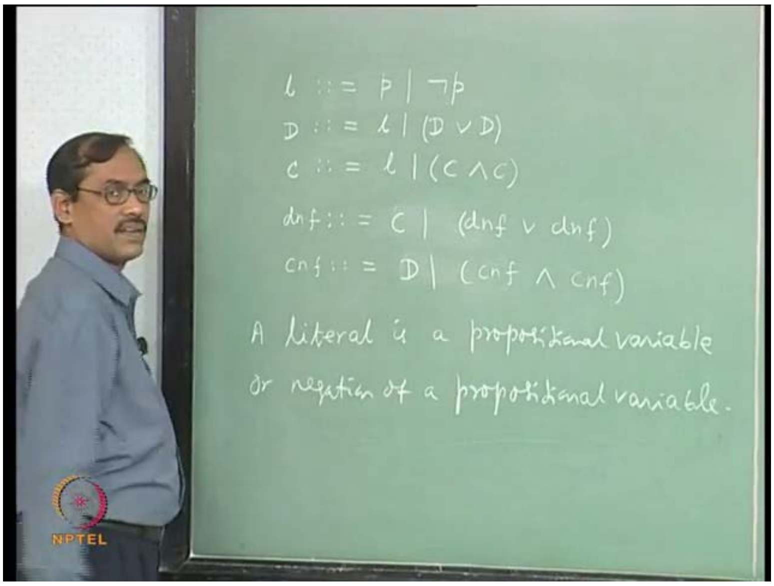
Mathematical Logic
Prof. Arindama Singh
Department of Mathematics
Indian Institute of Technology, Madras

Lecture - 9
Normal Forms

In the last class we have seen some consequences and some equivalences, and how to symbolize some ordinary language arguments into propositional logic, and then validate the ensuing consequences. Then we raised a question which tells that if some propositions are in some specified forms you can see their models or non models easily. Now, we will see that all propositions can be converted or brought into such forms; that is what you want to see. So, first let us give some name to those forms.

(Refer Slide Time: 00:58)

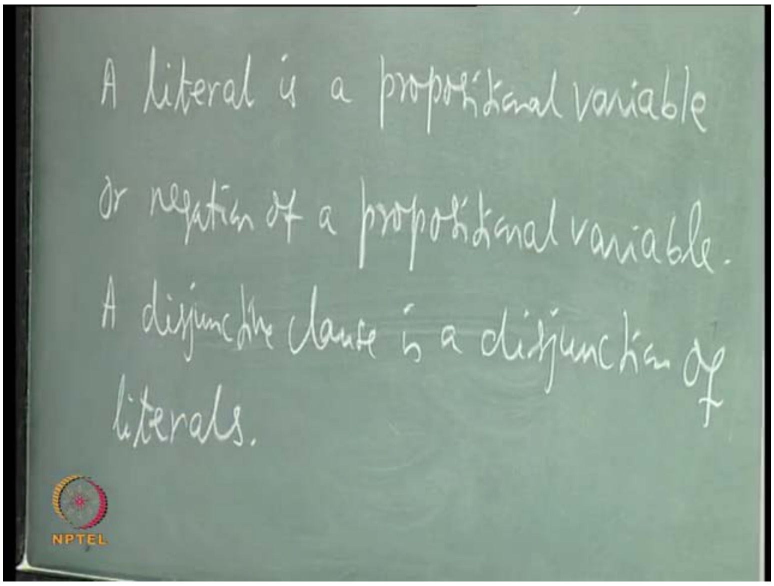


We define those forms: it starts with a literal; we say that a literal can be a propositional variable or negation of a propositional variable. So, here p stands for any generic propositional variable and l stands for a literal; a literal can be a propositional variable or negation of any propositional variable. Then we come to defining some clauses, so we say that a disjunctive clause can be a literal or it can be disjunction of two other clauses, so say, D or D.

If we are specific about the gammar, then you have to include a bracket here, parentheses, because once it will be combined with other clauses you need a bracket there, you need parenthesis. Similarly, a conjunctive clause can be either a literal or it can be conjunction of two other conjunctive clauses then finally, you define what we want. So, we say that a dnf can be disjunction of conjunctive clauses, so that way we have to write it as a disjunction should be, or of a conjunctive clause and another conjunctive clause, so which is C or C. So, once you have this, it means there can be finite number of disjunctions here not only two, it is recursive.

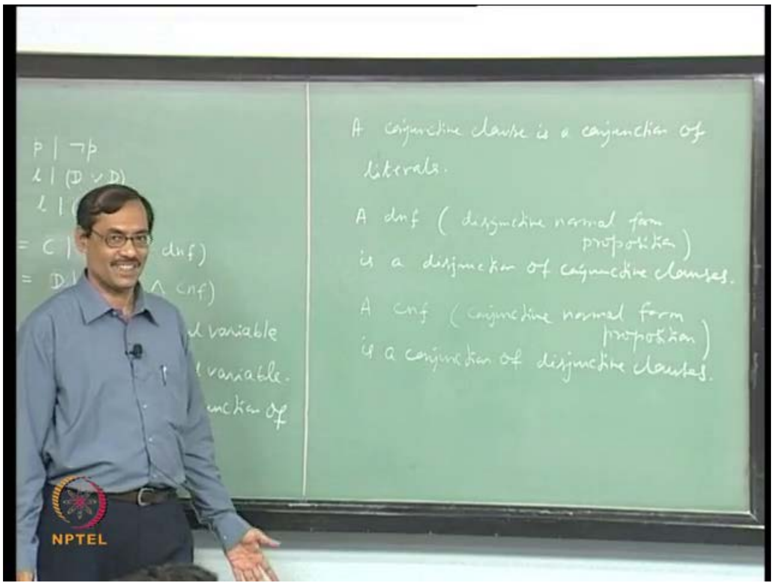
Similarly, a cnf can be a conjunction of two disjunctive clauses. But there is a problem when you give in to this recursive definition. It says only C or C, as a dnf, C is already there, so it allows two conjunctive clauses to be ored together, but we need really any finite number. So, what you do here is, just like this, we take a conjunctive clause, by definition, a dnf on then disjunct together many dnf's just like this. So, what we will do, we say that it can be a conjunctive clause or a dnf or another dnf. Similarly, here what do we do, we say cnf can be a disjunctive clause or a cnf and a cnf; if it is very critical.

