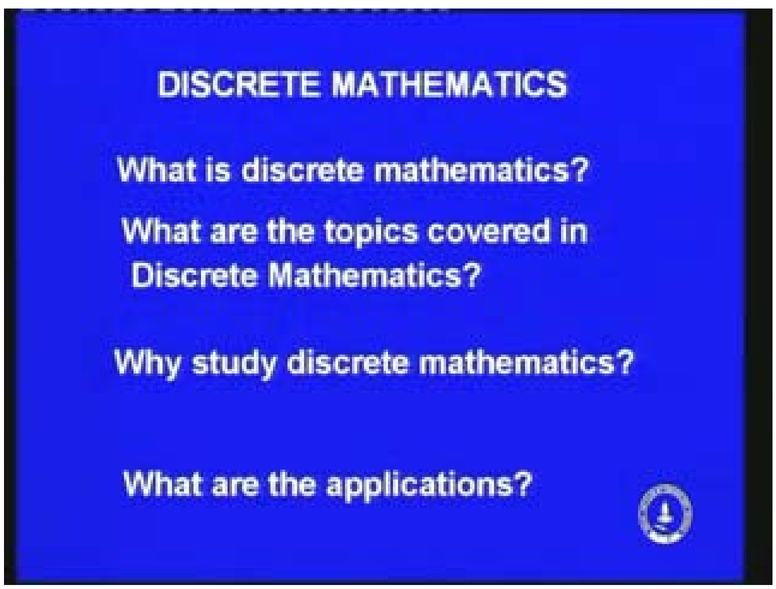
Discrete Mathematical Structures
Dr. Kamala Krithivasan
Department of Computer Science and Engineering
Indian Institute of Technology, Madras
Lecture - 1
Propositional Logic

This course is about discrete mathematical structures. It is a theoretical course intended for B. Tech or B. E. students in computer science in the second semester. What are the things we are going to study in this subject? First of all what is discrete mathematics and what are the topics to be covered here?
Discrete Mathematics is a study of discrete structures which are abstract mathematical models dealing with discrete objects and their relationship between them. The discrete objects could be like: sets, permutations, graphs, FSA, etc. We will see in a minute, what are the topics to be covered here? And why do we want to study Discrete Mathematics and why should one study this subject?

(Refer Slide Time: 05:02)



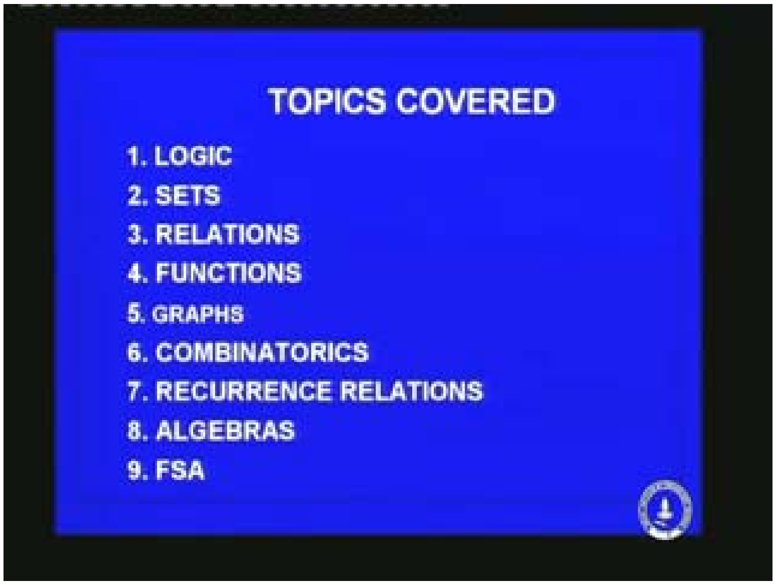
The reason is, in many fields of computer science, basic mathematical concepts will be used and in any books on them when such a concept is introduced they will not explain it in detail but briefly mention a few lines about it. But for a student to understand the subject completely it may not be enough. At the same time, one need not go in-depth in this subject also. One need not learn this subject as a mathematical student could learn Algebra or analysis.

You want to know something about each topic to a decent depth and that will help to understand the other subjects in computer science in a better way. The aim of this course is not only make people learn about these topics, but also help them to develop the habit of thinking mathematically. The student should develop to think in a mathematical manner. That is the aim of this course.

What are the applications or what are the fields in which this these concepts will be used? They are plenty; in fact in each and every subject in Computer Science some where or the other these concepts will be used. For example, when you want to study programming, after writing a program you would like to check that if it is correct and you want to prove it, so proving programs correct is a topic which will be studied in this course which will help you to write correct programs. And in artificial intelligence lot of ideas about Logic will be used.

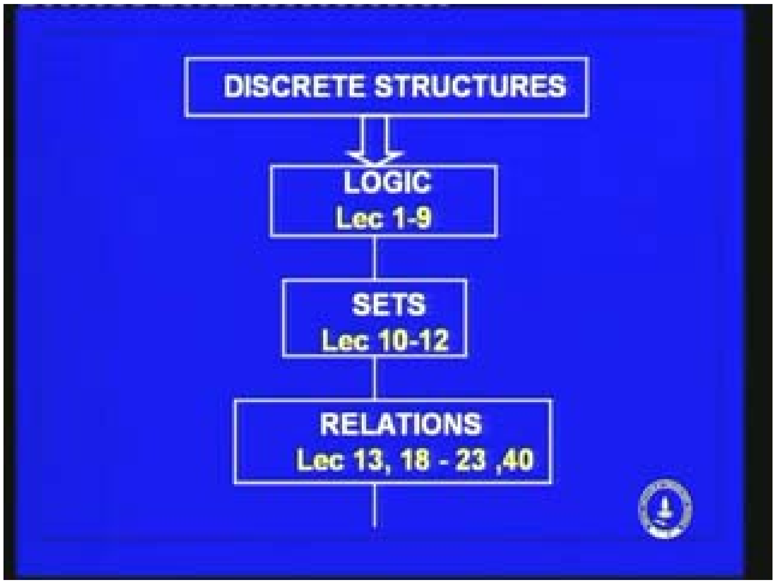
Prolog is a Logic programming language where the resolution principle is used and we will learn about these in this subject. And in computer networks lot of concepts about graph theory and final state automata are used, so that will be very useful there. In fact you can find that in any field you will have a few concepts from discrete mathematics which will help you to understand the subject in a better manner. In compilers you learnt final state automata concepts will be used, cramer concepts will be used, hash functions should be used and these are studied in this subject. And now what are the topics which will be covered in this subject?

(Refer Slide Time: 05:25)



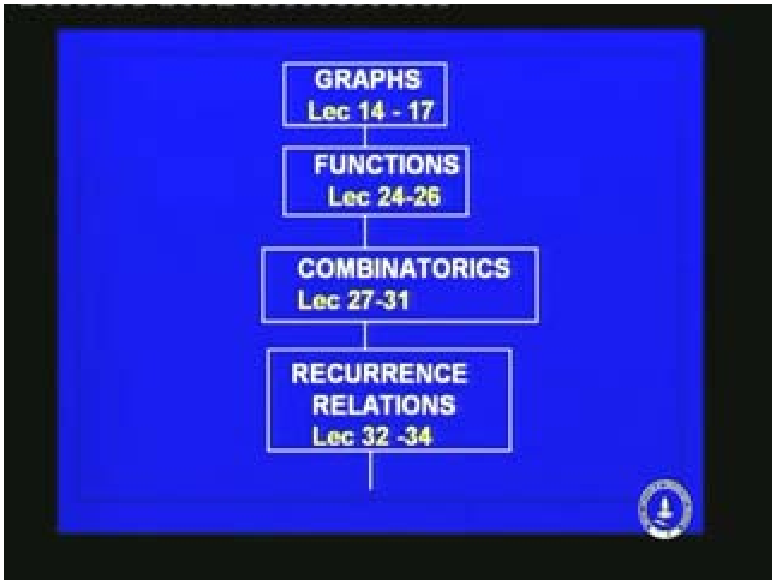
We will be studying about Logic, sets, relations, functions, graphs, combinatorics, recurrence relations, algebras and FSA. The layout of the subject will be like this.

(Refer Slide Time: 06:15)



First nine lectures will be devoted to Logic and its applications. Then three lectures will be on sets. Actually this subject does not pre-suppose any prior knowledge from the student except a reasonable mathematical maturity expected of a high school student. That is the only thing we assume and proceed with this subject, In fact after learning this subject the student should be able to think more mathematically and also should have acquired a basic knowledge about these topics. Then we go on to relations. In relations, first we have lecture 13, then afterwards we shift to graphs from lecture 14-17, then again come back to more properties of relations from 18-23.

(Refer Slide Time: 07:15)

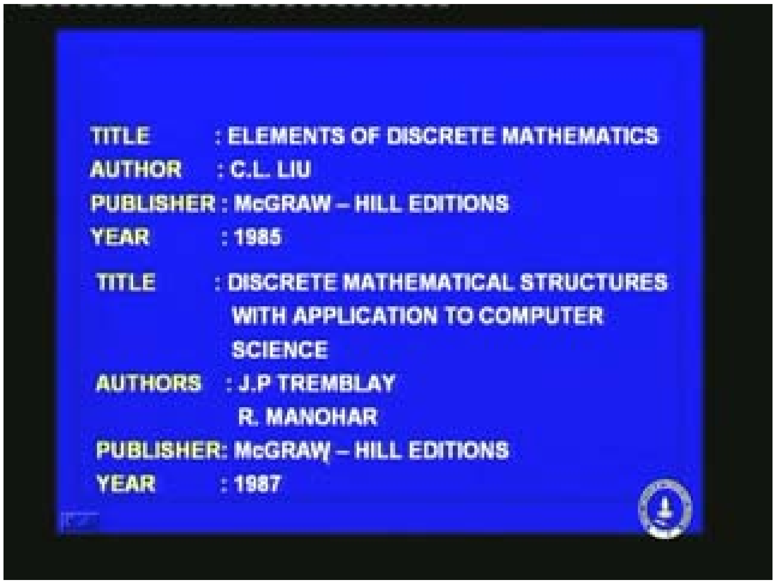


Then the last lecture will be about lattices which are also about partially ordered sets. We will learn certain basic facts about graphs in lecture 14-17, then we will learn about functions in lecture 24-26, then a few lectures on combinatorics where we will be studying about pigeon hole principle, permutation and combination, and generating functions. Recurrence relation is another topic which is of importance especially in algorithmic analysis.

