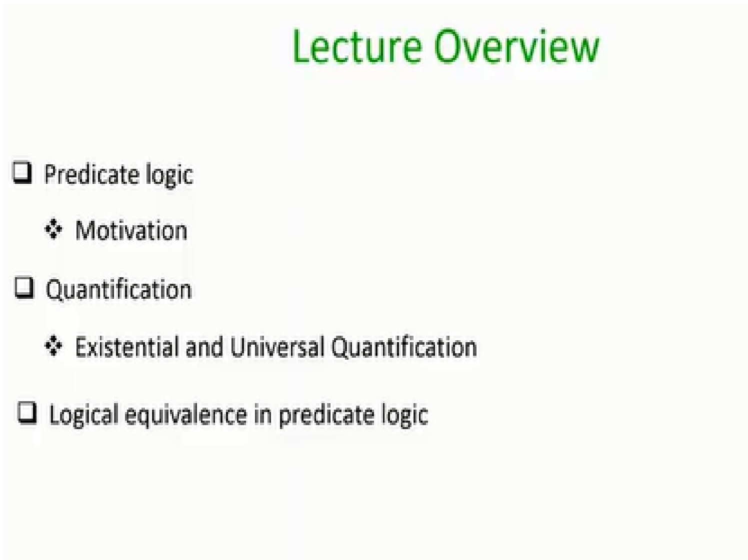
Discrete Mathematics

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

Lecture -08
Predicate Logic

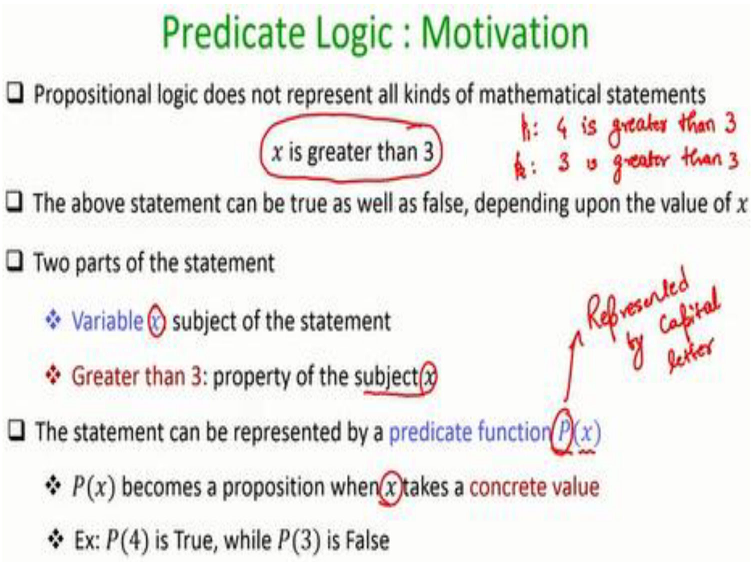
Hello everyone welcome to this lecture on predicate logic.

(Refer Slide Time: 00:25)



Just to recap, till now we had extensively discussed propositional logic how do we form compound propositions, how do we verify whether argument forms are valid, various ways of verifying argument forms are valid or not using rules of inferences and so on. The plan for this lecture is as follows: in this lecture we will discuss predicate logic and its motivation, we will discuss about quantification mechanisms and we will discuss logical equivalence in predicate logic.

(Refer Slide Time: 00:57)



So let us start with the motivation of studying predicate logic, so it turns out that even though propositional logic is very interesting it cannot represent all kinds of mathematical statements that we are interested in. So for instance consider this declarative statement at x is greater than 3, which is a declarative statement but it is not a proposition because until and unless the value of x is assigned, we cannot decide whether the statement is true or false.

