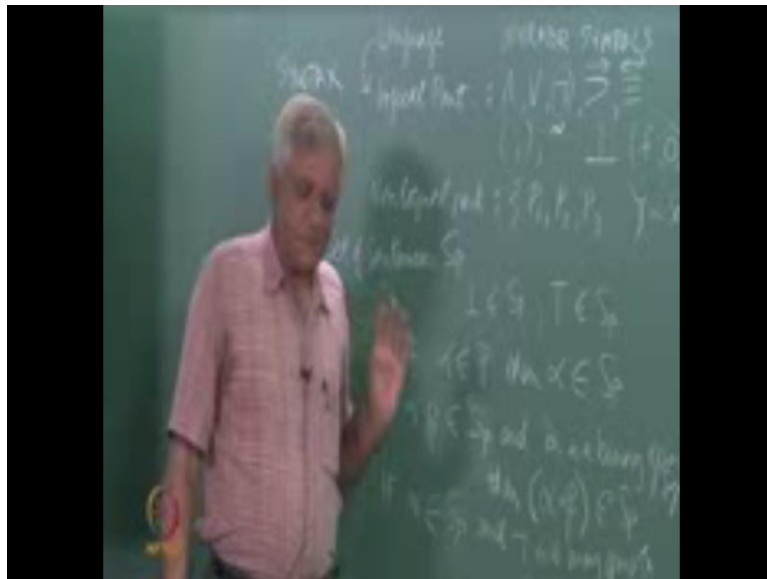


Artificial Intelligence
Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 42
Knowledge Based Systems PL

(Refer Slide Time: 00:16)



So, today we want to look at propositional logic. So, to keep things into prospective our motivation for studying logic is to be able to make inferences. So, in general if he given a set of facts and if he ask a statement about which is not in this set some statement. And we are ask the question is this statement true or not true then we should be able to answer that question. And you want to build the mechanism or a machinery to do this automatically or mechanically if not necessarily automatically. By mechanically we mean by applying a set of rules which can be then by human as well. So, whenever we talk about logic, we have first talk about the language part which like any formal language is a set of sentences that belongs to this language. So, what is the language of propositional logic? It has 2 components; one is called the logical part and it essentially contains the alphabet of these operators or operator symbols. So, you must keep in mind that we want to distinguish between the syntax of the language and the semantics of the language and do you not do you.

So, a very pedantically in this lectures, but we need to distinguish between the fact thus the syntax part which says that these are some symbols that we use. But off course, the meaning of this symbols calms from this semantics and which as you are familiar with is given by the through tables at we can construct for each of these operators. So, at one level other syntactic level we just treat them as symbols but when we talk about the semantics of sentences. Then we will sort of reverts to the idea which comes on the proof table essentially. So, for the movement we have focusing only on the syntax part just think of them only a symbols essentially. Then there are other things like brackets that we can use then there is a non logical part by thus we mean it varies from problem to problem essentially let us as assume that be since here. So, we have symbols which is set of proposition symbols. So, this is the alphabet with which we construct the language we have a concern at which we use and all progression logics. And we have a varying set which may be in principle and infinite or at least countable infinite or it could be fine at as well essentially.

We just talking about the 5 statements speak you are a steep then this set will be simple set of statement. So, this is a countable set of propositional symbols a we have 2 more symbols here one is I will use this notation. But you could also use f was 0 and the other is this, but you could also used t or 1 or even true and falls the full letters. It does not matter basically they are these 2 symbols which are they are in all logics this you often read as bottom. And this we read as top essentially bottom are falls or 0 are equivalent. Basically it is a set of 2 symbols it is a 2 valued logic that we have talking about. And when we talk about semantics then every sentence in our language will be map to one of these symbols. And that is how the semantics will be defined month we would implicitly between us say that it is stands for true sentences. And the other stands for fall sentences I mean this is fall sentences and this is true sentences for that is only for us our sake. As far as the logic itself is concerned the machine in self is concerned they are just 2 symbols to which sentences are mat on true essentially. But a travelers helps to keeping mind the semantics which is that we are talking about false or true statements. What is this machinery that we have talking about?

