UNIT-II

FUNCTIONS, ORDER RELATIONS AND BOOLEAN ALGEBRA

Functions: - Functions are an important part of discrete mathematics. This article is all about functions, their types, and other details of functions. A function assigns exactly one element of a set to each element of the other set. Functions are the rules that assign one input to one output. The function can be represented as f: A ⇢ B. A is called the domain of the function and B is called the codomain function.

What is Function?

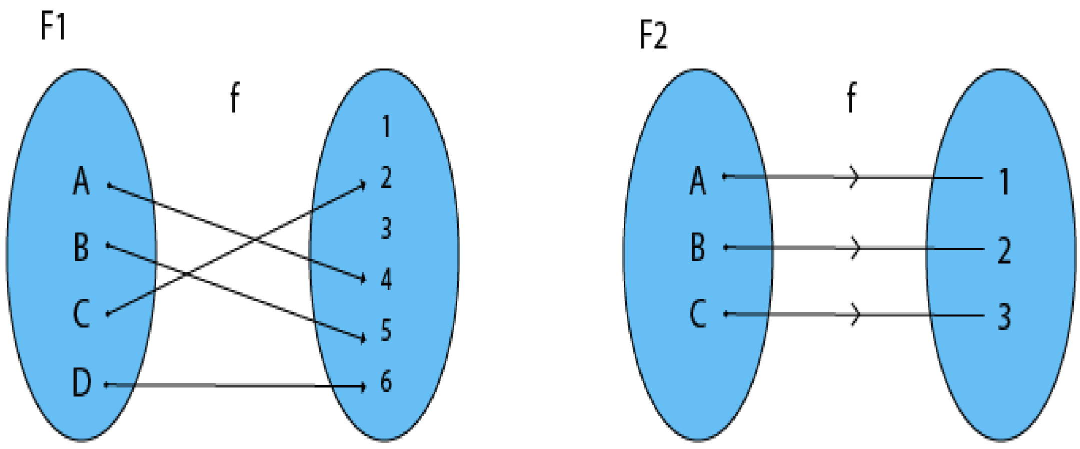
- A function assigns exactly one element of one set to each element of other sets.
- A function is a rule that assigns each input exactly one output.
- A function f from A to B is an assignment of exactly one element of B to each element of A (where A and B are non-empty sets).
- A function f from set A to set B is represented as **f: A ⇢ B** where A is called the domain of f and B is called as codomain of f.
- If b is a unique element of B to element a of A assigned by function F then, it is written as f(a) = b.
- Function f maps A to B means f is a function from A to B i.e. f: A ⇢ B

Types of Function

Some of the common types of functions are:

- One-One Function
- Many-One Function
- Onto Function
- Into Function
- One-One Correspondent Function

- One-One Into Function
- Many-One Onto Function
- Many-One Into Function
- **Injective (One-to-One) Functions:** A function in which one element of Domain Set is connected to one element of Co-Domain Set.

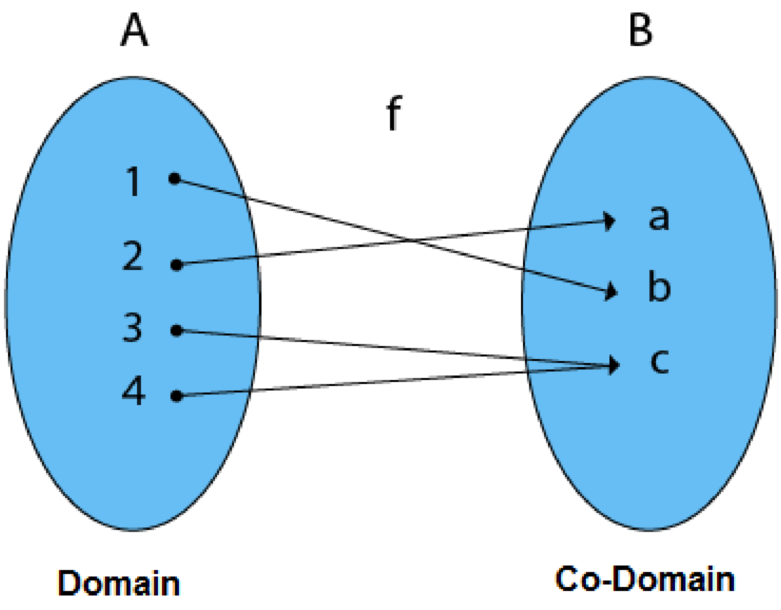


F1 and F2 show one to one Function

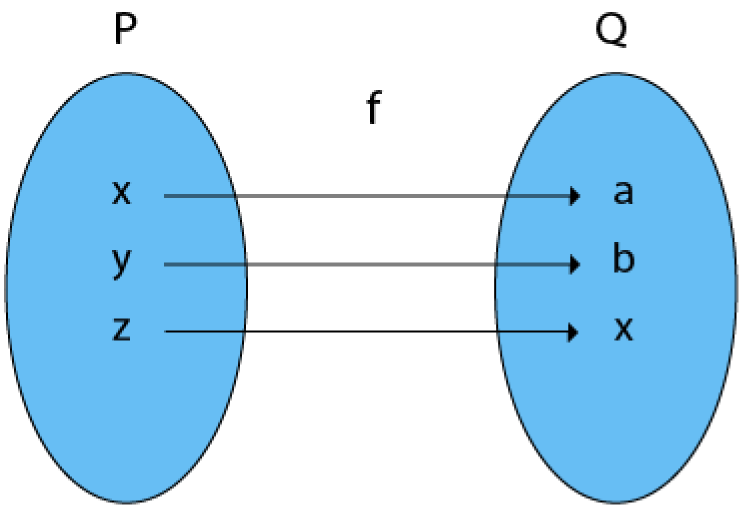
- **Surjective (Onto) Functions:** A function in which every element of Co-Domain Set has one pre-image.

  Example: Consider, A = {1, 2, 3, 4}, B = {a, b, c} and f = {(1, b), (2, a), (3, c), (4, c)}.

  It is a Surjective Function, as every element of B is the image of some A.



- **Bijective (One-to-One Onto) Functions:** A function which is both injective (one to - one) and surjective (onto) is called bijective (One-to-One Onto) Function.



Example: -

Consider P = {x, y, z}

Q = {a, b, c} and f: P → Q such that

f = {(x, a), (y, b), (z, c)}

The f is a one-to-one function and also it is onto. So it is a bijective function.

- **Into Functions:** A function in which there must be an element of co-domain Y does not have a pre-image in domain X.

