**Discrete Mathematical Structure**

**Department of Computer Science**

**Milan Kumar Nayak**

**Cont.6370158954**

**Basudev Godabari Degree College**

At /- Kesaibahal , Dist/- Sambalpur , Pin – 768228

bgdegreecollege@gmail.com **,Mob – 9438000807**

**Unit-1**

**:::::::::::Prepositional Logic – Definition::::::::**

A proposition is a collection of declarative statements that has either a truth value "true” or a truth value "false". A propositional consists of propositional variables and connectives. We denote the propositional variables by capital letters (A, B, etc). The connectives connect the propositional variables.

Some examples of Propositions are given below −

* "Man is Mortal", it returns truth value “TRUE”
* "12 + 9 = 3 – 2", it returns truth value “FALSE”

The following is not a Proposition −

* "A is less than 2". It is because unless we give a specific value of A, we cannot say whether the statement is true or false.

**Connectives:-** In propositional logic generally we use five connectives which are −

* OR (∨∨)
* AND (∧∧)
* Negation/ NOT (¬¬)
* Implication / if-then (→→)
* If and only if (⇔⇔).

**OR (**∨∨**)** –

The OR operation of two propositions A and B (written as A∨BA∨B) is true if at least any of the propositional variable A or B is true.

The truth table is as follows −

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A ∨ B** |
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

**AND (**∧∧**)** –

The AND operation of two propositions A and B (written as A∧BA∧B) is true if both the propositional variable A and B is true.

The truth table is as follows −

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A ∧ B** |
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

**Negation (**¬¬**)** –

The negation of a proposition A (written as ¬A¬A) is false when A is true and is true when A is false.

The truth table is as follows −

|  |  |
| --- | --- |
| **A** | **¬ A** |
| True | False |
| False | True |

**Implication / if-then (**→→**)** –

An implication A→BA→B is the proposition “if A, then B”. It is false if A is true and B is false. The rest cases are true.

The truth table is as follows −

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A → B** |
| True | True | True |
| True | False | False |
| False | True | True |
| False | False | True |

**If and only if (**⇔⇔**)** –

A⇔BA⇔B is bi-conditional logical connective which is true when p and q are same, i.e. both are false or both are true.

The truth table is as follows −

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A ⇔ B** |
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | True |

**Tautologies**

A Tautology is a formula which is always true for every value of its propositional variables.

**Example** − Prove [(A→B)∧A]→B[(A→B)∧A]→B is a tautology

The truth table is as follows −

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **A → B** | **(A → B) ∧ A** | **[( A → B ) ∧ A] → B** |
| True | True | True | True | True |
| True | False | False | False | True |
| False | True | True | False | True |
| False | False | True | False | True |

As we can see every value of [(A→B)∧A]→B[(A→B)∧A]→B is "True", it is a tautology.

**Contradictions:-**

A Contradiction is a formula which is always false for every value of its propositional variables.

**Example** − Prove (A∨B)∧[(¬A)∧(¬B)](A∨B)∧[(¬A)∧(¬B)] is a contradiction

The truth table is as follows −

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **A ∨ B** | **¬ A** | **¬ B** | **(¬ A) ∧ ( ¬ B)** | **(A ∨ B) ∧ [( ¬ A) ∧ (¬ B)]** |
| True | True | True | False | False | False | False |
| True | False | True | False | True | False | False |
| False | True | True | True | False | False | False |
| False | False | False | True | True | True | False |

As we can see every value of (A∨B)∧[(¬A)∧(¬B)](A∨B)∧[(¬A)∧(¬B)] is “False”, it is a contradiction.

**Contingency:-**

A Contingency is a formula which has both some true and some false values for every value of its propositional variables.

**Example** − Prove (A∨B)∧(¬A)(A∨B)∧(¬A) a contingency

The truth table is as follows −

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **A ∨ B** | **¬ A** | **(A ∨ B) ∧ (¬ A)** |
| True | True | True | False | False |
| True | False | True | False | False |
| False | True | True | True | True |
| False | False | False | True | False |

As we can see every value of (A∨B)∧(¬A)(A∨B)∧(¬A) has both “True” and “False”, it is a contingency.

