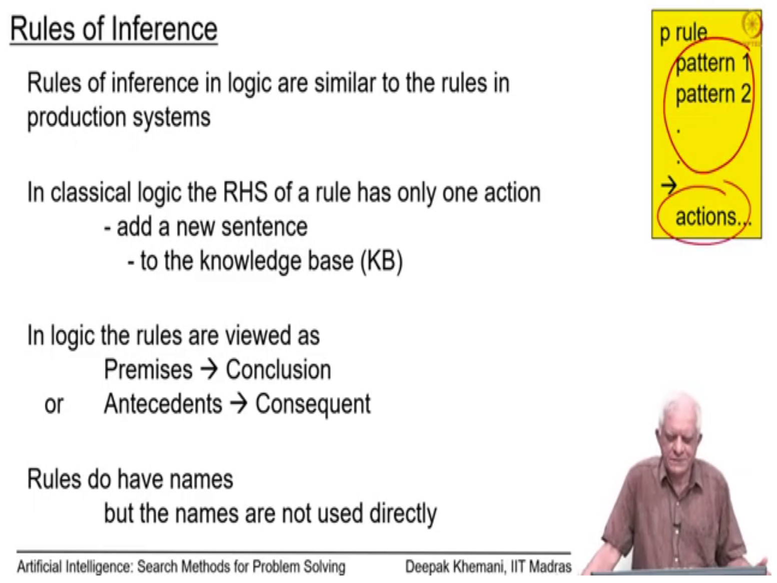
Artificial Intelligence: Search Methods for Problem Solving

Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Chapter – 11
A First Course in Artificial Intelligence
Lecture – 81
Deduction as Search
Rules of Inference

(Refer Slide Time: 00:17)



Rules of Inference

Rules of inference in logic are similar to the rules in production systems

In classical logic the RHS of a rule has only one action
- add a new sentence
  - to the knowledge base (KB)

In logic the rules are viewed as
Premises → Conclusion
or Antecedents → Consequent

Rules do have names
but the names are not used directly

p rule
pattern 1
pattern 2
.
.
→
actions...

Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

So, we are back and let us continue with our study of reasoning in first order logic. And let us start with Rules of Inference. So, we saw the syllogism; it is the syllogism was an example of a rule of inference, we have also seen rules when we talked about rule based systems.

So, rules had a particular structure that they had a set of patterns and they had a set of actions essentially. So, patterns are like antecedents in logic and actions are in classical logic are like add actions or make actions.

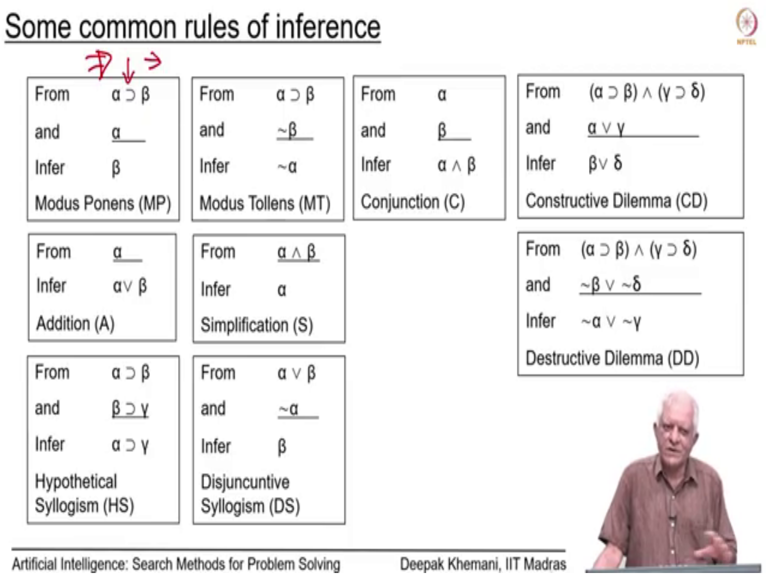
So, rules of inference in logic are similar to the rules in production systems that we have studied. In classical logic, the right hand side of the rule has only one action, which is to add a new sentence to the knowledge base essentially.

When we talk about rules in the logic, we normally see it as an association between premises and conclusion or some logicians would say antecedents and consequent and that is the basic pattern that rule of inference says.

