Artificial Intelligence: First Order Logic Semantics

Prof. Deepak Khemani

Department of Computer Science and Engineering

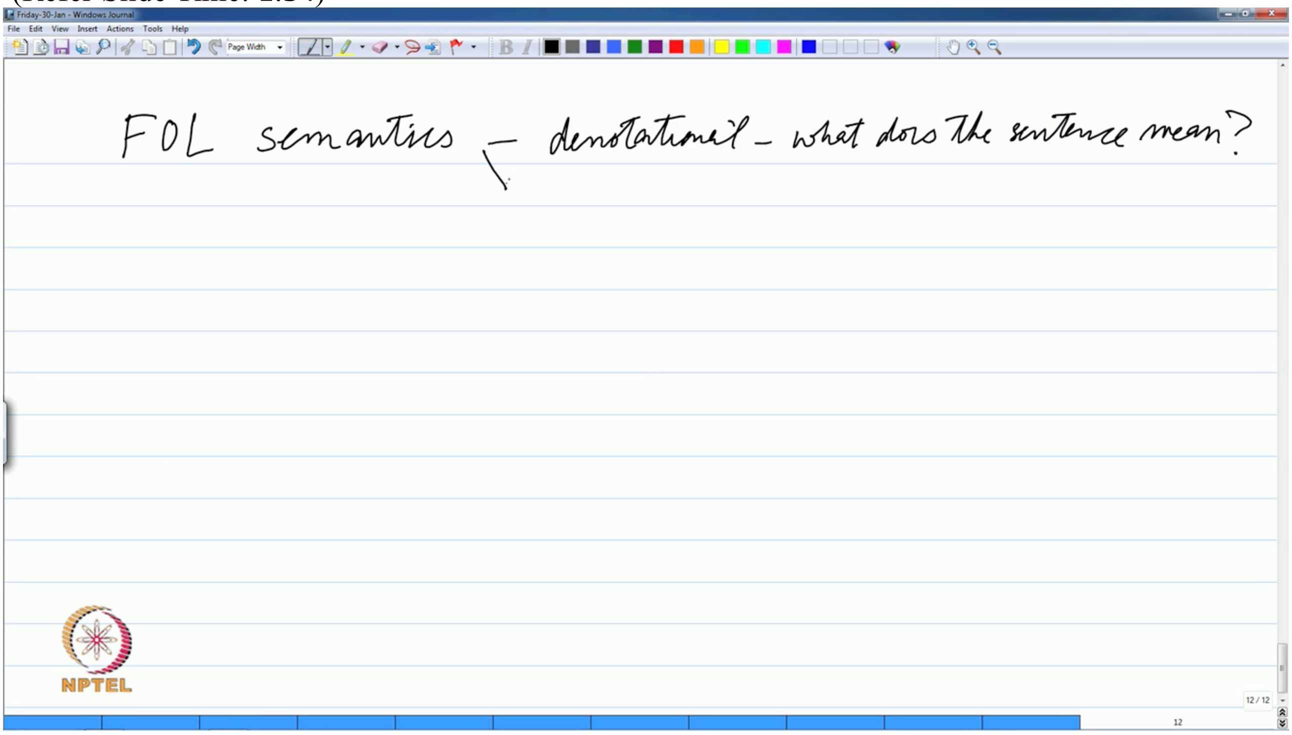
Indian Institute of Technology, Madras

Module – 03

Lecture - 02

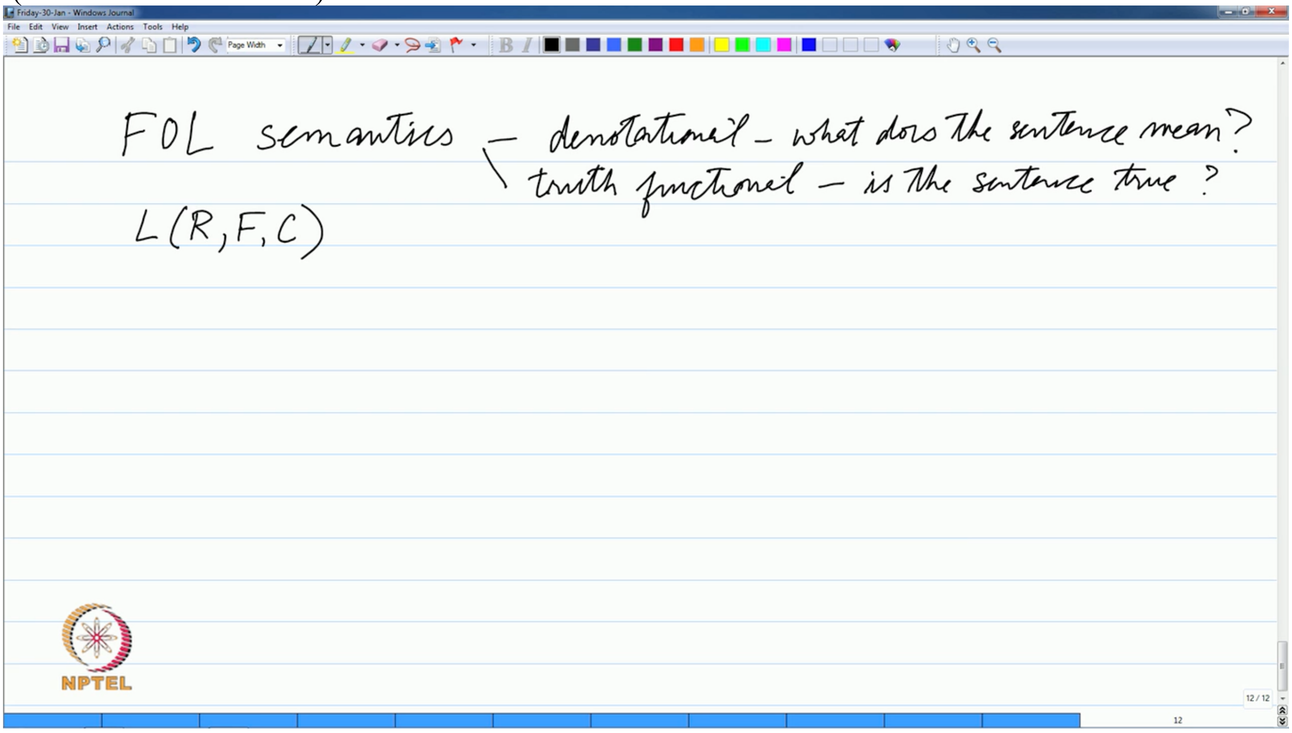
Ok so in the last class we defined the language for first order logic, the syntax of first order logic. And today we want to look at the semantics. Now there will be two aspects of semantics of first order logic. One is what we can say **Denotational** which will ask the questions what does the sentence mean. As opposed to propositional logic where we say that a formula stands for a sentence, a proposition stands for a sentence and P and Q stands for P is true and Q is true and so on so forth. But in first order logic we may really want to say what are we talking about. Are you talking about men and mortality and honesty? Or are are talking about numbers and some properties of numbers, prime numbers, even numbers, odd numbers. What does the sentence mean thats one aspect of the semantics?