 **::::::::::::::Propositional Equivalences:::::::::::::**

Two statements X and Y are logically equivalent if any of the following two conditions hold −

* The truth tables of each statement have the same truth values.
* The bi-conditional statement X⇔YX⇔Y is a tautology.

**Example** − Prove ¬(A∨B)and[(¬A)∧(¬B)]¬(A∨B)and[(¬A)∧(¬B)] are equivalent

Testing by 1st method (Matching truth table)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **A ∨ B** | **¬ (A ∨ B)** | **¬ A** | **¬ B** | **[(¬ A) ∧ (¬ B)]** |
| True | True | True | False | False | False | False |
| True | False | True | False | False | True | False |
| False | True | True | False | True | False | False |
| False | False | False | True | True | True | True |

Here, we can see the truth values of ¬(A∨B)and[(¬A)∧(¬B)]¬(A∨B)and[(¬A)∧(¬B)] are same, hence the statements are equivalent.

Testing by 2nd method (Bi-conditionality)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **¬ (A ∨ B )** | **[(¬ A) ∧ (¬ B)]** | **[¬ (A ∨ B)] ⇔ [(¬ A ) ∧ (¬ B)]** |
| True | True | False | False | True |
| True | False | False | False | True |
| False | True | False | False | True |
| False | False | True | True | True |

As [¬(A∨B)]⇔[(¬A)∧(¬B)][¬(A∨B)]⇔[(¬A)∧(¬B)] is a tautology, the statements are equivalent.

**Inverse, Converse, and Contra-positive**

Implication / if-then (→)(→) is also called a conditional statement. It has two parts −

* Hypothesis, p
* Conclusion, q

As mentioned earlier, it is denoted as p→qp→q.

**Example of Conditional Statement** − “If you do your homework, you will not be punished.” Here, "you do your homework" is the hypothesis, p, and "you will not be punished" is the conclusion, q.

**Inverse** –

 An inverse of the conditional statement is the negation of both the hypothesis and the conclusion. If the statement is “If p, then q”, the inverse will be “If not p, then not q”. Thus the inverse of p→qp→q is ¬p→¬q¬p→¬q.

**Example** –

 The inverse of “If you do your homework, you will not be punished” is “If you do not do your homework, you will be punished.”

**Converse** –

The converse of the conditional statement is computed by interchanging the hypothesis and the conclusion. If the statement is “If p, then q”, the converse will be “If q, then p”. The converse of p→qp→q is q→pq→p.

**Example** –

The converse of "If you do your homework, you will not be punished" is "If you will not be punished, you do your homework”.

**Contra-positive** –

 The contra-positive of the conditional is computed by interchanging the hypothesis and the conclusion of the inverse statement. If the statement is “If p, then q”, the contra-positive will be “If not q, then not p”. The contra-positive of p→qp→q is ¬q→¬p¬q→¬p.

**Example** − The Contra-positive of " If you do your homework, you will not be punished” is "If you are punished, you did not do your homework”.

**Duality Principle**

Duality principle states that for any true statement, the dual statement obtained by interchanging unions into intersections (and vice versa) and interchanging Universal set into Null set (and vice versa) is also true. If dual of any statement is the statement itself, it is said **self-dual** statement.

**Example** − The dual of (A∩B)∪C(A∩B)∪C is (A∪B)∩C(A∪B)∩C

**Normal Forms**

We can convert any proposition in two normal forms −

* Conjunctive normal form
* Disjunctive normal form

Conjunctive Normal Form

A compound statement is in conjunctive normal form if it is obtained by operating AND among variables (negation of variables included) connected with ORs. In terms of set operations, it is a compound statement obtained by Intersection among variables connected with Unions.

