Artificial Intelligence: Valid Arguments and Proof Systems

Prof. Deepak Khemani

Department of Computer Science and Engineering

Indian Institute of Technology, Madras

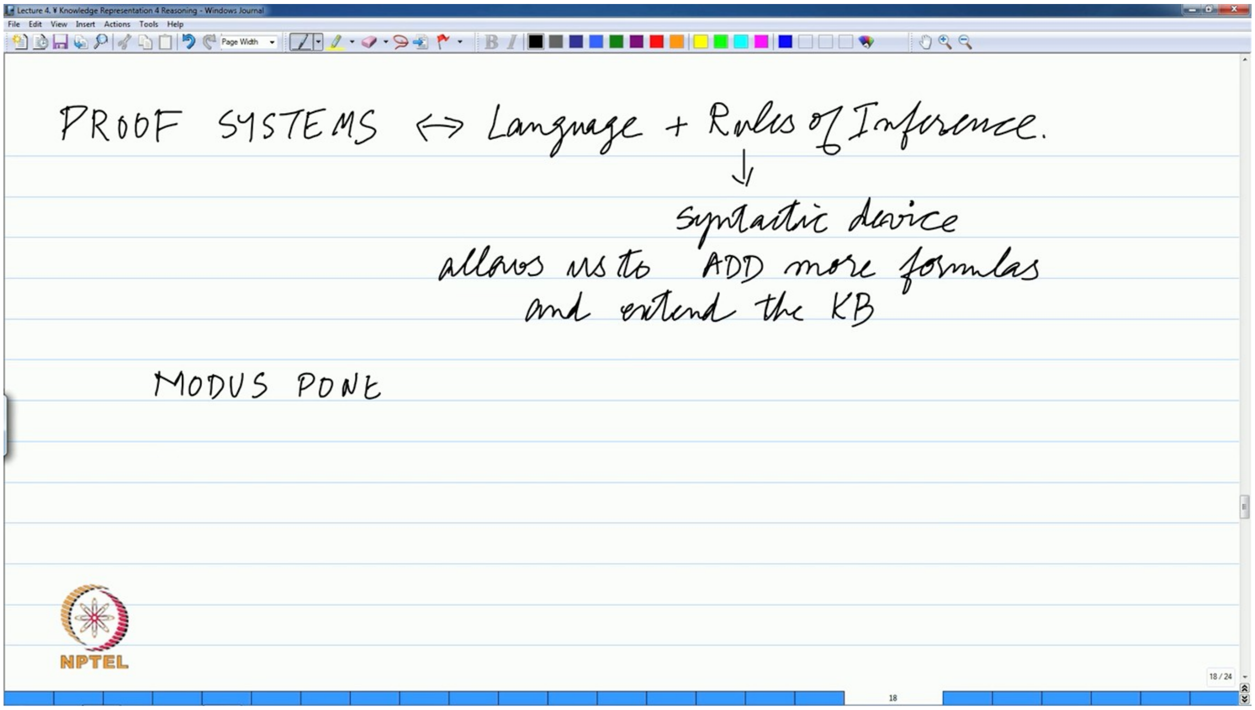
Module – 02

Lecture - 03

So in the last class we looked at the syntax and the semantics of propositional logic. So what semantics gives us is that it gives us a mechanism for determining which compound formulae are true and which compound formulae are false. We also saw that there were three kinds of sentences that we talked about set of tautologies, set of contingencies and set of contradiction. Now tautology is of particular interest to us and we are often interested in knowing whether a certain formula is always true or false. And the only way to construct to find out using the method that we have seen is by constructing a truth table for all. So what does a tautology mean that for every atomic sentence for every possible value there are two of them the final formula must be true. Now if we have two variables then we saw that we have four rows in the truth table. If we have three variables there would be eight rows. If there were four variables there would be sixteen rows. So the size of the truth table tends to increase exponentially with the number of propositional variables and therefore if you talking of a large problem it is not always viable idea to fill the truth table. Moreover, when we move to first order logic which is a more expressive language we will see that the idea of truth table just doesn't carry over at all.

So logicians have at all times been interested in some other ways of arriving at these formulae which are of interest to us. And the other way that we are talking about is a proof system. So proof systems are basically a language that we have already defined plus rules of inference. And when I put an algorithm on top of that then we have a proof procedure. We will come to that eventually so let's look at proofs system. So we started off with a language. The language itself is defined by an alphabet and a set of formulae that you can construct in the language. Then we define the semantics now we are talking about rules of inference. So what is a rule of inference. It's a syntactic device which allows us to add more formulas and extend the knowledge base. So proof systems are they don't look at meaning they don't look at truth value so they are not really concerned with truth value. They tend to treat knowledge base as a set a set of formula that is given to you and a proof system allows you to add more and more formulae to it.

