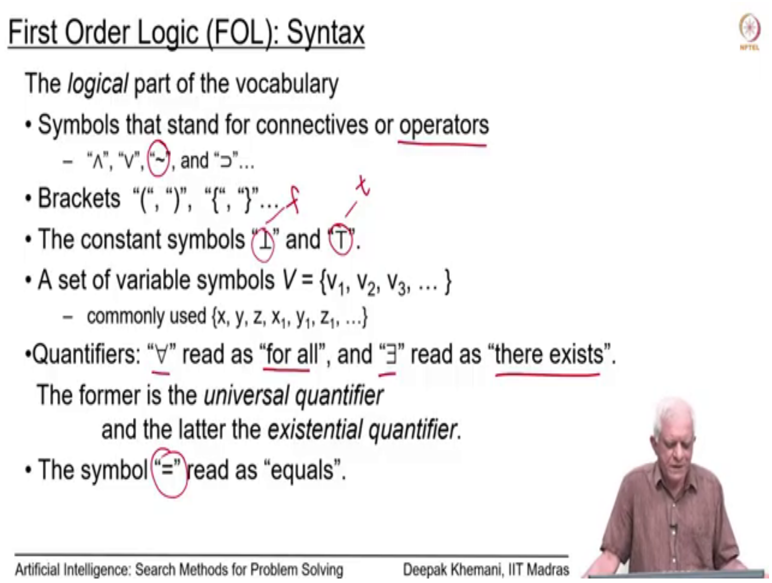
Artificial Intelligence: Search Methods for Problem Solving
Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Chapter – 11
A First Course in Artificial Intelligence
Lecture – 83
Deduction as Search
First Order Logic

(Refer Slide Time: 00:15)



So, here we are looking at the logic machine. Let us look at First Order Logic in a little bit more detail; it is a language that we have chosen to represent the sentences. What are sentences? Sentences are basically statements which are true or false essentially, that is the whole idea of a sentence from the logic perspective. That you give us a sentence and if it is a sentence, it must either be true or it must be false. In principle, you do not necessarily have to know whether it is true or false.

So, the my favorite example here is to say that, white always wins in chess. So, what do we mean by that? We mean that, whenever white is playing, whenever both players are playing perfectly; white will always win. But we do not know, because you know we have not explored the chess tree completely unlike the Tic-tac-toe tree.

So, we do not know whether it is true or not, but in principle the sentence can be true or false. So, anything which can in principle be true or false is a sentence essentially; which is of course different from other statements like questions, interrogative sentences, can you tell me what is the best way to solve this Sudoku or imperative statement, can you please come over to my lab in the afternoon.

These are different from statements which are true or false and logic is concerned with statements sentences which are in principle true or false and we have chosen the language of first order logic to represent sentences. So, let us look at this language very briefly; I am sure you are familiar with this, but just to serve as a recap. There is a logical part of the vocabulary, which has symbols which stands for connectives or operators.

So, we often call them as binary operators or unary operators; there is one unary operator here which is negation, but you have another symbol for negation as well. And there are operators that you are no doubt familiar with and or implies and so on and so forth; equivalence we talked about. We have brackets, curly brackets or braces or brackets as we call them; then we have constant symbols, these are just part of the language and we have two constant symbols.

We call the first one the bottom and we call the other one the top essentially. And sometimes some people may use false for bottom and true for top, it does not matter; there are two symbols which are associated with the language. And as you probably know, they are associated with two kinds of truth values. So, false is always false and true is always true essentially.