**Examples**

* (A∨B)∧(A∨C)∧(B∨C∨D)(A∨B)∧(A∨C)∧(B∨C∨D)
* (P∪Q)∩(Q∪R)(P∪Q)∩(Q∪R)

Disjunctive Normal Form

A compound statement is in disjunctive normal form if it is obtained by operating OR among variables (negation of variables included) connected with ANDs. In terms of set operations, it is a compound statement obtained by Union among variables connected with Intersections.

**Examples**

* (A∧B)∨(A∧C)∨(B∧C∧D)(A∧B)∨(A∧C)∨(B∧C∧D)
* (P∩Q)∪(Q∩R)

**:::::::::::::::Predicate:::::::::::**

A predicate is an expression of one or more variables defined on some specific domain. A predicate with variables can be made a proposition by either assigning a value to the variable or by quantifying the variable.

The following are some examples of predicates −

* Let E(x, y) denote "x = y"
* Let X(a, b, c) denote "a + b + c = 0"
* Let M(x, y) denote "x is married to y"

**Well Formed Formula**

Well Formed Formula (wff) is a predicate holding any of the following −

* All propositional constants and propositional variables are wffs
* If x is a variable and Y is a wff, ∀xY∀xY and ∃xY∃xY are also wff
* Truth value and false values are wffs
* Each atomic formula is a wff
* All connectives connecting wffs are wffs

**::::::::::Quantifiers:::::::::::**

The variable of predicates is quantified by quantifiers. There are two types of quantifier in predicate logic − Universal Quantifier and Existential Quantifier.

**Universal Quantifier**

Universal quantifier states that the statements within its scope are true for every value of the specific variable. It is denoted by the symbol ∀∀.

∀xP(x)∀xP(x) is read as for every value of x, P(x) is true.

**Example** –

 "Man is mortal" can be transformed into the propositional form ∀xP(x)∀xP(x) where P(x) is the predicate which denotes x is mortal and the universe of discourse is all men.

**Existential Quantifier**

Existential quantifier states that the statements within its scope are true for some values of the specific variable. It is denoted by the symbol ∃∃.

∃xP(x)∃xP(x) is read as for some values of x, P(x) is true.

**Example** –

"Some people are dishonest" can be transformed into the propositional form ∃xP(x)∃xP(x) where P(x) is the predicate which denotes x is dishonest and the universe of discourse is some people.

**::::::::::::::Nested Quantifiers::::::::::**

If we use a quantifier that appears within the scope of another quantifier, it is called nested quantifier.

**Example**

* ∀ a∃bP(x,y)∀ a∃bP(x,y) where P(a,b)P(a,b) denotes a+b=0a+b=0
* ∀ a∀b∀cP(a,b,c)∀ a∀b∀cP(a,b,c) where P(a,b)P(a,b) denotes a+(b+c)=(a+b)+ca+(b+c)=(a+b)+c

**Note** − ∀a∃bP(x,y)≠∃a∀bP(x,y)

**::::::::::: Rules of Inference :::::::::::**

* To deduce new statements from the statements whose truth that we already know, **Rules of Inference** are used.

Mathematical logic is often used for logical proofs. Proofs are valid arguments that determine the truth values of mathematical statements.

An argument is a sequence of statements. The last statement is the conclusion and all its preceding statements are called premises (or hypothesis). The symbol “∴∴”, (read therefore) is placed before the conclusion. A valid argument is one where the conclusion follows from the truth values of the premises.

Rules of Inference provide the templates or guidelines for constructing valid arguments from the statements that we already have.

Table of Rules of Inference

|  |  |  |  |
| --- | --- | --- | --- |
| **Rule of Inference** | **Name** | **Rule of Inference** | **Name** |
| P∴P∨QP∴P∨Q | Addition | P∨Q¬P∴QP∨Q¬P∴Q | Disjunctive Syllogism |
| PQ∴P∧QPQ∴P∧Q | Conjunction | P→QQ→R∴P→RP→QQ→R∴P→R | Hypothetical Syllogism |
| P∧Q∴PP∧Q∴P | Simplification | (P→Q)∧(R→S)P∨R∴Q∨S(P→Q)∧(R→S)P∨R∴Q∨S | Constructive Dilemma |
| P→QP∴QP→QP∴Q | Modus Ponens | (P→Q)∧(R→S)¬Q∨¬S∴¬P∨¬R(P→Q)∧(R→S)¬Q∨¬S∴¬P∨¬R | Destructive Dilemma |
| P→Q¬Q∴¬PP→Q¬Q∴¬P | Modus Tollens |  |  |

**Addition**

If P is a premise, we can use Addition rule to derive P∨QP∨Q.

