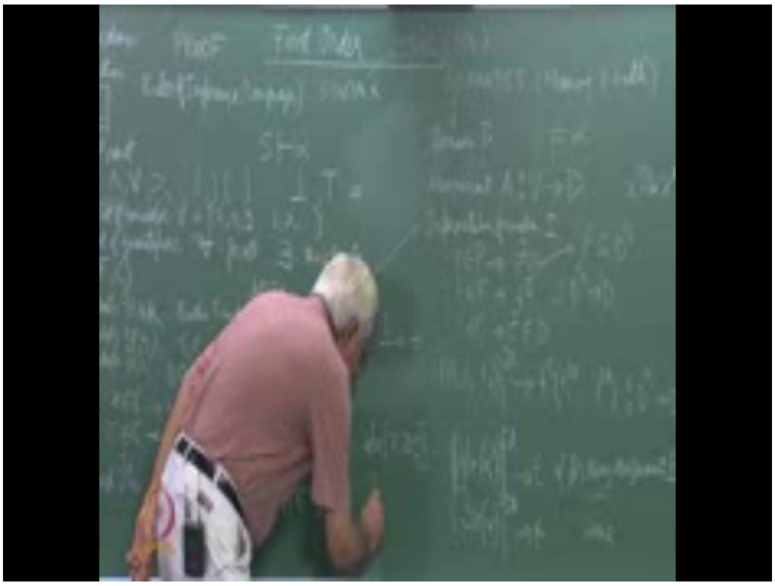
Artificial Intelligence

Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 45
First Order Logic (FOL)

(Refer Slide Time: 00:20)



So, we are in the some sense in the last lap of the course, and we are looking at logic. So, when we say logic there are many aspects to what we are talking about. The first one is syntax or the language part. So, we first define a formal language which is the language of logic that we are talking about and we have rules to say what is the sentence in this language and what is not. Then we have the lotion of semantics which tells us that when we make sentences out of smaller sentences, how do they combine, what is the meaning of those sentences? Or in general we are concern with meaning and truth. So, what do the sentences mean is one aspect of semantics and is a sentence true or not true is the other aspect of semantics. Because we are really interested to use logic as a mechanism to compute in coats true statements to arrive with true statements to arrive with true statements through a synthetic process essentially.

So, but to do that we first need to define whether the sentences true or not essentially that is done in the notion of semantics associated with the notion of semantics. So, given us so one of the thing we are interested in is a given a set of sentences yes which we could call is premises we could call as axioms. But something that we assume to be true without questioning that we want to find out what other sentences can be true. So, in gen so in particular we may be interested in a particular sentences call alpha and we may be interested asking whether alpha is true or not essentially. And to this end we have this notion of entailment and we say that the set of sentence and tail alpha if whenever those sentences are true in those circumstances alpha must be true also. In professional logic which is what we have looked at so far what do the mean by the circumstances?

Basically we mean the valuation function which assigns of valuation to each atomic proposition and then we can lift the valuation up to the compound sentences. And if we have a set of sentences which are true in some given valuation then alpha must be true in that valuations essentially that is the notion of entailment. So, this is the semantics we define, because we are interested in a using logic for doing something. But we have also the same time interested in doing it in a mechanical way. So, the promise of logic is that you can sit down with paper and pencil and decide whether the given sentences true or not essentially. Of course, in the modern context the promise is that you can write a computer program to tell you whether the given sentence is true or not. Because whatever you can do with paper and pencil you can do with a computer program as well.

So, corresponding to the notion of entailment which is a notion concern with truth value we have a notion of proof or drivability in the synthetic side of the language essentially. So, we say that we can derive the sentence alpha given the set s. So we are given the set of sentence which we call is a knowledge base or something like that. And we want to find out now, what else can be add to the knowledge base of the database or set of sentences s and the notion of proof. So, notion of proof is based on the notion of the rules of influence and rules of influences are basically rules which tell you that given some pattern already present in the set of sentences what new sentence can you add to the set up? So, rule basically will say if it has a set of antecedents so for example, model says that if you can see alpha. And if we can see alpha implies beta where alpha and beta could match to anything any sentence then you can add the sentence beta to the knowledge base. So, rules of basically synthetic pattern base mechanism for adding new sentences. And the idea off course you have different kind of proof procedures.

