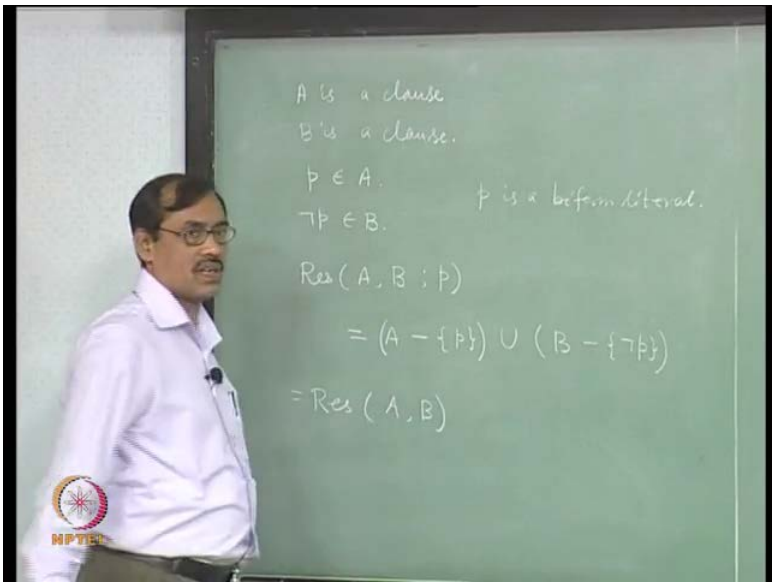
Mathematical Logic
Prof. Arindama Singh
Department of Mathematics
Indian Institute of Technology, Madras

Lecture - 12
Resolution

We have introduced resolution rule, have we? That if you take any two clauses A and B and there is a lateral p in A and its negation not p is in B, then the resolution will be deleting those two laterals and joining all the remaining together.

(Refer Slide Time: 00:36)



We just introduce that. Say, A is a clause; once we say a clause, it means it is a disjunctive clause, in this scenario; and then it is also written as a set. Go on writing clauses as ors of literals, any time we can think this as a set of literals; we will not write them explicitly, any one of the notation we will be using. So, A is a clause, B is also a clause, and we have a lateral p which is in A. That means, A looks like, some laterals, and p, some other laterals. Then, we also take not p belongs to B; we add just a comment, when you say p, it does not mean it is a variable, a propositional variable, it can also be a literal.

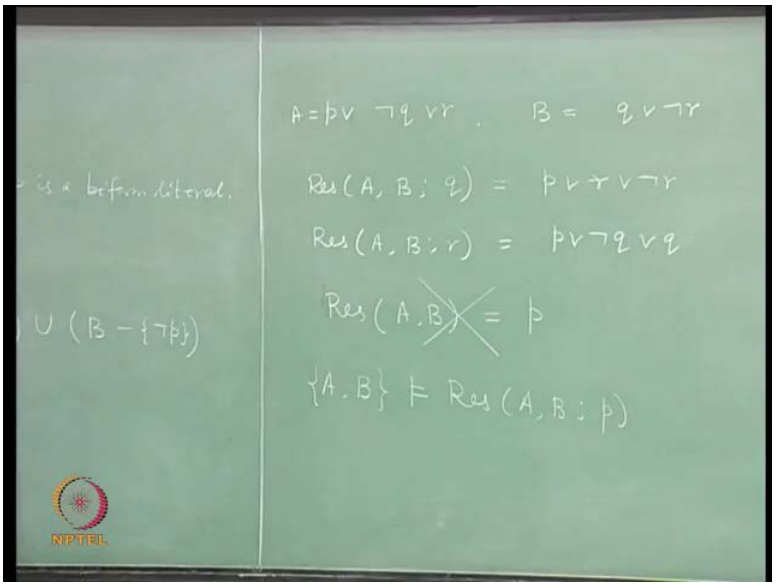
Suppose it is a lateral, it can look like not of q, in that case not p will be not of not of q, but it is written as q simply; double negation is assumed throughout. In that sense, when you say p belongs to A, not p belongs to B, it does not give a constraint that it has to be

the not of a propositional variable, in that form, and this has to be a propositional variable; it is not that. With this understanding, we are giving the notation here.

