Artificial Intelligence: An Introduction to Formal Logics

Prof. Deepak Khemani

Department of Computer Science and Engineering

Indian Institute of Technology, Madras

Module – 01

Lecture - 03

Let us now move towards the subject matter of the course that we are studying. We will use formal logics both for representation as well as for reasoning. So when you talk about representation then logic gives us a language, it is a formal language which allows us to represent certain kinds of things essentially and we will look at the formal languages in details as we go about. But not only does a logic allow you to represent something, it also allows you to reason about it in the sense that we can define rules for manipulating those representations and we will look at how we can manipulate things. So let's look at some of these ideas and we are looking at these from for... formal logics...so remember that when we say formal logic here the reasoning is based only on forms essentially.

So we saw syllogism a little while ago, it describes what is a valid argument. All men are mortals. Socrates is a man. Therefore, Socrates is mortal. The last sentence is acceptable to us because it confirms. the argument confirms the form which has been in some sense described as being a valid form of argument. And we are interested in valid forms of arguments. You can also have arguments which are not necessarily valid so which we will see that it is not necessary that what you are saying follows some form that is valid. So you can say that I see clouds in the sky. Therefore, it is going to rain. Now, your statement that its going to rain let's say it's going to rain at 5o clock or something, would be true if it actually rains at 5o clock. But it is a conclusion that you can draw something that you see heavy cloud, dark cloud and say that at 5o clock it will be raining. It is an inference you are making but it is not a valid inference in the sense it is not necessarily true that it will rain because you know often you see clouds and that it does not rain.

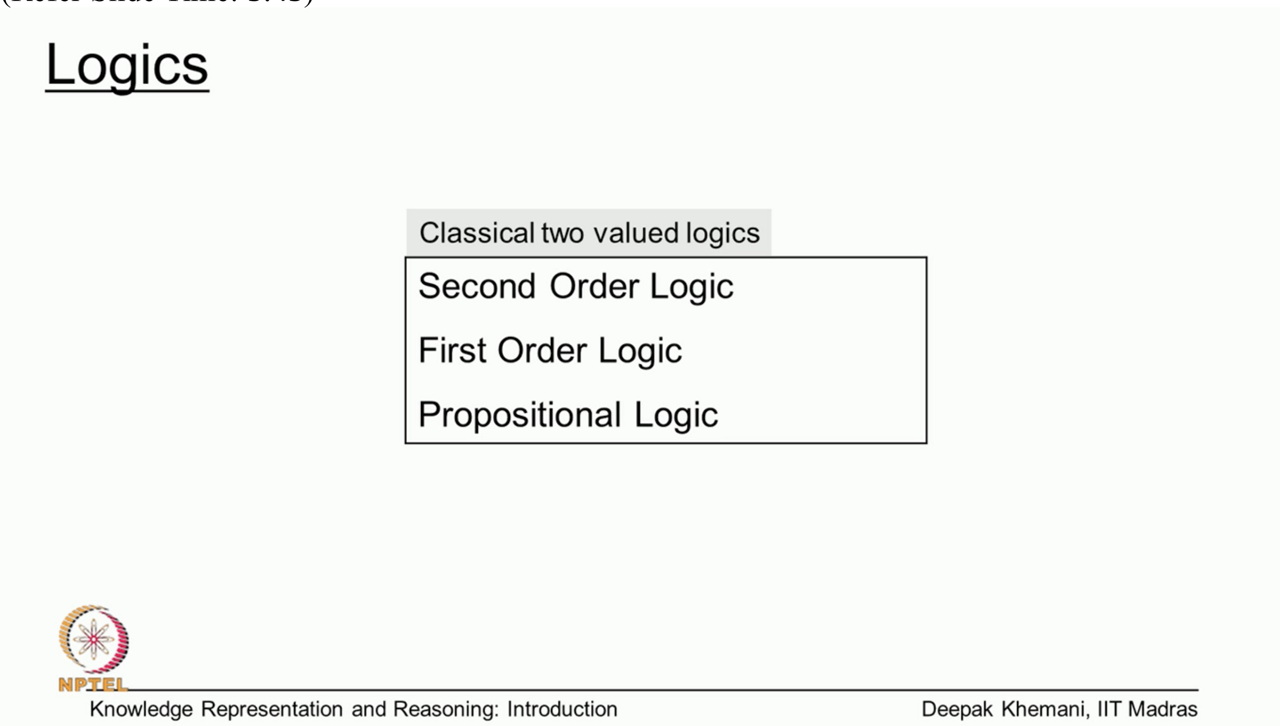
So we would be interested in largely in valid forms of reasoning or as we say deductive reasoning. But you will also be interested in non-valid forms of reasoning because that's very often we do that in the real world. And to with the complexity of the world we often in some sense draw some conclusions. So we should allow machine also to do that. To make inferences that are not necessarily true. And towards the later parts of the course we try to tie it together and say that how can you make inferences which are kind of you might say guesswork but how can you put them in a logical framework.

So, logics are formal language with well-defined rules for manipulations of representations. So first of all there are formal languages because they are used for representation and then on top of it we have what we call rules of inferences. We will do this formally once we get into a specific logic essentially. Out notion of a knowledge base or a KB is that is a set of sentences in a given logic. The logic language gives us a set of sentences. All formal languages you often say words...like these are the words of a language. You will be using a sentences because that's more common in logic. Knowledge base is set of sentences. It's a set in the sense that there is no ordering only its a just pool of statements.

