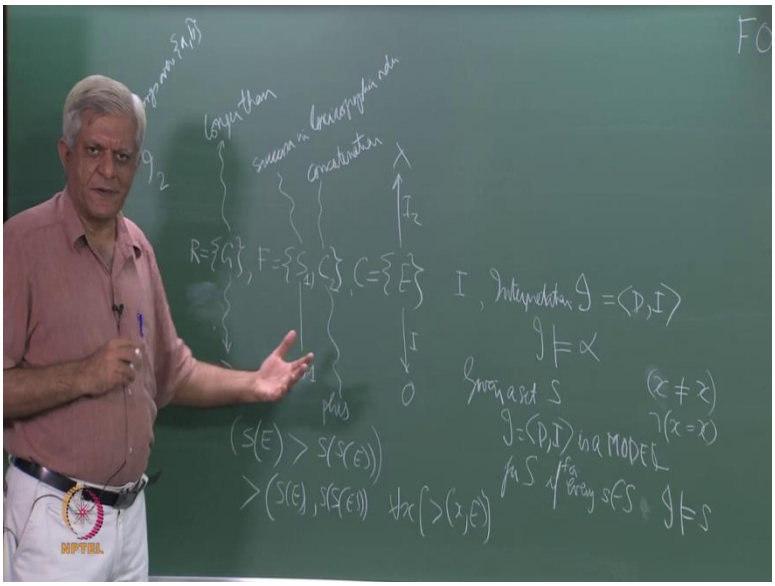
Artificial Intelligence
Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 46
Reasoning in FOL

So, let us continue with the study of FOL. In the class, we defined the language, essentially the syntax and symmetries. Now, let us look at how you can do reasoning or inferences with possible logic, but before we do that let us just recap that, a language is defined by the sets of RFC. So, let us say that r is a symbol called g and very small language, F is a symbol a 2 symbols, let us say one is called s and one is called c and c contains of one symbol, let us s say is called e. So, we said that we defined a mapping I to define the meaning of these 3 sets these things essentially.

(Refer Slide Time: 1:43)



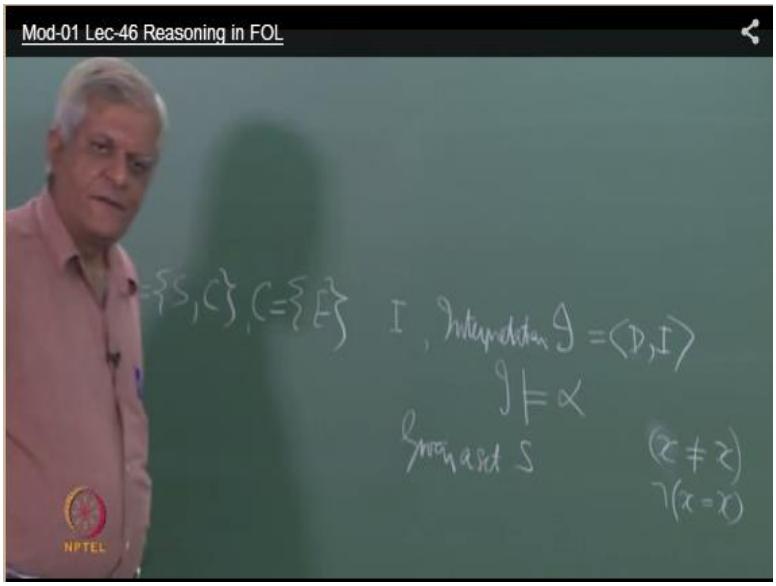
So, this mapping is called interpretation mapping, but we also call an interpretation that is use the symbol I is defined as a domain and an interpretation mapping. So, an interpretation of a set of sentences or a language is in terms of the domain about which, the language is making statements and a mapping which tells you what does is predicate symbol mean, what does each function symbol mean and what does each constant symbol mean essentially.

So, whenever a sentences alpha is true under an interpretation and we defined the symmetric in the last class. We say that is interpretation entails alpha. So, it is a notation that we use and

we talked about the motion of valid sentences. So, a valid sentence is something which is true in all interpretations, which means you can choose any domain and any mapping function and the sentence will be true. So, basically the set of tautologies that we can talk about. So, for example, the system that we discussed earlier.

Satisfiable sentences are those which are true under some interpretations and unsatisfiable sentences are those which are never true. So, an example of unsatisfiable sentences would be for example, x not equal to x. So, such a atomic formula can never be true essentially. I think which we can also read as write as x which is the same thing.

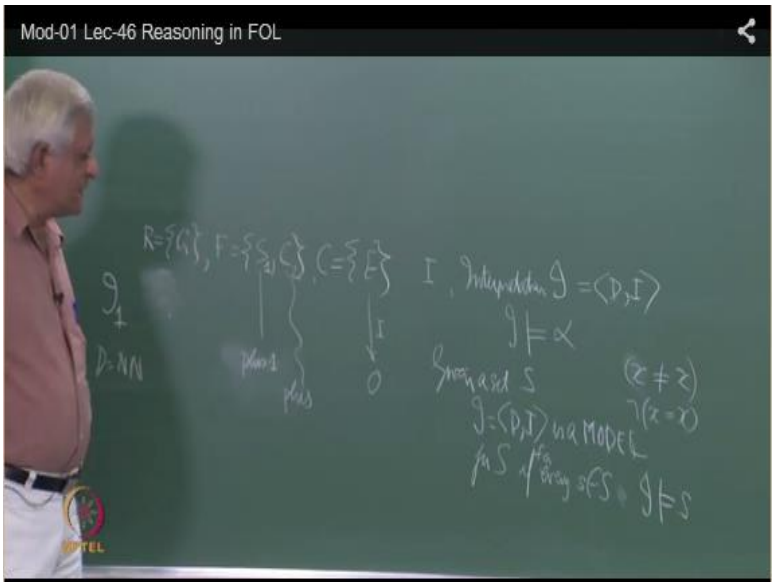
(Refer Slide Time: 03:37)



It is just a short form that we have used to write in. So, given a set of sentences s we say that a interpretation which means a domain an interpretation is a model for s. If every s belongs to s interpretation imply this sentence s. So, if you write a set of formulas or sentences in a given language and then you get find the domain and an interpretation which makes all those sentences true. Then we say that we have found a model for those set of sentences. Logics are always tried to say this is the logic system and does it have a model essentially.