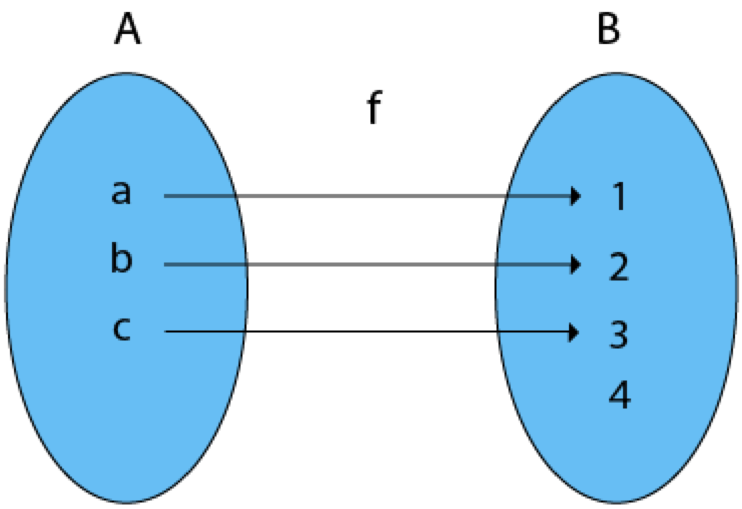
Example:

Consider, A = {a, b, c}

B = {1, 2, 3, 4} and f: A → B such that

f = {(a, 1), (b, 2), (c, 3)}

In the function f, the range i.e., {1, 2, 3} ≠ co-domain of Y i.e., {1, 2, 3, 4}



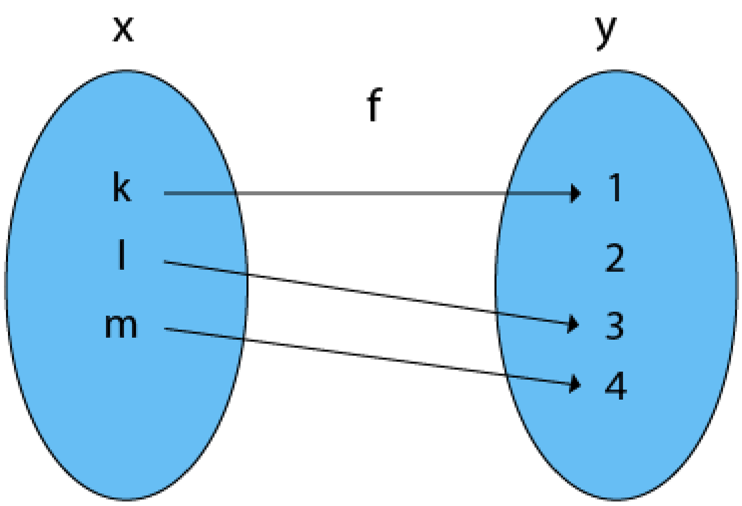
- **One-One Into Functions:** Let f: X → Y. The function f is called one-one into function if different elements of X have different unique images of Y.

Example:

Consider, X = {k, l, m}

Y = {1, 2, 3, 4} and f: X → Y such that

f = {(k, 1), (l, 3), (m, 4)}

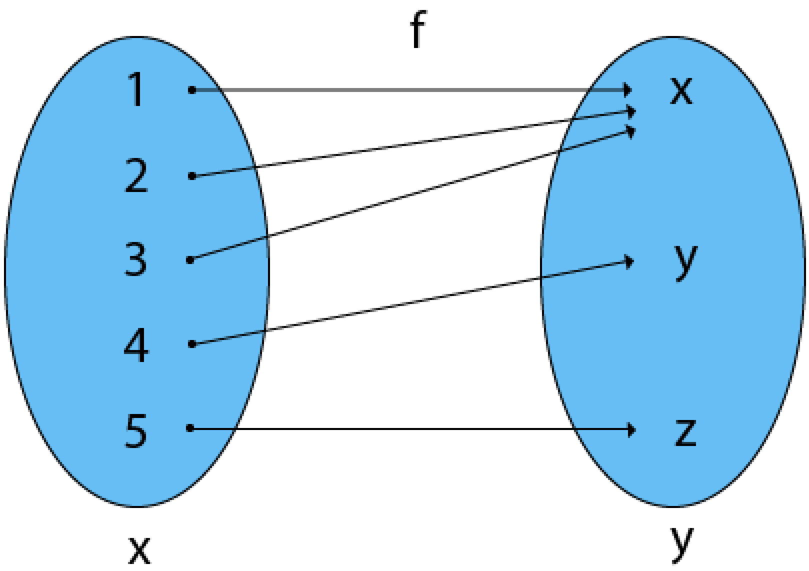


- **Many-One Functions:** Let f: X → Y. The function f is said to be many-one functions if there exist two or more than two different elements in X having the same image in Y.

Example:

Consider X = {1, 2, 3, 4, 5}

Y = {x, y, z} and f: X → Y such that



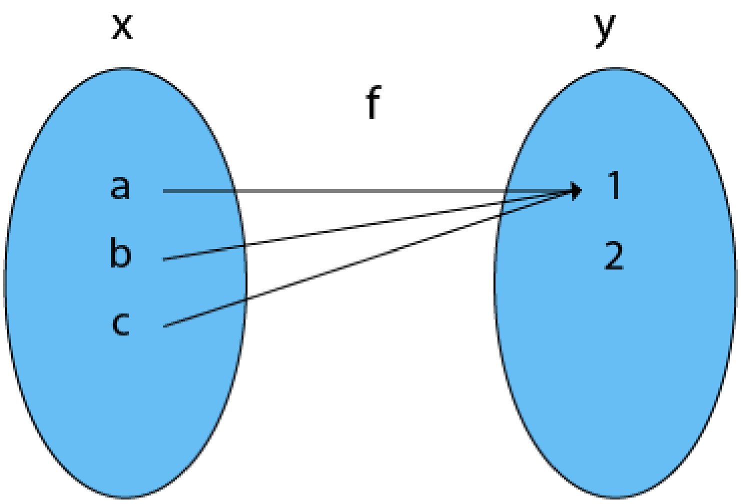
- **Many-One Into Functions:** Let f: X → Y. The function f is called the many-one function if and only if is both many one and into function.

  Example:

  Consider X = {a, b, c}

  Y = {1, 2} and f: X → Y such that

  f = {(a, 1), (b, 1), (c, 1)}



- **Many-One Onto Functions:** Let f: X → Y. The function f is called many-one onto function if and only if is both many one and onto.

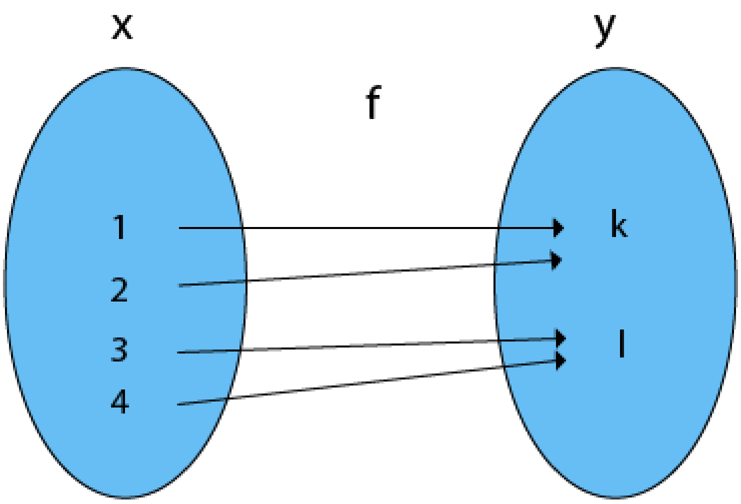
Example:

Consider X = {1, 2, 3, 4}

Y = {k, l} and f: X → Y such that

f = {(1, k), (2, k), (3, l), (4, l)}

The function f is a many-one (as the two elements have the same image in Y) and it is onto (as every element of Y is the image of some element X). So, it is many-one onto function



Permutation functions: -

A **permutation** of a set S is a bijective (one-to-one and onto) function from S to itself. If S has n elements, a permutation rearranges the elements of S in all possible ways.

