- An experiment is a procedure that yields one of a given set of possible outcomes.
- The sample space of the experiment is the set of possible outcomes.
  - an outcome is indivisible
- An event is a subset of the sample space.
- An atomic event is an outcome from the sample space.

Definition 1

If S is a finite non-empty sample space of equally likely outcomes, and E is an event, that is a subset of S, then the probability of E is
$$p(E) = \frac{|E|}{|S|}\ .$$

Theorem 2

Let E be an event in a sample space S. The probability of the event $\bar{E} = S - E$, the complementary event of E, is given by
$$p(\bar{E}) = 1 - p(E)$$

Definition 3 (Kolmogorov axioms, finite version)

Probability on a finite sample space S is a set function $p : \mathrm{pow}(S) \to [0, 1]$ satisfying three axioms

1. for all $E \subseteq S$, $p(E) \geq 0$,
2. $p(S) = 1$,
3. for all $E, F \subseteq S$ such that $E \cap F = \emptyset$, $p(E \cup F) = p(E) + p(F)$

Theorem 4 (Probability properties)

Given the Kolmogorov axioms

- $p(\emptyset) = 0$
- *if* $E \subseteq F$ *then* $p(E) \leq p(F)$
- $p(E - F) = p(E) - p(E \cap F)$
- $p(E \cup F) = p(E) + p(F) - p(E \cap F)$
- $p\left(\bigcup_{i=0}^{n} E_i\right) \leq \sum_{i=1}^{n} p(E_i)$
- *if* $E_1, \ldots, E_n$ *are mutually disjoint, then* $p\left(\bigcup_{i=0}^{n} E_i\right) = \sum_{i=1}^{n} p(E_i)$

4. What is the probability of getting "four of a kind" in a draw of 5 cards from a full deck?
   - sample space: C_5^{52} equally likely draws
   - successful outcome: first pick a rank, C_1^{13} then pick the four cards of this rank C_4^4, then pick a remaining card, C_1^{48}:
   - hence
$$p(\text{"four of a kind"}) = \frac{C_1^{13} C_4^4 C_1^{48}}{C_5^{52}} = \frac{13 \cdot 1 \cdot 48}{2598960} \approx 0.00024$$

Independence Definition:

Two events $A, B \subseteq S$ with $P[A] > 0, P[B] > 0$ are said to be **independent** if and only if
$$P[AB] = P[A]P[B]$$

Conditionally Independence Definition:

A sequence of events $B_1, \ldots, B_n$ are conditionally independent given event A if and only if for every subset of these events, $B_{i_1}, \ldots, B_{i_k}$,
$$P[B_{i_1} \cap B_{i_2} \cap \cdots \cap B_{i_k} | A] = P[B_{i_1}|A] \cdot P[B_{i_2}|A] \cdots P[B_{i_k}|A]$$

Independence Definition (II):

A sequence of events $A_1, \ldots, A_n$ are (mutually) independent if and only if for every subset of these events, $A_{i_1}, \ldots, A_{i_k}$,
$$\Pr[A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_k}] = \Pr[A_{i_1} A_{i_2} \cdots A_{i_k}] = \Pr[A_{i_1}] \cdot \Pr[A_{i_2}] \cdots \Pr[A_{i_k}]$$

Definition of Graph:

A *graph* $G = (V, E)$ consists of V, a nonempty set of *vertices* (or nodes) and E, a set of *edges*. Each edge has either one or two vertices associated with it, called its endpoints. An edge is said to *connect* its endpoints.

Adjacent vertices/incident edge:

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

Neighborhood:

The set of all neighbors of a vertex v of $G = (V, E)$, denoted by $N(v)$, is called the *neighborhood* 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, $N(A) = \cup_{v \in A} N(v)$.

Directed Graph:

A *directed graph* (or *digraph*) (V, E) consists of a nonempty set of vertices V and a set of *directed edges* (or *arcs*) E. Each *directed edge* is associated with an *ordered pair* of vertices. The directed edge associated with the ordered pair (u, v) is said to *start* at u and *end* at v.

Definition:

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.

Degree of a vertex:

The *degree (or valency) 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 $\deg(v)$.

Definition:

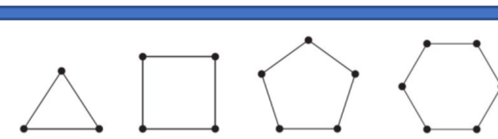
Let $G = (V, E)$ be a graph with *directed* edges. Then
$$\textstyle\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|$$

Theorem:

An *undirected* graph has an *even* number of vertices of *odd* degree.

