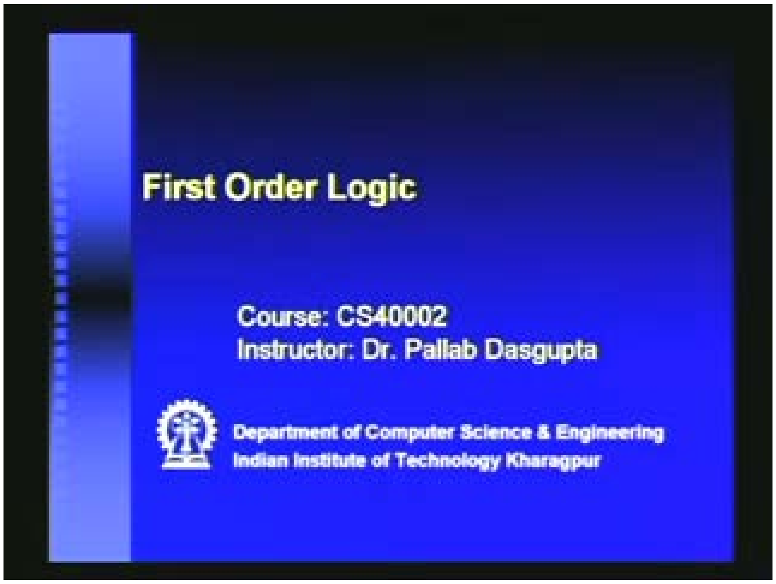
Artificial Intelligence
Prof. P. Dasgupta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture- 9
First Order Logic

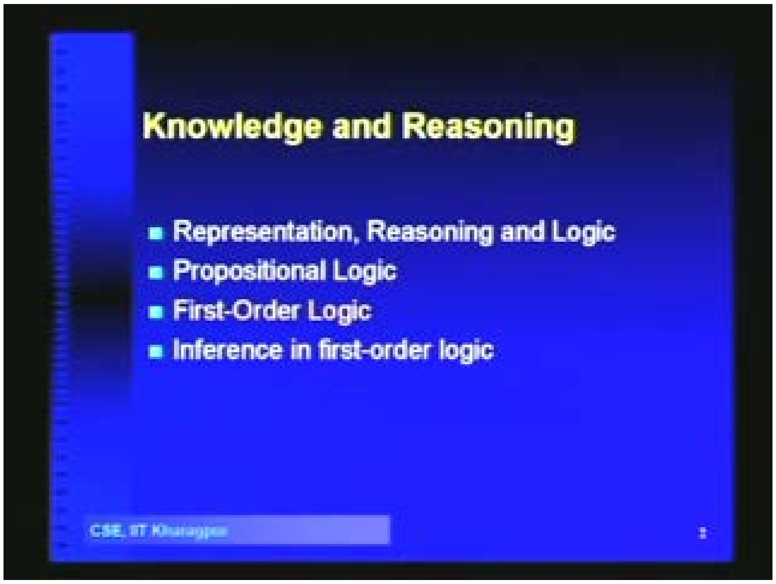
In the last class, we had seen we have studied propositional logic and we had seen some basic ways of doing reasoning with propositional logic. Today, we will extend the logic further and move onwards to what is called first order logic.

(Refer Slide Time: 00:01:12)

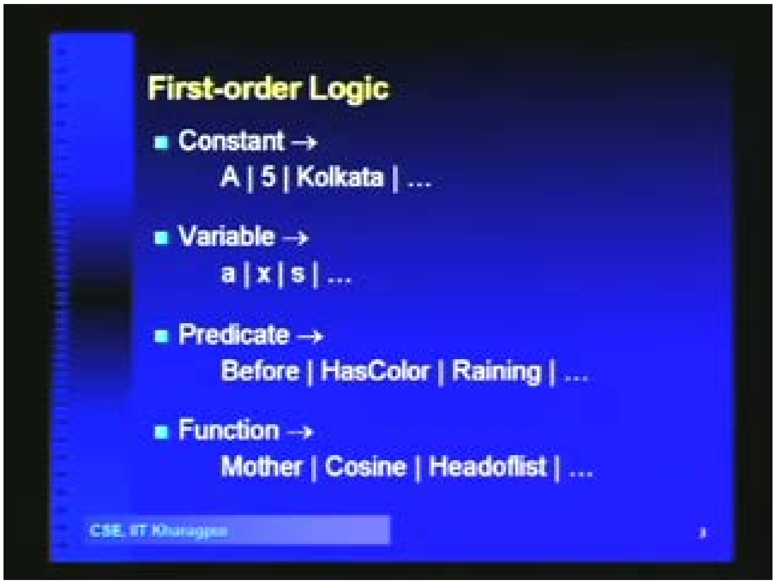


In first order logic, the main so what we have done so far is, we have seen how to represent a problem in logic. We have seen propositional logic; we will now see first order logic, and then, we will finally study how we can do inference in first order logic. The difference between first order logic and propositional logic is in the existence of predicates.

