Formal Language and Automata Theory

Chapter One

INTRODUCTION

Formal Language

- A formal language is an abstraction of the general characteristics of programming languages. A formal language consists of a set of symbols and some rules of formation by which these symbols can be combined into entities called sentences.

Con't

Formal Language

- Developed with strict rules.
- Predefined syntax and semantics.
- Precise
- Unambiguous

Disadvantage:

- Unfamiliar natation
- Initial learning effort

E.g. C++, Pascal,….

Natural language

- Rules come after the language
- Evolve and develop
- Highly flexible
- Quite powerful
- No special learning effort needed

Disadvantage:

- Vague
- Imprecise and ambiguous
- User and context dependent

E.g. Amharic, English, ….

Mathematical preliminaries and notations

- **Set**: is a collection of elements, without any structure other than membership.

  If m is an element of the set S, we write m ε S.

A set is specified by enclosing some description of its elements in curly braces. E.g. S={a,b,c}

Universal set *U* of all possible elements is, if *U* is specified, then $S=\{ x: x \varepsilon U, x \not{E} S\}$

The set with no element is called the **empty set** or **null set**. It is denoted by Ø.

Con’t

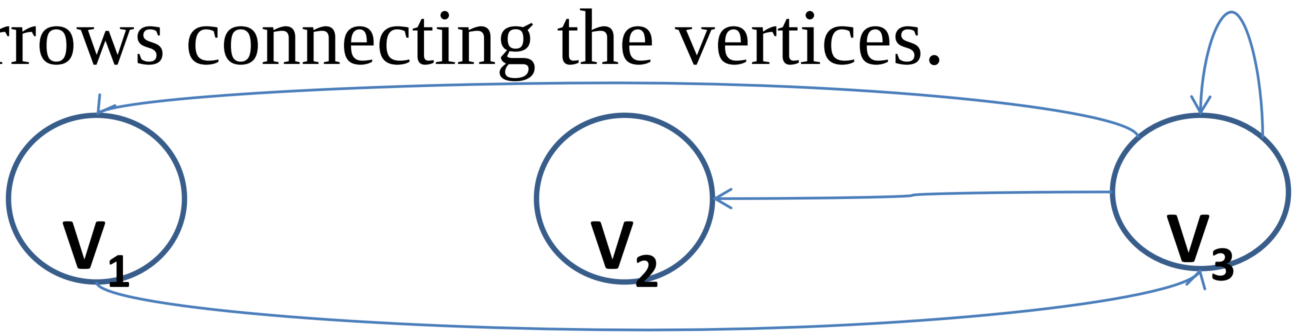
Operation on sets

- Let A and B be two sets and
  - A $\underline{c}$ B iff, for all x ε A, x ε B
  - A=B iff, A $\underline{c}$ B and B $\underline{c}$ A
  - A c B iff, A $\underline{c}$ B but A $\neq$ B
  - A u B={x: x ε A or x ε B}
  - A ∩ B={x: x ε A and x $\not E$ B}
  - A X B={ (x, y), x ε A and y ε B}

Con’t

Graphs and Trees

A graph is a construct consisting of two finite sets $V=\{ v_1,v_2, \ldots\ldots v_n\}$ of vertices and the set $E=\{e_1, e_2, \ldots\ldots e_m\}$ of edges. Each edge is a pair of vertices from V. e. g $e_i=(v_j, v_k\}$. The vertices are represented as circles and the edges as lines with arrows connecting the vertices.



Con't

- The degree of vertex *V* in graph is the number of edges with *V* as an end vertex.
- A path in a graph is an alternating sequence of vertices and edges of the form $v_1e_1v_2e_2\ldots e_{n-1}v_n$ and no edge or vertex is repeated in the sequence.