Cycles:

A *cycle* C_n, $n \geq 3$, 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 $\{v_n, v_1\}$.

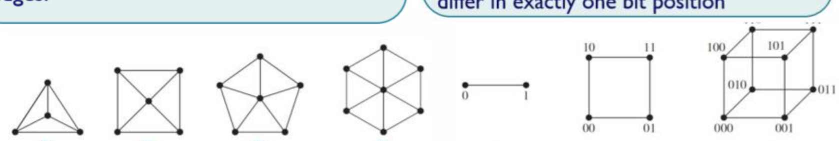


C_3 C_4 C_5 C_6

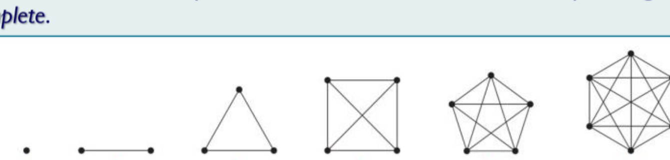
Wheels: We obtain a wheel W_n when we add an additional vertex to a cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n, by new edges.

n-Cubes: An *n-dimensional hypercube*, or *n-cube*, denoted by Q_n, is a graph that has vertices representing the 2^n bit strings of length n. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit position

Complete Graphs: A *complete graph on n vertices*, denoted by K_n, is a simple graph that contains exactly one edge between each pair of distinct vertices. A simple graph for which there is at least one pair of distinct vertex not connected by an edge is called *noncomplete*.



W_3 W_4 W_5 W_6 Q_1 Q_2 Q_3



K_1 K_2 K_3 K_4 K_5 K_6

Definition (path):

A *path* or a *walk* in a graph $G = (V, E)$ is a sequence of vertices $(v_0, v_1, \ldots, v_n)$ such that there exists an *edge* between any two *consecutive* vertices, i.e., $e_i = (v_{i-1}, v_i) \in E$ for $0 < i \leq n$. The path is said to *pass through* the vertices $v_1, \ldots, v_{n-1}$ or traverse the edges $e_1, e_2, \ldots, e_n$. The *length* of the path, is n (i.e., the number of edges).

Graph Isomorphism:

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are *isomorphic* if there exists a *bijection* $f : V_1 \to V_2$ such that $(u, v) \in E_1$ if and only if $\big(f(u), f(v)\big) \in E_2$. The bijection f is called the *isomorphism* from G_1 to G_2, and we use the notation $G_2 = f(G_1)$.

Definition (cycle):

A *cycle* or a *circuit* is a *path* where $n \geq 1$ and $v_0 = v_n$ (i.e., starts and ends at the same vertex). The *length* of the cycle is n (i.e., the number of edges).

Definition (Subgraph):

Given a graph $G = (V, E)$, a subgraph of G is simply a graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq (V' \times V') \cap E$; we denote subgraphs using $G' \subseteq G$.

Definition (connectivity):

An *undirected* graph is connected if there exists a path between *any* two nodes $u, v \in V$ (note that a graph containing a single node v is considered connected via the length 0 path (v)).

An undirected graph that is *not* connected is called *disconnected*.

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

Definition (connectivity):

A *directed* graph is *strongly connected* if there exists a *path* from *any node* u to *any node* v (so is from the node v to the node u).

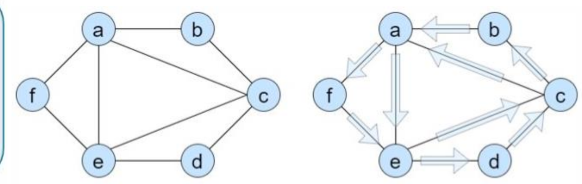
A *directed* graph is *weakly connected* if there is a path between every two vertices in the underlying undirected graph.

- A *connected component* of graph $G = (V, E)$ is a *maximal* connected subgraph; that is, it is a subgraph $H \subseteq G$ that is connected, and any larger subgraph H' (satisfying $H' \neq H, H \subseteq H' \subseteq G$) must be disconnected.
- We may similarly define a *strongly connected component* of a directed graph as a *maximal* strongly connected subgraph.

