Introduction to Formal Language Theory

Course outline

Chapter 1: Basics

- Set theory
- Relations & functions
- Mathematical induction
- Graphs & trees
- Strings & languages

Chapter 2: Introduction to grammars

Chapter 3: Regular languages

- Regular grammar
- Automata
- Regular expressions

Chapter 4 :Context Free Languages

- Context free grammars
- Normal forms

Chapter 5: Push Down Automata (PDA)

- NPDA
- DPDA

Basics: outline

- Overview of languages: natural vs formal
- Review of set theory and relations
  - Set theory
  - Relations and functions
- Mathematical induction
- Graphs and trees
- Strings and languages

Overview of languages : natural Vs formal

- Language is a set of strings or sentences.
- Natural Languages
  - rules come after the language
  - evolve and develop
  - highly flexible
  - quite powerful
  - no special learning effort needed

  Disadvantages
  - vague
  - imprecise
  - ambiguous
  - user and context dependent
  - Ex. Amharic, English, French, …

Overview of languages: cont'd

- Formal Languages
  - developed with strict rules

    →predefined syntax and semantics
  - precise
  - unambiguous

  →can be processed by machines!

  Disadvantages
  - unfamiliar notation
  - initial learning effort
  - Ex. Programming languages: Pascal, C++, …

Overview of languages: cont'd

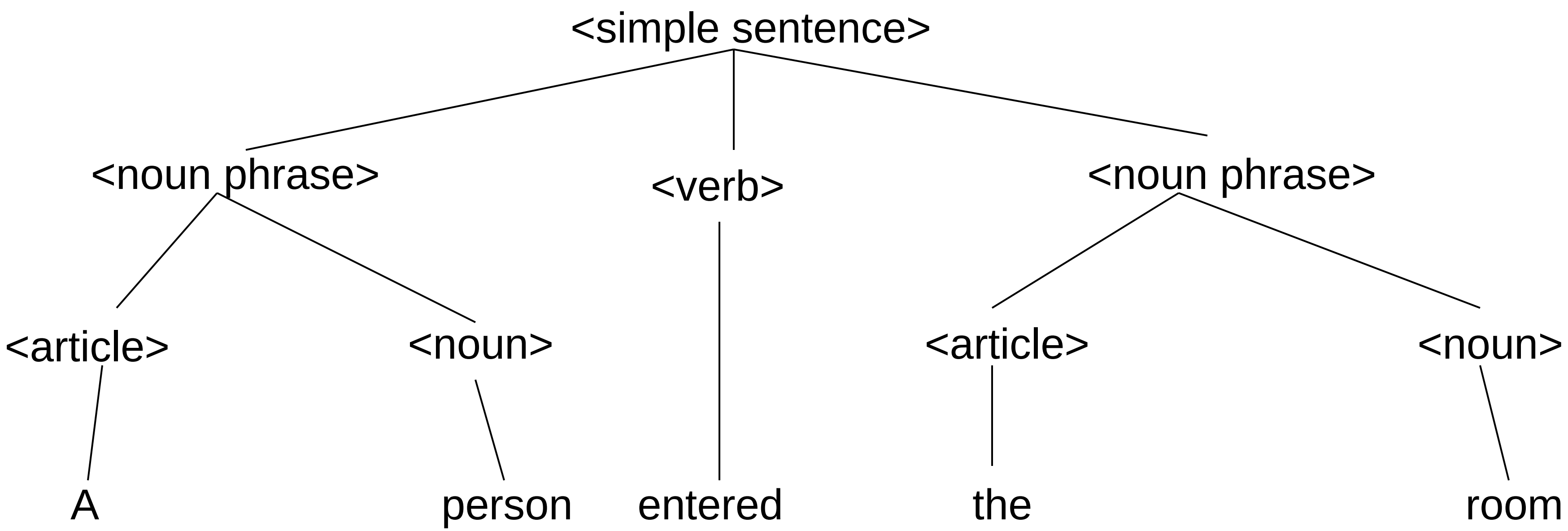
- Sentences: the basic building blocks of languages
- Sentence = Syntax + Semantics
- Grammar: the study of the structure of a sentence
- Ex:

<simple sentence> ::= <noun phrase><verb><noun phrase>

<noun phrase> ::= <article><noun>

→A person entered the room

Overview of languages: cont’d



Derivation tree for the simple sentence: A person entered the room.

