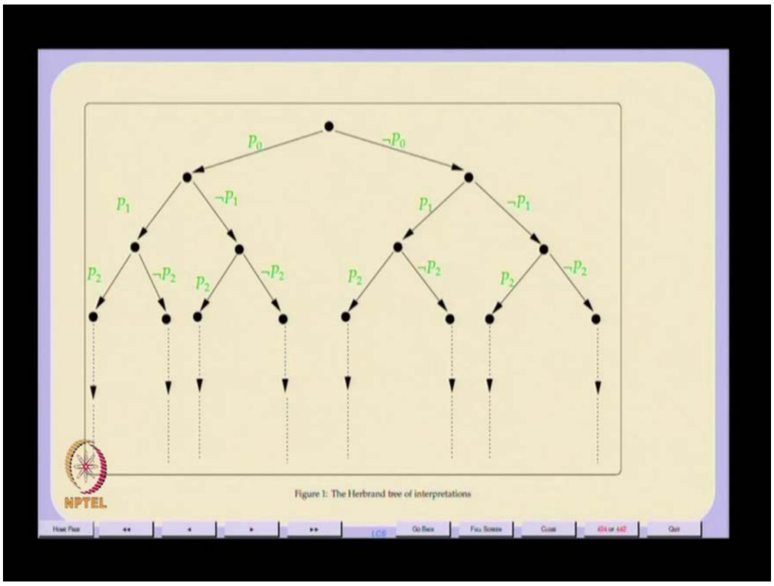
Logic for CS
Prof. Dr. S. Arun Kumar
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi

Lecture - 27
Substitutions and Instantiations

So, let us start. So, we did Herbrand's theorem. So, let us actually go back little bit and recap those things. So, actually let us look at this.

(Refer Slide Time: 00:45)

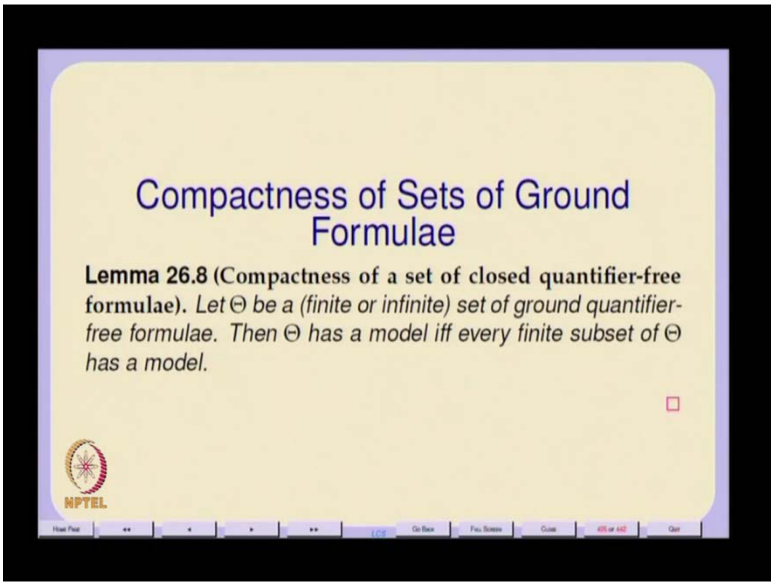


So, we had this Herbrand tree of interpretations and essentially what we what we're saying is that you take. Firstly, our Herbrand interpretation. So, this is the set of all possible interpretations Herbrand interpretations. So, which essentially means that you take any theory that you want you do not have any concrete representations the only concrete representations that you have are in the terms of terms defined on that signature and these p naught p one etcetera are essentially atomic predicate symbols applied to ground terms. So, which. Firstly, implies that you do need at least constant. So, we're are always talking about a signature which if it comes without a constant then we extend it expand it to include a constant at least.

So, that there are ground terms always the set of ground terms would be empty. So, you have a non-empty set of ground terms and. In fact, if you have even one constant then if you have even one constant symbol then you have an infinite collection of ground terms actually and those ground terms applied to the atomic predicates symbols of appropriate arity enumerated enumerate all those grounds formulae ground atomic predicates and this these capital p naught p naught naught p one and so on and so forth essentially signify ground atomic predicate formulae ground atoms and all possible truths assignments for these.

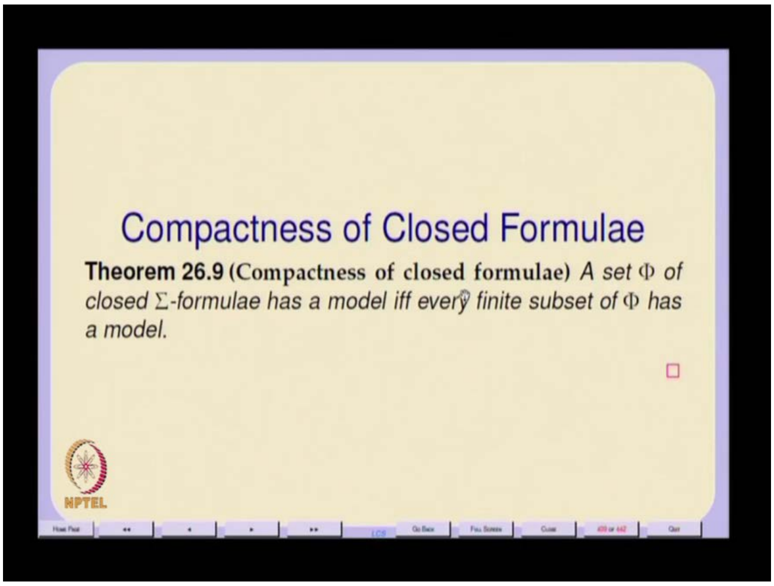
So, this tree in infinite as of course, you have only a accountably infinite number of them you have only a countable you're language of terms itself produces only a accountably infinite collection of terms and you anyway have only a finite number of atomic predicate symbols. So, the number of ground terms that you ground atoms that if you have is only countably infinite it cannot be more than that. So, this. So, the herbrand tree of interpretations essentially gives you all gives you power set of that yeah. So, it gives you all possible interpretations in the term algebra itself for all possible atomic predicates right.

