MTH 310

GRAPHS

Definition 1

Two vertices u and v in an undirected graph G are called adjacent (or neighbours) in G if u and v are endpoints of an edge e of G. Such an edge is called incident with the vertices u and v and e is said to connect u and v.

Definition 2

The set of all neighbours of a vertex v of $G = (V, E)$, denoted by $N(v)$, is called the neighbourhood of v. If A is a subset of V, we denote by $N(A)$ the set of all vertices in G that are adjacent to at least one vertex in A. So, $N(A) = \bigcup_{v \in A} N(v)$.

Definition 3

The degree of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex v is denoted by $\deg(v)$.

Example What are the degrees and neighbourhoods of the vertices in the graphs A and B displayed below?

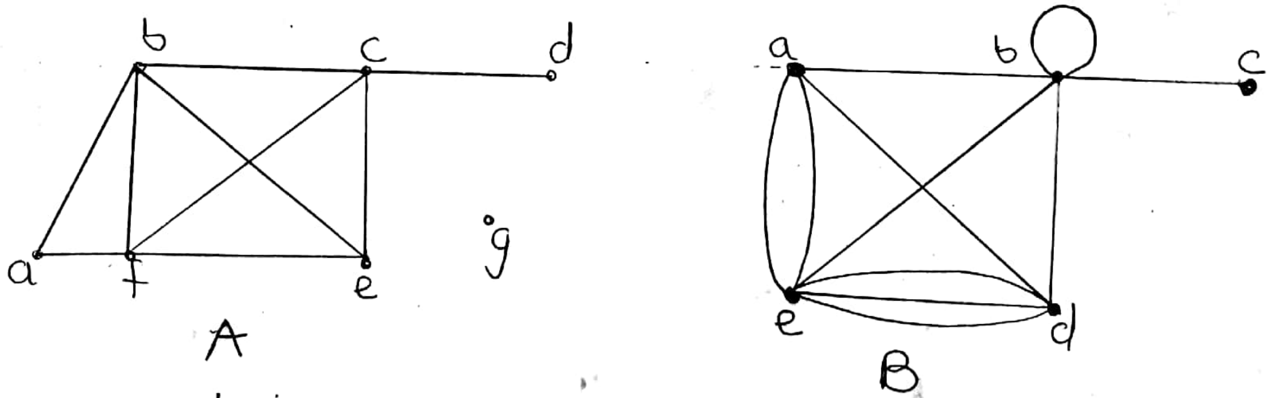


Figure 1. The Undirected Graphs A and B.

Solution:

In A, $\deg(a) = 2$, $\deg(b) = \deg(c) = \deg(f) = 4$, $\deg(d) = 1$, $\deg(e) = 3$ and $\deg(g) = 0$. The neighbourhoods of these vertices are $N(a) = \{b, f\}$, $N(b) = \{a, c, e, f\}$, $N(c) = \{b, d, e, f\}$, $N(d) = \{c\}$, $N(e) = \{b, c, f\}$, $N(f) = \{a, b, c, e\}$ and $N(g) = \emptyset$.

In B, $\deg(a) = 4$, $\deg(b) = \deg(e) = 6$, $\deg(c) = 1$ and $\deg(d) = 5$. The neighbourhoods of these vertices are $N(a) = \{b, d, e\}$, $N(b) = \{a, b, c, d, e\}$, $N(c) = \{b\}$, $N(d) = \{a, b, e\}$ and $N(e) = \{a, b, d\}$.

Definitions 4

A vertex of degree zero is called isolated. It follows that an isolated vertex is not adjacent to any vertex. Vertex g in graph A in the example above is isolated. A vertex is pendant if and only if it has degree one. Consequently a pendant if and only if it has degree one. Consequently, a pendant vertex is adjacent to exactly one other vertex. a pendant vertex is adjacent to exactly one other vertex. Vertex d in graph A in the example above is pendant.

Examining the degrees of vertices in a graph model can provide useful information about the model.

Example — Biological Networks

Many aspects of the biological sciences can be modeled using graphs.

Niche Overlap Graphs in Ecology: Graphs are used in many models involving the interaction of different species of animals. For instance, the competition between species in an ecosystem can be modeled using a niche overlap graph. Each species is represented by a vertex. An undirected edge connects two vertices if the two species represented by these vertices compete (that is, some of the food resources they use are the same). A niche overlap graph is a simple graph because no loops or multiple edges are needed in this model. The graph in Figure 2 models the ecosystem of a forest. From this graph, it can be seen that squirrels and raccoons compete but that crows and shrews do not.

deg (Raccoon) = 3 deg (owl) = 3
deg (Hawk) = 3 deg (Squirrel) = ~~3~~ 4
deg (Crow) = 3 deg (opossum) = 3
deg (woodpecker) = 3 deg (shrew) = 3
deg (mouse) = 1.

There are 13 edges and the sum of the degree of the vertices is 26.

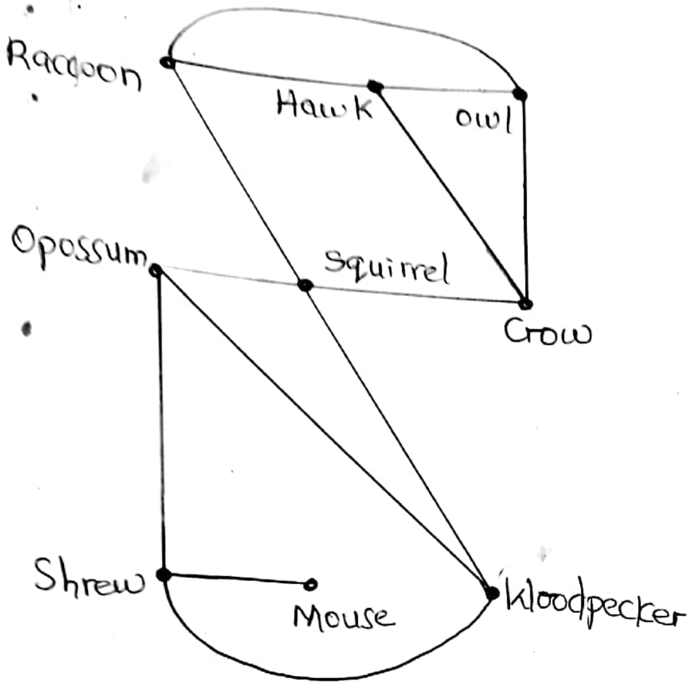


Figure 2 A Niche Overlap Graph

Question: What does the degree of a vertex in a niche overlap graph in figure 2 represent? Which vertices in this graph are pendant and which are isolated?

Solution: There is an edge between two vertices in a niche overlap graph if and only if the two species represented by these vertices compete. Hence, the degree of a vertex in a niche overlap graph is the number of species in the ecosystem that compete with the species represented by this vertex. A vertex is pendant if the species competes with exactly one other species in the ecosystem. Finally, the vertex representing a species is isolated if this species does not compete with any other species in the ecosystem.

For instance, the degree of the vertex representing the opossum in the niche overlap graph is three, because the opossum competes with three other species: the squirrel, the woodpecker and the shrew. In this niche overlap graph, the mouse is the only species represented by a pendant vertex, because the mouse competes only with the shrew and all other species compete with at least two other species. There are no isolated vertices in the graph in this niche overlap graph because every species in this ecosystem competes with at least one other species.

Note: When all the degrees of vertices of a graph are added, the sum of the degrees of the vertices is twice the number of edges because an edge is incident with exactly two (possibly equal) vertices. This leads to what is called the handshaking theorem (lemma). This is because of the analogy between an edge having two endpoints and a handshake involving two hands.

Theorem 1: (The Handshaking Theorem) Let $G=(V,E)$ be an undirected graph with m edges. Then

$$2m = \sum_{v \in V} \deg(v).$$

(Note that this applies even if multiple edges and loops are present.)

Example: How many edges are there in a graph with 18 vertices each of degree 4?

Solution: Because the sum of the degrees of the vertices is $4 \times 18 = 72$, $\Rightarrow 2m = 72$ where m is the number of edges. Therefore, $m = 36$. □

Theorem 1 shows that the sum of the degrees of the vertices of an undirected graph is even.

Theorem 2: An undirected graph has an even number of vertices of odd degree.

Proof: Let V_1 and V_2 be the set of vertices of even degree and the set of vertices of odd degree, respectively in an undirected graph $G=(V,E)$ with m edges. Then

$$2m = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v).$$

Because $\deg(v)$ is even for $v \in V_1$, the first term in the right-hand side of the last equality is even. Furthermore, the sum of the two terms on the right-hand side of the last equality is even because this sum is $2m$. Hence, the second term in the sum is also even. Because all the terms in this sum are odd, there must be an even number of such terms. Thus, there are an even number of vertices of odd degree. □

(46)

Theorem 2: An undirected graph has an even number of vertices of odd degree.

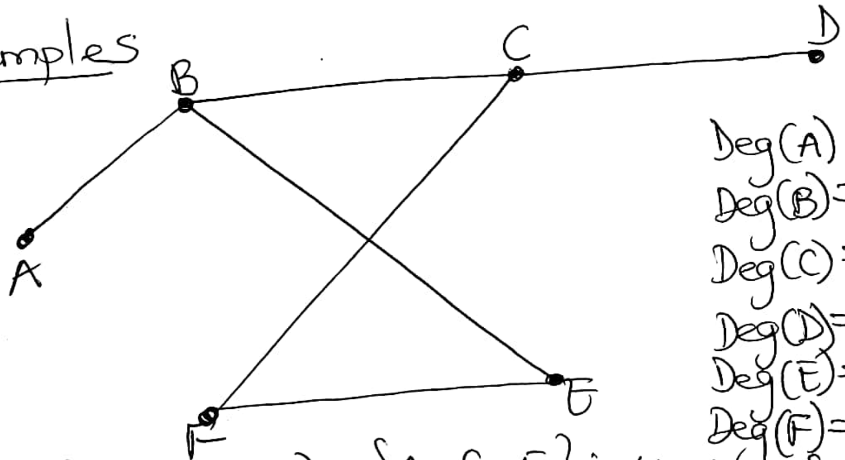
Proof:

The neighbourhood of a vertex has the same vertex as its neighbourhood and probably with some other vertices in an undirected graph. That is, at least a point in the neighbourhood of a vertex is a vertex to the point. Also, the neighbourhood of the vertex having a pendant vertex as a neighbourhood has odd numbered neighbourhood. The incident to a pendant vertex is ~~prime~~ odd, which can be prime.

Since every vertex of a circuit has even degree, then no vertex has odd degree. That is, every vertex has even incident in a simple circuit.
Hence every undirected graph has an even number of vertices with odd degree. □

Examples

(1)



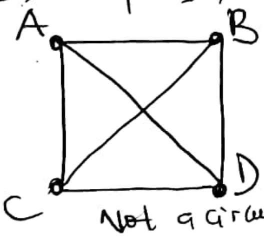
Deg(A) = 1
Deg(B) = 3
Deg(C) = 3
Deg(D) =

4 vertices has odd degree.

Deg(E) = 2
Deg(F) = 2

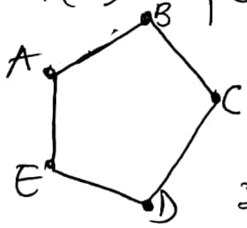
N(A) = {B}; N(B) = {A, C, E}; N(C) = {B, D, F};
N(D) = {C}; N(E) = {B, F}; N(F) = {C, E}.

(2)



Not a circuit

Deg(A)=3
Deg(B)=3
Deg(C)=3
Deg(D)=3
N(A)={B, C, D}
N(B)={A, C, D}
N(C)={A, B, D}
N(D)={A, B, C}.



Deg(A)=2; N(A)={B, E}
Deg(B)=2; N(B)={A, C}
Deg(C)=2; N(C)={B, D}
Deg(D)=2; N(D)={C, E}
Deg(E)=2; N(E)={A, D}

Discrete Mathematics and Its Applications.
Seventh Edition. by Kenneth H. Rosen.

⑤

Note: Terminology for graphs with directed edges reflects the fact that edges in directed graphs have directions.

Definitions 5:

- When (u, v) is an edge of the graph G with directed edges, u is said to be adjacent to v and v is said to be adjacent from u. The vertex u is called the initial vertex of (u,v), and v is called the terminal or end vertex of (u,v). The initial vertex and terminal vertex of a loop are the same.
- In a graph with directed edges the in-degree of a vertex v, denoted by $\deg^-(v)$, is the number of edges with v as their terminal vertex. The out-degree of v, denoted by $\deg^+(v)$, is the number of edges with v as their initial vertex. (Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex)

Example

Find the in-degree and out-degree of each vertex in the graph G with directed edges shown in Figure 3.