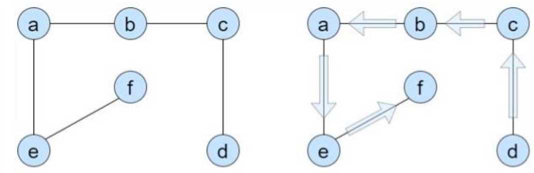
Theorem:

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.

Definition:
A *cycle* that uses *every edge* in a graph *exactly once* is called a ***Euler cycle***.
A *path* that uses *every edge* in a graph *exactly once* is called a ***Euler path***.



Euler Circuit: a-e-c-a-f-e-d-c-b-a



Euler Path: d-c-b-a-e-f

Theorem 1:
A *undirected graph* $G = (V, E)$ has an Euler cycle if and only if G is connected and every $v \in V$ has even degree. Similarly, a *directed graph* $G = (V, E)$ has an Euler cycle if and only if G is *strongly connected* and every $v \in V$ has *equal in-degree and out-degree*.

Theorem 2:
A *undirected graph* $G = (V, E)$ has an ***Euler path***, but not a *Euler cycle*, if and only if the graph is connected and exactly two nodes has an *odd degree*.

Definition (Hamilton Paths):
- A *simple* cycle in a graph G that passes through every vertex exactly once is called a ***Hamilton cycle***.
- A *simple* path in a graph G that passes through every vertex exactly once is called a ***Hamilton path***.

Bipartite Graphs:
A simple graph G is called ***bipartite*** if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2. When this condition holds, we call the pair (V_1, V_2) a ***bipartition*** of the vertex set V of G.

Theorem: G is bipartite if and only if G is connected and has no odd-length cycles.

Matchings:
A ***matching*** in a graph $G = (V, E)$ is a subset M of E such that no vertex in V is on more than one edge in M.
- M is a ***perfect matching*** if every vertex in V is on an edge in M.

Maximum vs Complete matching:
Let $G = (V, E)$ be a *bipartite* graph with partition $V = (V_1, V_2)$. A *maximum matching*, M, in G is a matching with largest possible in size $|M|$, and a *complete matching* from V_1 to V_2 is a matching such that every node in V_1 is matched (assuming $|V_1| \le |V_2|$).

Planar Graph: A graph is called ***planar*** if it can be drawn in the plane without any edges crossing (where a crossing of edges is the intersection of the lines or arcs representing them at a point other than their common endpoint). Such a drawing is called a *planar representation* of the graph.

If a graph is planar, so *will be* any graph obtained by *removing* an edge $\{u, v\}$ and adding a new vertex w together with edges $\{u, w\}$ and $\{w, v\}$.

Such an operation is called an ***elementary subdivision***.

Euler's Formula:
Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

Definition:
The graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called ***homeomorphic*** if they can be obtained from the same graph by a sequence of *elementary subdivisions*.

Corollary 1:
If G is a connected planar simple graph with e edges and v vertices, where $v \ge 3$, then $e \le 3v - 6$.

Theorem (K. Kuratowski):
A graph is nonplanar *if and only if* it contains a subgraph homeomorphic to $K_{3,3}$ or K_5.

Definition (coloring):
A ***coloring*** of a simple graph is the assignment of a *color* to each vertex of the graph so that no two adjacent vertices are assigned the same color.

Lemma:
A graph, G, with at least one edge is bipartite if and only iff $\chi(G) = 2$.

- An even-length closed cycle is 2-colorable: $\chi(C_{even}) = 2$.
- An odd-length closed cycle require 3 colors: $\chi(C_{odd}) = 3$.
- A complete graph K_n requires n colors: $\chi(K_n) = n$.
- All bipartite graph is 2-colorable: $\chi(K_{m,n}) = 2$.

Theorem:
A graph with maximum degree at most k is $(k + 1)$-colorable.

Definition (chromatic number):
The ***chromatic number*** of a graph is the *least number* of colors needed for a coloring of this graph. The chromatic number of a graph G is denoted by $\chi(G)$. (Here χ is the Greek letter *chi*).

The Four Color Theorem:
The *chromatic number* of a planar graph is no greater than four.

