8.	t	Modus ponens using Step 6 and 7
	$s \rightarrow t$	

Proof of contradiction:

The "Proof by Contradiction" is also known as reductio ad absurdum, which is probably Latin for "reduce it to something absurd".

Here's the idea:

1. Assume that a given proposition is untrue.
2. Based on that assumption reach two conclusions that contradict each other.

This is based on a classical formal logic construction known as Modus Tollens: If P implies Q and Q is false, then P is false. In this case, Q is a proposition of the form (R and not R) which is always false. P is the negation of the fact that we are trying to prove and if the negation is not true then the original proposition must have been true. If computers are not "not stupid" then they are stupid. (I hear that "stupid computer!" phrase a lot around here.)

Example:

Lets prove that there is no largest prime number (this is the idea of Euclid's original proof). Prime numbers are integers with no exact integer divisors except 1 and themselves.

1. To prove: "There is no largest prime number" by contradiction.
2. Assume: There is a largest prime number, call it p.
3. Consider the number N that is one larger than the product of all of the primes smaller than or equal to p. N=1*2*3*5*7*11...*p + 1. Is it prime?
4. N is at least as big as p+1 and so is larger than p and so, by Step 2, cannot be prime.
5. On the other hand, N has no prime factors between 1 and p because they would all leave

a remainder of 1. It has no prime factors larger than p because Step 2 says that there are no primes larger than p. So N has no prime factors and therefore must itself be prime (see note below).
We have reached a contradiction (N is not prime by Step 4, and N is prime by Step 5) and therefore our original assumption that there is a largest prime must be false.

Note: The conclusion in Step 5 makes implicit use of one other important theorem: The Fundamental Theorem of Arithmetic: Every integer can be uniquely represented as the product of primes. So if N had a composite (i.e. non-prime) factor, that factor would itself have prime factors which would also be factors of N.

Automatic Theorem Proving:

Automatic Theorem Proving (ATP) deals with the development of computer programs that show that some statement (the *conjecture*) is a *logical consequence* of a set of statements (the *axioms* and *hypotheses*). ATP systems are used in a wide variety of domains.

The language in which the conjecture, hypotheses, and axioms (generically known as *formulae*) are written is a logic, often classical 1st order logic, but possibly a non-classical logic and possibly a higher order logic. These languages allow a precise formal statement of the necessary information, which can then be manipulated by an ATP system. This formality is the underlying strength of ATP: there is no ambiguity in the statement of the problem, as is often the case when using a natural language such as English.

ATP systems are enormously powerful computer programs, capable of solving immensely difficult problems. Because of this extreme capability, their application and operation sometimes needs to be guided by an expert in the domain of application, in order to solve problems in a reasonable amount of time. Thus ATP systems, despite the name, are often used by domain experts in an interactive way. The interaction may be at a very detailed level, where the user guides the inferences made by the system, or at a much higher level where the user determines intermediate lemmas to be proved on the way to the proof of a conjecture. There is often a synergetic relationship between ATP system users and the systems themselves:

- The system needs a precise description of the problem written in some logical form,
- the user is forced to think carefully about the problem in order to produce an appropriate formulation and hence acquires a deeper understanding of the problem,
- the system attempts to solve the problem, if successful the proof is a useful output,
- if unsuccessful the user can provide guidance, or try to prove some intermediate result, or examine the formulae to ensure that the problem is correctly described,
- and so the process iterates.

ATP is thus a technology very suited to situations where a clear thinking domain expert can interact with a powerful tool, to solve interesting and deep problems. There are many ATP systems readily available for use.

UNIT II
Relations

Introduction

The elements of a set may be related to one another. For example, in the set of natural numbers there is the less than‘ relation between the elements. The elements of one set may also be related to the elements another set.

Binary Relation

A binary relation between two sets A and B is a rule R which decides, for any elements, whether a is in relation R to b. If so, then we write a R b. If a is not in relation R to b, then a /R b.

We can also consider a R b as the ordered pair (a, b) in which case we can define a binary relation from A to B as a subset of A X B. This subset is denoted by the relation R.

In general, any set of ordered pairs defines a binary relation.

For example, the relation of father to his child is F = {(a, b) / a is the father of b} In this relation F, the first member is the name of the father and the second is the name of the child.

The definition of relation permits any set of ordered pairs to define a relation.

For example, the set S given by

S = {(1, 2), (3, a), (b, a) ,(b, Joe)}

Definition

The domain D of a binary relation S is the set of all first elements of the ordered pairs in the relation.(i.e) D(S)= {a / $ b for which (a, b) Є S}

The range R of a binary relation S is the set of all second elements of the ordered pairs in the relation. (i.e) R(S) = {b / $ a for which (a, b) Є S}

For example

For the relation S = {(1, 2), (3, a), (b, a) ,(b, Joe)}

D(S) = {1, 3, b, b} and

R(S) = {2, a, a, Joe}

Let X and Y be any two sets. A subset of the Cartesian product X * Y defines a relation, say C. For any such relation C, we have D(C) Í X and R(C) Í Y, and the relation C is said to from X to Y. If Y = X, then C is said to be a relation form X to X. In such case, c is called a relation in X. Thus any relation in X is a subset of X * X . The set X * X is called a *universal relation* in X, while the empty set which is also a subset of X * X is called a *void relation* in X.

