Artificial Intelligence: First Order Logic Syntax

Prof. Deepak Khemani

Department of Computer Science and Engineering

Indian Institute of Technology, Madras

Module – 03

Lecture - 01

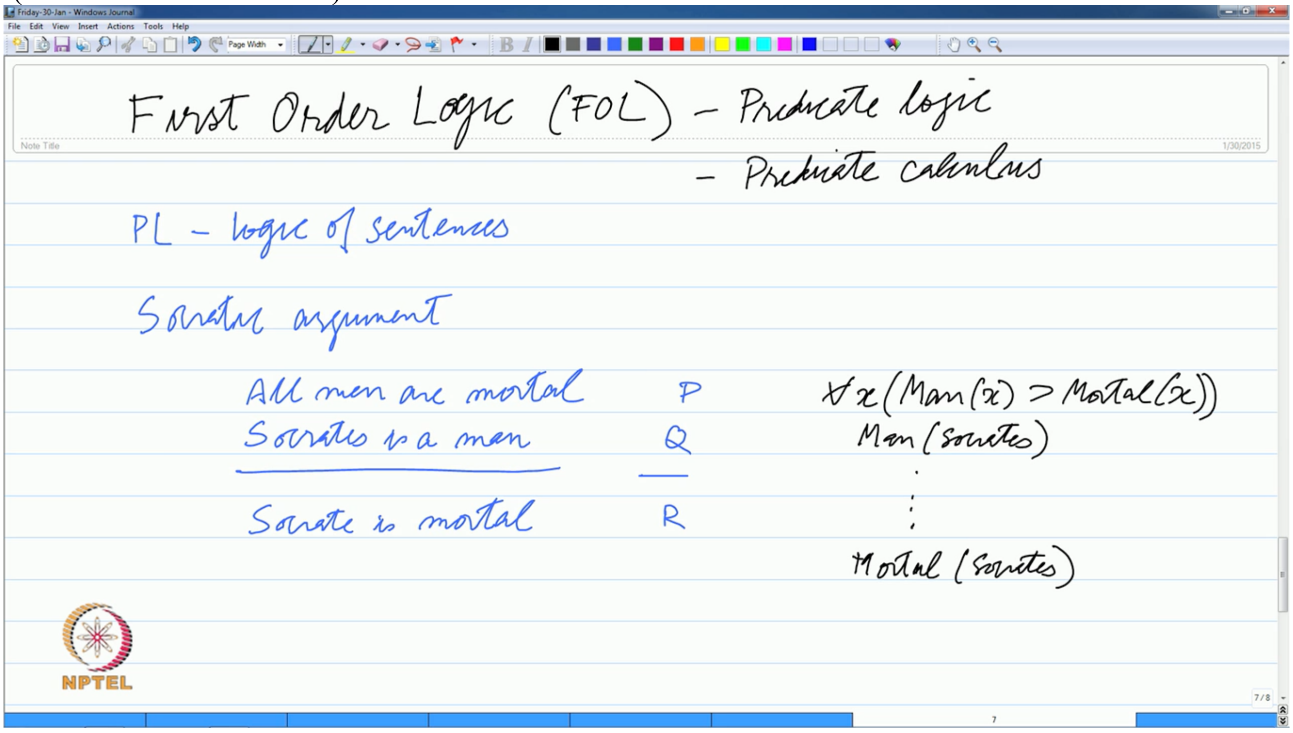
So so far we have been looking at propositional logic. Propositional logic is a logic of sentences and essentially if you are given a certain set of sentences and told that they are true then you can explore the logical connections between those sentences and also determine which other sentences are closely related. But we find that this logic is not expressive enough and that is the reason we want to move on to this first order logic which is the most popular form of logic that we know because it is quite expressive, we can say lot of things in first order logic and yet it is within reasonable amount of computational complexity.

So we begin looking at first order logic. We will use the notation FOL but we also sometimes call it predicate logic because it uses a notion of predicates which i am sure you are familiar with. Also called predicate calculus. For example, we said propositional logic was propositional calculus. So these are the terms we will use interchangeably but we will use the term FOL most of the time. Why are we interested in this? Because if you remember we started with the Socratic argument when we said that what do we mean by logical inference, what do we mean by deduction. So if we are given a sentence like All men are mortal and if we are given another sentence which says Socrates is a man then we should be able to infer that socrates is mortal. This form of reasoning we said was syllogism, one of the 16 or 19 syllogisms I don't know that the greeks had specified.

Now the trouble is that if we try to encode this in propositional logic, here we cannot really say much, so we could say, for example that this is P and this is Q and this is R that we want to show. And there is no connection between them because in propositional logic we accept the sentence as an atomic unit. We cannot look inside a sentence and say what is the sentence talking about. So in fact the fact that the sentence is talking about men being mortal and Socrates being a man is hidden from us because at the sentential logic level we treat the whole thing as an atomic unit. And at that point we had also mentioned that in the first order logic what you would end up writing is in the language something like this:

So our given statements would have been these two statements: The first statement is essentially a rendering of same english sentence all men are mortals in first order logic. And the second sentence is likewise the second sentence which is Socrates is a man. From this you would want to derive the sentence that Mortal. And as we will see it is indeed possible to do that in the first order logic. We will be able to derive. So what did we mean by derive. By derive we mean just use the syntactic procedure, the pattern matching and you are allowed to add certain formulas based on certain rules of inference. And then you can add the formula that you are interested in. We will see that it is indeed possible to derive Mortal Socrates but before we do that we need to sort of define our language and in this case we will see that we also need to ...we can also define what do we mean by sentences in our logic. We can define the semantics more expressively.

(Refer Slide Time: 05:10)