Then we say that resolution of A and B, these two clauses with respect to this lateral p, sometimes if it is a variable, we say with respect to the variable p. So, p is called a biform variable, here, if p belongs to A, not p belongs to B, it is called a biform variable or a biform literal. Biform literal is more appropriate; but once you know p is a variable, you also say p is a biform variable, so you say; give a comment, p is a biform literal.

Define resolution of this as A minus p, take out p from A, then similarly take out not p from B and then take their union. Here, union means what? We are getting a new clause where some literals from this and the literals from these are there. Once, as I said it will be of all those things, so it is equivalent to deleting p, deleting not p, and add them together. This, we will be calling as resolution of A and B with respect to the biform literal p. Sometimes, you delete the biform literal. If it is very clear what this p is, we just write resolution A, B as an informal notation, so also we write this as resolution of A, B if p is clear from the context.

(Refer Slide Time: 03:41)

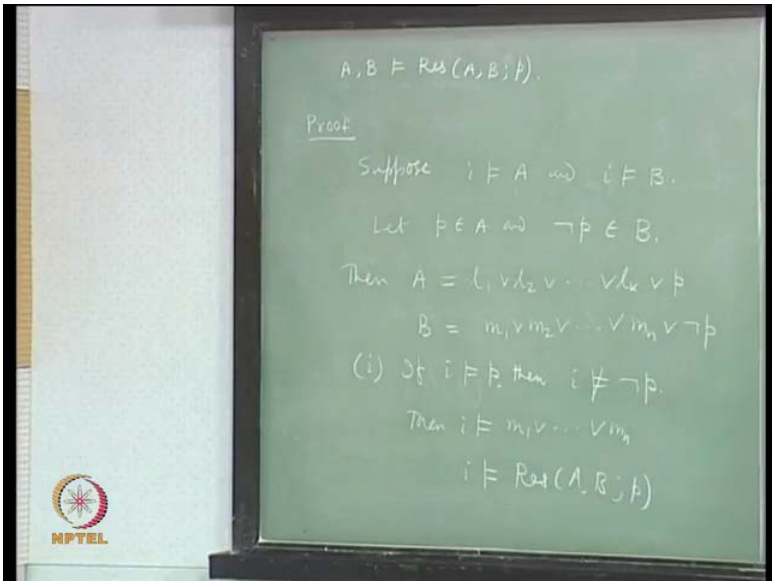


Let us see an example. Say, I have p or not q or r this is my A, and I have B as say q or not r, now it shows me a biform literal to be q, also, not q, any one of them you can say, a biform literal is this. Then what will be the resolution? We will write resolution A, B with respect to the biform literal q. Now, look at the definition. Here you do not say B

minus q is not permitted. Since, it is written in the B minus not p form there our convention works; once you say it is with respect to q or with respect to not q, they are the same, because of the convention, because not not q is taken as q itself.

This is equal to, you delete this not q, delete this q, so you get p of r. You could have also chosen r as the biform literal, right? Because r is there, not r is also in the other, so you chose that A, B, r. Now, once we choose r and not r here, add the other things as p or not q or q; then should I write this also? This could have been any one of these two. If I do not fix the context, so let be p. Now, I am just looking at it formally, forget the semantics, now I have this clause, so the procedure says when it is in one form, the other not of that from, the other. You choose this, choose this, I get this. Choose r, not r, I, why not q, not q, r, not r, both, then I get p, but that is not permitted. That is what resolution does not define. You have to fix a literal and then do it. You cannot fix both the literals at the same time; this is not permitted. That is why this notation really helps; which literal you are fixing. Accordingly, it will be taken as the right resolution; maybe you have to keep this in mind that you fix one literal and then write resolution of A, B; as it will have no meaning.