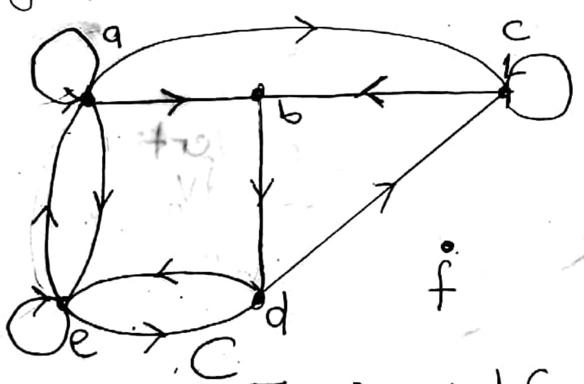


Figure 3 The Directed Graph C

Solution: The in-degrees in G are $\deg^-(a)=2$, $\deg^-(b)=2$, $\deg^-(c)=3$, $\deg^-(d)=2$, $\deg^-(e)=3$ and $\deg^-(f)=0$. The out-degrees are $\deg^+(a)=4$, $\deg^+(b)=1$, $\deg^+(c)=2$, $\deg^+(d)=2$, $\deg^+(e)=3$ and $\deg^+(f)=0$.

Note: Because each edge has an initial vertex and a terminal vertex, the sum of the in-degrees and the sum of the out-degrees of all vertices in a graph with directed edges are the same. Both of these sums are the number of edges in the graph as stated in the following theorem.

Theorem 3: Let $G=(V,E)$ be a graph with directed edges. Then

$$\sum_{v\in V}\deg^-(v)=\sum_{v\in V}\deg^+(v)=|E|.$$

Note: The undirected graph that results from ignoring directions of edges is called the underlying undirected graph.

Example & Definition:

Complete Graphs: A complete graph on n vertices, denoted by T_n, is a simple graph that contains exactly one edge between each pair of distinct vertices. The graphs T_n, for $n=1,2,3,4,5,6$ are displayed in Figure 4. A simple graph for which there is at least one pair of distinct vertex not connected by an edge is called noncomplete.

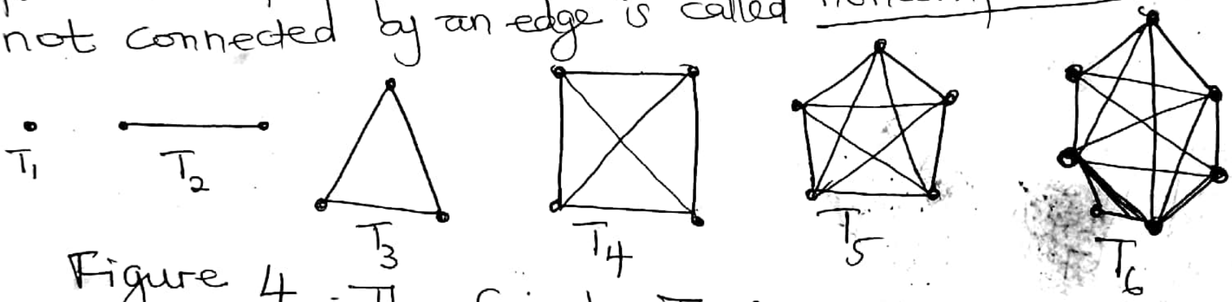


Figure 4. The Graphs T_n for $1\le n\le 6$.

Example:

Cycles: A cycle C_n, $n\ge 3$, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1,v_2\}$, $\{v_2,v_3\}$, $\ldots$, $\{v_{n-1},v_n\}$ and $\{v_n,v_1\}$. The cycles C_3, C_4, C_5 and C_6 are displayed in Figure 5.

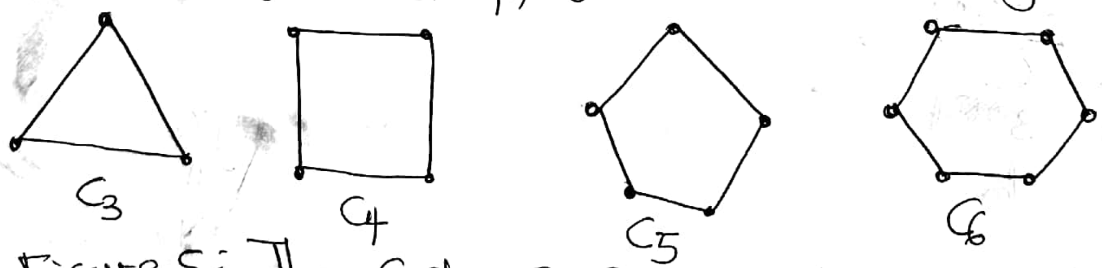


Figure 5: The Cycles C_3, C_4, C_5 and C_6.

Note: The complete graph of order n is defined to be the unique graph with n vertices and $\binom{n}{2}$ edges so that there is an edge connecting every pair of vertices and is denoted by T_n.

Example:

Wheels: We obtain a wheel W_n when we add an additional vertex to a cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n, by new edges. The wheels W_3, W_4, W_5 and W_6 are displayed in Figure 6.

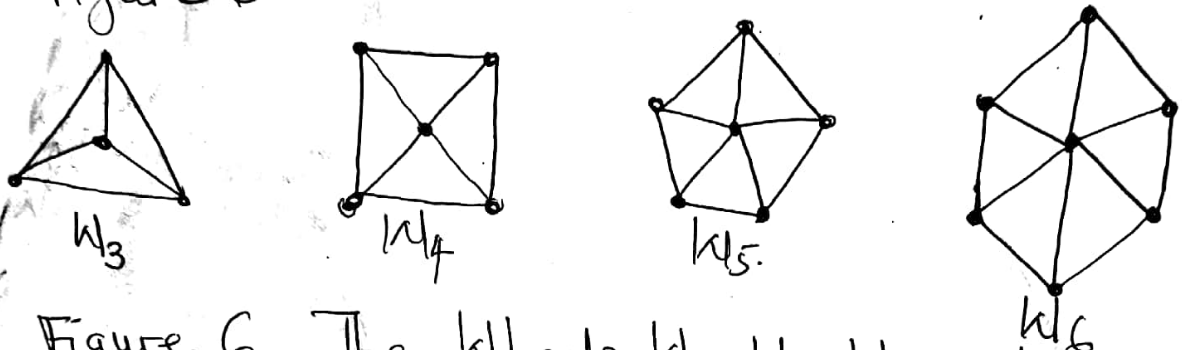


Figure 6 The Wheels W_3, W_4, W_5 and W_6.

New Graphs from Old

- Sometimes only part of a graph is needed to solve a problem. When edges and vertices are removed from a graph, without removing endpoints of any remaining edges, a smaller graph is obtained. Such a graph is called a subgraph of the original graph.

Definition 5.6:

- A subgraph of a graph $G=(V,E)$ is a graph $H=(W,F)$ where $W \subseteq V$ and $F \subseteq E$. A subgraph H of G is a proper subgraph of G if $H \neq G$.
- Let $G=(V,E)$ be a simple graph. The subgraph induced by a subset W of the vertex set V is the graph (W,F), where the edge set F contains an edge in E if and only if both endpoints of this edge are in W.

Subgraphs

Definitions 7: A graph H is a subgraph of a graph G, denoted by $H \subseteq G$, if $V_H \subseteq V_G$ and $E_H \subseteq E_G$. A subgraph $H \subseteq G$ spans G (and H is a spanning subgraph of G), if every vertex of G is in H, i.e., $V_H = V_G$.
Also, a subgraph $H \subseteq G$ is an induced subgraph, if $E_H = E_G \cap E(V_H)$. In this case, H is induced by its set V_H of vertices.

i.e. vertices of H contains all its edges in G.

A spanning subgraph is a maximal subgraph?

In an induced subgraph $H \subseteq G$, the set E_H of edges consists of all $e \in E_G$ such that $e \in E(V_H)$. To each nonempty subset $A \subseteq V_G$, there corresponds a unique Induced subgraph

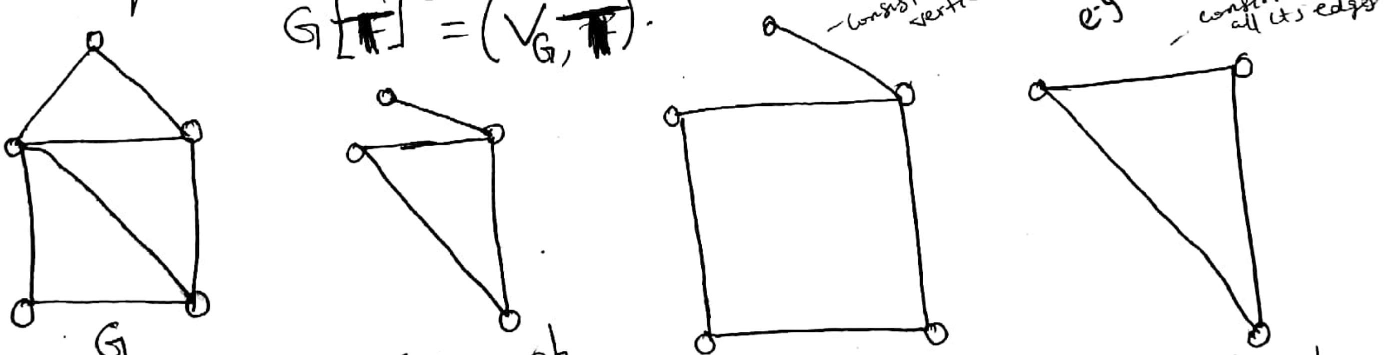
$$G[A] = (A, E_G \cap E(A)).$$

To each subset $T \subseteq E_G$ of edges there corresponds a unique spanning subgraph of G,

$$G[T] = (V_G, T).$$

e.g. — This is not induced because edge ab is missing — consists of all its edges

— consists of all vertices in G.



G   subgraph   spanning   induced

For a set $T \subseteq E_G$ of edges, let

$$G - T = G[E_G \setminus T]$$

That is if H contains all the edges of G that join vertices of V_H then we call H the subgraph of G induced by H.

be the subgraph of G obtained by removing (only) the edges $e \in F$ from G. In particular, $G - e$ is obtained from G by removing $e \in G$.
Similarly, we write $G + F$, if each $e \in F$ (for $F \subseteq E(V_G)$) is added to G.
For a subset $A \subseteq V_G$ of vertices, we let $G - A \subseteq G$ be the subgraph induced by $V_G \setminus A$, that is

$$G - A = G[V_G \setminus A],$$

and e.g., $G - v$ is obtained from G by removing the vertex v together with the edges that have v as their end.

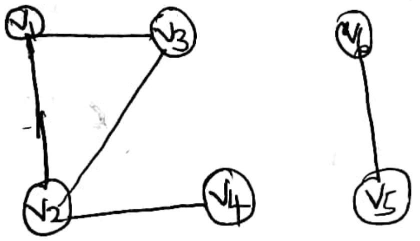
Definitions 8:

For a graph G, we denote

$$\nu_G = |V_G| \text{ and } \varepsilon_G = |E_G|.$$

The number ν_G of the vertices is called the <u>order</u> of G, and ε_G is the <u>size</u> of G. For an edge $e = uv \in G$, the vertices u and v are its ends. Vertices u and v are adjacent or neighbours, if $uv \in G$. Two edges $e_1 = uv$ and $e_2 = uw$ having a common end, are adjacent with each other. i.e.

A graph G can be represented as a plane figure by drawing a line (or a curve) between the points u and v (representing vertices) if $e = uv$ is an edge of G. The figure below is a geometric representation of the graph G with $V_G = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and $E_G = \{v_1v_2, v_1v_3, v_2v_3, v_2v_4, v_5v_6\}$.



This graph is of order 6 and size 5.

Graphs can be generalized by allowing loops uu and parallel (or multiple) edges between vertices to obtain a multigraph $G = (V, E, \psi)$ where $E = \{e_1, e_2, \dots, e_m\}$ is a set (of symbols), and $\psi : E \to E(V) \cup \{uu \mid v \in V\}$ is a function that attaches an unordered pair of vertices to each $e \in E$: $\psi(e) = uv$.

Note that we can have $\psi(e_1) = \psi(e_2)$ — in a multigraph

Definition 9: A directed graphs or <u>digraphs</u> $D(V, E)$ where the edges have a direction, that is, the edges are ordered: $E \subseteq V \times V$. In this case, $uv \neq vu$.

Isomorphism of Graphs.

Definition : Two graphs G and H are isomorphic, denoted by $G \cong H$, if there exists a bijection

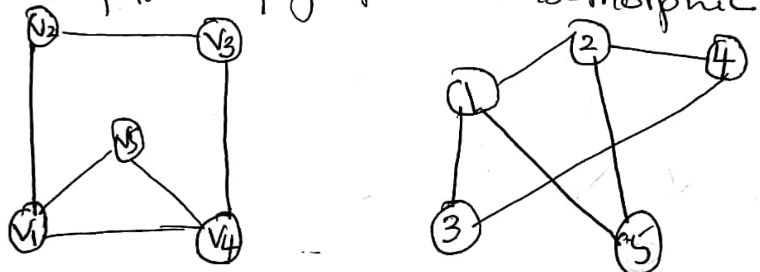
$$\alpha : V_G \longrightarrow V_H \text{ such that}$$

$$uv \in E_G \Leftrightarrow \alpha(u)\alpha(v) \in E_H$$

for all $u, v \in G$.

Hence G and H are isomorphic if the vertices of H are renamings of those of G. Two isomorphic graphs enjoy the same graph theoretical properties, and they are often identified.