First Order Logic (FOL) – Predicate logic

– Predicate calculus

PL – logic of sentences

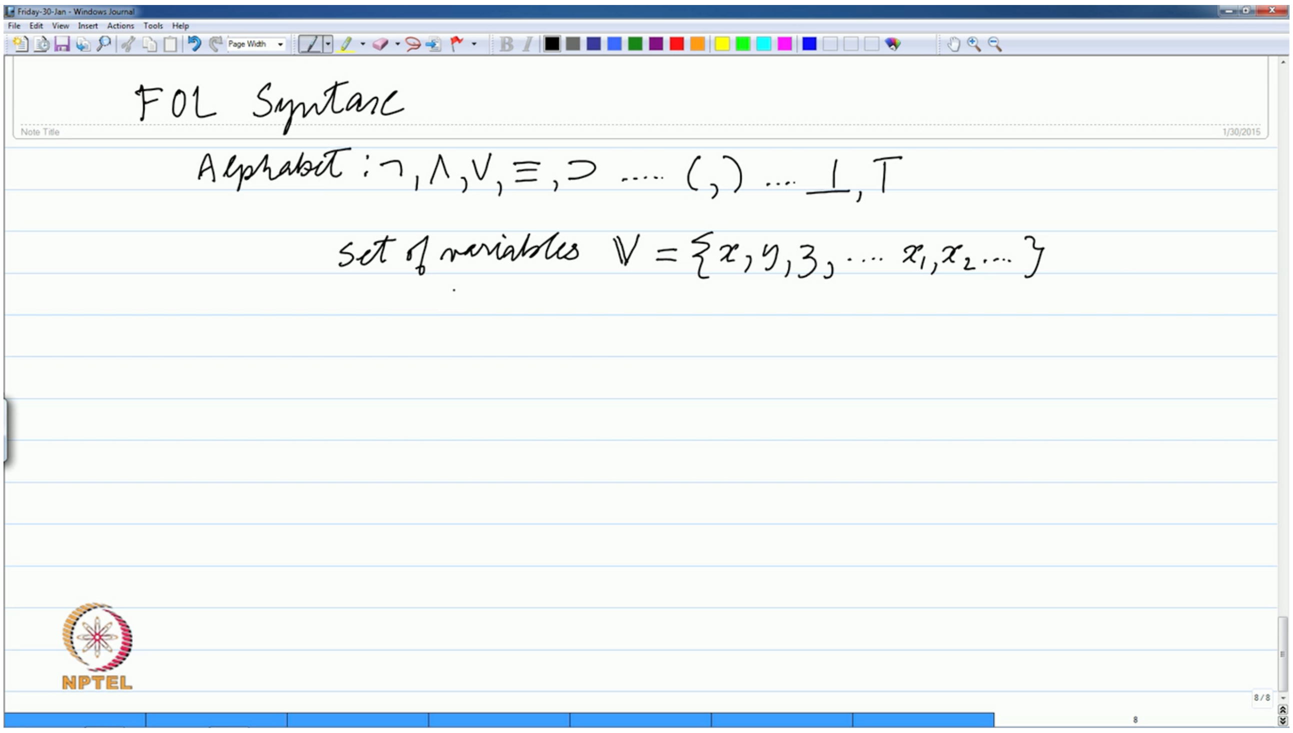
Socratic argument

All men are mortal	P	$\forall x(Man(x) \supset Mortal(x))$
Socrates is a man	Q	Man(Socrates)
Socrate is mortal	R	Mortal(Socrates)

In propositional logic the semantics simply said that P stands for certain statement, Q stands for some statement and if we have a statement of the kind P implies Q and if we have P then we can infer Q. The modus ponens rule will still carry to the first order logic but here we will also be able to talk about the meaning of sentences.

So we begin with syntax of FOL. And as we did in propositional logic we will see that the FOL alphabet is basically an augmentation of propositional logic arguments. We will add more symbols and we will be able to write in some sense complex sentences. So we start with the alphabet as usual. So we have the logical connectives which we had in the propositional logic. And as in propositional logic we can choose which connectives we want to work with. So these are some of the connectives that we would be interested in. And we will also carry forward the brackets that we used to break up compound formulas into simple formulas. And we will also carry forward the two constants that we had in propositional logic. So these were kind of constant sentences and the first sentence which we read as bottom stands for something which is always false and the second sentence which we read as top stands for something which is always true. But in addition to that in FOL we have a greater alphabet so we have a set of variables. These are not propositional variables these are simply set of variables which we call as a set V which would be made up of elements, typically we use things like x, y, z and so on or we may use x1, x2. So if you remember in the last example we said that for all x, Man x and so on. So x was a variable.

(Refer Slide Time: 8:02)



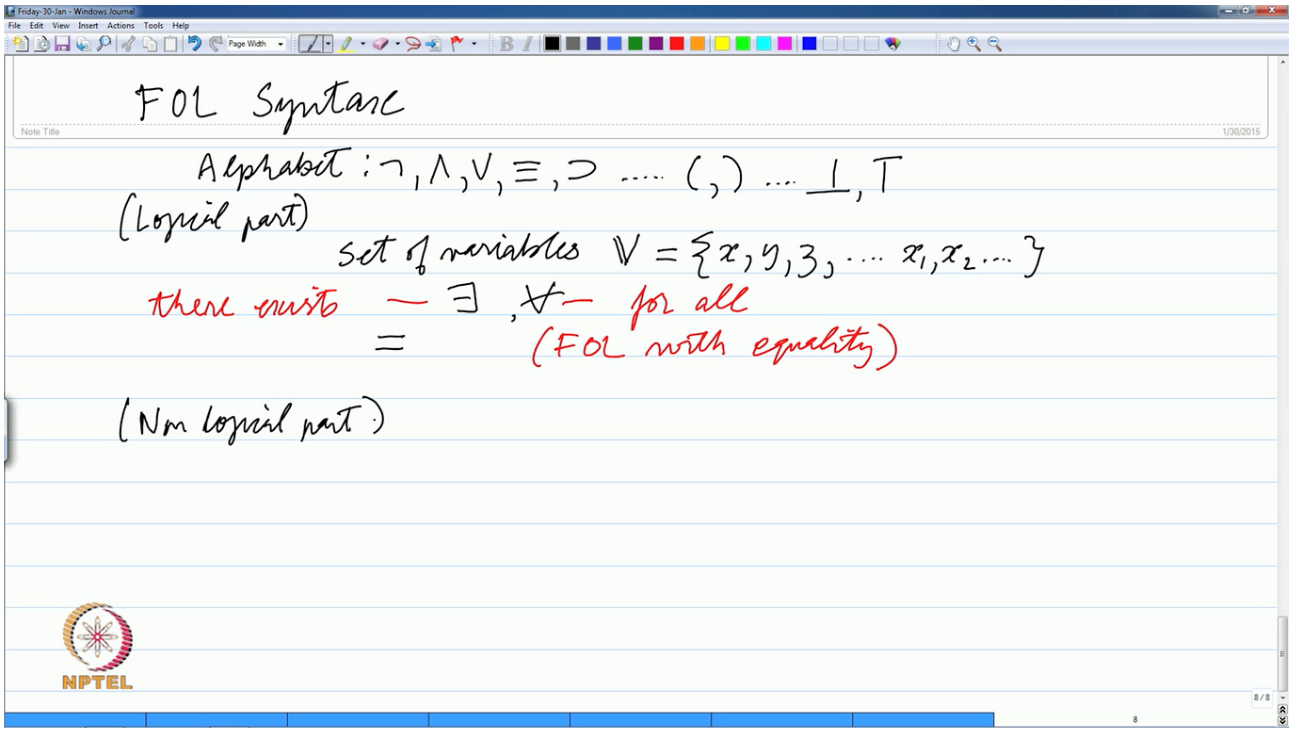
FOL Syntax

Alphabet : $\neg, \wedge, \vee, \equiv, \supset$ ..... (,) .... $\perp, \top$

set of variables $\mathbb{V} = \{x, y, z, \ldots x_1, x_2 \ldots\}$

Then we have two new symbols which are this (\exists) and this (\A). So the way we will read this is the first one we will read as there exists and the second one we will read as for all which is what we did when we talked about all men are mortal we said for *all x* something something. And then we may or may not include the symbol which is equality. So if we include it we call it FOL with Equality and if we do not then we have a slightly smaller version of FOL. So people tend to distinguish between two languages with equality and without equality essentially because equality has certain properties that we can exploit essentially. So all this was the logical part and by the logical part we mean which is independent of any particular logic system or logic language that we are defining essentially. Any logic that we define will have these kinds of symbols. Of course we may have subset of the connectives that's left to us but all these symbols would be there in any FO language.

