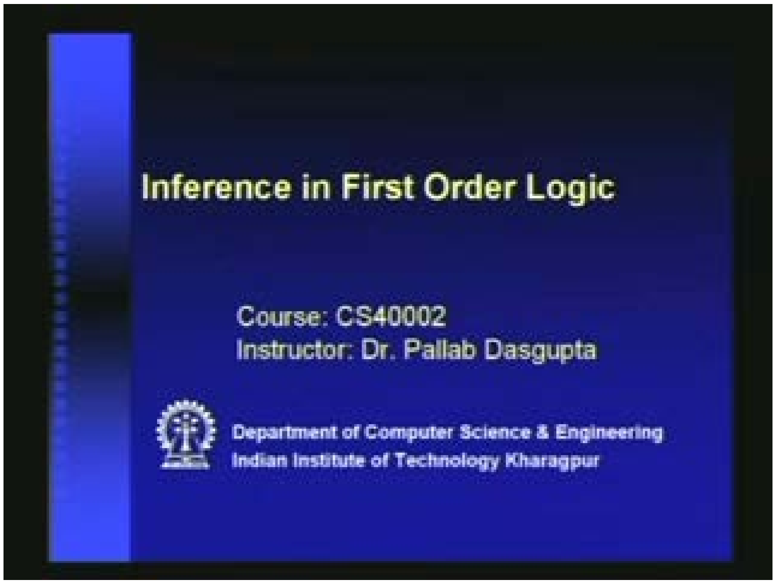
Artificial Intelligence
Prof. P. Dasgupta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture- 10
Inference in First Order Logic

I had introduced first order logic in the last lecture. So, from this lecture, we will start studying the mechanism of inferencing in first order logic.

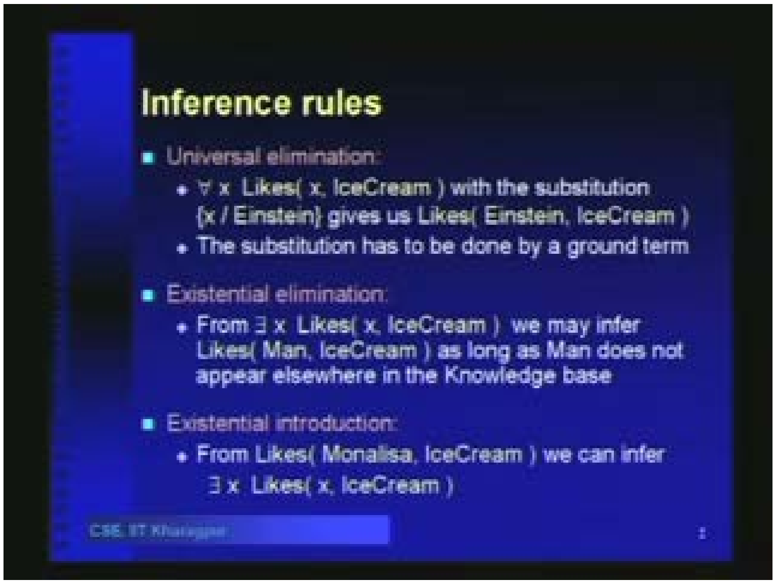
(Refer Slide Time: 00:01:03)



We will see how we can deduce new facts from the existing ones using first order logic. Now, just to do a brief recap: the difference that we saw between propositional logic and first order logic was that, in first order logic, you can have predicates, which have arguments. These arguments can be instantiated with values from a given domain, right, and we can have quantifiers, which says, whether there exists some member in the domain which can satisfy the predicate, which is- there exists the variable for that predicate, or it could be a universal quantification, which says that for all values of the variable x from each domain, the predicate is satisfied.

It is the existence of these quantifiers, which makes inferencing in first order logic more difficult than that of propositional logic. In propositional logic, we saw that we can reduce the inferencing problem to an equivalent sat problem of Boolean's satisfiability, and we have different Boolean satisfiability solvers, which can solve this. But when we go to first order logic, we the the domains of these variables can be very large and could be potentially infinite also, and therefore, it is not so easy to reduce the problem to a sat instance.