(Refer Slide Time: 04:31)



So, a definition will be a literal is a propositional variable or negation of a propositional variable; that sentence comes from this. Next, we go for disjunctive clauses, so we say, a disjunctive clause is a disjunction of literals. Then, we say a conjunctive clause is a conjunction of literals. Fine? We will read that disjunctive clause is a disjunction of literals. This says a finite number of literals can be add together that is how you have to read it. So, finite number includes 0 also; that creates problem; it uses zero number of literals, will create confusion, it will not have here, but zero number of disjunction allowed. That means a single literal is by itself a disjunctive clause, similarly here a single literal will be taken as a conjunctive clause also. Next we define a dnf which can be read as disjunctive normal form proposition is a disjunction of conjunctive clauses.

(Refer Slide Time: 05:31)



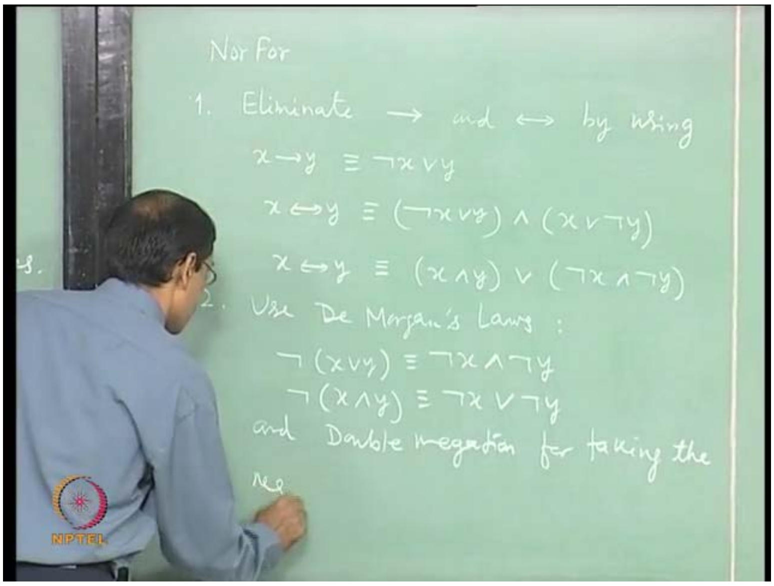
So, again with the same convention if you have single conjunctive clause that can also be considered as a dnf, the word dnf is used in two ways, we say a proposition is in dnf. So, there, you do not read this proposition, word, we say proposition is in disjunctive normal form also as it is for a dnf, a proposition which is in dnf, we say it is a dnf. In that case, we read dnf means disjunctive normal form proposition, both the ways it is being used now.

Similarly, we can define a cnf, the cnfs are conjunctive normal form propositions or just the normal form is a conjunction of disjunctive clauses. So, if we take a single literal, it is in dnf or cnf, both not either, both, ha, that is what you mean. Right? Because a literal by itself is a conjunctive clause; a conjunctive clause by itself is a dnf. So, literal is in dnf similarly, a literal by itself is a disjunctive clause and a disjunctive clause is again a cnf. Therefore, a literal is also in cnf.

Now, the question is how this conversion will take place? One way is to look at the truth table of any proposition and then get the models, so you go the reverse way, but that defeats the purpose. We want to see the models from the proposition, now we are considering the models and constructing the proposition. So, how does it help? Will it help in storing some knowledge there, storing that, this is the model of that, it is equivalent to telling that these are the models we have found out, fine. So, you should have some different ways of finding out dnf or cnf equivalent to a proposition.

It is easy to get it because all that you need is your proposition should have only negation symbol, or symbol, and the and symbol. First thing it says, you should eliminate this implies symbol and the biconditional symbol, using your laws. If you write an algorithm for this normal form conversion, it would first proceed with eliminating these two connectives.

(Refer Slide Time: 10:14)