Whereas, the definition of proposition is, it is a declarative statement for which we know that it is either true or false but not simultaneously it cannot take the value true or false. So that is not happening for the first statements of this form. So the question is how exactly we can represent statements of this form because in general when we are writing mathematical theorems and statements, very often want to characterize properties about arbitrary sets, arbitrary domains without worrying about the underlying values which are going to be taken in the domain and so on. So for that we need predicate logic, so it turns out that there are two parts of the above statement which we are interested to represent: the first part is the variable x which is the subject of the statement, because in this statement we are trying to say something about the property of x.

We want to say that x is some value which is greater than 3, we do not know whether that is true or not because that depends upon the exact value that x is going to take. But we want to characterize the property that whatever is the value of x it is greater than 3 or not. So greater than

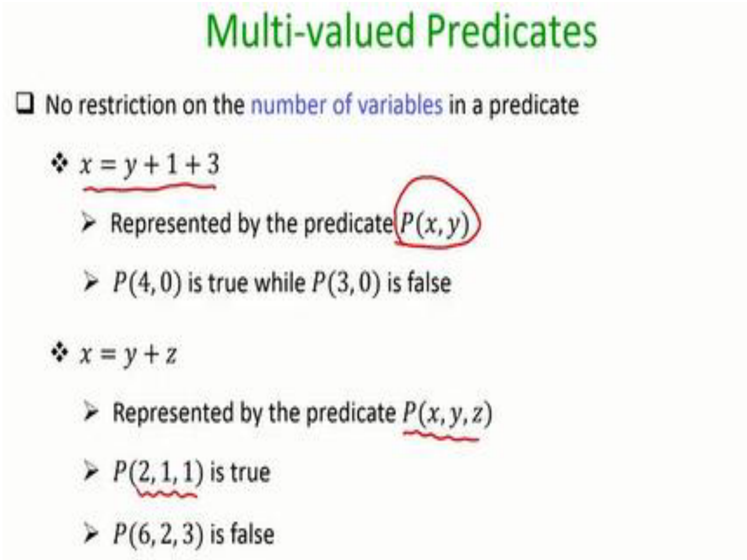
3 is the property for the subject x. So the way we represent these statements in the predicate logic is we introduce a function which I call as a predicate function.

And this predicate function typically they are represented by capital letters and the choice of exact capital letter is up to you. You can use M, A, B, C anything and why we are using capital letters to differentiate from propositional variables for which we use lower case letters. So the propositional variables they are represented by lower case letters predicate function variables, they are represented by capital letters.

So this is a function of variable x, where x can take any value and this P(x) becomes a proposition when we assign a concrete value to x. So for instance if I assign the value 4 to x, then the proposition that I obtained is p_1, which is 4 is greater than 3. Now the statement 4 is greater than 3 is indeed a proposition, it is a declarative statement and we know it is a true statement that means there is no ambiguity about the truth value or the truth status of p_1.

Whereas P(3) is another proposition, say p_2 which is the statement 3 is greater than 3. Now this is a proposition because this is the declarative statement and the truth status of p_2 is confirmed here, it is a false statement, there is no more ambiguity left. Now we can see how we can represent or how we can deal with statements of this form where we have some abstract variable and we want to state or declare some property about the variable using these predicate functions.

(Refer Slide Time: 04:53)