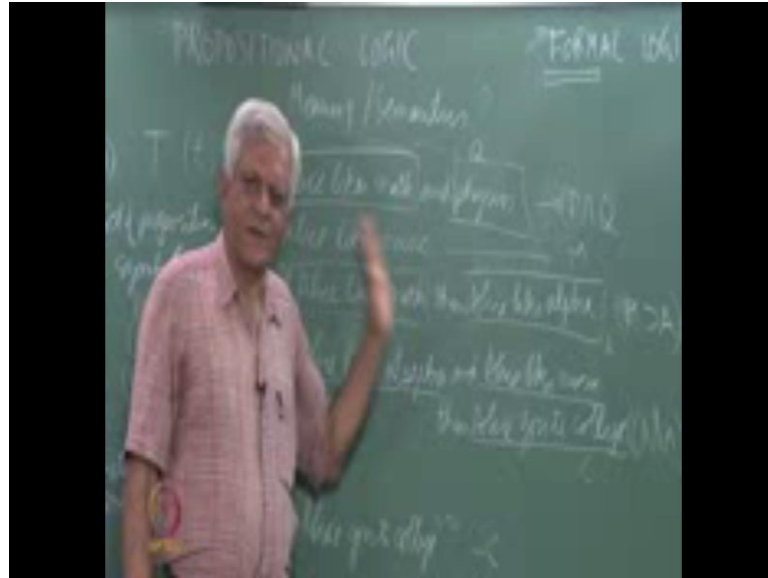
This machine is we want to built is essentially a machinery which first defines this language. And then tries to reach to sentences in this language essentially and a mapping

for to this anywhere will come to that in the movement. So, the set of sentences let us call it S where p stands for propositional logic is defined as follows that this belongs to this set S belongs to this set. So, that of course, you know they stand for something which is always true or and something which is the always false. So, this is the set of propositional symbols let us call this set P . Then we say that if α belongs to P then α belongs to S . So, every propositional symbol is the sentence and then we say that if α β belong to P . So, α β these symbols we will use as variables to stand for any sentence essentially α and β belong to S and α , these are finally, operator symbol. So, of these symbols we will distinguish between binary and unary and we will not talk about higher or operators and we do not need to α . So, this one is unary we often of always you also use the this sign for this α for this sign. We also use sometimes an arrow for this we sometimes use 2 arrows and there are the symbols which we I have not mentioned here, but we can build the language using other symbols.

So, I So, is your familiar that there are sixteen possible binary operates that you can device and these are only one 2 3 4 of them and there are 12 more which so if α β belongs to the set of sentences. And so instead of using this particular symbol let me here use some abstract symbol if says the, this is the binary operators symbol then α well I am. So, set I just defending the language here. So, for example, α if p and q are to propositional symbols then p and q belongs to this set and once this belongs to the set we can also say p and q or q belongs to the set.

So, essentially we can combined this any number of time and we can create sentences which are which are really long. So, and which infinites so the sentences in fact, we can create even if the set of propositional symbols is finite essentially likewise we have for the unary symbol. If α belongs to S and we just use this symbol in a unary operator then so essentially now we have defined we started with the central propositional symbols. And we have augmented this with more sentences and the sentences are using the operators. So, the or the connective essentially and the bracket symbols and so we can get in infinite set of symbol essentially. So, this is the syntax, what is the meaning of these sentences propositional logic is not concern with meaning. So the meaning is given by us essentially.

