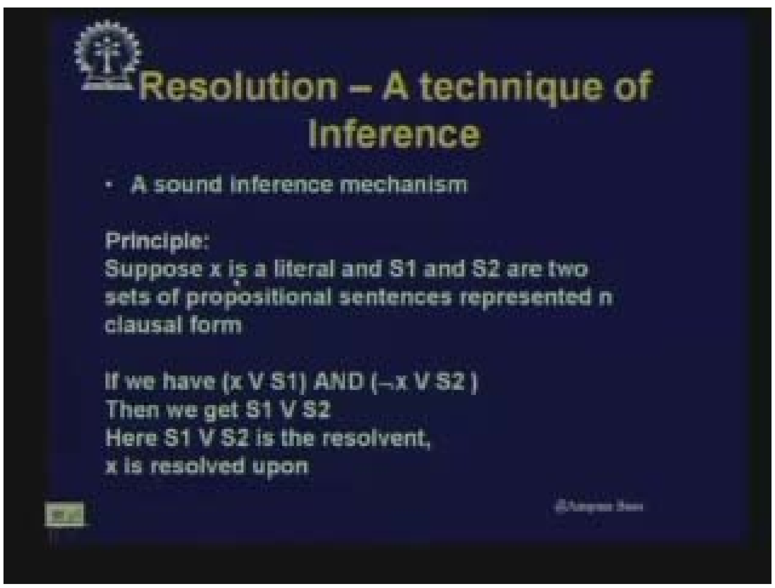
Artificial Intelligence
Prof. Anupam Basu
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture - 15
Resolution in FOPL

In the last lecture we had discussed about inferencing mechanism in first order predicate logic. We ended with examples of inferencing where we eliminated universal quantifiers, we eliminated existential quantifiers and also at times we introduced existential quantifiers. We also saw two very important operators as to how they are used. One is substitution and another is unification. Now, one very powerful inferencing mechanism used in inferencing in first order predicate logic is resolution. Resolution is an inferencing method that is sound and it can be automated as well. Therefore in today's lecture we will be discussing how we infer make inferences using the resolution method. Before we start discussing about resolution technique as applied in first order predicate logic we will have a brief revisit a revision of how it was applied in propositional logic. We discussed this earlier also. So let us quickly have a recapitulation of how resolution was applied in propositional logic and then we will see how it can be applied in the case of predicate logic.

(Refer Slide Time 03:00)

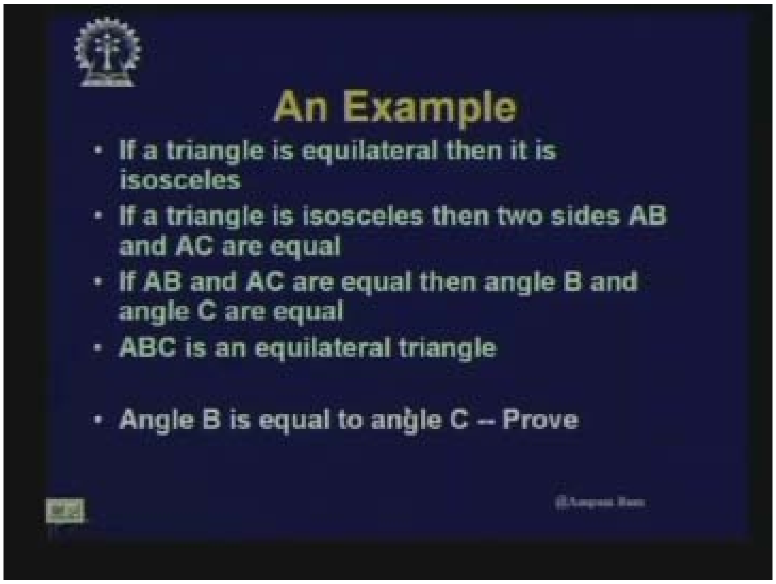


So here the basic principle is that, suppose (x) is a literal and S1 and S2 are two sets of propositional sentences, right now we are not discussing predicate logic cases but we are restricting ourselves to propositional logic cases. And what is the basic difference between propositional logic and predicate logic as we have seen? First, in predicate logic we can have variables whereas in propositional logic the parameters are all constant. And the second difference is that, in predicate logic we can have quantifiers over these variables and may need two type of quantifiers namely the existential quantifier and universal quantifier and such quantification is not there in propositional logic. So here suppose S1 and S2 the sets of propositional sentences represented in the clausal form, we already know what a Clausal form is, and clause is a disjunction of literals.

And also we mentioned what a horn clause is. In the horn clause on the right hand side of the implication there can be only one literal that means in the clausal form there can be at most one non negative literal because p AND q implies r in that case if I write in the clausal form p AND q implies r if I convert into the clausal form it will become NOT p OR NOT q OR. So if there can be only one literal on the right hand side and the left hand side are all conjunctions then when converted in the clausal form we will have at most one non negative literal. But clause is a disjunction of literals so suppose (x) is a literal and S1 and S2 are two sets of propositional sentences which we have represented in the clausal form. If we have (x) OR S1 AND NOT (x) OR S2 then we get S1 OR S2.

