# Artificial Intelligence: Propositional Logic: Language, Semantics and Reasoning

**Prof. Deepak Khemani**

**Department of Computer Science and Engineering**

**Indian Institute of Technology, Madras**

**Module – 02**

**Lecture - 01**

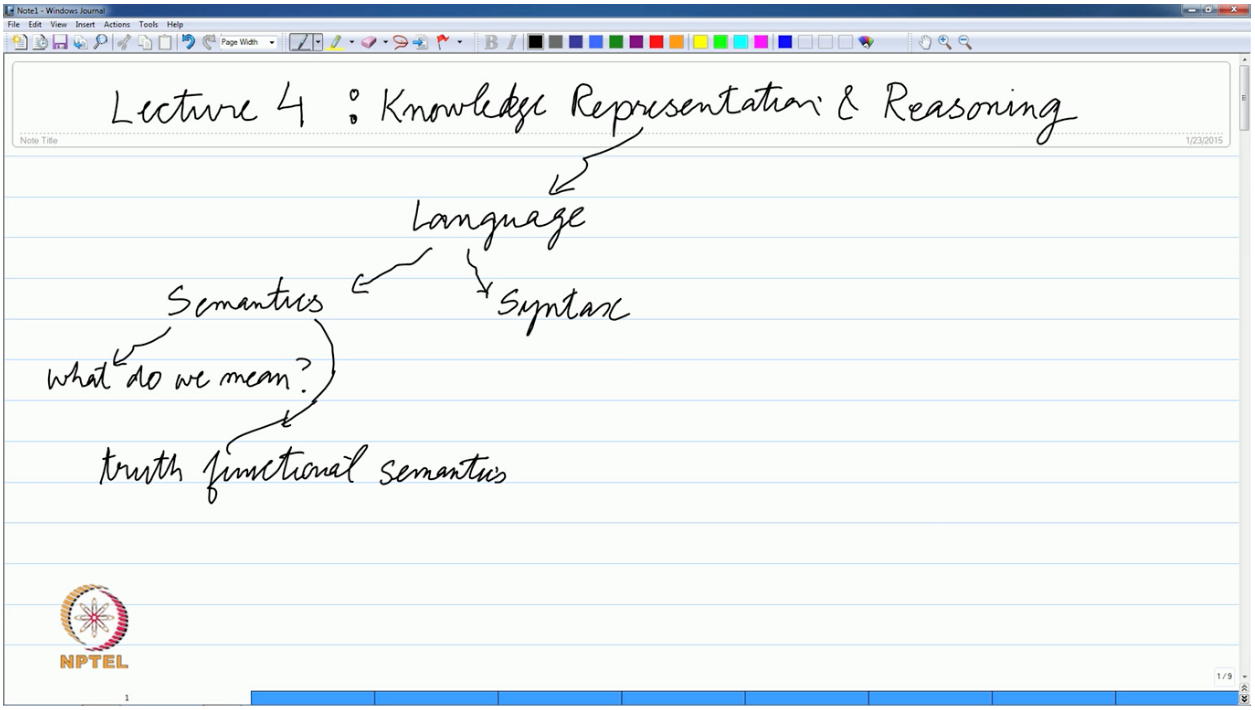
## Module 2 Lecture 1

In the last few classes we saw some history that led up to the topic we are interested in which is knowledge representation and reasoning. You saw that you know for many centuries people have been interested in how do we know what we know how can we present that and also quite interestingly how do we what do we mean when we say something. How do we mean what we mean? So that meaning part is a little bit of a conundrum we will see we will try to define semantics in some way but we are more interested in how can we represent things for the machine and how can we make machine reason about what we are representing. Of course we will keep addressing the notion of meaning along the way. So as far as representation goes what we need first is a language so we when we say a language what we mean like the formal language you must have studied somewhere. Language will consist of two parts one is the syntax and other is semantics. So we are of course interested in semantics because eventually whenever we write computer program we want them to do something useful for us. And we want it to be clear among everybody that what is the meaning of what we have written.

Now in this context I may say that when you often write programs you use words which are meaningful to you so instead of writing I and j you might use the word like count or total marks. Or marks for variables and things like that and you may store some number in that. Now when we talk about semantics we don’t really look at the English meaning of that word. So if I write count is equal to count plus 1 its only for my benefit that I am using the word count. Or if I say total marks is equal to quiz1 marks plus quiz2 marks plus end sem marks its only for my benefit that I am saying that the word marks has no meaning here. This is not what we mean by meaning so we will try to define meaning little bit more formally when we move on to first order logic.

