Formal Languages and Automata Theory

March 2023

By Ashenafi C.

Chapter 1

Mathematical Preliminaries and Notation

Introduction

- An **automaton** is a construct that possesses all the indispensable features of a digital computer.
  - It accepts input, produces output, may have some temporary storage, and can make decisions in transforming the input into the output.
- A **formal language** is an abstraction of the general characteristics of programming languages.
  - It consists of a **set of symbols** and **some rules** of formation by which these symbols can be combined into entities called **sentences**.
  - It is the set of all sentences permitted by the rules of formation.

Sets

Definition 1. A set is a group of objects. The objects in a set are called the **elements**, or **members**, of the set.

Example 1

The set of positive integers less than 100 can be denoted as $\{1,2,3,\ldots,99\}$.

Example 2

A set can also consists of seemingly **unrelated** elements: $\{a, 2, Fred, New\ Jersey\}$.

Definition 2. Two sets are equal if and only if they have the **same** elements.

Example 3

Set $\{1,3,3,3,5,5,5,5\}$ is the same as set $\{1,3,5\}$.

- A set can be described by using a set builder notation.

Example 3

$$O = \{x \mid x \text{ is an odd positive integer less than } 10\}$$

Example 4

$$\mathrm{N} = \{x \mid \text{natural numbers}\} = \{0,1,2,3,\ldots\}$$

$$\mathrm{Z} = \{x \mid \text{integers}\} = \{\ldots,-2,-1,0,1,\ldots\}$$

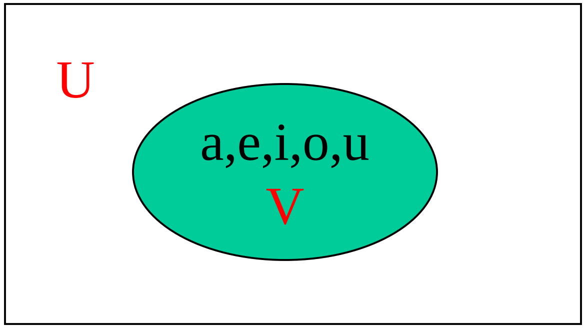
$$\mathrm{Z} = \{x \mid \text{positive integers}\} = \{1,2,3,\ldots\}$$

$$\mathrm{R} = \{x \mid \text{real numbers}\}$$

- A set can be described by using a Venn diagram.

Example 5

Draw a Venn diagram that presents V, the set of vowels in English alphabet.



- The set that has no elements is called **empty set**, denoted by ϕ.

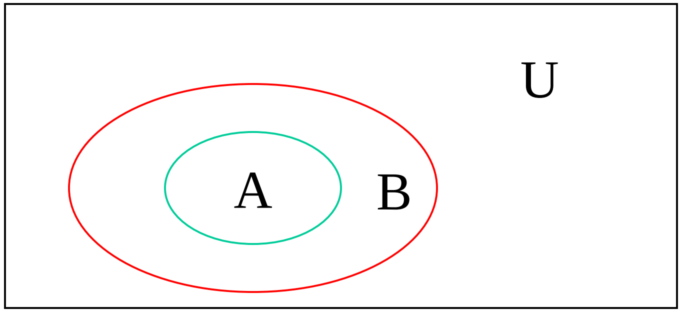
Definition 3. The set A is said to be a **subset** of B if and only if every element of A is also an element of B. We use the notation $A \subseteq B$ to indicate that A is a subset of the set B.

- $A \subseteq B$ if and only if $\forall x (x \in A \rightarrow x \in B)$
- For any set P, $\phi \subseteq P$ and $P \subseteq P$.
- If $A \subseteq B$ and $B \subseteq A$, then $A = B$.

Example 6

$\{\phi, \{a\}, \{b\}, \{a, b\}\} = \{x \mid x \text{ is a subset of the set } \{a, b\}\}$.

- If a set A is a subset of set B but that $A \neq B$, A is called to be a proper subset of B, denoted as $A \subset B$.



Definition 4. The power set of a set S is the set of all subsets of S, denoted by P(S) or 2^S.

Example 7

What is the power set of the set $S = \{0,1,2\}$?

Solution: $P(\{0,1,2\})$ (or 2^S) $= \{\phi, \{0\}, \{1\}, \{2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0,1,2\}\}$.