(Refer Slide Time: 07:18)

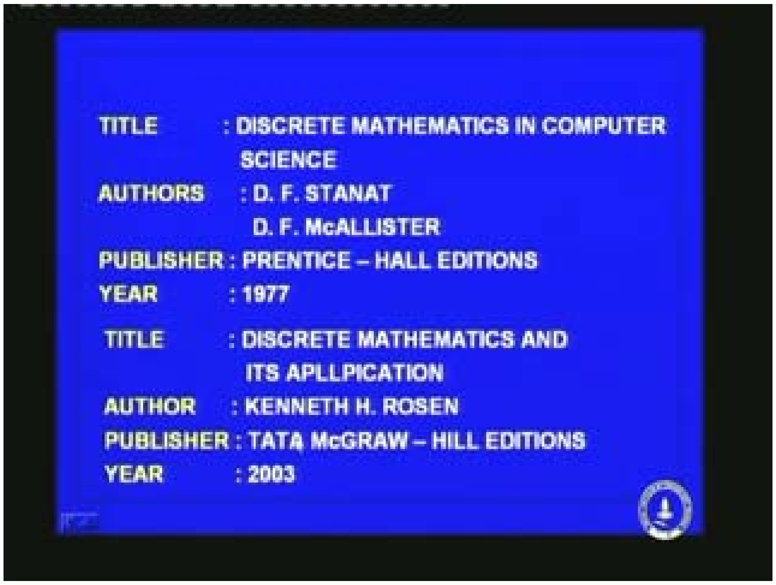
We spend three lectures on them, and then we will study a little bit about algebras where we study about groups, semi groups, rings, etc and about finite state automata. We have two lectures where we study only the basic principles of final state automata.

(Refer Slide Time: 08:00)



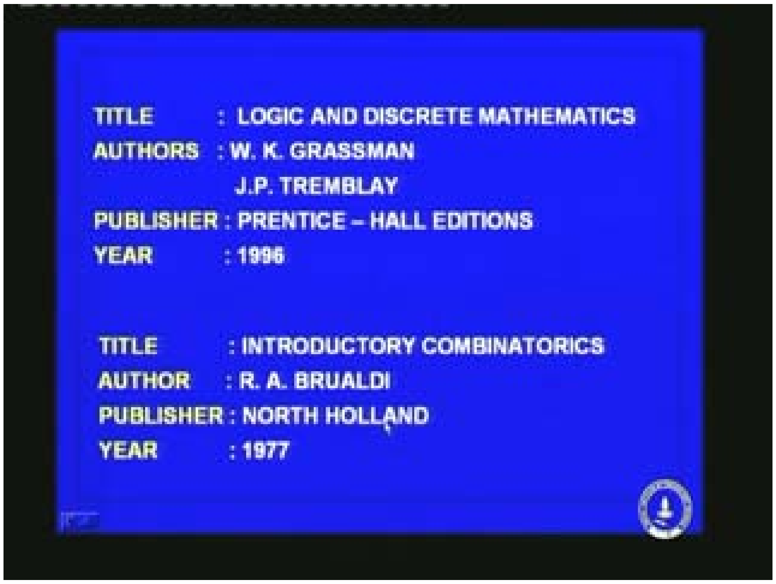
We shall give some reference books for the subject. Some of them are new and some of them are old. But, there have been recent editions of them, Elements of Discrete Mathematics by C. L. Liu and the second one is Discrete Mathematical Structures with Applications to Computer Science by Tremblay and Manohar. These are standard books.

(Refer Slide Time: 08:00)



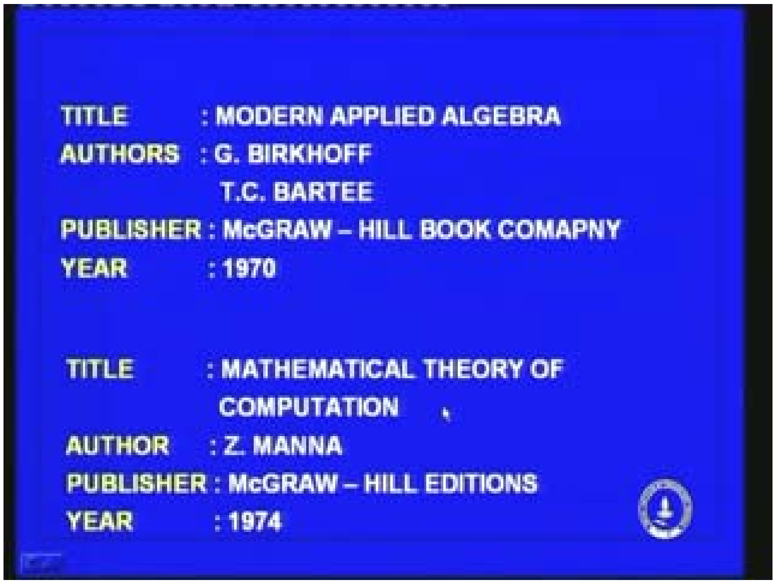
Discrete Mathematics in Computer Science by Stanat and McAllister is another book. Discrete Mathematics and its Application by Rosen,

(Refer Slide Time: 08:33)

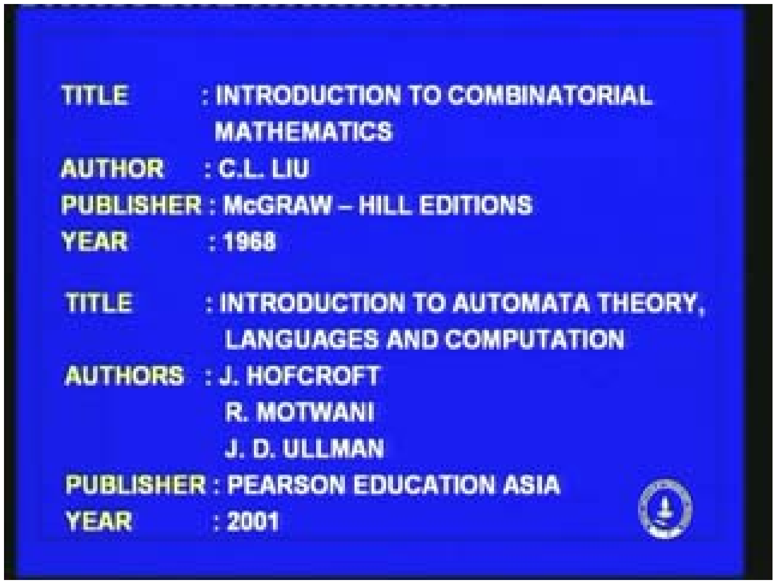


Logic and Discrete Mathematics by Grassman and Tremblay has some concepts about prolog and application of Logic to that. Introductory Combinatorics by Brualdi is a book that can be used to study some topics of Combinatory. Modern Applied Algebra by Birkhoff and Bartee is an old book but it has got some good concepts about Lattices and Final State Automata, Mathematical Theory of Computation by Manna is another book which will be useful for proving programs correct.

(Refer Slide Time: 09:27)



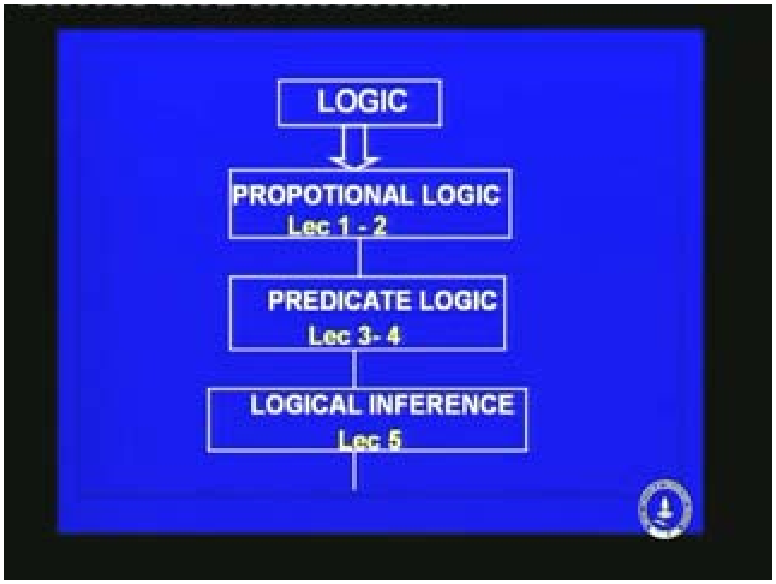
(Refer Slide Time: 09:32)



Introduction to Combinatorial Mathematics by Liu is an old book but still it has got some very good topics: permutation and combination, generating functions, recurrence relation etc. Then for Automata Theory this book can be used Introduction to Automata Theory, Languages and Computation by Hofcroft Motwani and Ullman are some of them among the few books. There are so many other books also.