That if the premises are given to you or if the antecedents are given to you; then you can add the conclusion or you can make the conclusion or you can infer the conclusion of the consequent and add it to the knowledge base. We are talking about processing, reasoning, the manipulation of symbols here. Rules do have names and I am sure you are familiar and we will see them very quickly.

But the names are not used directly in the reasoning engine if you want to call them; if you saw the offside like engine it said that, there was something called a conflict set in which the rule name followed by the timestamps of the working memory elements were added to the conflict set. Here the rule name is kind of behind the scenes most of the time; it may be given as a justification as we will see, but it is not part of the processing essentially.

(Refer Slide Time: 02:22)



Some common rules of inference

$\supset \downarrow \rightarrow$

From $\alpha \supset \beta$ and $\underline{\alpha}$ Infer β Modus Ponens (MP)	From $\alpha \supset \beta$ and $\underline{\sim\beta}$ Infer $\sim\alpha$ Modus Tollens (MT)	From α and $\underline{\beta}$ Infer $\alpha \wedge \beta$ Conjunction (C)	From $(\alpha \supset \beta) \wedge (\gamma \supset \delta)$ and $\underline{\alpha \vee \gamma}$ Infer $\beta \vee \delta$ Constructive Dilemma (CD)
From $\underline{\alpha}$ Infer $\alpha \vee \beta$ Addition (A)	From $\underline{\alpha \wedge \beta}$ Infer α Simplification (S)		From $(\alpha \supset \beta) \wedge (\gamma \supset \delta)$ and $\underline{\sim\beta \vee \sim\delta}$ Infer $\sim\alpha \vee \sim\gamma$ Destructive Dilemma (DD)
From $\alpha \supset \beta$ and $\underline{\beta \supset \gamma}$ Infer $\alpha \supset \gamma$ Hypothetical Syllogism (HS)	From $\alpha \vee \beta$ and $\underline{\sim\alpha}$ Infer β Disjuncutive Syllogism (DS)		

Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

So, here are some common rules of inference that you must have surely studied in when you studied logic. The most common of them is called modus ponens as you know doubt are familiar; it says that if you are given two statements, one statement is alpha implies beta that symbol that you see here, you should readers implies.

Some books will read this symbol, some books would use this symbol; but it does not matter, it is all the same. It is a symbol which stands for the implication operator in logic. And if you are given a statement which is alpha implies beta and if you are given a statement alpha; then you can go ahead and add beta to the knowledge base.

So, that is the most commonly used rule of inference called modus ponens. Another one is modus tollens, which is kind of converse of this; it says that if you are given alpha implies beta and if you are given that not beta is true, then you are forced to conclude that not alpha is true.

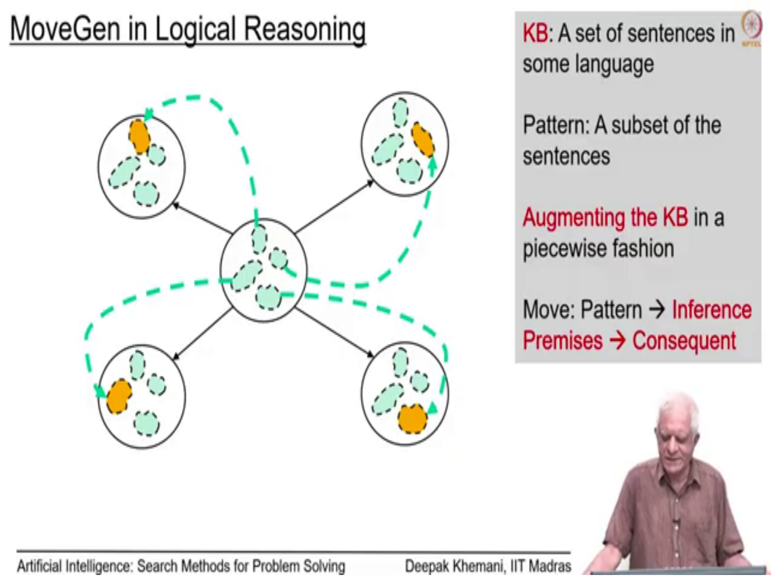
Which means you can add it to your knowledge base and then there are many others. So, for example, conjunction says if you are given alpha, if you are given beta; then you can add alpha and beta to the system.