For a set with 3 elements:

n=3n = 3n=3

The number of permutations is:

3!=3×2×1=63! = 3 * 2 * 1 = 6

3!=3×2×1=6

So, there are 6 distinct permutation functions for a set with 3 elements.

Listing All Permutations

If the set is {a,b,c}, the 6 permutations are:

1. (a,b,c)

2. (a,c,b)
3. (b,a,c)
4. (b,c,a)
5. (c,a,b)
6. (c,b,a)

These permutations represent all possible ways to rearrange the 3 elements in the set.

Partial Order Relations

A relation R on a set A is called a partial order relation if it satisfies the following three properties:

1. Relation R is Reflexive, i.e. aRa ∀ a∈A.
2. Relation R is Antisymmetric, i.e., aRb and bRa ⟹ a = b.
3. Relation R is transitive, i.e., aRb and bRc ⟹ aRc.

Example1: Show whether the relation (x, y) ∈ R, if, x ≥ y defined on the set of +ve integers is a partial order relation.

Solution: Consider the set A = {1, 2, 3, 4} containing four +ve integers. Find the

relation for this set such as R = {(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3), (1, 1), (2, 2), (3, 3), (4, 4)}.

Reflexive: The relation is reflexive as for every a ∈ A. (a, a) ∈ R, i.e. (1, 1), (2, 2), (3, 3), (4, 4) ∈ R.

Antisymmetric: The relation is antisymmetric as whenever (a, b) and (b, a) ∈ R, we have a = b.

Transitive: The relation is transitive as whenever (a, b) and (b, c) ∈ R, we have (a, c) ∈ R.

Example: (4, 2) ∈ R and (2, 1) ∈ R, implies (4, 1) ∈ R.

As the relation is reflexive, antisymmetric and transitive. Hence, it is a partial order relation.

Example2: Show that the relation 'Divides' defined on N is a partial order relation.

Solution:

Reflexive: We have a divides a, $\forall$ a$\in$N. Therefore, relation 'Divides' is reflexive.

Antisymmetric: Let a, b, c $\in$N, such that a divides b. It implies b divides a iff a = b. So, the relation is antisymmetric.

Transitive: Let a, b, c $\in$N, such that a divides b and b divides c.

Then a divides c. Hence the relation is transitive. Thus, the relation being reflexive, antisymmetric and transitive, the relation 'divides' is a partial order relation.

Example3: (a) The relation $\subseteq$ of a set of inclusion is a partial ordering or any collection of sets since set inclusion has three desired properties:

1. $A \subseteq A$ for any set A.
2. If $A \subseteq B$ and $B \subseteq A$ then $B = A$.
3. If $A \subseteq B$ and $B \subseteq C$ then $A \subseteq C$

(b) The relation $\leq$ on the set R of real no that is Reflexive, Antisymmetric and transitive.

(c) Relation $\leq$ is a Partial Order Relation.

Partial Order Set (POSET):

The set A together with a partial order relation R on the set A and is denoted by (A, R) is called a partial orders set or POSET.

Consider a relation R on a set S satisfying the following properties:

1. R is reflexive, i.e., xRx for every $x \in S$.
2. R is antisymmetric, i.e., if xRy and yRx, then $x = y$.
3. R is transitive, i.e., xRy and yRz, then xRz.

Then R is called a partial order relation, and the set S together with partial order is called a partially order set or POSET and is denoted by $(S, \leq)$.

Example:

1. The set N of natural numbers form a poset under the relation '$\leq$' because firstly $x \leq x$, secondly, if $x \leq y$ and $y \leq x$, then we have $x = y$ and lastly if $x \leq y$ and $y \leq z$, it implies $x \leq z$ for all x, y, $z \in N$.

2. The set N of natural numbers under divisibility i.e., 'x divides y' forms a poset because x/x for every x ∈ N. Also if x/y and y/x, we have x = y. Again if x/y, y/z we have x/z, for every x, y, z ∈ N.
3. Consider a set S = {1, 2} and power set of S is P(S). The relation of set inclusion ⊆ is a partial order. Since, for any sets A, B, C in P (S), firstly we have A ⊆ A, secondly, if A ⊆B and B⊆A, then we have A = B. Lastly, if A ⊆B and B ⊆C,then A⊆C. Hence, (P(S), ⊆) is a poset.

Hasse Diagrams

It is a useful tool, which completely describes the associated partial order. Therefore, it is also called an ordering diagram. It is very easy to convert a directed graph of a relation on a set A to an equivalent Hasse diagram. Therefore, while drawing a Hasse diagram following points must be remembered.

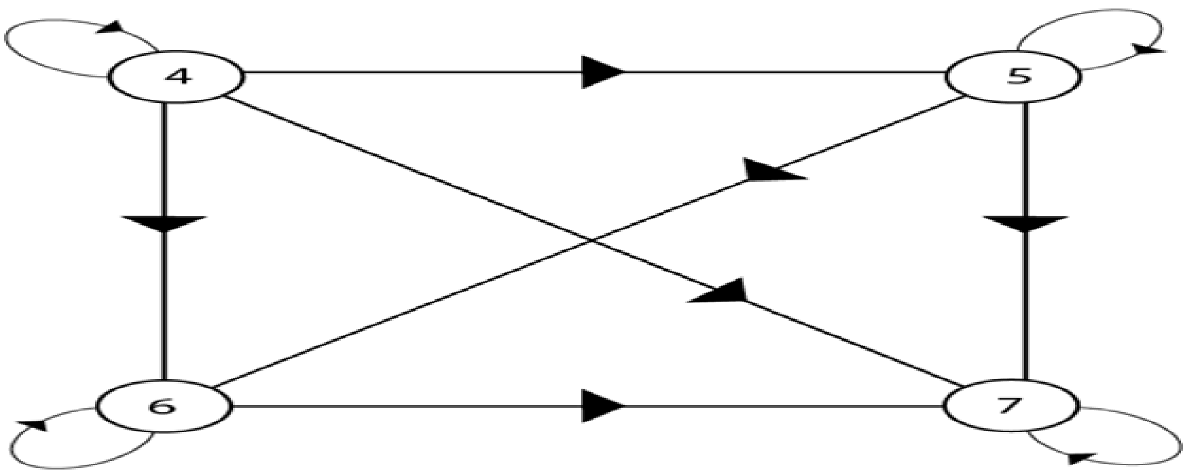
1. The vertices in the Hasse diagram are denoted by points rather than by circles.
2. Since a partial order is reflexive, hence each vertex of A must be related to itself, so the edges from a vertex to itself are deleted in Hasse diagram.
3. Since a partial order is transitive, hence whenever aRb, bRc, we have aRc. Eliminate all edges that are implied by the transitive property in Hasse diagram, i.e., Delete edge from a to c but retain the other two edges.
4. If a vertex 'a' is connected to vertex 'b' by an edge, i.e., aRb, then the vertex 'b' appears above vertex 'a'. Therefore, the arrow may be omitted from the edges in the Hasse diagram.

