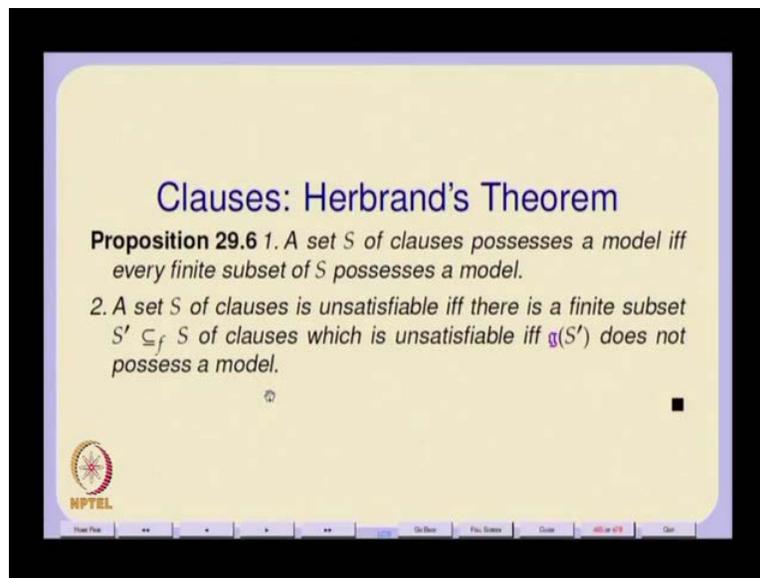


Logic for CS
Prof. Dr. S. Arun Kumar
Department of Computer Science
Indian Institute of Technology, Delhi

Lecture - 30
More on Resolution in FOL

So, we were looking at resolution. So, for a long time actually most theorem proving systems were actually resolution based proving systems so it is important in first order logic. Let, us just recapitulate what we did last time and then do some examples here. Then, proceed to a so today I will just do some examples to explain the basic things and I will not prove anything theorems today.

(Refer Slide Time: 01:00)



Clauses: Herbrand's Theorem

Proposition 29.6 1. A set S of clauses possesses a model iff every finite subset of S possesses a model.

2. A set S of clauses is unsatisfiable iff there is a finite subset $S' \subseteq_f S$ of clauses which is unsatisfiable iff $\mathcal{G}(S')$ does not possess a model.

■

NPTEL

So, we actually had so the whole point of course is that you what we.

(Refer Slide Time: 01:09)

Resolution in FOL

Let S be a set of clauses, $C_i, C_j \in S$ with $FV(C_i) \cap FV(C_j) = \emptyset$ and p an atomic predicate symbol such that

- $C_i = C'_i \cup \{p(s_{i'}) \mid 1 \leq i' \leq m_i\}$
- $C_j = C'_j \cup \{\neg p(t_{j'}) \mid 1 \leq j' \leq m_j\}$
- $L = \{p(s_{i'}) \mid 1 \leq i' \leq m_i\} \cup \{p(t_{j'}) \mid 1 \leq j' \leq m_j\}$ is a set of unifiable literals.
- $\mu = \text{UNIFY}(L)$ is an mgu of L .

Then

$$\text{Res1} \frac{S}{(S - \{C_i, C_j\}) \cup \{\mu(C'_i \cup C'_j)\}}$$



So, Resolution is a single rule one monolithic rule of which all these are the conditions I mean so these are like all the side conditions on this rule if you like. So, essentially all this says is first of all you should standardize the variables apart so you take a set of clauses. In which, means that there are each clause is implicitly universally quantified over all the variables of the clause. And, what we have from is column's theorem is that this set of clauses so if I had some original as formula or a set of formulae. Then, that this in set of clauses is as column conjunctive normal form of that original formula let us say ϕ . And, that original formula has a model if and only if this column conjunctive normal form has a model. And, of course it is the transformation unlike in the case of prenex normal forms and so on the transformation form of formula is ϕ to its column conjunctive normal form does is not preserved under logical equivalence ψ .

So, it only preserves models so it does not preserves so it preserves satisfiability and unsatisfiability but, it does not preserve logical equivalence. So, the existence of so that is all so a resolution is most often used in the form of a showing that showing unsatisfiability that is they are no models that is. And however, that is not just the only where the resolution can be used. So, let us assume that we have a set of clauses set S of clauses. And since, each clause is suppose to be implicitly universally quantified then they are universally as we assume that between two clauses we will use alpha renaming to standardize the variables apart.