- Set Operations:

Complement $\overline{A} = \{x \mid x \in U \text{ and } x \notin A\}$, where set U is called "universal" that contains all the elements we might ever consider}

Union $A \bigcup B = \{x \mid x \in A \text{ or } x \in B\}$

Intersection $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$

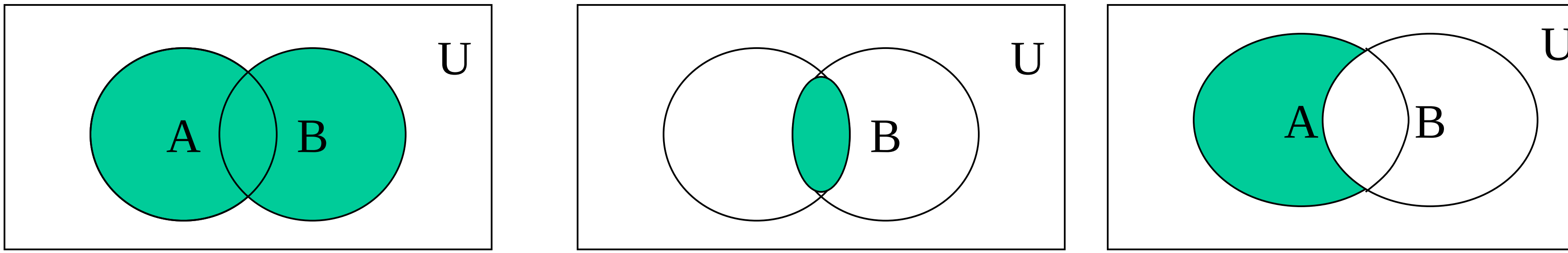
Difference $A - B = \{x \mid x \in A \text{ and } x \notin B\}$

Example 8

$\{1,3,5\} \cup \{1,2,3\} = \{1,2,3,5\}.$

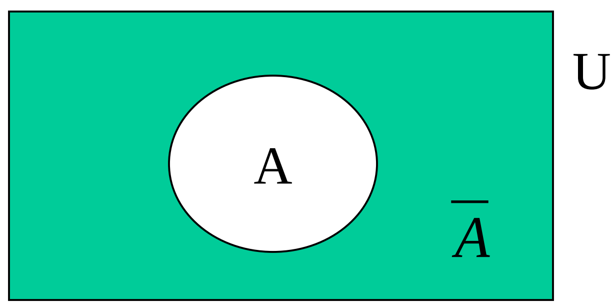
$\{1,3,5\} \cap \{1,2,3\} = \{1,3\}.$

$\{1,3,5\} - \{1,2,3\} = \{5\}.$



Example 9

Let $A = \{a, e, i, o, u\}$ and the universal set is the set of the letters of the English alphabet. Then $\overline{A} = \{b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z\}.$



- **Set Identities**

Table 1 Set Identities	
Identity	*Name*
$A \cup \phi = A$ $A \cap U = A$	Identity laws
$A \cup U = U$ $A \cap \phi = \phi$	Domination laws
$A \cup A = A$ $A \cap A = A$	Idempotent laws
$\overline{(\overline{A})} = A$	Complementation laws
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative laws
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative laws
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Distributive laws
$\overline{A \cup B} = \overline{A} \cap \overline{B}$ $\overline{A \cap B} = \overline{A} \cup \overline{B}$	*De Morgan's* law

- Cartisian product of A and B, denoted by $A \times B$, is the set of all ordered pairs (a,b), where $a \in A$ and $b \in B$. Hence, $A \times B = \{(a,b) \mid a \in A \land b \in B\}$.

$\boxed{\text{Example 10}}$

Let $A = \{1,2\}$ and $B = \{a,b,c\}$.

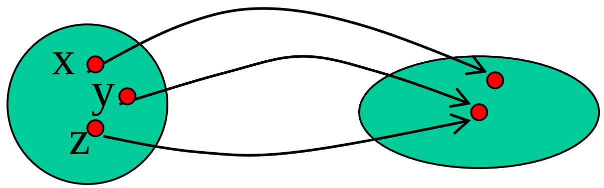
$A \times B = \{(1,a),(1,b),(1,c),(2,a),(2,b),(2,c)\}$

$B \times A = \{(a,1),(a,2),(b,1),(b,2),(c,1),(c,2)\}$

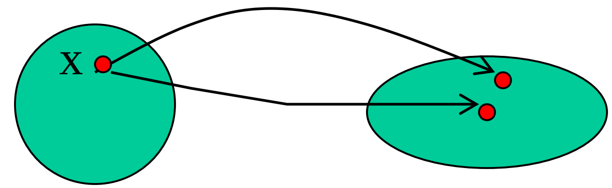
$A \times B \neq B \times A$

Functions and Relations

Definition 1. Let A and B be sets. A function f from A to B is an assignment of exactly one element of B to each element of A. We write $f(a)=b$ if b is the unique element of B assigned by the function f to the element a of A. If f is a function from A to B, we write $f : A \to B$.



