Artificial Intelligence: First Order Logic: Proof Systems

Prof. Deepak Khemani

Department of Computer Science and Engineering

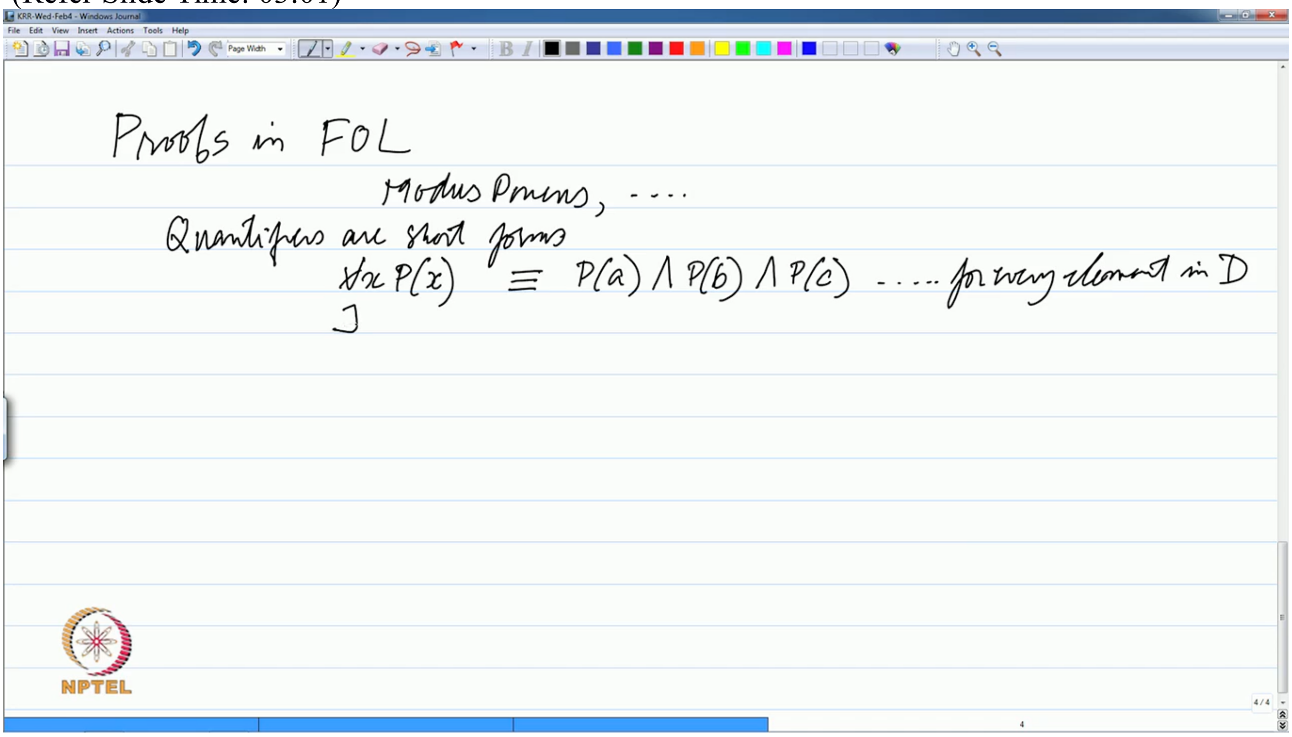
Indian Institute of Technology, Madras

Module – 03

Lecture - 04

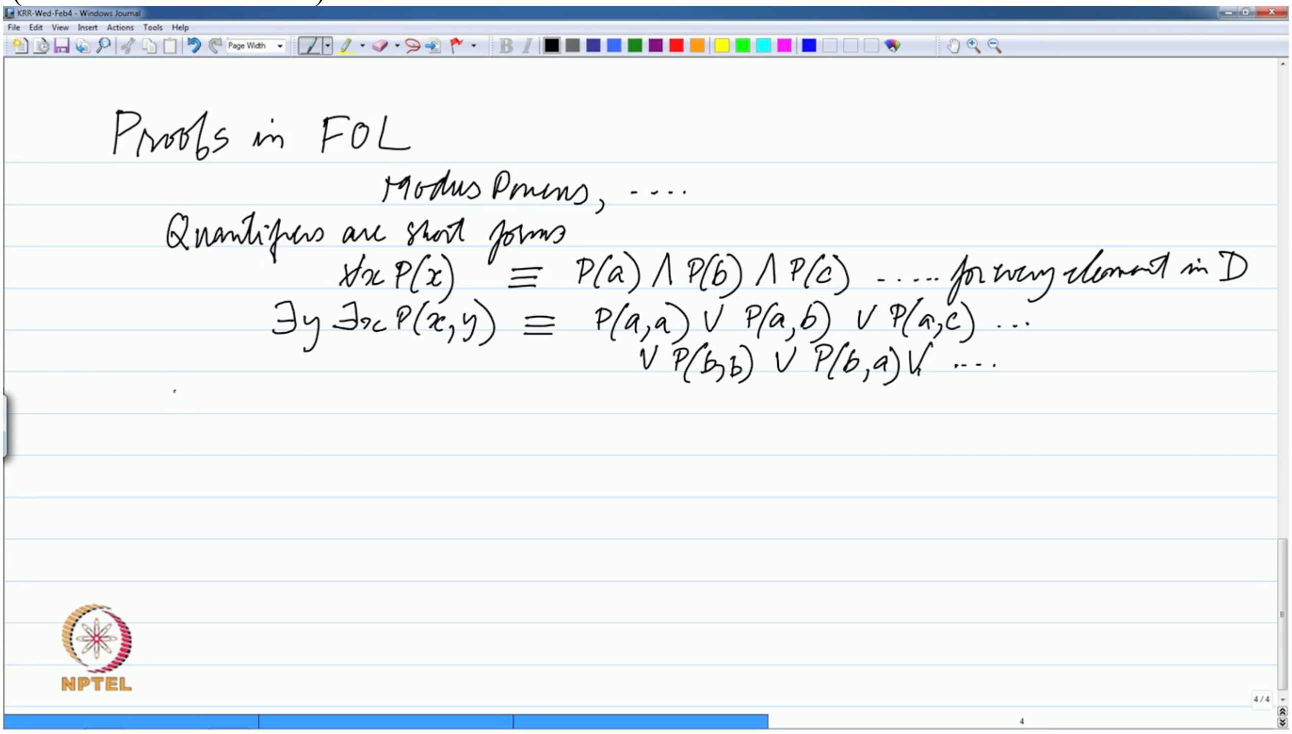
Okay so we have just defined the semantics of first order logic and now we want to move on to proof systems in FOL because eventually that's what we are interested in, writing programs which can produce truth statements. So the first thing to observe is that all the rules that we have studied so far, things like modus ponens, they still hold in FOL as well. So we carry forward everything that we have in propositional logic to first order logic. But what is different in first order logic is that we have quantifiers, we have predicates and we want to talk about how to handle those. Now the first thing to observe is that these quantifiers are essentially short forms. So when I say something like, okay so let me first state this. So if I were to make a statement like for all x Px where P is some formula or predicate then I am essentially making a statement which is logically equivalent to saying that P a where a is some constant of my domain or some element of my domain and P b where b is another element of my domain and P c and so on for every element which is true of all formalisms generally. It doesn't have to be arity one, could be P x y and things like that. So the first thing to observe is that a universal quantifier is a short form for making a statement which is true of every element of domain. So I could have written the larger conjunct especially if my domain is finite so let's say if I am talking about this particular class which has twenty odd students, I could have written some statements for each student that would be equivalent to saying for all x something like that.

(Refer Slide Time: 03:01)



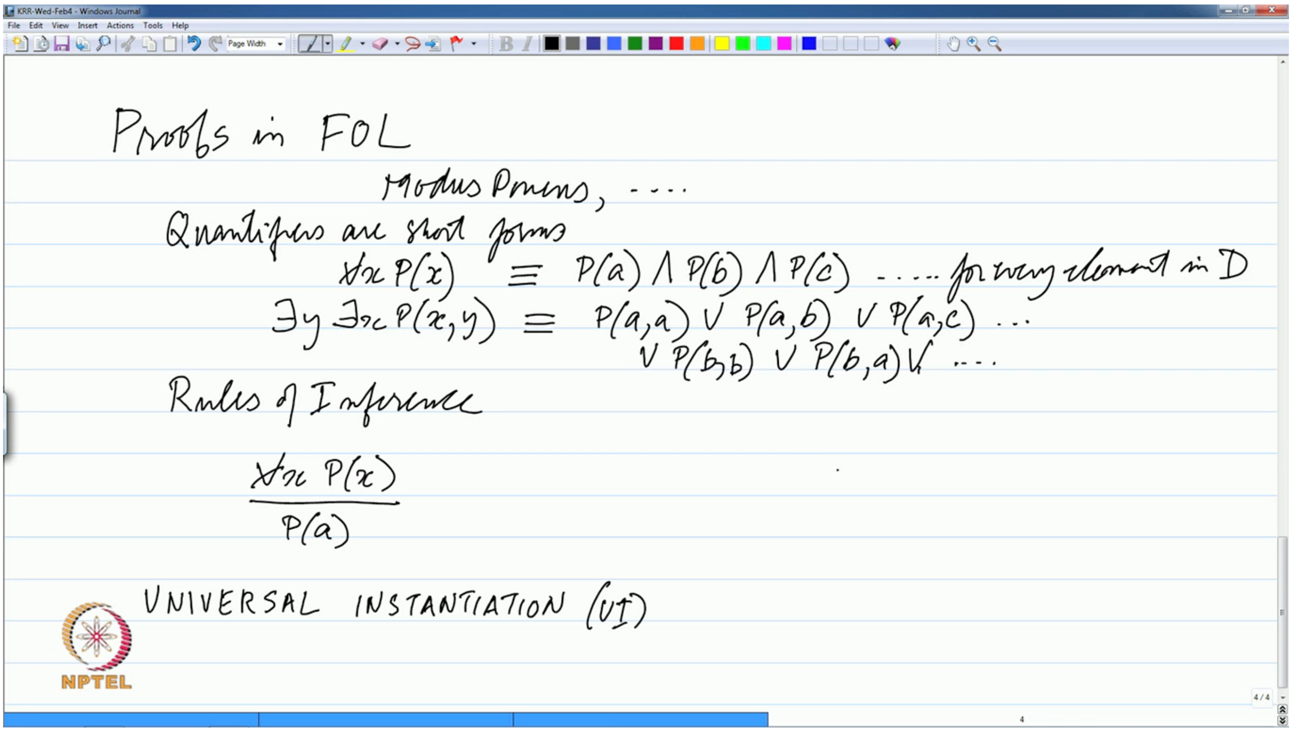
Likewise there exists x P x or let me use a predicate with two variables so let's say there exists y, there exists x is equivalent to saying that P a or P a b or P a c or P b b and so on .. It's just a large disjunct essentially which of course explains the semantics of these quantified sentences quite easily that a sentence like for all x P x will only be true when it is true for every element in the domain. Whereas a sentence like there exists an x P x so for example there exists a number which is even will be true as long as for some element it is true essentially. So this will give us insight into the two rules of inference that we need to FOL essentially.