(Refer Slide Time: 1:34)



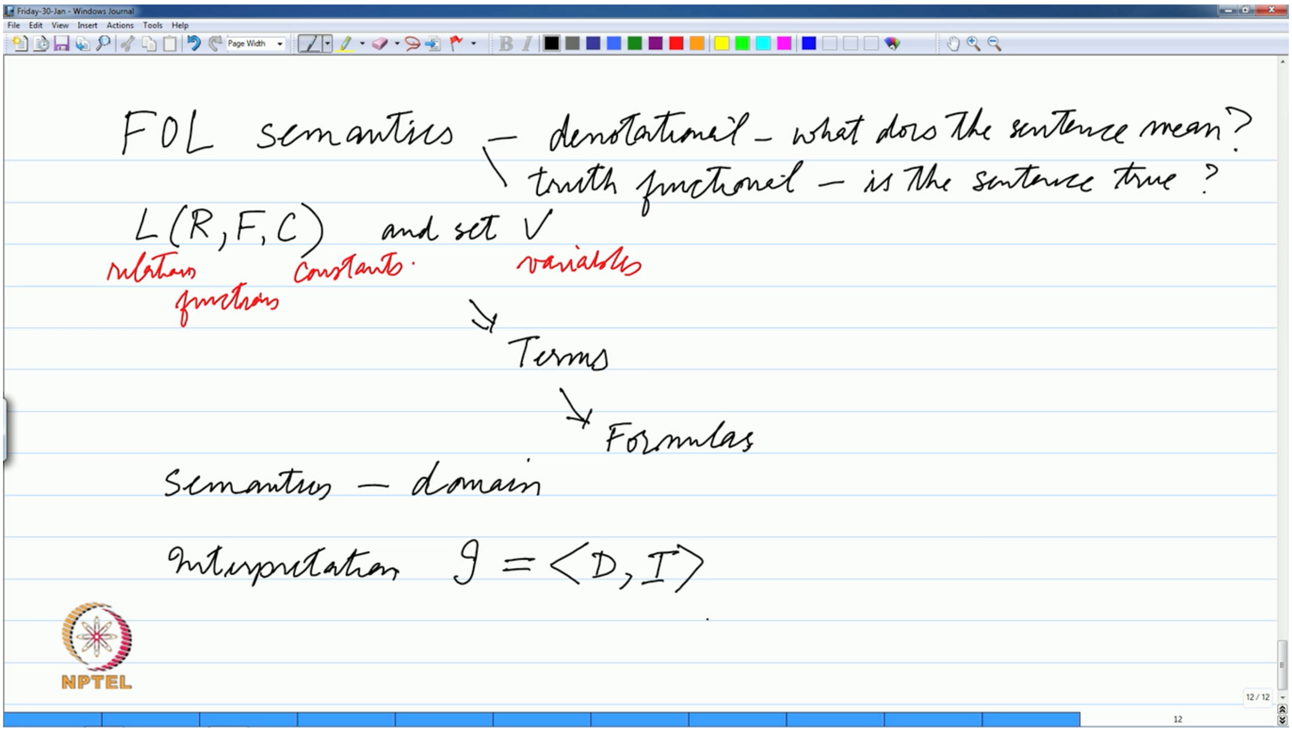
Other aspect that we are already familiar with is **truth functional** which is basically asking the question is the sentence true? So we look at both these aspects. Now remember that the language of first order logic in fact the language is defined by the three sets. So instead of using those complex characters for that we will use the notation R, F, C where R is the set of relation symbols, F is the set of function symbols and C is the set of constant symbols. Once we choose that set these three sets then you have defined the first order logic language. And of course you have also defined, you have at some point made a choice about what are the logical connectives that you are going to use essentially. We will assume for the moment that we will use all the unary and binary connectives and when we talk about proof systems then we will restrict the set of connectives to a smaller set. So for example when we come to the resolution method we will say that the logical connectives are just not, and, or.