Overview of languages: cont'd

- In Pascal (as well as in many other languages), for example, an identifier is specified as follows:

  <identifier> ::= <letter> | <letter> {<letter> | <digit>}*

  <letter> ::= a | b| c …

  <digit> ::= 0 | 1| 2 | … | 9

  Ex. a, x1, num, count1, …

Review of set theory and relations

- Sets
  - A well defined collection of objects (called members or elements)
  - Notation: a Є S → a is an element of the set S
- Operation on sets

  Let A and B be two sets and U the universal set
  - Subset: A C B
  - Proper subset: A c B
  - Equality: A = B
  - Union: A U B
  - Intersection: A ∩ B
  - Set difference: A \ B or A – B
  - Complement: A' or A bar
  - Cartesian product: A X B = {(a,b) | a Є A and b Є B}

    Note: (a,b) is called an **ordered pair**, and is different from (b,a)

Set theory and relations: cont'd

- Properties

  Let A, B, C be sets and U the universal set

  - Associative property: A U (B U C) = (A U B) U C
  - Commutative property: A U B = B U A
  - Demorgan's laws: (A U B)' = A' ∩ B', ...
  - Involution law: (A')' = A

  Definitions:

  - Let A be a set. The **cardinality of a set A** is a measure of the "number of elements of the **set**" and denoted by |A| or #(A).
  - The set of all subsets of a set A is called the power set of A, denoted by 2^{A}.

Set theory and relations: cont'd

Definition:

Let S be a set. A collection $\{A_1, A_2, \ldots, A_n\}$ of subsets of S is called a partition if $A_i \cap A_j = \emptyset$, $i \neq j$ and $S = A_1 \cup A_2 \cup \ldots \cup A_n$.

Ex. S = {1, 2, …, 10}

Let $A_1 = \{1, 3, 5, 7, 9\}$ and $A_2 = \{2, 4, 6, 8, 10\}$, then $\{A_1, A_2\} = \{\{1, 3, 5, 7, 9\}, \{2, 4, 6, 8, 10\}\}$ is a partition of S.

Q. Find other partitions of S

- Countability
  - A finite set is countable
  - If the elements of set A can be associated with $1^{st}, 2^{nd}, \ldots, i^{th}, \ldots$ elements of the set of Natural Numbers, then A is countable.

    Note: that in this case A may not be finite.

Ex.

1. N = {1, 2, …, i^{th}, …} is countable
2. Z = {…, -3, -2, -1, 0, 1, 2, 3, …} = {0, 1, -1, 2, -2, 3, -3, …} is countable
3. [0, 3] is uncountable (not countable)

Relations and functions

- Relations
  - <u>Definition</u>: A relation R is a set of ordered pairs of elements in S. (i.e is a subset of S X S)

    Notation: (x, y) Є R or x R y
  - Properties of relations
    - Let R be a relation on a set A, then
      a. R is **reflexive** if for all a Є A, a R a or (a, a) Є R
      b. R is **symmetric** if a R b => b R a
      c. R is **transitive** if a R b and b R c => a R c, for all a, b, c Є R
      d. R is an **equivalence relation** if (a), (b) and (c) above hold.
    - Let R be an equivalence relation on set A and let a Є A, then the equivalence class of a, denoted by [a], is defined as:

    [a] = {b ЄA | a R b}

Relations and functions: cont’d

Examples:

Check whether the following relations are reflexive, symmetric, and transitive

1. Let R be a relation in {1, 2, 3, 4, 5, 6} is given by {(1,2), (2, 3), (3, 4), (4, 4), (4, 5)}
2. Let R be a relation in {1, 2, 3, …, 10} defined as a R b if a divides b
3. Let R be defined on a set S such that aRb if a=b
4. Let R be defined on all people in Addis Ababa by aRb if a and b have the same date of birth.

Relations and functions: cont'd

- Functions
  - Definition: A function f from a set X to a set Y is a rule that associates to every element x in X a unique element in Y, which is denoted by f(x).
    - The element f(x) is called the image of x under f.
    - The function is denoted by f: X → Y
  - Functions can be defined in the following two ways:
    1. By giving the images of all elements of X

       Ex. f:{1, 2, 3, 4} → {2, 4, 6} can be defined by

       f(1) = 2, f(2) = 4, f(3) = 6, f(4) = 6
    2. By a computational rule which computes f(x) once x is given

       Ex. f:R → R can be defined by $f(x) = x^2 + 2x + 1$, x Є R (R = the set of all real numbers)

Relations and functions: cont'd

- Let $f: A \rightarrow B$ be a function
  1. f is an **into** function if $R_f \subseteq B$
  2. f is an **onto** function if $R_f = B$
  3. f is a **one-to-one** function
     if for x_1 & $x_2 \in A$, $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$
  4. f is bijective (one-to-one correspondence) if it satisfies (2) and (3) above.

