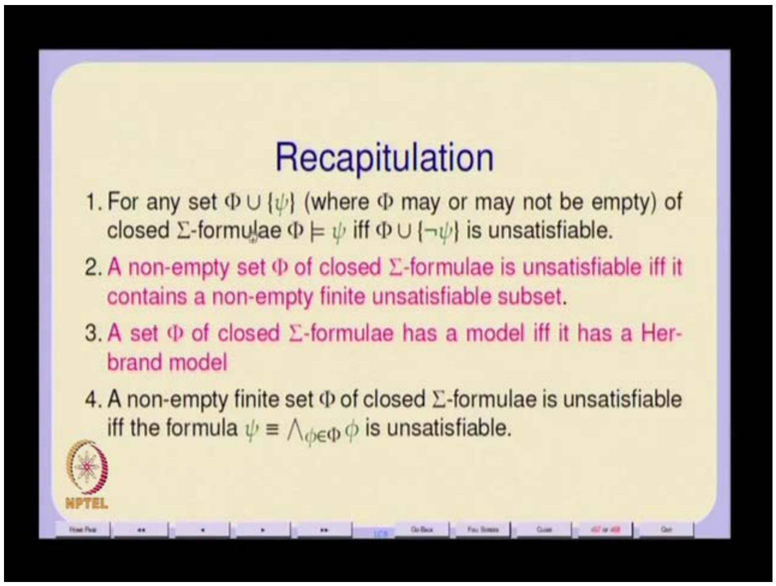
Logic for CS

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

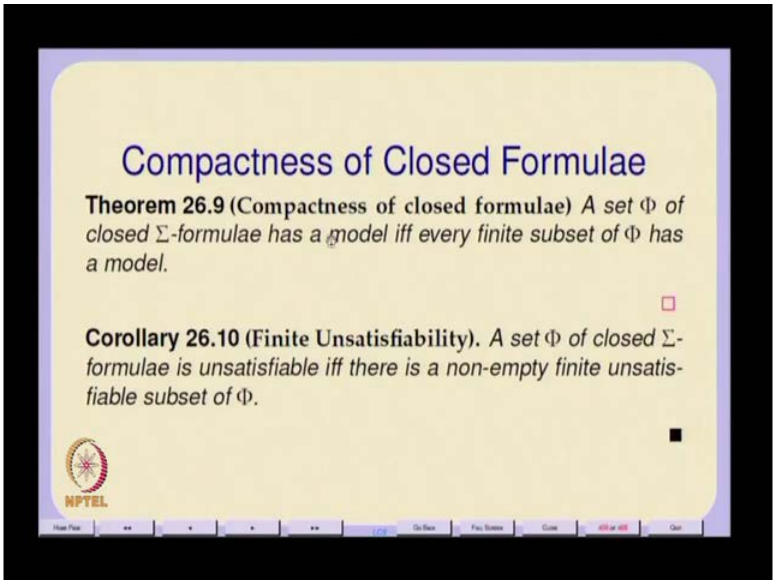
Lecture - 29
Resolution in FOL

(Refer Slide Time: 00:42)



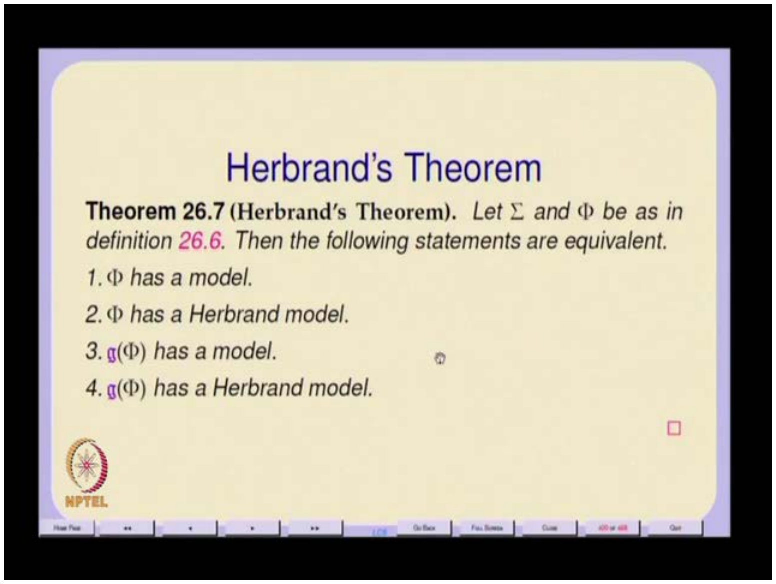
Will, start with get on to this Resolution, before we get on to the resolution. Let, us do some recap. So, one thing occurs it is easy to show that if, you take closed formulae basically all closed formulae behave like proposition that is. So, in fact so if, you take only closed formulae then the concept of logical consequence is very much like that of that in propositional logic. So, essentially of a closed formula phi logical consequence of a set of closed formulae phi if, and only if phi union naught psi is unsatisfier. So, some of the things that we have been looking at so for one thing is so if you look at all that you have done in essential in module theory and validity in terms of unsatisfiability. Then, essentially a non empty set phi of closed sigma formulae is unsatisfiable if and only if it contains a non empty finite unsatisfiable subset.

(Refer Slide Time: 01:56)



And, this impact follows straight from the Compactness theorem for Closed Formulae. So, this particular the essentially the contra positive of this if and only if gives you a essentially what I might call a finite unsatisfiability corollary. So, a set of sigma closed formulae has a model if and only if every finite of subset sigma has a module. So, it does not have a module if and only if there is at least one finite subset which, does not have a module. So, Unsatisfiability from compactness actually you get a finite unsatisfiability result for closed formulae. Then, a set phi of closed sigma formulae has a module if and only if it has a Herbrand model.

(Refer Slide Time: 02:52)



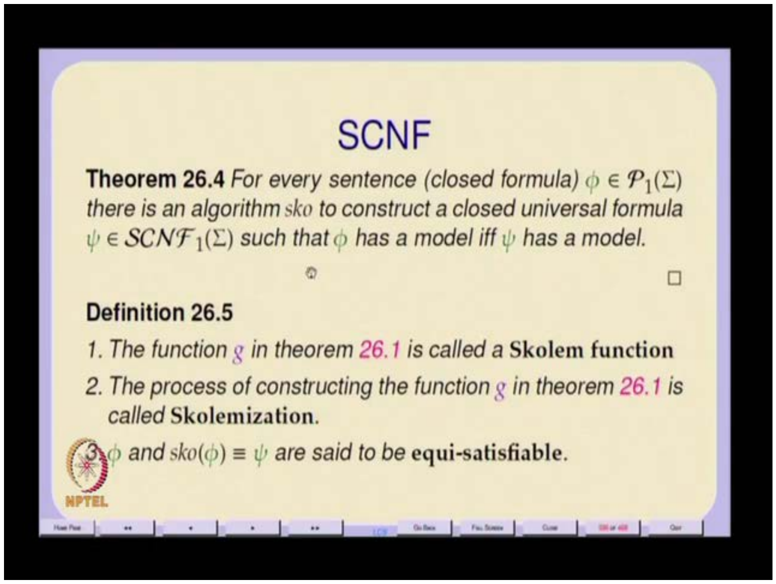
So, in fact this is Herbrand's Theorem is very powerful theorem which, actually so these four statements are equivalent to each other. And, if you look at the these g of phi and then g of phi essentially consist of number I mean there are no free variables in g of phi. So, valuation is of no effect there are no quantifiers also in g of phi. So, there are no free variables and there are no quantifiers then it means that it basically behaves like propositions. I mean so, this is actually the reason why we first learn proposition logic before we come on to predicate logic first. But, many of the technique that is we want in predicate logic have their direct analogs and proposition logics. And, that is also a tub down approach because, even if you look at predicates as parametric proposition then it is a good idea to first get a over view kind of language whose properties we understand thoroughly. And, after that reduce everything in a deeper language it is essentially a deeper languages right. You, are looking into the structure of propositions and parametric finding parameters to get parametric proposition of predicates.