(Refer Slide Time: 3:00)



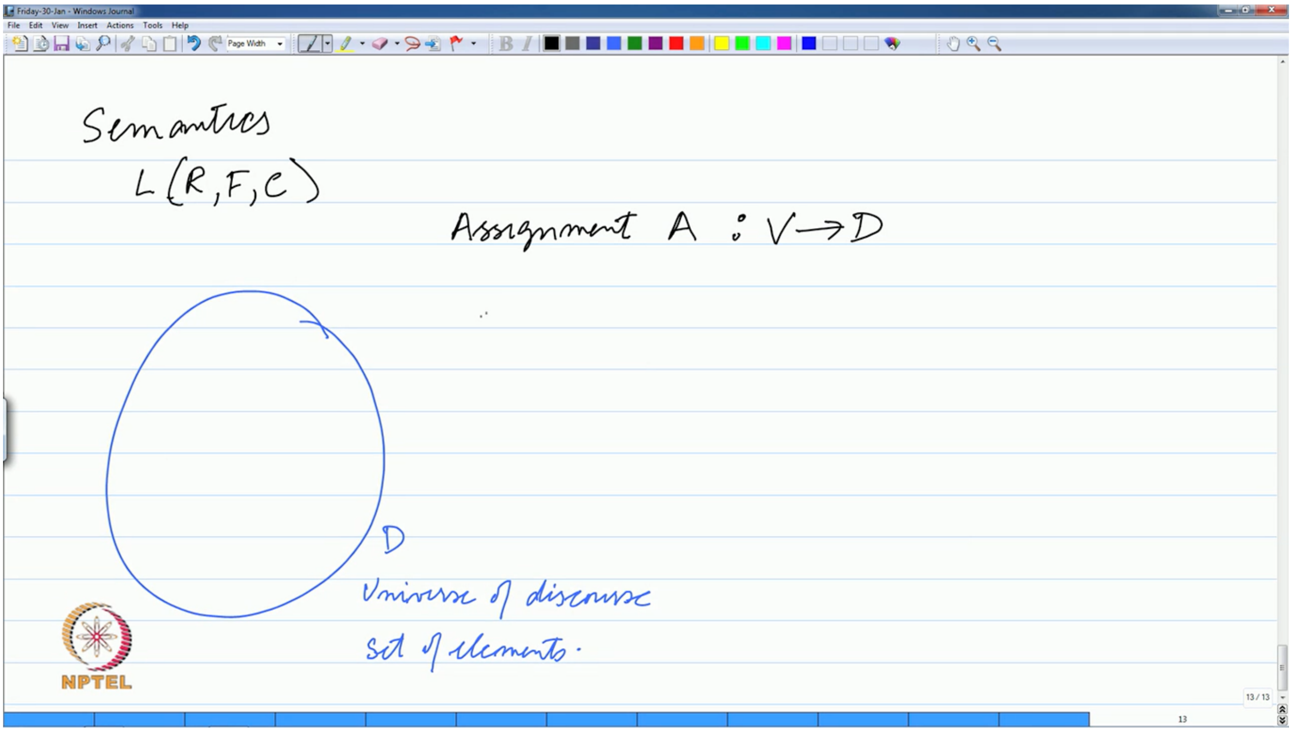
So we have these three sets R, F and C and we have the set V. So I am just using different characters here so these are the variables. These are relations, functions and constants. And then we constructed terms out of this language and then we constructed formulas. Now the semantics of the meaning of first order logic is defined in terms of a domain. And we say that we interpret the expressions, the terms and the formulas and the sentences over a domain. So we will talk of an interpretation I which is made up of a domain D and a mapping which we will also call I which is the interpretation mapping.

(Refer Slide Time: 4:47)



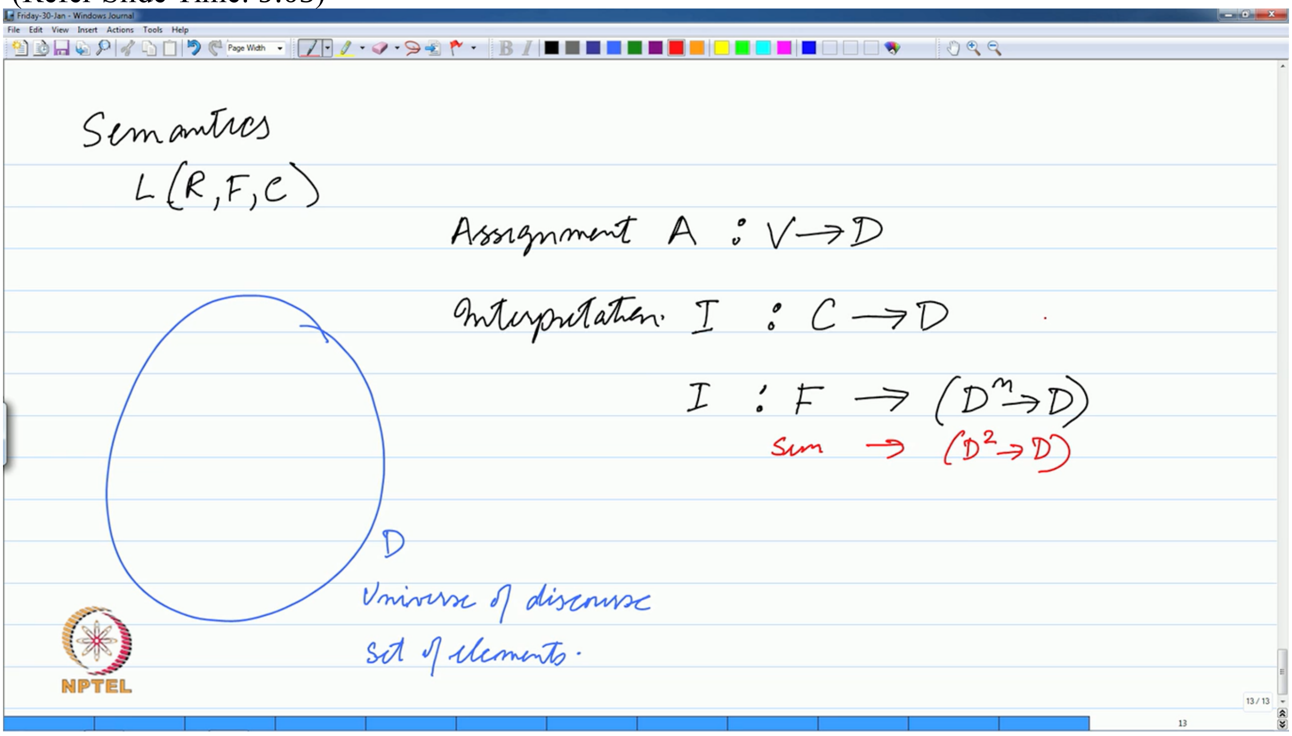
And we will basically be able to interpret every sentence, every formula in terms of the domain. So we have a language and now we are saying that we will define the semantics in terms of the domain D. So a domain is basically also known as the universe of discourse which is a set of elements. Now we will define an assignment function A which will map the set of variables V onto the domain D. So we can choose just like in propositional calculus we chose a valuation function which maps every propositional symbol to a value truth or false. In this case we are now saying that to every variable in my language I can map it to an element in my domain. It's like saying for example x is mapped to 4 or x is mapped to 5 or x is mapped to Sujeet. Of course we can talk about the meaning of sentences under different assignment. So every time we choose an assignment function A it means that for every variable in my language, so I may have x1, x2, x3..some finite number of variables, I am saying that this is what they stand for in the domain. Everything that we are talking about in FOL semantics is going to be with respect to this domain D essentially.