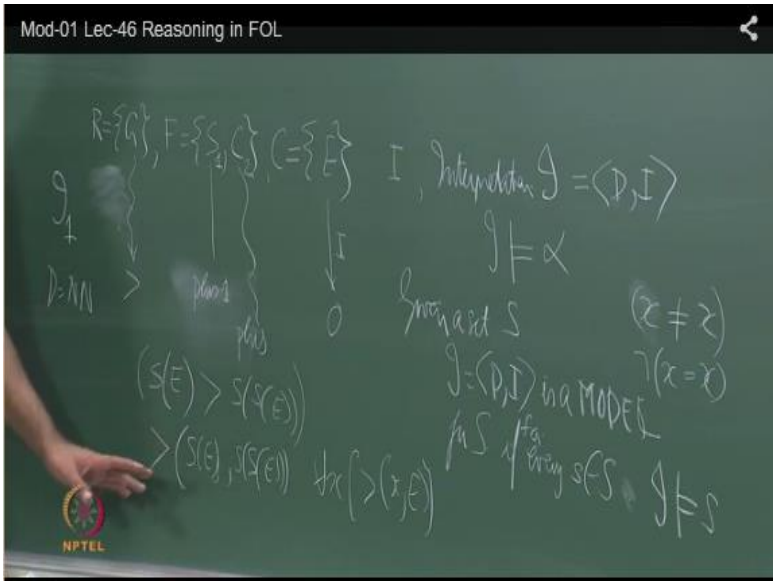
You means can you is it true in some domain and some interpretation it is just to for the sake of complete lesson. We not really go into too much detail there. So, let us look at this small example of a small language which is got one relation symbol which, we have called g and two function symbols we have called s and c and one constant symbol, which we have called e essentially. Now, we can have one interpretation I 1, which does the following.

(Refer Slide Time: 05:24)



Let us say domain is equal to national numbers and I can say my constant. So, this is the mapping I maps to 0, the number 0. This maps to successor plus 1, this let us say this is of I i t 2 and this is of ii t 1, this maps to plus or multiplication it does not matter some binary operation on this. This map to greater than let us say. Then I can write statements for example, I can say successor of e greater than successor of successor of e. I would use this as the prefix and then said the arguments are, but I used so this variations.

(Refer Slide Time: 07:13)

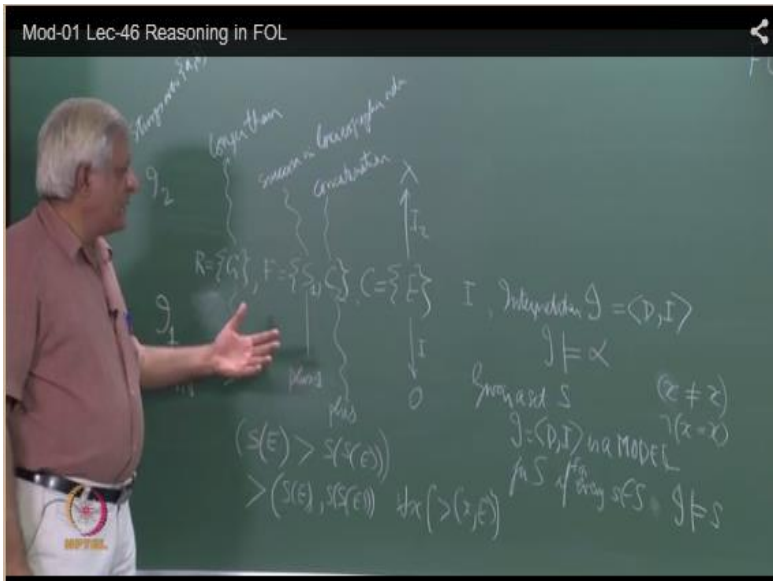


So, this is the predicate symbol and these are 2 arguments, which are terms and the terms are

used constructed using the function symbols and the constant symbols here. They are no variables in my sentence, but I can add variables 1 of them could be. So, I could write for all x greater than x e. I am not talking about the truth value of these sentences. I am not claiming that this sentences is true or this sentence is true. Always saying is that we can interpret what does the sentence mean and the meaning is given by a choosing a domain and choosing an interpretation mapping.

Then any sentence in my language here which I have define the syntax, the sentence like this is essentially saying as we all know that the successor of 0 is greater than the successor of successor of 0. Of course, it is not a clue set will, but that is what it saying. So, we can understand the meaning of the sentence and this is saying that every x is greater than 0, which may or may not be true or which is not true, but that that is not the point. The point is not we are not talking about the truth value here, we are talking about the meaning of the semantics of the syntax.

(Refer Slide Time: 08:27)

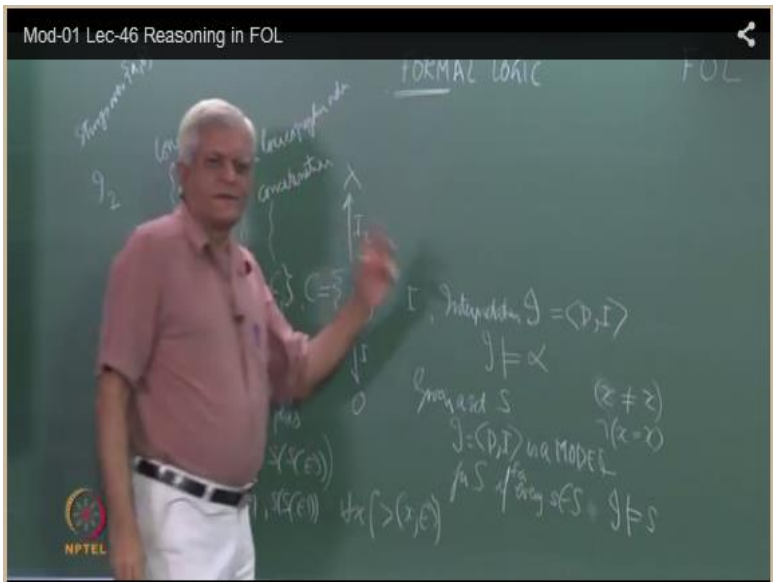


As the same time I can choose a different interpretation which is let us say streams over alphabet a and b and then I can say that this now this is i 2 maps to empty string, let us say lambda. This maps to concatenation, this maps to successor in lexicographic order and this let us say maps to longer than.