So, then reduce all that your languages it is an interesting way of dealing with the problem. And, so what are Herbrand models? What Herband's theorem essentially says that therefore is that. You, take this set of a phi of close sigma formulae all we need to do is to deal with the ground terms. The ground formulae are all quantifier free and variable free and therefore, they are essentially like propositions.

So, this has an model if and only if has a Herbrand model. So, which essentially means if the ground formulae of phi have a Herbrand model and the Herbrands theorem also tells you that we need to deal with only terms in a language we do naught need to go for further than that. So, any non empty finite set phi of closed sigma formulae is unsatisfiable if and only if the big AND of this set of closed formulae is unsatisfiable. Because, there are closed formulae they essentially behave like propositions and from this it is for the first point it is clear that we should just take a big AND. But, notice that none of these things might work directly if any of the formulae in phi work open. the, more the moment to have free variables and you have what might be called open formulae then, it becomes important unless the validity of the formula is established to be independent of those free variables it matters what interpretations what models are you looking at what valuations are you looking at that is.

So, this so for example the reason the deduction theorem from proposition law logic does not directly go into first start of logic it is prescribes because of the presence of open formulae. So, there are deep constrains that you have to put on the free variables before you can get an analog of the deduction theorem predicate logic. On the other hand what all this essentially say is that if you do not have any free variables. And, if you then you can treat predicate logic essentially like propositional logic. In the case of closed formulae there is an added complication that in as proposition universally closed formula can have a infinite number of ground terms. But, that is where, the compactness theorem comes to your risk. Both the compactness theorem of propositional logic and the compactness theorem of a stratctologic essentially tell you that, for sac there might be and infinite number of propositions generated by the ground instances. But, if I am looking at it not as validity but as satisfier unsatisfier ability. Then, I have to look for only a finite subset which is not in satisfiable which, is so that is so technically that is a very important result.

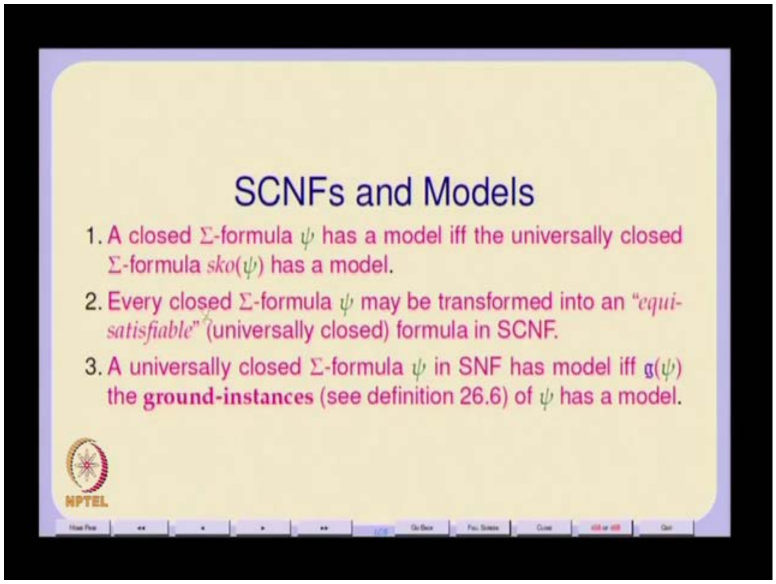
(Refer Slide Time: 08:08)



So, then what we did was we showed that Skolem normal form which, essentially says that for any closed formula phi there is a simple algorithm sko which, generates a formula psi which is not logically equivalent to phi. Unlike in the case of propositional logic where we had the conjunctive normal form here, it is only equi-satisfiable there set to be equi-satisfiable. They may not have the same models but, if one has a model then the other has a model. And, in fact actually what happens is that indubitably speaking the process of Skolemization is such that actually every model of the Skolem form would, also be a model of the non skolamized form. But, not necessarily the other way round I think there is an exercise some were where I think we showed that phi Skolem of Skolemized phi will logical consequence of fact. And, that shows that basically what we are seeing is that if phi does not have any models in the Skolemized form of phi also not going have any models.

So, model construction and unsatisfiability therefore go hand in hand. So, phi would be unsatisfiable if and only if sko phi is unsatisfiable. Every model of sko phi is a model of phi but naught necessarily the other way. So, this is what we are Skolem normal forms give you.

(Refer Slide Time: 10:15)

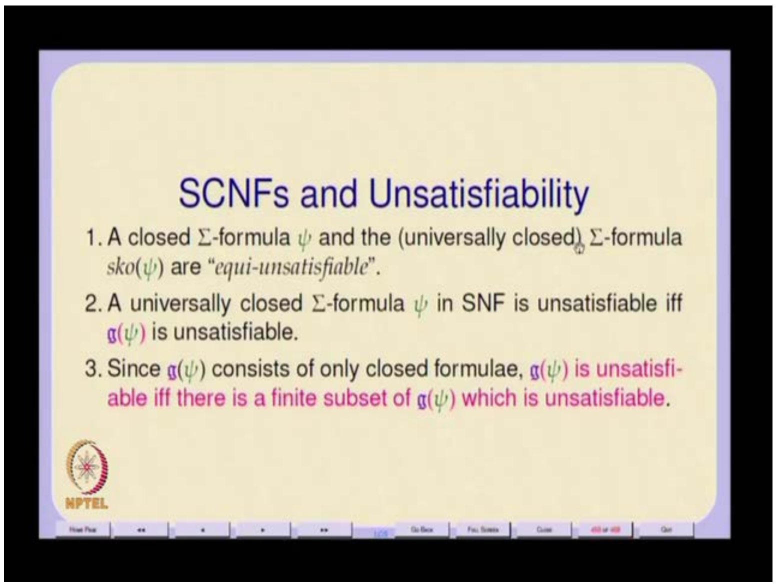


And, so essentially what we are saying therefore is so what Skolem normal theorem and the conjunctive normal form for propositions quantifier free formulae says essentially is that. I can therefore up to satisfiability every closed sigma formula may be transformed into a Skolem conjunctive normal form. So, that essentially means it is a universally closed form in a prenex normal form with a sequence of universal quantifier followed essentially by a body consisting of only proposition connectives and predicates. So, and what so this is what a this Skolem normal forms tells us. So, any universally closed sigma formula signs column normal form now not necessarily talking about conjunctive normal form it is skolem normal form has a model if an only if the ground instances of sigma psi have a model. So, this is what happens when you take a Skolem normal forms and apply Herbrand's theorem again on them let us after all your Skolem normal forms are also subset of closed formulae.

