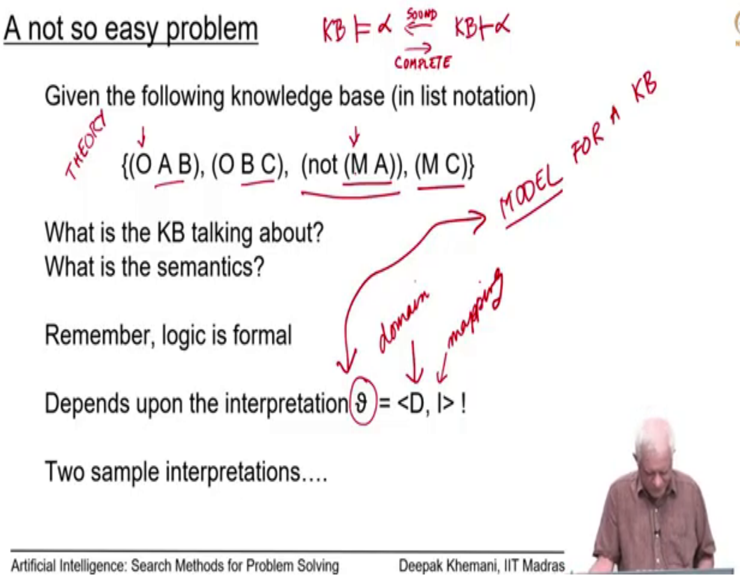
Artificial Intelligence: Search Methods for Problem Solving
Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Chapter – 11
A First Course in Artificial Intelligence
Lecture – 87
Deduction as Search Incompleteness

(Refer Slide Time: 00:14)



Welcome back for this last session on logic and reasoning and deduction. We have had a very whirlwind tour of logic, but hopefully we will have time to go over things in more detail.

So, just know a very quick recap there were two notions we started with. One was the notion of entailment. If you remember we said that a knowledge base entails a sentence alpha. And the meaning of this sentence is that if the knowledge base is true, then alpha must be necessarily true, so that was the notion of entailment.

We also looked at the notion of proof which we had a different notation here. And we said that a alpha is provable given the knowledge base that if the knowledge base is given to you, some reasoning algorithm will generate alpha.

So, there is an algorithm tied to proof, whereas entailment is a semantic property. It simply says that if the knowledge base is true, then alpha is true essentially, but semantics is not easily accessible to us.

In propositional logic, you can always construct truth tables and things like that, but in when you talk about first order logics it is not easy to work with semantics. So, there are proof methods which are closer to semantic methods which are called like model checking and so on, but we will not get into that.

We are focusing on proof methods which are syntactic in nature which means you have a proof procedure, you have a procedure, you have an algorithm, which can produce the sentence alpha.

And then we had talked about notions of soundness and completeness. We had said that a logic is sound, when you see a logic we also mean a proof procedure now is sound if anything it produces happens to be true. So, soundness goes from right to left. If alpha is provable, then alpha, alpha must be true which means our logic machine can prove only true statements.

The other direction was completeness. It says that if alpha is true or alpha has entailed which is a semantic property, then our proof process will procedure will find it essentially. So, our logic is complete. Anything that is true can be proven in our system essentially.

So, we will started looking at logic, then we looked at forward chaining, and then we looked at backward chaining. And we saw examples of both. And we saw that prologue is a language which does backward chaining. Today we want to end on a note which cautions us about these two methods. So, let us see what I am talking about here.

So, let us look at the not so easy problem. When you are looking at this problem, you must also remember that logic is formal in nature that we work with symbols, we manipulate symbols. The meaning behind those symbols is only in our heads at least as of now. I did say that for first order logic, you can create the semantics by creating an interpretation and so on, but let us not get into that here.

So, let us look at look at the problem which is not so straightforward to tackle. So, in the list notation that was introduced by Charlie and McDermott. We have a very small knowledge base. It has got four sentences. Remember that in the list notation, the first element in the list is a predicate. So, O is a predicate and it has two arguments A and B here essentially. In the second sentence again O is a predicate, and it has two arguments B and C.

Then we have a not of course that we are familiar with and it says that M A whatever the formula is false essentially, or not of that formula is true. that formula itself has a predicate called M which takes one argument A is essentially. So, that is a third sentence in our in our knowledge base.

The fourth sentence simply says that there is something C which is an M. So, you can read this as some relation between some predicate which relates A and B, and B and C. O is a relation, it is a binary predicate; it takes two arguments. M is a unary predicate, it takes one argument. And the last sentence says that C and M, and the third sentence says that A is not an M essentially.