So, the point I want to really emphasis here is that a language is one thing and what it means or what is the semantics of those sentences in that language is totally another thing. It is

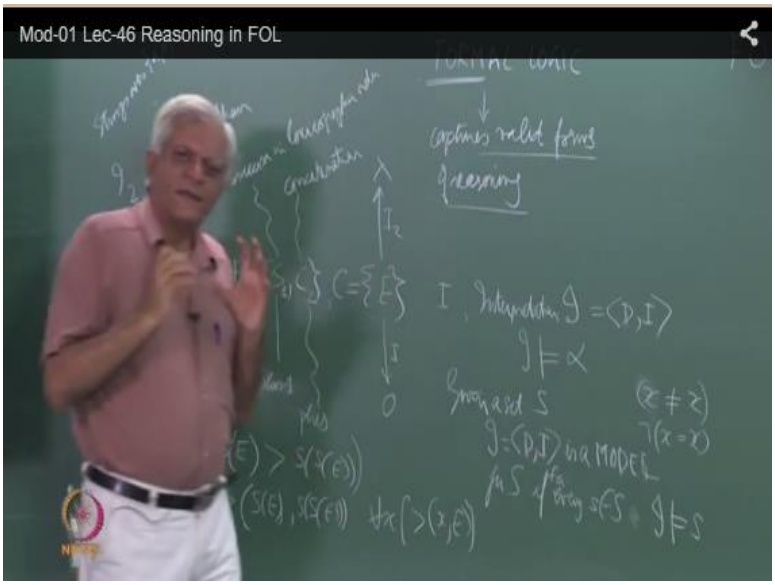
depended upon the interpretation. So, a same expression like this which is defined in my language would mean one thing if I am talking about numbers. It would mean a different thing if I am talking about strings, it could mean a totally different thing if I am talking about people. For example, I could map these things to some relation between people. So, that is one point which, I really want to emphasis that and this is something which is very central to in fact writing programs as well as the study of logic.

(Refer Slide Time: 10:03)



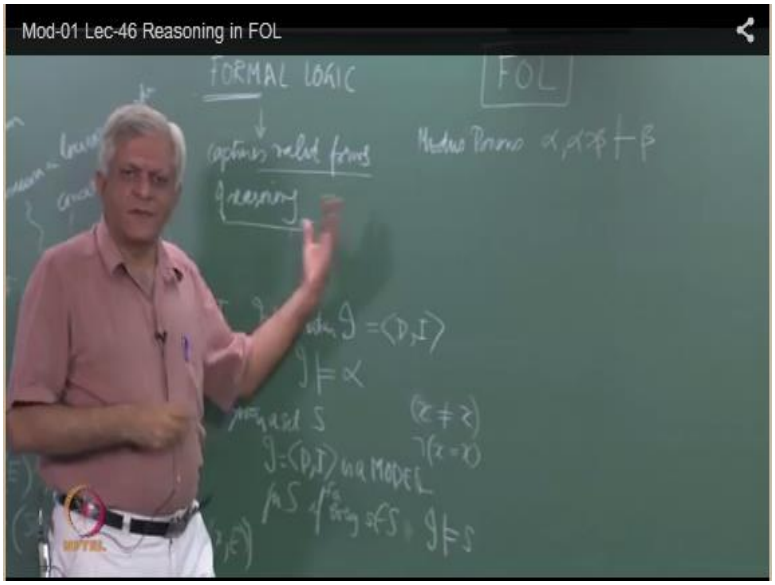
So, we cannot stop emphasizing the fact that everything that we do in logical reasoning is based on forms essentially. It is not ma based on meaning essentially any proof procedure that I have, if you remember it says I have a rule set of rules of inference, I gave a set of premise sand I keep adding new formulas. It is purely based on form it does not matter what we are talking about numbers or they are talking about strings or they are talking about something else.

(Refer Slide Time: 10:47)



So, the logical system is formal and the logic is basically captures valid forms of reasoning. So, this we have talked about earlier as well, so it is it says that when is an inference or sound inference, which we call as deduction essentially. How do we define deduction? Logic is basically concerned with making sound inferences and we have seen as to when can a rule of inference we sound. For example, it is based on tautological implications or tautological equivalences and so on. If you build logic around sound rules you will get a sound logic it, but everything is formal essentially.

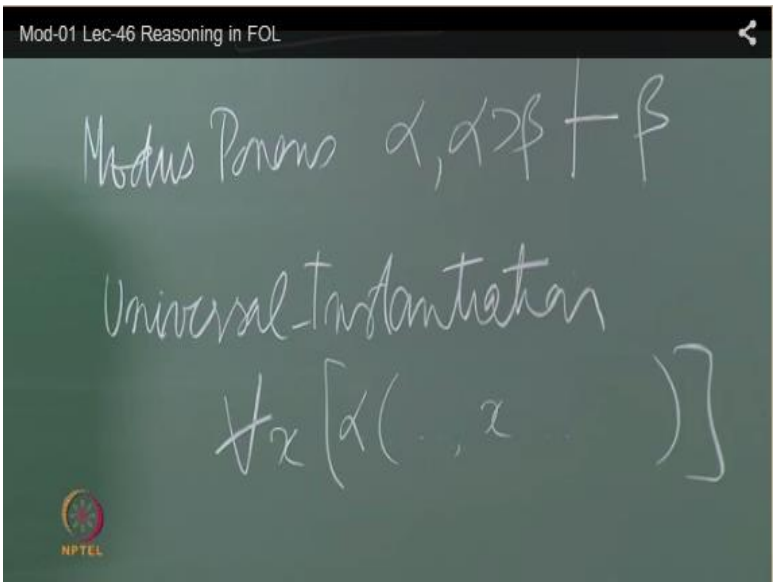
(Refer Slide Time: 11:38)



So, what is the reasoning mechanism that we can use in first set of logic. We already have a set of rules that we have spoken about. So, for example, still applies it still says the same thing that if we have alpha, if we have alpha implies beta, then you can derive beta. Thus, same rule we can carry forward to first set of logic because it is only talking about, how to interpret this implication sign essentially even that we have essentially. It not really worried about things like variables and so on.

So, it does not matter what alpha beta is. As long as they are sentences this rule applies essentially, but we also need some new rules of inference to take care of the fact that we have talking about quantifiers. The most important rule is call universal instantiation and what it says is that if you have. So, I will write it as top and bottom if you have a formula of the kind for all x and anything else inside this, but which contains an x essentially.