(Refer Slide Time: 00:02:40)



So, let us see some basic things that we need to do in first order logic: firstly, we need to have some basic inference rules: what can we deduce from what? The first of this is universal elimination. Universal elimination says, for example, that if we have for all x likes x ice cream, right, then, we can substitute x with Einstein and get likes Einstein ice cream. Now, we can do this, because we have this for all quantifier here. We are given that for all x likes x ice cream, in other words, it says that everyone likes ice cream, and so, if we replace x with Einstein, we get likes Einstein ice cream, and we can deduce this from this statement.

If this is given to us, then we can always deduce that Einstein also likes ice cream. The substitutions will be shown like this, when we have x slash Einstein, it means that we are substituting the variable x with Einstein. Then, we have a concept of a ground term- I will define this more formally. A ground term is- for the moment, let us say it is a constant, like we have Einstein here. It- by a ground term, we mean that it will not have any more variables within it, so, there is no further instantiation that you can do on a ground term. The second rule is existential elimination.

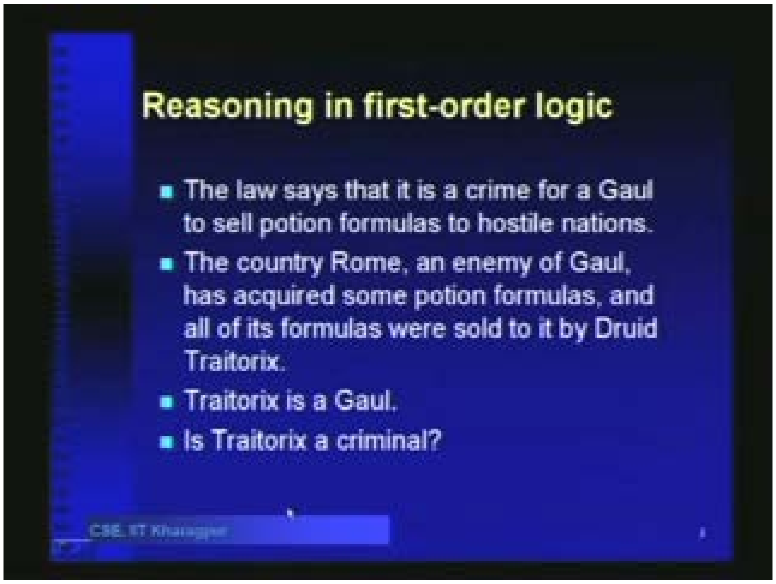
So, suppose we are given: there exists x likes x ice cream, right. Then, if we are given a particular person, say Einstein, can we infer that likes Einstein ice cream? No, we cannot, because we are given there exists x. But, which of these from the domain of x will like ice cream is not known, right? But we can do the following thing: we can replace x by, say, something like man, provided that man does not appear elsewhere in the knowledge base. If it is an entity which does not exist in the knowledge base, then we can actually replace x with man. So, this man here is a ground term; we are not going to further instantiate this, but this is the place holder for that x which likes ice cream. So, this 1 says there exists x, who likes ice cream. So we are say ok Let that person who likes ice cream be called man, right?

Now this man is a ground term, so, we do not have any other quantifier before it. But, at the same time, it is not 1 of the people whom we know, because if it is 1 of the people whom we know, then we do not know whether that person likes ice cream or not. (Student speaking). If if there is a person called man, then we cannot use man here. (Student speaking). Yes. No, the man is not a set- man is a ground term; it is 1 person, okay? It is like this- that we are told that someone likes ice cream, right, so, we are saying that okay, let that be man, right? And henceforth, we will say that man likes ice cream. It is also possible that someone else also likes ice cream; their existence does not prevent us from having more than 1 person, right? But we are naming this person man just to use man as a ground term in the inference procedure.

We will shortly see why this is useful, and when we generalize this thing, we will call this Skolemization, or replacing a variable by a skolem function. I will shortly describe that why this becomes useful. The third 1 is existential introduction. So, if we are given that likes Monalisa ice cream, then we can infer, that there exists x, likes x ice cream. If we are given that Monalisa likes ice cream, then, we can always say that there is someone who likes ice cream, right? So, in all of these 3, the first 1 is given to us and we can deduce the second 1 from that, right? For example, we can deduce likes Einstein ice cream from for all x likes x ice cream. We can deduce likes man ice cream from there exists x likes x ice cream, right? And likewise, if we are given likes Monalisa ice cream, then we can deduce there exists x likes x ice cream, okay?

