Counting

Six Basic Rules: Conceive of counting as a process (precisely defined).

Multi-Step (Product Rule): #(total) = product of #(each step)

Multi-Case (Sum Rule): #(total) = sum of #(each case)

Inclusion-Exclusion: **#**(AorB) = **#**(A) + **#**(B) − **#**(A&B)

Complements: #(not A) = #(all) − **#**(A)

(Decision) Tree: Combine Multi-Step and Multi-Case.

Special Order: #(total) = #(in special order) * #(re-orderings)

Examples: Bit strings, IP Addresses, Rubik's Cube, Poker Hands.

==========

Pigeon-hole Principle: if n objects are placed in k boxes ($n \geq k$), then at least one box contains $\uparrow n/k \uparrow$ (= n/k rounded up) objects.

Example: Among **100** people, there must be **9** with the same birth month.

Permutations/Combinations: #ways **k** out of **n** objects are selected:

Permutation (Ordered) **"n pick k"**	**Combination** (UnOrdered) **"n choose k"**
$_nP_k = P(n,k) = \frac{n!}{(n-k)!}$	$_nC_k = C(n,k) = \frac{n!}{(n-k)!\bullet k!}$

Identities: $_nP_0 = 1$ $_nP_1 = n$ $_nP_n = n!$

$_nC_k = {}_nC_{n-k}$ $_nC_0 = 1 = {}_nC_n$ $_nC_1 = n = {}_nC_{n-1}$

Pascal's Triangle: $_{n+1}C_k = {}_nC_k + {}_nC_{k-1}$

Binomial Theorem: $(a+b)^n = \sum_k {}_nC_k\, a^k b^{n-k}$

Corollary: $2^n = \sum {}_nC_k$ **(a=1=b)** $0 = \sum (-1)^k\, {}_nC_k$ **(a=1=-b)**

Propositional & Predicate (Formal) Logic

Propositions are declarative sentences which are definitely either T or F.

Propositional variables, **p, q**, … represent unspecified propositions.

The basic **propositional operators** are:

¬	**NOT**	∧	**AND , &**	∨	**OR**
→	**IMPLIES , IF THEN**	↔	**IFF , IF AND ONLY IF**		

Basic Truth Tables:

p	¬p
T	F
F	T

p	q	p ∧ q
T	T	T
T	F	F
F	T	F
F	F	F

p	q	p ∨ q
T	T	T
T	F	T
F	T	T
F	F	F

p	q	p → q
T	T	T
T	F	F
F	T	T
F	F	T

p	q	p ↔ q
T	T	T
T	F	F
F	T	F
F	F	T

Precedence Order: ¬ ∧ ∨ → ↔ **(& parentheses)**

Also: + XOR, "either or" **| NAND "not and"** **↓ NOR "not or"**

Given a proposition of the form: **p → q** : (**positive);**

q → p is the **converse ;**

¬p → ¬q is the **inverse ;**

¬q → ¬p is the **contrapositive .**

English Equivalents for : "p ↔ q"

p if and only if q | if p, then q, and conversely | p iff q

p is necessary and sufficient for q (and vice versa)

English Equivalents for : "p → q "

p implies q | if p, then q | if p, q

p only if q | q if p | q when p

q unless not p | q whenever p | q follows from p

p is sufficient for q | q is necessary for p

Logical Equivalences

$\neg\neg p \equiv \neg(\neg p) \equiv p$	Double Negation	
$p \wedge T \equiv p$	$p \vee F \equiv p$	Identity
$p \vee T \equiv T$	$p \wedge F \equiv F$	Domination
$p \vee p \equiv p$	$p \wedge p \equiv p$	Idempotent
$p \vee \neg p \equiv T$	$p \wedge \neg p \equiv F$	Negation
$p \wedge q \equiv q \wedge p$	$p \vee q \equiv q \vee p$	Commutative
$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	$(p \vee q) \vee r \equiv p \vee (q \vee r)$	Associative
$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	Distributive
$\neg(p \vee q) \equiv (\neg p \wedge \neg q)$	$\neg(p \wedge q) \equiv (\neg p \vee \neg q)$	DeMorgan
$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$	Absorption
$p \rightarrow q \equiv \neg p \vee q$	$p \wedge q \equiv \neg(p \rightarrow \neg q)$	
$p \rightarrow q \equiv \neg q \rightarrow \neg p$		Contrapositive
$\neg(p \rightarrow q) \equiv p \wedge \neg q$		Conditional DeMorgan
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$		Biconditional DeMorgan
$p \leftrightarrow q \equiv q \leftrightarrow p \equiv (p \rightarrow q) \wedge (q \rightarrow p)$		Necessary & Sufficient

Predicates (Propositional Functions)

P(x) "proposition" with a variable, denoted **x** , in some Domain

P(x,y,…) "proposition" with 2 or more variables in 1 or more Domains

When values are assigned to the variables in a propositional function, it "evaluates to" – yields as its output - a proposition.

Quantifiers applied to propositional functions yield propositions.