(Refer Slide Time: 13:11)

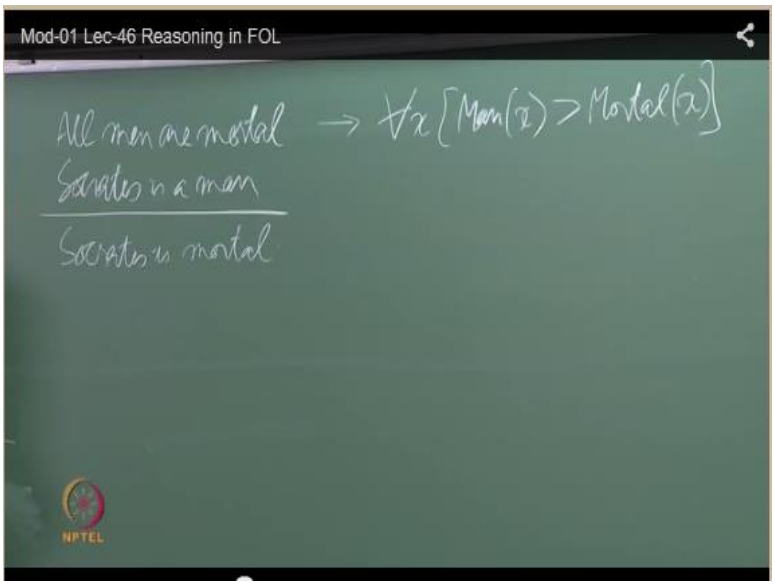


So, let us say an alpha with some x somewhere essentially, it could have other terms also. If you have a formula of this kind, you can have a formula of this kind alpha which the term replaced by a where a belongs to a set of constants. So, what we are saying is that you can always instantiate a universally quantified sentence to a sentence, which uses the constant essentially. So, let us see an example of this. So, if you go back to the argument that we were talking about.

The argument was all men are mortal, Socratic is the man and you want to show that Socratic is mortal. We had made an observation that we cannot do this in propagation logic and we

need something more expressive almost stronger to talk about this essentially. Now, we have the machinery for doing that. So, let us first translate these sentences into first logic and this is an exercise which you must have done i am sure at some point of time, but if you not done you must practice a little bit, but you can translate this into a language as follows.

(Refer Slide Time: 15:15)



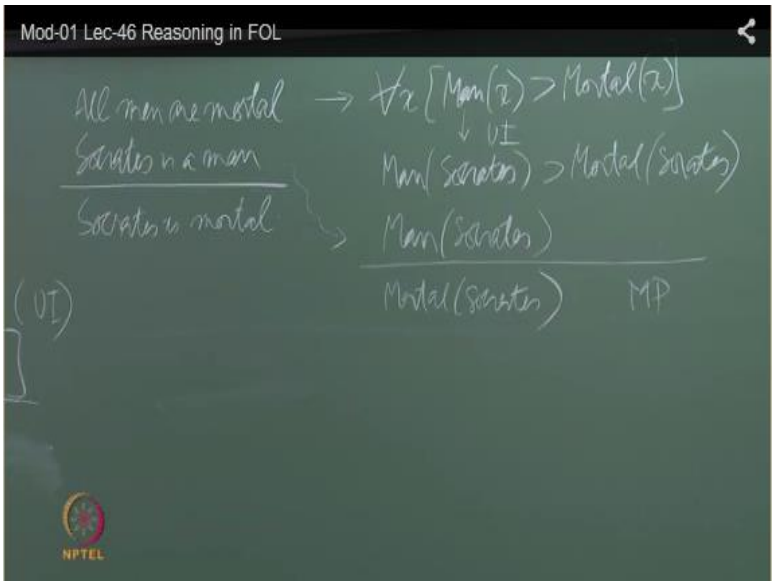
For all x, man x implies mortal x where, man and mortal predicates. Since unary predicates they basically define a subset of the universal discourse of the domain. What this statement is saying that, if any x belongs to this or any x satisfies the predicate man, it must satisfy the predicate mortal x essentially. In other words if x belongs to man interpretation, then x belongs to predicate man. So, what is this man interpretation?

Basically, it is a set in my domain subset in my domain which defines unary relation of man essentially. So, when you say man x we say that x belongs to the set of thing, which is call men. So, recall that unary relation is basically a subset relation essentially of the of the u universe of discourse essentially. You can see the same thing as saying that man is the subset of mortal i. So, this statement is essentially saying that the set of men and here off course we mean set of human beings is the subset of the set of things which have mortal essentially.

That is just set in English language we say that all men are mortal in FOL we will write it like this, but the semantics of that is that the set which corresponds to the set man. The mapping of this is a unary relation this is a subset you see call man I and essentially we are saying this. This sentence we can map to man Socratic and we want to show that mortal Socratic. Now,

with this rule of universal in sensation which we often abbreviate to UI. We can produce the formula which is very useful for us which is that I can instantiate this formula by substituting x equal to this. So, off course I should have clarify here the belongs to set constants. So, that is a inference step of UI and then off course this is the inference step of mortals followings.

(Refer Slide Time: 18:37)



Now, having used universal instantiation I have a formula which is very nice because it is saying alpha implies beta and then I have alpha and I can use which is this essentially. Again to emphasis the fact that it has nothing do with the factor it is man or it is mortal, it could be all students have bright or all leafs are green or all birds are small anything.

Any such statement if you have, if you accept the first statement, if you accept the second statement, we must accept the third statement that is all logic is telling us. It is form of reasoning this form of reasoning is valid essentially. If this is true and this is true this must necessarily will be true that is a lotion of ((Refer Time: 19:19)). What motion of proof is? This gives us a way of producing the sentence syntactically without looking at the meanings of things essentially.

As you can see if you want to talk about mechanizing this process, then there is a little bit of guess work involved there. What is this value of that? I should substitute essentially. Off course we look at this whole picture and we can see that a must be Socratic in that case of that x must be Socratic. Off course I will have this useful thing, but never the less if you are doing the forward search can of a process and you are saying I have all these universally

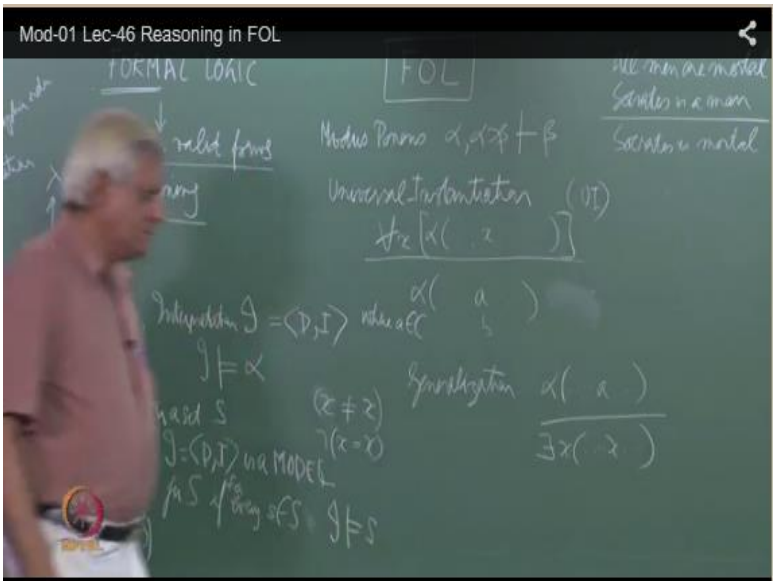
formulas, how do I proof something which means you have to keep.