P∴P∨QP∴P∨Q

Example

Let P be the proposition, “He studies very hard” is true

Therefore − "Either he studies very hard Or he is a very bad student." Here Q is the proposition “he is a very bad student”.

**Conjunction**

If P and Q are two premises, we can use Conjunction rule to derive P∧QP∧Q.

PQ∴P∧QPQ∴P∧Q

Example

Let P − “He studies very hard”

Let Q − “He is the best boy in the class”

Therefore − "He studies very hard and he is the best boy in the class"

**Simplification**

If P∧QP∧Q is a premise, we can use Simplification rule to derive P.

P∧Q∴PP∧Q∴P

Example

"He studies very hard and he is the best boy in the class", P∧QP∧Q

Therefore − "He studies very hard"

**Modus Ponens**

If P and P→QP→Q are two premises, we can use Modus Ponens to derive Q.

P→QP∴QP→QP∴Q

Example

"If you have a password, then you can log on to facebook", P→QP→Q

"You have a password", P

Therefore − "You can log on to facebook"

**Modus Tollens**

If P→QP→Q and ¬Q¬Q are two premises, we can use Modus Tollens to derive ¬P¬P.

P→Q¬Q∴¬PP→Q¬Q∴¬P

Example

"If you have a password, then you can log on to facebook", P→QP→Q

"You cannot log on to facebook", ¬Q¬Q

Therefore − "You do not have a password "

## :::::::::::Mathematical Induction ::::::::::

* **Mathematical induction**, is a technique for proving results or establishing statements for natural numbers. This part illustrates the method through a variety of examples.

**Mathematical Induction** is a mathematical technique which is used to prove a statement, a formula or a theorem is true for every natural number.

The technique involves two steps to prove a statement, as stated below −

**Step 1(Base step)** − It proves that a statement is true for the initial value.

**Step 2(Inductive step)** − It proves that if the statement is true for the nth iteration (or number *n*), then it is also true for *(n+1)th* iteration ( or number *n+1*).

## How to Do It

**Step 1** − Consider an initial value for which the statement is true. It is to be shown that the statement is true for n = initial value.

**Step 2** − Assume the statement is true for any value of *n = k*. Then prove the statement is true for *n = k+1*. We actually break *n = k+1* into two parts, one part is *n = k* (which is already proved) and try to prove the other part.

###  Problem 1:

3n−13n−1 is a multiple of 2 for n = 1, 2, ...

**Solution**

**Step 1** − For n=1,31−1=3−1=2n=1,31−1=3−1=2 which is a multiple of 2

**Step 2** − Let us assume 3n−13n−1 is true for n=kn=k, Hence, 3k−13k−1 is true (It is an assumption)

We have to prove that 3k+1−13k+1−1 is also a multiple of 2

3k+1−1=3×3k−1=(2×3k)+(3k−1)3k+1−1=3×3k−1=(2×3k)+(3k−1)

The first part (2×3k)(2×3k) is certain to be a multiple of 2 and the second part (3k−1)(3k−1) is also true as our previous assumption.

Hence, 3k+1–13k+1–1 is a multiple of 2.

So, it is proved that 3n–13n–1 is a multiple of 2.