$\forall$**xP(x)** $\equiv$ for every x (for all x), P(x) [is true]

$\exists$**xP(x)** $\equiv$ for some x (there exists an x), P(x) [is true]

$\exists$**!xP(x)** $\equiv$ for a unique x (there exists a unique x), P(x) [is true]

Quantifiers have highest precedence

Quantifiers bind variables within a certain scope, which extends until the first "possible" end which defines a propositional function.

DeMorgan for Quantifiers

$\neg\forall$**x(P(x))** $\equiv$ $\exists$**x($\neg$P(x))** $\neg\exists$**x(P(x))** $\equiv$ $\forall$**x($\neg$P(x))**

Associative for Quantifiers

$\forall$**x(P(x)$\wedge$Q(x))** $\equiv$ $\forall$**xP(x)** $\wedge$ $\forall$**xQ(x)** **(not for $\vee$)**

$\exists$**x(P(x)$\vee$Q(x))** $\equiv$ $\exists$**xP(x)** $\vee$ $\exists$**xQ(x)** **(not for $\wedge$)**

$\forall$**x(P(x)** $\rightarrow$ **Q(x))** $\equiv$ $\forall$**x($\neg$P(x)** $\vee$ **Q(x)**

But $\neg$ **(** $\forall$**x(P(x)$\rightarrow$Q(x))** $\equiv$ $\forall$**xP(x)** $\rightarrow$ $\forall$**xQ(x))**

Nested Quantifiers

$\forall$**x$\forall$yP(x,y)** $\equiv$ $\forall$**y$\forall$xP(x,y)** $\exists$**x$\exists$yP(x,y)** $\equiv$ $\exists$**y$\exists$xP(x,y)**

But: $\neg$**($\forall$x$\exists$yP(x,y)** $\equiv$ $\forall$**y$\exists$xP(x,y))** $\neg$**($\forall$x$\exists$yP(x,y)** $\equiv$ $\exists$**y$\forall$xP(x,y))**

Formal Logical Inference

Rule	Tautology	Latin Name
p $p \rightarrow q$ $\therefore q$	$[p \wedge (p \rightarrow q)] \rightarrow q$	Modus Ponens
$\neg q$ $p \rightarrow q$ $\therefore \neg p$	$[\neg q \wedge (p \rightarrow q)] \rightarrow \neg p$	Modus Tollens
$p \rightarrow q$ $q \rightarrow r$ $\therefore p \rightarrow r$	$[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$	Syllogism
$p \vee q$ $\neg p$ $\therefore q$	$[(p \vee q) \wedge \neg p] \rightarrow q$	Disjunctive Syllogism
$p \vee q$ $\neg p \vee r$ $\therefore q \vee r$	$[(p \vee q) \wedge (\neg p \vee r] \rightarrow (q \vee r)$	Resolution
p $\therefore p \vee q$	$p \rightarrow (p \vee q)$	Addition
$p \wedge q$ $\therefore p$	$(p \wedge q) \rightarrow p$	Simplification
p q $\therefore p \wedge q$	$[(p) \wedge (q)] \rightarrow (p \wedge q)$	Conjunction

Types of Logical Proof

Direct Proof: To prove

(i) p→q is true: assume p is true, argue why q must be true.

(ii) ∀x P(x)→Q(x): assume for any x, P(x), argue why Q(x).

Example: If integer **n** is odd, then $\mathbf{n^2}$ is odd.

Example: The difference between consecutive perfect squares is odd.

Contraposition Proof (Indirect): To prove

(i) p→q is true: assume ¬q is true, argue why ¬p must be true.

(ii) ∀x P(x)→Q(x): assume for any x, ¬Q(x), argue why ¬P(x);
or assume for no x, Q(x), argue why no x, P(x).

Example: If $\mathbf{n^2}$ is even, then **n** is even.

Contradiction Proof (Indirect): To prove

p is true: assume **¬p** and some cleverly chosen **r** are both true, argue why this implies **r∧¬r** are true (a contradiction), so **¬p** is false.

Example: $\sqrt{\mathbf{2}}$ is irrational.

Exhaustive Proof: – all inclusive **case by case** analysis.

Without Loss of Generality – use a simple case which subsumes all.

Example: Integer **n** is odd if and only if **n+1** is even.

Existence Proof: - may be **direct** (find an example) or **indirect.**

Example: There exist irrational numbers, **x,y,** with $\mathbf{x^y}$ rational.

Sets, Functions and Cardinality

A **set** is a collection of **elements.** An element is a member of a set.

x ε S means **x** is an element of **S**.

Sets are lists **{…}** or Truth Sets **{x | P(x)}** or one of these:

Null Set = ∅ = **{ }** has no elements**;**

Naturals $\mathbb{N}$**;** $\mathbb{N}_n$ **= { 1, 2, 3, …, n };**

Integers $\mathbb{Z}$**;** $\mathbb{Z}_n$ **= { 0, 1, 2, …, n – 1 };**

Rationals $\mathbb{Q}$**;** **Reals** $\mathbb{R}$ **;** **Complex #s** $\mathbb{C}$ *;*

$\mathbb{N}^+ = \mathbb{Z}^+, \mathbb{Q}^+, \mathbb{R}^+$ just the positive numbers in each.