(Refer Slide Time: 03:24)



And what we showed we showed the compactness which notice that this compactness is only for a set of closed quantifier free formulae here, and later it is only for closed formulae yeah compactness of closed formulae.

(Refer Slide Time: 03:38)

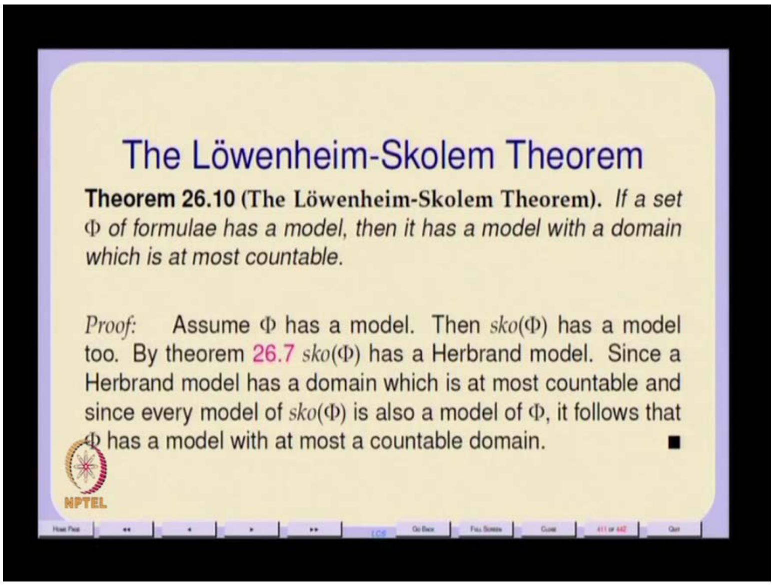


So, we are not really talking about formulae which have which are open and which have free variables and so on and so so and what we are are saying is that a set of a set phi of closed sigma formulae has a model if and only if every finite subset of phi has a model in this phi can only be countanbly infinite because we are constrained by a language. So, it cannot be an uncountable set of formulae for example.

So, it is only a countable set of formulae and there is there is one possibility which of having things uncountable which I am never which I have not actually ever mentioned the only way you could have a possibly uncountable number of formulae is actually if you're signature sigma has an uncountable number of atoms or atomic predicates symbols or function symbols if it does or it simply if it just has an uncountable number of constants take the real's for example, I think of every real number is a constant then I have a signature over the real's which is actually uncountable.

In which case then the number of terms that you have is actually uncountable which also means that you could therefore, have a an uncountable number of formulae, but let us not get into that there. So, so actually, so we. So, in theory actually the set phi could be uncountable and provided if you have an uncountable number of terms for example, and you can have an uncountable number of terms for example, if you have a uncountable number of constants in your signature, but let us not get into that we will stay with countability and our infiniteness will always at most countable right.

(Refer Slide Time: 06:03)



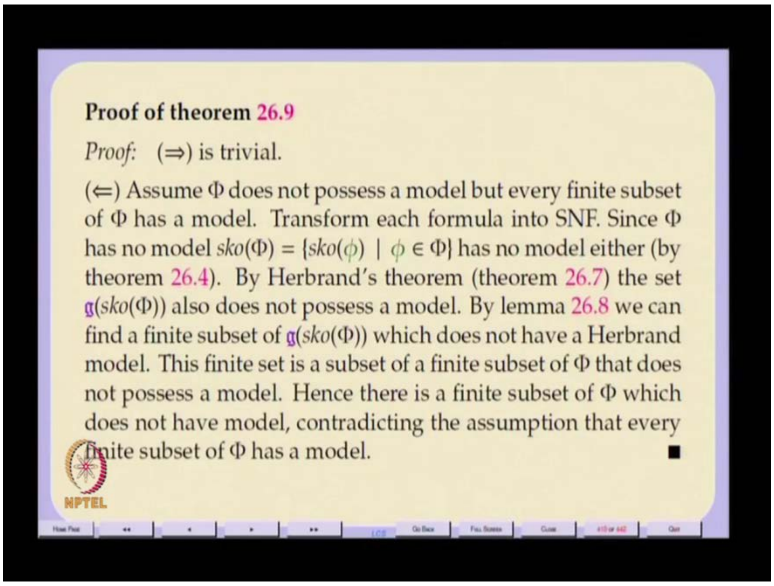
So, we proved the compactness theorem. Yeah.

Student: This how do we move from that g four phi to the finite set of sorry phi. This is this is because of the previous lemma right.

This is, but from the we found out a finite subset of g equal to phi g of whole phi and then the next line we claim is that there is this finite subset of phi which is not satisfied.

Student: This was it is…

(Refer Slide Time: 06:41)



This were you are saying is a formula.

Student: no no line after by lemma next sentence.

This finite set is a subset of a finite subset of phi that does not possess a model if it is a finite set then anyway it is consists of ground terms it is given that it is a finite set it should be a. So, there should be only a finite a finite number of. So, so this is after this is a finite set then you’re saying that this finite set need not be a finite set of capital phi.

Student: we are saying that how how come this pin terms can you explain it a little bit or it is just a finite set of g four phi or it is a subset of some finite subset of phi.

Ah ok let us let us look at score phi score phi is obtained from capital phi by solemnizing each of this solemnizing finding solemn functions for the existential quantifiers there are anyway all only a finite number of them right and when you when you find the set of all ground instances of these all you do is you’re replacing all the variables basic now after having solemnize they are all just universally quantified variables function universally quantified predicates and you’re instantiating those universally quantified predicates with ground ground terms and you’re g sko phi is essentially this and all this is countable at most all this is countable at most and with assumptions that it does not possess a model means that there is a finite subset which does not have a model and therefore, it means there is a finite subset of g sko phi which does not have a model.

But that g sko phi that finite subset of g sko phi comes from some finite subset of atoms in of phi that is it from some some finite subset of formulae in phi and there is just some ground instances of those some finite set of ground instances of those that is how you can get a finite subset you cannot get a finite subset otherwise.

And and if that finite subset does not have model then it means that you lift it back to phi there’s a finite subset of phi which does not a model right. So, that is. So, that what happens here hence there is a finite subset of phi which does not a model which contradicts our assumption that every finite subset of phi as a model.

Student: You could have g phi.