Then we have these various rules addition, simplification, constructive dilemma and destructive dilemma, hypothetical syllogism, disjunctive syllogism and things like that essentially. So, basically you can have many rules and a rule is meaningful or as we say sound in the language of logic, provided it takes you from true premises to only true conclusions.

So, in this course, we will not look at how to decide whether a rule is sound or meaningful or not; but you can take it for granted that there are a set of rules which can be used for making inferences. And the process of making inference is symbolic in nature is that, if you can see the pattern which the left hand side matches or the antecedents match; then you can simply add the consequent to the right to the knowledge base.

We are not concerned about what though sentences are what is alpha, what is beta we do not care. As long as these patterns are matched, we can make the inference essentially and that kind of emphasizes the fact that logic is essentially formal in nature; that this formal patterns must be adhered to in reasoning. And if you do so, then your conclusions would be justified or sound as we see.

(Refer Slide Time: 04:56)



MoveGen in Logical Reasoning

KB: A set of sentences in some language

Pattern: A subset of the sentences

Augmenting the KB in a piecewise fashion

Move: Pattern → Inference
Premises → Consequent

Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

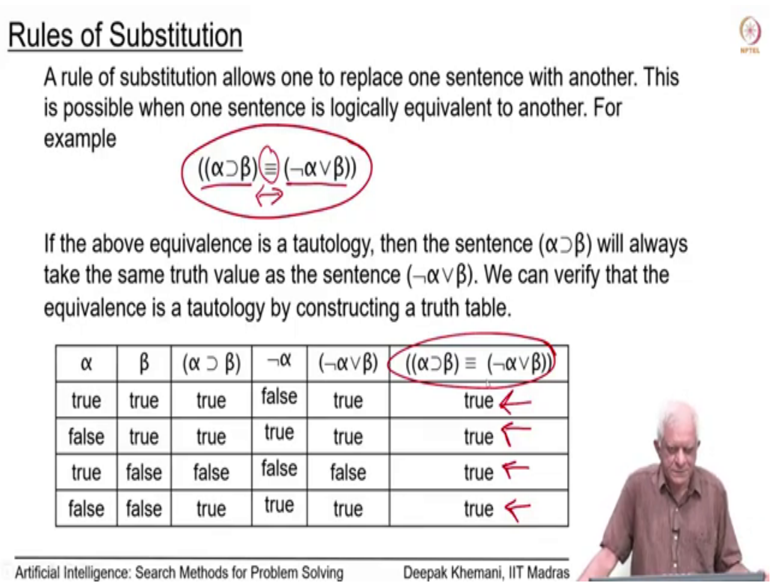
So, given these moves that you can make, inferences that you can make; the process is similar to what we discovered, discussed when we talked about rule base systems and therefore, you can see that the diagram is also similar.

What is different is what I have written in red here is that, it is a knowledge base that you are talking about, it is not a state. In that time we were talking about the state and so the knowledge base is a set of sentences in some language and obviously it could be describing a state, but that is another matter.

A pattern is a subset of those sentences; you can augment the knowledge base in a piecewise fashion by you know adding new sentences. And the new sentences are basically those with conformed to the pattern, inference pattern or the premise consequent pattern. And in that sense, every pattern that you see can lead to something else being added to the knowledge base.

So, it is very similar to what we saw in rule base system, except that the only actions you can do a add a statement to the knowledge, knowledge base. And we are talking about a knowledge base, rather than a state essentially.

(Refer Slide Time: 06:04)



Rules of Substitution

A rule of substitution allows one to replace one sentence with another. This is possible when one sentence is logically equivalent to another. For example

$$((\alpha \supset \beta) \equiv (\neg\alpha \vee \beta))$$

If the above equivalence is a tautology, then the sentence $(\alpha \supset \beta)$ will always take the same truth value as the sentence $(\neg\alpha \vee \beta)$. We can verify that the equivalence is a tautology by constructing a truth table.

α	β	$(\alpha \supset \beta)$	$\neg\alpha$	$(\neg\alpha \vee \beta)$	$((\alpha \supset \beta) \equiv (\neg\alpha \vee \beta))$
true	true	true	false	true	true
false	true	true	true	true	true
true	false	false	false	false	true
false	false	true	true	true	true

Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

There are also something called rules of substitution. So, a rule of substitution is like a two way rule; it simply says that, the left hand side in this case a statement which is alpha implies beta is equivalents to. So, this, these three arrows are to be read as equivalent to some books may use this symbol.