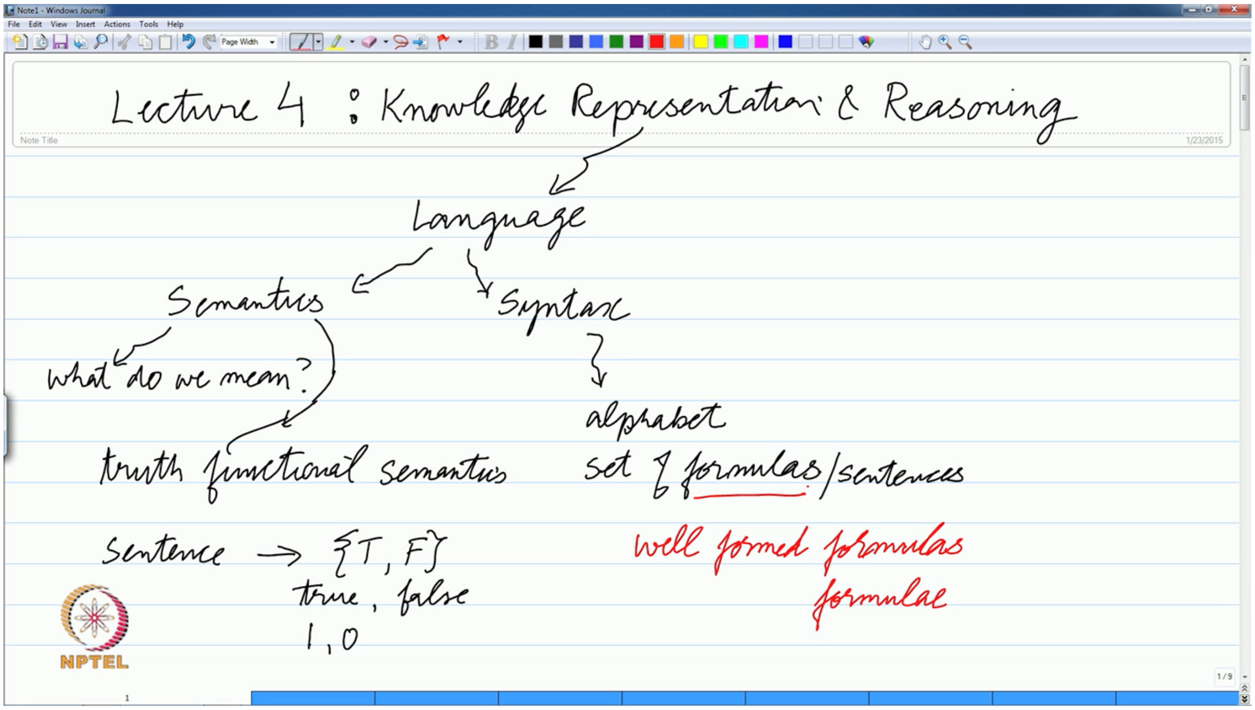
Today we will start with the simplest of all languages which is propositional logic. So just before that there are two aspects to semantics one is what do we mean by whatever we are expressing and second is what you can call truth functional semantics. By this we are concerned with truth values of statements.

(Refer Slide Time: 3:47)




So we said that we are looking at classical logic or truth value logic in our case every sentence will be mapped either to a value true or a value false. So which we could also write as true or false or we could write as 1 comma 0. So basically it is a two valued set one of them which we interpret to stand for sentences being true and other stands for sentences being false. Now when it comes to syntax of the language we need an alphabet and from this alphabet we construct a set of formulas or sentences. We often use the term well-formed formulae. So it's is conventional to use the word formulas as a plural though their older style used to be formulae. Doesn't matter we will use the term formulas. So this is what we mean by the term formula.

(Refer Slide Time: 5:45)




When we say formula we mean well-formed formula which basically means that a word in the language that you are trying to define. When it comes to reasoning we will basically mean inference which we can interpret as saying what else do we know. So the basic idea behind this whole effort is that given that you know some things you know some facts what else can you say that you know. So we saw the example of the Socratic argument of syllogism in the examples earlier where we said that if we know that all men are mortals and if we know that Socrates is a man then we also know even though it Is not stated that Socrates is a mortal.

So that's the kind of reasoning we want to do. And we will see that reasoning can be of different types in some places you are sure that what you infer is true in other places it may not be true. We also mentioned that there is a whole family of languages which logicians have defined and they differ in how much do they allow you to express. But the basic unit of logic is something called as sentences so a knowledge base is a collection of sentences. Now another thing that I should probably highlight at this very moment that whenever we talk about reasoning we are dealing with only syntax. Any computer program that you write is a syntactic entity.