(Refer Slide Time: 11:11)

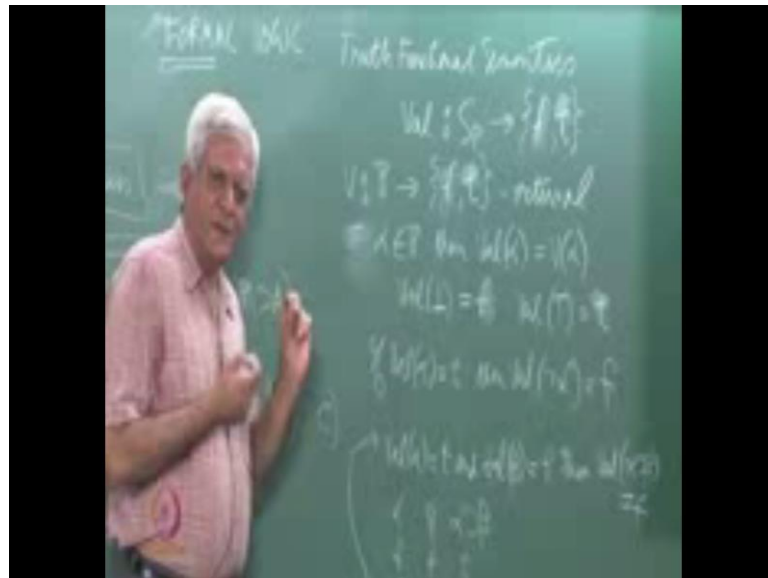


So, I can say that a if have set of sentences like Alice likes a math and physics and I call this sentence p . That is only for my own sake essentially I want to let say reason about Alice and I create a set of sentences is about Alice. So, I say I like math's and physics a Alice likes music. If Alice likes math then Alice likes algebra if Alice likes algebra and music. So, I should really say I will just likes music then Alice goes to college that is say I have this set of sentences. So, always thank to give you flavor of anyway familiar with this the kind of things we can do with professional logic and from this we want to ask the question that is this sentence. So is this sentence can be cannot be informed not essentially. So, I can now encode this problem in propositional logic and I can say a that Alice like math's is p and Alice likes physics is q .

Then I can call the sentences p and q because Alice like math's and Alice likes physics this I can call as m . And then I have a statement if Alice like math then Alice likes algebra. So, this sentence I can call as a let us say and this hole sentence I will represents and implies a . And this sentence says Alice like algebra Alice like music o implies a if Alice likes music and then Alice goes to college So, this let us call it is c So, this sentence I can write as a and m implies c why did I put brackets around here? But I am not doing it, because when if clear then we do not often put brackets and this is the sentence c and we are asking with the this sentences true or not. So, I am saying here that

m means that Alice likes music but that is only between us as for as the logical machinery is concerned it is not concern with this at all. So, like we set logic as formal essentially.

(Refer Slide Time: 14:55)



We say logic is it only depends upon the from so essentially what we have going to ask is a if a assume this is true; this is true; this is true; this is true. Then is this necessarily true number we said logic talks about for deduction essentially, but suddenly you have started talking about true statements for let us talk about truth first when we says semantics. We essentially have to kinds of well should I safely worst do it values it is a, what does the sentence mean? What does p mean? What does q mean? What does n mean? Now, we are saying that logic is not concern with that essentially. The other thing is truth; truth values or some people in the tram truth functional semantics. And what we have saying here is that there is a mapping can which goes from this site of sentence sp to in site. We want to define this mapping and that those 2 symbols we set that between us will assume this stand for false and truth essentially. So, what are what is the mapping a looking for we are looking for a mapping which takes an import any sentence in my language. And maps it to one of those 2 symbols which means it tells me, but that is again between us where there the sentences true or false essentially.

So, we have to define this mapping. So, we start with the function v which maps this at p to the set of essentially lets external what do mean by this? Somebody is telling me what is true and what is not true that I am saying that Alice likes music's. So, I am give mapping and to clue and mapping p to true q to true. And so in general given the mapping of the professional sentences we can derive a mapping or so this is general version we will come to specifics later sometimes you have uncertain solid.

So, I am I it is say p is true or r it is true in which a off course, a do not I am not given the mapping, but I am given the mapping for pm also. We will come to that a bit later first you all the defined that if you are given the mapping a which we call we which takes you from the set of propositional symbols to this. Then can be defined this value function and that we can do as follows. If help a belongs to p then $val\ alpha$ is equal to, because somebody is given a started it maps to he are very talking about to tables from this point onwards. Then we always define val is equal to this thing and val they will little bit of over loading happening here ideally at at should use different symbols essentially.