So, Herbrand's theorem applier's immediately there. So, which means you have quest for models and unsatisfier ability is essentially restricted to ground formulae. And, when you take the ground instances of psi what are you doing essentially you are thronging out quantifiers. And, we are replacing all free variables in the body by terms purely syntatic terms. And, you are essentially looking for variable free syntatic terms.

So, you are quest for models are equivalently the quest for unsatisfiability comes down to a quest for essentially ground models ground unsatisfiability which is important. Because, of the fact that which when sue are original thing if you want to prove your arguments to be valid. Then, you need to prove that the premises union the negation of the conclusion is some or unsatisfiable. So, the quest for models comes from where quest for unsatisfiablity comes from there. And, so logical validity is also somewhere related to unsatisfiability in the same way that we will use. So, this our this is a recap.

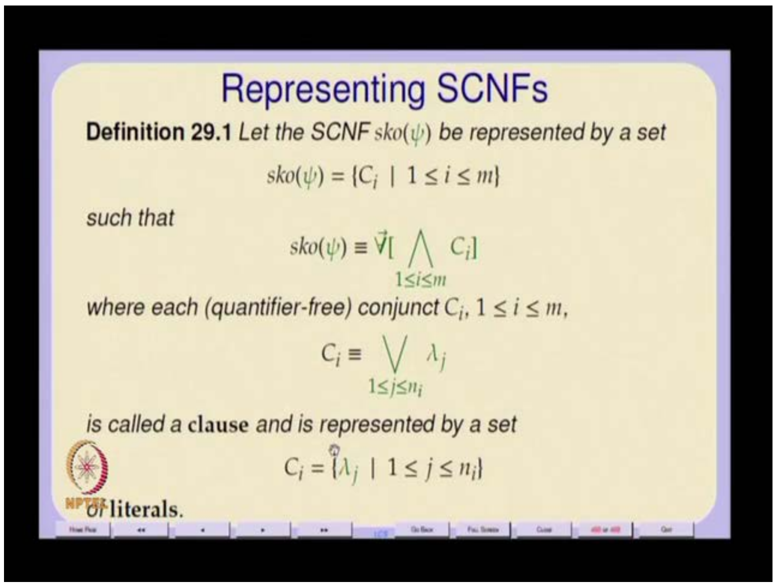
(Refer Slide Time: 13:14)



So, now let us look at this so a close sigma formula upside and the universally closed sigma formula sko phi are equi-unsatisfy if, they equi-satisfiable then there also equi-unsatisfiable. And, universally closed sigma formula phi in SNF is unsatisfiable if and only if the ground instances of a psi that means, you will throw away that the universal quantifier by essentially taking all possible substitution ground substitution of the free variables in the body of psi look at all those as essentially as propositions. If, that set is unsatisfiable then your original formula upside unsatisfiable. So, since g of psi consists of only closed formulae g of psi is unsatisfiable if and only if there is a finite subset of g of psi which is unsatisfiable.

So, everything falls in place we are interested in improving that proving valid arguments which, means there is a logical conclusion to a set of premises to all of set of axioms and that for close formula in general we are interested in first sort of theory. So, which means you are original set phi will actually be a collection of closed formulae. If, there are not closed formulae and if you are really looking at theories and if there are open and they have to be valid for all those models that we are interested in which, means you can also universally close in. So, they should be true for ground instances and essentially we are going to try to prove unsatisfiability. So, current goal now is to just go in to unsatisfiablity where, is of in order to actually show logical validity logical consequence of some statement from some other statement.

(Refer Slide Time: 15:21)



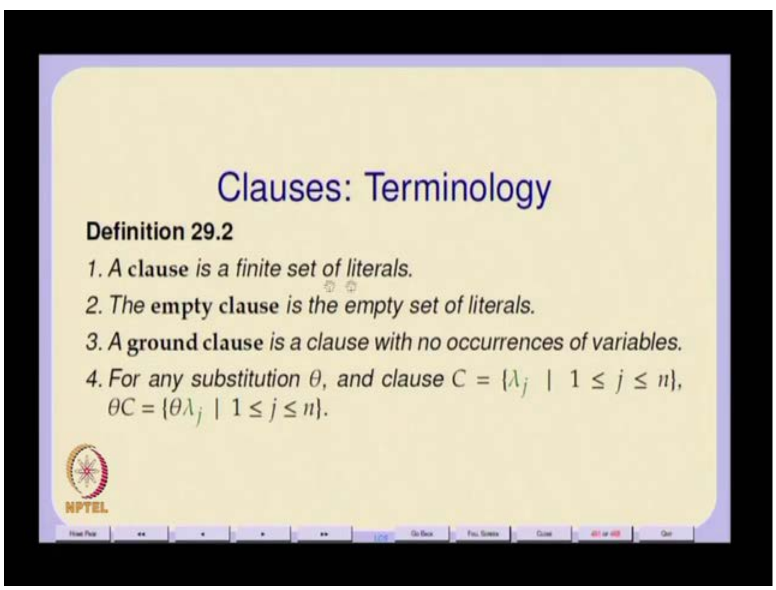
So, let us look at a Skolem Conjunctive Normal Forms.

Student: (Refer Time: 15:28)

Actually, what I meant was a you most of your first order theories the axioms are expressed open formulae under the assumptions that there are valid for all the models that you are interested in. If, there are valid for all the models you are interested in and there therefore valid for all possible valuations of the variables in the formulae. Then, there actually universal closure is also valid those models that is all that is really the implications. So, let us look at any look at Skolem conjunctive normal form. So, this essentially what is known as asset of clauses Ci. So, they might be for any formula there will be a finite set of clauses Ci. Such, that the original Skolem conjunctive normal form should be regarded as, the syntactically the same as a universal closure of the conjunction of these clauses. Where of course, each conjunct each clause consists of a set of literals basically where, the set of literals is to be regarded as so each clauses is to be regarded essentially as a disjunctions of collection of literal's lamlogic. So, essentially what we are saying is its Skolem conjunctive normal forms are as we did in the case of proposition logics they just going to be represented as, sets of sets of literals that is essentially there is an disjunction deep inside and there is a conjunction of doubt of here. So, that is what so we are all so now from now on what will do we will just look at a collection of clauses there all finite sets so these are all finite sets sub of clauses.