So, remember that the basic mechanism is pick a rule pick a piece of data for which the rule is applicable and apply the rule essentially. So, this universally statement could be applied to 100s of people. Every element in my database I could say Socratic is mortal, Ram is mortal whoever essentially, anybody you give me a name and I will say he is mortal that is why I could prove that thing. So, this off course should trigger this thoughts about forward reasoning versus backward reasoning for you and we will look at that little bit, if not today then we will tomorrow.

The same kind of strategy is come into play. When you do forward reasoning you have a choice of applying so many rules of inference so much data that we have whereas, if you are doing backward reasoning, then you know what you want to show true, but the thing is that we need a role of inference which allows us to. We need to be able to use backward reasoning here.

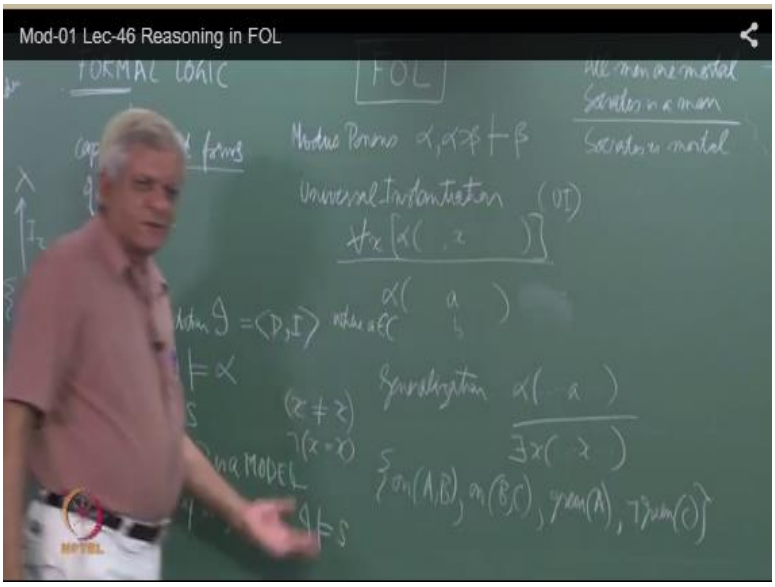
So, if you remember as you saw during planning, backward reasoning, backward state space search for planning and into some kind of problems essentially. So, are they similar problems here, we will have to see. Now, for a sake of completeness there another rule which is called generalization. What this rule says that, if you have a formula of this fine alpha a, you can replace it by a formula of the kind there exist x, x.

(Refer Slide Time: 22:03)



So, if I have the statement in my database, which says is Ramesh is bright, then I can generalized to say that their exist someone who is bright essentially. Why do we need this because sometimes the goal may be to show that there exist something, something, something essentially. So, an example which is given in one of the text book is as follows. It says that you are given the following facts on a b on b c green a, not green c. So, this is what is given to you the set of premises, the set s of sentences that a is on b, b is on c, a is green and c is not green.

(Refer Slide Time: 23:22)



Your goal is to show in exist on x then exist on y that on x y. So, the goal is we asking whether this formula is true. So, this is given to us, a is on b, b is on c and you can off course do any interpretation of the set of formulas. The most natural interpretation which comes to main is the blocks follow interpretation and where green stands for a color. So, let us accept that, but the thing is to show that this formula is true, we are not going to rely on the meaning of those sentences essentially.

You want to show that this formula which is there exist x exist y such that on x y and green x and not green y these true. So, what does this formula true or not true? First of all what is the intuition whether it is true or not true and secondly, off course the question that we will ask is can we prove it. Remember the motions of soundness and completeness that does a logical system prove every true formula essentially.

So, first let us let me ask is this formula true or not true. You people have done the career

course should not answer what is the intuition says? So, let us look at the meaning it saying that there is a block on another block and the block above is green and the block below is not green. Given to us is three blocks the top most block is green, the bottom most block is not green nothing is said about the block in between.

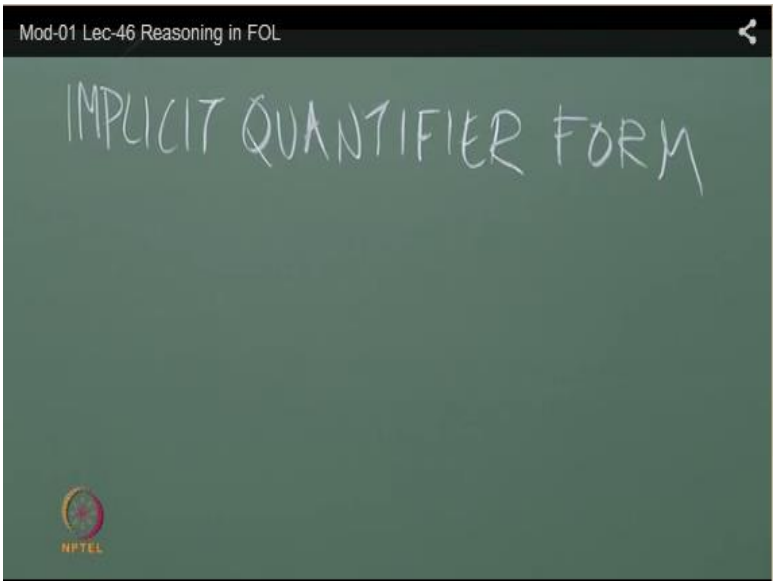
So, I will leave it as a small exercise for you to work out, but this is the kind of thing you may want to prove which is while sometimes you may need a rule which say something like that essentially. So, this is the process of forward chaining we are looking at. So, forward chaining says basically stringing together a set of facts by moving from the left inside of a rule where, we are using more exponents. So, the right end side given alpha, you are chaining into beta then given beta there is beta implies gamma then you chaining into gamma.

So, you moving in the forward direction essentially what is given to you and what can be derived. You move in the forward direction and this what we have doing here essentially. Even this formula we can derive this formula, given this you can derive this. So, you have moving in the forward direction. The first thing you want to do is avoid this guess work. You do not want to use this rule of universal because if you go to use it directly, then you would end up during lot of guess work as to what should I instantiate the values of the variable tool essentially.

It is clear if you look at this formula is that, you know here we are talking about x here we are talking about. So, it make sense to somehow say that I am instantiate x to because while looking at both this formulas. So, the difference is the universal instantiation rule only looks at this formula then says I must produce this whereas, if you also look that is then you would know the you have to produce this essentially.