And, so you have this the free variables of C_i into section of free variables C_j is empty. And, you choose any atomic predicate symbol which occurs in both which occurs in positive form mean one of the clauses and negative form in another of the clauses. So, the notion of complimentary pair as generalized from propositional logic should exist in some form here. It is what we should realize is that it is quite possible that since we are dealing with parameterized propositions so essentially if you predicates the parameterized propositions. This predicate symbol could appear in positive and negative forms within the same clause itself without there being any contradiction. And, so it may also happen in they occur in both positive and negative forms in this clause too. But, that is all we are saying is that you should have at least one positive existence in one clause and negative existence in another clause that is.

So, if it is available in such a form and if you can actually unify some subset of the positive occurrences in one clause C_i with some subset of the negative occurrences in the other clause C_j . So, here it is C_j . C_j prime union and naught of pt_j prime. So, we take this pt_j prime then we are saying now you take this union and if you and find essentially the most general unifier it has to be unifiable. And, if you can find this unifiable set and you take this most general unifier then the rule essentially says that I can take this I can remove these two clauses C_i and C_j from the set S . And, add a new clause which is obtained by applying this substitution the result of unification is a substitution if there are unifiable then a substitution μ which is the most general unifier is available.

So, apply this most general unifier on C_i prime union C_j prime notice that of course if had applied this μ of this set and the μ of this set will make complimentary pairs. We will make sets of complimentary pairs so essentially they sort of cancel out each other. So, what are left out are only those which are not being unified in this from this. Only those which are not in this set L only those terms are left. And, essentially this unified version is what is as a logical consequence of C_i and C_j and early in the game we proved to theorem which said that if you added more assumptions on so and so that you can carry forward those assumptions basically.

So, this the rest of S basically remains unchanged so at a time you take just two clauses identified positive and negative occurrences in the two clauses. Some subsets find positive and negative occurrences such that you have an identifiable subset which is unifiable and which is non-empty.

(Refer Slide Time: 07:51)

Algorithm: Unification
Require: $S \subseteq_f \mathcal{T}_\Omega(V)$ and $|S| > 1$
Ensure: If S is not unifiable then fail else $\exists \theta \in \mathcal{S}_\Omega(V) : |\theta S| = 1$ and θ is a mgu of S

- 1 UNIFY(S) $\stackrel{df}{=} \text{PARTIALUNIFY}(1, S)$ where
- 2 PARTIALUNIFY(θ, S) $\stackrel{df}{=} \dots$
- 3 let $T = \theta S$ in
- 4 if $|T| = 1$ then
- 5 return θ { θ is a mgu of S }
- 6 else {There is a position at which at least two terms are different}
- 7 let $D = \text{DISAGREEMENT}(T)$ in

NPTEL

And, actually calculate the unifier which is through this unification algorithm which of course is guaranteed to give the most general unifier. So, the rule is just this so this all that we have to do and you this there are no other axioms no other proof rules this is the only rule. So, you can think of first order this in many books this is called the resolution calculus that is it. So, this is a you take the first order logic and essentially take is column conjunctive normal forms, take the subset column conjunctive normal forms and just apply just have one rule that is it one monolithic rule and you can apply keep applying this rule repeatedly.

So, the first thing is of course so this unified version is a actually a logical or of course not remember that this is a clause. So, in that sense its simplicity universally quantified so it is a universe it is like some kind of an application of for all elimination. But, then that for all elimination also allows variables and then universally generalizing on that. So, it is like an application of firstly you have only the universal quantifier so if you want to get some intuition as to why there is this just one rule required it is very simple. This unification is a combination is like a combination of twisters. You have for all instantiation for all elimination which essentially replaces the variables and C_i and C_j with some terms and those terms are essentially the terms which give you the most general commonality between those bilaterals and L .