So, I so let the say we defined it to 2 now, what and really saying is that this bottom always bottom symbol always maps to the truth fell you false. So, let us call this is the set of truth values and here working with the binary true valued logics. So, that this truth values which one we call false and one we call true and bottom always must be false and top always maps to T with stands for true essentially. And then if $val\ alpha$ is equal to true then $val\ not\ alpha$ is equal to form. So, I right rules like this if $alpha$ maps true, and $alpha$ is the sentence in my language there. Then the another sentence in which I prefix $alpha$ with this or unary operator for we will off course, between us read as not $alpha$ then not $alpha$ maps to f essentially and like this you can right other rules. So, for example, we can say if well $alpha$ you control true and $val\ beta$ is equal to false then $val\ alpha$ implies $beta$ is equal to false. So, we are familiar with this. So, we will not spend our time righting this essentially you can convert these kind of statements into a table. So, for example, a $alpha\ beta\ alpha$ and and place beet.

So, false false true false true well I do not know where there you can make out between my maybe I should out chosen 0 and one that anyway you not the truth table for implication. So, this is the truth table for implication these are the simplification if I can

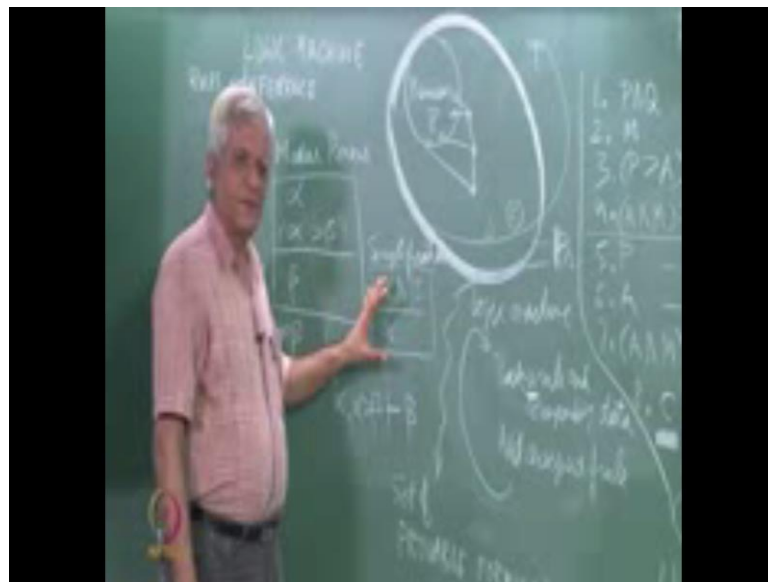
like this for each low I can write a sentence like this. So, this is this stands for the a third row here this statement call this third row. So, I considered the 2 table or I can rights statement like distance so on, but we are familiar with this as to how to defined this semantics of how to lift the semantics of smile sentences to compound sentences when they are made of using operators. And we have this logic rules that we all familiar with essentially. So, what have it done? We offset that we have define the language of professional logic, we have define what is the set of sentences which make up this proportional logic. And then we have define the mapping which tells us that if he knew the truth values of the constituent sentences.

How can we lift those truth values to more compounds in this is and we have rules for doing that which means somebody gives us this valuation which a that these are the 5 atomic sentences a proportional's symbols. And they map to true false true false respectively then for this entire set of infinite formalize. I have a mechanism for determining for there they map to true of where there they false and that I can do by using true tables or something like that. So, where, there the logic machinery come and here. So, logic machinery comes in the logic machinery as we set is deals only with the form it does not deal with content. It does not with semantics it is it does not deal with meaning what we have interested in is can we table of a syntactic machinery which will do that mapping for us essentially. Instead of having to apply these rules repeatedly you can keep applying the rules if some formula l for and some formula beta for what is the truthfully of this. It depends if both of them a truth then this compel formula is true and you can breakdown essentially. Instead of doing that in staff having to construct the true table can we have an alternate mechanism for doing that?