* Example : The following graphs are isomorphic. Indeed, the



required isomorphism is given by $V_1 \longmapsto 1$, $V_2 \longmapsto 3$, $V_3 \longmapsto 4$, $V_4 \longmapsto 2$, $V_5 \longmapsto 5$.

Let $V_G = \{V_1, \cdots, V_n\}$ be ordered. The adjacency matrix of G is the $n \times n$ matrix M with entries $M_{ij} = 1$ or $M_{ij} = 0$ according to whether $V_iV_j \in G$ or $V_iV_j \notin G$.

*Notice that the adjacency matrix is always symmetric (with respect to its diagonal consisting of zeros).

i.e. $\deg(V_1) = 3$

$\deg(V_2) = 2$

$\deg(V_3) = 2$

$\deg(V_4) = 3$

$\deg(V_5) = 2$

$\deg(1) = 3$

$\deg(2) = 3$

$\deg(3) = 2$

$\deg(4) = 2$

$\deg(5) = 2$

Connectivity

Many problems can be modeled with paths formed
by traveling along the edges of graphs. For instance,
the problem of determining whether a message can
be sent between two computers using intermediate
links can be studied with a graph model. Problems
of efficiently planning routes for mail delivery,
garbage pickup, diagnostics in computer networks,
and so on can be solved using models that involve
paths in graphs.

Definitions 10:

Let n be a nonnegative integer and G an undirected
graph. A path of length n from u to v in G is a
sequence of n edges $e_1, \ldots, e_n$ of G for which there
exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices
such that e_i has, for $i = 1, \ldots, n$, the endpoints
x_{i-1} and x_i. When the graph is simple, we denote
this path by its vertex sequence $x_0, x_1, \ldots x_n$
(because listing these vertices uniquely determines
the path). The path is a circuit if it begins and
ends at the same vertex, that is, $u = v$, and
has length greater than zero. The path or circuit
is said to pass through the vertices $x_1, x_2, \ldots, x_{n-1}$
or traverse the edges $e_1, e_2, \ldots, e_n$.
A path or circuit is simple if it does not contain
the same edge more than once. Some books refer to
path as walk. A walk is defined as an alternating
sequence of vertices and edges of a graph, $v_0, e_1, v_1, e_2,$
$\ldots, v_{n-1}, e_n, v_n$, where v_{i-1} and v_i are the endpoints
of e_i for $i = 1, 2, \ldots n$. Also, closed walk is used
instead of circuit to indicate a walk that begins

and ends at the same vertex, and trail is used to denote a walk that has no repeated edge (replacing the term simple path).

Definition 8.11:

An undirected graph is called connected if there is a path between every pair of distinct vertices of the graph. An undirected graph that is not connected is called disconnected. We say that we disconnect a graph when we remove vertices or edges, or both to produce a disconnected subgraph.

Thus, any two computers in the network can communicate if and only if the graph of this network is connected.

Example: The graph A in Figure 7 is connected, because for every pair of distinct vertices there is a path between them. However, the graph B in Figure 7 is not connected. For instance, there is no path in B between vertices a and d.

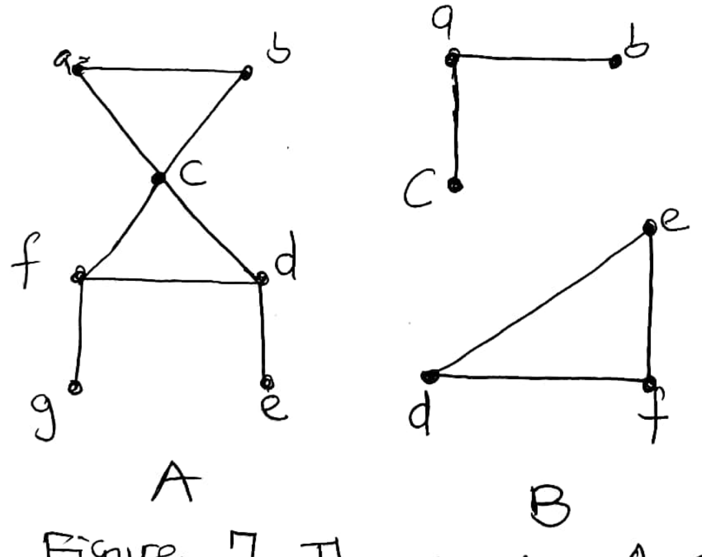


Figure 7 The Graphs A and B.

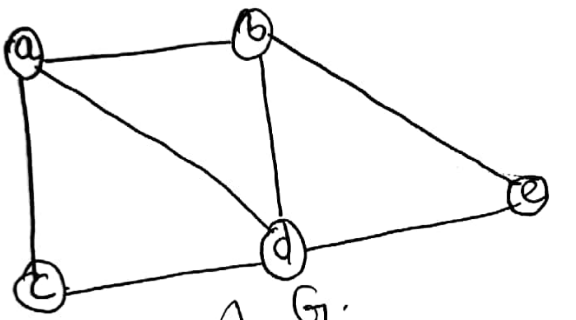
In graph A an example of path is g f d e ; e d c b a

Definitions 12:

— An Eulerian Path (Eulerian trail, Euler walk) in a graph is a path that uses each edge precisely once. If such a path exists, the graph is called traversable.

— An Eulerian Cycle (Eulerian circuit, Euler tour) in a graph is a cycle that uses each edge precisely once. If such a cycle exists, the graph is called Eulerian (also unicursal)

Representation Example: Graph G has Euler path a, c, d, e, b, d, a, b

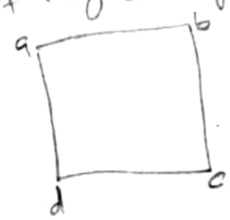


Graph G.

Definitions 13: A graph $G = (V, E)$ is trivial, if it has only one vertex, i.e., $V_G = 1$; otherwise G is nontrivial. The graph $G = K_V$ is the complete graph on V, if every two vertices are adjacent: $E = E(V)$. All complete graphs of order n are isomorphic with each other, and they will be denoted by K_n.

A graph G is said to be regular, if every vertex of G has the same degree. If this degree is equal to r, then G is r-regular or regular of degree r.

E.g. of regular graph



deg(a) = 2
deg(b) = 2
deg(c) = 2
deg(d) = 2

∴ it is a 2-regular graph.

Scanned by CamScanner

Connected Components

Definitions 14

A connected component of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G. That is, a connected component of a graph G is a maximal connected subgraph of G. A graph G that is not connected has two or more connected components that are disjoint and have G as their union.

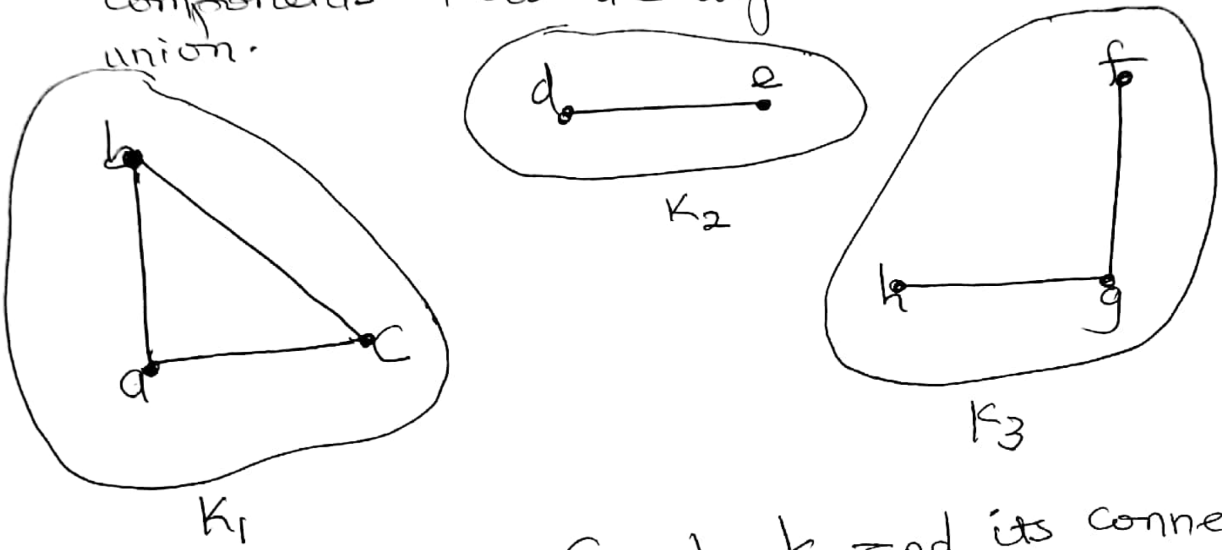


Figure 8: The Graph K and its connected Components K_1, K_2 and K_3.

Example: What are the connected components of the graph K shown in Figure 8?

Solution: The graph K is the union of three disjoint connected subgraphs K_1, K_2 and K_3, shown in Figure 8. These three subgraphs are the connected components of K.

Theorem: There is a simple path between every pair of distinct vertices of a connected undirected graph.

Proof: Let u and v be two distinct vertices of the connected undirected graph $G=(V,E)$. Because G is connected, there is at least one path between u and v. Let $x_0, x_1, \ldots, x_n$, where $x_0 = u$ and $x_n = v$, be the vertese sequence of a path of least length. This path of least length is simple. To see this, suppose it is not simple. Then $x_i = x_j$ for some i and j with $0 \le i < j$. This means that there is a path from u to v of shorter length with vertese sequence $x_0, x_1, \ldots, x_{i-1}, x_j, \ldots, x_n$ obtained by deleting the edges corresponding to the vertex sequence $x_i, \ldots, x_{j-1}$ i.e. $x_i, \ldots, x_{j-1}$ □

A graph G is called connected if for every pair $\{x, y\}$ of distinct vertices there is a path from x to y. The maximum connected subgraphs of a graph G are called the connected components of G. A forest is a graph with no cycles and a tree is a connected graph with no cycles. Therefore, the connected components of a forest are all trees.

A spanning tree of a graph G is a subgraph that is a tree and that contains every vertex of the graph G.

Example

trail (A trail does not contain an edge twice)

In the simple graph shown in Figure 9. a, d, c, f, e is a simple path (or trail) of length 4, because {a, d}, {d, c}, {c, f} and {f, e} are all edges. However, d, e, c, a is not a path, because {e, c} is not an edge. Note that b, c, f, e, b is a circuit of length 4 because {b, c}, {c, f}, {f, e} and {e, b} are edges and this path begins and ends at b. The path a, b, e, d, a, b which is of length 5, is not simple because it contains the edge {a, b} twice.

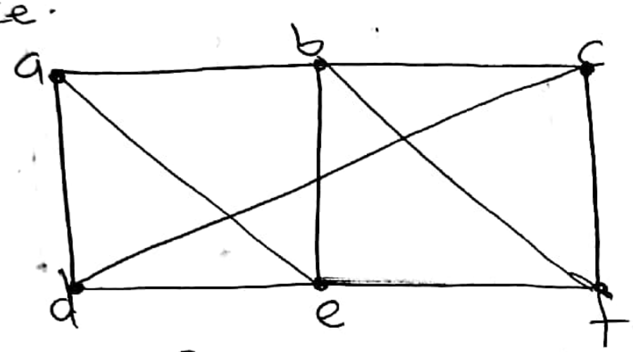


Figure 9: A Simple Graph

How Connected is a Graph?

Suppose that a graph represents a computer network. Knowing that this graph is connected means that any two computers on the network can communicate. However, it is of interest to understand how reliable this network is. For instance, will it still be possible for all computers to communicate after a router or a communications link fails? To answer this and similar questions the following concepts are necessary.

The removal from a graph of a vertex and all incident edges produces a subgraph with more connected components (maximal subgraph) and such vertices are called cut vertices (or articulation points). The removal of a cut vertex from a connected graph produces a subgraph that is not connected. Analogously, an edge whose removal produces a graph with more connected components (maximal connected subgraph of a graph) than in the original graph is called a cut edge or bridge.

Note that in a graph representing a computer network, a cut edge and a cut vertex represent an essential router and an essential link that cannot fail for all computers to be able to communicate.

Example: Find the cut vertices and cut edges in the graph A_1 shown in Figure 10.

