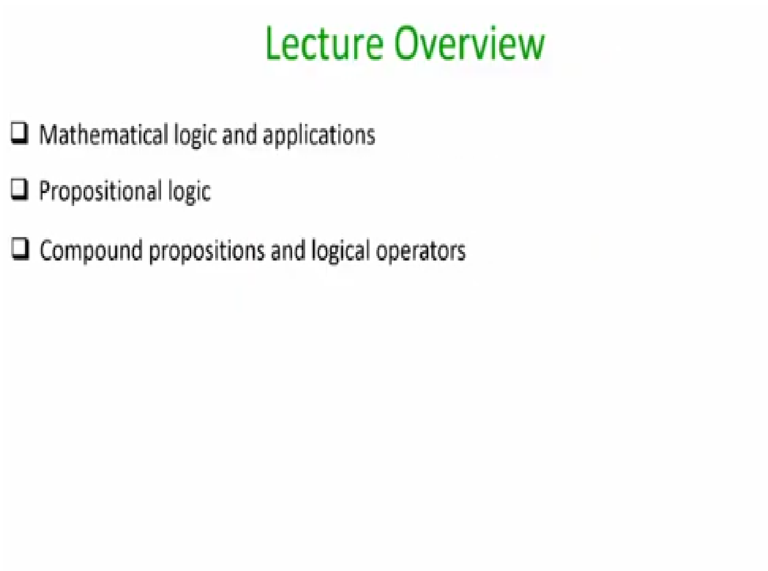
Discrete Mathematics
Prof. Ashish Choudhury
Department of Mathematics and Statistics
International Institute of Information Technology, Bangalore

Lecture -01
Introduction to Mathematical Logic

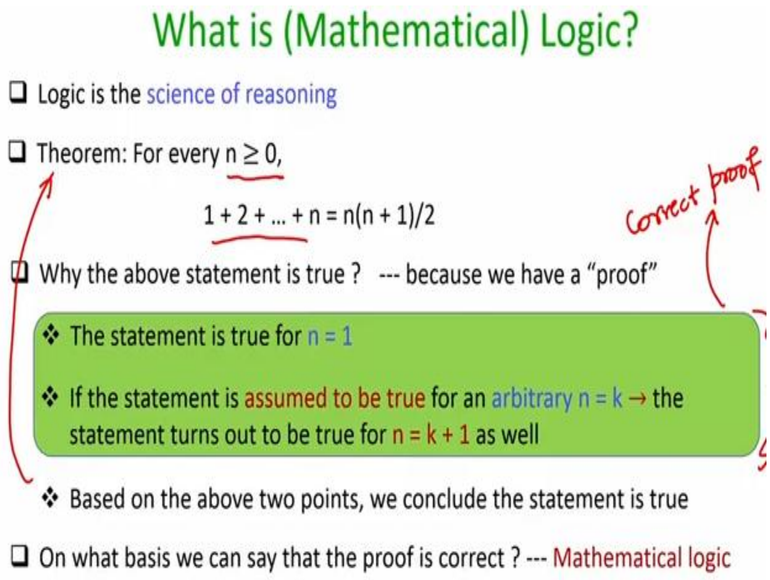
Hello everyone, welcome to this lecture on introduction to mathematical logic.

(Refer Slide Time: 00:24)



The plan for this lecture is as follows. In this lecture we will discuss mathematical logic and applications. We will discuss propositional logic and we will discuss compound propositions and logical operators.

(Refer Slide Time: 00:39)