A function

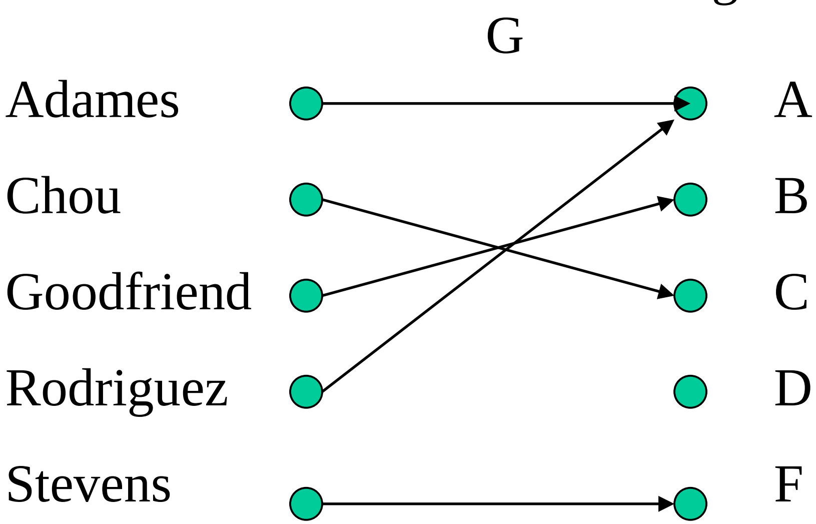


Not a function

Example 1

Let set A={Adams, Chou, Goodfriend, Rodriguez, Stevens} and B={A,B,C,D,F}.

Let G be the function that assigns a grade to a student in our theory of computation.



The domain of G is the set A={Adams, Chou, Goodfriend, Rodriguez, Stevens}, and the range of G is the set {A,B,C,F}.

- Considering the function whose **domain** and **range** are in the set of integers. We are often interested only in the behavior of these functions as their arguments become very large.

<u>Definition 2</u>

Let $f(n)$ and $g(n)$ be functions whose domain is a subset of the positive integers.

(1) If there exists a positive constant c such that for all n

if $|f(n)| \le c|g(n)|$, we say that f has order at most g, denoted as $f(n) = O(g(n))$,

if $|f(n)| \ge c|g(n)|$, then f has order at least g, denoted as $f(n) = \Omega(g(n)$

(2) If there exist constants c_1 and c_2, such that $c_1|g(n)| \le |f(n)| \le c_2|g(n)|$, f and g have the same order of magnitude, denoted as $f(n) = \Theta(g(n))$.

Example 2

Let $f(n) = 2n^2 + 3n$, $g(n) = n^3$, $h(n) = 10n^2 + 100$

Then $f(n) = O(g(n))$, $g(n) = \Omega(h(n))$, $f(n) = \Theta(n)$

Definition 3

Let A and B be the sets. A relation R from A to B is a subset of $A \times B$.

From the definition, relation R is a set of pairs.

If $(a,b) \in R$, we say a has a relation R with b, denoted as aRb.

- Functions can be considered as relations, but relations are more general than functions.

Example 3

Let A={1,2,3,4}. R={(1,2), (2,3),(3,1),(4,4)} is a relation from A to A, and it is also a function from A to A.

Example 4

R={(1,2), (2,3),(1,3),(1,4)} is a relation but **not** a function.

Example 5

- Let A be the set of students in the AASTU. Let B the set of courses. The set R that consists of those pairs (a,b), where a is a student enrolled in course b, is a relation from A to B.

Definition 4

R is an equivalence relation if for any pair *(x,y)* of *R*

xRx for all *x* (reflexivity)

If *xRy* then *yRx* (symmetry)

If *xRy* and *yRz*, then *xRz*. (transitivity)

We usually use $\equiv$ to denote equivalence relation.

Example 6

Let *I* be the integer set and let *R* be a relation from I to I,

where *(x,y)* $\in R$ if and only if $x \bmod 3 = y \bmod 3$.

Then $2R5$, $12R0$, and $0R36$.

It is an equivalence relation, as it satisfies reflexity, symmetry, and transitivity.