(Refer Slide Time: 00:01:26)

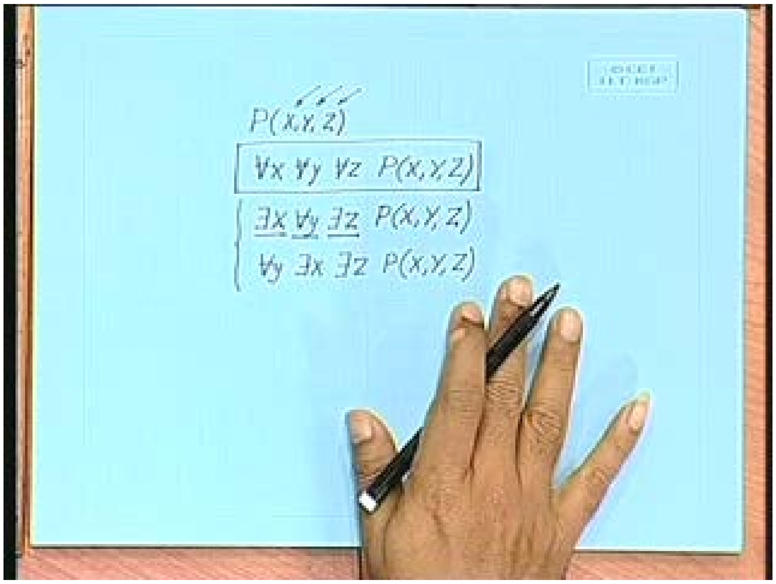


(Refer Slide Time: 00:01:35)



If you recollect, that in propositional logic, we had propositions which could take values of either 0 or 1. So, anything that we want to represent, is in terms of propositions which have a Boolean value- true or false- right? Now, unlike that, in first order logic, what we are going to have is- we will have things called predicates and predicates can have 1 or more arguments.

(Refer Slide Time: 00:06:12)



You can have a predicate which has arguments x, y and z, right? There is no truth value that we associate with this predicate as such, but we can always say, that for all x, for all y, for all z, I have p(x,y,z). Then, this whole thing has a truth value. So, this whole thing can be true or false. If it is the case, that for all values that x can take, for all values y can take, for all values z can take, this is true; then, this whole thing has is true. In other words, when- these are like variables; x, y, z, are variables. If you assign some value to them, now, these values need not to be numerical; it is some instantiation that we do to this variables; we assign them some value from the domain of these variables, and if we know or if the knowledge base already contains the fact, that for those variables, this predicate is true, then, we say that for that value, for for that value of x, for that value of y and for that value of z, p(x,y,z) is true, right?

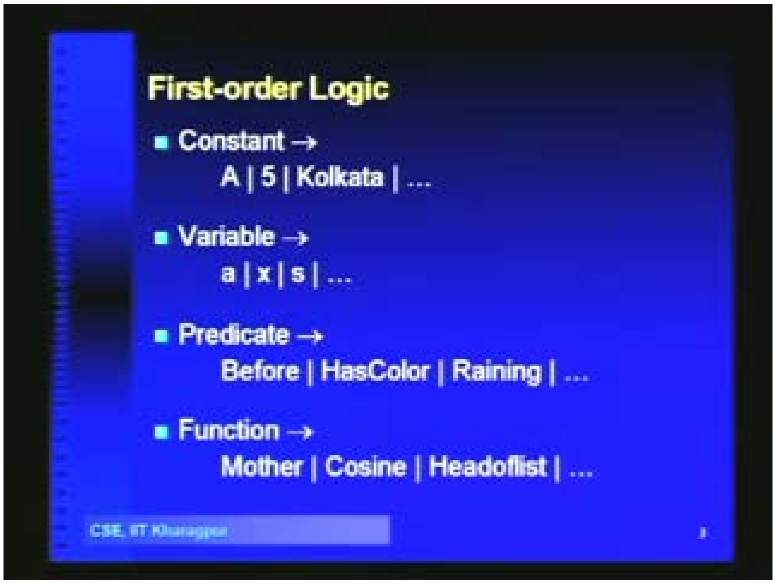
Similarly, you can have- there exists x, for all y; there exists z, p(x,y,z). That means that there exists some value of x, such that no matter how you pick your y, there will always be some z, which is able to satisfy p(x,y,z). (Student speaking). p(x,y,z) has a truth value, when x y and z are instantiated to some values, right? So, p of Tom, Dick, and Harry can have a truth value. But if you just leave them as person x, y, z, without instantiating them to any value, then, we cannot say whether whether that is true or not. But we we can have these quantifiers and we have 2 types of quantifiers, namely, there exists and for all, I think you are familiar with these things from discrete structure.

This means that there exists some value of x for all, such that for all values of y, there exists some value of z, which satisfies p(x,y,z). And it is not difficult to see, that the positions of these for alls and there exists y, they are not commutative. So it is this is not the same as writing for all y, there exists x, there exists z, p(x,y,z). Though it may be cases, where 1 implies the other, but these 2 are certainly not equivalent, right? So, we cannot actually move the quantifiers across each other; they will have different meanings.

And we will shortly describe a set of things that we can do and we cannot do, on this kind of logic.