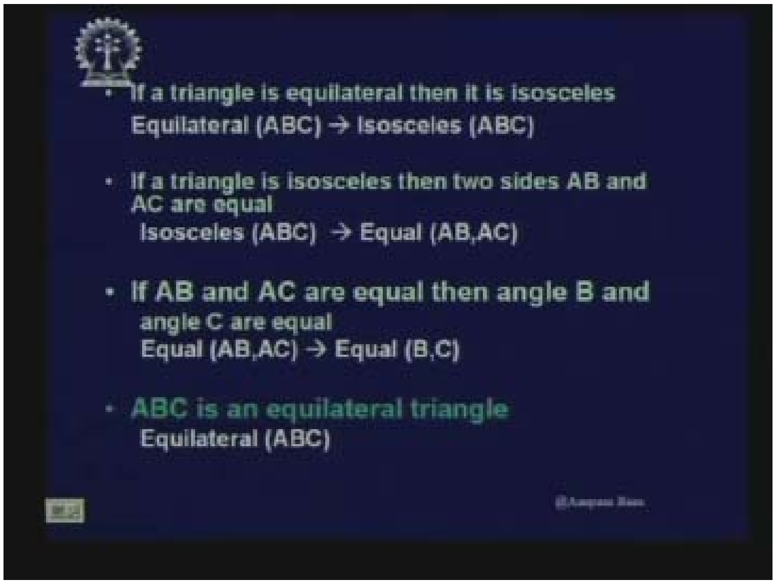
Look at this, this is a revisit, we have already shown the same slides while discussing the propositional logic. So here if I just do this AND then obviously (x) OR NOT (x) will be always be true. So S1 OR S2 will come true. Here S1 OR S2 this entire thing is called the resolvent. That means these two we have resolved in order to get this and (x) has been resolved upon.

(Refer Slide Time 06:15)



So the same old example, if a triangle is equilateral then it is isosceles, if a triangle is isosceles then the two sides AB and AC are equal, if AB and AC are equal then angle B and angle C are equal ABC is a equilateral triangle and we are supposed to prove angle B is equal to angle C.

(Refer Slide Time 06:38)



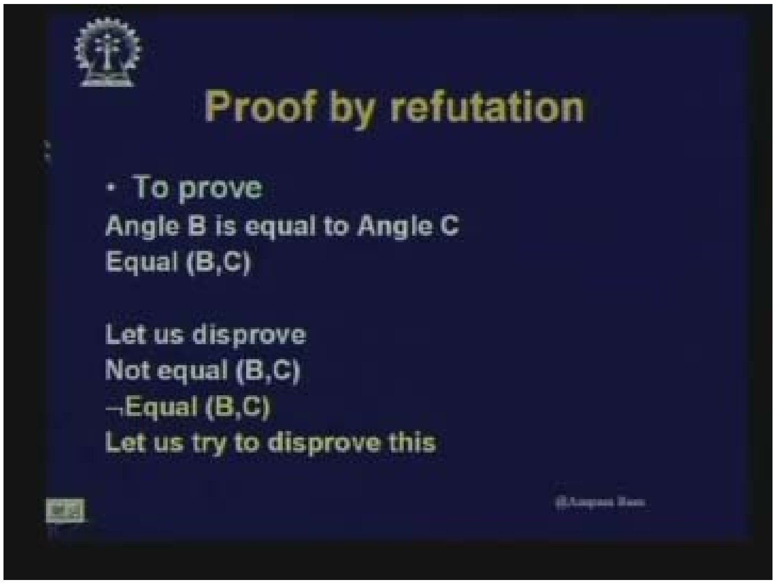
Now given this we next represent these in the form of propositions. If a triangle is equilateral then it is isosceles. So we can write equilateral ABC implies isosceles ABC. If a triangle is isosceles then the two sides AB and AC are equal that we can represent as isosceles ABC implies equal AB AC. If AB and AC are equal then angle B and angle C are equal. Written in the predicate form it turns out to be equal AB AC implies equal B and C where B and C are angles. ABC is an equilateral triangle so simply we can write equilateral ABC.

What is the next step?

We have to convert them to clausal form. Now we had equilateral ABC implies isosceles ABC converted to clause it becomes NOT equilateral ABC OR isosceles ABC. Isosceles ABC implies equal AB AC and that one translates to NOT isosceles ABC OR equal AB AC. Similarly, equal AB AC implies equal BC gets converted to NOT equal AB AC why is this NOT because if I convert this in the clausal form this antecedent part becomes negated, there is a mistake NOT equal AB AC should be changed to OR just has happened here NOT equal AB AC OR equal BC. And next one is equilateral ABC. So now we have got all these in the clausal form.

Now the basic of resolution is proof by refutation. That means to prove that angle B is equal to angle C that is equal BC. Our approach is to disprove that B and C are not unequal. This is a very well known proof method where in order to prove that B and C are equal we first try to prove B and C are not equal and fail to prove it. So we try to disprove NOT equal BC.

(Refer Slide Time 09:40)

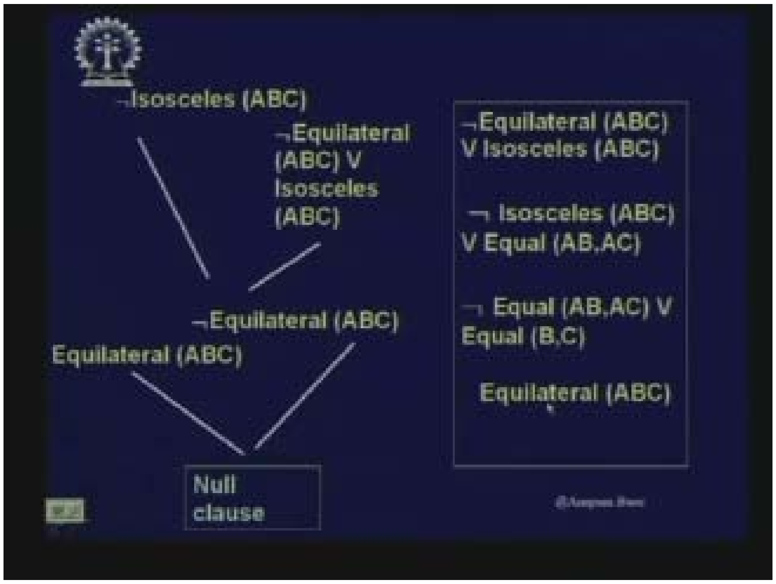


Our objective is to prove this equal BC. So we start with NOT equal BC and we try to disprove it. So what is there? We take the clausal form of the goal and negate it and try to disprove it and that is our approach. So let us look at this animated resolution. We have got in our knowledge base all these clauses equilateral ABC OR isosceles ABC etc that we have discussed and this is the one we want to prove. We want to disprove this NOT equal BC you have to disprove this.