Given two sets, **A** and **B** :

Equal: $A = B \equiv (x \in A \leftrightarrow x \in B)$

Subset: $A \subseteq B \equiv (x \in A \rightarrow x \in B)$

Proper Subset: $A \subset B \equiv (A \subseteq B \wedge \neg A = B)$

The set of all subsets of **S** is $P(S) = 2^S$ = **Power Set** : $\forall S\ (\varnothing \in P(S))$.

Given two sets, **A** and **B**:

Union: $A \cup B = \{ x \mid x \in A \vee x \in B \}$

Intersection: $A \cap B = \{ x \mid x \in A \wedge x \in B \}$

Difference: $A - B = \{ a \mid a \in A \wedge \neg a \in B \}$

Complement (universal U): $\overline{A} = A^c = U - A$

Symmetric Difference: $A \Delta B = (A - B) \cup (B - A)$

Product (ordered pairs): $A \times B = \{ (a,b) \mid a \in A \wedge b \in B \}$

De Morgan, Associative and Distributive Laws (prove with Venn Diagrams):

$(A \cap B)^c = (A^c \cup B^c)$ $\quad$ $(A \cup B)^c = (A^c \cap B^c)$

$(A \cup B) \cup C = A \cup (B \cup C)$ $\quad$ $(A \cap B) \cap C = A \cap (B \cap C)$

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $\quad$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

For **X finite**, the **cardinality** of **X** is: **|X| = #(X) =** number of elements in X.

$\#(A \cup B) = \#(A) + \#(B) - \#(A \cap B)$

$\#(A - B) = \#(A) - \#(A \cap B)$

$\#(P(A)) = 2\text{^}\#(A) = 2^{\#(A)}$

A **relation** from a set **X** to a set **Y**, **R:X —> Y**, is a rule which assigns to some x's some y's. Alternatively, $\mathbf{R \subseteq X \times Y}$.

X is called the **Domain**. **Y** is called the **Codomain**.

A **function** from a set **X** to a set **Y**, **f:X —> Y**, is a relation which assigns to **every x precisely one y=f(x).** I.e. $\forall \mathbf{x \varepsilon X}\ \exists \mathbf{! y \varepsilon Y}\ (\ \mathbf{y=f(x)}\ )$.

Graph of y=f(x) is $\mathbf{G(f) = \{\ (x,y)\ \varepsilon\ X \times Y \mid y = f(x)\ \} \subseteq X \times Y}$

Image of A $\subseteq$ X is $\mathbf{f(A) = \{\ y \varepsilon Y \mid \exists x \varepsilon A\ (y=f(x))\ \}}$. **Range of f** is **f(X).**

Preimage of B $\subseteq$ Y is $\mathbf{f^{-1}(B) = \{\ x \varepsilon X \mid \exists y \varepsilon B\ (y=f(x)\ )\ \}}$.

f:X $\rightarrow$ Y is: **injective or 1-1** iff $\forall x_1, x_2\ \varepsilon\ X\ (\ x_1 \neq x_2 \rightarrow f(x_1) \neq f(x_2)\ )$;

surjective or onto iff $\forall y \varepsilon Y\ \exists x \varepsilon X\ (\ y=f(x)\ )$ iff $f(X)=Y$;

bijective iff it is injective and surjective.

f:X—>Y and **g:Y—>Z,** the **composition** is**: g$\circ$f: X—>Z is g(f(x)).**

Theorem: f:X—>Y, g:Y—>Z, h**:Z—>W: (h$\circ$g) $\circ$f(x) = h$\circ$ (g$\circ$f)(x) =h(g(f(x)));**

f: X—>Y 1-1 $\wedge$ **g:Y—>Z 1-1** $\rightarrow$ **g$\circ$f:X—>Z 1-1**;

f: X—>Y onto $\wedge$ **g:Y—>Z onto** $\rightarrow$ **g$\circ$f:X—>Z onto**.

f:X—>Y and **g:Y—>X** , are **inverses** (Write **g** as $\mathbf{f^{-1}}$) if

$\forall \mathbf{x\ \varepsilon\ X\ (\ g \circ f(x) = x\ )} \wedge \forall \mathbf{y\ \varepsilon\ Y\ (\ f \circ g(y) = y\ )}$.

Theorem: (i) $\exists\ \mathbf{f^{-1}} \leftrightarrow \mathbf{f}$ is **bijective**; (ii) $\mathbf{(f \circ g)^{-1} = g^{-1} \circ f^{-1}}$.