So, we have seen some of them which something which we call national deduction which kind of flows with the rules of influence then there something call indirect. We saw an example of this when we said that make an assumption p or alpha and then it somehow we can show q then we can show p implies q. This form of proof is called indirect proof were you start with an assumption and then arrive at a formula where is the assumption becomes a left is side of a implication statement. Then we saw the resolution method for proportional logic. And there are other methods so for example, that is the tableaux method which is very popular in many logic circles essentially. So, proof procedures are basically mechanism for doing achieving this mechanically producing new form list or mechanically testing whether the given formula can be produce from a given set up formulas and arriving at alpha.

So, as you can see there are 2 routs so arriving to alpha 1 is a semantics rout which says that you look at the meaning of alpha look at the valuation function. Or look at the semantics as to which propositions are true and then decided the alpha is true or not that amounts to constructing a truth table. And we said that truth tables are too large very often if the number of proposition is high. And we want to use this mechanism of proof of generating new sentences till we have generates generated alpha for essentially. Now, today, we want to look at a different language. So, when we say logic is language primary it is language we can define different kinds of logic and we discuss this in the one of the early lectures that you know proposition logic is one of them. But there are more expressive language is which can express things in more details for example, there is epistemic logic which talk about people knowing what other people know and that kind of stuff.

But we are not going to that accent we want to look at the next most the next level of language which is in fact, such popular level that almost everything that we do in computing falls within the scope of this language. And this is known as first order logic and we want to look at that today FOL essentially. In fact, everything that we do on a computer program in a on a computer for example, you write a c program or a java program can be seen as working in first order logic. The characteristics feature of first order logic is there is a notion of variable which is not there in the notion of proposition logic. And we can talk about relation between the different elements essentially. So, let us define first the language first logic and I will do this set of simultaneously. I will work with the syntax in the semantics instead of first giving you the full syntax and then the semantics separately.

So, we will do it a slightly in formal way I will keep writing the syntax and will keep looking at what the semantic corresponded to and we will see what kind of expression is allows us essentially? So we are talking of FOL syntax now as in proposition logic there is the logical part which includes symbols like this. So, we are defining the alphabet of the language so and so on which is the very which is the same as what it was in proposition logic then symbols like brackets and so on. All this is borrowed from proposition logic essentially everything is same. So, at this point I would again like to emphasis the fact that you must discriminate between the symbol that we are using and the meaning of the symbol essentially. So, I have use this symbol or I could have use this symbol and if you want to call this a symbol. It does not a matter what you use this it is a symbol and the syntactic machinery basically looks at the symbols as the something which can be match with same symbol.

So, it is basically pattern matching were as the semantics of the symbol as we have all agreed is that it combines it determines a truth value of 2 sentences combine using this connective by in a specific manner. So, we know that alpha and beta is true whenever alpha is true and beta is true and only then essentially. So, that is a semantic side essentially. So, the semantics is just for our benefit to understand what the language is doing for us and valid and validate for the language is doing for it. The machinery itself does not use the semantic at all the machinery just does pattern matching and so on so forth. So, we are return the semantics of and s saying that alpha is true alpha and beta is true if alpha is true and beta is true. But when we talk about proofs we have different things operating upon this symbol. So, we say for example, is p n q is given to us or alpha and beta given to us we had a rule calls simplification which says we can add alpha itself essentially. Or if alpha is given to you and beta is given to you then you can write alpha and beta.