(Refer Slide Time: 09:47)



FOL Syntax

Alphabet : $\neg, \wedge, \vee, \equiv, \supset$ .... (,) .... $\perp, \top$

(Logical part)

set of variables $\mathbb{V} = \{x, y, z, \dots x_1, x_2 \dots\}$

there exists — $\exists$, $\forall$ — for all

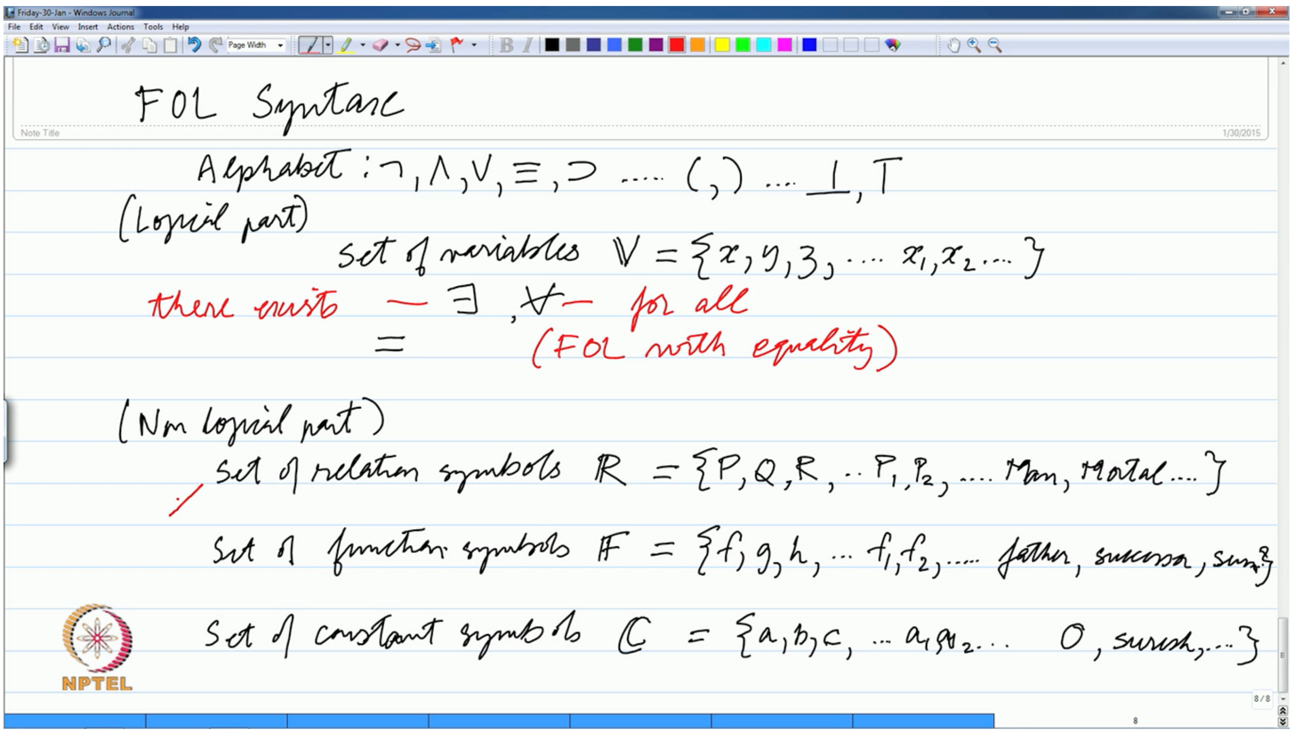
$=$ (FOL with equality)

(Non logical part)

And then there is a non-logical part and when we say non logical part we also mean domain dependent essentially, what is that we are trying to talk about. So there are three sets of symbols here, set of relation symbols, we call this set R. And this will typically include symbols like P, Q , R which are like the propositional symbols that we used in propositional logic. Or we can also use something like P1, P2 or we can also use something like words of english language which are sort of meaningful for us essentially, man mortal and so on. We think of man as an alphabet here, instead of saying P3, I am saying man because man is more meaningful to us essentially.

Then we have the set of function symbols and we will use this for that and typically we use things like f, g, h or f1, f2. But We could also use something like father, successor or sum if we are talking about numbers. And finally we have a set of constant symbols. We will call this set C and they will be made up of functions and typically we would be using the first few alphabets like a, b, c and so on but you could use a1, a2 and so on. But you also use 0 if you are talking about number or a person for example Suresh.

(Refer Slide Time: 12:52)



FOL Syntax

Alphabet : $\neg, \wedge, \vee, \equiv, \supset$ ..... (,) .... $\bot, \top$

(Logical part)

set of variables $\mathbb{V} = \{x, y, z, \ldots x_1, x_2 \ldots\}$

there exists — $\exists$, $\forall$ — for all

$=$ (FOL with equality)

(Non logical part)

Set of relation symbols $\mathbb{R} = \{P, Q, R, \ldots P_1, P_2, \ldots$ Man, Mortal ....$\}$

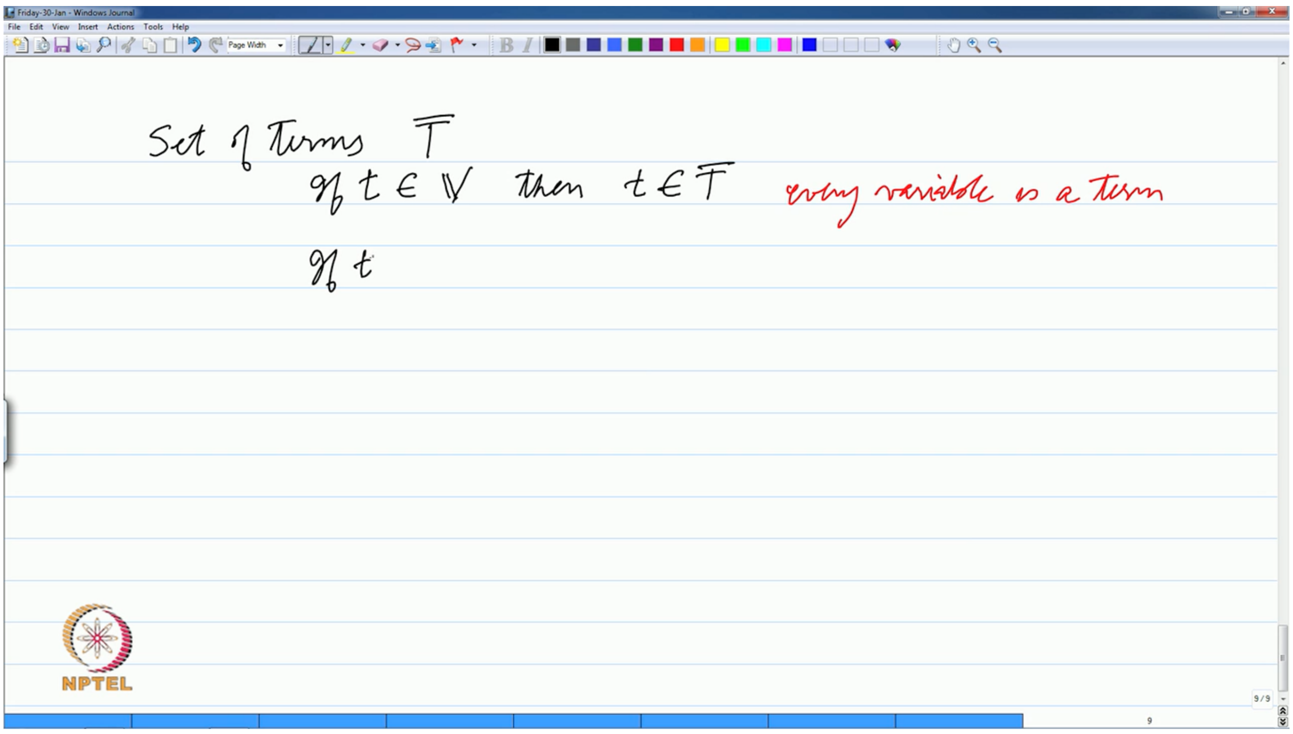
Set of function symbols $\mathbb{F} = \{f, g, h, \ldots f_1, f_2, \ldots$ father, successor, sum$\}$