And then, what are you doing? You are implicitly you so that those terms themselves have do have variables some variables might be new some variables might be old. But, by treating it as clauses what you are saying is that your implicitly applying a universal generalization for all introduction on the variables that are present there after the substitution of mu. So, it is like a combination of these two and that is so in a certain sense it does make intuitive sense is that you do not require anything else. We already have propositional resolution for other things so, that is not a serious issue the completeness the so our completeness will essentially involve trying to prove propositional completeness for resolution and predicate completeness but, that will come to at a later stage. Now, it makes intuitive sense that if you keep up applying universal instantiation that means for all elimination. And then, follow it up with for all introduction you get something that is a logical consequence of your original set of formula.

(Refer Slide Time: 11:33)

Remarks on Resolution

1. The free variables of distinct clauses *must* be disjoint and variables should be renamed if necessary. ⊗

Example 30.1 Let $S = \{C_1, C_2\}$ where

$$C_1 = \{p(x)\}$$

$$C_2 = \{p(f(x))\}$$

S represents the set of closed formulae

$$\{\forall x[p(x)], \forall x[\neg p(f(x))]\}$$

which is clearly unsatisfiable. However the empty clause cannot be derived (because of the *occurs check*) unless x is renamed in one of the clauses.

NPTEL

So, that is what this thing is so let us look at some examples of first resolution. First remark is that of course this is the standardizing apart the free variables of distinct clauses must be disjoint. And the variable should be renamed if necessary and that is equivalent to alpha renaming for non-variables. Otherwise if you were to take this by the way here there should be a negation sign here which I have forgotten. So, if you take the set of two clauses px and $\neg p(fx)$. Where, f is some very function symbol it is clear that this set of formulae is actually unsatisfied for all instances of x that you can find it is not guaranteed to simultaneously satisfy both of this.

But, however what will happen if you do not do a standardization apart is that any correct unification algorithm will encounter the occur check problem.

(Refer Slide Time: 13:01)

Example: Occurs Check

Example 28.13 Consider the set

$$S_2 = \{f(g(z), x, h(g(z))), f(z, h(y), h(y))\}$$

The disagreement set $\{g(z), z\}$ is such that S_2 is not unifiable, because for any substitution θ $\theta f \theta z$ can never be syntactically identical with $\theta g(z)$. This is an example of the notorious occurs check problem. Hence S_2 is not unifiable.

 NPTEL

If, you remember the Occurs Check this is you cannot unify say gz z mean if so you cannot unify f of x and x for example. So, any correct implementation of the unification algorithm will be faced with the occurs check problem. So, which means that you will not be even though this is unsatisfiable and therefore there is a equivalent of proof by contradiction in the resolution calculus is called a reputation we will see an example of reputation later.

Student: sir when is the last example this example also inspirable (Refer Time: 13:43) since for all x px and for all x p of f x .

No naught of p of x .

Student: naught of p of x .

That is there should be a negation symbol here.

Student: this is also a (Refer Time 13:54).

Look at this you get it this way intuitively if this second one is true then there is some value in some model supposing this is satisfiable. Then, it has some model that is and there is some value

of which you can give to x definitely. Which, makes this true which makes p of f of value a let us say false which is which clearly contradicts this. So, it is not satisfiable and what makes it unsatisfiable is just the fact that is everything is universally quantified that is what. So, which means that you have to be able to unified and if you try to unify this without standardizing the variables apart. You will never actually reach the conclusion of unsatisfiability though it is first to be unsatisfiable. So this is the first remark is that you have to actually separate throughout all the variables from the different clauses by doing an renaming of them it makes all variables unique basically.

(Refer Slide Time: 15:27)

Factoring

2. Unlike in the case of **propositional resolution** it should be possible to eliminate several literals at once

Example 30.2 Consider the set $S = \{C_1, C_2\}$ where

$$C_1 = \{p(x), p(y)\}$$

$$C_2 = \{\neg p(u), \neg p(v)\}$$

S is clearly unsatisfiable but by removing only one literal at a time with the substitution $\{x/u\}$ yields the new clause

$$C'_{12} = \{p(y), \neg p(v)\}$$

from which the **empty clause cannot be derived.**

NPTTEL

The second thing is called Factoring and that is it should be possible to eliminate several literals at the same time. So, that is why unlike the case of resolution in propositional logic. Here, we took a subset in each case in each clause if, you do not do that. Then, you have some consequences so take something like this if you have two clauses $p(x)$ and $p(y)$ a essentially there implicitly universally quantified. So, essentially what this is equivalent to saying for all x for all y $p(x)$ and $p(y)$. And this have standardized the variables apart is equivalent to saying for all u for all v $\neg p(u)$ and $\neg p(v)$. Now, this if you take this set of clauses has to be unsatisfiable you cannot have a model. If, it cannot have a model and you but I still do a resolution such that I just choose one of so let us say just choose this $p(x)$ and $\neg p(u)$ to resolve one. And, that is there are certain things which are not mentioned here.