So, given these four statements what are we talking about? When we use variable names like count is equal to count plus 1 in any programming language like C or something like that, the fact that we are use a variable named count is meaningful only to us. As far as the interpreter or the compiler is concerned, it has no meaning at all, it simply is just the variable.

So, here also I have I have taken deliberately I have taken predicate names and argument names which are not meaningful, because that is how logic works. Logic works only with the form; it does not work with meaning. So, if we have a knowledge base like this which has got only four statements or four sentences, what is it talking about, what is the semantics of these four statements? [vocalized-noise

We cannot say. It is, it is just a set of four statements. Whether they are true or not will depend upon what is it they are talking about, what is the domain they are describing, and if those particular facts that we are talking about are true in the domain or not?

So, that truth value comes later, the meaning comes later. To the knowledge base it is just set of sentences in the language which we have defined as the first order language. And this one does not even have any quantifiers here.

So, let us look at this process. Now, try to get behind the fact try to digest the fact that logic is formal that the meaning is not embedded in the knowledge base at all essentially. So, as I said there is a notion of an interpretation, we use a symbol I for interpretation in which we choose a domain.

We will not go into the details, and we choose a mapping. So, if we choose for example numbers as domain, then we can say the successor function means this, the sum function means this, the greater than predicate means this, so all that is given to us by mapping essentially.

Constants will map to elements in the domain; variables are variables which can map to anything in the domain; statements which have universally quantified variables will apply to all elements in the domain existentially quantified statements will apply to some elements in the domain.

And therefore, we can talk about what are we saying what are we talking about what is the meaning of our sentences, and also whether a knowledge base is true or not essentially.

So, if we pick a domain and an interpretation which makes a knowledge base true, then we are set to have a model for the knowledge base ok. So, I will just mentioned that the term here, but we will not go into the details. A model for a knowledge base is an interpretation which means you have to choose a domain, and you have to choose a meaning of the predicates and the constants.

So, for example, Socrates is a constant and if you are choosing the domain of people, then we are referring to one particular element in the domain. If we talk about Donald Trump, we are talking about one particular element in the domain of people. So, you have to choose a domain, and you have to choose an interpretation.

And you are given a knowledge base. In our case, there is a knowledge base of small set of four sentences which we say that an interpretation I which is made up of a domain and a mapping is a model for the knowledge base if all the statements in the knowledge base become true.

So, a model is that interpretation which is true for the theory as some people say. So, when we talk about knowledge base, some people use a word theory. And this for people who like physics will ring a bell because you always have theories in physics and so on.

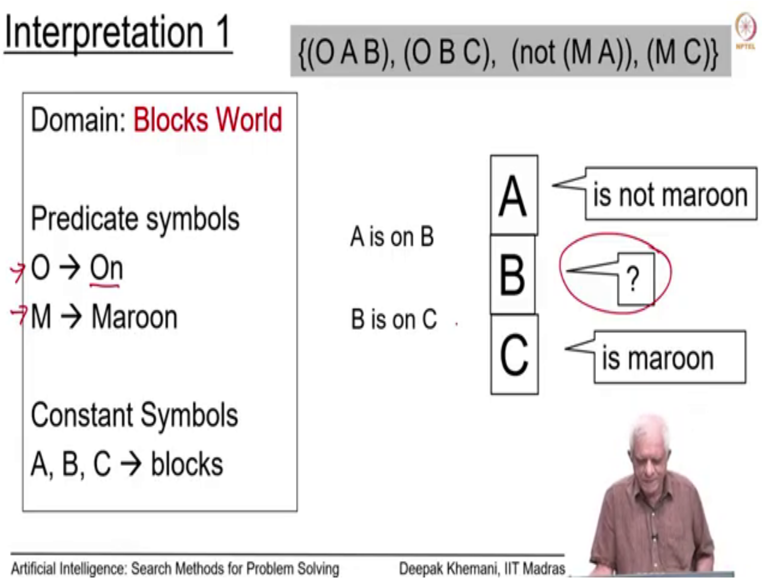
And then you have to go and validate does this theory hold in the real world, is there quantum gravity, is there is the string theory able to explain the whole world, how do we explain gravity you know all kinds of things people asks.

So, when we talk about formal systems mathematics or physics as the case may be or logic we are talking about a theory. And then we when you are talking about model is the can we find the real validation of our theory, can we see somewhere in the real world that whatever we are saying in the theory is in fact true? So, that is the notion of a model very quickly we will not go into that.

