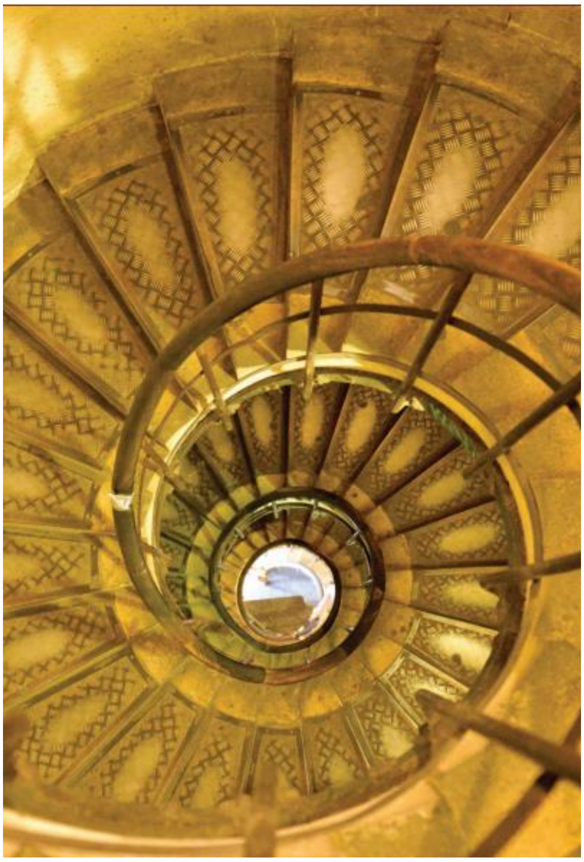
DISCRETE MATHEMATICS COURSE OF

GO CLASSES

Lectures by Deepak Poonia

References :

- Discreate Mathematics and its application by Kenneth H. Rosen 8th edition goclasses.in



Discrete Mathematics course of

Lectures by Deepak Poonia

INDEX

By Quantum City

Made by **Quantum City**

1. Mathematical Logic

1.1) Introduction to Mathematical Logic :

Language for mathematics : Mathematical logic gives precise, Unambiguous meaning to mathematical statements/theorems etc.

But why do we need a new language/logic for mathematics/computers?? Why can't we use a natural language, like English?? – Answer is Because all the natural languages are ambiguous. *Example* : "The sun is shining and I feel happy." What does this statement mean?

1. Does it mean that your friend is happy because the sun is shining?
2. Sun shining and his happiness are completely independent things?

Now, consider another statement "Cats are furry and elephants are heavy." Both these statements have exactly the same structure, but nobody would assume that elephants are heavy because of the furriness of cats.

Mathematical logic resolves this ambiguity.

1.2) Propositional Logic :

Simply a world of true, false. The variable used is known as *propositional variable* or *Boolean variable*.

A proposition is a declarative sentence (that is, a sentence that declares a fact) that can be either true or false; it must be one or the other, and it cannot be both. A proposition/statement is a declarative sentence to which It is possible to assign a value of either true or false.

Example of propositions :

1. "Jaipur is the capital of the India."
2. "Some cows are brown."
3. "2 x 2 = 5."

But What is NOT a proposition then? Answer is commands are not proposition, questions cannot be a proposition and "x+2 = 2x" is not a proposition. Now consider following sentence : S : "S is false". This is declarative sentence but still if you assign S to true then it says S is false. Same sentence has two values i.e. true and false. This type of sentence is called **paradox**.

Each proposition will be represented by a propositional variable. (generally, by any Upper alphabet letter). Example : S = false. Then false is the truth value of proposition S.

1.2.1) Atomic and compound proposition :

An Atomic proposition is one whose truth or falsity does not depend on the truth or falsity of any other proposition.

For example : "4 is a prime number." This is propositioning whose value is false but it cannot be derived from any other proposition. "4 is prime number, and New Delhi is the capital of India." This is also proposition but it is not atomic, is truth value depend on the truth values of two proposition. Atomic proposition is, by itself true or false.

A single Boolean(propositional) variable p is referred to as an atomic proposition, since it does not reduce further to other more basic propositions.

Note : New propositions (Compound) can be created from Atomic propositions with the help of Logical connectives.

Associative (Anderbahar) : (a + b) + c = a + (b + c) and communicative : a + b = b + a Commutative

Examples of logical connectives :

Let P : 4 is prime number, Q : New Delhi is the capital of India.

1. 4 is prime number, **AND** New Delhi is the capital of India.
2. 4 is **NOT** a prime number.
3. New Delhi is the capital of India **OR** 4 is not prime number.

A Compound propositions are the propositions which are constructed by combining one or more atomic propositions.

Some standard logical connectives :

1. NOT
2. AND
3. OR
4. Exclusive OR
5. Implication
6. Bi-implication or Double implication
7. NAND
8. NOR

1.2.2) Logical Connective :

Conditional Statements : The statement p → q is called a conditional statement because p → q asserts that q is true on the condition that p holds. A conditional statement is also called an **implication**.

antecedence of S $S : P \to Q$ Consequent of S

Question :

1) Filling the GATE exam application form is __ for cracking the GATE exam.

Answer : Always remember that sufficient means enough. So, filling the game exam application form is enough for cracking the GATE exam. False because in addition to this you should have marks more than cutoff then you are said to crake GATE exam. So, filling the GATE exam application form is necessary for cracking the GATE exam. Necessary means without gate exam application form you cannot crack GATE but it is one of the required conditions not sufficient condition. Necessary is subset of sufficient. Answer is necessary but hold on. If you not fill the GATE exam application then you cannot crack the GATE exam. This is true. If you write it in symbolic form then ¬p → ¬q. This is same as q → p. so, **P is necessary condition for q is same as q → p**.

2) "Being Natural number" is __ for "Being integer number"

Answer : If I want integer and you give me some natural number then it is not necessary but it is sufficient information that it is going to be integer only because every natural number is subset of integer. So, answer is sufficient. This is same as If one number is natural number then it is also integer number. But put that into reverse if one number is integer number then it is natural number. This is false because integer number can be negative which is not natural. So, first statement is correct. i.e. If one number is natural number then it is also integer number. This is nothing but p → q. So, **P is sufficient condition for q is same as p → q**.

3) Studying chemistry is __ for cracking GATE CSE exam.

Answer : Sometimes question itself is wrong in this case it is neither necessary nor sufficient.

4) N being even is __ for N+2 being even.

Answer : if N is even then N+2 is even so it is sufficient. But if N is not even then N+2 is not even. So, it is necessary also.

Different meanings of p → q :

- if p, then q"
- "p implies q"
- "if p, q"
- "p only if q"
- "p is sufficient for q"
- "a sufficient condition for q is p"
- "q if p"
- "q whenever p"
- "q when p"
- "q is necessary for p"
- "a necessary condition for p is q"
- "q follows from p"
- "q unless ¬p"
- "q provided that p"
- "p only when q"

Converse, Contrapositive, And Inverse :
The proposition q → p is called the converse of p → q.
The contrapositive of p → q is the proposition ¬q → ¬p.
The proposition ¬p → ¬q is called the inverse of p → q.

Biconditional Statement : The biconditional statement p ↔ q is the proposition "p if and only if q." The biconditional statement p ↔ q is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called bi-implications**.** This means p ↔ q is equal to p → q and q → p. Now p → q means q **if** p **and** q → p means q **only if** p so, if you combine them you get q if and only if p or p if and only if q.
Different meaning ***:***
"p is necessary and sufficient for q" "p iff q."
"if p then q, and conversely" "p exactly when q.

TABLE 8
Precedence of Logical Operators.

Operator	*Precedence*
¬	1
∧	2
∨	3
→	4
↔	5

1.2.3) Propositional formula/expression :
Every propositional variables or combination of variables are propositional formula or expression.
Example : (q → ¬p) → (¬t → r)
Examples which are not propositional formula : p q ¬, p → q r ¬

Propositional logic is collection/set of all propositional formulas.

Truth table : Truth table tell us about the truth values of a compound proposition for each combination of truth values of atomic propositions.
Tautology (**valid**) : A Tautology is a wff (well-formed formula) for which all truth table values are T.
Contradiction(**fallacy OR unsatisfiable**): A contradiction is a wff for which all truth tables values are F.
Contingency : A contingency is a wff that is neither a tautology nor a contradiction.
Example : P ∨ ¬P is a tautology. P ∧ ¬P is a contradiction. P → Q is a contingency.

NOTE :

1) **P is necessary condition for q only implies that without P, Q cannot happen (¬p → ¬q) which in turn implies q → p.**
2) **Implication tells us property and Bi-implication tells us definition. p → q This also means q is property of p. p ↔ q this means p is definition of q or q is definition of p.**
3) **Precedence of EXOR, NOR, NAND operator is not universally defined, it will be given in question. Or you can solve those by expanding them to simpler logical connectives.**
4) **Propositional formula and compound proposition are same.**

always true	sometimes true/sometimes false	always false
tautology	contingency	contradiction
logically true	contingency	logically false
valid	invalid	
unfalsifiable	falsifiable	
	satisfiable	unsatisfiable

1.3) Case method :

We assign one truth value to one variable then check for truth value of expression. We do not have to draw truth table.

Example : Consider p ∧ ¬(q ∨ p), Now consider only two case i.e. when p is true and when p is false. So, when p is true. T ∧ ¬(q ∨ T). which is false and when p is false expression is false so expression is contradiction.

1.4) Logical Equivalence :

The propositions are equal or logically equivalent if they always have the same truth value. If P and Q are logically equivalent, we write P ≡ Q. Some author uses ⇔ this symbol.

TABLE 6 Logical Equivalences.

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

If first proposition is same then sign is same
If Different then sign diff.

TABLE 7 Logical Equivalences Involving Conditional Statements.

$p \to q \equiv \neg p \vee q$
$p \to q \equiv \neg q \to \neg p$
$p \vee q \equiv \neg p \to q$
$p \wedge q \equiv \neg(p \to \neg q)$
$\neg(p \to q) \equiv p \wedge \neg q$
$(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$
$(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$
$(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$
$(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$

TABLE 8 Logical Equivalences Involving Biconditional Statements.

$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

1.5) English statements to propositional expression :

In logic, we have, AND = But = Although = Though = Even Though = However = Yet = Still = Moreover = Nevertheless = Nonetheless = Comma.

1.5.1) Logic translation of different English word :

Unless : here is an example, (imagine a stubborn child "jen")

"Jen won't go to the party UNLESS Mary goes to the party."; This sentence means if Mary doesn't go to the party then jen won't go to the party.

Let, p = Jen won't go to the party and q = Mary goes to the party. Above sentence can be written as **¬q → p**

Another example, "I won't study UNLESS you complete my demand";

Let, p = I won't study and q = you complete my demand. If you don't complete my demand then I won't study (¬q → p).

Unless P, Q = ¬P → Q (just replace **UNLESS** with **IF NOT/OR**)

P unless Q = P if not Q = ¬Q → P = Q ∨ P. Which means unless is same as OR operation.

Another example : If you study well, you will crack GATE exam unless you make silly mistakes. Now the question is which sentence to solve first. So, the answer is in English do not use precedence table. Just go with the feel. In above example, we take "if you study well, you will crack GATE exam" as first sentence so, let p = you study well, q = you will crack GATE exam and r = you make silly mistakes. So, (p → q) ∨ r. because **OR = UNLESS**.

1.6) Logical arguments :

Argument in propositional logic is a sequence of propositions which includes premises/knowledge-base/hypothesis/antecedents and conclusion.

TABLE 1 Rules of Inference.

Rule of Inference	*Tautology*	*Name*
p $p \to q$ $\therefore q$	$(p \land (p \to q)) \to q$	Modus ponens
$\neg q$ $p \to q$ $\therefore \neg p$	$(\neg q \land (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ $\therefore p \to r$	$((p \to q) \land (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \lor q$ $\neg p$ $\therefore q$	$((p \lor q) \land \neg p) \to q$	Disjunctive syllogism
p $\therefore p \lor q$	$p \to (p \lor q)$	Addition
$p \land q$ $\therefore p$	$(p \land q) \to p$	Simplification
p q $\therefore p \land q$	$((p) \land (q)) \to (p \land q)$	Conjunction
$p \lor q$ $\neg p \lor r$ $\therefore q \lor r$	$((p \lor q) \land (\neg p \lor r)) \to (q \lor r)$	Resolution

Using only these standard rules of inference we can check validity of any arguments.

Example : "If it is raining, He'll take umbrella.", "It is not raining." (This is called **premises**)

Can we infer "He'll not take umbrella" ? (This is called **conclusion**)

Answer is yes, modes ponens.

Argument = Premises + Conclusion

We say that above argument (example) is valid. Argument is valid if, whenever the premises are true, then the conclusion is true.

1.6.1) The inference symbol :

P1, P2, P3 hence C. This statement is valid argument iff $P1 \land P2 \land P3 \rightarrow C$ is a Tautology. We can also write this statement in short mathematical form. $P1, P2, P3 \models C$ or $P1, P2, P3 \vdash C$.

KB(knowledge base) $\models y$ is equivalent to KB infers y; KB entails y; KB implies y; y is a consequence of KB.

Fallacy of affirming the conclusion : $((p \rightarrow q) \land q) \rightarrow p$ (For p false and q true)

Fallacy of denying the hypothesis : $((p \rightarrow q) \land \neg p) \rightarrow \neg q$ (For q true and p false)

2. FIRST ORDER LOGIC

First order logic also known as Predicate calculus/ Predicate logic/ Quantificational logic.

WHY we need another type of logic when we already have propositional logic ? – Consider one argument. “All men are mortal. Socrates is a man.”, “Hence, Socrates is mortal.” Unfortunately, there is really no way to express this in propositional logic. Propositional logic has very limited expressive power. But we know that above argument is valid. But we can’t prove so we extend our idea to first order logic and second order logic.

So, what is the problem with propositional logic ? – It treats everything as True or false. We cannot assign number to any variable. In real world we should talk about more than true or false. So, we follow first order logic and higher order logic.

Definition : A world of objects, their properties, their relationships, their transformation(function)

So, what does FOL have, that propositional logic doesn’t ??

- First-order logic speaks about objects, which are the domain of discourse or the universe.
- First-order logic is also concerned about properties of these objects.
- Also, we have relations over/between/among objects (called predicates)
- In FOL, we also have functions of objects
- Another significant new concept in first-order logic is quantification: the ability to assert that a certain property holds all elements or that it holds for some element.

First order logic = Proposition logic + Objects (*Domain*) + Their properties (*Predicate*) + Multiple objects (*Quantifier*)

We are done with proposition logic. So, let’s start with remaining one.

Objects (Domain) : Set of value a variable can take. It is also called *Universe*. For exmple, the universe of propositional logic is true or false. The universe of age is number. So, *domain* cannot be empty unless it is explicitly given in question. And values in domain are called *constant*.

Predicate (Their properties and relationship) : Consider Domain be set of people : {ram, john, sita, gita}. Let say x is variable from this domain. This domain has different properties like ram is male. Sita is female. And They can also have relationship for example, ram is husband of sita. John is brother of gita, etc. We represent these predicates by alphabet for example, C(x) = “x is clever”. And some x of domain satisfies this property some don’t. This is called unary predicate. F(x, y) = “x is father of y”. where x and y have domain of set of people. This is called binary predicate. And we also have ternary predicate. So, <u>predicate is a sentence containing variables (every variable refers to the domain) such that it becomes a proposition once we replace each variable with specific value from its domain.</u>

This is same predicate that we had seen in last chapter. But here we learnt it more rigorously.

Quantifiers : Quantifiers tell us about quantity of objects. For example, some, many, all, none, etc. In FOL we have two quantifiers namely, for all and there exists or Universal quantifier and existential quantifiers respectively. For example, consider of FOL where domain : set of all people, Predicate : M(x) = “x is male”. And we say there exists x in the domain, M(x) is true. Which simply means every person is male which is false. But it is predicate.

