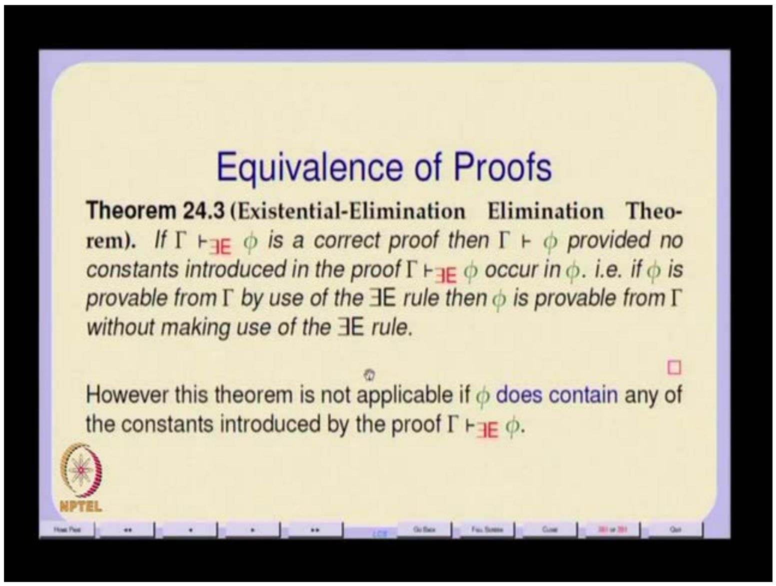
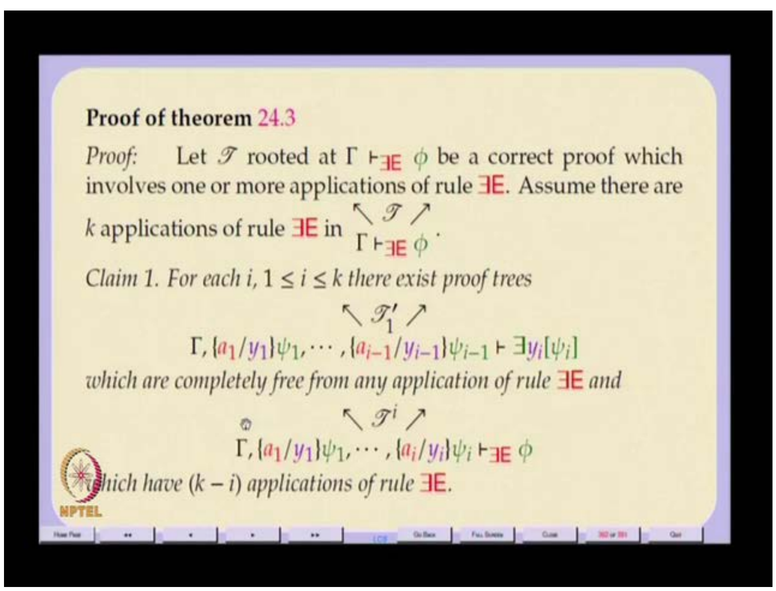
Logic for CS
Prof. Dr. S. Arun Kumar
Department of Computer Science
Indian Institute of Technology, Delhi

Lecture - 25
Normal Forms

(Refer Slide Time: 00:35)

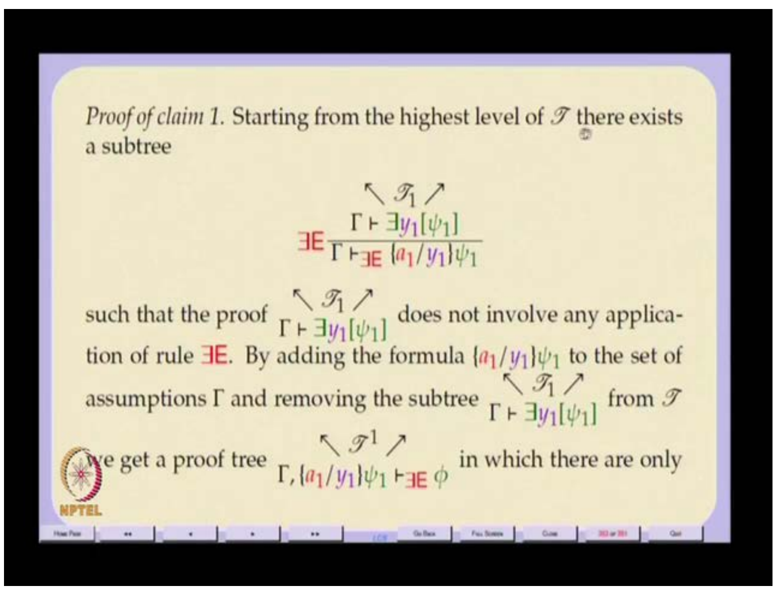


(Refer Slide Time: 00:40)



So, last time we were looking at this Existential Elimination theorem. So, I have sort of change the proof a bit there are some bugs that I realized in my original version there are very settled bugs. But, essentially the proof in its essence is correct but the whole point is this in a you start with a proof tree. Which, use there exists elimination and let us assume that there are some k applications of there exists elimination. Then, this k applications need not be of the in the leaves of the proof tree that is that is the first thing. So, what you need to do is you need to identify an order of the applications based on the level in the proof tree. So, that you start off, by eliminating the once closest to the leaves and once you start eliminated all those once that are closest to the leaves. You essentially you get a you get different proof trees because you will now add those things as assumptions. So, for so essentially what we are saying is you start from the highest level of the tree.

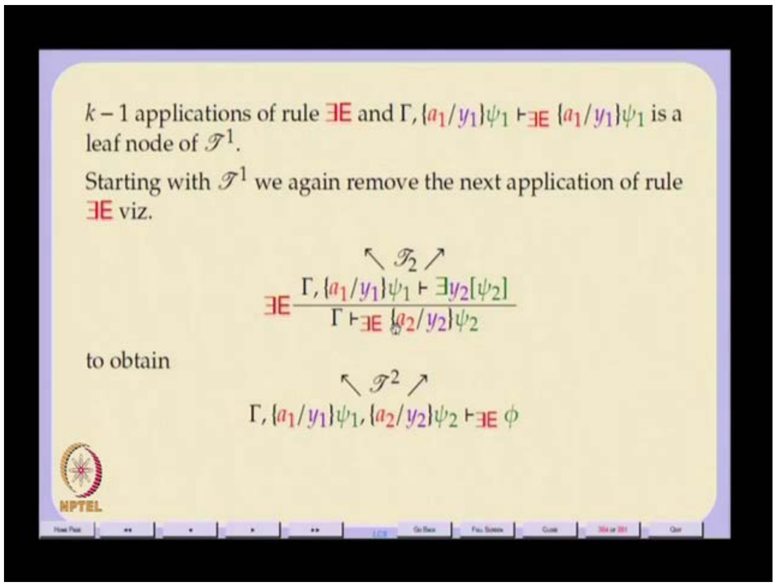
(Refer Slide Time: 02:14)



And, there is some first application of there exists elimination. So, you let us say that is a sub tree T1which and if it is an application of there exists elimination. Then, it is a sub tree T1 rooted at a formula like there exists y1 psi 1. And, because of which when you applied there exists an elimination you got this first constant a1. Now, it is clear that this tree T1 does not have any application of their excess elimination. So, now when you add a1 for y1 psi one to the set of assumptions you remove this entire tree T1. But, you retain this essentially this node with the assumption that a1 for y1 psi 1 is an extra assumption.