Numeric Valued Functions

For two functions, **f,g:X—>Y**, with **Y** a set of numbers, define:

$\mathbf{f \pm g(x) = f(x) \pm g(x)}$ $\mathbf{f \cdot g(x) = f(x) \cdot g(x)}$ $\mathbf{f/g(x) = f(x)/g(x)}$.

f: $\mathbb{R}$ —> $\mathbb{R}$ is **increasing** iff $\forall \mathbf{x_1, x_2:\ x_1 < x_2\ \rightarrow f(x_1) < f(x_2)}$.

Increasing functions, **f,g: $\mathbb{N}$ —> $\mathbb{R}^+$**, share **order of growth** $\mathbf{\Theta(f) = \Theta(g)}$

iff $\dfrac{\mathbf{f(n)}}{\mathbf{g(n)}}$ approaches a non-zero limit as n approaches infinity.

Standard Growths: $\mathbf{\Theta(1)=\Theta(\log_n(n))=\Theta(\sqrt[n]{n}) < \Theta(\log_3(n))=\Theta(\log_2(n)) < \Theta(\sqrt[3]{n}) < \Theta(\sqrt{n}) < \Theta(n) < \Theta(n^2) < \Theta(n^3) < \Theta(2^n) < \Theta(3^n) < \Theta(n!) < \Theta(n^n)}$.

Order of growth is used to measure the complexity or runtime of algorithms.

Important functions with codomain the Integers, $\mathbb{Z}$:

n! = n(n-1)(n-2)…(2)(1) = **n factorial**

$\uparrow x \uparrow$ **= <u>ceiling</u> of x** = x rounded up = least integer ≥ x

$\downarrow x \downarrow$ **= <u>floor</u> of x** = x rounded down = largest integer ≤ x

<u>Special Types of Functions:</u>

<u>Sequence</u>: a_k: $\mathbb{N}$ –> $\mathbb{R}$ or a_k: $\mathbb{N}_n$ –> $\mathbb{R}$

<u>Series (for a sequence)</u>: $\sum_{k=0} a_k = a_0 + a_1 + \dots$, or $\sum_{k=1} a_k$ or $\sum_{k} a_k$

<u>Matrix</u> (n by m)**: M_{ij} : $\mathbb{N}_n \times \mathbb{N}_m$ –> $\mathbb{R}$, i for rows, j for columns.**

<u>Matrix multiplication</u> (n by p × p by m) **is:** $(M \times N)_{ij} = \sum_{k} M_{ik} \cdot N_{kj}$

<u>Counting or "How Many?"</u>

Definition: ∃ bijection **X<–>Y** iff **X** and **Y** have the same **<u>cardinality:</u>**

X is **finite** iff **#(X) = #($\mathbb{N}_n$) = n;**

X is **<u>countably infinite</u>** iff **#(X) = #($\mathbb{N}$) = #($\mathbb{Z}$) = #($\mathbb{Q}$);**

the real numbers, $\mathbb{R}$, are uncountably infinite (Cantor).

Given a set, **X** , with **#(X) = n**:

#(P(X)) = #(subsets of X) = 2^n ;

#{1-1 functions X –> X} = n! ;

#{ functions X –> X} = n^n ;

#{ relations X –> X} = $2^{n\wedge 2}$.

<u>Operations and Abstract Algebra:</u>

A **<u>(binary) operation</u>** on a set **G** is a function, **° : G × G → G .**

A **<u>group</u>** is a set with an associative operation with identity and inverses.

A **<u>ring</u>** is a set with two associative operations, **"+,•"**, with identities, **"0,1"**:
where **"•"** distributes over **"+" (** r•(s+t) = r•s + r•t & (s+t)•r = s•r + t•r **)**;
and under **"+"** it is a **commutative** group.

A **<u>field</u>** is a **ring** where under **"•"** it is a **commutative** group excluding **"0"**.

Integers and Divisibility

"a divides b" iff **a|b** iff $\exists$**k (ak = b).**

Division Theorem: $\forall$**a** $\forall$**d>0** $\exists$**!q** $\exists$**!r (0≤r<d)** $\wedge$ **(a = dq + r).** Define **q=**quotient **= "a div-d"** and **r=**remainder **= "a mod-d"** (see algorithm).

Congruence: a ≡ b mod-d $\leftrightarrow$ **(a mod-d) = (b mod-d)** $\leftrightarrow$ **d | (a-b).**

Laws of Congruence: If **a ≡ b mod-m** and **c ≡ d mod-m,** then

(i) **(a+c) ≡ (b+d) mod-m**; (ii) **(a•c) ≡ (b•d) mod-m**;

(iii) **(a^c) ≡ (b^c) mod-m for c > 0**; BUT:

(iv) **$c^a \not\equiv c^b$ mod-m** and **$a^c \not\equiv b^d$ mod-m.**

Corollary: $\mathbb{Z}_m$ = **{0,1,…,m-1}** is a **Commutative Ring** under

Addition (+) and multiplication (•) **"mod-m"** , for **m > 1** ;

Always with additive inverses: **k + -k ≡ 0 mod-m (-k = m-k)** ;

Sometimes with multiplicative inverses: **$k \bullet k^{-1} \equiv 1$ mod-m**.