Universal Quantification : The universal quantification of P(x) is the proposition "P(x) is true for all values x in the universe of discourse". "For all x P(x)" or "For every x P(x)" is written ∀xP(x). Here ∀ is called the *universal quantifier*. "An element for which P(x) is false is called a counterexample of ∀xP(x)"

Let's consider domain made up of {a,b,c,d} a finite set and a predicate P(x). Now, ∀xP(x) means P(a) ∧ P(b) ∧ P(c) ∧ P(d). ∀ is conjunction over finite domain.

Existential Quantification : The Existential Quantification of P(x) is the proposition "There exists an element x in the universe of discourse such that P(x) is true". "There exists x such that P(x)" or "There is at least one x such that P(x)" is written ∃xP(x). ∃xP(x) is true iff there is at least one **witness** (an element for which P is true). Here ∃ is called existential quantifier. "All element for which P(x) is false is called a counterexample of ∃xP(x).

Let's consider domain made up of {a,b,c,d} a finite set and a predicate P(x). Now, ∃xP(x) means P(a) V P(b) V P(c) V P(d). ∃ is disjunction over finite domain.

Question :

1) If the domain is explicitly given as Empty, then How do Quantifiers behave ?

Answer : ∀xP(x) this will be false if there is counterexample. So, if we have empty domain, we cannot have any counterexample. So, Universal quantifier is true. Talking about existential quantifier ∃xP(x), ∃ is true if we have one witness but if the domain is empty then we cannot have any witness. So, if domain is empty then existential quantifier is always false.

Domain is always non-empty unless it is explicitly given

2) If there is no free variable then how does quantifier behave ?

Answer : Consider one proposition, P : "2+2 = 4". We know that this statement is true. So, for every element of domain this statement is true. That is why Universal quantifier is true. And if proposition would be "2+2=9" Then for every element this sentence is incorrect. That is why universal quantifier is false. Similar logic you can apply for Existential quantifier. Predicate will become proposition when there is no variable or no free variable (which we shall see later in this chapter). That means if we put value of x in predicate it will become proposition.

2.1) English-FOL translation :

Let, Domain : Set of all animals. And Predicate, cute(x) : x is cute.

1. Every animal is cute. = ∀x cute(x)
2. Some animal is cute. = ∃x cute(x)
3. Every rabbit is cute. = Now, ∀x(rabbit(x) ∧ cute(x)) This representation seems correct. But it is not. It says "Every animal is rabbit *and* cute". Correct alternative should be ∀x(rabbit(x) → cute(x)). If x is rabbit then it should be cute.
4. Some rabbit is cute. = ∃x(rabbit(x)→cute(x)). This is false. Consider domain consists of {lion, rabbit(which is not cute), tiger(which is cute)}. So now we know that there is no rabbit which is cute. But ∃x(rabbit(x)→cute(x)) this says it is true rabbit(x) is false for lion so the whole expression is true. Because false implies anything is true. ∃x(rabbit(x) ∧ cute(x)) this is true. Because it says There should be at least one x which is rabbit and cute. Now, consider previous domain (lion domain). This expression is true.

In general, we can say that

All A's are B's. = ∀x(A(x)→B(x)); *Some A's are B's* = ∃x(A(x) ∧ B(x)). That mean for Universal quantifier we use implication and for existential quantifier we use conjunction.

5. Consider R(x) is "x is a rabbit" and H(x) is "x hops" and the domain consists of all animals. Then translate ∃x(R(x)→H(x)). It does not say "Some rabbit hops". In this type of question where existential quantifier is given with implication, we write ∃x(¬R(x) V H(x)). Answer is "There are animals which are not rabbit or hops".
6. Consider domain : Set of all people and predicate, Male(x) : x is a male; Army(x) : x is in army.

Only males are in army. - ∀x(Army(x)→Male(x)) it also means ∀x(-Male(x)→-Army(x)).

All and only males are in army – This is combination of all males are in army and only males are in army. First means ∀x(Male(x)→Army(x)). And second means ∀x(Army(x)→Male(x)). If we combine these two, we get double implication.

2.2) Bounded variable and Free variable :

2.2.1) Bounded variable :

Consider : set of all-natural number. E(x) = x is even.

∀x E(x) – This is false we know but it is proposition. It says "All-natural number are even". But proposition should not contain any variable. And we know that each sentence means the same. So, does it really contain variable ?

Again, let's take another example,

Consider, Domain : {a,b,c} predicate, N(x) : x is nice.

∀x N(x) – N(a) ∧ N(b) ∧ N(c). Both means same. But one contains variable and one doesn't. This variable is called **dummy variable** or **bounded variable** or **quantified variable**. Why bounded because it is bounded by quantifier.

So, does it really contain variable ? – Answer is yes. (dummy variable)

2.2.2) Free variable or Real variable :

Now, consider, Domain : Set of all-natural numbers.

E(x) : x is even. Clearly this is not proposition because it contains variable. But here one thing to note that x can take any value. It can take one or more than one value. Don't think about it is true or false. But it is not bounded by any quantifiers so it is called *free variable*. In dummy variable, we cannot assign value to variable. For example, let's say similar domain but ∀x E(x) and take x = 3. ∀3 E(3) this makes no sense. But if you consider previous sentence E(x) only, it makes sense. That's the difference between these two variables.

Question :

1) Consider Domain : Set of all integers. Convert the following into a proposition. S(y) : ∃x(x>y)

Answer : ∃x(x>y) here x is not main variable it is bounded variable. Free variable is always main variable. So, y is main variable that is why S(y) is given. Y is free to take any value under given domain so this is not proposition. So, first we convert this sentence into proposition by putting some value into y. S(4) = ∃x(x>4) which says there must be one integer which is greater than 4 and yes, it is true. You can put any integer into y. Let's look at another approach.

Second approach, to convert any predicate into proposition is to remove free variable by converting it to quantified variable. As we know that y can take any integer value, we use universal quantifier and make y a quantified variable. ∀y∃x(x>y) This says for all integer there exists integer which is greater. And yes, it is also true. Now we use existential quantifier instead of universal and check for the truth value of proposition. ∃y∃x(x>y) which says there exists integer for which there exists integer greater than it. which is true. It is not necessary to use both universal and existential, you can use only one.

A proposition can only contain bound variables, no free variables.

NOTE :

1) **If "A" doesn't have any free variable x then ∃xA = A, ∀xA = A.**
2) **While solving English to quantifier always go with meaning of statement. What you have done is you have selected one correct option which matches with meaning of statement then you have tried to select those option which is derived from selected option. Method is correct like those options are also correct but look for meaning also.**

2.3) Scope of quantifier :

A part of logical expression to which a quantifier is applied is called the scope of this quantifier.

$\exists y[P_y \vee Hx]$ Free

scope

$\exists y[F_y \wedge \forall x[Fx]] \wedge L(y)$ Free

$\exists x(\underbrace{\,- - - x - - x\,}_{\text{Scope of } \exists x}) \wedge P(x)$

free

What is the importance of scope of quantifier ?

Consider, ∀x(smile(x) → wearhat(x)) and **∀x**(smile(x)) → ∀x(wearhat(x)). Both have different meaning. First say that "Every smiling person wear a hat" and second says "if all people are smiling then all people wears hat". So, in second one we can change dummy variable in second scope. i.e. ∀x(smile(x))→ ∀y(wearhat(y)).

Consider domain consists of integer and you have predicate I(x) : x is integer, R(x) : x is real. If I(x) appears in any quantified predicate then we know that I(x) is always true. And it is correct representation (translation) if something about integer is asked and I(x) is used in it.

2.4) Nested Quantifier :

∀x∀y (xy = yx) is the example of nested quantifier. But why do need nested quantifiers ? – Many interesting statements in first-order logic require a combination of quantifiers. Eg. For every natural number, there exists a greater natural number, some natural number is less than or equal to every natural number. <u>Note that the quantifier must be read from left to right</u>. Eg.

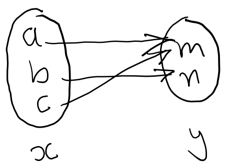
$\forall x \ \exists y \ P(x,y)$

For all x, there is a y, P(x,y) is true. We will study four combination of nested quantifier namely,

∀x∀y P(x,y) ∃x∀y P(x,y)

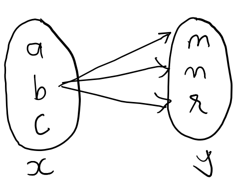
∀x∃y P(x,y) ∃x∃y P(x,y)

- ∀x∀y P(x,y) : This means for every value of x and for every value of y has property P(x,y).
- ∀x∃y P(x,y) : This means for all value of x there exists y such that P(x,y) is true. Remember here for all value x there may be different value. Eg. Let x be student and y be course then all students have taken some course.



P(a, m) = T
P(b, n) = T
P(c, m) = T

- ∃x∀y P(x,y) : There exists one value x for all y such that P(x,y) is true. There is a student x has taken all y course.



P(b, m) = T
P(b, n) = T
P(b, r) = T

- ∃x∃y P(x,y) : There is one value for one value such that P(x,y) is true. Some student x has taken some y course.

NOTE :

1) **If quantifiers used in sentence is same then you can change the order of quantifier. But if the quantifiers used is different for example, ∃x∀y to ∀y∃x or vice versa then it is not valid.**

2.4.1) Numerical Quantification :

Now we will learn how to use FOL to express numerical quantifiers as the following : at least two, at most one, exactly one, at least three , at most three, at most two, exactly two, etc.

Let's take simple example and then we will eventually take hard as it goes…

Question :

1) *There is at least one cube*. - ∃x Cube(x)
2) *There are at least two cubes*. - **∃x∃y** (Cube(x) **∧** Cube(y) **∧** (x ≠ y)) why we have not used implication because we have existential quantifier. You can extend this idea to at least three also. Just add another variable z and make three condition which includes ≠. But here is another question how do you express sentence when we say "There are at least 2 elements in the domain" – Here we know that values are from domain directly so we do not have to create any Cube(x) or something like that we can directly write ∃x∃y (x ≠ y). Similarly, for "There are at least 3 elements in the domain" - ∃x∃y∃z (x ≠ y ∧ y ≠ z ∧ x ≠ z).
3) *There is at most one cube*. – Meaning in a domain if we have many cubes those cubes must be same. Here you noticed we have used "if" and "many". ∀x∀y ((Cube(x) ∧ Cube(y))→(x=y)). Similar to question 2 we ask for "There is at most one element in the domain" - ∀x∀y (x=y).

4) *There is exactly one cube* – There are two methods let's see first. Exactly means at least one and at most one. We have seen these cases previously. First is at least one means $\exists x$ Cube(x) and at most means $\forall x \forall y$ ((Cube(x) $\wedge$ Cube(y))$\rightarrow$(x=y)). We combine this expression with and operator. So, final answer will be $\exists x$ Cube(x) $\wedge$ $\forall x \forall y$ ((Cube(x) $\wedge$ Cube(y))$\rightarrow$(x=y)). Second method is to observe that exactly means at least one and if that at least one cube matches with all cube then both must be same. i.e. $\exists x$ (Cube(x) $\wedge$ $\forall y$ (Cube(y)$\rightarrow$x=y)). At least one cube will make surety that there must be cube and comparing this cube with other make sure that selected cube is same as the cube in domain. Here x implies every shape in box or domain. But if we assign x to be cube only then we can have another representation as follows, $\exists x \forall y$ (Cube(y) $\leftrightarrow$ (x=y)). Why double implies because if we know x is cube(let's say C1) and if all shape is equal to x(C1) then they are also cube and C1.
5) *There is at most 2 cubes* – We follow the similar pattern as with at most one cube. We take three cubes from all element in the domain and if they are cube then two of them must be same. $\forall x \forall y \forall z$ ((Cube(x) $\wedge$ Cube(y) $\wedge$ Cube(z))$\rightarrow$(x=y $\vee$ y=z $\vee$ x=z)).
6) *There are exactly two cubes*. – for exactly two cubes we should be at least two cubes then and both should be different. Then we take all element and if that element is cube then it must be equal to first or second cube. $\exists x \exists y$ ((x $\neq$ y) $\wedge$ Cube(x) $\wedge$ Cube(y) $\wedge$ $\forall z$(Cube(z) $\rightarrow$ (x=z) $\vee$ (y=z))).

Now we will see example of each 1 to 6 question.

E1) "For every number there is a larger number" Assume domain is set of numbers. – $\forall x \exists y (x<y)$

E2) "There is a number that is larger than every other number". Assume domain is set of numbers.

$\exists x \forall y (x>y)$ This looks correct but it says other number meaning x should not be equal to y. Correct answer is $\exists x \forall y [(x \neq y) \longrightarrow (x > y)]$

E3) "If one number is less than another, then there is a number properly between the two". Assume domain is set of numbers. - $\boxed{\exists x} \exists y (x \neq y \wedge x < y \boxed{\Longrightarrow} \exists z ((x<z) \wedge (z<y)))$

This also looks correct but here they are not talking about specific two numbers. They are saying if one number is less than another meaning for all numbers. They are not saying there is a number which is less than another number. And you do not have to include not equal to condition because x<y make sure that they are not equal. $\forall x \forall y [x<y \longrightarrow \exists z ((x<z) \wedge (z<y))]$

E4) "There are infinitely many numbers which are prime." Assume domain is set of numbers. $\exists x (prime(x))$

This is incorrect. It says at least. "Infinitely many numbers" is nothing but some number which is greater than all other number and it should be prime. $\forall x \exists y [(x<y) \wedge prime(y)]$

E5) X is even number. – This means x can be represented as 2*something. $\exists y [x = 2y]$

E6) P(X) : X is prime. – This means number should be greater than 1 and any number other than 1 or x should not divide x. $(x>1) \wedge (\forall y [y|x \longrightarrow (y=1) \vee (y=x)])$

2.4.2) Negation of quantifier :

We directly go to example,

E1) At least one person has a car. –$\neg \exists x Car(x) \equiv \forall x \neg (Car(x))$

In short, $\neg \forall x P(x) \equiv \exists x (\neg P(x))$
$\neg \exists x P(x) \equiv \forall x (\neg P(x))$

2.5) Validity, satisfiability in FOL :

Let M be a first order logic expression.

- M is valid iff M is always true (Whatever non-empty domain we take and whatever predicate we take)
- M is satisfiable iff M is possible to be true (for some domain and some predicate)

So, $S : \forall x\, P(x)$ is it valid ? Is it satisfiable ?? – yes If we have domain as natural number and predicate as x > -1 then we can make S satisfiable. But can we say it is valid or always true no we can have cases where it is not always true. See the definition it says you give me anything and it should be always true then it is valid. I can take domain as natural number and predicate as x is even. Which is false as natural numbers being not only even. Similarly, for existential quantifier.

Let's take another example, $(\exists x\, P(x)) \vee (\exists y \neg P(y))$ this is satisfiable as we can always create specific case where it is true. But is it valid ? – We know that x and y have common domain by default, this is false when P(x) is false always and other is false. Let's say P(x) is false means there does not exist a single element for which P(x) is true. Which means ¬P(y) is true because as it says It is true when there does not exist a single element. So, it is always true. And that is why it is valid.

Question is how to check if a FOL expression M is valid or not ?

Method 1 : Intuition and logical thinking (we saw this)

Method 2 : A systematic procedure

Procedure :

1. Take an abstract domain (abstract means don't take natural number or something like that take domain with no meaning or relationship like abcd), like {a,b,c,d,…}
2. Try to make M false somehow…
   - If you can, M is invalid
   - If you can never, M is valid.

Similarly, how to check if a FOL expression M is satisfiable or not??