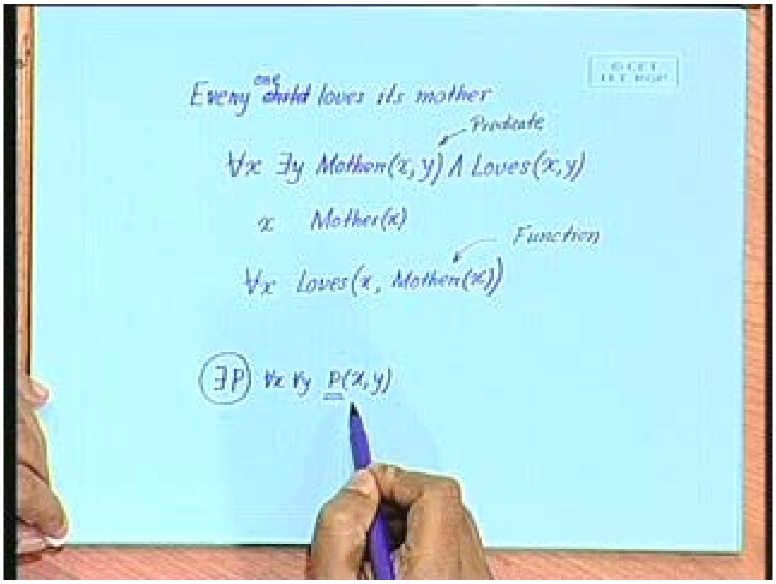
But first thing that we have to understand is, how do we actually use this kind of predicates to model actual scenarios? How do we use this kind of predicates to model actual scenarios? And then we will come to the kinds of reasoning that we can do with that logic. Firstly, let us study briefly, the syntax of the logic. The syntax of the logic is as follows: we have constants like a, 5, Kolkata, etc. We can have variables, which are usually will be written in short in in small capital in small letters. We will have predicates like, before, hascolor and raining, and we will have also functions. There is a different betw between what we have in predicates and what we have in functions.

(Refer Slide Time: 00:06:40)



Predicates: once you instantiate the arguments of the predicates, you have a true or false value, whereas functions can have any value. In case of predicate logic, in case of first order logic, your functions can return you values which are numeric, nonnumeric, characters, strings, whatever. So that they just have a return value, okay? For example, if you want to say that every child loves its mother, right, then we can write in 2 different ways. 1 is, okay, let us see, we want to express the fact that- or better still, let us say we want- instead of saying every child, let us say everyone. There are 2 ways in which we can write this: 1 is, we can say, that for all x, there exists a y, such that mother, which means that y is a mother of x. Right? This is 1 way of writing. We can also use a function called mother, where given x, mother x, will denote the mother of x. So, if we treat this as a function, then we will write- for all x loves x. Now, see the difference in representation.

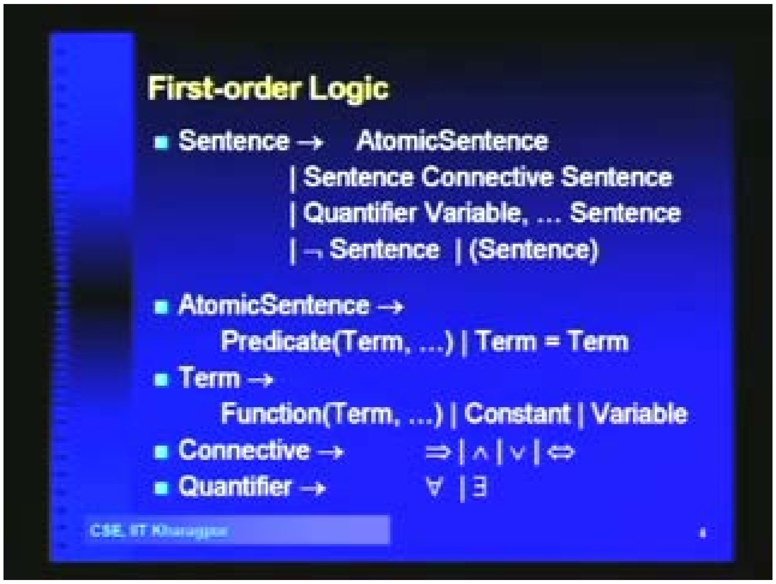
(Refer Slide Time: 00:15:30)



Here, in this case, mother was a predicate with 2 arguments; here, it was a predicate with 2 arguments; here, it is a function. So, here, the value, given the value of x and y, mother x y is true or false. Here, the given the value of x, mother x gives us another- returns as a value which we can use within the predicate. In first order logic, these functions will never appear, except as arguments to predicates. These functions will not appear outside any predicates. If you look at the syntax of the logic, we will have predicates, and within predicates, some of the arguments can use functions. Let us continue with the syntax of first order logic.

We can have different kinds of sentences; the first kind is atomic sentence, where we have a predicate. This is a basic form of a sentence predicate, which has a set of terms. The arguments of the predicates are terms, and a term can be a function of other terms recursively, or it can be a constant, or it can be a variable. It can be x, which is a variable; it can be Tom, Dick or Harry, which are constants; and it can be a function, like we had mother of x.

(Refer Slide Time: 00:11:00)