Though we are not looking at the meaning there we only looking at the well define procedure which is what the proof procedure we are talking about. So, these are the symbols which we borrow from proposition logic then we have set of variables that is called is set v and typically we use symbols like x y z or sometimes x 1 x 2 and so on. It does not matter basically it is a set of variable to given to us. A variable symbols you might want to say we do not we will sort of not keep distinguish between the syntax and semantics. So, when we say a variable we will say that it has its own well define semantics how is the semantics of FOL defined? The semantics is defined in terms of a domine D or some logic book that you might has seen also called the universe of discourse. Essentially the domine is a set of object or elements and the language of first set of logic allow you to talk about relations between elements.

So, for example, you might have the set might be a set of people and you might say that for example, ram is a brother of laxman. So, it is a relation between ram and laxman and it is a brother relation. FOL allows you to talk about things like that essentially of it is a domine is a set of natural numbers you might say 7 are less than 11 essentially. So, it is relation between 7 and 11 the less than relation and we are saying 7 comas 11 belong to this relation. So, everything is a terms of domine D and the semantics that we will associate with variables is an assignment function we will call this which matches which maps every element of every variable to an element of the domine essentially. We are doing syntax and semantics at the same time. So, as we write the syntax we will try to understand what is the meaning of those things? So, every variable would basically stand for element in the domine essentially. So, for example I might say there is a variable for x and the assignment will say x equal to 3 or x equal to 7 or x equal to 20. So that is that is what the assignment function is doing.

So, we would also write saying that x maps to x A. So, this notation we will use to say that this is x A belongs to domine and x belongs to the set of variables and this x maps that is the assignment function which is doing for us. So, x A will stands for the image of x under this mapping then we have a set of quantifiers so true be very pedantic should say set of quantifiers symbols, but we will set of use some slightly loosely. And the 2 most common quantifiers that we use are symbol like this which we read as for all and a symbol like this which we read as there exists. So, it is to additional to when you see this symbol additional read it is for all it additional to read the symbol is there exists directly we will see the semantics of this as he go. But essentially these are quantifiers for variables here and they dictate what assignments are we talking about essentially? So, as you probably know when we use for all so if say for all x p s were p is a predicate which you will define in a moment.

Then I am saying that take any assignment and this p x must be choose essentially. So, we will come to the more formal definition later, but at this moment let us understand and say there are quantifiers are use to quantify what assignments can be used for talking about those variables essentially. And just when we are describing this we can also make an observation that this is the characteristics of first order logic that you have variables in a quantify an over variables. If we talk about the so called second order logic then we have relations or as we will call them predicates. And we will have predicate variables and quantifier over predicate variables essentially. So that is the higher order logic which is we are not talking about in our logic predicate symbols will be fixed essentially. So, this is the logical part which is common in every language that we define and the non logical part in proposition logic and non logical part was a set up proposition that we start off which essentially. In this case it is a set of there are 3 sets.

So, we will define the language l to be define by 3 sets which we will call P F and C. So, these are 3 sets some books would use the symbol our instead of P add use p, because p is stands of predicate here if we use our then it stands for relation essentially. Both are talking about the same thing the predicate comes on the language of the logic. And relation is basically define the about the domine we do not expressly talk about relations here. But, predicates as you will see then intended to stand for relations essentially so what is this set? This is a set of predicate symbol. So very often we use things like P Q R and so on. But we can also use things like brother friend all of these would stand for predicate symbols essentially. So, basically a set we define a set and the language that we are talking about will use those symbols essentially. Then so this is P F is the set of function symbols typically we use small f small g small h or f 1 f 2, but we could also use things like plus father and so on and as you can guess from this plus in father that function symbols basically denote functions in the domain essentially.

Then there is a set of constraints symbols it could be c 1 c 2 or something. So, it could be something like for example, 0 which stands for 0 or which could stands for an empty string any anything which is in this thing. So, we are talking about the syntax of the language the any first order language is defined by these 3 sets; a set predicate symbol a set of function symbol and a set of constraint symbols what is the mapping of this sets? The mapping is defined by an interpretation function we will call is I and what this function does is that it maps every P belonging to P to use for this moment a symbol p I on the domine. So, we will just come to that in a moment essentially likewise every function symbol is map to we will come to this in a moment a function symbol f I and every constraint symbol is map to a constraint symbol t I. But in this case we can also specify at this stage that it something which belongs to D. So the interpretation is an interpretation of language.