The Hasse diagram is much simpler than the directed graph of the partial order.

Example: Consider the set A = {4, 5, 6, 7}. Let R be the relation ≤ on A. Draw the directed graph and the Hasse diagram of R.

Solution: The relation ≤ on the set A is given by

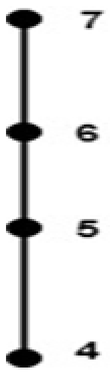
R = {{4, 5}, {4, 6}, {4, 7}, {5, 6}, {5, 7}, {6, 7}, {4, 4}, {5, 5}, {6, 6}, {7, 7}}



To draw the Hasse diagram of partial order, apply the following points:

1. Delete all edges implied by reflexive property i.e. (4, 4), (5, 5), (6, 6), (7, 7)
2. Delete all edges implied by transitive property i.e. (4, 7), (5, 7), (4, 6)
3. Replace the circles representing the vertices by dots.
4. Omit the arrows.

The Hasse diagram is as shown in fig:



Upper Bound: Consider B be a subset of a partially ordered set A. An element x ∈ A is called an upper bound of B if y ≤ x for every y ∈ B.

Lower Bound: Consider B be a subset of a partially ordered set A. An element z ∈ A is called a lower bound of B if z ≤ x for every x ∈ B.

Lattices:

Let L be a non-empty set closed under two binary operations called meet and join, denoted by $\wedge$ and $\vee$. Then L is called a lattice if the following axioms hold where a, b, c are elements in L:

1)Commutative Law: -
(a) $a \wedge b = b \wedge a$ (b) $a \vee b = b \vee a$

2) Associative Law:-
(a) $(a \wedge b) \wedge c = a \wedge (b \wedge c)$ (b) $(a \vee b) \vee c = a \vee (b \vee c)$

3) Absorption Law: -
(a) $a \wedge (a \vee b) = a$ (b) $a \vee (a \wedge b) = a$

Duality:

The dual of any statement in a lattice $(L,\wedge ,\vee)$ is defined to be a statement that is obtained by interchanging $\wedge$ an $\vee$.

For example, the dual of $a \wedge (b \vee a) = a \vee a$ is $a \vee (b \wedge a)= a \wedge a$

Bounded Lattices:

A lattice L is called a bounded lattice if it has greatest element 1 and a least element 0.

Example:

1. The power set P(S) of the set S under the operations of intersection and union is a bounded lattice since $\emptyset$ is the least element of P(S) and the set S is the greatest element of P(S).

2. The set of +ve integer I_+ under the usual order of $\leq$ is not a bounded lattice since it has a least element 1 but the greatest element does not exist.

Properties of Bounded Lattices:

If L is a bounded lattice, then for any element $a \in L$, we have the following identities:

1. $a \vee 1 = 1$
2. $a \wedge 1 = a$
3. $a \vee 0 = a$
4. $a \wedge 0 = 0$

Theorem: Prove that every finite lattice $L = \{a_1, a_2, a_3 \ldots a_n\}$ is bounded.

Proof: We have given the finite lattice:

$$L = \{a_1, a_2, a_3 \ldots a_n\}$$

Thus, the greatest element of Lattices L is $a_1 \vee a_2 \vee a_{3 \vee \ldots \vee an}$.

Also, the least element of lattice L is $a_1 \wedge a_2 \wedge a_3 \wedge \ldots \wedge a_n$.

Since, the greatest and least elements exist for every finite lattice. Hence, L is bounded.

Sub-Lattices:

Consider a non-empty subset L_1 of a lattice L. Then L_1 is called a sub-lattice of L if L_1 itself is a lattice i.e., the operation of L i.e., $a \vee b \in L_1$ and $a \wedge b \in L_1$ whenever $a \in L_1$ and $b \in L_1$.

Example: Consider the lattice of all +ve integers I_+ under the operation of divisibility. The lattice D_n of all divisors of $n > 1$ is a sub-lattice of I_+.

Determine all the sub-lattices of D_{30} that contain at least four elements, $D_{30}=\{1,2,3,5,6,10,15,30\}$.

Solution: The sub-lattices of D_{30} that contain at least four elements are as follows:

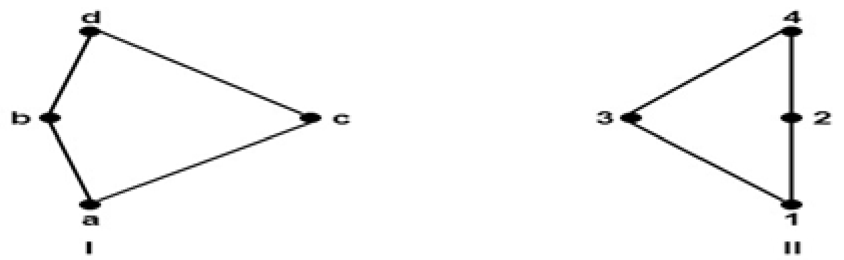
1. {1, 2, 6, 30}
2. {1, 2, 3, 30}
3. {1, 5, 15, 30}
4. {1, 3, 6, 30}
5. {1, 5, 10, 30}
6. {1, 3, 15, 30}
7. {2, 6, 10, 30}

Isomorphic Lattices:

Two lattices L_1 and L_2 are called isomorphic lattices if there is a bijection from L_1 to L_2 i.e., f: $L_1 \longrightarrow L_2$, such that f (a ∧ b) =f(a)∧ f(b) and f (a ∨ b) = f (a) ∨ f (b)

Example: Determine whether the lattices shown in fig are isomorphic.

Solution: The lattices shown in fig are isomorphic. Consider the mapping f = {(a, 1), (b, 2), (c, 3), (d, 4)}.For example f (b ∧ c) = f (a) = 1. Also, we have f (b) ∧ f(c) = 2 ∧ 3 = 1



Distributive Lattice:

A lattice L is called distributive lattice if for any elements a, b and c of L,it satisfies following distributive properties:

1. a ∧ (b ∨ c) = (a ∧ b) ∨ (a ∧ c)
2. a ∨ (b ∧ c) = (a ∨ b) ∧ (a ∨ c)

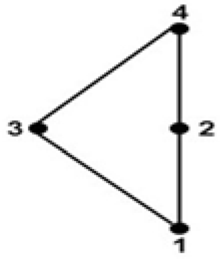
If the lattice L does not satisfies the above properties, it is called a non-distributive lattice.

Example:

1. The power set P (S) of the set S under the operation of intersection and union is a distributive function. Since, a ∩ (b ∪ c) = (a ∩ b) ∪ (a ∩ c)

and, also a ∪ (b ∩ c) = (a ∪ b) ∩ (a ∪c) for any sets a, b and c of P(S).