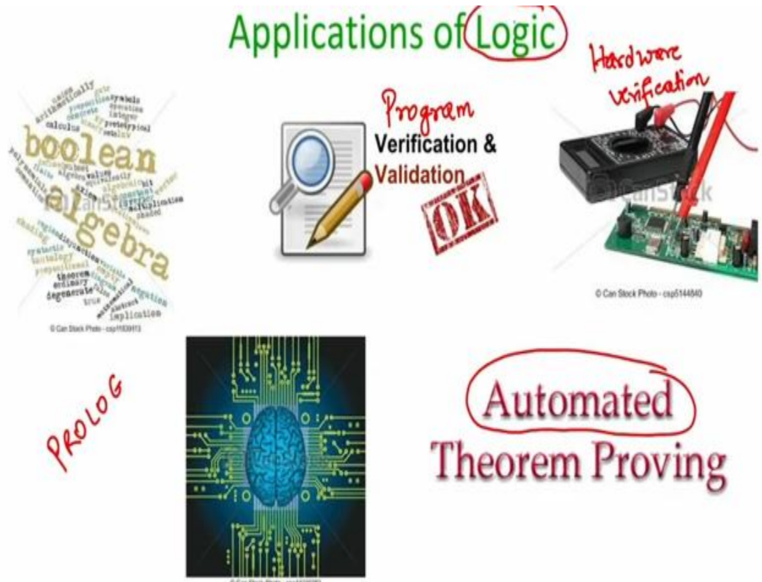
So to begin with let us ask the question what is logic or more precisely what is mathematical logic. So on a very high level, mathematical logic is the science of reasoning. Namely it tells you how to verify or how to conclude whether a statement or mathematical statement is true or mathematical statement is false. To be more precise consider this theorem statement, which is a very straight forward theorem statement, which says that, for every $n \geq 0$ the summation of first n numbers is $\frac{n(n+1)}{2}$.

So this is the mathematical statement and now we want to verify whether this statement is true or false. Well, it turns out that statement is true because we have something called what we call as proof and there are several proofs for proving this statement to be true, so one of the simplest proofs is the proof by mathematical induction. So the proof by mathematical induction basically argues that the statement is obviously true for $n = 1$.

And then we show that if you assume that the statement is true for any arbitrary $n = k$ then we also end up showing that the statement is true for $n = k + 1$ as well. And these two statements constitute a proof and based on the above two points we conclude that our theorem statement that we made here is true. Now the question here is on what basis we can say that these two statements that we have written here in this highlighted rectangular box constitutes a correct proof.

How do we argue that indeed this is a correct proof? So on a very high level mathematical logic or the science of reasoning is what helps us to conclude that two statements that we have written here indeed constitute a valid proof for proving that this statement is correct.

(Refer Slide Time: 02:44)



It turns out that there are several applications of logic, So for example, we use it extensively in Boolean algebra, which forms the basis of computer architecture. We also use it for program verification and validation. So what I mean by program verification, here is the following. Suppose we have written a software for particular application. How do we verify whether our software is performing its intended task?

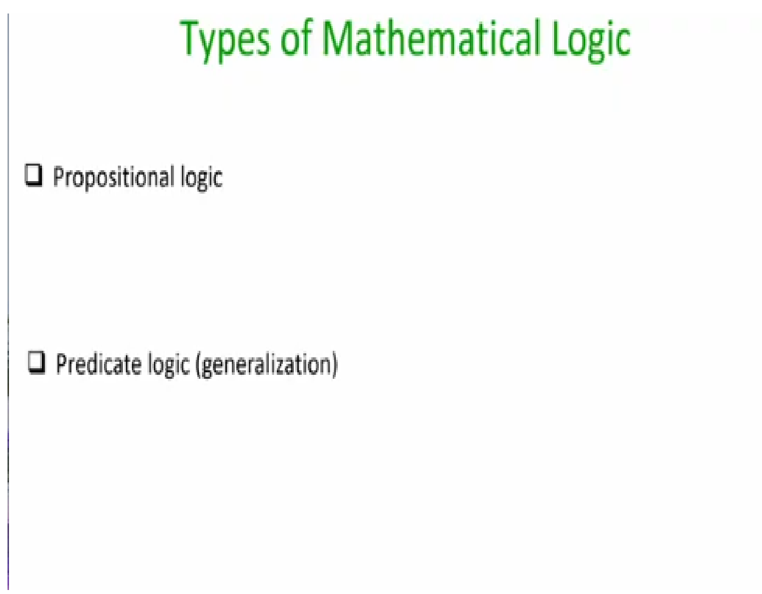
One way of verifying that is we run the program or the software for various inputs and verify whether it is giving the correct output or not and if it is giving the correct output for various inputs that we have tried then we believe that the program is indeed performing its intended task. But in this process there might be a possibility that there is some bad input which we never tried, for which the program may not be performing or may not be giving you the right output. So this way of verifying the program is not the right approach specifically for softwares or programs which are meant for very critical applications. So for example, if your software is developed for an Aeroplane application then we require the software to be completely robust. It should be foolproof. It should perform these actions always correctly.