But the it is a logical operator, one of the binary operators that are there in logic is equivalent to this other statement which is not alpha or beta. So, I am sure all of you are familiar with this equivalence and you can write a statement alpha implies beta or you can write it as not alpha or beta, and they mean the same thing; in the sense that they are true at the same time essentially.

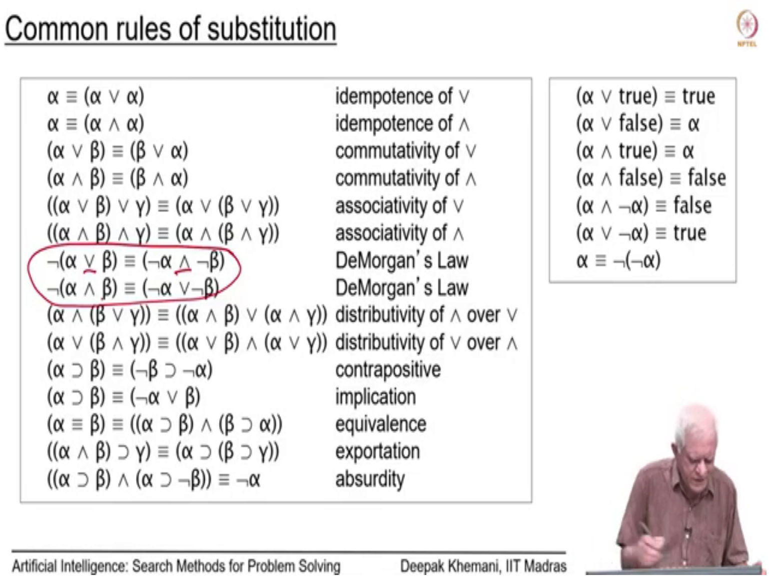
Whenever the alpha implies beta is true, not alpha or beta will be true and whenever alpha implies beta is false, not alpha or beta I mean. So, if they are logically equivalent; which means you always have the same truth value, given the truth values of alpha and beta, then we say these two are equivalent. And a rule of substitution says that, if you can see one of them in your knowledge base; you can go ahead and add the other one.

So, you can go from left to right or you can go from right to lift. In modus ponens, you could only go from left to right. If you are given alpha and you are given alpha implies beta, then you could add beta. But if you are given alpha implies beta and you are given beta, you could not add alpha; you should probably go back and look at why that is not a sound rule.

Rules of substitutions work in both directions; if you see the left hand, can add the right hand side; if you see the right hand side, you can add the left hand side. And the key to accepting them is that, they are based on the statement which is a tautology; that this above statement, this whole statement that we are talking about, if that is a tautology as you can see here in this truth table and you can go and construct the truth table yourself in more detail.

But what is important is that, in every one of these rows in the last column, it is true. So, this statement is always true. If it is a tautology, then you can use it as a rule of substitution. There is a similar reasoning for rules of inference; but these are, those are tautological implication statements, but we will not go into those details right now.

(Refer Slide Time: 08:36)



Common rules of substitution

$\alpha \equiv (\alpha \vee \alpha)$	idempotence of $\vee$
$\alpha \equiv (\alpha \wedge \alpha)$	idempotence of $\wedge$
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$
$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	DeMorgan's Law
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	DeMorgan's Law
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$
$(\alpha \supset \beta) \equiv (\neg\beta \supset \neg\alpha)$	contrapositive
$(\alpha \supset \beta) \equiv (\neg\alpha \vee \beta)$	implication
$(\alpha \equiv \beta) \equiv ((\alpha \supset \beta) \wedge (\beta \supset \alpha))$	equivalence
$((\alpha \wedge \beta) \supset \gamma) \equiv (\alpha \supset (\beta \supset \gamma))$	exportation
$((\alpha \supset \beta) \wedge (\alpha \supset \neg\beta)) \equiv \neg\alpha$	absurdity

$(\alpha \vee \text{true}) \equiv \text{true}$
$(\alpha \vee \text{false}) \equiv \alpha$
$(\alpha \wedge \text{true}) \equiv \alpha$
$(\alpha \wedge \text{false}) \equiv \text{false}$
$(\alpha \wedge \neg\alpha) \equiv \text{false}$
$(\alpha \vee \neg\alpha) \equiv \text{true}$
$\alpha \equiv \neg(\neg\alpha)$