Therefore, in order to prove equal BC we want to disprove NOT equal BC and then we have to resolve it and from here we will have to select one clause where we have got the negation of this equal BC that is here equal BC. This is NOT equal BC and here is equal BC so I select this clause and take this NOT equal AB AC OR equal BC. Now this part and this part will resolve together and we get NOT equal AB AC so this is the first level resolvent. Now we have to find the dual of NOT equal AB AC, there is equal AB AC here so that is this one.

Now I try to resolve this resolvent with this clause and as I do it what do I get?
I get equal AB AC and NOT equal AB AC resolved upon and we will get NOT isosceles ABC here. So this is a new resolvent and NOT isosceles ABC we start with that and which one should we select? Obviously it is this one because this one has got the negation of this. Here isosceles ABC is there so I select this and as I resolve these two what do I get? These two again resolve out and I get NOT equilateral ABC. And when I resolve out NOT equilateral ABC with this one equilateral ABC what do I get? I get a null clause. That means point from where I started is disproved. Here g is a null clause that means I am trying to resolve two contradictions. That means the goal with which I started is not consistent with the knowledge base that I have.

(Refer Slide Time 12:27)

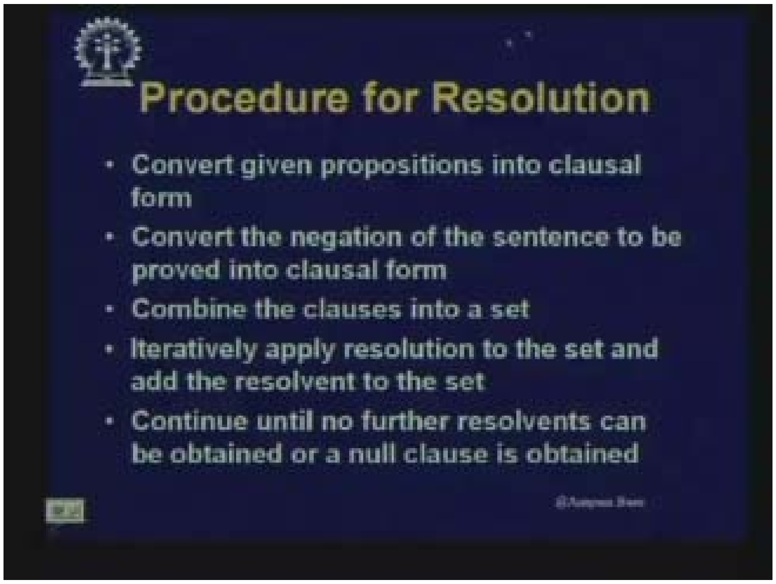


With the given knowledge base the goal which I started with is contradictory. Now what was the goal? The goal was the negation of the goal that I actually wanted to prove. So the negation of the goal that I wanted to prove is disproved. Therefore the actual goal is proved. That is the principle of resolution. And the same principle will hold for resolution in case of predicate logic as well. Next let us quickly see the basic steps of carrying out resolution for propositional calculus. First step is we convert the given propositions into clausal form. Next we convert the negation of the sentence to be proved. Please note that we take a sentence to be proved and then we first negate it and after this negation we convert it to the clausal form.

So convert the negation of the sentence to be proved into clausal form. Combine all the clauses in a set. And then iteratively apply resolution to the set. One of the candidates that we start with is the clausal form of the negation of the goal that I want to disprove. So I take that one and take a suitable clause from the remaining clauses in the set and try to resolve it. If the resolution is possible then through this resolution I will get a new clause. Then I will take the new clause and take another one and try to resolve it further until all the clauses are exhausted or I come across a contradiction that is a null clause.

As soon as I get a null clause then I immediately prove that the negation of the goal is contradictory therefore the goal is true. So I iteratively apply a resolution to the set and add the resolvent of every iteration to the sentence. Then we continue until no further resolvents can be obtained or a null clause is obtained and that is where we stop. That is the way resolution works. And we have seen it through an example. Next, when we try to do it for predicate logic there will be some differences but the major principle of resolution is remaining the same, the major approach is remaining the same. Here also we will start with the negation of the goal converted into clausal form and the procedure of generation of resolvents are the same.

(Refer Slide Time 15:31)



However since you realize that we are now going to deal with predicate logic and not with propositional logic we will have to handle the cases of variables as well as quantifiers. And in the last lecture we have already seen how these quantifiers can be handled and how we can get the clausal form. So those steps are to be followed in order to apply resolution for first order predicate logic as well. So let us start with another set of examples.

Let us consider a few statements as shown here. All people who are graduating are happy. All happy people smile. Someone is graduating. Is someone smiling? That is the conclusion I want to prove. Here it has been posed as is someone smiling? I want to have an answer yes or no but that is equivalent to proving a statement someone is smiling.

Now, if that is true then the answer is yes, if someone is smiling is disproved that means no one is smiling then for, is someone smiling the answer is no. Now, if you look at these sentences you will soon realize that none of these sentences can be really represented in the form of propositions. Let us forget about mechanical inferencing for the time being. If we just apply our common sense reasoning let us see what comes out. All people who are graduating are happy. All happy people smile. Therefore I can reason and say all people who are graduating are smiling. And then someone is graduating. That means there is at least one person who is graduating and anybody who is graduating is smiling. So, is someone smiling should be yes or if I write that someone is smiling that should be true. But however a computer will not apply common sense in that way. It needs a mechanical procedure by which the same reasoning can be carried out.