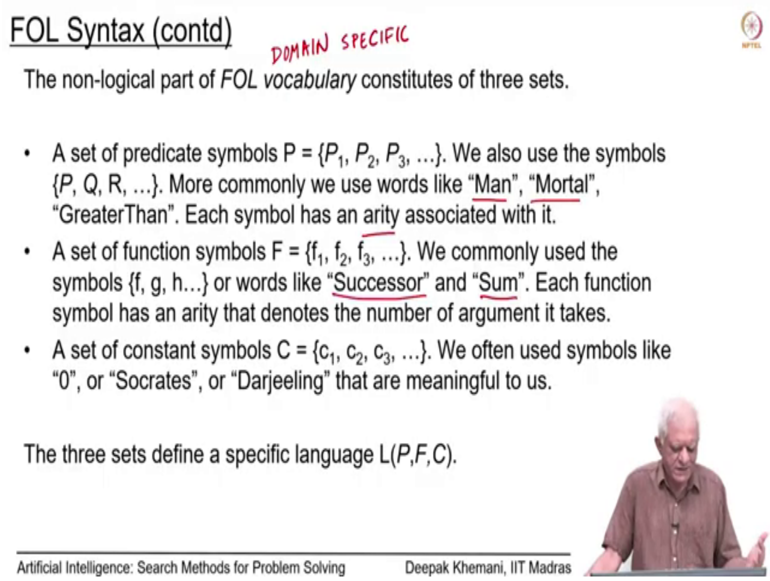
Then in first order logic and this is even from professional logic; we have a set of variable symbols v 1, v 2, v 3 and so on, sometimes we use x, y, z and x1, x 2 and so on. Then we have quantifiers, most often we just talk about two quantifiers; one is read as for all, and the other is read as there exists.

And the first one is called the universal quantifier and the second one is called the existential quantifier and we will look at their usage in a few minutes. And often we identify a separate symbol for equality, and people sometimes distinguish between first order logic and first order logic with equality.

And the reason for that is that, equality when you use in a language; you have to be aware that the equality relation has certain properties which must be respected, see otherwise your system will not be complete. What are these properties? The properties about reflexive that, x is always equal to x; or symmetry if x is equal to y, then y is equal to x or transitivity and so on.

So, there are certain axioms associated with equality. So, if you are using equality as a symbol in your language, you must pull in those axioms as well; but we will leave it here for the moment here.

(Refer Slide Time: 04:49)



Then there is a non logical part or the domain specific part of the language. So, whenever we are devising a language, we have to choose certain set of symbols which are to do which the problem that we are trying to solve.

That is sometimes it is called the non logical part; in the sense that, it is not common to every language that you will devise. And when I say language, you mean a set of sentences; which means you what are the constant symbols, what are the variable symbols and so on, the symbols that we are talking about are domain specific.

So, there are three kinds of such sets; one is a set of predicate symbols that you are familiar with no doubt, typically we use the word P 1, P 2, P 3 to stand for those predicates. We can also use P, Q, R essentially; but very often in logic textbook, we use things like Man and Mortal. So, we said all men are mortal right or Greater Than and things like that.

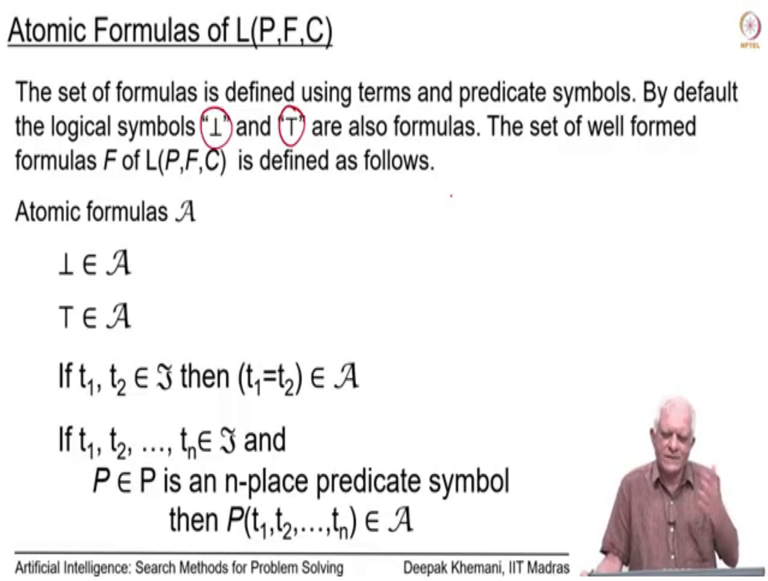
We also have an identity associated with it, which tells us how many arguments does that predicate or relation have essentially. Then we have a set of function symbols; typically we may choose to use lowercase English letters for that; so f, g, h or f 1, f 2, f 3 and so on. So, we have functions like the successor function in natural numbers or the sum function also in natural numbers, which basically maps elements onto other elements in the domain essentially.