  Ex. $f: Z \rightarrow Z$ is given by $f(x) = 2x$

  Show that f is one-to-one but not onto.

  - Definition: A set A is said to be countable iff there exists a function $f: A \rightarrow N$ such that f is bijective. (N=the set of natural numbers)

Mathematical induction

- Let P_n be a proposition that depends on $n \in Z^+$. Then P_n is true for all +ve n provided that:
  i. P_i is true
  ii. If P_k is true, so is P_{k+1}, for some $k \in Z^+$.

Three steps:

1. Base case: verify that P_1 holds
2. Inductive hypothesis: assume that P_k holds, for some $k \in Z^+$
3. Inductive step: show that P_{k+1} holds

Ex. Show that $1+2+\ldots+n = n(n+1)/2$, for all $n \in Z^+$.

Mathematical induction: cont'd

<u>Solution</u>:

Let P_n: $1+2+\ldots+n = n(n+1)/2$

<u>Step1</u>: for n = 1, P_1 holds

<u>Step2</u>: for some $k \in Z^+$, assume P_k is true

i.e. P_k: $1+2+\ldots+k = k(k+1)/2$

<u>Step3</u>: WTS P_{k+1} is true

P_{k+1} : $1+2+\ldots+k+(k+1) = (k+1)(k+2)/2$

: $P_k + (k+1) = (k+1)(k+2)/2$

: $k(k+1)/2 + (k+1) = (k+1)(k+2)/2$

: $[k(k+1) + 2(k+2)]/2 = (k+1)(k+2)/2$

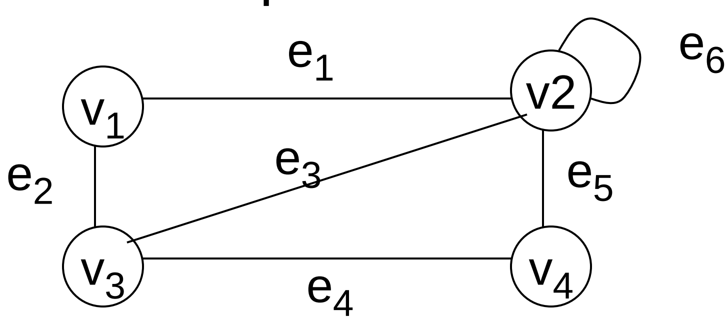
: $(k+1)(k+2)/2 = (k+1)(k+2)/2$

Therefore, Pn holds for all n $\in$Z+

Ex. Show that Pn = $\sum(i=1,n)(i^2) = (n+1)(n)(2n+1)/6$ for all n

Graphs and trees

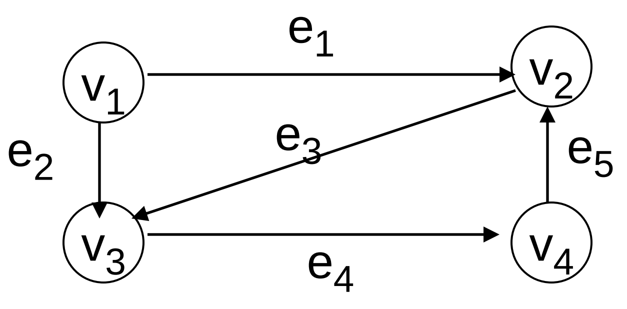
- Graphs
  - <u>Definition</u>: A graph (undirected graph) consists of:
    a. A non-empty set v called the set of vertices,
    b. A set E called the set of edges, and
    c. A map Φ (phi) which assigns to every edge a unique unordered pair of vertices



$e_1 = \{v_1, v_2\}$
$e_2 = \{v_1, v_3\}$
…
$e_6 = \{v_2, v_2\}$ (a self loop)

Graphs and trees: cont'd

- Definition: A directed graph (digraph) consists of:
  a. A non-empty set v called the set of vertices,
  b. A set E called the set of edges, and
  c. A map Φ (phi) which assigns to every edge a unique ordered pair of vertices



$e_1 = (v_1, v_2)$
v_1 : a predecessor of v_2
v_2 : a successor of v_1

Graphs and trees: cont'd

- Definition: The degree of a vertex v in a graph (directed or undirected) is the number of edges with v as an end vertex.