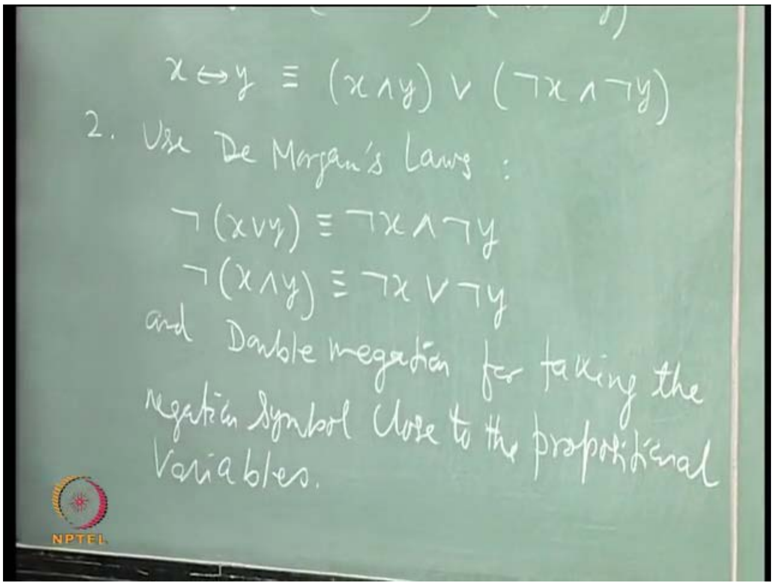


So, first step in the algorithm, let us call it normal form conversion; so then, we should first eliminate this and this by using the laws. Which laws you may be using? By using say x implies y is equivalent not x or y, and then x biconditional y is equivalent to and of both the things, you can take. So, it is a not x or y and x or not y. It is combination of x implies y and y implies x. Or, you can write it in a different way, you can also say x if and only if y is equivalent to not x not y.

You just see its models; from x and y are having the same truth value, so if x, y both are true you get x and y; or let us write it first, x and y or both are false, so not x and not y. Any one of this can be used; sometimes negation of all those things. If you can recognize, it becomes quicker, for example, not of x implies y you can straight forward write x and not y. Similarly, these two also, but anyway, we will take care of that later. So, second thing is, now once you have applied this step, your proposition does not have the symbols implies and biconditional. Then what should we do, see, all that we want is a conjunction of disjunction of literals or a disjunction of conjunction of literals.

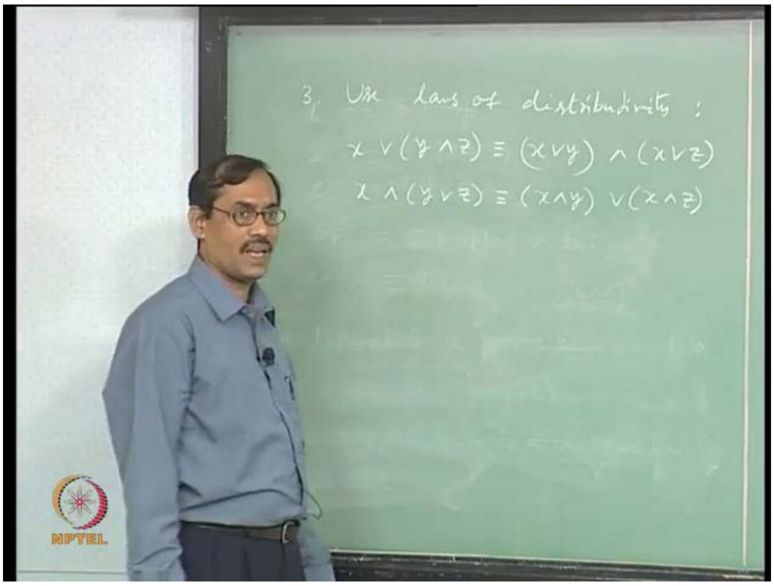
Literals, that means negation symbol should not come anywhere else except near the propositional variables. So you have to take the negation symbol near the propositional variables, fine? Use De Morgan. Use De Morgan law. What are the laws not of x or y is equivalent to not x and not y; the other one is not of x and y is equivalent to not x or not y. But literals also have another property. There cannot be two negation or three negation symbols; this might introduce many more negation symbols near the propositional variables.

(Refer Slide Time: 13:45)



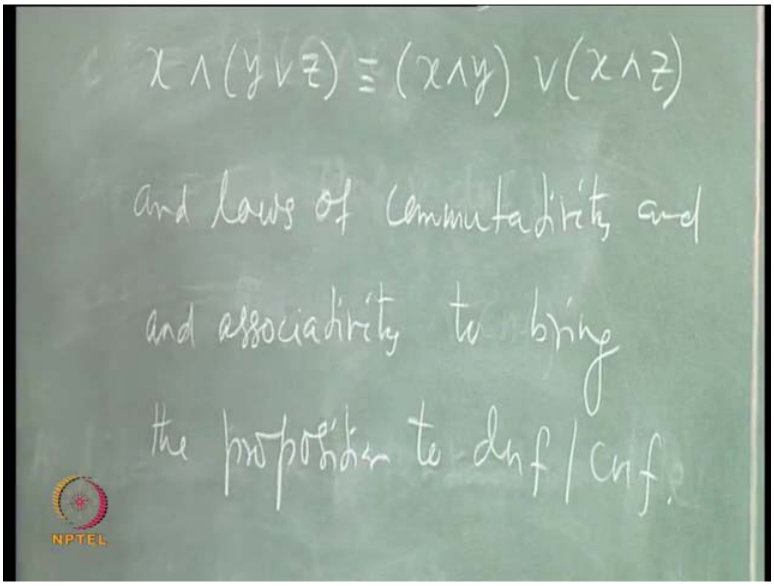
You have to eliminate them; that is, use our negation and double negation for taking the negation symbol close to the propositional variables. Next step? See, whatever we have done till now, has also a name. It is called a negation normal form or the negation form with the propositional variables, but and’s, or’s can be anywhere. In some way, they are there; that is all we want, a specific form again AND’s of OR’s, OR’s of AND’s; so finally, what you have to do is, use distributivity.