(Refer Slide Time: 4:32)



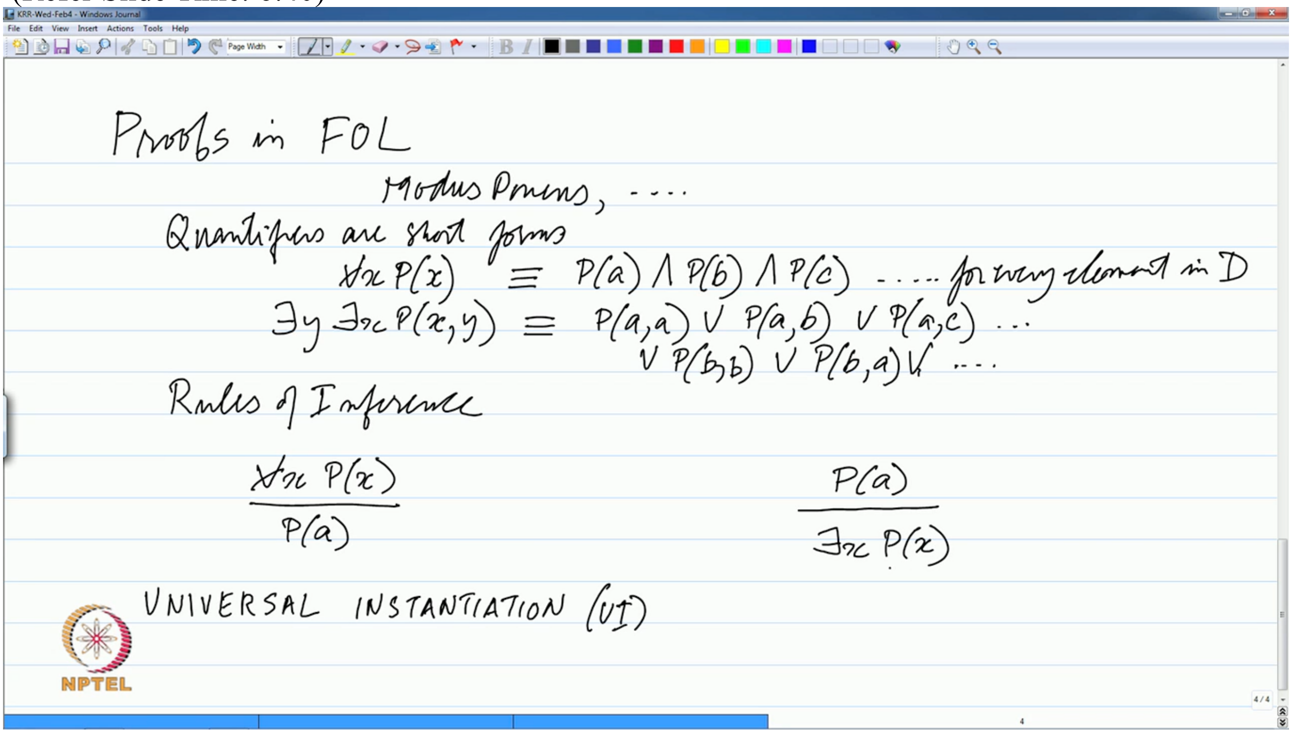
So essentially there are two rules we need, one is, and I will illustrate them using predicates of arity 1 but they carry forward to predicates with more arguments as well. So first rule says that if you have a formula of the kind for all x P x then you can infer P of a where a is some constant in the domain. Now you can see that if you look at the fact that for all x P x is just an abbreviation for a large conjunct, this is just an incense of simplification in some sense essentially because in propositional logic we said that if P is true and Q is true then you can infer that P is true essentially. This rule is called universal instantiation and its one of the most common rules that we use in first order logic. Essentially it says that for x you can substitute anything. You can even substitute a term which is let's say some function of x or something like that doesn't matter, some function of a function. Anything you put in, any term you give as an argument to this predicate this formula must be true so you can substitute it.

(Refer Slide Time: 6:26)



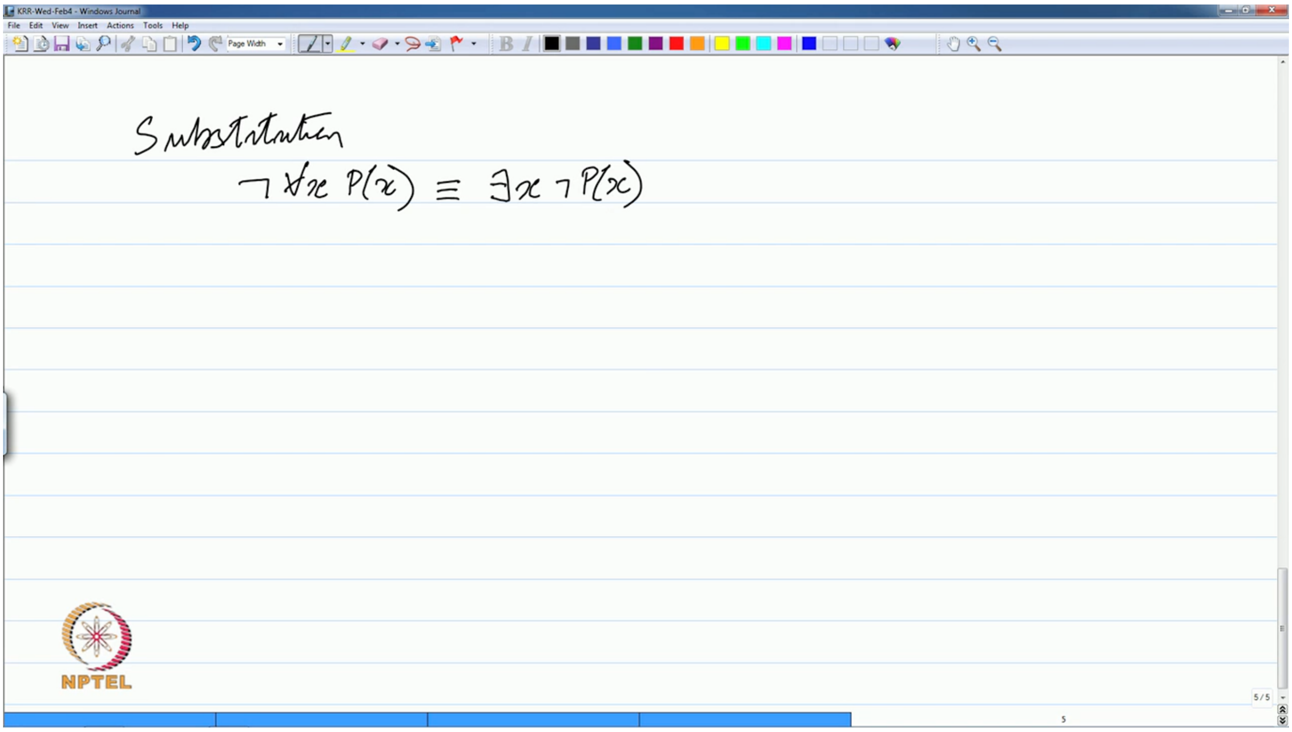
The other rule is opposite which says that if you are given that a predicate is true for a certain element a then you know that this formula is true and this formula says there exists x P x

(Refer Slide Time: 6:40)



Because this formula is only saying that there exists at least one element for which this formula must be true and we already know that it is true for a. So there exists a x P x must be true. This rule is called Generalisation. Then just as we had rules of substitution in propositional logic, we also have rules of substitution in FOL which means you can substitute one formula for another formula and you might be familiar with these kinds of problems. So the two most common rules that we will use are the fact that if you have a negation of a universally quantified formula then that is logically equivalent to a formula which is an existentially quantified formula which a negation inside.

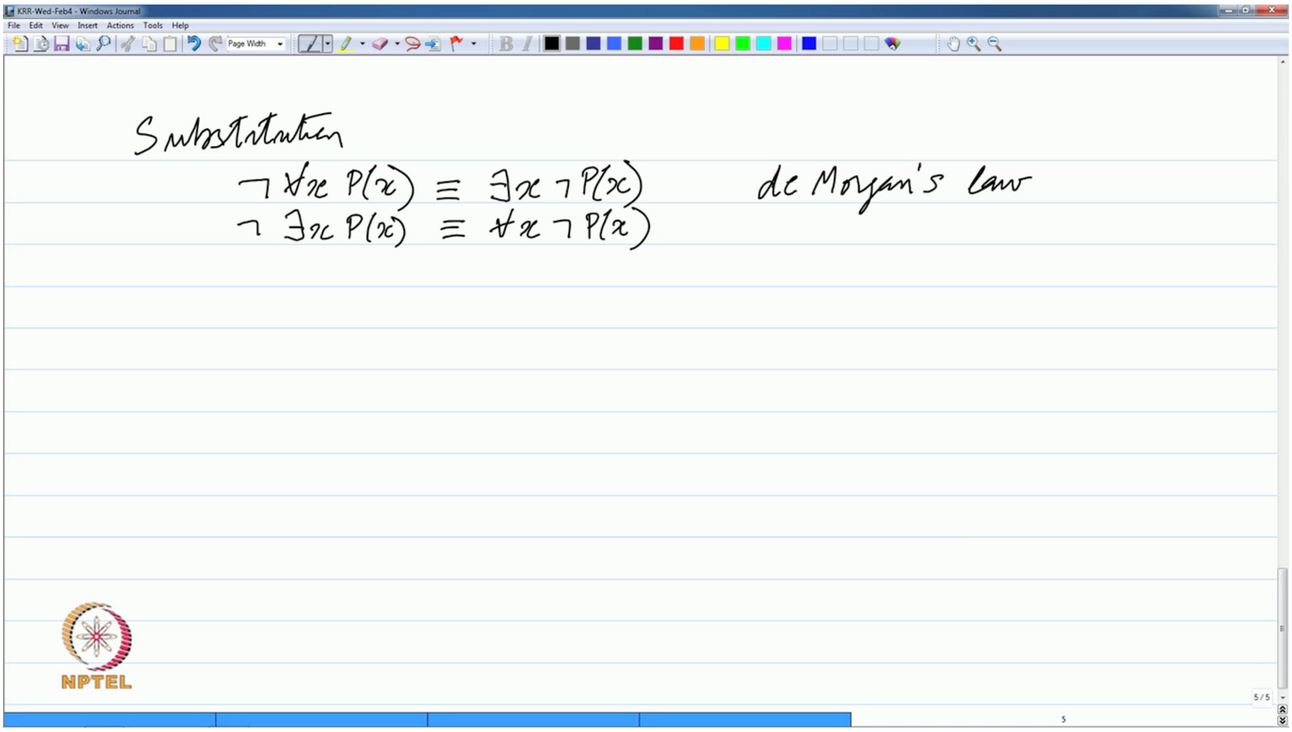
(Refer Slide Time: 8:17)



In other words, you can move the not inside and when you move it inside it changes the nature of the quantifier essentially. As you would expect this is called de Morgan's law and it's easy to understand if you remember the fact that a universally quantified formula is basically a large conjunction and we already know from the propositional logic that when you take a formula which is a conjunct and take a negation of that and when you push the negation inside then it becomes a disjunct. So that's what is happening here essentially.

