Introduction to Logic
Prof A.V. Ravishankar Sarma
Dept. of Humanities and Social Sciences
Indian Institute of Technology, Kanpur

Lecture - 39
Formation Trees for wff's in predicate Logic

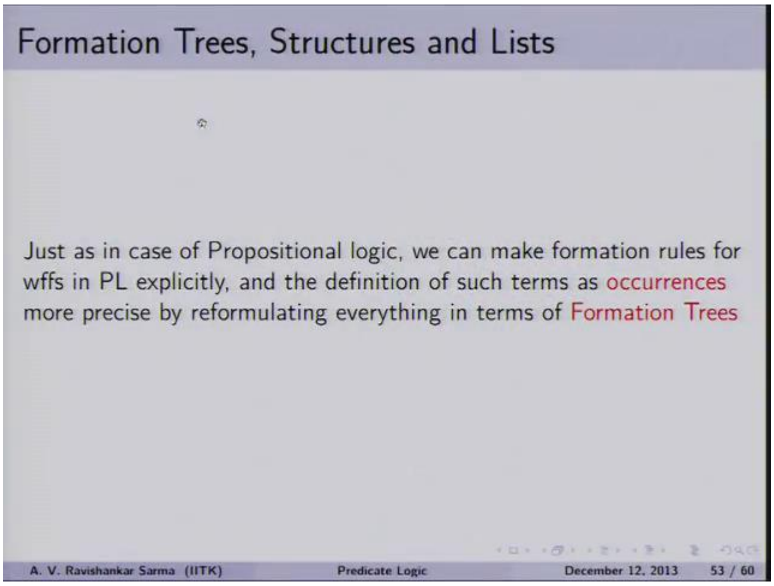
Welcome back in continuation to the last lecture, where we discussed primarily about syntax of predicate logic, where we discussed about some of the building blocks of predicate logic. So, predicate logic was one of the important building blocks of predicate logic are variables, constants, function symbols and predicates. This is the fourth thing that we have discussed in greater detail and we discussed about the meanings of these things.

Then, we discussed about one of the 2 important operators which is which makes predicate logic distinct from the propositional logic, so they are quantifies. So, we require these quantifiers especially, talk about certain things such as if we want say that all birds are black for example, we need to have quantifies. And then in that context we introduce 2 quantifiers: first is for all x and the second 1 there exist some x.

Then, later we discussed about the scope of a quantifier and when a given formula and a predicate logic is considered to be having free variables etcetera and when, we say that a given formula is a close formula or when a given formula has some kind of grand terms etcetera and all. All these things which we have discussed in greater detail in the last class and also we also discussed about some of the important formulas, some of the important properties of quantifiers especially, whether or not quantifies distributes etcetera and all.

These are the things which we have seen in that context we have seen that, when the quantifiers are the same for example, if you have for all x for all y and then p x y it is same as for all y for all x p x y. So, the order does not make any big difference especially, when you have the same quantifies, so now in continuation with the syntax of predicate logic.

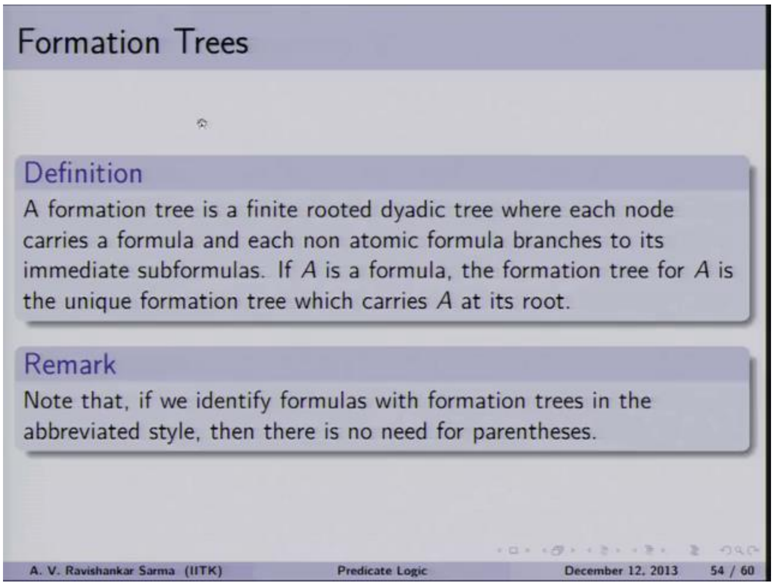
(Refer Slide Time: 02:31)



So, what will be doing now is will be discussing something about some of the formation trees of a given well formed formula in a predicate logic. So, just as in the case of 4 Propositional logic, we have a something called unique formation tree with respect to a given well form formula. Just like that in the case of a predicate logic as well we have for each given formula; we have a corresponding tree a formation tree. So, the advantage of having formation tree is simply this, that.

Suppose, you are not given any parenthesis etcetera and all, so once you draw the formation trees, then there will not be much ambiguity with respect to the parenthesis that are concerning the given formula. So, now the formation trees just as in the case of propositional logic, we can make formation rules for the well form formulas in predicate logic explicitly, and the definition of such terms as occurrences more precise by reformulating everything in terms of Formation Trees.

(Refer Slide Time: 03:41)



So, what we mean by Formation Tree? In the context of propositional logic here is the definition of formation tree, what I am trying to simply say is this that each and every given formula will come up it is own unique kind of tree or a formation tree. So, suppose if you say for all x p x y implies q x for example, that will have some second specific kind of formation; a unique kind of formation tree when you compare with another formula such p x y implies not q x y for example, that will have his own formation tree.

So, each formula comes with its unique formation tree. So, now what is the definition of a Formation Tree: A formation tree is a finite rooted dyadic tree where each node the 1 which is at the top most point that is considered to be the node the root point it is like a tree with trunk, and then you have branches and leaves etcetera and all. So, formation tree is a finite rooted dyadic tree, where each node carries a formula that is a given formula will be the sitting at the node and each non atomic formula.