So, to come to that let a say that somebody as given a this function. We with basically saves which professions of true and which are falls from that we can come compute this function val which says the something for compound sentences. And from that we can defined this said t is the set of 5 sentences. So, of all the sentences which I can construct did my language I the semantics tells me which have true and which have falls, but I have own and either waves of reaching those sentences. I do not know to apply these schools I do not want to construct to tables quite, because the side of the 2 tables then grow to be very large is essentially I have 2 constituent here alpha beta. And I have 4 set

4 rose it may true table if I have a sentence like alpha and beta implies or delta implies theta and I have 5 variables and I have 2 rest of 5 rose in may true table. So, the side of the truth table goes exponentially with the number propositional variables set I use in my sentence essentially. So, I do not know want to construct a true table what do we do we defined in set the notion of a rule of inference. So, let me of this part out. So, what are we interested it we I interested it we a interested in reaching those wet set of true sentences this.

(Refer Slide Time: 27:00)



And what do I mean by that if I somebody else given with the promises which is this set p that at talked about the semantics defines is set t devises I assume to be true. So, these assume to be true but from if I assume these to be true then there where hole set which is true. So, for example, I can say a and or not q is this true or not this is the hole they there infinite. So, the sentences I can construct using these symbols and I can ask the question of the self them were there that is true or not from that I can collect the true sentences. And I have this set true essentially now a are seen that he say the mechanism of reaching this set again truth true or different means now you talking about the logic machines. Or you can say and inference this use is something call rules of inference. So, what does the rule of inference? Rule of inference is a pattern and the most popular on that no thought the a familiar with is called nodes ponens and modus ponens says that if he can see a

sentence alpha. And the sentence belongs to the set of sentences that we have and if you can see a sentence alpha implies beta.

Then we can add in sentence beta what do I mean by this that if I have a set of sentences given to me to set of premises. For example, our data base of sentences if you want to use at a then I can keep adding new sentences to the data base essentially. And how do I do that? I do it by this pattern matching mechanism which says that if I have alpha and if I have alpha implied beta then I can add beta essentially. So, this is rule of inference there have many others you must see familiar with. So, one rule is called simplification we should will need for this it says if you can see a sentence of this kind then you can right the sentence of this kind essentially alpha. So, notice that once you have rules like simplification we do not feel complete for somebody to us the set of all atomic valuations for all atomic proposition. Somebody is given a of p and q from this by applying this we can always add q to or p to the. Then you say we can add this say sentence beta or we can add this new sentence alpha to our data base. We also use the turn that we can derive beta essentially or we also use the turn prove beta. So, if alpha is given to us and alpha implies beta is given to us when we can say we can derive beta or we can prove that beta is true essentially.

So, the notion of a proof is distinct from the notion of truth that we have talking about there the notion of truth is based on the semantics of the operators that we have defined what does it? What does and mean? What does all mean? what does the implication mean? What does negation mean? And when he says mean basically we are talking about the truth table essentially how does and combine the truth values of its components to give you truth value. That is the meaning of those operators and truth the notion of truth is based on the notion of semantics are those operators essentially. So, notion of proof or derivation is purely syntactic notion it is purely based on pattern matching. It says if you can see is such a pattern and if you can see is such a pattern. Then you can derive this pattern it is nothing to do at least on the surface which the notion of prove essentially ideally. Of course, we would not be in a machine like this a syntactic machine like this which is producing something which is not interested. And we are interested in true statement we want to know what is true even some premises not else is true. So, theory on proving is the name that we use. So, this is essentially So, let us see what happening

here before we look at the machine if further what is been given to us says p and q for
lets label number this now true is a .