Method 1 : Intuition and logical thinking (we saw this)

Method 2 : A systematic procedure

Procedure :

1. Take an abstract domain, like {a,b,c,d,…}
2. Try to make M True somehow…
   - If you can, M is satisfiable.
   - If you can never, M is not satisfiable.

Question :

1) $(\forall x\, P_x) \wedge (\exists y \neg P_y)$

Answer : First saying for all element, P is true. But second is saying for at least one value P is false. Which cannot happen simultaneously. This is contradiction meaning it is not valid and not satisfiable.

2) $(\forall x Px) \wedge (\forall y \neg Py)$

Answer : This is not satisfiable and not valid. Because they can't happen together. If OR connective is given then it is satisfiable and talking about validity so we have some property which is false for some element and true for some element in that each quantified predicate is false. So, not valid.

3) $(\exists x Px) \wedge (\exists y \sim P(y))$

Answer : It says P can be true for some element and false for another element. Yes, it is satisfiable. But Consider for all element P is true, then as cannot have element for which P is false which contradict second predicate. So, it is invalid.

4) $[\forall x P(x)] \longrightarrow [\exists x P(x)]$

Answer : It is always true because if every element satisfies P then some element also satisfies P. Which means it is valid and satisfiable. Now, here's a cache, assume domain to be empty. Now first predicate is true always and second predicate is false always so T→F is F. so; it is invalid after all? Answer is no. Both cases are correct. In second case we have assume that domain is empty. And that is why we do not take domain empty unless it is explicitly given.

Now, let's see more on implication…

5) $\forall x [Px \wedge Qx] \longrightarrow \forall x Px \wedge \forall x Qx$

Answer : We have to make first predicate true and second one false. This is always true because properties of universal quantifier. In fact, reverse is also true. In short, we can put double implication instead of implication.

6) $\forall x [Px \wedge Qx] \longrightarrow \forall y Py$

Answer : We know that for LHS to be true we have to make first and second predicate true. First says P is true for all element and Q is also true for all element which implies that P is true for all element. Which is always true. Now, take reverse i.e. $\forall y Py \longrightarrow \forall x [Px \wedge Qx]$

LHS says that P is true for all element and second predicate implies that P is true for element and Q is also true. Now, we know that first and second are equal but consider a case where Q is false for all or some element which means it is not always true. It is satisfiable but not valid.

7) $\forall x [Px \vee Qx] \longrightarrow \forall x Px \vee \forall x Qx$

Answer : This is satisfiable. If for all P and Q is true. It is invalid because if some half of the element of domain satisfies P and half of them satisfies Q then LHS is true but RHS is false. So, invalid. But take reverse direction. It says for all P element P is true or for all element Q is also true. Which means If one them is false then also RHS is true. So always true.

Distributive properties of quantifiers :

$$\forall x (P(x) \wedge Q(x)) \underset{\checkmark}{\overset{\checkmark}{\rightleftarrows}} \forall x P(x) \wedge \forall x Q(x)$$

$$\forall x (P(x) \vee Q(x)) \underset{\times}{\overset{\checkmark}{\leftrightarrows}} \forall x P(x) \vee \forall x Q(x)$$

This is distribution of quantifier over logical connectives : let's represent quantifier by some common notation (Π) and connectives by (#). We have to check,

$$\underbrace{\Pi x (P(x) \# Q(x))}_{\text{compact}} \underset{?}{\overset{?}{\leftrightarrows}} \underbrace{\Pi x P(x) \# \Pi x Q(x)}_{\text{Expanded}}$$

Let's take examples,

E1) $\exists x(P(x) \wedge Q(x)) \longrightarrow \exists x P(x) \wedge \exists x Q(x)$ - LHS says there is someone for which both P and Q is true simultaneously while RHL says there is x such that P and there is x (may or may not be same) such that Q. so it is true. Take reverse, it LHS says there is x for which P is true and there is x (may be different from P) such that Q is true. Now, both x can be different for which P and Q is true. So LHS is true but RHS is false. Meaning this is invalid.

E2) $\exists x(P(x) \oplus Q(x)) \longrightarrow \exists x P(x) \oplus \exists x Q(x)$ - It is invalid. As we can have some common x for which P is true but Q is false. By doing so we have LHS true. But we can have some x (not common) for which P is true and Q is also true. Then RHS is false. So invalid. But reverse is valid. Because ExOR is true for either one of them should be true. So, let $\exists x P(x)$ to be true and $\exists x Q(x)$ to be false. $\neg \exists x Q(x) \equiv \forall x \neg Q(x)$. LHS says there is x for which P is true and for all x Q is false. So RHS will automatically be true.

Similarly, we can prove many more and make table…

	$\forall$	$\exists$	$\exists!$
$\wedge$	$\leftrightarrow$	$\rightarrow$	
$\vee$	$\leftarrow$	$\leftrightarrow$	
$\rightarrow$	$\rightarrow$	$\leftarrow$	
$\leftrightarrow$	$\rightarrow$	×	
$\oplus$	×	$\leftarrow$	
(NAND) $\uparrow$	$\rightarrow$	$\leftarrow$	
(NOR) $\downarrow$	$\rightarrow$	$\leftarrow$	

$\rightarrow$ ≡ compact to expand

$\leftarrow$ ≡ Expand to compact

$\forall$ have ($\rightarrow$) in all arrow connectives

after filling $\forall$, fill $\exists$ by symmetry

× ≡ Invalid both sides

$\longrightarrow$ simple trick (NOR ≡ $\downarrow$ & NAND ≡ $\uparrow$)

We define a new quantifier, **uniqueness quantifier**, the symbol of which is ∃!. For any predicate P and universe U, ∃!xP(x) means there is exactly one element in the universe for which P is true.

E3) $\exists!x(P(x) \wedge Q(x)) \Rightarrow \exists!xP(x) \wedge \exists!xQ(x)$ – First it seems like it is valid but look carefully, if we have exactly one x for which both P and Q are true then RHS says we have exactly one x for which P is true and exactly one for which Q is true. But in domain we can have some x which is not true for both but it is true for either P or Q. In that case this will be false. Note that uniqueness means exactly one case should exists (not at least or at most). P is prime minister and Q is women then there may be some women who are not prime minister. In this case it fails.

Negation of uniqueness quantifier : Negation of uniqueness quantifier is nothing but either all should not satisfy or at least two witness for which it satisfies. We can say that ¬∃!xP(x) is for every x in domain either is should false or if there exists some x such that first witness and second witness is different. i.e. $\forall x[\neg P(x) \vee \exists y((x \neq y) \wedge P(y))]$

NOTE :

1) **If you try to find truth value of ∀x[p(x)→¬r(x)] don't replace it with expanded form. You can replace one wff by other wff when it is double implication. Here only one direction implication is there. i.e. ∀x[p(x)→¬r(x)] → [∀xp(x)→ ∀x¬r(x)] so you can't replace first by second. Try to find meaning instead, it says p(x)→¬r(x) should be true for all x. so come up with counterexample. In short if anything is given like ∀x or ∃x then follow definition.**

NULL Quantification :

We saw distributive properties of quantifier, Now see this

If A : "2+2=10" and we know that $\forall x A \equiv A$ & $\exists x A \equiv A$

$\forall x\, [P(x) \vee A]$

not affected by Quantifier

We call this NULL quantification. Which means A has no free x. Which also mean that if some part of expression has no free variable then expression is called NULL quantification. So, what so special about it ? Let's see one example,

$\forall x\, [P(x) \vee A(x)] \longrightarrow \forall x P(x) \vee \forall x A(x)$ ✗

$\forall x\, [P(x) \vee A] \longrightarrow \forall x P(x) \vee A$ ✓

i.e. **If "A" does not have any free variable x then : create two cases : A = true ; A = false**.

Expression is valid iff valid in both cases. If in some case invalid, then expression invalid. Do not blindly see null quantification and select options, prove from above bold expression.

Sometimes we can use alternative form, example $A \rightarrow B \equiv \neg A \vee B$

$\forall x [P(x) \rightarrow W] \equiv \forall x P(x) \rightarrow W$? $\quad \forall x \neg P(x) \vee W \equiv \neg \exists x P(x) \vee W \equiv \exists x P(x) \rightarrow W$

2.6) Interpretation, Model in logic :

2.6.1) Interpretation, model in propositional logic :

If we have some propositional formula Let's say G : $a \vee (b \rightarrow c)$

Then Interpretation means some truth combination of G (it might be true or false)

Definition : An interpretation I assigns a truth value to each atom.

So, let's say $\boxed{I_1 : a = T, b = F, c = F}$ and we say $I_1(G) = T$

Interpretation of G

For 2 variables How many interpretations of G ? – 2^2 interpretation. For n variable – 2^n

We know that for some interpretation expression has true value and some have false values. In propositional logic, **Model** is an interpretation for which expression is true. So, if we have system consists of n variables and expression contains only m variable and P are models for expression then we have total of $2^{(n-m)}$ x P models. **Co-model** is an interpretation for which expression is false.

Interpretation = Model + Co-model

Let set of propositional logic expressions

KB = {E1, E2, E3,… En} then model is an interpretation for which E1 and E2 and … and En is true.

Example, KB = $\{P \rightarrow Q,\ Q,\ R\}$ # interpretation = 8, #model = 2, #co-model = 6.

So, in propositional logic in terms of interpretation and models we can say that, *Tautology* : Every interpretation is model, *Contradiction* : There is no model, *Satisfiable* : There exists a model, *Valid* = satisfiable (both have same meaning in propositional logic but not in FOL)

2.6.2) Interpretation and model in first order logic :

Consider G : $\forall x\, P(x)$

What do we need to "interpret" G ? – You want predicate P(x) and domain. Combination of these two things is called *interpretation in first order logic*.

Interpretation 1

$\left.\begin{array}{l}\text{Domain : } \mathbb{N} \\ P(x) : x \text{ is even}\end{array}\right\}$ This interpretation is co-model

Then G = false

Finite model : model with finite domain. Similarly, we can define finite co-model. Above example was finite co-model.

Consider, $\forall x\, P(x) \longrightarrow \exists x\, P(x)$ we know that it is true if domain is not empty. But what if domain is empty then is it true ? – Yes, it is always true (whatever interpretation we take and whatever predicate we take). Every interpretation is a model in this example. So, we call this expression valid.

So, in first order logic, *valid* : always true (no co-model), *satisfiable* : there exists a model, *unsatisfiable* : All interpretation is co-model.

Tautology in first order logic : A tautology in first-order logic is a sentence that can be obtained by taking a tautology of propositional logic and uniformly replacing each propositional variable by a first-order formula (one formula per propositional variable). For example, because $A \vee \neg A$ is a tautology of propositional logic, $(\forall x\,(x = x)) \vee (\neg \forall x\,(x = x))$ is a tautology in first order logic. Here expression only contains one variable i.e. A. But if we have more than one variable then we assign each variable (like A, B, C) a unary relation symbol ($\forall x R(x), \exists x P(x), \neg \forall x P(x), \ldots$).

In short, we can summarize the meaning of each terms

Terms	Propositional Logic	First order logic
Valid	There exists a model	There is no co-model
Satisfiable	There exists a model	There exists a model
Unsatisfiable	There is no model	There is no model
Tautology	There is no co-model	Discussed
Contradiction	There is no model	There is no model

3. Set theory

3.1) Introduction :

Definition : Set is a collection of objects.

Set notation : Curly braces with commas separating out the element.

For example, set of English vowels can be represented by {a, e, i, o, u}.

Two sets are *equal* when they have the same contents, ignoring order. {a, b, c} = {b, a, c}

Set cannot contain duplicate elements. Any repeated elements are ignored. {a, b, b, b, c, c} = {a, b, c}

Set may contain anything. {a, b, c, 1, 2, {Gujarat, Bengal, Maharashtra}, hoes, b*tch}

So more formally we can say that **A set is an unordered collection of distinct objects, which may be anything (including other set).**

How many elements does the following sets have ? – {a, b} = 2 elements; {a, b, a, a} = 2 elements; {b, a} = 2 elements; {a, b, a, a, 1, 1, 2, 3, 2, 2} = 5 elements.

Membership : Let S = {a, b, c}, we say element "a" belongs to set S. We represent this using a ∈ S, element "d" is not belonging to set S can be represented by d ∉ S. ∈ is called *set membership symbol*.

Question :

1) 1 = {1} Are these objects equal ? – first 1 is number and {1} is set containing 1 number. So, both are different things.
2) Let S = {a, b, c}.
   (i) a ∈ S – True. Because a is element of S.
   (ii) {a} ∈ S – false. Because {a} is not element of S.
   (iii) {a, b, c} ∈ S – False. Because {a, b, c} is not an element of S.
   (iv) {a, b} ∈ S – False.
3) What is the set of all prime numbers between 14 to 16 (inclusive) – We know that there are no prime number between 14 to 16. Therefore, S = { } = ∅. This is called **NULL set** or **Empty set** where number of elements in set is zero. We represent this empty set by ∅ symbol.
4) S = {∅} = ∅ are these objects equal ? – first is set containing empty set and second is empty set. So, cardinality of first set is 1 and that of second is 0.

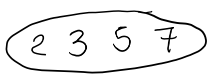
Terminology related to sets :

- *Finite set* : A set with a limited number of elements **or** A set in which number of elements is equal to some whole number (0, 1, 2, 3,..). Example : D = {dob, cat, fish, frog}
- *Infinite set* : A set with an unlimited number of elements. Example : N = {1, 2, 3, 4, …}
- *Cardinality of Set* : Number of elements it contains. (already discussed)

3.1.1) Set representation :

Consider the set of all prime numbers less than 10.

Above question is also a representation of some set it is called *verbal representation*. {2, 3, 5, 7} this is called *list representation or roster representation*. Below is called *Venn diagram*.



Another representation is famous known as *Set-builder representation*.

S = {x|some property that x must satisfy} = { x | p(x)}

element which satisfies this property.

In our example, we write S = {2, 3, 5, 7} = { x | x is prime and x < 10}

Let's take another example, Even natural numbers can be represented as

{ n | n ∈ N and n is even }

The set of all n where n is a natural number and n is even

3.1.2) More about sets :

1) **Subset** : It is sub collection of set. If set A is subset of set S then we represent it as $A \subseteq S$. We can represent $A \subseteq S$ in first-order logic as $\forall x(x \in A \rightarrow x \in S)$. $\subseteq$ this symbol is called "subset of".

Let S = {a, b, c}, subsets of S : $\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}$

From above subset of S we can conclude that every set is subset of itself. Which means $S \subseteq S, \emptyset \subseteq \emptyset$

Note that if we have S = {a, b}, $\emptyset \subseteq S$. This is true because empty set is *subset of* every set. $\emptyset \in S$ This is false because empty set is not *an element* of S.

Let's take one beautiful example,

S = {1, {2, 3}, 4}. We know that it has three elements. Now answer following question.

1) $\{1\} \subseteq S$ True. Because {1} is subset of S. but $1 \subseteq S$ this is false. As here 1 does not represents set it represents number 1 not set.
2) $\{1\} \in S$ False. Because {1} is not member of S. But $1 \in S$ this is true. Because 1 is member of S it is not subset of S.
3) $\{2\} \subseteq S$ false. Because in {2} we have element 2 but in set S we don't have element 2 we have {2,3} as an element.
4) $\{2, 3\} \subseteq S$ false. Because {2, 3} is not set it is element but $\{\{2,3\}\} \subseteq S$ is true and $\{2, 3\} \in S$ is true.

2) **Proper subsets** : A proper subset of a set S is a set T such that
- $T \subseteq S$
- $T \neq S$

There are multiple notations for this; they all mean the same thing: $T \subset S, T \subsetneq S$.