(Refer Slide Time: 14:46)



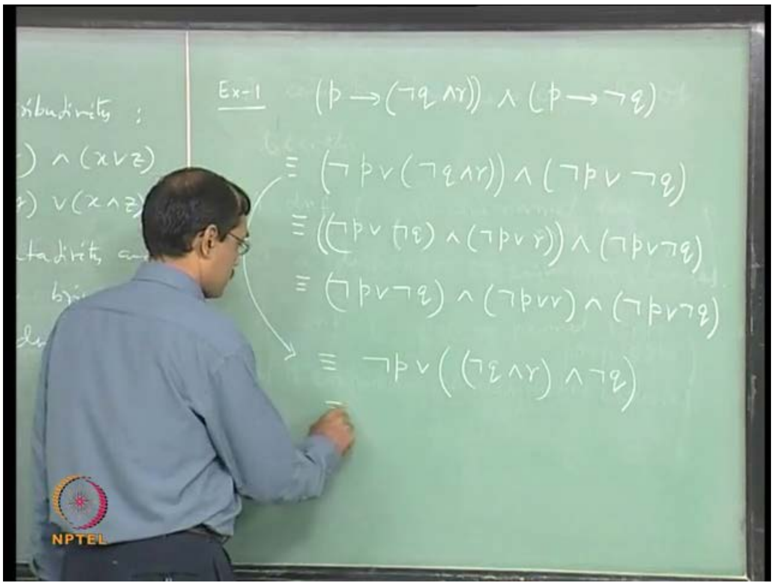
First step is use laws of distributivity which can be written as x or y and z is equivalent to x or y and x or z also the other way. So, this has, if you have something, some propositional variable, let us, here p or y and z, then you can put in this form, but if it is in the form y and z or p?

(Refer Slide Time: 16:25)



So, you need commutativity, you need also associativity because you are not going to write all those parenthesis; and laws of commutativity and associativity will bring the proposition to d n f or c n f.

(Refer Slide Time: 16:23)

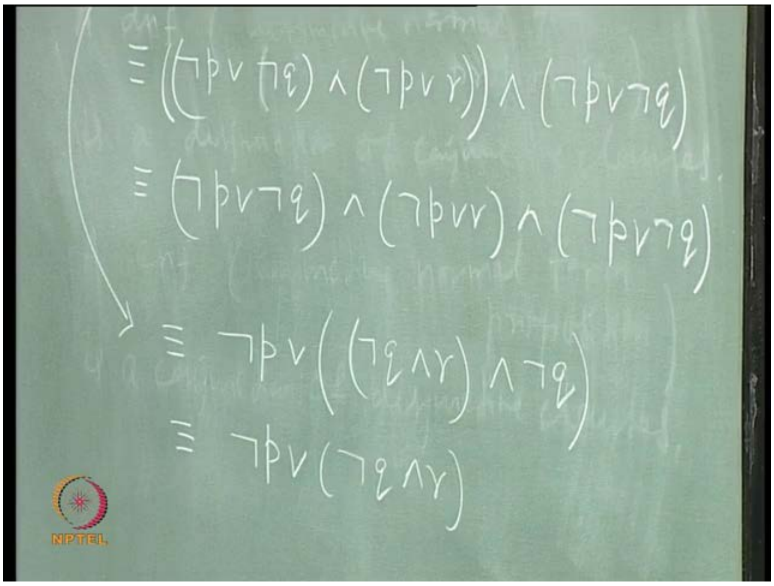


So, let us see the first problem, and p implies not q, so it says first we have to eliminate this two occurrences of implies. This is equivalent to not of p or not q and r and not p or not q. Next, there is no need of using De Morgan, here, not is never outside; it is near the propositional variables, they are already literals. So, second step is done. We go for distributivity, now this not p when distributed over, gives if not p or not q and not p or r and not p or not q, fine?

Due to associativity, you can forget the brackets; so you just write it as not p or not q and not p or r and not p or not q. Fine, but still there are some other ways, like if you recognize here not p or not something and not p or something. So, you could have used distributed in the other direction, so what does that give? It will be not p or not q and r and not q, you check it back not p or not q and r and not p or not q, now what should I do? This one, not q and r and not q, so use the law of idempotency, which is not specified in the algorithm, does not need it anywhere, but if you know some laws, of absorption laws of idempotency. They will help you in simplifying the cnf or dnf; now you can modify even the algorithm if you want to use them.

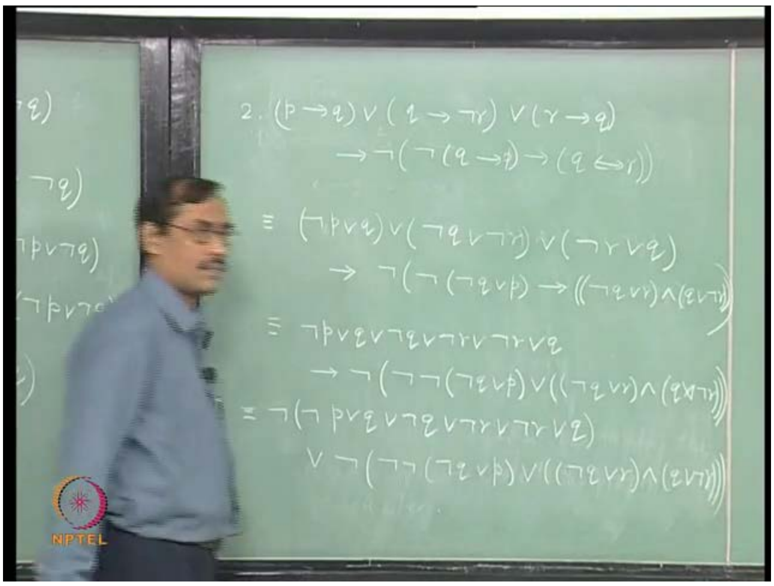
Now, what does it say? not p of this becomes not q and r that is all because not q and not q is equivalent to not q. This gives not p or not q and r, this is a simplified version. See, sometimes you may reach at one proposition which is in dnf, but you want a cnf, so what should you do there? Just blindly use distributivity, you would get it; but that is not easy in terms of computer time because so many factors are there, they will be multiplied out , that is, you can get a cnf.