2. The lattice shown in fig II is a distributive. Since, it satisfies the distributive properties for all ordered triples which are taken from 1, 2, 3, and 4.



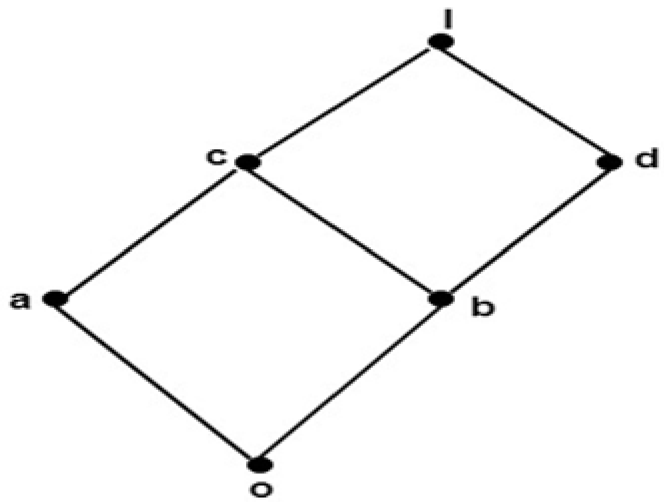
II

Complements and complemented lattices:

Let L be a bounded lattice with lower bound o and upper bound I. Let a be an element if L. An element x in L is called a complement of a if a ∨ x = I and a ∧ x = 0

A lattice L is said to be complemented if L is bounded and every element in L has a complement.

Example: Determine the complement of a and c in fig:



Solution: The complement of a is d. Since, a ∨ d = 1 and a ∧ d = 0

The complement of c does not exist. Since, there does not exist any element c such that c ∨ c'=1 and c ∧ c'= 0.

Modular Lattice:

A lattice (L, ∧,∨) is called a modular lattice if a ∨ (b ∧ c) = (a ∨ b) ∧ c whenever a ≤ c.

Direct Product of Lattices:

Let $(L_1 \vee_1 \wedge_1)$and $(L_2 \vee_2 \wedge_2)$ be two lattices. Then (L, ∧,∨) is the direct product of lattices, where $L = L_1 \times L_2$ in which the binary operation ∨(join) and ∧(meet) on L are such that for any $(a_1,b_1)$and (a_2,b_2) in L.

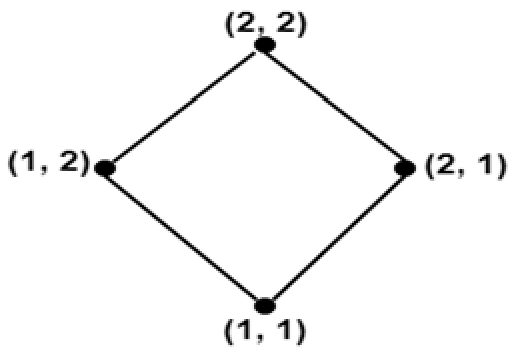
$$(a_1,b_1)\vee(a_2,b_2)=(a_1 \vee_1 a_2,b_1 \vee_2 b_2)$$

and $(a_1,b_1) \wedge (a_2,b_2)=(a_1 \wedge_1 a_2,b_1 \wedge_2 b_2)$.

Example: Consider a lattice (L, ≤) as shown in fig. where L = {1, 2}. Determine the lattices $(L^2, \leq)$, where L^2=L x L.

(L, <=)

Solution: The lattice $(L^2, \leq)$ is shown in fig:



(L2, <=)

BOOLEAN ALGEBRA:

A complemented distributive lattice is known as a Boolean Algebra. It is denoted by (B, ∧,∨,',0,1), where B is a set on which two binary operations ∧ (*) and ∨(+) and a unary operation (complement) are defined. Here 0 and 1 are two distinct elements of B.

Since (B,∧,∨) is a complemented distributive lattice, therefore each element of B has a unique complement.

Properties of Boolean Algebra:

1. Commutative Properties:

(i)a+b = b+a
(ii)a*b=b *a

2. Distributive Properties

(i) a+(b*c)=(a+b)*(a+c)
(ii)a*(b+c)=(a*b)+(a*c)

3. Identity Properties

(i) a+0=a
(ii) a *1=a

4. Complemented Laws:

(i) a+a'=1
(ii)a * a'=0

The table shows all the basic properties of a Boolean algebra (B, *, +, ', 0, 1) for any elements a, b, c belongs to B. The greatest and least elements of B are denoted by 1 and 0 respectively.

1. a ≤b iff a+b=b

2. a ≤b iff a * b = a

3. Idempotent Laws

(i)a+b=a
(ii) a * a = a

4. Commutative Property

(i)a+b=b+a
(ii)a*b=b*a

5. Associative Property

(i)a+(b+c)=(a+b)+c

(ii)a*(b*c)=(a*b)*c

6. Absorption Laws

(i)a+(a*b)=a

(ii)a*(a+b)=a

7. Identity Laws

(i) a+0=a

(ii) a*1=a

8. Null Laws

(i)a*0=0

(ii)a+1=1

9. Distributive Laws

(i)a*(b+c)=(a*b)+(a*c)

(ii) a+(b*c) = (a+b)*(a+c)

10. Complement Laws

(i)0'=1

(ii)1'=0

(iii)a+a'=1

(iv)a*a'=0

11. Involution Law

(a')'=a

12.De Morgan's Laws

(i)(a *b)'=(a' +b')

(ii) (a+b)'=(a' *b')

Simplification Using Algebraic Functions

In this approach, one Boolean expression is minimized into an equivalent expression by applying Boolean identities.

Problem 1

Minimize the following Boolean expression using Boolean identities −

F(A,B,C)=A′B+BC′+BC+AB′C′$F(A,B,C)=A'B+BC'+BC+AB'C'$

Solution

Given,

F(A,B,C)=A′B+BC′+BC+AB′C′$F(A,B,C)=A'B+BC'+BC+AB'C'$

Or,

F(A,B,C)=A′B+(BC′+BC′)+BC+AB′C′$F(A,B,C)=A'B+(BC'+BC')+BC+AB'C'$

[By idempotent law, BC’ = BC’ + BC’]

Or,

F(A,B,C)=A′B+(BC′+BC)+(BC′+AB′C′)$F(A,B,C)=A'B+(BC'+BC)+(BC'+AB'C')$

Or,

F(A,B,C)=A′B+B(C′+C)+C′(B+AB′)$F(A,B,C)=A'B+B(C'+C)+C'(B+AB')$

[By distributive laws]

Or,

F(A,B,C)=A′B+B.1+C′(B+A)$F(A,B,C)=A'B+B.1+C'(B+A)$

[(C' + C) = 1 and absorption law (B + AB')= (B + A)]

Or,

F(A,B,C)=A′B+B+C′(B+A)$F(A,B,C)=A'B+B+C'(B+A)$

[B.1 = B]

Or,

F(A,B,C)=B(A′+1)+C′(B+A)$F(A,B,C)=B(A'+1)+C'(B+A)$

Or,

F(A,B,C)=B.1+C′(B+A)$F(A,B,C)=B.1+C'(B+A)$

[(A' + 1) = 1]

Or,

F(A,B,C)=B+C′(B+A)$F(A,B,C)=B+C'(B+A)$ [As, B.1 = B]

Or,

F(A,B,C)=B+BC′+AC′$F(A,B,C)=B+BC'+AC'$

Or,

F(A,B,C)=B(1+C′)+AC′$F(A,B,C)=B(1+C')+AC'$

Or,

F(A,B,C)=B.1+AC′$F(A,B,C)=B.1+AC'$ [As, (1 + C') = 1]

Or,

F(A,B,C)=B+AC′$F(A,B,C)=B+AC'$ [As, B.1 = B]

So,

F(A,B,C)=B+AC′$F(A,B,C)=B+AC'$is the minimized form.

