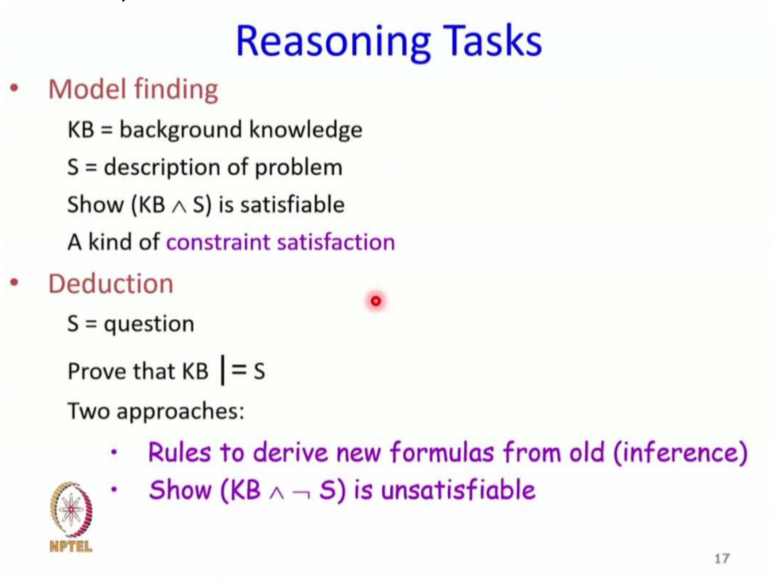
Artificial Intelligence

Prof. Mausam
Department of Computer Science and Engineering,
Indian Institute of Technology Delhi

Lecture-45
Logic in Artificial Intelligence: Forward Chaining, Part-4

So now, I think we are getting closer to the interesting algorithms and the reasoning tasks that we have to solve with this. So let us say I give you so there are 2 tasks that we will study.

(Refer Slide Time: 00:27)



Reasoning Tasks

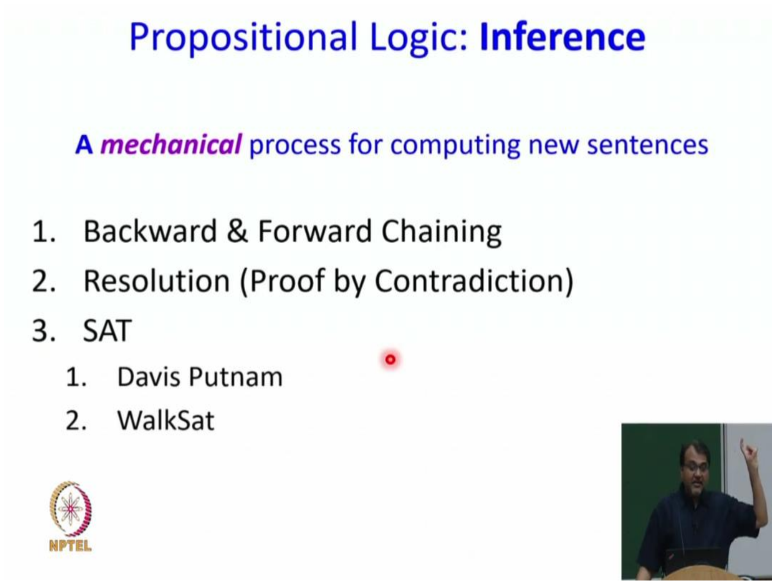
- Model finding
  KB = background knowledge
  S = description of problem
  Show (KB $\wedge$ S) is satisfiable
  A kind of constraint satisfaction
- Deduction
  S = question
  Prove that KB $\models$ S
  Two approaches:
  - Rules to derive new formulas from old (inference)
  - Show (KB $\wedge \neg$ S) is unsatisfiable

17

One is called model finding and one is called deduction. Model finding says, I give you a knowledge base I give you a new problem. Tell me if there is a model for KB and S, which basically means S the new problem is consistent with your knowledge base, this is sort of like constraint satisfaction and we will show you examples. Whereas direction says, I have given you a knowledge base I have given you a new question S.

Can you prove S some knowledge base KB is a true and of course, if I want to prove S some KB then either I can use some rules to derive new formula and finally get to S or I can show that KB and not S is unsatisfiable. When do I ever show KB and not S unsatisfiable have you ever done this that you have to prove something but you said not S and then showed that it is not possible when you do this proof by contradiction? So, this style of proving KB and not S unsatisfiable is equivalent to mimicking proof by contradiction. So, now we are getting closer to our first inference algorithm.