Artificial Intelligence: Search Methods for Problem Solving   Deepak Khemani, IIT Madras

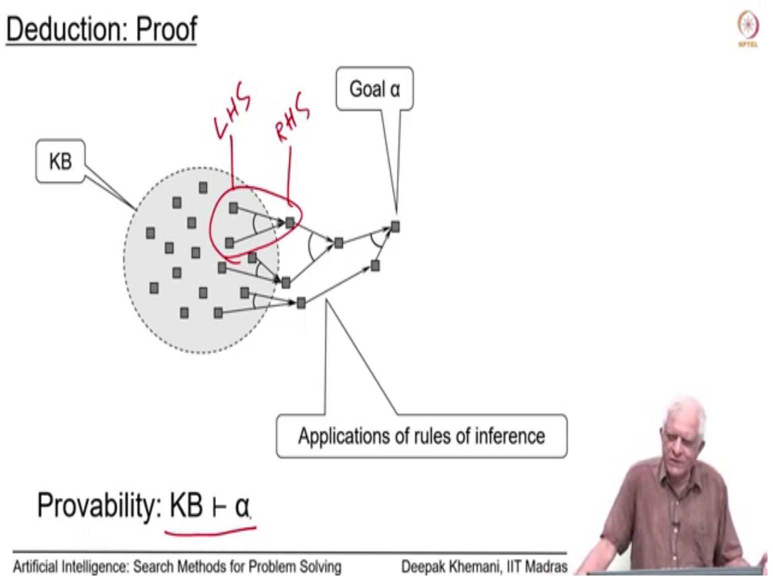
There are many rules of substitution, again you would have studied them. Our intention is not to study logic here. So, it is not that I expect you to go and revise these rules or something; but it is just to point out that there are many such things.

One very common one that you might remember or two very common ones are these De Morgan's Laws, which says that you can push the not inside a bracket and when you push it inside; it changes the sign of the, changes the nature of the connectives. So, if it was or to start with, it will become an and, and vice versa; so these are De Morgan's law.

So, if you see something on the left hand side, you can replace it with the right hand side and vice versa. And there is a whole host of rules of inferences; we need these rules of substitution, because logic is so formal in nature that you know if the pattern has to match exactly.

And only if it matches exactly, then you can go ahead and make an inference and we will see this as we look at making logical inferences. So, here is our rule of substitution that we discuss in the last few minutes, ok.

(Refer Slide Time: 09:46)



Deduction: Proof

Provability: KB ⊢ α

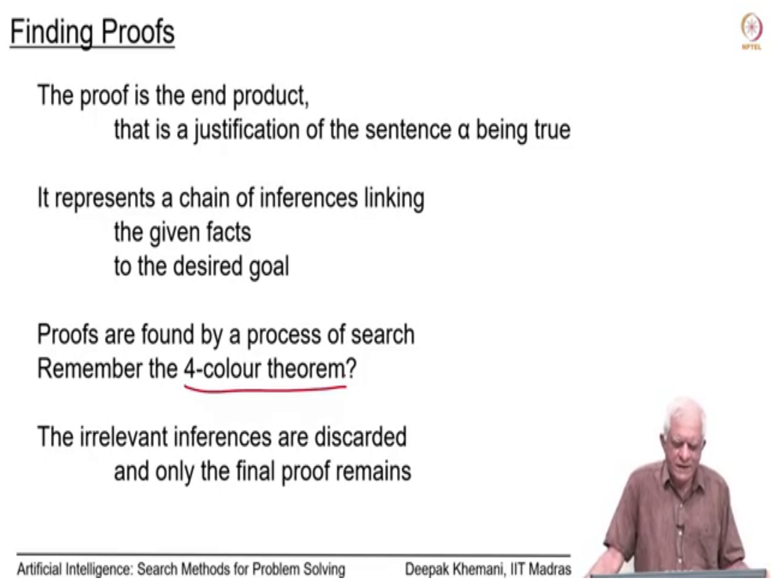
Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

So, we have said that deduction is about connecting true statements to other true statements. And we have also said that the process of doing so is formal in nature, symbolic in nature, syntactic in nature. And what we are driving at is the notion of a proof essentially. So, a notion of a proof is different from the notion of that being true.