(Refer Slide Time: 07:52)

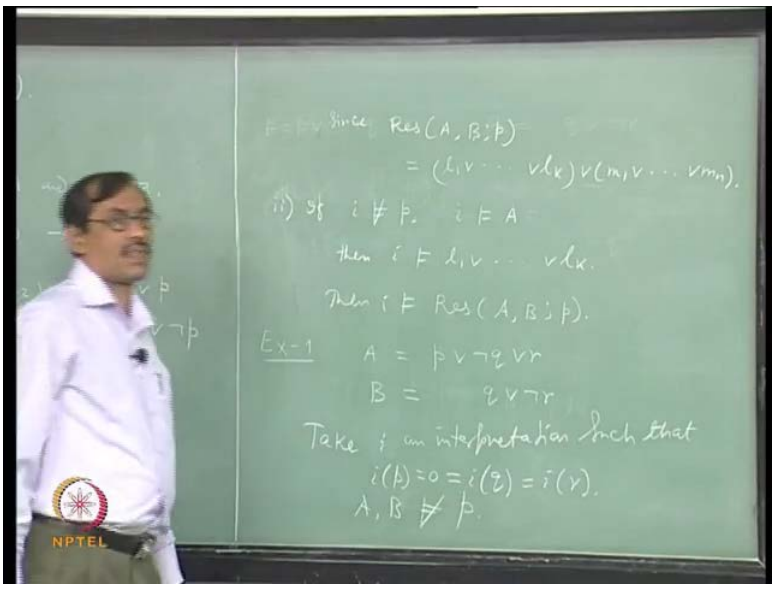


We will see what is the reason. Let us see the reason, why you are doing it? The reason is, if you take A and B as a set of two propositions, of two propositions. It will really entail resolution of A, B with respect to whatever literal you have chosen. This

entailment is really semantic entailment. It says, chose this proposition A, as propositions you are thinking, this is a clause, fine, that is also a proposition. If you assume A and B, you can deduce the resolution of A, B, p as this consequences, as a direct consequence, that is what it says; we can prove it, it is not difficult to prove, let us see.

To prove this what we do, let us rewrite it here, but our p you choose. Now, suppose I will prove it, if you have not forgotten, you start with a model of the set of propositions sigma so that, that is also model of w. We start with, suppose i is a model of A and i is also a model of B, since this is possible. That means i of, p as a literal, which is occurring in A and not p is occurring in b, or otherwise you could have started with not p belongs to A, p belongs to B; that is similar, so that, p be a literal in A and not p be in B. Now, then how will A look like? It is a clause where p is there, so p and some other literals. Let us write it as, say, l1 or l2 or lk or p. Similarly, B might look like some m1 or m2 where all literals are, mn or not p.

(Refer Slide Time: 10:55)



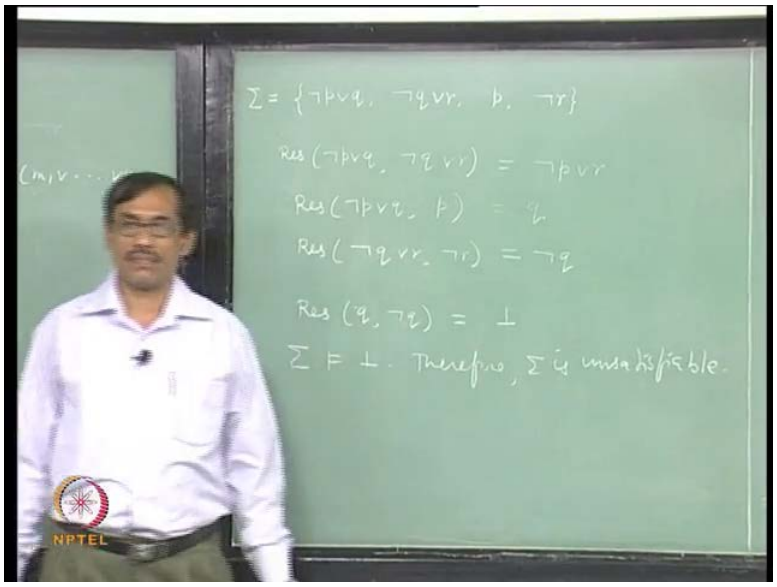
Now i is a model of A; we do not know of what it satisfies, it might satisfy any one of them only that is, also possible, right? So, let us bring out two possibilities: this i satisfies p or it does not satisfies p. There are two cases, so let us take first case. If i satisfies p, then what happens, i does not satisfy not p. Look at B, i does not satisfy not p, but i satisfies B, therefore, i satisfies the other one. Then i satisfies m1 or mn. In that case i satisfies resolution A, B; because resolution A, B, p equal to l1 to lk or m1 to mn, one of

them is satisfied, so the whole thing is satisfied. Since resolution of A, B, p is equal to l1or lk or m1 or mn, so that gives case one.