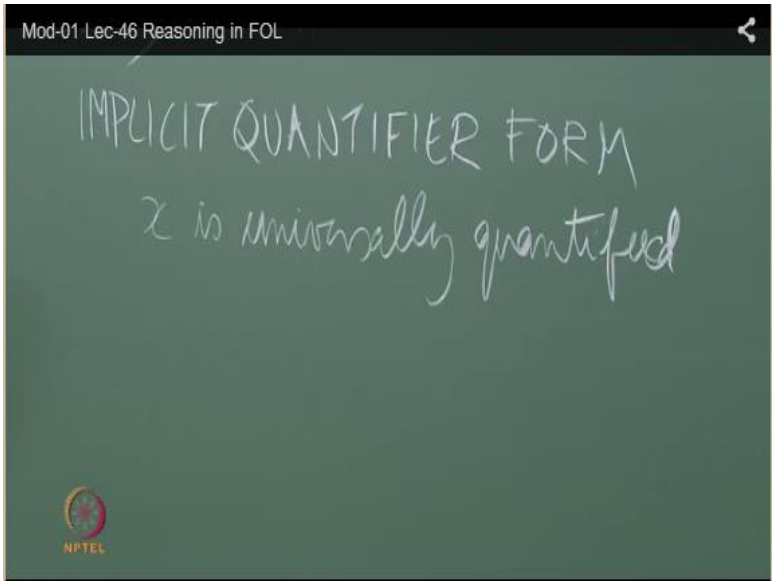
So, let us define a new role which combines these 2 steps into 1 step, but to that let us first modify our representation little bit to make it simpler for us. To do that we define something in what is called an implicit quantifier form. So, we want to express our formulas without actually lighting down the quantifies because remember that if you going to write programs to process these sentences, then you have to in worry about how to interpret the quantifies and so on. So, you will have to parse them in you know construct quantifies and then take them as a stuff.

(Refer Slide Time: 28:30)



So, instead what is normally done is to express the quantifies implicitly and for today's class we will only look at the universal quantify because that is all we have here. We are not worry about the existential quantify in the next class for a universal quantifier. So, if x is universally quantified.

(Refer Slide Time: 29:09)

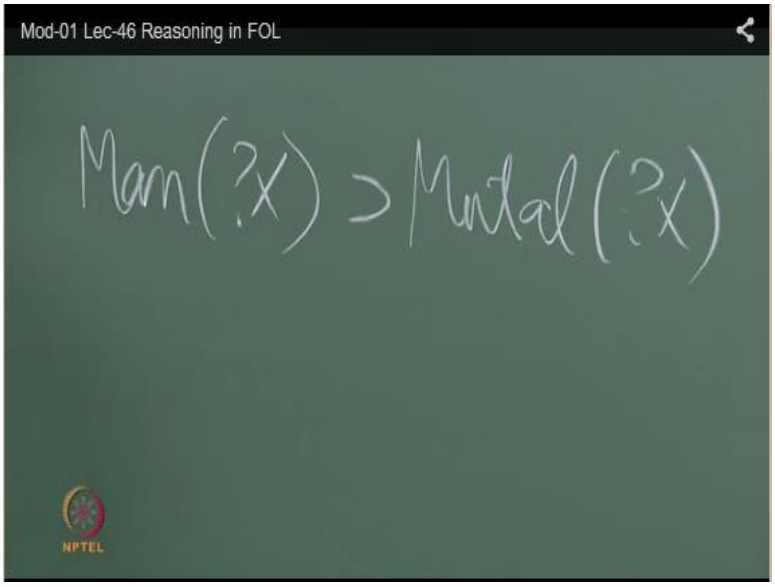


So, for the moment let us assume that we know the nature of a variable. So, in our logic that we have defined there only 2 kinds of quantifiers and in the sentences there are no free variables. So, start with there are only bound variables and variables are bound either by a

universal quantifier or by a existential quantifier. So, let us assume that we know the nature of the quantifier and we can somehow say that is a universally quantified variable, which is easy in sentences like this, but in more complex sentences we will see later it is not.

So, let us assume that we can do that than in the implicit quantifier form, we simply place it with x souse the same symbol x. So, this is just a convention which says that I am going to use the question mark to stands for a variable, which is universally quantified, which means I will now rewrite this whole thing as follows that man x implies mortal x.

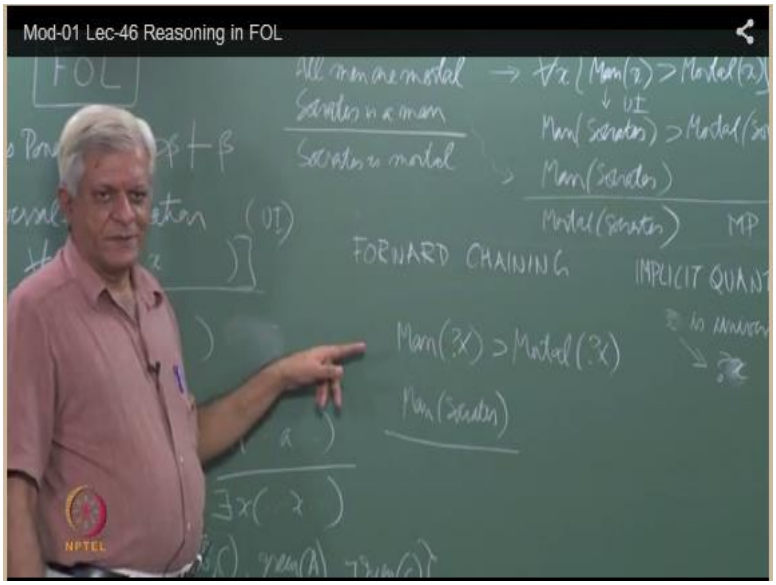
(Refer Slide Time: 30:29)



So, instead of writing is like that I am writing like this, but it does not mean that i have changed the sentence, the sentence is still the same except that the universal quantifiers implicit. I have indicated it in my sentence by putting a question mark before the variable, which tells we that this variable all these occurrences of this variable has one occurrence of a universal quantifier before the sentence is implicit. It is their like in that, but it is implicit essentially.

You can see that this is basically motivated to simplify the processing that you do when you write a program to do forward chaining essentially. So, given this and given this you can see that it is becoming a little bit simpler to do this processing. Somehow if I could have a version of more exponents, which did not require an exact match, but which allowed us to substitute anything for a universally quantified variable, then life would become simpler. I would say I am trying to match this with this, how can I do this?

(Refer Slide Time: 31:53)



If you can simply say x equal to you can do it and then this will become like the step of universal instantiation. Again I would like to emphasis, I am not doing it independently first I am now looking at a modified version, which will have this built into it essentially. So, in some sense I know the target to which I want to instantiate because I have this fact here essentially.