So there we use mathematical logic to verify whether the software is indeed performing its right task or not. So there is a huge body of work, a discipline of computer science, which we call as program verification and validation where the goal is to verify whether the given software or the program is performing its correct intended application or not. So there also we use mathematical logic.

In the similar way we use mathematical logic for hardware verification. So for instance, if you have a motherboard and there are several chips or various units which are embedded there which are very small in size. So how do you verify whether the embedding that you have done on your motherboard is correct or whether it is performing the right task or not. So again we use mathematical logic to do hardware verification. Mathematical logic is very useful in artificial intelligence.

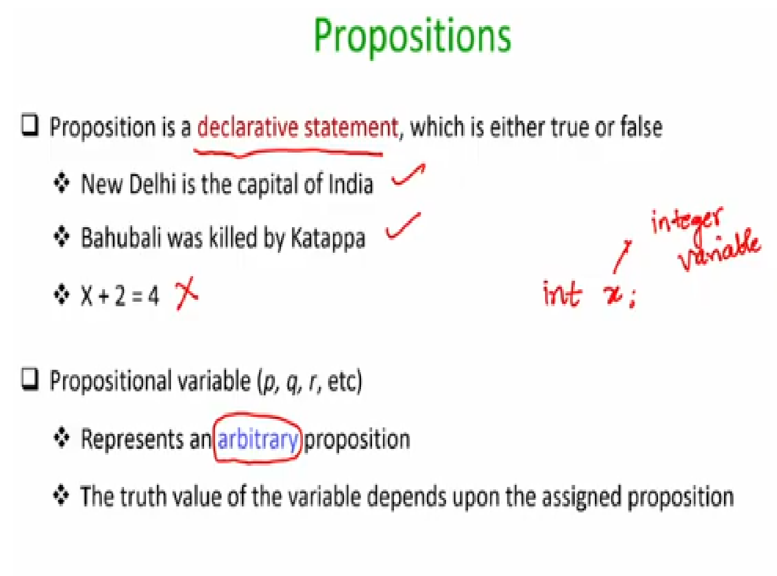
In fact there is a programming language, which is called as PROLOG which is used extensively in AI applications and the basis for PROLOG is mathematical logic and we also use mathematical logic for proving theorem in the form of automatic theorem proving., where in automated theorem proving we write computer programs to prove whether a statement is true or false, so there also we use mathematical logic extensively. So it turns out that mathematical logic is very important, very significant.

(Refer Slide Time: 05:42)



And there are various types of mathematical logic, the basic form of mathematical logic is called as propositional logic and a generalization of propositional logic is called as predicate logic.

(Refer Slide Time: 05:57)



So let us begin our discussion on propositional logic. So we first define what is a proposition. So informally, proposition is a declarative statement which is either true or false. But it cannot take both the values simultaneously that means it has to be either true or it has to be either false. So for instance consider the statement that New Delhi is the capital of India. It is a declarative statement because it is declaring something about the city called New Delhi. It is declaring that the city New Delhi is capital of India or not.