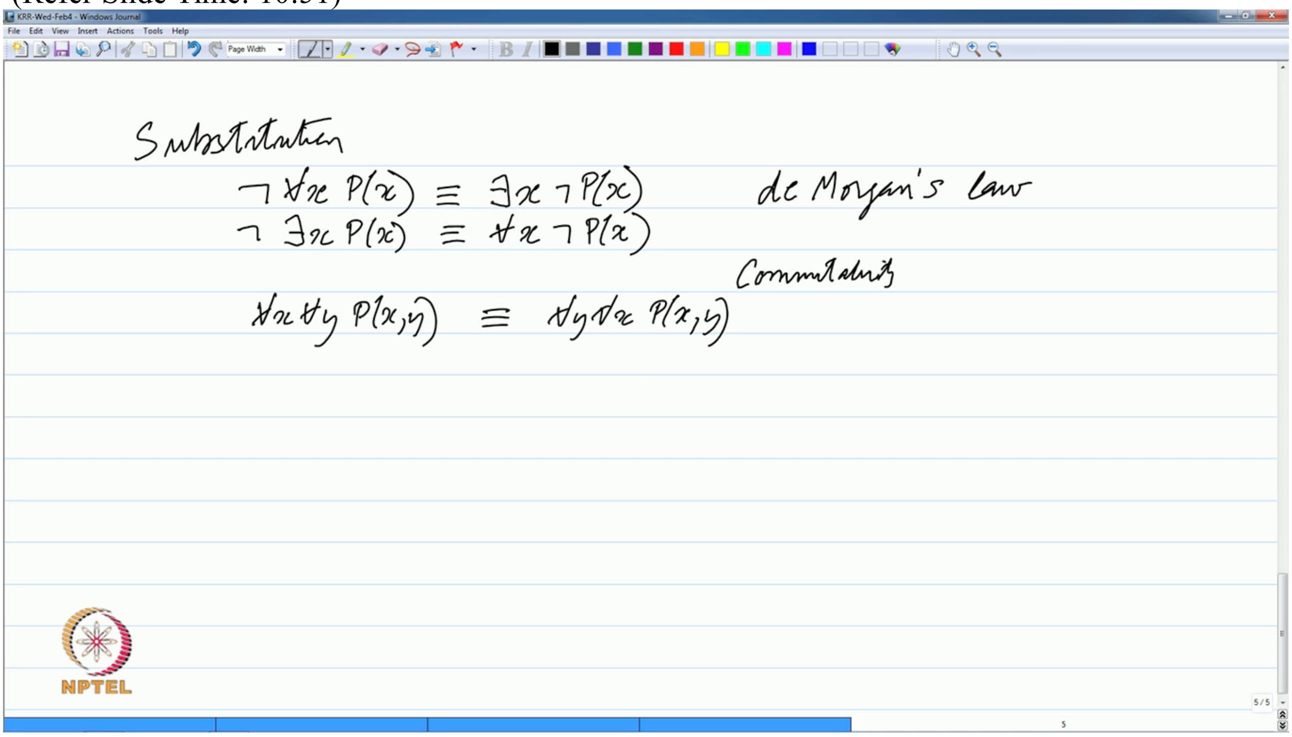
Or let's say P stood for mortal essentially, so P x says that x is moral essentially. This sentence is saying that it is not the case that everything is mortal which of course is logically equivalent to saying that there exists at least one element which is not mortal. So to say that it's not the case that everything is mortal is equivalent to saying that there exists at least one element which is not mortal. And likewise if you move the across the existential quantifier we get universal quantifier.

(Refer Slide Time: 9:53)



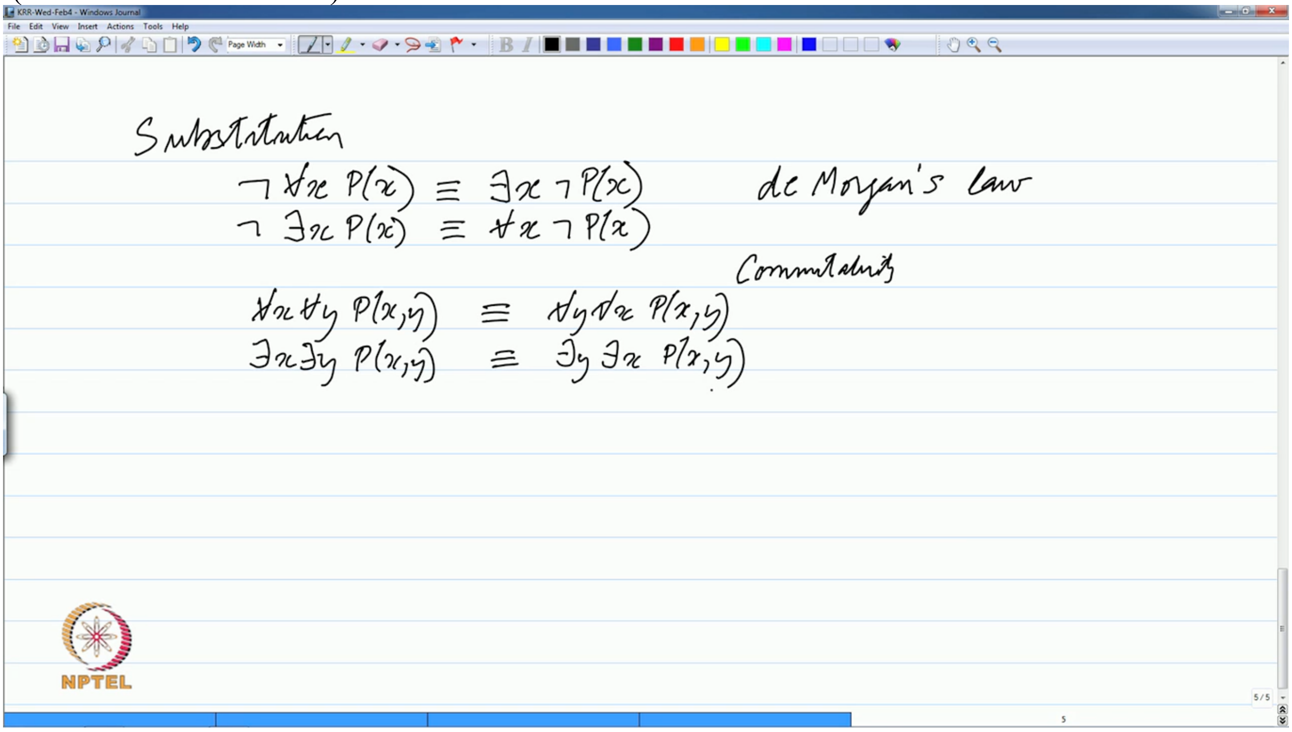
So if you were to say now let's say that P were to stand for immortal. Then you are saying on the left hand side that there is no one who is immortal. There does not exist anyone who is immortal. Its equivalent to saying that everyone is not mortal, everyone is not immortal essentially. And then you have the usual rules which are like commutativity and so on. So if you have a formula for all x for all y P x y you can write equivalently for all y for all x P x y. It doesn't matter as long as the quantifer is of the same nature essentially. With a universal quantifier you can change the order.

(Refer Slide Time: 10:51)



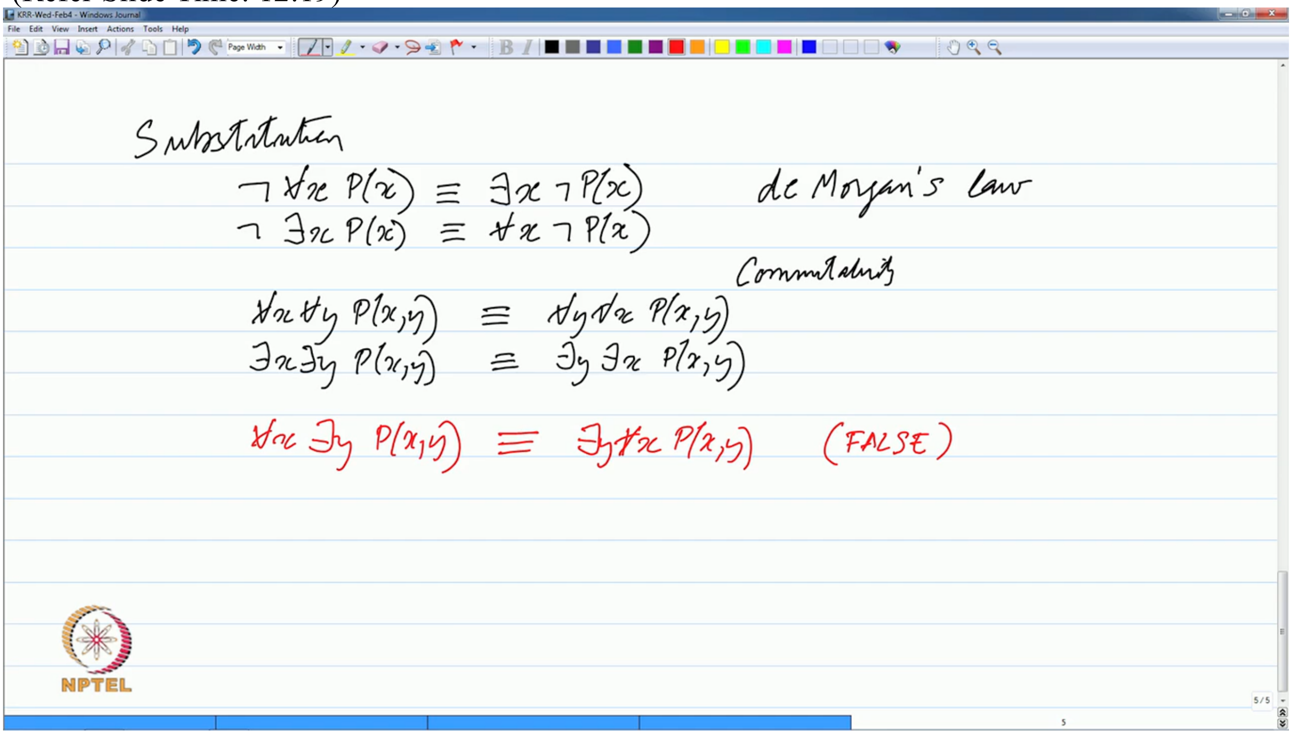
Why do we need these rules because we are talking about now formal systems, you are talking about manipulating formulae to either produce a formula we are interested in or produce a proof of some kind essentially? Then I will encourage you to look at, so if you look at this formula for example so the same thing holds for existential quantifier

(Refer Slide Time: 11:30)