###  Problem 2:

1+3+5+...+(2n−1)=n21+3+5+...+(2n−1)=n2 for n=1,2,…n=1,2,…

**Solution**

**Step 1** − For n=1,1=12n=1,1=12, Hence, step 1 is satisfied.

**Step 2** − Let us assume the statement is true for n=kn=k.

Hence, 1+3+5+⋯+(2k−1)=k21+3+5+⋯+(2k−1)=k2 is true (It is an assumption)

We have to prove that 1+3+5+...+(2(k+1)−1)=(k+1)21+3+5+...+(2(k+1)−1)=(k+1)2 also holds

1+3+5+⋯+(2(k+1)−1)1+3+5+⋯+(2(k+1)−1)

=1+3+5+⋯+(2k+2−1)=1+3+5+⋯+(2k+2−1)

=1+3+5+⋯+(2k+1)=1+3+5+⋯+(2k+1)

=1+3+5+⋯+(2k−1)+(2k+1)=1+3+5+⋯+(2k−1)+(2k+1)

=k2+(2k+1)=k2+(2k+1)

=(k+1)2=(k+1)2

So, 1+3+5+⋯+(2(k+1)−1)=(k+1)21+3+5+⋯+(2(k+1)−1)=(k+1)2 hold which satisfies the step 2.

Hence, 1+3+5+⋯+(2n−1)=n21+3+5+⋯+(2n−1)=n2 is proved.

### Problem 3:-

Prove that (ab)n=anbn(ab)n=anbn is true for every natural number nn

**Solution**

**Step 1** − For n=1,(ab)1=a1b1=abn=1,(ab)1=a1b1=ab, Hence, step 1 is satisfied.

**Step 2** − Let us assume the statement is true for n=kn=k, Hence, (ab)k=akbk(ab)k=akbk is true (It is an assumption).

We have to prove that (ab)k+1=ak+1bk+1(ab)k+1=ak+1bk+1 also hold

Given, (ab)k=akbk(ab)k=akbk

Or, (ab)k(ab)=(akbk)(ab)(ab)k(ab)=(akbk)(ab) [Multiplying both side by 'ab']

Or, (ab)k+1=(aak)(bbk)(ab)k+1=(aak)(bbk)

Or, (ab)k+1=(ak+1bk+1)(ab)k+1=(ak+1bk+1)

Hence, step 2 is proved.

So, (ab)n=anbn(ab)n=anbn is true for every natural number n.

**Unit-2**

**Set And Function**

* German mathematician **G. Cantor** introduced the concept of sets. He had defined a set as a collection of definite and distinguishable objects selected by the means of certain rules or description.

 **Set** theory forms the basis of several other fields of study like counting theory,

relations, graph theory and finite state machines. In this chapter, we will cover the

 different aspects of **Set Theory**.

**::::::::::Set :::::::::**

 A set is an unordered collection of different elements. A set can be written explicitly by listing its elements using set bracket. If the order of the elements is changed or any element of a set is repeated, it does not make any changes in the set.

Some Example of Sets

* A set of all positive integers
* A set of all the planets in the solar system
* A set of all the states in India
* A set of all the lowercase letters of the alphabet

Representation of a Set

Sets can be represented in two ways −

* Roster or Tabular Form
* Set Builder Notation

Roster or Tabular Form

The set is represented by listing all the elements comprising it. The elements are enclosed within braces and separated by commas.

**Example 1** − Set of vowels in English alphabet, A={a,e,i,o,u}A={a,e,i,o,u}

**Example 2** − Set of odd numbers less than 10, B={1,3,5,7,9}B={1,3,5,7,9}

Set Builder Notation

The set is defined by specifying a property that elements of the set have in common. The set is described as A={x:p(x)}A={x:p(x)}

**Example 1** − The set {a,e,i,o,u}{a,e,i,o,u} is written as −

A={x:x is a vowel in English alphabet}A={x:x is a vowel in English alphabet}

**Example 2** − The set {1,3,5,7,9}{1,3,5,7,9} is written as −

B={x:1≤x<10 and (x%2)≠0}B={x:1≤x<10 and (x%2)≠0}

If an element x is a member of any set S, it is denoted by x∈Sx∈S and if an element y is not a member of set S, it is denoted by y∉Sy∉S.