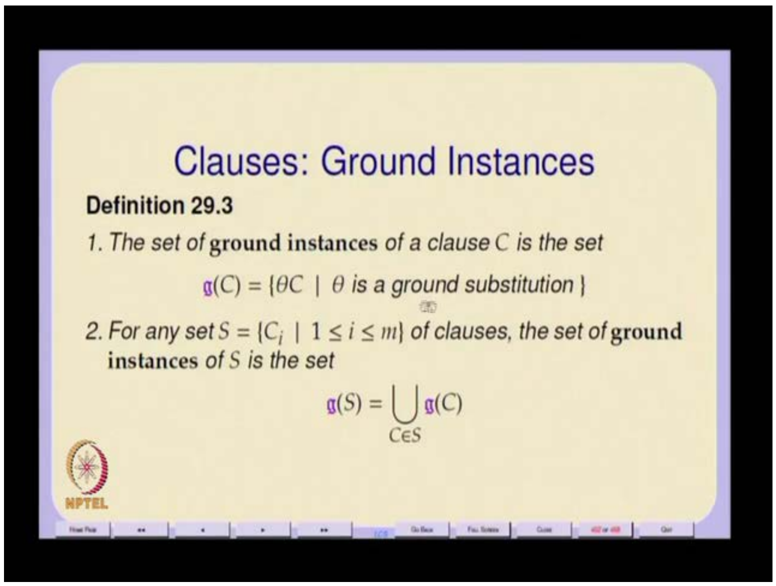
So, we just look at a set of clauses where each clause is a set of literals. Of course now, what happens is since we have so will actually talk about there being free variables in the clauses. Because, we are not we actually physically omitting the universal quntive universal closure so we are actually looking at clauses with free variables. But, there are not to be regarded as open terms because there suppose to be regarded as implicitly universally closed. So, but however will talk about a free variables because, what we going to do is you are going to deal with substitutions of this free variable.

(Refer Slide Time: 18:33)



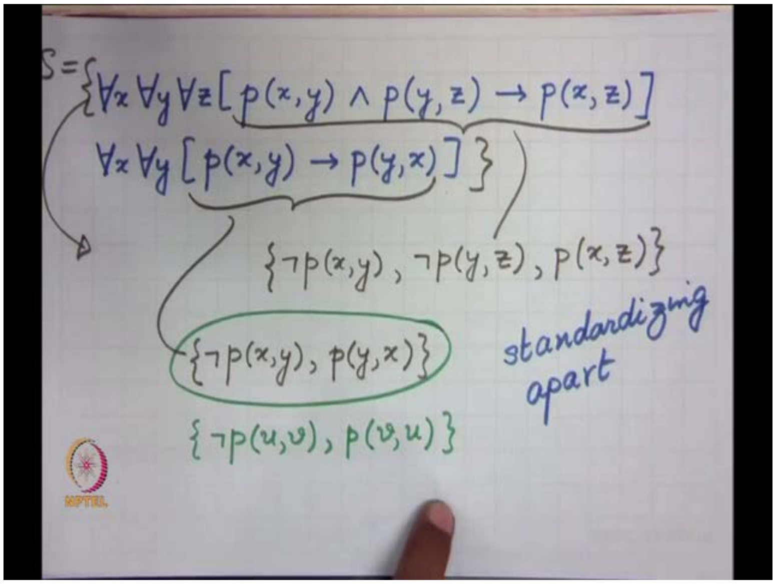
So, clauses of finite sets of literals the empty clauses is the empty set of literals. So, the empty clauses essentially represents for clause. A ground clause is a clause with no occurrences of variables. So, you can take any substitution theta and a clause C each literal might have free variables. Each literal in the clause might have free variables and you can actually apply the substitution in theta two the free variables. In particular so theta C is just a theta applied to all the literals in the clause. And, in particularly if theta is a ground substitution which means all the variables are replaced by ground terms. Then, this become what you get in is aground clause collection of ground literal basically. So, for each substitution that is what you get.

(Refer Slide Time: 19:47)



So, I will you saw if I take a class C and remember that it is essentially to be regarded as being universally closed. So, for all possible values of the variables in the clauses here essentially saying that the corresponding ground predicates is true. So, you can take the set of all ground instances of clauses and that is a essentially take all possible ground substitution for the free variable of the clause. And, just take this huge essentially take we going to essentially take this huge union. But, at the moment I am regarding it at as creating for each clause as collection of ground clauses. So, for each ground substitution theta there is a new clause theta C which, is completely variable free and this is a ground clause. If, I have a set of clauses of finite set of clauses then the set of ground instances of S is essentially big UNION of the sets of ground clauses. So, notice that all this is consistence with are interpretation that this set S is implicitly universally quantified over all the variables. The only problem sometimes is that so I have some clause let me start with the formula. Let, us say some particular relation is transitive.

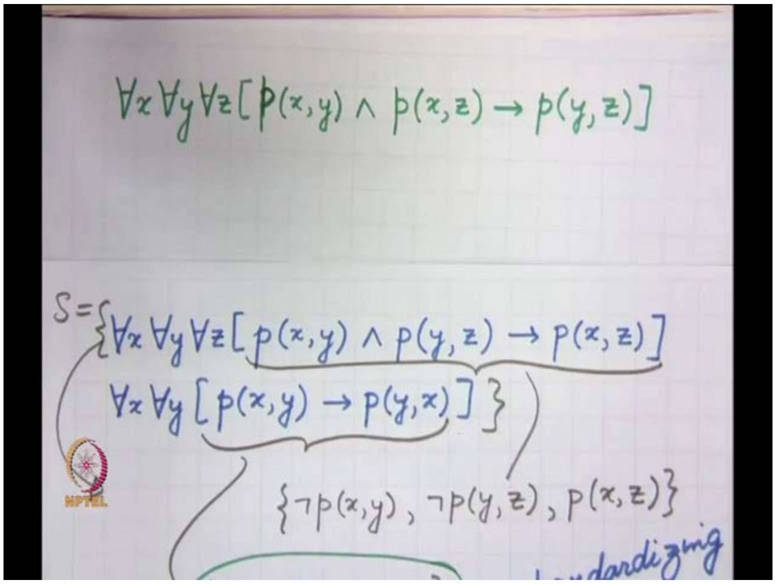
(Refer Slide Time: 21:45)



So, then what we are saying is for all x, for all y, for all z let us say this is some atomic predicate binary predicate p. p of x, y and p of y, z arrow p of x, z so this is tranduvity. And, let us say you have a symmetry. So, symmetry would also be written as for all x, for all y. p x, y arrow p of y, x now if I was going to convert these. So, I can convert them into Skolem conjunctive normal form so, this one gives me or let us get worry about this body. So, this body is essentially is naught of this. So, naught p x, y coma naught p y, z coma p x, z. So, essentially as a clause this is what it is. And, if I want to take this is just naught p x, y p y, x. So, as a clause throwing out the universal quantifiers I essentially get these two clauses. However, since we are talking about these being implicitly universally quantifier it means that this x here, and this x here are different x's. So, that is so actually what we need to do therefore is we need to do an alpha remaining. So, that different clause since we are throwing out the quantifier there is a possibilities of confusions. So, you will need to do an implicit alpha renaming you going to use implicit universal quantifier you should do an implicit alpha renaming.