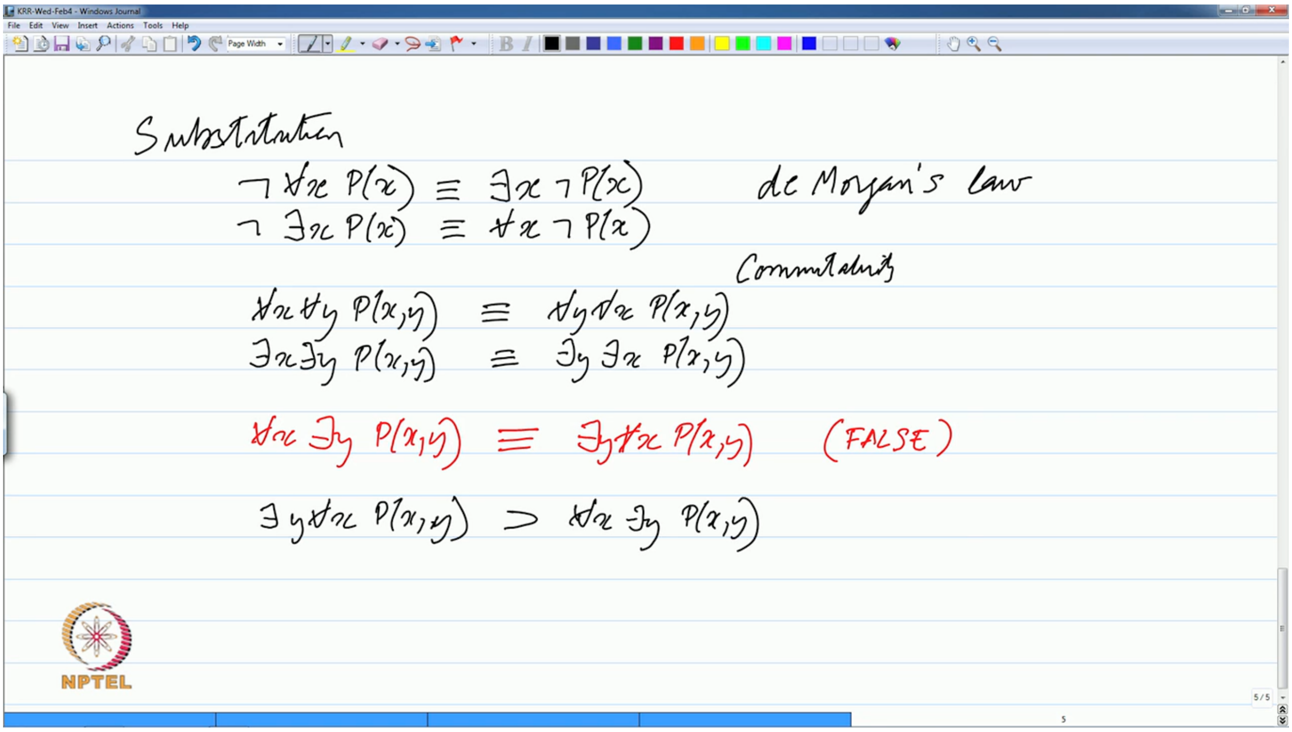
But if I were to treat, look at the formula like this for all x there exists a y P x y and if I were to try to say that there exists a y for all x P x y which means I try to interchange the two quantifiers then this formula is false, it's not true. So you cannot use it as a rule of substitution that if you have the left hand side for example, you cannot replace with the right hand side.

(Refer Slide Time: 12:19)



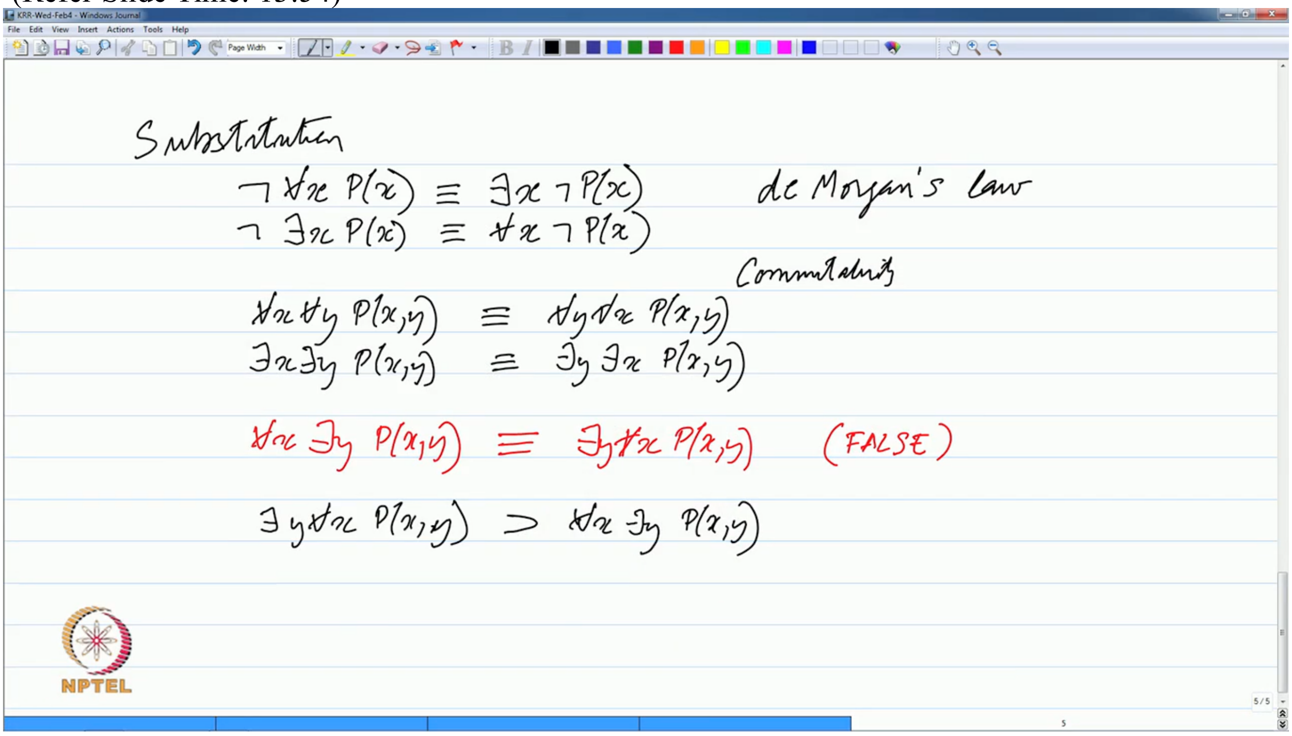
In this particular example it so happens that if you have the right hand side you can still replace with the left hand side. But that means you can write a rule of inference which says that there exists a y for all x P x y implies for all x there exists a y P x y.

(Refer Slide Time: 12:50)



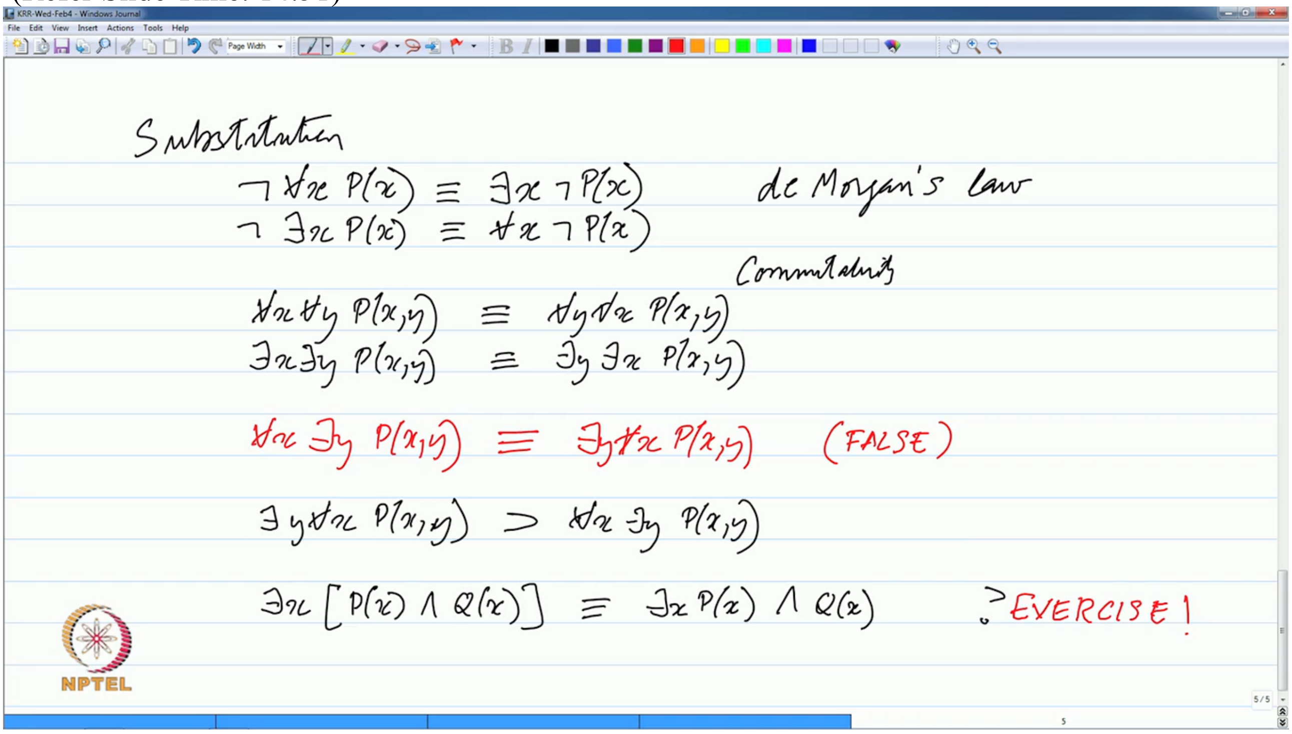
If you look at the left hand side, let us say P stands for admires. So P x y stands for the fact that x admires y. So the left hand side is saying that there exists some y who everyone admires. There exists a y such that for all x P x y is true. So let's say Sachin, a non-controversial example, you can say that everyone admires Sachin which is true essentially. The right hand is saying that everyone admires someone, for all x there exists a y such that P x y is true. Now since we know that eveyone admires Sachin, this right hand side is true because for every person there is atleast one person whom we admire. So certain things hold in one direction.

(Refer Slide Time: 13:54)



So I will encourage you to look at quantified formulae where you have two quantifiers or maybe you have just one quantifier and you look at formulae like this. There exists an x , this is an exercise P x and Q x is equivalent to there exists P x and Q x. So I am leaving it as an exercise. So look at both the quantifiers, universal quantifier and the existential quantifier, look at these common conjuncts which is and, or and try to see which sets of formulae are equivalent.

