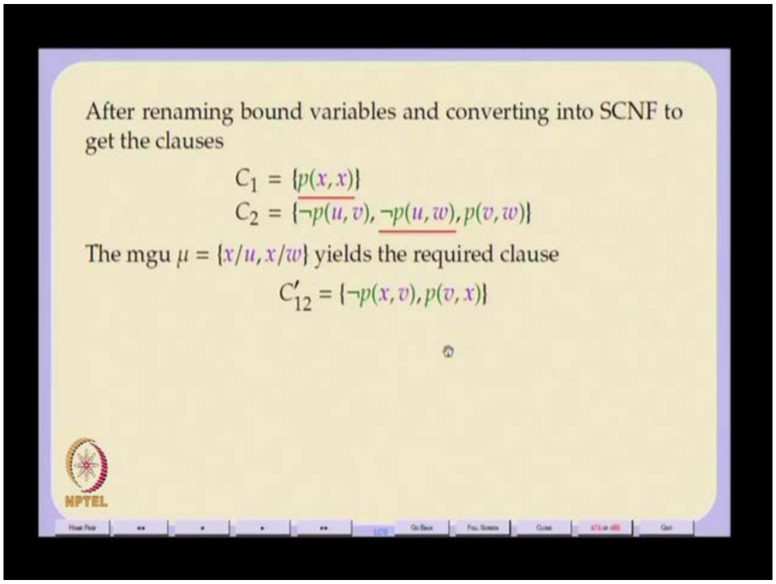
Logic for CS
Prof. Dr. S. Arun Kumar
Department of Computer Science
Indian Institute of Technology, Delhi

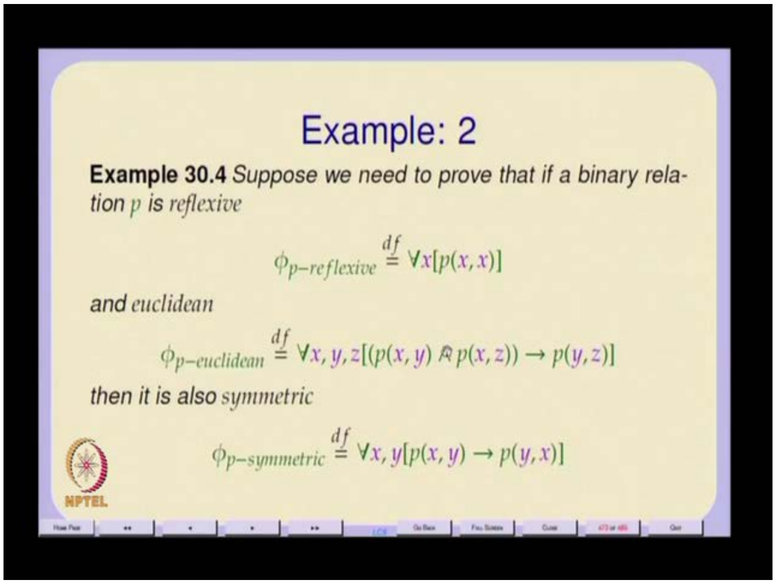
Lecture - 31
Resolution: Soundness and Completeness

So, today I will we will look at the Soundness and Completeness of the resolution method. But, before that let us just you recap what we I have done earlier so one of the things we did was that so this example was the best. So, we showed that you could actually derive logical consequences from resolution and that is what.

(Refer Slide Time: 00:52)

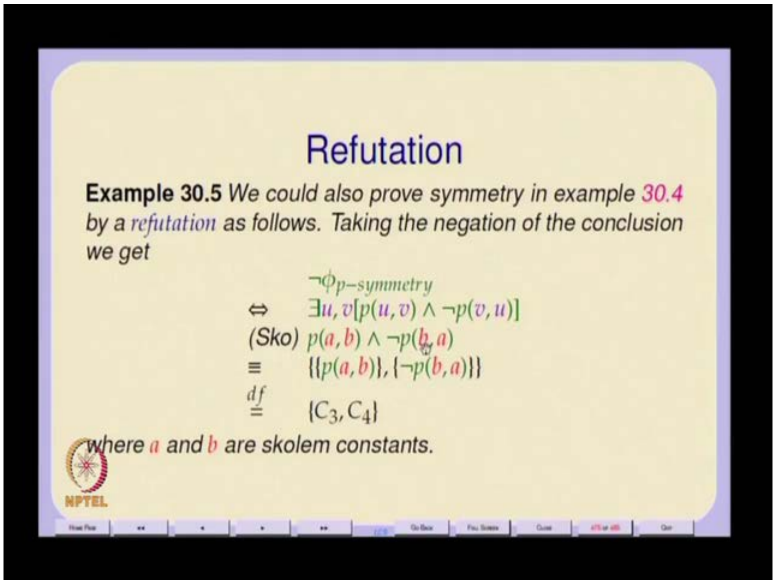


(Refer Slide Time: 00:57)



