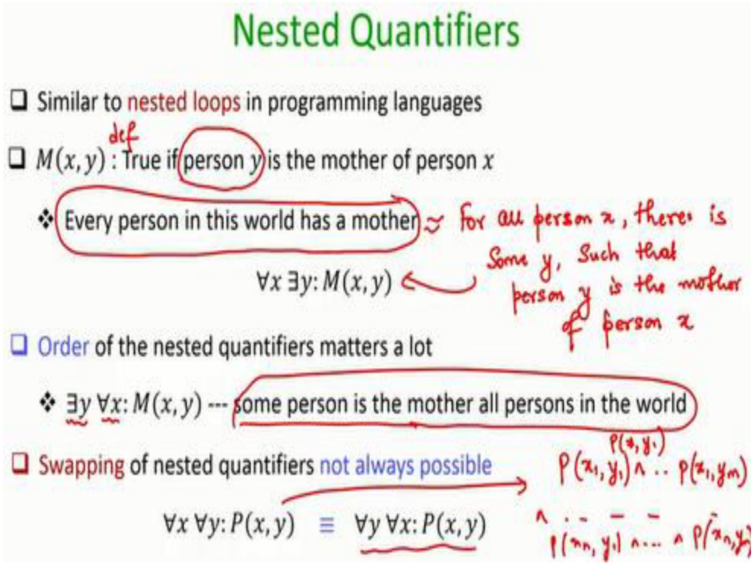


Now let us try to understand Nested Quantifiers, so there is very often we encounter statements where we need to have a nested form of quantification and this is similar to nested loops in programming languages. So let us see an example here, so say the predicate M(x, y) is defined in such a way that it is true if person y is the mother of person x, that is the definition of the predicate M(x, y).

And, I want to represent a statement that every person in this world has a mother. So my claim is that this can be represented by this expression for all x there exist y such that M(x,y) is true and this is an example of nested quantification. You want to say that you fix a value of x, for that fixed value x there exists some y, you change x then for the new x there might be another y, you change x then for the new x you have another y such that this property M(x, y) is true.

And why this is the expression representing every person in this world has a mother; well, this is equivalent to saying that for all person x, there is some y, such that person y is the mother of person x which is indeed what is represented by this expression. Now when you are dealing with nested quantification the order of the quantification matters a lot because if you change the order of the quantification then the logical interpretation of the statement changes completely.

For instance if I write an expression there exist y for all x, M(x,y); where M(x,y) is as defined above, the interpretation of that is you have there exist coming outside first, that means you want to say that there is some person y, such that all the x are related to that y. Namely the same y is the mother of all persons x in the world, that is not what we want to interpret. This statement some person is the mother of all persons in the world and every person in this world has a mother, they are two different logical statements.

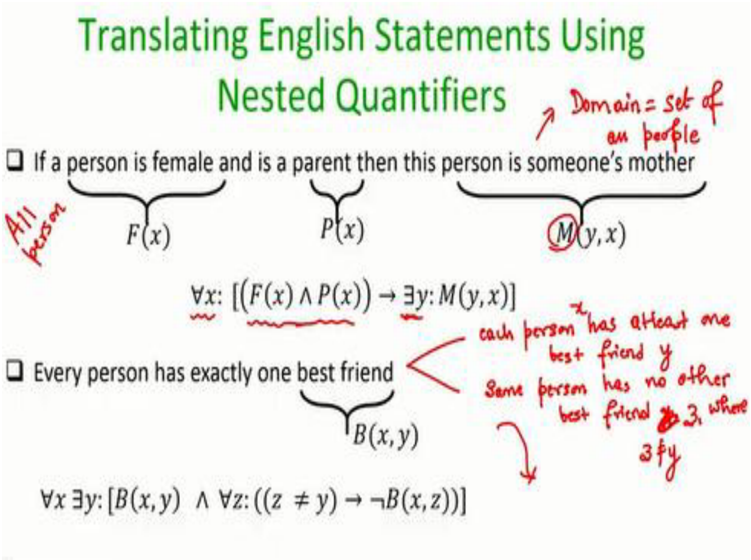
And hence they are represented by two different nested quantifications. So that is why swapping of quantifications are not always possible, it is possible only when you have the quantifications of the same type occurring throughout the expression. That means if you have an expression of the form for all x for all y or a sequence of quantifications which are of the same type then it does not matter whether it is y appearing first or whether it is x appearing first.