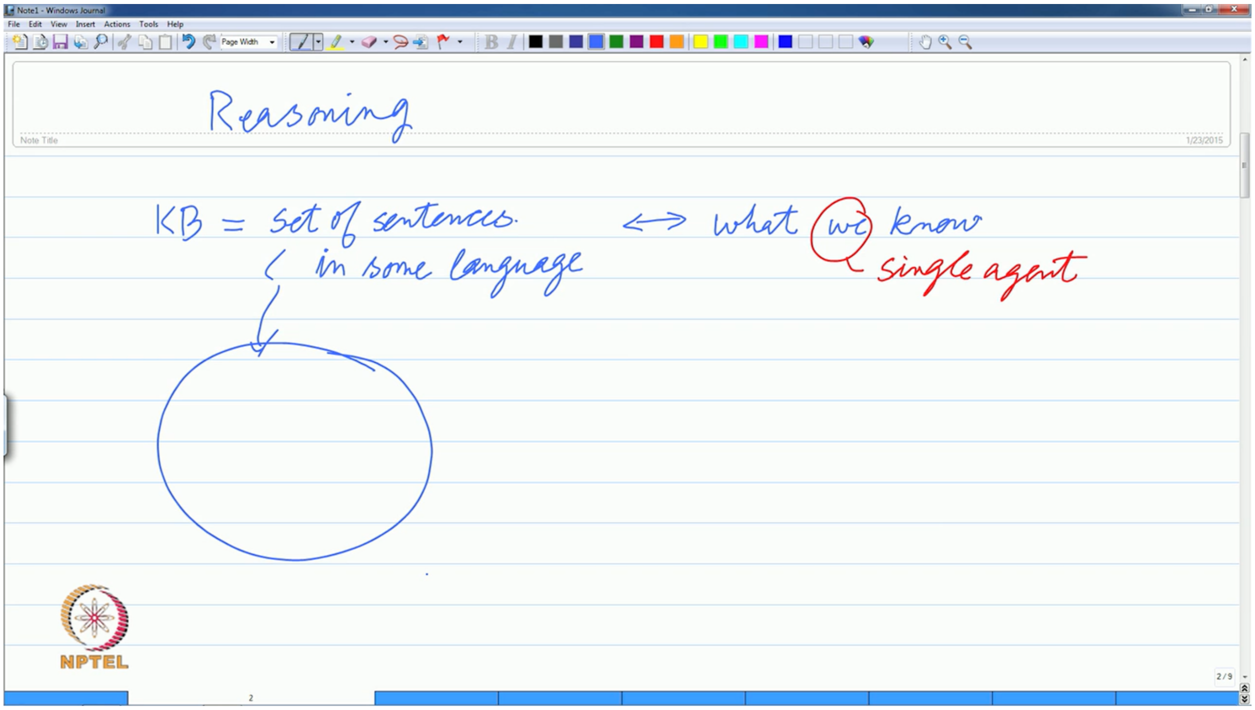
It looks at a certain pattern and according to rules that you have written it produces certain new patterns. It does not look at the meaning at all. And we will see that we will associate the word reasoning with proof procedures and since we are interested in automated reasoning since we want computer programs to do the reasoning for us we will be interested in those approaches or those algorithms which can be implemented over these logic systems. So at the beginning may be today sometime I will show you that there are proof methods which are difficult to implement in algorithms because they involve making guesses or making choices which is easy somehow for us to do at this moment but it's not easy to put down in an algorithm.

So as we go along this course we will sort of oscillate between representation and reasoning. Representation is our concern as to how can we represent what we really mean or what we want to say. How expressive the language is how can we represent complex facts whereas reasoning is concerned with the proof procedures or algorithms that we will apply on the representation. So when you look at the kind of domain of logic there is a whole class of people who are you may call as theoreticians mathematicians or logicians who are not so much concerned with representations so they will assume that you somehow they tell you this is the language and somehow you represent what you want to say in that language but they are more concerned with reasoning and things like you know whether algorithms which terminate and whether everything can be inferred completeness that we mentioned the last time.

Whereas as somebody who might be wanting to develop applications you would also be interested in representations so how can you represent what you want to represent and we will see that that's quite involved subject. It's not straightforward to represent all kinds of things that they can express in English. English as we mentioned is a or hindi or tamil for that matter are very expressive languages you can say a lot in that but also it is difficult to interpret difficult to parse. So you must have encountered situations where you must have ambiguous sentences which are not very easy to parse.