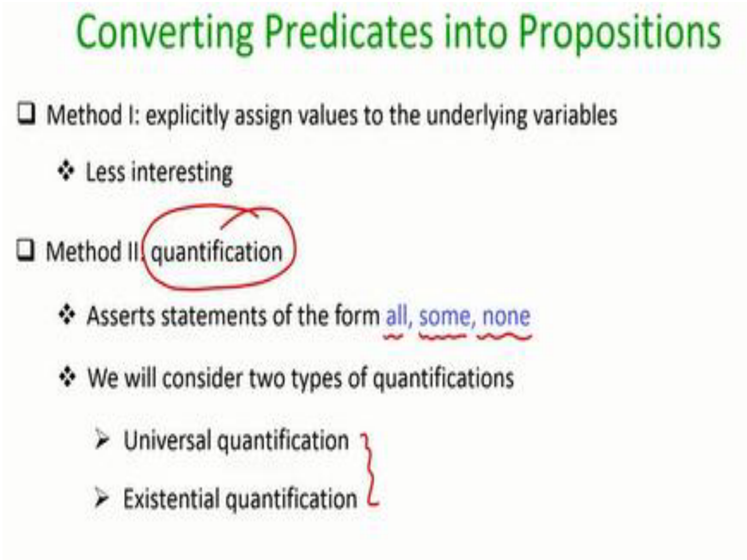


It turns out that we can define multi-valued predicate functions right? So the previous example was for statements where we had only one subject namely the subject x but now you might be dealing with statements where you have multiple subjects. For example, we want to represent declarative statements of the form x equal to y + 1 + 3. So this is not a proposition because until and unless we do not assign values to x and y, we do not know what is the status of the resultant proposition.

So I can represent this statement by a predicate variable, I use the predicate variable capital P and it is a function of two variables x and y. When I assign the value 4 to x and zero to y, I get the proposition say p_1 that 4 equal to 0 + 1 + 3 and this is a true statement, this is a true proposition. So I will say that P(4,0) is true whereas if I assign the value x equal to 3, y equal to 0 then this is a false proposition.

So now you can see that this predicate variable is acting as an abstract function as a placeholder and it can represent multiple propositions depending upon what exact values you assign to the corresponding variables, which are there in the corresponding predicate function. In the same way you can have three valued predicate functions. So, for example, if I want to represent x equal to y + z and I can introduce a predicate function P of 3 variables and depending upon the values which you assign to x, y, z you get propositions, which can be either true or false.

(Refer Slide Time: 07:02)

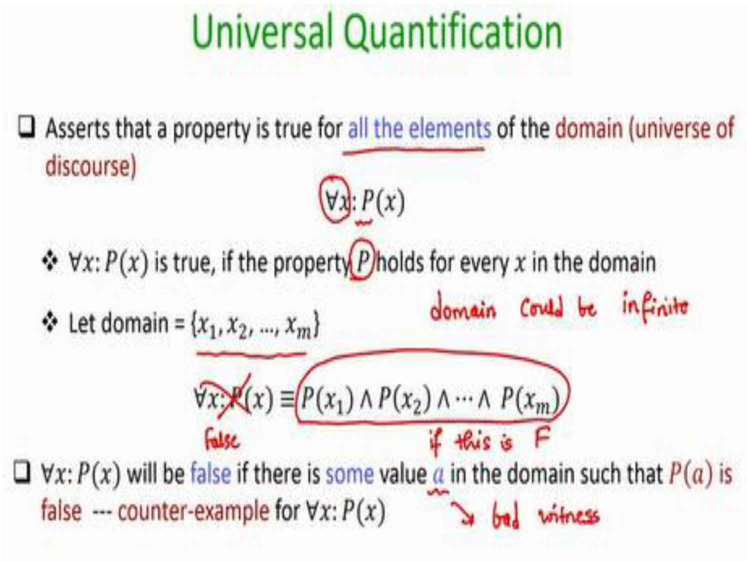


So now once we have predicates we will be interested to convert them into propositions because then only we can apply the rules of inferences, rules of mathematical logic and do something meaningful with the resultant propositions. It turns out that there are two methods of converting your predicates into propositions. The method number one is you assign explicitly, manually the values to your underlying variables, but it is very less interesting.

We are not interested in this method and this is what we were doing till now in all the examples that I have demonstrated. I have manually assigned the values to my underlying variables and convert the resultant predicates into propositions. We will be interested in method 2, which we call as the quantification method. Because we will be interested in representing quantified statements of the form that something is true for all values in my domain, something is true for some values in my domain something is not at all true for any of the values for my domain.

So we will be interested in these three forms of quantifications and we will be discussing about various mechanisms how to deal with such statements and so on. So we will be first starting with two forms of quantifications; namely the universal quantification and existential quantification.