There are many different logics that one can devise. We will look at ...we start to look at propositional logic or the logic of sentences. Then we will look at First order logic which is also known as predicate calculus. And then we will look at other logics. So I will shortly try to list some

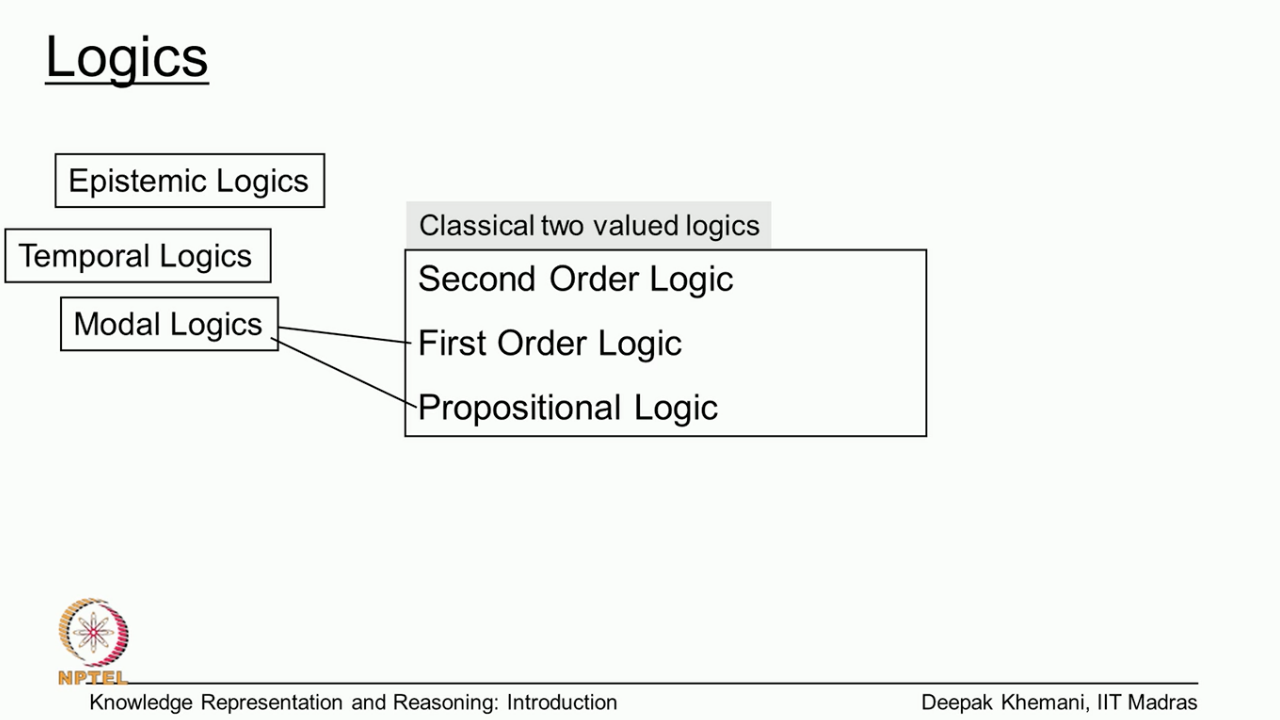
logic; so there are diff logics which vary on expressivity how much can you say in a logic essentially and how much can u say in a manner that you can reason with that. SO representation and reasoning go hand in hand.
Typically, more expressivity comes with greater cost of computational complexity. So the expressive a language is the more computationally complex it is to prove new things in the logic. So that’s why we always have to have a trade-off between what is the language you will choose which does allows us to express enough to solve any problem that we are trying to solve. At the worst case you have logic in which reasoning is un decidable it means you cannot be sure that the program will terminate or not if you try to do something. But in the simplest of languages reasoning is reasonably simple guaranteed to terminate.

(Refer Slide Time: 5:45)



So there is a small family of logics which ...from bottom to top the lowest is propositional logic then slightly more expressive if first order logic and slightly more expressive than that is second order logic. We will look at these in more detail later. But this a family of what is known as classical two valued logics. So what do you mean by two valued? That every sentence is either true or it is false. It can only take two truth values so this is two valued logics. People have thought of other kind of logics for example you can say true false and may be or true false and don’t know. People have tried to construct different kinds of logic. We will be interested to start in two valued logics and this is a classical logic and there is an increasing order of expressivity as you go from propositional to first order to second order logic. In fact, Gordon's celebrated incompleteness theorem applies to second order logic he says that. And we will talk about the notion of completeness as we go along. Essentially second order logic is so powerful in term of expressivity that you cannot even make all the inferences that you should be able to make. There is a notion of completeness we will come to that as you go along. So these are the classical logics and we will spend some time studying these and principally we will be studying first order logic because that the most common thing we used; all programming language that you see which have variables in them can be seen as instances of first order logic.

(Refer Slide Time: 7:44)



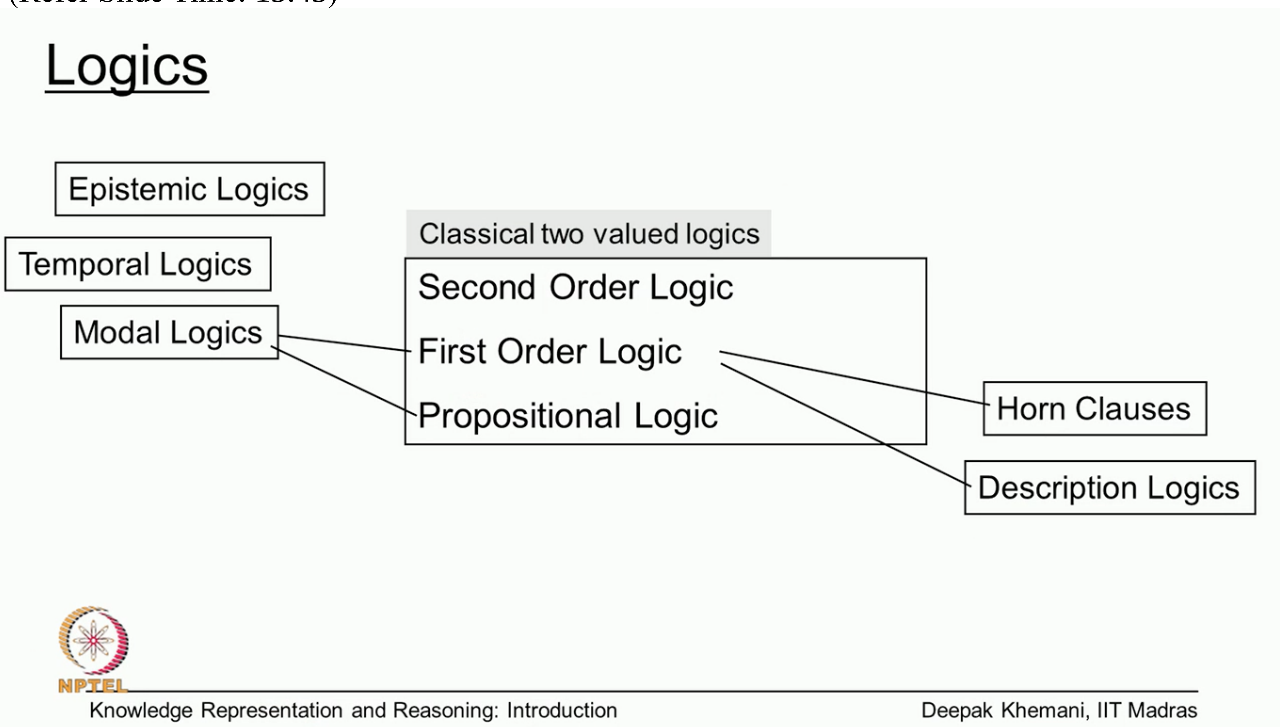
Then there are other kinds of logics which you can extend for or you can extend prop logic by something which are called modal logic essentially. So in classical logics you can say a sentence and you can say that it is either t or it is f but in a modal logic you can include modes so in the sense you can say that this sentence is necessarily true or that this sentence is possibly true. So these are two common modal operator as we call them. Necessity and possibility. And you can prefix these operators in the formal language that we talk about. May be we will get time we will look at a little bit of this. By saying that so a diamond symbol is used for possibility so if you have a sentence and you put a diamond before that then in modal logic it makes that this sentence if possibly true. So if I made a statement that I see dark clouds at 4 o clock in the afternoon and it is possible that it may rain at 5 o clock then obviously it is a valid inference because the possibility definitely exists. So modal logics allow you to do these kind of things. Or you can see if is necessarily true essentially that a sentence is necessarily true, so its necessarily true that that the sun rises that or the sun rises in the morn or something like that.