Example 7

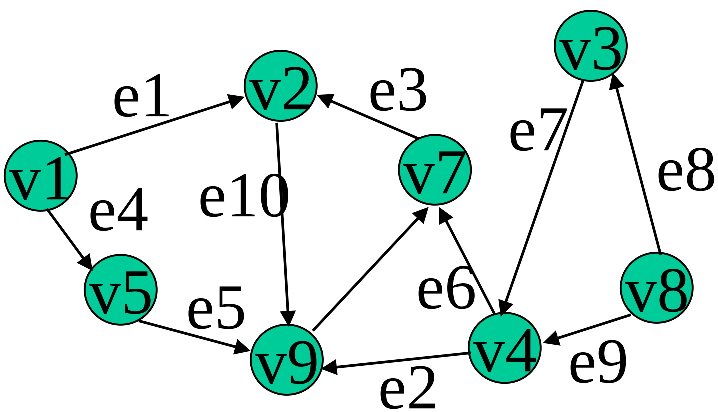
'=' on the set of integers is an equivalence relation.

Graphs and Trees

Definition1

A graph is a construct consisting of two sets, denoted as $G = (V,E)$,
where $V = \{v_1, v_2, \dots, v_n\}$ is a set of vertices and $E = \{e_1, e_2, \dots, e_m\}$ is a set of edges.
Each edge is a pair from V.

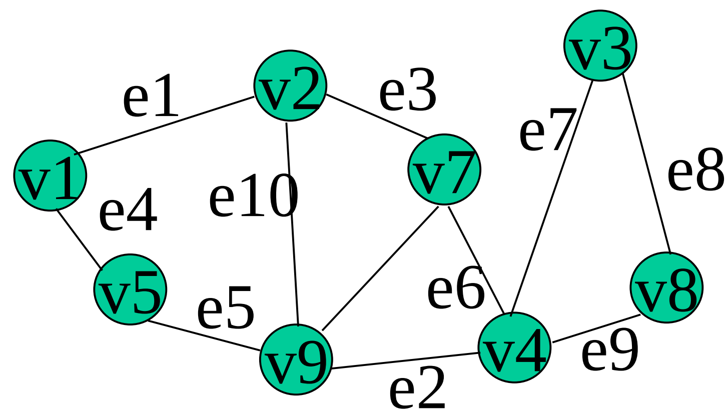
Example 1



A directed graph

(digraph)

Example 2



An undirected graph

Definition 2

(1) A sequence of edges $(v_i, v_j), (v_j, v_k), \ldots, (v_m, v_n)$ is a *walk.*

(2) The *length* of a walk is the total number of edges traversed in going from the intial vertex to the final one.

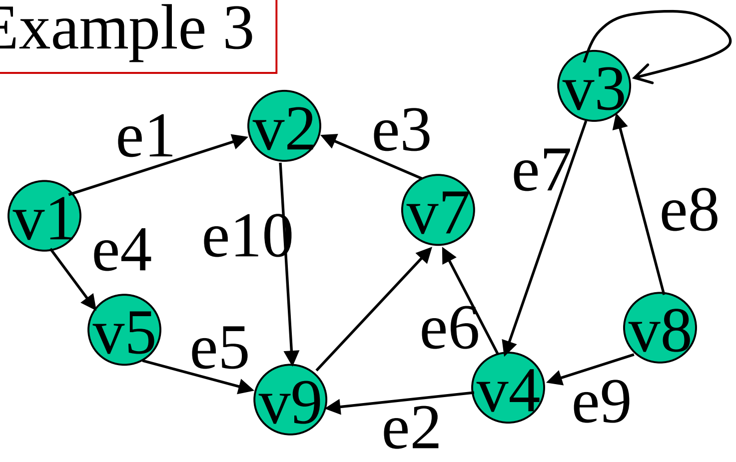
(3) A walk wit h no repeated is said to be a *path*.

(4) A path is *simple* if no vertex is repeated.

(5) A walk from v_i to itself with no repeated edges is called as a *cycle* with base v_i.

(6) An edge from a vertex to itself is called a *loop*.