Suppose, if is an atomic formula the tree ends there itself. If it is considered to be non atomic formula, in the context of proposition logic p all the individual letters such propositional variables such p q r etcetera and all they where the atomic formulas. In the context of predicate logic constant, individuals etcetera they all come under the category of atomic formulas.

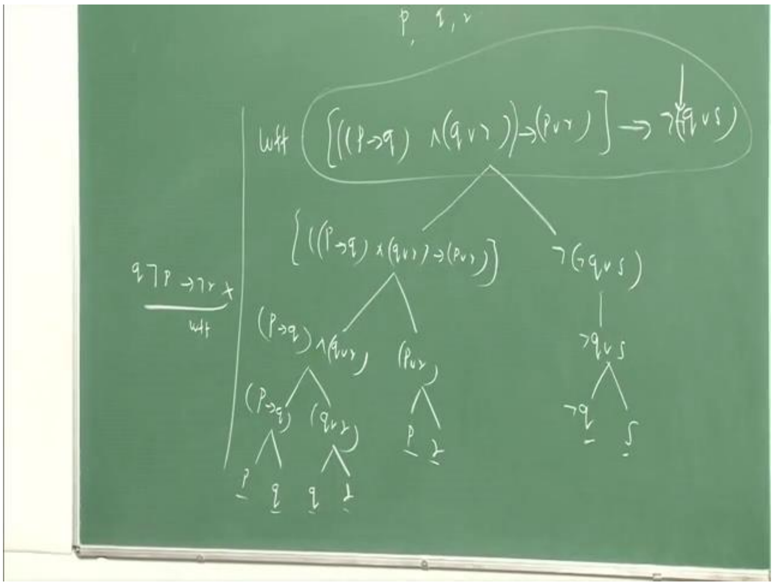
So, it carries a formula and each non atomic formula branches into its immediate formulas and these formulas are called as sub formulas. For instance if A is considered to be a formula then the formation tree for A is considered to be kind of unique formation tree which carries that particular kind of formula at its root. So, 1 important remark that I like to make here is this that, if we can identify the formulas each and every given formula with respect its formation trees in the abbreviated style.

Then obviously, there will be no confusion ambiguity etcetera and all because, we face this particular kind of problem earlier especially in the context that suppose if the parenthesis are not given. Then, we follow certain kind of priority which we will be using on the logic operator that we are using in the given formula. So, in the context of propositional logic and the connective end is given the smart most important.

Then, followed by that or and then of course, negation is there and then implication and double implication that is order that we follow in the case of propositional logic this just a more convention. So, one of the important thing which I like to point out here is this that if you can have a formation tree for a given formula and then obviously, there will be no confusion and all.

So, that formula represent that unique tree represents given kind of formula. So, that is a advantage of having the formation trees. So for example, in the context of propositional logic, suppose if you are given a formula like this, then its corresponding formation tree will be like this.

(Refer Slide Time: 07:31)



For example, if you have a formation tree like this p implies q and q implies r implies p or r. So, this is 1 particular kind of string implies not q or s. So, this tells us that p implies q and q implies r implies this 1 and the whole thing implies this 1. So, now the formation tree for this particular kind of formula this is considered to be well form formula. So, if you write p q implies not r this not considered to be well formed formula; that we have discussed in later detail in the context of propositional logic.

So, now this formula will come up with this unique formation tree. So, the first thing that we will be doing is this, so this is considered to be non atomic formula. If it is a atomic formula it will end with only letter such as p q r etcetera not p etcetera; this is not an atomic formula. So, then this we will be expanding the formula into the branches or leaves which you can all. So, that till to such an extent that will end with only atomic formulas.

So, now this is going to be like this first, so these branches into the first formula that you have on the left hand side is this thing p implies q and q implies r implies p or r this is a first 1 branches into not q or s. So, now these further changes into not q of s still become q or s and then this q or s still it is not an atomic formula. So, this is considered to be complex molecular formula, this consists of 2 atomic formulas and all.

So, even 2 atomic formulas combine and then it will become molecular formula, then it further reduces to thing q s. So, this is on the right hand side now, this branch leads to p implies q and q or r and then the next formula that you will be reading is p or r. So, this is the way with in which you pass this formula or you will read this particular kind formula is mare kind of convention and all which 1 will be following.

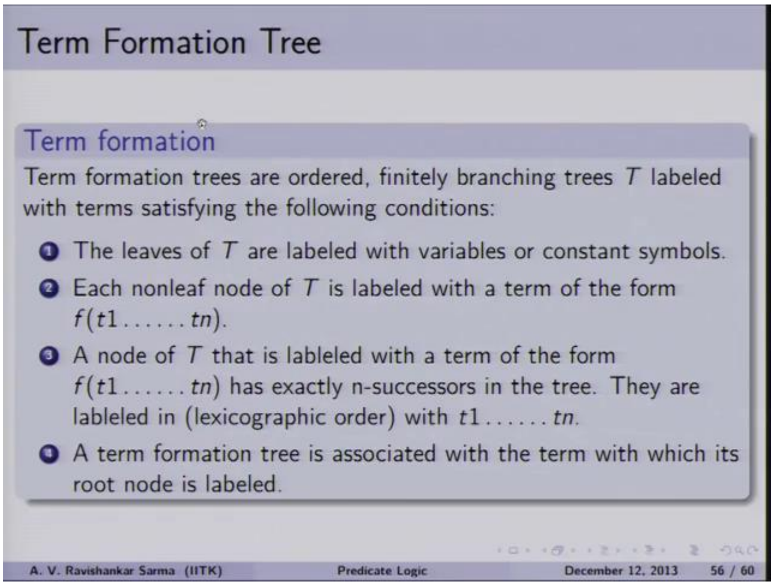
So, this is still a non atomic formula this reduces to p and r and then this reduces to p implies q q or r and this further reduces to p and q and q and r. So, these are things so now, we ended up with all atomic sentences and all. And this completes our tree, the formation tree for a given formula whole be, suppose if we change this particular kind of formula into some other thing and all not q or something like that. Then, it will have its own formation tree and all.