No no think think of what happens when you go back from g sko phi. So, you take a finite subset of g sko phi which does not have a model when you go back from g sko phi to phi or when you go back from g sko phi to find the appropriate elements in sko phi.

What do you have you have all universal quantifiers yeah you have all universal quantifiers and you have found a you have found an instance for which the universal quantifier is not satisfied. So, that finite subset of sko phi is not satisfiable because it is all the all the variables are universally quantified that sko phi would have come from a finite subset of phi and. So, what you're saying and that might have its existential quantifiers, but if there isn't what we know if there isn't a model for the ground term then there is no model for that subset too right so.

So, that is how this works out yeah. So, a simple therefore, the interesting thing now about since I was talking about countability and uncountability. Now, therefore, it is therefore, very clear that is there is this important theorem by lowenheim-skolem which says that if a set phi and this set phi could be infinite has a model then it has a model with the domain which is at most countable right. So, set phi of course, again this set phi satisfies the same constraints that I have not mentioned it here as the compactness theorem. So, which means phi is going to be closed sigma formulae there are no free variables right and. So, what we're seeing now is that for if you're limited therefore, what the lowenheim-skolem essentially says is that if you're limited to a signature which give you at most the countable number of terms ok.

Then any model of of a set of closed formulae of accountably at most countable of enumerable set of closed formulae it is not necessary to look among the uncountable sets it is sufficient to look within the countable sets which essentially means that therefore, you you need the limitation of a finite or a countable sigma to produce only a countable number of terms is not actually a limitation for closed formulae the moment your formulae become open then the whole theorem come this right.

So, as far as closed formulae is concerned which means that if you're looking at very general purpose theorems even about things like the real numbers without the assumption of an uncountable number of constants for examples if you're looking at very general theorems which had to be which have to be let us say universally quantified formulae.

Then they do have denumerable models if those theorems are true right. So, they do have denumerable models and it is not necessary to search through an uncountable set and those denumerable models are essentially constructible from your signature yeah and. So, so let us just quickly look at the proof this is actually a very simple proof, but it it is phylophisically it it has a great deal of influence I mean what it shows is that they are not particularly limited even if you are talking about the first order theory of the real's.

If you're looking at the first order theory of the real's and you're looking for theorems which are of a universal nature then you're not limited by a language which is only countable and therefore, cannot represent all of the real's yeah that is that is the that is actually the important consequences of this.

So, let us look at the proof. So, assume phi has a model right then sko phi has a model this solemnized version of phi has a model by theorem twenty six point seven what is it. So, this is which has a herbrand model there is basically herbrand's theorem. So, it has a model if and only if it has a herbrand model and every herbrand model has a for a for a finite countable signature it has only a of enumerable terms right and and the terms the term language itself is the domain right and that is countable that is a countable set right and. So, if it has a model if and only if the sko phi has a model and that is at most the countable number of formulae and therefore, any this set of closed formulae phi even if it is countably infinite it has a if it has a model then i'am not constrained if I just look for terms within the herbrand universe yeah and can find I can find a model essentially in the herbrand tree of interpretations right so.

So, therefore, phi has a model with at most a countable domain there are there are extensions of the lowenheim-skolem theorem actually to uncountable signature and finding uncountable models and. So, on. So, for, but I think there're not of a practical significance really in all our main concern where when we look about look at unaccountability is really things that the real the complex numbers.

So, what happens in the first order theory of real's what happens in the first order theory of complex numbers there the theorems are not open they are always closed formulae and you're looking at and if you have to find examples or counter examples which is what models are about right examples are counter examples then you're not constraint by the fact that you have only countable language or countable domain you can construct your herbrand terms for the models or counter examples from a countable language it is a form of countable construction so.

So,. So, what herbrand's theorem and therefore, solemnization lowenheim-skolem theorem all of them they essentially says that for any theory that any first order theory at least that i'am interested in I do not need to have any concrete models I need to have only a language of terms if i;am interested only in the in the in the theorems of closed formulae right. So, so one. So, these. So, these the fact that you need to limit yourself only to a language of terms essentially therefore, it makes it possible to look at first order logic theorem proving and therefore, logic programming and. So, on and. So, for as feasible entities which you can work with just in a linguistic frame work you do not actually require full mathematical frame work you have a simple linguistic frame work and within which you can do your programming you can do your theorem proving and for most of the closed formulae not necessarily the open formulae.

So, for open formulae of course, you're there're going to be limitations and. In fact, the expressibility of various theorem is going to be there, but closed formulae including existential formulae actually can be done entirely within a countable language which can be freely generated by a grammar that is that is actually the model of the story and which the whole concept of theorem proving or logic programming first order logic actually comes right.

Student: In this. So, in this yeah.

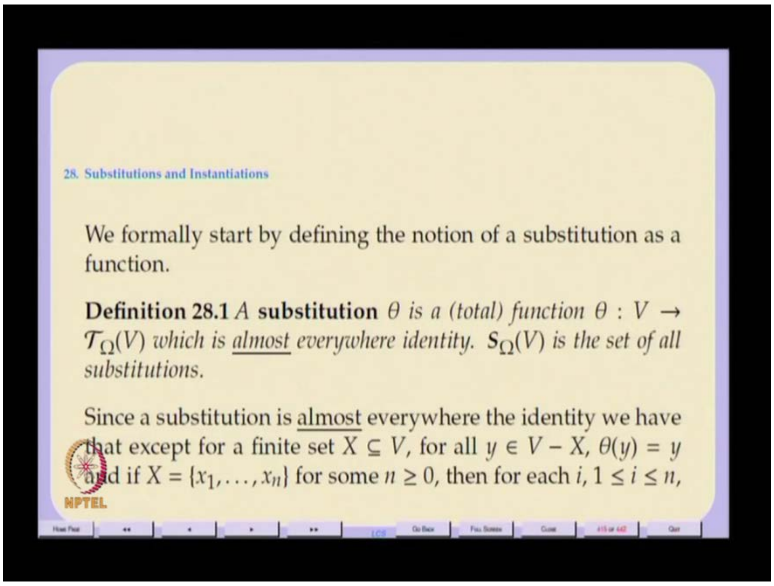
In the theory closed formulae.