(Refer Slide Time: 19:41)



So, there is a way. Fine. This is a theorem again, what we have done till now through an algorithm; it says that every proposition can be converted to a cnf and every proposition can be converted to a dnf equivalently, preserving equivalence. You can convert anyway anything right, but all that it says that, equivalence is preserved; fine.

(Refer Slide Time: 20:53)



Let us see another, second problem, p implies q or q implies not r or r implies q implies not of not of q implies p implies q if and only if r, first parenthesis is not required. What we do? We have to eliminate the connectives implies and biconditional, so not p or q, that is, for this implication, or not q or not r, for this implication, r not r or q. For this implication, let us keep it, it will create problem otherwise; so implies, not of not, so say, not q or p, again keep it.

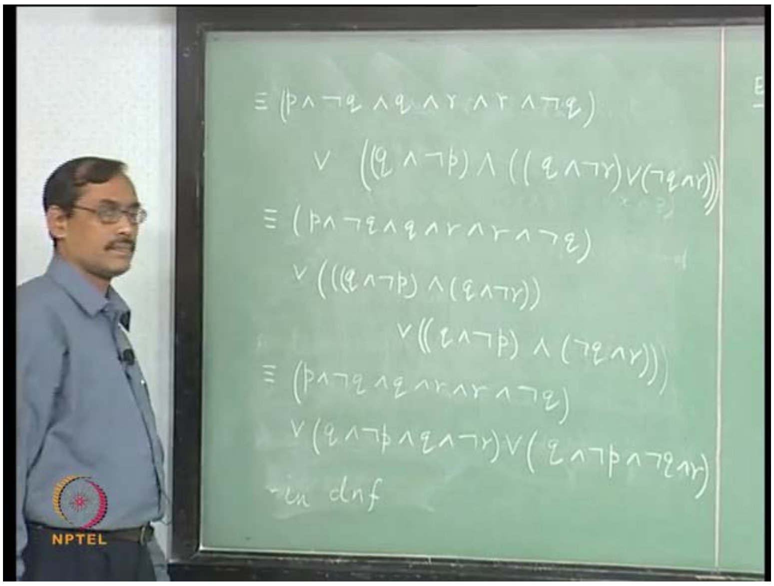
Now, q biconditional r, which is, not q or r and q or not r, one more bracket, so there is a nested implication, one is here outside, this is, say inside, first, let us take care of this; you can take care of both at a time. Let us take care of this first, this gives, these are all OR's, so I can put them together with associativity.

This implies, not of, now this becomes not of this or this and not of this is; what you can, use double negation straight forward, anything can be used anywhere; but we wanted to follow the algorithm. So, we have to write this way here, double negation we are omitting, so once you take out this, it becomes not q or p or this one, is it is confusing? Let us take not of this thing, that is, not of whole thing, not x or y, you need one more, so that gives. See, if you want to write all those things you can or some simplification you can do. See that q or not q is occurring, with that if you make any, or it will be remaining that only, because this is equivalent to top.

You can simplify here itself; you take top as this, and then top implies x is equivalent to what? x? That also you can do, but if you do not want to do, still we can proceed, does not matter, let us take that as it is. This will be negated, because this simplification is there and then, or with the whole thing, that gives or the whole thing, that is not so as per the algorithm, you have not used any simplification till now.

Now, algorithm says: next step is to go for the distributions, before that we may need De Morgan's law. If you use De Morgan's law, what does it say; this not will go inside. So, De Morgan's law says that, for it will go inside, it goes on changing the OR's to AND's, AND's to OR's, that is what it will do, and if you use double negation along with that, it goes, it becomes p and not q and q and so on. You can write as equivalent of, and that not goes inside, is a p and not q and q and r and r and not q, when this not goes inside, then or is as it is. This not does not goes inside. When this not goes this double negation is not there, as if, now, what happens not q becomes q and not p first term gives q and not p. Then this becomes and next is this not is going, so here it becomes q and not r q and not r, this becomes or, then not q and r just verify with yours. Next, what should we do is,write in any form. Here is one conjunctive clause or there should be other conjunctive clause it is not a conjunctive clause.

(Refer Slide Time: 26:52)



You have to distribute; you just keep it as it is, or, which or to distribute, this and only or is here inside, also you can distribute, but then suppose I take this and distribute this and it becomes this and this. So, you need another distribution; any other simplification can be done, suppose you distribute this, can you distribute this?

Student: And first one

Which one?

Student: First one.

This?

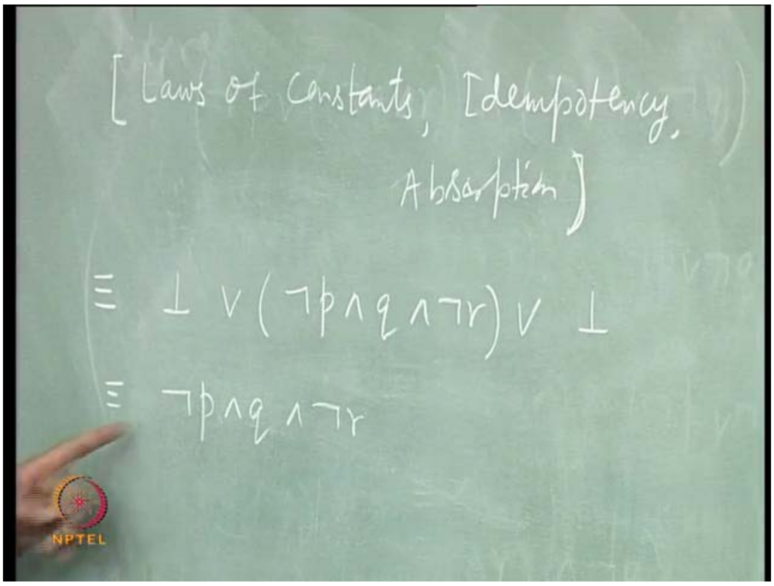
Student: Yeah.