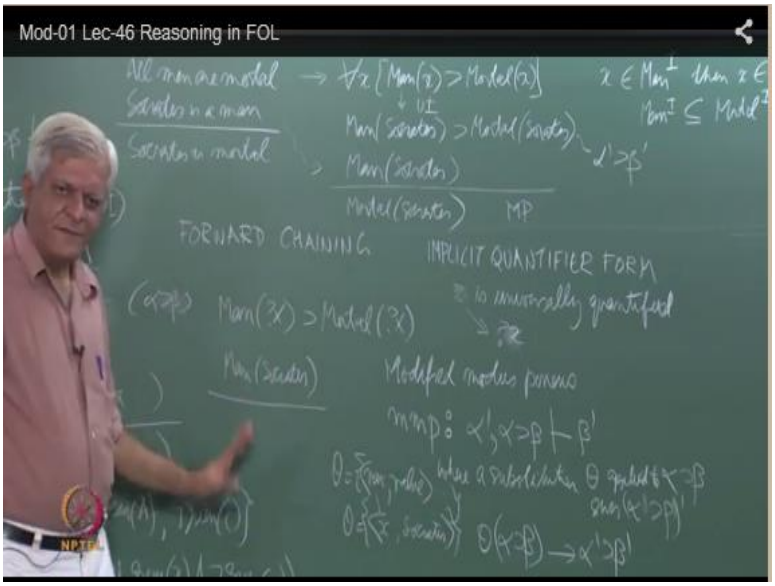
So, this rule is call in that calling modified and the rule is as follows. It says that let us say m, m p and it says that from alpha prime and alpha implies beta, beta prime where, a substitution let us say theta applied to alpha gives alpha prime. So, we are saying the substitution is a function which when you applied to any formula, then it replaces some occurrences of variables in that formula with other occurrences essentially. So in fact, when applied alpha implies beta, it gives you alpha prime implies beta prime or it gives you alpha implies beta whole prime.

What is the substitution? A substitution is basically a set of variable value pairs and in our case, the variable is x and the value that is Socratic that is the substitution we are interested in. What does it do? It says that when I apply theta to alpha implies beta I am looking at this specific example. In general if you apply theta to any formula this is just one kind of a formula which we have interested in. It gives you alpha prime implies beta prime and how does it give you that it substitutes.

So, what is the substitution telling we it saying that you substitute every occurrence of x with

Socratic essentially. So, if I apply the substitution to this if this is my alpha implies beta, then my alpha prime implies beta prime is this. So, this is alpha implies beta and this becomes alpha prime implies beta prime under the substitution theta is equal to x.

(Refer Slide Time: 35:39)



So, how we are saying is that the substitution tells you what to substitute for the set of variables named in the substitution. In this substitution only one variable is named which is x and a very simple substitution is given which is, but it could be anything else. I could substitute any for example, arbitrary term for it I could say the grandfather of which means let us say the father of mother of Socratic. That is a term remember that is also a term I could have substituted this in my formula, then I would have formula which says if grandfather is the man, then grandfather is mortal.

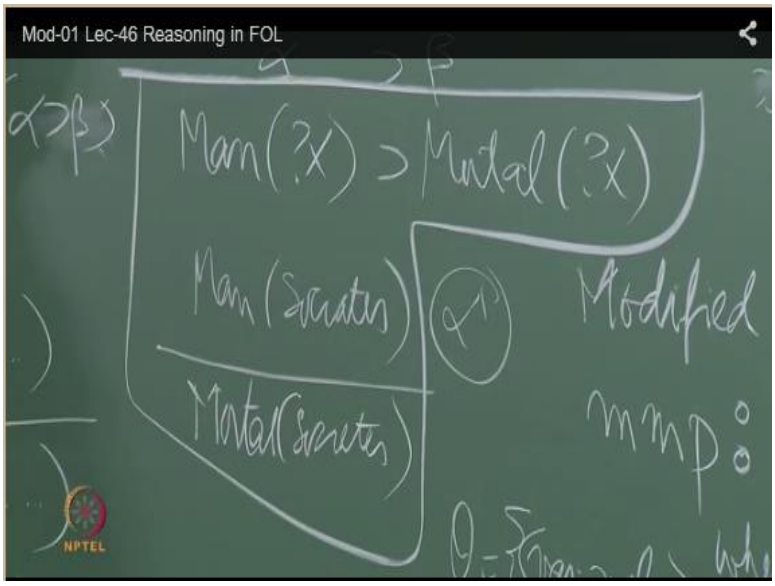
I would have got that statement instead the substitution that I am doing here is which for x. So, I am getting this statement that if is a man then is mortal that is alpha prime implies beta prime. I already have alpha prime here which has got inside. So, now I can apply more directly essentially. So, this whole 2 step process is collapse into one step process in this modified exponents rules where, thus instantiation is taking place somewhere inside by means of a substitution.

It is saying that if you given a formula alpha prime and a formula alpha implies beta, if you can somehow found find the substitution which will make these two equal. So, the technical term you uses the unifier, which we will study in the next class, but for the moment just let us

think of it does a substitution, which will make this formula equal to this formula which is what the universal instantiation step is doing.

So, the technical way of saying that is if you can unify alpha prime with alpha and you can always unify by means of a substitution, which means you make them look the same. Then you can imply beta prime directly, which means apply the same substitution to beta. What is beta here, mortal x man x implies mortal x. So, this is alpha and this is beta and this is alpha prime. I can make alpha prime and alpha the same by saying applying the substitution x equal to this. This rule says apply the same substitution to the right hand side of the expression and directly infra mortal.

(Refer Slide Time: 38:50)



So, this step is, this is the modified rule that says that I can substitute directly here jump to this conclusion in some sense without having into go through that step of instantiation explicitly, but this is a simple example. It has unary predicate with only one argument and so it is easy to understand. In general modifies says that given any alpha prime and anything of alpha implies beta.

In fact, I do not need have to call it alpha I could call it gamma. I can unify gamma with alpha, but it traditionally we see alpha prime in alpha. If you can somehow make them the same, then go higher and make the inference and apply the same way of making them the same to the conclusion as well essentially. So, apply the same substitution to beta to get beta prime.