(Refer Slide Time: 08:31)



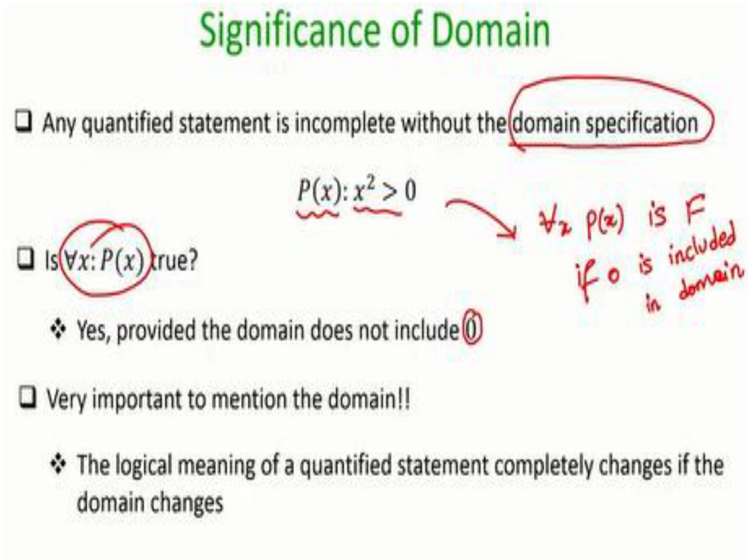
What is a universal quantification? Well whenever we want to assert that a property is true for all the elements of my domain, then I use universal quantification. So very often you might have encountered this notation (∀) for all x in your theorem statements. You encounter this whenever we say that some property is true for all integers all real numbers. So very often we use this notation for all x.

And say we want to say that all integers satisfy something, something, that something, something is nothing but a predicate function. You want to say that some property P is true for all the x values in your underlying domain. If that is the case then we use this notation. So this expression for all x, P(x) is true, if the property P holds for every value x in your domain that means for simplicity assume your domain has m possible values of x.

Well your domain could be infinite as well say your domain could be the set of integers that is possible because you might want to assert a property which is true for all integers. But just for simplicity I am assuming here that my domain consists of m number of elements. Then the quantification for all x, P(x) is logically equivalent to the conjunction of m propositions, where proposition $P(x_1)$ denotes that property P is true for x_1, $P(x_2)$ denotes that property P is true for x_2 and so on and $P(x_m)$ denotes that property P is true for x_m.

That means if this conjunction on your right hand side, if this is false, then for all x, P(x) is not true, that will be false. That means even if you find one counter example or a bad witness, your counter example is nothing but a bad witness, even if you find a bad witness, at least one bad witness for which the property P fails, I can conclude that for all x, P(x) is false.

(Refer Slide Time: 11:11)



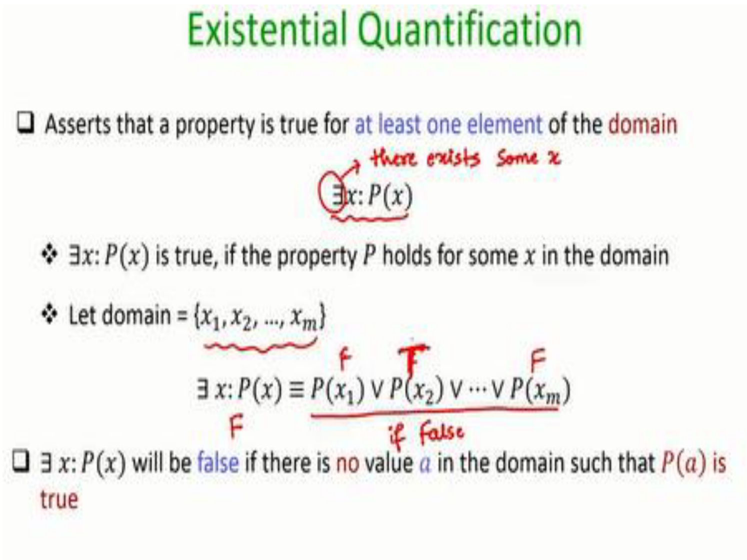
So before proceeding further, I would like to stress here on the significance of the domain here. Whenever we are making quantified statements, it could be any form of quantification, it is very important that you clearly and explicitly say what is the domain of x. Say for instance I define a predicate P(x) this, says that x^2 is greater than zero. This is a predicate function which will be true if x^2 is greater than 0, it will be false if x^2 is not greater than 0.