Let us try that. This is q and not p and q and not r that is one other factor is q and not p and not q and r is that fine this is in correct form.

Student: Sir, it is actually

We do not need the brackets; we have to take only parenthesis, that is all to bring it to looking good, only for elegance. What we do, q and not p and q and not r another is q and not p and not q and r. So, it is in dnf. Now bring it to cnf, distribute everything. How many factors are here? One, two, three, four, five, six, here are one, two, three, four here are four. So, it will give you how many clauses? There 6 into 4 clauses, that becomes clumsy. What you need is, use the absorption and idempotency and the laws of constants, because you know it is not q and q. This becomes bottom which will help, so laws of constants also will be helpful.

(Refer Slide Time: 32:08)



What we are using is: laws of constants, idempotency and absorption. What is absorption? Idempotency means x or-ed with x, x or x is equivalent to x; x and x is equivalent to x. What is absorption? If you have x or y then it will take and x. That, x is absorbed in x or y, so it is as x or y and x is equivalent to x or y or x, so x. And you think of union and intersection and say we just said a union b and a and is intersection.

Student: x is x and x.

x, not x and y. Now, what about x and y or x?

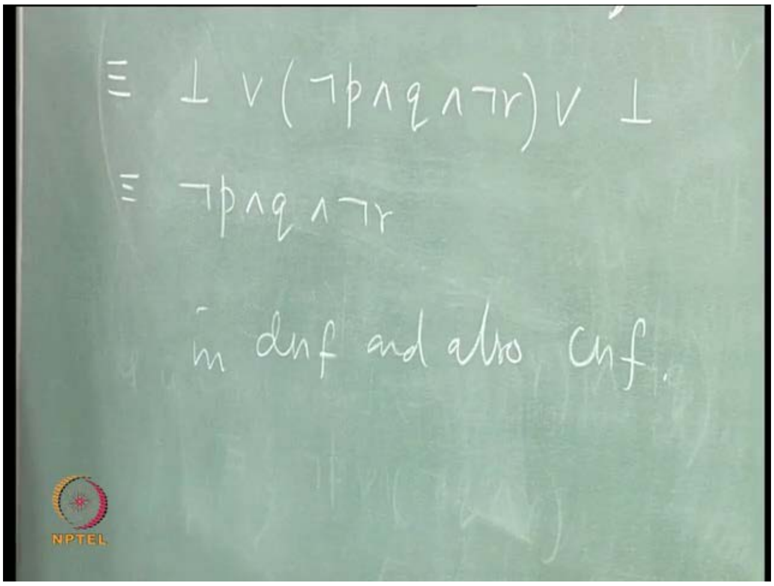
Student: x and x

That second.

Student: x and

x, so these are laws of absorption, now let us see what comes out of this.

(Refer Slide Time: 34:06)

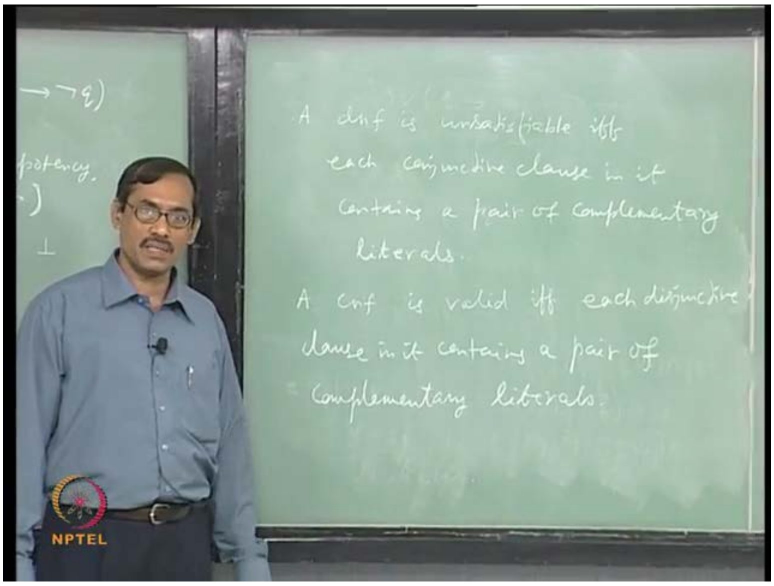


This is equivalent to, now we observe this, and then, that bottom or what else? u and not v and q is idempotency, so not p and q and not r, next is q and not q is bottom again. Now, this is in dnf or cnf? Both? Now, it is in dnf and also in cnf, ha, as a dnf, it has a single clause, single conjunctive clause, this itself is the clause. Conjunctive clause? As a cnf. The clauses are the literals, there are three clauses in the cnf, in this cnf; and there is a single clause in the dnf; so did not have to write all those 6 into 4 into 4 disjunctions!