Con't

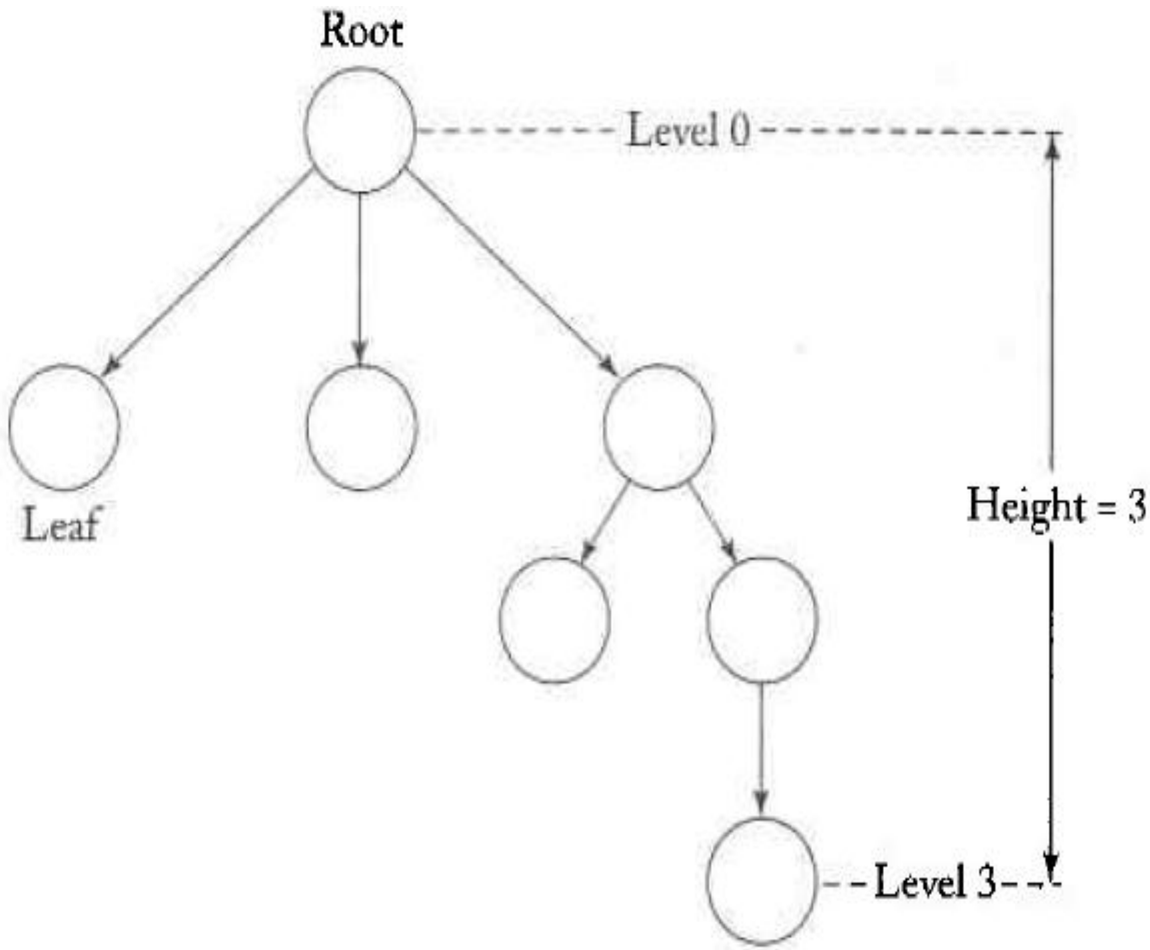
Tree

- Trees are a particular type of graph.
- A tree is a directed graph that has no cycles, and that has one distinct vertex called the **root**.
- There are some vertices without outgoing edges, these are called the **leaves** of the tree.
- If there is an edge from v_i to v_j, then v_i is said to be the parent of v_j, and v_j the child of v_i

Con’t

- A tree with n vertices have n-1 edges.
- A leaf in a tree is a vertex of degree one.
- The number of edges in the path is called the length of the path.
- The height of the tree is the length of the longest path from the root.
- A vertex v in a tree is at level k if there is a path of length k from the root to the vertex v.

Con’t



Alphabet and Strings

An alphabet is a finite set of symbols which are used to form words in a language. An example of an alphabet might be a set like $\{a, b\}$.