And you are familiar with this process p implies a or a and m implies c here the fourth
facts is a terms sometime we used given to us that this 4 sentences at true. Then a
conclusion that we have see is see true or not will Alice go to college or not essentially,
but we have not looking at the meaning of this sentences. Any more we have not even
looking at the truth values of the sentences. We are asking that will this machinery that
we have we have talked about tools of inferences. There are more rules that we can
talked about and if necessary he will introduce the can we have this machinery produce
this formula c or derive this formula c a prove this formula c . So, what is this machinery?
This logic machine essentially says pick our rule and corresponding data by data had be
mean the sentences which are already there. So, for example, these sentences are already
given to us essentially and if some rule is applicable data take that rule. And take that
data and add consequent of rule of a there as the matching component at a stating
explicitly here what is the consequent? The consequent is return below the line here
return below the line here we also use another way of signifying this rule.

We say α implies β there is another way of righting this rule it is say that given
 α and given α β you can derive β . So, this symbol is used for proving or
deriving thing and this is stating a rule of inference modus porous is say that if I give this
to you can add that corresponding to this is corresponding is also the notion of truth.
Then we are asking if this is true and this is this is true this is true then is it necessary that
 c is true. We would like this as follows p and q and n and p implies a and this one a and n
implies c we have. So, corresponding of sentences we have the social of truth value and
correspondence notion of a truth value. We have saying that these 4 sentence entail c . So,
the premises is what are the premises is 4 things even to us premises entail the sentence c
and that more formally we like this these first premise, second premise, third premise,
fourth premise. And that this similar symbol, but it 2 lines is for entailment an entailment
says that if these are true than that is necessarily true that you cannot the that true that is
notion of entailment. So, that is set of truths 2 sentence that we talk about an entailments
basically asked the question thus is see belong to the set of 2 sentence or not those this
premises entail c the question. We are asking here is can you derive see from the

premises could derive we see are use this rules of inference for entailment; we have to use the truth stable for the operators they are quite this things.

So, I am so fact that they are different. So, logic machinery basically says packable and corresponding data and add consequent of the rule. And we put this essentially and what this will produce is another set of formulize which we will call is what is the call this that is let me, because I use the symbol p earlier. Let us say this is p r something some symbol what is the set? This is a set of I should write it here set of provable formulas. So, I have in adamantly sort of switch to the terminology of formulas but essentially a formula is a sentence in my language essentially a well form formula sp . It is a short for well form formula anything that I can construct using the sentence that I have defines is a formula. And this set p r is a set of all those formulas which are provable given the set of premises and the set of rules of influences that are given to me essentially. So, the question we are asking is the c belong to this set or not and instead of writing this loop which will keep going indefinitely, because the set is actually infinite. We can put a the termination criteria that if c if the consequent equal to c then stop essentially we can just put in the condition that.

So, that means we are writing the program to generate a proof for only c and not worrying about other provable formulas essentially. So, there a small variation essentially so how do we go about this? This is given to us we can add p here why because there is a role called simplification which says if you have α and β and α matches p and β matches q . Therefore, I can produce p here by simplification there another rule. So, I can use now modus ponens because I have p and I have p implies a I can add a remember modus ponens say if you have α , α is p and α implies β is p implies. Therefore, I can β becomes a and I can add a there is a another rule which says that if I have 2 things. Then the conjunction of those 2 things can be added which is called addition, but I can write a m , because I have a and I have m I can write a and m I can add this is another rule of influence which I have not written here explicitly and then again I can use modus ponens. So, ideally I should say how would I get this? I got it from one and simplification how did I get this? I got this from 5 and 3 and modus ponens hence so on essentially. So, you are familiar with this process having studied geometry and algebra in school.

So, now I have got a m I got this by addition of a and m and then the last thing is c , because I have this a and m and a and m matches alpha and a and m implies c matches alpha implies beta. So, c matches beta. So, I can use modus ponens. So, this is a syntactic machinery of arriving at proof of c which does not look at the meaning of the operators. It does not understand what m stands for? What does implication stand for? It only knows all it has is a set of rules of inference which says if you see this pattern. And you see this pattern you can add a new formula to your data base and we can put this to a cycle and say can we add c . So, at this point we have added c . So, we can say yes c is provable in Γ . So, that is a notion of proof notion of provability that is syntactic machinery we are after, because it is easy to write pattern matching algorithms and in this case the patterns are exact. So, you know matching is not even hard and you can keep looking for rules and matching data and keep producing new data.