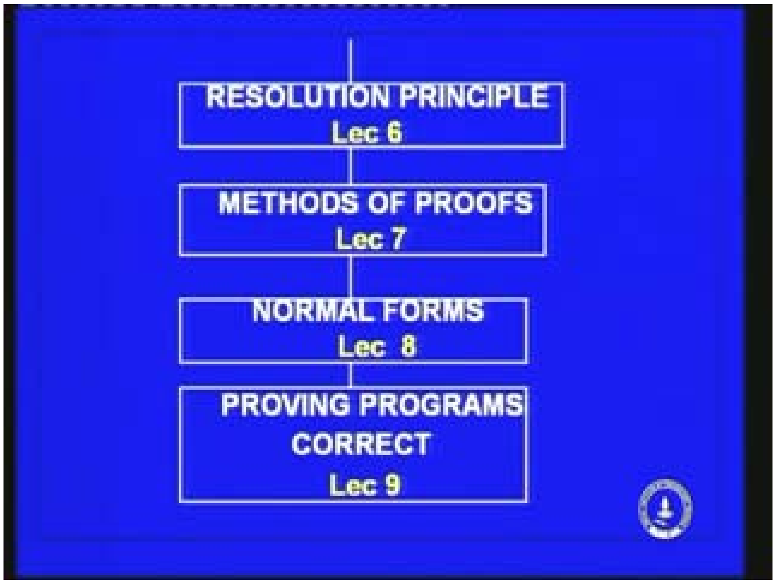
Any student studying this subject can use the prescribed book by the respective University or College. Now we shall go on to the first topic Logic. Logic is a very useful topic. Every student in Computer Science should learn this topic because it is applied in proving programs correct in data bases where you study about relational algebra and relational calculus and also in artificial intelligence about reasoning and things like that. So, first, coming to Logic, there will be 9 lectures on Logic. They are divided like this:

(Refer Slide Time: 10:33)



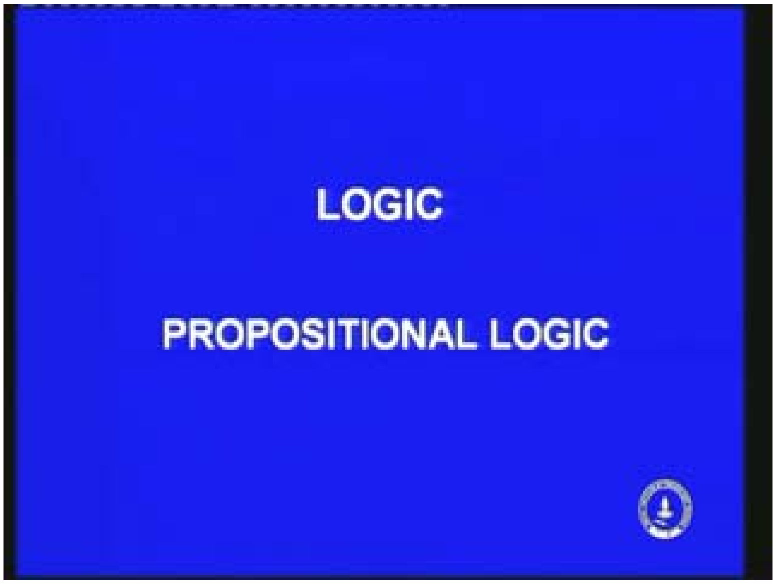
The first 2 lectures will be devoted to Propositional Logic, then we go on to Predicate Logic, then Logical Inference, then Resolution Principle, then Methods of Proofs, Normal Forms and 1 lecture on Proving Programs Correct.

(Refer Slide Time: 10:46)



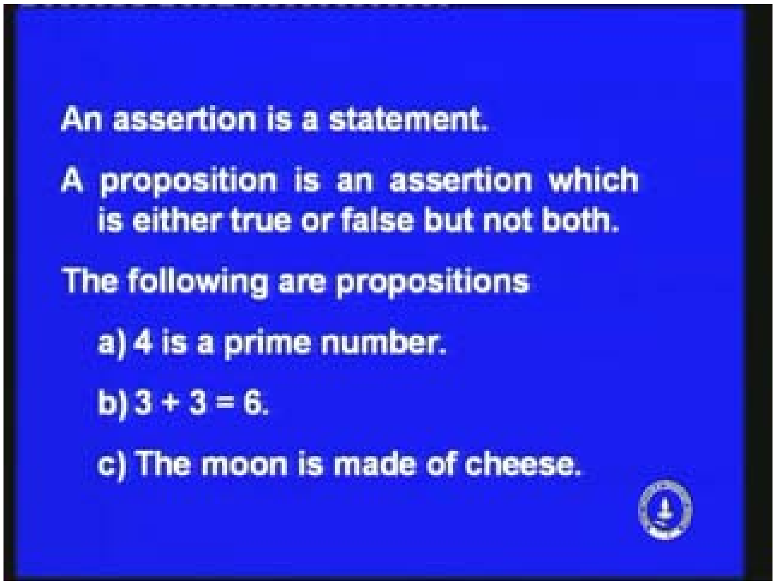
So let us start first with Logic and Propositional Logic.

(Refer Slide Time: 10:53)



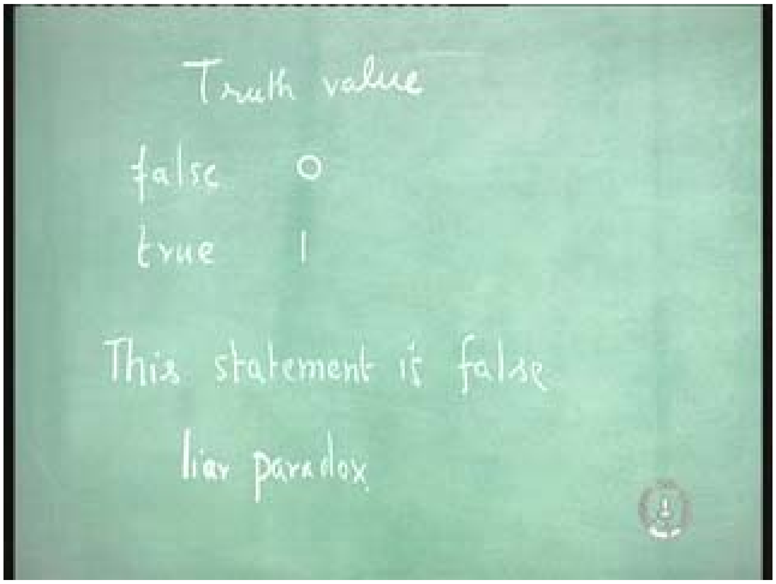
What is Propositional Logic? So for that first we have to start with what is a proposition. In English language we talk about sentences, we talk about assertions, we talk about orders, questions and so on. What is a proposition? First of all it should be an assertion. An assertion is a statement.

(Refer Slide Time: 13:04)



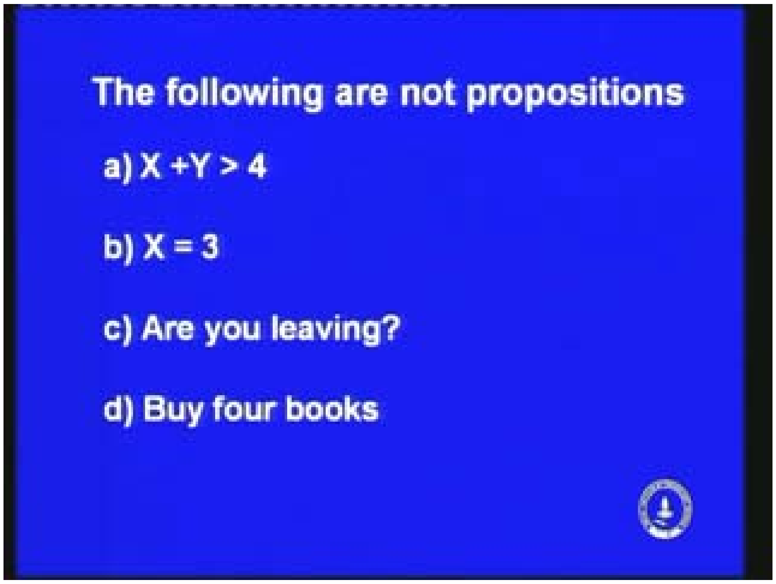
It is different from asking a question or giving an order, the proposition is an assertion which is either true or false but not both. So you must be able to associate a truth value with an assertion.

(Refer Slide Time: 16:08)



The truth value will be false or true. Usually you denote false by 0 and true by 1. So to any sentence or to any assertion to which you can associate a truth value is called a proposition. Consider the following examples: the following are propositions, 4 is a prime number, it is an assertion but it is a false statement. So the truth value associated with 4 is a prime number is false. Then take the next sentence, 3 plus 3 is equal to 6. Again, it is an assertion and it is a correct statement. So the truth value associated with that is true. The moon is made of green cheese or the moon is made of cheese. This is again an assertion and a wrong statement. So the truth value associated with that is false. Let us see what are not propositions.

(Refer Slide Time: 14:53)



Consider the statement X plus Y greater than 4, is it true or false? It depends on what value you are going to give for X and Y. If you give X the value 2, Y the value 2, then 2 plus 2 is greater that 4, is not correct. Or, if I give the value X is equal to 3, Y is equal to 4, then it takes the value 3 plus 4 is greater than 4, that is 7 is greater than 4 which is correct.

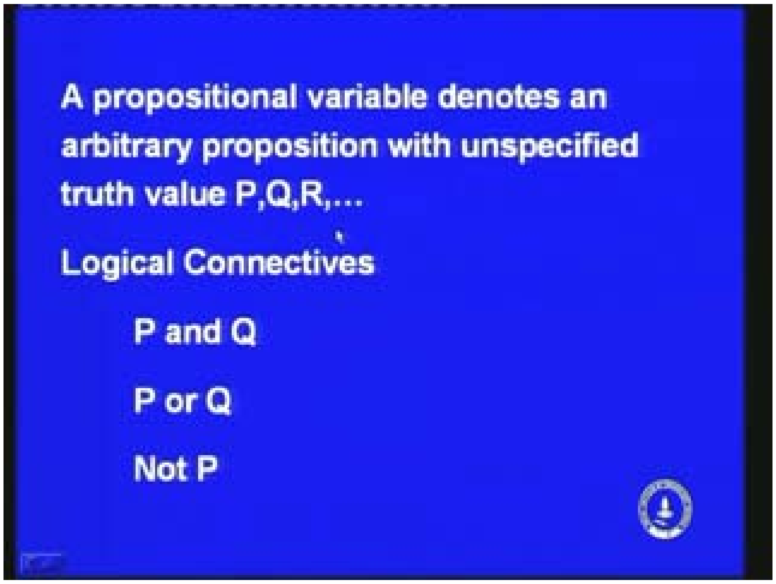
So, depending on the value you give for X and Y the statement takes a true or a false value. It does not have a unique value, it depends on the value you are going to give for the variables X and Y. X and Y are called individual variables. So you cannot associate a unique truth value to this. Next, take the sentence X is equal to 3. Here you find that you cannot associate a truth value to this. If you give the value 3 to X it will be true, if you give some other value to X, it will not be true. So again this is not a proposition.