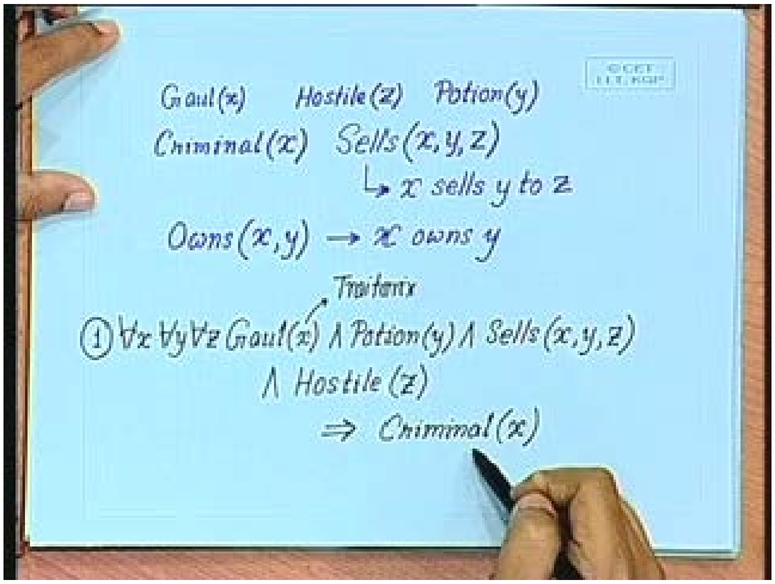
Now, let us first study 1 example of reasoning in first order logic, and then we will go to the formal way of doing it. So, this is an example which is inspired from the asterisk comics. The loss is that it is a crime for a Gaul to sell potion formulas to hostile nations, so this is our first sentence. The second sentence says, the country Rome, which is an enemy of Gaul, has acquired some potion formulas, and all of its formulas were sold to it by a Druid Traitorix. And then, we are given that Traitorix is a Gaul, and the question that we have to solve is that- is Traitorix a criminal? So, let us first see, that how we translate each of these statements into first order logic, and then, we will see how to do the deduction of the final statement from the first 3.

(Refer Slide Time: 00:08:03)



First 1 is: the law says that it is a crime for a Gaul to sell potion formulas to hostile nations. What predicates will we require? Okay.

(Refer Slide Time: 00:27:58)

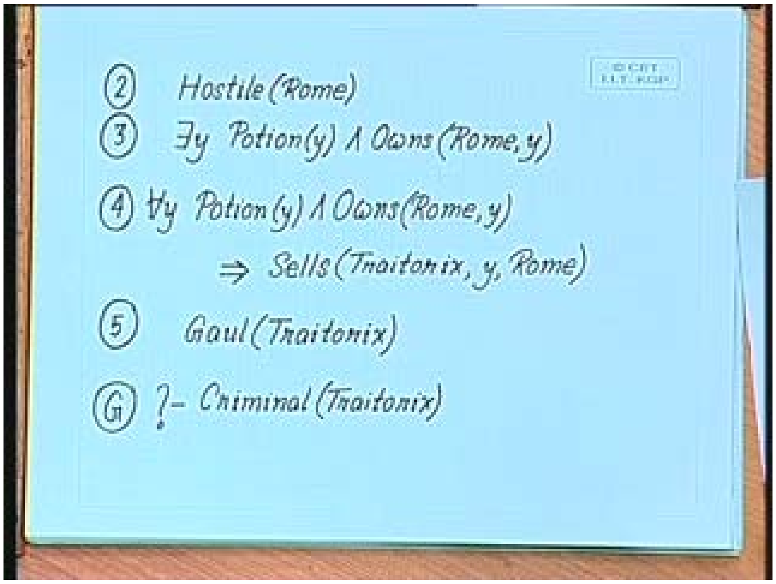


We will use predicates like, okay, let us write them down. Gaul x to indicate that x is a Gaul. Okay, hostile, say z, indicating that z is a hostile nation. Then, we will use potion y to indicate that y is a potion, right? And, we will also use criminal, say x, to indicate x is a criminal, right, and we will also have selling- sells- this will indicate, let us say, x sells y to z. So, x is is going to indicate x sells y to z, where y is a potion formula and x is a person and z is a country is a nation. And I think we will also require owns- so owns x y, which will indicate x owns y, where x is a potion formula; y is a nation, right? Now, let us look at the first statement, which says that the that the law says that it is a crime for a Gaul to sell potion formulas to hostile nations.