Set of constant symbols $\mathbb{C} = \{a, b, c, \ldots a_1, a_2 \ldots$ 0, suresh, ...$\}$

Now both these set of relation symbols and the set of constant symbols have an arity associated with them and we will see that when we define the language. An arity tells you how many other symbols it interacts with or takes in some sense. Some people say that set of constant symbols are essentially a subset of the set of function symbols, where arity is ...g..... Anyway this is just for the sake of knowing, we should know that this is often done but we will distinguish between set of function symbols and constant symbols because we tend to think of set of function symbols as we will see shortly as elements of a domain or named elements of the domain.

So that was the alphabet and then we define a set of terms. Let’s use this notation of the set of terms, we define it by structural recursion as follows: So remember what did we have. We had a set of variables V, we had a set of relation symbols, we had a set of function symbols and we had a set of constant symbols, and then we had these other logical connectives, then we had these there exists and for all and equality. So the set of terms T is defined as follows that If, I will just use T as a kind of a meta level variable belongs to the set of variables, then t belongs to the set of terms. In other words, every variable is a term.

(Refer Slide Time: 15:22)



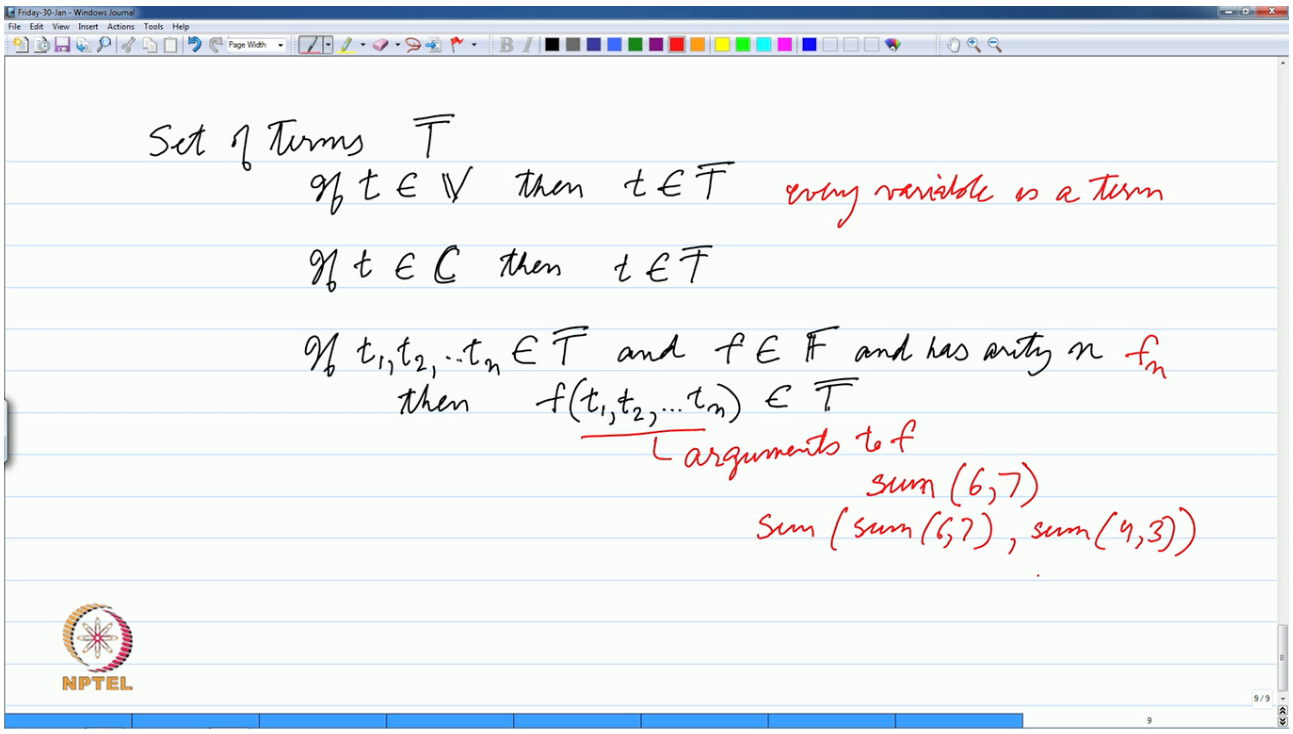
Set of Terms T

If $t \in V$ then $t \in T$ every variable is a term

If t

Likewise, if t belongs to the set of constant symbols, then t belongs to the set of terms. So every constant is a term. Then if t1 t2 tn belong to terms, t1 t2, up to tn are terms and f belongs to the set of function symbols and has arity n. Sometimes we would write this a f n where this n subscript you know is the arity of the function symbol. Then the function symbol followed by a bracket followed by the n terms is also a term. Now notice here that so we say that these terms are arguments to f essentially. And the number of arguments they take is defined by the arity of the function essentially. But notice that arguments are terms essentially which means they could themselves be made up of another function symbol with another set of arguments which are terms, so recursively we can write terms which are more and more compound essentially. So examples of terms in some language for example we are dealing with number you may write something like sum 6 7 where sum is a function with arity 2. It corresponds to the sum we are used to dealing with in numbers. So that's a term essentially. I could also have written as sum sum 6 7 sum 4 3 and that should also be a term essentially. So the arguments of terms need to be terms that's the only condition that we have essentially.