Take the next one, are you leaving? This is not an assertion. A proposition should be an assertion. Are you leaving is a question and it is not an assertion. So it is not a proposition. Take the next one, buy 4 books, this again is not an assertion, it is an order, go and buy 3 books, this is an order and not an assertion so it cannot be a proposition. So whenever something is not an assertion, it is a question or an order it cannot be a proposition. But in special cases there can be assertions which are not propositions.

For example, consider this sentence. This statement is false. This is an assertion, is it a proposition? It is not a proposition because you cannot associate a truth value to this. If it is true, it is false. If it is false, it is true. So you cannot associate a true or a false value with this statement. This is called a paradox. This is actually called a liar paradox. And such a statement even though it is an assertion, it is not a proposition because you are not able to associate a truth value to that.

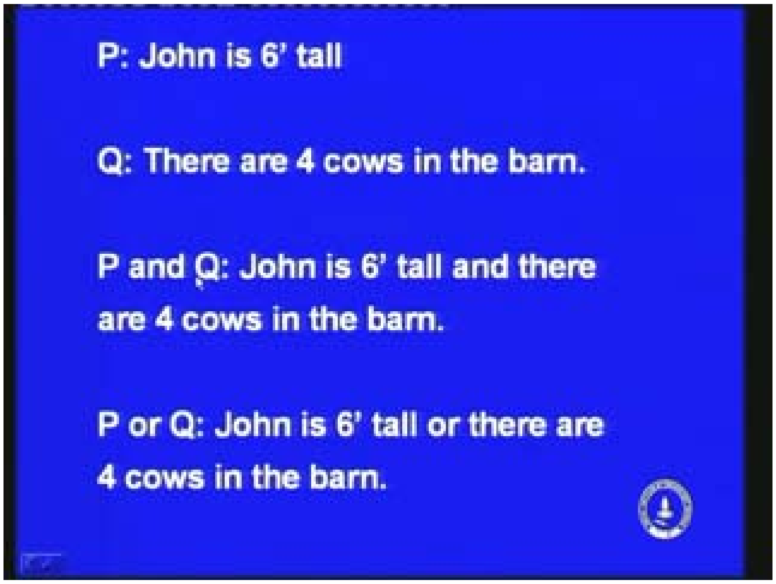
This has, you have individual variables X and Y to which we associate numbers. You can talk about proposition variables. A propositional variable denotes an arbitrary proposition with unspecified truth value like P, Q and R.

(Refer Slide Time: 17:23)



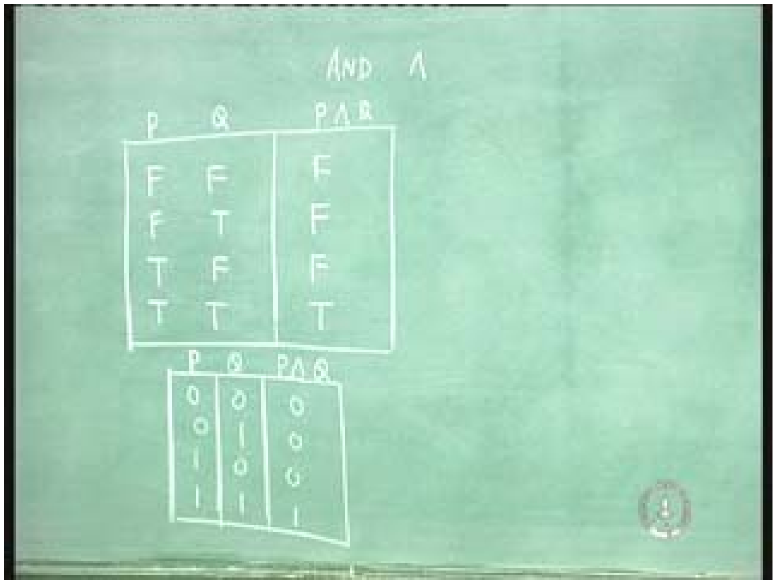
They are all propositional variables and you can assign a proposition to them. P is just a propositional variable, just as we assigned the value 4 to X, you can assign the value of a proposition to the variable P. Similarly, you can denote these as propositional variables and you can assign propositions as values to them. Then when you have propositional variables or actually propositions also, you can connect them with logical Connectives. There are several Connectives. First, we shall study three of them: and, or, and not.

(Refer Slide Time: 17:39)



Consider the proposition, John is 6 feet tall, and the proposition Q, there are 4 cows in the barn. Then the compound statement P and Q is John is 6 feet tall and there are 4 cows in the barn. When is this true? P and Q will be true when both P is true and Q is true. So we have a truth table for them. So and is denoted by this symbol and the truth table for that will be P Q or P and Q.

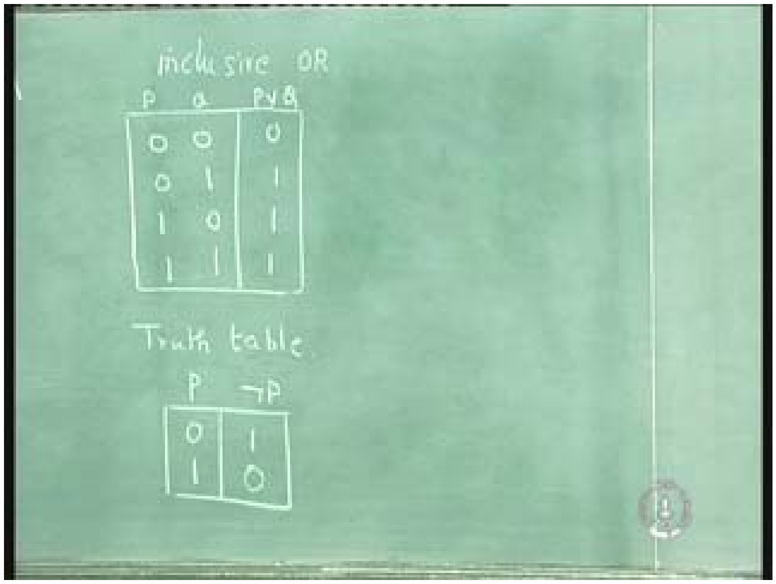
(Refer Slide Time: 19:29)



The possibilities are, both can be false or P can be false and Q can be true. This can be true and this can be false and both can be true, these are the possibilities. And when both are false the compound statement will be false and when one is false also the compound statement will be false but when both are true the compound statement will be true. Usually we denote false by 0 and true by 1. So the truth table will be like this, P Q, P and Q 0 0, 0 1, 1 0, 1 1. And in these three cases it is 0 and 1 in this case. So the compound statement P and Q will be true only when both P is true and Q is true. In all the other cases it will be false.

What about the Connectives or P or Q? If P denotes this sentence and Q denotes this sentence P or Q will denote the sentence John is 6 feet tall or there are 4 cows in the barn.

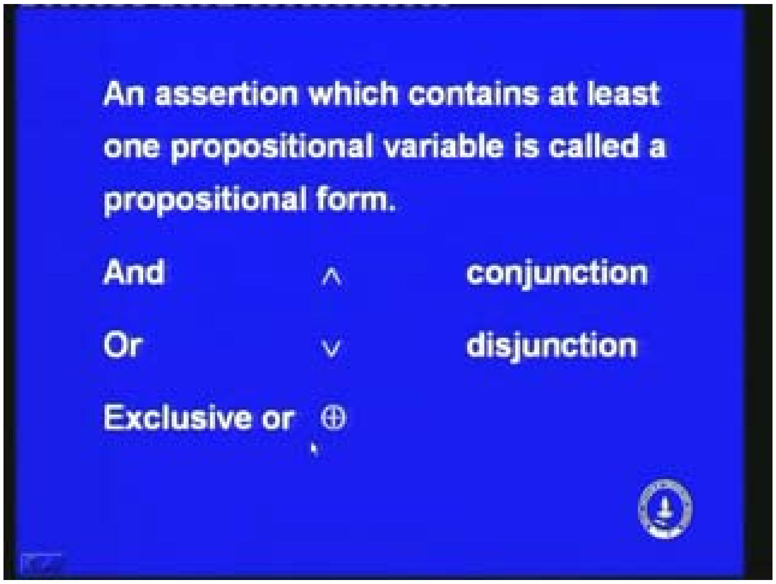
(Refer Slide Time: 20:15)



When can you say that this is true? When one of them is true the compound statement will be true or when both of them are true also the compound statement will be true. This is called inclusive OR and the truth table for that will be like this.