You can conveniently swap the order of the quantification and both LHS and RHS will be equivalent if you want to check that well for all x for all y can be considered as follows if you expand the for all x and for all y then it will be considered, imagine that x takes values from x_1 to x_n and y takes values from y_1 to y_m, right? I can expand this left hand side in this form and everything is conjunction here.

And, then I can swap and can shuffle around all the $P(y_1)$ first all of so I can shuffle around all the expressions of the form anything P of anything followed by y_1 and take them together and then followed by conjunctions of all P anything of y_2 and so on and that will be equivalent to the second expression right and this shuffling around is possible because everywhere AND is appearing and it satisfies the associative law.

But if you have an expression where you have quantifications of different form, then this kind of swapping may not be possible. The logical interpretations might be completely different.

(Refer Slide Time: 34:13)



So, now let us see some more examples here how we can start translating statements using the help of nested quantification. So suppose I want to represent a statement that if a person is female and is a parent then this person is someone's mother. So we have to first identify or define the predicates that we have to use here and here again, the domain is not explicitly given but you can imagine here that the domain is the set of all people.

So let me introduce this predicate F(x) which is true if person x is female and I also need a predicate P(x) to represent that person x is parent and I had already introduced a predicate M in the previous slide which I am retaining here. So first of all, this is a universally quantified statement because I am making a statement about all persons here, I am not making a statement about some specific person, I am making a statement about all persons.

So that is why this will be a universally quantified statement and this is an if statement of the form if-then your premise is for all person x in the domain I want to state that if the person x is female and if the person is a parent, so that is why conjunction of F(x) and P(x) then for the same x there exist a y, a person y such that x is the mother of y and you see how carefully I have put the parentheses here.

If I do not put the parentheses then the expression becomes ambiguous it will not be clear that whether it is x which is appearing first and then followed by y and so on. So x is occurring on a higher level and for each x there will be some y. Similarly if I want to represent statements of the form that every person has exactly one best friend, so this statement has two parts.

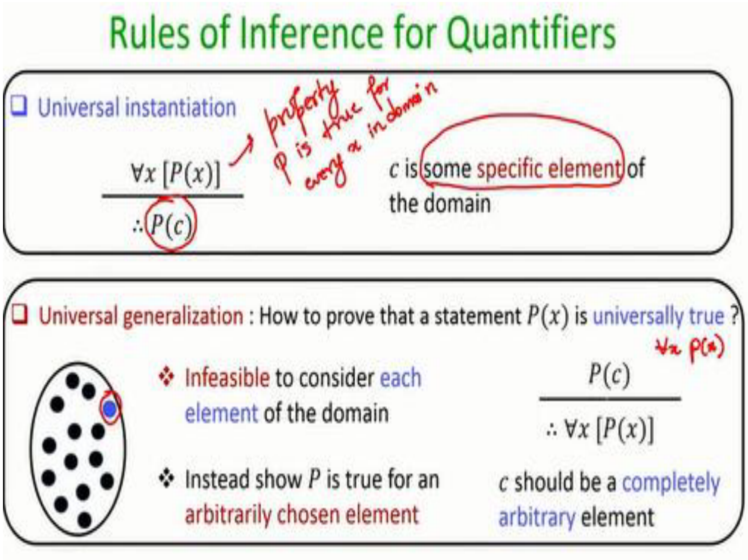
The first part is that each person has at least one best friend definitely, thats the first part of this statement : is one best friend in fact each person x has at least one best friend y and the second part is the same person x has no other best friend z, where z is different from y and this is true for all x that is what is the logical interpretation of this statement. So let me first introduce the required predicates here, so I introduce a predicate B(x, y) which is true if person y is the best friend of person x that is a definition of my predicate B(x, y).

And, now you can see here that since I have identified the two parts of this English statement, the first part is that for every person x there is some y such that y is the best friend of x and I want to state that for the same x there is no different person z different from y who is also the best friend of x that should not be possible, so that is why the left hand side represents the first part of this expression represents that person x has at least one best friend.

And the second part of the expression represents that person x has the possibility of a second best friend as well I want to avoid that and that is why I put a negation in front of that if I put the negation in front of that then that rules out the possibility that there is no second person z different from y who is also the best friend of x because of the occurrence of this negation.

