Artificial Intelligence:

Propositional Logic: The Resolution Refutation Method

Prof. Deepak Khemani

Department of Computer Science and Engineering

Indian Institute of Technology, Madras

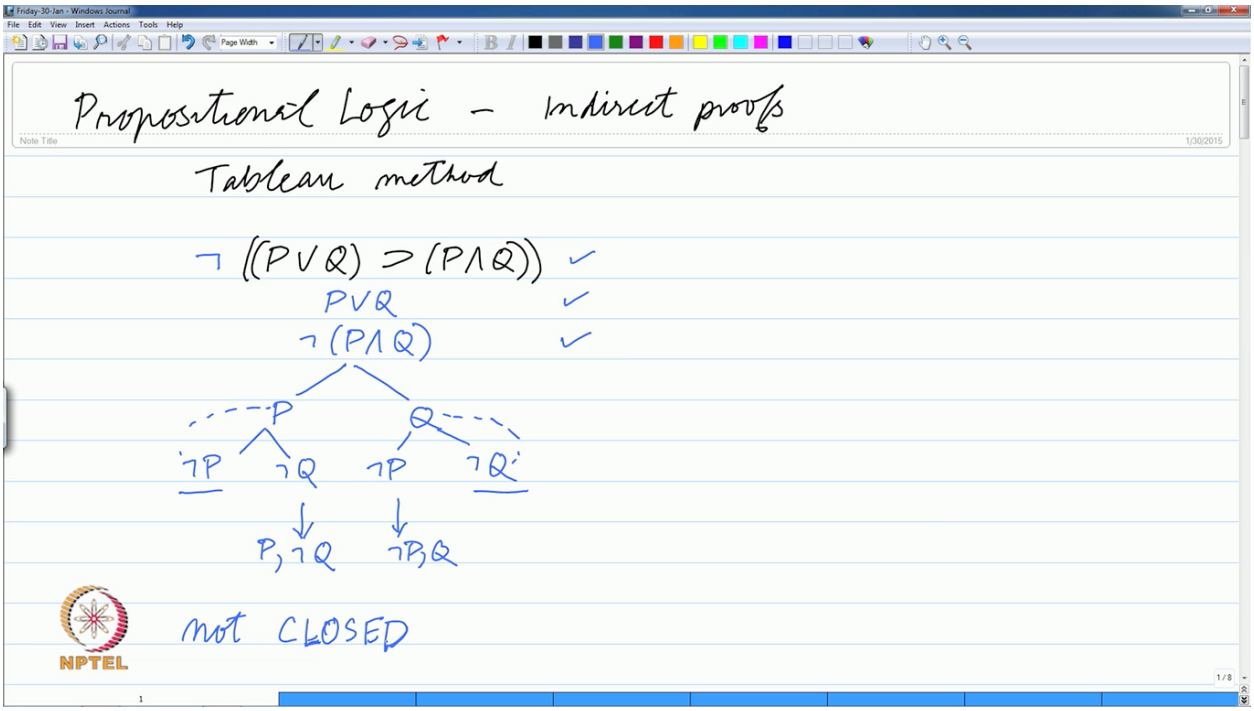
Module – 02

Lecture – 07

We have been looking at propositional logic and this is probably the last class that we will look at this and in particular we were looking at indirect proof methods. So we saw this system called tableau method just to do a very quick recap. If we take a small example so supposing we want to show a particular formula to be true so let's say that this is the formula that we are interested in it says that P or Q implies P and Q. so lets us say for argument sake that we want to show that this formula is true or valid under all valuations. So the method says that what you do is you take its negation. Both the indirect proof methods that we are going to lookat take the negation of the given formula and then we try to show something. In the case of tableau method, we try to find a satisficing valuation for the negation of the formula. And if we cannot find a satisficing valuation then we conclude that the formula cannot be negated. So let's just quickly look at this method. So we apply the rules that we had for different connectives.