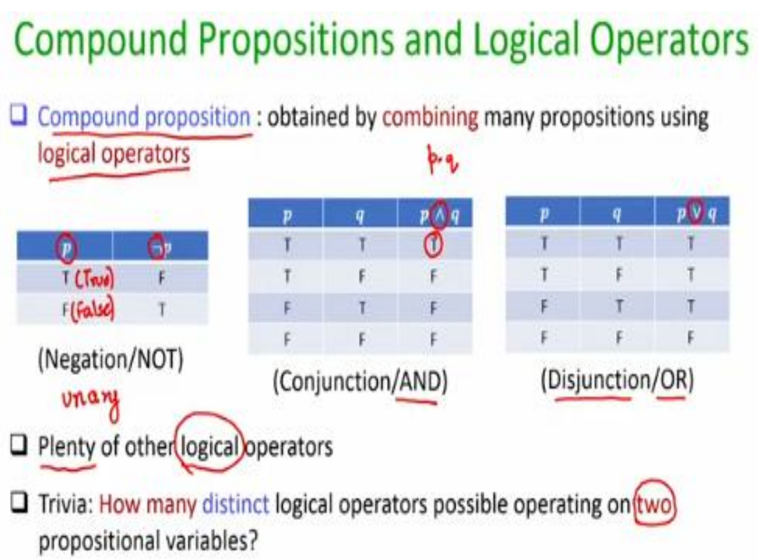
And indeed this statement can be either true or false because if you take the city New Delhi then either it will be the capital of India or it will not be the capital of India, but it cannot happen that it is simultaneously capital of India as well as not the capital of India. So, that is why it is a proposition. In the same way if I make a statement like Bahubali was killed by Katappa or then this is a declarative statement because it is declaring something about a character called Bahubali and Katappa and the statement is indeed true. Because we now have approved whoever has seen the movie Bahubali part 2, we now have the witness that indeed in that movie the character Katappa killed Bahubali. So this is also a declarative statement which can be either true or false and hence it is a proposition. Whereas the statement X+ 2 is equal to 4 this is not a proposition. Well, it is a declarative statement because it is declaring something about X, 2 and 4.

But we do not know what is the value of X. Depending upon the value of X this statement can be either true or it can be either false. So it can simultaneously take values true as well as false. So hence, it is not a proposition. Now let us next define what we call as propositional variables and these propositional variables are typically denoted by lower cap letters. So for instance p, q, r etc.

So, what is the propositional variable? It is a variable which represents an arbitrary proposition. That means it is a placeholder to store an arbitrary proposition and the truth value of this propositional variable depends upon the exact proposition which we assign to these variables p, q, r. So for instance in your programming language, say for instance, in your C programming language, we define variables like integer variables.

So if I make a statement declaration, like int x that means I am declaring here that x is an integer variable and this variable can store any integer value. It can store values 1, 2, 3 any integer value. In the same way a propositional variable it is a placeholder or an arbitrary proposition and depending upon what exact proposition we assign to that variable the variable can take the truth value either true or false.

(Refer Slide Time: 08:54)



Now let us next define compound propositions. So a compound proposition is a larger proposition or a bigger proposition, which is obtained by combining many propositions using what we call as logical operators. So the simplest form of the logical operator is the ¬ operator, which is an unary operator because it operates on a single variable and the truth table or the truth assignment for this negation operator is as follows.

So imagine p is a variable propositional variable and this propositional variable can take two values either true or false. So T here stands for true and F stands for here false. So this negation operator is defined as follows. If the variable p is true then the ¬ of p will be false whereas if the variable p is false then the ¬ of that variable will be true. That is how this unary operator negation is defined.

So we will be using this notation for denoting the negation operator. In various other books there are different notations which are used for the same negation operator. In some books, they use this tilde symbol ~p to denote negation of p. In some books they also use this p complement that means we put this bar on top of p ($\overline{p}$). So there are various notations for the same negation operator, but we will be extensively using the notation that is there in this table ($\neg p$).

Now we define another logical operator which is called as the conjunction and it is also called as AND, logical AND. We denote this operator by this notation (∧) again in some books they used a notation p dot q. But I will be using this notation p and this symbol q and this is a binary operator because it operates on two propositional variables. So if your p is a variable and q is a variable then p conjunction q is another propositional variable.