Now, till to this extent it is same now it will become not q or s not q or s suppose if you introduced this particular negation, so this is completely different from formula the formation tree that you have seen earlier. So that means, each and every formula in a propositional logic come up with its own formation tree and that has to be unique; it cannot be the case that any 2 formulas will have the same kind of formation tree.

If there if you have the same kind of formation tree then they are considered to be logically equivalent otherwise, each and every formula comes up its own formation tree. So, this kind of idea will be extending it to the predicate logic and then we will try to talk about what we mean by saying that, a given well formed formula in a predicate logic will have its own formation tree.

So, why we are doing this particular kind of thing? Is just because of the case that each and every formula comes up with its own formation tree and that is a reason why there is there will no ambiguity; when you draw a diagram for formation tree for a given well formed formula in a given logic. Then, the ambiguity with respect to parenthesis and the parenthesis will not arise once you form a kind of formation tree.

(Refer Slide Time: 12:35)



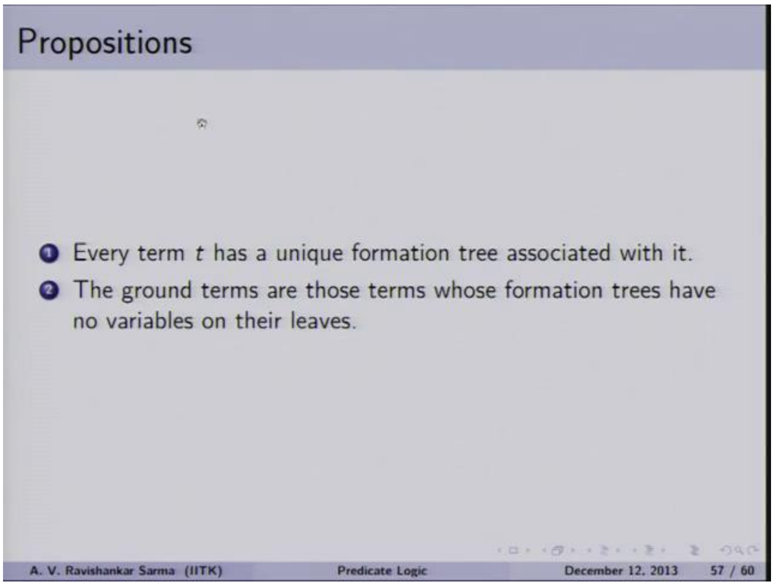
So, now in the context of predicate logic here is the definition of this thing Term Formation tree we have 2 different things: 1 is term formation tree and the second 1 is atomic formula formation tree. So, those in the predicate logic we have constants, variables, predicates and functional symbols and then with that you can these constants will combine with the help of functions etcetera form some kind of terms.

So, now how we form this particular kind of terms, so term we have formation tree for terms first of all and it will be like this. A term formation trees are usually considered to be ordered and they are considered to be finitely branching trees. It will not go forever ever and on ultimately it ends up with some kind of atomic formulas it is label with terms satisfying the following conditions.

So, what are these conditions are like this: The leaves of the given term that is t what is going to sit at the root of the formation tree and these are label with variables and constant symbols and each non leaf node of T is label with a term of the form f and then t1 t t t n are considered to be terms.

So, now in node of t that is what is at that top of the tree is going to be labeled with a term of the from f t1 t2 tn that needs to be expanded and all which has exactly some kind of successors n successors in the tree. It can be 1, it can be 2 etcetera and all. So, there are labeled with t1 to tn. So, a term formation tree is associated with the term with which its root node is labeled.

(Refer Slide Time: 14:40)

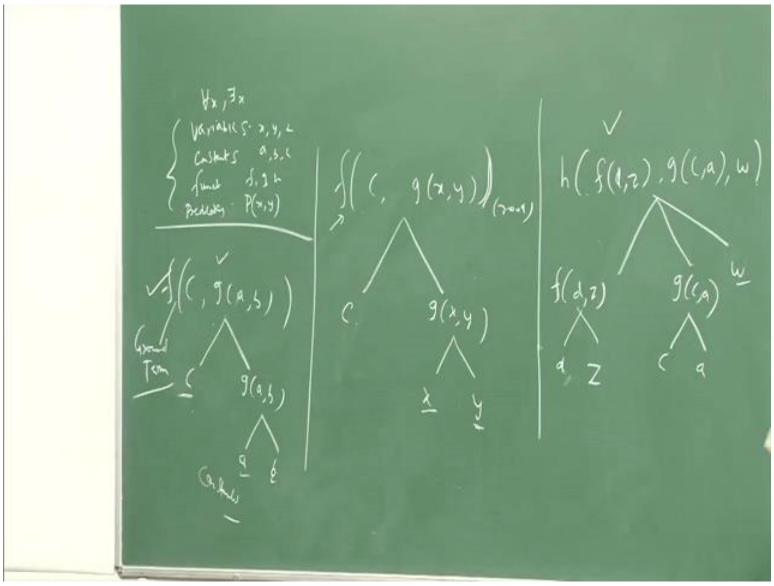


So, these are things which we need note every term t has his own unique formation tree associated with it. Just as in the case of propositional logic I given well form formal will have its own unique tree. Just like that, we are trying constructing a formation tree for a given term. So, that is going to be if you change some of the things in that term and that will have its own formation tree.

So, node 2 formulas will have same kind of unique formation tree, unless until they are logically equivalent or there the same kinds of terms. You are talking about the same kinds term they will have the same formation tree. So, the ground terms are in the context of this formation tree; the ground terms are those terms whose formation trees will have no variables on their leaves.

So, just imagine a tree which has its root at the node and it is like some kind upside down kind of tree. It starts with root that is a node that is where you start with the given term, and then you will start if it is starts branching it out and it forms goes to the leaves etcetera. And then if you find that there is no leaves etcetera, then it is considered to be a kind of ground term. So, let us consider some examples so that, we will get this point clearly. For example, if you have 2 terms like this, then what is going to be the tree diagram for these kinds of terms?