Base b: Using **b > 1** "digits", **{0,1, … ,b-1}**, express integers: $\Sigma\, d_k b^k$.

Convert $()_b$ to $()_{10}$: do the arithmetic implied by the sum above.

Convert $()_{10}$ to $()_b$: repeatedly divide by b, use remainders.

Convert binary=$()_2$ to Octal=$()_8$ to hex=$()_{16}$: group bits by 3's or 4's.

Arithmetic Algorithms base b are "the same" as usual ones base 10.

(http://www.javascripter.net/math/calculators/100digitbigintcalculator.htm)

Primes: **p prime iff (p>1)** $\wedge$ **($\forall$n,m (p=nm** $\rightarrow$ **p=n** $\vee$ **p=m)) ;**

Mersenne prime = $2^p - 1$, with p prime;

n is perfect iff n is sum of its proper divisors (6=1+2+3).

P-Theorems: (i) Every integer > 1 is the product of primes (see algorithm);

(ii) There are an Infinity of Primes with **#(primes ≤ n) ≈ $^n/_{\ln(n)}$, n** $\rightarrow\infty$;

(iii) If **p prime** $\wedge$ **p|nm,** then **p|n** $\vee$ **p|m** ;

(iv) If **p≠q primes** $\wedge$ **n≡1 mod-p** $\wedge$ **n≡1 mod-q,** then **n≡1 mod-pq** ;

(v) $(2^p - 1)2^{p-1}$ is **Perfect** iff **$(2^p - 1)$** is a **Mersenne prime.**

Greatest Common Divisor & Least Common Multiple:

greatest common divisor = gcd(a,b) = '∩'(prime factors)
least common multiple = lcm(a,b) = 'U'(prime factors)
a,b are **relatively prime** iff **gcd(a,b) = 1.**

G-Theorems: (i) ab = gcd(a,b)•lcm(a,b) & gcd(a,b) = gcd(b,a mod-b);
(ii) For **a,b > 0**, $\mathbf{a^{-1}}$**mod-b** exists iff **gcd(a,b)** = 1;
(iii) $\mathbb{Z}_p$ = **{0,1,…,p-1}** is a **Field (** $\exists \mathbf{x^{-1}}$ **mod-p for x≠0)** ;
and hence may solve equations by usual algebraic methods**.**

Unsolved Conjectures: Infinity of Mersenne Primes?
Odd Perfect Numbers? Twin Primes? Goldbach Conjecture?

Fermat's Little Theorem: If **p prime** and **¬p|a ,** then $\mathbf{(a^{p-1} \equiv 1) mod\text{-}p.}$
(Trick: (241^{241}) mod11 = $((241^{10})^{24})$ mod11•(241) mod11 = 1•10 =10.)

Encryption by Simple Cipher:

(i) blocks of one or more characters assigned numbers**:** $\mathbb{Z}_n$;
(ii) encrypt by: **X → C = X+K mod-n** (K is private key);
(iii) decrypt by: **X → X-K mod-n .**

Public Key Encryption (Cocks, RSA=Rivest, Shamir, Adleman):

Let **p,q primes**, **n=pq** , **e** with $\exists \mathbf{d=e^{-1}}$ **mod((p-1)(q-1))** , and **X<n** :
if $\mathbf{C = X^e}$ **mod-n ,** then $\mathbf{X = C^d}$ **mod-n .**

Proof: $\mathbf{C^d}$ **mod-n** = $\mathbf{(X^e}$ **mod-n)**$^{\mathbf{d}}$ **mod-n =** $\mathbf{X^{ed}}$ **mod-n** (by LofC(iii)) **=**
$\mathbf{X^{1+k(p-1)(q-1)}}$ **mod-pq** (by n=pq & d = e^{-1} mod(p-1)(q-1)) **=**
X•($\mathbf{X^{k(p-1)(q-1)}}$ **mod-pq) mod-pq =** (by LofC(ii) & X<n)
(X•1) mod-n (by FLT for p-1 & q-1 and PT(iv)) **= X** (X<n)**.** ▌

Protocol: R:= Receiver and **S:=** Sender
R: Choose public key **p,q,e.** Transmit only **n (=pq)** & **e** to **S:**
S: Break message into blocks, **B**, with maximum value **< n**
S: Encrypt block as $\mathbf{C = B^e}$ **mod n** & transmit **C** to **R:**
R: Decrypt block as $\mathbf{B = C^d}$ **mod n (** $\mathbf{d=e^{-1}}$**mod(p-1)(q-1))**
and re-assemble blocks into message.

Induction and Recursion

Induction: To prove a predicate, P(n), true for all natural numbers:

(i) Prove(or Verify): **P(1)** is true (recommend do P(2) too);

(ii) Prove: **P(k) → P(k+1)** for **k ≥ 1**

(Assume P(n) is true and then use this to show P(n+1) is true, too).