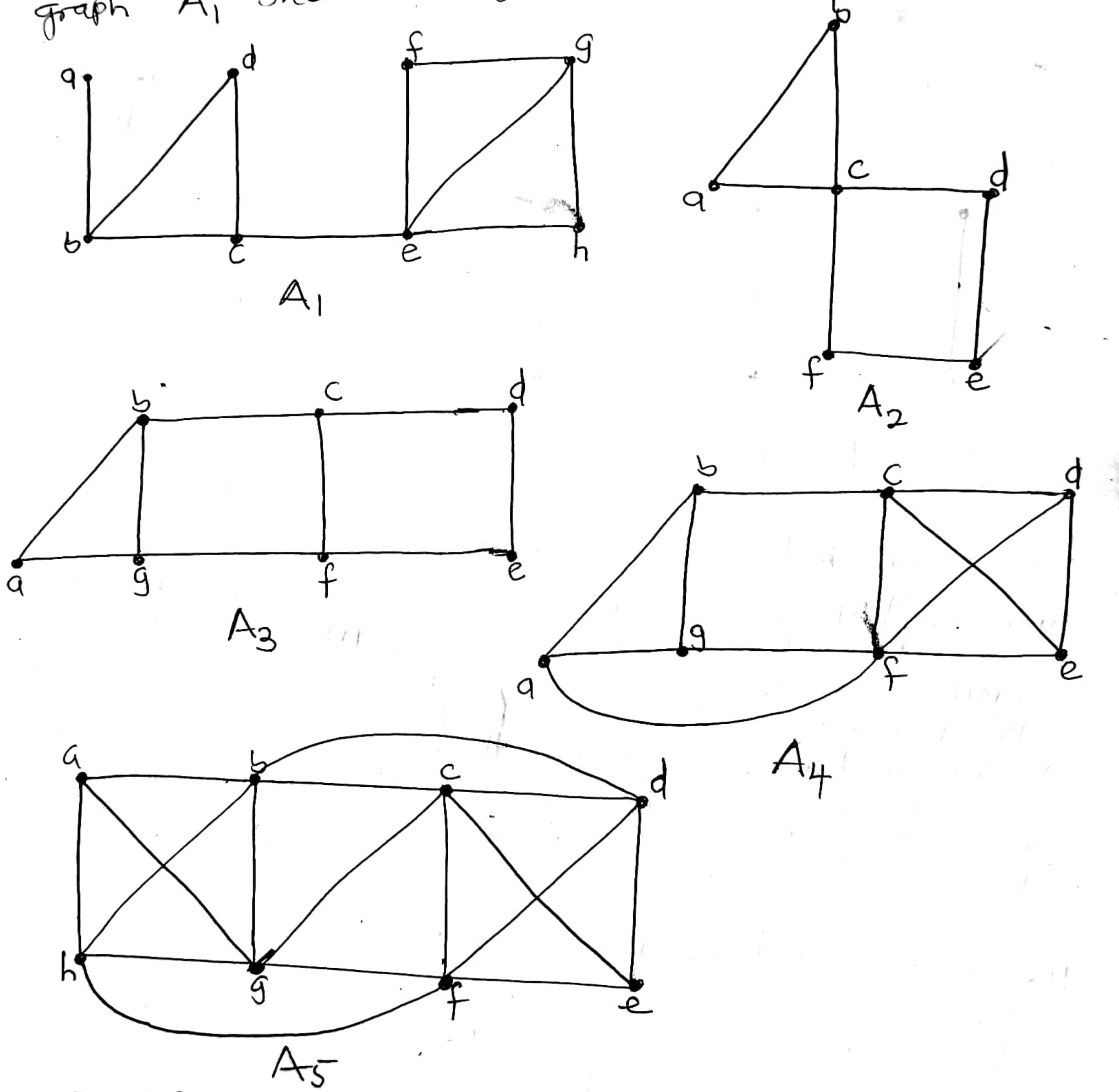


Figure 10: Some connected Graphs.

Solution: The cut vertices of A_1 are b, c and e. The removal of one of these vertices (and its adjacent edges) disconnects the graph. The cut edges are $\{a, b\}$ and $\{c, e\}$. Removing either one of these edges disconnects A_1.

Vertex Connectivity

Not all graphs have cut vertices. For example, the complete graph T_n, where $n \geq 3$, has no cut vertices. When a vertex is removed from T_n and all edges incident to it, the resulting subgraph is the complete graph T_{n-1}, a connected graph. Connected graphs without cut vertices are called nonseparable graphs.

It should be noted that every connected graph except a complete graph, has a vertex cut. The vertex connectivity of a non complete graph G, denoted by $K(G)$ is defined as the minimum number of vertices in a vertex cut.

When G is a complete graph, it has no vertex cuts, because removing any subset of its vertices and all incident edges still leaves a complete graph. Consequently, $K(G)$ cannot be defined as the minimum number of vertices in a vertex cut when G is complete. Instead, we set $K(T_n) = n-1$, the number of vertices needed to be removed to produce a graph with a single vertex.

For every graph G, $K(G)$ is minimum number of vertices that can be removed from G to either disconnect G or produce a graph with a single vertex.

That is, $0 \leq K(G) \leq n-1$ if G has n vertices, $K(G)=0$ if and only if G is disconnected or $G = T_1$ and $K(G) = n-1$ if and only if G is complete.

The larger $K(G)$ is, the more connected G is considered to be. Disconnected graphs and T_1 have $K(G) = 0$, connected graphs with cut vertices and T_2 have $K(G)=1$, graphs without cut vertices that can be disconnected by removing two vertices and K_3 have $K(G)=2$, and so on. A graph is K-Connected (or K-Vertex-connected), if $K(G) \geq K$. A graph G is 1-connected if it is connected and not a graph containing a single vertex; a graph is 2-connected or biconnected, if it is nonseparable and has at least three vertices. Note that if G is a K-connected graph, then G is a j-connected graph for all j with $0 \leq j \leq K$.

Example: Find the vertex connectivity for each of the graphs in Figure 10.

Solution: Each of the five graphs in Figure 10 is connected, and has more than one vertex. A_1 is a connected graph with a cut vertex as shown before, and $\kappa(A_1)=1$. Similarly, $\kappa(A_2)=1$, because c is a cut vertex of A_2. A_3 has no cut vertices, but $\{b, g\}$ is a vertex cut. Hence $\kappa(A_3)=2$. Similarly, because A_4 has a vertex cut of size two, $\{c, f\}$, but no cut vertices. It follows that $\kappa(A_4)=2$. A_5 has no vertex cut of size two, but $\{b, c, f\}$ is a vertex cut of A_5. Hence, $\kappa(A_5)=3$.

Edge Connectivity

The connectivity of a connected graph $G(V,E)$ can also be be measured in terms of the minimum number of edges that we can remove to disconnect it. If a graph has a cut edge, then there is need to remove it to disconnect G. If G does not have a cut edge, we look for the smallest set of edges that can be removed to disconnect it.

A set of edges E' is called an edge cut of G if the subgraph $G-E'$ is disconnected. The edge connectivity defines $\lambda(G)$ for all connected graphs with more than one vertex because it is always possible to disconnect such a graph by removing all edges incident to one of its vertices.

Note that $\lambda(G)=0$ if G is not connected. We also specify that $\lambda(G)=0$ if G is a graph consisting of a single vertex. It follows that if G is a graph with n vertices, then $0 \leq \lambda(G) \leq n-1$. It should be noted that $\lambda(G)=n-1$ where G is a graph with n vertices if and only if $G=K_n$ which is equivalent to the statement that $\lambda(G) \leq n-2$ when G is not a complete graph.

Boolean Expressions ... The set
n $\{(x_1, x_2, \ldots, x_n) \mid x_i \in B$ for $\{0,1\}$

Example: Find the edge connectivity of each of the graphs in figure 10.

Solution: Each of the five graphs in figure 10 is connected and has more than one vertex, and that all of them have positive edge connectivity.

A_1 has a cut edge, so $\lambda(A_1) = 1$.

The graph A_2 has no cut edges, but the removal of the two edges $\{a,b\}$ and $\{a,c\}$ disconnects it. Hence, $\lambda(A_2) = 2$. Similarly, $\lambda(A_3) = 2$ because A_3 has no cut edges, but the removal of the two edges $\{b,c\}$ and $\{f,g\}$ disconnects it.

The removal of no two edges disconnects A_4, but the removal of the three edges $\{b,c\}, \{a,f\}$ and $\{f,g\}$ disconnects it. Hence, $\lambda(A_4) = 3$. Also $\lambda(A_5) = 3$, because the removal of any two of its edges does not disconnect it, but the removal of $\{a,b\}$, $\{a,g\}$ and $\{a,h\}$ does.

An Inequality for Vertex Connectivity and Edge Connectivity

When $G = (V, E)$ is a noncomplete connected graph with at least three vertices, the minimum degree of a vertex of G is an upper bound for both the vertex connectivity of G and the edge connectivity of G. That is, ~~$\kappa(G$~~

$\kappa(G) \leq \min_{v \in V} \deg(v)$ and $\lambda(G) \leq \min_{v \in V} \deg(v)$. It should be observed that deleting all the neighbours of a fixed vertex of minimum degree disconnects G and deleting all the edges that have a fixed vertex of minimum degree as an endpoint disconnects G. It has been shown that $\kappa(G) \leq \lambda(G)$ when G is a connected noncomplete graph. Note also that $\kappa(K_n) = \lambda(K_n) = \min_{v \in V} \deg(v) = n-1$ when n is a positive integer, and that $\kappa(G) = \lambda(G) = 0$ when G is a disconnected graph. With these facts, it has been established that for ~~all~~ all graphs G

$$\kappa(G) \leq \lambda(G) \leq \min_{v \in V} \deg(v).$$

Applications of Vertex and Edge Connectivity.

Graph connectivity plays an important role in many problems involving the reliability of networks. Data network can be modelled using vertices to represent routers and edges to represent links between them. The vertex connectivity of the resulting graph equals the minimum number of routers that disconnect the network when they are out of service. If fewer routers are down, ~~data~~ transmission between every pair of routers is still possible. The edge connectivity represents the minimum number of fiber-optic links (or fibre channel ~~that~~ of an optical fiber communications system which provides a data connection between two points) that can be down to disconnect the network. If fewer links are down, it will still be possible for ~~data~~ to be transmitted between every pair of routers.

We can model a highway network, using vertices to represent highway intersections and edges to represent sections of roads running between intersections. The vertex connectivity of the resulting graph represents the minimum number of intersections that can be closed ~~at~~ a particular time that makes it impossible to travel between every two intersections. If fewer intersections ~~are~~ are closed, travel between every pair of intersections is still possible. The edge connectivity represents the minimum number of roads that can be closed to disconnect the highway network. If fewer highways are closed, it will still be possible to travel between any two intersections. This information is important for road repairs.

$S(x_1, x_2, \ldots, x_n) \mid x_i \in B$ for $\{0,1$

Connectedness in Directed Graphs

Definition: A directed graph is strongly connected if there is a path from a to b and from b to a whenever a and b are vertices in the graph.

Note that a directed graph can fail to be strongly connected but still be in "one piece" as seen in the following definition.

Definition: A directed graph is weakly connected if there is a path between every two vertices in the underlying undirected graph.

That is, a directed graph is weakly connected if and only if there is always a path between two vertices when the directions of the edges are disregarded. Clearly, any strongly connected directed graph is also weakly connected.

Example

Are the directed graphs A and B shown in Figure 11 strongly connected? Are they weakly connected?

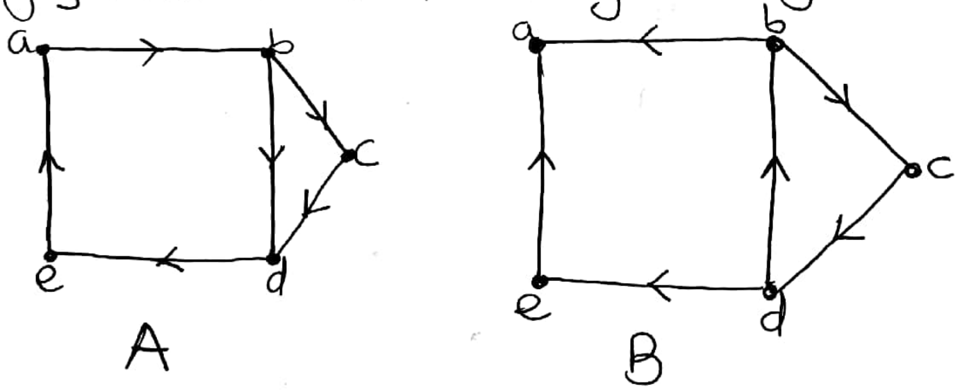


Figure 11 The Directed Graphs A and B.

Solution: A is strongly connected because there is a path between any two vertices in this directed graph. Hence, A is also weakly connected. The graph B is not strongly connected. There is no directed path from a to b in this graph. However, B is weakly connected, because there is a path between any two vertices in the underlying undirected graph of B.

Representing Graphs

Another way to represent a graph with no multiple edges is to use adjacency lists, which specify the vertices that are adjacent to each vertex of the graph.

<u>Example</u>: Use adjacency lists to describe the simple graph given in Figure 12.

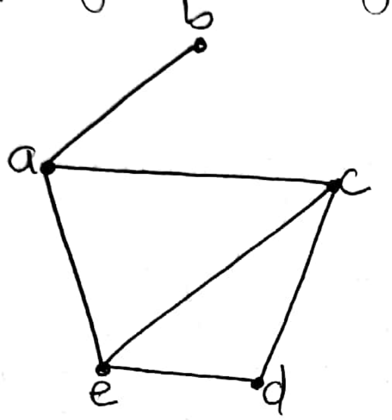


Figure 12 A Simple Graph.

Table 1 An Adjacency List for a Simple Graph

Vertex	Adjacent Vertices
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

<u>Example</u>: Represent the directed graph shown in Figure 13 by listing all the vertices that are terminal vertices of edges starting at each vertex of the graph.