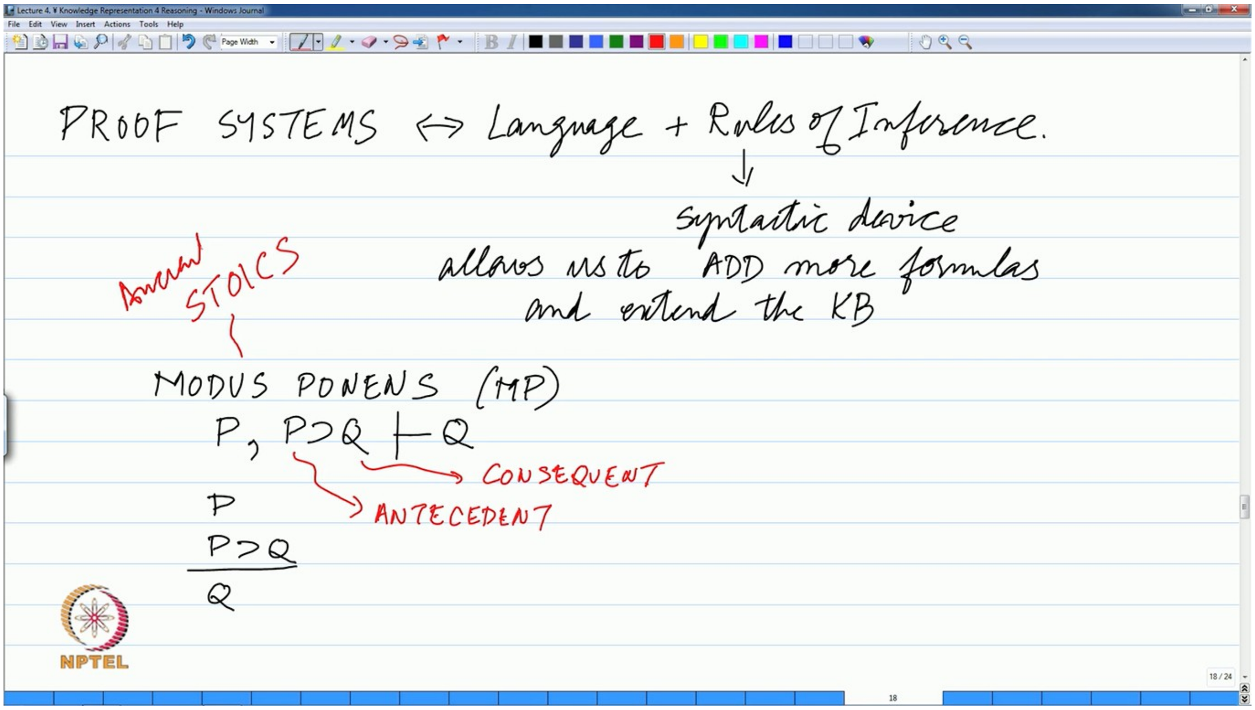
(Refer Slide Time: 4:39)



And how do we add new formulae we do that by applying some rules of inferences so let's start off with some example of rules of inference. The most common that no doubt you are familiar with is called modus ponens. We will use the term MP. Modus ponens says that if you have a propositional value P so remember it's a variable and you can plug in any formula instead of P. and if you have a formula which uses a propositional variable P and another propositional variable Q then you can add. It's a rule it's not really concerned with truth values. It simply says that if you can see if we can plug in something into P and the same thing into a formula P implies Q then you can infer or you can add Q. we have been used to write this also as P P implies Q hence Q. it's a same thing just different notation for writing. So basically what this says is that in your knowledge base if you have these two patterns one corresponds to some formula and the other corresponds to a larger formula where this is the antecedent of a implication.

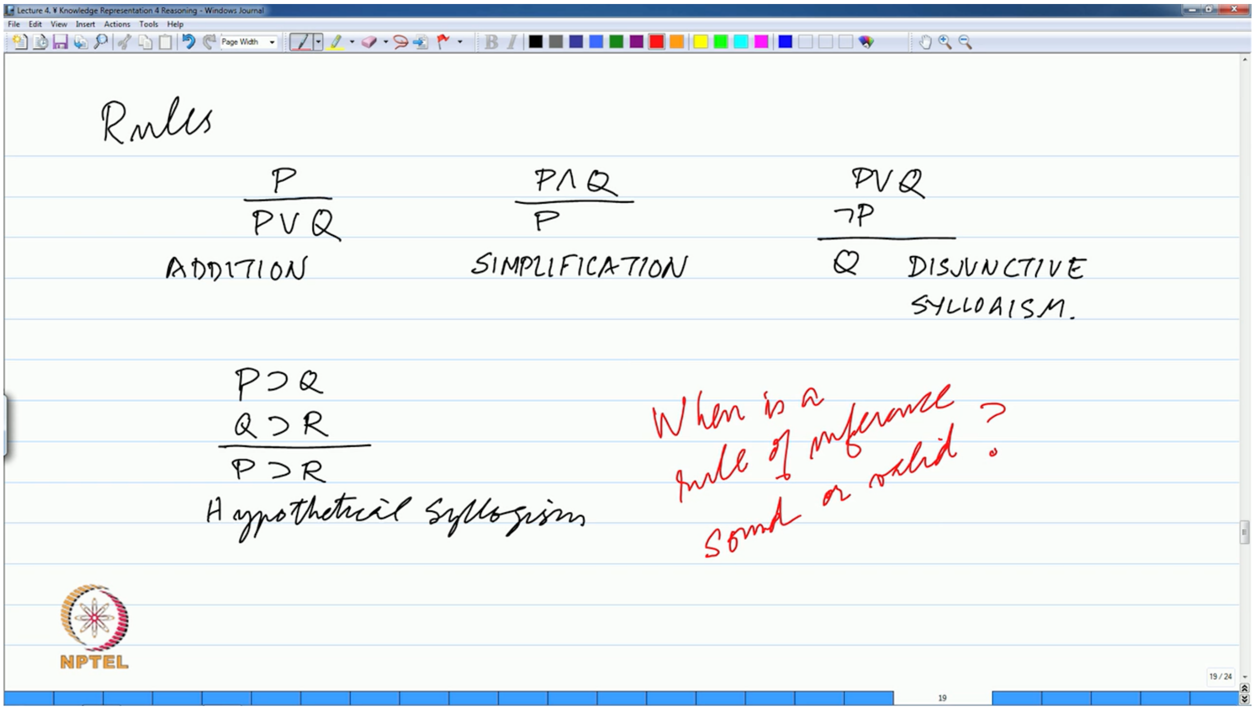
So whenever we talk of implication we call this antecedent and this consequent. And modus ponens is the most commonly used rule of inference. In fact if you look at any textbook on logic you will see that it was known to Stoics. People in the early greek times which was a school of philosophers and they had used modus ponens and the rules of inference quite may be two thousand years ago or whatever something like that.