So, for our very small knowledge base, remember there are two predicates O which is a binary predicate and M which is a unary predicate. Let us look at two sample interpretation. So, you before you do that, maybe you should take a pause, and try to construct a interpretation for this sentence.

There are four sentences. And just construct a world give a meaning to what is O and what is M and what are A, B, C, D, or A, B, C such that the statements become true essentially. So, we will start off by looking at two models for this knowledge base. So, so take a moment and think about it, and then move on to the video.

(Refer Slide Time: 11:48)



Here is one model. The knowledge base is given on the top or the theory is given on the top. And the domain that we have chosen is a blocks world domain that is something that we have been talking about quite a bit in our course here. The predicate O which is a binary predicate, we will treat it as on in the blocks world ok. We are familiar with that.

And the predicate M we will treat it as the colour predicate that M of x is true if x is maroon in colour. So, maroon M stands for maroon. So, we are talking about blocks which may be on top of each other, and we are talking about colours of those blocks. So, now, of course, having chosen a domain which is this we have an interpretation and we know what we are talking about A, B, C are blocks in our case.

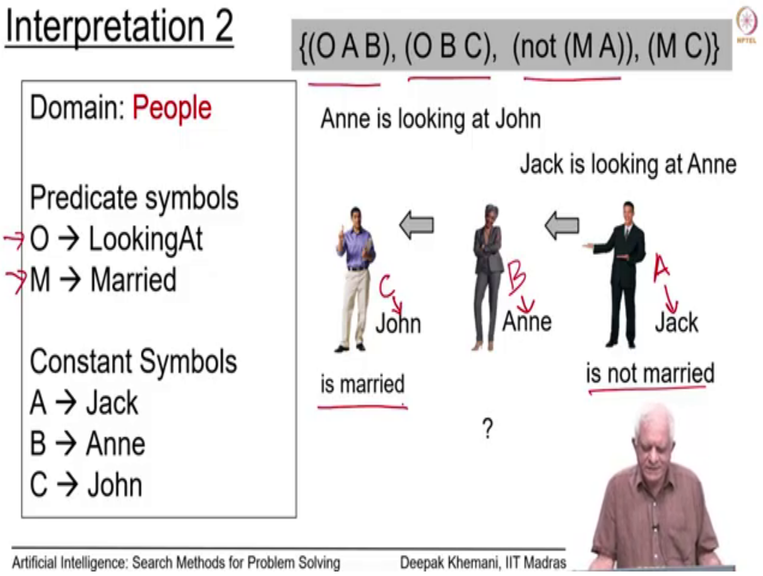
So, what are we talking about? We are saying that A is on B that is the first statement in our knowledge base. The second statement it says that B is on C. So, we have this blocks world. The third statement is saying that A is not maroon, and the fourth statement is saying that C is maroon ok. So, now, that we have created a domain and an interpretation is clear to us what are we talking about, what is the logic what is the knowledge base talking about.

But that comes only after you have pinned down a domain and an interpretation or domain and mapping which is also called an interpretation. So, this is of course a model for the knowledge base that we have essentially.

Can you think of another domain? So, the same set of sentences therefore, the same four sentences O, AB, OBC not MA and MC, it could be talking about something entirely different.

So, let us look at an example, another example. Observe here that we are not said anything about what is the colour of B essentially. We have simply said A is not maroon, we have said C is maroon. And we have said that A is on B, and B is on C that is all we have said about the world essentially.

(Refer Slide Time: 14:06)



Here is another domain. In this domain which is a domain of people. Now, O stands for looking at, it is a binary predicate remember which stands for a binary relation. So, if you say O A B, it means A is looking at B.

And M matches stands for married essentially. M of x means x is married essentially. And we have three constants and let us just give them human like names here. So, A is Jack, and B is Anne and C is john essentially. um