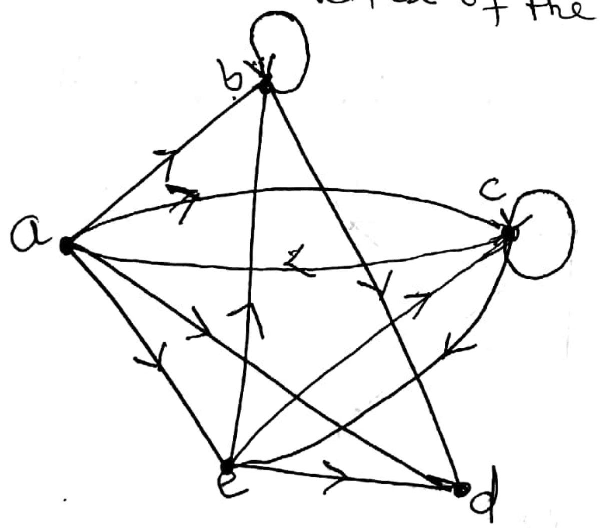


Figure 13 A Directed Graph

Table 2 An Adjacency List for a Directed Graph.

Initial Vertex	Terminal vertices
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

Adjacency Matrices

Graphs can be represented using matrices to simplify computation that arise from carrying out graph algorithms using the representation of graphs by lists of edges or by adjacency lists. Listing edges can be burdensome if there are many edges in the graph.

Two types of matrices that are commonly used to represent graphs are discussed here. One is based on the adjacency of vertices and the other is based on incidence of vertices and edges.

Suppose that $G = (V, E)$ is a simple graph where $|V| = n$. Suppose that the vertices of G are listed arbitrarily as $v_1, v_2, \ldots, v_n$. The adjacency matrix A (or A_G) of G, with respect to this listing of the vertices, is the $n \times n$ zero-one matrix with 1 as its (i, j)th entry when v_i and v_j are adjacent and 0 as its (i, j)th entry when they are not adjacent. In other words, if its adjacency matrix is $A = [a_{ij}]$, then

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

Example: Use an adjacency matrix to represent the graph shown in Figure 14.

Solution:

The vertices are ordered as a, b, c, d. The matrix representing this graph is

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

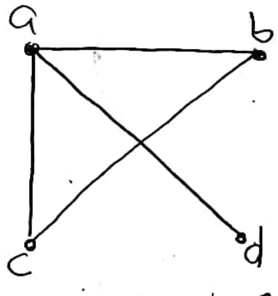


Figure 14 Simple Graph.

Example: Draw a graph with the adjacency matrix

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

solution

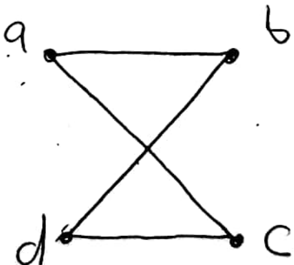
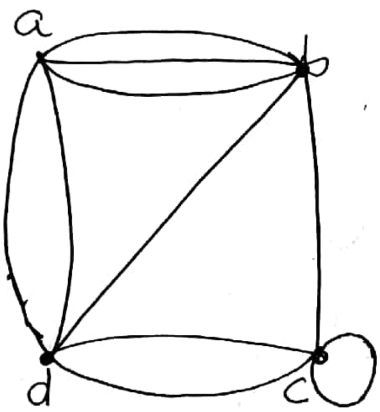


Figure 15 : A Graph with the Given Adjacency Matrix

Adjacency matrices can also be used to represent undirected graphs with loops and with multiple edges. A loop at the vertex v_i is represented by a 1 at the (i,i)th position of the adjacency matrix. When multiple edges connecting the same pair of vertices v_i and v_j, or multiple loops at the same vertex, are present, <u>the adjacency matrix is no longer a zero-one matrix</u>, because the (i,j)th entry of this matrix equals the number of edges that are associated to $\{v_i, v_j\}$. All undirected graphs, including pseudographs and multigraphs, have symmetric adjacency matrices.

<u>Example</u> Use an adjacency matrix to represent the pseudograph shown in Figure 16.



<u>Solution</u>: The adjacency matrix using the ordering of vertices a, b, c, d is

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

Figure 16 A Pseudograph

Note that if $A = [a_{ij}]$ is the adjacency matrix for the directed graph with respect to listing of the vertices, then

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \text{ is an edge of } G \\ 0 & \text{otherwise.} \end{cases}$$

The adjacency matrix for a directed graph does not have to be symmetric.

Scanned by CamScanner

Incidence Matrices.

Another common way to represent graphs is to use incidence matrices. Let $G=(V,E)$ be an undirected graph. Suppose that $v_1, v_2, \dots, v_n$ are the vertices and $e_1, e_2, \dots, e_m$ are the edges of G. Then the incidence matrix with respect to this ordering of V and E is the $n \times m$ matrix $M=[m_{ij}]$, where

$$m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise} \end{cases}$$

Example Represent the graph shown in Figure 17 with an incidence matrix.

Solution: The incidence matrix is

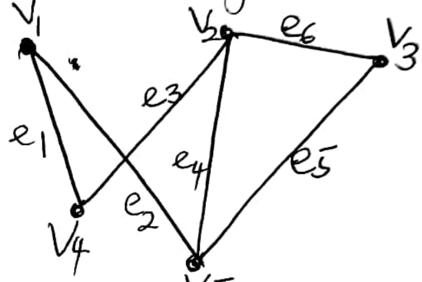


Figure. 17 An Undirected Graph.

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Note : Multiple edges are represented in the incidence matrix using columns with identical entries, because these edges are incident with the same pair of vertices. Loops are represented using a column with exactly one entry equal to 1, corresponding to the vertex that is incident with this loop.

Example: Represent the pseudograph shown in Figure 18 using an incidence matrix.

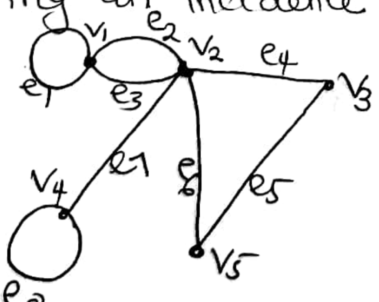


Figure 18 A Pseudograph

Solution: The incidence matrix for this graph is

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

Counting Paths Between Vertices.

The number of paths between two vertices in a graph can be determined using its adjacency matrix.

Theorem Let G be a graph with adjacency matrix A with respect to the ordering $v_1, v_2, \ldots, v_n$ of the vertices of the graph (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length r from v_i to v_j, where r is a positive integer, equals the (i,j)th entry of A^r.

Example: How many paths of length four are there from a to d in the simple graph G in figure 19?

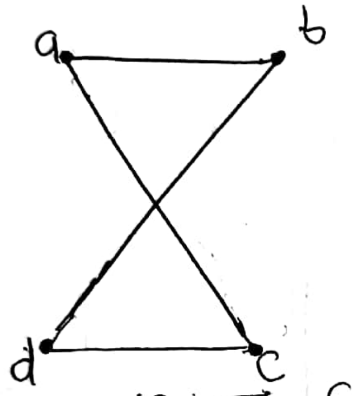


Figure 19 The Graph G.

The adjacency matrix of G (ordering the vertices as a, b, c, d) is

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Hence, the number of paths of length four from a to d is the $(1,4)$th of A^4.

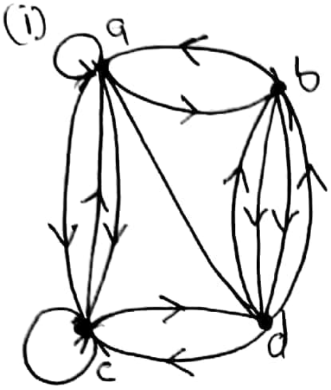
$$A^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

Because there are exactly eight paths of length four from a to d. By inspection of the graph we see that a, b, a, b, d; a, b, a, c, d; a, b, d, b, d; a, b, d, c, d; a, c, a, b, d; a, c, a, c, d; a, c, d, b, d; and a, c, d, c, d are the eight paths of length four from a to d.

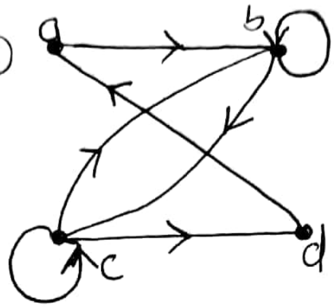
Exercise

① Find the adjacency matrix of each of the following directed multigraph with respect to the vertices listed in alphabetic order.

(i)



(ii)

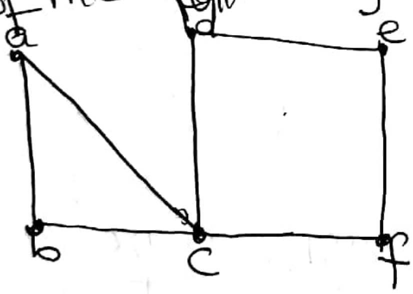


② Draw the graph represented by each of the following adjacency matrices

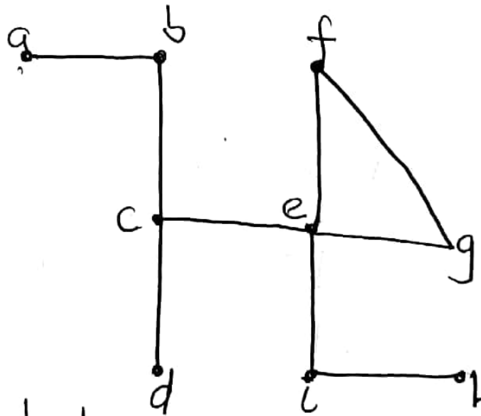
(i) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ (ii) $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix}$ (iii) $\begin{bmatrix} 0 & 2 & 3 & 0 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 1 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix}$

③ Find all the cut vertices and cut edges of each of the following graphs:

(i)

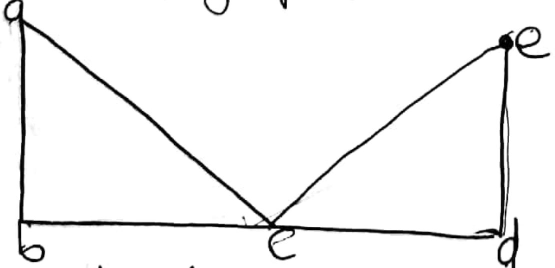


(ii)

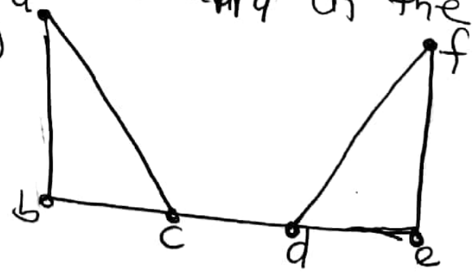


④ Find the number of path between a and d in the following graphs

(i)



(ii)



of length (a) 2 (b) 3 (c) 4

Boolean Algebra

The circuits in computers and other electronic devices have inputs, each of which is either a 0 or a 1, and produce outputs that are also 0s and 1s. Circuits can be constructed using any basic element that has two different states.

Boolean Functions

Boolean algebra provides the operations and the rules for working with the set {0, 1}. Electronic and optical switches can be studied using this set and the rules of Boolean algebra. The three operations in Boolean algebra mostly used are complementation, the Boolean sum and the Boolean product.

The complement of an element, denoted with a bar, is defined by $\overline{0} = 1$ and $\overline{1} = 0$. The Boolean sum, denoted by + or by OR, has the following values.

$1+1=1$, $1+0=1$, $0+1=1$, $0+0=0$.

The Boolean product, denoted by · or by AND, has the following values

$1\cdot 1=1$, $1\cdot 0=0$, $0\cdot 1=0$, $0\cdot 0=0$.

The rules of precedence for Boolean operators (unless parentheses are used) are: first, all complements are computed, followed by all Boolean products, followed by all Boolean sums. rule = CPS: complement, product, sum

Example: Find the value of $1\cdot 0 + \overline{(0+1)}$

$= 0 + \overline{1} = 0 + 0 = 0$

The complement, Boolean sum and Boolean product correspond to the logical operators $\neg$, $\vee$ and $\wedge$ respectively, where 0 corresponds to F (false) and 1 corresponds to T (true). Equality in Boolean algebra can be directly translated into equivalences of compound propositions.

Example: Translate $1\cdot 0 + \overline{(0+1)} = 0$, into a logical equivalence

Solution: A logical equivalence is obtained when we translate each 1 into a T, each 0 into an F, each Boolean sum into a disjunction, each Boolean product into a conjunction and each complementation into a negation. We obtain

$(T \wedge F) \vee \neg(F \vee T) \equiv F$

The following example illustrates the translation from propositional logic to Boolean algebra.

Example: Translate the logical equivalence $(T\wedge T) \vee \neg F \equiv T$ into an identity in Boolean algebra. That is $(1\cdot 1) + \overline{0} = 1$.