Next, case two is: i does not satisfy p, then i satisfies, we do not need i satisfies, let us go there itself. If i does not satisfy p, then we look at i satisfies A. What do we get? i satisfies l1 or lk, is it clear? i does not satisfy this, it is evaluated to 0, but the literal p is evaluated to 1. Therefore, whatever remains, that is evaluated 1. Then job is over, then i satisfies. That is the end of the proof. Therefore, resolution A, B, p is a consequence of the sets A, B. So, if you have a cnf having two clauses A and B, you take their resolution, it is really implied by the cnf.

Let us comeback to our problem. What happens if you cancel both the literals? In that case, what was happening, you have A as p or not q or r B as p or q or not r, there was no p. q or not r. In that case whether I take any interpretation which satisfies A which satisfies B, will that satisfy p? And I would have to show that. No, it is not necessary. So, the counter example should start with taking p equal to 0, so take i, an interpretation, such that i of p equal to 0, but I have to satisfy these.

(Refer Slide Time: 15:10)



Take q as 0 that is satisfied, r? I take r as 0, now what happens? i satisfies A because not q is 1, all the others, I forget, I do not need to see i satisfies B because not r is 1, but i does not satisfies p, so I see that A, B does not entail p. That means what we have done earlier should not be allowed, it gives a hint, because we want this to be preserved A, B

should entail resolution of A, B. That is why we have defined resolution in such a way that two variables, if are there p not p, q not q, they cannot be canceled simultaneously, only one of them can be canceled.

Let us see an example. How to proceed? Here, to go for the resolutions, what resolutions can give us. Say, resolution of not p or q, not q or r, so biform variable is q, not q, that is the biform literal. Here, I fix q or not q, so the other one remains not p, r. I can see that it is not p or r, to be specific, we should have written here semicolon q, resolution of not p or q with p, now p is the biform of literal. That gives q, resolution of not p or q with not r, I cannot tell, it will not give anything.

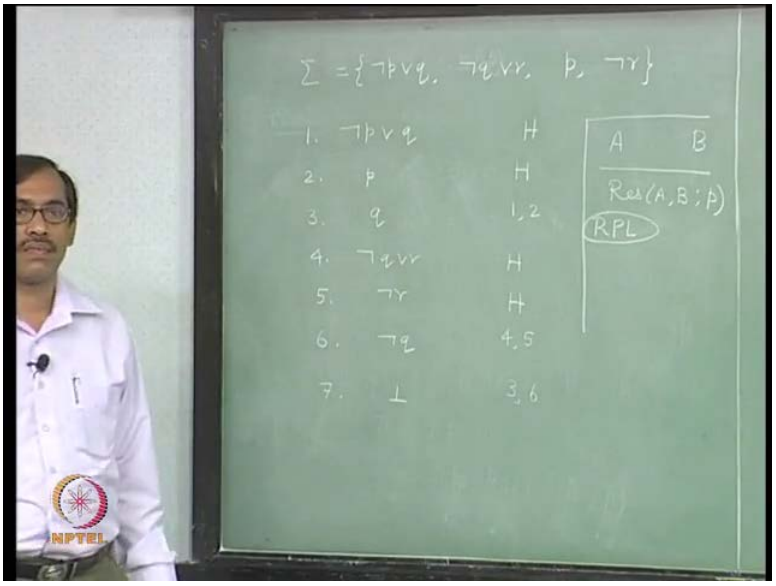
It cannot apply, because in not r there is only one literal not r and r is not here, so I cannot do anything. Now everything else, I can do with not q? Nothing, all are over. Let me start with the next one: not q, r, with that I do not have to go back. Let it be A, B or B, A they will be the same, by the definition, so we go further; p, I cannot take not of s, so r, not r goes, I get not q; and p, not r, nothing can come. I get this. It says that if this is my set sigma, then sigma entails not p or r, because of that result sigma also entails not q, sigma entails q. All these are consequences from the same set of premises, so once they are consequences, I can take further resolutions; I can apply resolution again, there will also be consequences.