(Refer Slide Time: 14:51)

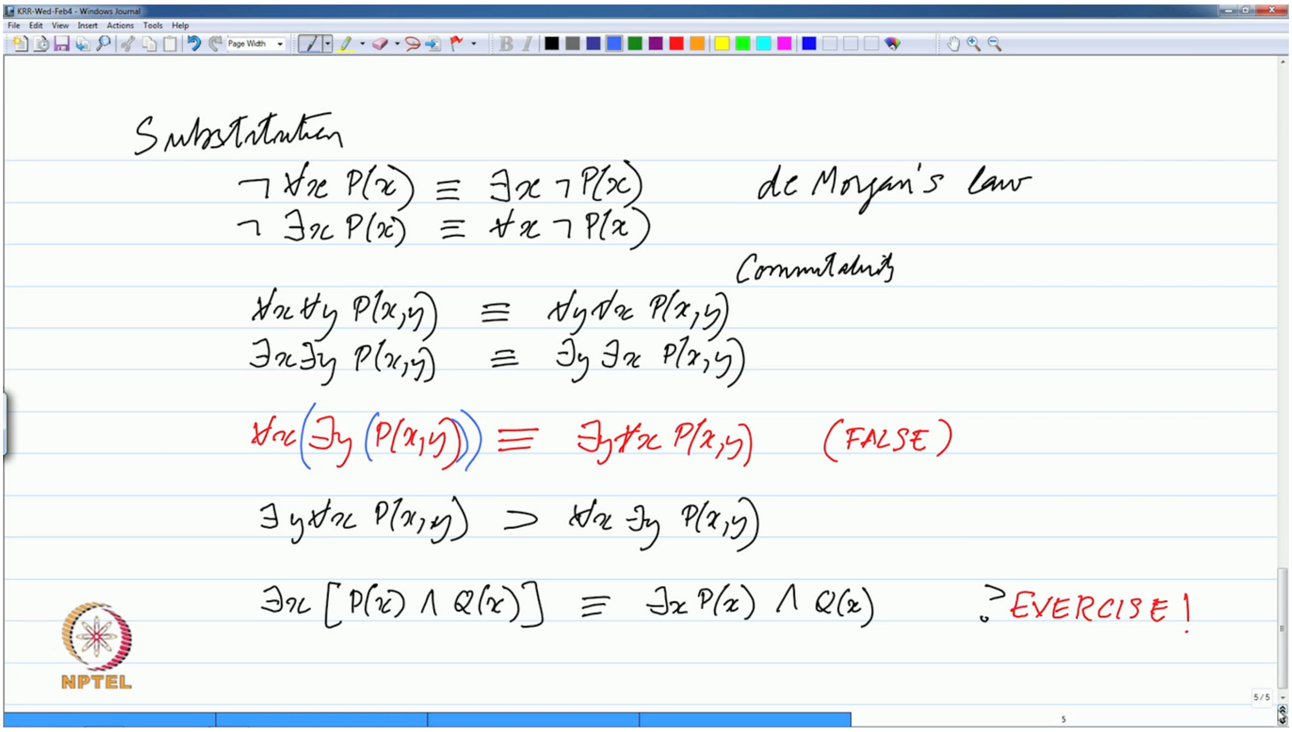


So let me try to give you an example, so if you say for example P stands for the fact that a number is even and Q stands for a number is odd essentially. Then if there is a formula which says that there exists a number which is even or odd and I say the other side is there exists a number which is even or there exists a number which is odd then you can see that both the sides are true and both the sides can be replaced with each other. So I wanted to somewhat study these formulae and look at the combinations of the AND, OR and the two quantifiers essentially.

question from audience No, no, no we have not talked about precedence at all because we have used brackets everywhere. Wherever it is clear that brackets can be thrown away we will throw them away. So we never say things like you know multiplication has a precedence over addition. That is only if you don't use brackets, if you use brackets you don't need precedence.

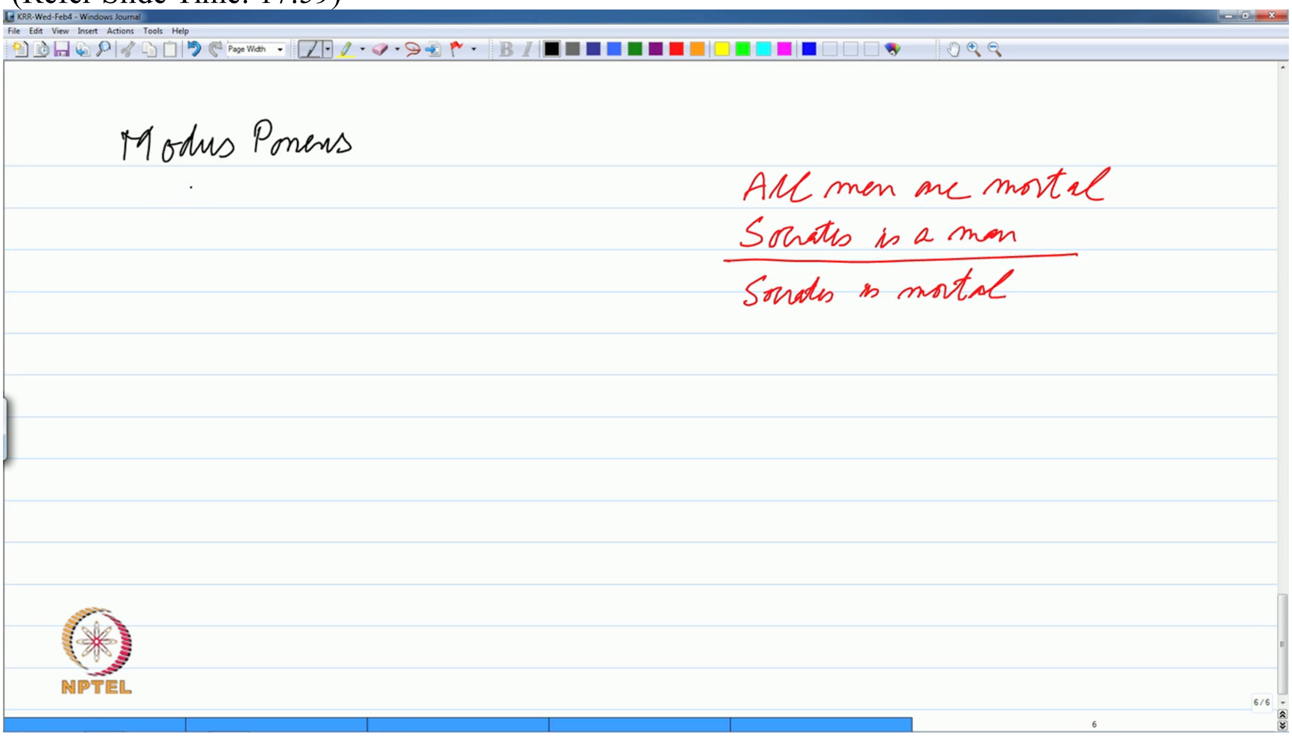
question from audience
There is no notion of precedence. The scope of a quantifier is whatever follows the quantifier. I could have put brackets essentially. So for this statement for all x there exists a y, if I were to really put brackets then I would first need to bracket which says that this is the scope of for all x and then I would say that whatever follows this is a scope of existential quantifier. That is implicit. If you remember when I had defined the syntax I had put brackets but in practice, we don't really put brackets unless there is a need to.

(Refer Slide Time: 16:54)



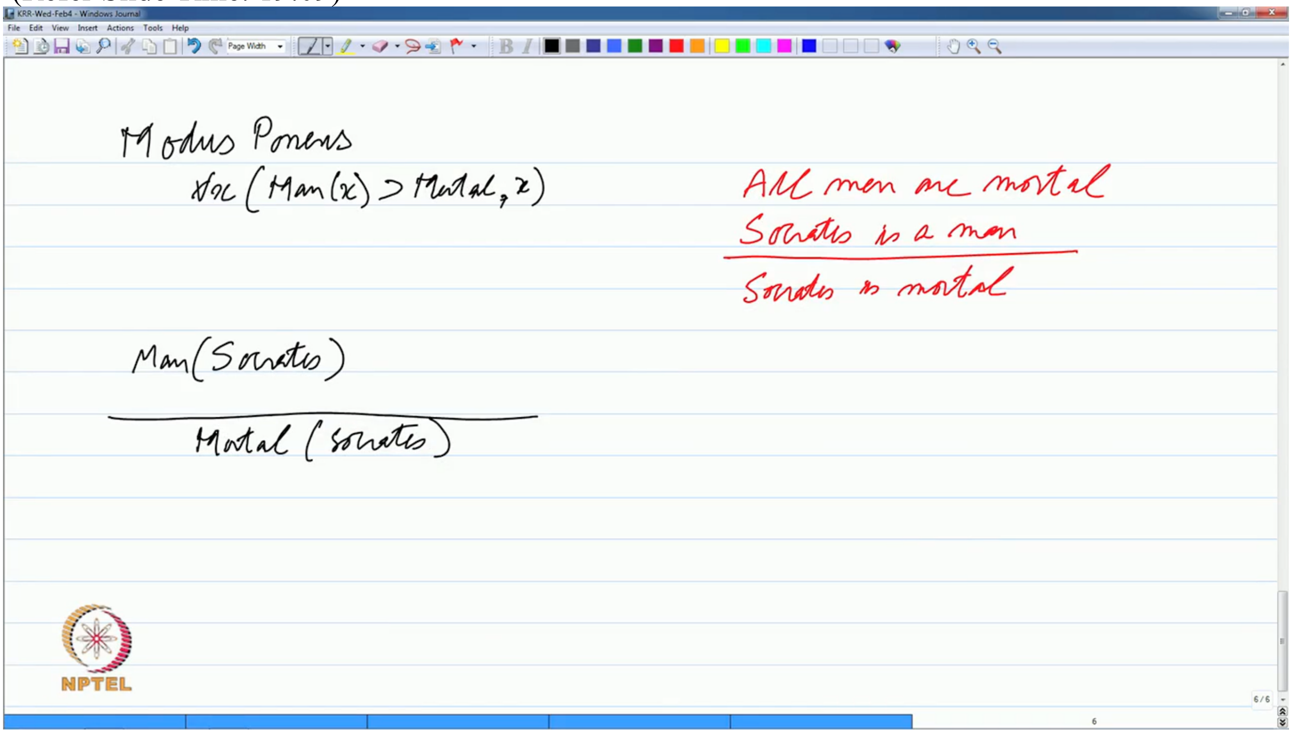
Okay so these are the kinds of rules of inference that we need. Now let's try talking about proofs now and let's take our favourite rule of inference which is modus ponens essentially and let's work with our favourite example. Since we have started the course we have not been able to prove that Socrates is mortal. Today let's try and do that. What are we given? We are given the example that all men are mortal. Then we are given Socrates is a man and we want to be able to prove that Socrates is mortal. So from the semantics of the language we know that this is indeed entailed.