(Refer Slide Time 19:07)

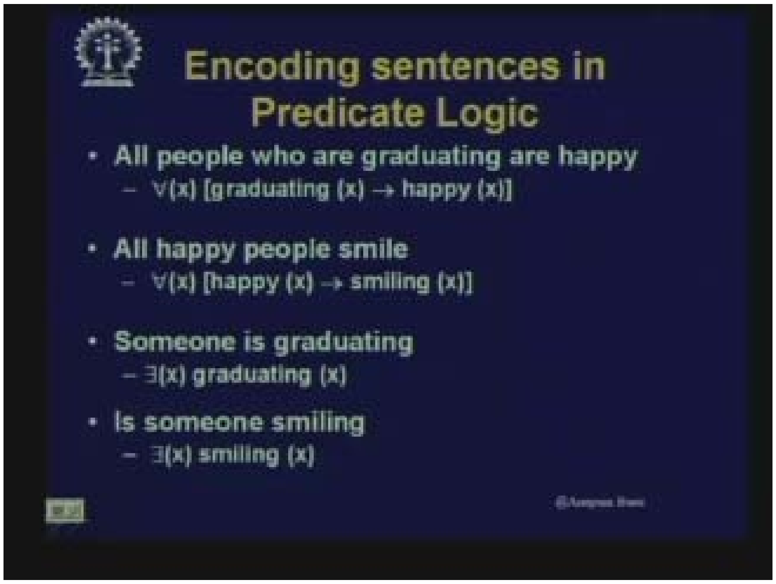
Therefore we intend to code this problem in first order predicate calculus. And then we will use resolution refutation, this resolution is also known as resolution refutation. Now why this word refutation is coming in? You can very simply guess. What are we trying to refute? In order to prove a particular goal g we are trying to refute NOT g and if we can refute NOT g then we are proving g. That is what resolution is all about. So we are using resolution refutation to solve a problem.

What is solving a problem?

Solving means finding out whether the conclusion can be answered from the given set of sentences. The conclusion is whether we can prove the statement; is someone smiling using whatever is there. Now, in order to capture the sentences we had shown we will have to first convert them in the form of predicates. In order to make the statement (x) is graduating someone is graduating we have to select the predicate and we have selected the predicate graduating (x) and that shows (x) is graduating.

Depending on the programming language we are using there will be some syntax's, here the conventions should have been in the order that all these predicates start with small letters. Here, (x) is happy I can write as happy (x), (x) is smiling, smiling (x). Now we try to encode these sentences into predicate logic using the predicates that I have just now stated. All people who are graduating are happy.

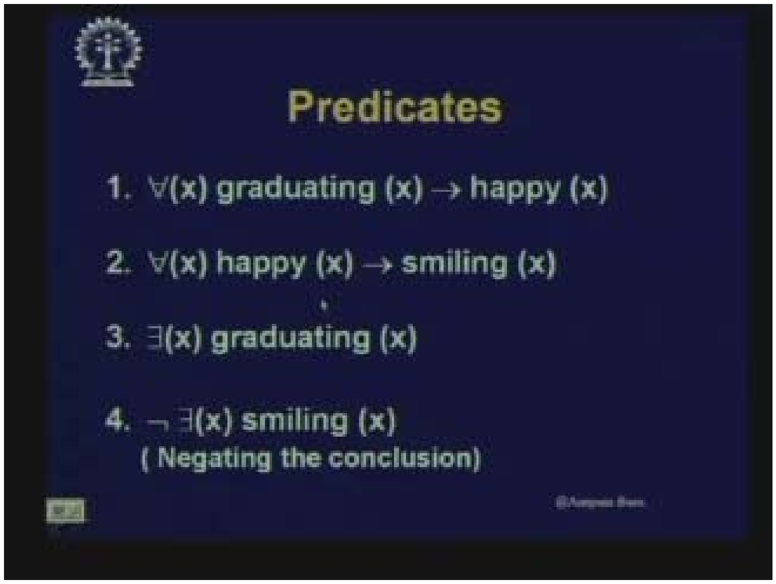
(Refer Slide Time 21:50)



That can be certainly written as; for all (x) graduating (x) implies happy (x). All people who are graduating are happy and that is universal quantification so there is no problem with that. All happy people smile is again universal quantification. someone is graduating, here it is shown but you could have also done that, someone is graduating, and this parenthesis is not required like this here because there is no scope of it, there exists (x) such that graduating (x) that is someone is graduating. Is someone smiling? That was my objective but let us consider that this statement is; someone is smiling. That means there exists (x) such that smiling (x) is true.

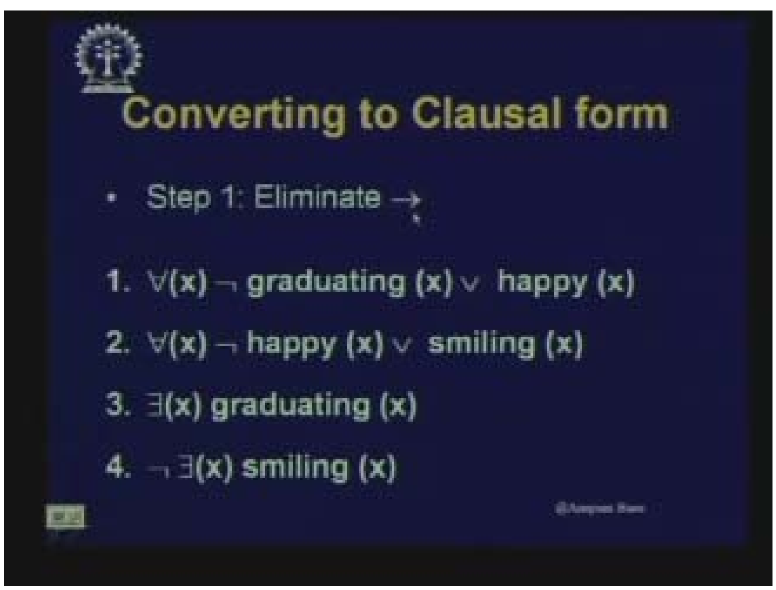
So, if there exists (x) smiling (x) is found to be true then the answer to this question is someone smiling is yes. So the first step was selecting the suitable predicates. And once the suitable predicates are chosen then the next step you do is write down these sentences express those sentences using predicate logic, using implication, using quantifiers. But the next step would be, you take these predicates here as they are but here I have made a small change, is someone smiling or someone is smiling was stated as there exists (x) such that smiling (x). And that was the conclusion and the negation of the conclusion therefore is there does not exist (x) such that smiling (x). If I can prove this that there does not exist (x) such that smiling (x) that means no one is smiling. But if I disprove it that means someone is smiling. So we first negate the conclusion and then keep the others as they are. I have not made any change in this.