We choose these once, so I get, that is my second chance, so resolution of not p or r with what? q or not q, nothing can be taken, so only these two can be taken, let us take that, with q, not q, each of these is the crucial thing, entails empty set. That is, according to us, but this says that sigma is unsatisfiable, because sigma entails bottom, so we see that sigma entails bottom. Therefore, sigma is unsatisfiable. So, we can prove unsatisfiability by resolution, now target is to derive bottom, somehow; you do not have to go everything like this. In fact, this was useless, this was never used, so you can know the target; possibly some simplification can be done, then you can write it as a proof.

The proof means, I will introduce one as my premise, not p or q, then I would have introduced my premise as p, then I conclude my resolution q. Next line, I introduce not q or, that is already there, now I introduce this as a premise, next I introduce as not r as a premise. Then, I find not q; then from q and not q, I take resolution, and get bottom, any doubt?

See there can be one clause which can be used many times, it does not limit us anyway, last step of resolution of q and not q, so add two conclusions, this, add two conclusions from the earlier premises. By our result, it says sigma entails q, sigma also entails not q, so whatever they entail that is also entailed by sigma, so I can use the resolvents. They are called the resolvents; resolution of A, B is called a resolvent of A and B. Resolvents can be further reused, so if you write it as a proof in which you are using the same line numbers.

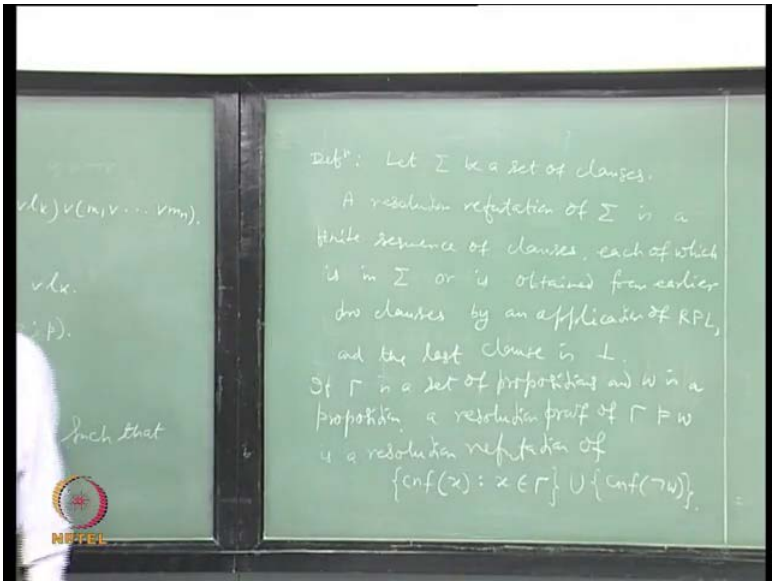
(Refer Slide Time: 20:37)



Let us rewrite it. I have sigma equal to not p or q, next not q or r, p, not r. So you have to start, to prove this way, my first is not p or q, which is a premise, so we will write H, as a hypothesis. You can write P also, for premise, this already is the second line. I will introduce p, this is also hypothesis, now third, I will derive from this q by using 1 and 2, and there is only one rule here, right, the rule of taking resolution. You do not have to write anything: pm, t anything. I have to write, there is only one rule, so why to write? I will just omit it, give the line numbers, but what is the rule? f you have A, you have B, then from this you deduce their resolvent. This rule. Let us give it a name: it is called resolution for propositional logic. The rule RPL. I can do this, so write it. p is a biform literal; that means p should be in A, its negation should be in B, or not p is in A, p is in B. If that happens, in that case only, here you should have written RPL anyhow, that is the only one.

We are omitting. Next, what we do? You can see what we have done. There, we have to derive not q from these two. So, introduce them as hypotheses: not q or r, that is a hypothesis, and fifth one, I take not r, which is a hypothesis, sixth, from these two I conclude not q, as a resolution. It is going smoothly. Next, I take three and six and get bottom, resolution from three and six. Such a three column style is called a resolution refutation of sigma.

(Refer Slide Time: 23:19)



Let sigma be a set of clauses. A resolution refutation of sigma is a finite sequence of clauses each of which is in sigma or is obtained from earlier two clauses by an application of that one rule of resolution. The last clause is bottom. That is what we have done here. In this refutation, it is a sequence of clauses, this is the sequence of clause, so each one is either in sigma, which I have written as H, or is obtained from earlier two.