So, we are asking here that, given a knowledge base, the same knowledge base given the same goal alpha; can you apply a sequence of rules of inference? You can see that each of them is like an application of a rule of inference; the left hand side is the antecedents, the right hand side is the consequent for each rule. So, every such thing is a rule essentially; can you apply a sequence of rules, so that you end up adding the goal to the knowledge base?

If you can do that, and these rules of inference must be from the allowed set of rules of inferences; then we say that we have a proof of alpha. And we say that alpha is provable and we use this symbol here that the knowledge base proves alpha. So, that is a notion of a proof essentially; if you have a proof, then you are willing to accept alpha essentially.

(Refer Slide Time: 11:24)



Finding Proofs

The proof is the end product,
that is a justification of the sentence α being true

It represents a chain of inferences linking
the given facts
to the desired goal

Proofs are found by a process of search
Remember the 4-colour theorem?

The irrelevant inferences are discarded
and only the final proof remains

Artificial Intelligence: Search Methods for Problem Solving Deepak Khemani, IIT Madras

How do you get proofs? Now, finding proofs is a process of going through search; the proof is only the end product that is a justification of a sentence being true essentially. It is a sequence of inferences; you must be remembering that, in geometry you write a proof you say that this is given and therefore, you can conclude this.

And therefore, you can conclude this and therefore, you can conclude this and so on. So, you have a sequence of lemmas on the way that you prove, and eventually you prove the theorem that you want to prove. But those sequence of statements are the proof essentially. How do you get the proof is the question that we are asking? Because we want machines to find proofs for us.

The proof represents the chain of inferences linking the given facts to the desired goal essentially; they are found by a process of search essentially. So, you must have heard about the 4 colour theorem; it is a very well known theorem which says that, any map on a planar surface or even on a globe for that matter; can be coloured using 4 colours. And what do you mean by can be coloured?

Basically two regions or two countries must which are adjacent to each other, must not have the same colour. And the 4 colour theorem says that, you give us any map which is on a planar surface, not in some three dimensional world or something like that; then you can colour it using 4 colours.

And it took more than a couple of hundred years for someone to arrive with a proof for this statement. And in fact, the proof was obtained by our computer program, which has a program of the kind that we are going to be talking about.

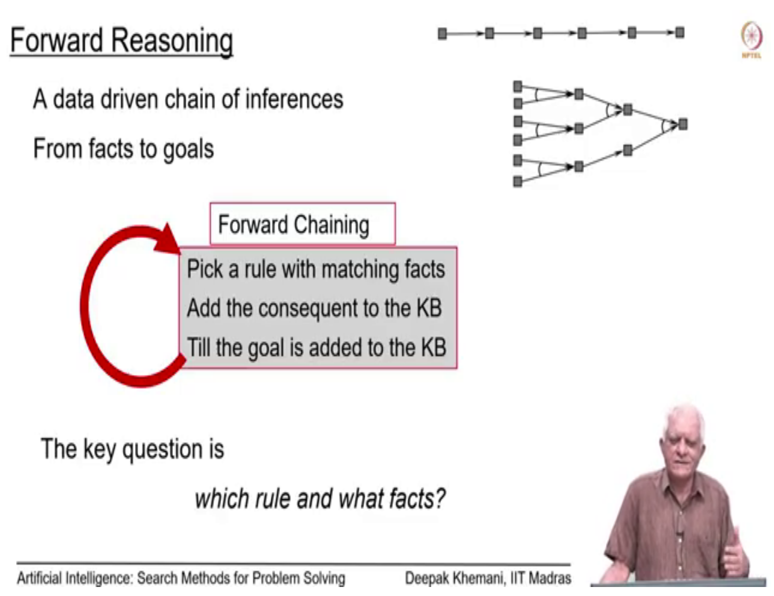
Or you must have heard about Fermat's last theorem; Andrew Wiles after spending tens of years in an isolated cottage pondering over it. Finally found a true for the Fermat's theorem which Fermat in those times had written in the margin of his notebook saying that, there is a very concise proof for this theorem; but it took a few hundred years for the rest of humanity to find that out.

So, finding proofs is a not a straightforward task, and finding proofs is what mathematicians occupied themselves (Refer Time: 13:54). So, they keep pondering about how to, prove things.

I have a friend a classmate of mine who for the last at least 10 years has been trying to find the proof for something called the Beal's conjecture, which is a generalization of Fermat's last theorem, but he is still working on it. So, the proof finding processes is not straightforward, it is involved search; but when you have found a proof, mathematicians do not give you the process by which they found of proof, they just give you the proof.