(Refer Slide Time: 01:58)

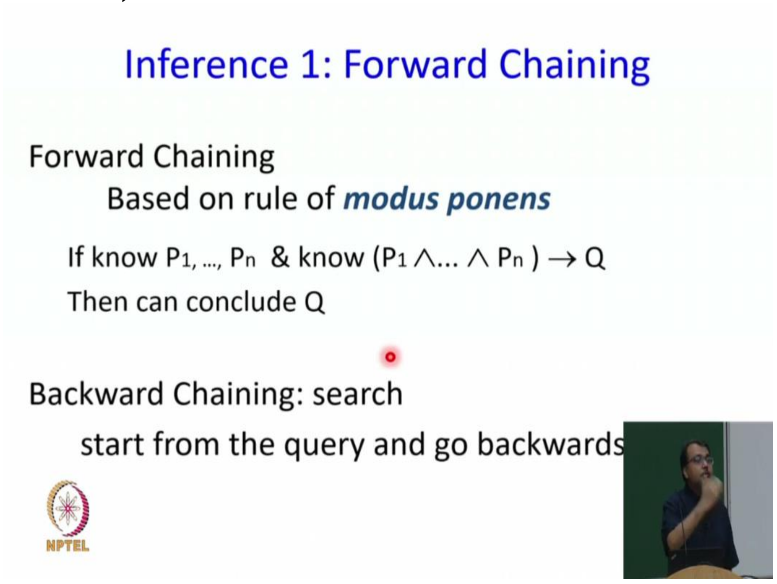

Propositional Logic: **Inference**

A ***mechanical*** process for computing new sentences

1. Backward & Forward Chaining
2. Resolution (Proof by Contradiction)
3. SAT
    1. Davis Putnam
    2. WalkSat

Now, what is an inference algorithm? It is a mechanical process of computing new sentences and we will study 3 of them backward and forward chaining resolution which is equivalent to prove by contradiction and 3 satisfiability problems which is equivalent to model checking model finding.

(Refer Slide Time: 02:18)


Inference 1: Forward Chaining

Forward Chaining

Based on rule of ***modus ponens***

If know $P_1, \ldots, P_n$ & know $(P_1 \wedge \ldots \wedge P_n) \rightarrow Q$

Then can conclude Q

Backward Chaining: search

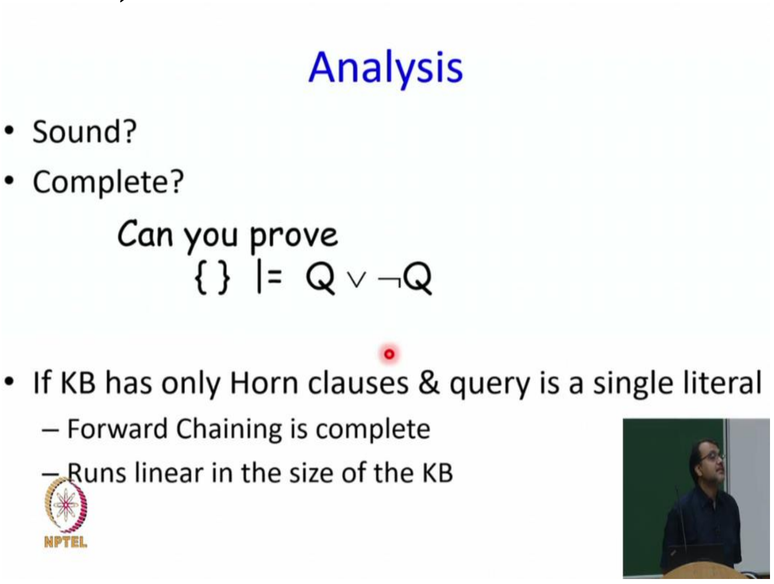
start from the query and go backwards

So let us look at the very first simple inference procedure and that is called the forward checking procedure. The forward checking is based on the rule of modus ponens the modus ponens basically says if I know P 1 and P 2 and P n imply Q and I also know that P 1 is true P 2 is true and P n is true, then I can prove Q to be true. It is as simple as rule A and B, A implies B, A implies B and A entails B.

This is how you will accurately say it in the language of logic that in any world in which A implies B and A is true, then B is true. Now this rule if you apply appropriately and prove new things, this is called forward chaining. I am starting with what I know and using this rule to prove new things in the forward direction. There is always a backward version of things like you had forward search, you had backward search and then you had by directional search the same way.

Backward chaining says that there is something I need to prove. Now, let me see what things should I have had in my knowledge base to be able to prove it. And then let me see how I can prove these things. And I keep going back, and that is called backwards. Anyway, if I only do for modus ponens in the forward direction,

(Refer Slide Time: 03:51)



Analysis

- Sound?
- Complete?