(Refer Slide Time: 07:00)



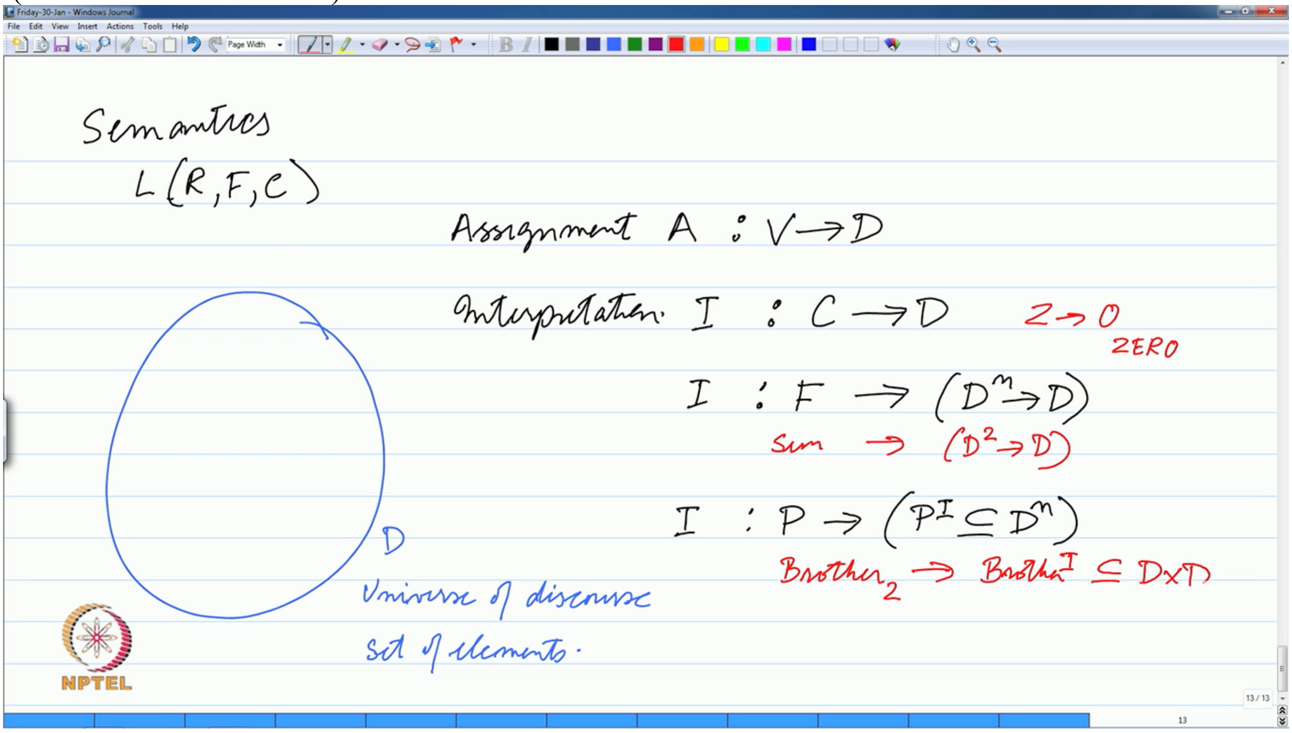
Then we have the interpretation function I where it does the following. It maps every constant to D. Just like a variable the interpretation is telling us what does this particular constant stand for. Its not something which is a variable so assignment is done only for variables. It maps every function symbol to an appropriate function which we write a follows. D raise to the power n to D. This is some notation that I am inventing on the fly. But essentially what we are saying is that every function symbol in my language will be mapped to some function the domain. And the function in the domain of arity n is a mapping for D raise to n to D. So for example, the sum, if I write sum then it could be mapped to D square. It takes two arguments and returns one argument. And that argument belongs to the domain essentially, it's a function from elements to domain.

(Refer Slide Time: 9:05)



So the function could be for example a symbol for example Z could map to 0. And then we map every predicate symbol to a relation which we will call as P i which is a subset of D raise to n. Again this is nonstandard notation. So essentially every predicate symbol will map to a relation on the domain. So for example I may have a predicate symbol for brother which has arity 2, it will map to D2. It will map to a symbol which let me just call it Brother i which is essentially a subset of *D cross D or a subset of D square essentially.*