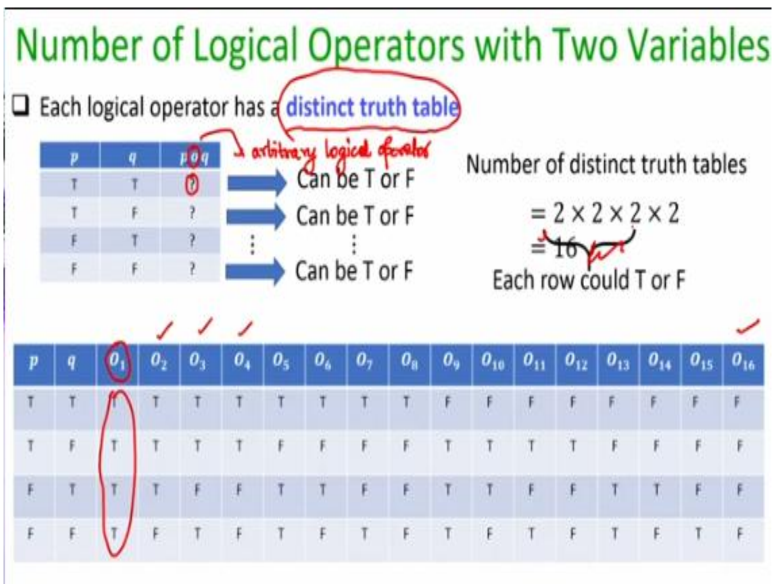
And a truth value of p conjunction q is defined as follows; p conjunction q is defined to be true only when both the variable p as well as the variable q are simultaneously true. If any of them is false then the conjunction of p and q is defined to be false. The next logical operator is the disjunction operator which is also called as OR operator denoted by this notation (∨) and it is defined as follows.

If any of the variables p or q is true then the disjunction of p and q is defined to be true. But if both p and q is false then disjunction is defined to be false. And it turns out that there are plenty of other logical operators that we can define on propositional variables. By the way why I am calling them as logical operator? Because it is operating on propositional variables, that is why I am calling it as logical operator.

And the result of any logical operator will be either true or false because we are in two valued logic where the propositional variable of the result of applying the logical operator can take only one of the two possible values. Now here is the question. How many distinct logical operators are possible operating on two propositional variables? That means suppose I have a propositional variable p and a propositional variable q.

My question is how many distinct types of logical operators I can define on these two propositional variables. So you can think in your mind and try to come up with an answer.

(Refer Slide Time: 13:05)



Let us try to derive the answer for the question. So to begin with you should understand that each logical operator has a distinct truth table. That means it does not matter what is the name of the logical operator that you give. You can call it conjunction, disjunction, x, y, z, alpha, beta, gamma. You can call it by any name. Once you fix the name of a logical operator operating on variables p and q then it has a distinct truth table. And what do I mean by a distinct truth table?

Well, since we are trying to define a logical operator on two variables and the variables p and q can take these four combinations. p can be true q can be true p to be true q to be false p to be false q to be true and p to be false q to be false. So we have these four possible combinations and here o is denoting an arbitrary propositional variable. My goal is to identify how many different types of arbitrary logical operator o I can define here.

That is my question here. And the question mark here below is what will be the truth value of p operation q for the various truth assignments of p and q. That means here the first question mark says what can be the possible value of p operation q if both p and q takes the truth value true. The second row here denotes what is the value of this operation operator p operation q if p takes the value true q takes the value false and so on.

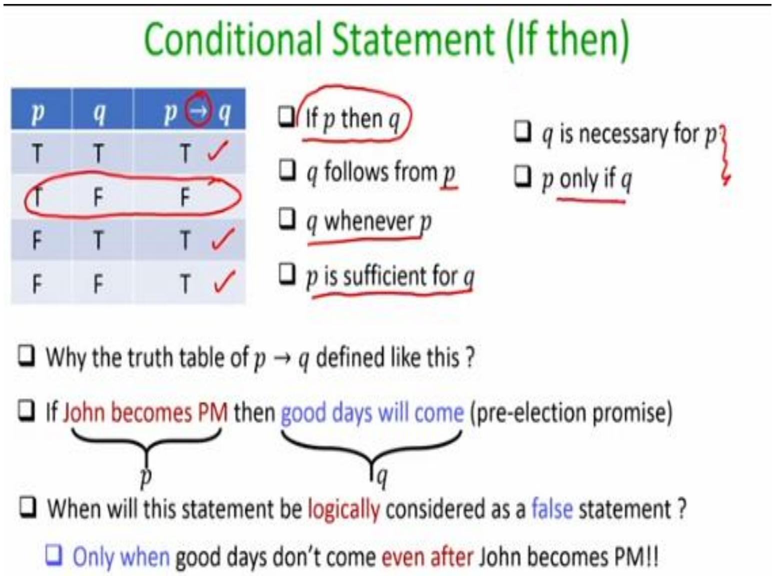
It turns out that since we are dealing with mathematical logic where each variable can take only two possible values then each of this possible question marks can be either true or false that means my operation o can be such that for p to be true and q to be true the result of p operation q can be either true or it can be either false. In the same way my operation o can be such that for p to be true and q to be false the operation p operation q may result in true or false.