Proved: $\sum_{1} k = \frac{n(n+1)}{2}$ $\sum_{1} k^2 = \frac{n(n+1)(2n+1)}{6}$

$\sum_{1} k^3 = \frac{n^2(n+1)^2}{2^2}$ $\sum_{0} r^k = \frac{r^{n+1} - 1}{r - 1}$

Strong Induction:

(i) Prove(or Verify): **P(1)** is true;

(ii) Prove: **P(1) ∧ P(2) ∧ … ∧ P(k) → P(k+1) for** k ≥ 1**.**

Bell's Theorem: For any lattice polygon (vertices are integers) in the plane, let **B** = # boundary lattice points, **I** = # interior lattice points, then **Area = I + B/2 – 1** .

Sequences: A sequence is a function: $a_n : \mathbb{N} \rightarrow \mathbb{R}$

Explicit Definition: $a_n = f(n)$

Recursive Definition: a_0 **or** a_1 is specified, $a_{n+1} = f(a_n)$

Algorithmic Definition: a procedure to calculate the terms

Example: **Arithmetic:** $a_{n+1} = a_n + d$; (d = common difference)

Geometric: $g_{n+1} = r \cdot g_n$; (r = common ration)

Fibonacci: $f_{n+1} = f_n + f_{n-1}$; (f_1=?, f_2=?, two starts)

Logistic: $p_{n+1} = r \cdot p_n \cdot (1 - p_n)$; (fecundity rate r < 4)

Pseudorandom: $x_{n+1} = (a \cdot x_n + b) \bmod m$.

Recursive sequences often approach **limits** as n →∞ :

Geometric: $g_n \rightarrow 0$ **for** $|r| < 1$

Logistic: $p_n \rightarrow (r - 1)/r$ **for** $1 \le r \le 3$ (? for r > 3 & **Chaos** for r → 4)

"3n + 1" Conjecture: if $x_{n+1} = \{ x_n/2$ **if even;** $3x_n + 1$ **if odd }**,

then does the x_n always terminate in 4,2,1 loop?.

Recursive Procedures: A procedure which "calls" itself.

Example:
```
proc      factorial( n, int ≥ 0 )
if n = 0  then      x := 1
else      x := n × factorial(n-1)
return(x)
```

Recursive Structures are defined by a Basic Step and a Recursive Step.

String: $\sum$ **= Alphabet (Set of Letters) and** $\sum^*$ **= Set of Strings**

BS: empty string, $\lambda \in \sum^*$ **;**

RS: if $a \in \sum$ **and** $\omega \in \sum^*$ **, then** $\omega a \in \sum^*$ **.**

Length = $L(\theta) = \{ 0$ for $\theta=\lambda$ **or** $L(\omega) + 1$ for $\theta = \omega a \}$

Connected Graph (CG): vertices connected by edges.

BS: (single vertex) • is a CG;

RS: given a CG, to build a new CG:

1. add a new edge connecting it to a new vertex; or,
2. add a new edge connecting 2 existing vertices .

Simple Graph (SG): CG without loops or multiple edges.

BS: (single vertex) • is a SG;

RS: given a SCG, to build a new SG:

1. add a new edge connecting it to a new vertex; or,
2. add a new edge connecting 2 non-adjacent vertices.

Planar Graph (PG): SG drawable in the plane without crossing edges.

(RS2 for simple SG require the new edge not cross existing edges).

Tree(Rooted) : SG (always PG) without any "circuits" or "closed paths".

BS: (single vertex) • is a Tree (the root);

RS: given a Tree, to build a new Tree (with same root) by adding a new edge connecting an existing vertex to a new vertex.

Rooted Binary Tree (BT): (Hard to define non-recursively)

BS: (single vertex) • is a BT (the root)

RS: given two BTs, $\mathbf{T_1}$ **&** $\mathbf{T_2}$, to build a new one, $\mathbf{T_1+T_2}$, by adding a new root with two edges connecting it to the roots of $\mathbf{T_1}$ & $\mathbf{T_2}$.

Probability

Classical probability space is a finite set, **S**, of the equi-likely **outcomes** of a **random process**. An **event** , **E** , is a subset of **S,** where the **probability of an event** is: $\textbf{Prob(E)} = \textbf{P(E)} = \frac{\#(E)}{\#(S)}$.

Flip a fair coin three times:

S = { <HHH>,<HHT>,<HTH>,<HTT>,<THH>,<THT>,<TTH>,<TTT> }

Prob(2 heads) = 3/8 = 0.375.

Roll two fair dice:

S = 36 ordered pairs {(1,1),(1,2), …,(1,6),(2,1), …,(6,6) }

Prob(Sum = 4) = Prob{(1,3),(2,2),(3,1)} = 3/36 = 1/12.