This one is obtained from these two. This is obtained from these two, this is obtained from these two; and the last one should be bottom. So once we have a resolution refutation of sigma, this would tell us that sigma is unsatisfiable, because of that result. Again, that is to be proved by induction and the length of this finite sequence; because one step you go, it preserves entailment, many steps you go, it also preserves entailment. That is what. Then you define: suppose you just generalize it a bit. If gamma is a set of propositions and w is a proposition a resolution proof of gamma entails w.

Here what is that we do? You want to show gamma entails w, yeah, you have only resolution refutation; so you should use reductio ad absurdum, add not w to this end. That is unsatisfiable, and unsatisfiability can be proved by resolution refutation, but you need cnf; so what we do, we write it: A resolution proof is a resolution refutation of what? of gamma union not w, but I cannot write gamma union not w. In the resolution refutation, I let this to be a set of clauses; so gamma union not w first must be expressed as a set of clauses, then of that set, it will be a resolution refutation.

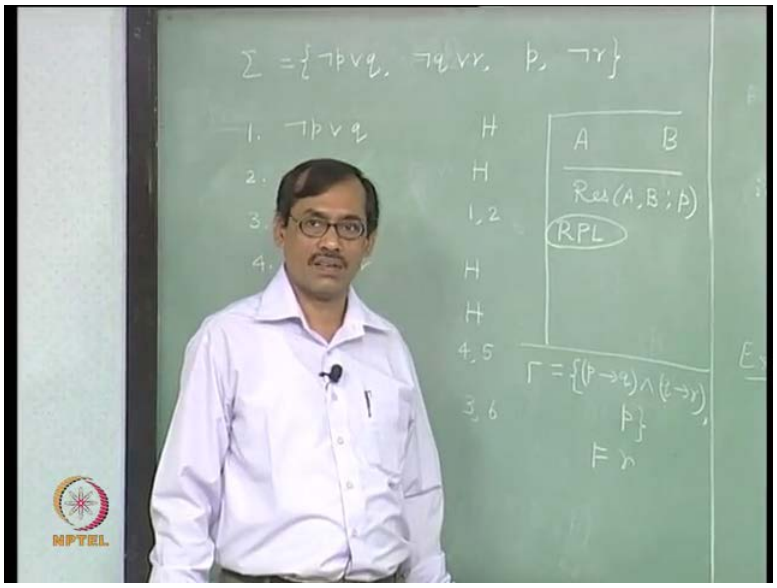
We write that as follows. It is a resolution of, resolution refutation of cnf of say x such that x belongs to gamma union cnf of not w; is it clear? It says, you take each member from gamma, convert that to cnf, that is a cnf representation. Then, take the cnf, there are really clauses, but we need clauses not this, so it should be what union of each cnf; it is not cnf as members. There should be sets and there union, so it should have been union over cnf of x, is that clear? This should be something like this, because each cnf of x i is a set of clauses.

Take all those sets of collection of, all those sets of clauses and take union over those, is that clear? If it is countable, it should be or it is finite. Then, you can build in a better way, you know, right? cnf of x i, and then this itself is a clause, so forget this and then take union over this? Right? Union of all these things, cnf of x i. So, cnf of x1 union cnf of x2 union cnf of x3, and so on. That gives you the set gamma, so that we write this way here, it is just a notation.

Read it carefully. It says, you take an element of gamma, convert it to cnf, now take the clauses, now take another x, another element of gamma, convert it to cnf, take all these clauses, with those together, continue like this. So, get a set of clauses which is obtained from each member of gamma, now that becomes a complete set, full set, with that you add cnf of not w. Then consider resolution of that, resolution refutation of that. That means, refutation of, is already in built in a definition, so that means.