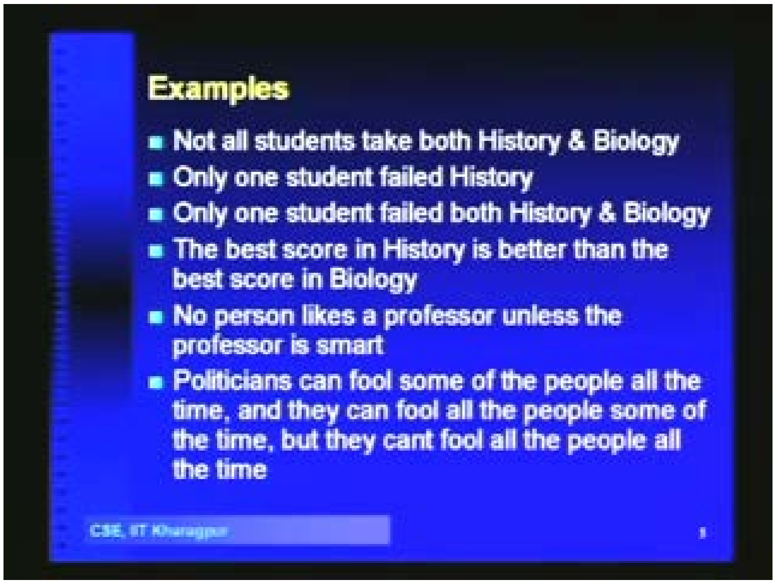
so and these These terms are the argu that form the arguments of the predicate. We also have term equal to term, as a special kind of predicate. This is a special predicate; this is a predicate which finds out whether these 2 are equal. If you have this check for equality, we shall see shortly that this is a very important predicate; it is very important to have this predicate. (Student speaking). No, see, what we want to check is, whether, say, I want to find out whether 2 variables have the same value or not. So, I can say x equal to y; so, you can treat this as a predicate, which says, equal, within brackets, x comma y. It is another predicate, but we cannot actually enumerate in our knowledge base, the facts associated with this equally. That would be too large.

You will have to say, this is equal to this, this is equal to this, like that. Checking for the equality is something which the reasoning system will do internally. We just have this special predicate, which tests for equality between 2 terms, and returns true if they are equal and returns false otherwise. We will As we progress, we will you will see why this special predicate becomes necessary. Then, in conjunction with negation, which we had, oh, I missed out the negation. Oh, no, I have it here. Right, right. So, here, you have see you have the negation here, so, in conjunction with a negation, you can actually also express term not equal to term. That is going to be not of term equal to term.

The other kinds of sentences that you have is, 2 sentences with a connective in between and a connective can be implication, or, and, and both ways implication, or or equivalent. And then, we can have quantifier variable, which means- there exist x, like that, and you can have several of these, followed by a sentence, right? Then, you can have not sentence and we can also have compound sentences. And we have just 2 types of quantifiers- for all, on their exists. But before we proceed further with first order logic, let us understand that why do we call this first order? What would be there in second order logic which is not there in first order?

Here, what we are allowing is, we are allowing the quantification over variables. We are allowing quantification over variables x, y, etc. We are not allowing quantification over predicates; we are not saying that there exists a predicate p, such that for all x, for all y, p x y. See, this is something which is trying to quantify the predicate itself. That is more complex, rather than having well defined predicates and quantification only over variables. In this way, you can see that we can have a hierarchy of logics: first order, second order and higher order logics. We are not going to go into details of those, but when we study the complexity of first order logic, we will get a glimpse of how difficult it is going to be to reason with higher order levels. But there are actually several tools which enable you to do reasoning with higher order level. Now, let us do 1 thing; let us have a look at examples and try to formulate these examples in first order logic.

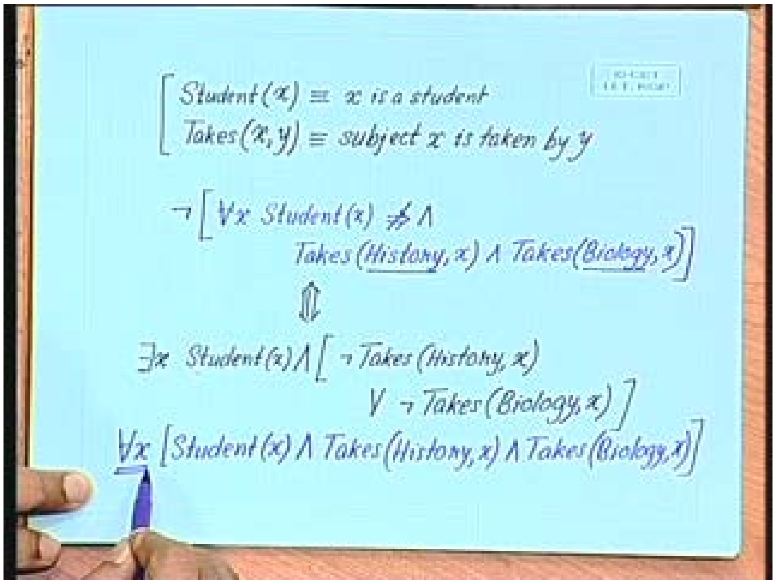
(Refer Slide Time: 00:50:55)



The first 1 says, not all students take both history and biology. The first step in solving these kind or modeling these kind of sentences in first order logic is, to decide on what is the set of predicates that we are going to have. First thing is that, it is not that, not everybody takes history and biology; not all students. There will be some predicate, which tells us whether a given x is a student. First predicate that we will define is, student x which says- right, then, we can have a predicate which says who takes what subject; takes x, y, where- which says subject x is taken by y.

(Student speaking). No, no, no. See, these are all variables; it depends on what you instantiate them with; these are all variables, right? I actually purposefully used x here, to create this confusion. This is this is another variable- this variable and this variable are not the same. (Student speaking). Why? x is just a variable. It is a place holder, so if you instantiate x with graph theory; then, if you try student graph theory, it will fail. If you instantiate x with Anshuman, and you have student Anshuman, then, that will be a fact, that you already have in your knowledge base, so it will match, agreed?