And this holds for all the four possible rows. It means that if I consider the number of distinct truth tables here, I can construct at most in fact exactly 16 distinct truth tables because for each of the possible rows here for each of the possible question marks here I have two possible options and each of them is independent of each other. Each of them are independent. So that is why I have to tell 16 number of distinct truth tables are possible.

Sorry for this rendering issue here. This parenthesis should come down here. So since each row should take the value true or false that is how I get a total number of distinct truth tables to be 16. And what I have done here is I have written down the 16 distinct truth tables which can be possible. I am calling those distinct 16 distinct truth tables by various operators. So operator 1 denotes a truth table where for p to be true and q to be true the result is true. For p to be true q to be false the result is true for p to be false q to be true the result is true and for p to be false q to be false the result is true. That is one possible truth table and I am saying that that corresponding operator is operator O1. In the same way, I have operator O2, O3, O4 and operator O6. And now I can give fancy names to each of these operators. I can give some name conjunction, disjunction, XOR, exclusive OR, exclusive NOR etc etc.

But I cannot give more than 16 distinct names because I do not have more than 16 possible truth tables. So that is why there are 16 possible logical operators distinct logical operators, which are possible with two propositional variables.

(Refer Slide Time: 17:24)



Now let us define another logical operator, which is the conditional statement. This is also called as if then statement and we use this notation p → q and the truth table for p implies q is defined as this. So you can see that p implies q is true for three possible combinations and it is defined to be false only when p is true, but q is false. Now, the question is why the truth table of p → q is defined like this?

That means why p → q is true, even if both p and q are false or why p → q is defined to be true even if p is false, but q is true. So to understand that let me give an example, a very simple example. Suppose we make this logical statement. Suppose there is a PM candidate called John and he makes the election promise that if he becomes the prime minister then good days will come for the country.

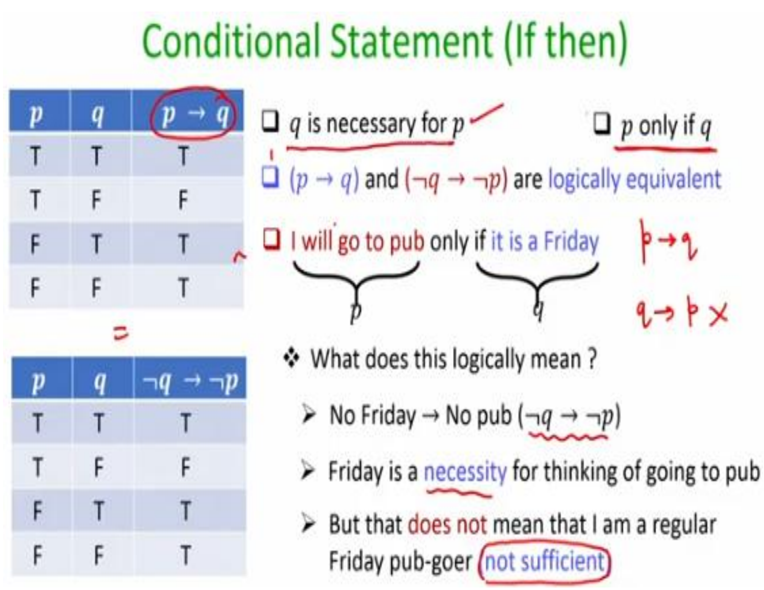
This is a pre-election promise which John makes. Now the question is when exactly will you consider this logical statement to be false? It turns out that this logical statement will be considered as a false statement only when good days does not come even if John becomes the PM. For all other cases the statement will be considered as a true statement; that means if John does not become the PM at the first place then I do not care whether good days come or not.