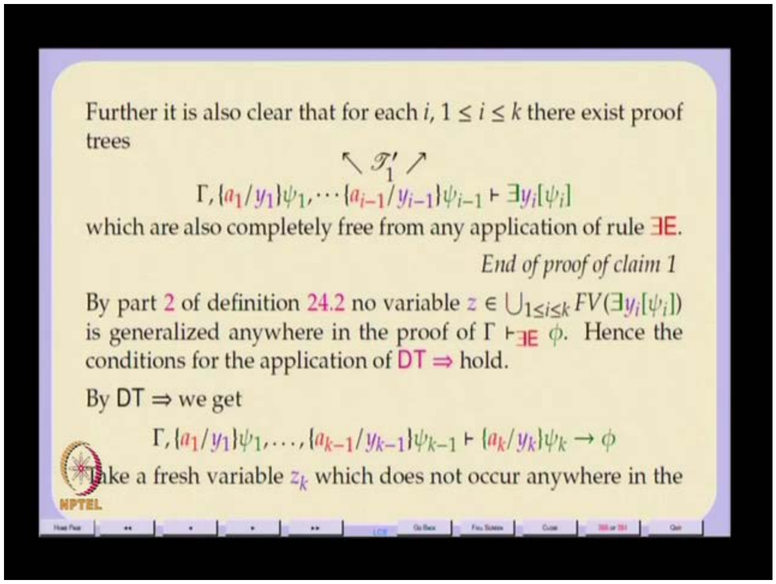
So, then that becomes a leaf of essentially a modified tree. Then, on this modified tree you look for essentially the second highest occurrence of there exist elimination and remove it replace it. And, add this an extra assumption a2 for y2 psi 1 then, look at the modified tree and then look at the highest and so on. So, there is a order which there is a total order or which you can impose starting based on the levels starting from the leaves and proceeding towards the root of the tree.

(Refer Slide Time: 04:00)



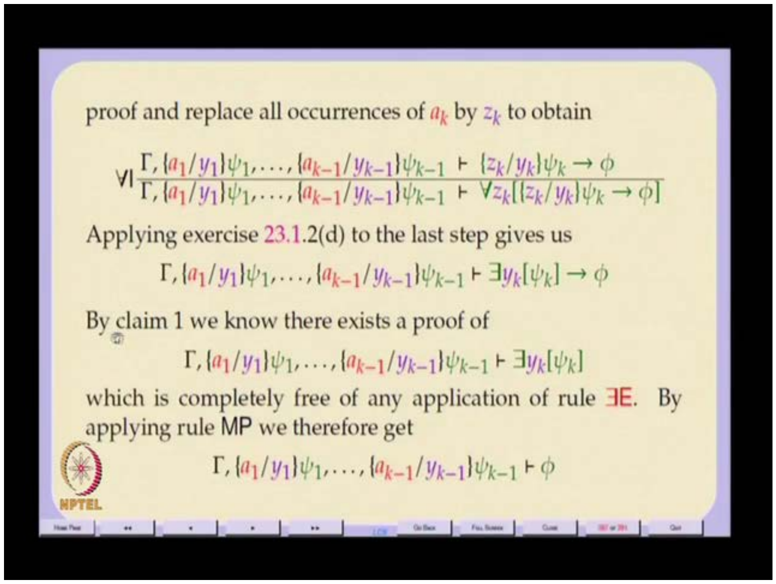
So, each of these trees T1, T2 each tree Ti that you eliminate actually is of a modified tree of the original proof. So, this claim here essentially says that for each i 1 there are k applications of the rule in the original proof for each i there exists proof trees T1 prime. This, should be this should be Ti prime gamma with constants a1 to ai minus 1 replacing y1 to yi minus 1 proves there exists yi psi i. And, this proof tree does not have any application of there exists elimination. And, the act of removing this sub tree is therefore getting modified proof tree means is that these all these proof trees Ti primes are completely free from any application of the rule there exists elimination. And, the modified original proof tree with each modification at the i-th at the i-th stage you get a proof tree Ti T superscript i. Which, has these i as some i extra assumptions and which proves phi. And, of course there are k minus i applications of there exists eliminations in this proof tree that is.

(Refer Slide Time: 06:22)



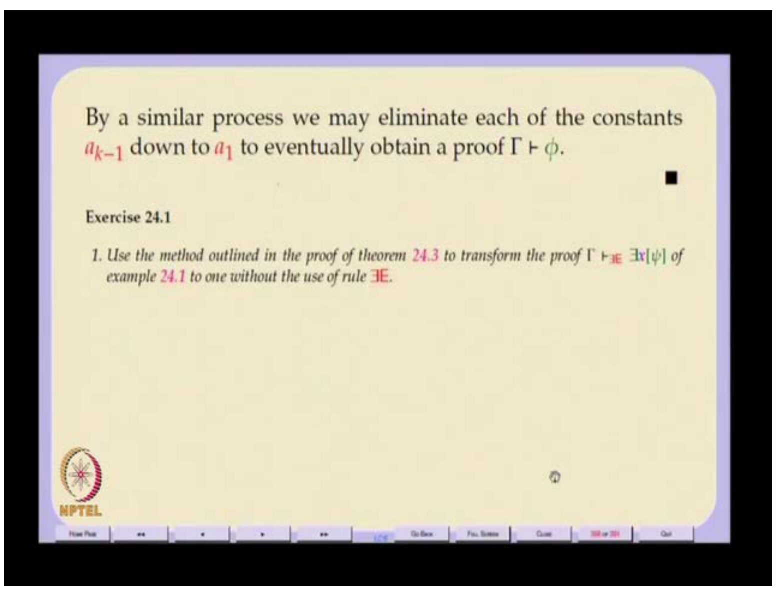
So, this claim is important in my in my original proof the other day I actually assumed that all these applications could be taken in the leaves. But, that is not true because if I had existential quantifiers deeply nested inside the formulae. Then I will have to extract out all the root nodes and go through several sets before I eliminate them. So, this is the modification so essentially this prove of this claim can be proven and it goes into several so there it ends. And, after that we essentially use the fact that that each of those Ti's did not use there exists elimination in order to apply this modus ponens.

(Refer Slide Time: 06:40)



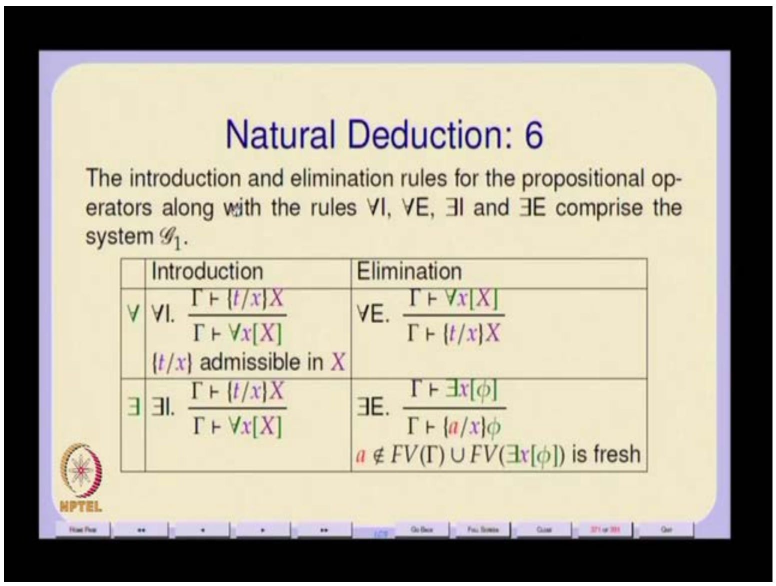
So, here I had written something which was wrong so by claim 1 we know that there exists a proof of this a1 to ak minus 1 psi k minus 1 proves their exists yk psi k. You, know that there is a proof of this which is completely free of any application of this proof. And, you also have obtained through the last step and deduction theorem you have applied you obtained this. And, therefore you can now apply the rule MP and, you can get this get this result.