4 of a kind at 5-card poker:

#(S) = #(All possible deals) = $_{52}\mathbf{C}_5$; #(4-kinds) = $_{13}\mathbf{C}_1\ {}_4\mathbf{C}_4\ {}_{48}\mathbf{C}_1$;

Probability = $\frac{(13\cdot1\cdot48)\cdot(5\cdot4\cdot3\cdot2\cdot1)}{52\cdot51\cdot50\cdot49\cdot48} = \frac{1}{4165}$

Law of Large Numbers: In a (valid) classical probability space, if N **trials** of the random process are done, then the limiting relative frequency of an event is the probability: $\textbf{Prob(E)} = \lim_{N\to\infty} \frac{\#(E \text{ occurs})}{N}$

General probability space is a finite set, S, of outcomes of a random process and a function: $p: S \to R^+$, satisfying: $\sum p(x) = 1$; where the **probability of an event** is: $\textbf{Prob(E)} = \textbf{P(E)} = \sum_{x\varepsilon E} p(x)$.

The **"p"** is usually computed experimentally as the relative frequency. In the Classical situation, $\mathbf{p(x)} = {}^1\!/_{\#(S)}$.

Conditional Probability and Independence:

Conditional Probability of E given F is: $P(E|F) = P(E\cap F) / P(F)$;

E & F are Independent if: $P(E|F) = P(E)$ or $P(E\cap F) = P(E)P(F)$;

note that if **E & F** are independent, then **$P(E|F) = P(E)$.**

Relations

A (binary) **relation between** sets **A** and **B** is $R \subseteq A \times B$. Write $(a,b) \in R$ **as** aRb. $\varnothing$ and $A \times B$ are trivial relations. If $A = B$, **R** is **relation on A**.

An **N-ary relation (among)** is $R \subseteq A_1 \times A_2 \times \ldots \times A_n = \prod A_i$. N-ary relations, viewed as **tables**, are the foundation of relational databases.

For $R_1, R_2 \subseteq A \times B$, each of these 4 is also a relation $\subseteq A \times B$:

Union:	$R_1 \cup R_2$
Intersection:	$R_1 \cap R_2$
Difference:	$R_1 - R_2$ and $R_2 - R_1$.

For $R \subseteq A \times B$, define: **not-R** $= \overline{R} = A \times B - R$ (complement of **R**).

For $R_1 \subseteq A \times B$, $R_2 \subseteq B \times C$, define:

Composition	$R_2 \circ R_1 \subseteq A \times C$ as $\{ (a,c) \mid \exists b \in B (aR_1b \wedge bR_2c) \}$
B-join	$R_1 \times_B R_2 \subseteq A \times B \times C$ as $\{ (a,b,c) \mid (aR_1b \wedge bR_2c) \}$
Product	$R_1 \times R_2 \subseteq A \times B \times B \times C$.

Relations on A: A relation $\approx \subseteq A \times A$, is:

Reflexive iff $\forall a \in A (a \approx a)$ (i.e. the diagonal, $\Delta = \{(a,a)\} \subseteq R$)
Symmetric iff $\forall a,b \in A (a \approx b \leftrightarrow b \approx a)$
Transitive iff $\forall a,b,c \in A (a \approx b \wedge b \approx c \rightarrow a \approx c)$
Equivalence relation iff it is reflexive, symmetric and transitive.

Examples: **(i)** $a \equiv b \bmod p$ (congruence) is an equivalence relation on $\mathbb{Z}$
(ii) $(a,b) \approx (c,d)$ iff $ad=bc$ is an equivalence relation on $\mathbb{Z} \times \mathbb{Z}^+$
(this is the precise definition of the rational numbers)

Proposition: If $\#(A) = n$; then: $\#($relations on $A) = 2^{n \cdot n} = 2\text{^}n^2$;
$\#($reflexive relations on $A) = 2\text{^}(n^2 - n)$;
$\#($symmetric, transitive or equivalence relations on $A) = ?$.

An equivalence relation, $\approx$, on **A** **partitions** **A** into a collection of disjoint subsets called **equivalence classes** , $[a] = \{ b \in A \mid a \approx b \}$.

The set of equivalence classes of $\approx$ is the **quotient set:** $A / \approx$.

Graph Theory

Graph: **vertices** and **edges** joining 2 vertices (ends).

Di-graph: each edge has a direction (will not discuss in detail).
Weighted (Di)Graph: each edge has a numeric weight.
Multi-edges: **> 1** edge joining same pair of vertices.
Loop: an edge joining a vertex to itself.
Simple Graph: has no loops and no multi-edges.
Complete Graph: precisely one edge joins each pair of vertices.
Adjacent vertices: have an edge joining them.
Adjacent edges: share a common vertex.

Path: a sequence of adjacent vertices without repeated edges.

Circuit: a closed path (begins and ends at the same vertex).
Connected Graph: any two vertices are connected by a path.
A graph is the disjoint "union" of its **connected components**.
Bridge: edge which disconnects the graph if removed.
Hair: edge with a terminal end (a vertex with only one edge).
Connected Digraphs are complicated by directions.
Tree: connected simple graph without any circuits.
DiTree or Directed Tree: tree with directed edges.
Rooted Tree: directed Tree, with a "beginning" vertex (root).