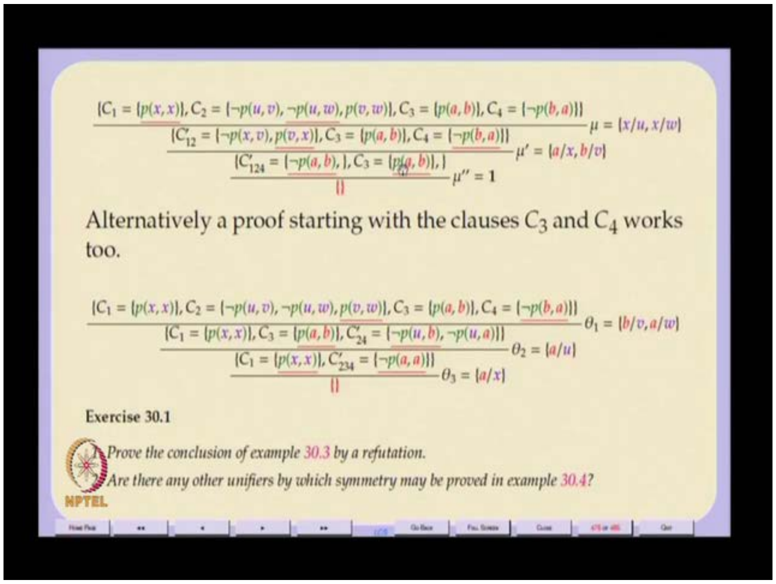
So, this was like a direct proof of this property of symmetry from reflexivity and euclideaness. So, this was so this was a direct proof which is which essentially the idea is that the resolvent should be a logical consequence of the two parent formulae from which it is derived.

(Refer Slide Time: 01:29)



The second thing we did was Refutation. So, what you could actually do is you could prove the symmetry by assuming the negation of symmetry.

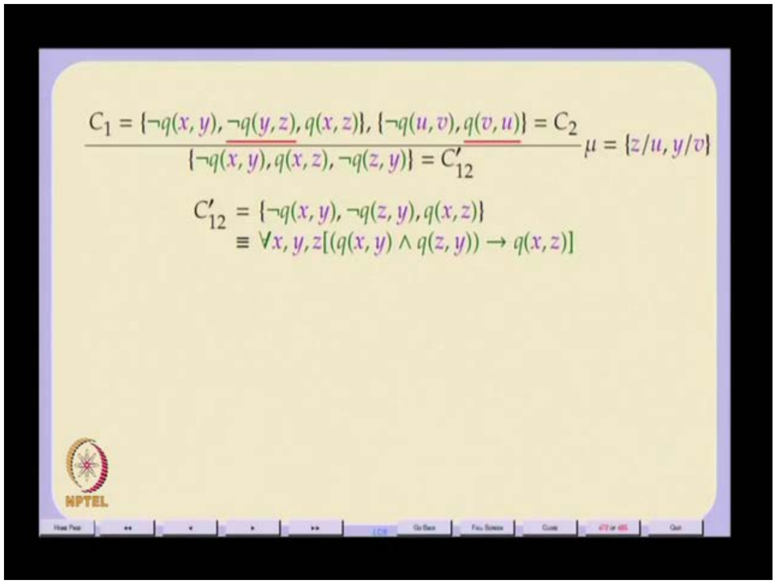
(Refer Slide Time: 01:40)



And, doing the clauses and if you start with and then you could have several possible ways of just using resolution to derive an empty clause so and this was called as refutation. In general of course, one of the early properties we saw about proof theory if you look at resolution as a proof rule. You have to realize that the derivation of the empty clause I mean in all the examples that we have done we ended up with just the empty clause. But, in general it possible that you can you have a whole lot of other clauses too in the set. So, one of our early theorems about formal theories said that if you could derive something from gamma from a set gamma and delta was a superset of gamma. Then, you can derive the same thing also from that so in the case of resolution what that means is that. If, you start with some larger set t of clauses then beside the empty clause you will also have some of those clauses hanging around. But, the fact that you have a set of clauses in which one of the clauses is an empty clause implies that you have essentially derived a contradiction. Because, the set of clauses intuitively represent a big consumption of a formulae and the empty clause represents the formula false basically.

So, you have to be able to make the empty clause a member of the set of clauses that you derived then that is sufficient as soon as you can do that you have essentially done a refutation. A refutation means that you are essentially trying to refute the claim refute some claim. So, that is why you take this negation and that is it is equivalent to a proof by contradiction. So, both logical consequences and refutations can be done through resolutions. So, that is in fact what often happens is that resolution in general tends to be more deterministic if you use it as a refutation than as a logical consequences. Because, it has the same problems of trying to guide the proof towards the logical consequence that you want which is something.

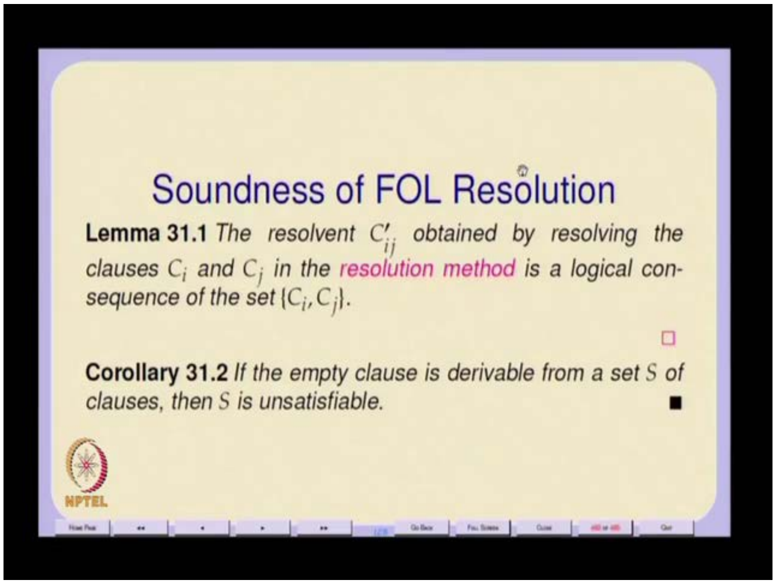
(Refer Slide Time: 04:43)