So let's look at what we meant by reasoning at a very high level. So what we have is a knowledge base so we use the term KB for the knowledge base is a set of sentences. So when you say a set of sentences we also mean in some language. And by we will define shortly what do you mean by language. We will start by defining the syntax of propositional language or propositional logic. So this kind of corresponds to what we know. When I say the word we are basically saying that there is single agent. And in fact we are basically reasoning on behalf of single agent. So whenever we say there is certain sentences in the knowledge base we are saying what does that particular agent know. Now so we can kind of represent a knowledge base via set and it has a set of sentences in that.

(Refer Slide Time: 12:55)



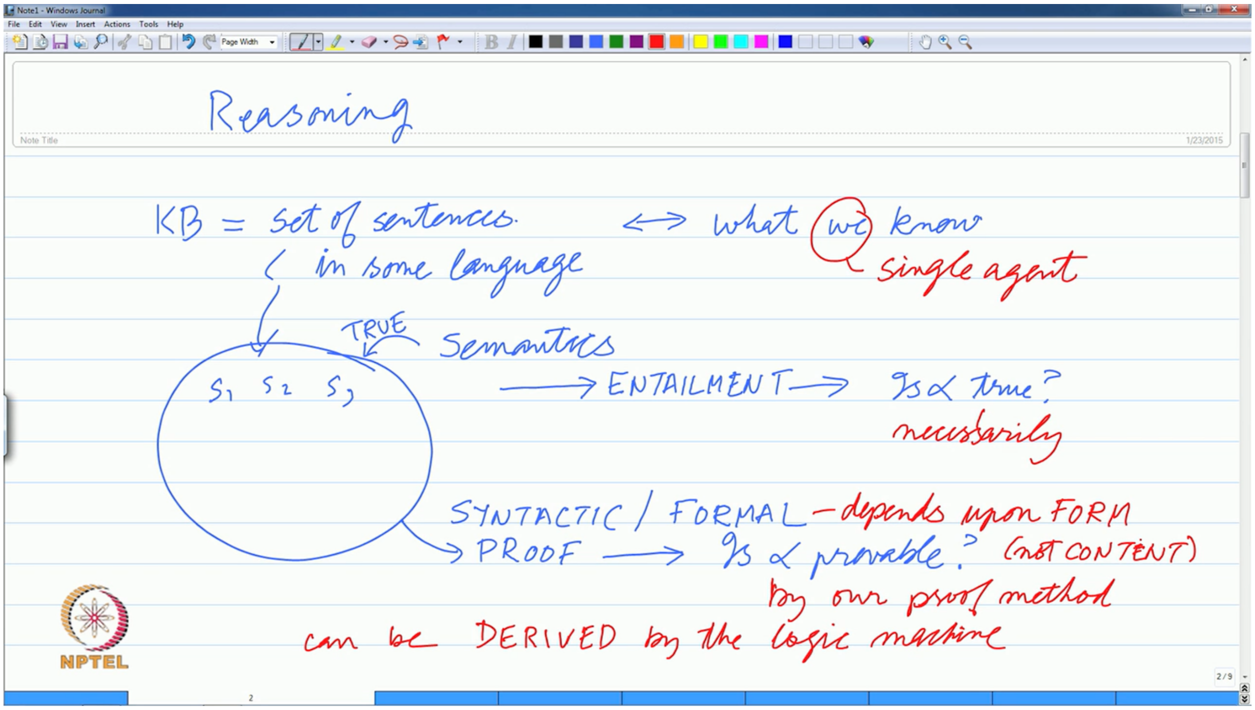

Now the semantics says that whatever is inside this is true. So when you say that you have a set of sentences then you are assuming that what is given to you is true. So if you have this sentence S1 S2 S3 and so on you are saying that what else is true. Then in reasoning what are asking is what else is true that's what we said right. What else do we know. The term that we use is entailment and we want to ask in general whether a given sentence alpha so given a set of sentences which we assume to be true entailment defines a set of sentences which are also true as consequences which are necessarily true as a consequence. That's what we had said.

That's the notion of entailment whereas reasoning is concerned with a proof procedure and the question we ask is is alpha provable? And when you say provable we will say by our proof method. So while on one hand we are interested in the semantics we are interested in saying as to what else is true as a consequence of what we assume to be true or what is given to us to be true. The route to arrive at the truth is going to be via a proof procedure and corresponding question would be what else is true. So when we use the word provable we can use the other phrases like what else can be derived.

So we will imagine that there is an algorithm which we will also call as the logic machine.