(Refer Slide Time 24:58)



But the next step is we have to convert them into clausal form. In order to convert them into clausal form the first step would be to eliminate the implication operation and to eliminate this implication operation you know that we will take this clause for example. We are retaining this for all (x) for the time being and graduating (x) implies happy (x) will be not graduating (x) OR happy (x).

(Refer Slide Time 25:31)



Similarly we can do it for the others. Happy (x) was implying smiling (x) so that gets converted to NOT happy (x) OR smiling (x). There exists (x) graduating (x) and this did not have any implication at all and that was as it is and the second one was also not having any implication, this one the fourth one was also not having any implication. Now I have got a set of clauses which are all free of the implication sign. The next step is to convert them to canonical form. Now it may be that this canonical form often called normal form may be new to you but let us not get too much bogged down with it, it is a form where quantifiers are there and we have got the examples of inference.

(Refer Slide Time 27:02)

Ultimately we are trying to go to the clausal form and it is basically a conjunctive normal form so there will be disjunctions and there can be disjunctions in clauses. The next step to do that is to reduce the scope of negation. So let us have a quick re look at what we had earlier. For the first two the scope of the negation is over this predicate graduating (x). And for this the scope of this negation is over this entire thing even over this quantifier. So the scope of the negation is fine over here I do not make any change here. However, just compare the earlier clause number four and the present clause number four. Here the scope of negation was over this quantifier as well as this statement.

Now I want to push this negation as much as close to the particular predicate. So negation of the existential quantifier is a universal quantifier. For example, I say there does not exist a person, there does not exist any (x) such that NOT mortal (x) but (x) is a man. So there does not exist I can convert it to for all (x) mortal x. I say that there does not exist a person there does not exist (x) such that NOT mortal x. So the same meaning is communicated if I say for all people for all (x) mortal (x) so I convert that there exists for all and NOT mortal to mortal. So I push the negation inside so that is exactly what is being done over here. This earlier one gets transferred to; for all (x) NOT smiling (x). So I have reduced the scope of negation here.

The next thing we need to do is, we have to standardize the variables apart. It is possible that we can put in the same variable that might have been used at different points. For example, here graduating (x) happy (x) here again NOT happy (x) smiling (x) all those things but when I am trying to do this what I will try to do is, for every clause I will change the different variables; every sentence will have a distinct variable. Now if you recollect the idea of unification, now this automatically gives rise to some problem that if we have got different types of variables then we must have some mechanical means of converting those variables in some way in order that we can do the unification.

(Refer Slide Time 32:22)

After we do this the next thing that we need to do is standardize the variables. So here you can see for graduating (x) happy (x) happy (x) or smiling (x) we take the liberty of using different variables. For all (x) NOT graduating (x) OR happy (x) so this one has got x, this has got y. Now if there were if there was a clause like graduating (x) and happy (y) in that case both (x) and (y) are reserved for that clause. In the other clauses we will have to use y z p q all those things. And only later as we will see we will try to see whether these variables can be compared or unified. You remember what we discussed about unification.

Unification is the process of finding a particular substitution of variables such that two different clauses become identical. Therefore we will have to do that unification anyway. Here we have selected smiling (y). Now here again we have introduced another one z and for all w smiling (w) so here we have made a change again. You can very well realize that still it is not a clause. So let us quickly recapitulate whatever we have done till now. What we have done is we are trying to convert it to clausal form.

The first step is we have selected some predicates and after selection of the predicates we encoded the given sentences in the form of first order predicate logic. And then we negated the goal and put it in the knowledge base then we will start converting then we took the canonical form or the clausal form, how do we do it? In order to do that we will first eliminate the implication sign and that can be achieved by the simple rule that we have come across so many times p implies q is not p OR q and there also we convert the entire knowledge base including the negation of the goal.

After that we reduce the scope of implication and after that we try to bring the negation as close as possible to the variables and in that process we may require to change the quantifiers also. For example, NOT for all (x) will become there exists (x) NOT there exists (x) will become for all x. After reducing the scope of negation we standardize the variables. Once these variables are standardized then we try to push all the quantifiers to the left. In this example the quantifiers were not very much within but there can be complicated clauses.

(Refer Slide Time 36:08)

For example, there can be something like this; for all (x) I am not bothering about too much of meaning here say P(x) implies there can be something else there exists y such that Q y it is possible like to express something like this it is possible. So in such cases there are two quantifiers but ultimately my objective will be to push all the quantifiers to the outside as much as possible. So the next thing that we try to do is, we move all the quantifiers to the left and using which we get in this case there was not much of change we come to these four clauses. Then we have to start working on elimination of the quantifiers.

First of all let us try to eliminate the existential quantifier. And we are already introduced to the process of skolemization. skolemization is a process of finding a constant or a function for the existentially quantified variable that will make this clause to be true. For example, here there were no existential quantification at all. The only existential quantification in this case we had was this. That was the only existential quantification. It was, there exists z such that graduating z we remove that and in place of that we find out (name1).

(Refer Slide Time 39:05)

(name1) is a skolemization constant. It is a name that somehow we are finding out who is graduating. So that particular value of (x) or (z) is the case is satisfying this statement graduating (x). In the last lecture we had also shown that in some cases we replace it NOT with a constant but with a skolemization function. Like in the prime divisibility case we had selected a function p d. So we have removed the existential quantifier that was the step five.