We did here somewhere I mean where so from transitivity and symmetry you have to do you have to guide the proof in such a way that you take some appropriate most general unifier which unifies exactly the kinds of things that you want in order to get this conclusion. Whereas, if you did a refutation and proof is more deterministic you can just take some whatever you whatever most general unifier you can find. And, just do a blind derivation then continue to find a most general unifier just till you somehow get the empty clause so that is in some sense more deterministic process. And therefore, more useful sometimes for theorem proving. So, now what will do is will do more meta theoretic aspects. So I will try to do to the soundness and part of the completeness the rest of the completeness I will just put up this as slides. So, the most important theorem on completeness I will put up on this as slides. And, so that then the next which you can study for yourself but after the vacation basically when we will worry about the completeness of the Hilbert system which is still pending and will worry about undecidability and may be formal theories like number theories or axiomatic set theory.

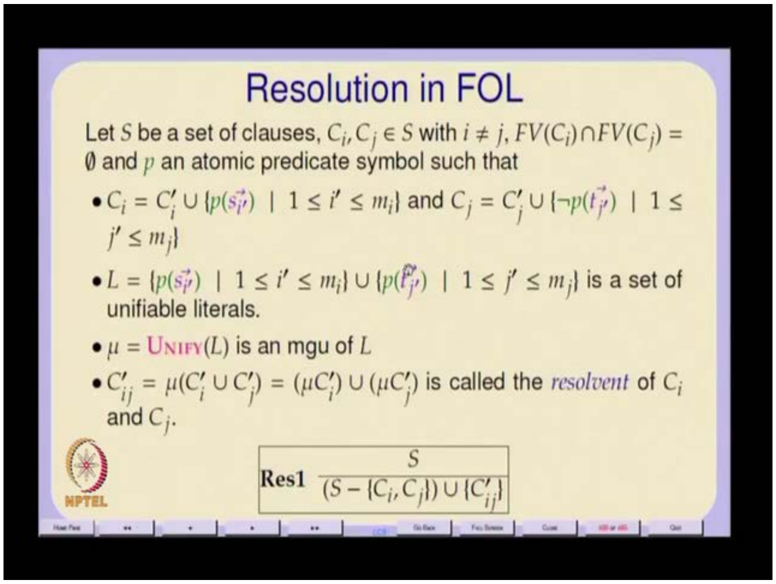
You know we have to do some applications.

(Refer Slide Time: 06:25)



So, we shall start with Soundness.

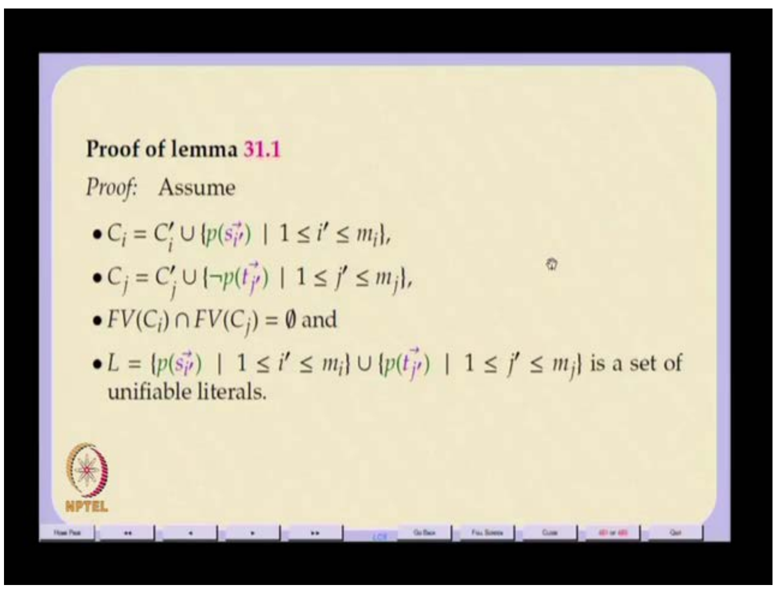
(Refer Slide Time: 06:30)



So, we have this resolution method which is with all these conditions. So, whatever those conditions are assuming those conditions hold you have you derive sets of clauses after using unification algorithm which gives you a most general unifier that most general unifier is important for the purpose of completeness but, at the moment we are just dealing with soundness. So let us all that we are saying is that this Cij prime that was defined. So, this Cij prime which is defined here is called a resolvent. This resolvent is a logical consequence of Ci and Cj where Ci and Cj are the two clauses against which the resolution was done.