(Refer Slide Time: 07:26)



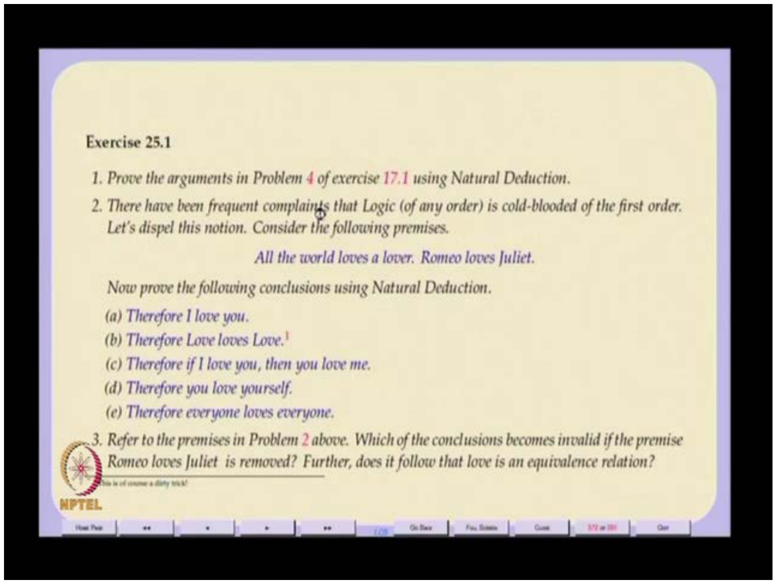
And, now with this new tree proof tree now this proof tree, you can now start eliminating ak minus 1 you will get a new proof tree and so on and so for till your eliminated all of them. So, this is by the way I have also added a large number of exercises which you should look at some of them you may have to submit. So, now what is means is that so there are so we have got existential elimination and we have got existential introduction. So, essentially what we have is of full natural deduction proof system that is it there ends the matter.

(Refer Slide Time: 07:58)

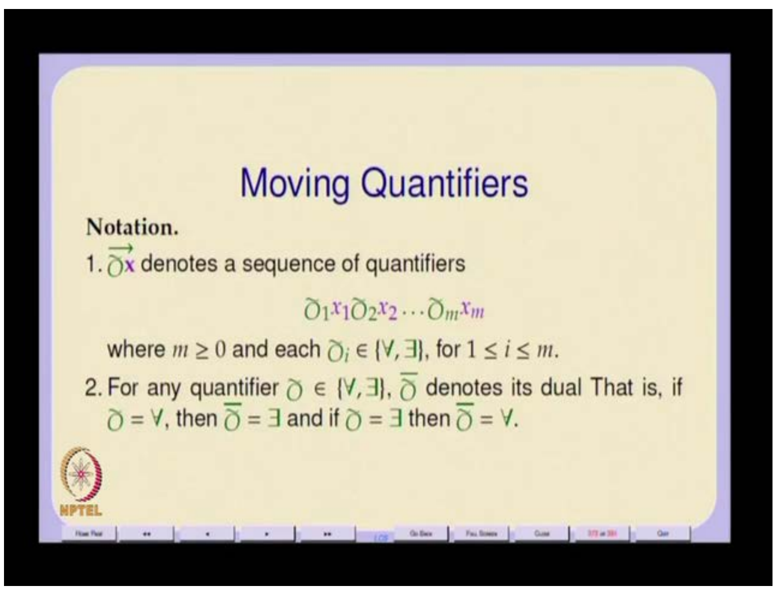


As far as Natural Deduction is concerned there ends the matter So, you take this for all introduction and there exist introduction and elimination rules and add it to those propositional introduction. And, elimination rules and you have a complete natural deductions system except that the proofs in natural deduction. Now because of there exists elimination will look different from the proofs of the universal system. Whereas, in the propositional case we could actually prove every rule of the natural deduction system as derived from the Hilbert style system. So, now the proofs are different there are a lot of this is all for you to have fun.

(Refer Slide Time: 08:48)

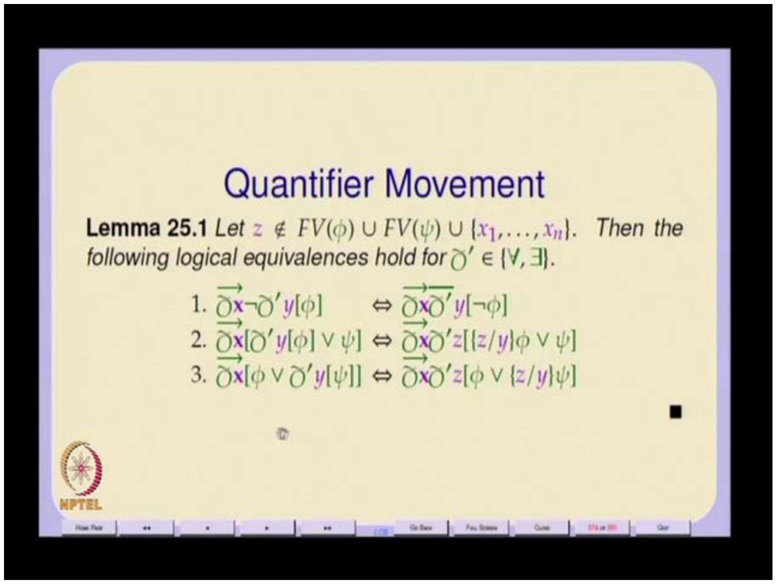


(Refer Slide Time: 08:51)



So, now let us get on to essentially the main thing now of Moving Quantifiers. So, I am going to introduce some heavy Notation. So, I am going to use a vector like this to denote a sequence of quantifiers where these quantifiers should be either for all or there exists any of them. And for any quantifier Q it is dual i will denote it with a bar here to show that it is clearly different from them from Q.

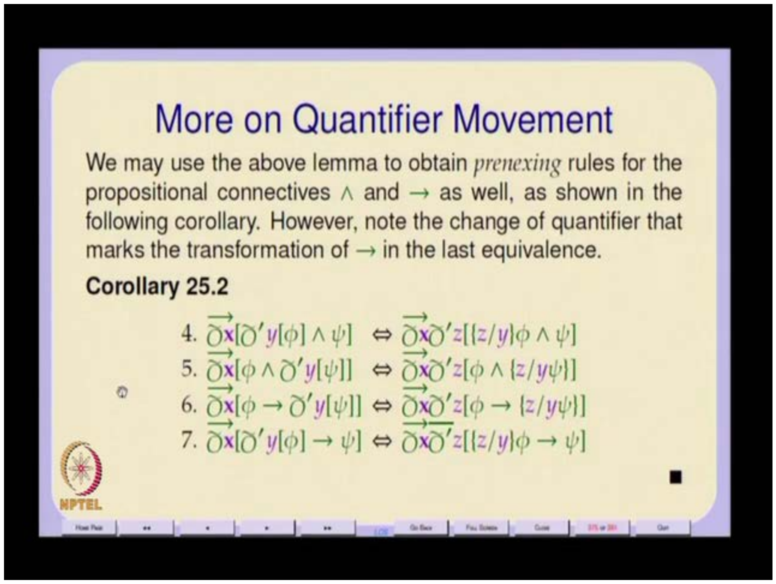
(Refer Slide Time: 09:34)



So that is so now what we have we have, this fantastic equivalences which is homework for you to prove. Notice that there are some conditions you have to fulfill this equivalence equivalences have to be fulfilled. Firstly I am assuming that there is there is a variable z which is not free anywhere in any of these and it is not any of these variables which, are essentially going to be bound. Then, for any vector of quantifiers followed by naught Q prime where Q prime is 1 of the quantifiers of course this is just like the Demorgan’s law. Then, you invert the quantifier Q prime so you get Q prime bar.