Boolean algebra provides the operations and the rules working ①

Boolean Expressions and Boolean Functions

Let $B = \{0, 1\}$. Then $B^n = \{(x_1, x_2, \ldots, x_n) \mid x_i \in B$ for $1 \le i \le n\}$ is the set of all possible n-tuples of 0s and 1s. The variable x is called a Boolean variable if it assumes values only from B, that is, if its only possible values are 0 and 1. A function from B^n to B is called a Boolean function of degree n.

Boolean operations are the product and complementation.

Example: The function $F(x, y) = \bar{x}\bar{y}$ from the set of ordered pairs of Boolean variables to the set {0, 1} is a Boolean function of degree 2 with $F(1,1) = 0, F(0,1) = 0$, $F(1,0) = 0$ and $F(0,0) = 1$. That is

Table 1

x	y	F(x,y)
1	1	0
1	0	0
0	1	0
0	0	1

Boolean functions can be represented using expressions make up from variables and Boolean operations. The Boolean expressions in the variables $x_1, x_2, \ldots, x_n$ are defined recursively as:

applying a formula to preceding terms to get new ones

$0, 1, x_1, x_2, \ldots x_n$ are Boolean expressions;

If E_1 and E_2 are Boolean expressions, then $\overline{E_1}$, $(E_1 E_2)$ and $(E_1 + E_2)$ are Boolean expressions.

Each Boolean expression represents a Boolean function.

Example: Find the values of the Boolean function represented by $F(x, y, z) = y\bar{x} + \bar{z}$

Solution Table 2

just like logic TTT TTF

x	y	z	$y\bar{x}$	$\bar{x}$	$\bar{z}$	$F(x,y,z) = y\bar{x} + \bar{z}$
1	1	1	0	0	0	0
1	1	0	0	0	1	1
1	0	1	0	0	0	0
1	0	0	0	0	1	1
0	1	1	1	1	0	1
0	1	0	1	1	1	1
0	0	1	0	1	0	0
0	0	0	0	1	1	1

equal
Boolean functions A and B of n variables are equal if and only if $A(b_1, b_2, \ldots b_n) = B(b_1, b_2, \ldots b_n)$ whenever $b_1, b_2, \ldots$, belong to B. Two different Boolean expressions that represent the same function are called equivalent. For instance $y\bar{x} + 0$ and $y\bar{x} \cdot 1$ are equivalent. The complement of the Boolean function A is the function $\bar{A}$ where $\bar{A}(x_1, \ldots x_n) = \overline{A(x_1, \ldots x_n)}$.

Let A and P be Boolean functions of degree n. The Boolean sum $A+P$ and the Boolean product AP are defined by

$$(A+P)(x_1, \ldots x_n) = A(x_1, \ldots x_n) + P(x_1, \ldots x_n)$$
$$(AP)(x_1, \ldots x_n) = A(x_1, x_2, \ldots x_n) P(x_1\, x_2 \ldots x_n).$$

Question: How many different Boolean functions of degree n are there?

Solution: From the product rule for counting, it follows that there are 2^n different n-tuples of 0s and 1s. Because a Boolean function is an assignment of 0 or 1 to each of these 2^n different n-tuples, the product rule shows that there are 2^{2^n} different Boolean functions of degree n.

Table 3 Boolean Identities

Identity	Name
$\bar{\bar{x}} = x$	Law of the double complement
$x + x = x$, $x \cdot x = x$	Idempotent laws
$x + 0 = x$, $x \cdot 1 = x$	Identity laws
$x + 1 = 1$, $x \cdot 0 = 0$	Domination laws
$x + y = y + x$, $xy = yx$	Commutative laws
$x + (y+z) = (x+y)+z$, $x(yz) = (xy)z$	Associative laws
$x(y+z) = xy + xz$	Distributive laws
$\overline{(xy)} = \bar{x} + \bar{y}$, $\overline{(x+y)} = \bar{x}\bar{y}$	De Morgan's Laws
$x + xy = x$, $x(x+y) = x$	Absorption laws
$x + \bar{x} = 1$	Unit Property
$x\bar{x} = 0$	zero property

(3)

Exercise: Translate the distributive law $x(y+z) = xy + xz$ into a logical equivalence.

x	y	z	$y+z$	xy	xz	$x(y+z)$	$xy+xz$

Duality

The dual of a Boolean expression is obtained by interchanging Boolean sums and Boolean products and interchanging 0s and 1s.

Example:
Find the duals of $z(y+1)$ and $\bar{y} \cdot 0 + (x+z)$

Solution: Interchanging $\cdot$ signs and $+$ signs and interchanging 0s and 1s in these expressions produces their duals.
The duals are: $z + (y \cdot 0)$ and $(\bar{y} + 1)(xz)$ respectively.

Note: There are 2^{2^n} different Boolean functions of degree n.

Example: A Boolean function of degree two is a function from a set with four elements, i.e. from $B = \{0, 1\}$ to $B = \{0, 1\}$.
Hence there are $2^4 = 16$ different Boolean functions of degree two labeled $F_1, F_2, \ldots, F_{16}$.

Table 3 - The 16 Boolean Functions of Degree Two

x	y	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	2^3 = 0's & 1's
1	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	2^2
0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	2^1
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	2^0

Table 4: The Number of Boolean Functions of Degree n.

Degree	Number
1	4
2	16
3	256
4	65,536
5	4,294,967,296 (trillion)
6	18,446,744,073,709,551,616

quintillion 10^{18}

quadrillion 10^{15}

18 ~~sext~~ quintillion, four hundred and ~~forty~~ six ~~quad~~ quadrillion, seven hundred and forty four ~~trillion~~ trillion seventy-three trillion, seven hundred and nine million, five hundred and fifty-one thousand, six hundred and sixteen.

What is Coding Theory?

The study of methods for efficient and accurate transfer of information: detecting and correcting transmission errors.

Some applications of error-correcting codes include correction of errors that occur in information transmitted through the Internet, data stored in computer, and music encoded on a compact disc. Other examples are Telephone lines, Radio, cell phone, Hard drives, Disks, DVDs, solid state memory, ...

ISBN (International Standard Book Number) Codes — ~~An~~ Error-Detecting code

A sequence $x_1 - x_2 x_3 x_4 x_5 - x_6 x_7 x_8 x_9 - x_{10}$, where x_{10} is a check digit chosen so that

$$S = x_1 + 2x_2 + 3x_3 + \cdots + 9x_9 + 10x_{10} = 0 \mod 11.$$

can detect all single and transposition errors.

The first string (x_1) is the country code.
The second string ($x_2 x_3 x_4 x_5$) identifies the publishing company;
The third string ($x_6 x_7 x_8 x_9$) in an ISBN is the number that is given to each book ~~the~~ corresponding company publishes.
The last digit (x_{10}) is a check digit. It is computed from the other digits by a simple rule: multiply the digits of your ISBN by 1, 2, 3, ..., and form the sum: the check digits is the remainder of this sum modulo 11.

ISSN- International standard serial number

Example:

For the ISBN 5 — 783 — 25249 — K we find $1\cdot5 + 2\cdot7 + 3\cdot8 + 4\cdot3 + 5\cdot2 + 6\cdot5 + 7\cdot2 + 8\cdot4 + 9\cdot9$

=200 $\equiv 2 \bmod 11$, hence the complete ISBN is

5 - 783 - 25249 - 2.

The codes of some countries:

Code	Language	Country
0	English	UK, US, Astralia, Newzealand, Canada,
1	English	South Africa, Zimbabwe
2	French	France, Belgium, Canada, Switzerland
3	German	Germany, Austria, Switzerland
4.	Japanese	Japan
5	Russian	Russia, states of the former USSR
7	Chinese	China
88	Italian	Italy, Switzerland
978	English	Nigeria

Note that Switzerland has three official languages.

Check: https://www.everything2.com/title/ISBN+Country+codes

for ISBN codes of some other countries.

GS1 Global standard which is either 978 or 979

Note: Now all ISBNs are 13-digits long.

(GS1) (EAN) - Group (group of Countries) - Publisher - Title - Check digit

European Article Number

The Calculation of 13-digit ISBN: The calculation of an ISBN-13 check digit begins with the first 12 digits of the 13-digit ISBN (excluding the check digit itself). Each digit from left to right is alternately multiplied by 1 or 3, then those products are summed modulo 10 to give a value ranging from 0 to 9. Subtracted from 10 that leaves a result from 1 to 10. A zero (0) replaces a ten (10). For example: The ISBN-13 Check digit of 978-3-11-022406-? is as follows: ? = 9x1 + 7x3 + 8x1 + 3x3 + 1x1 + 1x3 + 0x1 + 2x3 + 2x1 + 4x3 + 0x1 + 6x3 = 89 modulo 10 = 9 ∴ 10 - 9 = 1

∴ the check digit is 1.

(postal code)
Zone Improvement Plan

The ZIP bar code can be found on many bulk-mail envelopes (e.g. on business reply cards). (6)

Each block of 5 lines represents a digit; the correspondence is

1

2

3

4

5

6

7

8

9

10

The sum of all digits including the check digits (i.e. ZIP + 4 + last two digits of address) should be $\equiv 0 \bmod 10$.

The ZIP code has error-correcting capability: observe that each digit is represented by a string of 5 binary digits exactly two of which are nonzero (i.e. ıııll $\equiv$ 00011; ııl ıl $\equiv$ 00101; ...). So if there is a string of 5 binary digits with 2 zeros, implies there must have been an error which can be corrected using the check digit.

For example:

2 6

The first and the last bar define the border; the 50 bars in between encode 10 numbers: ZIP + 4 + Check digit

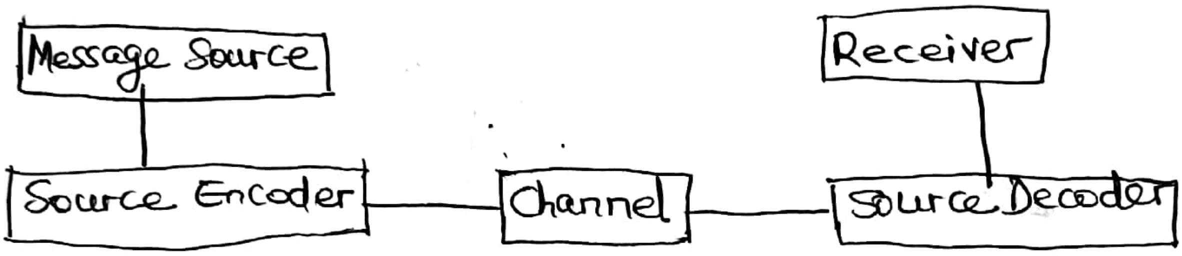
For decoding, we partition the bars between the two border bars into blocks of 5; the first block is ııl ıl representing 00101, that is 2. The whole ZIP code is 26715 − 0181 and the check digit is 0.

In fact, $2+6+7+1+5+0+1+8+0 = 30 \equiv 0 \bmod 10$

Note: Check digits are not always appended to the end of the number.

It is observed that each codeword has exactly two 1s and three 0s; since there are exactly $\frac{5!}{3!\,2!} = \binom{5}{2} = 10$ 5-letter words made out of two 1s and three 0s, the codewords exhaust all such 5-letter words.
If one permutes any codeword (e.g. switching the second and fourth digit), we will get another codeword.

A Simple Communication Model



Scanned by CamScanner

ERROR – CORRECTING CODES

A code is a set of messages called codewords that can be transmitted between two parties.
An "error-correcting" code is a code for which it is sometimes possible to detect and correct errors that occur during transmission of the codewords.

Let $\mathbb{Z}_2^n$ denote the space of vectors of length n over $\mathbb{Z}_2$ in which the codewords are vectors. Hence, codes here will be subsets of $\mathbb{Z}_2^n$ for some n, which may not necessarily be subspace of $\mathbb{Z}_2^n$.
If for instance C is a subspace of $\mathbb{Z}_2^n$ then C is called a <u>linear code</u>. (moreso, a vector space is a linear space)

In order to know if an error occurred during the transmission of a <u>codeword</u> in a code C is by determining if the received vector is in C. The aim is to detect and correct errors in received vectors.
In general, the "nearest neighbour policy" will be used to correct a received vector that contains errors.

<u>Example 1</u>: Consider the code $C = \{(1101), (1100), (1110)\}$ in $\mathbb{Z}_2^4$. Suppose a codeword is transmitted and we receive the vector $r_1 = (0101)$. Checking through C implies that $c = (1101)$ is the codeword from which r_1 differs in the fewest positions. Hence r_1 is corrected to c and assume that the error in r_1 is $e = r_1 - c = (1000)$.

<u>Definition</u>
Let C be a code in $\mathbb{Z}_2^n$. For vectors $x, y \in C$, we define the Hamming distance $d(x,y)$ from x to y to the number of positions in which x and y differ. Hence if $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$, then $d(x,y) = \sum_{h=1}^{\infty} |x_i - y_i|$.

The smallest Hamming distance between any two codewords in a code C is called the minimum distance of C and it is denoted by $d(C)$ or d. For example 1, $d = 1$.

is observed ...

An important part of coding theory is to be able to determine the number of errors that are guaranteed to be uniquely correctable in a given code.