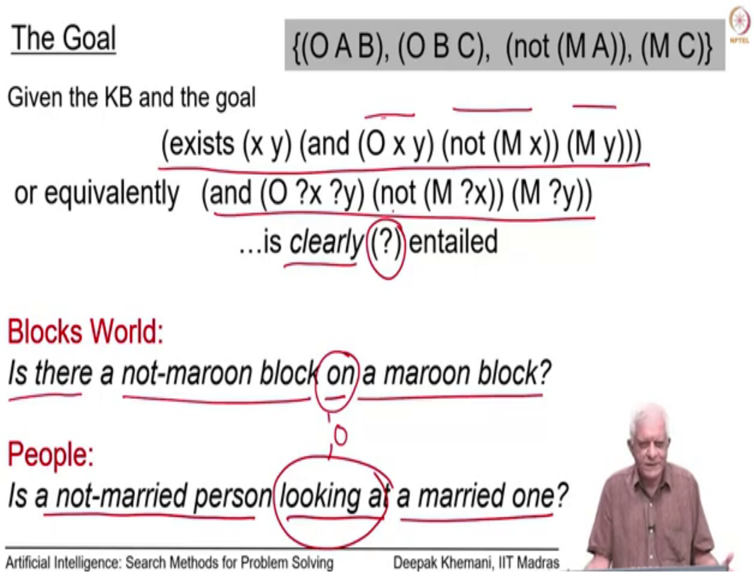
Now, what are we talking about we are talking about a world in which the first Anne is looking at John, Anne is B, and John is C. So, actually this is the second statement in our logic. What is shown in the picture that Anne is looking at John is because O stands for looking at and B is Anne.

So, let me write it here. B is Anne and John is C. So, obviously, Jack is going to be A. And the first statement says that Jack is looking at B, and B is Anne. So, Jack is looking at Anne. So, A, A maps to Jack that is a mapping, B maps to Anne, C maps to John actually I should point to the people, but anyway it is the same thing..

And, and the other two statements are saying that A is not married. So, we have written it here, A is not married, and C is married. So, this is the same knowledge base with two different interpretations. I hope that drives from the point that logic is formal that whatever you do with logic the meaning is only in our heads essentially.

So, one person could say that this is talking about the blocks world; another person could say that this is talking about people. And in both cases all the four sentences are true in the world that we are describing. But that is not the reason why we are discussing this example. The reason we are discussing this example is that reasoning is not always so simple.

(Refer Slide Time: 16:40)



The questions that we are asking in the blocks world is if you look at the bottom of the slide, that is there a non-maroon block on a maroon block that is a question we are asking? Which if you are talking about people would be is a non married person looking at a married one essentially.

So, here of course, the relation was on, and here it is looking at, but this both are for the relation on essentially. So, that is our query. It is an existential query remember that and we can write it as an existential query which is here, it says does there exist an x and does there exists a y such that O x y is true, O x y is true, not of M x is true and not of M y is true.

Let us look at the block world block. In block world, we are saying are there two blocks x and y such that x is on y, x is not maroon and y is maroon. If we look at the people domain, the same query translates to other two persons x and y such that x is looking at y, and x is not married, and y is married. So, is a not married person looking at a married one essentially that is a query? So, the first thing I would like to ask you is about entailment. Is this sentence true?

The second line here simply has it in a implicit quantifier form, but by now we are familiar with the fact that that variables in queries are existential in nature essentially. And the top sentences explicitly is essential. So, I have I should have hidden that sentence before asking you. The question we want to ask is that is the sentence true or not?

I have written here it is clearly true I have also put a question mark, because it is not so straightforward to answer whether this query is true. And many people have conducted this test over a whole set of people, and very few people end up saying that yes this sentence is true. Are they anybody says it is false? But most people say that we cannot say, we cannot see, we cannot this thing.

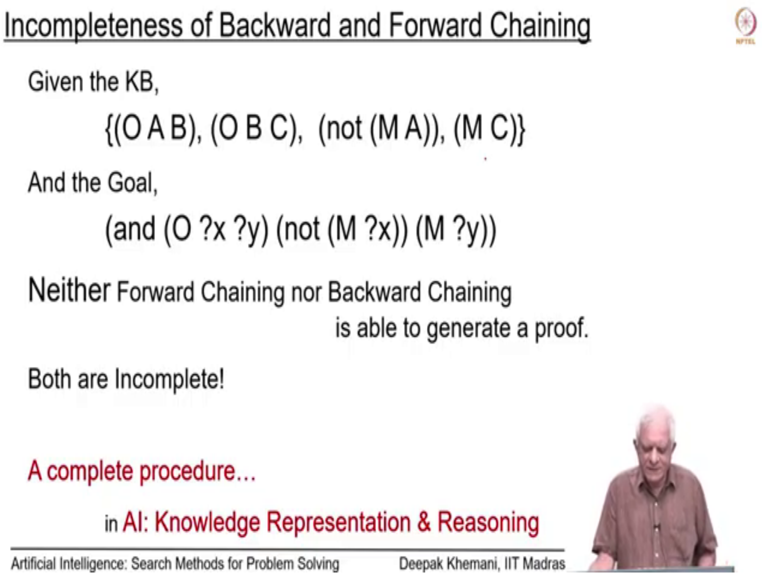
But the semantics is well-defined and entailment is well-defined. So, we should be able to say whether the sentence is true or not. And if you if you listen if you if you give some thought to this, you will find that indeed this sentence is true, and the logic is not so hard. It revolves if you can reason by cases, and we do not have a proof method which can reason my cases as of now.