For example: Let L denote the relation —less than or equal to< and D denote the relation —divides< where x D y means — x divides y< . Both L and D are defined on the set {1, 2, 3, 4}

L = {(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4,4)}
D = {(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 4), (3, 3), (4, 4)}

L Ç D = {(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 4), (3, 3), (4, 4)} = D

Properties of Binary Relations:

Definition: A binary relation R in a set X is **reflexive if, for every x Є X, x R x, That is (x, x) Є** R, or R is reflexive in X **ó (x) (x Є X ® x R x).**

For example:-

- The relation £ is reflexive in the set of real numbers.
- The set inclusion is reflexive in the family of all subsets of a universal set.
- The relation equality of set is also reflexive.
- The relation is parallel in the set lines in a plane.
- The relation of similarity in the set of triangles in a plane isreflexive.

Definition: A relation R in a set X is symmetric if for every x and y in X, whenever x R y, then y R x.(i.e) R is symmetric in X ó (x) (y) (x Є X ^ y Є X ^ x R y ® y R x}

For example:-

- The relation equality of set is symmetric.
- The relation of similarity in the set of triangles in a plane is symmetric.
- The relation of being a sister is not symmetric in the set of all people.
- However, in the set females it is symmetric.

Definition: **A relation R in a set X is whenever x R y and y R z , then x R z. (i.e)** transitive **if, for every x, y, and z are in X, R is transitive in X ó** (x) (y) (z) (x Є X^ y Є X^ z Є X ^x R y ^ y R z ® x Rz)

For example:-

- The relations **<, £, >, ³** and **=** are transitive in the set of real numbers
- The relations Í, Ì, Ê, É and equality are also transitive in the family of sets.
- The relation of similarity in the set of triangles in a plane is transitive.

Definition: A relation R in a set X is **irreflexive if, for every x Є X , (x, x)ÏX.**

For example:-

- < The relation < is irreflexive in the set of all real numbers.
- < The relation proper inclusion is irreflexive in the set of all nonempty subsets of a universal set.
- Let X = {1, 2, 3} and S = {(1, 1), (1, 2), (3, 2), (2, 3), (3, 3)} is neither irreflexive nor reflexive.

Definition: A relation R in a set x is **anti symmetric** if , for every x and y in X,
whenever x R y and y R x, Then x = y.
Symbolically,(x) (y) (x Є X ^ y Є X ^ x R y ^ y R x ® x = y)

For example

- The relations £ , ³ and = are anti symmetric
- The relation Í is anti symmetric in set of subsets.
- The relation —divides< is anti symmetric in set of real numbers.
- Consider the relation —is a son of< on the male children in a family. Evidently the relation is not symmetric, transitive and reflexive.
- The relation — is a divisor of — is reflexive and transitive but not symmetric on the set of natural numbers.
- Consider the set H of all human beings. Let r be a relation — is married to — R is symmetric.
- Let I be the set of integers. R on I is defined as a R b if a – b is an even number. R is an reflexive, symmetric and transitive

Equivalence Relation:

Definition:A relation R in a set A is called an **equivalence** relation if

- a R a for every i.e. R is reflexive
- **a R b => b R a for every a, b Є A i.e. R is symmetric**
- a R b and b R c => a R c for every **a, b, c Є A, i.e. R is transitive.**

For example

- < The relation equality of numbers on set of real numbers.
- < The relation being parallel on a set of lines in a plane.

us consider the set T of triangles in a plane. Let us define a relation

Problem1: Let

R in T as R= {(a, b) / (a, b Є T and a is similar to b}

We have to show that relation R is an Equivalence relation

Solution :

< A triangle a is similar to itself. a R a

< If the triangle a is similar to the triangle b, then triangle b is similar to the triangle a then a R b => b R a

< If a is similar to b and b is similar to c, then a is similar to c (i.e) a R b and b R c => a R c.

Hence R is an equivalence relation.

Problem 2: Let x = {1, 2, 3, … 7} and R = {(x, y) / x – y is divisible by 3} Show that R is an equivalence relation.

Solution: For any a Є X, a- a is divisible by 3, Hence a R a, R is reflexive

For any a, b Є X, if a – b is divisible by 3, then b – a is also divisible by 3, R is symmetric.

For any a, b, c Є, if a R b and b R c, then a – b is divisible by 3 and b–c is divisible by 3. So that (a – b) + (b – c) is also divisible by 3, hence a – c is also divisible by 3. Thus R is transitive.

Hence R is equivalence.

Problem3 .Let Z be the set of all integers. Let m be a fixed integer. Two integers a and b are said to be congruent modulo m if and only if m divides a-b, in which case we write a º b (mod m). This relation is called the relation of congruence modulo m and we can show that is an equivalence relation.

Solution :

< a - a=0 and m divides a – a (i.e) **a R a, (a, a) Є R, R is** reflexive .