(Refer Slide Time: 7:34)



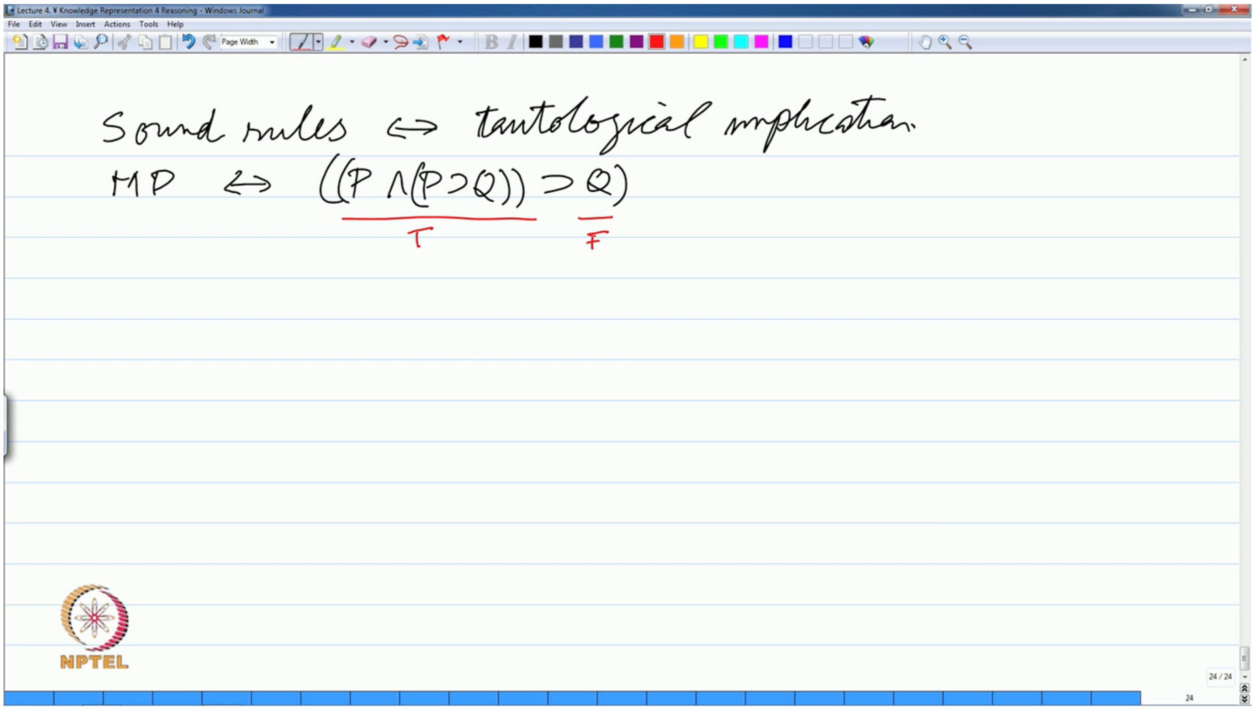
So let's look at a couple of more rules of inference. There is another rule called modus tollen which can be written as if we have not Q and P implies Q then not P. which we can also write as not Q P implies Q. so let's look at a few more rules. There is a rule that says that if you have P then you can add anything to it. this rule is called addition. There is another rule which says that if you have P and Q you can extract one part of it called Simplification. There is another rule which says if you have P or Q and you have negation of one of it then you can infer the other. It's called Disjunctive Syllogism. There are other rules for example P implies Q and Q implies R then P implies R. it's called Hypothetical Syllogism. So basically a rule of inference says that you can add new formulae using this pattern matching rules. The question that we want to ask is when is a rule sound or valid?