So, here let us look at this so let us take a simple example. So, here is something that I just picked it up from model logic but, we are not going to do model logic. So, essentially take a binary relation so our models let us say are some either you can think of them as some kind of graphs or some kind of just relations on sets. So, we are looking at some binary relation which is reflexive and it is Euclidian. Euclidian is the property of some model logic but let us not worry about it is possible to bring it down to just notion of relation.

So, euclidianness just says that if x, y belongs to a relation and x, z belongs to a relation then y, z also belongs to a relation so its closed front to that. Euclidian think is because, of some if they form some kind of a triangle I mean so it is the qualitative notion of a triangle in equality kind of thing so that is why the name comes otherwise the name has always no significance. Since, it is a technical name you might have to use it so basically what so what you are saying is that then, it is also symmetry and actually this is quite obvious. But, we have to look at so let us look how to go about proving this. So, this is the notion of symmetry for all x, y $p(x,y)$ implies $p(y,x)$. Remember that it does not matter if I replace this arrow by a double arrow by-conditional because with this universal quantifier the two things are equivalent mean.

So, this is an example of two different first order logic statement is being logically equivalent even though they are not like propositional tautologies and we cannot in propositions you cannot easily replace this arrow by the by-condition. But, the presence of this universal quantifier and the notions of validity so on and so forth make it equal but anyway it is easier to live with their just a single arrow so let us go with that.

(Refer Slide Time: 22:42)

After renaming bound variables and converting into SCNF to get the clauses

$$C_1 = \{p(x, x)\}$$
$$C_2 = \{\neg p(u, v), \neg p(u, w), p(v, w)\}$$

The mgu $\mu = \{x/u, x/w, y/v\}$ yields the required clause

$$C'_{12} = \{\neg p(x, y), p(y, x)\}$$

The slide includes the NPTEL logo in the bottom left corner and a navigation bar at the bottom with icons for Home, Back, Forward, and other controls.

So, we renamed bound variables and convert to column conjunctive normal form. But, that in this particular case this is the reflexivity property so it is just $p(x, x)$ that is it its nothing else of it. In the second case what you have therefore is that essentially am going to write it as for all u for all y . p of u, v for all u for all v p of u, v arrow p of v, u which is not of p of u, v or p of v, u which is what I get.

How it is what get is that second this is the euclidianness. So, for all uvw that is what should get for all uvw naught p of u, v comma naught p of u, w comma p of v, w that is naught u of p v naught p of u, w p of v, w .

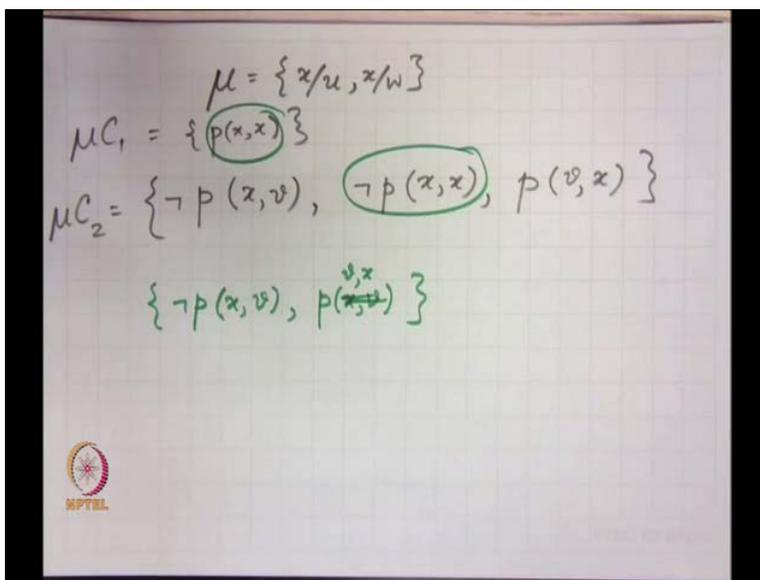
So, look now we have only one atomic symbol here and you find some most general unifier it is clear that so look at it. In this clause C_2 there are both positive and negative occurrences of the same atomic predicates. In the class C_1 there is only a positive occurrence in the predicate symbol thing which means that this anyway has to be a candidate for resolution. There is no other go and if I take positive symbols from here positive symbols of occurrences from here then, I have to take some negative occurrences of p from here. Of course, I could have taken this or could have taken this or could have taken both. But, supposing I decide to resolve these two and supposing I decide to take all of them. Then, what I can do is can actually take x for u, x for w, y for v that is one general unifier.