(Refer Slide Time: 18:25)


Friday-30-Jan - Windows Journal
File Edit View Insert Actions Tools Help

Set of Terms $\mathbb{T}$

If $t \in \mathbb{V}$ then $t \in \mathbb{T}$ every variable is a term

If $t \in \mathbb{C}$ then $t \in \mathbb{T}$

If $t_1, t_2, \ldots t_n \in \mathbb{T}$ and $f \in \mathbb{F}$ and has arity n f_n

then $\underbrace{f(t_1, t_2, \ldots t_n)}_{\text{arguments to } f} \in \mathbb{T}$

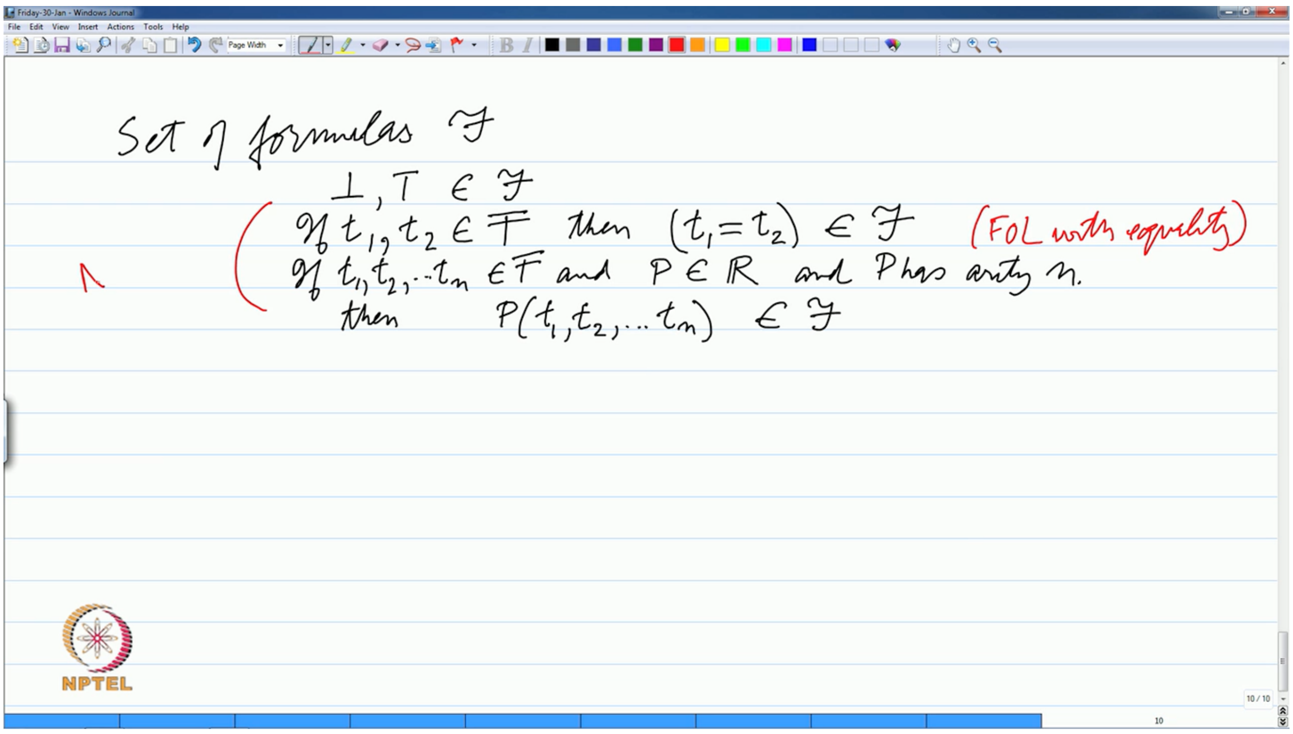
sum (6,7)

Sum (sum (6,7) , sum (4,3))

So we define a set of terms or as people say a family of terms that's the building block of our language essentially. As we will see in the semantics the set of terms will correspond to the elements of the domain. So if our domain is going to be the set of natural numbers for eg. then this term which says sum of 6 and 7 will stand for the number 13 . Likewise, this term will stand for the sum of 13 and 7 which is 20 essentially. So every term essentially will stand for something which is an element of the domain. Variables are the terms that we dont know what they stand for and we are sometimes trying to find out. Constants are terms which are fixed essentially, we know that they stand for something, so individual names for example zero or Suresh or Rakesh, they could be constants in the language essentially.

Then we have the set of formulae. So let's just call it a set f like this. I just need a different f which should not be confused with the function symbols. So that's it this is set of formulas f and in the similar fashion we define the set of formulae. So initially we say that these two are formulae, Top and bottom are formulae like in propositional calculus they are something which we call them as sentences. We will shortly define sentences as well. Then if t1 and t2 are terms then t1 equal to t2 belongs to set of formulae. This is obviously in the case when we have FOL with equality. Whenever we have FOL with equality we allow this particular formula in the language essentially. We can add brackets here from clarity and we generally often prefer to add brackets. Then if t1 t2 tn are terms and P is a relation symbol and P has arity n then P followed by t1,t2, tn belongs to set of formulae. So now we are using a relation symbol which also is known as the predicate symbol and we can take a predicate symbol of a cetain arity provided there are a certain number fo arguments and we have a formula essentially. So we say for example, in the example that I disucssed man x or man socrates, both x and socrates are terms remember. X is a variable, Socrates is a constant but they are both terms and when you said man x then man was a predicate symbol of arity 1 and the formula that thing became a formula essentially.

(Refer Slide Time: 22:55)



Set of formulas $\mathcal{F}$

$\perp, \top \in \mathcal{F}$

If $t_1, t_2 \in \mathcal{T}$ then $(t_1 = t_2) \in \mathcal{F}$ (FOL with equality)

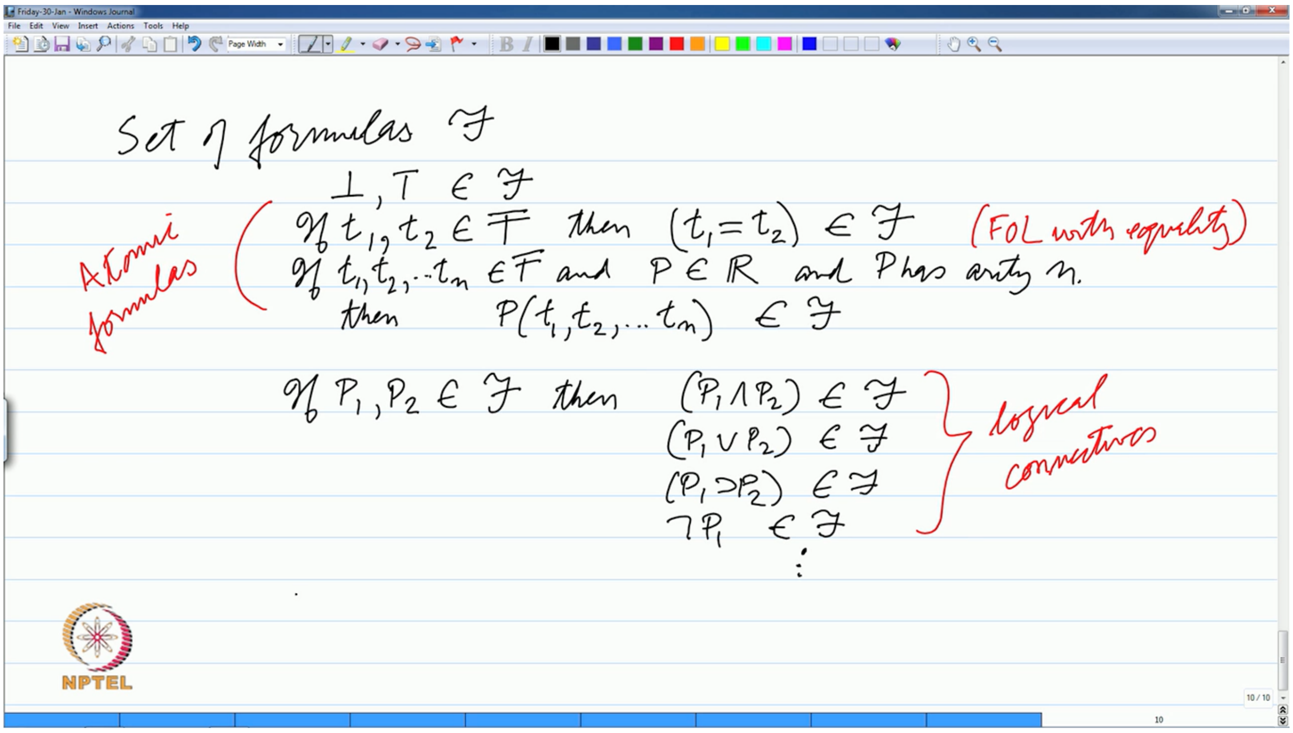
If $t_1, t_2, \ldots t_n \in \mathcal{T}$ and $P \in \mathcal{R}$ and P has arity n.