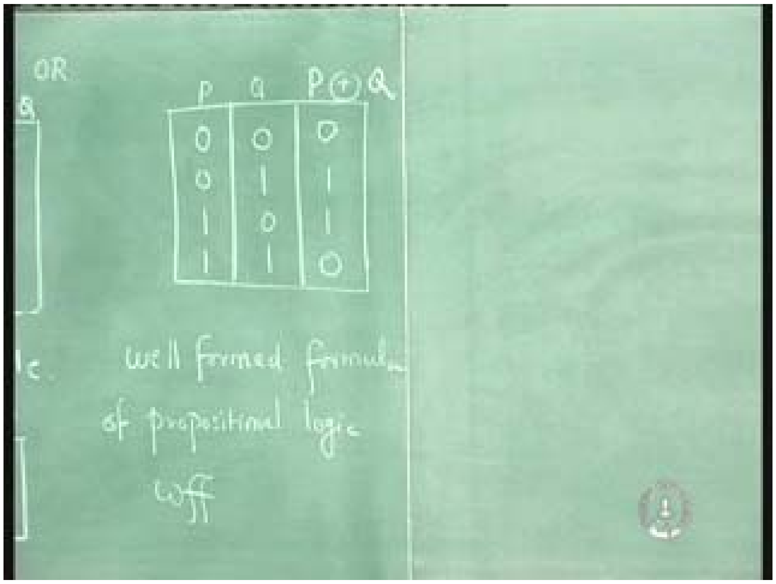
P Q, P or Q, 0 0, 0 1, 1 0, 1 1, 0 denotes false and 1 denotes true and when both of them are false, this is false. When one of them or both of them are true, this will be true. This is the truth table. This is called a truth table for OR. And for the unary operator NOT P and NOT P, when P is false NOT P will be true and when P is true NOT P will be false. So, what is the statement NOT P? John is not 6 feet tall. That will be the statement for NOT P. So, we are studying the operators And, OR, and NOT. There is one more operator called the Exclusive or.

(Refer Slide Time: 21:58)



An assertion which contains at least one propositional variable is called a propositional form. So when you connect two propositional variables by And or OR it is a propositional form.

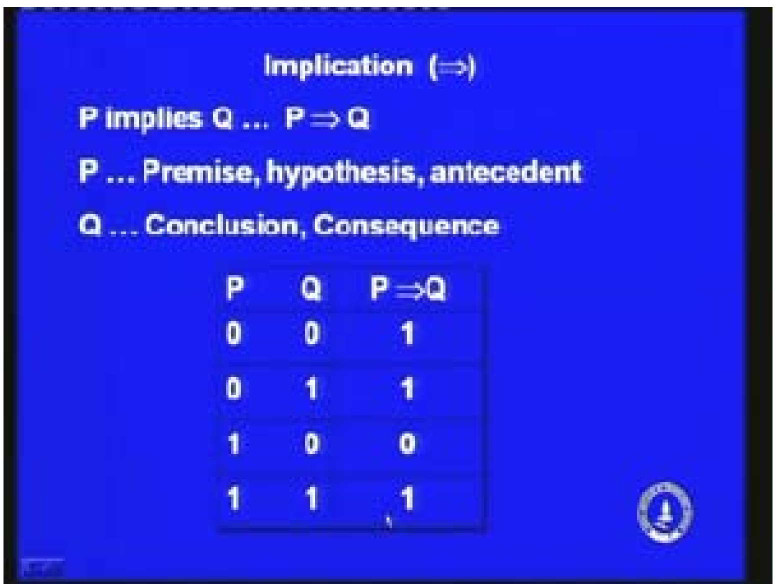
(Refer Slide Time: 24:31)



You can also use the Exclusive or operator. Exclusive or means the truth table for that will be like this -> 0 1, 1 0, 1 1. In the Exclusive or case it is true only when one of them is true and the other is false. If both of them are true or both of them are false, it is false. So when you say something like that, John is 6 feet tall, there are 4 curves in the bond. Both statements can be true and the compound statement can be true. But in some cases like say Sudha is wearing a saree and Sudha is wearing a blue chudidhar, and if I say Sudha is wearing a saree or Sudha is wearing a blue chudidhar both of them cannot be true at the same time. Only one can be true, the other will be false. So in such cases you use Exclusive or.

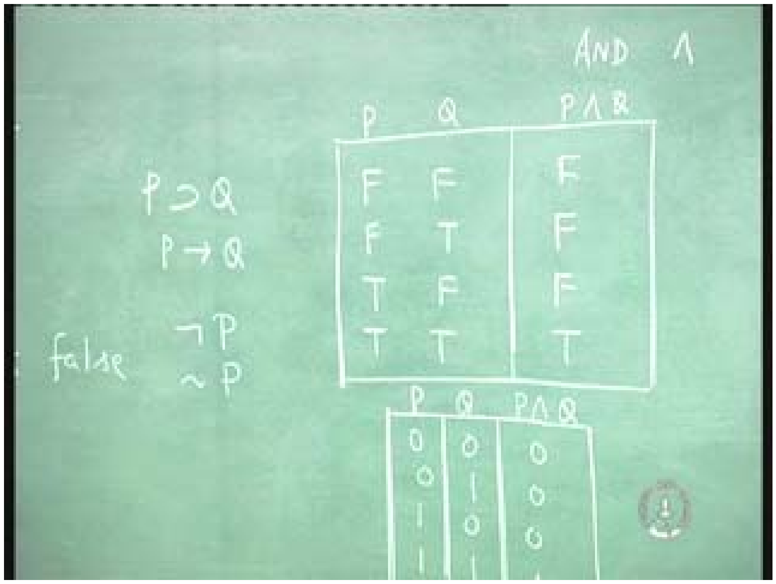
Usually when you say either this or that it means Exclusive or. When you just say or it means inclusive or. But you have to be very careful because sometimes they are used in a slightly ambiguous way. So when you want to transform an English sentence into logical notation or transform some logical formula into English sentence you have to be very careful whether you are using inclusive or, Exclusive or. Usually Exclusive or means you must say either or and for inclusive or you just say or without that either. And any propositional form which is connecting variables, it is called a well formed formula of Propositional Logic. Well formed formula is denoted by wff usually called as wff. So P and Q is a well formed formula. P or Q is a well formed formula, P Exclusive or Q is a well formed formula. So P Exclusive or will be denoted like P Exclusive or.

(Refer Slide Time: 26:30)



Apart from these operators, there are two more operators which are used in Logic. One is called the implication and another is called equivalence. What is implication? Let us look at this. P implies Q, it is denoted like P implies Q. In some books, it may be written like this and in some books they rarely use this sort of an arrow for P implies Q. So when you take a book you must be clear to find what sort of a notation they are using.

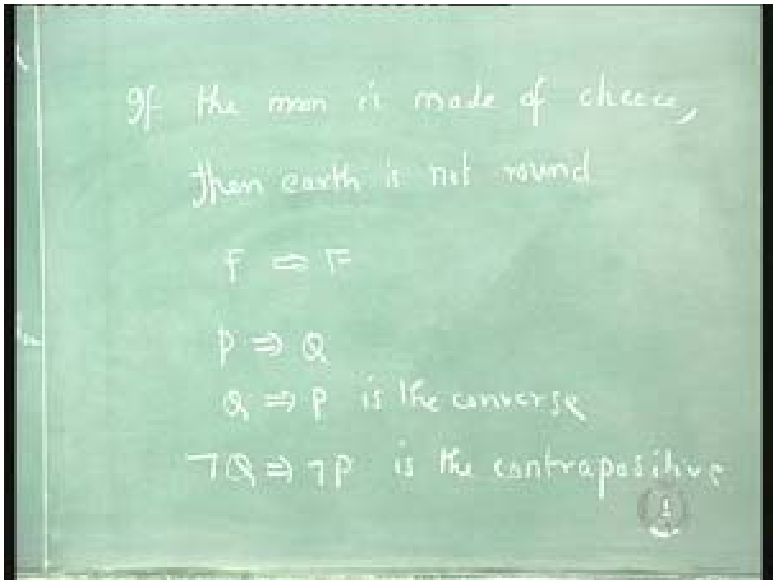
(Refer Slide Time: 25:45)



Similarly, NOT P is usually denoted like this. Sometimes it is also denoted like this. So when you see a book you must be careful to find what notation they are using.

So in this case, P is called the premise, hypothesis or antecedent and Q is called the conclusion or consequence. And what is the truth table for P implies Q? For P implies Q you have: 0 0, 0 1, 1 0, 1 1, then when both are false, it is true. When P is false and Q is true, it is true. When P is true and Q is false, it is false. When both of them are true, it is true. There is a slight problem here, it is not a problem but you must feel convinced about this truth table. Usually when we say Logic or what we study is Greek Logic which was formulated by Aristotle Socrates and Greek philosophers, in such Logic this is the truth table for implication.

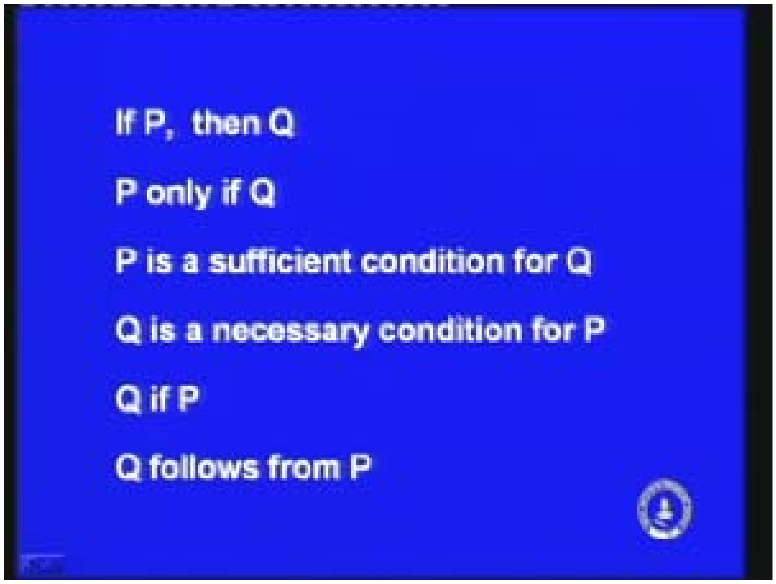
(Refer Slide Time: 31:45)



The antecedent and the consequent need not be related at all, they may be quiet unrelated statements like the moon is made of cheese, then the earth is not round, it could be something like that. There is no relation between the antecedent and the consequence. But the compound statement is true because this is of the form false implies false. And for that in the first line of the truth table tells you if P is 0 and Q is 0, P implies Q will be true. Greek Logic allows that. Usually in Indian Logic you look at Aryabhatta or Bascrads explanation on Aryabhatta's work, such things are not allowed.