So, the process of unifying these two I can actually define this and that yields just this so this is equivalent to essentially for all x, y p of x, y arrow p of y what we are fine. So, I have chosen this unifier that is a strange thing you know but this mean could have just chosen x for u and x for w without a y for v could have I chosen x for u , x for w and x for v .

Student: (Refer Time: 26:41)

So, I if had chosen x for u , x for w , x for v . That would be less general than this. Supposing, and actually x for u , x for w alone would be more general than x for u , x for w , y for v . Because, I am making the least number of commitments supposing we have done that. Then, what would have happened is if, I had just taken x for u , x for w .

(Refer Slide Time: 27:32)



Then, I would have got essentially naught p x for u is so my new C_2 basically so chose my μ to be just x for u and x for w . Then, what I would have got is naught p x and v and naught p x , x p v , x where that is what would have got for. So, this would be $\mu C_2 \mu C_1$ of course continues to remain just p x , x . And, the result of resolving would have essentially gotten ridden of these two and would have given me naught p x , v , p x , v and actually this would be the most general v x actually this would be the most general. But, anyway so but so this is actually less general than that this would so the unification algorithm would have just done this. Because, this given you which is equally good which is what this is. Because, it is just I wanted to show that it is

sometimes possible to choose some unifier which is not necessarily the most general and still give you the some logical consequences. But, that is dangerous sometimes.

(Refer Slide Time: 29:36)

Refutation

Example 30.4 We could also prove symmetry in example 30.3 by a *refutation* as follows. Taking the negation of the conclusion we get

$$\begin{aligned} & \neg \phi_{p\text{-symmetry}} \\ \Leftrightarrow & \exists u, v [p(u, v) \wedge \neg p(v, u)] \\ \text{(Sko)} & p(a, b) \wedge \neg p(b, a) \\ \equiv & \{p(a, b), \neg p(b, a)\} \\ \stackrel{df \text{ } \oplus}{=} & [C_3, C_4] \end{aligned}$$

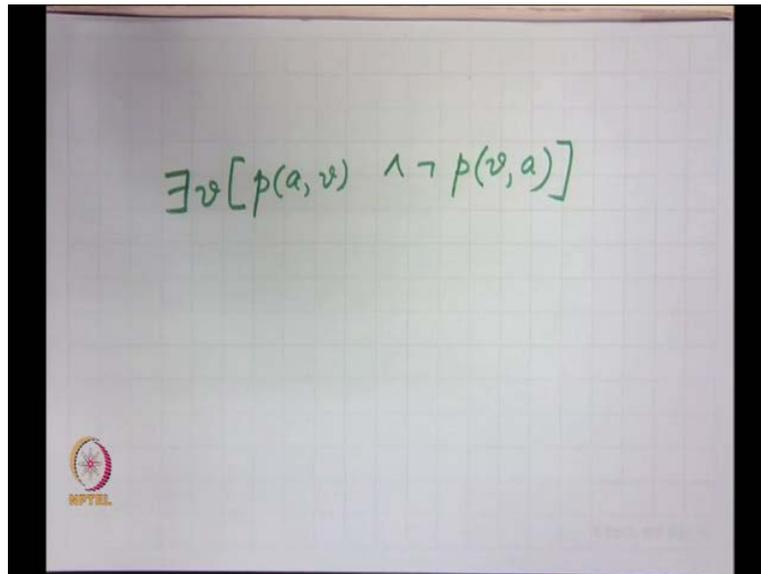
where a and b are skolem constants.