Let S = {1, 2} proper subsets of S are $\emptyset, \{1\}, \{2\}$

3) **Powerset of a set** : The set of all subsets of a set S is called powerset of S. The notation for the powerset of S is P(S). $S \in P(S)$ and $S \nsubseteq P(S)$

Let S = {a, b, c}, subsets of S : $\emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{b,c\}, \{a,c\}, \{a,b,c\}$ and powerset of S (set of all subsets of S) : $\{\emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{b,c\}, \{a,c\}, \{a,b,c\}\}$. Cardinality of powerset is 2^n. where n is number of elements in set S.

Now, what is P({}) or $P(\emptyset)$? – Powerset of $\emptyset$: $\{\emptyset\}$. We know that $|\emptyset| = 0$ but $|P(\emptyset)| = |\{\emptyset\}| = 1$. If you have doubt then in powerset or while taking subsets we select element from { } we do not consider whole { }.

Again, what is P({∅}) ? – powerset of {∅} : {∅, {∅}}. We know that $|P(\{\emptyset\})| = |\{\emptyset, \{\emptyset\}\}| = 2$

What is |P(P(S))| ? – It is talking about cardinality of powerset of powerset. We know that if set contains n element then powerset contains 2^n element. Now, cardinality of powerset of those 2^n element will be $2^{(2^n)}$.

3.1.3) Set operations :

Universal set : The Universal set U is the set containing everything currently under consideration. Content depends on the context. Sometimes explicitly stated, sometimes implicit.

Let's take an example of problem which includes vowels. Then possible universal sets are :

1) {a, e, i, o, u}
2) English alphabet
3) All alphabets of all languages
4) {a, e, i, o} – This cannot be universal set as it does not contain element u which completes set asked in problem.

Recall that we have +, -, x,… operations for numbers and we have ∧, ∨, →,… for propositions. So, question is what kind of operations do we have for sets ? – *Union, intersection, difference, complement,* … and not only that but we apply operations on number we get number as output, we apply operations on propositions we get proposition as output. Similarly, we do operations on set and we get output as set only.

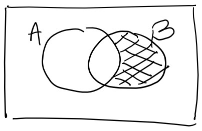
1) **Intersection** : Suppose M is the set of students who love mangoes, and N is the set of students who love kiwis. Then, $M \cap N$: the set of students who love mangoes and kiwis. We represent it in set builder form as $M \cap N = \{x | (x \in M) \wedge (x \in N)\}$.
   If A and B are sets and $A \cap B = \emptyset$ then we say that A and B are *disjoint*, or **disjoint sets**.
2) **Union :** Suppose we take same set as in intersection. $M \cup N = \{x | (x \in M) \vee (x \in N)\}$ = The set of students who love mangoes or kiwis (or both).
3) **Set difference :** The set of students who love mangoes but not kiwis = $M - N = \{x | (x \in M) \wedge (x \notin N)\} = M \cap \overline{N}$
4) **Symmetric difference (exclusive or) :** The set of students who love or mangoes but not both. $M \triangle N = M \oplus N = \{x | (x \in M) \oplus (x \in N)\} = (M \cup N) - (M \cap N) = (M - N) \cup (N - M)$
5) **Complement :** The set of students who don't love mangoes = $\overline{M} = \{x | x \notin M\}$
6) **Cross product :** The set of ordered pair of students = $M \times N = \{(x, y) | \, x \in A, y \in B\}$

These operations are not binary in nature you can apply operation on multiple set also.

Let's look at some membership problem which includes set operations :

1) $x \in (A \cap B)$ – means $x \in A \; and \; x \in B$
2) $x \in (A \cup B)$ – means $x \in A \; or \; x \in B \; or \; both$

3) $x \in (\bar{A} \cap B)$ – means $x \notin A$ *and* $x \in B$



$= B - A$

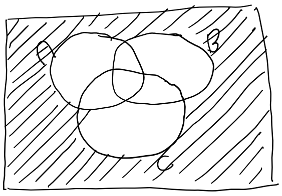
3.1.4) Set equality : Suppose A and B are sets. Then $\boldsymbol{A = B}$ **if and only if** $\boldsymbol{A \subseteq B}$ **and** $\boldsymbol{B \subseteq A}$**.** This fact is often used to prove set identities.

Let's discuss two approach to conclude that two sets are equal.

$$(\bar{A} - B) - C = (\bar{A} - C) - (B - C)$$

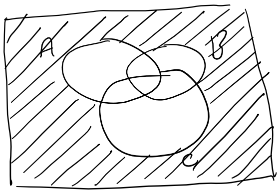
Method 1 : using Venn diagram,

LHS:



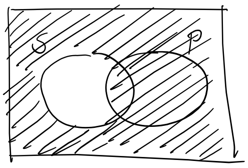
=

RHS:



Method 2 : Using analysis, we use definition of A = B discussed above.

$(\bar{A} - B) - C \subseteq (\bar{A} - C) - (B - C)$



Assume $x \in (\bar{A} - B) \cap \bar{C}$

$x \in \bar{A} \cap \bar{B} \cap \bar{C}$

$x \in \bar{A} \cap \bar{B} \cap \bar{C} \cap \bar{C}$

$x \in (\bar{A} - C) \cap (\bar{B}) \cap (\bar{C})$

Note that if $\boxed{x \notin S \text{ then } x \notin S - P}$ But

if $x \notin S - P$ then $x \notin S$ ← This is false

if $\boxed{x \notin S - P \text{ then } x \notin S \text{ or } x \in P}$

we can ommit this condition as it is extra

→ $x \in (\bar{A} - C)$ & $x \notin B$

$\underbrace{x \in (\bar{A} - C) \ \& \ x \notin (B - C)}$

$x \in (\bar{A} - C) - (B - C)$

hence, $(\bar{A} - B) - C \subseteq (\bar{A} - C) - (B - C)$

Similarly, we prove that $(\bar{A} - C) - (B - C) \subseteq (\bar{A} - B) - C$

Assume, $x \in (\bar{A} - C) - (B - C)$

$x \in (\bar{A} - C)$ & $x \notin (B - C)$

$x \in \bar{A}$ & $x \notin C$ & $x \notin B - C$

$x \in \bar{A}$ & $x \notin C$ & $x \notin B$ or $x \in C$

$x \in \bar{A}$ & $x \notin C$ & $x \notin B$

$x \in (\bar{A} - B)$ & $x \notin C$

$x \in (\bar{A} - B) - C$, hence both sets are equal.

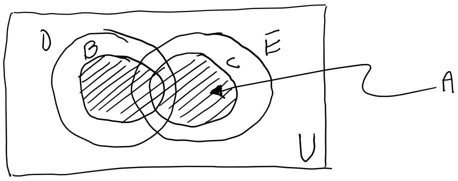
Question :

1) How may subsets and proper subsets does set A have if it contains 3 elements ?

Answer : n = 3, subsets = $2^n = 2^3 = 8$, proper subsets = $2^n - 1 = 7$

2) For any sets A, B, C, D and E where $A \subseteq B \cup C, B \subseteq D, and\ C \subseteq E$, show that we have $A \subseteq D \cup E$.

Answer :



3.1.5) Set Identities :

- Identity laws : $A \cup \emptyset = A, \quad A \cap U = A$

Let # be any operator on set, *Identity element* "e" for operation should satisfy X # e = X **and** e # X = X.

$A \oplus ? = A \Longrightarrow A \oplus \emptyset = A$ so, here identity element is $\emptyset$ for ExOR operation. But for set difference operator $A - \emptyset = A$ is true but $\emptyset - A \neq A$. So, no identity element exists for set difference operator.

Similarly, $A \times \emptyset \neq A$ Not only this but there does not exits identity element for cross product operator. But $\boldsymbol{A \times B = A\ iff\ A = \emptyset}$

- Domination laws : $A \cup U = U, \quad A \cap \emptyset = \emptyset$
- Idempotent laws : $A \cup A = A, \quad A \cap A = A$
- Complementation law : $\bar{\bar{A}} = A$
- Complement laws : $A \cap \bar{A} = \emptyset, \quad A \cup \bar{A} = U$
- Commutative laws : $A \cup B = B \cup A, \quad A \cap B = B \cap A$
- Associative laws : $A \cup (B \cup C) = (A \cup B) \cup C, A \cap (B \cap C) = (A \cap B) \cap C$
- You know distributive laws and De Morgan's laws :
- Absorption laws : $A \cup (A \cap B) = A, A \cap (A \cup B) = A$

Precedence of set operators : $\bar{A} > \cap > \cup$

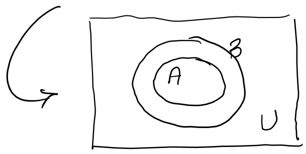
3.1.6) Proofs involving power sets :

1) We know that *if* $x \in P(A)$ *then* $x \subseteq A$ and *if* $x \subseteq A$ *then* $x \in P(A)$. From these two sentences we can conclude that $x \in P(A)\ iff\ x \subseteq A$.
2) Every element in P(A) is a set. (empty or non-empty)

Now, using these two properties let's prove see one example,

For any sets A and B, show that we have $A \cap B = A$ if and only if $A \in P(B)$ – Here if and only if is given which means we have to prove in both directions. First let's prove right implication. From first we can say that RHS is equivalent to $A \subseteq B$. Therefore, now we have to prove if $A \cap B = A$ then $A \subseteq B$.

Converse is also true. Means both directions are valid.



Suppose A and B are sets. Show that $P(A) \cap P(B) = P(A \cap B)$. – This P is not of probability. This is powerset notation. LHS means some subsets of A and B are common. This also means some element of A and B are common which is nothing but RHS.

Show that if A and B are sets, and $P(A) \subseteq P(B)$, then $A \subseteq B$. – Powerset contains all the possible subsets of a set. If one set of subsets are subsets of some other set of subsets meaning those elements are common in both sets. So, $A \subseteq B$.

3.1.7) Ordered Pairs and Cartesian (cross) Product :

It is a sequence of two elements a, b, where order matters and repetition matters. Notation : (a, b)

You can also compare it's features with sets : In set, Order doesn't matter, repetition doesn't matter. In ordered pairs, order matters, repetition matters.

sets	Pairs
1) $\{3,2\} = \{2,3\}$	(1) $(a,b) \neq (b,a)$
2) $\{2,2\} = \{2\}$	2) $(b,b) \neq (b)$
Notation : $\{\ \}$	Notation : $(\ ,\ )$

When can we say that ordered pairs are equal ? – this is same as asking when (a, b) = (c, d). This can only happen when a = c and b = d.

Similarly, we also have another type of structure also called **sequence**.

1) *Set* (finite or infinite) : Order doesn't matter, repetition doesn't matter.
2) *Ordered n-tuple* (finite) : Order matters, repetition matters.
3) *Sequence* (finite or infinite) : Order matters, repetition matters. Notation for sequence is < > where we can write <1, 2, 3, 4, …> Only difference between ordered n-tuple and sequence is that ordered n-tuple is finite and sequence can be finite or infinite and yes notation.

Cartesian Product :

$$A \times B = \{(x, y) | x \in A \text{ and } y \in B\}$$

$$B \times A = \{(x, y) | x \in B \text{ and } y \in A\}$$

In general, we can say that $A_1 \times A_2 \times \ldots \times A_n = \{(x_1, x_2, x_3, \ldots) | x_i \in A_i\}$

Cardinality of cartesian product : Let we have set A and B such that |A|=m and |B|=n then $|A \times B| = mn$

What is $A \times \emptyset$? – Let's say A = {1, 2} so, $A \times \emptyset = \{(x, y) | x \in A; y \in \emptyset\}$. $A \times \emptyset = \{(1, ?), (2, ?)\}$ but $\emptyset$ does not contains any elements so, no pairing is possible and it will be empty set. $A \times \emptyset = \emptyset$.

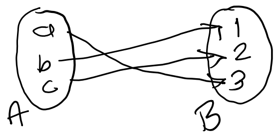
When $A \times B = B \times A$? – *if* A = B or *if* $A = \emptyset$ *or* $B = \emptyset$.

Some useful properties of cartesian product : it is neither associative nor commutative.

Q : What is the meaning of $\{1, 2\}^4$? – we know that simple {1, 2} is it a set now, $\{1, 2\}^2$ meaning it is a relation {1, 2} × {1, 2} = {(11), (12), (21), (22)} similarly $\{1, 2\}^4$ = {(1111), (1112), (1121),…(2222)}

3.2) Relations :

Relation can be seen in many areas. Let's take Facebook friendship example,



a has 3 friend
b has 1 friend
c has 2 friend

So somehow A is Related to B

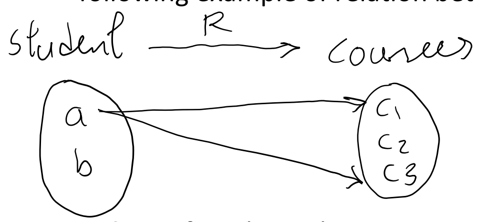
We write it as a R 3 or b R 1… Can we say cross product is relation because it is also ordered pair (a, b) where a is related to b ? – Answer is yes. Let's look at definition,

Definition : A *binary relation* R between two sets A and B (which may be the same) is a subset of the Cartesian product A X B. If the element $x \in A$ is related by R to the element $b \in B$, we denote this fact by writing $(a, b) \in R$, or alternately, by a R b. We say that *R is a relation on A and B*.

A relation on a set A is a subset of A x A.

Let's answer some interesting question,

1) Is relation from A to B are one-way ? – question asks to prove if a R b then b R a ?. Consider following example of relation between student and course offered.



xRy

student course

xRy iff x takes y.

Can C_1 R a ? – Definitely not how come course can take student this is nonsense sentence.

Let's take some problem related to notation,

2) If A is related to B we write $R : A \rightarrow B$. we also write this as x T y iff R (some relation like x+y=1 or something). How to write $T: N \times N \rightarrow N$?

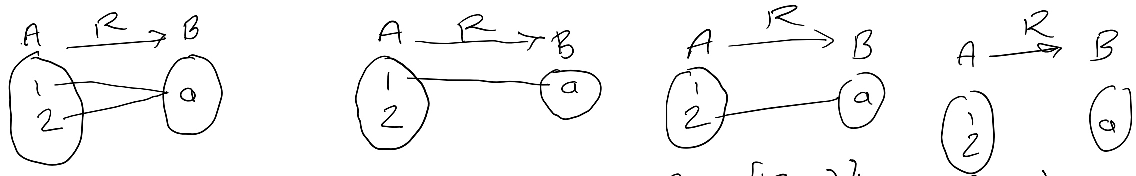
Answer : Note that relation is from N x N to N which means we have (a, b) relation on LHS and a on RHS. So, we write (a, b) T y iff a+b=y. Here a+b=y defines the relation and (a, b) represents the terms of N x N. we can also say that relation $T \subseteq (N \times N) \times N$.

3) We know that $0 \div 0 = Inderminate$ but what is 0 / 0 ?

Answer : Remember $0 \div 0$ here answer is some number and 0/0 is asking is 0 divides 0. In such cases we have to apply definition. a/b means b = an where n is some integer. So, 0/0 is also true as 0 = a. 0.

3.2.1) Counting number of relations :

Now, let's take one question say if we have two sets A and B. How many different numbers of relationship can they have ? – Let's say A = {1, 2} and B = {a}.



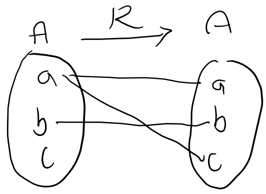
R={(1,a), (2,a)} R={(1,a)} R={(2,a)} R=ϕ

We know that $R: A \rightarrow B$ and $R \subseteq A \times B$.

i.e. every relation from A to B is subset of A x B. Every subset of A x B is a relation from A to B. Which means **Number of relations = Number of subsets of A x B. = $2^{|A \times B|}$ = $2^{|A| \times |B|}$**

3.2.2) Relation on a set :

Say we have relation $R: A \to A$. Then which of the following is true ?



a is related to c – True

c is related to a – False

a, c are related – Ignore writing this this is nonsense

a, c are related to each other. – It means a R c and c R a which is false here in this case.