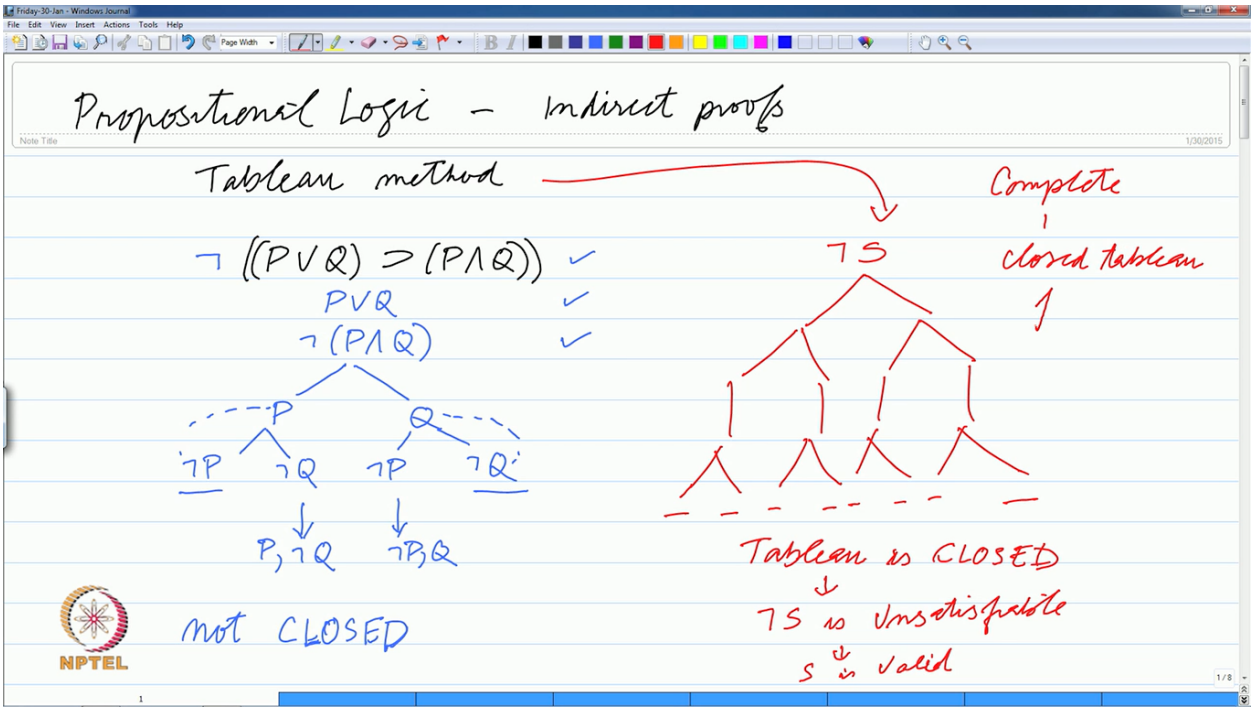
So to make this formula false we have to make the left hand side true and right hand side false. So we will just as our custom we will put a tick mark when we have dealt with this. Now we have two formulas both of them are going to introduce branching so we don't really have too much of a choice here. So let's take the first one. We introduce P or Q. so we have dealt with the second formula. Then we do the third formula which says not P or not Q. Now the way to break down the third formula we have to do it in each of the two branches this is probably the first time we are looking at it so it serves as a reminder that we have to do that as well. At this point we notice that this is closed because there is a contradiction along this path. We also see that this leaf is closed because there is a contradiction along this path. But the other paths don't have a contradiction. Further we cannot do anything anymore breaking down because everything is already at the atomic level. So we cannot refine this tree any further. So this means that we are able to find satisficing valuation two valuations one is given by this branch this says that if P is true and if not Q is true which means Q is false then we can make the original formula false. We can make its negation true and hence we can make the original formula false. Likewise, in this valuation if we have not P and if we Q then we can make it false. So we are left with a derivation which is as we say not closed

(Refer Slide Time: 4:40)



So what we need in the tableau method is given a formula sentence S we need some kind of a breakup of the formula where every branch is to be closed. Now if every branch is closed then we say that the tableau is closed. And if we started with negation of this then this means that negation of S is unsatisfiable which means that S is valid or S is a tautology. So a proof in the tableau method is derived by producing a tableau where every branch that we can see is closed. We would say that a tableau method is complete. What is the definition of complete that for every valid formula we are able to produce a closed tableau.