It is a crime for a Gaul to sell potion formulas to hostile nations. So, we can say that Gaul x and potion y and sells, and if x is a Gaul, y is a potion and z is a hostile nation, and x sells y to z, which means that this Gaul sells the potion y to the hostile nation z, then implies criminal x, right? Okay. When we write this kind of rules without saying anything here, it will mean that we always have a for all x, for all y, for all z. If there is any existential quantification, we will explicitly write it down, right? So, this is our first sentence- that the law says that it is a crime for a Gaul to sell potion formulas to hostile nations.

So, if anybody has done that, then it implies that that person is a criminal, okay? The second sentence says that the country Rome is an enemy of Gaul, right? So, what should we have for that? Hostile Rome. This is our second sentence: that Rome is hostile, for Gauls, of course. And then, we have that Rome has acquired some potion formulas, right? Rome has acquired some potion formulas. There exists a y, such that potion y and owns… this says that Rome owns some potion, right? And then, we are given that all of its formulas were sold to it by Druid territories, okay?

(Refer Slide Time: 00:18:43)

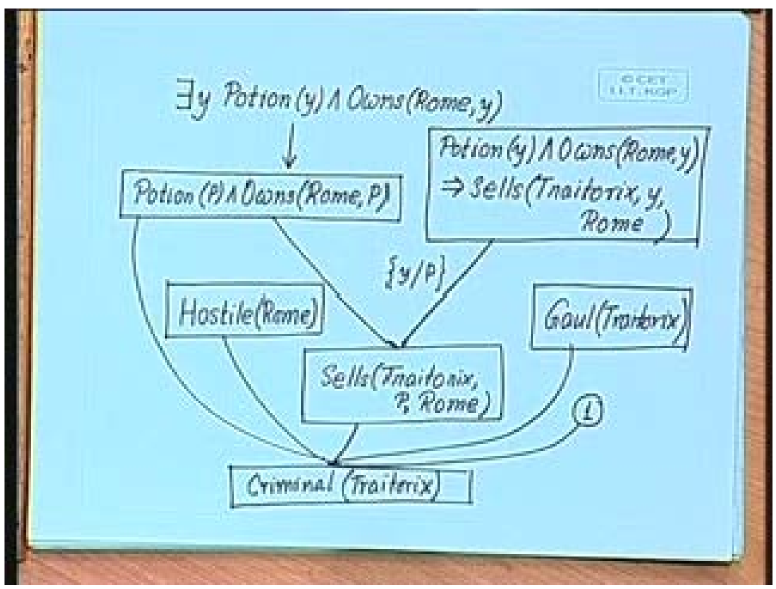


That means that for all y which is a potion- potion y- and owns Rome y, means that this was sold to Rome by Traitorix, so sells… right? So, for all y which is a potion and which is owned by Rome, so, all potion formulas that are owned by Rome were sold to Rome by Traitorix, right? Finally, we are given that Traitorix is a Gaul, so Gaul…, right? (Student speaking). Yes, the second 1 is to show that at least 1 potion is there, because we are also told that the country Rome and enemy of Gaul has acquired some potion formulas, right? So, that set of potion formulas acquired by Rome is no zero non-empty, right, and further, all this formulas all these formulas were sold to it by Traitorix.

Now, let us see- what do we have to deduce? We have to deduce that is Traitorix a criminal? The goal that we have the goal that we have is, we want to deduce this and we want to deduce this from this set of formulas. First 1 is this, right, and then, we have this as the second one, this as the third one, this as the 4th one, and this as the fifth one. So, we are trying to see whether 1 through 5 implies this goal g. Firstly, what we are going to study is- we will try to use modus ponens, as we did in propositional logic, which means that we will try to see whether the left hand side is satisfied, and if the left hand side of an implication is satisfied, then we can deduce the right hand side.

In an implication of the form a implies b, if a holds, then we can deduce b. We will try to go in the same way, and as we had seen previously, that there are 2 ways of doing this: 1 is forward chaining, where we start from the given set of rules and repeatedly apply them until we arrive at the goal, and the other option is to start with the goal, and then go backwards, to see that what do we need in order to derive this goal. We will study both of this, inferencing, using simple modus ponens, or the extension of the modus ponens for propositional logic, as we having first order logic.