Yeah this phi is of closed formulae and if you have open formulae you get into trouble. So, actually what happen what. So, what happens is that if you have open formulae then your that herbrand tree interpretations is not sufficient for all the ground terms you know for all the ground instances this is this is sufficient just for the closed formulae. So, in this so.

So, what it? So, if you're going to do things completely linguistically then clearly the one of the most important things that we need to worry about is a notion of substitutions, and basically you look at your ground instances so and so all of them actually deal with substitutions. So, what I'll do is I will quickly go through the theory of substitutions and take it for granted in future basically so and this. So, this lecture basically on substitution instantiation which is going to be independent of logic because the notion of substitutions instantiation is extremely general and whatever we say here is done from a purely universal algebra frame work and of which your first order logic language is also just a certain term algebra in the universal algebra frame work right so.

So, it does not matter whether we are taking the lambda calculus the combinatory logic or first order logic or whatever or even propositional logic with substitution of meta variables by propositions you take any algebra essentially any term algebra, and you need to perform substitutions and first order logic is no exception so. So, we will look at the substitutions in substitution theory in essentially a completely a general framework completely removed from first order logic and as far as this theory is concerned it is just first order logic is just another term algebra in which the theory of substitutions can be applied.

(Refer Slide Time: 21:53)

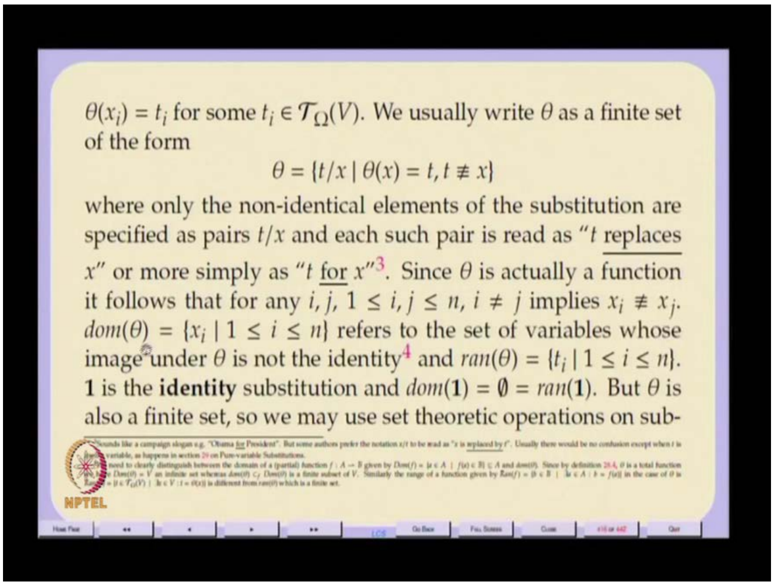


So, that is. So, so the slides are not completely well prepared for this lecture because it is a it is a write up which I have imported straight into this into this thing and for my for the purpose of this lecture. So, so you will see certain discontinuities on which I will clean up later.

Substitution theta as far as we are concerned is a total function in fundamental to everything every term algebra p which also includes first order logic is the set of variables v right and a substitution is just of a total function I am here I am a total function of the set of variables also the set of terms and this is a here the important thing here in the case of the substitution is that it is almost everywhere the entity function, which means that you have a you have an infinite collection of variables v, but there are only a finite number of variables which you're actually replacing by terms.

All other variables just retain their identity. So, the effect of this there is no effect of the substitution to all other variables. So, that is what I mean by almost everywhere identity this almost everywhere is a is a technical term used in first order logic to say that used in second order logic to essentially say that in all except a finite number of cases yeah so. So, now what we are saying is there are only a finite number of variables which are actually replaces by some terms which are not the same variables themselves. So, this will call the domain of this substitution by the you should not confuse this.

(Refer Slide Time: 23:55)



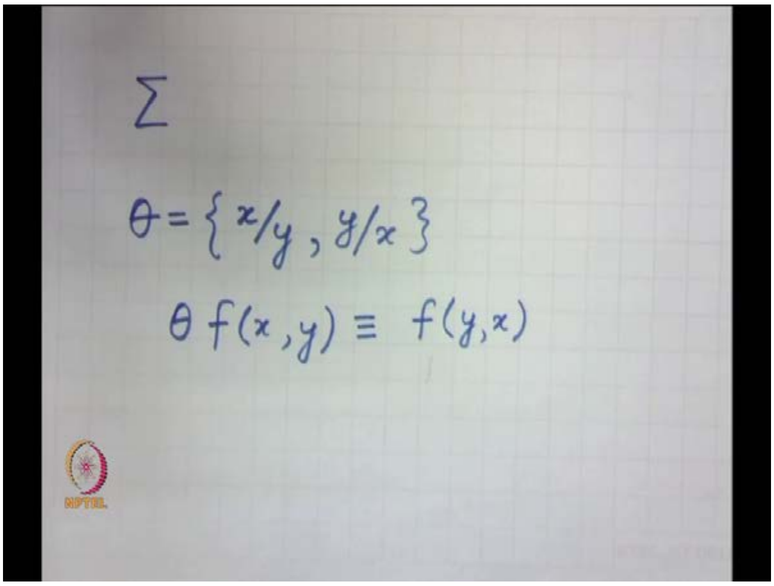
You should not confuse this dome this is this is some dome right wherever it wherever written so.

Student: For many of the signature.

Omega is signature yeah. So or. So, I use this dome small dom to denote only those variables which are being replaced by other terms other than themselves, so all those variables y for which theta y is just y itself I am excluding from this. So, remember that theta is a total function. So, as a function its domain is the whole set of variables and it. So, domain is the whole set of terms t omega v, but this small dom essentially give you the finite set of variables x I such that theta of x I is different from x I yeah. So, that is what here. So, so look at this a theta as represented as a finite set t for x where theta of x where t is different from x yeah. So, that is that is a theta and corresponding to dom you also have this concept of a range here. So, range is just this this set of terms t which act as the image of the variables which are actually being replaced. So, there is this identity substitution the identity substitution of course, is the identity function on from variables itself and therefore, its domain and range are both empty is dom and range above the empty here in that the small dom and range is ok.

in general let us let us keep this in general because it is not relevant to first order logic, but you may be you should read it we had this usual notion of admissibility of a admissibility of a substitution for in a term or in a predicate yeah. So, for all the substitutions that we considered in first order logic were substitutions of essentially a single variable by a term, but what I am doing here now is I am generalizing that notion to a set of possible variables by a set of possible terms and these are what this is what is known as a simultaneous substitution.