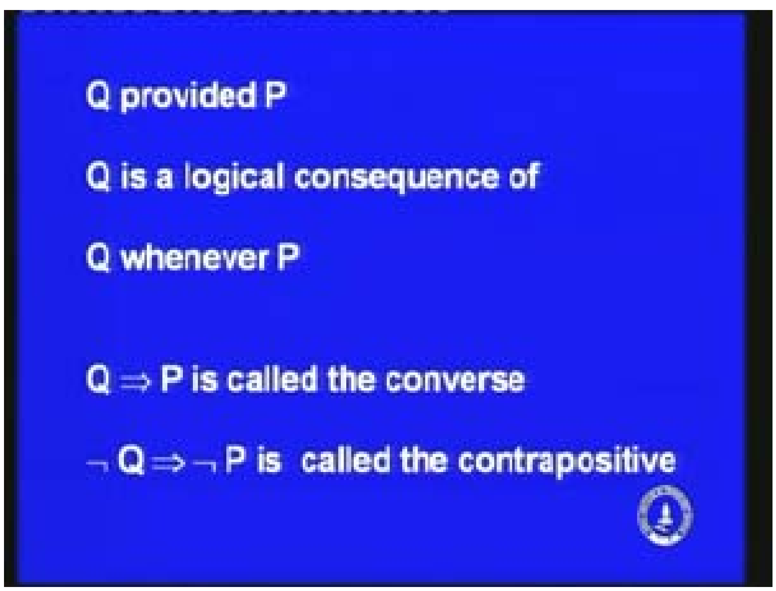
Usually antecedent and the consequence will be related. For example, something like this, you can very easily see, if I fall into the lake I will get wet. Obviously the premise and the conclusion or the consequence is related and you can see the correspondence between them. But generally in the truth table you must also give a value at the compound statement even though the premise and antecedent are not related. This, you must remember very well. So the implication is true when the premise or the antecedent is false or the consequence is true. It is false only in the case when premise is true and the conclusion is false or the antecedent is true and the consequence is false. Only in that case it will be false. In the other three cases it is taken to be true. This can be read in several ways.

(Refer Slide Time: 29:41)



You can read it as; if P then Q, P only if Q, P is a sufficient condition for Q, Q is a necessary condition for P, Q if P, Q follows from P, Q provided P, Q is a logical consequence of P, Q whenever P. These are the different ways of saying this. You would have studied about sufficient conditions and necessary conditions in High school level and in many theorems where the statement will be in the form if then.

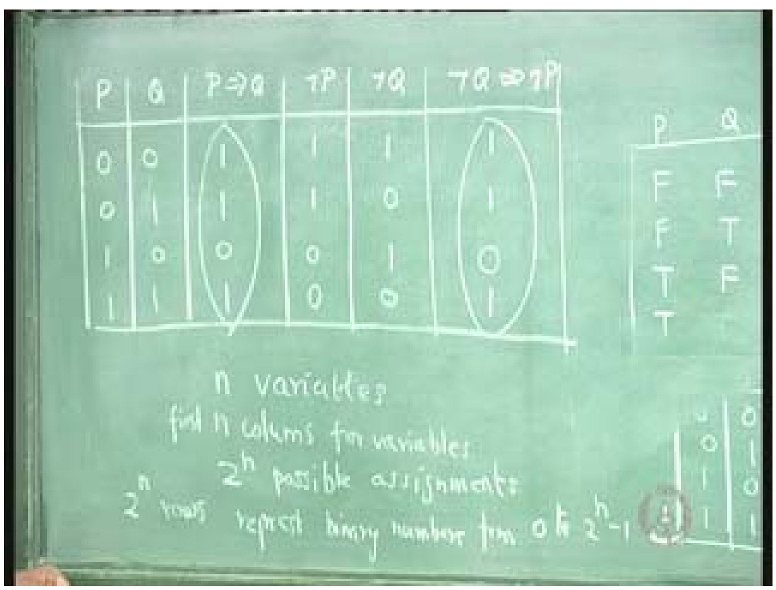
(Refer Slide Time: 29:47)



For example, you would have studied the Pythagoras theorem. If a, b, c is a triangle right angled at b, then ac square is equal to ab square plus bc square So the statement of the theorem is of the form if something, then something. And sometimes you talk about the converse of a theorem. Sometimes the theorem will be proved in a slightly different manner. When we study methods of proof we will get into that in detail. But generally, the converse of a theorem is denoted as Q implies P. P implies Q is a statement and the converse is Q implies P and NOT Q implies NOT P is called the contra positive. So you have this P implies Q, Q implies P is the converse, NOT Q implies NOT P is the contra positive. So suppose Pythagoras theorem is, if a, b, c is a triangle right angled at b then ac square is ab square plus bc square.

What will be the converse of that theorem? Converse also you would have studied in school. If in a triangle ac square is ab square plus bc square, then the triangle is right angled at b. So that is a converse. In this case it so happens that both the theorem and converse are true. But several situations may arise where the theorem may be true but the converse need not be true, because P implies Q it does not mean Q implies P is true. Whereas the contra positive NOT Q implies NOt P will be true whenever P implies Q is true. Let us draw the truth table and see this. So you have P, you have P Q, P implies Q NOT P NOT Q, NOT Q implies NOT P. So consider this table. You have the possibilities 0 0,, 0 1, 1 0 and 1 1.

(Refer Slide Time: 36:55)



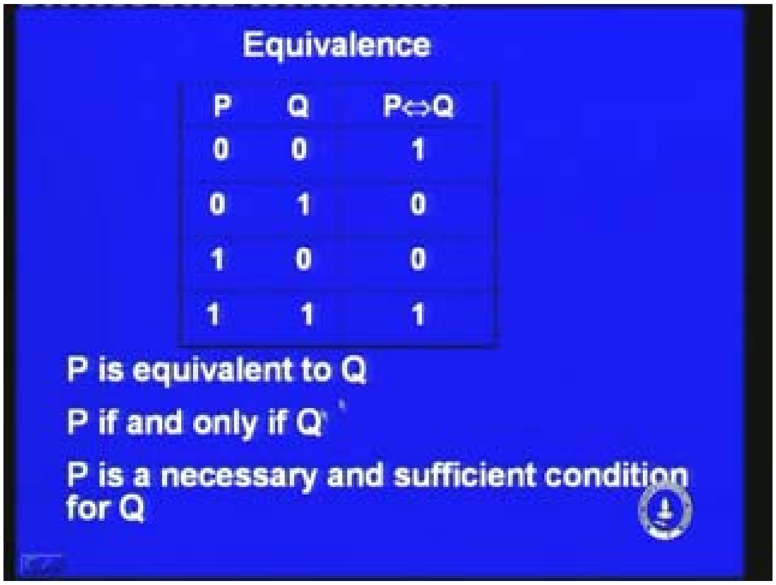
Another thing you must note before going into that is, let us see if there are variables say P, Q, R, S like that, in the truth table there will be one column for each one of the variables and there will be one row for each assignment of the variables. Each variable can be assigned the value true or false that is, 0 or 1. So if there are 4 variables there will be 2 into 2 into 2 into 2, where we get 16 possibilities. So there will be 16 rows in the truth table.

In general when you deal with N propositional variables and you want to study the truth table involving them for an expression involving them, there will be N columns, first N columns will be for variables and later on there will be more columns. Here there are two variables, two columns are for the variables. And here you consider every possible assignment for the variables. So there will be 2 power n possible assignments of truth values for the variables and so there will be 2 power n rows and if you look at them carefully they are representing binary numbers 0 1 2 3. So, they will represent binary numbers from 0 to 2 power n minus 1.

Now coming to this, this is NOT Q implies NOT P is the contra positive. P implies Q is true here, it is false here, what about NOT P? NOT P will be true if P is false, it will be false when P is true, what about NOT Q? If Q is false, this will be true, if Q is true, this will be false and when is the implication false? When the premise is true and the conclusion is false. So, in this case it will be false 1 and 0. If the premise is false the conclusion will be true. I mean the result, the implication will be true.

If the consequence is true then also it is true. Here both are 1, so it will be true as 1 represents true. So you see that these two columns are identical. So P implies Q is the same as saying NOT Q implies NOT P. Many times you will make use of such statements. For example, if a prime number is not a perfect number, so you may say it as if X is a prime it is not a perfect number, or you may also say a perfect number is not a prime, it is saying the other way round in the contra positive manner.

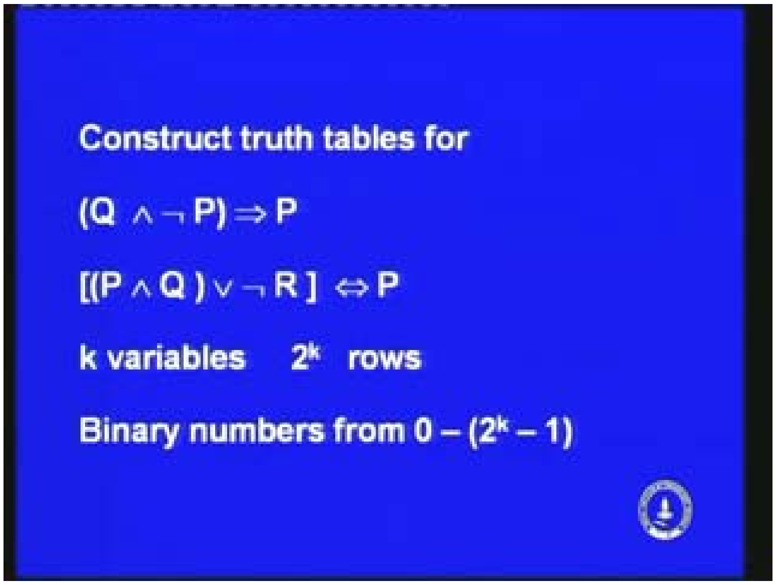
(Refer Slide Time: 37:55)



So the next thing is equivalence. You also have a logical connective equivalence which is denoted by a double arrow with arrow like this which is called equivalence. This is read as P is equivalent to Q or P if and only if Q or P is a necessary and sufficient condition for Q. Generally you read it as P is equivalent to Q or P, if and only if Q. This is also correct. Now, P is a necessary and sufficient condition for Q. When is this compound statement true? This compound statement is true when both P and Q have the same values. When both of them are false or when both of them are true, the compound statement takes the value. If one of them is false and the other is true, then it takes the value false. So this is the truth table for equivalence.