**Example** − If S={1,1.2,1.7,2},1∈SS={1,1.2,1.7,2},1∈S but 1.5∉S1.5∉S

Some Important Sets

**N** − the set of all natural numbers = {1,2,3,4,.....}{1,2,3,4,.....}

**Z** − the set of all integers = {.....,−3,−2,−1,0,1,2,3,.....}{.....,−3,−2,−1,0,1,2,3,.....}

**Z+** − the set of all positive integers

**Q** − the set of all rational numbers

**R** − the set of all real numbers

**W** − the set of all whole numbers

Cardinality of a Set

Cardinality of a set S, denoted by |S||S|, is the number of elements of the set. The number is also referred as the cardinal number. If a set has an infinite number of elements, its cardinality is ∞∞.

**Example** − |{1,4,3,5}|=4,|{1,2,3,4,5,…}|=∞|{1,4,3,5}|=4,|{1,2,3,4,5,…}|=∞

If there are two sets X and Y,

* |X|=|Y||X|=|Y| denotes two sets X and Y having same cardinality. It occurs when the number of elements in X is exactly equal to the number of elements in Y. In this case, there exists a bijective function ‘f’ from X to Y.
* |X|≤|Y||X|≤|Y| denotes that set X’s cardinality is less than or equal to set Y’s cardinality. It occurs when number of elements in X is less than or equal to that of Y. Here, there exists an injective function ‘f’ from X to Y.
* |X|<|Y||X|<|Y| denotes that set X’s cardinality is less than set Y’s cardinality. It occurs when number of elements in X is less than that of Y. Here, the function ‘f’ from X to Y is injective function but not bijective.
* If |X|≤|Y|If |X|≤|Y| and |X|≥|Y||X|≥|Y| then |X|=|Y||X|=|Y|. The sets X and Y are commonly referred as equivalent sets.

Types of Sets :-

Sets can be classified into many types. Some of which are finite, infinite, subset, universal, proper, singleton set, etc.

Finite Set

A set which contains a definite number of elements is called a finite set.

**Example** − S={x|x∈NS={x|x∈N and 70>x>50}70>x>50}

Infinite Set

A set which contains infinite number of elements is called an infinite set.

**Example** − S={x|x∈NS={x|x∈N and x>10}x>10}

Subset

A set X is a subset of set Y (Written as X⊆YX⊆Y) if every element of X is an element of set Y.

**Example 1** − Let, X={1,2,3,4,5,6}X={1,2,3,4,5,6} and Y={1,2}Y={1,2}. Here set Y is a subset of set X as all the elements of set Y is in set X. Hence, we can write Y⊆XY⊆X.

**Example 2** − Let, X={1,2,3}X={1,2,3} and Y={1,2,3}Y={1,2,3}. Here set Y is a subset (Not a proper subset) of set X as all the elements of set Y is in set X. Hence, we can write Y⊆XY⊆X.

**Proper Subset**

The term “proper subset” can be defined as “subset of but not equal to”. A Set X is a proper subset of set Y (Written as X⊂YX⊂Y) if every element of X is an element of set Y and |X|<|Y||X|<|Y|.

**Example** − Let, X={1,2,3,4,5,6}X={1,2,3,4,5,6} and Y={1,2}Y={1,2}. Here set Y⊂XY⊂X since all elements in YY are contained in XX too and XX has at least one element is more than set YY.

**Universal Set**

It is a collection of all elements in a particular context or application. All the sets in that context or application are essentially subsets of this universal set. Universal sets are represented as UU.

**Example** − We may define UU as the set of all animals on earth. In this case, set of all mammals is a subset of UU, set of all fishes is a subset of UU, set of all insects is a subset of UU, and so on.

**Empty Set or Null Set**

An empty set contains no elements. It is denoted by ∅∅. As the number of elements in an empty set is finite, empty set is a finite set. The cardinality of empty set or null set is zero.

**Example** − S={x|x∈NS={x|x∈N and 7<x<8}=∅7<x<8}=∅

Singleton Set or Unit Set

Singleton set or unit set contains only one element. A singleton set is denoted by {s}{s}.