(Refer Slide Time: 16:40)



So, a term can be conceited by these things c is constant, g is a function and x and y are variables. So, this is 1 particular kind of term for which we are trying to form formation tree and the other 1 is, something like f of d, z and g c, a and then w. So, first we will talk about the formation tree this particular kind of thing. So, now this is what is sitting at the root and all or you can imagine that its trunk of a tree.

So, now so the first term you are trying see is this 1 c and then you have g x and y and then this can be further x and y. So, this formula it has it is not these are not considered to be ground terms because, this formula consist of variables x and y. Then, suppose if you are formation tree has only ground terms; does not have any free variables and that is considered to be the ground term and all.

For example, if take the same formula like this f of c, g of a, b for example, this is considered to be term that we trying to form formation tree for this particular kind of term. So, what is important in predicate logics is this thing first you have variables x, y, z etcetera, and then we have constants like a, b, c which are referring to some kind of individuals. And then you have functional symbols like f, g, h etcetera and then you have predicates.

So, predicates can be something like p, x and y etcetera where x and y are related in such a way they have some kind of property like x is a father of y, x is a brother of y etcetera and all. So, these are things which are the building blocks of this thing and then we have 2 quantifies for all x there exist some x. So, now this is considered to be a term because, you have constants and you have function and then you have followed by some kinds of constants and all.

So, this considered to be term which we have defined the meaning of terms in the last class the last few classes. So, we can go back to that particular kind of definition of that term. So, now what we are trying to do here is this thing you are trying to draw the formation tree for this 1 go this is going to be like this. The first letter is going to see is c and then g of a and b. So, this is the way in which you read the formula.

So, its starts with c and then we go on to the next 1 that is g a, b; that is the way computer reads all the things and all. It goes from left to right that is the convention that we follow. So, now this further reduces to this 1 a and b. So, now you need to 1 need to observe that in this particular kind of formula these are considered to be leaves of you are root of the tree. So, all these terms are constants; they are not considered to be free variables.

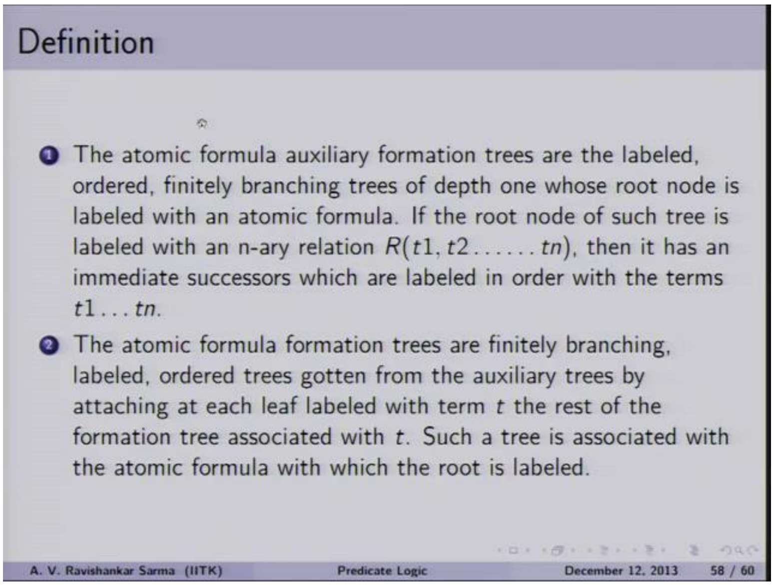
So, variables are likes x, y, z etcetera and all where a y thing you can substitute into x, y, z etcetera. So, now in this formula all are considered to be constants only and the and hence, this particular kind of thing is called as ground term. A ground term is a term, which has only constants it does not have free variables at the leaves. So, you have c a b etcetera all these things are constants.

So, that is why this is considered to be a ground term whereas, this 1 is not considered to be a ground term because, it has variables x and y. So, anything we can substitute for x and y depends upon what you substitute for x and y and interpretation of this 1 changes. So, now coming back to this particular kind of formula, so this is going to be this thing f d, z and you have g c, a and then whatever you have this now.

So now this stop here itself you have term which consist of constant or something like that variable and now, this changes to c and a and this goes to d and Z. So, this is the formation tree for this particular kind of term whereas, the formation tree for this 1 is considered to be this 1. So, that is a reason why note this 2 terms are considered to be different because, they have different kind of tree structure a tree formation tree.

So, it is for this reason that every term will have its own unique formation tree. So, now we talked about how to draw formation tree for a term. And then 1 important observation that we made is this that, if it is all it is considered to be ground term it will have only constants; it does not have any free variables at the leaves. If it has free variables at the leaves at the part of leaves, then it is not considered to be a ground term.

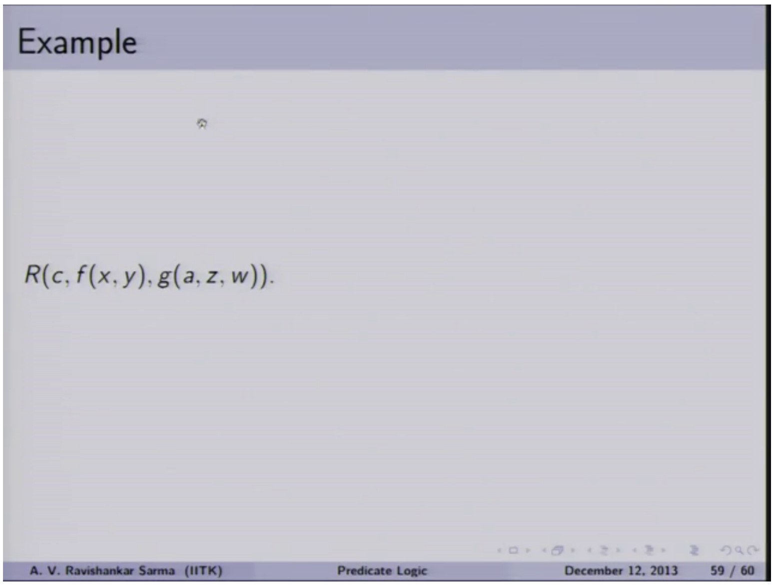
(Refer Slide Time: 22:31)



So, now, let us consider the definition of... So now, we have we talked about the formation tree for the terms. Now let us talk, about something about atomic formula auxiliary formation trees. So, what are these atomic formula auxiliary formation did they are the labeled, ordered, finitely, branching trees again as is the case of the 1 which have seen earlier of depth 1. It is depth 1 because, is all atomic formulas and whose root node is labeled with some kind of atomic formula and if the root node of such a tree is labeled with n array relations like r of t1 t2 to tn.