And then conjunction of both these conditions will represent what I am interested to assert. Of course now, if you do want to apply the De Morgan's law of quantifications, you can take the negation, this negation that is here, and you can take it inside and then conjunctions get converted into disjunctions and so on and then you can apply the rule that negation P OR Q is equivalent to $P \rightarrow Q$ and this is another equivalent form of the same expression.

(Refer Slide Time: 39:25)



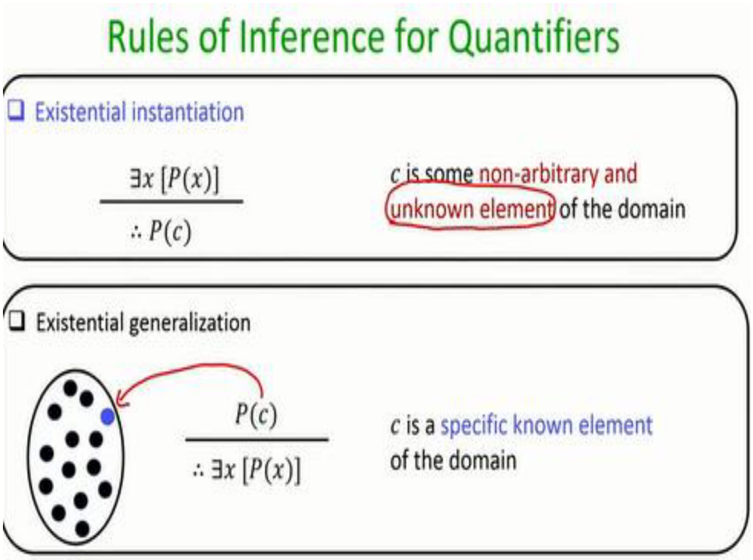
Now let us do some rules of inferences for quantified statements, so which are very important the first rules of inference is universal instantiation and argument form of this universal instantiation is if you are given the premise that for all x, P(x) is true, then you can come to the conclusion that the predicate P is true for some element c in the domain, where c is some specific element that you are interested in that you want to explicitly specify.

And, this is because since the premise for all x, P(x) is true that means property P is true for every x in the domain, property P is true for every x in the domain. So of course, it will be true for the element c as well, ok, whereas universal generalization has a different argument form, so what exactly universal generalization is used for so imagine you want to prove that a property P is true for every x in the domain that means you want to prove or assert that for all x, P(x) is true.

How do you do that? One option could be that you check whether property P is indeed true for x_1 or not, x_2 or not, x_3 or not and so on, where x_1, x_2, x_3 etc are the various values in your domain but this becomes infeasible if your domain is infinitely large. So to prove statements of the form that prove that something is true for every x in the domain where domain is infinitely large, very often we encounter statements of the form that prove some property is true for every integer x.

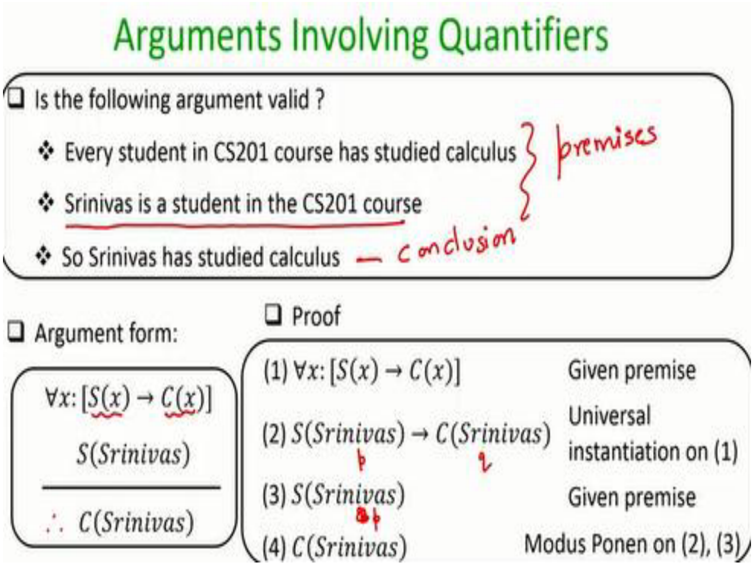
How do we prove it? We cannot take each and every integer and show that indeed the property is true for every integer that you have chosen. So to prove statements of that form, what we do is we pick some arbitrary element of the domain when I say arbitrarily element of the domain that means there is no specific property of that element, it is just some arbitrary element and show that the property P is true for that arbitrarily chosen element; if it is true for that arbitrarily chosen element, you can come to the conclusion that P is true for any element in the domain because the sample point that you have chosen was arbitrary. So the argument form here is if you show or if you know the premise that property P is true for element c where c is some arbitrarily chosen element then you can come to the conclusion that for all x, P(x) is true. So this is called universal generalization.