The language has as part of it is consitauance these constraint symbols and what interpretation tells you is at what does the constraint symbol stands for in my domine essentially. So, this symbol for example, might stand for the number 0 which is a element of a domine or this might stands for n p string or a symbol liked might stands for example, which is we can treat to be constraint and so on. So, there are 2 symbols which I have forgotten here which is this bottom in the top which is there in every language let me talk about. So, this is the vocabulary of the language then we define what we call as the family of terms sometime we call and set of terms. So, the language we are working towards defining what is the sentences in our language so far we have only define the alphabet for the language. And now, we are gradually working towards a set of sentence is, but before we do that we need to define what we call is the set of terms. And the set of terms let us call the T is defined as follows. That if for every edge which belongs to be for every variable that is we have edges belongs to so in other words we are saying every variable is the term then for every c belonging to T we say C belongs to p.

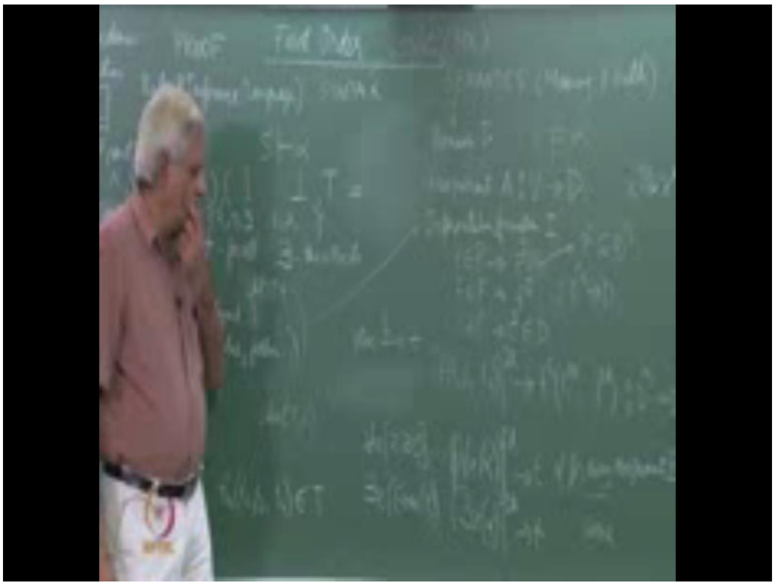
So, every constant is a term so what are these terms that we are talking about? The set of term is essentially the set of object in the domine essentially. Basically when we say x belongs to t we the semantics of that is that x A belongs D were x belongs to v x A belongs D that is what we are saying a set of terms. So, x is a term here we are saying x is a term here we are saying a cons here we are saying variable is a term here we are saying constraint is term and by term basically we mean something in my domine essentially. So, this C I also belong to D and x A is also belongs to D. So, one thing which I have not at mentioned is that each of these predicate symbols or each of these function symbols has associated with an arity essentially. So, they have an associated with an arity which basically tells you how many arguments it can take? In the case of functions, we will define them terms in the case of predicate we will define atomic formulas.

But basically in both the cases they take a set of arguments and the arity tells you what the arguments are so very often we write the arity below. For example, if we write arity 2 below plus it basically means it takes 2 arguments. If we take arity 3 below plus we say that it is a function 3 argument essentially. So how many arguments it take essentially? Now, we have variable system constraint system and therefore, every f n when I write f n like this it means this says arity n belonging to my set of function symbols and a set p 1 p n belonging to set off term. So, this is a recursive definition as you will see this structural recursion which we offer new to define a language. The expression f n followed by t 1, t 2 t n, so we can take a function symbol of given arity then take that many agreements which must be terms put them together and we get a new term essentially. So, for example, I could say something like plus 7 6 if plus is a function symbol of arity 2 then this expression stands for the term which takes to arguments.