Overall, the statement will be considered as a true statement because John is not breaking his promise. We will be saying that John is breaking his promise only when John becomes PM but still good days are not coming for the country and that corresponds to the truth assignment p to be true and q to be false and as a result for that combination I define p implies q to be false. There are other interpretations of this if then statement.

So the usual interpretation of p implies q is this if p then q where there are other interpretations like q follows from p. So q follows from p means if you ensure p to be true we can denote we can come to the conclusion q. p is sufficient for q tells that if you ensure that p is happening or p is true then the conclusion is q, right? q whenever p is another way of interpreting if p can then q. Now, the question is why p implies q denotes q is necessary for p and why p implies q denotes p only if q.

So these are the two common forms of p implies q, which is a common source of confusion for the students.

(Refer Slide Time: 20:20)



So, let us try to understand that why p implies q or why if p then q can also be interpreted as q is necessary for p and why it can be interpreted as p only if q. So to understand that let me first claim here that the statement $p \rightarrow q$ and the statement $\neg q \rightarrow \neg p$ are both logically equivalent. What I mean by both are logically equivalent is that you have the truth table of $p \rightarrow q$ and this is the truth table of $\neg q \rightarrow \neg p$.

And you can see that both of them have the same truth table. That means both $p \rightarrow q$ as well as $\neg q \rightarrow \neg p$ have the same truth table. So if p is true, q is true then p implies q is true and same is as the case for $\neg q \rightarrow \neg p$. That means row wise the first row here and the first row here are the same. Row wise the second row here and here are same. Row wise the third row of the two tables are the same and row wise the fourth rows of both the tables are the same.

And that is why I can say that $p \rightarrow q$ and $\neg q \rightarrow \neg p$ are both logically equivalent. Now consider this statement. I make the statement my statement is I will go to pub only if it is a Friday and say the statement I will go to the pub is represented by p variable p. So p is a propositional variable denoting the statement I will go to the pub and q is another propositional variable denoting the statement it is a Friday. My claim is that this statement can be represented by $p \rightarrow q$. Why so?

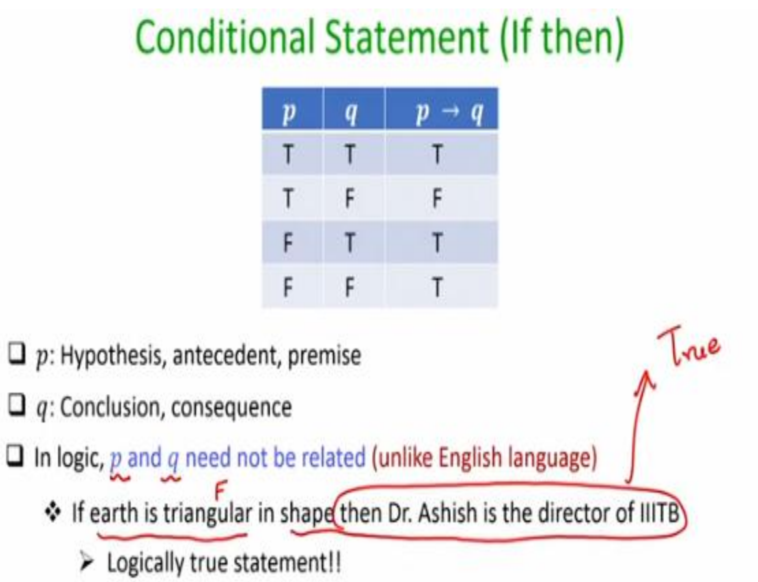
Because if you try to understand the logical meaning of this English statement, then what I am trying to say here is that if it is not a Friday definitely I will not go to the pub. Because that is how the statement “only if it” is interpreted and no Friday implies no pub can be represented by $\neg q \rightarrow \neg p$ because I am representing it is a Friday by the variable q. So it is not a Friday will be represented by $\neg q$ and I will not go to the pub will be represented by $\neg p$ because p as per my definition denotes I will go to the pub.