MPTEL

And, so the other way and somewhat more direct way is to do what is known as a Refutation. So, a refutation is essentially a proof by contradiction so you assume the negation of the conclusion that you are going to take. So, in this we let us do the same example so we have the same thing that this binary relation is reflexive and Euclidian. And, you have to prove symmetry and this is the formula this is the representation of the symmetry which of course whose negation will convert all the universal quantifiers to existential quantifiers.

So, which means you will get there exists u, v p of u, v and naught of p v, u . Which, on solemnization means that I will replace first one existential quantifier by a new function symbol which is a function of all the universal quantifiers which preceded it of course there are no universal quantifiers as precede there exists so, what you get is a constant a . And, having replaced that by the constant a . What you would have got therefore is...

(Refer Slide Time: 30:55)



There exists v p a , v where a is the constant. And, $\neg p$ v , a here and further skolemization would have only replaced this v by a new constant because there are no universal quantifiers before it and that is the constant b . So, basically you will get this p a , v and $\neg p$ v , a . p a , b and $\neg p$ b , a is actually two clauses so which I will call $C3$ and $C4$. Now, what you can do is you can actually just add this $C3$ and $C4$ to those two clauses $C1$ and $C2$. Make sure there are variables that anyway have been standardized apart and go through and apply the resolution.

(Refer Slide Time: 32:06)

$$\frac{\{C_1 = \{p(x, x)\}, C_2 = \{\neg p(u, v), \neg p(u, w), p(v, w)\}, C_3 = \{p(a, b)\}, C_4 = \{\neg p(b, a)\}\}}{\{C_{12} = \{\neg p(x, y), p(y, x)\}, C_3 = \{p(a, b)\}, C_4 = \{\neg p(b, a)\}\}} \mu = \{x/u, x/w, y/v\}$$

$$\frac{\{C_{12} = \{\neg p(x, y), p(y, x)\}, C_3 = \{p(a, b)\}, C_4 = \{\neg p(b, a)\}\}}{\{C_3 = \{p(a, b)\}, C_4 = \{\neg p(b, a)\}\}} \mu' = \{a/x, b/y\}$$

Alternatively a proof starting with the clauses C_3 and C_4 works too.

$$\frac{\{C_1 = \{p(x, x)\}, C_2 = \{\neg p(u, v), \neg p(u, w), p(v, w)\}, C_3 = \{p(a, b)\}, C_4 = \{\neg p(b, a)\}\}}{\{C_1 = \{p(x, x)\}, C_3 = \{p(a, b)\}, C_{24} = \{\neg p(u, b), \neg p(u, a)\}\}} \theta_1 = \{b/v, a/w\}$$

$$\frac{\{C_1 = \{p(x, x)\}, C_3 = \{p(a, b)\}, C_{24} = \{\neg p(u, b), \neg p(u, a)\}\}}{\{C_1 = \{p(x, x)\}, C_{234} = \{\neg p(a, a)\}\}} \theta_2 = \{a/u\}$$

$$\frac{\{C_1 = \{p(x, x)\}, C_{234} = \{\neg p(a, a)\}\}}{\{\}} \theta_3 = \{a/x\}$$

Exercise 30.1

- Are there any other unifiers by which symmetry may be proved in example 30.3.
- Try a refutation proof using $v = \{x/u, x/w, x/v\}$ as the first unifier in example 30.4

MPTEL

So, when you do that of you get this here I have given the most some general unifier is not necessarily the most general but that is by the way this y for v can be removed off and does not matter and actually a unification algorithm will give you only x for u and x for w . And now, what will happen is that that you can do you can follow essentially the same proof as this and but now what the refutation gives you is it gives you an empty clause after the another unification. Where, you have to in order to be able to unify this $p(a, b)$ or $\neg p(b, a)$ with this $\neg p(x, y)$ and $p(y, x)$ what you have to do is you need to give some you have to some notice that its completely symmetric. So, I could have either chosen a for x and b for y or I could have chosen a for y and b for x it is completely symmetric.

But, depending on which one you chose you will be resolving against two different clauses. But, the effect is that you would have derived an empty clause it does not matter whatever clauses are hanging around but you would have derived an empty clause. And, the fact that you can derive an empty clause essentially means that you proved a contradiction. But, normally any refutation procedure would not actually follow this path. So, the path here follow is that of this previous proof it would actually start with this constants so that you try to get your contradiction as early as possible.