Standard Simple Graphs:

$\mathbf{L_n}$ = Linear with n edges
$\mathbf{C_n}$ = Circle with n edges
$\mathbf{S_n}$ = Star with n+1 vertices
$\mathbf{W_n}$ = Wheel with n+1 vertices
$\mathbf{K_n}$ = Complete with n vertices
($\mathbf{K_3}$ triangle, $\mathbf{K_4}$ tetrahedron
$\mathbf{Q_n}$ = n-dimensional cube
$\mathbf{Q_2}$ square, $\mathbf{Q_3}$ cube)
$\mathbf{B_n}$ = Full n-level Binary Tree
$\mathbf{D_n}$ = Dihedron on n-polygon.

Degree of a vertex: deg(v) = number of edge endings at the vertex, **v** .

Adjacency Matrix: $\mathbf{M_{ij}}$ = #edge endings for edges joining vertices $\mathbf{v_i}$ and $\mathbf{v_j}$, for a **graph** with vertices $\mathbf{V = \{v_i\}}$. (Note: diagonal is twice # loops at **v**)

Theorem: $\sum \mathbf{deg(v_i) = 2\#(E) = 2(\#\ of\ edges)}$
$\mathbf{deg(v_i) = \sum M_{ij} = sum\ of\ i^{th}\ row = \sum M_{ji} = sum\ of\ i^{th}\ column}$

Formal Definition: A **Graph** consists of **<V,E,end>**, where :

(i) Finite Set of Vertices, **V** ; (ii) Finite Set of edges, **E** ;

(iii) Function, **end: E → V×V** (product for di-graph); or

end: E → V•V (symmetric product for graph).

A graph is **simple** if **end** is **1-1** and with **range ∩ Δ = ∅ = { }** , where **Δ = "diagonal" = { (v,v) } or { {v,v} }** for directed and undirected, respectively. For simple graphs, **E** may be defined (viewed) as the **range** of **end**.

Definition: Two graphs, $\mathbf{<V_1,E_1,end_1>}$ and $\mathbf{<V_2,E_2,end_2>}$ are **<u>isomorphic</u>** (same graph with different pictures) iff there are bijections:

$$\varphi: V_1 \to V_2 \quad \text{and} \quad \psi: E_1 \to E_2$$

which make the diagram "**commute**":

$$\begin{array}{ccc} E_1 & \to & V_1 \times V_1 \text{ or } V_1 \bullet V_1 \\ \Psi\downarrow & \rightleftharpoons & \downarrow \varphi \times \varphi \text{ or } \varphi \bullet \varphi \\ E_2 & \to & V_2 \times V_2 \text{ or } V_2 \bullet V_2 \end{array}$$

Such a pair of bijections is called an **<u>isomorphism</u>** (bijection of the vertices which preserves edges). An isomorphism of a graph with itself is an **<u>automorphism</u>**. The set of all automorphisms for a graph is the **<u>automorphism group</u>** of the graph.

Example: For $\mathbf{K_4}$, there are **4! = 24** automorphisms
(or only 12 if physically impossible reflections excluded)
For $\mathbf{Q_3}$ and $\mathbf{D_4}$, there are **48** automorphisms.

<u>Euler Paths and Circuits:</u>

Euler path (EP): traverses every **edge** exactly once.
Euler circuit (EC): circuit which is an Euler path.

Euler's Theorem: A connected graph has an:

EC if & only if all vertices have even degree.

EP if & only if exactly 2 vertices have odd degree.

Fleury's Algorithm ("don't disconnect") finds an EC or EP as above.

Eulerizing a graph adds multiple edges until all vertices are even.

Hamilton Paths and Circuits:

Hamilton path (HP): includes every **vertex** exactly once.
Hamilton circuit (HC): circuit which is also a Hamilton path.

Dirac's and **Ore's** Theorem give sufficient existence criteria for HCs.

Dijkstra's Algorithm: finds shortest path joining any two vertices in a weighted simple connected graph.

Traveling Salesman Problem (TSP): For a weighted graph, find the Hamilton circuit of minimum total weight.

Brute Force Algorithm is **Θ((n-1)!)** complexity.
Nearest-Neighbor & **Cheapest-Link Approx-Algorithms** are **$\Theta(n^2)$**.

Planar Graphs :

Planar graphs are connected simple graphs which may be drawn in the plane without crossing edges (edges may be curves or lines). (Any graph may be drawn in 3 dimensions without crossing edges.)

A **region** of a planar graph is the area enclosed by a simple circuit or the area exterior to the graph.

Theorem: Letting let **v = #(V) , e = #(E) , and** f **= #(regions)**, then the **Euler Characteristic, χ = v – e + f = 2** , for any planar graph.

This result is also true for simple polyhedron where regions are faces. A polyhedral torus or donut is not simple and **χ = v – e + f = 0.**

Map Colorings

A simple map may be viewed as a planar graph with vertices the nations and edges the borders. A map **coloring** of a planar graph assigns a color to each nation so that nations with a common border are different colors. A map which may be colored with two colors is **bi-partite**.

Four Color Theorem asserts that any simple map may be colored with only 4 colors. It was proven only by using a computer.