**Example** − S={x|x∈N, 7<x<9}S={x|x∈N, 7<x<9} = {8}{8}

**Equal Set**

If two sets contain the same elements they are said to be equal.

**Example** − If A={1,2,6}A={1,2,6} and B={6,1,2}B={6,1,2}, they are equal as every element of set A is an element of set B and every element of set B is an element of set A.

**Equivalent Set**

If the cardinalities of two sets are same, they are called equivalent sets.

**Example** − If A={1,2,6}A={1,2,6} and B={16,17,22}B={16,17,22}, they are equivalent as cardinality of A is equal to the cardinality of B. i.e. |A|=|B|=3|A|=|B|=3

**Overlapping Set**

Two sets that have at least one common element are called overlapping sets.

In case of overlapping sets −

* n(A∪B)=n(A)+n(B)−n(A∩B)n(A∪B)=n(A)+n(B)−n(A∩B)
* n(A∪B)=n(A−B)+n(B−A)+n(A∩B)n(A∪B)=n(A−B)+n(B−A)+n(A∩B)
* n(A)=n(A−B)+n(A∩B)n(A)=n(A−B)+n(A∩B)
* n(B)=n(B−A)+n(A∩B)n(B)=n(B−A)+n(A∩B)

**Example** − Let, A={1,2,6}A={1,2,6} and B={6,12,42}B={6,12,42}. There is a common element ‘6’, hence these sets are overlapping sets.

**Disjoint Set**

Two sets A and B are called disjoint sets if they do not have even one element in common. Therefore, disjoint sets have the following properties −

* n(A∩B)=∅n(A∩B)=∅
* n(A∪B)=n(A)+n(B)n(A∪B)=n(A)+n(B)

**Example** − Let, A={1,2,6}A={1,2,6} and B={7,9,14}B={7,9,14}, there is not a single common element, hence these sets are overlapping sets.

Venn diagram, invented in 1880 by John Venn, is a schematic diagram that shows all possible logical relations between different mathematical sets.

**Examples**

**::::::::::Set Operations:::::::::**

Set Operations include Set Union, Set Intersection, Set Difference, Complement of Set, and Cartesian Product.

**Set Union**

The union of sets A and B (denoted by A∪BA∪B) is the set of elements which are in A, in B, or in both A and B. Hence, A∪B={x|x∈A OR x∈B}A∪B={x|x∈A OR x∈B}.

**Example** –

 If A={10,11,12,13}A={10,11,12,13} and B = {13,14,15}{13,14,15}, then A∪B={10,11,12,13,14,15}A∪B={10,11,12,13,14,15}. (The common element occurs only once)



**Set Intersection**

The intersection of sets A and B (denoted by A∩BA∩B) is the set of elements which are in both A and B. Hence, A∩B={x|x∈A AND x∈B}A∩B={x|x∈A AND x∈B}.

**Example** − If A={11,12,13}A={11,12,13} and B={13,14,15}B={13,14,15}, then A∩B={13}A∩B={13}.



 **Set Difference/ Relative Complement**

 The set difference of sets A and B (denoted by A–BA–B) is the set of elements which are only in A but not in B. Hence, A−B={x|x∈A AND x∉B}A−B={x|x∈A AND x∉B}.

**Example** –

If A={10,11,12,13}A={10,11,12,13} and B={13,14,15}B={13,14,15}, then (A−B)={10,11,12}(A−B)={10,11,12} and (B−A)={14,15}(B−A)={14,15}. Here, we can see (A−B)≠(B−A)(A−B)≠(B−A)



 **Complement of a Set**

 The complement of a set A (denoted by A′A′) is the set of elements which are not in set A. Hence, A′={x|x∉A}A′={x|x∉A}.

More specifically, A′=(U−A)A′=(U−A) where UU is a universal set which contains all objects.