(Refer Slide Time: 42:09)



Now the duels of these rules are existential instantiation, which says that if you have the premise there exist P(x) then you can conclude at proposition P(c) is true where c is some non arbitrary but unknown element, I stress here that you may not be knowing what exactly is the element but you will be knowing that since P is true for some x in the domain, let c be the x for which it is true what exactly is that c you may not know that whereas existential generalization says that if you know that property P is true for element c in the domain where c is some fixed element, which you are aware of, that means you have a witness c explicitly for which the property P is true, then you can come to the conclusion that there exist x, P(x) is true.

(Refer Slide Time: 43:05)



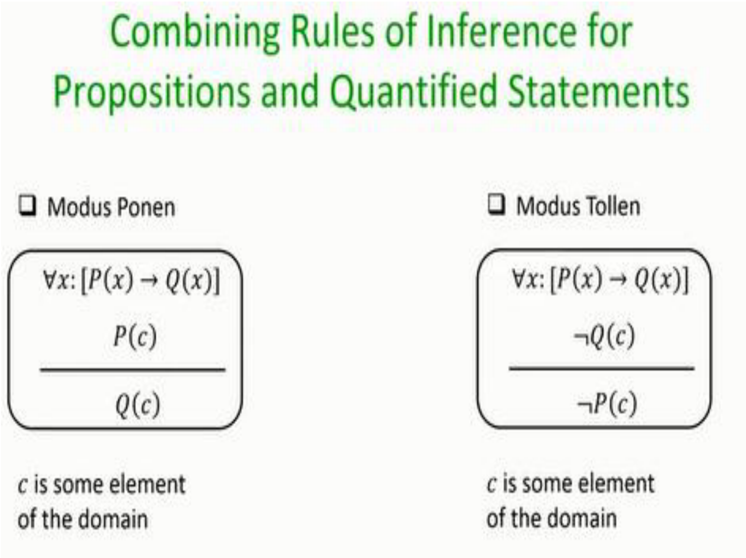
So these are four popular rules of inferences which we use involving which we use while dealing with quantifications. So now let us do an example to verify how to verify whether argument forms are valid or not, in predicate logic. So here you are given two premises and conclusion here. So I am retaining the same predicates S(x) and C(x) that we have defined in some earlier slides.

So the first statement is : every student in CS201 has studied calculus. So that is represented by for all x, S(x) → C(x) that is your first premise and the second premise is Srinivas is a student in the CS201 course that means the property S(x) is true for x equal to Srinivas; that means S(Srinivas) which is now a proposition is true, that is your premise. I said this is now a proposition because you have now assigned a value x equal to Srinivas.

The conclusion you are drawing here is that Srinivas has studied calculus that means you have to show that C(Srinivas) is true. So, let us see whether this argument form is valid or not, so you are given the premise for all x, S(x) → C(x) so what you can do is you can apply the universal instantiation and you can substitute x equal to Srinivas and get the proposition S(Srinivas) → C(Srinivas) to be true.

You are also given the premise S(Srinivas) to be true, now what you can do is you can think that this is now P → Q a proposition and a proposition P both these premises are true so you can apply Modus Ponen and come to the conclusion that C(Srinivas) is true.

(Refer Slide Time: 45:00)



So that leads us to the Modus Ponen and Modus Tollen rules. These are the generalizations of Modus Ponen and Modus Tollen to the predicate world. Modus Ponen says the following if you are given the premises for all x, P(x) → Q(x) and if P is true for some element c in the domain then you can come to the conclusion Q(c) and then same way Modus Tollen is generalized.

So that brings me to the end of this lecture. Just to summarize. In this lecture we saw how to convert English statements using predicates and logical connectives, we saw some rules of inferences using predicate logic and we saw how to verify whether a given argument form involving predicates is a valid argument form or not, thank you.