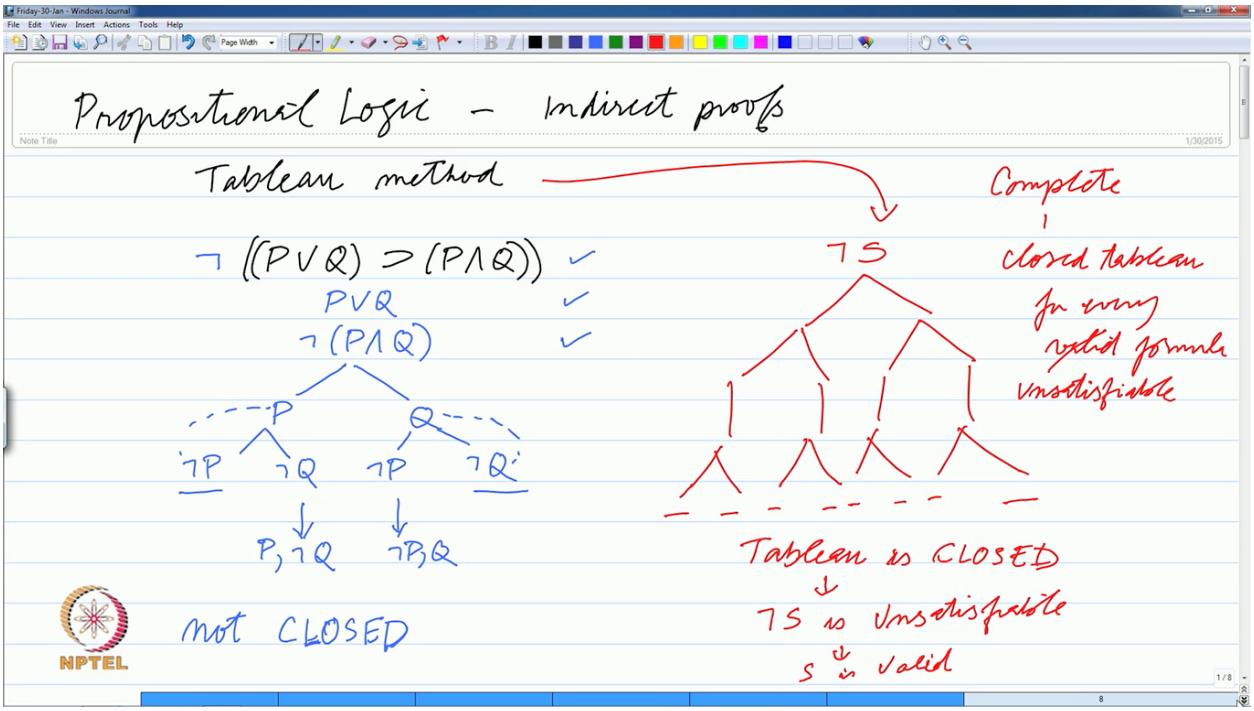
(Refer Slide Time: 6:20)



Well I have written valid formula here but you must keep in mind that when we have to negate it when we start constructing the tableau so I should may be write unsatisfiable formula. So what a choice that the algorithm will need to make here. Basically the choices are as to which of the constituent compound sentences we need to break up. And we have mentioned in the last class that if you break those sentences which do not include branches then you will have to end up doing less work. But there also another choice which we don't often talk about which is that sometimes you can close a branch quickly without going through many breaking up. If you could do that somehow if it's a large formula especially then you could have a shorter tableau which is closed.

But an important feature is that the tableau must be closed which means every branch must be closed. So you can imagine that the proof can sometimes be quite long.