Now, suppose someone says that, is it the case that for all x, P(x) is true, if someone asks this question, how will you verify that? Well, it depends upon what exactly is the domain, if the domain does not include 0, that means you are considering a domain where x can take any value, except 0, then yes this statement for all x, P(x) is true because the property P namely x^2 greater than 0 will be true for every x except 0.

But the same predicate P(x) will not satisfy the condition for all x, P(x) for all x, P(x) will be false if zero is included in the domain that means your x can take the value zero. So it is very important to explicitly mention what exactly is your domain, if you do not specify your domain then universal quantification does not make very sense. The logical meaning will completely change as soon as you change the domain.

That means for the same P(x) for all x P(x) may become true for one domain, but as soon as you change the domain for all x, P(x) may become false.

(Refer Slide Time: 13:01)

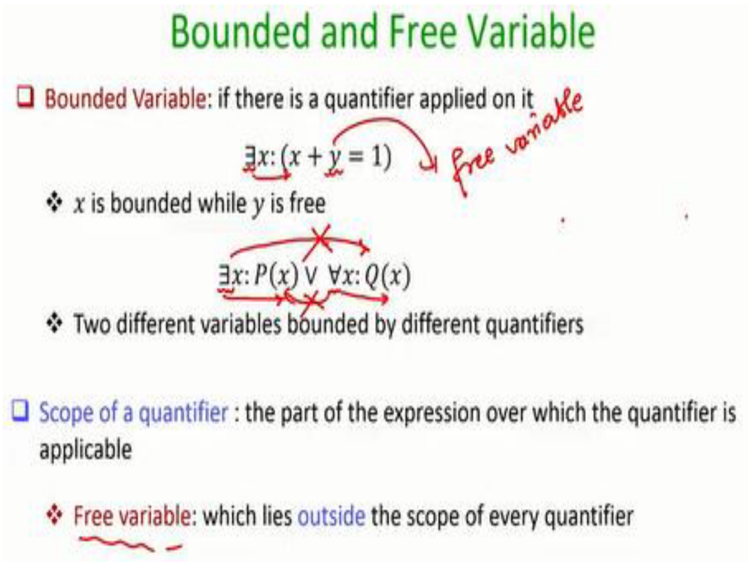


Now, let us go to the next form of quantification which we call as existential quantification and this quantification asserts that a property is true for at least one element of my domain. Here I am not interested to state that my property is true for every value in the domain. I want to stress that it is true for at least one value of the domain, well it might be true for every element of the domain that I am not worried about, I am interested to assert that it is true for at least one element of my domain.

And this is represented by this expression there exists x (∃), so this notation stands for there exists. So whenever property P is true for at least for some x in your domain the expression, there exists x, P(x) becomes true. Again for simplicity assume your domain consists of some m number of elements, then I can say that there exist x, P(x) is logically equivalent to disjunction of this m propositions where the first proposition is property P is true for x_1, second proposition is property P is true for x_2 and so on.

That means if your RHS is false and when can be RHS false? When this disjunction is false and when can this disjunction in your RHS will be false? When the property fails or the property does not hold for any of the x values; that means $P(x_1)$ is false, $P(x_2)$ is false and like that $P(x_m)$ is false, that means for none of the x values in your domain the property P is true, in that case the statement there exist x, P(x) will be false. But even if at least one of the x values satisfies the property P then the statement there exists x, P(x) becomes true.