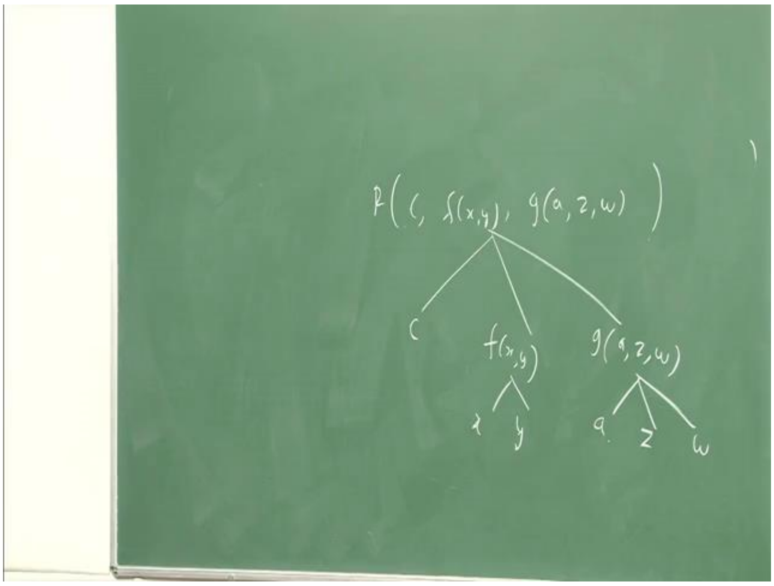
Then, it has an immediate successors which are labeled in order with the terms t1 to tn that is first thing that you need to note. The second 1 is, this that an atomic formula formation trees are finitely, branching labeled, ordered trees, which are obtained from some kind of auxiliary trees by attaching at each leaf labeled with some kind of term t and the rest of the formation tree are associated with another term tree. Such tree is associated with the atomic formula with which the root is considered to be labeled.

(Refer Slide Time: 24:08)



So, now let us considered some examples of atomic formation trees, so then we will understand this particular kind of thing the definition that we have discussed earlier. So, what I am exactly trying say is this that a given term will have its own corresponding formation tree and the same an atomic formula will have its own corresponding formation tree.

(Refer Slide Time: 24:27)

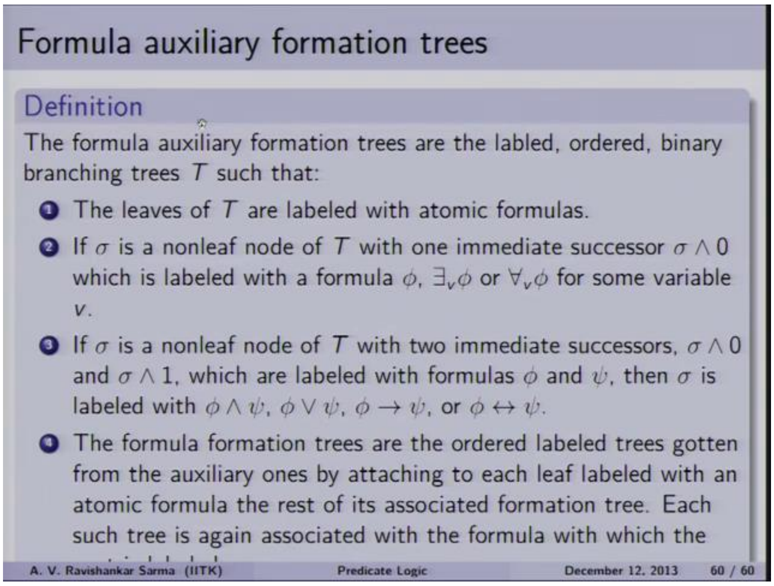


Let us consider an atomic formation tree and all like this c, f of x, y and g of let us a, z and w. So, this is considered to be an atomic formula. So, now the first 1 it has this 3 leaves now the first 1 is c and then followed by that x and y and g of a, z, w. So, now this ends here itself and then goes to x and y, and then you have 3 letters out of these 1 is some is some are constant, some are variables and then a, z, w.

So, this is considered to be the atomic formation tree for this particular kind of formula. So, these are atomic kind of well formed formulas and all. So, in the context of predicate logic we have 3 different things first is a given term will have his own corresponding formation tree. And then the second one is atomic formation, atomic formula which will have its own corresponding formation tree and the second 1 is anything which is considered to be a formula.

A formula in predicate logic is simply like this that it will have at least 1 free variable. If it has no free variable, then it is considered to be close kind of formula. Or it is also considered to be a sentence in the predicate logic; all those definitions which we have considered in the last class.

(Refer Slide Time: 26:11)



So, now we discussed about the formation tree for term and then when we have no free variables in that particular kind of formation tree then we called it as ground term otherwise, it is not a considered to be ground term. And then we discussed about atomic formulas and then it is own corresponding trees. And then let us, talk about some kind of formulas which exists in the predicate logic in general.

Then, its starts with there exists some x for all x etcetera and all starts with the quantifies. So, now the formula auxiliary formation trees are again considered to be labeled, ordered, binary branching trees, that tree is labeled as T where, the leaves of T are labeled with atomic formulas ultimately at the end of you are formation tree. All these things are considered to be atomic formulas.