So, we when we say sum of 2 plus 5, it represents another element and that element is as we know 7 essentially. So, functions basically maps, map elements to other elements essentially. In relations mother is a function, because everybody has a unique mother. So, if I say mother of John, then I know that what is the person I am talking about.

Then we have a set of constant symbols, in every domain we may have constants; so we have Socrates and we have numbers like you know 0, 1, 2, 3 and so on and so forth or Darjeeling, all these are constant symbols and they are just part of the language.

So, we have these three sets; the set of predicate symbols, the set of functional symbols, and the set of constant symbols and they together define a language which is used to talk about the domain, in some sense it corresponds to the short term memory that we had spoken about earlier.

(Refer Slide Time: 07:27)



So, languages have sentences and the simplest kind of sentences in language logic are called atomic formulas. And we can define a set of atomic formulas; the simplest are the two symbols bottom and top. Remember that I said that, sentences are associated with truth values.

So, either they are true or they are false; anything which in in principle is can be true is a sentence. Bottom is always false and top is always true. So, these are constant sentences (Refer Time: 08:01); other sentences may depend upon the domain essentially. So, bottom and top are atomic formulas and we have a notion of a term; term basically stands for an element in the domain.

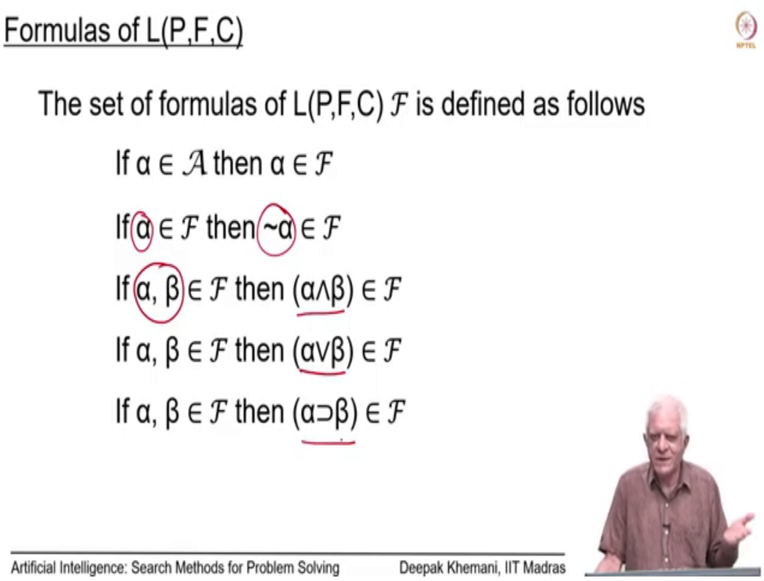
So, so if there is a domain we are talking about and you have some element; it can be either a variable or it can be a constant or it can be an output of a function, output in the sense it can be denoted by a function. So, sum of 2 and 5 for example, denotes a term; a term is basically an element of the domain.

So, if there are two elements t 1 and t 2 and they belong to this set of terms, which we have not defined, but I have done it very informally here; because they are kind of rushing through this whole thing here. Then this expression t 1 equal to t 2 is a atomic formula. What is an atomic formula? It is a sentence which is either true or false and it is the smallest possible sentence and that is why it is called atomic essentially.

So, you might say the two terms for example, New Delhi and the capital city of India these are two terms; because they refer to two specific entities in some domain which is a political map of a country let us say, they are equivalence. If I say Delhi is equal to capital of India and then essentially we are saying that we are talking about the same thing essentially. The other kind of atomic sentences are made up using predicate symbols.

So, if P is a predicate symbol and it is an n place symbol which means identity is n; then it can take n terms as arguments and it becomes a atomic formula. So, for example, Suresh, brothers Suresh, Ramesh; it takes two terms and it becomes a sentence. And being a sentence, it is either true or it is not true; it depends on whether Suresh is really the brother of Ramesh or not essentially.