Now a class of modal logics are called temporal logics because they deal with time. The notion of time. So whenever in classical logic there is a notion of time. A sentence is true or not true it’s kind of you know timeless. Mathematics is kind of like that so mathematics forms in the domain of classical logics. So if you say that the Pythagoras theorem is true simply I mean you cannot say that its true today and not true tomorrow. Temporal logics allow to bring the notion of time and you can say that at some point in the future this will become true. So you can make statements like that. So as you can see its some modality. That you are saying the statement become true at some point in the future and there are people who work with causal dist. System n so on they try to work with things like these and they want to argue that at some point someone let’s say queue will be cleared or something like that. And another kind of modal logic is Epistemic logic. In classical logic we just make statement for example We say that the earth is round so either its true or false. In epistemic logic we talk about agents and what agents know and what agents believe. So I can say that john knows that the earth is round, it’s a statement made in classical logic does not make too much sense as we will see. But in epistemic logic you will be able to see that this agent knows this fact. Or you can say john believes that the earth is flat. The statement may be true. Its truth value does not depend upon the constituent beliefs the fact that the earth is flat is a false statement at least to all of us agree upon that. Doesn’t influence the truth value of the largest sentence that john believes that the earth is flat. So epistemic logic allows you to do these sort of things they taught you to they allow to talk about beliefs of agents. So we can represent some of those puzzle s often encounter that A knows that b knows that C knows that something then A knows that you know there are two people that say I know the sum of these two numbers other person says I know product of these numbers and then you know...they start talking and they say I know thins and i know this and eventually hey come to what people call common knowledge. Sometimes onwards the later part of the course we may look at the epistemic logic.

Now FOL can be intractable. It is only semi decidable. Which means as we will see later that if you give certain kinds of queries you can guarantee that the algorithm is terminates but if you give certain kind of queries then it will go into an infinite loop. In particular if you give a query whose answer is yes or whose answer is true then algorithm will terminate or if the answer is known then the algorithm will never terminate. So in that sense it is semi decidable, even if it is decidable sometimes poof can be exponentially long it seems. So we cannot afford to wait for such periods. So we have looked at diff subsets of possible logics which are motivated by trying to reduce the complexity of computation it seems. So one subset hat you will look at is horn clauses which is a basis of programming language Prolog as you will see. It has been shown that if you work with this particular subset of the language then you can have proofs which are linear in length.

Another subset that we will look at is description logic which allows us to talk about domains and things like that. DL are the foundation of what we call as ontologies. So if we have a language like owl which allows you to describe a domain then the reason that you doing owl is set of basic or there is at least one version of this language called as owl-dl which is based on DL. And DL allows reasoning which is tractable in nature. So obviously you cannot say everything but whatever you can say you can reason about that in a reasonable amount of time, so both these languages are motivated from reducing complexity.