So, you can imagine the forwards search algorithm which says this is what is given to me this is a rule available to me from this. And this I can infer from this and this I can infer this from this and something else I can infer this. So, I can move forward and that c is somewhere. So, if I can reach this formula c then I have stopped. So, the power state space search that we talked about is essentially being done here in. So, notice that we have said pick a rule we have not said which rule that of course, is a very tricky question you can go off in some direction and infer meaningless things. So, that is obviously the thing that we need to tackle next essentially. So, what have we done? We have defined the language of propositional logic this in fact then the semantics the truth function semantics which says that given a sentence. We can arrive at its truth value using the truth table approach then we talked about this machinery of making inferences which which does not look at semantics. But it only looks at the syntax, it is a syntactic machinery and basically adds new formulas to that we can sort of build a machinery to produce the formula.

So, is this machinery, good machinery or not? This is the question which is asked by all logicians. So, what do we have? We have a logic machinery is this logic machinery doing what we want us want it to do? So, there are 2 notions that we use here one is the notion of soundness. The notion of soundness says that my machinery will only produce true formulas we can express this as follows if. So, let say L is a language

give to us or set of premises given to us if Γ entail α then Γ sorry if Γ derives α then Γ entails α notion, so soundness is that my logic machinery will produce only true formula if it produces a formula then that formula must be true. Of course, trivially if your logic machinery does not produce any formulas that will be a sound machine. But that of course, is not a very good machinery as a properties completeness that we have already spoken about is that if Γ entails α . Then Γ completeness says that if a formula is true then there is a derivation for that formula in my logic machinery that if I had a non deterministic machine which knew which rule to pick and which data to apply to it will produce a derivation essentially.

Of course, in the absence of having non determinism we have to use search and we will sort of focus on that as we go along tie to soundness is a notion which is basically is derived from soundness. But it is useful to talk about is it is consistency, consistency says Γ logic is consistent if you can derive either α or negation of α but never both essentially. So, says that Γ derives α and Γ must use the exclusive or Γ derives negation of α . So, this stands for exclusive which is not a symbol I have defined here but we can define it in terms of this. So, consistency is basically de tied up to soundness is if you just think about this a little bit only one of those things will be true and soundness says that I can prove only those true 2 things essentially if I prove something it must be true. So, you cannot have α and not α being derived. So, which is basically the other side of soundness, but very often we talk about consistency soundness, completeness and consistency essentially. So when we begin the next class, we will talk about these properties. We will ask the question when is my system going to be sound why is this rule sound how do I say that this rule is sound.

So, we will ask this question, we will also talk about completeness a little bit, but we will not prove it is completeness properties the proofs are simply little bit longer. And then we will move on to higher order logics essentially may I will give you an example of logical systems which were know as axiomatic systems. So, last 200 years mathematician have been talking about building axiomatic system. So, they have been saying if you use this set of operators and this rule of then your logic machinery is complete essentially. So, I will just give you an example you would not prove their completeness but we can talk about it essentially. So, in the next class, we will begin with soundness and completeness

and then move on to logic.

Eventually we will also focus on this machinery what really this algorithm should be essentially this seems to wage and it needs all this power of non determinism to work. We need something more efficient to be able to derive the proofs essentially observe that you know of course, we are not talking about first sort of logic yes. But in first sort of logic you would have this problem you could say if a number is odd then it is successor is even essentially. And if somebody tells you 5 is odd then you will say go into a direction saying 6 is even, 7 is odd, 8 is of course, you have to have the other rule also that if a number is even and it is successor is odd. Then you will just keep making statements 5 is odd, 6 is even, 7 is odd, 8 is even, 9 is odd and you will never come out that loop essentially. So, obviously that is like depth first search going into an infinite branch that is danger lucks even in this algorithm here and we have to look at some of those algorithm issues. So, we will stop here.