Problem 2

Minimize the following Boolean expression using Boolean identities −

F(A,B,C)=(A+B)(A+C)$F(A,B,C)=(A+B)(A+C)$

Solution

Given, F(A,B,C)=(A+B)(A+C)$F(A,B,C)=(A+B)(A+C)$

Or, F(A,B,C)=A.A+A.C+B.A+B.C$F(A,B,C)=A.A+A.C+B.A+B.C$ [Applying distributive Rule]

Or, F(A,B,C)=A+A.C+B.A+B.C$F(A,B,C)=A+A.C+B.A+B.C$ [Applying Idempotent Law]

Or, F(A,B,C)=A(1+C)+B.A+B.C$F(A,B,C)=A(1+C)+B.A+B.C$ [Applying distributive Law]

Or, F(A,B,C)=A+B.A+B.C$F(A,B,C)=A+B.A+B.C$ [Applying dominance Law]

Or, F(A,B,C)=(A+1).A+B.C$F(A,B,C)=(A+1).A+B.C$ [Applying distributive Law]

Or, F(A,B,C)=1.A+B.C$F(A,B,C)=1.A+B.C$ [Applying dominance Law]

Or, F(A,B,C)=A+B.C$F(A,B,C)=A+B.C$ [Applying dominance Law]

So, F(A,B,C)=A+BC$F(A,B,C)=A+BC$ is the minimized form.

QUESTIONS: -

- $F(A,B,C)=AB+A'B+AB'$

Step 1: Factor common terms

$$F=B(A+A')+AB'$$

Step 2: Apply complement law $A+A'=1$

$$F=B\cdot 1+AB'=B+AB'$$

Step 3: Apply absorption law $B+AB'=B+B'=1$

$$F=1$$

Final Solution:

$$F(A,B,C)=1$$

- $F(A,B,C)=A'BC+AB'C'+A'B'C$

  Step 1: Group terms with common factors

  $F=A'BC+A'B'C+AB'C'$

  Step 2: Factor common terms

  $F=A'C(B+B')+AB'C'$

  Step 3: Apply complement law $B+B'=1$

  $F=A'C\cdot 1+AB'C'=A'C+AB'C'$

  Final Expression:

  $F(A,B,C)=A'C+AB'C'$

- **Using Boolean identities, reduce the given Boolean expression:**

  $F(X, Y, Z) = X'Y + YZ' + YZ + XY'Z'$

  Solution:

  Given, $F(X, Y, Z) = X'Y + YZ' + YZ + XY'Z'$

  Using the idempotent law, we can write $YZ' = YZ' + YZ'$

  $\Rightarrow F(X, Y, Z) = X'Y+(YZ'+YZ')+YZ + XY'Z'$

  Now, interchange the second and third term, we get

  $\Rightarrow F(X, Y, Z) = X'Y+(YZ'+YZ)+(YZ'+XY'Z')$

  By using distributive law,

  $\Rightarrow F(X, Y, Z) = X'Y+Y(Z'+Z)+Z'(Y+XY')$

  Using $Z' + Z = 1$ and absorption law $(Y + XY')= (Y + X)$,

  $\Rightarrow F(X, Y, Z) = X'Y+Y.1+Z'(Y+X)$

  $\Rightarrow F(X, Y, Z) = X'Y+Y+Z'(Y+X)$ [Since $Y.1 = Y$]

  $\Rightarrow F(X, Y, Z) = Y(X'+1)+Z'(Y+X)$

$\Rightarrow$ F(X, Y, Z) = Y.1+Z′(Y+X) [As (X' + 1) = 1]

$\Rightarrow$ F(X, Y, Z) = Y +Z′(Y+X) [As, Y.1 = Y]

$\Rightarrow$ F(X, Y, Z) = Y+YZ'+XZ'

$\Rightarrow$ F(X, Y, Z) = Y(1+Z′)+XZ′

$\Rightarrow$ F(X, Y, Z) = Y.1+XZ′ [Since (1 + Z') = 1]

$\Rightarrow$ F(X, Y, Z) = Y+XZ′ [Since Y.1 = Y]

Hence, the simplified form of the given Boolean expression is F(X, Y, Z) = Y+XZ′.

- **Reduce the following Boolean expression: F(P ,Q, R)=(P+Q)(P+R)**

  Solution:

  Given, F(P ,Q, R)=(P+Q)(P+R)

  Using distributive law,

  $\Rightarrow$ F(P, Q, R) = P.P + P.R +Q.P + Q.R

  Using Idempotent law,

  $\Rightarrow$ F(P, Q, R) = P + P.R +Q.P + Q.R

  Again using distributive law, we get

  $\Rightarrow$ F(P, Q, R) = P(1+R) + Q.P + Q.R

  Using dominance law, we can write

  $\Rightarrow$ F(P, Q, R) = P + Q.P + Q.R

  Again using distributive law, we get

  $\Rightarrow$ F(P, Q, R) = (P+1).P+ Q.R

  Therefore, using dominance law, we can get the reduced form as follows:

⇒ F(P, Q, R) = 1.P+Q.R

⇒ F(P, Q, R) = P+Q.R

Hence, the reduced form of F(P, Q, R) = (P+Q)(P+R) is F(P, Q, R) = P+Q.R.

- **What is the equivalent expression for the Boolean expression x’y’z +yz+ xz?**

Given Boolean expression: x’y’z +yz+ xz

x’y’z +yz+ xz = z(x’y’+y+x)

Now, apply distributive law for the first two terms inside the bracket.

x’y’z +yz+ xz = z[(x’+y) (y+y’)+ x]

x’y’z +yz+ xz = z [(x’ + y) . 1 + x] [Since A+A’ = 1]

x’y’z +yz+ xz = z [x’ + y + x]

Further x+x’ = 1

So, x’y’z +yz+ xz = z(1+y)

Now, using null law, 1+y = 1

x’y’z +yz+ xz = z.1

Now, using identity law, A.1 = A

Therefore, x’y’z +yz+ xz = z.

Hence, the Boolean expression equivalent to x’y’z +yz+ xz is z.

RECURRENCE RELATIONS: -

A recurrence relation is a functional relation between the independent variable x, dependent variable f(x) and the differences of various order of f (x). A recurrence relation is also called a difference equation, and we will use these two terms interchangeably.