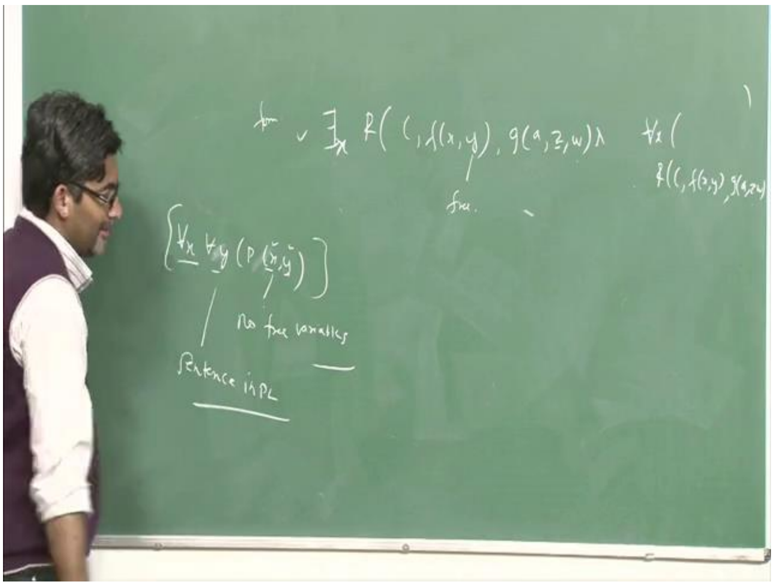
So, now if you consider sigma as non leaf node that is what going to extend to leaves node of T with 1 immediate successor. So, that is let us considered at sigma conjunction 0 which is labeled with a formula either, it can be simply a formula phi or a quantifier followed by a formula it can be existential quantifier or universal quantifier for some particular kind of variable v.

So, that is what is the case in the first instance and suppose if sigma is considered to be a non leaf node again T with 2 immediate successors. That is sigma conjunction 0 and sigma conjunction 1 that is stands for 2 immediate successors, then which are labeled with formulas let us say phi and psi. Then, that sigma has to be labeled with either the conjunction of these 2 formulas, that disjunction of these 2 formulas are implication or double implication, so there are things which we can have.

So, this just tells us that you know how to form this formation tree for a given kind of formula. Just we are giving some piece for the analysis for how to form this particular kind of formation tree. So, the formula formation trees are again considered to be ordered, labeled trees gotten from auxiliary 1s by attaching to each leaf labeled with an atomic formula and the rest of its is associated with the formation tree.

So, each tree is again associated with the formula with which it is marked, which is labeled. So let us, consider some examples for these formula auxiliary trees then we will see the difference between let see, so now 1 particular kind of example which we will be talking about, which where we talk about formula formation tree.

(Refer Slide Time: 29:24)



So, its starts with thing auxiliary formula formation tree and all easily starts with there exist some x we started with terms and then atomic formulas and then we have these things. Given formula in predicate logic it will have its own formation tree. So, now quantify followed by some kind of atomic formula f of x and y and g of a, z and w conjunction for all x.

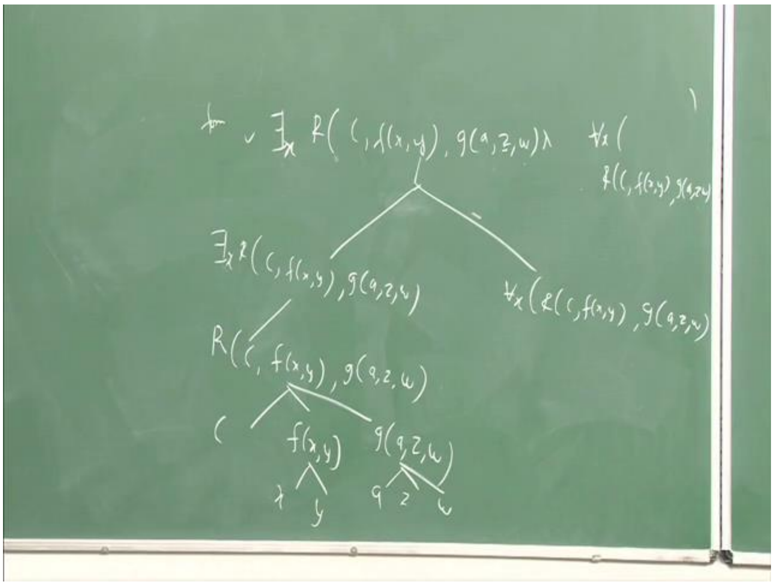
Let us consider, another term such as this thing r c, f, x, y g a, z, w is a conjunction of these 2 formulas. There exist some x r c, f, x, y a etcetera and all for all x or c f, x, y etcetera and all. First you need to note that, it is not considered to be close formula; a close formulas will be like for example, for all x for all y p, x and y. So, this is considered to be a close formula, because it does not have any free variables no free variables.

So, that is why it is considered to be a closed kind of formula. So, now in this case with respect to x this y is free and even other things such z etcetera there also free a formula in that context this is considered to be a sentence in predicate logic. So, where as this particular kind of thing is considered to be formula and predicate logic. Because, it has free variables y, z and respect to this a conjunction with respect to this quantifier we have free variables such as x etcetera.

So, in that context this is considered to be a formula, but not a sentence in the predicate logic. A sentence will have a no free variables; all the variables that existing your formula are bonded by the given quantifiers. For example, x and y are both founded in this particular kind of formula with respect to for all x and for all y with respect to these 2 quantifier x and y does not have any freedom.

So, now we are trying to form a formation tree for this particular kind of formula. So, this will have quantifiers followed by atomic formulas, and then quantifiers followed by some kind of atomic formulas.