  Note: that a self loop is counted twice when calculating the degree of a vertex.

  Ex. In the previous graph, $deg(v_1)$ = ? $deg(v_2)$ = ?

- Definition: A path in a graph (directed or undirected) is an alternating sequence of vertices and edges of the form $v_1e_1v_2e_2\ldots e_{n-1}v_n$, beginning and ending with vertices such that e_i has v_i and v_{i+1} as its end vertices and no edge or vertex is repeated in the sequence.

  The path is said to be from v_1 to v_n.

Ex. In the previous graph, $v_1e_1v_2e_3v_3e_4v_4$ is a path from v_1 to v_4.
Note: that a path may be directed (if all the edges in the path have the same direction.)

Graphs and trees: cont'd

- Definition: A graph (directed or undirected) is **connected** if there is a path between every pair of vertices.

  Q. Are the previous two graphs connected?

- Definition: A circuit in a graph is an alternating sequence $v_1e_1v_2e_2\ldots e_{n-1}v_1$ of vertices and edges starting and ending with the same vertex such that e_i has v_i and v_{i+1} as end vertices and no edge or vertex other than v_1 is repeated.

  Ex. $V_2e_3v_3e_4v_4e_5v_2$ is a circuit in the previous graph

Graphs and trees: cont'd

- Trees
  - Definition: A graph (directed or undirected) is called a tree if it is connected and has no circuits.

    Q. Are the previous two graphs trees?
  - Properties of trees:
    - In a tree there is one and only one path between every pair of vertices (nodes)
    - A tree with n vertices has n-1 edges
    - A **leaf** in a tree can be defined as a vertex of degree one
    - Vertices other than leaves are called internal vertices

Graphs and trees: cont'd

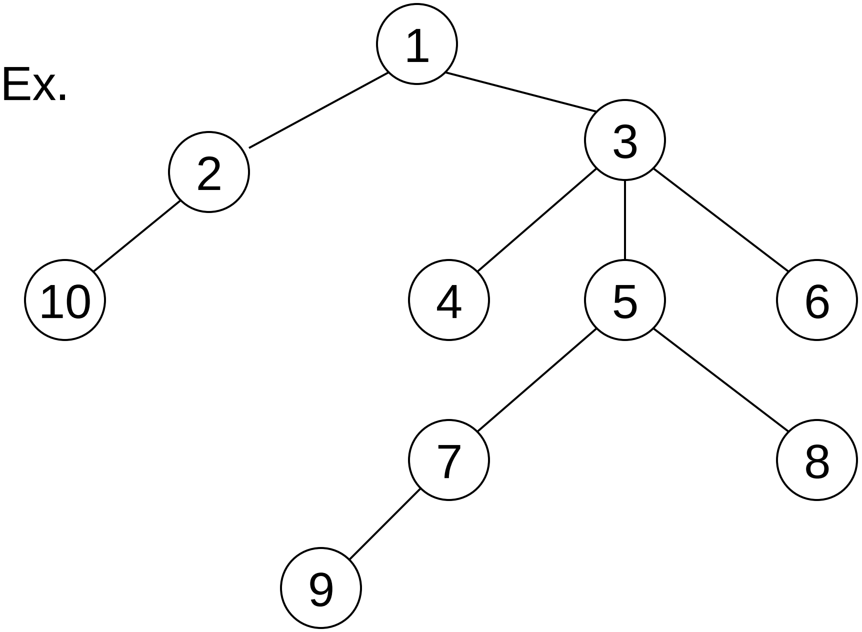
- Definition: An ordered directed tree is a digraph satisfying the following conditions:
  - There is one vertex called the **root** of the tree which is distinguished from all other vertices and the root has no predecessors.
  - There is a directed path from the root to every other vertex.
  - Every vertex except the root has exactly one predecessor.

    (For the sake of simplicity, we refer to ordered directed trees as simply trees.)
- The number of edges in a path is called the **length** of the path.
- The height of a 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.

  Q. what is the maximum possible level in a tree?
- There are several types of trees: binary, balanced binary, binary search tree, heap, general tree, …

Graphs and trees: cont'd

Ex.