Example 3



walk : $(v1, v2), (v2, v9), (v9, v7), (v7, v2), (v2, v9)$

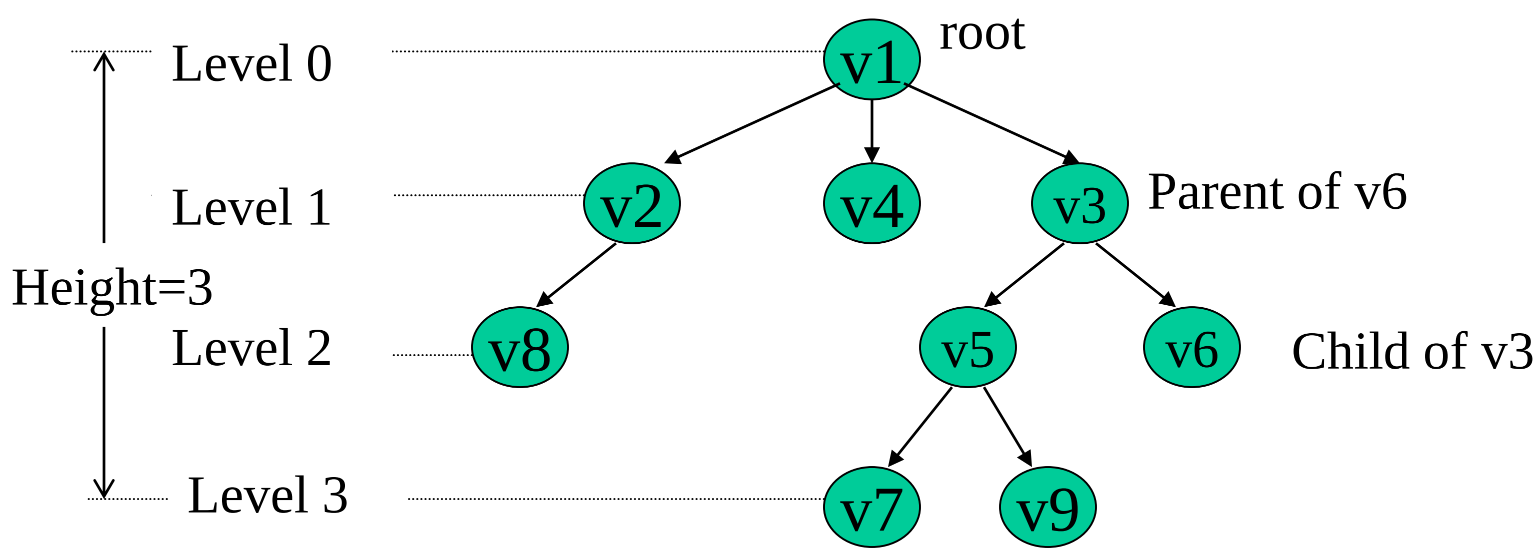
path : $(v1, v2), (v2, v9), (v9, v7), (v7, v2)$

cycle: $(v2, v9), (v9, v7), (v7, v2)$

loop: $(v3, v3)$

Definition 3

A *tree* is a directed graph that has no cycles. There is a one distinct vertex in tree, called the *root*.



A Proof Technique - Mathematical Induction

Definition1

Mathematical induction is a method we use to prove a sequence of statements $P_1, P_2, \ldots$ to be true.

(i) Basis: Proving that for some $k = 1$, P_1 is true.

(ii) Inductive Assumption: Supposing for any $n \geq k$, P_n is true.

(iii) Inductive Step: Proving that P_{n+1} is true.

Example 1

A binary tree (no parent has more than two chidren) of height n has at most 2^n leaves.

Proof: We use $l(n)$ to denote the maximum number of leaves in a binary tree of height n.

Basis: Clealy $l(0) = 1 = 2^0$.

Inductive Assumption: $l(i) \leq 2^i$, for $i = 0,1,\ldots,n$.

Inductive Step: To get a binary tree of height $n+1$ from one of height n, we can create, at most, two leaves in place of each previous one.

Therefore, $l(n+1) = 2l(n)$.

Using the inductive assumption, we get $l(n+1) \leq 2 \times l(n) = 2^{n+1}$.

Example 2

Show that $S_n = \sum_{i=0}^{n} i = \frac{n(n+1)}{2}.$

Proof:

Basis: $S_1 = \sum_{i=0}^{1} i = 1 = \frac{1(1+1)}{2}$

Inductive Assumption: Assuming that $S_n = \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$ for all $n \geq 1$.