(Refer Slide Time: 32:06)

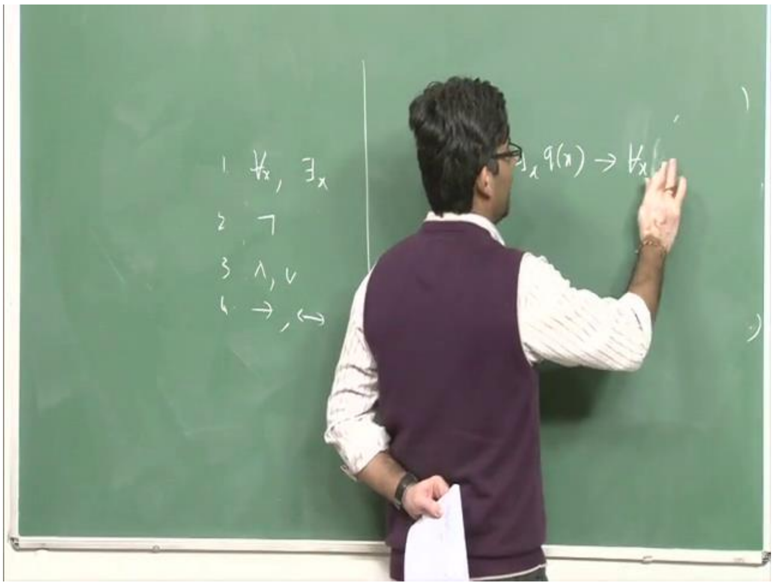


So, now this will have this thing first we need to write this thing R c, f f x, y, g a, z, w and the second 1 is this 1 for all x R c, f x, y, g a, z, w. So, now left hand side can be expanded to this thing. So, now we have to eliminate this quantifier and then this will be not eliminating. So, this the way which to form the trees and all x and y g of a, z, w will be expanding only the left hand side of this thing; right hand side you can expand it by using same kind thing.

So, now it will be like this further expanded to f of x and y and then g of a, z, w. So, this further expense to this 1 x and y and then g expands to a, z, w. So, in this way we can expand this particular we can construct a given formation tree for a given formula. So, only 1 needs to notice is this that each and every given formula will have its unique formation tree. So, with that particular kind of thing the important thing which will noticing is this that we can if we have a formation tree, then need have worry much about the parenthesis.

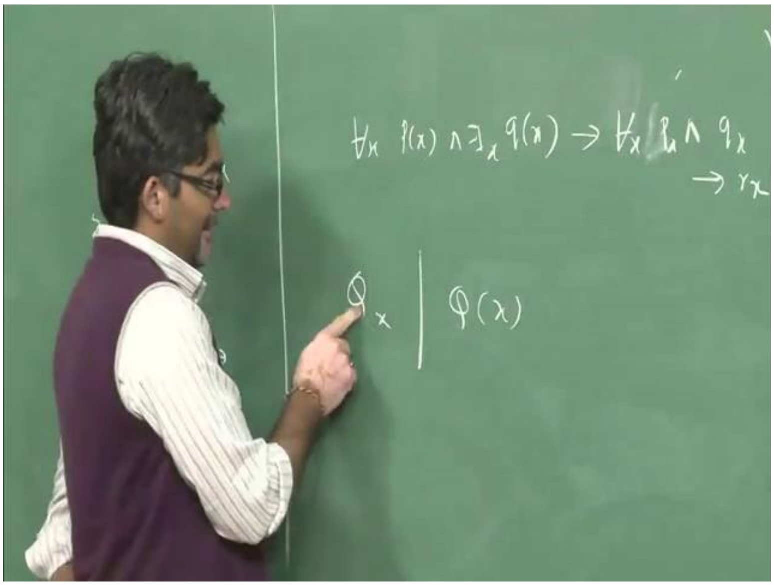
For example if you take particular kind of formula like this. So, usually this is the convention that we will be following we have consider some examples earlier and the preference ordering will be given like this.

(Refer Slide Time: 34:17)



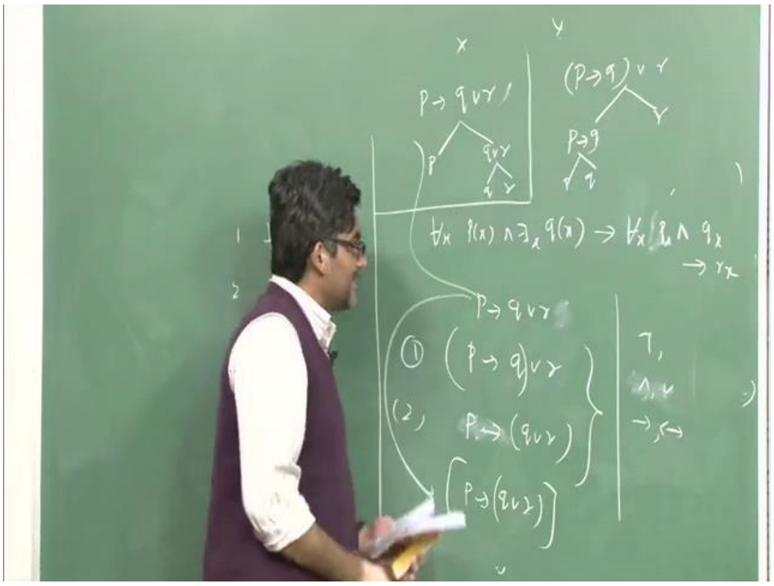
First preference will be given to for all x and followed by that there exist some x and then negation, and then followed by that you have disjunction convention and disjunction And then implication and double implication and all for example, if you have formula like this thing like you do not have any parenthesis or anything. Then let us see, how to write the particular kind of formula x q x implies for all x p x and q x etcetera.

(Refer Slide Time: 35:19)



So, then if you have if you are given formula like this thing where there no parenthesis which are given here let us you can consider R x also, where p, q, r are all predicates and then followed by then there are variables know. Sometimes I will be writing like this and you can even write like this also, in some text books Qx is written in this sense I will be writing in this particular kind of way. So, now suppose if you are given a formula like this, then where there are no parenthesis here.