1. List the leaves.
2. List the internal nodes.
3. What is the length of the path from 1 to 9?
4. What is the height of the tree?

- Note: a path from vertex (node) n_1 to node n_k can be simply expressed as the sequence of nodes n_i, i=1,…,k such that n_i is the parent (predecessor) of n_{i+1} (1<= I <=k)

Strings and languages

- Strings
  - An alphabet, ∑, is a set of finite symbols.
  - A string over an alphabet ∑ is a sequence of symbols from ∑.
  - An empty string is a string without symbols, and is denoted by λ.
  - Let w be a string, then its length, denoted by /w/, is the number of symbols of w.

  Ex. Let ∑ = {0, 1}, the following are some strings over ∑

  w = λ, /w/ = 0; w = 01, /w/ = 2; w = 010110, /w/ = 6
  - Given an alphabet ∑, Σ^* denotes the set of all strings (including λ) over ∑.
  - $\Sigma^+ = \Sigma^* - \{\lambda\}$

  Ex. $\Sigma = \{0, 1\} \Rightarrow \Sigma^* = \{\lambda, 0, 1, 01, 00, 11, 111, 0101, 0000, \ldots\}$
  - Σ^i is a set of strings of length i, i = 0, 1, 2, …
  - Let $x \in \Sigma^*$ and /x/ = n, then $x = a_1a_2\ldots a_n$, $a_i \in \Sigma$

Strings and languages: cont'd

- Operations on strings
  - Concatenation operation
    - Let x, y $\in \Sigma^*$ and $/x/ = n$ and $/y/ = m$. Then xy, concatenation of x and y, $= a_1a_2\ldots a_nb_1b_2\ldots b_m$, a_i, $b_i \in \Sigma$
    - The set Σ^* has an identity element λ with respect to the binary operation of concatenation.

      Ex. $x \in \Sigma^*$, $x\lambda = \lambda x = x$
    - Σ^* has left and right cancellation

      For x, y, z $\in \Sigma^*$,

      $zx = zy \Rightarrow x = y$ (left cancellation)

      $xz = yz \Rightarrow x = y$ (right cancellation)
    - For x, y $\in \Sigma^*$, we have $/xy/ = /x/ + /y/$

Strings and languages: cont'd

- Transpose operation
  - For any x in $\sum^*$ and a in $\sum$, $(xa)^T = a(x)^T$

    Ex. $(aaabab)^T$ = babaaa
  - A **palindrome** of even length can be obtained by the concatenation of a string and its transpose.
  - A **prefix** of a string is a substring of leading symbols of that string.

    w is a prefix of y if there exists y' in $\sum^*$ such that y=wy'

    Ex. y = 123, list all prefixes of y.
  - A **suffix** of a string is a substring of trailing symbols of that string.

    w is a prefix of y if there exists y' in $\sum^*$ such that y=y'w

    Ex. y = 123, list all suffixes of y.

Strings and languages: cont'd

- A **terminal** symbol is a unique indivisible object used in the generation of strings.
- A **nonterminal** symbol is a unique object but divisible, used in the generation of strings.

  Ex. In English, a, b, A, B, etc are terminals and the words *boy*, *cat*, *dog*, … are nonterminals.

  In programming languages, a, A, :, ;, =, if, then, … are terminals

Strings and languages: cont'd

- Languages
  - Definition: A language, L, is a set (collection) of strings over a given alphabet, ∑.
    - A string in L is called a sentence or word.

    Ex. $\sum = \{0, 1\}$, $\sum^* = \{\lambda, 0, 1, 01, 00, 11, \ldots\}$

    $L_1 = \{\lambda\}$, $L_2 = \{0, 1, 01\}$ over ∑

    $L_3 = \{a^n \mid n >= 0\}$ over $\sum = \{a\}$

  - Let L_1 , L_2 be languages over ∑, then
    - $L_1L_2 = \{xy \mid x \in L_1, y \in L_2\}$
    - $L\{\lambda\} = \{\lambda\}L = L$, for any language L
    - $L^0 = \{\lambda\}$
    - $L^1 = L$
    - $L^2 = LL \quad \equiv \{xx \mid x \in L\}$
    - …
    - $L^i = L^iL^{i-1}$, for $i>=2$
    - $L^* = U(i=0,\infty)(L^i)$