(Refer Slide Time: 11:10)



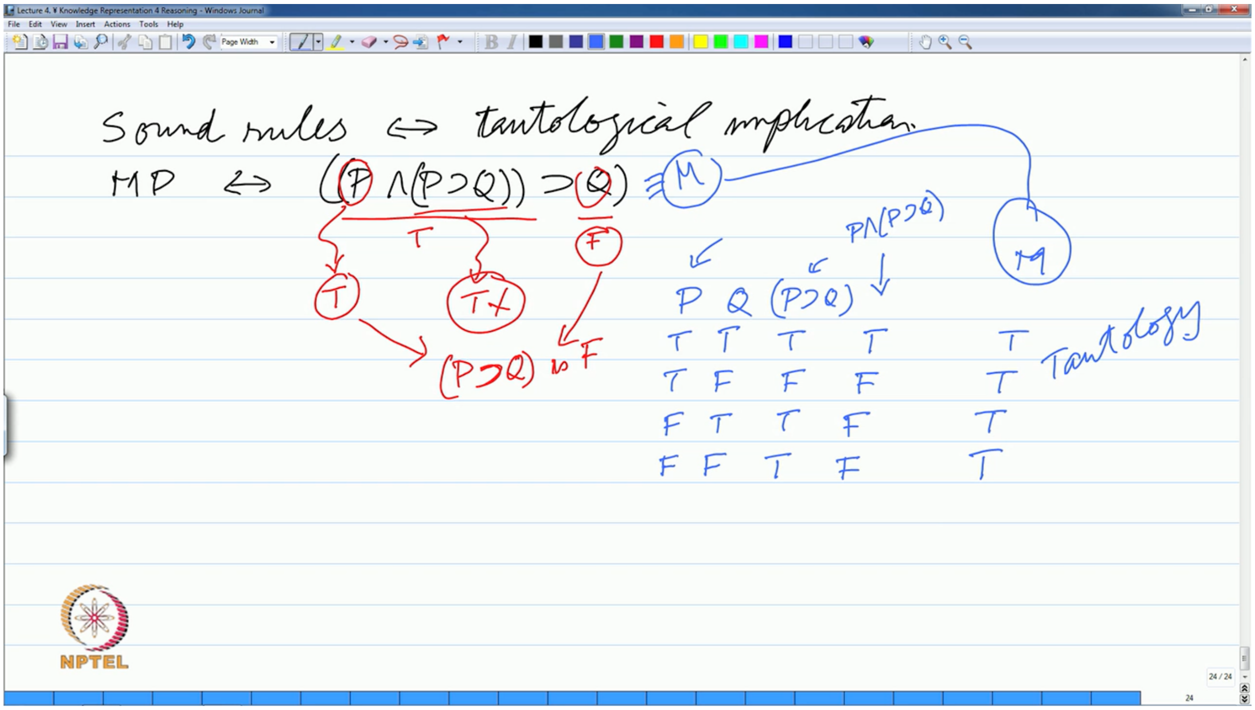
When is it okay to add this new formula. What do we mean by that is that if you are looking at a set of true formulae then a sound rule of inference will only allow you to add formulae which are necessarily true? So the answer to that is that it must be based on a tautological implication. So what do we mean by that. So basically what it means is that it must be based on a true statement and the true statement must be of a form where the main connective is implication. So for modus ponens the corresponding tautological implication is P and P implies Q the whole thing implies Q. so what we are saying now is that if this statement is a tautology then the modus ponens rule is a sound rule. So let's see whether it's a tautology or not. there are a couple of ways of trying to make it a to show that it's a tautology. One way is to try to show that it's not a tautology which means to try to make it false. Now if you want to make this formula false you know that if you look at the truth table for implication you will have to make this true and you will have to make this false.

(Refer Slide Time: 13:18)



So if you want to make this true then you will have to make this true and this second part also true. So if we are saying that P is true and Q is false then P implies Q is false by the truth table because true implies false is false that's the only case where an implication sign is false. And therefore this is false this cannot be true and therefore we cannot make the whole formula false. We could have also constructed the truth table for this so let's do that. So we start with P and Q so they can be true or false these are the four combinations we always keep with two variables. Then we have P implies Q we know that in this case it is true in this case it is false in this case it is true in this case it is true. Then we have this other formula which is P and P implies Q which is basically and of this one and this one. T and T will give you T T and false will give false false and true will give false false and true will give false. Now I am saying that this larger formula which is let's call it M. let's write the truth value of M so I am saying that this implies this that's the larger formula right. P and P implies Q implies Q so when this is true and this is true this is true. And this is true and this is false this is true and this is false and this is true this is true and this is false and this is true this is true. So this is M and we have shown that this is tautology. That this formula will always be true. The largest column is always true so that's one way we said we can show that something is a tautology.