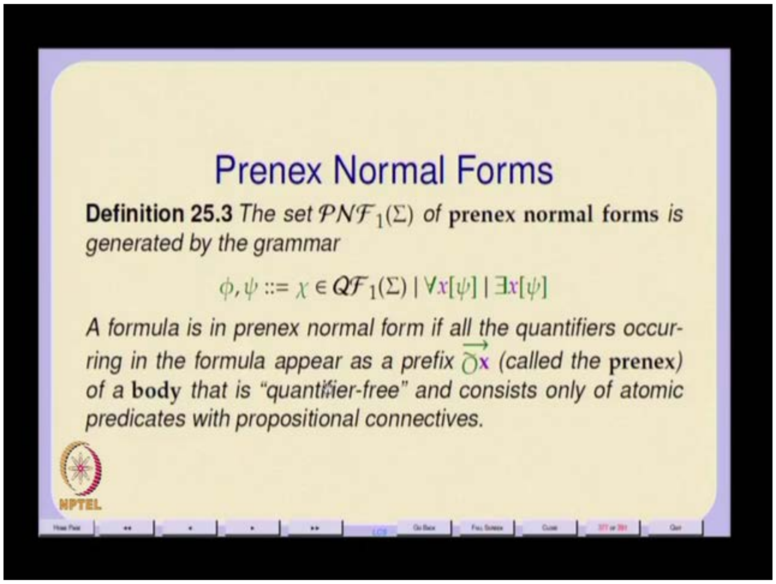
And, you may get the body of this, thing and the here you have q prime y phi or psi. Then what I can do is this I can do an alpha conversion and replace that y by a z and since z does not occur anywhere in all this I can also move this quantifier prime z outside. So, and this is this is for or and similarly if the quantifier occurs on this side then I can rename it and move this quantifier outside. So, look at these logical equivalences as operating from left to right. So, what are we doing we are moving quantifiers towards the left from the right of a unit term. And, of course once you have proved it not and or for once you have done this for naught and or it follows as a corollary that you can do similar things for the other operators.

(Refer Slide Time: 11:43)



So, now I mean now that we have got a natural deduction system and so on we can consider the full language and so we will work with a full language that is. So, here you have for and if you have Q prime and y then I replace a alpha convert that Q prime y phi two Q prime z, z for y and phi. And, since z is new I can move this quantifier out. And, similarly in this in the case of these arrow of course in arrow it includes that particular thing. That is this inversion of quantifier in the last which we have already seen we used it in our proof of there exist elimination theorem. So, this is the last one which corresponds to that and actually it works for both quantifiers the inversion of the quantifiers happens for both quantifiers.

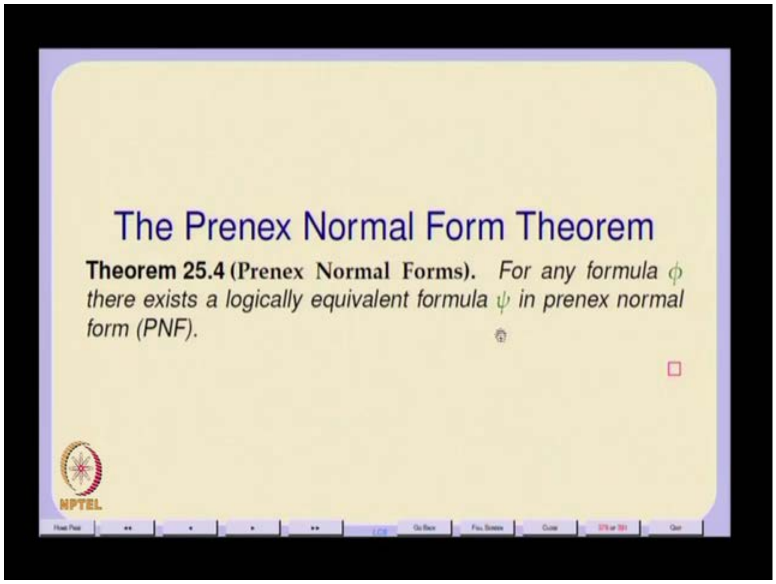
(Refer Slide Time: 12:50)



And, that is what this is suppose to once you have this if you lo at these seven rules from left to right that left hand side being replaced by the right hand side. Then, what happens in each case is that quantifiers are being moved from inside to the outer levels of the abstract intact tree I mean they have been moved closer to the root of the syntax tree. So, this seven logical equivalences essentially take care of negation and or and arrow there is no as similar logical equivalence for the bi-condition or no obvious logical equivalence for bi-condition.

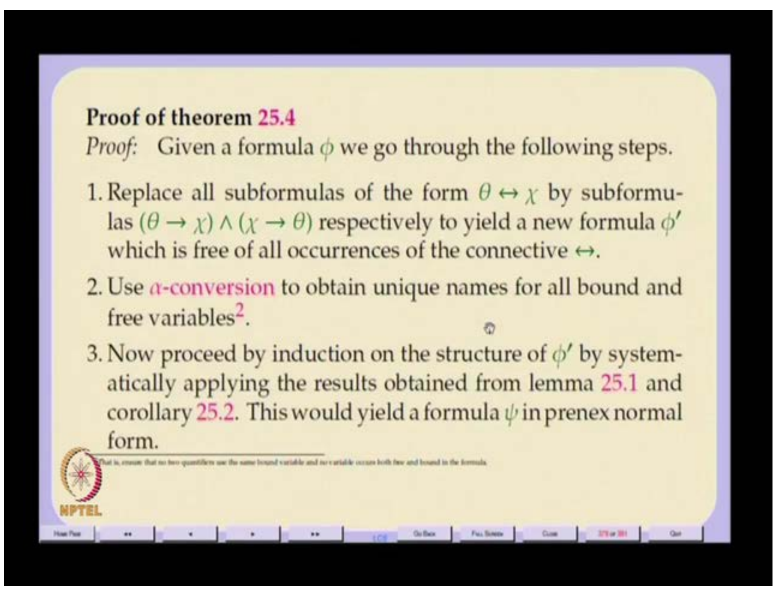
Anyways so what this means is that now we can define a sub language of the full language of first order predicate logic called the set of Prenex Normal Forms. So, this is so you take any quantifier free formula in the quantifier free, subset of for first order logic. So that, is chi belonging to k of 1 sigma and then you can keep adding quantifiers to the left. So the formulas that are generated are such that, there is a sequence of quantifiers. And, then there is a quantifier free body that is so that a prenex normal form. So, our formula is in prenex normal form if all the quantifiers occurring the formula appear as a prefix. Which, is a vector of quantifiers which is called the prenex and then there is a body that is quantifier free. And, consists of only atomic predicates and the propositional connectives not or and so on.