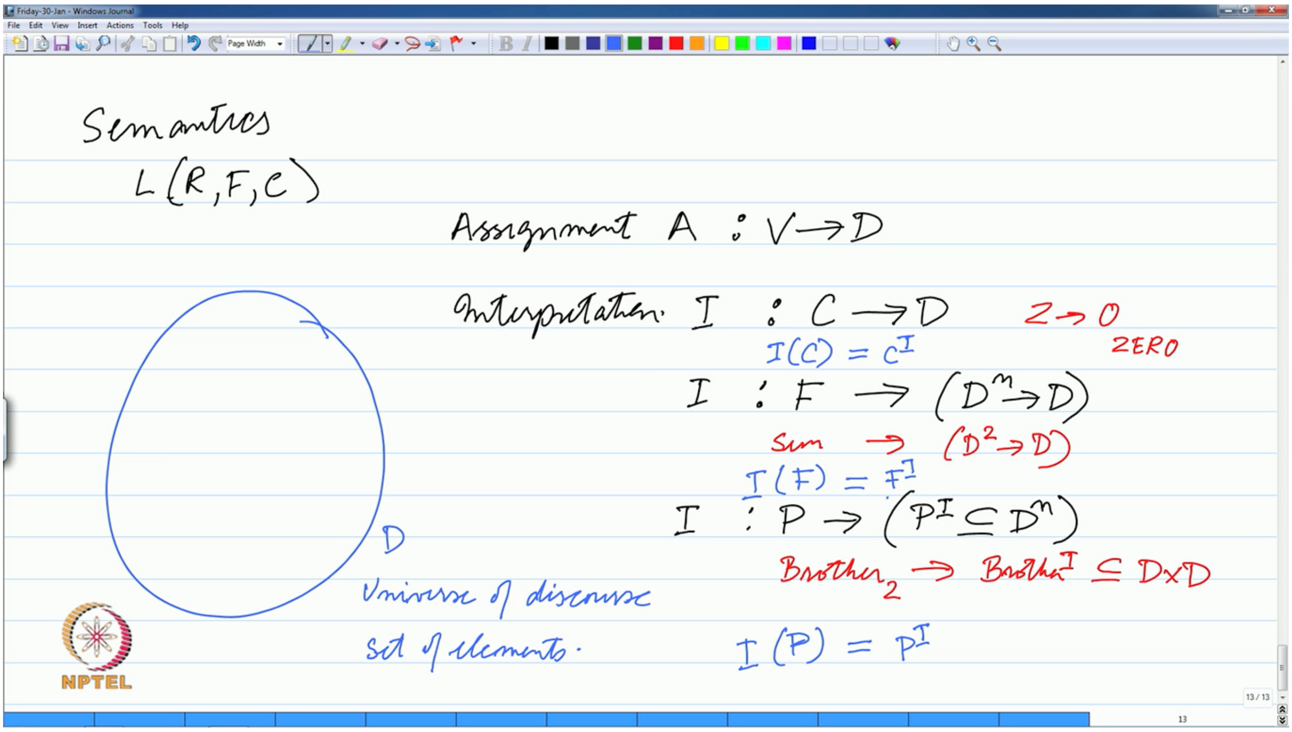
(Refer Slide Time: 10:31)



Because the brother relation is a binary relation and it relates two elements of the domain. And the relation will actually be an enumeration of all those elements which are related by the relation brother.

So for any constant C the interpretation of that constant will be its image in the domain which we will denote by C raise to I. This I is the interpretation function. Likewise, for every function its image under the interpretation mapping would be something which we simply call F raise to I and likewise for every predicate its image in the mapping would be the predicate name or relation name in the domain.

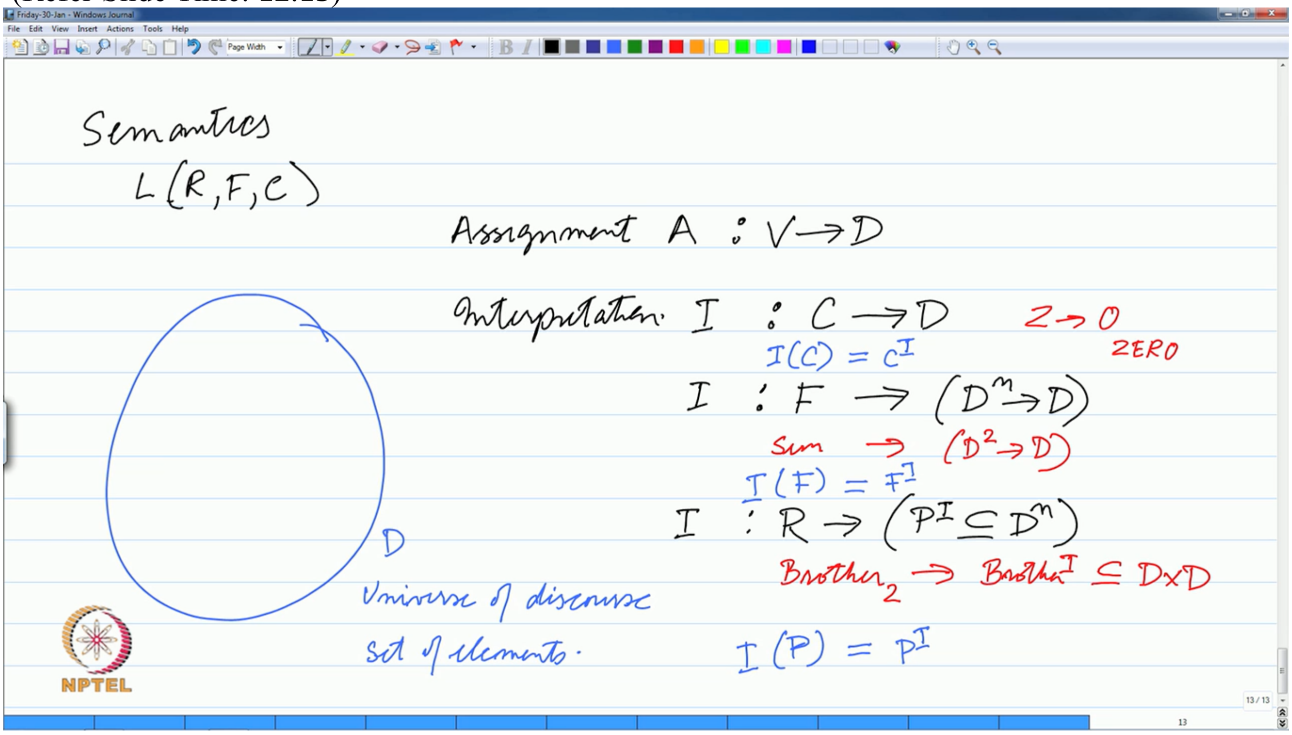
(Refer Slide Time: 11:41)



And this P raise to I will be a subset of D cross N, this F raise to I function will be a function from some N variables to one variable and constant is a function of 0 variable essentially.