And we would be interested in knowing what can be derived in this logic machine. So the logic machine is a syntactic entity. The term that we used earlier when we started talking about logic was formal. We say arguments are valid if they conform to a given form. so by formal we mean depends upon form. and by this we mean not content. It doesn't matter what you are saying

(Refer Slide Time: 17:19)



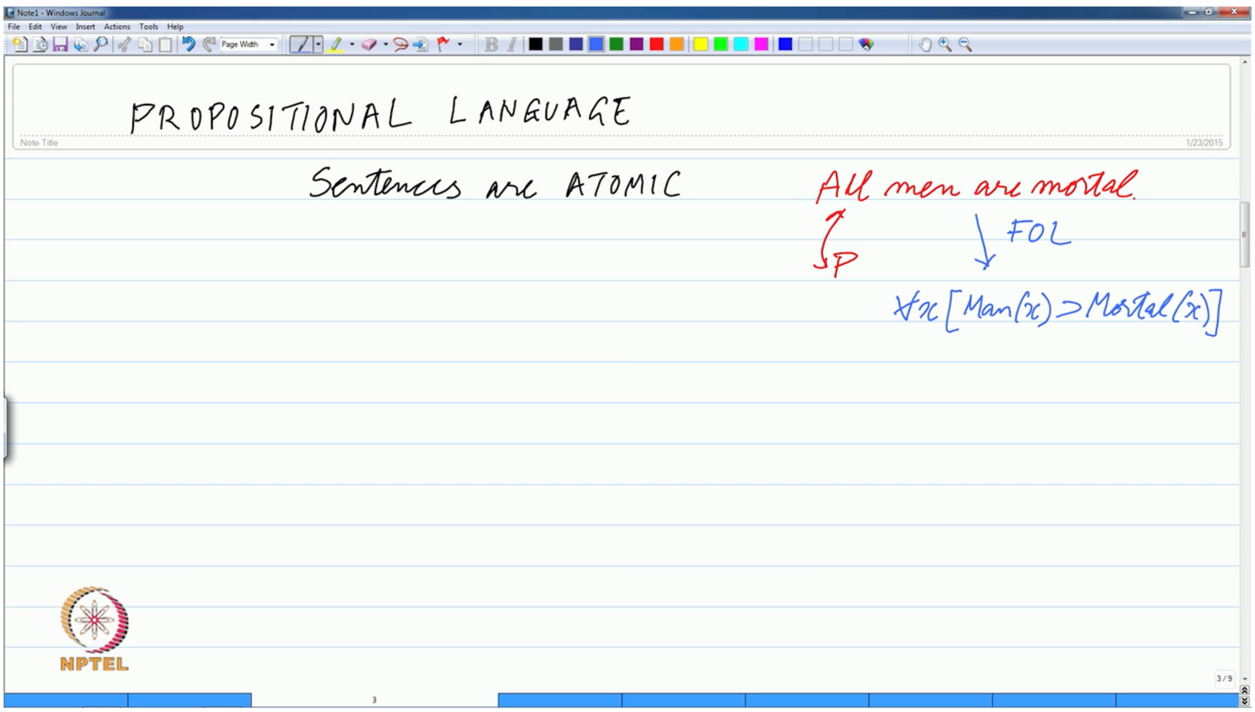

What is the difference between? By true we mean something is necessarily true as a consequence and we will define this. Very shortly when we define the propositional language we will see what does this mean in that context. This I am talking at a very general level but uses a semantic notion so if we know fact a fact b fact c fact d what else is true as a consequence of that. That's the notion of truth it's a semantic notion. Whereas proof is a syntactic notion which says that can you write an algorithm which will take a given fact given to you. You have a fact named fact a fact b fact c and fact d what else will the program produce. So program is just its just a program which will look at what it has and applies some rules as we will see and produces new sentences.

So this production of new sentences is a syntactic process. And we refer to that as proof. And we will see that what will be the condition that we will put on proof what constitutes a proof. What makes a proof a proof we will look at these things shortly. So it's a mechanical process the process of generating a proof is a mechanical process in the sense that you don't look at the meaning of the word. You only look at the form of relation between words. And as we go along I think this will become a little bit clearer.

So we begin with the simplest of all languages which is propositional logic. So first we define a language which is called as propositional language. And the distinguishing feature about propositional language is that sentences are atomic. We cannot break down a sentence any further we can only treat it as atomic entity. Its atomic unit which is defined in this language. So any sentence you want to reason about will be treated as an atomic entity. So for example if I want to talk about the sentence all men are mortals then in propositional logic I will say that this corresponds to a sentence I will call P. so we will be working as we will see shortly with propositional symbols or propositional variables which will stand for sentences. It is up to us what we want to plugin into P but as far as propositional language is concerned one sentence is denoted by one atomic variable.