< a R b = m divides a-b

m divides b - a b º a (mod m) b R a that is R is symmetric.

- a R b and b R c => a ºb (mod m) and bº c (mod m) O m divides a – b and m divides b-c

  O a – b = km and b **– c = lm for some k ,l Є z**

  O (a – b) + (b – c) = km+lm

  O a – c = (k +l) m

- **a ~ c (mod m)**
- a R c
- **R is transitive**

Hence the congruence relation is an equivalence relation.

Equivalence Classes:

Let R be an equivalence relation on a set A. For any a ЄA, the equivalence class generated by a is the set of all elements b Є A such a R b and is denoted [a]. It is also called the R – equivalence class and denoted by a Є A. i.e., [a] = {b Є A / b R a}

Let Z be the set of integer and R be the relation **called —congruence modulo 3<** defined by R = {(x, y)/ xÎ Z Ù yÎZ Ù (x-y) is divisible by 3}

Then the equivalence classes are

[0] = {… -6, -3, 0, 3, 6, **…}**

[1] = {…, -5, **-2, 1, 4, 7, …}**

[2] = {…, -4, **-1, 2, 5, 8, …}**

Composition of binary relations:

Definition:**Let R be a relation from X to Y and S be a relation from Y to Z. Then the relation R** o S is given by R o S = {(x, z) / xÎX Ù z Î Z Ù y Î Y such that (x, y) Î R Ù (y, z) Î S)} is called the composite relation of R and S.

The operation of obtaining R o S is called the **composition of relations**.

Example**:** Let R = {(1, 2), (3, 4), (2, 2)} and
S = {(4, 2), (2, 5), (3, 1),(1,3)}
Then R o S = {(1, 5), (3, 2), (2, 5)} and S o R = {(4, 2), (3, 2), (1, 4)}
It is to be noted that R o S ≠ S o R.
Also Ro(S o T) = (R o S) o T = R o S o T

Note**: We write R o R as R2; R o R o R as R3 and so on.**

Definition

Let R be a relation from X to Y, a relation R from Y to X is called the converse of R, where the ordered pairs of Ř are obtained by interchanging the numbers in each of the ordered pairs of R. This means for x Î X and y Î Y, that x R y ó y Ř x.

Then the relation Ř is given by R = {(x, y) / (y, x) Î R} is called the converse of R Example:

Let R = {(1, 2),(3, 4),(2, 2)}
Then Ř = {(2, 1),(4, 3),(2, 2)}

Note: If R is an equivalence rel**ation, then Ř is also an equivalence relation.**

Definition **Let X be any finite set and R be a relation in X. The relation R+ = R U R2 U R3…in X. is called the *transitive closure*** of R in X

Example: Let R = {(a, b), (b, c), (c, a)}.
Now R2 = R o R = {(a, c), (b, a), (c, b)}
R3 = R2 o R = {(a, a), (b, b), (c, c)}
R4 = R3 o R = {(a, b), (b, c), (c, a)} = R
R5= R3o R2 = R2 and so on.

Thus, R+ = R U R2 U R3 U R4 U…
= R U R2 U R3.
={(a, b),(b, c),(c, a),(a, c),(b, a),(c ,b),(a, b),(b, b),(c, c)}

We see that R+ is a transitive relation containing R. In fact, it is the smallest transitive relation containing R.

Partial Ordering Relations:

Definition

A binary relation R in a set P is called ***partial order relation*** or *partial ordering* in P iff R is reflexive, anti symmetric, and transitive.

A partial order relation is denoted by the symbol £., If £ is a partial ordering on P, then the ordered pair (P, £) is called a *partially ordered set* or a *poset.*

< Let R be the set of real numbers. The relation —less than or equal to < or
O , is a partial ordering on R.

< Let X be a set and r(X) be its power set. The relation subset, Í on X is partial ordering.

< Let Sn be the set of divisors of n. The relation D means —divides< on Sn ,is partial ordering on Sn .

In a partially ordered set (P, £) , an element y Î P is said to cover an element x Î P if x <y and if there does not exist any element z Î P such that x £ z and z £ y; that is, y covers x Û (x < y Ù (x £ z £ y Þ x = z Ú z = y))

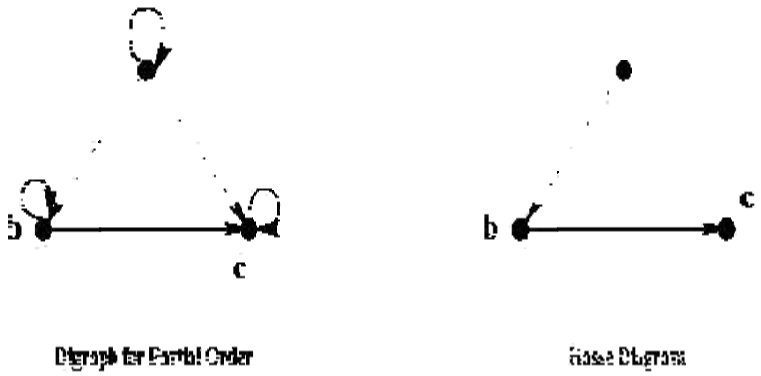
A partial order relation £ on a set P can be represented by means of a diagram known as a Hasse diagram or partial order set diagram of (P, £). In such a diagram, each element is represented by a small circle or a dot. The circle for x Î P is drawn below the circle for y Î P if x < y, and a line is drawn between x and y if y covers x.