(Refer Slide Time 40:01)

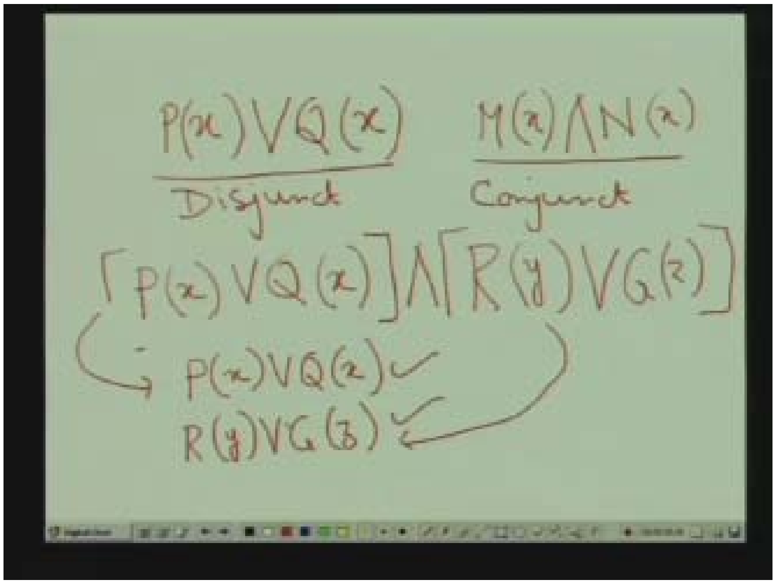
What is the sixth step?

We drop all the universal quantifiers because of the simple reason we do not really need it. By now I have removed the existential quantifiers by now I have reduced the scope of negation to as close as possible to the predicates. And by now I have pushed all the quantifiers to the left and the existential quantifiers have been removed already. So what is left is just the universal quantifier and all these clauses are now implicitly universally quantified. So I need not specify any further and so I can simply drop them. And as I drop them I get these clauses. In this case again we only had disjuncts.

In a canonical form it can be conjuncts of disjuncts what is a disjunct?
Again let us go here; I have got P(x) OR Q(x) this is a disjunction. And say M(x) AND N(x) this is a conjunct. This is a disjunct and this is a conjunct. Now, canonical form is a conjunction of disjuncts. So I can say that P(x) OR Q(x) if I have a form M(x) OR N(x). Now this is R(y) OR G(z). Now this is another disjunct. So this is a disjunct, this is a disjunct and I have got a conjunction of this disjunct so it is a conjunct. Therefore canonical form will be a conjunction of disjuncts only and that is the canonical form.

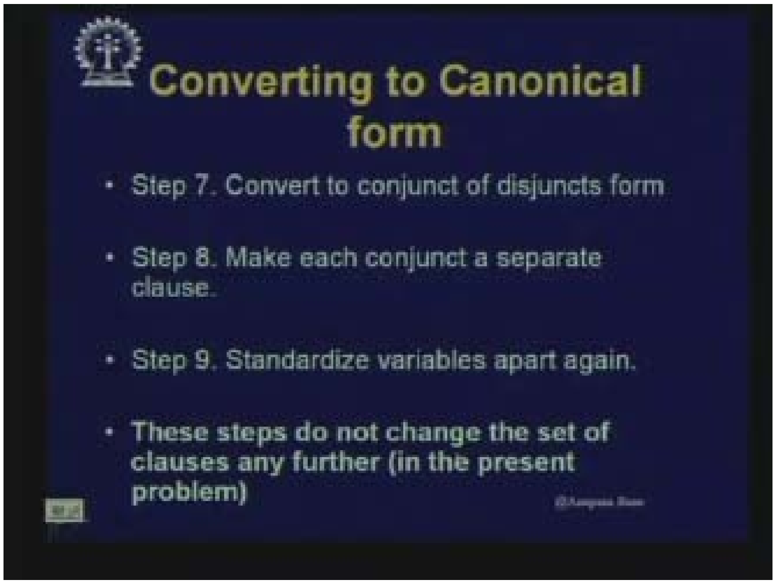
What is the implication of this?
The implication of this is, if I have conjunction of disjuncts, whatever I have written here I just take that as an example and if you just think a little while you will realize that once I have conjuncts of disjuncts then each of these disjuncts I can write down separately. And this one I can write down separately as P(x) OR Q(x) and this one I can write down separately as R(y) OR G(z). Now according to our definition both these are clauses and a conjunction of disjuncts therefore each of these disjuncts can be written down as a separate clause and that is our objective, convert in the clausal form.

(Refer Slide Time 44:33)



So, after we have dropped all these quantifiers we convert to conjunct of disjunct form and make each of these as a separate clause and we standardize variables apart again. So each of these clauses will now have separate variables. These steps do not change the set of clauses in the present problem.

(Refer Slide Time 44:51)

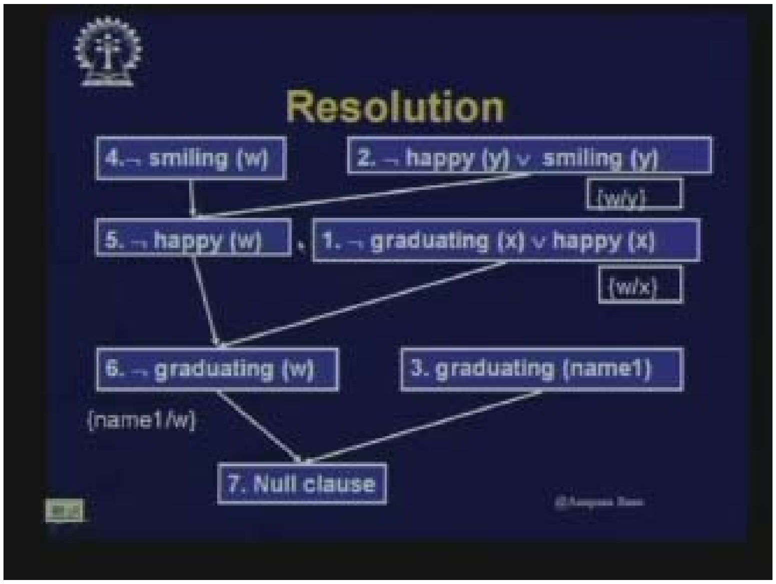


Now we come to the resolution problem. The steps 7 and 8 are not so much important here because by step 6 here we have already got them in the clausal form. Therefore we now can apply resolution and let us recall how should we start resolution? We will start with the clause that is the negation of the goal to be proved. Then from the remaining clauses in our knowledge base we will select the clause that can resolve this. Now start with this NOT smiling (w) and I will try to resolve with this with which one should I resolve?