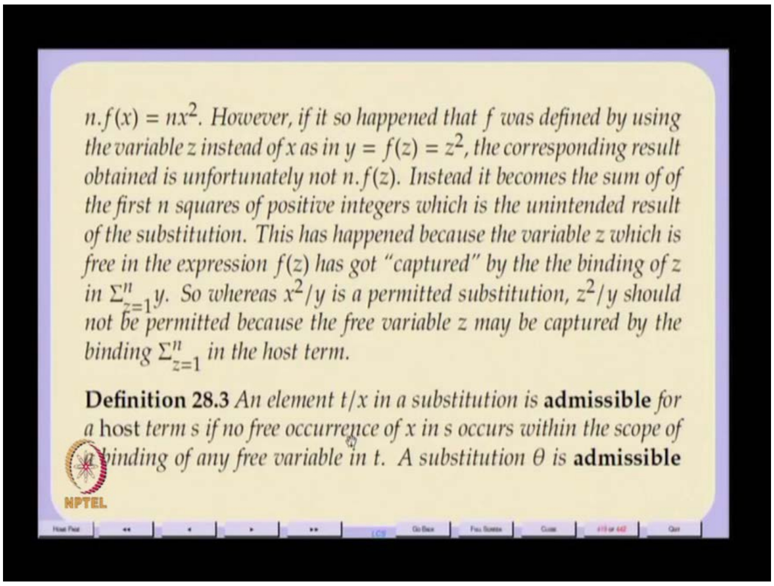
(Refer Slide Time: 26:40)



It is quite possible that I have a substitution theta which replaces x for y and y for x right ok.

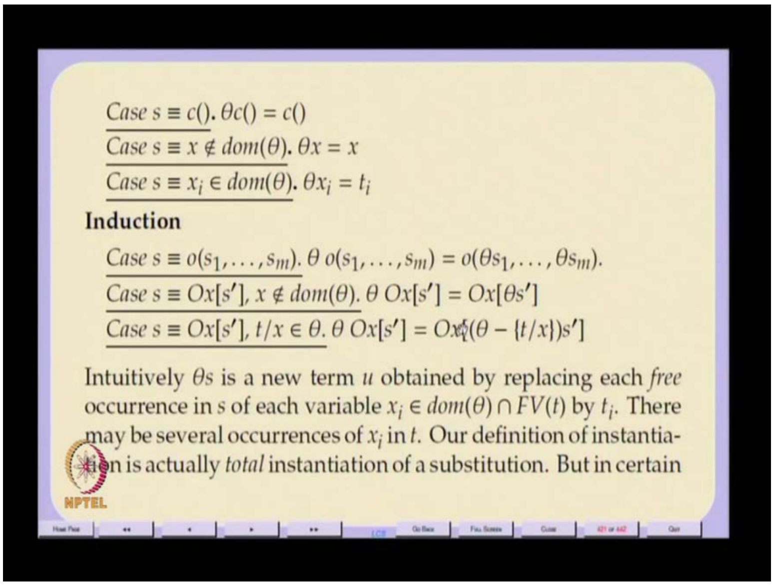
So, if I have term let us say f of x y then the notion of a of simultaneity of application essentially means that theta of f of x y is equal to f of y of x is simultaneous and not sequential there is no particular order it is actually simultaneous. So, you you look at all all the occurrences of variables which have to be replaced and replace them simultaneous. So, that where as if you if there was some sequence you applied here then you will you will most likely get f of x x or f of y y you will not get f of y x know you will not be able to switch terms right.

(Refer Slide Time: 27:45)



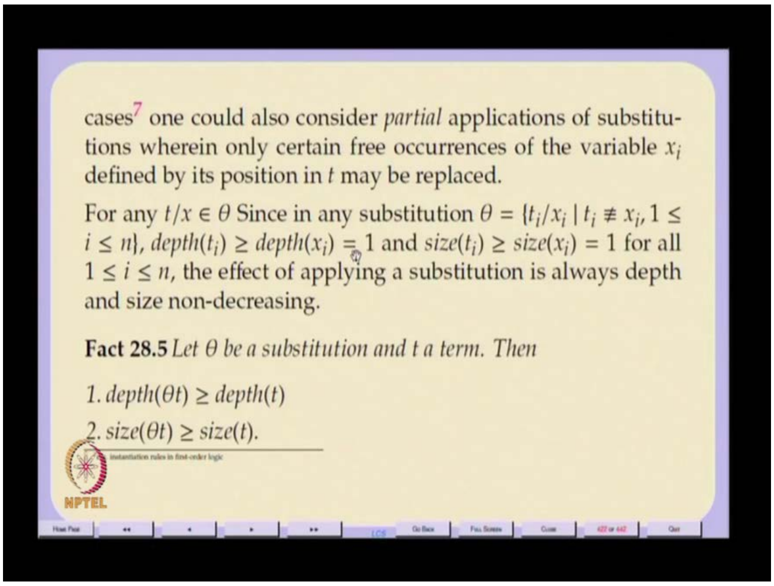
So, this the notion of simultaneity is important and the notion of and along with the notion of simultaneity also comes the notion of admissibility which needs to be generalized from a single term term for variable substitution to a set of possible substitutions simultaneous substitutions so, but the basic principle in admissibility is still remains the same you do a simultaneous substitution and there should be no capture of free variables that that should be the that is the basic principle on which it is defined. So, in in a more general fashion what we are saying is that a. So, a substitution theta which replaces a set of variables let us say x one to x n by a respective set of terms t one to t n is admissibility if each of them each of the substitutions in that set is a admissible right.

(Refer Slide Time: 28:41)



Then we have the notion of application of theta to s which is what I said what I have given. So, it is it is by by induction on the structure of substitutions it has to be simultaneous, but what I have also taken into account is the possibility that you might have bound variables occurring in terms this is something we are not we did not actually think of it in first order logic in the logic of terms in the language of terms of first order logic they were there was no they were no binding operators I mean and this treatment being general enough has to take into account the fact of binding operators. So, this this capital o x is like a typical binding operator think of it as either a quantifier in first order logic or think of it as lambda x and lambda calculus for example. So, what happens is substitutions. So, it is. So, is defined by induction yeah.