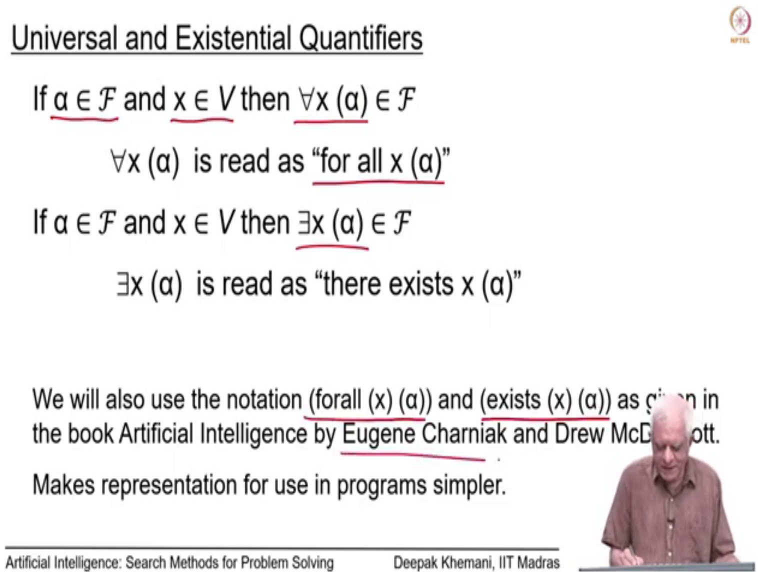
(Refer Slide Time: 10:12)



So, then we have atomic sentences, and like in all languages that you would have studied; you can construct more complex sentences by using the operators in the language. So, if alpha is a formula, then not alpha is a formula; you can add a negation sign and produce another formula. If alpha and beta are formulas, then alpha and beta are formulas; alpha or beta is a formula, alpha implies beta is a formula. So, you can keep adding.

So, as you can see, the language will have infinite sentences, infinite formulas essentially and that is because you can keep using connectives and keep adding more and more formulas to that. So, that is the language that we are talking about.

(Refer Slide Time: 10:52)



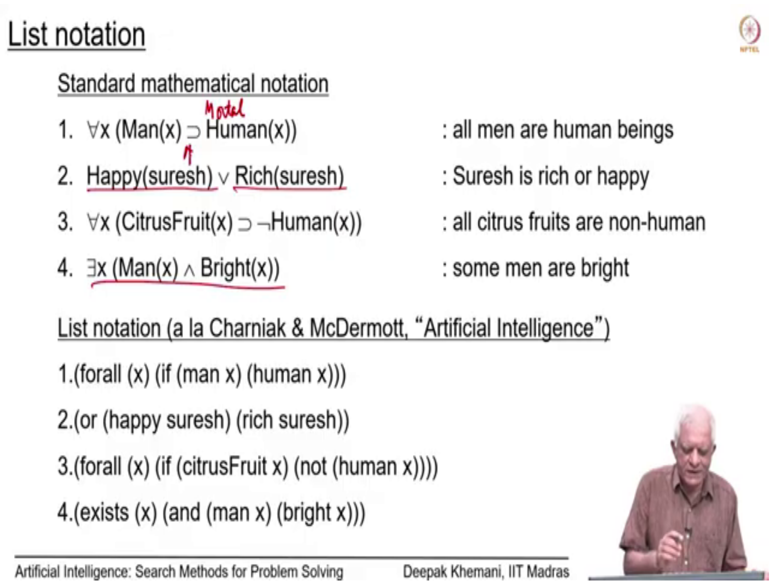
We had mentioned that in first order logic, it is different from propositional logic in the sense that it uses the two quantifiers; existential and universal quantifier and we had also mentioned this.

So, we can conduct, construct sentences using these quantifiers; we say that if alpha is a formula and x is a variable, then for all x alpha is a formula. So, you can take any formula, you can take a variable; quantify the variable using a quantifier, in this case it is a universal quantifier and say that, that sentence is a formula.

And the way we read this is, for all x alpha as your (Refer Time: 11:38) formula. Likewise if for the existential quantifier, we have there exists x alpha as a formula and it is read as; there exists an x such that alpha is true essentially. For example, you might say, there exists a number which is even essentially, even x; or for the universal formula you can say that, that for all numbers x plus 1 is greater than x or something like that essentially.