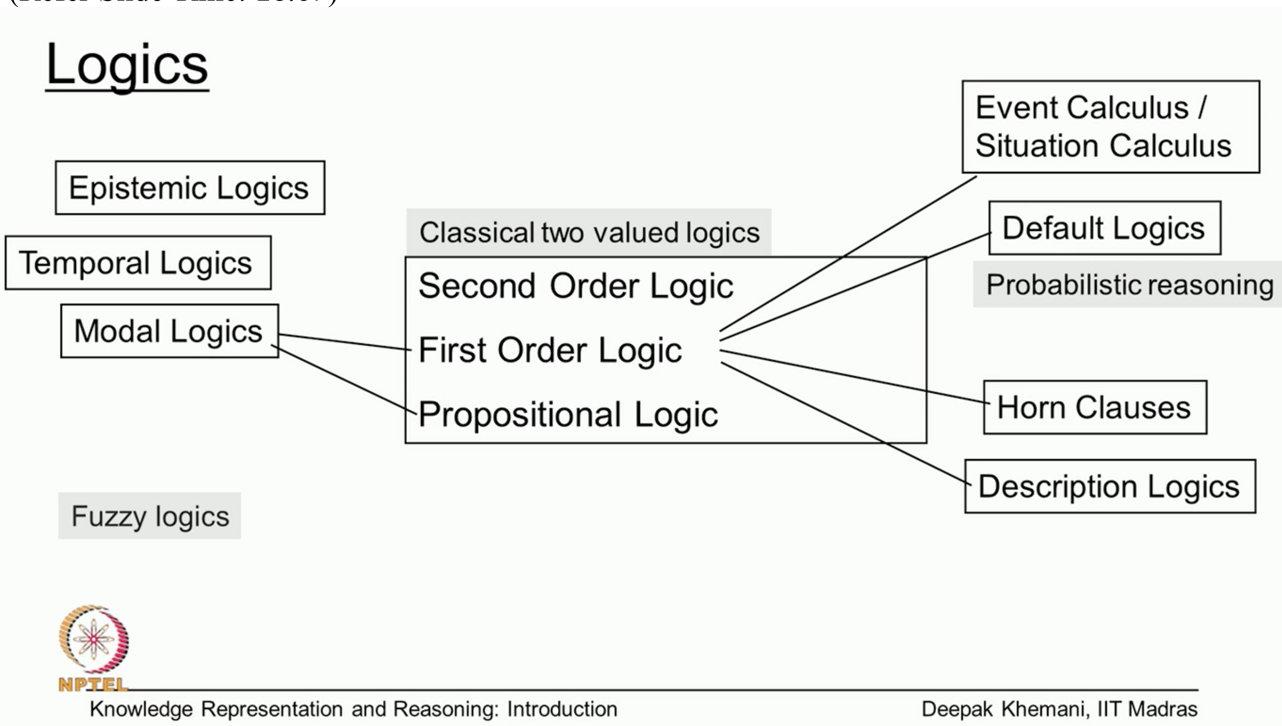
(Refer Slide Time: 13:43)



Then we will look at some variations of FOL which are motivated from increasing expressivities. Already you talked about modal logic they can add to what you can say in FOL you can add modalities you can add time; or about agents. The ... logics allow you to reason in the face of incomplete information. That you can make an inference but at the later point of time when you get more information you may take back that inferences, so such reasoning is called non-monotonic reasoning and such logics are called non monotonic logics. Because the set of state you believe to be true may vary over time. And default logics are or default reasoning is ta kind of reasoning we do so often that we don’t need to study. For example If you have come here on a bicycle you may say ok once this lecture gets over I will go and hop on to my bicycle and go to the hostel.

You are doing some default reasoning on the way. for example, you are making an assumption that your bicycle will still have air in the tyres then if you down its a default step that we do it without thinking. But if you want to be absolutely certain then you cannot say that. You cannot say with certainty that just because had air in the tyres when you came in you will have air in the tyres when you are going out because other things could have happened on the way. Some of these things are addressed in what we call as event calculus. so event calculus is kind of default reasoning.... which allow you to talk about events essentially. So when you say that you will hop on to bicycle and go home then you are sort of in some sense making default assumption that there is something else that is happening. that no enemy of yours for example came in to ...the air from the cycle or you didn’t run over a nail while you were coming in so that the bicycle got punctured. Or your bicycle is not stolen for example So all these assumptions mean certain events did not happen. And this kind of reasoning we will look at in some detail.

(Refer Slide Time: 16:07)



There are other kinds of logics of reasoning mechanism that we will not look at. Which is probability reasoning fuzzy logics constraint logic programming. Qualitative reasoning or graph sets So let me just have a couple of words about each of these, probability reasoning is basically bringing an element of probability there. Probability that it will rain at 5o clock is 75% so if you can come up with such a statement then you are doing some kind of probability reasoning.

Fuzzy logics are logics in which the set of true statement is not a crisp set its s fuzzy set. So the idea of fuzzy set is so typically sets are crisps set so either an element belong to the set or it doesn’t belong to the set. So For example If you want to say a statement about people being tall that’s my favourite example so if you want to say john is tall is that a true statement or a false statement it really depends on what we mean what do we define to be meaning of tall. So you say that okay tall means 6 feet or more essentially. Now that defines a crisp set. Anybody who is more than 6 feet ...6 feet or more falls in the set of tall people and anybody who is less that that is not tall. But then of course a question arises what happens if somebody is 5'11 half inches is that person tall or not.

Now fuzzy sets were devised to be able to think of these people also as tall so define predicates which in some sense gradually increase the membership value as some proposed. So what is hot when do you say that the water is hot, do you put a temperature that above 45 degrees the water is hot or do you say that gradually as it becomes warmer its property of being hot increases ..the truth value of the sentences that it is hot increases, so fuzzy logics which are devised by Zardeh are designed with these sort of things you might have heard of think about fuzzy logic controllers to control washing machine and so on. Graph sets in some sense are corresponding to modal logic in the sense that a graph sets don't have a one boundary. They have two boundaries so within one boundary everything is necessarily there. And within an outer boundary its possibly there. So such sets are called graph sets. So either we necessarily belong to a set or you possibly belong to a set or you don't belong to a set. So there are three reasons in some sense. The innermost region is where you necessarily belong to the set and there is a surrounding which is possibly belong to a set and a region outside where you don't belong to the set. Such sets are called graph sets.