(Refer Slide Time: 29:43)



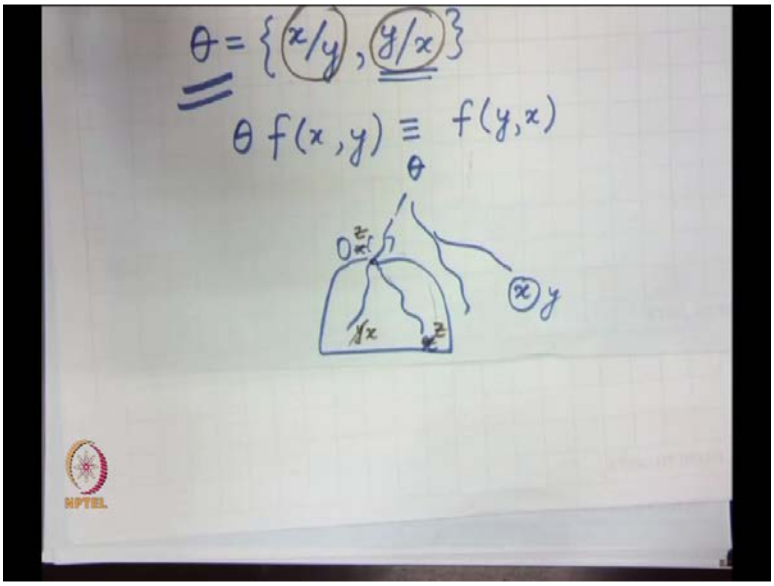
So, the the basic facts about substitutions are that their size not decreasing the effect of a substitution can never reduce the size of the abstracts in syntax tree on which it is applied yeah it has to be either size expanding or the size remains the same the size remains the same when you and this is because you are replacing a single variable which is a leaf note of an abstract syntax tree by a term t which is a tree in itself right.

So, the only way the size can remain the same is if you replace a variable by another variable in all other cases the size will be increased yeah. So, as I said this whole theory of substitution it is it is not a it is meant to be general generally applicable in a universal algebra frame work yeah. So, in. So, in general if you had if you have an operator with its own parameterized on a variable and having a scope which is what happened in the case of quantifiers it happens in the case of lambda and the lambda calculus lambda abstractions then what does the notion of an applying a substitution main in that case basically if any substitution of that bound variable cannot result in any substitution because the variable itself it is just that there is a co-incidence of name in the substitution and in the binding, but actually the two variables are different right in this.

What I am saying is if x belongs to the domain of theta; that means, in theta there is a replacement for x by a term t and I I am applying this theta to binding term of the form some operator x s prime then clearly this x in the binding is different from the x mentioned in the theta and therefore, the effect should be that this bound x throughout does not change in anyway. So, what it means is that you remove that particular substitution t for x out of theta treating the new substitution theta prime as one which leaves x unchanged and apply that to the body of this s prime that is that is all I am saying.

All I am saying is substitutions when pushed in through a binding operator should not affect that bound operator should not affect that binding in anyway yeah they do not have any effect on that binding otherwise for all other variables they actually pushed through right through the terms which in the terms in in when you are looking at abstract syntax trees what you are saying is when you apply it from the top to the root they actually filtered down to the sub-trees it is like applying the substitution uniformly down till you reach the leaves and then make the replace the leaf by a corresponding tree t yeah of course, if that leaf is x and x has a binding ups up there in the syntax tree somewhere like this.

(Refer Slide Time: 33:48)



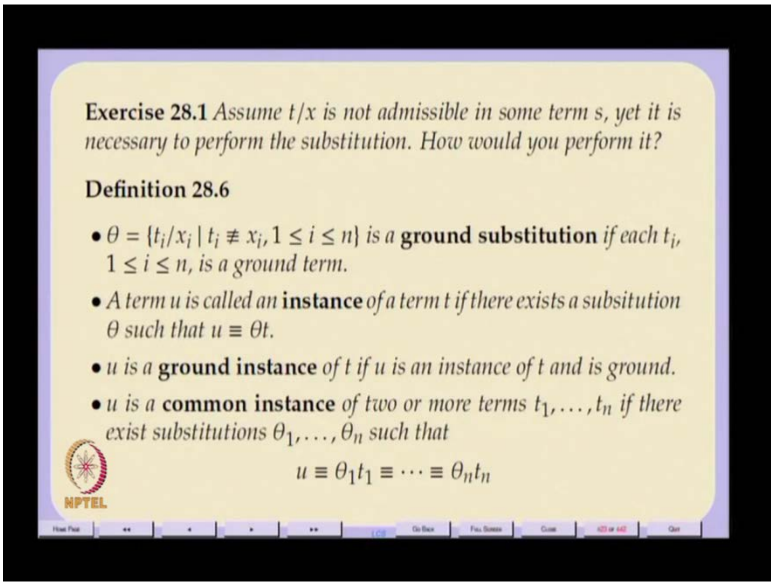
Then what you are saying is. So, let us take the same theta I have some syntax tree with x occurring here and may be a x occurring in a binding. So, basically what we are saying is this entire sub-tree is in the is in the scope of this x. So, if I apply this theta on this tree then this theta at least can have no effect on this x on any of the on on any part of this sub-tree which has x in the leaf theta cannot have any effect.

On the other hand if you look at this. So, there so; that means, this y for x has no effect on this sub-tree because x is bound on the other hand x for y is no is not admissible even if there is no binding for y because of the fact that the by replacing y for x I might be capturing a free a free occurrence. So, this this substitution theta has no effect at all on this sub-tree, but if there is some free occurrence of x here then essentially theta will purpulate down and replace this x by y for example, yeah.

So, these are these are some basic the the important I mean these are not intuitively very hard to understand, but they are something that when you have program them you have to be careful where does the binding take place therefore, how does it purpulate down what is the effect. So, the effect of this. So, here a of course, there is problem of admissibility because of which x for y cannot be done, but you have to take care of both admissibility and bound variables one the the natural thing to do is to use alpha conversion right.