Inductive Step: $S_{n+1} = \sum_{i=0}^{n+1} i = S_n + n + 1 = \frac{n(n+1)}{2} + n + 1 = \frac{(n+1)(n+2)}{2}.$

Strings and Languages

- **Alphabet**: Finite, nonempty set of symbols
  - Examples:
    - $\Sigma = \{0, 1\}$: binary alphabet
    - $\Sigma = \{a, b, c, \ldots, z\}$: the set of all lower case letters
    - The set of all ASCII characters

- **String**: Finite sequence of symbols from an alphabet Σ
  - Examples:
    - 01101 where $\Sigma = \{0, 1\}$
    - *abracadabra* where $\Sigma = \{a, b, c, \ldots, z\}$

Strings and Languages …

- **Empty String**: The string with **zero** occurrences of symbols from Σ and is denoted $\boldsymbol{\varepsilon}$ or λ
- **Length of String**: Number of symbols in the string
  - The length of a string w is usually written $|w|$
  - $|1010| = 4$
  - $|\boldsymbol{\varepsilon}| = 0$
  - $|uv| = |u| + |v|$
- **Reverse** : w^R
  - If w = abc, w^R = cba

Strings and Languages …

- **Concatenation**: if ***x*** and ***y*** are strings, then ***xy*** is the string obtained by placing a copy of ***y*** immediately after a copy of ***x***
  - $x = a_1a_2 \ldots a_i,\ y = b_1b_2 \ldots b_j$
  - $xy = a_1a_2 \ldots a_ib_1b_2 \ldots b_j$
  - Example: x = 01101, y = 110, xy = 01101110
  - $x\varepsilon = \varepsilon x = x$

Strings and Languages …

- **Power of an Alphabet**: Σ^k = the set of strings of length k with symbols from Σ
  - Example: $\Sigma = \{0, 1\}$
  - $\Sigma^1 = \Sigma = \{0, 1\}$
  - $\Sigma^2 = \{00, 01, 10, 11\}$
  - $\Sigma^0 = \{\varepsilon\}$

- **Question**: How many strings are there in Σ^3?

- The set of all strings over Σ is denoted Σ^*
  - $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$
- Also
  - $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$
  - $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$
  - $\Sigma^+ = \Sigma^* - \{\varepsilon\}$

Strings and Languages…

- **Substring**: any string of consecutive characters in some string *w*
  - If *w* = abc
  - □ ε, a, ab, abc are substrings of *w*

- **Prefix** and **suffix**:
  - if w = vu
  - v is a prefix of w
  - u is a suffix of w
  - Example
    - If w = abc
    - a, ab , abc are prefixes of w
    - c, bc, abc are suffixes of w

Strings and Languages…

- Suppose: S is the string banana
  - **Prefix : ban, banana**
  - **Suffix : ana, banana**
  - **Substring : nan, ban, ana, banana**

Strings and Languages…

- **Language**: set of strings chosen from some alphabet.
- A language is a subset of Σ^*
  - Example of languages:
    - The set of valid Arabic words
    - The set of strings consisting of *n* **0**'s followed by *n* **1**'s
      - $\{\varepsilon, 01, 0011, 000111, \ldots\}$
    - The set of strings with equal number of **0**'s and **1**'s
      - $\{\varepsilon, 01, 10, 0011, 0101, 1010, 1001, 1100, \ldots\}$
- **Empty language**: $\varnothing = \{\ \}$
- The language $\{\varepsilon\}$ consisting of the empty string.
- Note: $\varnothing \neq \{\varepsilon\}$

Strings and Languages …

- Can concatenate languages
  - $L_1L_2 = \{xy \mid x \in L_1, y \in L_2\}$
  - L^n = L concatenated with itself n times
  - $L^0 = \{\varepsilon\}$; $L^1 = L$
- Star-closure
  - $L^* = L^0 \cup L^1 \cup L^2 \cup \ \ldots$
  - $L^+ = L^* - L^0$
- Languages can be finite or infinite
  - $L = \{a, aba, bba\}$
  - $L = \{a^n \mid n > 0\}$

Strings and Languages …

OPERATION	DEFINITION
union **of L and M written $L \cup M$**	$L \cup M = \{s \mid s \text{ is in } L \text{ or } s \text{ is in } M\}$
concatenation **of L and M written LM**	$LM = \{st \mid s \text{ is in } L \text{ and } t \text{ is in } M\}$
Kleene closure **of L written L***	$L^* = \cup L^i$, $i=0,.., \infty$ L* denotes "zero or more concatenations of " L
positive closure **of L written L^+**	$L^+ = \cup L^i$, $i=1,.., \infty$ L^+ denotes "one or more concatenations of " L $L^+ = LL^*$

Strings and Languages …

$L = \{A, B, \ldots, Z, a, b, \ldots z\}$ $D = \{1, 2, \ldots, 9\}$

- $L \cup D$ = the set of letters and digits
- LD = all strings consisting of a letter followed by a digit
- L^2 = the set of all two-letter strings
- $L^4 = L^2\ L^2$ = the set of all four-letter strings
- L^* = { All possible strings of L plus ε }, $L^+ = L^* - \varepsilon$
- D^+ = set of strings of one or more digits
- $L(L \cup D)$ = set of all strings consisting of a letter followed by a a letter or a digit
- $L(L \cup D)^*$ = set of all strings consisting of letters and digits beginning with a letter

Strings and Languages …

- The language **L** consists of strings over **{a,b}** in which **each** string begins with **a** should have an **even** length.