So, here is a first theta one which is just does b for v and an a for w and this it resolves this p v, w to against this clause naught p of v, a does this substitution and you get this whole thing. Then, you got this p a, b and you got this naught p u, b naught p u, a. So, one possibility is to just do an a for u and get this p of x, x and naught p of a which of course derives anticlause with an a for x. And, so there are it is possible to play around with various kinds of unifiers some of them will give you the results. But, if there are not most general that is the problem there might be a problem with what you consider to be logical consequence.

So, that logical consequence often is guaranteed to hold only if you take the most general unifier. In this particular case it was lucky with this y for v because it was just the renaming of a variable you did not matter. But, as you can see here this example clearly shows that if you did not take the most general unifier if you did not unifier as many as literals as possible the goal should not be to eliminate as many clauses as many as literals as possible. But, the goal should be to get the most general unifier and eliminate all that unifier can eliminate. So, and once you do that you can actually you will actually get it a fairly automated and deterministic result. So, this is the alternative proof with the clauses C3 and C4.

(Refer Slide Time: 36:04)

Example 30.5 Consider the set $S = \{C_1, C_2\}$ where

$$C_1 = \{\neg q(x, y), \neg q(y, z), q(x, z)\}$$

$$\equiv \forall x, y, z [(q(x, y) \wedge q(y, z)) \rightarrow q(x, z)]$$

which represents transitivity and

$$C_2 = \{\neg q(u, v), q(v, u)\}$$

$$\equiv \forall u, v [q(u, v) \rightarrow q(v, u)]$$

which represents symmetry.

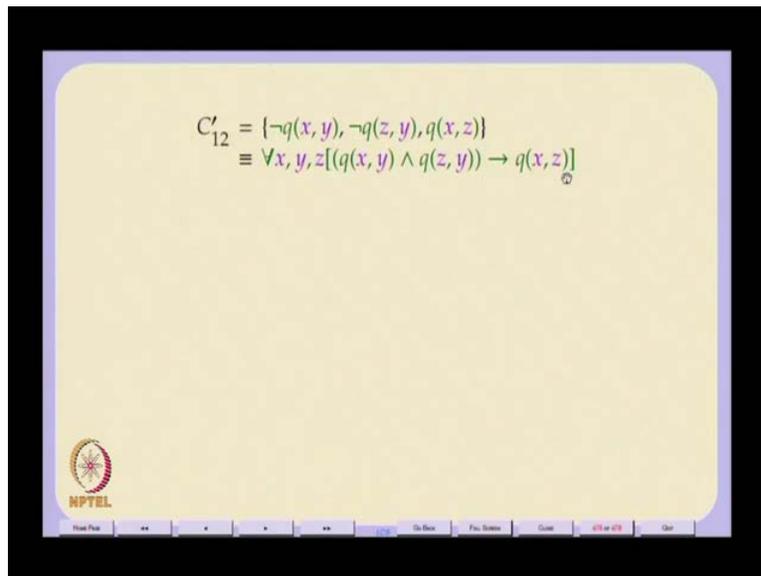
A logical consequence of these two properties is the property derived below.

$$C_1 = \{\neg q(x, y), \neg q(y, z), q(x, z)\}, \{\neg q(u, v), q(v, u)\} = C_2$$

$$\frac{\{\neg q(x, y), q(x, z), \neg q(z, y)\}}{\{\neg q(x, y), q(x, z), \neg q(z, y)\} = C'_{12}} \mu = \{z/u, y/v\}$$


Let us just do another example. So, here is a set which essentially represents this is the example that pointed out last time which we did not do. So, if you have transitivity and symmetry of a relation.

(Refer Slide Time: 36:30)



Then, you have you can derive this. So, let us take a predicate q let us just go about it systematically this one is the transitivity property. Which, says that for all x, y, z . $q(x, y)$ and $q(y, z)$ then $q(x, z)$ which in clausal form is these three literals. $\neg q(x, y)$ $\neg q(y, z)$ $q(x, z)$. You take symmetry we have already seen symmetry and so of course they have to standardize variables apart. So, I take two new variables u and v and I get $\neg q(u, v)$ and $q(v, u)$. A logical consequence of these two is something which says for all x, y, z $q(x, y)$ and $q(x, z)$ then $q(y, z)$. So, there is a permutation of the bound variables so that is why this is. So, of course what you do is you just find the most general unifier one possible way is to actually take z for u and y for v and resolve against resolve against this substitution. And I did not try to resolve I did not try to eliminate anymore variables because, I did not try to eliminate any more clauses. I just took one simple most general unifier and the result is this C'_{12} which essentially is this logical concept.