On the other hand, when you look at FOL first order logic which we would do in due course of time we would represent this sentence by breaking up into smaller parts and you will write it like this. If you don't know this language it doesn't matter.

(Refer Slide Time: 21:10)



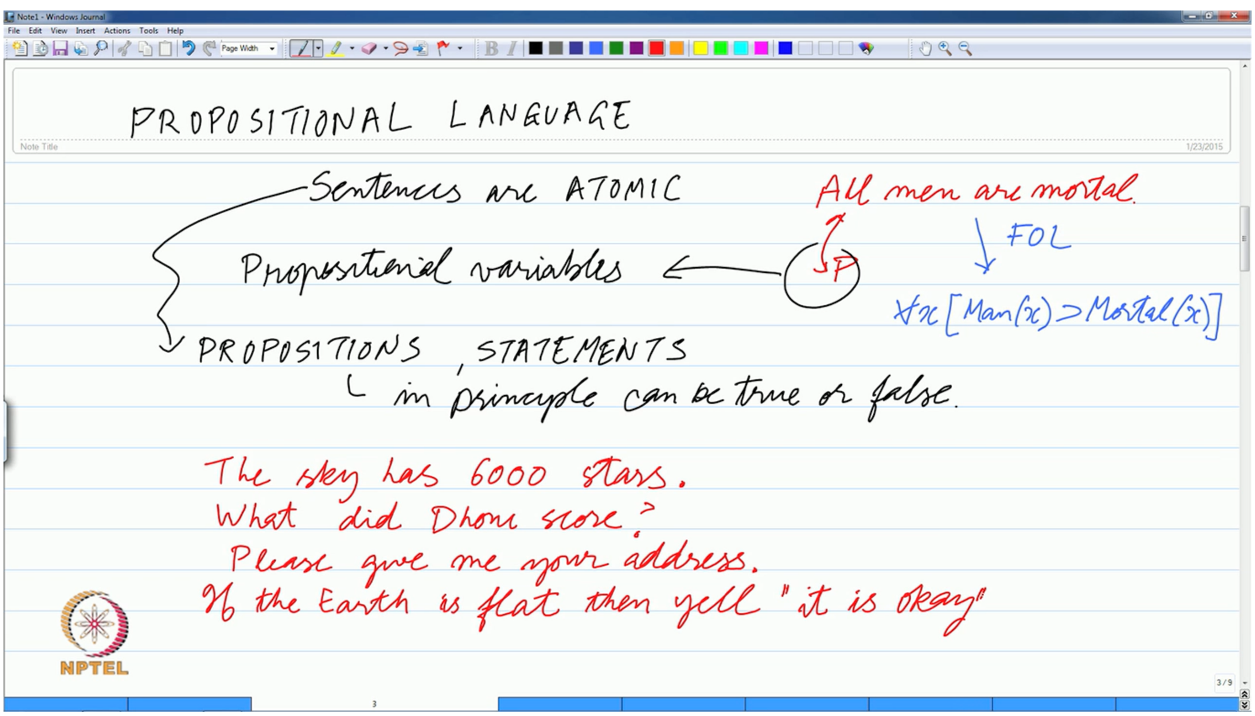

For the moment you can simply say that we are distinguishing between categories of things called man or humans and categories of things which are called mortal. And all we are saying is in this expression is that for all x so this symbol stands for for all x if x is a man then x is mortal. So we are stating the same sentence in first order logic but we have broken it down into parts and you can see once you break down things into parts we can reason with different parts. In fact that is what will allow us to eventually relate it to the fact that because Socrates is also a man then Socrates must be mortal. But in propositional logic we find that that would not be possible.

So this is a key feature about propositional logic which you can think of these as propositional variables. Which means that we will build our logic assuming that these are propositional variables but you are allowed to plugin anything in that. So instead of plugging in an English language sentence like this you could have plugged in a first order sentence like this into this symbol P because it's a propositional variable you can plug in any sentence in any language. So whatever we learn from propositional logic will carry forward to first order logic and other logic. So whatever machinery we talk about in this language we will carry forward to other languages also. So propositional logic is good in that sense it doesn't it's not cumbersome to deal with sentences in this. These sentences are atomic and we are really interested in how can we talk about truth and provability the question that you asked.

Now one more thing what do we mean by a center we also need things like propositions some books will say propositions so it's a logic of propositions or logic of statements. The basic unit is a proposition or a sentence or a statement we will use this term interchangeably. But what are all these things something which in principle can be true or false. Anything which in principle can be true or false is a statement or a sentence or a proposition. So a sentence like all men are mortal is also a proposition. If I say something like the sky has 6000 stars, then of I say something like let's look at an example. So what did Dhoni score?