  - aa, ab $\in$ L
  - aaaa,aaab,aaba,aabb,abaa,abab,abba,abbb $\in$ L

  - baa $\notin$ L
  - a $\notin$ L

Strings and Languages …

- The language **L** consists of strings over **{a,b}** in which each occurring of **b** is **immediately preceded** by an **a.**
  - $\varepsilon \in L$
  - $a \in L$
  - $abaab \in L$

  - $bb \notin L$
  - $bab \notin L$
  - $abb \notin L$

Strings and Languages …

- Let $X = \{a,b,c\}$ and $Y = \{abb, ba\}$. Then
  - $XY = \{aabb, babb, cabb, aba, bba, cba\}$
  - $X^0 = \{\varepsilon\}$
  - $X^1 = X = \{a,b,c\}$
  - $X^2 = XX =$
    - {aa,ab,ac,ba,bb,bc,ca,cb,cc}
  - $X^3 = XXX =$
    - {aaa,aab,aac,aba,abb,abc,aca,acb,acc,baa,bab,bac,bba,bbb,bbc,bca,bcb,bcc,caa,cab,cac,cba,cbb,cbc,cca,ccb,ccc}

Strings and Languages …

- The language $L = \{a,b\}^*\{bb\}\{a,b\}^* = \Sigma^*\{bb\}\Sigma^*$ consists of the strings over {a,b} that contain the substring **bb.**
  - $bb \in L$
  - $abb \in L$
  - $bbb \in L$
  - $aabb \in L$
  - $bbaaa \in L$
  - $bbabba \in L$
  - $abab \notin L$
  - $bab \notin L$
  - $b \notin L$

Strings and Languages …

- Let **L** be the language that consists of all strings that begin with **aa** or end with **bb**
  - $L_1 = \{aa\}\{a,b\}^*$
  - $L_2 = \{a,b\}^*\{bb\}$
  - $L = L_1 \cup L_2 = \{aa\}\{a,b\}^* \cup \{a,b\}^*\{bb\}$
  - $bb \in L$
  - $abb \in L$
  - $bbb \in L$
  - $aabb \in L$
  - $bbaaa \notin L$
  - $bbabba \notin L$
  - $abab \notin L$
  - $bab \notin L$
  - $ba \notin L$

Strings and Languages …

- Let $\mathbf{L_1}$ = {bb} and $\mathbf{L_2}$ = {ε, bb, bbbb} over **b**

- The languages L_1^* and L_2^* both contain precisely the strings consisting of an even number of **b**'s.

- ε , with length zero, is an element of L_1^* and L_2^*

Strings and Languages … Exercise

- What is the language of all even-length strings over **{a,b}?**

- What is the language of all odd-length strings over **{a,b}?**

Grammars

Let us see a grammar for English. Typically, we are told "a sentence can

Consist of a noun phrase followed by a predicate". We can write this grammar as follows:

variables

start variables

$$\langle \boxed{sentence} \rangle \to \langle noun\_ phrase \rangle \langle predicate \rangle$$

Furthermore, we have

$$\langle noun\_ phrase \rangle \to \langle \boxed{article} \rangle \langle \boxed{noun} \rangle,$$

productions

$$\langle predicate \rangle \to \langle \boxed{verb} \rangle$$

terminal symbols

From this grammar, we can produce the sentence like "a boy runs".

Definition1

A grammar G is defined as a quadruple

$$G = (V,T,S,P),$$

where V is a finite set of objects called variables,

T is a finite set of objects called terminal symbols,

$S \in V$ is a special symbol called the start variable,

P is a finite set of productions.

We assume V and T are nonempty and disjoint

Grammars…

Production rules are the heart of a grammar. We let them be of the form $x \to y$, where $x \in (V \cup T)^+$ and $y \in (V \cup T)^*$.

From a string $w = uxv$ and a production rule $x \to y$ we can obtain a new string $z = uyv$.
This is said that w derives z, denoted as $w \Rightarrow z$.