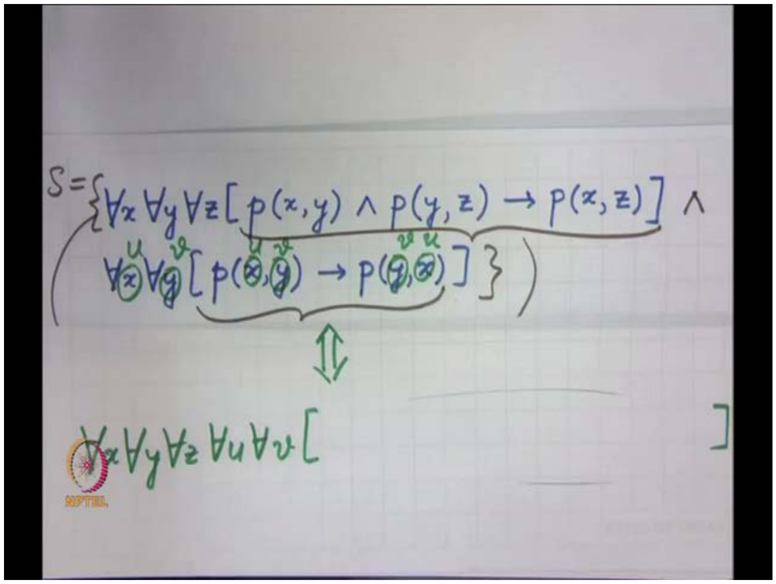
So, if you do an implicit alpha renaming essentially what we are saying is. This, will choose some two new variables which do not occur anywhere else. So, I get a essentially naught of p let us say u, v and p v, u. And, this process say if I had if might set original set S contained these two clauses here. Then, this process of separating of the variable is called standardizing apart. So, because of the fact that your universal quantifiers are implicit and they have a scope restrict to the clause you need to do alpha renaming if, you going to throw out the universal quantifiers. Remember one thing that therefore essentially what we are saying is, that this is if you are looking at logical consequences of this set S. Then, you are essentially looking at a conjunction of these two formulae so, there is a logical consequence so here is a logical consequences.

(Refer Slide Time: 26:30)



And, one logical consequences for all x, for all y do I need a third variable for all z it is also symmetric. Would this be true? So, if p is a transitive in symmetric relation then essentially this should this statement should. So, what are you saying now? We are saying that S so this statement is a logical consequence of this statement of this two statement. So, which means that I am saying that I can take the and of these two statements. And, this would be a logical consequences of the conjunction of these two statements. Of course, what I am what I said was that we were going to standardize the part which, means you going to rename the variables. What does happen now is? So, let us keep this away for the moment. When, you standardize as a part you no longer have it in Skolem conjunctive normal form. When you do the when you include this and when you no longer have a Skolem conjunctive to normal form. But, then what you have is that this and is I am going to do some here. So, this and after having done a standardization apart which means I replace this by u and this by v.

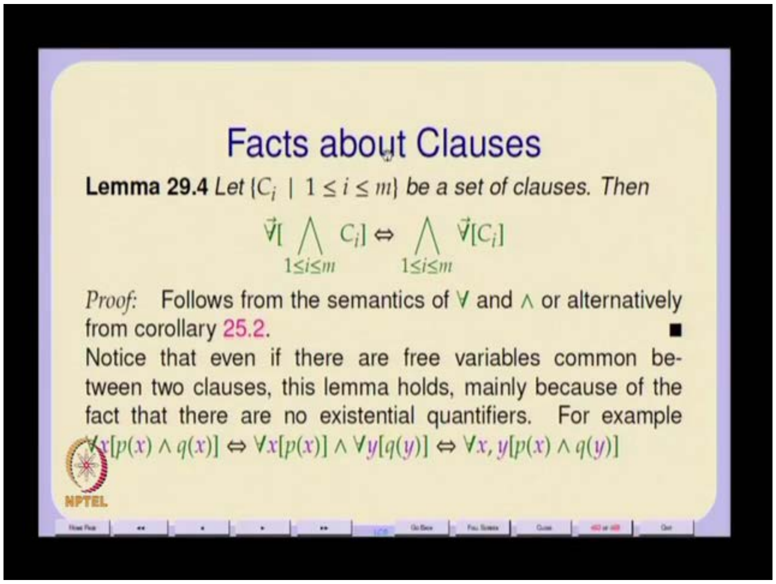
(Refer Slide Time: 29:54)



This is logically equivalent to actually having for all x, for all y, for all z, for all u, for all v and having a consumption of the points. And, this is something we I think is there an exercise to for moving quantifiers and let us so when we more quantify so essentially. Then, what it means is that I just deal with this essentially with this entire body. So, when I take a set of it is clear that these kinds of manipulations cannot be done if there are open formulae they have to be done with only if closed formulae. So, anyway now let us look at so essentially we are going to look at this. So, we have these clauses and their ground instances. And, let us look at some other clauses.

So, one of the things that we are going to do and in fact which for example all resolution theorem provers do and all prologue systems do is that. A regardless of what the user has given they generate internal temporary variables to do standardization apart. And, everything in within the prologue system actually works only with its internal variables. It is only but they do maintain a table somewhere during the scanning process which, gives a correspondence between the users defined variable names and the internally generated variables. So, that when they have to give error messages they can use user defined variable names. Otherwise, all prologue systems actually do this generation of temporary variables regardless or what the user has specified.

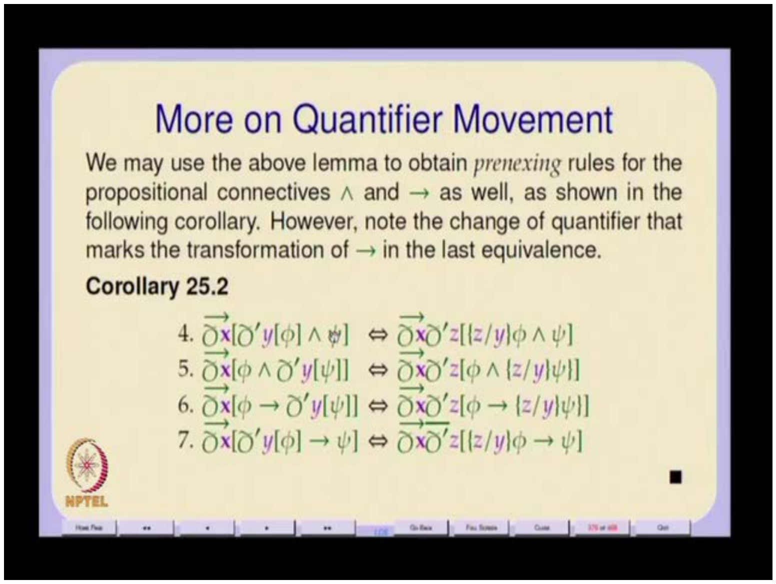
(Refer Slide Time: 32:01)



So, they do standardization apart in that form but the reason for doing that is absolutely essential. So, but we will not let us look at some more properties of clauses. So, the first thing is that I take what this has shown me here from here is that. I can blindly take universal closures whether, there are free variables or not. So, you take a collection of clauses Ci. Ci and Cj, may having been standardized apart will have different. Say we will have different sets of variables. I take the conjunction in the universal closure and that is logically equivalent to taking the clauses apart and taking the universal closure. This is because, your universal quantifier is essentially generalization an infantry generalization of your and…