In above example we saw interesting type of relation which is $R: A \to A$. We call it **R is on set A**. Which means if I say R is **on set** N x N. it means $R: N \times N \to N \times N$. Which also means $R \subseteq (N \times N) \times (N \times N)$.

When we say R is on set N x N. we know all about this from previous example. How to represent it in a R b iff R form. Answer is **(a, b) R (c, d) iff a = c or b = d.** From this answer following question :

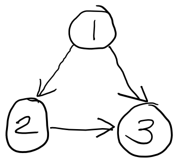
1) (1, 1) R (1, 1) – Correct because a = c or b = d
2) (0, 1) R (0, 2) – Incorrect because (0, 1) is not in our base set N x N (here N is natural number and 0 is not natural number)
3) (-1, 1) R (-1, 1) – Incorrect because (-1, 1) is not in our base set N x N.

Representations of Relations : We have already seen one type of graph representation. Consider R **on set** A. and A = {1, 2, 3} and x R y iff x<y. Therefore, R = {(1, 2), (1, 3), (2, 3)}

Matrix representation :

	1	2	3
1	✗	✓	✓
2	✗	✗	✓
3	✗	✗	✗

Another type graph representation :



3.2.3) Type of Binary Relations :

When a relation is defined **on** a set A : $R: A \to A$

Some special types of relations :

1) Reflexive, irreflexive
2) Symmetric, Anti-symmetric, Asymmetric
3) Transitive

Remember that this categorization is not for relation of the type $R: N \to Z$, meaning this categorization is only for relation **on set** type.

1) **Reflexive Relation** (reflexive, not reflexive, irreflexive) :

If every element is related to itself then we call this type of relation as *reflexive relation*. In other words, If we have relation R on set A (Base set) $\equiv R: A \to A$ then relation R is reflexive iff every element $\forall x \in A$ is related to itself.

Let's look at some example, A = {1, 2, 3}

- R = {(1, 1), (1, 2), (3, 3)} – False (Not reflexive)
- R = {(1, 1), (2, 2), (3, 2), (3, 3)} – True (Reflexive)
- R = {(1, 1), (2, 2)} – False

From above example, Reflexive relation R on A : R = {(1, 1), (2, 2), (3, 3), (1, 2), ...}

There must be there   There can be some extra

So, in FOL we say **Relation R *on set* A is *reflexive* iff** $\forall_{x \in A}(\boldsymbol{x\ R\ x})$

Question :

1) Equality relation on Z is reflexive ?

Answer : Question asks us for $R: Z \rightarrow Z$; aRb iff a=b. This is true because for every pair (a, b) as a = b we can write (a, a) which is obviously reflexive.

2) Let A = {1, 2, 3} and we ask subset relation on P(A) is reflexive ?

Answer : We can simplify question as $R: P(A) \rightarrow P(A)$; a R b iff $a \subseteq b$. Is this reflexive ? Yes, because every subset is subset of itself. **One thing to note that in all problems of relations we first find out base set and then move to actual problem.**

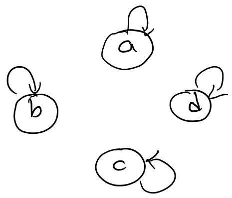
3) What will be matrix representation and graph representation of reflexive relation ?

Answer : Let A = {a, b, c, d}; Relation R on A.

R	a	b	c	d
a	1	☆	☆	☆
b	☆	1	☆	☆
c	☆	☆	1	☆
d	☆	☆	☆	1

1 ≡ ✓
0 ≡ ×
☆ ≡ (1 or 0)

matrix representation.



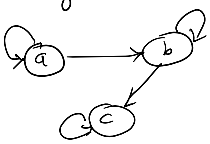
Graph representation

We have discussed reflexive and not reflexive relation. Now, we understand irreflexive relation.

Irreflexive and not reflexive are not same. Irreflexive is extreme opposite of reflexive relation. It says no element should be related to itself.

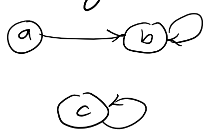
So, in FOL we say **Relation R *on set* A is *irreflexive* iff** $\forall_{x \in A}(\boldsymbol{x\ \not R\ x})$. Consider A = {a, b, c} then

Reflexive



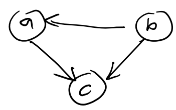
$\forall x \in A\ (x R x)$

NOT Reflexive



$\exists x \in A\ (x \not R x)$

Irreflexive



$\forall x \in A\ (x \not R x)$

2) **Symmetric Relation** :

We are now going to see *symmetric*, *anti-symmetric* and *asymmetric relation.*

If a R b then b R a then relation is *symmetric* relation.

If Base set : A

Relation R **on** A ≡ $R: A \to A$

Relation R is symmetric iff $\forall_{a,b\in A}(aRb \to bRa)$

One thing to note that if a R a then there is no problem to symmetric relation and if a !R a then also there is no problem to symmetric relation. Where !R is called "not related to".

Questions :

1) Equality relation **on** Z is symmetric ?

Answer : Z is set of integers (positive or negative). $R: Z \to Z; aRa\ iff\ a = b$. Yes, it is true as if two integers are equal then it is reflexive so it is also symmetric.

2) Let set A = {1, 2, 3} and subset relation on P(A) is symmetric ?

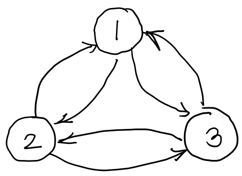
Answer : Base set is P(A) and question is $R: P(A) \to P(A); aRb\ iff\ a \subseteq b$. Here a and b should be element of P(A) and we have counter example as {1} ⊆ {1, 2} but {1, 2} ⊈ {1}. So, not symmetric.

3) Matrix and graph representation of symmetric relation. If base set A = {1, 2, 3} and some random relation

Answer :

R	1	2	3
1	*	✓	✓
2	✓	*	✓
3	✓	✓	*

Matrix Representation



Graph Representation

In short, we can say that if graph representation is given and if relation is symmetric then there should be no unidirectional edges.

4) When can you say that relation is "not symmetric" ?

Answer : $\exists_{a,b\in A}(aRb \to b \not R a)$

5) When (a, b) R (c, d) is called symmetric ? Remember here base set is A x A not A and $R: A \times A \to A \times A$.

Answer : Relation is reflexive when (a, b) R (a, b) satisfy. It is symmetric when (a, b) R (b, a) satisfy ? It's wrong it is symmetric when (a, b) R (c, d) and (c, d) R (a, b) satisfies plzz read this again. It is transitive if (a, b) R (c, d) and (c, d) R (e, f) then (a, b) R (e, f).

Anti-symmetric relation : If you have two <u>different element</u> a, b then this <u>should not happen</u> $a\ R\ b\ and\ b\ R\ a$. Which means

$R: A \to A$ and R is antisymmetric iff $\forall_{a,b\in A}((a \neq b \wedge\ a\ R\ b) \to b \not R\ a))$

OR $R: A \to A$ and R is antisymmetric iff $\forall_{a,b\in A}((b\ R\ a \wedge\ a\ R\ b) \to b = a))$

Asymmetric Relation (counter-symmetric relation) : <u>Asymmetric relation means relation should be anti-symmetric and irreflexive</u>. This is extension of anti-symmetric relation.

$\forall_{x,y \in A} \neg (aRb \wedge bRa)$ OR $\forall_{x,y \in A} (aRb \longrightarrow b \not R a)$

In short, we can say that asymmetric relation is anti-symmetric relation with no self-loop in graph representation.

3) Transitive relation :

It says if a R b and b R c then a R c. $\forall_{(a,b,c) \in A} (aRb \,\&\, bRc \longrightarrow aRc)$

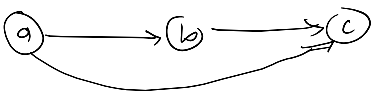
same / different

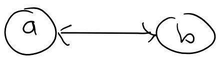
Now, it is very simple to prove that equality relation is transitive. And subset relation is also transitive because if x is subset of y and y is subset of z then x is subset of z. But now consider same subset relation on P(A) i.e. power set of A then also it is transitive.

But when relation is "not transitive" relation ? - $\exists_{(a,b,c) \in A} (aRb \wedge bRc \wedge a \not R c)$

same / different

Graph representation of transitive relation :



Is this transitive ? -  This is not transitive as a R b and b R a but a !R a.

3.2.4) Different type of closure :

If R is relation on a set A, it may or may not have some property P, such as reflexivity, symmetric, or transitivity. When R does not enjoy property P (same as saying when R does not have property P, this means some element of R satisfy and some not), we would like to find the smallest relation S on A with property P that contains R.

Which means we have to find out those relation pairs who satisfies property P and it is in A x A but it is not in Relation R. For example, the relation R = {(1,1), (1, 2), (2, 1), (3, 2)} on the set A = {1, 2, 3} is not reflexive. How can we produce a reflexive relation containing R that is as small as possible ? – This can be done by adding (2, 2) and (3, 3) to R, because these are the only pairs of the form (a, a) that are not in R but they are in A x A. So, now S = {(1, 1), (2, 2), (3, 3), (1, 2), (2, 1), (3, 2)} is reflexive closure of R.

Similarly, we have symmetric closure, transitive closure.

The time complexity of computing the transitive closure of a binary relation on a set of elements is $O(n^3)$

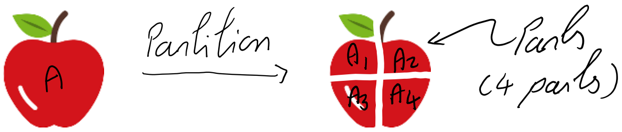
3.2.5) Equivalence relation :

A relation R is **equivalence relation** iff it is *reflexive*, *symmetric* and *transitive*.

Example, Consider relations on the set of integers. R = {(a, b) | a = b or a = -b} – If you look carefully it is absolute function i.e. |x| so, R = {(a, b) | |a| = |b|}. It is reflexive as a R a. It is also symmetric. Which means it is not asymmetric. It is also not anti-symmetric as 2 R -2 and -2 R 2 but 2 != -2. It is transitive. So, it is reflexive, symmetric and transitive and that is why it is *equivalence relation*.

Partition of a set :

Consider you have an apple, and you don't want to eat it alone. You want to share it with family so do partition of apple.



There is some observation that we can make

- $A_i \subseteq A$
- $A_i \cap A_j = \emptyset\ for\ 0 < i, j \leq 4\ and\ i \neq j$
- $A_1 \cup A_2 \cup A_3 \cup A_4 = A$
- $A_i \neq \emptyset$

Similar in sets consider a set S = {1, 2, 3, 4} and P : {{1}, {2, 3}, {4}} is partition of set S it is not only partition but it is one of the possible partitions of set S.

Definition : *Partition of set S is non-empty subset of S which are pairwise disjoint such that it includes all the element of set S.*

A set S is partitioned into k non-empty subsets $A_1, A_2, A_3, \ldots, A_k$ *if :*

1. *Every pair of subsets is disjoint.*
2. $A_1 \cup A_2 \cup A_3 \ldots \cup A_k = S.$

But we are not concern about partition of set we want total no. of partition possible. Let's take one example,

Let S = {1, 2, 3}, we know that we can divide this set to at most 3 parts. We have to make partition such that all four-point discussed in apple example should satisfy.

One part partition : {S} → 1

Two part partition : {{1}, {2, 3}}
{{2}, {1, 3}}
{{3}, {1, 2}} → 3

Three part partition : {{1}, {2}, {3}} → 1

In total there can be 5 partition of set S.

From now on we are not going to write like {{1}, {2, 3}} instead we write it like { 1, 23} (for our convenience) Let's answer questions.

{prime numbers, composite numbers} is a partition of N ? – False. Prime number contains {2, 3, 5, …} and composite numbers contains {4, 6, 8, 9, …} both sets are partition but 1 element is missing i.e. 1 number. That is why not a partition of N.

Question :

1) Let A = {2, 3, 4, 5, 6, 7, 8} and R a relation over A. Draw the directed graph of R, after realizing that x R y iff x – y = 3n for some n ∈ Z. Check that if every relation R satisfy.

Answer : x and y are pairs which should satisfy x – y = 3n condition for some n ∈ Z. x – y = 3n means x – y should be divisible by 3. Which means $x \equiv y \bmod 3$ or

x and y should have common remainder. If x = y then 0 is divisible by 3 so it is reflexive. If x-y is divisible 3 then y-x is also divisible by 3 which means it is symmetric.

Antisymmetric : ✗ because 3 R 6 and 6 R 3 but 6 != 3

Asymmetric : ✗ because it is not irreflexive.

Transitive : ✓ because if a R b means a mod 3 = r and b mod 3 = r as they share common remainder. Similarly, if b R c means b mod 3 = r and c mod 3 = r which means a R c. So, this is equivalence relation.

Now, we are going to introduce new notation which only applies on *equivalence relation* $[a]_R$. This is called **equivalence class of "a"**. where a is element of set.

$[a]_R = \{b | a\ R\ b\}$ It represents all the element which a relates to under relation R.

In above example, $[2]_R = \{2, 5, 8\}$, $[3]_R = \{3, 6\}$, $[4]_R = \{4, 7\}$ This is nothing but partition of set A.

$[5]_R = \{2, 5, 8\}$, $[6]_R = \{3, 6\}$, $[7]_R = \{4, 7\}$

One thing we can conclude with the above example that if set S is base set and R is equivalence relation on set S then R will partition the base set.

2) Consider base set A and relation which is complete relation then do relation analysis.

Answer : *If relation is complete relation then R = A x A*. It is reflexive and symmetric. It is also transitive but it is not antisymmetric but it is antisymmetric when |A| = 1. And it has only equivalence class because $[0]_R = \{0, 1, 2, 3, ...\}$ and $[1]R = \{0, 1, 2, 3, ...\}$ and so on if A = N.

3) Let S = {1, 2, 3, …, 19, 20} and define an equivalence relation R on S by $xRy \Leftrightarrow 4|(x - y)$. Determine the equivalence classes of R.

Answer : $R \Leftrightarrow x \equiv y \bmod 4$

$[1]_R = \{1, 5, 9, 13, 17\}$ equivalence class of 1

$[2]_R = \{2, 6, 10, 14, 18\}$

$[3]_R = \{3, 7, 11, 15, 19\}$ partition of S

$[4]_R = \{4, 8, 12, 16, 20\}$

because $[1]_R = [5]_R = [9]_R = [13]_R = [17]_R$

$[2]_R = [6]_R = [10]_R = \ldots$

$\vdots$

What if we ask another question ? What will be the cardinality of relation R on S. can we predict it from equivalence classes of R. Answer is yes. From equivalence classes of S we get useful information like

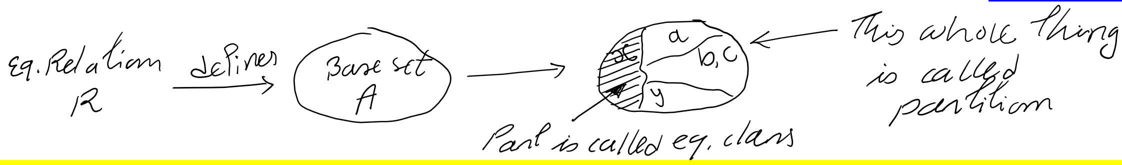
$R = \{(1,1), (1,5), (1,9), (1,13), (1,17),$
$(2,2), (2,6), (2,10), \ldots$
$\vdots \}$

We can $|R| = |[1]|^2 + |[2]|^2 + |[3]|^2 + |[4]|^2 \ \therefore |R| = 25 \times 4 = 100$

We say that Relation R has total 4 equivalence classes.