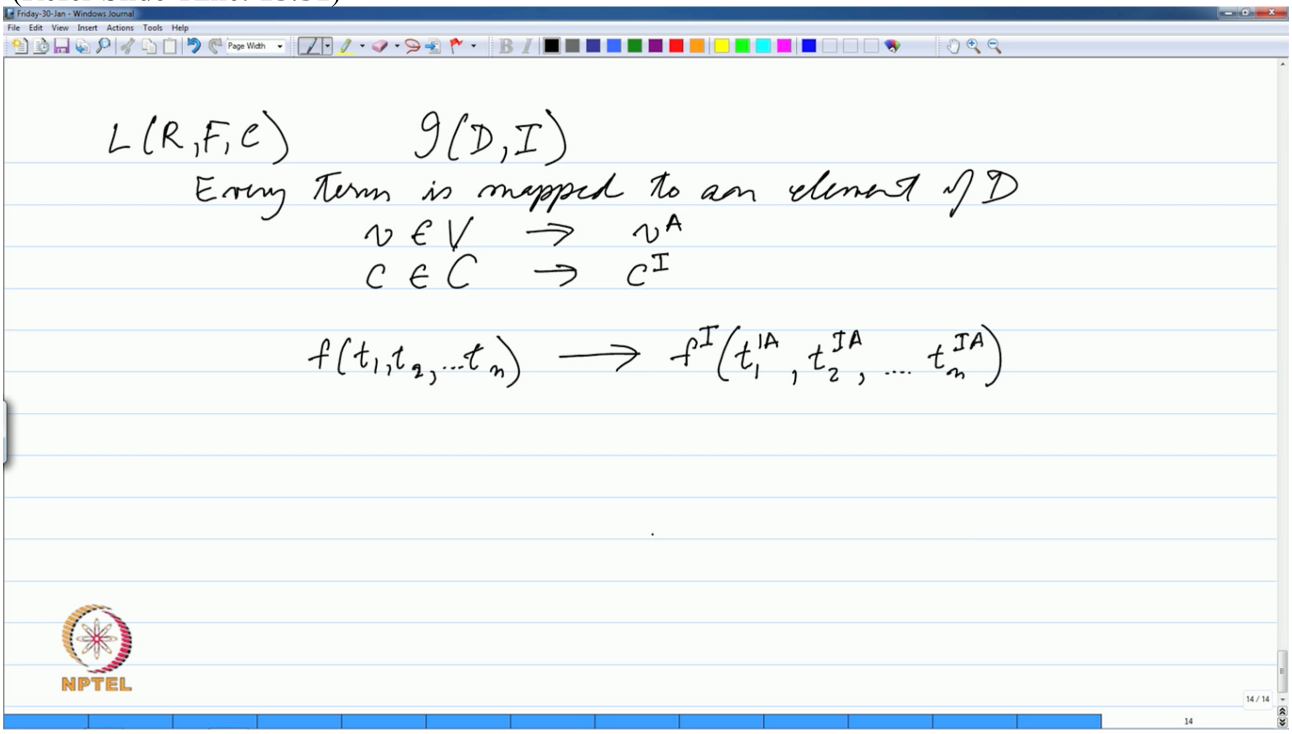
From audience: So C and F and mapped but R is not.
Faculty: So we should map R too? Is that what you are saying? Ok Thanks

(Refer Slide Time: 12:15)



So we have these two mappings now. So given a language *L R F C*, we have an interpretation which is given by D and interpretation mapping I in which every term is mapped to an element of D which we write as follows that if the term is a variable v belongs to the set V is mapped to its image which is given by its assignment which we will write as v A another assignment A. Every constant is mapped to c I. Every term belonging to T is mapped to. So let me just define that again. So if the terms are variables and constants then they are simply, the mapping is defined by the assignment function and the interpretation function. If the term is of the kind of t1, t2, tn then that is mapped to the function symbol which is interpreted by I by the terms t1 which is mapped by I and A. Remember a term can be either. It may have variables it may have constants and so on so we have to use both the mappings t2 I A .. tn I A where this notation t1 raise to I A stands for the fact that the term is interpreted under the mapping, the two mappings that we have, the assignment mapping which maps variables to elements and interpretation mapping which tells us what do we mean by a function name and what do we mean by a constant name and what do we mean by ... In this case only function and constant names essentially.