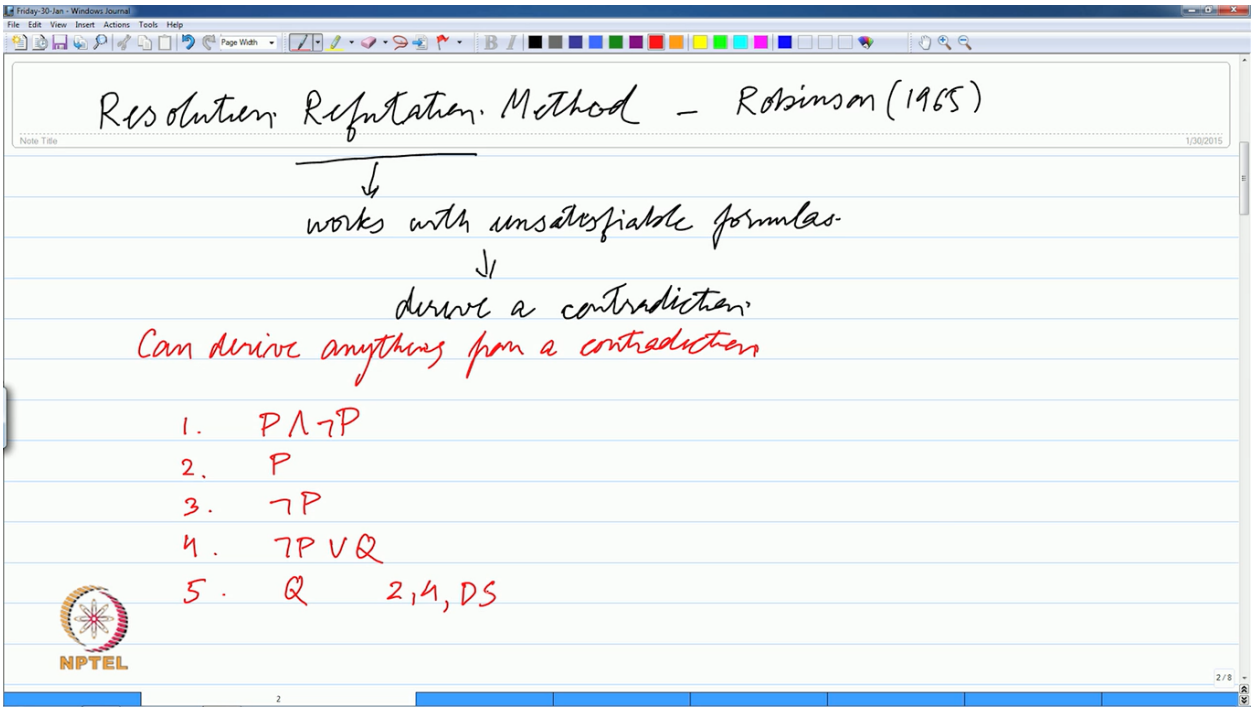
(Refer Slide Time: 7:36)



So today I want to look at a different method which is called the resolution refutation method. So as I mentioned towards the end of the last class this was invented by Robinson around 1965 or so. And since then it has been proven to be a very popular method. So like the tableau method this is also a refutation method in the sense that it works with unsatisfiable formulae. And it shows in this case that you can derive a contradiction if you start with an unsatisfiable formula then you can derive a contradiction. Which means you can refute the fact that the formula is true. Why is a contradiction bad for a logic system? The reason for that as logicians say is that if you have if your knowledge base is inconsistent or if you can derive a contradiction from your knowledge base then in fact you can derive anything.

So let's just do a small example to show this. Can derive anything from a contradiction. So let's take a simple contradiction which is just two sentence which is P and not P which obviously is a contradiction because we cannot make both P and not P true at the same time. Then by using simplification infer P we can also infer not P. you can also infer not P or Q which is addition the rule of addition which says that you can always use the disjunction to add something else and from this as we can see you can infer Q which is from 2 and 4 and disjunctive syllogism so I will just write DS here.

(Refer Slide Time: 11:01)