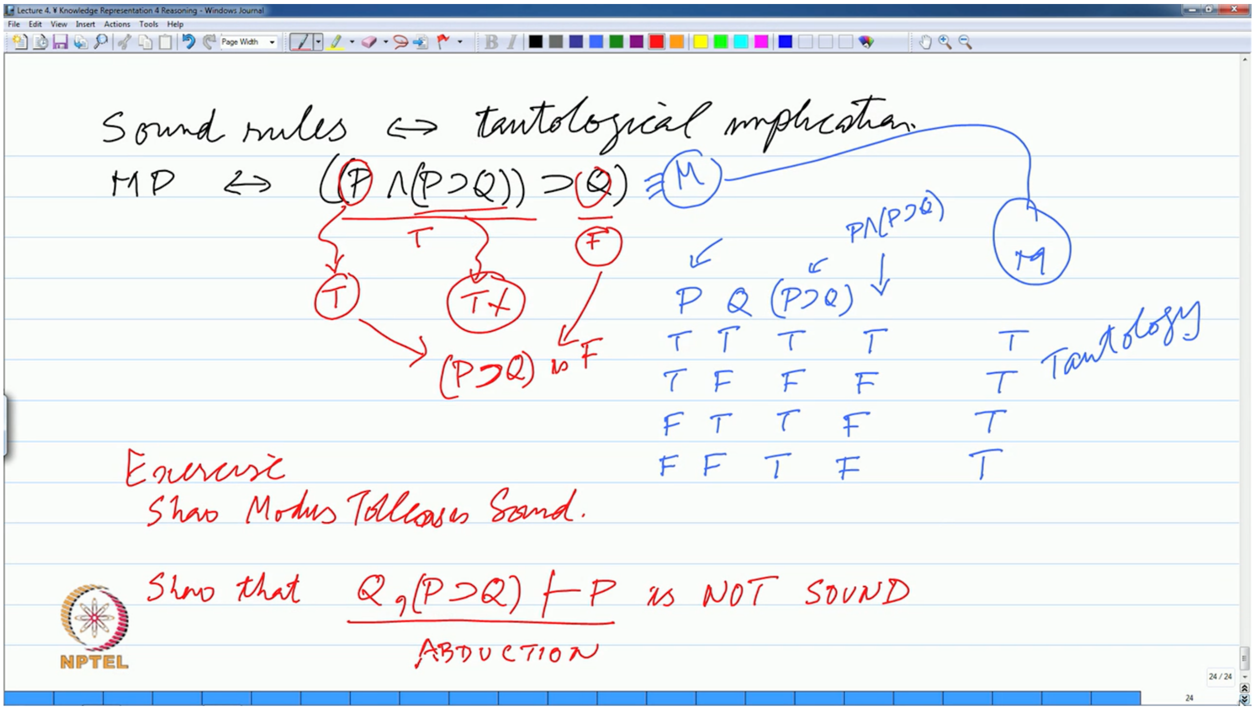
(Refer Slide Time: 16:36)



So we can show that the underlying implication for modus ponens because what is modus ponens saying it is saying that if you have P and if you have P implies Q then you can add Q to your knowledge base. And this is the underlying implication. And once that's a tautology we know that it's a sound rule of inference. So for every such implication statement you can create a new rule. So for example if we had an exercise show that modus tollens is sound. Or any of the rules we have spoken about. Also as an exercise show that this rule Q and P implies Q implies P is not sound. So you can separate the sound rules from the unsound rules by simply looking at the underlying implication and show that if it happens to be a tautology then it is sound otherwise it is not sound. Incidentally this rule is called abduction so even though it's not a sound rule of inference what do you mean by its not a sound rule of inference. Say that it's possible that the inputs the antecedents which is in this case Q and P implies it is possible that P is false.

You can see that whenever supposing Q is true whatever the formula Q is and if P is false then you can see that the left hand side or the antecedent's Q is true because we said its true and because P is false P implies Q must be true so that is true but the P consequent is false so this is not a valid rule of inference which means we cannot use it in our proof methods.

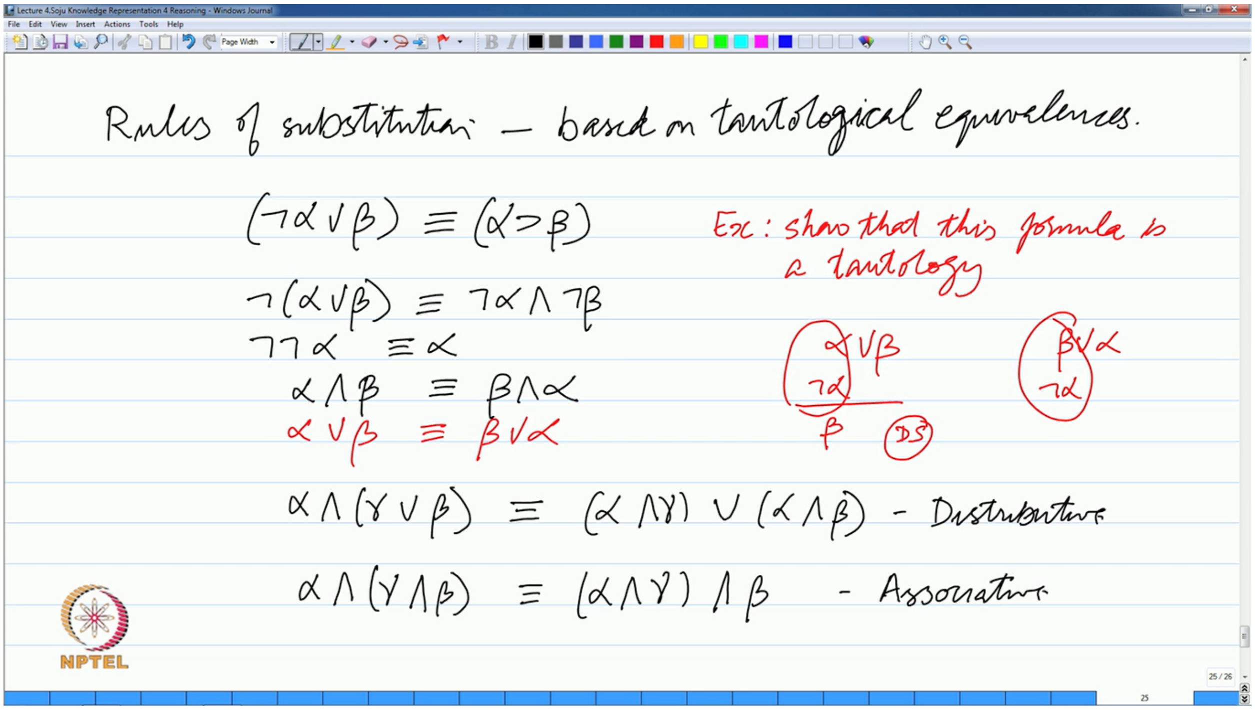
(Refer Slide Time: 19:11)