Now, you know that this plus for example, is mapped so for example, I might say something like this that plus under the interpretation I maps to plus which I would have call plus I. But let see more we're saying that plus map maps so this plus is and arithmetic operator on the domine of natural numbers. Let us see and if 7 and 6 are terms which map to respectively number 7 and 6 so which is off course very obvious for all of us. Then these expression stands for the number 13 given the fact that I am interpreting the plus as a addition symbol a symbol which stands for addition of 2 numbers and given that am giving this 2 agreements which respectively stands for 7 and 6. Then in domine if I add it 7 and 6 I would have got the number 13 and this term essentially stands for this number 13. So, we can define this here that given a term it is mapping under interpretation and assignment gives us a term which interpret function name which is interpreted followed by every term who is interpretation I have to do.

So, the semantics of terms is that either there variables or they are constraints or they are function symbols apply to an appropriate number of terms essentially. And what they gave us is a term essentially. So, this itself is a mapping from D raise to n to D. Basically a function symbol defines a term a function symbol also defines some mapping in the domine which is from D raise to D I could have written here actually D raise in to D it is a mapping from D raise in to D. So, it takes the argument n agreements and gives the one value or one result and all the agreements and all the values they comes from the domine essentially. So, plus for example, is D cross D to D essentially it takes 2 agreements and it gives one agreements.

(Refer Slide Time: 30:02)



We want look at formulas eventually we want to move towards sentences. But before we come to formulas so want to define this set F of formulas before that we define a set A of atomic formulas. And this atomic formulas will some sense of correspond to the propositional symbols in propositional logic essentially. So, first these things they belong to atomic formulas then if P belongs to P is a set of predicate symbols and t 1 t 2 t n belongs to a set of terms. So, t is a as an arty n then this P with arity n t 1 t 2 t n belongs to set up so an. Atomic formula in first order logic which is also called predicate logic or predicate calculus is made up of taking proposition symbol of arity n and n arguments which must be terms essentially. Now, you must keep this in mind this is strictly what keeps first order logic to define the boundary the first order logic what can what is the sentences of first order logic and what is not a sentences of.

So, if want to say for example, the predicate name believes and if I say john believe that the earth is flat. Then I can try and define a predicate which says 2 agreements as to who is believer and what is the thing that is believed, in then the believer is john. And the thing that believed is believed in is the flat earth essentially now, but flat earth itself a relation. It is basically saying that the earth is flat essentially something which is true or false essentially. So, this sentences john believes at the earth is flat is not a sentences in first order logic. Because you can only give terms as inputs and cannot gave formulas as the input that essentially. So the predicate symbol that we are taking about is basically map to a subset so p I is a subset of d rise to n. If you were they were the subset is a it is an relation on the domine D and the relation is this subset which has the which belongs to this set essentially. Sometime logician talk about first order logic and they distinguished from first order logic with equality. Some people do not include equality inside the language, but some people do.

So, let us talk about the first order logic with equality here in which case we have one more atomic formula. So, we have this top and bottom is atomic formula is an we have taking a predicate symbol and put giving a appropriate number of arguments gives a atomic formula. And now, if t i and t j belongs to a set of terms then t i equal to t j belongs to the set of atomic. So, in first order logic with equality we have atomic formula is which talk about equality as well essentially. So, we have an expression essentially so which means when we talk about first order logic with equality we must have the equality symbol also thrown in here. So, this is a set atomic formulas so before moving on to compound formulas or our formula is in general let us talk about the proof values of these atomic formulas. Because we want to talk about the semantic in terms of which formula is are true and which formula is are not true. We are so remember this truth or valuation is a mapping to this set of true or false sentences.

So, one thing we always do that we always map this true false and we always map this true and then we map atomic formulas of the first kind. So, I will note write and so it basically let assume it is implicit t n to true. If and only if the corresponding which is t 1 i A t 2 when I write i A on the top it means by applying the interpretation and by applying the assignment if there are any variables inside. So, this is off course something which we are familiar with formula is in this notion. That a predicate like this is true if I say brother ram laxman and I say brother ram laxman maps true. If in the real world ram and laxman the pair belongs to the set up pairs which define the set of brothers essentially. Or I can say that 7 less than 13 is true if in my domain I define the less than relation let us say I define in to then 7 coma 13 belongs to the set of pairs which define the less than relationship in this we are familiar with. The second kind t 1 t i equal to t j this maps to true if t i and stands for the same element if they stand for the same element essentially.