(Refer Slide Time: 14:51)



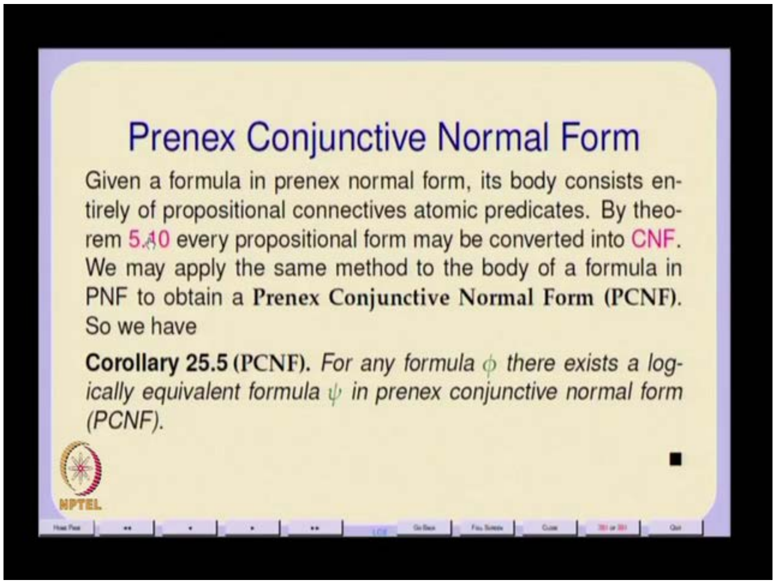
So, The Prenex Normal Form Theorem says that for any first order logical formula there is a logically equivalent formula in prenex normal form.

(Refer Slide Time: 15:02)

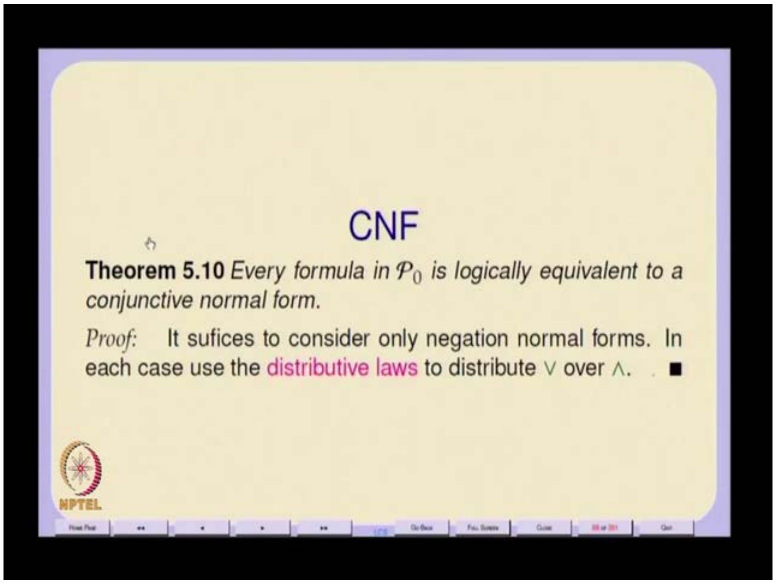


And, the justification is essentially just that where when I have bi-condition I replace the bi-condition by two conditions and after that so, I replace all bi-conditionals. But, the two conditionals given by this and after that what I do is I use those seven logical equivalences essentially to move the quantifiers. So, when I basically I use for one things it is if I am moving quantifiers and forming a sequence of quantifiers I want all the bound variables to be distinct for example. So, which means that one has to use alpha conversion buT1 but now we know that alpha conversion can be done to those to something we to for granted all our lives till we saw the proof of alpha conversion. So, that you can get unique names for all bound and free variables so there is absolutely no clash of names. And, then what you do is proceed by induction on the structure of that formula to move all the quantifiers by applying appropriate logically equal nesses here.

(Refer Slide Time: 16:14)

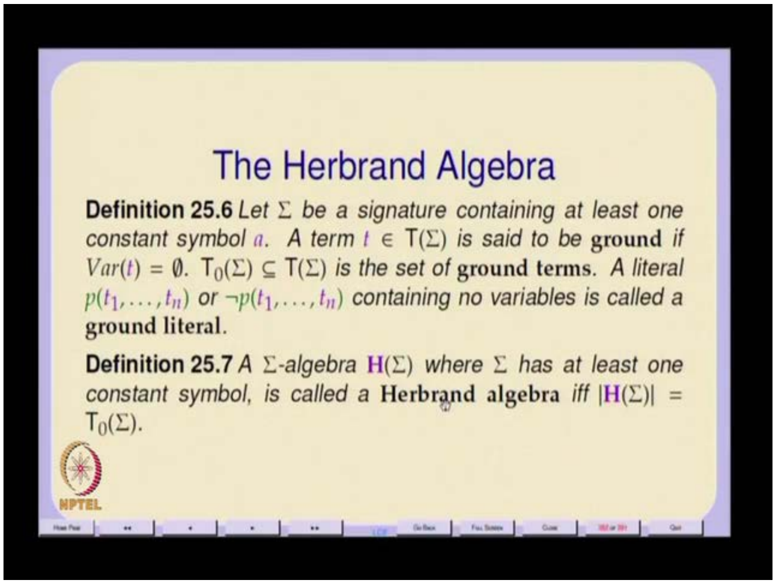


(Refer Slide Time: 16:15)



So, this would yield a formula of psi in prenex normal form basically. What, we also have from propositional logic is that every formula in propositional logic can be converted into a conjunctive normal form. So, if basically one just uses the distributive laws to distribute or over and one uses negation and so on so forth. And, so now what it means is since we have a prenex normal form we can take the body. The body is just a propositional form and use the conjunctive normal form algorithm to transform it into a prenex conjunctive normal form. So, one obvious corollary is that for any formula phi there exists a logically equivalent formula psi in prenex conjunctive normal form.

(Refer Slide Time: 17:01)

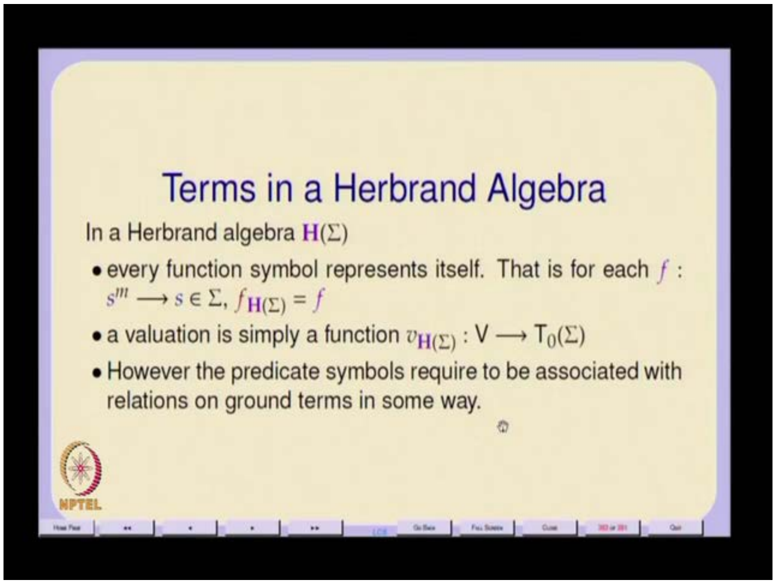


Now, we are going into the interface between proof theory and model theory one of the things that we did was that we define the semantics in terms of certain structures. So, let us I do not since I do not have any link think I will not go by there. So, now what I can also do is I can for any signature containing at least a one constant symbol. Basically, this means a function of as it is 0 that is how let us do it. A, term is set to be set to be ground if it is variable free if there are no variables in the term. So T naught of sigma is the set of ground terms T sigma is the entire term algebra I mean looseness we are speaking the t naught and the T are all there are not things in ground really there are all things is violet color that is by my color coding conventions. So, a literal is a ground literal if it is on atomic predicate or a negation of an atomic predicate in which all the parameters are variable free.