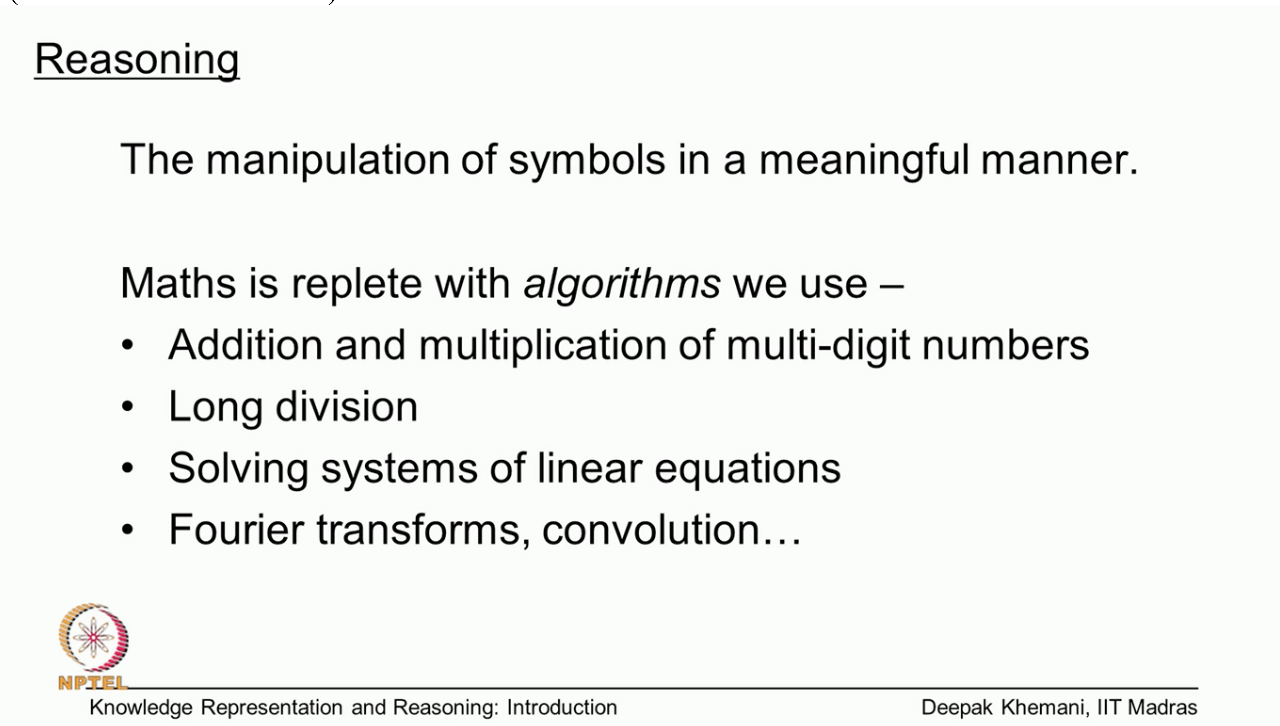
Qualitative reasoning is reasoning without numbers essentially. So it's basically in some sense reasoning of what physical systems you would have used numbers but you don't want to use numbers. And essentially you want to capture associations between certain kinds of things essentially. So you want to be able to make statements like as the level of water rises in a lake for example. Then the pressure at the bottom increases. So if you want this state association between level of water and the pressure being felt at the bottom. you could of course write equations to show that you know this is how much the pressure but you can also say that there is correspondence between if the level is increasing then the pressure is increasing. So qualitative reasoning has been used in modelling systems with these kinds of reasoning.

So I know some people who have used it in diagnosis for example say when i press the brake then the car steers or wears towards the left. So if you make such a statement then you can use qualitative models to reason with such systems and say what has really gone wrong. So you will reason about physical system is normally you would have quantitative models with equations and things like that but here you don't have quantitative models so you say when you throw a ball up then as it goes higher its velocity decreases ok so you are just talking about the effect of gravity here without using the equations of the gravity. So this form of reasoning is called qualitative reasoning.

Constraint Logic P basically combines logic programming. Which is what comes out of horn clauses and constraint processing. Where you basically model the world as variables and constraints between variables and then you find satisfying valuations for those variables. So SAT problem for example is an example of Constraint satisfactory problem. So CLP has emerged as a combination of these things.

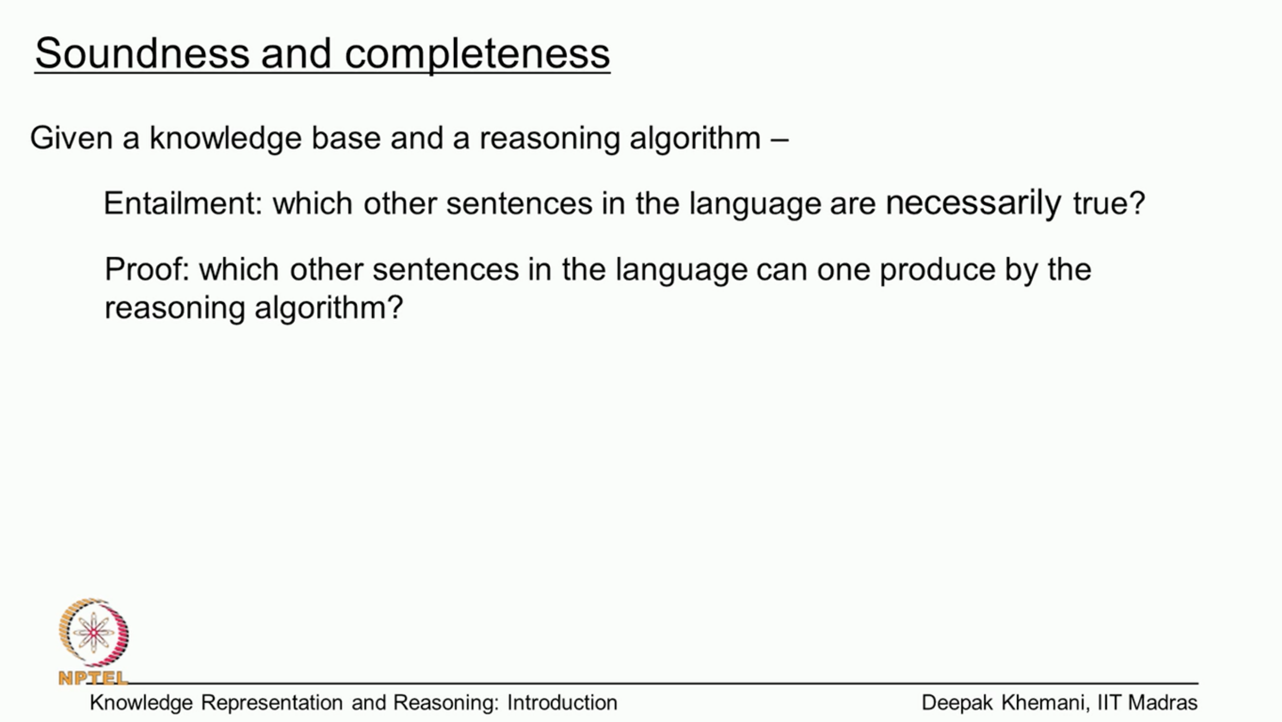
Reasoning as we have already said is the manipulation of symbols in a meaningful manner. Maths is replete with algorithms that we use so we are interested in these sort of things. So remember we are manipulating the representations...we are not manipulating ideas. When we manipulate numerals for example when you do addition or multi of multi-digit numbers you have an algorithm sitting behind that, so you look at five and you look at seven below that and you look up a table in some sense of course you do it mentally you say that if you see five and you if you see seven then you must write two below that. And then you carry 1 to the next column.so these are algorithm that we are using which are simply used in, I don't even need to be able to add myself I can just look up a table that when I see five and seven what should I write. So these are all syntactic in nature so whenever we are talking about reasoning we are talking about syntactic processes without looking at meaning but which should give something which is meaningful that's a kind of a conundrum that we have addressed.