So, this is what we will said out to prove today one corollary of this is that actually there is something you can state a theorem essentially. For which this Lemma is important but, essentially we could have rewritten this lemma as a full theorem. We says that the S prime that we derive from S is a logical consequence. S prime is a logical consequence of S. Where, S is the set of clauses in which Ci and Cj are some member clauses. And Cij S prime is S minus is so this this denominator is S prime. So, basically what we are saying is that resolution preserves logical consequence preserves validity. So if, S is valid then S prime is also valid so in fact it might be a good idea to treat this lemma as essentially saying that S prime is a logical consequence of S though we will focus on only a single step of resolution. So, that it most of that it is clearer.

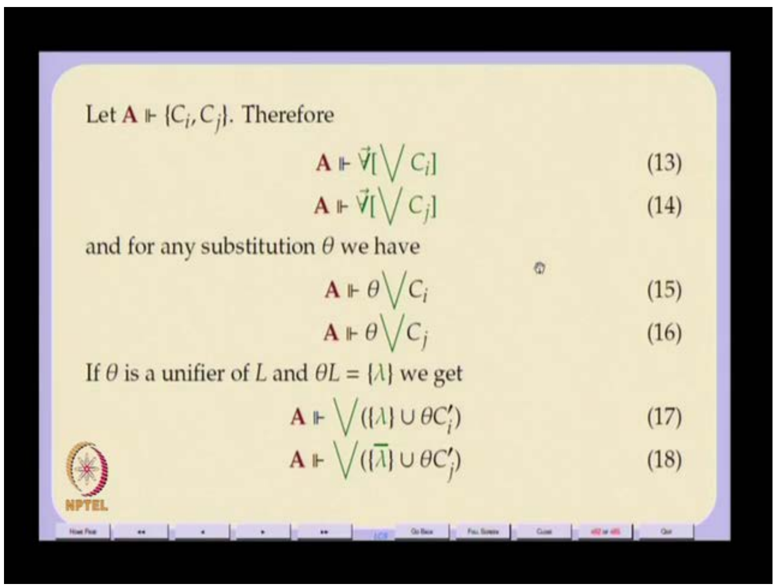
(Refer Slide Time: 08:59)



So, we just assume that we have these things as given in that resolution rule we have some we have Ci some Ci prime union some p of si prime. Where, there are various different set vectors of terms si prime and then there is Cj contains Cj prime union naught p of tj prime. Where, there are various kinds of tj primes vectors of triples all depending on the arity of p from the examples it is clear that. Ci prime could have other occurrences of p it could also have occurrences of naught p with other terms. Similarly, Cj prime could have occurrences of p and could have occurrences of naught p with other terms.

All we are saying is that there is some unifiable set of subset of terms in Ci and Cj such that the terms you select from Cj prime are complimentary to the terms you select from Ci. And, of course there free variables of course they are all we standardize the variables apart. So, all the free variables are distinctly named renamed so there are no so the free variables of Ci and Cj are empty. And, this set p si prime and p tj prime this chosen's set is a set of unifiable literals. Which, the unification algorithm gives us some for which the unification algorithm gives us some substitution.

(Refer Slide Time: 10:50)

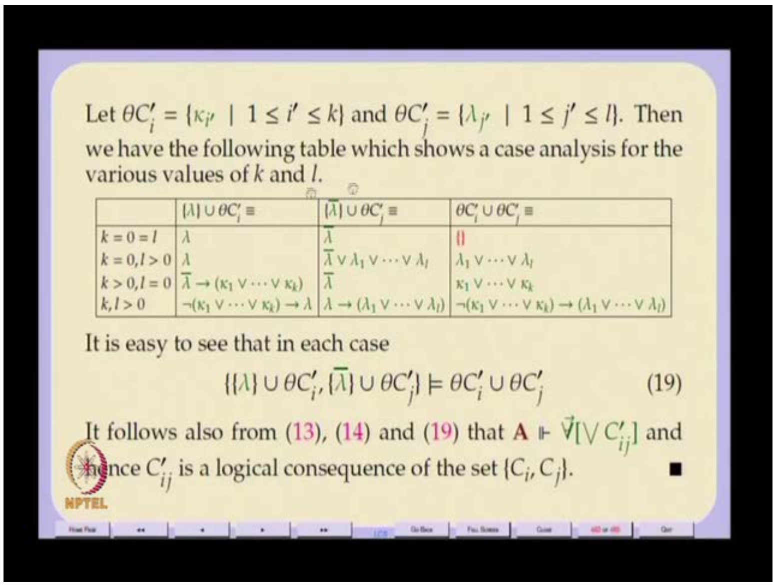


So, one thing is so supposing so will do this proof of logical consequence we need to show that Ci C prime ij is logical consequence of Ci and Cj. So, going strait by definition let us just assume that there is some sigma algebra A which is a model of Ci and Cj. Effectively that means, that it is a model of each of them the universal closure this should be this should this is an or. Or of all the literals in Ci and this is a or the or of all the literals in Cj. So, it is individually if it is a model of this then it is in model of the conjunction universally closed. But, of course they have distinct free variables so the universal closure can be distributed redundant universal universally closed variables can be removed from the individual clauses. And therefore, the universal closure of this and the universal closure of this are both satisfied by it and since this is a universal closure any substitution that you have is also satisfied by this way.

So, A is a model of any substitution theta of or of Ci and theta of or of Cj if further it has unifier of L. Then, essentially what we are saying is that this set L are unification algorithm keeps collapsing the set. I mean set of all distinct terms after unification till there is exactly one term left. So, if theta L is a unifier then let us say it gives you a single term lambda then when you do this substitution theta essentially a lot of those terms get collapsed. So, what you actually have therefore moreover this theta goes through goes across to Ci and here goes across to Cj and the result of unification is that all those p si primes which were chosen become a single term lambda. All those naught p tj primes that were chosen become a single term lambda bar.