(Refer Slide Time: 17:59)



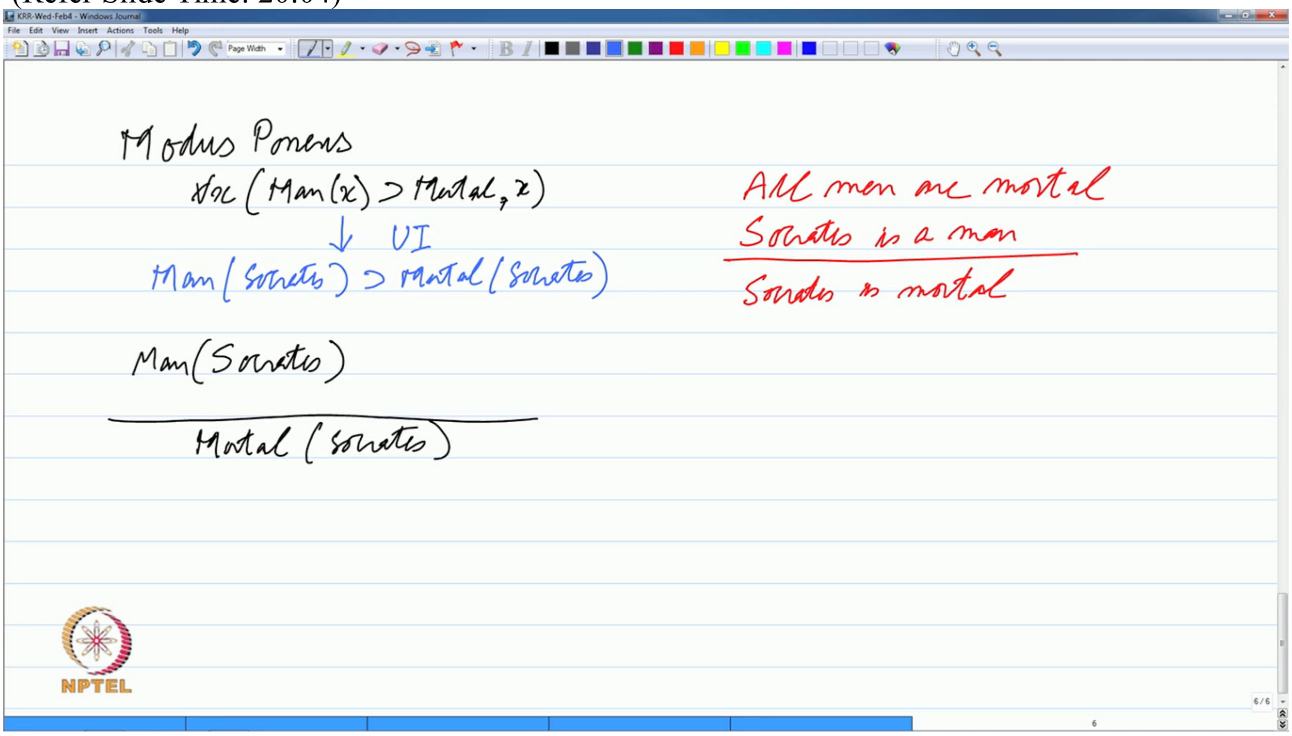
So let's write this in first order logic. For all x Man x implies Mortal x. And let me a little bit of gap because i need to make a couple of inferences. Then we are given Socrates is a man which we express as Man Socrates. And we want to infer that Mortal Socrates. We were not able to do this in propositional logic because if you put each of these three sentences as propositional variables there is no connection that you can see between them. But now we have a richer language and we have more rules of inference that we can exploit essentially.

(Refer Slide Time: 19:09)



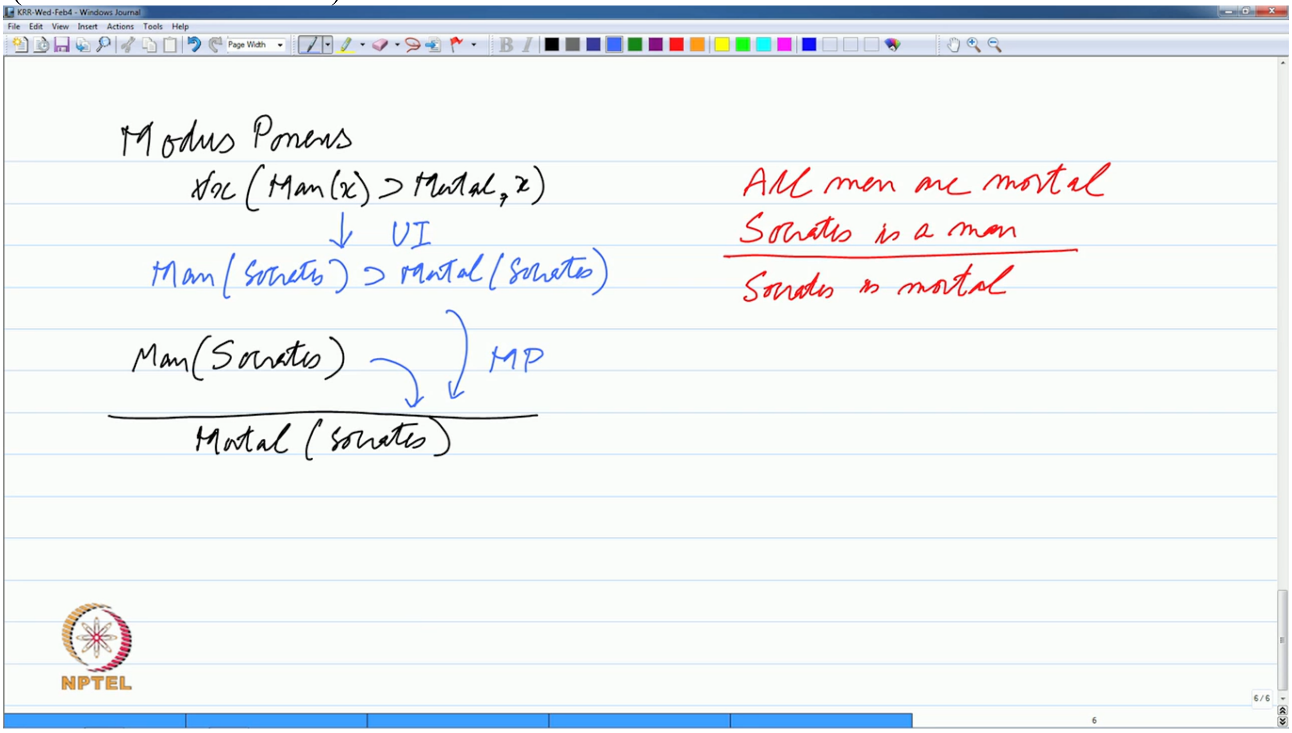
So what do we do? We first apply the new rule that we have just learned which is the rule of universal instantiation. This rule says that in any formula which has a quantified variable, universally quantified variable x, I can replace x with any thing I want from the domain essentially. In particular i can replace x with Socrates so I get a new formula, a new sentence which says that Man Socrates implies Mortal Socrates.

(Refer Slide Time: 20:04)



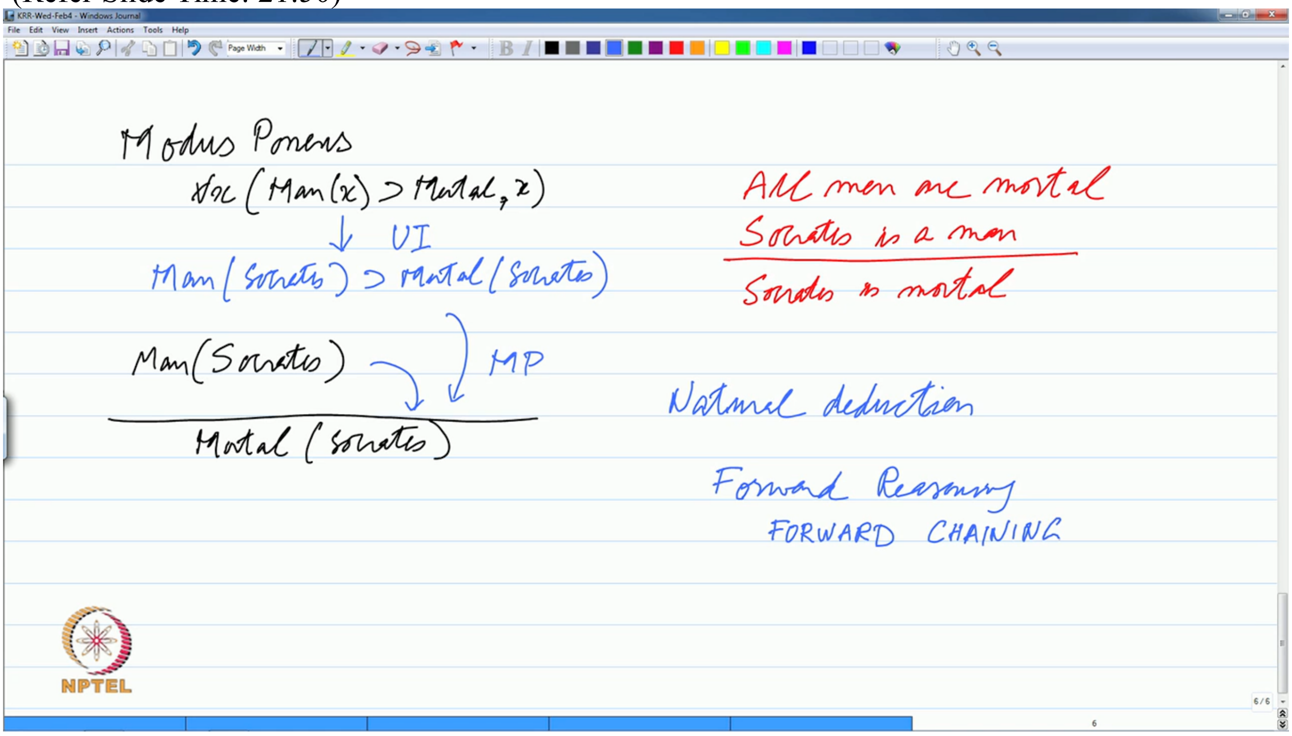
Now you can see things are simple. I have a statement so if I put Man Socrates as P and Mortal Socrates and Q then I have P implies Q and I will P below that so I can infer Q which is the rule of modul ponens that we inherited from propositional logic. So we can just simply apply that, using modus ponens and we have a proof for Mortal Socrates. We have generated the formula that Socrates is mortal and that is the notion of a proof.

(Refer Slide Time: 20:44)



So this form of reasoning as you remember is called natural deduction. So we will start with natural deduction and we will show that natural deduction or direct proofs as we call them are not complete. They were not complete in the case of propositional logic and they are not complete in the case of first order logic. But we will start with that because that is a very useful system to have. We will also call this as forward reasoning. And in particular we will use the term forward chaining which is a natural form of forward reasoning and I will shortly describe what we mean by this.