Sometimes, logic will help you this way, first you observe, here is writing your first clause as bottom because q is there not q is there. Now, suppose you find finally that in every clause you have a literal and its negation. You have a propositional variable and its negation, then automatically it will be bottom, it will be unsatisfiable. That means, if it is a dnf, and you find that every clause is having a pair of complementary literals, then you say that it is an unsatisfiable proposition, this is the observation.

Now, can you prove the converse? If it is in dnf, it is unsatisfiable, then every clause should contain a pair of complementary literals. Yeah? Clear? You see this by contraposition. If not, suppose there is a clause where there is not a pair of complementary literals. Then what you do: give 1 to each of the literals, you are defining an interpretation, so define the interpretation this way: each of the literals will get 1, a literal can be not of p, so that p will become 0. That interpretation becomes a model for the proposition; that is a model, so it is not unsatisfiable.

(Refer Slide Time: 37:11)

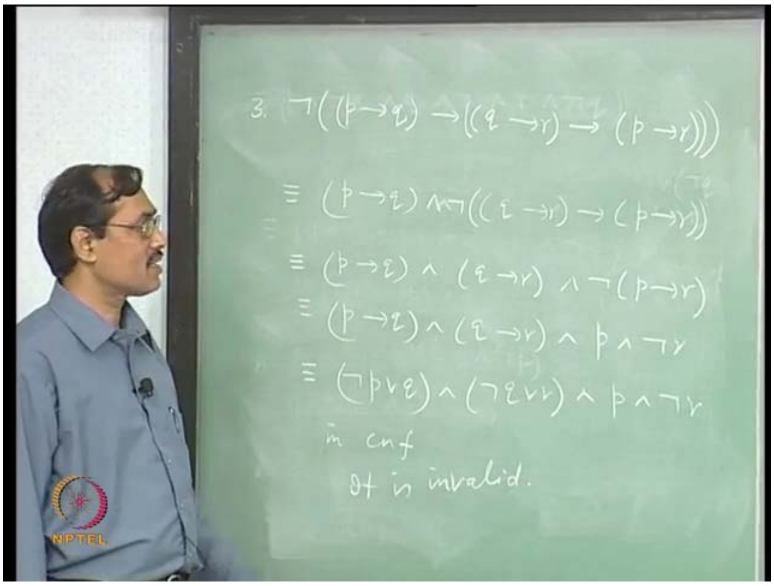


What we observe is, a dnf is unsatisfiable if and only if each conjunctive clause in it contains a pair of complementary literals. A pair of complementary literal means, for some propositional variable p, there is p, there is not p; p and not p are called complementary literals, complementary to each other.

The same we can say: a cnf is valid because you take a cnf, consider a disjunctive clause in it, that clause is having a pair of complementary literals p and, p or not p, and then same for other or's; so that p or not p becomes top. Now, every clause is top there, and-ed together, so finally it is top. So, a cnf is valid if and only if each disjunctive clause in it contains a pair of complementary literals.

That means if you convert to a normal form, then you can decide whether it is valid or it is not. Validity, can we decide it by changing it to cnf? If you want to find out that is satisfiable or not then convert it to dnf. You can find out whether it is satisfiable or not, is that clear? But unfortunately, one of the forms does not tell you both. You may need distribution and some other laws to be applied, so it might be difficult that whether a dnf is valid or not; that decision can be difficult unless you convert it to a cnf again. Similarly, whether cnf is satisfiable or not, it might be difficult to decide; difficult in the sense of what? Not that we cannot do it, it might take some time at least we have to convert it to the other form and then decide.

(Refer Slide Time: 40:20)

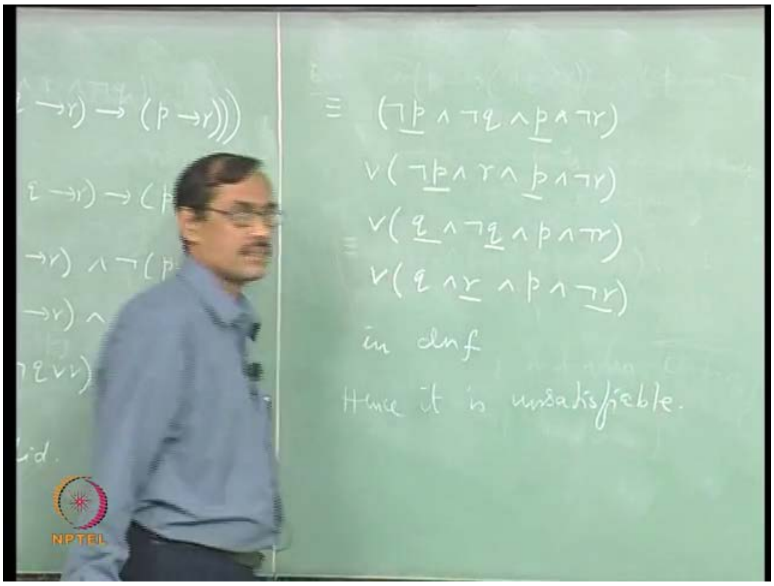


Let us take the problem three, where you want to convert it to dnf or cnf and then decide whether this is satisfiable or it is valid, unsatisfiable or invalid, so let us first convert it. This is, first, negation symbol, see some simplification. You can see here, it is in the form of not implies, so you can straight forward use not of x implies y is equivalent to x and not y instead of writing or. Then applying De Morgan, so this gives p implies q and q implies r implies p implies r its negation right what we are doing is not of x implies y is equivalent to x and not y.