So, from above three question we can conclude that

- Every equivalence relation R on set A gives unique partition of A. and each part of partition is called equivalence class of R. $y, x, a, b, c \in A$



- If R is equivalence relation on set A and there are n equivalence classes of R which means it is n parts in partition. Let one part or equivalence class is E1 and nth part or equivalence class is En then

Disjoint

$$R = (E_1 \times E_1) \cup (E_2 \times E_2) \cup \dots \cup (E_n \times E_n)$$

$$\therefore |R| = |E_1|^2 + |E_2|^2 + \dots + |E_n|^2$$

- Complete relation has only one equivalent class which is set itself.

4) Why do we call such relations to equivalence relation ?

Answer : Because in every equivalence relation, there is some sense/type of "equality" involved. Normally, an equivalence relation R on a set A has the following structure. For all a, b in A; aRb iff a, b has same ******. ← what is this ?

Let's understand with equivalence relation example,

- x and y have the same color
- x = y ← same value
- x and y have the same area
- x and y are programs that produce the same output.

Two elements a and b that are related by an equivalence relation are called *equivalent*. The notation a ∼ b is often used to denote that a and b are equivalent elements with respect to a particular equivalent relation.

From question 3rd we can conclude more things like

Let R be an equivalence relation on a set A. These statements for elements a and b of A are equivalent : (i) a R b, (ii) [a] = [b], (iii) [a] ∩ [b] ≠ ∅

Because if a R b is relation between a and b and relation is equivalence then classes of a and b should contains same element because if a is related to some element which means that element is related to b (by symmetric) and same as b and a is related to b (by transitive) so element in classes of a and b are same and not null.

5) Let R be an equivalence relation on set A. Which of the following is/are true?

Answer :

- Every equivalence class of R is subset of R. ✗
- Every equivalence class of R is subset of A. ✓
- Set of equivalence classes of R is subset of p(A) ✓
- Set of equivalence classes of R is subset of A. ✗
- Two different equivalence classes are disjoint. ✓
- Set of equivalence classes of R is partition of A. ✓
- Set of equivalence classes of R is partition of P(A) ✗
- Set of equivalence classes of R is partition of R. ✗

6) What is the largest and smallest ER (Equivalence relation) on A (largest and smallest in terms of cardinality) ?

Answer : Largest relation on A is R = A x A which has cardinality of $|A|^2$,Which has only one equivalent class as all element are related to each other. Smallest equivalent relation on A is when R is identity relation. Meaning it contains all reflexive element. Cardinality of such relation is |A| and have |A| number of equivalent class as one element is related to itself.

7) What is the No. of equivalence relation with n element in set ?

Answer : We know that equivalence relation has equivalence class which partition the base set. Which means **no. of different partition = no. of equivalence relation**. Because same partition means same class combination. No. of different partition of set with n element is given by BELL number B_n. nth bell number can be found easily from the bell triangle as follows : Here,

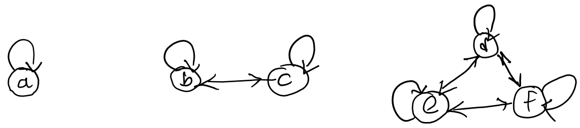
$E_{(i,j)} = E_{(i-1,j-1)} + E_{(i,j-1)}; i, j > 1,$ 1

$E_{(1,1)} = 1, E_{(i,1)} = E_{(i-1,i-1)}$ →

NOTE :

1) **a | b is called divides "relation" and it return true or false. a / b is called division operator and it return value.**
2) **If base set is empty and we have some relation R on empty set then obviously R = empty set. Now, recall universal quantifier applied on empty set always results in true. So, for reflexive, symmetric, antisy-+*mmetric, asymmetric, irreflexive, transitive it will result in true. So, every relation on empty set is both equivalence and partial.**
3) **If question asks for number of equivalence class of n then remember number of equivalence class of n = Number of partitions of n = no. of equivalence relation.**

Graph of equivalence relation : If R is equivalence relation on set A having equivalence class E1, E2, E3. Now, consider E1 = {a}, E2 = {b, c}, E3 = {d, e, f} then graph will look like this,



3.3) Partial Order relation :

When we use relation to order some or all of the element of set.

For instant, we order words using the relation containing pairs of words (x, y), where x comes before y in the dictionary. We schedule projects using the relation consisting of pairs (x, y), where x and y are tasks in a project such that x must be completed before y begins.

We order the set of integers using the relation containing the pairs (x, y), where x is less than y. When we add all of the pairs of the form (x, x) to these relations, we obtain a relation that is reflexive, antisymmetric, and transitive. There are the properties that characterize relations used to order the element of sets.

Consider an example of gate aspirants when at the beginning of his preparation he doesn't know the order to which subjects should be follow. So, in that case he doesn't get full order or total order instead he would get partial order like

$$\left.\begin{array}{l} DM \rightarrow TOC \rightarrow CD \\ DLD \rightarrow CO \rightarrow OS \\ EM \\ GA \end{array}\right\} \text{Partial order}$$

But now, consider another example of order of standard in which parents wants their child to be in. In that case we have total order.

$$KG \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow \dots \rightarrow 12 \rightarrow \dots \} \text{ Total order}$$

Now, here if you see if we want order in element of set, do we need symmetric relation ? – Answer is no. because symmetric relation tells us that a R b and b R a which means there is no order. But we want order so that is why for ordering elements of set we use antisymmetric order. And similarly, we want transitive relation between element of set. Because if DM R TOC and TOC R CD then definitely DM R CD.

But why we need reflexive relation ? - Consider new gate aspirant who want to order apt, TOC, CN but there is no relation between subjects (apt, TOC, CN can be studied independently) so there is no relation means there is no order in element of set. But we know we have to follow some order (to study these subjects) that is why reflexive relation should be there in partial or total order.

Definition : A relation R on a set A is *partial order* (or partial ordering) for A if R is reflexive, antisymmetric and transitive. A set A with a partial order is called a partially ordered set, or **POSET**.

Notation : (Base set, relation) for example, (N, <) although this is partial order relation (just example). A set S together with a partial ordering R is called a *partially ordered set*, or *poset*, and is denoted by (S, R)

Some standard partial order relation :

- $(N, \leq)$ is partial order relation because it is reflexive, antisymmetric and transitive.
- $(N, \geq)$ is also partial order relation it is RAT.
- $(P(A), \subseteq)$ is also partial order relation it is RAT.
- *Divisibility relation* : R on N, x R y iff x | y.

In equivalence relation if we want to express relation between two elements, we use ~ symbol, in partial order relation we use ≼ symbol.

$(\{1,2,3,4\}, R)$ $\xleftrightarrow{\text{Same}}$ $(\{1,2,3,4\}, \preccurlyeq)$ Partial order symbol

xRy iff $x|y$ $\qquad$ $x \preccurlyeq y$ iff $x|y$

This is not less than equal to

Some terms we need to know before moving on :

Comparable and non-comparable : If R is partial order relation on S then we say that a and b are comparable iff one of them is related to another ($a \preccurlyeq b$ *or* $b \preccurlyeq a$). If none of them are related to one-another we say they are incomparable or non-comparable.

For example, ({1, 2, 3, 4}, |} then

Comparable elements : 1 | 2, 2|4, 1|3,...

Non-comparable elements : 2|3, 3|1,...

$1 \leq 2 \leq 4$
$1 \leq 3$
$2 \nleq 3$

Partial order

Not total order

Not partial order relation
(They are partially related)

But when we have ({1, 2, 4, 8}, |} then we have total order. Every element is comparable.

$$1 \preccurlyeq 2 \preccurlyeq 4 \preccurlyeq 8$$

Thus, **Total order or Total ordering relation** is relation which is partial order relation + every element is comparable. Example, poset $(Z, \leq)$ is totally ordered, because $a \leq b$ or $b \leq a$ whenever a and b are integer.

3.3.1) Hasse diagram :

We know that any relation R on set A has many representations :

- Graph representation
- Matrix representation
- Set representation
- Arrow diagram representation $(DM \rightarrow TOC)$

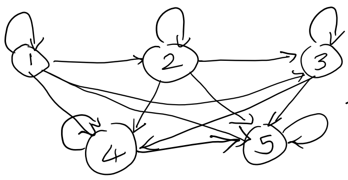
But when we have partial order relation, we have one more type of representation called hasse diagram and there is a relationship between hasse diagram and partial order relation.

For every partial order relation → we have hasse diagram

And if *hasse diagram for some relation is given → relation is partial order relation*

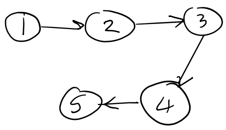
So, in short, we can say that <u>it is a property and definition of partial order relation</u>.

Now, consider a partial order relation $(\{1, 2, 3, 4, 5\}, \leq)$, graph representation is



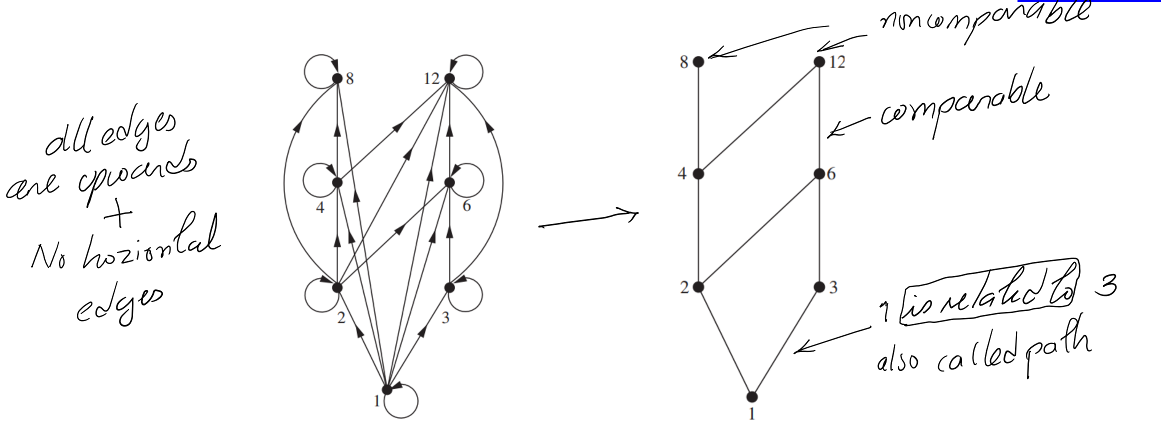
This is getting ugly

This is good

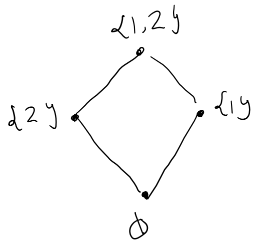


So, to reduce complexity if we know that certain relation has some common property, we omit those obvious conditions. So, if relation is partial order then we omit reflexive edges, transitive edges. By doing so we have graph like representation which is clean but it is not hasse diagram. In hasse diagram we also omit arrows. But after omitting arrows we lost sense of relation, for that we do this

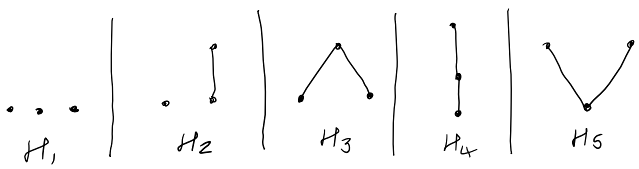
By doing we have not lost order of relation and all property of partial order is also satisfies.



Let's take another example, Consider POR (partial order relation) $(P(\{1,2\}), \subseteq)$



What are the possible hasse diagram of POR with 3 elements ? –



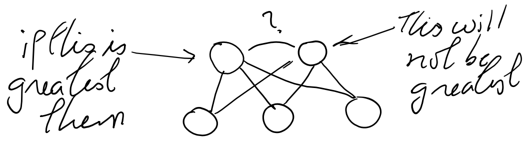
3.3.2) Special elements of POSET :

We shall go to see maximal, minimal, greatest, least elements of poset.

- **Maximal element** : It is element of poset which is not related to <u>any</u> element of same poset. so, in hasse diagram there is no path from m to anyone else (no edge)
- **Minimal element** : It is element of poset which is not related by <u>any</u> element of same poset. so, in hasse diagram, there is no path from element to minimal element.

- **Greatest/ Maximum element** : It is element of poset which is related by <u>every</u> element of poset.
- **Least/ Minimum element** : It is element of poset which is related to <u>every</u> element of poset.

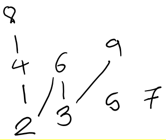
In a poset, can we have more than one greatest/maximum element ? – Let's say they are possible then by definition of greatest element, it is related by every element of poset. Let's create possible cases,



Which means if greatest and least exists in some hasse then they must be unique. This also implies *if some poset has more than one maximal or minimal element then we cannot have greatest and least element respectively.*

Can we have elements which are both minimal and maximal ? – Let's look at below example,

$(\{2,3,4,5,6,7,8,9\}, |)$

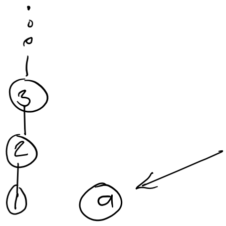


maximal = 8, 6, 9, 5, 7

minimal = 2, 3, 5, 7

So, one element can be both minimal and maximal (condition is it should not be related to or by related to any element).

If some POSET has one unique maximal element then it is maximum or greatest ? – If you think you will say it is true but take this beautiful case where POSET is $(N, \leq) \cup \{(a, a)\}$ this is POSET. and if you look at hasse diagram.



maximal but not greatest

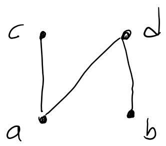
Similarly, we can conclude that if some POSET has unique minimal element then it is not necessary to be minimum or least element.

But in above case, POSET was infinite. What if POSET is finite will it still hold ? – then answer becomes yes in both minimal and maximal.

Now we are going to see more of these special elements namely upper, lower, greatest lower, least upper bound. For this consider a set A and X be any subset of A.

- **Upper bound** : Upper bound of X is the element a ∈ A such that all the element of X is related to a.
- **Lower bound :** Lower bound of X is the element a ∈ A such that all the element of X is related by a.

So, in short, upper and lower bound is same as finding maximum and minimum element of subset of poset. Consider example given below :



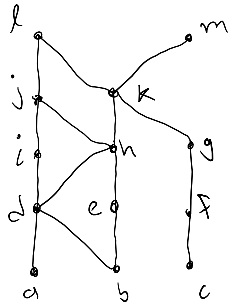
minimal = {a, b}
maximal = {c, d}
minimum = Φ
maximum = Φ

consider X = {c, d}
lower bound of X = a
we write LB(X) = {a}

LB({c, a}) = {a} ← a is related to a & c
UB({c, a}) = {c} ← c & a is related to c

- **Least Upper bound** : It is the least element in the upper bound of subset of poset. (first you find upper bound of X then you find least element)
- **Greatest Lower bound** : It is the greatest element in the lower bound of subset of poset. (first you find lower bound of X then you find greatest element).