(Refer Slide Time: 21:50)

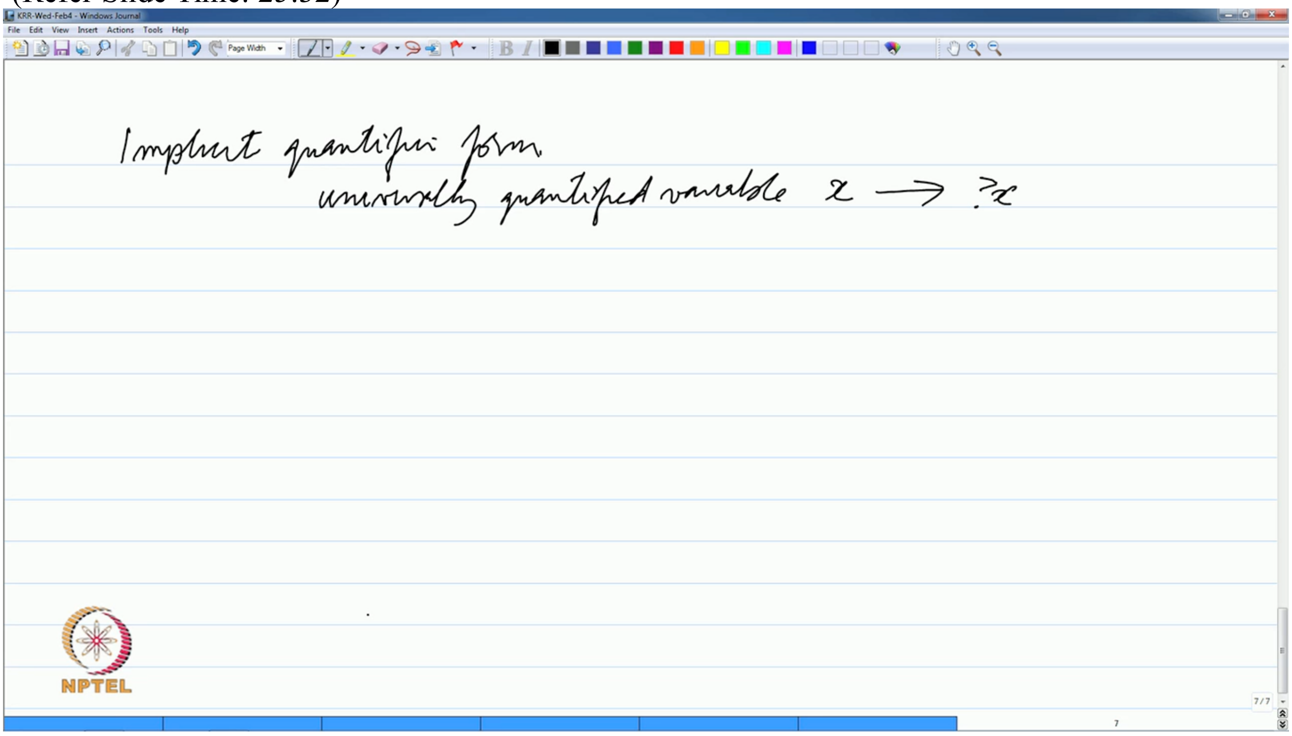


Now one problem here is that there is a guess symbol. Of course we can make informed guesses as to what to substitute for a variable but nevertheless remember we are trying to write programs do this essentially and if there is a whole set of statements we are working with and a whole set of guesses we have to make then it will not be such a simple task. So there is a problem with forward reasoning that we have to make a guess essentially. The other slightly smaller problem is that you have to, remember we are talking about writing programs that you have to process the quantifiers you know and you know handle them, pass them, prove, which can be more work in terms of processing essentially.

So what we will do is we will look at a variation of this rule which is used in forward chaining and which is to use implicit quantifier form which will also address this problem of guessing essentially. Because it is clear when we look at the target that we want. What is the target that we want? The target is this instance of this universally quantified statement where x is substituted with Socrates. It's clear by looking at what you want to prove that we are talking about Socrates essentially. So we want to basically associate the fact that Socrates is a man to Socrates being mortal. So in this implicit quantifier form we do not write the quantifiers. So today we will only look at the universal quantifier because a lot of knowledge that we want to store is going to be universally quantified.

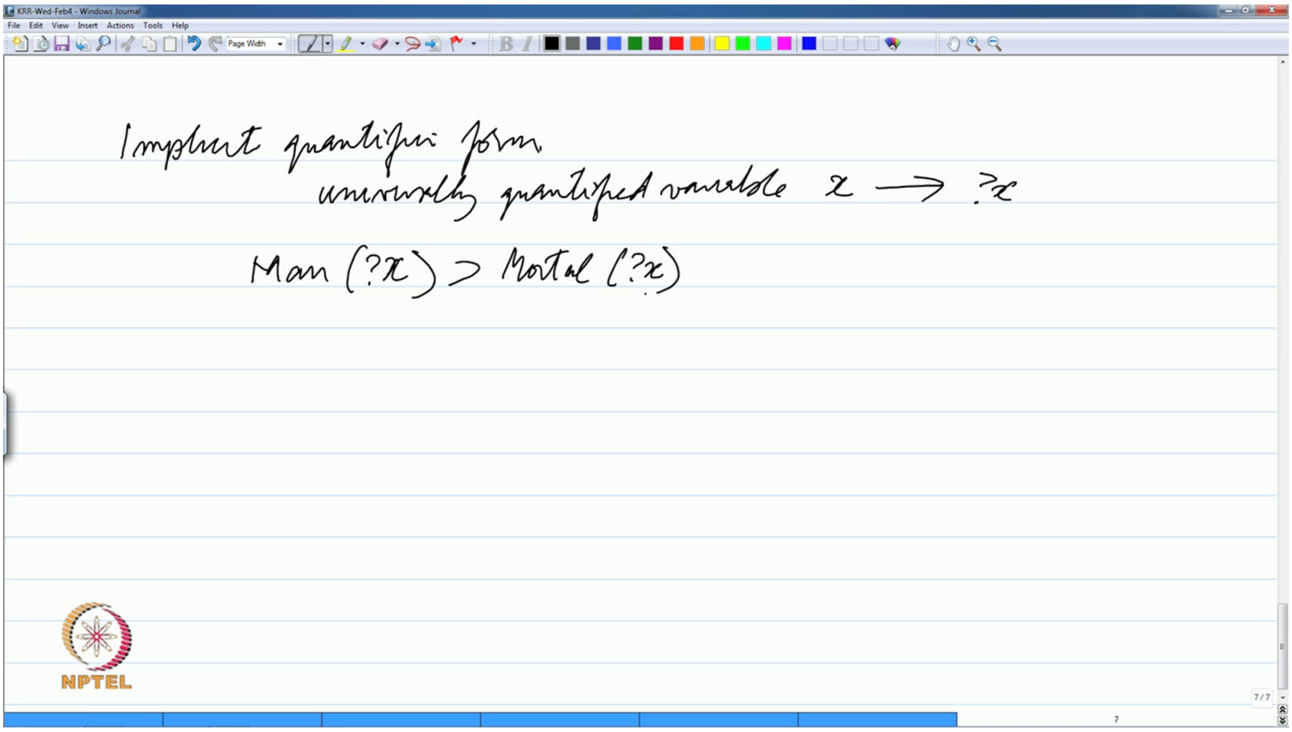
You know things like all men are mortal, all students are bright, all leaves are green, you know all kinds of relations between things which are universally quantified. And we first focus on that. We will look at existentially quantified variables a little bit later. So what we will do is we take a universally quantified variable let's call it x and replace it with distinct form which will make us kind of remember the variable which is universally quantified. So what we will do is we will just put a question mark below the variable

(Refer Slide Time: 25:32)



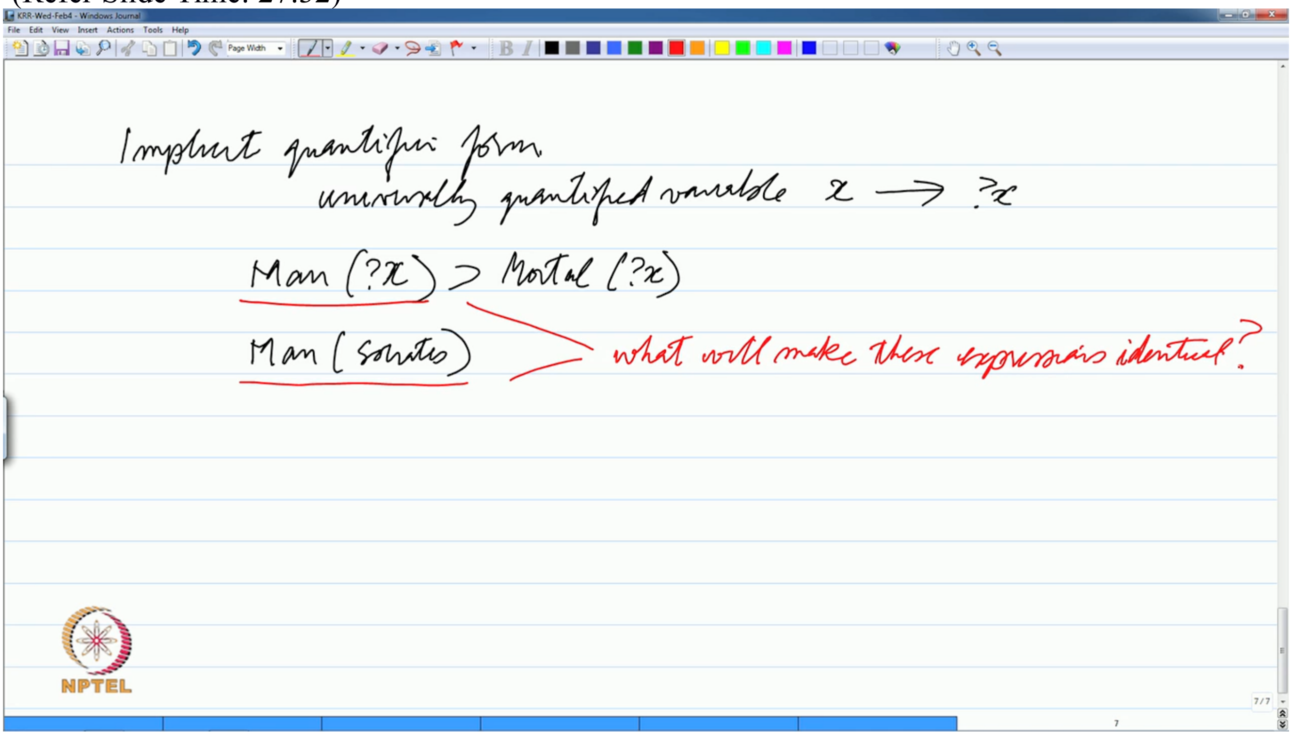
And keep in mind when we are writing our programs that the question mark stands for the fact that x is universally quantified. So instead of saying that for all x Man x implies Mortal x we will now write this as Man (question mark) x implies Mortal (question mark) x. This is just a change of notation.