So in that sense, Friday is a necessity or it is a necessary condition for thinking of going to the pub. That means I will think about going to the pub only if it is Friday. If it is not a Friday definitely I would not go to the pub. So that is why Friday is a necessary condition here. But Friday is not a sufficient condition for me going to the pub. Because it might be possible that there is some Friday on which I am ill or I may have some personal work for which I am not going to the pub.

That means in this case being Friday is not a sufficient condition; that means you cannot simply conclude that since it is a Friday definitely this guy will go to the pub and that is why this statement I will go to the pub only if it is a Friday will be represented by p → q. It would not be represented by q → p because q → p denotes that if it is Friday then I will go to the pub, but that is not what I want to represent here.

I want to state here the necessity of the condition that it is a Friday then only I can think of going to the pub and that is why the statements of the form that q is necessary for p or the statements of the form p only if q are represented by p → q.

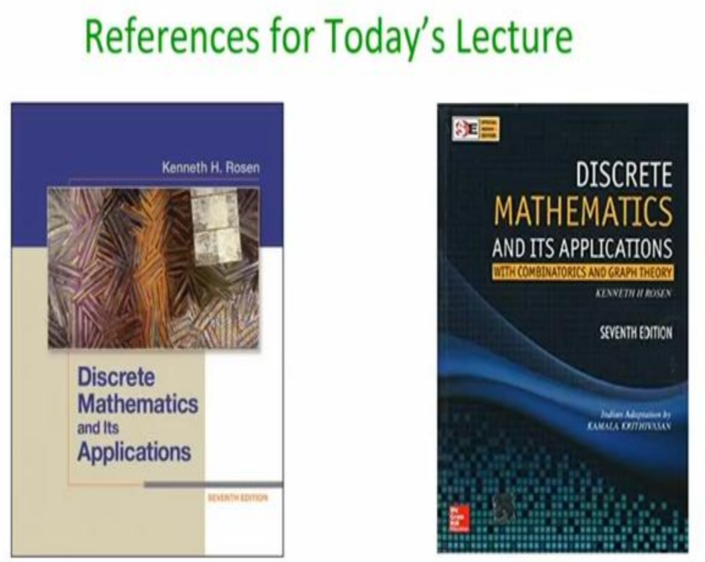
(Refer Slide Time: 24:21)



So whenever we write a conditional statement of form, if then, p → q, then the p statement or whatever statements you have for the propositional variable p they are called as hypothesis or antecedent or premise and q denotes the conclusion or the consequence. It turns out that in mathematical logic the statements in p and in q need not be related in the English sense. So, for example, if I make a logical statement that if the earth is triangular in shape then Dr. Ashish namely myself is the director of IIIT Bangalore.

So in English Language, this is an absurd statement. This is a completely incorrect statement. But in mathematical logic this is a correct statement because earth is definitely not in triangular shape. So this is false. And remember for p → q I do not care what is the q part. As soon as my p part is false I can conclude that p → q is an overall true statement. That means even though the conclusion here namely Dr. Ashish is the director of IIIT Bangalore, which is actually a false conclusion in English, it does not matter whether it is true or false since my premise is false here namely since our earth is not triangular in shape, the overall statement is a true statement logically true statement and that is why in mathematical logic the statement p and the statement q may not have any relation with each other. They are just logical statements.

(Refer Slide Time: 25:58)



So that brings me to the end of this lecture. To conclude in this lecture we started our discussion on mathematical logic. We saw various applications of mathematical logic. We started our discussion on propositional logic. We defined what is a proposition? We defined compound propositions and how compound propositions can be formed from simple propositions using logical operators. We discussed various logical operators, like disjunction, conjunction, negation, if then statement and so on. Thank you.