(Refer Slide Time: 00:34:04)



So, it depends on what you are instantiating x with. It is these These things are like relations; they are you know some of the facts and you are trying to use them to de to derive other facts out of them, right? Now, if we have these 2 predicates and what we wanted it to express was, that not all students take both history and biology. There are there are many ways to write this statement: 1 is that we first try to write that all students take history and biology, and then negate that. If we want to write all students takes history and biology, then we will write- for all x- student- x implies: takes and and then, put a negation outside the whole thing.

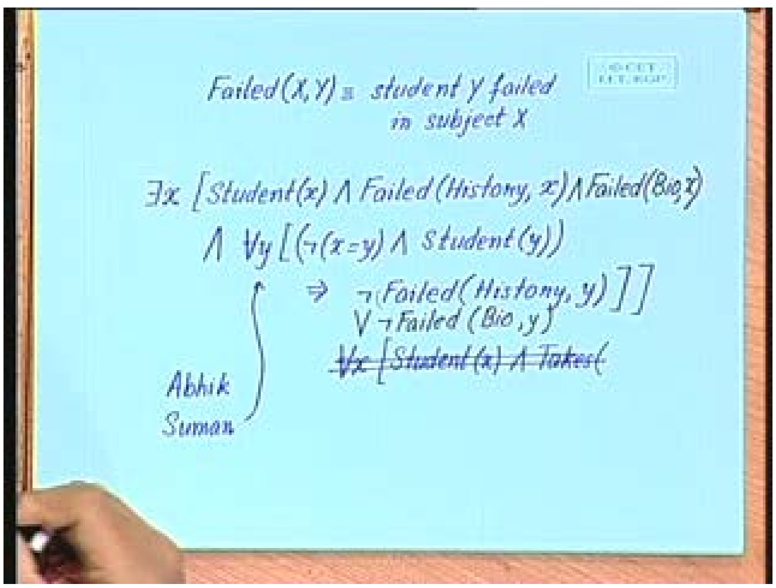
The thing inside the negation says that for all x student x, for all x, if x is a student, then x takes history and x takes biology. But what we wanted to express was that, this is not the case that not all students take both history and biology. Now, note that in this, these this history and biology, these are constants. x is the variable that we have and the scope of the x is for this whole formula here. Now, we can also express this same statement in another form: we can say that there exists a student, such that the student does not take history or the student does not take biology. If you want to write that, then we will say that there exists x- student x- and either not takes, or. Actually, these 2 statements are equivalent. And how do we have this equivalence?

If we push the negation in, then the for exists x, becomes there exists x, right, and then, when you have the negation here, recall that this is of the form a implies b, so, that is not a or b, and then, if you use De Morgan’s to push the negation in; we have student x and then, we have this not takes history x or this and or becomes this and becomes or and we have this, okay? Yes. (Student speaking). In the first statement, and no, instead of implication, I mean, how will you connect this? If you if you do not have this implication, you have to have an and or or; some connective to connect this. No, I did not possibly I did not get your question; I mean, you wanted to know that suppose, we do not keep this implication here, okay?

Let us let us see- suppose, instead of implication, if we had and here, then we have a problem, because then, every x has to be a student, but then every x in the universe is not a student, right? I can always find out some value to instantiate x, such that that value is not a student. Suppose I instantiate x to table, and then, in your knowledge base, you will not find student table. So then, what will happen is, that this thing will fail. Are you getting me? Instead, what we are trying to say is that, if f x is a student, then, the student takes history or takes biology. If x is not a student, we do not care. That is why we have this implication.

In the second case, because we are looking for existential quantification, so, we are saying that okay, there exists some student, such that the student either does not take history or does not take biology. Clear? Let us look at the second example: only 1 student failed history. So, now, we will have another additional predicate, which says failed …y, where this says that student y failed in …, alright?

(Refer Slide Time: 00:37:28)



What we want to express is, only 1 student failed history. How are we going to express that? First thing that we need to express is that there is somebody who failed history. There is 1 who failed history, and then, we will try to also express that only 1 failed history. So, we say that there exists x and we have student x, there is some student x, right? And failed history, x, right? And now, we have to express, that for all the other students, they did not fail in history, to express that only 1 student failed in history. We have said, there exists x who have failed in history and then for all the other ones- they were they passed history, right? So, we sill say that for all y, and then not, x equal to y, and student y- (Student speaking)- wait, we are coming to that, and student y, this whole thing implies- now let us see this once again. We have this This tells us that for all y, x not equal to y, and student y, so, if y is a student other than this x, then not failed history y, then y did not fail in history.

So, if y is a student other than x, then y did not fail in history; is this alright? (Student speaking). Let us let us see- here, we have this- there exists x, right? We are saying that there exists some x, who is a student and who failed in history, because we wanted to specify that only 1 student failed history. Now, that means that 0 students failed history, should be eliminated. (Student speaking). No. No. No, no, no, no. Then, for all y, you will not have this. See, if there is more than 1 person who failed in history, then this is going to fail, because suppose, let us say, that say Abhik and Suman failed in history, right? Then, when you instantiate x to Abhik, right, then, this part is fine.

Abhik is a student; Abhik failed in history, fine. And then, for all y, you have to satisfy this, but for all are y, means that 1 of those y's is Suman, right? And Suman is not equal to Abhik, and Suman is a student, right? Then, it should be a case, that not failed history Suman. But that is a contradiction, because Suman has failed, so therefore, this is going to fail- this whole thing is going to fail, right? So, this whole statement is going to be true only when there is exactly 1 student who has failed history. (Student speaking). No, then this itself will fail. (Student speaking). No, no, see, this is there exists x, this is there exists x, right? So, for all those x's where x is not a student, this will fail, but if you have at least 1 x for which this is satisfied, then you are done, right, and that x has to exist, because you are you know at least 1 student failed in history.