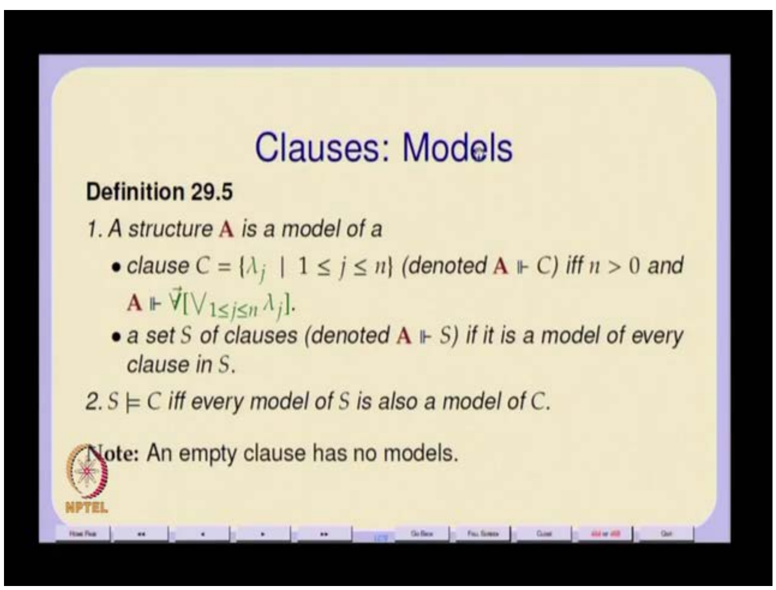
So, it can be moved around but this universal closure therefore leaves unspecified what variables actually are free in Ci. So, if you take these two clauses for example this universal closure on the left side corresponds to this the closure of five variables. Whereas, on the right side it corresponds to one clause with three variables universally closed and the other clause is two variables universally closed.

(Refer Slide Time: 33:38)



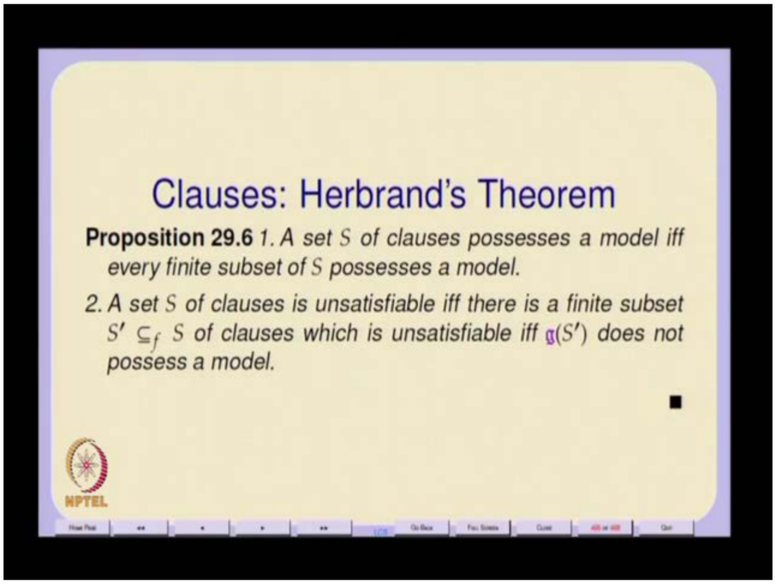
So, here is the reference this quantifier movement essentially tells us that we can do all these kinds of manipulations. So, that is what our universal closure gives us so this is one fact.

(Refer Slide Time: 33:55)



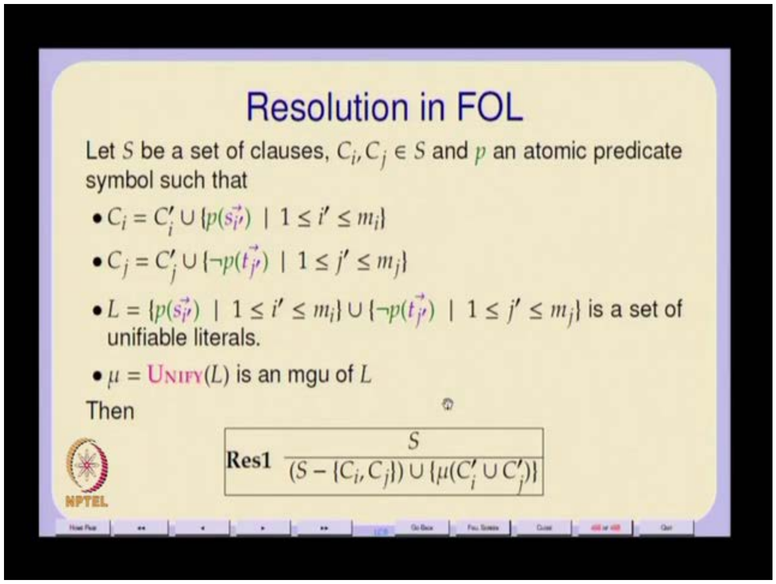
The, second is let us look at Models of Clauses. So, we just so structure A is a model of a clause C. And, we will denote this if and only if the universal closure of the disjunction of literals in the clause is a model, A is a model of that. I notice that a clause is a set of literals and we implicitly regarded as a disjunction an, or of the literals. A set of clauses has a model A if, A is a model of every clause in S because, a set of clauses is actually a conjunction of clauses you know implicitly. And, a clause C is logical consequence of a set of clauses S if and only if every model of S is also a model of C. An empty clause is no models because, an empty clause typically refers to the constant false.

(Refer Slide Time: 35:02)



So, now in the Herbrand's theorem brought down to the level of clauses essentially says that a set of clauses possesses a model if and only if every finite subset possesses a model. And, essentially this is more important a set S of clauses is unsatisfiable if and only if there is a finite subset S prime which, is unsatisfiable with if and only if this g the ground instances of S prime is not a model. And, actually we can go further there is a finite subset of ground instances which do not possess a model that is Herbrand's theorem using.

(Refer Slide Time: 35:28)



So, now which actually brings us to the Resolution Method. So, for uniformity think so let us look at it even from here operational semantics. And, programming languages you would have realized that many of the structural rules there I had side conditions. Evaluation of if than else would have two rules where, one rule that then clause is executed and the side condition is that the Boolean is true and other rule the else clause is executed. And, the side condition has the Boolean forms. So, think of all this as a side condition for this rule this, just think of all this whole thing as a side condition. So, here what are we doing now? We have a set of clauses S and we choose two clauses Ci and Cj. The way you choose two clauses Ci and Cj is such that they have some common atomic predicate p occurring in them. Such that in one of them p occurs in positive form over some terms by the way this si prime arrow is meant to signify that vector tuple of ground terms.

The, tuple of terms if p is an anary atomic predicate then you assume that this vector has a length m, there are n terms there. So, basically it has some set of terms of p and the other one has some negations of terms of p. But, of course there is no reason to believe that this si prime in any case is related to this tj prime. Firstly, there are two different clauses so the one of the first things that would happen is that you would standardize them apart so anywhere the variables are going to be different. But, that aparts so what you do is. Supposing, you have this and what you can do is you take these two so, Ci is Ci prime union some you can have more than one occurrence of the same atom with different parameters within the same clause. So, that is why I have taken this to be a set assume there are mi clauses Ci which, basically have the atom p occurring with various kinds of and tuples of terms. And, similarly in Cj may be there are mj different occurrences of the negation of p with various kinds of n tuples and terms.