It really depends upon what whether Anne is married or not. There are two possibilities Anne is married, then you know that Jack is looking at Anne, and Jack is not married and Anne is married, so therefore, a not married person is looking at a married person.

On the other hand, it could be that Anne is not married, these are only two possibilities either Anne is married or Anne is not married essentially. So, if Anne is not married, it turns out that because she is looking at John the statement is again true. So, the statement that an unmarried person is looking at a married person is true in this knowledge base.

Many people are not able to arrive at this conclusion specifically because while the statement is true we do not know which married which unmarried person is looking at which married person, because we do not know whether Anne is married or not. But it turns out that irrespective of whether she is married or not this statement is true. So, it is a true statement.

(Refer Slide Time: 21:28)



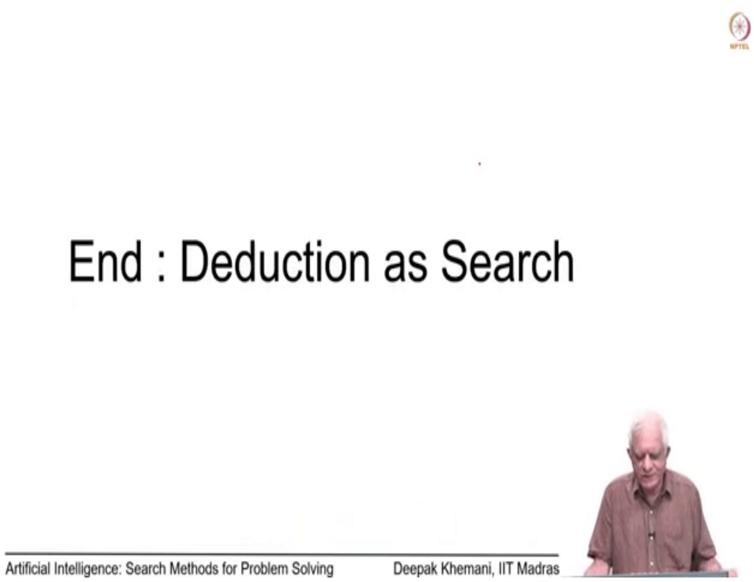
Which means if our logic is complete, we should be able to generate a proof for that essentially. It turns out that neither forward chaining nor backward chaining is able to generate a proof. You can try it for yourself. In fact, there is hardly any implication sentences there to chain over. And this is a very simple example which talks about the incompleteness of forward chaining and backward chaining.

Does that mean that first order logic is incomplete? No, we have already mentioned then Godel had shown almost a century ago, he had a he had he had a theorem called the completeness theorem. We says that first order logic you can always construct, construct complete reasoning systems. He had a theorem called the incompleteness theorem very famously known as Godel's incompleteness theorem, but that was talking about second order logic essentially.

First order logic is known to be both sound and complete which is why it is a very powerful tool that we use almost everywhere in computer science. But it is not complete if you are talking about backward chaining or forward chaining. And this example is a counter example it is see here is a problem that you cannot generate a proof for.

So, these methods are not complete. But there exists complete methods for theorem proving in first order logic, but we will leave that for another time and hopefully some of you will come and join this course which is called AI knowledge representation and reasoning, and that will be offered in the next semester.

(Refer Slide Time: 23:18)



So, with that we will end our study of deduction as search. Our goal here was so that while logical reasoning is a very important part of an agent's capabilities underlying this logical reasoning is a search algorithm essentially.

We saw two search algorithms forward chaining and backward chaining, but they turned out to be incomplete. There is another search algorithm called the resolution reputation method. Many of you must have studied this. It is complete for first order logic I think.

So, we have one more week left. In the next week, we will come back and look at an alternative way of representing problems which is using constraints and these representations are called constraint satisfaction problems. And we want to have a quick look at that.

The nice thing about constraint satisfaction problems is that you can flexibly combine search and reasoning together to solve a problem. The same problem you can solve entirely with search or you can solve entirely with reasoning. But of course, now we know that behind reasoning there is a underlying search algorithm, but nevertheless we like to think of that as reasoning essentially.

So, we will do that in the next week which will be the last week of the course.