Consider following example : let's say X = {d, k, f}



minimal = {a, b, c}
maximal = {l, m}
Greatest = Φ
least = Φ
LB(X) = Φ
UB(X) = {l, m, k}

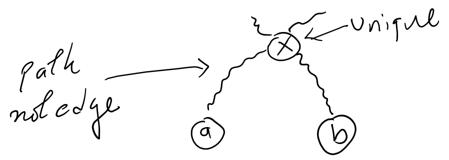
always unique why?
LUB(X) = {k}
GLB(X) = Φ

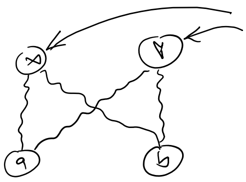
NOTE :

1) **In most of the cases we want to find GLB and LUB for pair of elements. So, we denote GLB{a, b} as a Λ b (also known as meet) and LUB{a, b} = a V b (also known as join)**
2) **In finding different cases to prove some option wrong, you can also take infinite POSET for example, (Z, <=) this is special type of POSET where there is no minimal or maximal element.**
3) $(a_o \preccurlyeq a)$ **means a_o is related to a. which also means a0 is at bottom in hasse diagram.**
4) **In relation, base set is by default, non-empty unless it is explicitly mentioned.**

Some standard result you must think about :

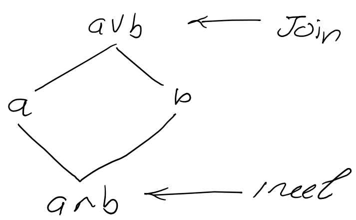
- a Λ a = a
- if a R b (comparable) then a Λ b = a and a V b = b
- if a and b are incomparable then a V b = *Unique* first **joining** point in upward direction. (if not unique then a V b does not exists)



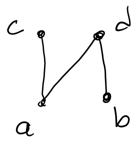


a v b but not unique because a v b should be unique

- if a and b are incomparable then a Λ b = *Unique* first **meeting** point in downward direction. (if not unique then a Λ b does not exists)

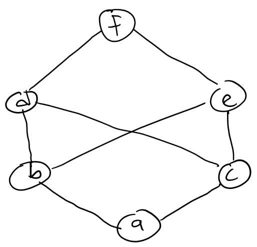


Trick is J comes before m



c v c = c
a ∧ a = a } standard
a v c = c } a, c comparable
a ∧ c = a

a v b = d
a ∧ b = DNE (Doesnot...)
c v b = DNE

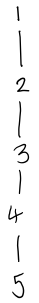


b v c = Not unique so DNE
OR UB{b, c} = {d, e, f} now least element is both d & e but it should be unique
b ∧ a = a
d ∧ e = Not unique so DNE

If we have poset P (poset is nothing but partial order set P) then what is ∧ P and ∨ P ? - ∧ P = LUB of all element in set P = Greatest element, and ∨ P = GLB of all element in set P = least element (think)

3.3.3) Hasse diagram of TOR :

Consider following poset and its hasse diagram , ({1, 2, 3, 4, 5}, >=), (N, |)



chain & it is TOR
every element is comparable
Thus
TOR ⟷ Hasse diagram chain

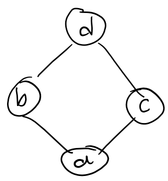
$2 \vee 3 = 6$
$10 \vee 8 = 40$ (LCM)
In short,
any $a, b \in N$
$a \wedge b = GCD(a,b)$
$a \vee b = LCM(a,b)$

One thing to note that POR $(p(A), \subseteq)$ is TOR iff |A|<=1. (think)

3.4) Lattice :

Definition : A *lattice* is a poset $(A, \preccurlyeq)$ in which any two elements a, b has an LUB(a, b) and a GLB(a, b). which means $\forall_{a,b \in A}(a \wedge b \text{ should exists and } a \vee b \text{ should exists})$

Now what we will need to do to check for lattice. We have to check all these cases.



$\cancel{a \vee a = a}$
$\cancel{a \wedge a = a}$
$\cancel{a \vee b = b}$
$\cancel{a \wedge b = a}$
$\cancel{a \vee c = c}$
$\cancel{a \wedge c = a}$

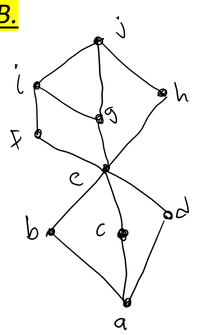
$\cancel{a \vee d = d}$
$\cancel{a \wedge d = a}$
$b \vee c = d$
$b \wedge c = a$
$\cancel{b \wedge b = b}$
$\cancel{b \vee b = b}$

We know that aRb then there always LUB & GLB

This also mean a & b are comp.

To check if a given poset (of n elements) is lattice or not, we have to check for EVERY pair of elements to see if every pair of elements has LUB and GLB. It will take a lot of time as number of pairs of elements is n^2. Can we do better than this ? – Answer is check only pairs of incomparable elements. Because Pair of comparable elements always have GLB and LUB in every poset. So, there is no need of checking of them.

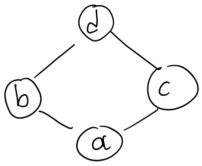
So, another definition of lattice : ***A POSET in which every two non-comparable elements have Unique GLB and LUB.***



In this you only have to check for,
b, c, d, f, g, h, i

Can we count cardinality of POR through hasse diagram ? – Let's consider following hasse diagram of POSET.

|R| = ?



$= 4 + 3 + 1 + 1 = 9$

for b

for c

4 element so 4 reflexive pairs

asymmetric & transitive pairs of (a, x)

any element other than a

Consider total order relation instead what would be the cardinality of TOR through hasse diagram ? –

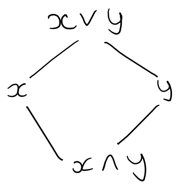
$|R| = (a,x) + (b,x) + (c,x)$

$3 \quad 2 \quad 1 = 6 = \frac{3\times 4}{2}$

(1 reflexive + 1 anti + 1 transitive)

And this will be the maximum cardinality of relation with 3 elements (i.e. chain like hasse diagram). And the Minimum cardinality of relation with 3 elements will be relation which is reflexive only.

One thing to note that in every lattice L, $\forall_{x,y} \in L$



$xR(x\vee y)$
$yR(x\vee y)$
$(x\wedge y)Rx$
$(x\wedge y)Ry$

$(x\wedge y)R(x\vee y)$

Properties of lattices : Any lattices have the following properties. (**ICAA**)

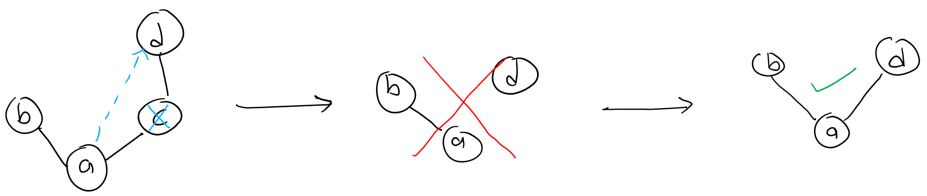
- Idempotent $(x \wedge x = x; x \vee x = x)$
- Commutative $(x \wedge y = y \wedge x; x \vee y = y \vee x)$
- Associativity $((x \wedge y) \wedge z = x \wedge (y \wedge z); (x \vee y) \vee z = x \vee (y \vee z))$
- Absorption $(x \wedge (x \vee y) = x; x \vee (x \wedge y) = x)$

But Identity, complement, de-morgen, distributive properties are only satisfied by some lattices, not by all lattices. (think why they do not satisfy in case you forgot)

3.4.1) Sublattice :

Before starting new topic, we will address some interesting questions,

If we delete an element from Base set in any POSET will it still be POSET ? – If you delete any element from base set then its relation would also affect the same so every entry where that element appears will get deleted from relation. And therefore, it will still be POSET because deletion of element from base set also deletes relation associated with it so it will not affect reflexive, anti-symmetric and transitive relation.



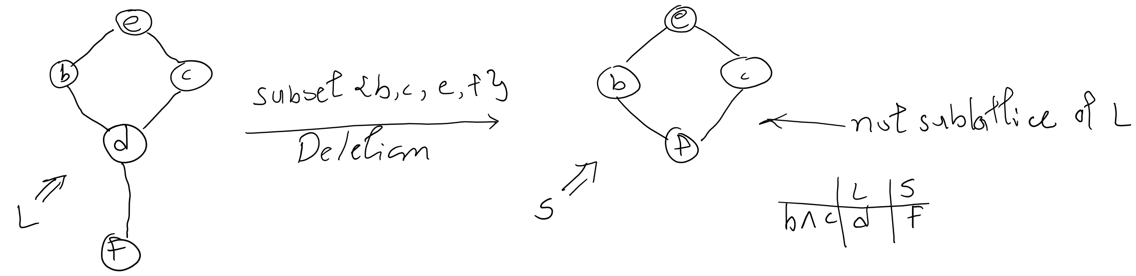
In short, we can say that If (A, R) is POSET then (B, R) is also POSET $\forall B \subseteq A$

Similarly, if (A, R) is Eq. relation then $\forall B \subseteq A$, (B, R) also an eq. relation

If we delete some element from relation, can I guaranty that result will be POR ? – NO because If you delete any element from relation then consider that element is of the form (a, a) then it will not satisfy reflexive relation and that is why relation will not be POR.

But why we are explaining this under sublattice ?

So, sublattice S of L is nothing but (i) Subset of L, (ii) S should be lattice and (iii) GLB, LUB for any a, b∈S must be same in S and L.



Intuition behind sublattice : If you take subset of lattice then take their GLB, LUB also. So, in previous example we are taking b and c but not talking its GLB from base set.

Clearly, above diagram shows that **if L is lattice and S is lattice and subset of L then it is not necessary that S is sublattice.**

Definition : A *sublattice of a lattice L* is a nonempty subset of L that is a lattice with the same meet and join operations as L.

3.4.2) Types of lattice :

We know that all lattice satisfies four properties i.e. ICAA, but talking about identity, distributive, complement then some lattices satisfy them, some don't. So, that is why if some lattice satisfies

- **Bounded lattice** (lattice which satisfies *identity* property)
- **Complemented lattice** (Lattice which satisfies *complement* property)
- **Distributive lattice** (Lattice which satisfies *distributive* property)
- **Boolean lattice** (Lattice which satisfies *all these three* property)

1) Bounded lattice *:*

A lattice $\langle A, \preccurlyeq \rangle$ is **bounded** iff it has a minimum element and a maximum element. These are denoted by 0 and 1 respectively.

Bounded lattice has least and greatest element which means all elements are bounded by these two elements.

Can infinite lattice be bounded lattice ? – Answer is Not possible. Because take simple example of TOSET (total order set) $(Z, \leq)$ which do not have least and greatest element. But you are wrong, It is possible take one beautiful example of TOSET $([a, b], \leq)$ this is real interval from a to b including a and b. This is infinite but it has least element (a) and greatest element (b).

Now, we know that every finite lattice has greatest and least element.

Every finite lattice is bounded

Some facts about bounded lattice :

- Suppose $\langle A, \preccurlyeq \rangle$ is a bounded lattice having minimum 0 and maximum 1, and let x be any element in A. then

$$0 \vee x = x = x \vee 0; 1 \wedge x = x = x \wedge 1$$

$$0 \wedge x = 0 = x \wedge 0; 1 \vee x = 1 = x \vee 1$$

- For entire lattice we know that upper bound is greatest element and lower bound is least element. So, another definition of bounded lattice can be lattice L is bounded iff it has entire lattice L has Upper bound and Lower bound.
- *Infinite lattice which is not bounded*

$(N, |)$

least element = 1

Greatest element = DNE

Infinite lattice which is bounded

$(P(N), \subseteq)$

least element = ϕ

Greatest element = N

$(W, |)$

$W = \{0, 1, 2, 3, 4, \ldots\}$

least element = 1

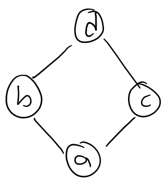
Greatest element = 0

Wait, we have mention involvement of identity property in bounded lattice but by far we have not seen it. So, let's see…

Identity property : We say "e" is *identity element* for operator # iff for all a, a # e = a **and** e # a = a.

For addition, identity element is 0 because for all a + 0 = a.

For subtraction, we don't have identity element because 0 – a = -a and a – 0 = a both are different but according to definition both should be same.



take operator # as GLB then

$d \wedge d = d$

$b \wedge d = b$

$c \wedge d = c$

$a \wedge d = a$

therefore d is identity element for GLB

Similarly, <u>for *any* lattice, least element (if exists) will be identity element for LUB (join or ∨) and greatest element (if exists) will be identity element for GLB (meet or ∧).</u>

That is why another definition of bounded lattice can be **in lattice, if identity element is least element and greatest element for LUB and GLB respectively then it is bounded lattice**.

Domination law : In set theory, if M is universal set then $S \cup M = M$. Here we say that M is dominator because it is dominating S.

Then dominator for GLB in lattice – least element

Dominator for LUB in lattice – greatest element

For example, $(N, \leq)$, Dominator for LUB = DNE; dominator of GLB = 1.

In short, we have seen three definition of bounded lattice :

(i) Lattice with least and greatest element
(ii) Lattice with Identity element for GLB and LUB be greatest and least respectively.
(iii) Lattice with dominator for GLB and LUB be least and greatest respectively.

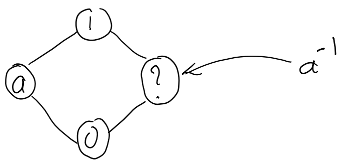
2) Complemented Lattice :

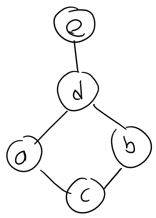
Lattice is complemented iff every element has at least 1 complement. First of all, complement of an element "a" be "b" iff $a \vee b = 1 \; and \; a \wedge b = 0$. Here 1 means greatest and 0 means least.

And if you see carefully here greatest and least element exists which means it is also bounded lattice so, for lattice to be complemented it first has to be bounded lattice.

Complemented lattice $\underset{\times}{\overset{\checkmark}{\rightleftarrows}}$ bounded lattice

Idea is in a bounded lattice, complement of a ≡ a^{-1}



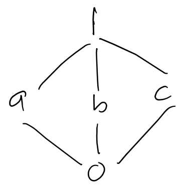


$a \vee b = d \neq e$
$a \wedge b = c$
$b \neq a^{-1}$
$a^{-1} = DNE$

This is not bounded so not complement

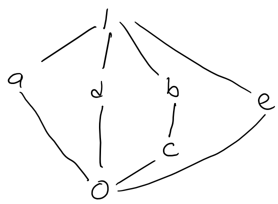
Example of complemented lattice :

1)



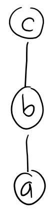
$1^{-1} = 0$ & $0^{-1} = 1$ because $(\text{greatest})^{-1} = \text{least}$ & vv
$a^{-1} = b, c$
$b^{-1} = a, c$
$c^{-1} = a, b$

2)



$1^{-1} = 0$ & $0^{-1} = 1$
$a^{-1} = b, c, d, e$
$b^{-1} = a, d, e$
$c^{-1} = a, e, d$
⋮

3)



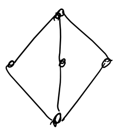
$b^{-1} = ?$
$b^{-1} = DNE$

Not complemented

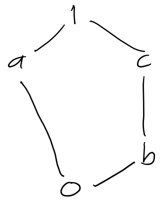
Q : Can a TOSET be complemented lattice ? – From previous example it seems like answer is no. But answer is yes, it is possible when iff <= 2 elements.

3) Distributive Lattice :

Lattice is distributive iff $\forall_{a,b,c} \; a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \; and \; a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$



kite lattice & Diamond & M_3
It is not-Distributive lattice

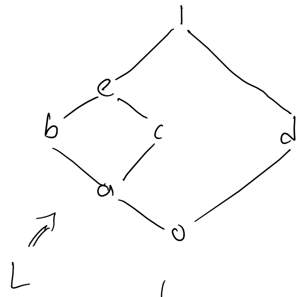


Not Distributive lattice
This is called pentagon lattice (N_5)

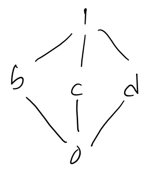
Above lattice is special lattices.

Theorem : if *Kite* or *pentagon* sublattice then lattice is not distributive. **OR** L is distributive iff there is no sublattice of L which is *kite* or *pentagon*.

Q : If lattice L has <= 4 elements then L is definitely distributive ? – Answer is yes. Because if less than 5 elements assure that there is no kite or pentagon sublattice. But if you have 5 elements than also it is possible to have distributive lattice why because there are many lattices which do not have kite or pentagon sublattice with 5 elements.