**Example** − If A={x|x belongstosetofoddintegers}A={x|x belongstosetofoddintegers} then A′={y|y doesnotbelongtosetofoddintegers}A′={y|y doesnotbelongtosetofoddintegers}



**Cartesian Product / Cross Product**

The Cartesian product of n number of sets A1,A2,…AnA1,A2,…An denoted as A1×A2⋯×AnA1×A2⋯×An can be defined as all possible ordered pairs (x1,x2,…xn)(x1,x2,…xn) where x1∈A1,x2∈A2,…xn∈Anx1∈A1,x2∈A2,…xn∈An

**Example** –

 If we take two sets A={a,b}A={a,b} and B={1,2}B={1,2},

The Cartesian product of A and B is written as − A×B={(a,1),(a,2),(b,1),(b,2)}A×B={(a,1),(a,2),(b,1),(b,2)}

The Cartesian product of B and A is written as − B×A={(1,a),(1,b),(2,a),(2,b)}B×A={(1,a),(1,b),(2,a),(2,b)}

**::::::::Power Set::::::**

Power set of a set S is the set of all subsets of S including the empty set. The cardinality of a power set of a set S of cardinality n is 2n2n. Power set is denoted as P(S)P(S).

**Example −**

For a set S={a,b,c,d}S={a,b,c,d} let us calculate the subsets −

* Subsets with 0 elements − {∅}{∅} (the empty set)
* Subsets with 1 element − {a},{b},{c},{d}{a},{b},{c},{d}
* Subsets with 2 elements − {a,b},{a,c},{a,d},{b,c},{b,d},{c,d}{a,b},{a,c},{a,d},{b,c},{b,d},{c,d}
* Subsets with 3 elements − {a,b,c},{a,b,d},{a,c,d},{b,c,d}{a,b,c},{a,b,d},{a,c,d},{b,c,d}
* Subsets with 4 elements − {a,b,c,d}{a,b,c,d}

 Hence, P(S)=P(S)=

{{∅},{a},{b},{c},{d},{a,b},{a,c},{a,d},{b,c},{b,d},{c,d},{a,b,c},{a,b,d},{a,c,d},{b,c,d},{a,b,c,d}}{{∅},{a},{b},{c},{d},{a,b},{a,c},{a,d},{b,c},{b,d},{c,d},{a,b,c},{a,b,d},{a,c,d},{b,c,d},{a,b,c,d}}

|P(S)|=24=16|P(S)|=24=16

**Note** − The power set of an empty set is also an empty set.

|P({∅})|=20=1|P({∅})|=20=1

**:::::::::::Partitioning of a Set:::::::::**

Partition of a set, say *S*, is a collection of *n* disjoint subsets, say P1,P2,…PnP1,P2,…Pn that satisfies the following three conditions −

* PiPi does not contain the empty set.

[Pi≠{∅} for all 0<i≤n][Pi≠{∅} for all 0<i≤n]

* The union of the subsets must equal the entire original set.

[P1∪P2∪⋯∪Pn=S][P1∪P2∪⋯∪Pn=S]

* The intersection of any two distinct sets is empty.

[Pa∩Pb={∅}, for a≠b where n≥a,b≥0][Pa∩Pb={∅}, for a≠b where n≥a,b≥0]

**Example**

Let S={a,b,c,d,e,f,g,h}S={a,b,c,d,e,f,g,h}

One probable partitioning is {a},{b,c,d},{e,f,g,h}{a},{b,c,d},{e,f,g,h}

Another probable partitioning is {a,b},{c,d},{e,f,g,h}{a,b},{c,d},{e,f,g,h}

Bell Numbers

Bell numbers give the count of the number of ways to partition a set. They are denoted by BnBn where n is the cardinality of the set.

**Example** −

Let S={1,2,3}S={1,2,3}, n=|S|=3n=|S|=3

The alternate partitions are −

1. ∅,{1,2,3}∅,{1,2,3}

2. {1},{2,3}{1},{2,3}

3. {1,2},{3}{1,2},{3}

4. {1,3},{2}{1,3},{2}

5. {1},{2},{3}{1},{2},{3}

Hence B3=5