Example1: The equation $f(x + 3h) + 3f(x + 2h) + 6f(x + h) + 9f(x) = 0$ is a recurrence relation.

Example2: The Fibonacci sequence is defined by the recurrence relation $a_r = a_{r-2} + a_{r-1}$, $r \geq 2$, with the initial conditions $a_0=1$ and $a_1=1$.

ORDER OF THE RECURRENCE RELATION: -

The order of the recurrence relation or difference equation is defined to be the difference between the highest and lowest subscripts of f(x) or $a_r=y_k$.

Example1: The equation $13a_r+20a_{r-1}=0$ is a first order recurrence relation.

Example2: The equation $8f(x) + 4f(x + 1) + 8f(x+2) = k(x)$

DEGREE OF THE DIFFERENCE EQUATION: -

The degree of a difference equation is defined to be the highest power of f (x) or $a_r=y_k$

Example1: The equation $y^3_{k+3}+2y^2_{k+2}+2y_{k+1}=0$ has the degree 3, as the highest power of y_k is 3.

Example2: The equation $a^4_r+3a^3_{r-1}+6a^2_{r-2}+4a_{r-3}=0$ has the degree 4, as the highest power of a_r is 4.

Example3: The equation $y_{k+3} + 2y_{k+2} + 4y_{k+1} + 2y_k = k(x)$ has the degree 1, because the highest power of y_k is 1 and its order is 3.

Example4: The equation $f(x+2h) - 4f(x+h) + 2f(x) = 0$ has the degree1 and its order is 2.

LINEAR RECURRENCE RELATIONS WITH CONSTANT COEFFICIENTS

A Recurrence Relations is called linear if its degree is one.

The general form of linear recurrence relation with constant coefficient is

$$C_0\, y_{n+r} + C_1\, y_{n+r-1} + C_2\, y_{n+r-2} + \cdots + C_r\, y_n = R\,(n)$$

Where $C_0, C_1, C_2 \ldots\ldots C_n$ are constant and R (n) is same function of independent variable n.

A solution of a recurrence relation in any function which satisfies the given equation.

Linear Homogeneous Recurrence Relations with Constant Coefficients:

The equation is said to be linear homogeneous difference equation if and only if R (n) = 0 and it will be of order n.

The equation is said to be linear non-homogeneous difference equation if R (n) ≠ 0.

Example1: The equation $a_{r+3} + 6a_{r+2} + 12a_{r+1} + 8a_r = 0$ is a linear non-homogeneous equation of order 3.

Example2: The equation $a_{r+2} - 4a_{r+1} + 4a_r = 3r + 2^r$ is a linear non-homogeneous equation of order 2.

A linear homogeneous difference equation with constant coefficients is given by

$$C_0\, y_n + C_1\, y_{n-1} + C_2\, y_{n-2} + \cdots\ldots\ldots + C_r\, y_{n-r} = 0 \text{ .... equation (i)}$$

Where $\mathbf{C_0, C_1, C_2 \ldots\ldots C_n}$ are constants.

The solution of the equation (i) is of the form $A\alpha_1^K$, where α_1 is the characteristics root and A is constant.

Substitute the values of $A\,\alpha^K$ for y_n in equation (1), we have

$$C_0\,A\alpha^K + C_1\,A\alpha^{K-1} + C_2\,A\alpha^{K-2} + \cdots \ldots + C_r\,A\alpha^{K-r} = 0 \ldots\ldots\ldots \text{equation (ii)}$$

After simplifying equation (ii), we have

$$C_0\,\alpha^r + C_1\,\alpha^{r-1} + C_2\,\alpha^{r-2} + \cdots C_r = 0 \ldots\ldots\ldots\ldots \text{equation (iii)}$$

The equation (iii) is called the characteristics equation of the difference equation.

If α_1 is one of the roots of the characteristics equation, then $A\alpha_1^K$ is a homogeneous solution to the difference equation.

To find the solution of the linear homogeneous difference equations, we have the four cases that are discussed as follows:

Case1: If the characteristic equation has n distinct real rootsα_1, α_2, $\alpha_3, \ldots\ldots\alpha_n$.

Thus, $\alpha_1^K, \alpha_2^K, \ldots\ldots\ \alpha_n^K$ are all solutions of equation (i).

Also, we have $A_1\ \alpha_1^K, A_2\ \alpha_1^K, \ldots\ldots\ldots A_n\ \alpha_n^K$ are all solutions of equation (i). The sums of solutions are also solutions.

Hence, the homogeneous solutions of the difference equation are

$$y_k = A_1\ \alpha_1^K, A_2\ \alpha_2^K, \ldots\ldots\ldots A_n\ \alpha_n^K.$$

Case2: If the characteristics equation has repeated real roots.

If $\alpha_1 = \alpha_2$, then $(A_1 + A_2\,K)\ \alpha_1^K$ is also a solution.

If $\alpha_1 = \alpha_2 = \alpha_3$ then $(A_1 + A_2\,K + A_3\,K^2)\ \alpha_1^K$ is also a solution.

Similarly, if root α_1 is repeated n times, then.

$(A_1+A_2 K+A_3 K^2+......+A_n K_{n-1}) \propto_1^K$

The solution to the homogeneous equation.

Case3: If the characteristics equation has one imaginary root.

If $\alpha+i\beta$ is the root of the characteristics equation, then $\alpha-i\beta$ is also the root, where α and β are real.

Thus, $(\alpha+i\beta)^K$ and $(\alpha-i\beta)^K$ are solutions of the equations. This implies

$(\alpha+i\beta)^K A_1+\alpha-i\beta)^K A_2$

Is also a solution to the characteristics equation, where A_1 and A_2 are constants which are to be determined.

Case4: If the characteristics equation has repeated imaginary roots.

When the characteristics equation has repeated imaginary roots,

$(C_1+C_2 k) (\alpha+i\beta)^K +(C_3+C_4 K)(\alpha-i\beta)^K$

Is the solution to the homogeneous equation.

Example1: Solve the difference equation $a_{r-3ar-1}+2a_{r-2}=0$.

Solution: The characteristics equation is given by

$s^2-3s+2=0$ or $(s-1)(s-2)=0$
$\Rightarrow s = 1, 2$

Therefore, the homogeneous solution of the equation is given by

$a_r=C^1{}_r+C^2.2^r$.

Example2: Solve the difference equation $9y_{K+2}-6y_{K+1}+y_K=0$.

Solution: The characteristics equation is

$9s^2-6s+1=0$ or $(3s-1)^2=0$

$\Rightarrow s = \frac{1}{3}$ and $\frac{1}{3}$

Therefore, the homogeneous solution of the equation is given by

$y_K=(C_1+C_2\, k).\left(\frac{1}{3}\right)^K$

Example3: Solve the difference equation $y_K-y_{K-1}-y_{K-2}=0$.