So before we move on we should also look at rules of substitution. They are often very handy devices. These are based on tautological equivalences. So rules of inference say if the left hand side is given to you then you can add the right hand side or if the formulae above the line are given to you then you can add the formulae below the line. Rules of substitution allow a two way exchange you can substitute either formula for any formula. So we have seen examples of this. For example, we have seen that not alpha or beta is equivalent to alpha implies beta. So based on this we can have a rule of substitution which says that if you see the left hand side somewhere you are free to substitute the right hand side. Likewise, if you see the right hand side somewhere you are free to substitute the left hand side. So these are called rules of substitution. And they are based on the fact that this underlying formula is a tautology.

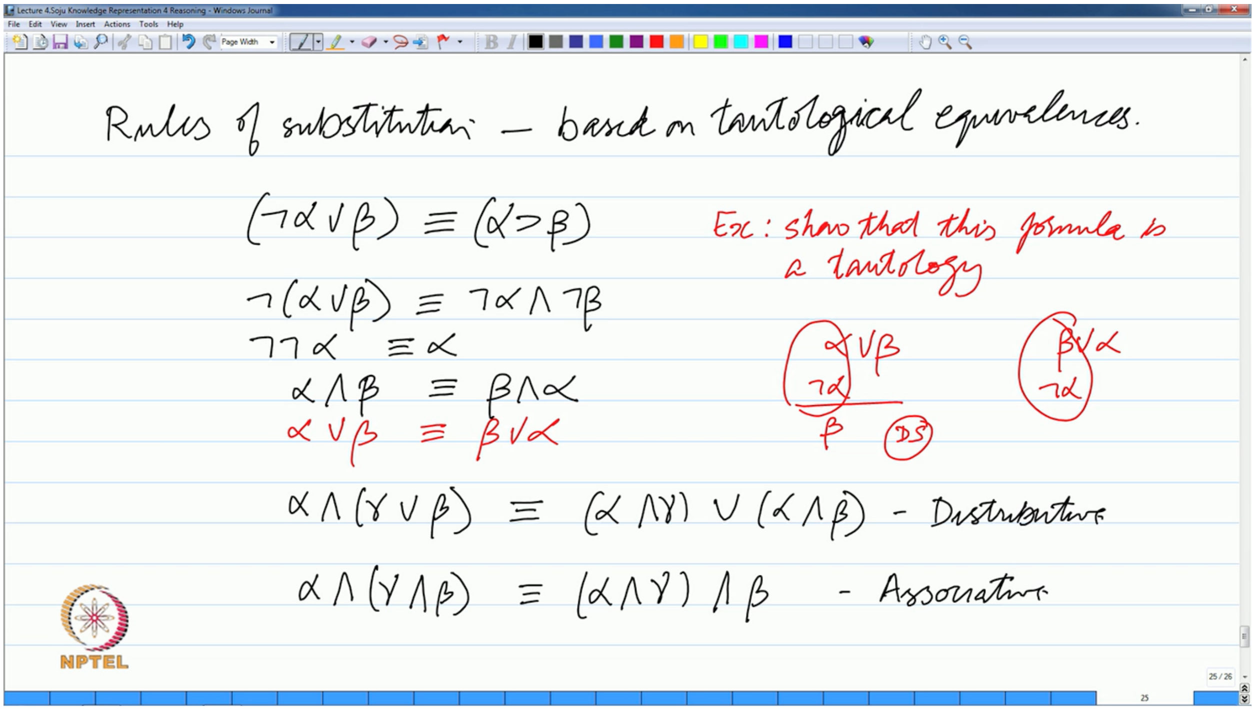
So as an exercise show that this is a tautology. All rules of substitution are based on tautological equivalences. What are the other formulae you must be familiar with many of them? These are known as De Morgan laws. And even simpler formulae like alpha and beta is equivalent to beta and alpha. Remember that the proof procedure is a syntactic procedure. It just looks at pattern matching. So if you see alpha and beta somewhere so for example we saw a rule earlier. So for example we see this rule which says that the disjunctive syllogism it says that if alpha or beta not alpha implies you can infer beta.