Those who have learnt a little bit about Boolean algebra will know the similarity between propositional logic and Boolean algebra. In Boolean algebra you also study about two operators and NOT, which we will not study in Logic and instead in Logic we study about equivalence and implication.

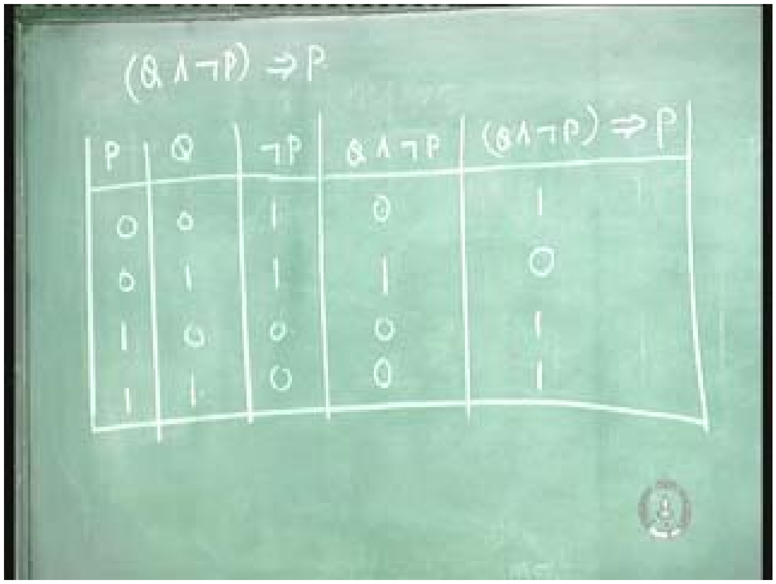
(Refer Slide Time: 39:45)



Now let us take some examples: Let us construct truth tables for Q and NOT P implies P and P and Q or NOT R is equivalent to P. So as I mentioned to you earlier, when you have k variables there will be k columns for each of the variables.

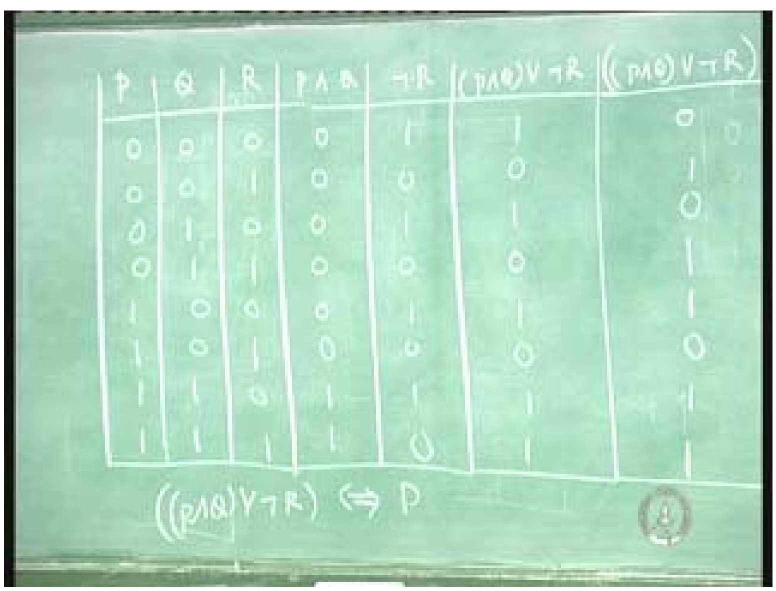
Then some more columns, but there will be 2 power k rows representing the assignments and they will be representing binary numbers from 0 to 2 power k minus1. Now let us draw the truth tables for these expressions.

(Refer Slide Time: 41:40)



Q and NOT P implies P. There are two variables P and Q. So there will be two columns for them and you will give the values 0 0, 0 1, 1 0, 1 1 for them. Then NOT P, Q and NOT P in the whole expression Q and not P implies. Now, when is not P true? When P is false NOT P is true and when P is true NOT P will be false and the adding of that when both of them are true this is true. In other cases it will be false. Now this is the premise and this is the consequence. When is the implication true?

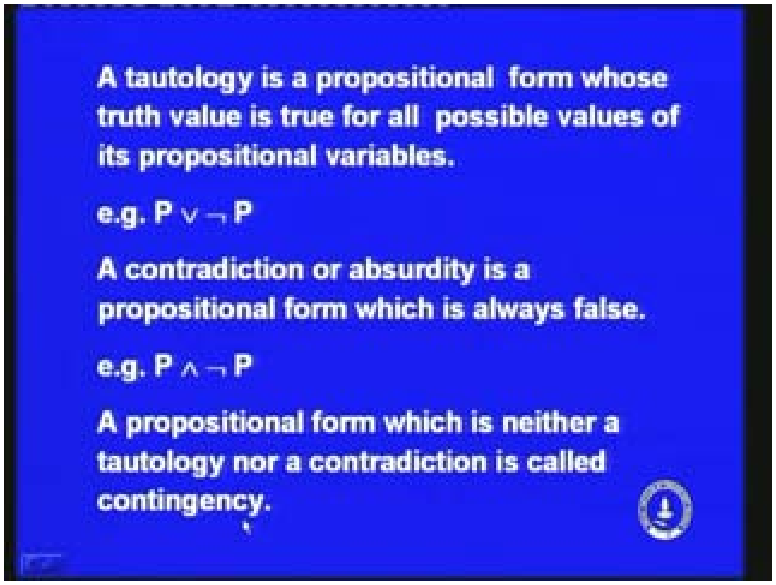
(Refer Slide Time: 45:10)



When the premise is false the implication will be true. And in this case the premise is true and the conclusions are there, consequence is false. So in that case you know that it is false. So this is the truth table for Q and NOT P implies P. Now let us take the other one P and Q or NOT R is equivalent to P. There are three variables P, Q and R. So, there will be three columns for them and there will be all possible assignments for them. That will be: 0, 0 and 0. Usually, you write them in this order so that they are representing number 0, 1, 2 and 3 etc, 1 0 1, 1 1 0, 1 1 1. Now, what is the statement? The statement is this, I will write here, P and Q or NOT R is equivalent to P. So next you must have a column for P and Q, you must have a column for NOT R, So when is P and Q true? When both of them are true, when one of them is false, it will be false. So you get this for P and Q. When is not of r true? When r is false and it is false when r is true. So you will get 1 0 1 0 1 0 and 1 0. Now you have the expression for P and Q or NOT of R. you are orring this and this. When you Orr, when one of them is true the result is true. So when one of them is true, it is true. When both of them are false, it is false. In this case it will be true, in this case it will be false, true and true.

Then the last column is like this: P and Q or NOT R, this is equivalent to P. When are two expressions equivalent? When they take the same value 0 0 or 1. So you have to compare this and this, when they are the same, it is true here, 0 0, it is true but here it is 0 and 1, so it is false, 0 and 0 it is true, 1 and 1 it is true, 0 and 1 it is false, 1 and 1, 1 and 1 is true. So this is the truth table for P and Q or NOT R is equivalent to P. So for any given expression, you can draw the truth table and find out when the whole expression will be true and when the whole expression will be false and so on.

(Refer Slide Time: 47:28)

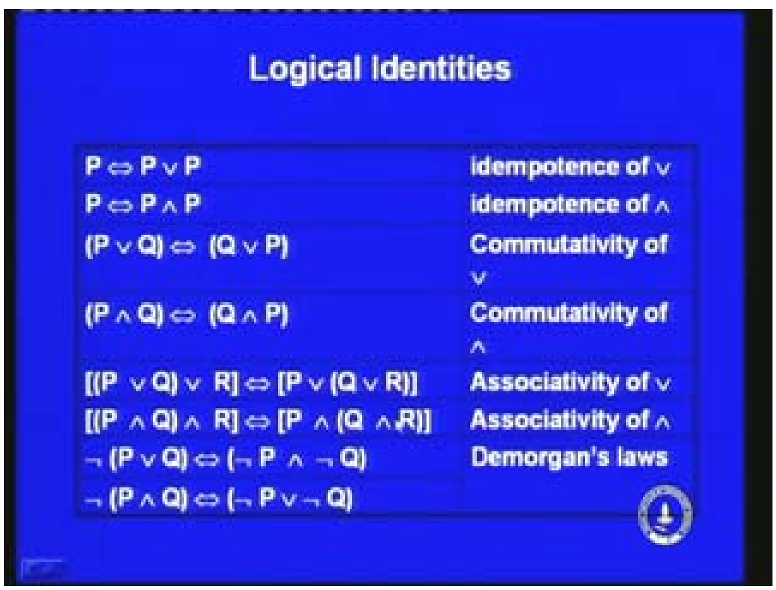


Now a tautology is a propositional form whose truth value is true for all possible values of its propositional variables. Sometimes an expression may be such that, always it takes the truth value true. For example, P or not P, consider this expression P or not P. There are two possibilities, P can be true or P can be false. When P is true, this compound expression is true because one of the component is true. When P is false NOT P will be true. So this compound expression will be true because one of the components is true. So whether you give the value 0 or 1 or false or true to P, this compound expression is always true. Such an expression is called a tautology.