I will select from these clauses the clause which has got complementary of the negation of this so obviously this clause clause two should be selected. And if we resolve these two then what we get? The resolvent is NOT happy (y). But there is a big question here. In case of propositional logic the problem was not there but here since I have got variables and I have standardized the variables NOT smiling (w) and smiling (y) can be resolved. they can be resolved only if there is a possibility of substitution, there is a possible correct substitution, if I can substitute y by w then only I can unify these two, then only this smiling (y) and smiling (w) will become Q y. So, merely resolution will not do.

And what will be the result of the resolvent? Will it be happy (y)?
Here I have NOT smiling (w) and here I had the clause two NOT happy (y) OR smiling (y) and I can resolve this only under a substitution where w is substituting y. So this one becomes happy (w), smiling (w) and this smiling (w) and NOT smiling (w) resolve out and I get NOT happy (w).

(Refer Slide Time 48:46)

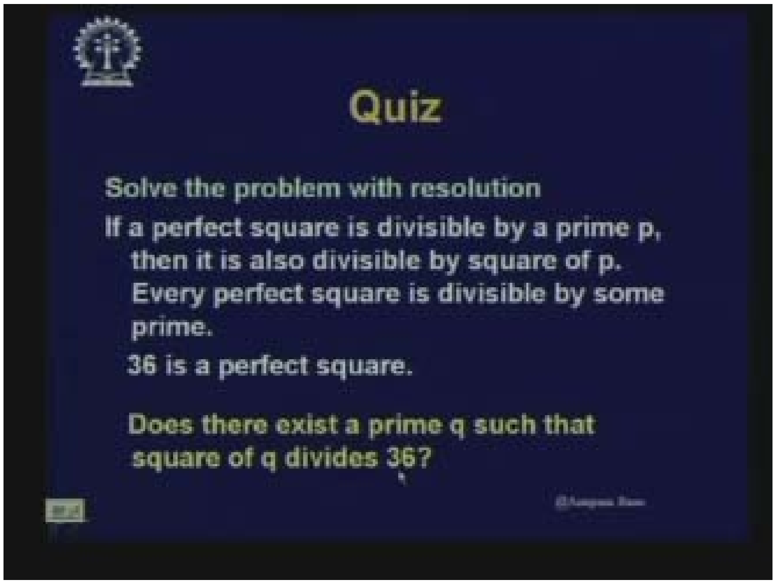


Again here when I get NOT happy (w) the next thing I have to select is some clause from here. For example, here it is happy (x) so I will try to unify them. Now, this unification is very important. Here I selected this graduating (x) OR happy (x) so I now replace this (x) with w so I get happy (w) graduating (w) and I combine these two and I get NOT graduating (w). Here I replaced resolving for the sake of resolution I substituted y with w, here I substituted (x) with w and there was no contradiction. And I came to NOT graduating (w) and there was a clause graduating name that is (name1). So now this variable here w is substituted with the constant (name1). And with this I resolve what do I get? I get a null clause. And what does this null clause imply? It implies that the goal with which you started is contradictory. And what is the goal with which I started; it was the negation of the goal to be proven. So the contradiction of the goal to be proven has been disproved so the goal is proved. Is someone smiling? Yes, someone is smiling.

See the major difference between these two. It is the resolution in propositional logic and resolution in predicate logic. The problem is, here we have got variables so we have to take some extra pain in order to convert them to clausal form. We had to convert them into clausal form in propositional case also but here we had to take a little extra pain. And after that we have resolved them and in order to resolve them we had to unify, we had to give a proper substitution. So this is a very powerful inference mechanism applied in many mechanical proving or theorem proving systems and other applications.

Here is a problem to solve. This is the problem we solved in the last lecture using other inference mechanisms. But the same problem you try to solve using resolution. The problem is if a perfect square is divisible by a prime p then is also divisible by the square of p. Every perfect square is divisible by some prime, 36 is a perfect square, does there exist a prime q such that the square of q divides 36. So the same thing you convert into clausal form and try to obtain the solution.

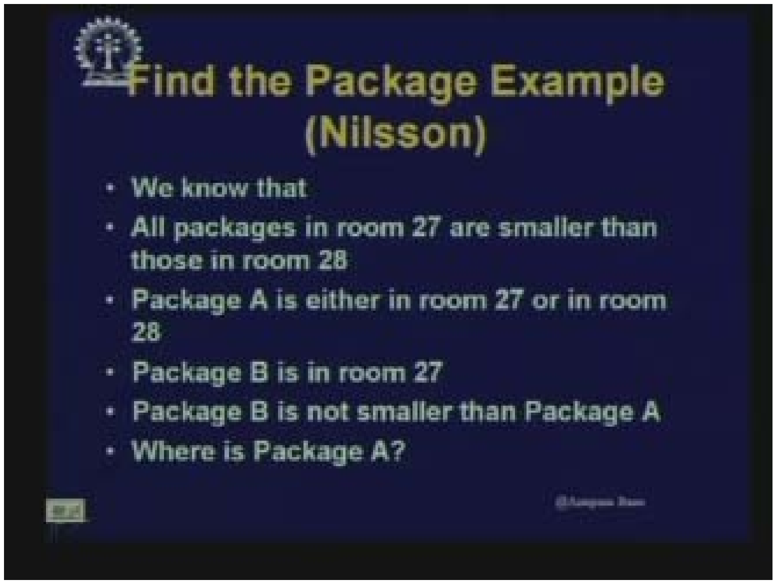
(Refer Slide Time 52:01)