If x < y but y does not cover x, then x and y are not connected directly by a single line.However, they are connected through one or more elements of P.

Hasse Diagram:

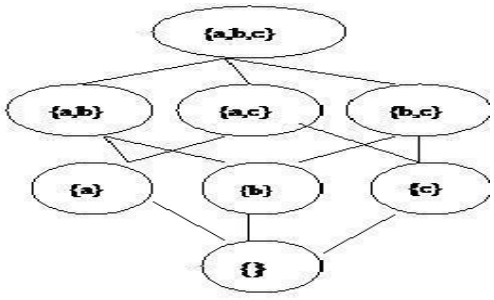
A Hasse diagram is a digraph for a poset which does not have loops and arcs implied by the transitivity.

Example 10: For the relation {< a, a >, < a, b >, < a, c >, < b, b >, < b, c >, < c, c >} on set {a, b,c}, the Hasse diagram has the arcs {< a, b >, < b, c >} as shown below



Digraph for Partial Order Hasse Diagram

Ex: Let A be a given finite set and r(A) its power set. Let Í be the subset relation on the elements of r(A). Draw Hasse diagram of (r(A), Í) for A = {a, b, c}



Lattice and its Properties:

Introduction:

A lattice is partially ordered set (L, £) in which every pair of elements a, b ÎL has a greatest lower bound and a least upper bound.

The glb of a subset, {a, b} Í L will be denoted by a * b and the lub by a Å b.

.

Usually, for any pair a, b Î L, GLB {a, b} = a * b, is called the meet or product and LUB{a, b} = a Å b, is called the join or sum of a and b.

Example1 Consider a non-empty set S and let P(S) be its power set. The relation Í —contained in< is a partial ordering on P(S). For any two subsets A, BÎ P(S)

GLB {A, B} and LUB {A, B} are evidently A Ç B and A È B respectively.

Example2 Let I+ be the set of positive integers, and D denote the relation of —division< in I+ such that for any a, b Î I+ , a D b iff a divides b. Then (I+, D) is a lattice in which

the join of a and b is given by the least common multiple(LCM) of a and b, that is,

a Å b = LCM of a and b, and the meet of a and b, that is , a * b is the greatest common divisor (GCD) of a and b.

A lattice can be conveniently represented by a diagram.

For example, let Sn be the set of all divisors of n, where n is a positive integer. Let D denote the relation —division< such that for any a, b Î Sn, a D b iff a divides b.

Then (Sn, D) is a lattice with a * b = gcd(a, b) and a Å b = lcm(a, b).

Take n=6. Then S6 = {1, 2, 3, 6}. It can be represented by a diagram in Fig(1). Take n=8. Then S8 = {1, 2, 4, 8}

Two lattices can have the same diagram. For example if S = {1, 2, 3} then (p(s), Í) and (S6,D) Have the same diagram viz. fig(1), but the nodes are differently labeled .

We observe that for any partial ordering relation £ on a set S the converse relation ³ is also partial ordering relation on S. If (S, £) is a lattice With meet a * b and join a Å b , then (S, ³) is the also a lattice with meet a Å b and join a * b i.e., the GLB and LUB get interchanged . Thus we have **the principle of duality of lattice as follows.**

Any statement about lattices involving the operations ^ and V and the relations £ and ³ remains true if ^, V, ³ and £ are replaced by V, ^, £ and ³ respectively.

The operation ^ and V are called duals of each other as are the relations £ and ³.. Also, the lattice (L, £) and (L, ³) are called the duals of each other.

Properties of lattices:

Let (L, £) be a lattice with the binary operations * and Å then for any a, b, c Î L,

< a * a = a a Å a = a (Idempotent)

< a * b = b * a, a Å b = b Å a (Commutative)

< (a * b) * c = a * (b * c) , (a Å) Å c = a Å (b Å c)

O (Associative) <

a * (a Å b) = a , a Å (a * b) = a (absorption)

For any a ÎL, a £ a, a £ LUB {a, b} => a £ a * (a Å b). On the other hand, GLB {a, a Å b} £ a i.e., (a Å b) Å a, hence a * (a Å b) = a

Theorem 1

Let (L, £) be a lattice with the binary operations * and Å denote the operations of meet and join respectively For any a, b Î L,

a £ b ó a * b = a ó a Å b = b

Proof

Suppose that a £ b. we know that a £ a, a £ GLB {a, b}, i.e., a £ a * b.
But from the definition of a * b, we get a * b £ a.
Hence a £ b => a * b = a …… (1)

Now we assume that a * b = a; but is possible only if a £ b,
that is a * b = a => a £ b **……** (2)
From (1) and (2), we get a £ b ó a * b = a.