So, a literal is a so as we did it in the case of resolution and proposition logic you have the notion of the positive and a negative literal. And, now what we can say is we can think of this T sigma itself as a carrier of a sigma algebra. So as a domain of sigma algebra if, T sigma itself is a domain of a sigma algebra. And, of course T sigma is closed under functions of sigma under the all the operations and sigma. So, therefore T sigma is a natural carrier for sigma itself. So, you consider the algebra instead of considering an algebra a sigma comma a you consider the algebra comma sigma T sigma and that is a Herbrand Algebra. So, we will call that H of sigma so in particular so though important thing here to realize is that there should be at least one constraint symbol. Because, what happens otherwise is that your essentially applying functions on functions and functions and that can be infinite regress. So, if you want to have a, bases somewhere there has to be constraint symbol. Since, you are not going outside for a model you are looking for models within the language of terms itself. So, we do require some ground terms if we did not have any constant then there would not be any ground terms. There, would probably there would only be terms with variables in them but there would be no ground terms.

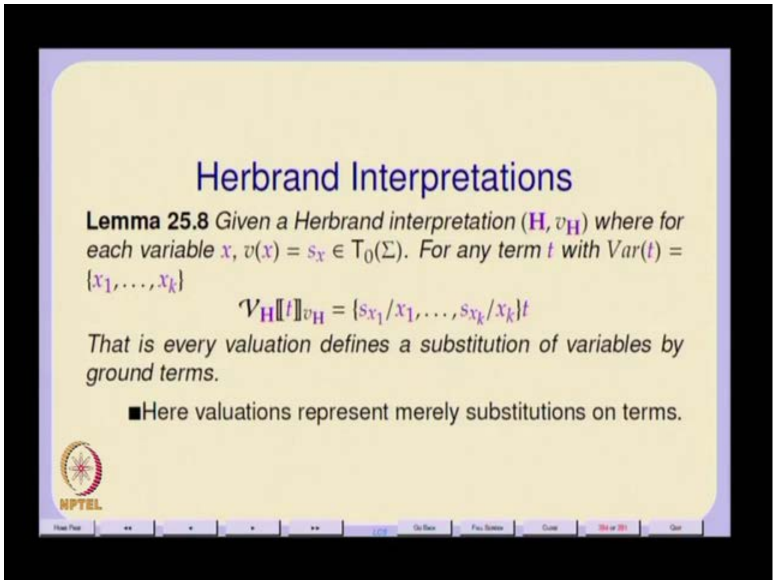
So, that is why these it is important to have at least one constraint. And, so it is quite remonstrant of whatever you have done about many of you must have done something's about freely generated groups. So, it is essentially something like that it is a free algebra generated from sigma itself. So, that basically what I am saying is there is no brown color anywhere everywhere its violet the carrier set is violet in color and not in brown. So, this is called herbrand algebra and it is and essentially the domain of the algebra are just a set of all brown terms. And, if there was not constant symbol then this set of ground terms would be empty that is a problem. So, you wanted to be non-empty so that is so you have this.

(Refer Slide Time: 21:23)



So, you take a Herbrand algebra so now what we can do is now of course we talk of valuations. So, a valuation now will be a function from the variables be to essentially ground terms of result. So, any so for every so essentially what we are saying is so I can define if I can define an interpretation over H of sigma such that every function symbol f in the signature is interpreted as itself in H of sigma. And, a valuation is simply a function which associates with each variable of ground term. So, the only thing the only difference is that we have not associating anything with the predicates symbols so at the moment we are leaving the predicate symbols alone.

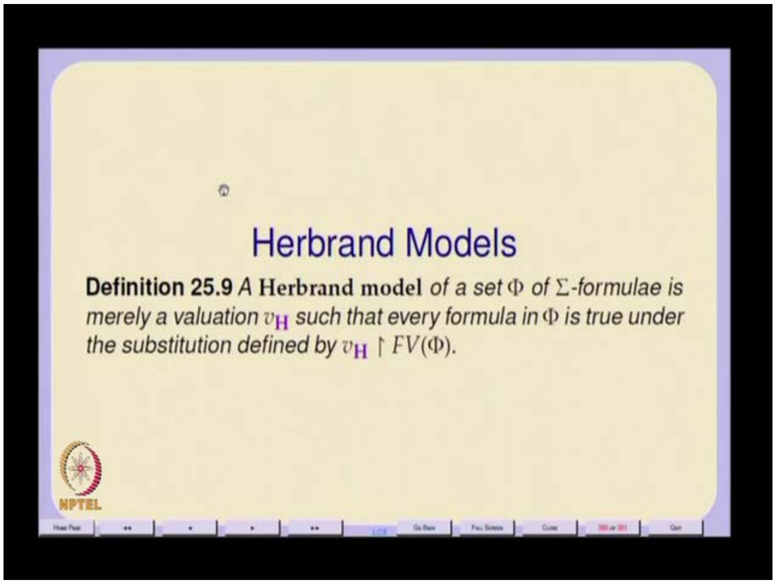
(Refer Slide Time: 22:37)



So, now a Herbrand Interpretations is just a herbrand algebra here I have just a herbrand algebra this should be a H of sigma. But, often I will just this sigma understood I will not mentioned it and then there is a valuation. Therefore, each variable x I have a term a ground term that is why its purple in its violet in color sx and T naught of sigma. And, so for any term T which has a variables x1 to xk. If, you were to evaluate this term in this valuation Vh. Then, what you are essentially going to get is a substitution is a pure syntactic substitution.

So, every occurrence of the variable x1 will be replaced by the terms sx1. Of, course sx1 itself of course the sx1 in this case would be ground terms so it will be variable free. So, what this means is that every valuation is essentially defines the substitution one of the things I did when we did all those coincidence lamas and so on and so forth. What, we showed how variations and valuations are equivalent to substitutions in a certain sense. But, now when your valuations themselves are in the same set of terms then everything collapses to this substitutions the only thing here it is not arbitrary substitution it is a substitution of ground terms for variables.

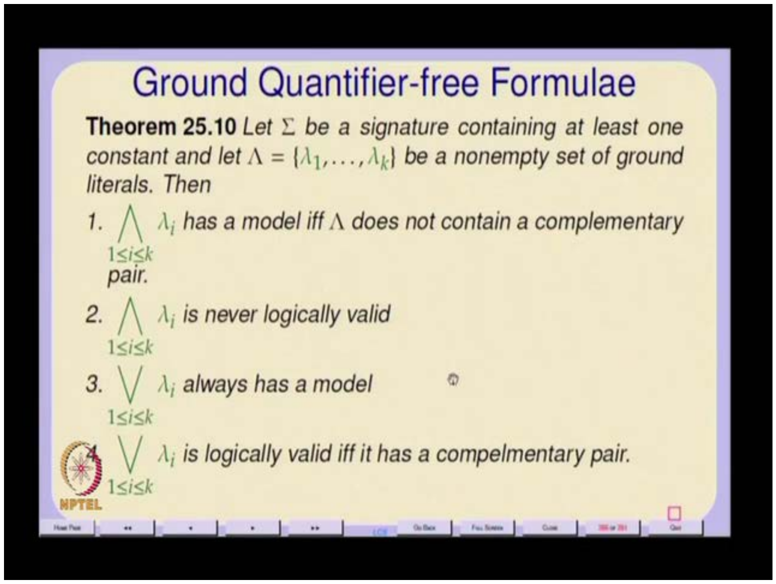
(Refer Slide Time: 24:41)



So, valuations represent merely substitutions on terms so we would say so now we have still not actually interpreted the predicates symbols. In this a Herbrand Models of a set phi of sigma formulae is a nearly a valuation I mean this can associate it with the valuation vH Which makes every formula in phi true. And, if it makes every formula and phi I mean phi true then you are not really interested in the entire set of if i is infinite set of formulae. Then, it will anyway only have a finite set of free variables. And, if it has only a finite set of free variables then you only need to consider the valuation restricted to that finite set of free variables.