So, there are some essentially certain settle issues to be resolved one is how to make resolution completely deterministic part of the way of making resolution completely deterministic to just limit yourself to the most general unifier so that is crucial in some sense.

But, then there is this question of, Are you guaranteed whatever most general unifier you get is it guaranteed that that you are refutation procedure will always work? By the way logical consequence is a slightly non deterministic process refutation is more deterministic. So, you are most general unification and refutation to whether give you a more deterministic way of approaching the problem of proof. And, of course there is a question of is it complete, Can you actually prove everything that you can prove in any other complete system?

Whereas, on the other had compared to all these other systems like Hilbert style systems and natural deductions systems and so on. This, resolution system has a, has a potential to become extremely deterministic. And, therefore and you do not and there is of course the problem that with all theorem proving procedures is that they the reasoning mechanisms becomes counter intuitive. I mean it is not a it is not easy to reconcile those procedures with your normal methods of reasoning that there is that problem. But, besides that essentially it gives you a fairly deterministic method which can be therefore be affirmed.

So, instead of an interactive theorem prover, where the user supplies essential intuition in proving the theorem that is what would have happen. Supposing, you try to you take a Hilbert style proof system or a natural deduction proof system. Every rule in it is decidable that means given the patterns of the hypothesis. And, the pattern of the conclusion it is possible by an algorithm to decide whether it is a correct application of the rule or not.

If, you have a finite set of rules you can go further given any two patterns a given any a set of patterns and another pattern you can determine by an algorithm exhaustively checking out all the rules patterns. Which, rule was applied in order to obtain it, or whether that an application of rule at all or not you can give a deterministic answer.

But, you cannot proceed to do the proof in a deterministic fashion for example you cannot run a theorem prover to just go and find a proof which, you can do with something like resolution. So, what resolution what any resolution prover would do therefore is you give it set of formal f to prove to you give it some logical consequence to prove. So, essentially it will do a negation it

will do as column conjunctive normal form it will do standardizing apart all those are simple algorithms. Which, can be done and then it will actually rename all the variables knew it will generate temporary names for all the variables maintain a table of correspondence. And, then it will run the unification and it is essentially called unification. Which, will give it some unifier and it will keep resolving against them. So, actually what it will do is it will do a, what is known as the linear resolutions which is not very important. But, basically what is important is to be able to see that there exists positive and negative occurrences of some atom in two different clauses. And, that they are unifier so basically you can exhaustively look at all that.

And, you can find a unifier do the substitution and then continue with to find positive and negative occurrences of this of some other atom maybe and, again do the resolution. And, so this process can be done in a completely batch more without any user intervention. Whereas, any proof which involves a natural deduction or a Hilbert style system would require user interaction. Because, the user will have to provide an essential proof technique which is not deterministic. For, example if you is we know that in the Hilbert style system one of the one of the exercises is somewhere actually showed that. For, every proof which is direct that means you assumed only all the assumptions and tried to get the conclusion from the assumptions. By for every direct proof there is also a proof by contradiction which is equally correct. That, is as far as the larger proof techniques are concerned. But, between these two there are a huge number of variations possible can start off with the direct proof. Somewhere, in the middle make an assumption to derive a small contradiction. And, then I continue with the direct proof or I take make more and more define essentially within that proof I define a series of lemmas which are all proven by contradiction let us say.

So, there are all these variations which for any theorem prover would require guidance it cannot do it by itself. Whereas, this resolution and actually the tableau method mean we have to do the tableau method also for first tautology. They, have a genuine potential for essentially feeding the theorem to be proved to the system in a batch mode just give it to the system. And, let it crunch it out it will crunch it out with an answer finally either it could do it or it cannot do it. And, that is essentially automatic really automatic theorem proving otherwise most theorem provers which use other systems are not automatic. There are interactive theorem provers so at every now and then the user has to give some essential intuition to the system for it can prove.