(Refer Slide Time: 22:04)



Long division for example. When children learn long divide they just learn it as a blind procedure they don't understand what's happening. And we look at Fourier transform or convolution we often do it as a procedure. So we work with algorithms and we want to look at algorithms So whenever we talk about logic systems we talk...I will introduce the notion of soundness and completeness of logical systems now. So given a knowledge base. Remember the knowledge base is a set of sentences in a given language and a reasoning algorithm which allows you to manipulate symbols in that particular language. The notion of entailment is a very basic notion that we will start with. Given a knowledge base which other sentences in the language are necessarily true. Remember that we are talking of a given language so everything that we can express in that language and we will look at this when we look at propositional logic again whichever sentences which are expressive within the language are necessarily true or which are entailed but the knowledge base, which are entailed by the premises. So given a set of sentences or given some things that we know what else is necessarily true. And when you say what else, you are always talking in the context of a language. What else which is expressible in the language is necessarily true. So that's a notion of entailment. We will look at this in more detail as we go along.

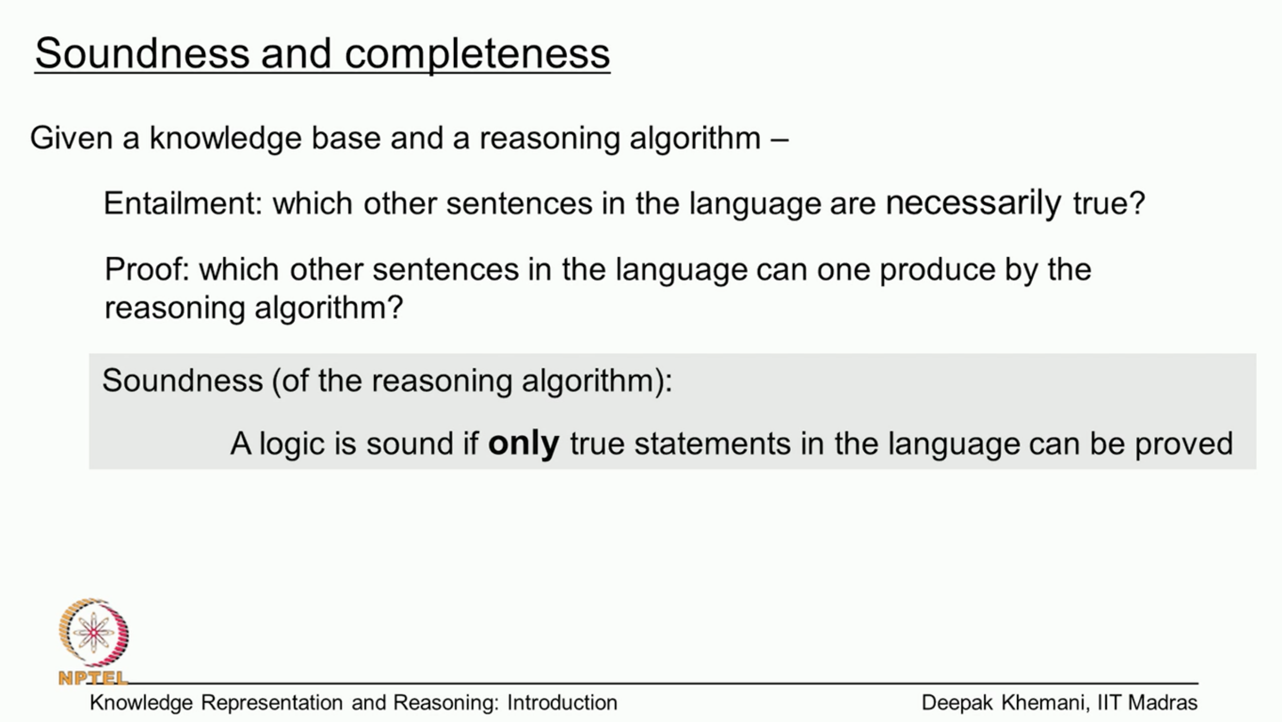
(Refer Slide Time: 24:10)



We have the notion of a proof. So essentially we are talking of a reasoning algorithm which manipulates symbols in a syntactic fashion. Which other sentences in the language can one produce by reasoning algorithm. So there are these two notions: you are given the knowledge base and one question is kind of semantics ...it is concerned with semantics or meanings. It says that given that these things are true, what else is necessarily true. So given that All men are mortals and given that Socrates is a man one of the sentences which is necessarily true is that Socrates is mortal but what other sentences are true. So that is concerned with entailment. Meanings truth value.