Next, what should we use? Negation again, it is p implies q and q implies r, not of p implies r, once more. Now, this becomes not p or q and not q or r and p and not r, which is in cnf; yeah? Each one is a disjunctive clause; naturally it comes to cnf, but cnf says, it is valid or not can be decided. Is it valid? It is not, because there is one clause where there is not a pair of complementary literals; there is a clause where there is not a pair of complementary literals; if not p is there, p is not there. If q is there not q is not there. For validity each clause should have a pair of complementary literals; there is one that itself is a literal, a propositional variable. So, no pair can be available there, so it is not valid; that must be; we know it is invalid. Now immediately we conclude it is invalid.

But invalid means it can be satisfiable; it can be unsatisfiable. It looks to be unsatisfiable? So to apply that observation what we need is convert it to a dnf. Then only you can do it, so let us convert and see. You can go back and do some tricks, but from here itself we can do, so let us see. A principle says if you distribute it is correct, so distribute. This is one clause, this is another, it is like multiplying x plus y into y plus z into z plus x into x plus w or something, four factors, chose one factor from each and write them, that is what you will be doing.

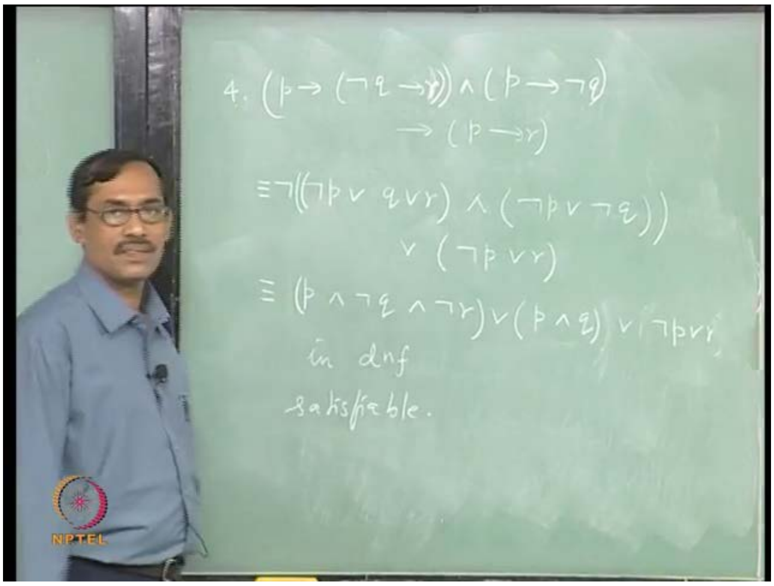
(Refer Slide Time: 44:53)



It gives not p and not q and p and not r or not p and not q, not q is gone, so I take r; now and p and not r, or I take the next factor: q and q and p and not r or q and r and p and not r. So, this is in dnf; and now you go back to the principle, apply it. Here we see that p, not p, there is a pair of complementary literals, in the next clause there is also p, not p, in the next clause, q, not q, in the next clause, r, not r; so it is unsatisfiable.

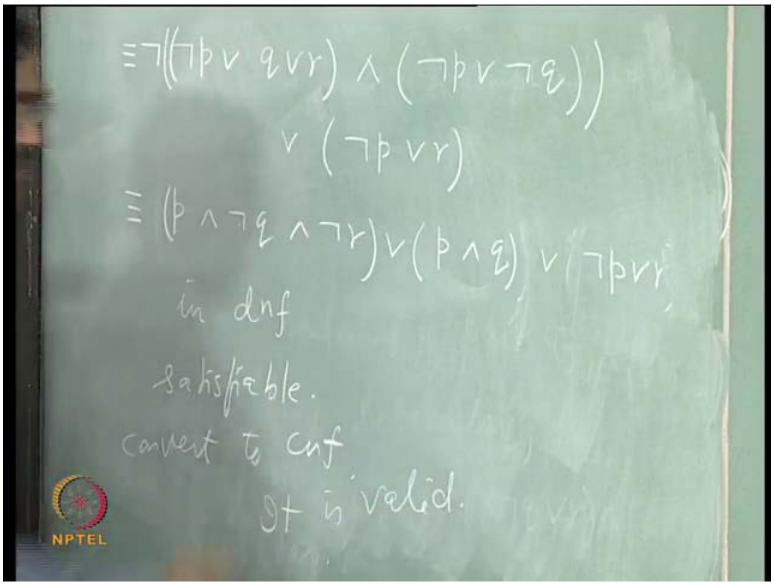
Let us take one more, forth one. This is p implies not q implies r and p implies not q implies p implies r. We start eliminating; this is equivalent to not p or first clause; I am taking first proposition before this, so not p or this. That will be again not of not q which is q or r and not p or not q, so not of this whole thing, or not p or r; do it yourself. Next, I take not inside, that gives p and q or not r or again, here p and q or not p or r. Now, you observe, this is a conjunctive clause, this is a conjunctive clause, or is here or is also here.

(Refer Slide Time: 46:13)



So, I should remove these two brackets, that is all. There all or's, so that not p itself will become a conjunctive clause, just remove and remove these; this is in dnf. So, dnf says, what? Satisfiable or unsatisfiable. This is not unsatisfiable because there is a clause having no pair of complementary literals; so it is satisfiable.

(Refer Slide Time: 50:08)



But it may be valid, it is a weaker thing, it can be stronger; it can be valid. We change it to cnf and check. What will happen there? p, not p, that clause is gone, equivalent to top; then p, q, not p once, not p is there, p is gone. You check the next, not q with q will give top, and similarly, not r with r will give, so that will become valid. You can simplify also, where to simplify first? It is not q, this is q; it does not matter, it is a smaller one. So, you just multiply out and convert it to cnf by distribution; just distribute the OR’s AND’s then you get that it is valid.