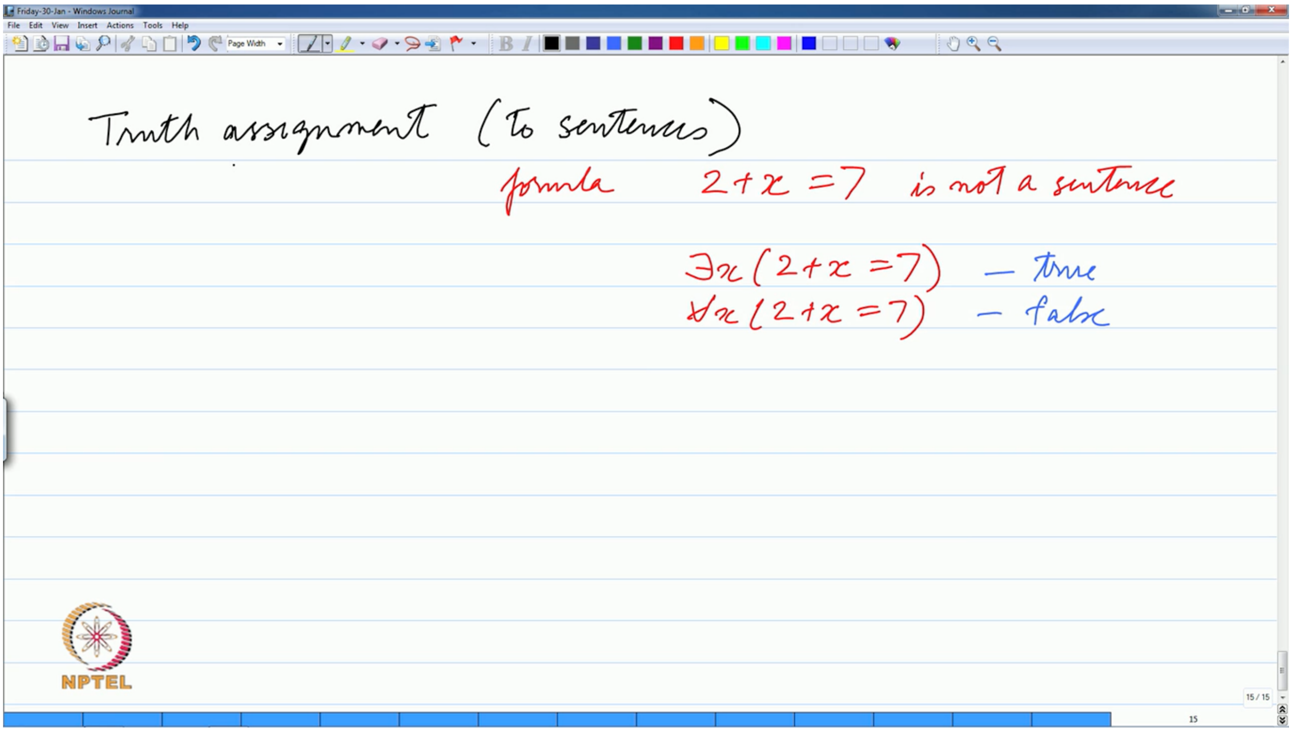
(Refer Slide Time: 15:31)



We have already said that the relation symbols are mapped onto let me write that here. Formulas are mapped to relations on the domain of appropriate arity and so on and so forth. So let's look at the truth functional aspect of this thing and remember that truth assignment is done only to sentences of the language essentially. So a formula like for example 2 plus x is equal to 7 has no truth value it is not a sentence. Why is it not a sentence? Because it has a free variable which is x which is not quantified essentially which means you can assign a truth value to this sentence essentially it really depends on the assignment x essentially.

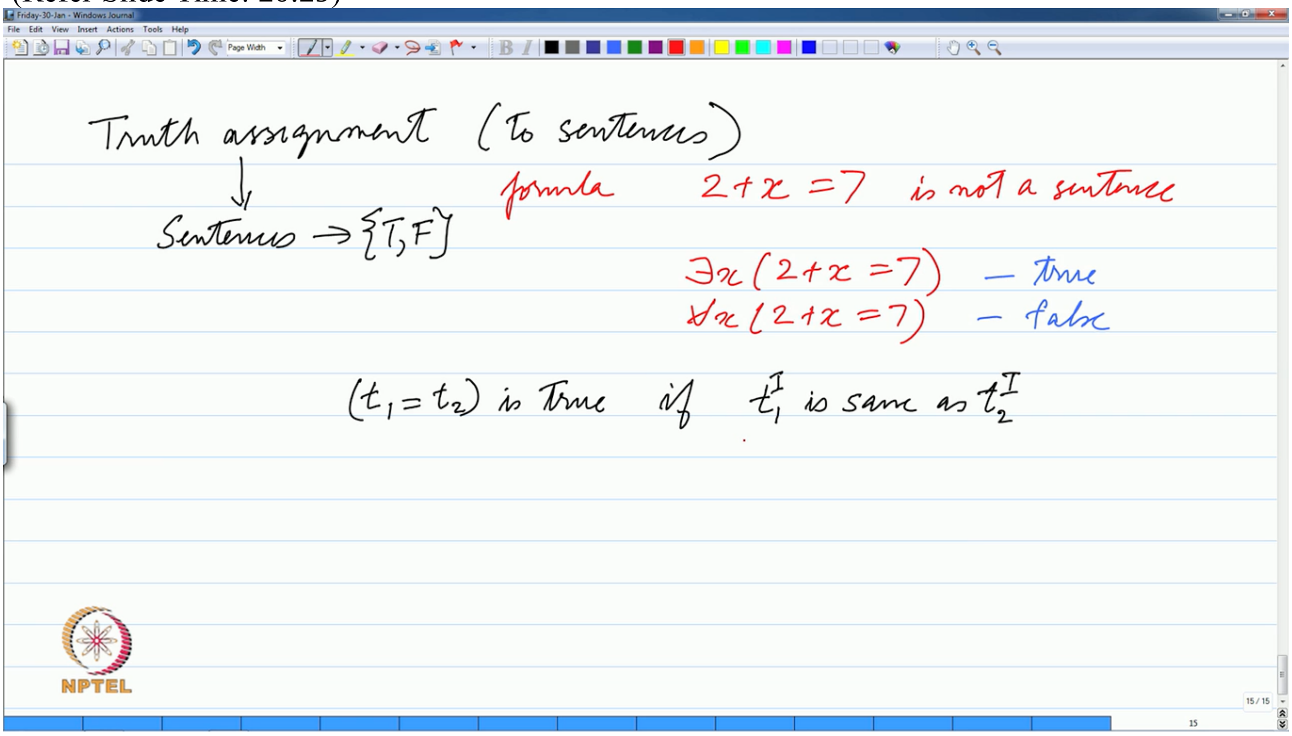
Whereas if I had a formula like there exists an x such that 2 plus x is equal to 7 or if I had said for all x 2 plus x is equal to 7, that is also a sentence because now the only variable in my formula is x and it is quantified in both the formulas that I have and it turns out as you know that in this example this sentence must be true and in this example this should be false. We will come, we will shortly define how do we come to this mapping essentially but if you look at it informally the first formula or the first sentence is saying that there exists an x such that 2 plus x is equal to 7 and therefore the sentence must be true. The second formula is saying that for all x 2 plus x is equal to 7 and therefore that formula must be evaluated to false. So we want to now come to a formal definition of how do we arrive at this truth function semantics.