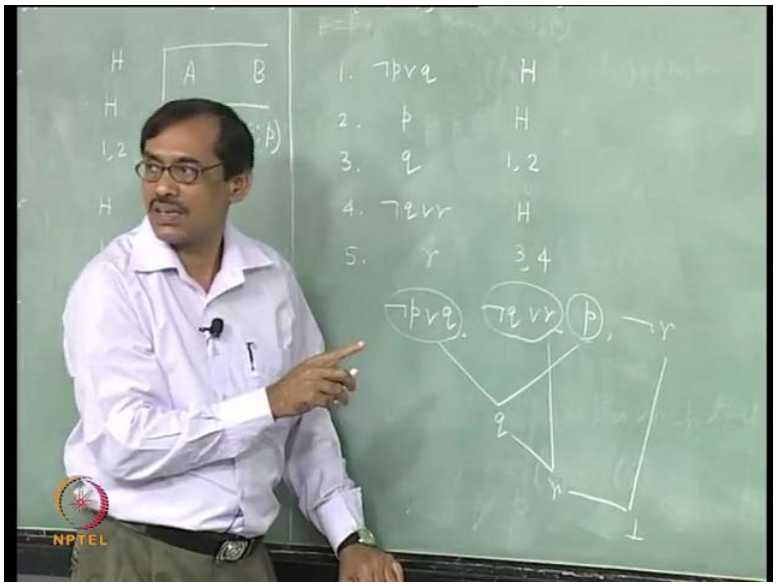
Suppose we revisit this example. I would have given this sigma in a different way. I would say that I will take my set gamma equal to, this gives me p implies q and q implies r, this is my one proposition there, another proposition is p. That entails r, then what I do, I take cnf of p which is p itself, it will be p, then another set bracket.

(Refer Slide Time: 29:59)



Now, cnf of this will be this, plus this, plus and another bracket there, union, so it will be p another bracket then cnf of not r, that will also be union with this, that will give not r. All that you have to do is, starting from this you will first convert it to cnf and then get this set, get a resolution refutation of this; in fact to use that result. You need this resolution refutation. you can always start from this directly and derive r that is also allowed, because but to make it uniform we make this way: reductio ad absurdum. Then, always you have one fixed target bottom, it is just to make it uniform, there is no need of course, you can always think of this and derive r, for example.

(Refer Slide Time: 31:42)



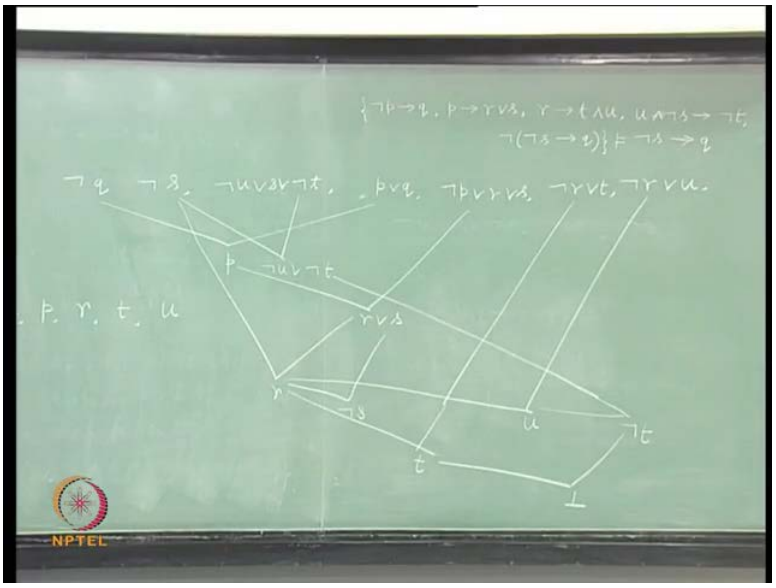
Here, what should we do? We could have started with this set, not p or q, not q or r, p entails r. This is what it is, right, p is kept as p, p implies not q and q implies r is written as not p or q, set, not q or r, two clauses and p is there. So, entails r. Then resolution would proceed. Let us take these two: not p or q then 2, I take 2, 3, I get q; then I introduce not q or r, and this is then, fifth will be r. That could always be done. It is, it is equivalent. Why is it equivalent? Suppose in gamma you have x1 or x2, xn. Now, you want gamma entails w, now gamma entails w means x1 and x2 and x3 and x4 and xn implies not w is unsatisfiable, sorry, implies w is valid, which will say x1 and up to xn and not w is unsatisfiable.

So, when you convert to cnf each one, you convert each, instead of converting the totality, because? There all ends. So, that is what we are taking, cnf of x1, cnf of x2, cnf of x3, so, one should take cnf of x1, that will look like 1, clause c1 and c2 and c3 an cn. That is same thing as writing a set of c1 comma c2, cn. Do you see what is happening? Clear? Here also you could have done some simplifications to make it clear, how we are proceeding. Instead of this style of proof you could have written as a graph. What you do, you start with that set p and then you take p with this resolution as q, and then with q and this, you get resolution as r.