Suppose a * b = a.

then b Å (a * b) = b Å a = **a Å b ……** (3)
but b Å (a * b) = b (by iv)**……..** (4)
Hence a Å b = b, from (3) => (4)
Suppose aÅ b = b, i.e., LUB {a, b} = b, this is possible only if a£ b, thus(3) => (1)
(1) => (2) => (3) => (1). Hence these are equivalent.

Let us assume a * b = a.
Now (a * b) Å b = a Å b
We know that by absorption law , (a * b) Å b = b
so that a Å b = b, therefore a * b = a Þ a Å b = b (5)
similarly, we can prove a Å b = b Þ a * b = a (6)
From (5) and (6), we get

a * b = a Û a Å b = b
Hence the theorem.

Theorem2 **For any a, b, c Î L, where (L, £) is a lattice. b**

£ c => { a * b £ a * c and
{ a Å b £ a Å c

Proof Suppose b £ c. we have proved that b £ a ó b * c = b…..... (1)

Now consider (a * b) * (a * c) = (a * a) * (b * c)
= a * (b * c)(by Idempotent)
= a * b (by (1))

Thus (a * b) * (a * c) = a * b which => (a * b) £ (a * c) Similarly (a Å b) Å (a Å c) = (a Å a) Å (b Å c)
= a Å (b Å c)
= a Å c
which => (a Å b) £ (a Å c)

note:These properties are known as **isotonicity.**

Functions

Introduction

A function is a special type of relation. It may be considered as a relation in which each element of the domain belongs to only one ordered pair in the relation. Thus a function from A **to B is a subset of A X B having the property that for each a Є A, there is one and only one b Є B** such that (a, b) Î G.

Definition

Let A and B be any two sets. A relation f from A to B is called a function **if for every a Є A there is a unique b Є B such that (a, b) Є f .**

Note that the definition of function requires that a relation must satisfy two additional conditions in order to qualify as a function.

The first condition is that every a Є A must be related to some b Є B, (i.e) the domain of f must be A and not merely subset of A. The second requirement of uniqueness can be expressed **as (a, b) Є f ^ (b, c) Є f => b = c**

Intuitively, a function from a set A to a set B is a rule which assigns to every element of A, a unique element of B. **If a ЄA, then the unique element of B assigned to a under f is denoted by f**

(a).The usual notation for a function f from A to B is f: A® B defi**ned by a ® f (a) where a Є A,** f(a) is called the image of **a** under f and **a** is called pre image of f(a).

- Let X = Y = **R** and f(x) = x2 + 2. Df = R and Rf Í R.
- Let X be the set of all statements in logic and let Y = {True, False}. A mapping f: X®Y is a function.
- A program written in high level language is mapped into a machine language by a compiler. Similarly, the output from a compiler is a function of its input.
- Let X = Y = **R** and f(x) = x2 is a function from X ® Y,and g(x2) = x is not a function from X ® Y.

A mapping f: A ® B is called one-to-one **(injective or 1 –1) if distinct elements of A are mapped into distinct elements of B. (i.e) f is one-to-one if**

a1 = a2 => f (a1) = f(a2) or equivalently f(a1) ¹ f(a2) => a1 ¹ a2

For example, **f: N ® N given by f(x) = x is 1-1 where N is the set of a natural numbers.**

A mapping f: A® B is called **onto (surjective) if for every b Є B there is an a Є A such** that f (a) = B. i.e. if every element of B has a pre-image in A. Otherwise it is called **into**.

For example, **f: Z®Z given by f(x) = x + 1 is an onto mapping. A mapping is both 1-1 and onto is called bijective**

.

For example f: R®R given by f(x) = X + 1 is bijective.

Definition: A mapping f: R® b is called a **constant mapping** if, for all aÎA, f (a) = b, a fixed element.

For example **f: Z®Z given by f(x) = 0, for all x ÎZ is a constant mapping.**

Definition

A mapping f: A®A is called the identity mapping **of A if f (a) = a, for all aÎA. Usually it is denoted by IA or simply I.**

Composition of functions:

If f: A®B and g: B®C are two functions, then the composition of functions f and g, denoted by g o f, is the function is given by g o f : A®C and is given by

g o f = {(a, c) / a Є A ^ c Є C ^ $bÎ B ': f(a)= b ^ g(b) = c} **and (g of)(a) = ((f(a))**

Example 1: **Consider the sets A = {1, 2, 3},B={a, b} and C = {x, y}. Let f: A® B be defined by f (1) = a ; f(2) = b and f(3)=b and Let g: B® C be defined by g(a) = x and g(b) = y**

(i.e) f = {(1, a), (2, b), (3, b)} and g = {(a, x), (b, y)}. Then g o f: A®C is defined by