(Refer Slide Time: 15:12)



Now let us define what we call as bounded and free variables. So a variable is called a bounded if there is a quantifier, which is applied on it. So for example if I write an expression of this form then x is bounded because the quantification there exists is applicable on this x, but what about this y? This is acting as a free variable here. There is no quantification applicable on this variable y whereas consider this expression, there exists x, P(x) disjunction for all x, Q(x).

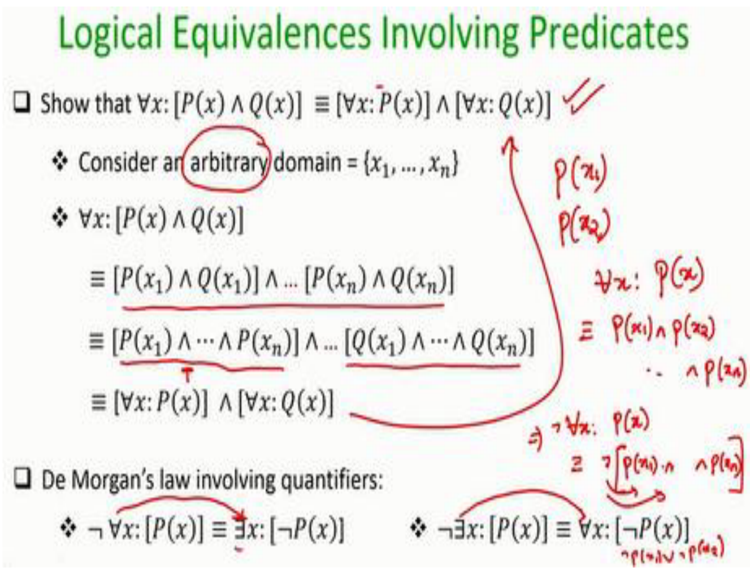
Now question here, am I talking about 1 x or am I talking about 2 x, it turns out there are 2 different variables, the 2 x s are represented by the same x here, which is the common source of confusion here. So there are two different variables and two different quantifiers are applied on it. The there exists quantification is applicable on this x which is the subject for predicate P and the quantification for all is applicable on the second x variable, which is the subject for proposition Q.

So what you can do is to avoid confusion, you can either put x explicitly the brackets to denote that there exist is applicable on P(x) and for all is applicable on Q(x) or it is recommended to use different variables, if you do not want to put parentheses to avoid confusion. Use different variables for different predicate functions, if they are applicable with respect to different quantifiers. What is the scope of a quantifier?

The scope of a quantifier is that part of the expression over which the quantifier is applicable. So for instance, if I take this expression, this example; the scope of this quantifier is only limited to P(x), it is not applicable to for all x, Q(x), no. In the same way the scope of this for all quantifier is applicable only to this Q(x), it is not applicable to P(x). So that means what is a free variable? A variable is free if it is outside the scope of every quantifier, right?

So if I take the first expression here, the scope of this there exists a quantifier is this x, but if I take this variable y, it does not come within the scope of this there exist, because there is no quantification which is applicable over this y and that is why this y is a free variable here. And if you have expressions involving predicate functions where you have free variables, then it is completely ambiguous, you cannot make any meaning out of that.

(Refer Slide Time: 18:53)



So now we can define logical equivalences even for the predicate world, we can define predicates, we can have quantified statements and we can have two different expressions and we can verify whether the two expressions are logically equivalent or not. So for instance if I want to verify whether the expression in your LHS part and the expression in your RHS part, they are logically equivalent or not we can verify that.

And, see I have explicitly used parenthesis here to distinguish the quantification of this for all x and this for all x. So how do we prove logical equivalences involving predicates? Well, what was our definition of logical equivalence in the propositional world? If you have two compound propositions x and y we said that x is logically equivalent to y if and only if x bi-implication y is a tautology.