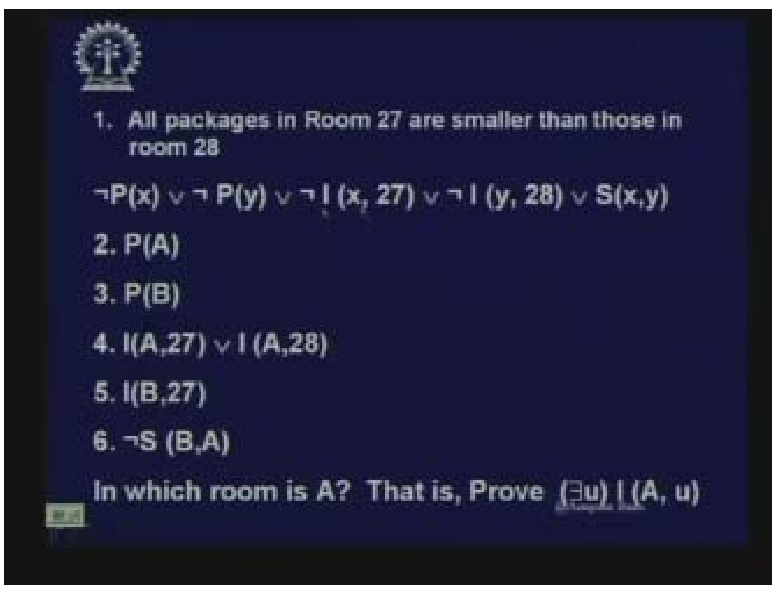
Next, we will quickly look at some very interesting application of this resolution theorem proving. Answer extraction: Given a set of clauses a knowledge base we want to find the answer to a question. If you were noticing, here also ultimately when we got the null clause we could answer the question is someone smiling? We got a null clause because we found that if I had really asked who is smiling you have proved that someone is smiling, but who is smiling? The answer lies here because here this (name1) which was a constant at least he or she is smiling so that answer is there. But now I want to mechanically have that answer at the leaf of my resolution tree. Here is a nice problem: This example has been taken from the book of N. J. Nilsson. All packages in room 27 a particular room is smaller than those in room 28. Room 28 can have some smaller packages also. But all packages in room 27 are smaller than those in room 28.

Package A is either in room 27 or in room 28 the robot does not know. Package B is in room 27. Package B is not smaller than package A. Now the question posed is, where is package A? Now how can we answer this question? Here what we will do is we will just forget about this part for the time being. We will add another literal to each clause that will come from the negation of the theorem. For example, it says all packages in room 27 are smaller than those in room 28 can be converted in the clausal form in this way because all packages in room 27 are smaller than those in room 28 so (x) is a P(x) AND P(y) AND in (x) 27 that means package (x) is in 27, package (y) is in room 28 and (x) is smaller than (y).

(Refer Slide Time 54:57)



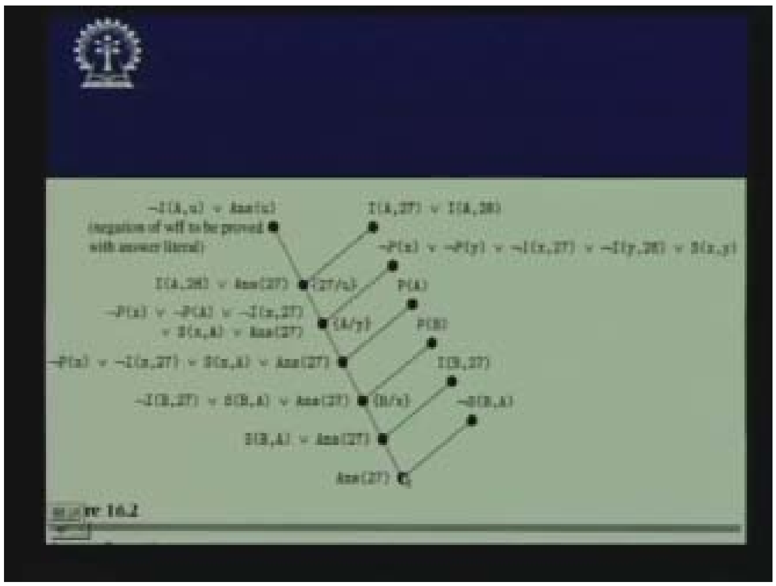
(Refer Slide Time 55:18)



Now, if I convert them in clausal form I get this: A is a package B is a package A is in room 27 OR A is in room twenty eight that was the problem here. Package A is in either in room 27 or 28 and package B is in room 27. And package B is not smaller than package A. Now my question is in which room is A. that is, I have to prove that there exists a some room number there exists u such that in I standing for in A u, A is in u. so that is my goal query. So I have to negate it for resolution. So the negation of the clause is this is as straight away taken from Nilsson's book: NOT of A, u and only thing I have added is for this u I have got an answer literal.

As I go on resolving this obviously I will select a clause which can resolve this out so I take this and I can resolve it with the substitution of 27 for u otherwise this is not substituted. So u is substituted with 27 and I get the resolvent A twenty eight my resolvent becomes this. And with this now I look for a particular clause which can resolve this out. I take the first clause and as I resolve this I resolve it with a particular substitution that is y is replaced with A and based on that when I go on doing it I am remembering the substitution and in this way as I come here I substitute this and ultimately when I get this contradiction this leaf node will come to 27 and that is my answer.

(Refer Slide Time 57:55)



In the normal resolution case this part was not there, if it was there I would have started with this and ultimately I would have come to a null clause here. But if I start with adding with the negation of the goal a clausal form of that or I put this answer of that literal u in which I want to hold the answer. Then the substitution values will automatically put me in the resolve I want and that will come as a leaf node. So the same resolution method can be applied to carry this out.

So in this lecture we have seen that we can apply the resolution refutation method for the case of predicate logic with some additional techniques as discussed namely the clausal form, the unification part and also we can apply this to find out answers to some questions by applying it in an intelligent manner.