You see it as a directed acyclic graph, this happens to be a tree here, but always you may not get a tree. There will be crossings, so you get a directed acyclic graph. Let us see this resolution refutation. How do you see it as a directed acyclic graph? Almost the same way with not r only you have to proceed. We have another premise here as not r, when you take r, not r? That gives not r, same thing, all the things are there, only not r is not there.

I have all these premises now. From there I get bottom, yes, and that to, one literal and the other one has to be negation of it, yes; thatis how clauses taken as a node, is the level of that node; see here, this is one node, this is another node. From this two nodes I get one here. Suppose you have p or s. Here p or s, then you would get q or r, s here. That is all. You are not getting two, you are getting one, they are odded together, we are taking only A minus p union B minus not p, so that itself is a clause, so only a single clause, there are not two here.

(Refer Slide Time: 37:36)



What about this? What does it tell us? To do here, this is a set, this is set of premises. So, first thing it asks, is to convert everything that to cnf and put them in as a set of clauses. First, not p or q, that gives p or q, p implies r, s, which is not p or r or s, r implies t and u, so that is not r or t and u which is equivalent to not r or t or not r or u, the distribution. You have got a cnf, now as a clause set we will be taking one clause as this the other clause as not r or u. We will be writing it as not r or t, another is not r or u, that was your doubt? Then, let us see, u and not s implies not t, so that gives not r, u and not s or not t, which is not u or s or not t; so I get not u or s or not t, next not of this. This is not s and not q, that is all.

These are the two clauses, in that, in not s and not q, I have two clauses, one is not s and another is not q, so you put a comma instead of or. Next negation of these, so not of not this implies q, which is not s and not q; again I get that. What we have done is, the same thing, both the sides, it is r, so I said I do not have to do anything, one is negation, another is simply there. One should take negation of that, it will be added to this, it is same thing as there, I do not have to add anything here.

Now, this is my set, right, from which you have to get the resolution, let us start. Do you need a bigger board, let us copy it here, now what should we do? Start taking resolutions, that is what it says. It is usually easier if you start with the unit clauses, where there is a single literal then resolutions will not increase the length. If you start with others

resolution might give you a clause, sorry, bigger length, more literals. So, let us start with that, not q where there is q, p is there, so I take with that, so I get p. Now, you can start with p itself, where not s also, let us start with not s.

So, these two give not u or not t. Well now with p, I get r or s, with not s, I get r. Well, it is not a tree, because of this, because these things can be reused. In general it will be a directed acyclic graph; there is no cycle found, but it is not a tree. If you reverse, it will be a tree, from r you will get a child as from not u and not t also you get this child, that is not possible in a tree. Now, we have r; here r or s, that again gives not s; also this gives t, with r, also with this I get u, then I have not q, I have not s, I have p or not s, is there already, I need not have this, this is wasteful, already I got. Getting not s from these two is wasteful it does not help, but you can get, so many wasteful things, so what of the things I have got? not q, not s, p or t, u, not u or not t with u if I take, what will happen? This will give me not t, now I have t somewhere, t is here, so I get.

But if you rewrite in some more different order this crossings might not be there. But we are not worried about that. Which one? That is, it does not matter, not with an implied p, What is wrong if p is true? p is valid? Not that we can imply p, r and r or s, so give s, not s, from this I am getting r or s. Therefore, this you are getting it from, remove, this is not correct. Anything else? Verify everything now. We just go back, see what are wasteful, what are, not u ended bottom. It comes from not and t, now t we are getting from r, and not r or t and not t we are getting from u, so from u and not u or not t, is it correct? Yes, on u we are resolving, so you verify.

This, you are getting from not s, so that is canceled, so you get not u or not t then r or s comes from p and not means that is canceled. So, r or s, not q and p or q that gives p, so it is equivalent to telling, if you do not have it here, also it will be alright, do you see? Suppose you do not have this here. Your gamma is only up to this, fine, still it will entail, because not w will add this, is it clear? That is, the redundant premise is there.