That was the definition of logical equivalence, that means x and y takes the same truth value, whenever x is true y is true, whenever x is false y is false. But now we are talking about arbitrary domains, arbitrary predicate functions and so on. Intuitively, we will say that my expressions involving predicate functions are logically equivalent if they are equivalent with respect to any possible domain.

That means it should not happen that the two expressions are equivalent for one domain but there is some bad domain for which the two expressions are not equivalent. Even if there is one bad domain for which the two expressions are not equivalent, the overall expressions will not be considered as logically equivalent expressions. But now the problem here is that in this particular case I am not specifying what is the domain of x and I cannot do that.

Because whenever I have specified logical equivalences involving predicates, my predicate could be over any possible arbitrary domain. Because I want to state here that this condition holds, so the way you can interpret this statement is I want to prove that my LHS and RHS are equivalent. It does not matter what is the domain of x that is what is the meaning of logical equivalence in the predicate world.

So you might be wondering that without even talking about the domain, how can I verify whether the LHS and RHS are true for every possible domain, the way we do it, we say that because it has some arbitrary domain, that arbitrary domain could be the set of integers it could be anything and my property P, my property Q could be arbitrary properties, I am not explicitly specifying what are property P and Q.

So I am considering arbitrary domains and arbitrary predicates P and Q and for simplicity I am taking a finite domain, but whatever I am saying here, you can generalize it to infinite domains as well. Now our goal will be to show that with respect to this arbitrary domain and arbitrary predicates P and Q, the LHS expression and RHS expression have the same truth value. If we do that then that establishes the logical equivalence of this expression, because I am considering an arbitrary domain for which I am showing the equivalence here.

So let us start with the LHS part here, what was your LHS part? The LHS part says that for all x conjunction of P(x) and Q(x) is true. Now if you remember for all x, P(x) conjunction Q(x) is true if the property P and Q is simultaneously true for every x value in your domain that means this conjunction should be true. Now what I do is, I just shuffle the terms here, I bring all the P terms together and all the Q terms together.

So remember this $P(x_1)$, $P(x_2)$ all these are propositions. So what I am saying is that for all x, P(x) and Q(x) is true, that means $P(x_1)$ and $Q(x_1)$ is true, the proposition $P(x_2)$ and $Q(x_2)$ is true and so on and now I am shuffling around the individual propositions. Now if the conjunction of all the P prepositions is true that is equivalent to saying that for all x, P(x) is true and the same way if the conjunction of all the Q propositions are true that is equivalent to saying that for all x, Q(x) is true and this is nothing but your RHS expression.

In the same way we can prove De Morgan's law involving quantified statements, so for instance we can prove that if you have a negation symbol outside for all x, you can take the negation inside and for all become there exist and you take the negation and put it before your predicate P(x) and the dual property will be if you have a negation before there exist, you can take the negation inside, there exists become for all and property P is replaced by negation of property P.

You can prove that using the same thing, so you can say that for all x, P(x) is equivalent to $P(x_1)$ and $P(x_2)$ and so on. That means negation of for all x, P(x) is nothing but negation of this entire conjunction and then you can apply De Morgan's law of propositions and you can take the negation inside each of them and the conjunctions get converted into disjunction, so you get negation of $P(x_1)$ disjunction negation of $P(x_2)$ and so on.

And since we want to assert that negation $P(x_1)$ is true, or negation of $P(x_2)$ is true and so on, that is equivalent to saying that the negation of P property is true for at least 1 value of x in my domain. Same thing you can do for the second dual form of De Morgan's law. So that brings to the end of this lecture, the references for this lecture is Rosen’s book.

And to summarize in this lecture, we started discussing predicate logic, we saw the motivation of predicate logic and we discussed how we can represent quantified statements. We saw two forms of quantifications namely the universal quantification and existential quantification. Universal quantification is true when the property P is true for all the values in your domain. Existential quantification is true when the property is true for at least one value on your domain.

And, we saw some logical equivalences involving statements having predicate functions. Thank you.