(Refer Slide Time: 18:31)



So as before when we talked about truth semantics we are talking about an assignment or an function which will map the sentences to a set which as before we will use the terms T and F, true formulae and false formulae essentially. So how do we do that? Let's first talk about atomic formulae. A formula like t1 is equal to t2, this is an atomic formula is true which means its mapped to this symbol T if the interpretation or the image of the first term both under interpretation and mapping. Remember this term may have variables inside essentially and if it has variables inside essentially then it will not be sentence so we will just talk about the interpretation function. So the term under the interpretation I is same as the term 2 in the interpretation I essentially.

(Refer Slide Time: 20:29)

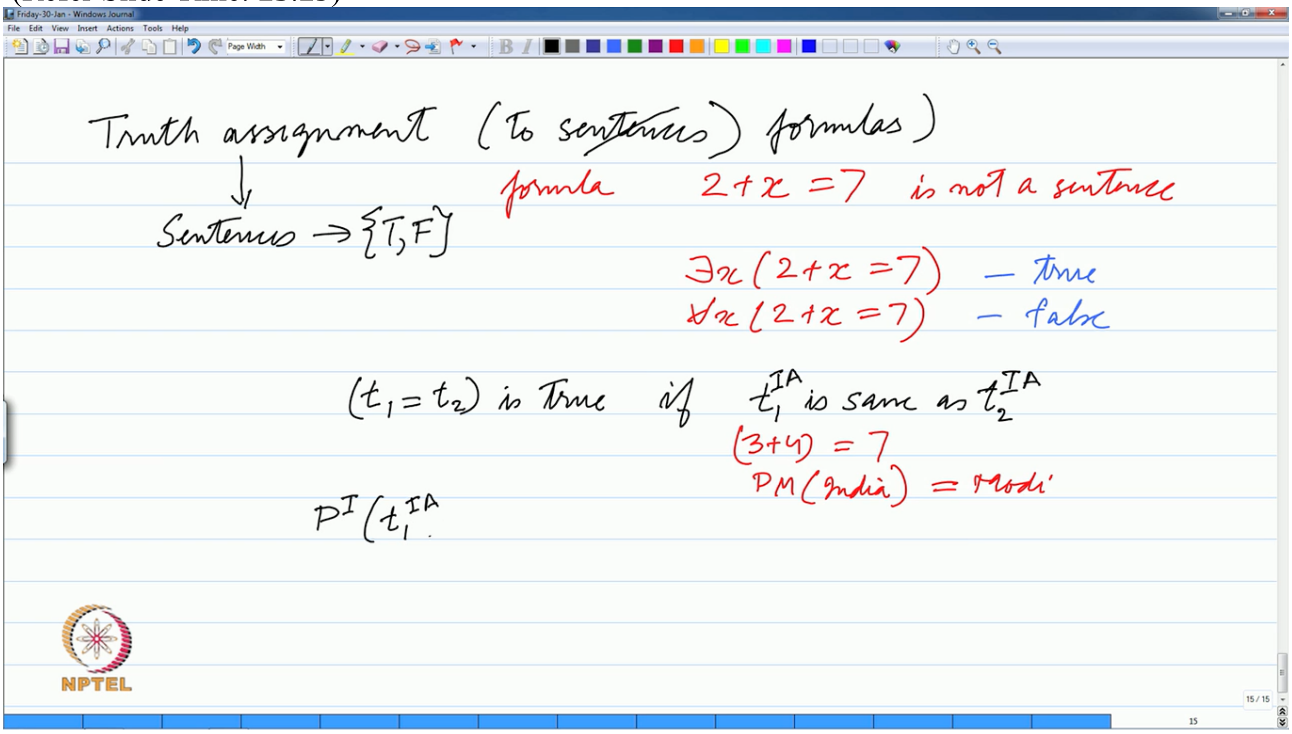


So as an example we can say that 3 plus 4 is equal to 7. Where 3 and 4 and 7 are let us say constants in my language. And the meanings of constants are determined by the interpretation function which corresponds to the numbers 3 and 4 and 7 so we must distinguish between numbers and their representations. What we are talking about in the language is only the representation and we are saying that if the two elements are the same then this thing.

If I say PM India is equal to Modi where PM is a function which takes a country and tells you who is the prime minister of that country. Remember the terms point to the elements of the domain and functions basically are defined terms and Modi is a constant so this sentence happens to be true at this particular moment.

Then an atomic sentence of the kind P under an interpretation I with some terms...actually to me more kind of complete we can avoid our restriction to the set of sentences but instead we will say to the set of formulas. So we can even still talk about formulas being true or false which means that my original notation which is both under interpretation and assignment is true or false essentially. So which means that a formula like 2 x + 7. 2 plus x is equal to 7 would be true under some assignment but it may be true under some other assignment. So the assignment in which x is mapped to 5 that formula will be true. So we can essentially talk about the truth value of formulae also but when we talk about the largest family of formulae then we have to talk about the assignment function essentially. When it comes to sentences we don’t need assignment functions because there are no free variables. But if there are free variables then we have to consider an assignment into place. So we will do that here as well.

(Refer Slide Time: 23:29)



So atomic formula with the predicate symbol P or relation symbol P would map to true if the tuple that we form t1 I A t2 I A ...tn IA belongs to the image of relation symbol Pi. So we are still talking about formulae in general essentially. Then logical connectives are treated as in propositional logic. Which means the semantics of and or implies negation are treated as before essentially.