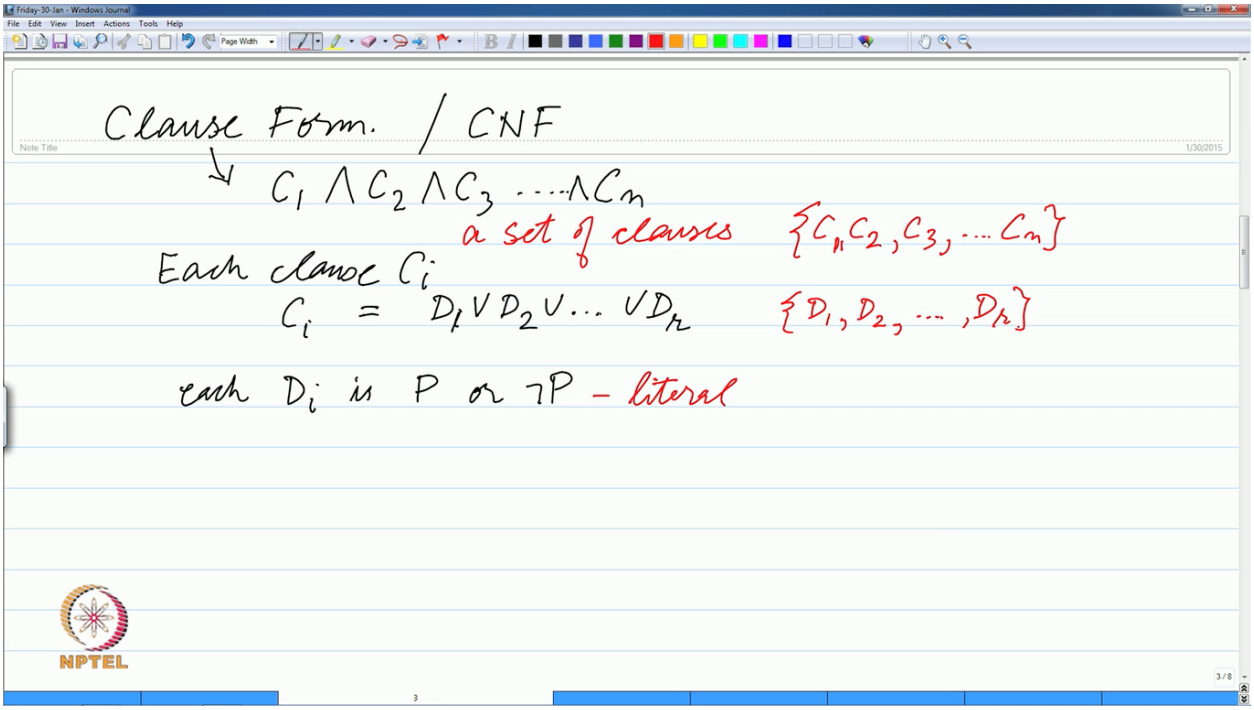
So you started with P and not P and you could produce Q. so obviously you can see that there is no connection of Q with P and not P and you can practically derive anything from a contradiction which is the reason why when we look for knowledge bases and logic systems we are interested in three properties one is the property of soundness which if you remember says that if a knowledge base derives a formula alpha then that that formula should be in KB. Then we have completeness which says that if a formula is entailed then you should be able to derive it and the third is consistency which actually is connected to soundness but we often state it separately that cannot derive both a formula alpha and its negation because then you have inconsistent knowledge base then you can virtually do anything.

Okay so the resolution starts with an unsatisfiable formula and then it shows that you can derive a contradiction we will look at that shortly how it does it. but before we do it one comment that the formulas that it can look at are in conjunctive normal form which when we move to higher order logic we will also call clause form. so I will begin by defining what is conjunctive normal form and then we will look at the method.

Or CNF which is a short form for conjunctive normal form. so a formula is in clause form if it is in the form where it is a set of clauses C1 C2 Cn. A set of clauses so I am using the term a set of clauses but obviously the formula that I have written is not in a set notation it is in the logical notation that we have in the logic language that we have. So when we write the set of clauses we sometime may just denote it as C1 comma C2 comma C3 comma Cn so we often write clause form formulae as a set. Each clause Ci can be written as D1 or D2 or Dr and each Di is an atomic formula or a negation of an atomic formula we call this a literal. So each Di is a positive literal or