So, which is essentially a substitution because a substitution strictly speaking is a finite replacement of variables it is a almost everywhere the identity function. That, means you have an infinite set of variables but what you are saying is there are only a finite set of those variables which, are which are being replaced by non identical terms. Which are not being replaced by themselves. So, you will so in the case of any valuation vH essentially a herbrand model of a set phi will be restricted to the substitution created by restricting this valuation to the free variables of phi.

(Refer Slide Time: 26:15)



Now, supposing so let sigma be a signature and let us consider a finite set of literals lambda 1 to lambda k. So, these are ground literals so there are variable free. So, there are essentially atomic predicates on ground terms or negation of atomic predicates on ground terms that is what they are. Then, lambda i has a model this, the big AND of lambda i has a model has a by a model we are saying actually we are referring to herbrand model. But, at the moment let us just write it this way a proof will actually construct a herbrand model. So, the big AND of lambda i has a model if and only if capital LAMBDA does not contain a complimentary pair. And, the big AND of lambda i's can never be logically valid and or of big OR of lambda i always as a model and big OR of lambda i is logically valid if and only if it has a complementary pair. The whole point about all this is that in order to look for models for formulae we do not need to go anywhere else.

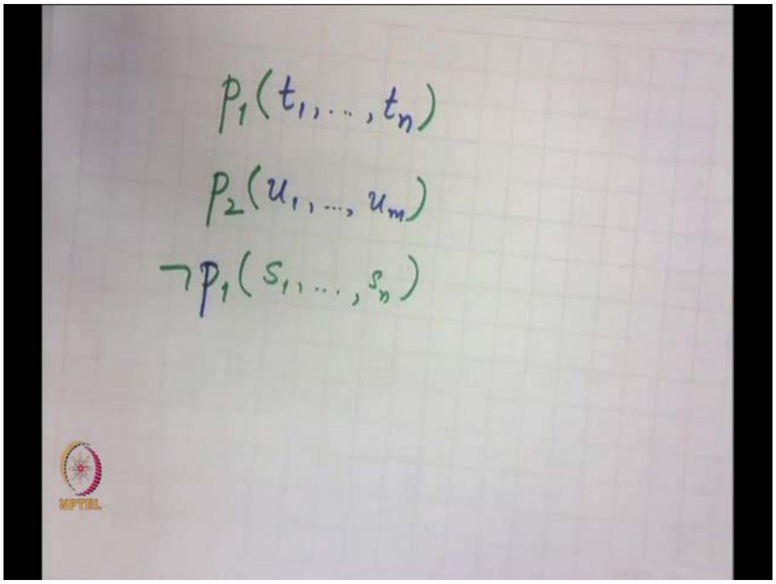
We could actually look within the term algebra itself by looking for herbrand models. So, one of the important theorems which I may not so today but I do not think I can do it today. But, which we will use this so, all this that we are doing is what leads us eventually to finding to logic programming.

Basically let us look at it this way if you have a logic program which has to do some computation it essentially tries to find some models for the axioms you give it. So, it cannot no computation mechanism can deal with arbitrary mathematical theories to find models. So, it will have to look within itself within the language of terms itself in order to determine this or whether not. And, so we require a theorem which says that there is a model in the external world if and only if there is a herbrand model. If, you have such a theorem then what it means is your computation mechanism needs to look only within itself it does not have to look outside in the rest of the world.

If, your theorem is weaker which says that if there is a herbrand model then there is also a model in the outside world. Then, it is not sufficient because what it means is that they might be models in since outside world and you will never be able to find models in this. So, this is the first theorem first towards moving towards essentially inward looking models so, what are known as term models. So, these herbrand models intruster a logic or in the larger terminology of universal algebra they would be called term models. Basically, in a certain sense all are computations are dealing only with term models. So, you can think of them as to as terms you can think of all your models of computation involving a machine. Let us say as essentially looking at that machine as an as a sigma algebra. Where, the sigma are all the operations that the machine can perform and all the expressions that are dealt with are just the terms of that sigma algebra.

So, this is a fairly powerful notion that which actually brings us to computation and in fact it brings us to the notion of computation from logic much later than people realize I mean it is only the 60s. That, some that Robinson realized and this could be done though people like herbrand had actually proved these theorems 30s. So, let us prove this theorem so what do we have to do in order to so let us take this conjunction of lambda i’s. And, if it does if it does have a complimentary pair of course there is no way it can have model I means it is already gone. So, let us assume that it does not have a complimentary. So, if it has a model then it definitely cannot have a complimentary pair that is one thing. So, let us assume that it does not contain a complimentary pair.

(Refer Slide Time: 32:28)

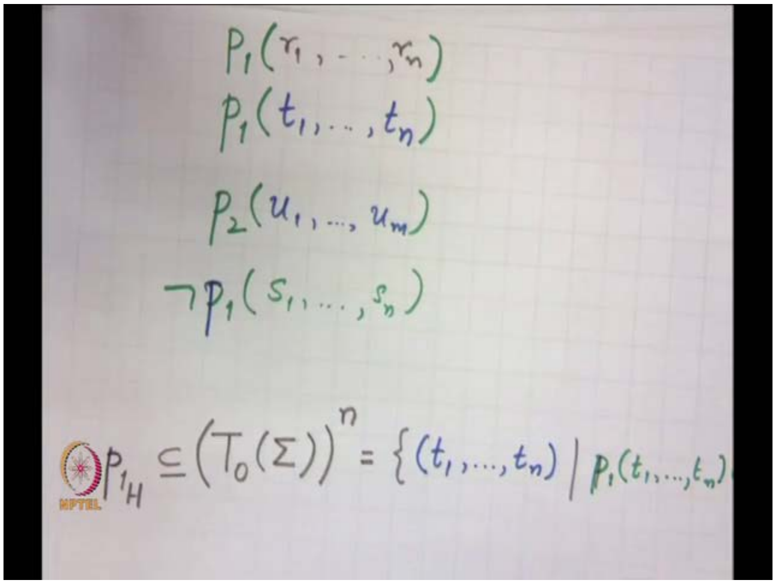


So, these lambda 1 to lambda k is all of them are of the form let us say some p1 of sum ground purple ground terms sum p1 sum p2 something maybe of some other terms. Let us say u1 to un and maybe another 1 is maybe this p1 itself. But, with some different set of terms S1 to Sn. So, this k could be a very large set I mean this k could be a very large number lambda 1 to lambda k. So, all these lambdas are essentially these kinds are atomic formulae so it should it could for example you have this. So, you have got a set of in your sigma you got a set of atomic predicates symbols with there are it is and you have you take various and you take your lambda 1 to lambda k is constructed out of various ground terms I mean this somewhere are ground literals. So, there are all ground terms which means that there are completely variable free.

Now, what you are saying is and you are saying that there are there is no complimentary pair. Note that if S1 to Sn are different from T1 to tnt element is not a complimentary we have moved away from the preposition to parameterized preposition. So, a complimentary pair these two would be complimentary pairs if only they if, they looked exactly identically. They had identical abstract syntax trees or rather this was an identical abstract sub tree of this formula. Otherwise, they would not be complimentary pairs.