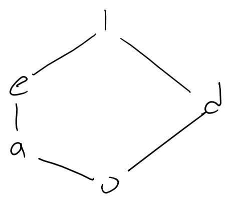
$S = \{1, b, c, d, 0\}$



This is not even a sublattice of L because GLB & LUB of b & c is different in L & S.

~~This is kite & that is why L is not distributive~~

$R = \{0, a, e, d, 1\}$



this is sublattice of L & it is pentagon that is why L is not distributive

NOTE : To make *sublattice* take elements with their GLB and LUB. It is not compulsory to take maximum and minimum element in sublattice you can even ignore both.

Theorem : **If** lattice is distributive **then** there is at most 1 complement for every element. (this is one-way theorem)

Contrapositive is in lattice if any element has more than 1 complement then it is not distributive.

In short, to check if given lattice is distributive ;

(i) If |L|<=4 then it is distributive
(ii) If |L|=5 then distributive iff not kite or pentagon
(iii) If L is total order then always distributive because we cannot create kite or pentagon.
(iv) Find complement of elements and if some element has > 1 complement then not distributive. But if every element has <= 1 complement then find kite or pentagon sublattice.

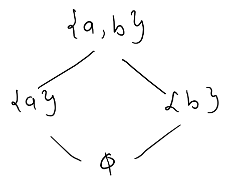
NOTE : If every element has exactly one complement then it is complemented (which is obvious) but it may not be distributive (because theorem was if distributive then at most 1 complement)

4) Boolean Lattice (or Boolean algebra) :

A lattice is Boolean if it is bounded, complemented and distributed lattice. Or if it is complemented and distributive. Because every complemented lattice is also bounded lattice.

Boolean lattice is complemented distributive lattice

Example, A = {a, b} then $(P(A), \leq)$ this is Boolean algebra or lattice.



<= 4 so distributive

← square structure

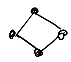
Q : Can single element lattice be Boolean algebra ? – Single element means it has greatest and least element so it is bounded, next it has less than 5 elements so it is distributive and it is complemented as complement of single element is that element itself.

Boolean algebra with :

1 element = A = φ

2 elements =  A = {1}

3 elements = Not possible

4 elements =  A = {1, 2}

5 elements = DNE

6 elements = DNE

NOTE :

1) **Every Boolean algebra has the same structure as $(P(A), \subseteq)$ structure. Because then only we can have unique complement.**
2) **Above sentence also implies that every Boolean algebra have 2^n elements. Remember this is if Boolean then 2^n elements. Reverse is not true because chain containing 2^n element is not Boolean.**

Boolean lattice satisfies all properties such as ICAA, identity, distributive, complement, absorption, etc. and also propositional logic, digital logic with 2 elements also satisfies all these properties that is why all are called Boolean algebra.

Q : What are the important elements of Boolean algebra $(P(S), \subseteq)$ and $(P(S), \supseteq)$? –

LUB = Union
GLB = Intersection
Greatest = S
Least = φ

← $(P(S), \subseteq)$

$(P(S), \supseteq)$

LUB = Intersection
GLB = Union
Greatest = φ
Least = S

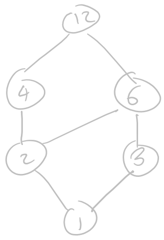
3.4.3) Analysis on Divisibility relation :

Now, we are going to see one special lattice also known as *divisibility lattice* which is denoted by Dn = set of all divisors of n. if n = 4, D_4 = {1, 2, 4} similarly if n = 6, D_6 = {1, 2, 3, 6}.

$(D_4, |)$ $D_4 = \{1, 2, 4\}$

$(D_{12}, |)$, $D_{12} = \{1, 2, 3, 4, 6, 12\}$

lattice ✓
bounded ✓
complemented ✗
distributive ✓



lattice ✓
bounded ✓
complemented ✓
distributive ✓

In all hasse diagram of divisibility lattice, one thing you can notice i.e. LUB = LCM and GLB = GCD

Q : Is $(D_n, |)$ distributive and complemented ? - All divisibility lattice is distributive lattice because LCM and GCD distribute over each other. And talking about complemented lattice so we know that some divisibility lattice contains chain lattice and we know that chain containing number of elements greater than 2 will not be complemented lattice.

Q : Which divisibility lattice is complemented ? – if n (in D_n) is prime square free number then divisibility lattice is complemented. Prime square free number is a number n such that p^2 | n should not happen, where p is prime. Let's say n = 8 then 8 = 2^2 x 2. 2 is prime and prime square is present so $(D_8, |)$ is not complemented. n = 6 then 6 = 2 x 3, No prime square is present thus $(D_6, |)$ is complemented lattice.

And some interesting fact that if n is prime square free number then complement of any element "a" is n/a.

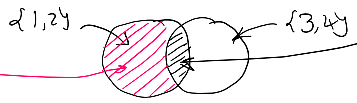
So, in short, we can say that $(D_n, |)$ is complemented iff prime factorization of n do not contains repetitive prime number. And if you observe this same condition applies to Boolean lattice or Boolean algebra.

Still two topics are remaining 1. Refinement of a partition and closure of relations which will do it after completion of theory of computation subject.

Silly Mistakes :

(1, 2) 3 4 5 6 7 = B_6

Let A = {1, 2, 3, 4, 5, 6, 7} then how many equivalence relations R on A are possible in which (1, 2) ∈ R but (3, 4) ∉ R ? -



All equiv. Relation which include (1,2) & (3,4)

(1, 2) (3, 4) 5 6 7 = B_5

Answer = $B_6 - B_5$

Q #23 | Numerical Type | Award: 1 | Penalty: 0 | Combinatory

The number of partitions of the set $X = \{a, b, c, d\}$ with "a" and "b" in the same block is ______

Your Answer: 4 | Correct Answer: 5 | Incorrect | Discuss

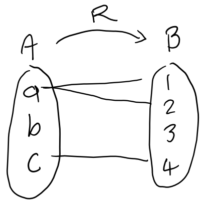
{a, b}{c}{d} is also partition.

4. FUNCTION

// Lecture 1

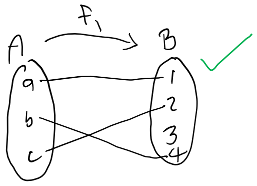
Function is type of relation. First consider a relation from set A to set B, $R: A \to B$

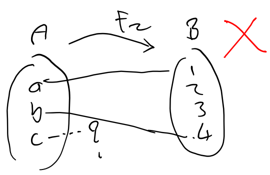
We know that relation $R \subseteq A \times B$



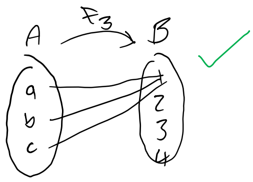
Q: $x \in A$ then $\underbrace{|\{y \mid xRy,\ y \in B\}|}_{\text{set of all those elements of B to which } x \text{ is Related.}} = ? = 0 \text{ to } 4$

A binary relation R from A to B is said to be a *function* if **for every** element a in A, there is a unique element b in B so that (a, b) is in R. For a function R from A to B, instead of writing (a, b) ∈ R, we also use the notation R(a) = b, where b is called **image** of a or a is **preimage** of b. The set A is called the **domain** of the function R, and the set B is called **range** or **codomain** of the function R.





Relation but not function



4.1) Introduction :

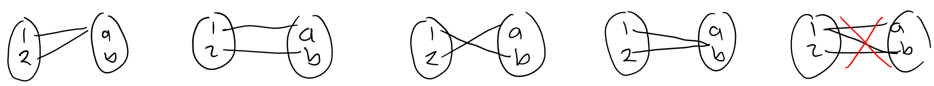
Function requirement :

- No element of the domain must be left unmapped.
- No element of the domain may map to more than one element of the co-domain.

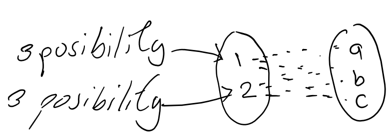
// Lecture 2

4.1.1) Number of functions :

Q : how many functions are possible for set A = {1, 2} to B = {a, b} ? – No. of function is 4.



Consider same question but from 2 elements set to 3 elements set.



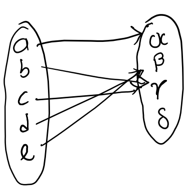
$3 \times 3 = 9$ # of functions

Similarly from m to n we would have n^m functions.

Therefore, **number of functions = |codomain|$^{|\text{domain}|}$**

// Lecture 3

4.1.2) Representation of functions :



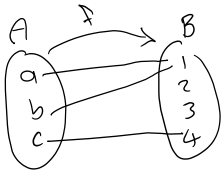
	α	β	γ	δ
a	✓			
b			✓	
c			✓	
d		✓		
e		✓		

	R
a	α
b	γ
c	γ
d	β
e	β

As we know that function is relation so we can also have set representation of function.

$R = \{(a,\alpha),(b,\gamma),(c,\gamma),(d,\beta),(e,\beta)\}$ ← Not popular

Image of a subset of domain :



$f(a) = 1$

$f(\{a,b\}) = \{1\}$

$f(\{a,c\}) = \{1,4\}$

$f(\text{domain}) = \{\text{Range}\}$

Some terminology used in function :

Real valued function : function whose co-domain = R

Integer valued function : Function whose co-domain = Z

Natural valued function : Function whose co-domain = N

Boolean valued function : Function whose co-domain = {0, 1}

Two real-valued function or two integer valued function with the same domain can be added, as well as multiplied.

$f: \{1,2,3\} \to R \qquad f(x) = x^2$

$g: \{1,2,3\} \to R \qquad g(x) = x+1$

$f+g: \{1,2,3\} \to R \qquad (f+g)(x) = x^2+x+1$

& similarly $fg: \{1,2,3\} \to R$; $(fg)(x) = x^2(x+1)$

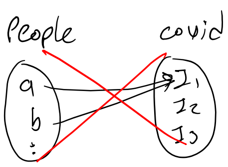
// Lecture 4

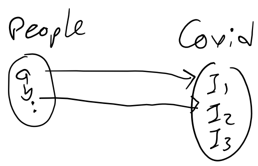
4.1.3) Types of functions :

There are mainly three types of function : One-one function (Injective), Onto function(surjective), Bijection (Bijective)

1) **One-one function** : This is also called injective or injection why ? let's see

If I have function from people to injection then no two people can have dose from same injection, they must have used different injection.



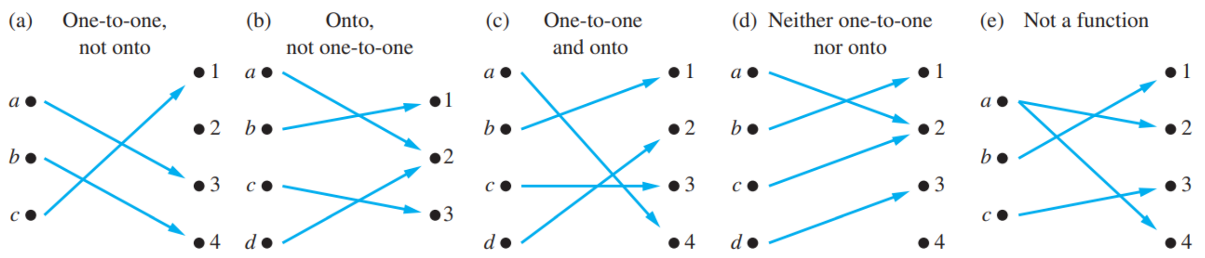


In one-one function, every element of domain is mapped to different distinct element of co-domain. In other word no element of co-domain is related by more than one element from domain.

If f(a) = f(b) then a = b this condition should satisfy for every one-one function.

2) **Onto function** :

In onto function, every element of co-domain has at least one preimage.



3) **Bijective** :

A function that associates each element of the codomain with a unique element of the domain is called *bijection*. Or <u>bijection is a function that is both injective and surjective</u>.

<u>In bijection, if $R : A \rightarrow B$ then $|A| = |B|$ this result is the consequences of below statement.</u>

<u>And if |A| > |B| then one-one function is not possible</u>

<u>If |A|<|B| then onto function is not possible</u>. (prove this in book)

NOTE :

1) **If X^2 = 25 then X = ? – same as asking which value of X satisfies X^2 = 25, X = +5, -5 and if X = sqrt(25) then X = 5 only not X = -5 (because codomain of sqrt function is nonnegative). Both are different operation.**
2) **Relation is subset of A X B which is nothing but set so, every relation is special type of set. Thus, we can apply set operation like union, intersection, set difference, etc on relation to get new relation.**

//Lecture 7

4.2) Composition and inverse of function :

<u>Composition is relation operation. It is operation that creates new relation transitively</u>.

Example,

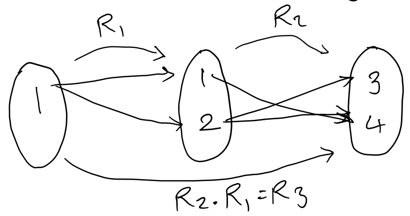
$R_1 = \{(1,1),(1,2)\}$

$R_2 = \{(2,3),(2,4),(1,4)\}$

$R_2 . R_1 = \{(1,3),(1,4)\}$ ← composition

$1 \rightarrow 2 \rightarrow 3$

$1 \rightarrow 2 \rightarrow 4$



why $R_2.R_1$ & not $R_1.R_2$?

$R_2.R_1 = f_{R_2}(f_{R_1})$

output of R_1 becomes input of R_2

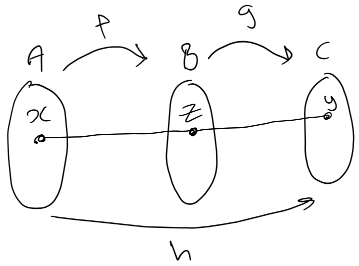
R2.R1 means first you apply R1 then you apply R2.

Composition of relations is associative. i.e. A.(B.C) = (A.B).C

Now, let's take our attention back to function. We do same composition on function as function is nothing but special type of relation only.

Consider function f, g such that $f: A \rightarrow B$ and $g: B \rightarrow C$

And composition function $h: A \rightarrow C$ or $h = gof$

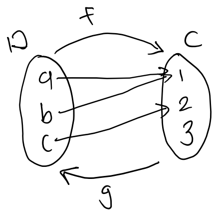


$h(x)=y$ iff $f(x)=z$ & $g(z)=y$

Q : What is the necessary condition for $f: A \to B$ and $g: M \to N$ to defined gof ? – Codomain of f should be domain of g. In other words, B = M.

4.2.1) Inverse of function :

First consider a function $f: D \to C$



f is function from D to c & it is not one-one and that's why function g is not function. Because 1 have two images.

Similarly, if F is not onto function then reverse of f function is not even a function. So, question is

Q : When will function be a function even in reverse direction ? – It should be bijection (from above two observation). And we call this reverse direction function as inverse of function.

$f: R \to [0, \infty)$ $\quad f(x) = x^2 \leftarrow$ first check is it a function?

(i) One-one $\to$ X $\quad$ -1 & 1 both have $f(x)=1$

(ii) Onto $\to$ ✓ $\quad$ every codomain has some Image

(iii) Invertible $\to$ X $\quad$ because it is not bijection

SILLY MISTAKES :

Let ◊ ⊆ X × X be an arbitrary binary relation over an arbitrary set X. Which of the following statements is/are true?

(a) ◊ o ◊$^{-1}$ = ◊$^{-1}$ o ◊

(b) ◊ ∪ ◊$^{-1}$ is symmetric

(c) If ◊ is transitive then ◊$^{-1}$ is transitive.

(d) If ◊ is antisymmetric then ◊$^{-1}$ is antisymmetric.

$◊ : x \to y$, $y \to z$ $\quad ◊^{-1} : y \to x$, $z \to y$

$◊ \circ ◊^{-1} = \{(x,x),(y,y)\}$ $\neq$ $◊^{-1} \circ ◊ = \{(y,y),(z,z)\}$