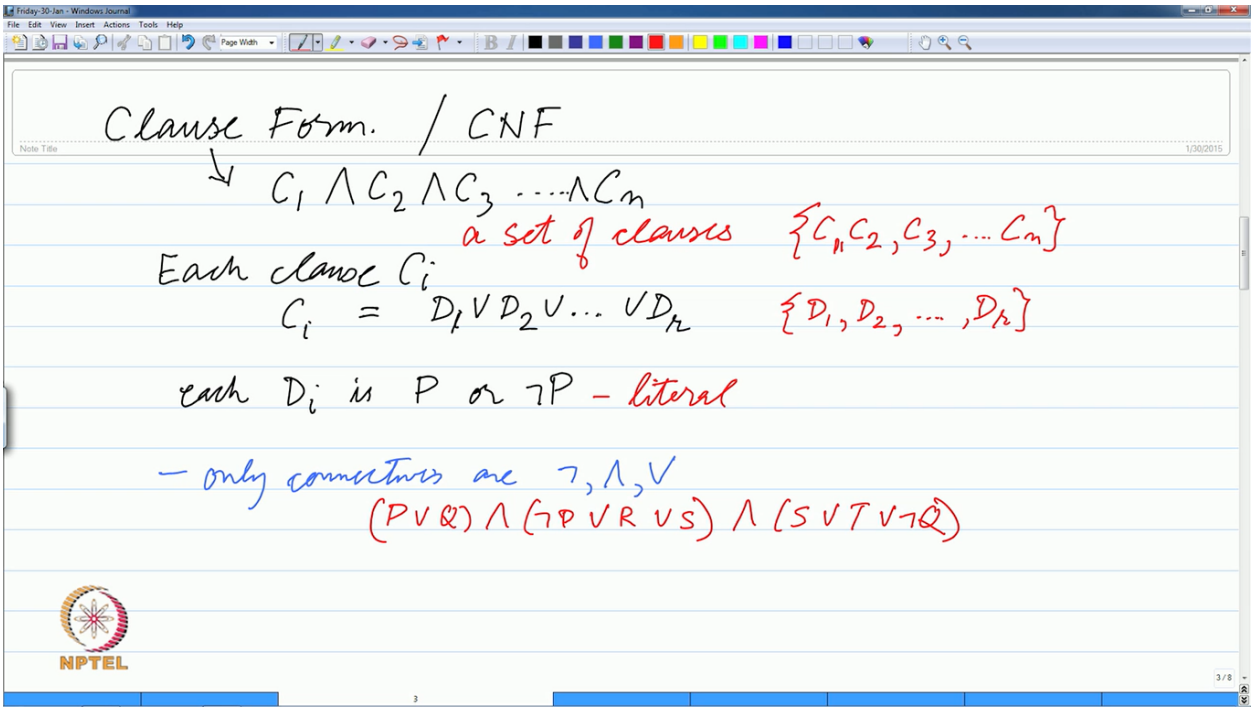
negative literal. And by literal we mean an atomic formula and like we do for clauses we may also use a set notation for the Dth clause so you can write it as D1 comma D2 comma Dr.

(Refer Slide Time: 16:33)



So what can we observe about this. That the only connectives that we use are negation and or. And in addition the negation is applied in the innermost level in some sense and then in the next outer layer or is applied and in the outermost layer and is applied. So for example we may have a formula like this P and Q and not P or R or S and S or T or not Q. such a formula is in conjunctive normal form.

(Refer Slide Time: 17:36)



So this is C1 this isC2 and this is C3 and this is the conjunct of three clauses. So unlike in our earlier notations these are literals. They are not propositional variables anymore they are literals which mean they are atomic propositions or negations of atomic propositions. now We can convert any formula to clause form. we will not do this now but may be when we look at first order logic where we will also be interested we will look at the procedure. But the basic idea is to replace every other connective with either one of these 3 connectives combination of one of these 3 connectives. And then apply de morgans laws and distributive laws and associative laws to basically arrive at the particular structure that we are interested in where at the outermost level we have clauses then inside each clause there is a disjunction of literals.

Now this comes with a warning may blow up. That a formula when you convert into CNF form it could possibly blow up into length so as an example if you were to try to simply expand a formula like P is equivalent to Q you will see that it will have 3 or 4 connectives. Okay now we assume that we have a formula in CNF form and the resolution method is based on this on a tautological equivalence which is the following that if you are given a formula which is of this kind then this formula is logically equivalent to the same formula repeated which ofcourse is trivial but also a new formula which is added a new clause which is added which is Q or S. so if you accept this as a tautology then you can accept resolution method as the sound method. And I will leave this as a small exercise for you to show that this is a tautology.

(Refer Slide Time: 21:25)

Friday-30-Jan - Windows Journal

File Edit View Insert Actions Tools Help

Page Width

Resolution method – Tautological equivalence

Note Title 1/30/2015

$$(P \vee Q) \wedge (\neg P \vee S) \equiv (P \vee Q) \wedge (\neg P \vee S) \wedge (Q \vee S)$$

4 / 8

4