(Refer Slide Time: 24:15)



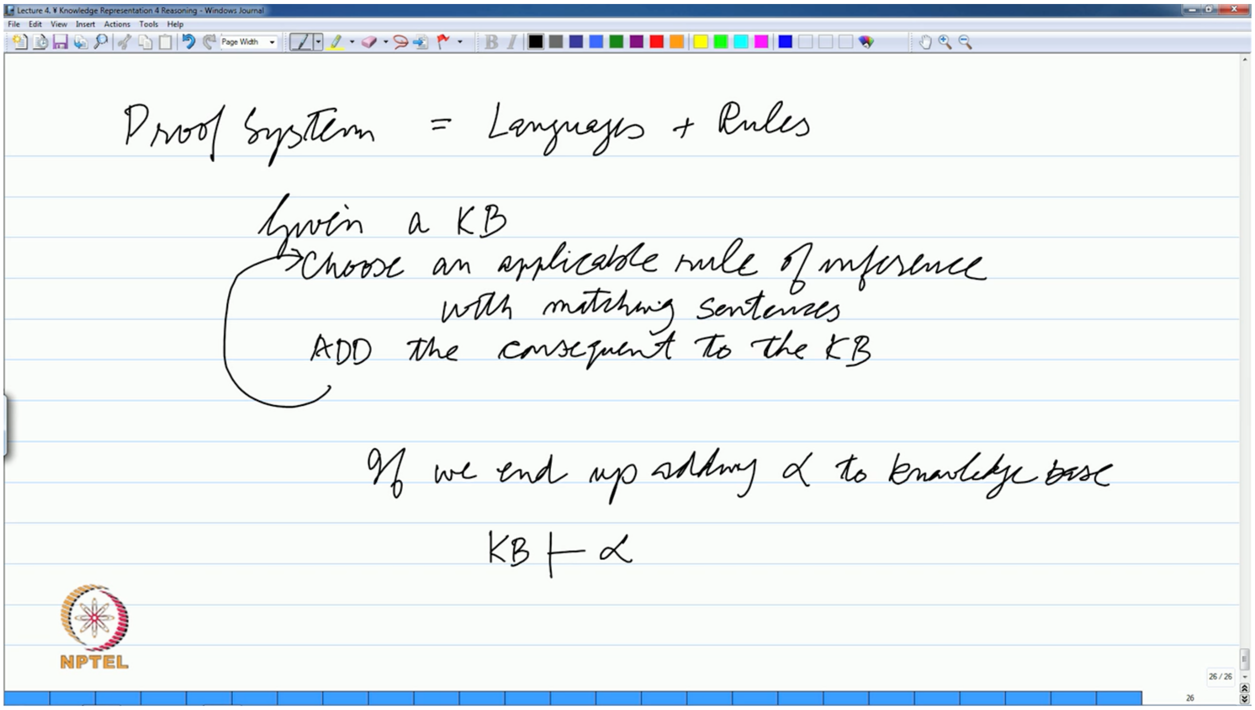
Remember this was disjunctive syllogism but what if you are given beta or alpha and not alpha obviously the pattern matching will not help here because in some sense this is saying that these two formula should be matched whereas these two formulae don't match. Whereas this kind of rule which is a as you know commutativity which I have written here alpha or beta is equivalent to beta or alpha. So I could have substituted alpha or beta instead of beta or alpha and then I could have applied the disjunctive syllogism. It allows us to do all these kinds of things. We also have rules like alpha and gamma or beta is equivalent to alpha and gamma or alpha and beta which is a well-known distributive property. Since we are talking about binary connectives we may need to use rules like alpha and gamma and beta is equivalent to alpha and gamma and beta which is a well know associative law. So with these kinds of laws we can substitute one formula with the equivalent formula which can be used in other places.

(Refer Slide Time: 24:14)