So, these are true statements, you can also of course make false statements. If you are interested in programming, you may want to use these alternative notation which is a list notation, which is given to us by Charniak and McDermott, who wrote a very popular book in 1976 or something like that, which would be easier to write programs with; whereas these mathematical notations are not so easy to type in to text programs. So, here you can just confine yourself to text and write programs for doing logical reasoning.

(Refer Slide Time: 12:47)



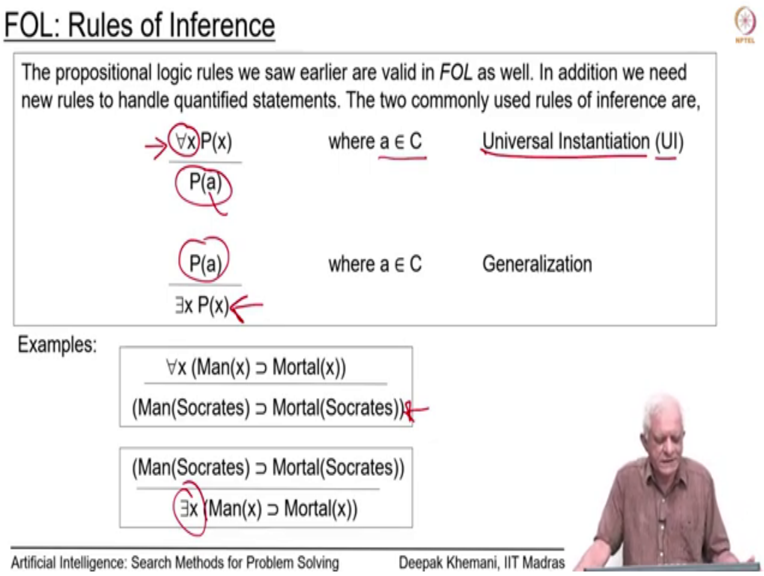
That is called the list notation. So, if you have standard mathematical notation; the first one says that, all men are human, all men are human beings, this is in the language of logic that we have been talking about.

If you replace this with mortal, then it will give us the one of the statements we use in the Socratic argument; man x implies mortal x, all men are mortal. The second one says Suresh is rich or Suresh is happy; the third one says that, all citrus fruits are not human. So, we know of course, that fruits are not human. And the fourth one says that, some men are bright, which logically speaking says there exists an x, such that x is bright and x is a man essentially.

So, the difference between all men are bright and some men are bright is denoted differently and maybe you should pay some thought of attention to that. You can use that the same set of sentences in list notation, which was given to us by Charniak and McDermott is written slightly differently. As you can see, one difference is that in the mathematical notation; the operator comes between the two formulas that that you are connecting to each other.

So, this is alpha, this is beta and this is alpha implies beta. In this list notation, the f comes first; the antecedent comes here and the consequent comes here. Notation is different, the meaning is the same essentially; but you can see that the second set of sentences that we have here, they would be much easier to process if you are writing a simple program to do that. So, all the four sentences on the top are repeated here in the list notation.

(Refer Slide Time: 14:50)



We spoke about rules of influence which were general in nature, which are some people refer to them as proportional rules of influence; because when we say alpha implies beta and alpha means, you can add beta, the modus ponens, we are not saying what language alpha is in.

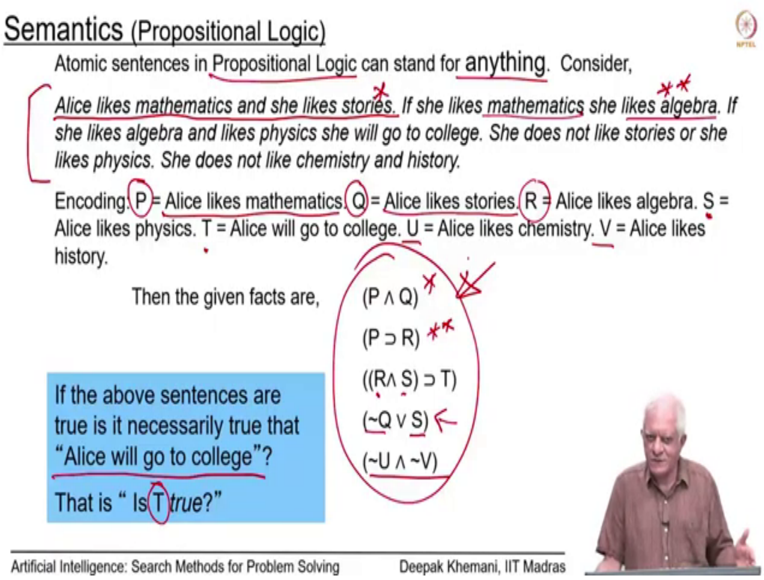
But there are certain sentences which are certain rules of inference, which are specific to first order logic. And there are two of them; one is called universal instantiation, we will use the term UI for short. And what this says is that, if you are given a sentence of the kind which says that, for all x P x is true; then you can infer the sentence that a is true, where a is a constant here.