I am not saying this is a sentence we will look at each of these in a moment. One last one. Some random sentences.

(Refer Slide Time: 25:32)

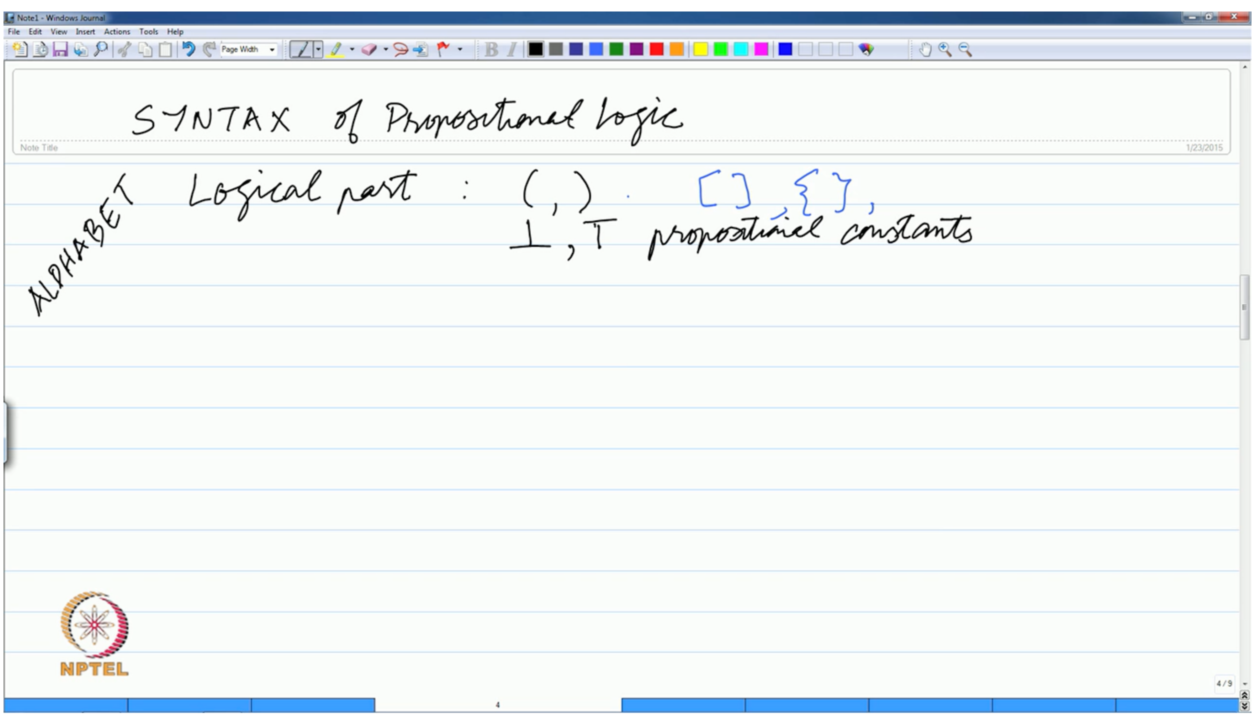



So sentence or some statements in principle can be true or false. We are not really concerned if they are true or false as long as they can be true or false we call it a sentence. So the first one the sky has 6000 stars is obviously a sentence because see it may be false but it's still a sentence it can in principle be true or false. But the other three are not sentences. What did Dhoni score is not a sentence it's an interrogative. Its asking something it's a question. Please give me your address it's not a sentence we cannot assign a truth value to this. It's a request or command or imperative statement. So that's why that's not the sentence and this is not the sentence. What about the last one if the earth is flat then yell it is okay? It is also a command of course it is based upon it has a part which you can see as a sentence if some statement is true but on the right hand side what we have is an imperative statement. So this is also not a sentence.

So anything which in principle can be true or false is a sentence. Okay so now let's go to the syntax of our language. Or in other words the propositional logic that we are talking about. So some people distinguish every language by two parts one is logical part and by logical part we mean something which belongs to every logic every propositional logic in this case. Okay we are talking about the alphabet at this moment. every language has an alphabet the alphabet has two parts one is the logical part which consists in the propositional logic things like brackets so two brackets are in principle enough but you could also add square brackets or curly brackets to improve readability for us. We often get very confused with many brackets at the same time.

Then we have two special symbols which are I will use this notation. These are propositional constants as opposed to variables.