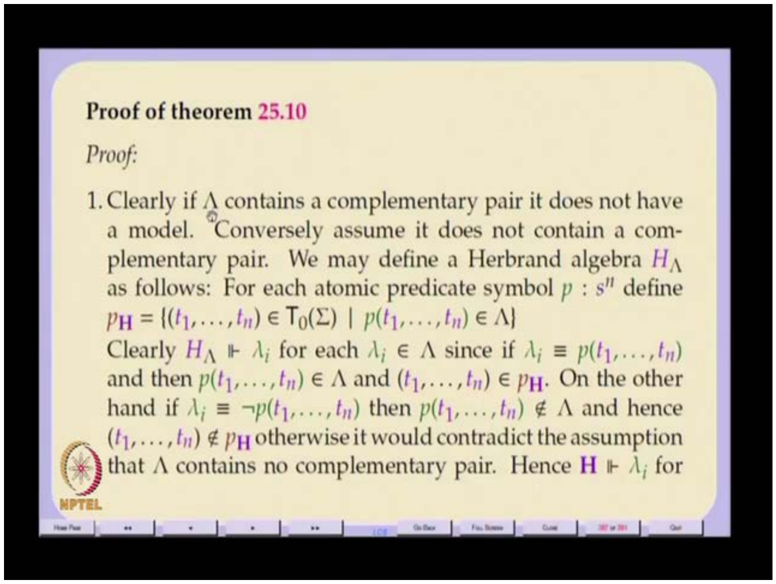
So, a complimentary pairs and parameters should also match. So, then what do we do, we essentially we have to create a model and we are going to create a model as a herbrand model. So we are going to create a model within terms itself. So, a model is going to be entirely purple in color violet in color. So, what we do is we take such we take this stopple supposing this is 1 of the lambda i's.

(Refer Slide Time: 35:10)



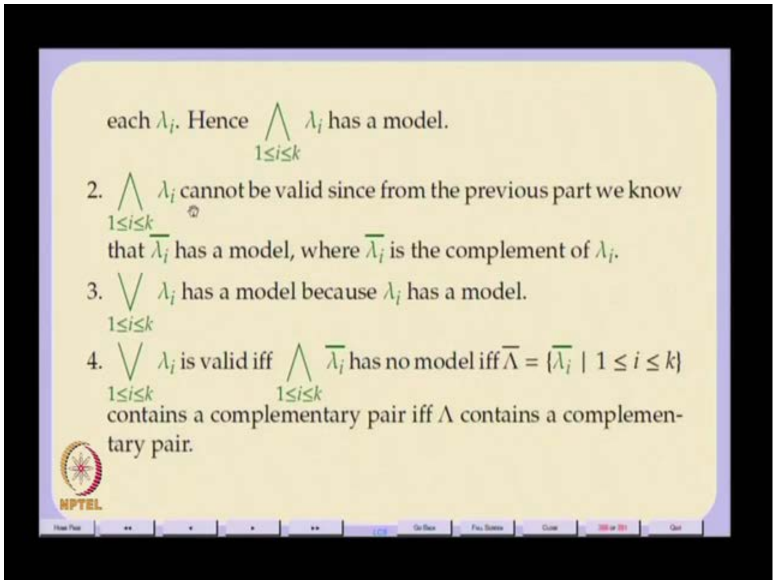
Then, our model actually should give me define a relation p1 H lets say should actually be a subset of T naught sigma the whole raise to n. If for p1 it has to have this it has to have this kind of structure. And, we have to take the ground terms so what we do is we define the interpretation of p1 as essentially this set of all these stupples t1 to tn such that p1 of t1to tn belongs to capital lambda. So, you take all these n stupples of terms where so for example you could also have some other term you could also have another term with the same atom you predicate you also occurring in positive form. Which, is some term of let us say r1 to r1 to rn where r1 to r n are terms. So, essentially what we are saying is from this set there our interpretation of p one will contain these two stupples r1 to rn and T1 to tn. But, it is not going to contain the tupple as (Refer time: 37:08) that is it is important to realize that. So, for any formula that so essentially any literal that occurs positively in lambda then it is parameters which are ground literals. Which, are ground terms there are included in the interpretation of the in the meaning of that relation of that, the relation corresponding to that atomic predicate.

(Refer Slide Time: 37:52)



So, this how you construct so now this is a herbrand model of this set lambda. You take you take any literal in this set you can show that this is a lambda. So, you can so this entire interpretation can this algebra is a model of that literal. So, that essentially I am calling this H lambda satisfies lambda i or a H lambda is a model of lambda i for each lambda i. So, there are only two possibilities for lambda i. Lambda i occurs in positive form like this, then it is parameters are bearing a model and so it is a true.If lambda i occurs in a negative form or like this, then of course its parameters are not in this. So, therefore by the semantics of the predicate naught p1 of that tupple is true. So, every literal lambda i is modeled by this H lambda. What, is there in h lambda? The only thing new in H lambda is that H is already herbrand model. So, anyway the terms are your t naught sigma there is no valuation now. Because, we are anyway talking about ground literals so it does not matter what valuation you have. The only thing you need to do is to interpret the predicate symbols and so we interpret the predicate symbols. So, every lambda i is modeled by this H lambda defined essentially like this. So, the relation PH for each atomic predicate symbol p contains a tupple t1 to tn and only if P t1 to tn is the formula is a ground literally lambda that is it. So, essentially you ignore all the negative literals consider only the positive literals and interpret them as the tupples as the parameters as being in the relation that is it. So, if so this H lambda models every lambda i therefore it also models the big wedge the big AND of this lambda i.

(Refer Slide Time: 40:28)



One thing is clear any such any such set of ground literals there big AND cannot be logical valid. What, is logical validity logical validity? Means it should hold for all models for all interpretations. But, now how many there are uncountable number of interpretation outside. So, the only way is to look for a, model within the herbrand algebra look for some interpretation within the herbrand algebra. So, that it does not model it and then you have found a model for which you found an interpretation for which the big AND is not true and therefore big AND is not valid that is.

I can take any literal there is a model for the compliment of the literal. I can take that singleton set and there is a model for it. So, and now so that interpretation is clearly not a model of this big AND of lambda i. And, therefore big AND of lambda i cannot be valid cannot be logically valid. Big OR lambda i has a model because I can take any singleton set lambda i and just create a model for it in particular with this lambda i that i choose was a negative literal. Then, essentially I can restrict my interpretation to the empty relation. And, I have and then I have obtained a model for it. So, I need to look only inward into the term algebra in order to look for models. And, this any big OR is always is valid if and only if well if and only is the big AND of the compliments as no models if and only if the compliment. This set contains a complimentary pair if and not and this lambda bar contains a complimentary pair if and only lambda contains a complimentary pair.

So, this is the first step towards looking inwards of model and thereby restricting your quest for models of sets of formulae in some tractable fashion we just stay within your term algebra and do not go outside it. So, basically what we have to prove which I am not going to do today which I will do next time is really that it is sufficient to look inside your term algebra. It, is not necessary to go outside we will after that we will do some more model theory where we will talk about countable and uncountable models and so on so. But, essentially if it is sufficient to look inside your term algebra models that means there is a model in the outside world if and only if there is a model in your terms algebra. Supposing that is true then what it means is it gives you a very amazing consequence that your term algebra anyway has only a countable number of terms. Which means that any first order theory which has a model should be having a countable model should be having an at most countable.

So, the question of first order theories of sets of first order formulae which do not have countable models and have only uncountable models. That is the question that gets answered also by this the herbrand theorem. So, we will look at all these things and so this you can see that this notion of countable models is a direct extension of compactness. In the case of compactness it was countable set finite subsets here even for uncountable models you are saying that you do not need to look for uncountable models you look for just countable models such sufficient. Finite models are not always guaranteed but, where but if every subset is finite then you still have to have a countable number of finite subsets for a countable set. So that is so that that may not be guaranteed so we will so now what we are doing is we are progressing out of proof theory gradually into model theory. And, then we will get into first order theories like of numbers and so on. We, also have to prove completeness and we have to prove un-decidability of there are some important things. But, let us go through herbrand models and which essentially forms the bases of all logic programming actually and then we will proceed.