Solution: The characteristics equation is $s^2-s-1=0$

$s=\frac{1\pm\sqrt{1+4}}{2}=\frac{1\pm\sqrt{5}}{2}$

Therefore, the homogeneous solution of the equation is

$$y_K = C_1\left[\frac{1+\sqrt{5}}{2}\right]^K+C_2\left[\frac{1-\sqrt{5}}{2}\right]^K$$

Example4: Solve the difference equation
$y_{K+4}+4y_{K+3}+8y_{K+2}+8y_{K+1}+4y_K=0$.

Solution: The characteristics equation is $s^4+4s^3+8s^2+8s+4=0$
$(s^2+2s+2)\,(s^2+2s+2)=0$

$s = -1\pm i,-1\pm i$

Therefore, the homogeneous solution of the equation is given by

$$y_K=(C_1+C_2\, K)(-1+i)^K+(C_3 +C_4\, K)(-1-i)^K$$

GENERATING FUNCTIONS

Generating function is a method to solve the recurrence relations.

Let us consider, the sequence a_0, a_1, a_2....a_r of real numbers. For some interval of real numbers containing zero values at t is given, the function G(t) is defined by the series

$$G(t)= a_0, a_1t+a_2 t^2+\cdots+a_r t_r+\text{...........equation (i)}$$

This function G(t) is called the generating function of the sequence a_r.

Now, for the constant sequence 1, 1, 1, 1.....the generating function is

$$G(t) = \frac{1}{(1-t)}$$

It can be expressed as

$$G(t) =(1-t)^{-1}=1+t+t^2 +t^3+t^4+\cdots\text{[By binomial expansion]}$$

Comparing, this with equation (i), we get

$a_0=1,a_1=1,a_2=1$ and so on.

For, the constant sequence 1,2,3,4,5,..the generating function is

$G(t) = \frac{1}{(1-t)^2}$ because it can be expressed as

$G(t) =(1-t)^{-2}=1+2t+3t^2 +4t^3+\cdots+(r+1) t^r$

Comparing, this with equation (i), we get

$a_0=1,a_1=2,a_2=3,a_3=4$ and so on.

The generating function of Z^r,($Z\neq0$ and Z is a constant)is given by

$G(t)= 1+Zt+Z^2 t^2+Z^3 t^3+\cdots+Z^r t^r$

$G(t)=\frac{1}{(1 - Zt)}$ [Assume $|Zt|<1$]

So, $G(t)=\frac{1}{(1 - Zt)}$ generates $Z^r,Z\neq0$

Also,If $a^{(1)}_r$ has the generating function $G_1(t)$ and $a^{(2)}_r$ has the generating function $G_2(t)$, then $\lambda_1 a^{(1)}_r+\lambda_2 a^{(2)}_r$ has the generating function $\lambda_1 G_1(t)+ \lambda_2 G_2(t)$. Here λ_1 and λ_2 are constants.

Application Areas:

Generating functions can be used for the following purposes -

- For solving recurrence relations
- For proving some of the combinatorial identities
- For finding asymptotic formulae for terms of sequences

Example: Solve the recurrence relation $a_{r+2}-3a_{r+1}+2a_r=0$

By the method of generating functions with the initial conditions $a_0=2$ and $a_1=3$.

Solution: Let us assume that

$$G(t)=\sum_0^\infty a_r t^r$$

Multiply equation (i) by t^r and summing from r = 0 to ∞, we have

$$\sum_0^\infty a_{r+2} t^r - 3\sum_0^\infty a_{r+1} t^r + 2\sum_0^\infty a_r t^r=0$$

$(a_2+a_3 t+a_4 t^2+\cdots)-3(a_1+a_2 t+a_3 t^2+\cdots)+2(a_0+a_1 t+a_2 t^2+\cdots)=0$
$[\therefore G(t)=a_0+a_1 t+a_2 t^2+\cdots]$

$$\therefore \quad \frac{G(t)-a_0-a_1 t}{t^2} - 3\left(\frac{G(t)-a_0}{t}\right) +2G(t)=0 \text{............equation (ii)}$$

Now, put $a_0=2$ and $a_1=3$ in equation (ii) and solving, we get

$$G(t) = \frac{2-3t}{1-3t+2t^2} \text{ or } G(t) = \frac{2-3t}{(1-t)(1-2t)}$$

Now, Let $\frac{2-3t}{(1-t)(1-2t)} = \frac{A}{1-t} + \frac{B}{1-2t}$

i.e., $\quad 2-3t = A(1-2t)+B(1-t)$........................equation (iii)

Put t=1 on both sides of equation (iii) to find A. Hence

$-1 = -A \quad \therefore A = 1$

Put $t=\frac{1}{2}$ on both sides of equation (iii) to find B. Hence

$\frac{1}{2}=\frac{1}{2} B \quad \therefore B = 1$

Thus $G(t) = \frac{1}{1-t} + \frac{1}{1-2t}$. Hence, $a_r = 1+2^r$.

- **<u>ADVANCED MASTER THEOREM FOR DIVIDE AND CONQUER RECURRENCES: -</u>**

Divide and conquer is an algorithm that works on the paradigm based on recursively branching problem into multiple sub-problems of similar type that can be solved easily.

Example

Let's take an example to learn more about the divide and conquer technique −

```
function recursive(input x size n)
  if(n < k)
    Divide the input into m subproblems of size n/p.
    and call f recursively of each sub problem
  else
    Solve x and return
```

Combine the results of all subproblems and return the solution to the original problem.

Explanation − In the above problem, the problem set is to be subdivided into smaller subproblems that can be solved easily.

MASTERS THEOREM FOR DIVIDE AND CONQUER is an analysis theorem that can be used to determine a big-0 value for recursive relation algorithms. It is used to find the time required by the algorithm and represent it in asymptotic notation form.

Example of runtime value of the problem in the above example −

T(n) = f(n) + m.T(n/p)

For most of the recursive algorithm, you will be able to find the Time complexity For the algorithm using the master's theorem, but there are some cases master's theorem may not be applicable. These are the cases in which the master's theorem is not applicable. When the problem T(n) is not monotone, for example, T(n) = sin n. Problem function f(n) is not a polynomial.

As the master theorem to find time complexity is not hot efficient in these cases, and advanced master theorem for recursive recurrence was designed. It is design to handle recurrence problem of the form −

T(n) = aT(n/b) + ø((n^k)logpn)

Where n is the size of the problem.

a = number of subproblems in recursion, a > 0

n/b = size of each subproblem b > 1, k >= 0 and p is a real number.

For solving this type of problem, we will use the following solutions,

- If $a>b_k$, then T(n) = ∅ (nlogba)
- If $a = b_k$, then
  - If p> -1, then T(n) = ∅(nlogba $\log^{p+1}$n)
  - If p = -1, then T(n) = ∅($n\log_{ba}$ loglogn)
  - If p < -1, then T(n) = ∅($n\log_{ba}$)
- If $a<b_k$, then

- If p> = 0, then T(n)= Ø($n_k$log$_p$n)
- If p< 0, then T(n) = Ø(nk)

Using the advanced master algorithm, we will calculate the complexity of some algorithms −

Binary search − t(n) = θ(logn)

Merge sort − T(n) = θ(nlogn)