So, that is what happens here so I can push the substitution in and essentially what I get is that A is a model of this disjunction. Where, all those p si primes have all gotten collapsed because of the substitution theta into a single term lambda. And, then the substitution is applied on the rest of the clause Ci similarly here, all those terms naught p tj primes have all gotten collapsed into a lambda bar and theta is applied to rest of the terms Cj prime.

(Refer Slide Time: 14:29)



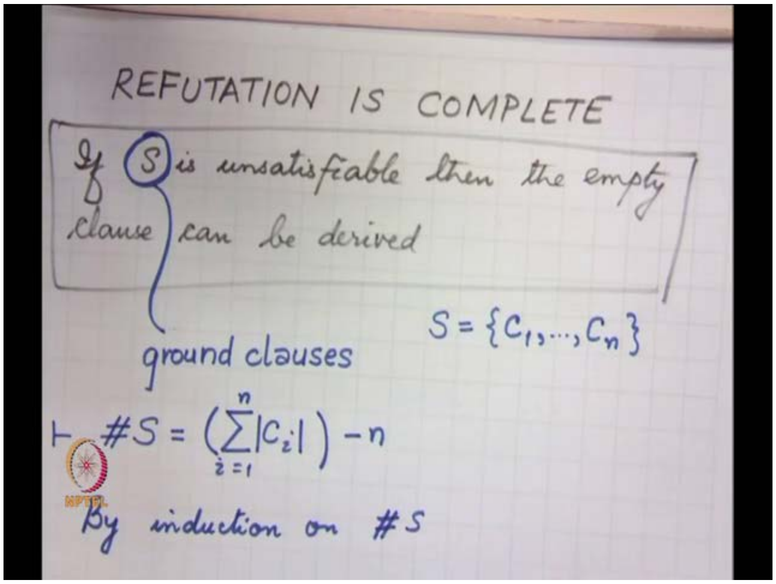
Now, what we are essentially going to show is so you take this theta Ci prime and theta Cj prime. So, basically let us assume that there are some literals a some set of literals capa i prime and lambda j prime. So, there are k literals in theta Ci prime and l literals and theta Cj prime. And what do they represent? So, depending on the values of k and l we actually have this kind of a case analysis. Now, I had to contact the table so that it fix in the screen but I think you will be able to see most of what is there so when you have a when k and l are above 0. Then, what it means is that your theta Ci prime is empty and theta Cj prime is also empty. Which means, what you have here is just lambda and lambda bar which are a complementary pair and they derive the empty clause. So, this union which is what you are this is your C prime ij though of course notice that I have not actually taken theta to be the most general unifier this lemma holds for any unifier it does not matter.

So, essentially this is like union is your C prime ij the resolvent and you can see that this empty clause is actually derivable and is not just derivable from these two. But, is actually a logical consequence of lambda, lambda bar it is like an and of lambda, lambda bar. And, in the other cases so if you were to take the case when k is 0 which means that theta Ci prime is empty. And therefore, you have only lambda there whereas l is non 0 then what will you have is. So, you essentially you have lambda bar or that entire clause represents an or of literals or of all these. And lambda and lambda bar or all this essentially is logically implies lambda 1 or 2 lambda l. And this is so essentially these are validity preserving a logical consequences. I mean, so validity is preserved almost exactly like so you mean we have to think of it this way they look propositional. But, we know that all propositional validity is also work for universally quantified first-order formulae. So, they are you have to so these are these are actually universally quantified on their free variables. But, validity is preserved over that only because of the universal quantifier. So, these all these propositional logical consequences also work for universal closures of first-order logic formulae and they preserve this validity. So, in this particular in the case of when k is non 0 and l is 0. I still have lambda or capa 1 or dot or capa k but that or is equivalent to lambda bar arrow this and here I have lambda bar.

So, by modus ponens I actually have capa 1 or dot capa k and when both k and l are non 0. I prefer to write it this way so again I have lambda or capa 1 or capa k. But, that is logically equivalent to taking the negation of capa 1 or dot capa k arrow lambda. And, this is lambda bar or lambda 1 or lambda l. But, then that is the same as lambda arrow this and then we have the transitivity property of the arrow. So, which allows us to essentially claim this so this is logically valid and it remains logically valid even when it is universally closed. And, therefore it preserves logical validity. So, in each case what we actually have is that this set lambda union theta Ci prime lambda bar union theta Cj prime it has theta Ci prime union theta Cj prime as a logical consequence. And, a universal closure does not change a logical consequence and therefore what it means is that your A, a model A that you started off as a model of Ci and Cj is a model of C prime ij. And, therefore it is a logical consequence because A was an arbitrarily chosen model so for all such models which for all models of Ci and Cj you have shown that there also models of C prime ij. And, therefore it is C prime ij is a logical consequence of Ci and Cj so what this essentially says is so now in particular if you derived an empty clause.