(Refer Slide Time: 35:52)



So, where if there are no parenthesis the problem as in the case of propositional logic. For example, if you have a formula like this p implies q or r and this formula can be read in varieties of way is it the case that p implies q or r. So, I am writing it like this or is it the case that this 1 way of reading this particular kind of formula and the second way of reading the formula is this thing p implies q or r this 1 way of reading it.

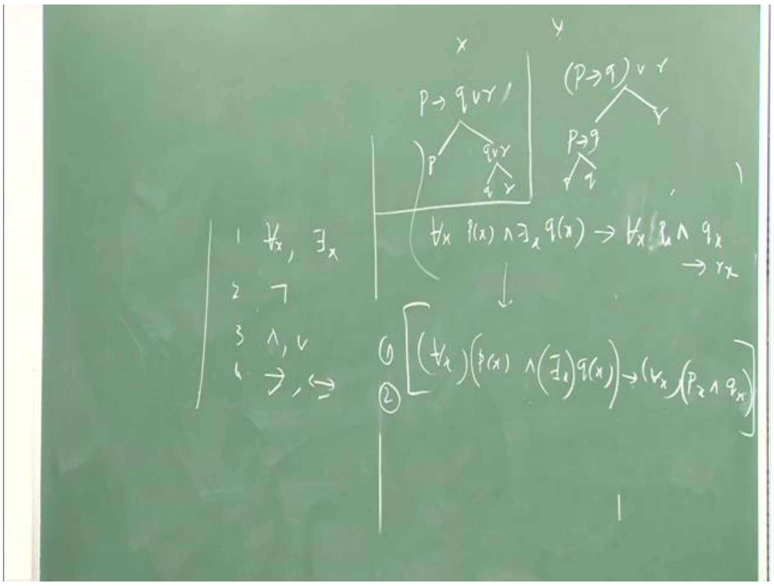
So, there is confusion of how to read this particular kind of formula and all. So, for that we have we used some kind of convention with which you know we have come up with this thing. So, in the case in the context of propositional logic we have this thing conjunction or disjunction, implications and if and only if, so now if you are given a formula like this p implies q or r.

So, now we have only disjunction and the implication here. So, the first preference out of this implication and disjunction is this 1. So, that is why you put bracket here q or r and then you put bracket on this 1 the next 1 is per implication. So, this formula should be read as p implies q or r and you can draw formation tree for this 1 it will be like this p implies q or r.

So, now it will be like p and then q or r so now, it will be like this 1 q or r. So, this is the formation tree for this particular kind of thing. Suppose, if you draw if you have a formula like this thing p implies q or r, then the formation tree for this 1 is like p implies q and r p and q. So, this x and y the formation tree for these 2 things are looking quite different and all.

So, that is why each and every formula will come of its own formation tree. In the same way for example, in this particular kind of formula the predicate logic formula first we need to do what we need to do is, we need to follow this particular kind of preferential kind of ordering; for all x there exist some x negation and or implication and this 1.

(Refer Slide Time: 38:37)



So, now in this particular kind of formula, so wherever for all x is there we have to put in brackets that is the 1 first thing which we need to do and that the first step you will do this thing. And then we will follow the second step later for all x p x and q x is the first step and the second step is this that you need to give preference to there exist some x. So. now this will remain as it is expect that you need to put bracket here, and then there is no existential quantifier to worry much now the next preference is given to and.

So, now so this will be like this, so for all x p x now for this 1 we to put bracket here this will become p x and q x and wherever end is there here you need to put bracket here p x and the whole formula. Next, this the formula which is closer to this particular kind of thing within the scope of this quantifier you put brackets here. So, now this is what is going to be the case since there are no implication; implication is there here.

So, now this formula will become like this for all x p x p x and there exist some x q x implies for all x p x and q x now, it will have his own formation tree. So, I think we will end this lecture by saying that what we done in this lecture is this that, each and every predicate logical formula comes up with its own formation tree to start with we have contracted a formation tree for term.

Then we said that if there are no free variables in that particular kind of term we said that, it is considered to be ground term. And then we formed we constructed well form we constructed formation trees for the atomic formulas and then obviously, we have when atomic formulas are combined, will have compound kind of formula auxiliary kind of formula.

Then, we talked about formula auxiliary formation trees by constructing its own formation tree in the next class what will be doing is this that. So, what will be doing is will be talking about the symmetric of predicate logic; symmetric of predicate logic is not as simple as the case of the propositional logic. In the propositional for example, if you want say that if you want to talk about the truth value of grass is green and moon is made up of green cheese.

What you will be doing is simply this, that you have the symmetric of conjunction, so a conjunction is going to be true only, when both the conjunctions are true. And if you just know the truth value of these 2 individual propositional variables, this is a sentence which is represented by propositional variables if we know the value of p what value p takes, what you know what value of q takes and all.

There another you can talk about the truth value of a given compound sentence. Whereas, this not as simple as in the case of propositional logic; in the case of predicate logic we have variables we have constants, we have functional symbols and we have predicates. Then, what is required is we need to go into the details of deep structure of these particular kinds of thing, where we need to assign some kind of things to this variables constants etcetera and all, constants predicates etcetera.

Then, in that context we will be talking about something called as interpretation; interpretation is nothing, but assigning some kind of values to some kind of things to this variables called constants, predicates and functional symbols within the domain. And other important thing is this that, it does not make any sense to talk about truth value of a given predicate logic formula without referring to any domain.

So, we need to talk about I need to fix the domain, then only we can talk about truth value of a given predicate logical formula. It does not make any sense to talk about whether or not given formula is true without referring to any domain and all. So, whatever is true with respect to suppose if you take the domain as people, then if you take the domain as for example, real numbers or some other thing it might change you know.

So, the interpretation might change, then will be talking about the certain things which are considered to be true in all structures and we call it has tautologies and all. So, all these things which will be talking about in the next class where we will be dealing with the symmetric of predicate logic.