I take the union of these two sets. Except that I this negation should not be there, this assume I take this union remove this negation and I see that this is unifiable. So, one of the things that we did initially was we standardized them apart. Then, now what we are saying is can they we all made look alike if, they can all be made to look alike then my unification algorithm gives me some most general unifier mu. Notice, that these si primes and pj primes they do have variables in them. So, at this stage we are not talking of a unifier that will be ground it will still have variables in it. It will have variables in it not ground so, we are not actually using that last part of Herbrands theorem. But, what we are saying now is in S I can remove these two Ci and Cj. And, take the unified version of Ci prime union Cj prime. And, essentially what I am saying is that this so this is a clause the unified version of Ci prime union Cj prime. It can have free variables your most general unifier does not necessarily make things ground. So, it leaves some it keeps lots of free variables which means that as a clause it is implicitly universally quantified on all those free variables.

And, so essentially this is the resolution rule which just says that. I start with this S and I take and then I essentially replace this Ci and Cj and S by this unified Ci prime union Cj prime. Unlike the case of propositional resolution there are various differences here. One is that in propositional resolution we had only one atom in its negation. Here, we are saying that it is possible to take a set so but, this is not just a cleanup problem. When, this is I mean in the case of propositional resolution I said that you have to clean it up so that you did not have multiply duplicate copies of atoms lying around. Because, you did not have to destroy your empty clause it will make your empty clause a non-empty clause. But, here it is not here it is deliberate it is not even that you choose all possible terms which with p in it here, all possible terms with not p in it here no. It, is not even that you choose a maximal subset which is unifiable no not even that you choose some unifiable subset that is it.

So, which means that in Ci prime there should be various occurrences of p hanging around still there could be occurrences of naught p also hanging around there. And, similarly in Cj prime there should be occurrences of p various other terms and occurrences of naught p also with various terms. It does not matter while we are saying is you take we are not even saying you take only one you take some subset which is unifier. And, the result of this application is essentially that the universal closure of mu Ci prime union Cj prime is the logical consequence of Ci and Cj.

Student: so what happens when we are naught able to find. Then it does not unifiable.

So, there are two aspects of resolution when you look at resolution theorem proving. You, are looking at logical consequences when, you look at, resolution refutation you are looking at, unsatisfiability. So, all your prologue systems work on a form of refutation resolution refutation. But, I will come to that later basic resolution refutation essentially means that I should be eventually able to by applying these resolution rules several times I should eventually be able to derive the empty clause as one of the members of the set. And, if the empty clause occurs in the set. So, take this denominator to be the set as prime if the empty clause is member of this set then what you are saying is that you are refuted.

So, forgot what you going to take is so you can either try to prove for a given set phi for a given set of formulae phi. You, can either prove size or logical consequence of phi or you can include naught psi there and prove our refutation both of them are possible. In, the second case what you have to do is you have to explicitly derive empty clause. In, the first case if you can actually derive psi from s that is just as good in fact that is what we going to do now.

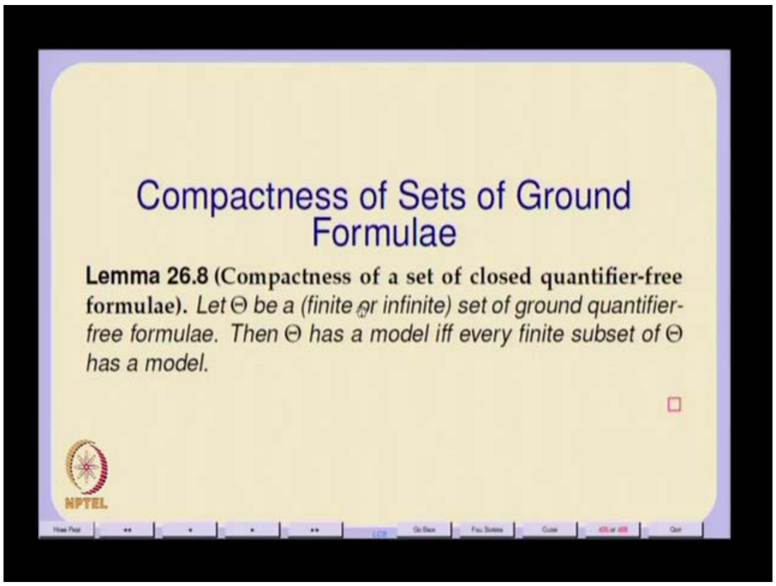
So, let us take this example so the interesting things about resolution is that this is like a new single proof rule. And, you do not need any other axioms of proof rules. So, what I am saying is throughout the whole of Hilbert slide and proof systems throw them out completely take predicate logic. And, have just one rule with these complicated side conditions and what we will be able to show later, is that this one rule is sound and complete so predicate logic.

So, the resolution so the entire you can have so this is called the resolution calculus. The, resolution calculus just defines the language and has this one rule finished that is it. And, with this rule hopefully you can you do not require any other axioms nothing. You, just have to apply this rule repeatedly it might require some intelligence to apply this rule in order to get logical consequences. So, logical getting logical consequences by just this rule is not deterministic it will require some thought but, refutation procedure can be deterministic. So, if you logical if you want calculate logical consequences by taking the negation of the conclusion. And, then deriving a refutation that can be done deterministic. And, determinism actually stems from the mgu what does the mgu give you it says that I only do as much I do not know rush into doing a ground substitution.

I make them look alike with the minimum number of substitution changes which, will make them look alike and I still retain the variables. Because, what I am going to do essentially is after this when I got S prime I am going to look for another resolution mechanism another application of resolution. That, application of resolution might again involve only this the same atomic predicate p. But, what at each time so the I basically composing by substitutions gradually without making commitments and doing it sort of opportunistically. I am waiting till the moment in order to do a certain minimal substitution to make things look alike derive a new clause. And, then apply again and so on and so forth till I get may be an empty clause from the form itself.

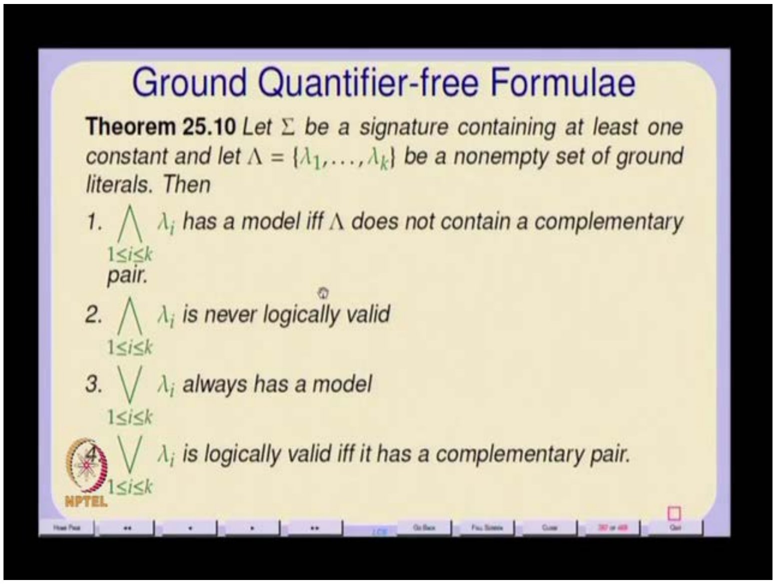
So, I may naught get your empty clause directly from directly from ground instances. So, there was that theorem we proved before Herbrand's theorem that if I had a set of ground instances in the form of a clause,

(Refer Slide Time: 49:49)