So, the irrelevant inferences are discarded and the neat connection from the knowledge base to the goal is presented as a proof essentially; but finding proofs is a search process.

(Refer Slide Time: 14:42)



Forward Reasoning

A data driven chain of inferences

From facts to goals

Forward Chaining

Pick a rule with matching facts
Add the consequent to the KB
Till the goal is added to the KB

The key question is

which rule and what facts?

Artificial Intelligence: Search Methods for Problem Solving   Deepak Khemani, IIT Madras

We will start by looking at something called forward reasoning, which is very similar to what we have been talking about earlier forward state space, search forward state space, planning and rule based systems.

And it the process not surprisingly is similar; it moves from facts towards the goals, it is a data driven chain of inferences that you are making. And it can be kind of captured by process which is you are seeing on the top right of your screen; that one by one you are moving away from what is given to you towards some new facts and you keep adding new facts.

The top row shows only rules with one antecedent, the bottom row shows rules with two antecedents and you can see the kind of process forward chaining process that happens. Now, what you are seeing there are proofs. How do you find the proofs? You as I said you find it through a process of search. And we have a forward chaining algorithm, which is very similar to what we did earlier when you did search.

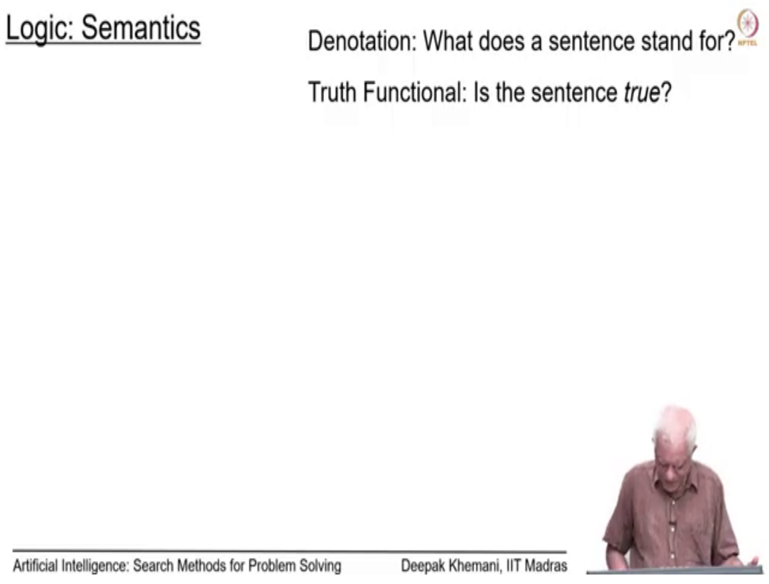
It says that, pick a rule with matching facts; add the consequence to the knowledge base and keep doing that till the goal is added to the knowledge base. It is very much similar to the basic search algorithm that we have been talking about, high level search algorithm that we have been talking about. The key question of course is, which rules should we pick and what matching facts we should apply to?

So, we saw a glimpse of this when we talked about conflict resolution strategies in production systems, that you can have different kinds of strategies. We are talking about automated strategies ok; we are not talking about a mathematician which combines other processes to finding the proof, to work towards finding the proof.

Not just simple search, you know they make hypotheses; they make guesses and they try something, then they discard something and all those are many other processes that real mathematicians work about.

But we want our program to you know in some systematic way go about doing that and eventually find a proof and that is how in fact the 4 colour theorem was proved essentially, ok.

(Refer Slide Time: 16:56)



Logic: Semantics

Denotation: What does a sentence stand for?

Truth Functional: Is the sentence *true*?

Artificial Intelligence: Search Methods for Problem Solving   Deepak Khemani, IIT Madras

So, we have this notion of finding proofs. So, let us take a very quick break at this point. In the next segment, we will try and talk about, when is a proof acceptable to us? Remember that a proof is a syntactic process; its a symbolic process, you are looking for a pattern and adding the consequent to the knowledge base and keep doing that till this process, till the goal is added. When is it going to be acceptable to us and what kind of a logic will allow us to do that essentially?

Or what kind of an algorithm with, what kind of representation will allow us to do this? The representation we have frozen here which is first order logic; but we will talk about algorithms starting in the next segment. So, we will take a short break and come back.