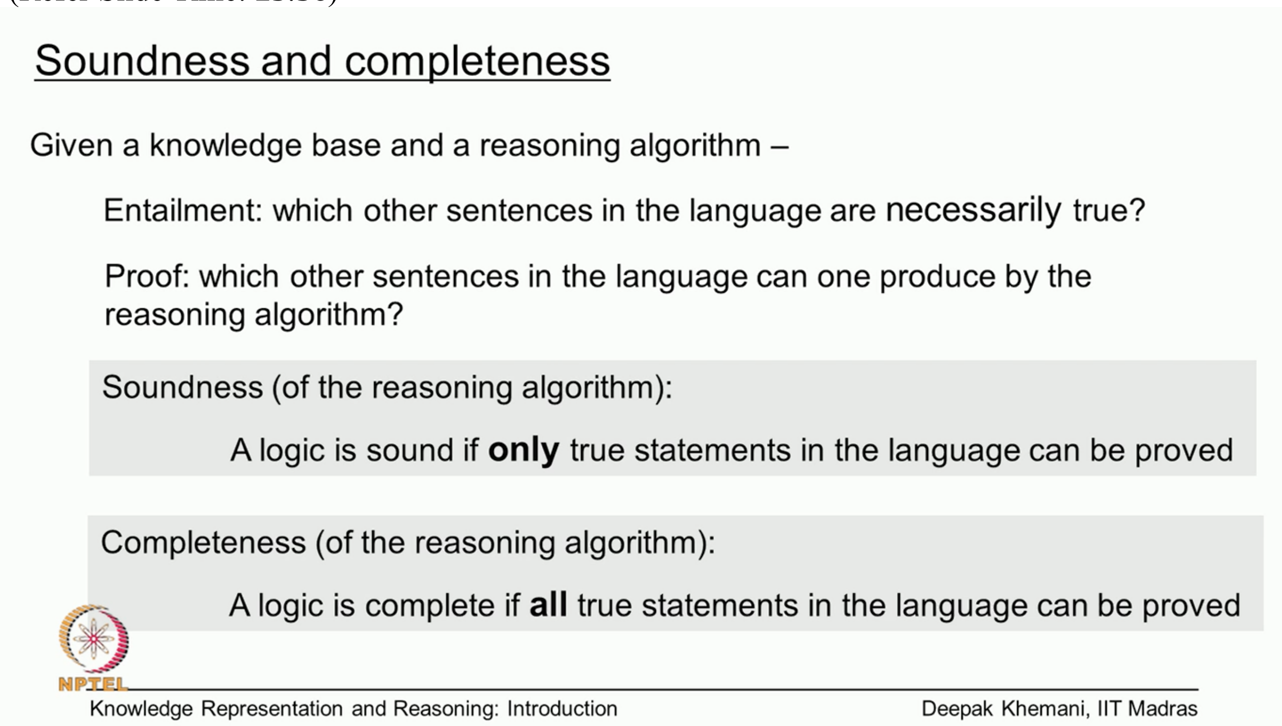
Proof is concerned with what does your algorithm produce. So can my algorithm look at these two sentences and produce the third sentence via syntactic processes. That's the notion of a proof. Again we will look at it in more detail as we go along but that's the basics whenever we are talking about logics. The notion of entailment which is the semantic notion which is what else is true. The notion of proofs which is a syntactic notion which says what can be proved or what can be derived by the algorithm that you are working with.

(Refer Slide Time: 25:29)



And there are relations between these two things, we talk about soundness of a reasoning algorithm in a given language. A logic is sound if and only if two statements in a language can be true. Only true statements can be true. So whatever the system produces should be true. Then we say it is sound or valid. We say a logic is complete or a reasoning algorithm is complete if all true statements can be true. So there is no true statement which cannot be true. Now it's true that as we will see that probabilistic logic is sound and complete. First order logic is also sound and complete. But as goodwill....told second order logic is not sound and complete. You cannot prove all statements in second order logic.

(Refer Slide Time: 25:56)

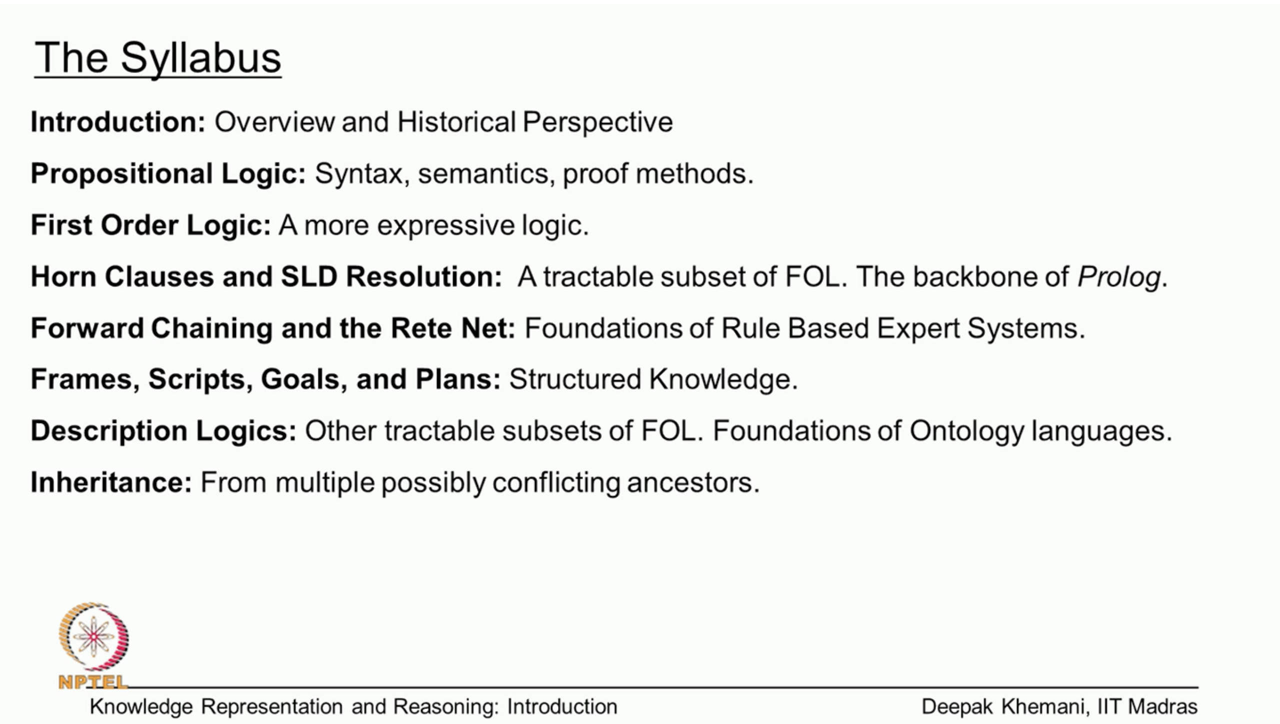


Okay so let me end with the syllabus for our course. What we just did actually was overview and historical perspective. Then in the next class we will move on to propositional logic. We will look at the formal logic, we will look at the syntax, we will look at semantics and we will look at proof methods. We will talk about soundness there. Then we will move on to First order logic, which is a more expressive logic. The difference between the propositional logic and first order logic is that the first order logic allows you to talk about variables. Whenever you say All men implicitly you are using variables. In propositional logic we cannot introduce the syllogism which in first order logic as we will see, we can. because we are allowed the mechanism of using a variable. So basically the difference between propositional and first order logic is that you are allowed to use variables, you are allowed to use quantifiers over variables. You can say All men or some men or these kinds of statements. Second order logic allows you to talk about quantifiers over predicates essentially which we will see later but we are not going to study second order logic.