(g of) (1) = g (f(1)) = g(a) = x
(g o f) (2) = g (f(2)) = g(b) = y
(g o f) (3) = g (f(3)) = g(b) = y

i.e., g o f = {(1, x), (2, y),(3, y)}

If f: A® A and g: A®A, where A= {1, 2, 3}, are given by

f = {(1, 2), (2, 3), (3, 1)} and g = {(1, 3), (2, 2), (3, 1)}
Then g of = {(1, 2), (2, 1), (3, 3)}, fog= {(1, 1), (2, 3), (3, 2)}
f of = {(1, 3), (2, 1), (3, 2)} and gog= {(1, 1), (2, 2), (3, 3)}

Example 2: **Let f(x) = x+2, g(x) = x – 2 and h(x) = 3x for x Î R, where R is the set of real numbers.**

Then f o f = {(x, x+4)/xÎ R}

f o g = {(x, x)/ x Î X}
g o f = {(x, x)/ xÎ X}
g o g = {(x, x-4)/x Î X}
h o g = {(x,3x-6)/ x Î X}
h o f = {(x, 3x+6)/ x Î X}

Inverse functions:

Let f: A® B be a one-to-one and onto mapping. Then, its inverse, denoted by f -1 is given by f - 1 = {(b, a) / (a, b) Î f} Clearly f-1: B® A is one-to-one and onto.

Also we observe that f o f -1 = IB and f -1o f = IA. If f -1 exists then f is called invertible**.**

*For example:L***et f: R ®R be defined by f(x) = x + 2**
Then f -1: R® R is defined by f -1(x) = x - 2

Theorem: **Let f: X ®Y and g: Y ® Z be two one to one and onto functions. Then gof is also one to one and onto function.**

Proof

Let f:X ® Y g : Y ® Z be two one to one and onto functions. Let x1, x2 Î X

< g o f (x1) = g o f(x2),
< g (f(x1)) = g(f(x2)),

< g(x1) = g(x2) since [f is 1-1]

x1 = x2 since [g is 1-1}
so that gof is 1-1.
By the definition of composition, gof : X ® Z is a function.
We have to prove that every element of z Î Z an image element for some x Î X under gof.
Since g is onto $ y ÎY ': g(y) = z and f is onto from X to Y,
$ x ÎX ': f(x) = y.
Now, gof (x) = g (f (x))
= g(y) [since f(x) = y]
= z [since g(y) = z] which shows that gof is onto.

Theorem (g o f) -1 = f -1 o g -1 **(i.e) the inverse of a composite function can be expressed in terms of the composition of the inverses in the reverse order.**

Proof. f: A ® B is one to one and onto. g: B ® C is one to one and onto.
gof: A ® C is also one to one and onto. Þ (gof) -1:
C ® A is one to one and onto.
Let a Î A, then there exists an element b Î b such that f (a) = b Þ a = f-1
(c). Now b Î B Þ there exists an element c Î C such that g (b) = c Þ b = g - 1(c).
Then (gof)(a) = g[f(a)] = g(b) = c Þ a = (gof) -1(c)..... **(1)**
(f -1 o g-1) (c) = f -1(g -1 (c)) = f -1(b) = a Þ a = (f -1 o g -1)(c) ….(2) Combining (1) and (2), we have (gof) -1 = f -1 o g -1

Theorem: If f: A ® B is an invertible mapping ,
then f o f -1 = I B and f-1 o f = IA

Proof: f is invertible, then f -1 is defined by f(a) = b ó f-1(b)
= a where a Î A and bÎ B .

Now we have to prove that f of -1 = IB

. Let bÎ B and f -1(b) = a, a Î A
then fof-1(b) = f(f-1(b))
= f(a) = b

therefore f o f -1 (b) = b " b Î B => f o f -1 =
IB Now f -1 o f(a) = f -1 (f(a)) = f -1 (b) = a
therefore f -1 o f(a) = a " a Î A => f -1 o f = IA.
Hence the theorem.

Recursive Functions:

The term "recursive function" is often used informally to describe any function that is defined with recursion. There are several formal counterparts to this informal definition, many of which only differ in trivial respects.

Kleene (1952) defines a "partial recursive function" of nonnegative integers to be any function f that is defined by a noncontradictory system of equations whose left and right sides are composed from
(1) function symbols (for example, f, g, h, etc.), (2) variables for nonnegative integers (for example, x, y, z, etc.), (3) the constant 0, and (4) the successor function $S(x) = x + 1$.

For example,

$$f(x, 0) = 0 \quad (1)$$

$$f(x, S(y)) = g(f(x, y), x) \quad (2)$$

$$g(x, 0) = x \quad (3)$$

$$g(x, S(y)) = S(g(x, y)) \quad \textbf{(4)}$$

defines $f(x, y)$ to be the function $x\,y$ that computes the product of x and y.

Note that the equations might not uniquely determine the value of f for every possible input, and in that sense the definition is "partial." If the system of equations determines the value of f for every input, then the definition is said to be "total." When the term "recursive function" is used alone, it is usually implicit that "total recursive function" is intended. Note that some authors use the term "general recursive function to mean partial recursive function, although others use it to mean "total recursive function."

The set of functions that can be defined recursively in this manner is known to be equivalent to the set of functions computed by Turing machines and by the lambda calculus.