So, if I say the prime minister of India and if i say manmohan singh then I say these 2 terms are the same. In other words manmohan singh is a prime minister of India if in the domine they map to the same person essentially so that gives us the set up atomic formulas. Now, let us talk about the set up formulas many people use the term formulas some people use the term formula. But I think the more moderns style seems to be formulas essentially. So here first of all we borrow everything we do in proposition logic which means we define things like alpha and beta alpha implies beta and so on. So,, this is like in P l exactly we borrow there so if alpha is formula and beta is a formula then alpha in beta is formula essentially. So, you must suppose start of by saying that all atomic formula is formula essentially and then you can use logical connective so construct more formulas. But we also have formula is of this kind that for all followed by variable name followed by a formula is a formula.

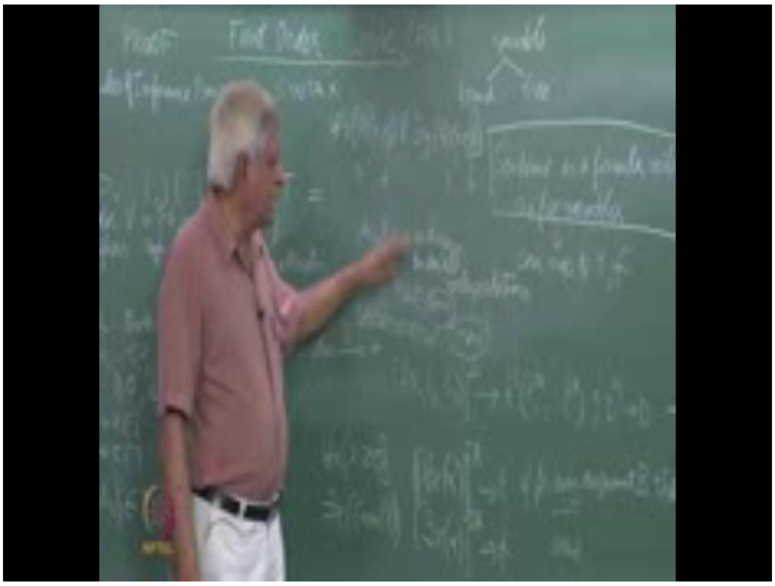
And then exist followed by a name followed by a formula they both belong to the set of formulas F that we are talking about. So, this is something new in first order logic which is that you can. So recall that we just made an observation that atomic predicates they correspond to proportional symbols into proposition logic. In fact, you can think of proportional and proposition logic as a predicate with 0 arguments essentially. So, there nothing to follow after that it just becomes to proportional symbols you cannot break it down in to future where as the predicate breaking down in it into further. So, if I say Socrates is a man we treat it that as a proposition earlier. But now, we would talk about it as man Socrates that we are breaking it down. And saying that man is a relation of arity one which is defines over the set and basically Socrates belongs to this. So, man is a subset of the elements of the domine and Socrates is 1 those elements essentially. So the atomic formula is define here and the of compound formula is define exactly like we do in proposition logic that we use. In fact, this is what gives a definition to what does this symbol and mean what does the symbol implies mean and so on.

And then we have this new formula is which say that alpha and there exists x beta essentially. So, when these formulas are is true? So,, we say so let me write it here for on x alpha on the way interpretation and an assignment maps to true if for every assignment B that is an x variant a. So, let me first treat this out we are saying that a formula for all like self under an interpretation an assignment so what does the interpretation do? Interpretation defines all the predicate symbol and the function symbols as you what the stand for an assignment tells you what every variable is being map to in the this thing. We say that such a thing is true under a given interpretation and an assignment. If for assignment B so we are talking of other assignment know what are assignments? Assignments give map variables to elements in the domine that is an x variant of a so when we say an x variant of a we mean that the assignment B defers from the assignment a only in the mapping of these variable x.