Then, what you are saying is if you derived an empty clause from Ci and Cj then you are essentially saying that first-order logical consequence of Ci and Cj. And, so that the universe holds the false set has of course no models therefore the original parent would also not have any models. Which, is what we have in this corollary if the empty clause is derivable from a set S of clauses then S is unsatisfiable. And, this lemma can of course be written more generally as if S prime is obtained from S by a single application of resolution. Then S prime is a logical consequence of S and therefore repeated applications preserved logical consequence remember that. Now, we have moved away from logical equivalence in two ways. One in if you look at arguments in general the fact that we convert arguments into sets of using a skolem conjunctive normal form means that we have moved away from logically equivalence. And, remained only in equi satisfiability of models. And, your resolution does not preserve logically equivalence it preserves logical consequence. So, what it means is that you cannot apply even if it were feasible in other ways you cannot actually go through the derivation backwards but that is true of any proof system actually. It preserves only logical consequence there is a way of looking at proof system is that if what a theorem gives you only a logical consequence of your axioms. Then, there is a way of looking at it as information theoretically a theorem has no value it does not provide any fresh information. So, that is where that is a different philosophy let us not get into that. So, we are looking at so essentially therefore resolution also preserves logical validity. So, if your original set was logically valid then, any resolvent will also be logically valid. And, definitely it preserves logical consequence and you take the contra positive and that essentially justifies refutation very much like proofs of contradiction. So, if the empty clause is derivable from a set S. Then, that derived S prime does not have a model and therefore S does not have a model therefore s must be unsatisfiable. So, I do not actually have any more slides but will go through a proof on this. So what i will do now is I will justify essentially a completeness. So, if you look at this. This says so look at this corollary so what will do is.

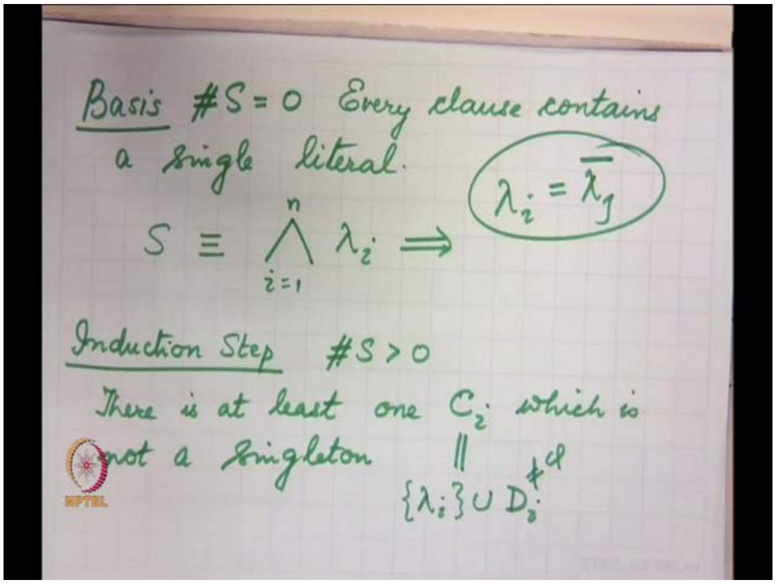
(Refer Slide Time: 24:35)



So, will just say that refutation itself is complete and, what I mean by that is essentially this is this corollary is like the soundness for refutation. If, I can derive an empty clause using resolution from a set S of clauses then, their original set S is unsatisfiable. And, what I need to do is to show that if S is unsatisfiable then the empty clause can be derived. So, this is what is going to be a completeness theorem. However, at the moment what I am going to do is restrict this S to only ground clauses. So, this is like a preliminary lemma for ground for and it some sense also justifies resolution the resolution method for propositional logic. Because, it is trivially follows that propositional resolution is complete for propositional logic. So, but let us go through this so the interesting thing here is. So, I have to show this assuming that S is set of ground clauses if it is unsatisfiable then the empty clause can be derived and we need to show that.

So, let us proceed with this proof as follows. What I am going to do is I am going to do this proof using some measure. So, I define hash S as so let S be the set of clauses C1 to Cn. So, there are n clauses and hash S consist a hash S is a number and this number is sigma Ci i equals 1 to n. I take the number of literals in each Ci so basically I take the cardinality of the set Ci. And, I add up all of them and then from this I also subtract n this is my measure. I take this measure and the proof then is by induction on hashes. This is the measure has been chosen in such a way that essentially you can start in induction by 0 from 0. So, if you were to start your induction.

(Refer Slide Time: 28:48)