In general let $x \in Z_2^n$ (n - length of each word, code (0,1)) and positive integer r, and $S_r(x) = \{y \in Z_2^n / d(x,y) \le r\}$, which is a ball of radius r around x. Let C be a code with minimum distance d and let t be the largest integer such that $t < \frac{d}{2}$. Then $S_t(x) \cap S_t(y)$ is empty for every pair x, y of distinct codewords in C. If z is a received vector in Z_2^n with $d(u,z) \le t$ for some $u \in C$, then $z \in S_t(u)$ and $z \notin S_t(v)$ for all other $v \in C$. That is, if a received vector $z \in Z_2^n$ differs from a codeword $u \in C$ in t or fewer positions, then every other codeword in C will differ from z in more than t positions. Thus, the nearest neighbour policy will always allow t or fewer errors to be corrected in the code. The code C is said to be t-error correcting.

Example :2: Let $C = \{(11111111), (00011100), (11100000), (00000011)\}$

The minimum distance of C is d = 5. Since t = 2 is the largest integer such that $t < \frac{d}{2}$, then C is 2-error correcting.

If C is a t-error correcting code in $V = Z_2^n$, the problem of determining the number of vectors in V that are guaranteed to be correctable in C has to be tackled.

For any $x \in V$, there are $\binom{n}{i}$ vectors in V that differ from x in exactly i positions. Also, any vector in V that differs from x in i positions will be in $S_t(x)$ provided $i \le t$. Hence the number of vectors in $S_t(x)$ will be $\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}$. The number of vectors in V that differ from one of the codewords in C in t or fewer positions and are consequently guaranteed to be uniquely correctable in C is $|C| \cdot \left[\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}\right]$, since the sets $S_t(x)$ are pairwise disjoint.

i.e. $S_1(x) \neq S_2(x) \neq \cdots$

$|V| = 2^n$ yields a theorem (theorem 1) that gives a bound on the number of vectors in Z_2^n that ~~are~~ are guaranteed to be correctable in a t-error correcting code in Z_2^n. This bound is called the Hamming bound.

<u>Theorem 1</u>: Suppose C is a t-error correcting code in Z_2^n. Then

$$|C| \cdot \left[\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}\right] \leq 2^n.$$

A code C in Z_2^n is said to be perfect if every vector in Z_2^n is guaranteed to be correctable in C. That is, a code C in Z_2^n is perfect if the inequality in theorem 1 with C is an equality.

From example 2:

$$S_5(11111111) = \{00011100,\ 11100000\}.$$

Since $d = 5$ and $t < \frac{d}{2} \Rightarrow t = 2$

$$\therefore S_2(11111111) \cap S_2(00011100) \cap S_2(11100000) = \phi$$

Let $z = (10011111)$ and $u = (11111111)$ with $d(u, t) = t = 2$ then $z \in S_2(11111111)$ and $z \notin S_2(V)$

$V \in$ other vectors in C.

FINITE FIELDS

The characteristic of ring R: Let R be an arbitrary ring. If there excists an integer n such that $n.1 = 1+1+\cdots+1=0$, then the smallest such integer is called the characteristic of the ring; If no such n excists, we say that R has characteristic 0. If R is a finite ring, the characteristic of R is always nonzero.

The rings Z/nZ have characteristic n, so every integer $n \geq 1$ occurs as the characteristic of a ring

Lemma: If E is a subfield of F, then E and F have the same zero and identity elements.

Proof: Let 0 be the zero element of F and $0'$ that of E. Then $0'+0'=0'$ in E (and therefore in F). Moreover, $0+0'=0'$ in F, so comparing yields $0+0'=0'+0'$ and cancelling $0'$ gives $0=0'$.

Similarly, let 1 and $1'$ denote the identities in F and in E, respectively. Then $1'.1'=1'$ in E and therefore in F, as well as $1.1'=1'$ in F. Comparing these equations gives $1.1'=1'.1'$; but in arbitrary rings, we cannot simply cancel $1'$. If F is a field, however, then ~~$1\neq 0$~~ $1'\neq 0'=0$ by the field axioms and cancelling the non-zero factor $1'$ yields $1=1'$ □

Quadratic Excetensions

A field is a commultative ring in which every nonzero element has an inverse. A field with only finctely many elements is called a finite field. For example $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ for primes p is a finite field.

Note that finite fields with q elements are denoted by $\mathbb{F}_q$.

$v = \lambda_1 v_1 + \cdots + \lambda_n v_n$ for $v_1, \ldots, v_n \in V$

If there are elements $v_1, \ldots, v_n \in V$ such that every $v \in V$ can be written in the form

$$v = \lambda_1 v_1 + \cdots + \lambda_n v_n, \quad \ldots (1)$$

then V is said to be finitely generated.

Let F be a finite field, and let p denote its characteristic. The subfield of F generated by 1 is the finite field with p elements, and so we can view F as a vector space over $\mathbb{F}_p$. Let $n = \dim F$ denote its dimension. Then every element of F has a unique representation of the form (1), where $\lambda_1, \ldots, \lambda_n$ run through $\mathbb{F}_p$. This implies that F has exactly p^n elements.

Proposition: If F is a finite field of characteristic p, then F has exactly p^n elements, where $n \geq 1$ is some natural numbers.

Ideals

An ideal I in a ring R is a subring of R with the additional property that $RI = I$ i.e., $ri \in I$ for all $r \in R$ and $i \in I$.

Ideals allow us to define quotient structure. If R is a ring and I is an ideal in R, then we can form the quotient ring R/I. Its elements are the cosets $r + I = \{r + i : i \in I\}$ for $r \in R$, and we have $r + I = r' + I$ if $r - r' \in I$. Addition and multiplication in R/I are defined by

$$(r+I) + (r'+I) = (r+r') + I \text{ and } (r+I)(r'+I) = rr' + I.$$

As for vector spaces, we define kernel and image of ring homomorphisms $f : R \to S$ by

$$\ker f = \{r \in R : f(r) = 0\} \subseteq R \text{ and}$$
$$\operatorname{Im} f = \{f(r) : r \in R\} \subseteq S.$$

If $a, b \in R$ and $a \simeq b$ is defined to mean that $a - b \in Z$. Prove that $\simeq$ is an equivalence relation in R.

Proof: To see that $\simeq$ is reflexive, let $a \in R$. Then $a - a = 0$ and $0 \in Z$, so ~~a~~ $a \simeq a$.

To show that $\simeq$ is symmetric, let $a, b \in R$. Suppose $a \simeq b$. Then $a - b \in Z$ — say $a - b = n$ where $n \in Z$. Then $b - a = -(a - b) = -n$ and $-n \in Z$. Thus $b \simeq a$

Transitivety of $\simeq$ is shown as follows:

Let $a, b, c \in R$. Suppose that $a \simeq b$ and $b \simeq c$. Thus, $a - b \in Z$ and $b - c \in Z$. Suppose $a - b = p$ and $b - c = q$, where $p, q \in Z$. Then $a - c = (a - b) + (b - c) = p + q$. Now $p + q \in Z$; that is $a - c \in Z$. Therefore ~~a~~ $a \simeq c$. $\therefore \simeq$ is an equivalence relation on the set R.

Scanned by CamScanner

FINITE-STATE MACHINES WITH OUTPUT

A finite-state machine $M = (S, I, O, f, g, S_0)$ consists of a finite set S of states, a finite input I, a finite output O, a transition function f that assigns to each state and input pair a new state, an output function g that assigns to each state and input pair an output and an initial state S_0.

A state table can be used to represent the values of the transition function f and the output function g for all pairs of states and input.

Another way to represent a finite-state machine is to use a state diagram, which is a directed graph with labeled edges. Each state is represented by a circle. Arrows labeled with the input and output pair are shown for each transition.

Th.

Example of a Finite-State Machine (A vending machine)

The following is an example of how finite-state machines can be used to model a vending machine, ~~a machine that delays input~~.

Assuming that a vending machine accepts ₦5, ₦10 and ₦20. When a total of ₦30 or more has been deposited, the machine immediately returns the amount in excess of ₦30. When ₦30 has been deposited and any excess refunded, the customer can push a ~~pineapple~~ pink button and receive a pineapple juice or push a blue button and receive an apple juice.

The machine can be in any of seven different states S_i, $i = 0, 1, 2, \ldots 6$ where S_i is the state where the machine has collected ₦$5i$. The machine starts in state S_0, with ₦0 received. The possible inputs are ₦5, ₦10, ~~and~~ ₦20, the pink~~apple~~ button (P) and the blue button (B). The possible outputs are nothing (n), ₦5, ₦10, ₦15, ₦20, ~~₦25~~ a pineapple juice and an apple juice.

Suppose that a student puts in ₦20 followed by ₦20, receives ₦10 back and then pushes the pink button for a pineapple juice. The machine starts in state S_0. The first input is ₦20 which changes the state of the machine to S_4 and gives no output. The second input is ₦20. This changes the state from S_4 to S_6 and gives ₦10 as output. The next input is the pink button which changes the state from S_6 back to S_0 (because the machine returns to the start state) and gives a pineapple juice as its output.

Table 1: State Table for a Vending Machine

State	Next State: Input 5	10	20	P	B	Output: Input 5	10	20	P	B
S_0	S_1	S_2	S_4	S_0	S_0	n	n	n	n	n
S_1	S_2	S_3	S_5	S_1	S_1	n	n	n	n	n
S_2	S_3	S_4	S_6	S_2	S_2	n	n	n	n	n
S_3	S_4	S_5	S_6	S_3	S_3	n	n	5	n	n
S_4	S_5	S_6	S_6	S_4	S_4	n	n	10	n	n
S_5	S_6	S_6	S_6	S_5	S_5	n	5	15	n	n
S_6	S_6	S_6	S_6	S_0	S_0	5	10	20	PJ	AJ

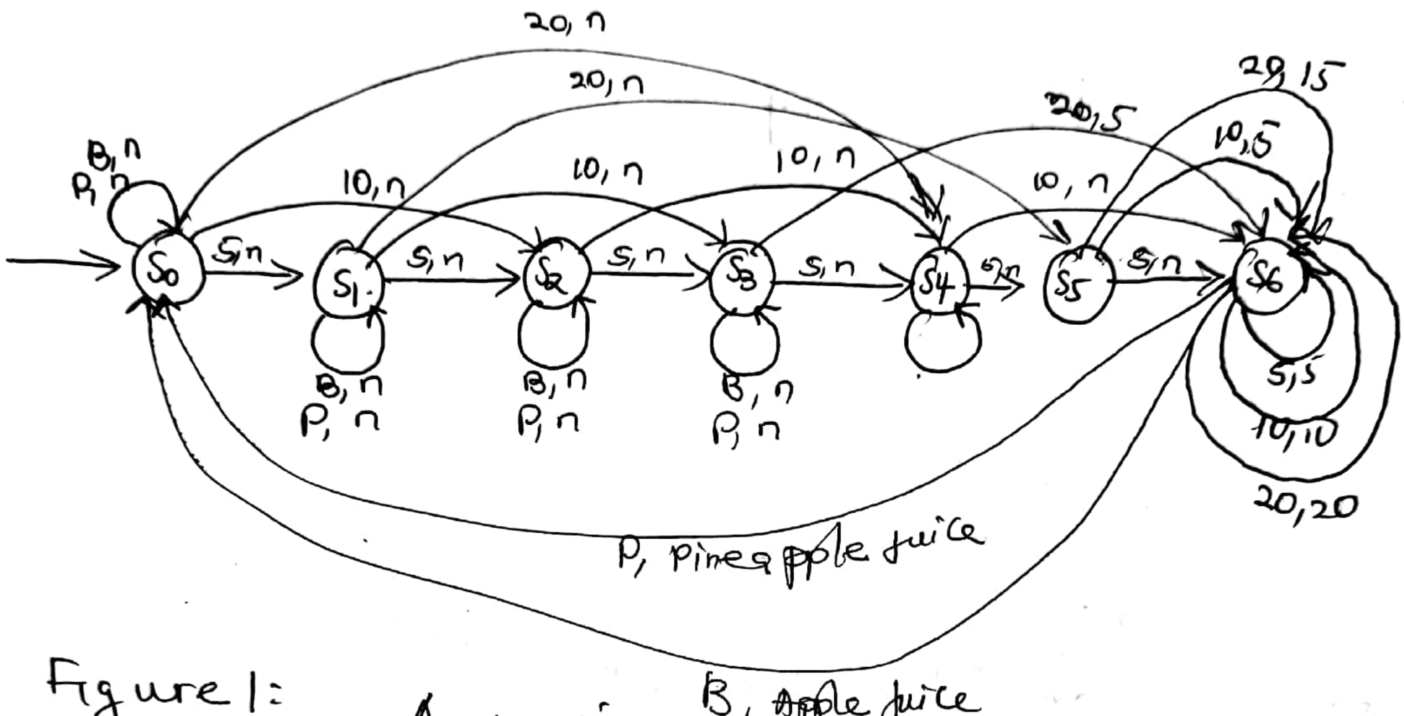


Figure 1: A vending machine

Example: Construct the state diagram for the finite state machine with the state table shown in Table 2.