In which case what will you do you will replace this x by let us say a brand new variable z which means this x becomes z here and now this x for y becomes admissible and. So, any free occurrence of y will be get replaced by x that x will continue to remain free y for x because of the fact there is no free x here y for x has yeah. So, so that is those are the things. So, you you we we use alpha conversion quite intensively and all this to give uniqueness of names to distinguish bind bound variables from free variables and so on.

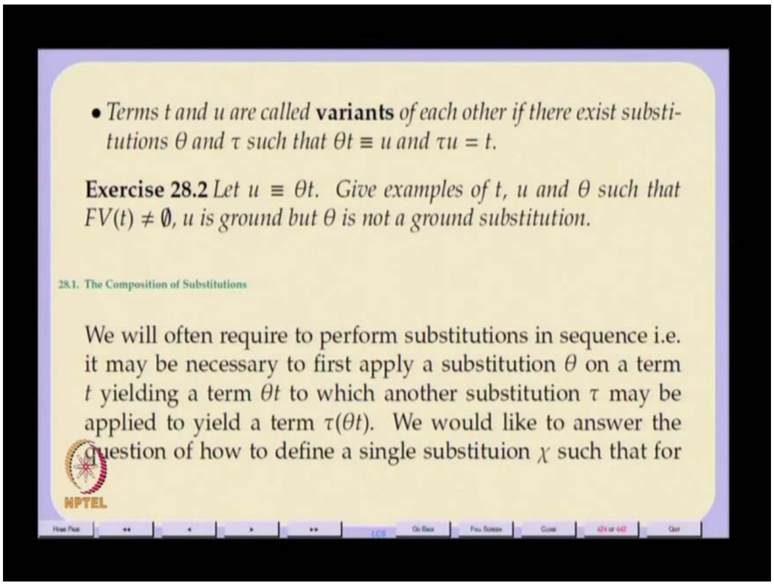
(Refer Slide Time: 37:00)



So, the basic fact of life here is that substitutions are a length and our depth and size non-decreasing and then we have the usual notion of a ground substitution if all the terms which are being. So, if you have a set of substitutions simultaneous substitutions t I x I and all the t I are ground terms when think then it is a ground substitution and you also think of common instances. So, if I can there are two terms are there are two or more terms t one to t n which as syntax trees do not look alike, but if there is a substitution which can be applied uniformly to all of them. So, that the resulting trees look identical then you'll say that this that resulting e u is a common instance of these terms yeah.

I am sorry there is not a single substitution there is actually a sequence substitution there is actually a sequence of substitution there is a corresponding set of substitutions theta one to t n theta n, such that you can make all these trees equivalent to the look the same as u then you say that u as a common instance yeah this is slightly different from that of unify unification which will come to ok.

(Refer Slide Time: 38:25)

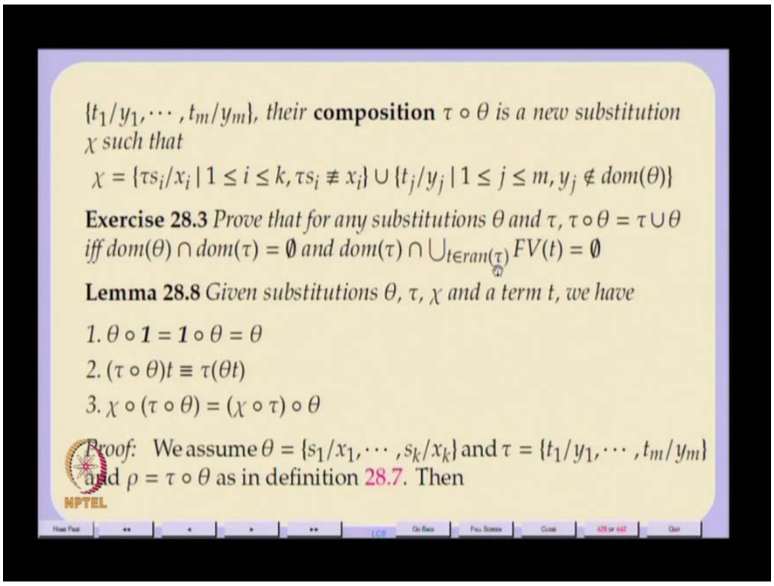


The two terms t and u are called variants of each other if I can find two substitutions theta and tau such that theta applied to one of them gives me the other tau applied to the other gives me the first one right, but more often we will not be looking at two different substitutions wall be what we will be looking at our. So, I can take this I can look at substitutions. So, this was simultaneous substitutions, but I can compose simultaneous to different simultaneous substitutions together in which case I have a composition on substitutions which I need to define yeah.

The effect and basically it is defined by its effect on this on their term. So, so essentially. So, I supposing I have two different substitutions theta and tau and for a given term t I first apply theta. So, I get a new term t prime and then I applied tau on t prime I get a new term t double prime. So, the effect of doing tau after theta is the composition. So, what is the composition. So, what is the composite effect of what is the substitution which is which can be considered the composite effect of two substitutions yeah performed in sequence.

Ah that. So, that. So, essentially this composition operation shows their substitutions of closed under composition and and there a there is a huge amount of case analysis, but let me let us. So, this composition is essentially base.