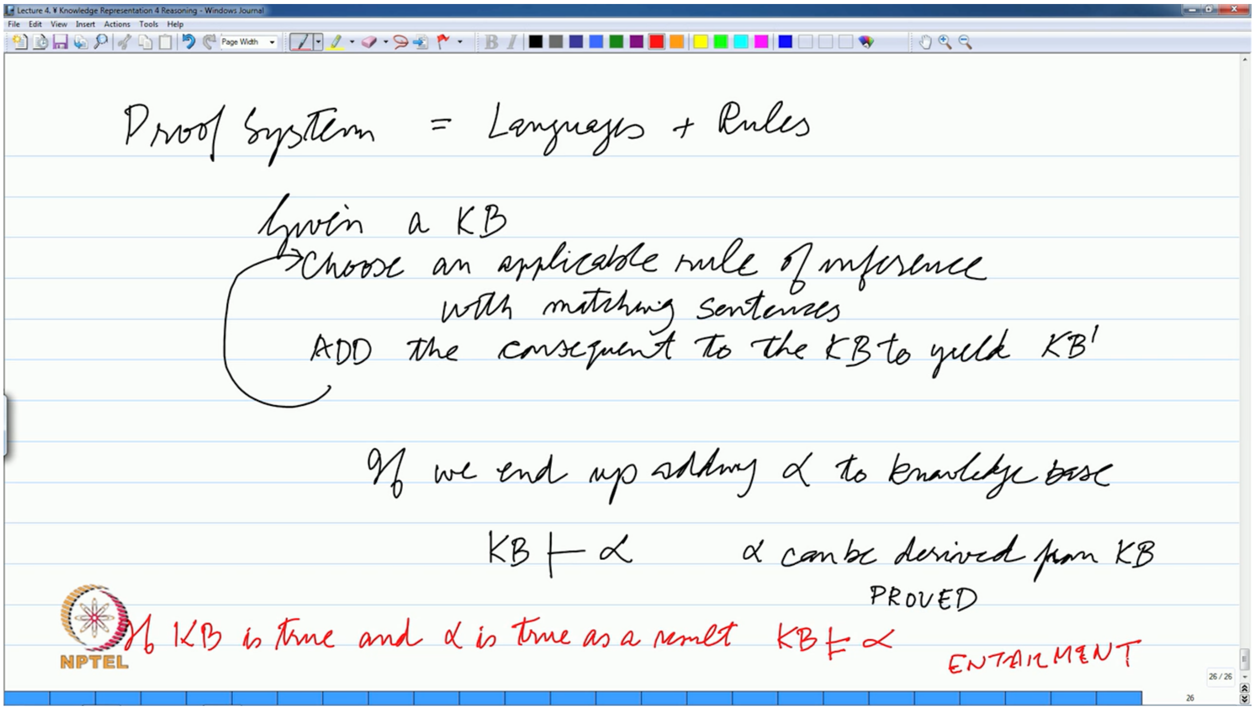
So what is a proof procedure a proof system. So remember we had said that its language plus rules. So I will include both rules of inference and rules of substitution. And you can see that the rules of substitution is equal to two rules of inference. It can be broken down into two rules of inferences where one is given and other is the thing that can be added it can be done in both ways. So given a KB choose an applicable rule. So let me just write a rule of inference because I just said that a rule of substitution can be broken down into two rules of inference anyway with matching data. I am saying data but its matching sentences. Add the consequent to the KB. So this thing if I put it into a loop. Take a rule of inference apply it add something to the KB, take another rule of inference apply it add something to the KB. So I have a means of growing the knowledge base. So at the end of it if I end up adding some formula alpha to the knowledge base then we write this notation that from the knowledge base we can derive alpha.

(Refer Slide Time: 27:04)



So let me just add here to yield KB prime. Just to separate the fact that KB is the original knowledge base that we started with and then if you can apply a succession of rules of inference and produce this formula alpha or add it to the knowledge base then we say that alpha can be derived from KB or alpha can be proved. So given the KB we can produce alpha and add it to the knowledge base. We have already seen a notion earlier when we said that given a knowledge base what else is true we can determine using the truth table. If so let me use a different here. So if KB is true and by this we mean that every sentence in the knowledge base is true and alpha is true as a result we have already said that alpha is entailed by the KB. We write this a KB entailed alpha. A similar symbol but with two lines. So this is entailment.

(Refer Slide Time: 29:17)



Do you think we have two ways of looking at alpha one is through a process of semantics by looking at truth table and finding out that if my every sentence in my knowledge base is true is alpha true. Then we say that the alpha is entailed by knowledge base. The other procedure is the procedure that we have just defined which is a proof procedure which says that I can keep applying a rule of inference now remember rules of inference are not part of the language they are outside language. That's why we say that a proof system is language plus rules of inference. But if I choose a language and if I choose a set of rules of inference and if I repeated application of rules of inference to deduce a formula alpha then we say that alpha can be proved in this proof system or alpha is derivable or alpha can be derived. The question we want to ask is are these two approaches arriving at the same answer or not because our interest is in truth values. Is alpha true or not but we have started by constructing a machine or a proof system which we can a machine which can produce alpha may be. So the question that we had mentioned towards the end of the introductory module was this notion of soundness and completeness. Just to repeat it quickly a logic system or a proof system is sound if it only produces true formulae. This procedure will only add an alpha if alpha happens to be true. A logic machine or a proof system is said to be complete if it will add all true formulae whenever alpha happens to be true I can be rest assured that my logic machine will produce it at some time. So if we had such a machine then obviously we won't have to consider about truth table and semantics and all these kinds of stuff. We could just entrust the task of finding whether a formula is true or not to the logic machine. if it produces it we will say it is true if it doesn't we will say it is false. So will look at this notion of soundness and completeness again little bit in the next class and we will look at different kinds of algorithms which can be used to apply the rules of inference and as I said there are certain algorithms which are easy to program because there is no guess work whereas there are other algorithms where guess work is involved but you will not be able to write a program to do a kind of guess work. So we will look at different options and then we will settle down with one or two algorithms of proof systems this exercise is also called theorem proving. So our task is to design a theorem prover by theorem we mean something which is true. So we will do that in the next class.