(Refer Slide Time: 26:11)



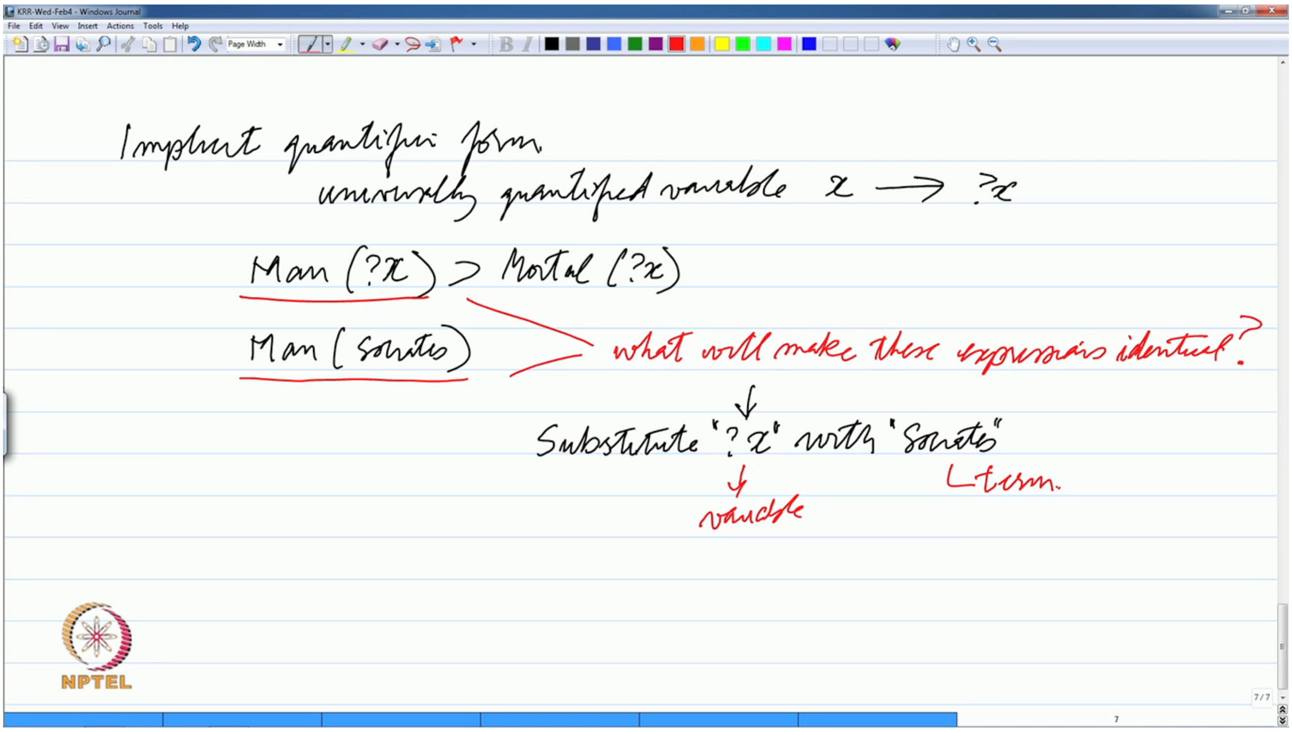
It doesn't change the sentence; the sentence is still the same. It's just that we are not writing the quantifier and we are remembering the fact by putting a question mark below the variable essentially. So every time you see a variable with a question mark you must remember that it is universally quantified. So i can still apply universal instantiation to this for example and produce a formula which is true for Socrates, but we will not do that. What we will do is we will now try to apply directly. We will ask the question as to what is it that will make these two formulas the same. I will just use the term expressions and then you say it identical

(Refer Slide Time: 27:32)



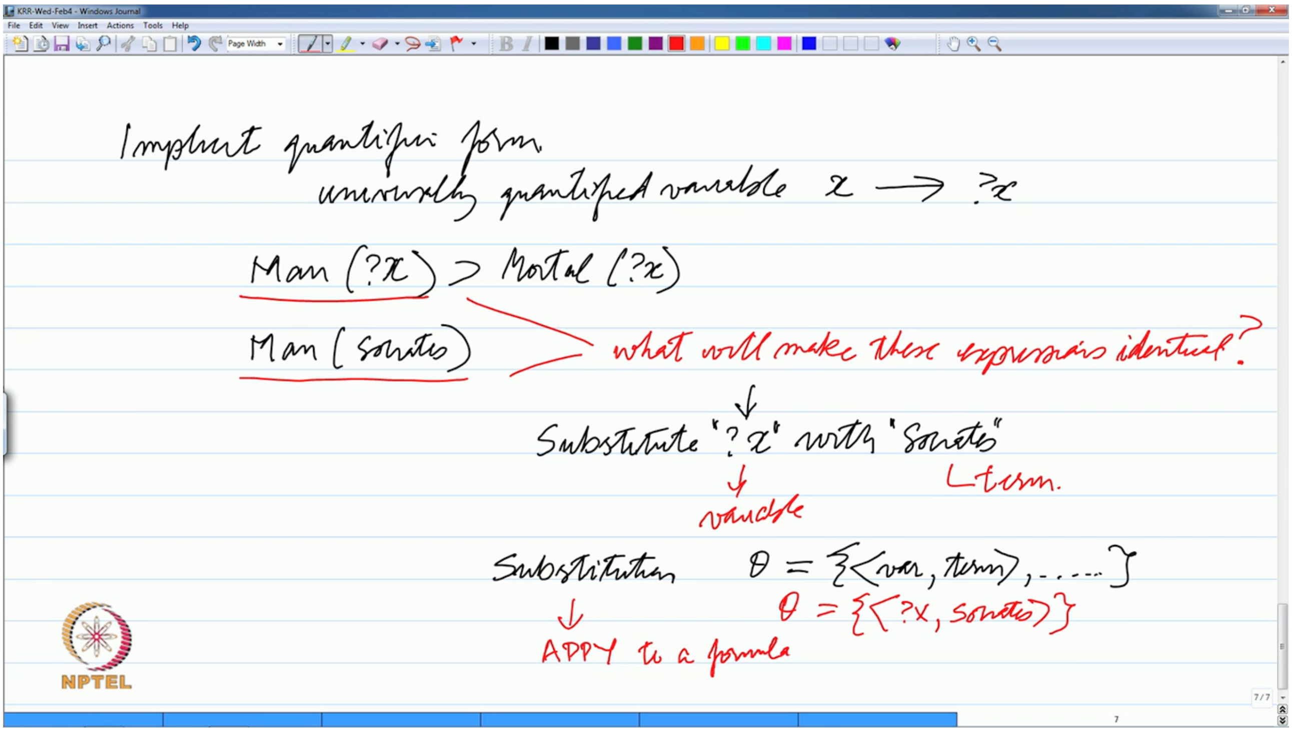
And the answer to this is a substitution x with Socrates. I use not x so substitute the string not x with the string Socrates. So again you must keep in mind that this is a constant in my language so I can substitute it with a constant. In fact, I can substitute it with any term essentially in this case it happens to be a constant. And this is a variable. So i can substitute a variable with a constant or any term essentially. So let me actually write term here

(Refer Slide Time: 28:46)



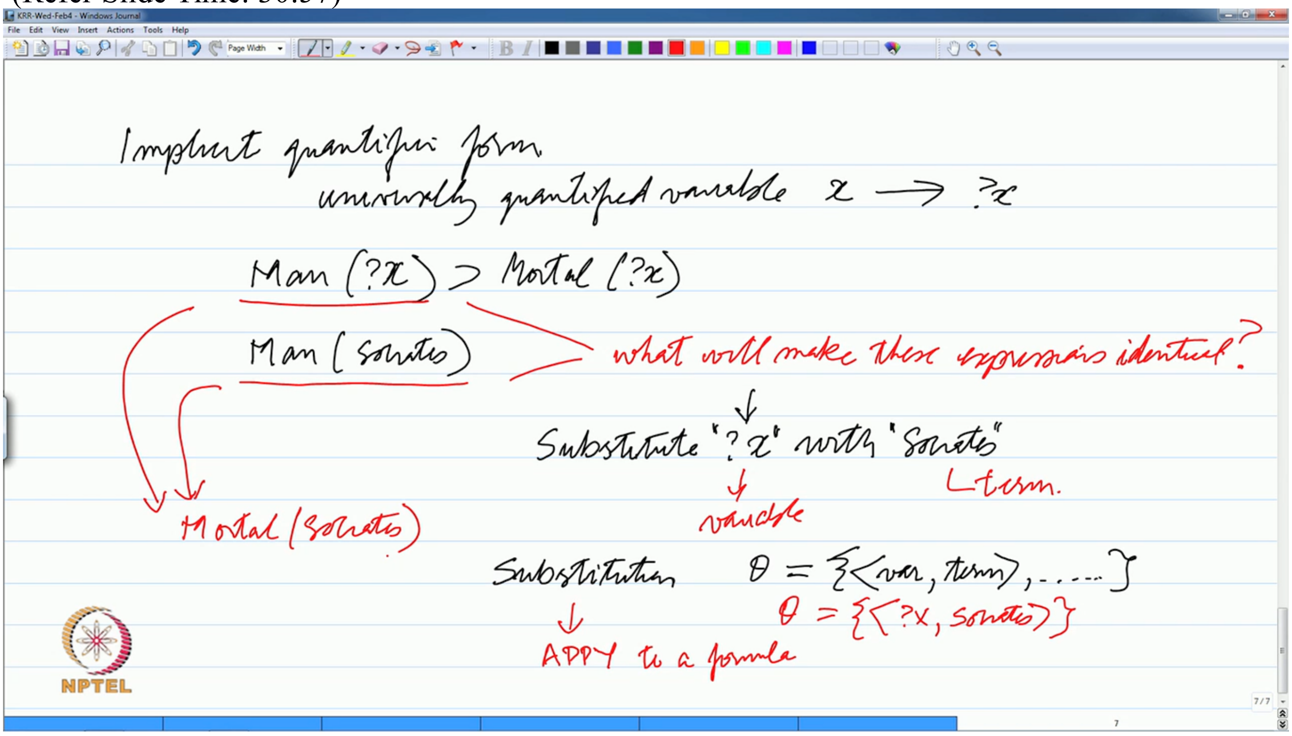
So we call this thing a substitution and a substitution often donated by greek letters for example theta says that I have a set of pairs which are variable and term. So substitution is a collection of variable term pairs or a variable value pairs as some books would say and in particular the substitution that we are interested in is in our example theta, say the variable we are talking about is x, maybe we will sometimes put the question mark here sometimes not because it is really clear from the context what we are talking about. We are substituting x with Socrates and this is what our theta is in this example. Now if we apply this substitution. So we have this notion of applying substitution to a formula

(Refer Slide Time: 30:17)



It means basically substitute whatever are the variables we have talked about in substitution with the values that are given in the substitution. Remember substitution is a list of variable value pairs and we apply this to get a new formula. So once we do that then we can directly infer Mortal Socrates.

(Refer Slide Time: 30:57)



So this rule which allows us to do this, takes these two formulae and infer this formula by applying a substitution so theta that we are applying is this one here is we will call this as modified modus ponens or MMP. So a modified modus ponens can be applied to a formula in an implicit quantifier form and what it needs is that you must be able to make the left hand sides of the implication and other statements somehow identical by applying a substitution. And then you can apply the same substitution to the right hand side of the implication to infer that essentially. So I will stop here. We will formalise this notion in the next class and we will also look at the algorithm which is needed for making these expressions identical and this algorithm is called the unification algorithm which we will study in the next class.