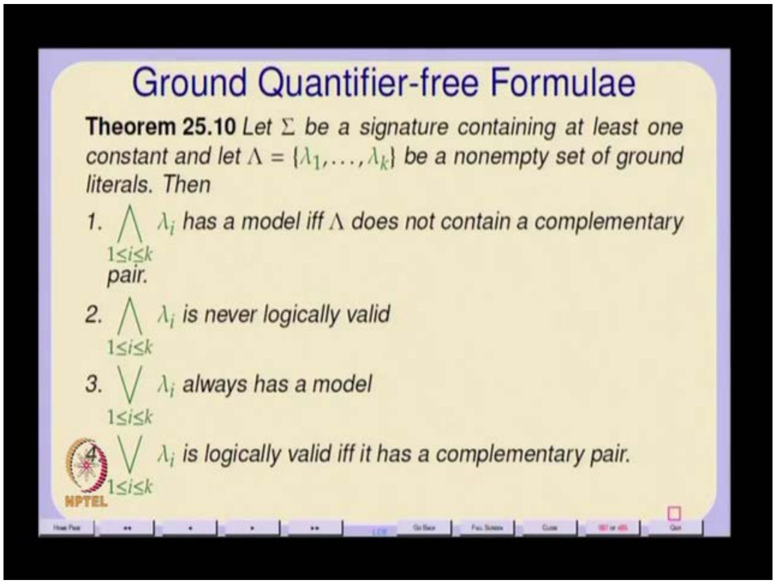


So, if hashes so you take your Basis and if hash S is 0. Then, what this means is that this summation is equal to n. Now, this summation can be equal to n if and only if every clauses is singleton clause. So, this summation cannot be equal to n on so which means that the case when hash S is 0. Means that every clause is a singleton every clause contains a single literal. If, every clause contains a single literal so essentially what we have is that. Your S represents the logical formula and of lambda i. Where, lambda i is the singleton is a single literal representing clause Ci from i equals 1 to n. So, if S is this and you start with the assumption that S is unsatisfiable. S has no models then, we had a theorem somewhere which showed that there should be a complementary pair. Where is that theorem was it an normal forms skolemization I think Herbrand's theorem Herbrand tree of interpretations.

Student: Before this

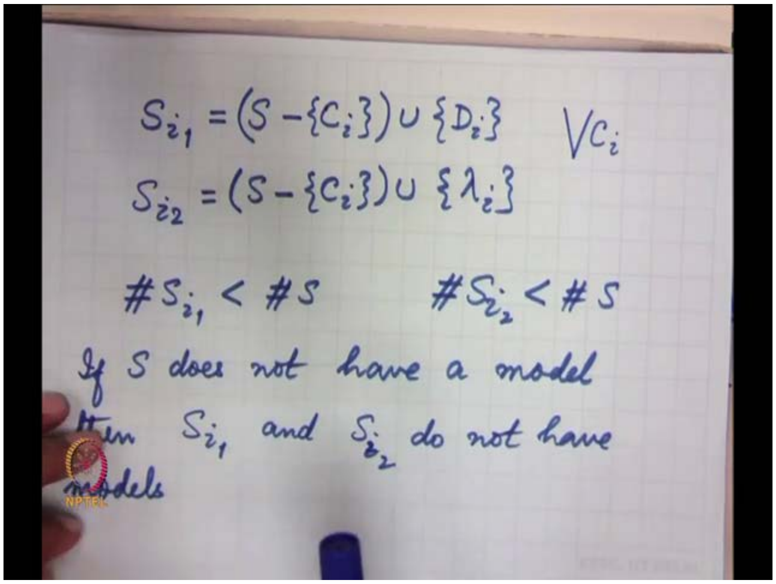
Before skolemization I think I had sometime after moving quantifiers prenex normal form next conjunctive herbrand algebra. Here, this is the one.

(Refer Slide Time: 31:28)



So, which means that there is a complementary pair so if there is a complementary pair then of course you have derived the empty clause. So, the basis is trivial so let us take the induction step so the induction step so this implies that some lambda i is the same as some lambda j bar for some i and j. So, the induction step is when hash S is greater than 0 if, hash S is greater than 0 and you have n clauses that means there is at least one clause one Ci which is not a singleton. So, let me assume that this Ci let me isolate one literal from it. So, I let me say that this is so it contains at least two. Two literals of which of course this Di is not empty.

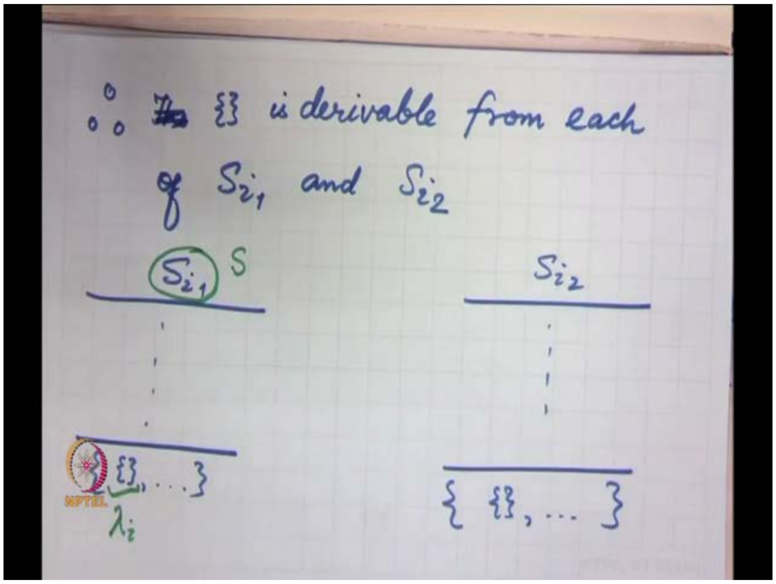
(Refer Slide Time: 33:26)



Now, let us look at so Si1 consist of S minus Ci union Di. So, basically this is nothing and let me consider Si2 consist of S minus Ci union this lambda i. So, we split one clause Ci into two parts which separated out one literal from the rest of the clause. And, now we are considering two sets where it is clear that hash Si1 is less than hash S and hash Si2 is less than hash S. Now, we have to do some reasoning the claim is that if S does not have a model. Then, Si1 and Si2 do not have models either because, from the soundness if it follows that. If, Si1 has a model then S would have a model because what are you doing this lambda i you separated out that Ci with this lambda I essentially what you are saying is that this Ci is logically equivalent to an or of the literals. So, if Si1 which does not have lambda i if it does have a model.