Then we will look at horn clauses which is one of the subsets which is more tractable. We will look at a particular proof procedure or a particular reasoning algorithm called SLV resolution and we will see how this forms the backbone of the language Prolog which you must be familiar with or some of you must be familiar with. Then we will look at forward chaining and the structure of Rete network which was the basis for what people called as expert systems in the 1980s which uses something called a forward chaining, inference and ....or expert systems shell as it is usually called. The key idea behind that was the weightage network, we will look at that.

Then we will look at structured knowledge. We will look at frames which was introduced by Minskey....Then scripts, goals and plans which was introduced by Rogere shank..... and we will see how they are used for reasoning. Then we will move on to description logic. As I said this is another subset which is tractable of the whole and we will see that this forms the basis of many ontology languages like OWL-DL for example.

(Refer Slide Time: 28:56)



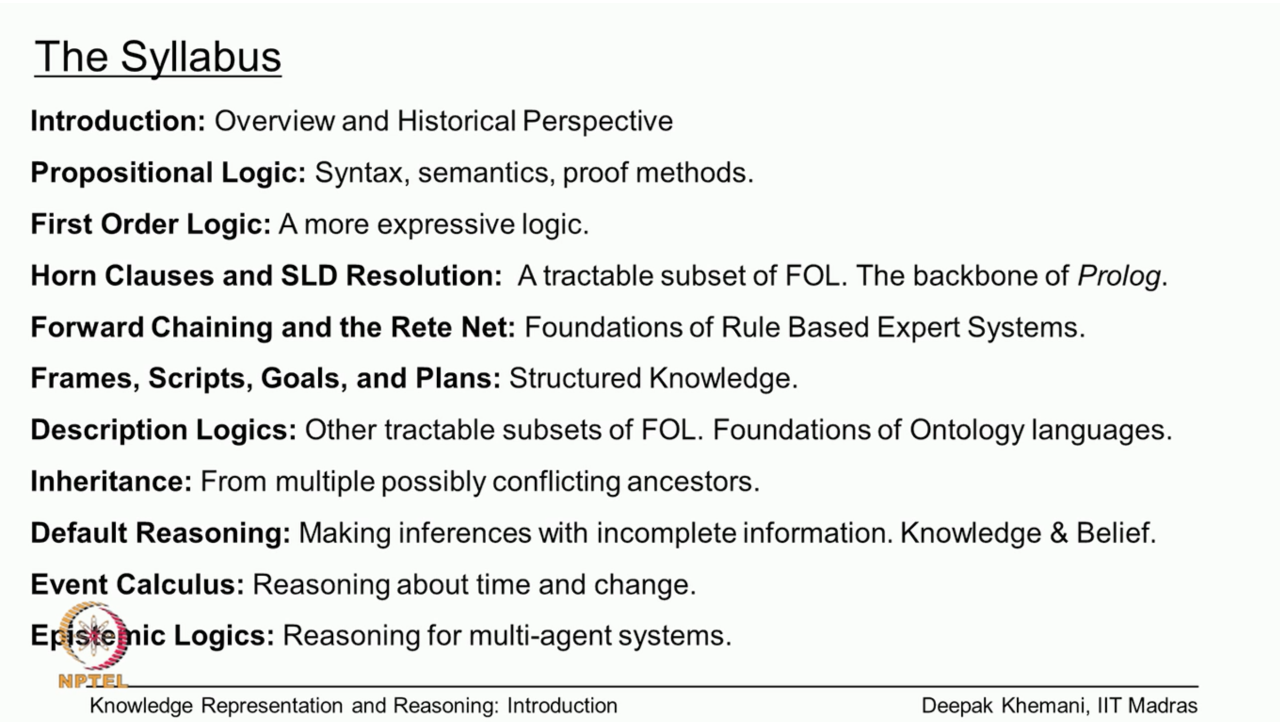
Then we will look at inheritance. Whenever you will talk about the taxonomies of the object oriented programming. Object oriented programming can be traced back to the idea of frames which was introduced by Minskey. And one of the feature of object oriented programming is inheritance so you can inherit something from your ancestors. What if you have multiple ancestors, do you inherit everything from them. What if there is conflict between them, one of them says that a certain property is true, another one says another property is not true. Which is the most acceptable feature to inherit. So some of those issues you can look at when we study inheritance.

Then we will move on to default reasoning making inferences with incomplete information. Then we will talk about knowledge and beliefs. Then Event calculus as I said, reasoning about time and change. Something happens, something else happens and so on. When I pick up this bottle then I should be able to conclude that I am holding this bottle. So there was an action which happened or an event which happened as a result of which something became true, these kind of logics are using event calculus. Then we will reason about time and change.

And finally based on the time we have, we will look at a form of epistemic logic which we reason using multi agent systems.

So these are the two textbooks that I has said we are going to use. One is this book by Brachman and Lewiss.....the first reference. You can find an Indian edition available it will not be too expensive if you want to buy it. And the other one is my book, which is a textbook of AI and you can find only Indian edition or south Asian edition. So apart from these we will use some of these references and so there are couple of references by Schank which talks about fits and goals and plans that you can see. Then Shanahan talks about the calculus of events as you can see the title. Then John Sowa who we show the cartoon from. It talks about one form of representation called conceptual graphs.

(Refer Slide Time: 31:25)



Okay so when we meet in the next class we will start with Propositional logic which is what we have seen here. The next thing is propositional logic, we will start with syntax, semantics and proof methods.