Can you prove

$\{\} \models Q \vee \neg Q$

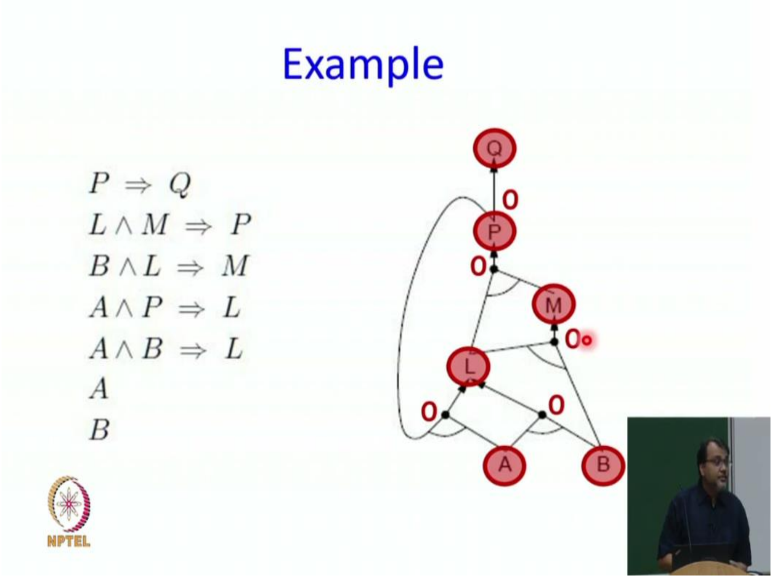
- If KB has only Horn clauses & query is a single literal
  - Forward Chaining is complete
  - Runs linear in the size of the KB

Would it be a sound procedure? Whenever I apply modus ponens and prove a new thing, will it actually be entailed? Come on this is not a very hard question. So, the question is if this is the only rule I was applying that whenever I know P 1 to P n, and I also know that P 1 to P n implies Q and therefore, based on this knowledge I prove that Q is true, will Q really be true? Yes. So, will it be a sound algorithm? Yes, it will never say something which is incorrect.

Will it be a complete algorithm? Will it be able to prove everything that is true? Can you give me an example? It will take you some time to think about, but I will give you a simple example. Can you use this procedure to prove that nothing entails Q or not Q? Is this true? Yes. I do not need anything to prove Q or not Q. It is a tautology. But can I use forward checking to prove it forward chaining to prove it? No, because forward chaining only applies modus ponens and this rule this cannot be proved by modus ponens.

However, we know that if KB only has horn clauses and query is a single literal then forward checking is also complete here is the importance of horn clauses. And in fact, not only is it complete it runs linear in the size of the knowledge base. Let us look at this example.

(Refer Slide Time: 05:31)



Example

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$
A
B

And I think that will be the last thing we will do today. So, suppose these are the things I know in my knowledge base P implies Q and L M implies P B and L implies M and P implies L A and B implies L is true B is true. And now I asked the question is Q true? So, what I can do is I can convert this particular knowledge base into a graph so, here every node is a variable a proposition, then I also know that A and B implies L.

So, the way represented is I take it A an arc out from A arc out from B and put it in this intermediate node. And I make an arc which says 2 of these things need to be true for me to get to L and I make an arc this dot to the 2 L. So basically what I am doing, I am keeping track of how many incident variables do I have in the antecedent, antecedent is A and P and what can I prove somewhere that is the consequent.

And I create this graph structure. And now what I do I put A and B in my Q, because those are the 2 things I know to be true. Now, I can run a simple forward checking algorithm how can I run a simple forward checking algorithm? Because A is true I pop it out from the Q.

And what do I do from there as soon as I know A is true, I know that the dot for A and B the node for A and B, no longer needs 2 things to be true it only needs one more thing to be true.

So the antecedent has already got one thing satisfied, then I pop B and as soon as I pop B, now I know that I do not need to satisfy both the things of this antecedent are satisfied. Because both the things of this antecedent are satisfied now I can prove L because I can prove L I put it in the Q and I repeat this process. So now that I know L I have put it in the Q. From the Q and now, I send the messages out saying I have been through and now you do check for yourself.

So, now this other node which represented B and L now has been satisfied, that means I have been able to move M, because I have been able to move M, I put M into the Q and so on. And suppose, I keep running this procedure, and at some point, I get to Q to be true and when I can get Q to be true, that means I have solved the question I had at hand.

So now you see that this is an inference procedure, which is using a mechanical process for computing new sentences. And in this case, if all the sentences are horn clauses of the form, A and B and C and D implies something and my query is a single variable, or a conjunction of multiple variables, but not a disjunction. So, if my query is a single variable, then I can say that this procedure is guaranteed to prove it and it will be also complete.

I think this is a good point to stop. We have to talk about 2 other inference procedures. One is resolution and one is satisfiability. But before we finish his homework for you, I want you to read about what is a CNF clause and how do you convert any given clause into a CNF class? We will start from there in the next class, and we will work up to the other element. Thank you.