If $w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n$, we say that w_1 derives w_n and write as $w_1 \overset{*}{\Rightarrow} w_n$.

The $*$ means that an unspecified number of steps (including zero).
Therefore, $w \overset{*}{\Rightarrow} w$.

Definition1.2

Let $G=(V,T,S,P)$ be a grammer. Then the set

$$L(G)=\{w\in T^*:\ S\overset{*}{\Rightarrow} w\}$$

is the language generated by G.

If $w\in L(G)$, then the sequence

$$S\Rightarrow w_1\Rightarrow w_2\Rightarrow\cdots\Rightarrow w_n\Rightarrow w$$

is a derivation of the sentence w.

Example 1

Consider the grammar

$$G=(\{S\},\{a,b\},S,P),$$

with P given by

$$S\rightarrow aSb$$

$$S\rightarrow\lambda..$$

We have $S\Rightarrow aSb\Rightarrow aaSbb\Rightarrow aabb$,

that is $S\overset{*}{\Rightarrow} aabb$.

Thus, $aabb$ is a sentence in $L(G)$.

We can prove that $L(G)=\{a^n b^n:\ n\geq 0\}$.

Proof:

Whenever we generate sentences we use

$S\rightarrow aSb$ n $(n\geq 0)$ times and then use $\mathrm{S}\rightarrow\lambda$ once.

It means that G generates and only generates

the sentence $a^n b^n (n\geq 0)$.

Example 2

Find a grammar that generates

$$L = \{a^n b^{n+1} : \ n \geq 0\}.$$

Solution:

The idea of example 1 can be used here. All we need to do is generate an extra b.

This can be done by a production rule $S \to Ab$, and let A derive the language in example 1.

Therefore, we have

$G = \{\{S, A\}, \{a, b\}, S, P\}$, where P consists of

$S \to Ab,$

$A \to aAb,$

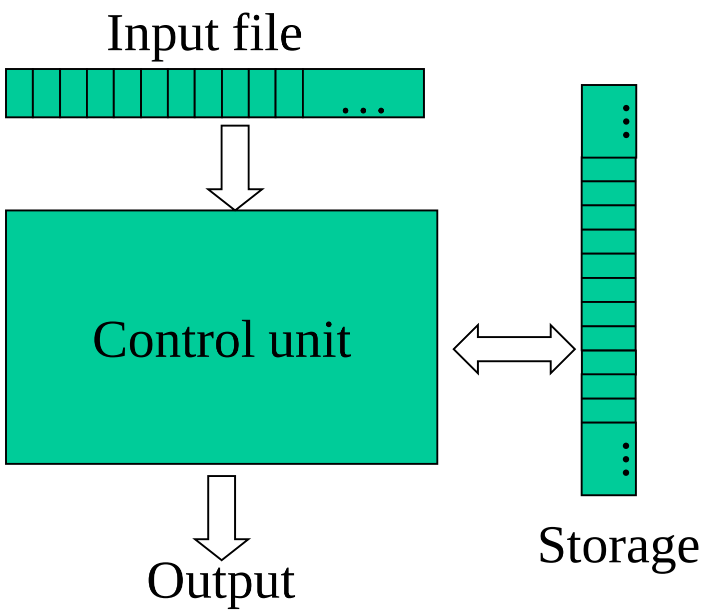
$A \to \lambda.$

A given language may have many grammars that generate it.

We say that two grammars G_1 and G_2 are equivalent if $L(G_1) = L(G_2)$.

Automata

- An **automaton** is an abstract model of a digital computer.



Current configuration

(current state, input and storage)

MOVE

(using Transition function)

Next configuration

(Next state, input and storage)

- Deterministic automata: each move is uniquely determined.

  Nondeterministic automata: the moves are not uniquely determined.

- An automata is called an accepter if its output response is limited to “yes” or “no”.
- An automata is called a transducer if it is capable of producing strings of symbols as output.

Applications

- Compiler design
- Pattern matching
- Example:

The rules for variable identifiers in C are

1. An identifier is a sequence of letters, digits, and underscores.

2. An identifier must start with a letter or an underscore.

3. Identifiers allow upper- and lower-case letters.

Formally, these rules can be described by a grammar.

$$<id> \rightarrow <letter><rest>|<undrscr><rest>$$
$$<rest> \rightarrow <letter><rest>|<digit> <rest>|<undrscr><rest>|\lambda$$
$$<letter> \rightarrow a|b|...|z|A|B|...|Z$$
$$<digit> \rightarrow 0|1|...|9$$
$$<undrscr> \rightarrow \_$$