Algebraic systems:

An algebraic system, loosely speaking, is a set, together with some operations on the set. Before formally defining what an algebraic system is, let us recall that a *n* -ary operation (or operator) on a set *A* is a function whose domain is *An* and whose range is a subset of *A* . Here, *n* is a non- negative integer. When *n*=0 , the operation is usually called a nullary operation, or a constant, since one element of *A* is singled out to be the (sole) value of this operation. A finitary operation on *A* is just an *n* -ary operation for some non-negative integer *n* .

Definition**. An *algebraic system* is an ordered pair (*A O*) , where *A* is a set, called the underlying set of the algebraic system, and *O* is a set, called the operator set, of finitary operations on *A* .**

We usually write *A* **, instead of (*A O*) , for brevity.**

A prototypical example of an algebraic system is a group, which consists of the underlying set *G* , and a set *O* consisting of three operators: a constant *e* called the multiplicative identity, a unary operator called the multiplicative inverse, and a binary operator called the multiplication.

For a more comprehensive listing of examples, please see this entry.

Remarks.

< An algebraic system is also called algebra for short. Some authors require that *A* be non-empty. Note that *A* is automatically non-empty if *O*contains constants. A *finite algebra* is an algebra
whose underlying set is finite.

< By definition, all operators in an algebraic system are finitary. If we allow *O* to contain infinitary operations, we have an *infinitary algebraic system*. Other generalizations are possible. For example, if the operations are allowed to be multivalued, the algebra is said to be a *multialgebra*. If the operations are not everywhere defined, we get a *partial algebra*. Finally, if more than one underlying set is involved, then the algebra is said to be *many-sorted*.

The study of algebraic systems is called the theory of universal algebra. The first important thing in studying algebraic system is to compare systems that are of the same ``type''. Two algebras are said to have the same *type* if there is a one-to-one correspondence between their operator sets such that an *n* -ary operator in one algebra is mapped to an *n* -ary operator in the otheralgebra.

Examples:

Some recurring universes: N=natural numbers; Z=integers; Q=rational numbers; R=real numbers; C=complex numbers.

N is a pointed unary system, and under addition and multiplication, is both the standard interpretation of Peano arithmetic and a commutative semiring.

Boolean algebras are at once semigroups, lattices, and rings. They would even be abelian groups if the identity and inverse elements were identical instead of complements.

Group-like structures

< Nonzero **N** under addition (+) is a magma.
< **N** under addition is a magma with an identity.
< **Z under subtraction (−) is a quasigroup.**
< Nonzero **Q** under division (÷) is a quasigroup. $y * a = b$ if $a * x = b$ if and only if $x = a^{-1} * b$, and
< Every group is a loop, because $a * x = b$ if and only if $x = a^{-1} * b$, and $y * a = b$ if and only if $y = b * a^{-1}$.
< 2x2 matrices(of non-zero determinant) with matrix multiplication form a group.
< **Z** under addition (+) is an abelian group.
< Nonzero **Q** under multiplication (×) is an abelian group.
< Every cyclic group G is abelian, because if x, y are in G, then $xy = a^m a^n = a^{m+n} = a^{n+m} = a^n a^m = yx$. In particular, **Z** is an abelian group under addition, as is the integers modulo n **Z**/n**Z**.
< A monoid is a category with a single object, in which case the composition of morphisms and the identity morphism interpret monoid multiplication and identity element, respectively.
< The Boolean algebra **2** is a boundary algebra.

General Properties:

Property of Closure

If we take two *real numbers* and multiply them together, we get another real number. (The real numbers are all the rational numbers and all the irrational numbers.) Because this is always true, we say that the real numbers are "closed under the operation of multiplication": there is no way to escape the set. When you combine any two elements of the set, the result is also included in theset.

Real numbers are also closed under addition and subtraction. They are not closed under the square root operation, because the square root of -1 is not a real number.

Inverse

The inverse of something is that thing turned inside out or upside down. The inverse of an operation undoes the operation: division undoes multiplication.

A number's *additive inverse* is another number that you can add to the original number to get the additive identity. For example, the additive inverse of 67 is -67, because 67 + -67 = 0, the additive identity.

Similarly, if the product of two numbers is the *multiplicative identity*, the numbers are *multiplicative inverses.* Since 6 * 1/6 = 1 (the multiplicative identity), the multiplicative inverse of 6 is 1/6.

Zero does not have a multiplicative inverse, since no matter what you multiply it by, the answer is always 0, not 1.

Equality

The equals sign in an equation is like a scale: both sides, left and right, must be the same in order for the scale to stay in balance and the equation to be true.

The *addition property of equality* says that if a = b, then a + c = b + c: if you add the same number to (or subtract the same number from) both sides of an equation, the equation continues to be true.

The *multiplication property of equality* says that if a = b, then a * c = b * c: if you multiply (or divide) by the same number on both sides of an equation, the equation continues to be true.

The *reflexive property of equality* just says that a = a: anything is congruent to itself: the equals sign is like a mirror, and the image it "reflects" is the same as the original.

The *symmetric property of equality* says that if a = b, then b = a.

The *transitive property of equality* says that if a = b and b = c, then a = c.