- ***Symbol:*** Element such as $a, b, c, \dots, 0, 1, 2, \dots$
- ***Alphabet/Vocabulary:*** Collection of symbols. E.g., $\{a, b\}, \{0, 1, 2, c, d\}, \dots$
- ***String:*** Sequence of symbols, E.g., $aa, bbc, a0b1c2, \dots$
  - Empty string is denoted by λ or ϵ (which is different from an empty set $\emptyset$)
- ***Language:*** Set of strings. E.g., $\{0, 1, 00, 01, 10, 11\}$
  - It is not a natural language or a programming language

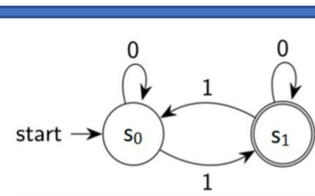


Figure 1: *A simple DFA's state diagram.*

State	Input	
	0	1
$\to s_0$	s_0	s_1
s_1	s_1	s_0

Table 1: *Transition function in a table format.*

An NFA transition function

State	Input	
	0	1
s_0	s_0, s_1	s_3
s_1	s_0	s_1, s_3
s_2	—	s_0, s_2
s_3	s_0, s_1, s_2	s_1

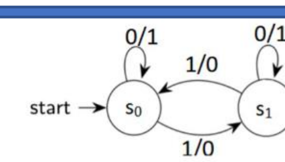


Figure 2: *An FSM with output.*

Current State	f - Input		g - Input	
	0	1	0	1
$\to s_0$	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

Finite-State Automaton:
A finite-state automaton $M = (S, I, f, s_0, F)$ consists of a finite set S of states, a finite input alphabet I, a transition function f that assigns a next state to every pair of state and input (so that $f : S \times I \to S$), an initial or start state s_0, and a subset F of S consisting of final states (or accepting states).

Definition (NFA):
A ***nondeterministic*** finite-state automaton $M = (S, I, f, s_0, F)$ consists of a set S of states, an input alphabet I, a transition function f that assigns *a set of states* to each pair of state and input (so that $\boldsymbol{f: S \times I \to \wp(S)}$), a starting state s_0, and a subset F of S consisting of the final states.

Definition: Let $M = (S, I, O, f, g, s_0)$ be a finite-state machine and $L \subseteq I^*$. We say that M *recognizes* (or *accepts*) L if an input string x belongs to L *if and only if* the *last output bit* produced by M when given x as input is a 1.

Definition: A string x is said to be ***recognized*** or ***accepted*** by the machine $M = (S, I, f, s_0, F)$ if it takes the initial state s_0 to a final state, that is, $f(s_0, x)$ is a state in F. The language recognized or accepted by the machine M, denoted by $L(M)$, is the set of all strings that are recognized by M.

Definition: Two finite-state automata are called ***equivalent*** if they recognize the *same* language.

Theorem: If the language L is recognized by a nondeterministic finite-state automaton M_0, then L is also recognized by a deterministic finite-state automaton M_1.

Finite-State Machine 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 alphabet* I, a *finite output alphabet* O, a *transition function* f that assigns a next state to every pair of state and input ($f : S \times I \to S$), an *output function* g that assigns to each state and input pair an output ($g : S \times I \to O$), and an *initial* state s_0.

Definition I: Suppose that A and B are subsets of V^*, where V is an alphabet. The *concatenation* of A and B, denoted by AB, is the set of all strings of the form xy, where x is a string in A and y is a string in B.

Definition II (Kleene Closure): Suppose that A is a subset of V^*. Then the *Kleene closure* of A, denoted by A^*, is the set consisting of *concatenations* of arbitrarily *many strings* from A. That is, $A^* = \bigcup_{k=0}^{\infty} A^k$.

Definition: III The ***regular expressions*** over a set I are defined recursively by:
- the symbol $\emptyset$, i.e., *an empty string*, is a regular expression;
- the symbol λ, i.e., the set $\{\emptyset\}$, is a regular expression;
- the symbol x is a regular expression whenever $x \in I$;
- the symbols $(\boldsymbol{AB})$, $(\boldsymbol{A} \cup \boldsymbol{B})$, and $\boldsymbol{A}^*$ are regular expressions whenever $\boldsymbol{A}$ and $\boldsymbol{B}$ are regular expressions.

Kleene's Theorem A set is *regular if and only if* it is *recognized* by a finite-state automaton.

TABLE 1

Expression	*Strings*
$\mathbf{10^*}$	a 1 followed by any number of 0s (including no zeros)
$\mathbf{(10)^*}$	any number of copies of 10 (including the null string)
$\mathbf{0 \cup 01}$	the string 0 or the string 01
$\mathbf{0(0 \cup 1)^*}$	any string beginning with 0
$\mathbf{(0^*1)^*}$	any string not ending with 0