A tautology is a propositional form whose truth value is true for all possible values of its propositional variables. The other way round, a contradiction or an absurdity is a propositional form which is always false, take this expression P and NOT P. There are two possibilities, P is true or P is false. If P is true NOT P is false, so this expression will be false because and will be true only when both of them are true. It is not possible for both P and NOT P to be true either one of them will be true. In any case, one of them will be true, the other will be false. The possibility is, this is true, this is false or this is false, this is true. In any case this compound expression P and NOT P will always be false. Such an expression is called a contradiction.

A propositional form which is neither a tautology nor a contradiction is called a contingency. These are the examples which we consider: Q and NOT P implies P and Q and P or NOT R is equivalent to P. These two are the examples of contingency. In some cases they are true but in some cases they are false. So they are examples of contingency. Given a propositional form, it is always possible to find out whether it is a tautology or not because you can always draw the truth table and if the last column is always 1 1 and 1 then it is a tautology. If the last column is always 0 0 and 0 then it is a contradiction and sometimes when it is 0 and sometimes it is 1, then it is called a contingency.

(Refer Slide Time: 50:15)



Then, we have a lot of logical identities. These logical identities can be used in simplifying logical expressions. Let us see one by one. P is equivalent to saying P or P. When you say P or P, it is not necessary to say twice, it is enough to say once. This is called idempotence of R. Then, if you have P and P, it is not necessary to write like this. It is enough to say like this: So P is equivalent to P and P. This is called idempotence of and. So, whenever in an expression you get something like P and P, you can replace it by

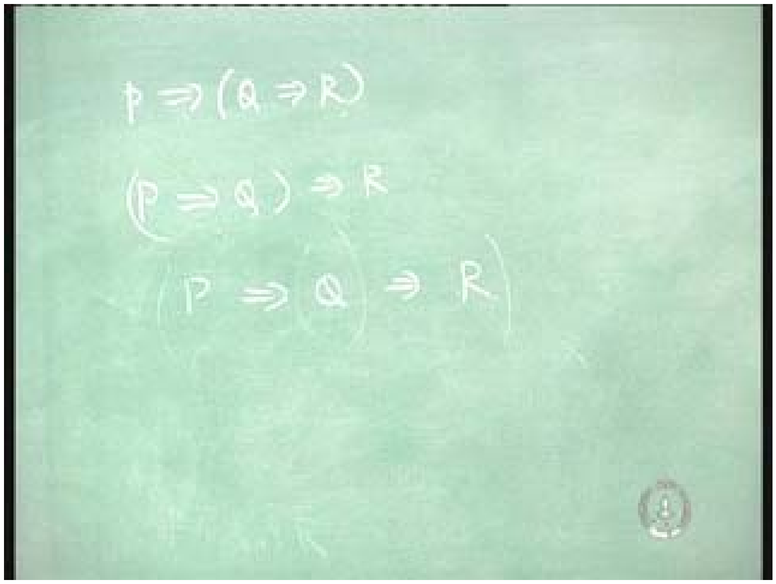
P. Then you have the commutative laws. It is equivalent to saying Q or P is equivalent to saying P or Q. You can interchange the variables, it does not affect the meaning, either this or that. A or B is B or A, so P or Q is same as Q or P. So, this is called commutativity law.

Similarly, you have commutativity for an, P and Q is equivalent to saying Q and P and you have associative laws P or Q, then or R. That is first or P or Q. Then you take R and combine, that is equivalent to having P or Q or R and P and Q and R is equivalent to saying P and Q and R. This is called associative law. And in general whenever you write an expression you try to use parenthesis. There is no priority but generally NOT has higher priority than and or if you use only these three it is because you can write NOT P and Q. This is NOT P and Q that is it is equivalent to saying NOT P and Q. You can write without the parenthesis, it is not NOT P and Q, this is different. If you want to write like this, you must put the parenthesis in a proper way.

(Refer Slide Time: 51:20)

So, we see that, but P and Q and R, you can write without parenthesis because of associativity. Whether we group this first, or this first is immaterial. When both the operators are without any problem, you can write like this. That is what is meant by associativity of law and similarly for OR.

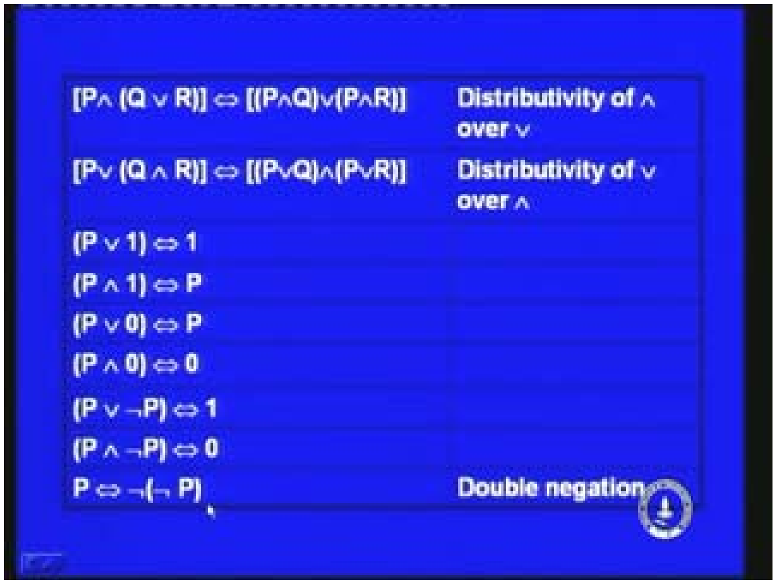
(Refer Slide Time: 52:00)



Whereas, you can realize that P implies Q implies, if you want to write like that, P implies Q implies R, this is different from saying P implies Q implies R. You cannot say something like this; P implies Q implies R because you do not know whether you mean this or that. You can draw the truth table and verify that this has got a different value, they are not the same, so you cannot write something like this. You have to put parenthesis wherever is either you mean this or you mean that. So, implication is not associative.

And there are Demorgans laws which are very common and very useful also. NOT of P or Q, you can bring the NOT inside which will mean NOT P and NOT Q. You can again draw the truth table for this and see that they will be identical, the columns representing them will be identical. So, when you bring the NOT inside, R becomes similarly NOT of P and Q will be NOT of P or NOT of Q, So, when you bring the NOT inside, it becomes R and R becomes And. These are called Demorgans laws.

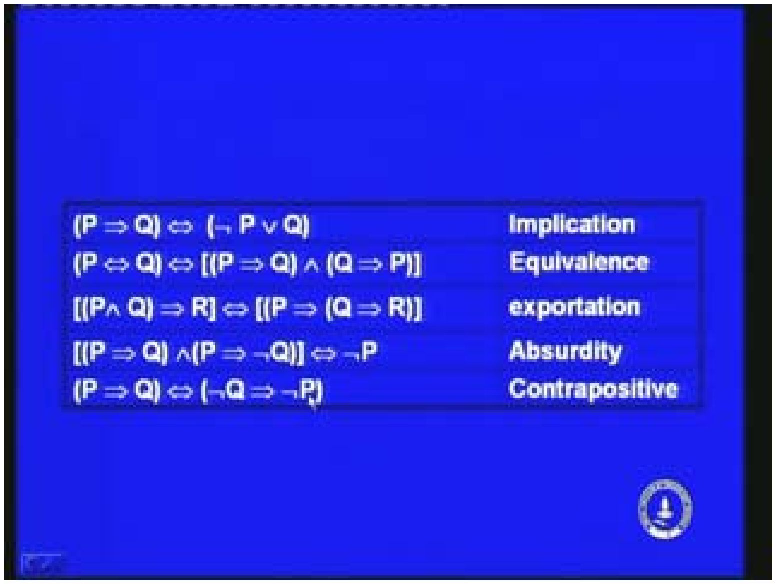
(Refer Slide Time: 55:00)



Then you have distributive laws and distributes over R. That is P and Q or R is equivalent to saying P and Q or P or R. So you can distribute and over R and write like this. Sometimes you can use the distributive laws in the reverse direction also. Another thing is, R distributes over and, P or Q and R is the same as saying P or Q and P or R. Again, here also in the distributive law sometimes you can apply in the reverse. Then you have these rules implying when one of the operands is true or false, P or 1 is always 1. So this is a compound statement. But when one of the operands is true, the compound statement is always true, this we know. So even when one of them is true, this is true and now we are taking one operand as true only so this is 1, P and 1 if P is true this will be true, if P is false this will be false. So P and 1 has the same value as P. Similarly for R, P R 0, if P is true, this will be true if P is false, this will be false. So P or 0 takes the same value as P and the and with 0, P and 0 because one of the operands is false, the compound statement will be false, so you will have P and 0 which is the same as 0 or false.

P or NOT P is always 1, which is always true. This is called a tautology. This is what we have seen earlier. P and NOT P is called a contradiction or absurdity if it always takes the value 0. And you have double negation saying P is equivalent to NOT of P. So if you use negative negation operation twice, it is equal to the original statement.

(Refer Slide Time: 56:10)



There are some more statements like this. We shall briefly look into them. We shall look into them in more detail in the next lecture, but just we have a brief look at them. P implies Q, you can write it as NOT P or Q. This is equivalent to saying P implies Q. And similarly P is equivalent to Q, you can equivalently say it as P implies Q and Q implies P. This rule is also true P and Q implies R, is equivalent to saying P implies Q implies R and P implies Q and P implies NOT Q is equivalent to saying NOT P. It is called absurdity. This is used in proves by contradiction. This is the most commonly used. This we have seen earlier P implies Q is equivalent to saying NOT Q implies NOT P which is the contra positive and so on. So in this class we have learnt about a few logical connectives and on how to write the truth table and so on. We shall continue with these concepts in the next lecture.