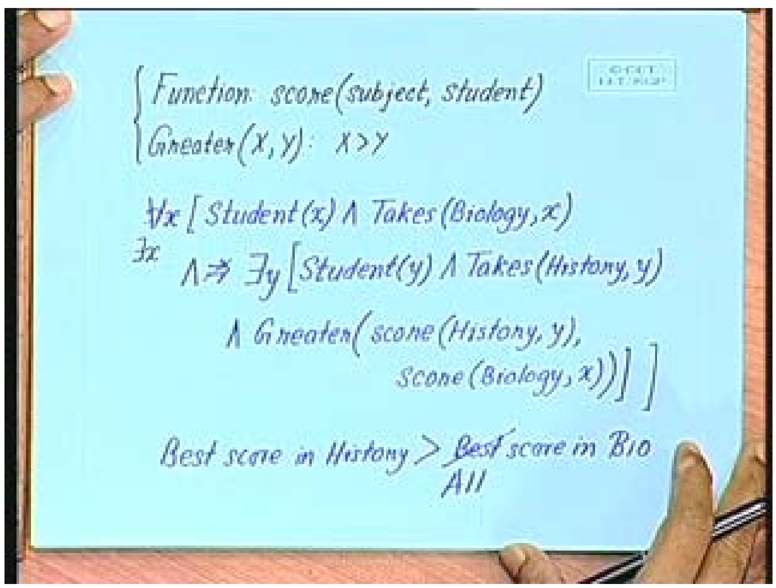
There has to be at least that person and that person will satisfies student x failed history x, right? Had these been for all, then your problem would have arised. But because this is there exists and we are required to guarantee that there exists a student who failed in history, so this is fine. You have to be careful in choosing places where you use and, and where you use implication. When you are looking for existential quantification; when when you are looking for the existence of something, then, usually, you will have scenarios where you have this there exists x such that this and this and this. (Student speaking). Let me take it down. For all x? Yes. Yes. (Student speaking). We do not have biology here at all. For all x, let me go back to the previous example, then we will- it is better to write it there. Yes yes. What is- how do you want it want to write it? For all x? (Student speaking). Yes, this is not going to work.

I will explain why. Biology. See, that is what we have discussed here. Actually, suppose instead of this implication, we have this and. See, then, because we have for all x here, you are making it mandatory that all x has to be student x. (Student speaking). Yes. No, but the whole thing will fail, because this is for all, so if it fails for 1 x, it is going to fail totally. Because you want this to be true for all x, had this been there exist x, then if 1 x is not a student, there can be another x we use as student and can satisfy that. See, this is a for all quantifier, so if it fails for any x, then it fails totally. If you have a there exist quantifier, if it fails for 1 x, you can still have another x which is satisfied. If it fails for all of the x's, then that there exist will fail. Is that clear? Okay.

Let us move into a little more complicated stuff. We now want only 1 student fail both history and biology; only 1 student fail both history and biology. So, how are we going to write this? Let us compare with what we have to do here. I will slightly modify this 1 here to get the new query. No, not exactly, not always. Let us see. Here, we are looking for a student who has failed both history and biology and we want to express that there is no other student who has failed both. So, here it will come with and, right. I am writing bio in short for biology. So, there exists x student x and failed history x and failed bio x, right, and for all y, x not equal to y, and student y means not failed history y or not failed, exactly, or not failed, right?

So, this is the query, for there exists x student x and failed history x and failed bio x and for all y, x not equal to y, student and student y, implies there either that person did not fail history or that person did not fail biology. (Student speaking). Where? Yes, yes, yes. you can have you You mean that you can take this not out and have failed history y and failed bio y? That means just moving these 2 negations out and using the De Morgan again; yes, you can always do that, right? Next example: the best score in history is better than the best score in biology. Now, how do we express score? Yes, we- here is a case, which seems, you know, at first glance, and indeed so it appears that we need a function, right? A function which is going to return the score. Because score is something which is not a its not a true false value; it is some numeric value that is going to get returned .

(Refer Slide Time: 00:46:12)