Then, oring a lambda i there does not change the model it does not make the model invalid the same model will hold and in fact on the. So, the same argument holds if this Si2 has a model then by adding of a few more literals in an or there you have not you are not destroying that model the satisfiability of the model. So, which means that so it follows that if S does not have a model then Si1 and Si2 do not have models they have measures which, are smaller than S. And, if Si1 and Si2 do not have models then from by the induction hypothesis it is possible to derive the empty clause from each of them. So, that is a next conclusion.

(Refer Slide Time: 36:41)



So, essentially what you are saying is that the empty clause I am just going to use this is derivable from each of Si1 and Si2. So, there are two resolution proof trees basically you start with Si1 and you go through some sequence of resolutions and you end up at one place where you have an empty clause and probably other thing too. But, the empty clauses are important thing. And, you start from Si2 you go through your resolution proof and it is possible to find an empty clause. Whatever, else might be there but that does not prove a done thing. Now, what I do is at look at this proof tree this Si1 did not have lambda i. So, I look at this proof tree and supposing and I say supposing, I had taken Si1 with a lambda i inserted there in Di. So, that this thing again I restore Ci back and then, supposing I look at this proof tree so. Which, means I have essentially replace this Si1 by S the proof tree still is valid except that there are two possibilities.

One possibility is that this empty clause is still derivable. The other possibility is that instead of empty clause being derivable I do not have the empty clause but I have this lambda i occurring as a singleton theory. This those are the only two possibilities if I consider the same proof tree then this is what will happen if I put back my lambda i into Di to restore Ci. Then, this is what can happen to this proof tree. Now, let us look at Si2 now Si2 had lambda occurring in it lambda I occurring in it. I claim that if, this empty clause here if this does not have lambda i occurring in it. Then, lambda i went as part of a resolution step somewhere here. If, lambda i went as part of some resolution step here. And, that resolution step was never used here then that resolution step can now be used here. If, lambda i is if lambda i continues to reside here. I mean in Si2 throughout this proof tree then the proof of this empty clause is independent on lambda i anyway. And, would be can be incorporated here so which means that lambda i would not have occurred in this empty clause. In either case I can derive the empty clause from S that is I hope they are disconvincing is that any there might I might be a sub-case I would left out.

Student: is that the theory relations will of lambda where I mean where we got lambda out of that.

Student: of that clear that how do you formally see how do we formally claim that indirectly.

No

Student: kind of the same relations and the relation of Si2 should be.

No we are not just doing a copy and paste it of the resolution steps we are doing something more complicated all we are saying is. If, you can act the exact occurrence of lambda i in this proof there are two possibilities either, lambda i went away as part of something so that some other clause had lambda i bar and lambda i and that lambda i bar were chosen literals for resolution and it went away. So, that means in this case there are two possibilities of course remember that so what can what so that means that if lambda i continue to remain here. Then, that lambda i bar was step was never used so there is a lambda i bar hanging around here. So now, I can resolve those two and I can derive the empty set.

Now, if what was the other case.

Student: there are still a lambda i here.

If, there is still a lambda i here if this lambda i that was there continues to resolve it somewhere here. Then, anywhere lambda i is not interest was not involved in the proof of the in the derivation of empty clause. So, which means I can still derive the empty clause here whether there is a lambda i or not. So, I could not have derived that without it and adding that lambda i I would still get this empty clause within this lambda i would not be in this in either case therefore I would have derived an empty clause from S.

Student: sir a lambda i bar still in

No I am considering two particular two separate cases.

Student: in that there should be a lambda i bar in the resolving down the Si1. So, my question is the is it not possible that lambda i bar could have been used in some other the resolution step or no.

But, then what I am saying is if then what you are saying you are using the fact that the resolution might involve more than one pair of literals. Because, you are talking of subsets of literals the same kind that is what you are saying. Then all I am saying is in this case also modify this proof so that this lambda i also goes as part of that proof. So, that is what the problem is but in either case I would have derived an empty clause from S. And therefore, for ground clauses at least this step is complete I mean in this resolution is complete.

So, this is actually the most interesting lemma because the other lemma the main theorem which we need to show is that in the presence of free variables and basically what are you saying there now. We will do resolution there if I have let us say p of x and naught p of y somewhere. Then, I will do this substitution x for y or y for x and may be do this do the resolution. But, that is essentially like saying take all ground instances of p of x, take all ground instances on of naught p of y. And, for every p of some term here I can find the naught p of the same term there and I can do propositional resolution. Except that could be infinitary the number of terms instances could be infinitary and so we resolve p x with not p y for example.

So, the propositional resolution proof is actually the most interesting. Because, the rest of the thing is really a formulization that if I did it with variables and then I can find various ways. The only other interesting aspect which is different from the propositional resolution is the fact that we are taking whole subset set of literals delta i that needs to be justified. So, for that we require some extra machinery so this lemma called a lifting lemma which I have to prove it. And then, I pro completeness for the main theory but what I am saying is I will read this lines and put them up. And I live it to you for a self study since on Friday we are having essentially an open session which we do not want to record here.