Semi groups and monoids:

In the previous section, we have seen several algebraic system with binary operations. Here we consider an algebraic system consisting of a set and an associative binary operation on the set and then the algebraic system which possess an associative property with an identity element. These algebraic systems are called semigroups and monoids.

Semi group

Let S be a nonempty set and let * be a binary operation on S. The algebraic system (S, *) is called a semi-group if * is associative

if a * (b*c) = (a * b) * c for all a, b, c Î S.

Example The N of natural numbers is a semi-group under the operation of usual addition of numbers.

Monoids

Let M be a nonempty set with a binary operation * defined on it. Then (M, *) is called a monoid if

- * is associative

(i.e) a * (b * c) = (a * b) * c for all a, b, c Î M and
there exists an element e in M such that
a * e = e * a = a for all a Î M
e is called the identity element in (M,*).

It is easy to prove that the identity element is unique. From the definition it follows that (M,*) is a semigroup with identity.

Example1 Let S be a nonempty set and r(S) be its power set. The algebras (r(S),U) and (r(S), Ç) are monoids with the identities f and S respectively.

Example2 Let N be the set of natural numbers, then (N,+), (N, X) are monoids with the identities 0 and 1 respectively.

Groups Sub Groups:

Recalling that an algebraic system (S, *) is a semigroup if the binary operation * is associative. If there exists an identity element e Î S, then (S,*) is monoid. A further condition is imposed on the elements of the monoid, i.e., the existence of an inverse for each element of S then the algebraic system is called a group.

Definition

Let G be a nonempty set, with a binary operation * defined on it. Then the algebraic system (G,*) is called a group if

- * is associative i.e. a * (b * c) = (a * b) * c for all a, b, c,Î G.
- there exists an element e in G such that a * e = e * a = a for all a Î G
- for each a Î G there is an element denoted by a-1 in G such that a * a-1 = a-1 * a = e, a-1 is called the inverse of a.

From the definition it follows that (G,*) is a monoid in which each element has an inverse w.r.t. * in G.

A group (G,*) in which * is commutative is called an abelian group or a commutative group. If * is not commutative then (G,*) is called a non-abelian group or non-commutative group.

The order of a group (G,*) is the number of elements of G, when G is finite and is denoted by o(G) or |G|

Examples 1. (Z5, +5) is an abelian group of order 5.

2. G = {1, -1, i, -i} is an abelian group with the binary operation x is defined as 1 x 1 = 1, -1 x -1 = 1, i x i = -1 , -i x -i = 1, …

Homomorphism of semigroups and monoids

Semigroup homomorphism.

Let (S, *) and (T, D) be any two semigroups. A mapping g: S ® T such that any two elements a, b Î S , g(a * b) = g(a) D g(b) is called a semigroup homomorphism.

Monoid homomorphism

Let (M, *,eM) and (T, D,eT) be any two monoids. A mapping g: M® T such that any two elements a, b Î M ,

g(a * b) = g(a) D g(b)
and g(eM) = eT

is called a monoid homomorphism.

Theorem 1 Let (s, *) , (T, D) and (V, Å) be semigroups. A mapping g: S ® T and h: T ® V be semigroup homomorphisms. Then (hog): S ® V is a semigroup homomorphism from (S,*) to(V,Å).

Proof. Let a, b Î S. Then

(h o g)(a * b) = h(g(a* b))
= h(g(a) D g(b))
= h(g(a)) Å h(g(b))
= (h o g)(a) Å (h o g)(b)

Theorem 2 **Let (s,*) be a given semigroup. There exists a homomorphism g: S ® SS, where (SS, o) is a semigroup of function from S to S under the operation of composition.**

Proof For any element a Î S, let g(a) = fa where f aÎ SS and f a is defined by

f a(b) = a * b for any a, bÎ S
g(a * b) = f a*b

Now f a*b(c) = (a * b) * c = a*(b * c)
where = f a(f b(c)) = (f a o f b) (c).

Therefore, g(a * b) = f a*b = f a o f b = g(a) o g(b), this shows that g: S ® SS is a homomorphism.

Theorem 3 **For any commutative monoid (M, *),the set of idempotent elements of M forms a submonoid.**

Proof. Let S be the set of idempotent elements of M.
Since the identity element e Î M is idempotent, e Î S.
Let a, b Î S, so that a* a = a and b * b = b
Now (a * b) * (a * b) = (a * b) * (b * a) [(M, *) is a commutative monoid]
= a * (b * b) * a
= a * b * a
= a * a * b
= a * b
Hence a * b Î S and (S, *) is a submonoid.

Isomorphism:

In abstract algebra, an isomorphism **is a bijective map *f* such that both *f* and its inverse f^{-1} are** homomorphisms, i.e., *structure-preserving* mappings. In the more general setting of category theory, an **isomorphism** is a morphism $f: X \to Y$ in a category for which there exists an **"inverse"** $f^{-1}: Y \to X$**, with the property that both *f***

$f = \mathrm{id}_X$ and $f^{-1} = \mathrm{id}_Y$.