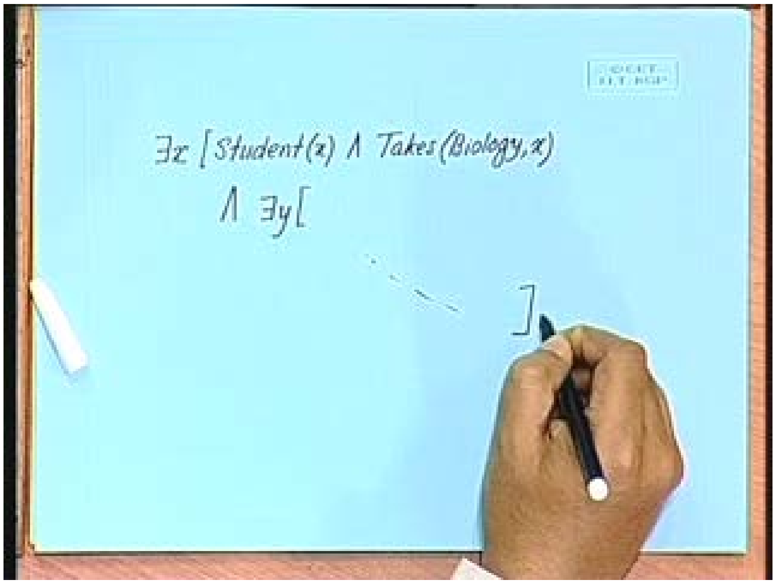


So, we will introduce a function and the function will be as follows: the function will be- score- the first argument of the function will be a subject and the second argument will be the student. So, given the subject and the student, this function will return the score obtained by that student in that subject, right? Now, what we want to express is, the best score in history is better than the best score in biology. (Student speaking). What function? Right. So, we need we need another function which compares 2 scores and tells us which is greater and which is less. Now, that is a predicate, right, it is not a function, because comparison is going to return you a true or false value. Let us add this predicate and this predicate.

If you have these 2, then let us try to see how we can write this. Now, what we want to express is: the best score in history is better than the best score in biology. There are many ways to write this: 1 way to write this is say, that we say, that okay, if we have an x who is a student and who has taken biology, let x be a student who has taken biology. Then there exists a y- a student y- who has taken history and the score of y in history is greater than the score of x in biology. (Student speaking). No, if you can do this for all x, then we are done, right? So, let me write down the first thing. What we want is for all x- student x- and takes biology x. This says, that for all x, who is a student and who has taken biology implies, there must exist some y- student y- and, right? There must exist a student of history. For every student of biology, there must exist a student of history who has scored more, right?

So, greater, and here, we will use the function, score- yes, history- y. And this is from here to here. Let us study this again. What we have expressed is that, for all x, who is a student of x and takes biology, there exists some student y, who is a student of history, and has a score in history, which is greater than the score of x in biology, right? This is to express this thing, that the best score in history best score in history is greater than, yes, best score in biology, so, all scores in biology, best score in bio, and effectively, what this is expressing is all.(Student speaking). No, come again, what how do you want to write this? Okay. Let us see. Let me rewrite this, okay? So, you are saying first term will be, there exists x, then? Student x and takes biology x.

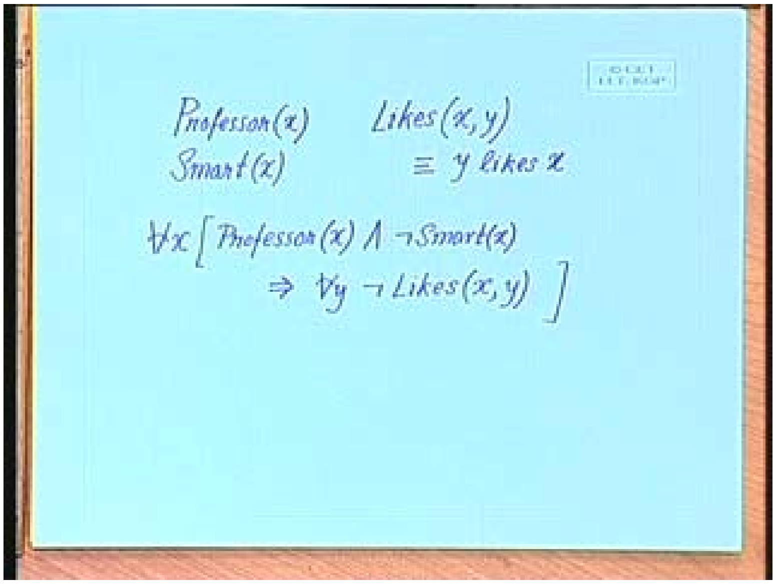
(Refer Slide Time: 00:45:34)



Instead of this implication, you have and, and there exists y, the rest is similar. Let me get it right- what you want here is that, instead of this implication, you want and, right? (Student speaking). Instead of here. So, if you write this, so let me get it right. You want this to be changed to there exists x, and you want to replace this by and, is that right? No, because this is going to get satisfied when any student of history scores more than any student of biology. See, this is going to be satisfied when any student of history scores more than any student of biology. (Student speaking). Yes, right. But, it does not express the intent that the best score in history is better than the better than all scores in biology. Just think it over.

You have to spend some time in digesting the logic and you have to try writing out a few more statements yourself to get the hang of things. And it is good to debate on what are the different ways to write the same property, and whether we can actually write 1 property in many ways.No person likes a professor unless the professor is smart, right? So, try writing this out. No professor no person likes a professor unless the professor is smart. Instead of no person, just make it no one. No 1 likes a professor unless the professor is smart. When you say no one, then you would just will not have to write that person, yes? Right.

(Refer Slide Time: 00:49:36)