And all other variables it maps identically as in a. So, b maps everything like a does except for x which it maps differently. So that is call an x variant of a so for every b that is an x variant o f a. In other words x can take any value the formula now without the quantify alpha I under that assignment B map to true. Likewise the exits x alpha under i A maps true all this is a same except that instead of every we have some. So, for example, if I have a sentences so let us say that there is only one variable in a system so far, but and let us see we are talking about natural number. Then if I say for all x x greater than equal to 0 I say that such a sentences true under an assignment a if for any value that I plug in for x this part x greater than 0 greater than equal to 0 will be true essentially. You can do things like this for all x x greater than y.

And then we can say that in this assignment a y is equal to something. That is a little bit complicated we do not want to get into there and that is not the sort of thing we normally do so let us not try and do that even essentially. But the cuts of the matter are that a sentence quantified by for all x would be true. If you plug in any value for x and the sentences becomes true and you remove the quantify. Likewise something like their existence x such that even x is a sentences which would be true. If we can find some assignments which means some value of x which make this second part true. So I can say even 4 for example, because even 4 is true I can say there and x even number is true essentially. This is a sentence which will have not decide define what is the sentence?

(Refer Slide Time: 45:50)



But to define a sentence first we have to characterize variables in to 2 kinds variable x can be either bound or free. So, a variable is set to be bound if it is quantified essentially. So, if I say for example, for all x p x y were p is some p is some predicate and x and y are variables I can say that x is bound, but y is 3 essentially. So, if variable has a quantify then it is bound and if a variable does not have a quantify then it is free. So, if I say something like this for all x p x y or exists y q x y something like this. Then you can see that this is bound this is 3; this is bound and this is also bound why? This occurrence of y bound, because it comes it in the scope of quantifies which is defined by the bracket around here. And this occurrence of y goes not have a quantify so this occurrence y is 3, but x is quantified by x in likewise that x is quantified by this end, because they come within the larger brackets. So, a sentence is a formula with no free variables this is definition of a sentence.

So, it must first be a formula well form formula to all those definitions that we have defining this formula set formula is set of atomic formula and so on it must be a well form formula. So, notice that this is also well form formula, but it is not a sentence because it is got a free variable inside essentially. So, a sentence is the well form formula which we are calling as formulas without any free variables and the intuition behind this is that. Because every variable bound which means every variables assignment is controlled by some quantifier we can talk about the truth value of that variable essentially. So, this for example, we can map to true or false so all of you will agree that this is true this is true. But if I said something like for all x even x it is a sentence it has got only one variable and it is something which is not true.

So, I can say that this is not fall this is not true, because the definition of the truth value according to this definition it says that for I plug in any value of for x and then even x must be true. Now, if i plug in 3 for example, when even 3 is not true so there for the sentence not true. So, if I say something like this for all x; x greater than y now this is not a sentence. Because it has got a 3 variable which is y and you can see that we cannot say whether this sentence is true or false. Because why could be anything essentially, but if I said something like this there exist y such that for all x x is greater than y then I can say whether the sentence true or not will that. Of course depends on the domine and the interpretation function that we are talking about so a sentence something to which we can assign a truth value sentences are like this which can be map to truth values. A valid sentence is true under all interpretation you can choose any domine and any mapping for the predicates symbols and function symbols and the sentence will be true.

So, we can see sentence like x equal to x or P of x implies p of x will always be true irrespective of what domains be true. Satisfiable instead of all we say some and unsatisfiable instead of some are say none. So, with this have a basic machinery for working with first order logic we have define the language start it off by a alphabet then the set of terms then the set of atomic formulas and then the set of formulence. And the finally, the set sentences and we defined how to assign truth value to each of them. And we can now talk about this whether the sentences valid or satisfiable or unsatisfiable. We will upon this part, little bit in the next class when you meet will stop here.