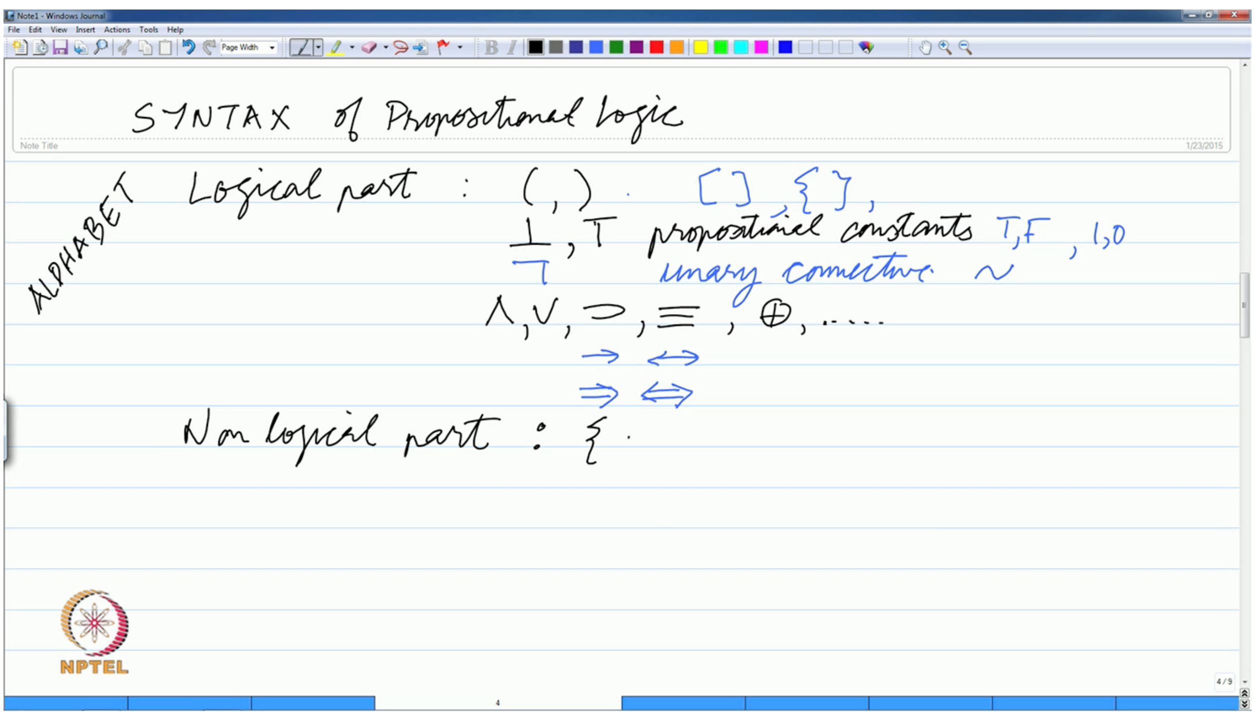
(Refer Slide Time: 29:32)



But you can use any two symbols so other people for example might use P F or 1 0 it doesn't matter but two symbols. One will stand for truth one stands for a sentence which is always true. One stands for the sentence which is always false. We will represent these two sentences by this. So they will be mapped to this two valued set that we have been talking about.

Then we will use a symbol like this this is a binary connective. Sometimes people use this symbol as well it doesn't matter we will use this symbol from the left hand side. And then we have symbols like this so you are all familiar with these kind of things and there are other symbols. So books may use this notation so instead of this somebody may use a single arrow or somebody else may use double arrow. Instead of this they will put two arrows like this it doesn't matter. These are just part of the alphabet they are used to construct sentence. And the other thing which some people call as non-logical part.

(Refer Slide Time: 31:57)




In set P Q R … accountable set of propositional variables. We also use things like P1 P2 P3 Palpha. So all this constitutes the alphabet of propositional logic. At this point maybe I should point out that a given language may choose a subset of connectives. So for example. So for example you might see somewhere a notation that if you use L to stand for a language then they will specify what are the connectives you are using. So for example this could be one language which is using 3 connectives we have seen one unary connective which all know we will read as NOT and these two symbols we read as OR and AND.

Or you may build a logic and we will see examples of this which uses only this two symbols or there may be other possibilities. So you can define a language by choosing which set of connectives you will use. And about 200 years or so people were trying out different experiments and trying to see which set is a good set to use and we will see shortly what do we mean by this sentence. Okay so I will stop here in the next lecture we will see the language we have seen the alphabet so far. In the next lecture we will look at what is the language for propositional logic then you will look at the semantics and then you will look at the proof methods.