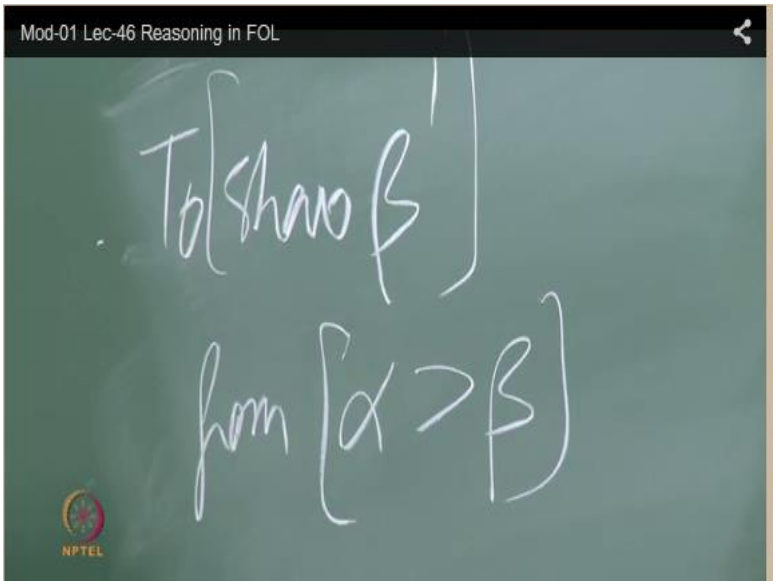
So, if I have a simple database and by simple I mean, I have only statements of this kind something x implies something x. So, b b x implies 2 x which one let us say there is only 1 variable could have more than 1 variable essentially. So, you could have a statement which says friend x y implies friend y x essentially, then I could say Suresh is the friend of Ramesh. I could infer that Ramesh is the friend of Suresh essentially by applying this rule. So, if I have a simple database of the kind where, I can apply modest exponents rule then I have a simple mechanism for keep applying rule repeatedly till i generated formula that I have essentially.

Let us assume we are using only modified modest exponents because our data is of that nature, but still the question of which facts to apply to is still open. You still do not know there is no sense of direction saying that this is an inference I should make. So, the same problem of huge branching factor in the forward direction exist in forward chaining essentially. So, can we do backward chaining. So, what would backward chaining be like. So, let me just give you a hint here and we will take it up the next class.

So, backward chaining needs to distinguish between statements which are there in the database. Let us call them facts and statements which you want to show to be true essentially. So, forward chaining is simple you have a set of facts given to you which includes no rules and statements like all these kind of statements. You can keep adding new statements and you can terminate when the required statement is produced essentially.

If I want to do backward chaining what is the mechanism that I have been used to do backward chaining. So, let me take the same rules, but I want to express something like this. To show what I am trying to highlight here is that you need to distinguish between what you have and what you do not have.

(Refer Slide Time: 42:46)



What you have is the set of facts, what you do not have is something you want to show to be true. So, we could use a marker this is only for programming purposes. It is not part of the language. It says that if you have a goal beta prime and if you have a rule of the kind alpha implies beta. So, we call this a rule left hand side, right hand side, then you can replace the goal of showing beta prime with the goal of showing alpha prime essentially.

If you implementing this we will need to keep the facts that we have in one set of formulas, which are separated from the goals that we are trying to show. So, we can do that simply here on the board by putting a marker called show. So, when you have marker called show it means, it is a goal it is not something which is true. It is something you want to show to be true and this is saying that to show that beta prime is true, if you have a rule of a kind alpha implies beta and you can do this unification process.

So, you can find the substitution then you can reduce it to a sub goal of saying show that alpha prime is true or does a translate to our problem here. It says that if you want to show that Socratic is mortal and if you have a rule which says all men are mortal, then substitute or replace the goal of showing that Socratic is mortal with the goal of showing that Socratic is a man. This is the process of backward chaining.

So, from goals you are moving to sub goals essentially. Now, you can see that this process is going to become a little bit complicated. If my left inside had more than one statements. So for example, you can define a grandfather by saying that x is a grandfather of y, if x is a

father of z and z is the parent of y, you could define it like this. Now to show that x is a grandfather of y, you would have to show both those things that show that there exist a z whose father is x and there exist I mean that same z is a father of y essentially.

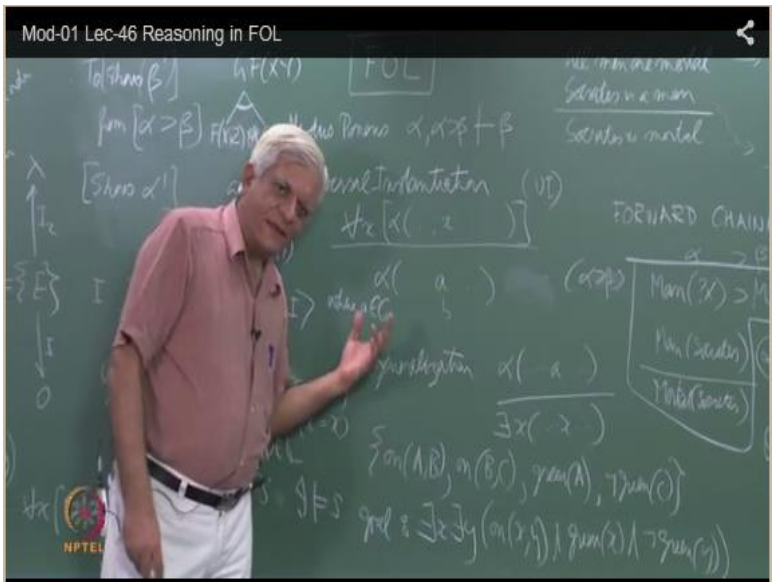
So, grandfather x y if father x z and p stands for parent z y. In the forward direction I would have this fact somewhere that says Peter is the father of marry and marry is a mother of John, then I can show that Peter is a grandfather of john by going in a forward direction. In a backward direction I have to put and here. So, you can reduce the sub goal to these 2 sub goals, but there is and here essentially.

May be there is another rule which says that grandfather of x y. So, let me say instead of parent I have mother here father x z father z y I have 2 separate rules. This rule says that x is a father of father of y, this rule says that x is a father of mother of y essentially. For some reasons I have these 2 separate rule, then you can see that to show that Peter is a grandfather of John I could either use this rule or I could use this rule. So, you can see that it is becoming an AND-OR tree essentially.

You could either use one rule or you could use another rule essentially. So, backward chaining has this complication. You know that how do you get that to think like this. So, essentially it maps to AND-OR tree and we will look at that in the next class essentially. Essentially, what backward chaining is saying is that from a goal you can move to a sub goal essentially.

So, I will stop here and in the next class we will look at this process of finding the substitution here and there is a very nice algorithm call unification algorithm which some of you must have studied for doing so. We will see how this kind of AND-OR trees is a tackled. We will also briefly mention as to this kind of backward chaining process is what really this language prolonged does. Just to complete the course may be we will look at the resolution method in first set of logic and the motivation for that method can be this problem that we have here.

(Refer Slide Time: 48:21)



So, let me reveal to you that this sentence is indeed true, but it cannot be proven either by a forward chaining or backward chaining. Those methods are not complete, but resolution method we showed this or at least we talked about it. In proportional logic case also it is a complete method essentially which is why it is so attractive essentially. So, we will end the course with resolution method for first set of logic, which is in the next two classes I think.