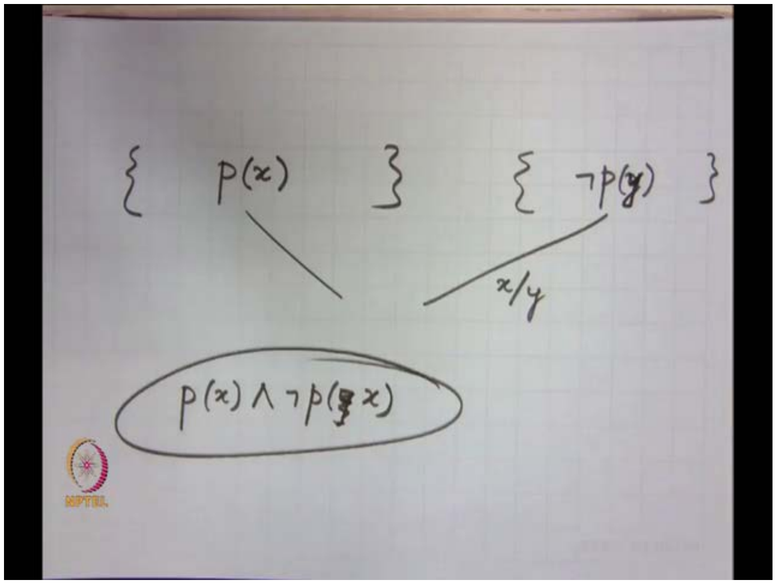
There, is a sets of ground formulae so take look at this lemma this is a set of ground quantifier free formulae. But, the difference is that by taking mgu we are not rushing into ground formulae because what this s allows us to do for example.

(Refer Slide Time: 50:45)



This, ground quantifier free formula we had this theorem which showed us that you can just do pattern matching. So, this or of lambda is like a clause and checking complementary pairs is just a pattern matching looking for identical tree with a negation on one side. So, instead of going down to ground formulae using mgu's. We, are staying within the level of clauses with free variables in them but, if the patterns still look like complimentary pairs. Then, anyway it is not necessary to go down to the ground.

(Refer Slide Time: 51:52)

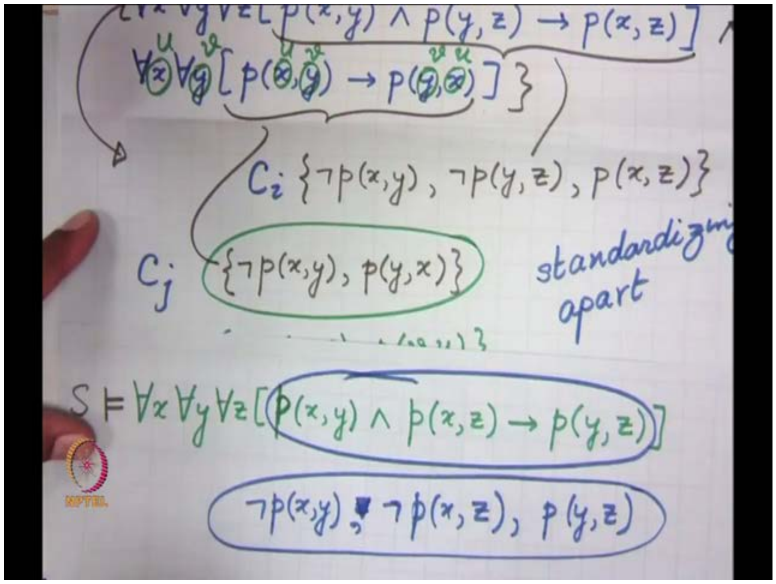


So, what I mean is if I have even if I have variables and after unification I get 1 clause which has the same p of x. And, another clause which has naught p of x naught p of I have of course after if I can actually decide to resolve between them. And, do a common substitution of a variable minimum absolutely minimum. So, maybe I decide to replace y by x then I get a complimentary pair. But, this complimentary pair is naught ground at the level of all I am saying is that it does naught matter this is always a complimentary pair whatever substitution you make for it.

So, our mgu allows us to postpone that decision till an appropriate moment. So, we are not using that so we are just using mgu's. Because, of we are just using mgu's so that we look for an appropriate place when we can decide there is something as a complimentary pair. But, it is a complimentary pair not necessarily of ground instances it is not our ground instances. Then, there is naught propositions in their ground instances they are essentially propositions. But, these are like propositional forms with variables this is like a typical contradiction in propositional logic. But, with variables and if this is always going to be a, contradiction.

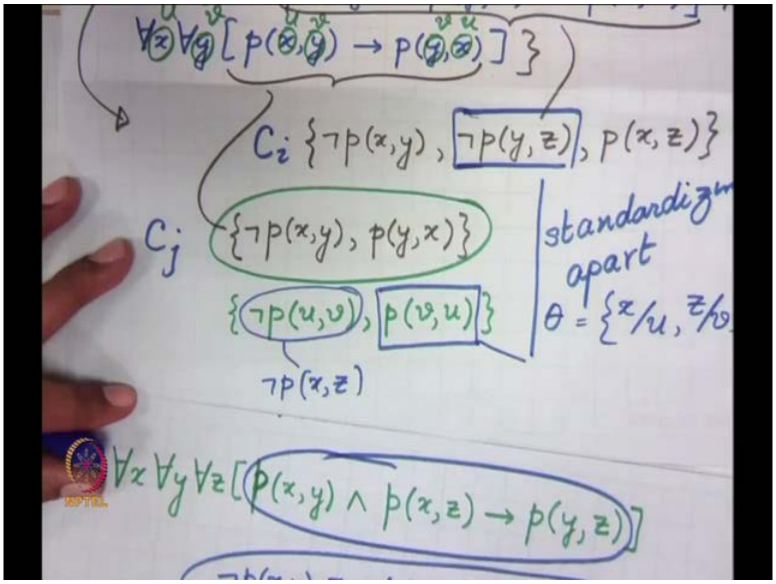
So, just look at this so what we do so we can actually prove so having standardized apart what you can actually do is I want to prove this xy and xz p y,z. And, actually I have to choose an appropriate.

(Refer Slide Time: 54:27)



So, this one by the way what is the form of this one would work out to naught p x,y or naught p x, z p y,z. So, what I am going to try to do is so these are my clauses Ci and Cj I have to choose some subset here and some subset here which, are complimentary unify them find the appropriate substitution do that substitution throughout and essentially derive this. So, the one thing is clear so there are two negations here one positive occurrence here is one negation one positive occurrence. So, either so I have to take either this, positive occurrence and this negation. So, that I get these two I get these two negations in this. So, you have to find one substitution and there is a substitution which at the moment I am naught able to clearly see. But, I think what you can do is or either way actually suppose, you may take this.

(Refer Slide Time: 56:17)



And, we take this and we unify then I get this naught p x,y which I get this I get naught p u,v. But, then I am going to unify it with this then this u,v will get substituted by x,z. Such that I let us put it this way I find a substitution theta. So, that this thing becomes naught p x,z so which means I am going to replace u by x and v by z probably this might work is it. Now, it may not work it may work with this. So, y,z so there is one particular substitution you can find which at the moment I am not able to see that I will do and show. But, basically what you can do is you can find a, substitution apply it and then you will get this. And, then that is a like a logical consequence.