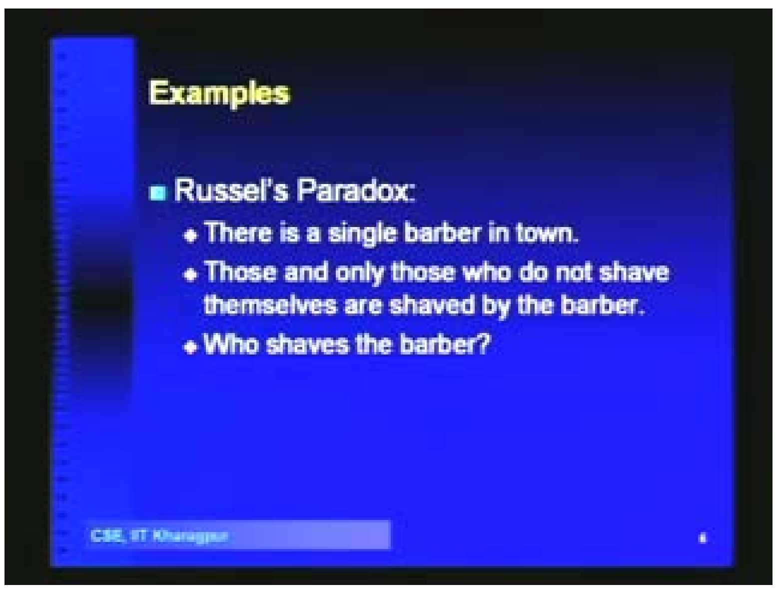
How are we going to write this? We will have predicate professor- professor x, and we will also have a predicate which says likes x y. Let us clarify this- this says that y likes x; it could be x likes y also, so, you have to specify that what is the semantics of your predicate. Then, for all x, right, we will need also, a predicate called smart of x, which says x is smart. Professor x says x is professor is smart x says x is smart, right? For all x, professor x and not smart x, implies for all y. (Student speaking). Yes, this is fine with us.

If x is not a professor, then we do not care whether y likes x or not. If x is not a professor, then this thing is going to become vacuously true; this is going to become vacuously true. (Student speaking). If you remove the for all x, and just leave it like this, we have to define some quantification on x, right? Otherwise, it remains ambiguous. It remains ambiguous whether x is quantified, I mean, whether you want this to hold for all, or you want some x for which you want this, right? Unless you specify this, you have not completely specified the statement. But later on, we will see, that in that, default we will use for knowledge basis is for all x; if you do not specify anything else, we will use that, okay?

Look at this last one- please take this down. This is these statements are actually from the book of Russell and Norvig. These are this is 1 of the exercises given in that book. Politicians can fool some of the people all the time, and they can fool all of the people some of the time, but they cannot fool all the people all the time. Please take this down- politicians can fool some of the people all the time and they can fool all the people some of the time, but they cannot fool all the people all the time. Fair enough. Try to write this down in first order logic. Assume the proper predicates, etc.

We will conclude this lecture with 1 very famous paradox, which is called Russell's paradox. And this is a very famous paradox, which says that there is a single barber in town. Those and only those who do not shave themselves are shaved by the barber and then the question is, who shaves the barber? Because if the barber shaves himself, then, by this step, mean, it says that the barber does not shave himself.

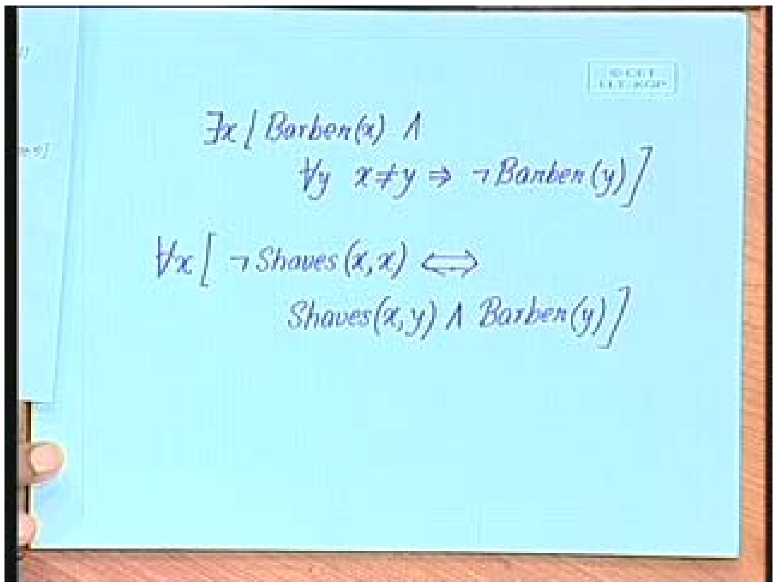
(Refer Slide Time: 00:52:12)



Because if the barber shaves himself, then he is not shaved by the barber, which is a contradiction. And if the barber does not shave himself, then he has to be shaved by the barber, which again is a contradiction. Now, if you try to model this in first order logic, let us see what happens. So, first thing we have to express is, there is a single barber in town. So, there exists x and for all y, x not equal to y, which is not of x equal to y, this implies, right? This expresses the fact that there is only 1 barber; there is only 1 barber. And then, the second statement says, that for all x, if it is the case that x does not shave himself, then and only then- so, we have both ways implication- x y, and, right? Now, where is the problem in this?

(Student speaking). The problem here is that, see this y- this barber, this person, is also in the domain of this x, right? This person is also in the domain of this x. that that this person is also a member of the town, right? Now, unless we encode that. Once we encode that, then we will see that the 3 together cannot be satisfied anymore. So, there has to be the inconsistence is in this class, which has those and only those- slides please, slides please. Yes.

(Refer Slide Time: 00:54:33)



Here you have: those and only those who do not shave themselves are shaved by the barber, and this those those other than the barber, right? So, this those actually can take values from the entire domain of the town, but then, unless we exclude the barber from that, there is an inconsistencies in this specification, right? So, in the next lecture, what we are going to do is, we will start formulizing the inference mechanism that will go with first order logic. Today, what we have seen is, what is first order logic and how we can write out different sentences in first logic. From the next class onwards, we will start studying the inference procedure in first order logic.