(Refer Slide Time: 40:05)



So, I have the tau composed with theta I am just using since there like function composition I am just using the function composition operation. So, there is new substitution ki such that now what you need to do is you need to look at the doms of theta and tau and since you are performing things in sequence the you have to actually perform this. So, so you take. So, you separate out those which are not in the domain of theta separate out those variables which are not in the domain of theta which means that any application of theta will not disturb those variables. So, if those variables have to be disturbed they'll be disturbed by only by tau and. So, so this second addend here this second of this union essentially it tells you that all those variables which are not in the domain of theta get replaced by whatever tau specifies for that yeah and if you look at variables which are in the. So, now, look at the variables which are in the domain of theta

So, theta will first replace each of those excise by some term s I that s I will then has its own collection of variables which tau will again replace some of them right. So, that is equivalent to actually taking tau applied to each of these terms s I and replacing x I by tau applied to s I yeah and this is for all the x i's which are in the domain of theta for all the x i's which are not in the domain of theta you just take the appropriate substitution from tau and this is this t j y j is just a substitution from tau yeah ah

And this union because notice that this is a the domain of these two substitutions is disjoint. So, the union is. So, composite substitution and that is the effect of composing tau with theta yeah and essentially what this means now is supposing theta and tau were such that their domains were disjoint then their composition will be exactly the same as taking the union of the two yeah

the domains and range is where all completely disjoint then the composition will just be the union otherwise you will have to apply this tau on each of these s i's that is right now if you look at substitutions then says as an algebra with a composition operation then the substitutions actually form a monoid, yeah. So, you you take any signature omega you take a an infinite collection of variables v. So, and you take the set of all possible substitutions let us say some capital s set of all possible substitutions on this

That itself forms a monoid with this one as the identity element and the composition of substitutions is associative that is something you can prove and you have a very nice structure like this yeah notice that I am making yeah

A monoid is a set which is with a product operation which is associative and there is an identity element. So, the natural numbers are a monoid and radish and for example, yeah with zero being included in the naturals yeah take take any set of take any alphabet the set of all strings forms a monoid with the empty string being identity element yeah it is not abalian the natural numbers and under addition are abalian

Yeah yeah it is not necessary yeah. So, that is. So, the inverse is may or may may not exist actually. So, yeah.

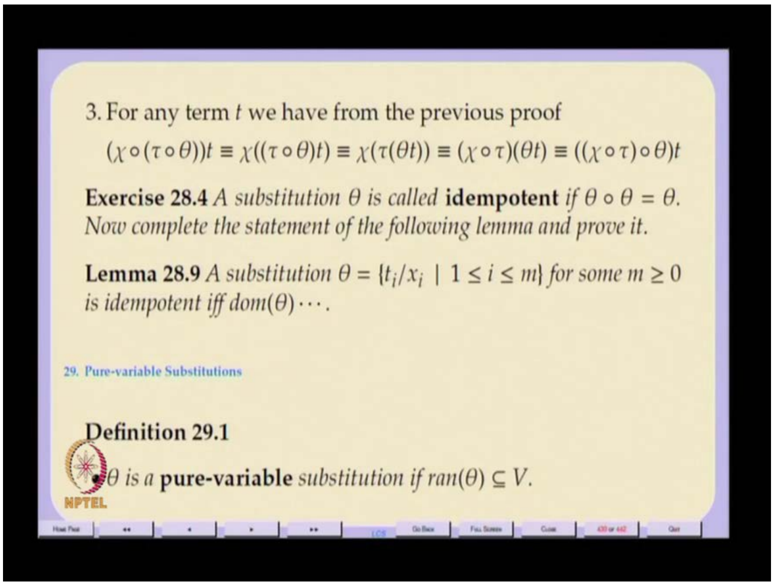
Student: What is the second condition.

This is the associativity.

Student: ok

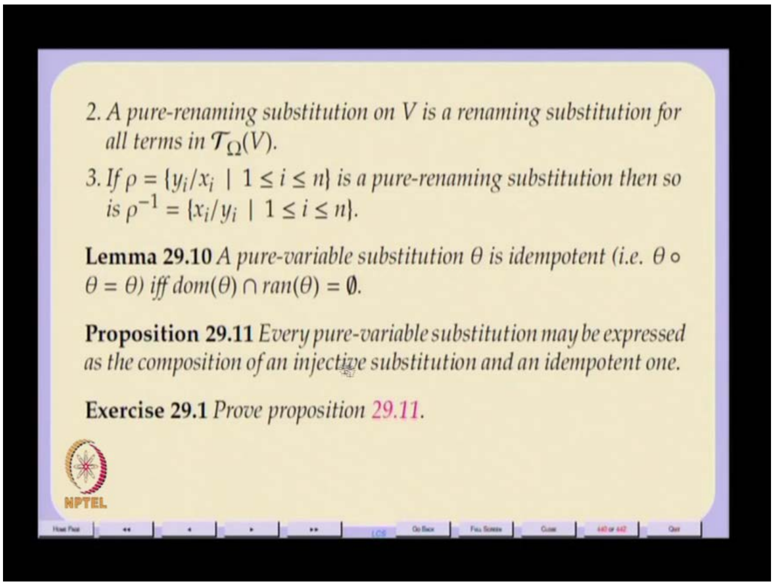
Basically this domain of theta into section domain of tau is empty and domain of tau intersection free variables of t all t is which belong to the range of tau. Yes, I think it should be in the range of theta there is clearly a mistake there domain of tau intersection yeah I think it should be the range of theta yeah I think it because we are looking at a tau compose theta being just the union right. So, this has to be theta definitely thanks I will correct that mistake yeah, notice that I am making a clear distinction between the monoid equality and syntactically identical yeah most books do not do that and next it is quite confusing sometimes.

(Refer Slide Time: 46:32)



So, this this they form a monoid and then there is a there is a notion of let me see now this may not be very important for the rest of this lecture. So, let us let us keep this business of pure variable substitution you can you can actually go and read it yourself the most important thing is another two may theorems there which yeah.

(Refer Slide Time: 47:00)



So, let us actually. So, this is this is the notion of substitution which we want I do not think we require these notions of pure variable substitutions and substitution change there are lots of interesting things, you can prove which which we will skip for the moment ah

But the more important thing is how are we going to use this notion of substitutions couple with the notion of herbrand with the consequences of the herbrand's theorem and the loyanamse colum theorem in order to come with essentially a a programming mechanism yeah. So, that is. So, basically we need to be able to define this we need to be able to use this notion of substitutions to find what is known as a correct answer substitution in a logic program. So, we will we will work our way towards essentially the theory of logic programming because that that is what herbrand's theorem gives us the loyanamse colunm theorem also gives us that to look for models you need to search only within accountable language of terms we do not need to go anywhere yeah

So, I have rushed through this notion of substitution, but please go through it in sometime, but it is it is really ancillary, but we need to know the notion of composition of substitutions and we need to know the notion of simultaneous substitutions in order to be able to in order to be able to actually deals with the notions of with what is known as a correct answer substitution and logic programming yeah. So, go through it and we will start that next time yeah I will stop here .