Table 2

State	f		g	
	Input		Input	
	0	1	0	1
S_0	S_1	S_0	1	0
S_1	S_3	S_0	1	1
S_2	S_1	S_2	0	1
S_3	S_2	S_1	0	0

Solution: The state diagram is

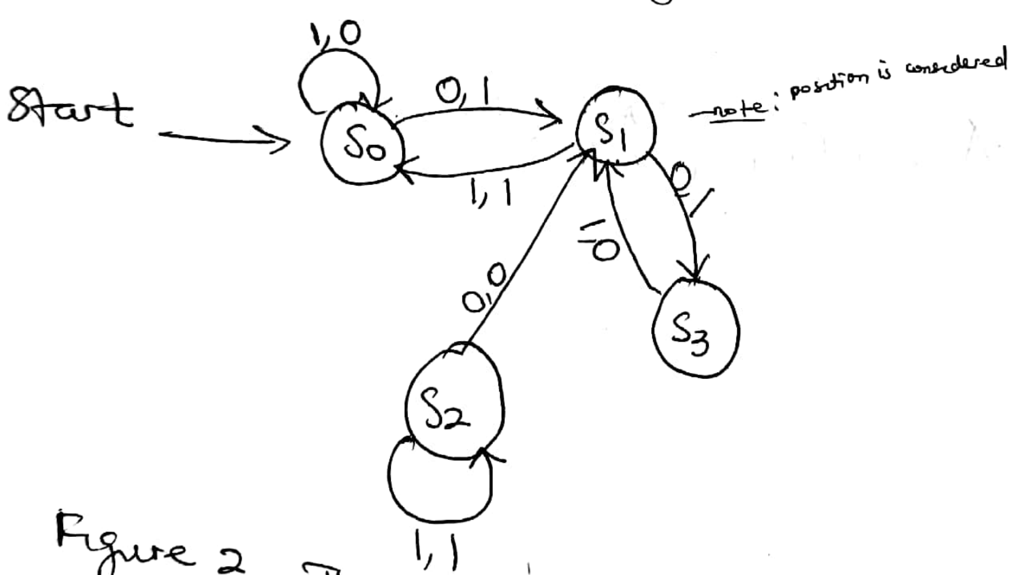


Figure 2 - The State Diagram for the Finite-State Machine in Table 2.

Example: Construct the state table for the finite-state machine with the state diagram shown in figure 3.

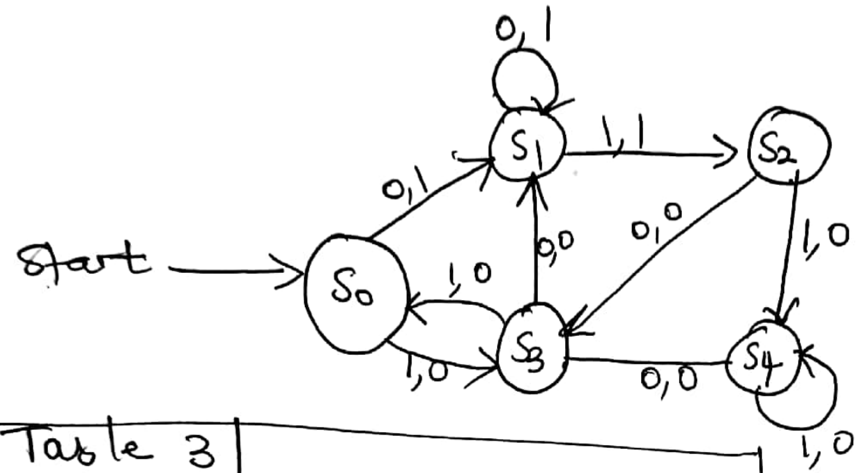


Fig-3. A Finite-State machine

Table 3

	f		g	
	Input		Input	
State	0	1	0	1
s_0	s_1	s_3	1	0
s_1	s_1	s_2	1	1
s_2	s_3	s_4	0	0
s_3	s_1	s_0	0	0
s_4	s_3	s_4	0	0

F

Example of Finite State Machine (FSM) using Vending Machine.

Assumptions:

- Vending Machine only accepts ₦10
- soft drinks cost ₦40
- Flavours are Fanta, Coke and Malt.
- No change return

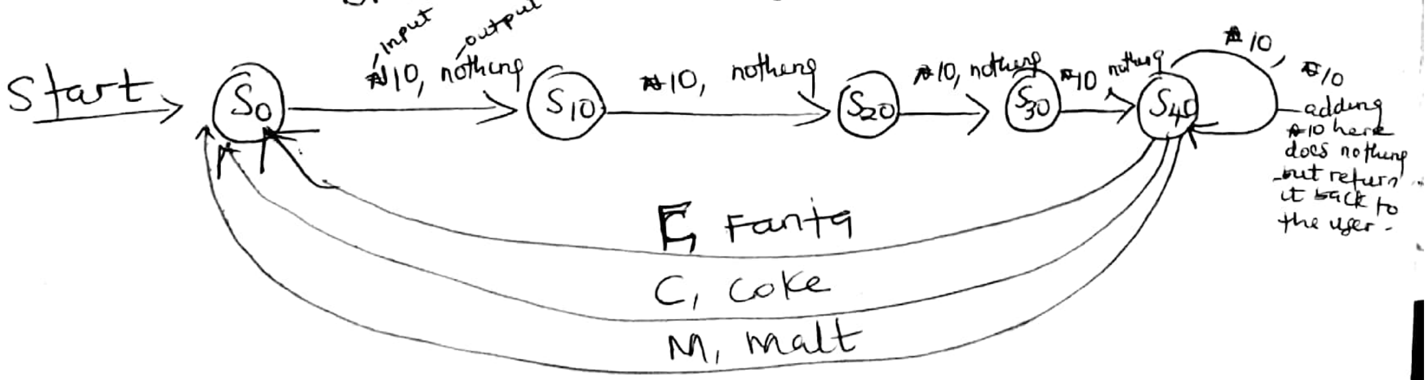
$M = (S, I, O, f, g, S_0)$

What are States? Inputs? Outputs?

States?		Inputs?	Outputs?
S_{40}:	4 · ₦10	press fanta button: Fanta Button	F → Get Fanta
S_{40}:	4 ₦10	press coke button: Coke Button	C → Get Coke
S_{40}:	4 ₦10	press malt button: Malt Button	M → Get Malt
		inserting another ₦10: ₦10	₦10 → Return ₦10
S_{50}:	4 · ₦10		F, C, M → Nothing
S_{30}:	3 · ₦10 fewer money in machine	Any Button	F, C, M → Nothing
S_{20}:	2 · ₦10 "	Any Button	✓ → ✓
S_{10}:	1 · ₦10 "	✓	✓ → ✓
S_0:	0 · ₦10 " no money	✓	

$S = \{S_0, S_{10}, S_{20}, S_{30}, S_{40}\}$ input $I = \{₦10, F, C, M\}$ (F: Pressing Fanta Button) $O = \{$Fanta, Coke, Malt, ₦10, nothing$\}$

Draw State Diagram.



Scanned by CamScanner

Partial Ordering

A relation R on a set S is called a partial ordering or partial order if it is reflexive, antisymmetric and transitive. A set S together with a partial ordering R is called a partially ordered set or poset and is denoted by (S, R). Members of S are called elements of the poset.

Examples:

(1) Show that the "greater than or equal" relation $(\geq)$ is a partial ordering on the set of integers.

Solution: Because $x \geq x$ for every integer x, $\geq$ is reflexive. If $x \geq y$ and $y \geq x$, then $a = b$. Hence, $\geq$ is antisymmetric. Also, $\geq$ is transitive because $x \geq y$ and $y \geq z$ imply $x \geq z$. It follows that $\geq$ is a partial ordering on the set of integers and $(Z, \geq)$ is a poset.

(2) Show that the inclusion relation $\subseteq$ is a partial ordering on the power set of a set S.

Solution: $A \subseteq A$ whenever A is a subset of S, $\subseteq$ is reflexive. It is antisymmetric because $A \subseteq B$ and $B \subseteq A$ imply that $A = B$. Finally, $\subseteq$ is transitive, because $A \subseteq B$ and $B \subseteq C$ imply that $A \subseteq C$. Hence, $\subseteq$ is a partial ordering on $P(S)$ and $(P(S), \subseteq)$ is a poset.

power set of S

Definition: Two elements x and y of a poset $(S, \preceq)$ are called comparable if either $x \preceq y$ or $y \preceq x$. When x and y are elements of S such that neither $x \preceq y$ nor $y \preceq x$, x and y are called incomparable.

Example: In the poset $(Z^+, |)$, are integers 5 and 10 comparable? Are 3 and 8 comparable?

Solution: The integers 5 and 10 are comparable, because $5 | 10$. The integers 3 and 8 are incomparable because $3 \nmid 8$ and $8 \nmid 3$.

P6

The adjective partial is used to describe partial orderings because pairs of elements may be incomparable. When every two elements in the set are comparable, the relation is called a total ordering.

Definition If $(S, \preceq)$ is a poset and every two elements of S are comparable, S is called a totally ordered or linearly ordered set and $\preceq$ is called a total order or a linear order. A totally ordered set is also called a chain.

Example

) The poset $(Z, \leq)$ is totally ordered, because $x \leq y$ or $y \leq x$ whenever x and y are integers.

) The poset $(Z^+, |)$ is not totally ordered.

Hasse Diagrams

The Hasse diagram of $(S, \preccurlyeq)$, named after the twentieth-century German mathematician Helmut Hasse who made use of them.

Example: Draw the Hasse diagram representing the partial ordering $\{(a, b) \mid a \text{ divides } b\}$ on $\{1, 2, 4, 6, 8, 10, 12, 14, 16\}$

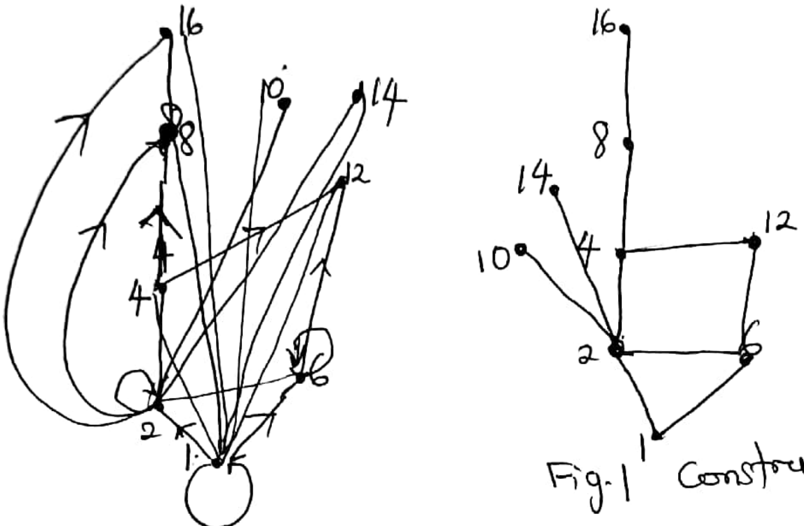


Fig. 1 Constructing the Hasse Diagram of $(\{1, 2, 4, 6, 8, 10, 12, 14, 16\}, |)$

An element of a poset is called maximal if it is not less than any element of the poset. That is, x is maximal in the poset $(S, \preceq)$ if there is no $y \in S$ such that $x \prec y$. Similarly, an element of a poset is called minimal if it is not greater than any element of the poset. That is, x is minimal if there is no element $y \in S$ such that $y \prec x$. Maximal and minimal elements are easy to spot using a Hasse diagram. From fig. 1, 10, 14, 12 and 16 are maximal elements while 1 is the minimal element.

Sometimes there is an element in a poset that is greater than every other element. Such an element is called the greatest element. That is, 16 is the greatest element of the poset $(S, \preccurlyeq)$ in fig. 1. i.e. x is the greatest element of the poset $(S, \preceq)$ if $y \preccurlyeq x$ for all $y \in S$. The greatest element is unique when it exists. An element is called the least element if it is less than all the other elements in the poset. That is, x is the least element of $(S, \preceq)$ if $x \preccurlyeq y$ for all $y \in S$. The least element is unique when it exists. For example, 1 is the least element in figure 1.

It is possible to find an element that is greater than or equal to all the elements in a subset P of a poset $(S, \preccurlyeq)$. If x is an element of S such that $p \preccurlyeq x$ for all $p \in P$, then x is called an upper bound of P. Also if y is an element of S such that $y \preccurlyeq p$ for all $p \in P$ then y is called a lower bound of P.

The element a is called the least upper bound (lub) of the subset P if a is an upper bound that is less than every other upper bound of P. Similarly, the element b is called the greatest lower bound (glb) of P if b is a lower bound of P and $c \preceq b$ whenever c is a lower bound of A.

Lattices

A partially ordered set in which every pair of elements has both a least upper bound and a greatest lower bound is called a lattice.

Exercise: Draw the Hasse diagram for the partial ordering $\{(X, Y) \mid X \subseteq Y\}$ on the power set $P(S)$ where $S = \{a, b, c, d\}$.