- **Example**: $\Sigma = \{a, b, c, d\}$ is an **alphabet set** where ‘a’, ‘b’, ‘c’, and ‘d’ are **alphabets**.

Con’t

- **String:**
  - *string* over E is some number of elements of E (possibly none) placed in order. So if $E = \{ a, b\}$ then strings over E can be a, ab, bbaa, abab and so on.
  - **Length of a String**

    It is the number of symbols present in a string. (Denoted by **|S|**). **E.g.**
    - If S=‘cabcad’, |S|= 6
    - If |S|= 0, it is called an **empty string** (Denoted by **λ** or **ε**)

Con't

Kleene Star

- **Definition**: The set **Σ*** is the infinite set of all possible strings of all possible lengths over **Σ** including **λ**.
- **Representation**: $\Sigma^* = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \ldots\ldots$
- **Example**: If $\Sigma = \{a, b\}$, $\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, \ldots\ldots\ldots\}$

Kleene Closure / Plus

- **Definition**: The set **Σ+** is the infinite set of all possible strings of all possible lengths over Σ excluding λ.
- **Representation**: $\Sigma^+ = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \ldots\ldots$

  $\Sigma^+ = \Sigma^* - \{ \lambda \}$
- **Example**: If $\Sigma = \{ a, b \}$, $\Sigma^+ = \{ a, b, aa, ab, ba, bb, \ldots\ldots\ldots \}$

Con’t

Transpose Operation

- for any string $x\ \varepsilon\ \Sigma^*$ *and* $a\ \varepsilon\ \Sigma$, $(xa)^T = a(x)^T$.
- A palindrome of even length can be obtained by the concatenation of a string and its transpose.

e.g. Let $\Sigma=\{0,1\}$

$Y= (01001)T$

$(xa)^T=(0100\ \ 1)^T$

$=1(0100)^T$

$=10(010)^T$

$=100(01)^T$

$=10010$

Language

- **Definition** : A language is a subset of Σ^* for some alphabet Σ. It can be finite or infinite.
- **Example** : If the language takes all possible strings of length 2 over $\Sigma = \{a, b\}$, then L = { ab, bb, ba, bb}

Grammars

- Grammars denote syntactical rules for conversation in natural languages. Linguistics have attempted to define grammars since the inception of natural languages like English, Amharic.
- Strings may be derived from other strings using the productions in a grammar.
- The set of all strings that can be derived from a grammar is said to be the language generated from that grammar. More on chapter 3& 5

Automata

- The term "Automata" is derived from the Greek word "αὐτόματα" which means "self-acting". An automaton (Automata in plural) is an abstract self-propelled computing device which follows a predetermined sequence of operations automatically.
- An automaton with a finite number of states is called a **Finite Automaton** (FA) or **Finite State Machine** (FSM).

Con’t

- Finite automata are useful model for many kinds of hardware and software.
  - Software for designing and checking the behavior of digital circuits.
  - The lexical analyzer of typical compiler.
  - Software for scanning large bodies of a text.
  - Software for verifying systems of all types with a finite distinct state. Such as communication protocols or protocols for secure information exchange.

Con't

- Finite Automata (Finite state machine)
  - We denote a FA by the 5-tuple(Q, E, q_0, δ, A) where Q is a set of states, E is an alphabet, q_0 is the starting state, δ is a transition function, and A is the set of accepting states.
- Finite Automaton can be classified into two types:
  - Deterministic Finite Automata (DFA)
  - Non-deterministic Finite Automata (NDFA)

Con't

Deterministic Finite Automaton (DFA)

- In DFA, for each input symbol, one can determine the state to which the machine will move. Hence, it is called **Deterministic Automaton.**
- As it has a finite number of states, the machine is called **Deterministic Finite Machine** or **Deterministic Finite Automaton**.

Con’t

Non-deterministic Finite Automaton (NDFA)

- In NDFA, for a particular input symbol, the machine can move to any combination of the states in the machine. In other words, the exact state to which the machine moves cannot be determined.
- Hence, it is called **Non-deterministic Automaton**. As it has finite number of states, the machine is called **Non-deterministic Finite Machine** or **Non-deterministic Finite Automaton**.