Now, obviously, we can see this makes sense; because when you say for all x P x is true, that means you take any element from the domain and it must be true. So, in particular if you take any particular element a, it must be true for that. And the converse in some sense is that, if you know that P a is true; then you also know that there exists an x such that P x is true, because you know that P a is true essentially.

So, examples of these are, if you were to say all men are mortal; you can infer from that this sentence which says that, Socrates is mortal. So, in the logic language all men are mortal dose represented as for all x; if x is a man, then x is mortal. The second of the inferred statement is, if Socrates is a man; then Socrates is mortal.

So, the first statement was universal or universally quantified; the second statement is specific to Socrates, in fact you can say it is a proficiency. And likewise example for generalization; if you say that Socrates is mortal, then you can say that all there, there exists a man who is mortal. So, we have to add these two rules of inference. Why are we interested in all these? Because we know we want our logics to be sound and complete essentially.

(Refer Slide Time: 17:01)



Now, we will start looking at proof procedures. Let us begin with propositional logic and let us first start with the representation and state the problem. You are given a set of sentences and which is given in this paragraph that you can see here. And for each sentence, we have coded them using propositional symbols. In propositional logic, which is simpler than first order logic; any sentence can stand for anything essentially.

So, we have a sentence called P, it says that Alice likes mathematics. And we have another sentence called Q, a professional symbol called Q and this says Alice likes stories. So, if you have a sentence in our story, which says that Alice likes mathematics and Alice likes stories; we can represent this sentence by this formula here which is P and Q. Why?

Because P says Alice likes mathematics and Q says that Alice likes stories. The second sentence is, if she likes mathematics, she likes algebra. So, well of course, we are kind of making sense of the fact that, she is a pronoun and were talking about Alice and so on; all that natural language processing we are doing mentally.

But essentially there is a relation between liking mathematics which is P and liking algebra which is R here. So, here we have a statement which says that, P implies R stands for the second statement in our story essentially. Then the third statement says that, if she likes algebra and likes physics; then she will go to college.

So, as you can guess likes algebra is stands for R and S which is there somewhere here, there you can see stands for likes physics. So, if both these are true; then T stands for Alice will go to college. And then there are two more statements; she does not like stories or she likes physics, this is the statement here, she does not like stories or she likes physics.

Notice that this is not an exclusive hour; it does not mean that one of them only can be true; it means that, either one or both can be true. And she does not like chemistry and she does not like history; so chemistry is U here and history is V here and we know that both she does not like essentially.

So, this is a formalization of the story which is there in the English language above this. And now we are ready to go around looking at the proof process essentially. What are you, what do we want to prove? The question we want to ask is; is it true that given this knowledge base, what is the knowledge base?

Alice likes mathematics and she likes stories; then the relation that, if she likes math's, she likes algebra; if she likes algebra and she likes physics, she will go to college, she and there are two other facts which may or may not be relevant here. But given this whole set of sentences which is a knowledge base; we are asking, is it true that Alice will go to college?

And by now we have resorted to looking at a proof process for arriving at truth value statements; which means if you can find a proof for the statement, what is the statement T. Is T true or not? Will she go to college or not? If you can find a proof for T, given this knowledge base which is now expressed in propositional logic; can we prove T? And because we have accepted that our logic is sound and, completeness does not matter here.

Because we have accepted to be sound; if we can prove it, then we will be happy to accept that this statement is true. So, we look at the proof after a short break in the next this thing. And we will also extend this story to first order logic essentially; because that is what we are really interested in. So, again we will take a short break and we will look at the proof process after this, see you.