then $P(t_1, t_2, \ldots t_n) \in \mathcal{F}$

So sometimes we associate these with atomic formulae in the sense that you cannot break them down any further just like we had propositional variables which were kind of atomic you could not break it down any further. Atomic formulae are similar in nature that you cannot break them down any further. Essentially as far as the language that we are trying to define. You remember there is a language that we are trying to define is going to be a set of sentences. Then there are other kind of formulae that we can say are compound formulae which sort of deal with the connectives and so on.

So if you say that P1 P2 are formulae then P1 and P2 is a formula. P1 or P2 is a formula. P1 implies P2 is a formula. Negation the symbol we are using is different, so negation of P1 is a formula and so on. So this is like in propositional calculus so if you are given two formulas then you can use the logical connectives to construct larger formulas. The same thing carries over here and the semantics of the logical connectives will be identical as we will see.

(Refer Slide Time: 24:54)


Set of formulas $\mathcal{F}$

$\bot, \top \in \mathcal{F}$

Atomic formulas

If $t_1, t_2 \in \mathcal{T}$ then $(t_1 = t_2) \in \mathcal{F}$ (FOL with equality)

If $t_1, t_2, \ldots t_n \in \mathcal{T}$ and $P \in \mathcal{R}$ and P has arity n.

then $P(t_1, t_2, \ldots t_n) \in \mathcal{F}$

If $P_1, P_2 \in \mathcal{F}$ then $(P_1 \wedge P_2) \in \mathcal{F}$

$(P_1 \vee P_2) \in \mathcal{F}$

$(P_1 \supset P_2) \in \mathcal{F}$

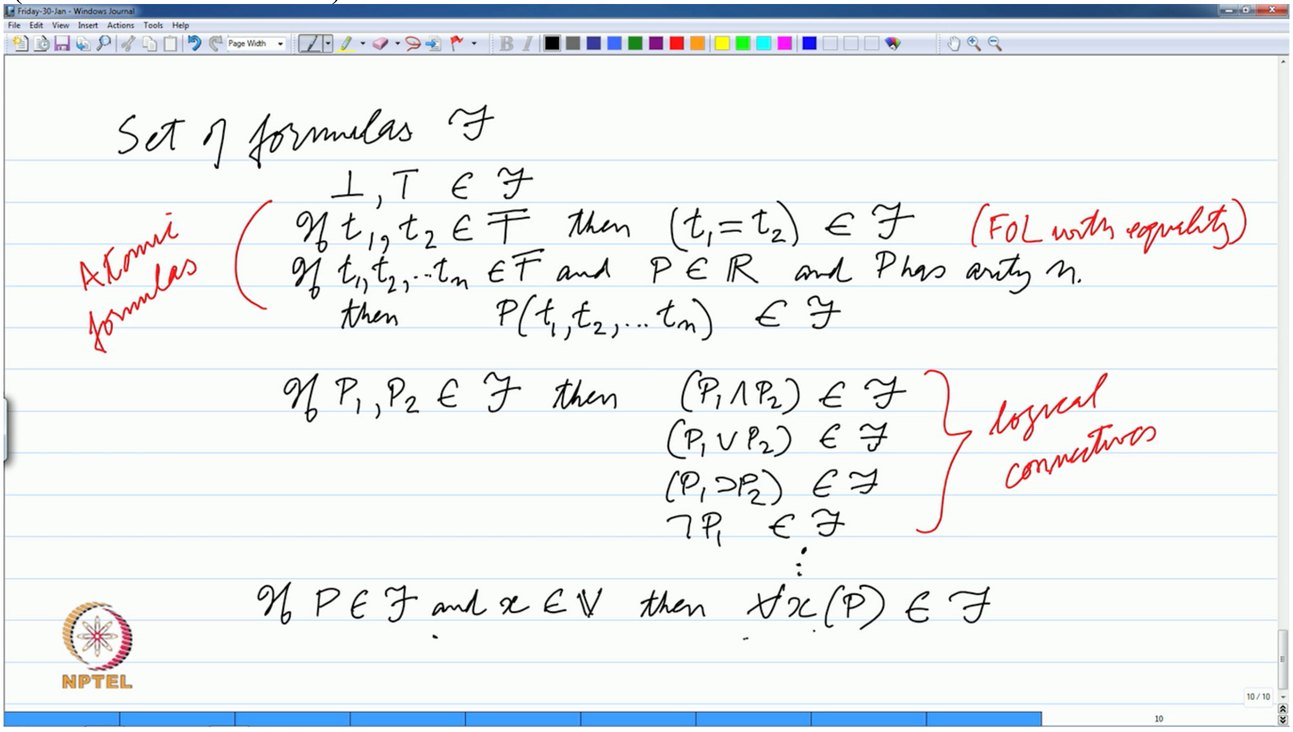
$\neg P_1 \in \mathcal{F}$

$\vdots$

logical connectives

Then if there is a formula P which is this and there is a variable x which is set V then. So you are given any formula which we call P here and if you are given any variable x then we can construct a formula by prefixing this P with the symbol for all and followed by a variable x essentially. And we will read this for all x essentially. So for all x P is true essentially.

(Refer Slide Time: 25:46)


Set of formulas $\mathcal{F}$

$\bot, \top \in \mathcal{F}$

Atomic formulas

If $t_1, t_2 \in \mathcal{T}$ then $(t_1 = t_2) \in \mathcal{F}$ (FOL with equality)

If $t_1, t_2, \ldots t_n \in \mathcal{T}$ and $P \in \mathcal{R}$ and P has arity n.

then $P(t_1, t_2, \ldots t_n) \in \mathcal{F}$

If $P_1, P_2 \in \mathcal{F}$ then $(P_1 \wedge P_2) \in \mathcal{F}$

$(P_1 \vee P_2) \in \mathcal{F}$

$(P_1 \supset P_2) \in \mathcal{F}$

$\neg P_1 \in \mathcal{F}$

$\vdots$

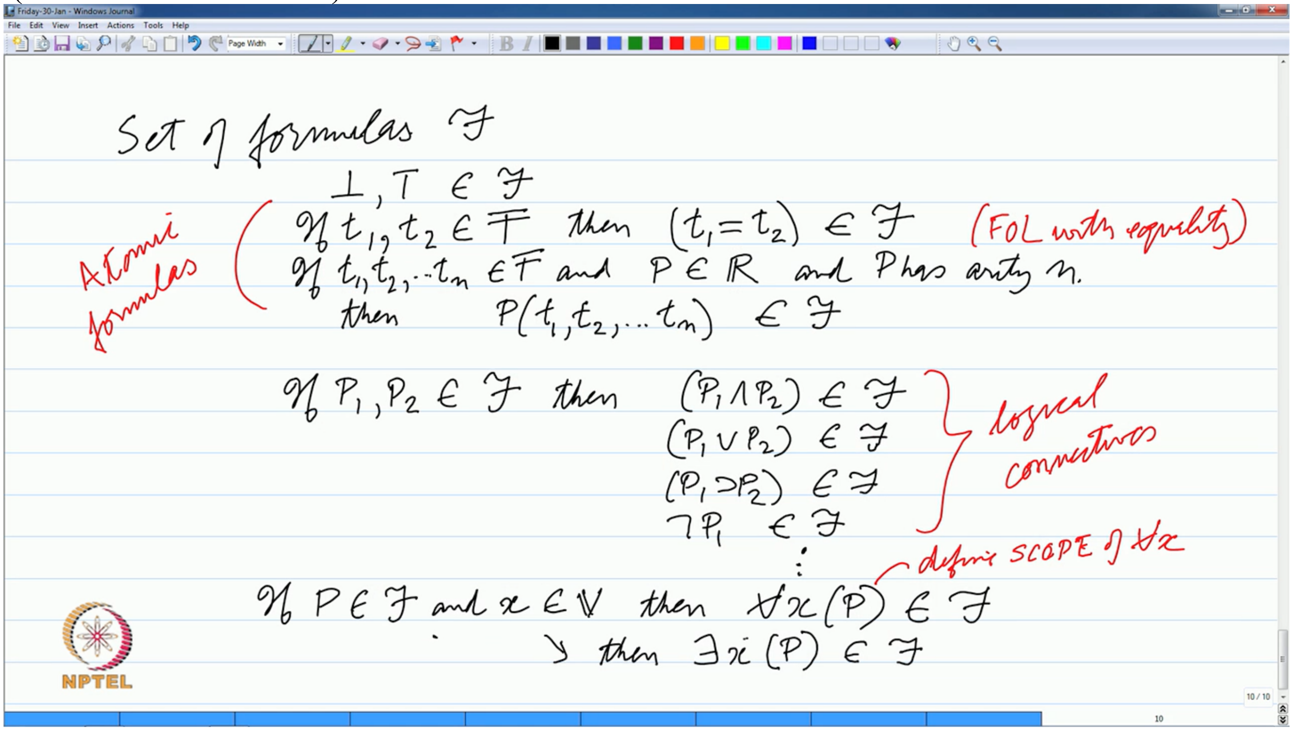
logical connectives

If $P \in \mathcal{F}$ and $x \in \mathcal{V}$ then $\forall x (P) \in \mathcal{F}$

Notice that P may or may not have x inside, it does not really matter. It's a formula. Any formula you can similarly.

We use the brackets to define a scope of this for all x which means that this for all quantifier, these two are known as quantifiers, existential quantifiers and universal quantifiers. They have a scope and this applies to whatever is in the bracket essentially. Then so the same thing follows there exists x P. So the two quantifiers we have, for all x and there exists x, can be used to consider larger formulae. So you can take any formula and prefix for all x or there exists x and you get a new formula essentially.

(Refer Slide Time: 27:20)



Set of formulas $\mathcal{F}$

$\bot, \top \in \mathcal{F}$

Atomic formulas: If $t_1, t_2 \in \mathbb{T}$ then $(t_1 = t_2) \in \mathcal{F}$ (FOL with equality)

If $t_1, t_2, \ldots t_n \in \mathbb{T}$ and $P \in \mathcal{R}$ and P has arity n.

then $P(t_1, t_2, \ldots t_n) \in \mathcal{F}$

If $P_1, P_2 \in \mathcal{F}$ then $(P_1 \wedge P_2) \in \mathcal{F}$

$(P_1 \vee P_2) \in \mathcal{F}$

$(P_1 \supset P_2) \in \mathcal{F}$

$\neg P_1 \in \mathcal{F}$

logical connectives

$\vdots$

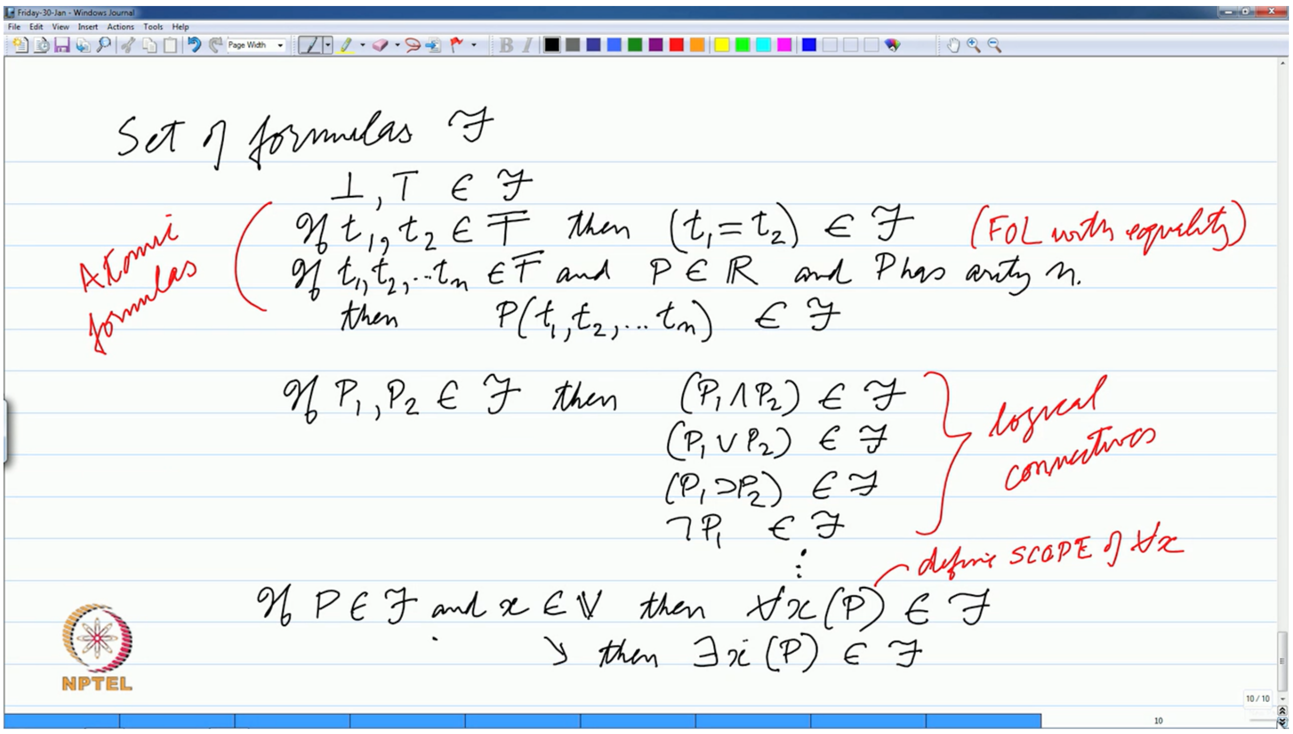
If $P \in \mathcal{F}$ and $x \in \mathbb{V}$ then $\forall x(P) \in \mathcal{F}$ — define SCOPE of $\forall x$

then $\exists x(P) \in \mathcal{F}$

So in the logic that we are working with we are using these two quantifiers but there is no particular restriction that these are the only two quantifiers. People you know have talked about other kinds of quantifiers as well. So if you look at some basic books on logic for example some books would talk about another quantifier which says that there exists a unique x. But anyway we will look at these two quantifiers.

So we have now defined a set of formulas. The formulas that we have are defined by a set of atomic formulae which says top and bottom are atomic formulae, a term equal to another term is an atomic formula and then if you take a predicate symbol supply appropriate number of terms to that then we get another atomic formula. Then we can use any formulae and use logical connectives to form a larger formula, and then we can apply the two quantifiers that we have to get more formulas. So all this defines the set of formulae that we have.

(Refer Slide Time: 28:25)



Set of formulas $\mathcal{F}$

$\bot, \top \in \mathcal{F}$

Atomic formulas

If $t_1, t_2 \in \mathcal{T}$ then $(t_1 = t_2) \in \mathcal{F}$ (FOL with equality)

If $t_1, t_2, \ldots t_n \in \mathcal{T}$ and $P \in \mathcal{R}$ and P has arity n.

then $P(t_1, t_2, \ldots t_n) \in \mathcal{F}$

If $P_1, P_2 \in \mathcal{F}$ then $(P_1 \wedge P_2) \in \mathcal{F}$

$(P_1 \vee P_2) \in \mathcal{F}$

$(P_1 \supset P_2) \in \mathcal{F}$

$\neg P_1 \in \mathcal{F}$

$\vdots$

logical connectives

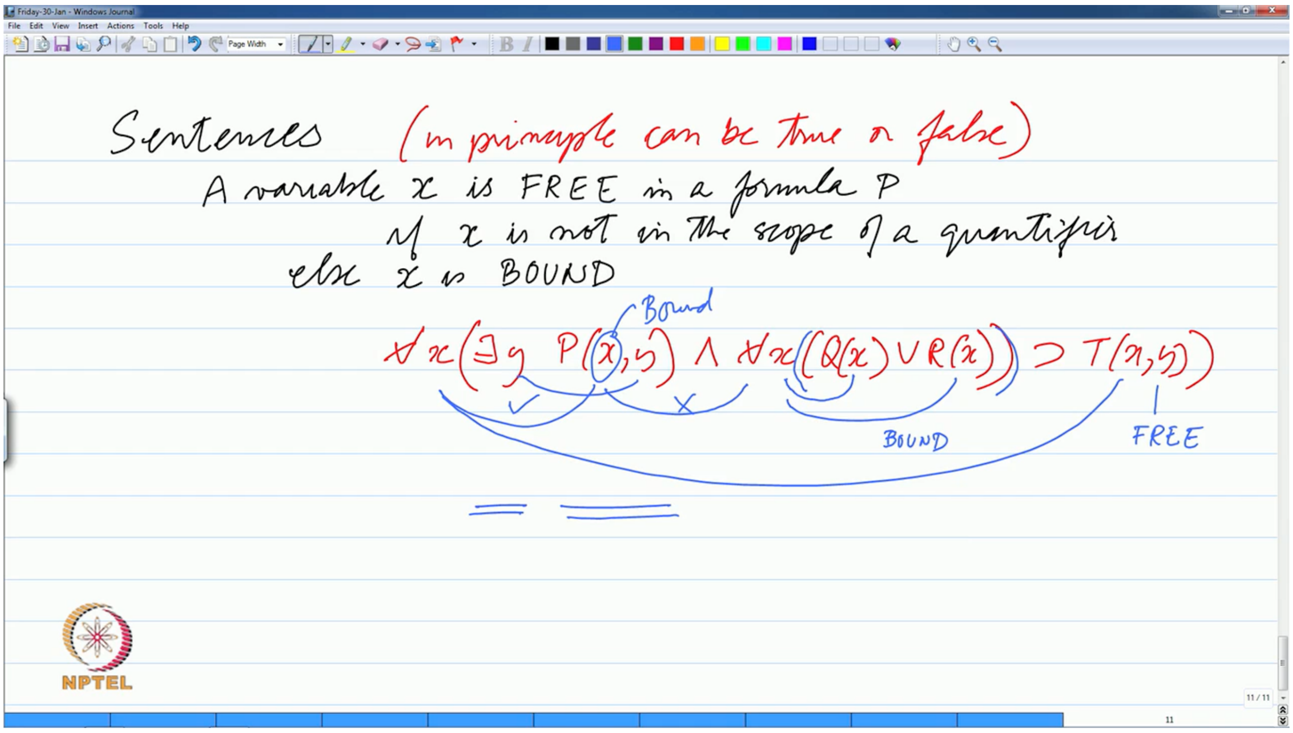
define SCOPE of $\forall x$

If $P \in \mathcal{F}$ and $x \in \mathbb{V}$ then $\forall x (P) \in \mathcal{F}$

then $\exists x (P) \in \mathcal{F}$

Now let's talk about sentences which is what we are really interested in. So what are sentences? Sentences are those things which in principle can be true or false. It can be mapped to true or false. In next lecture we will define the semantics of first order logic. But we first need to define what we mean by a sentence. To do that we first need to define the notion of free variable. So we say that a variable x is free in a formula, let's call it some formula P if x is not in the scope. Remember we had defined the notion of a scope of a quantifier. So if x is not in the scope of a quantifier, the occurrence of x is not in the scope of the quantifier then we say that x is free. And otherwise we say x is bound. So let me give you an example. Supposing we have a formula which looks like this for all x there exists y P x y and for all x Q x or R x implies let's say T x y. So some random formula I have written. So we need to identify which occurrences of variables are free and which occurrences are bound. So remember that if a variable is in the scope of a quantifier it is bound otherwise it is free. So you can see that this for all x is bound by this quantifier here because it comes in the scope here. Likewise, this is bound by this. So these are bound. This x is also bound but it's not bound by this one but it is indeed bound by this one. I will just put a cross on that. So this is this x is bound. Likewise, this y is bound because it comes within the scope of this there exists y. It is bound by this quantifier. This x, the last x we have is also bound by this. But this last y is free. Because the scope of this existential quantifier the last y that we have extends only up to in face this one. So the scope of this y is only up to T x y. So that's why the last y is free

(Refer Slide Time: 33:09)



Sentences (in principle can be true or false)

A variable x is FREE in a formula P

if x is not in the scope of a quantifier

else x is BOUND

$$\forall x(\exists y\, P(x,y) \wedge \forall x((Q(x) \vee R(x))) \supset T(x,y))$$

So in this way you can basically identify free variables and bound variables, and a sentence of a FOL is a formula of FOL without any free variables. So any formula in which there are no free variables is a sentence of first order logic. And having chosen a set of relation symbols, a set of function symbols, a set of constant symbols we can now define a whole set of sentences. So that's a language that we have. These are the elements of our language. Atomic formulae, compound formulae, quantified formulae, as long a these are formulae which don't have free variables it's a set of sentences. So in the next class we will talk about the semantics of first order logic. And we will see that whenever we devise a first order language we are talking of certain domains and we are talking about relations in the domain.