(Refer Slide Time: 00:26:34)



Let us see; we want to start we start with the formula, so this is forward chaining.
We start with- there exists y, potion y and…, okay? Now, there exist some y, so, what we will do first here is, we will eliminate this implication by replacing y with something; something which does not exist in a knowledge base right now. You remember, we did this for that- there exists x likes x ice cream and we replace x by man? So, we will replace this y by, let us say, we will replace it by some potion p. So, once we do that, that gives us potion p and owns…, okay, and then, we also have the rule which says, portion y and owns Rome y implies that sells Traitorix y Rome.

Now, if we use these 2 now, we will do a thing called an unification- we will try to see whether we can use this to match the left hand side of this; in order to do that, we see that we have to substitute this variable y here by p. And there is no problem in doing so, because recall, that in all of these, we have this for all y outside. So, if this holds for all y, then it will also hold for p. Therefore, whenever you have for all, you can always replace the variable by any other ground term of your choice. We say that the left hand side of this rule will unify with this one, and that will happen if we substitute this y here with p. This is the substitution that we are applying on this and that is going to unify their left hand sides and therefore we can deduce the right hand side.

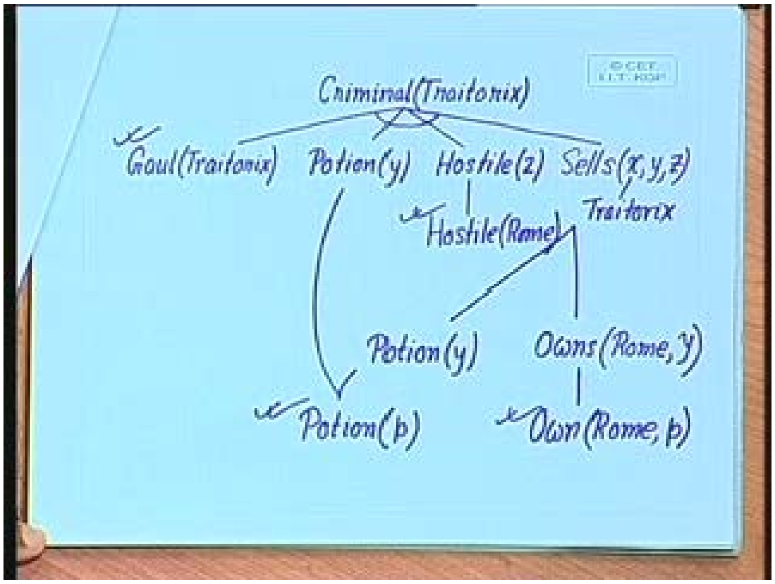
But again, when we unify it, when we do this substitution of y by p, we will we will have to do it on both sides. Therefore, we have we have deduced that sells Traitorix p Rome. We have sells and further, we are given hostile Rome, right, and we are also given Gaul Traitorix and from here we have potion p. Let us see: what do we have? We have potion p, hostile Rome, Gaul Traitorix and sells Traitorix p Rome, right? Now then, we can use this rule- Gaul x portion y sells x y hostile z implies criminal x. But the left hand sides will now have to be unified with these predicates that we have here. That means that here, x is going to be replaced by Traitorix, y is going to be replaced by p and z is going to be replaced by Rome.

If we do that substitution on the left hand side, then the left hand side matches with what we have here, in the knowledge base. We have potion p- we have hostile Rome, we have Gaul Traitorix, and we have sells Traitorix p Rome. These 4 facts I have already been deduced and are there in the knowledge base, so, they unify with the left hand side of this rule and it gives us the right hand side again, with the same substitution, which means that it gives us criminal Traitorix. Out of these 4 and that rule, and the rule 1 that we had, will give us…. (Student speaking). Here? Yes. No, no, because, see, you need to have Gaul x, potion y, sells xyz and hostile z- all 4 together, to infer criminal x.

If you just put potion p here, then you do not get that Traitorix is a criminal. You just get that for all y x Gaul x and potion p for all z sells xyz and hostile z is implies criminal x. See, once you instantiate something, that particular quantifier will go, but the others will still be there. You can only deduce criminal Traitorix if you have instantiated all of these variables, or else, if you are if suppose, you replace x with Traitorix and then, you have to find y and z, which satisfies potion y and sells xyz, only then, you can deduce criminal Traitorix. Once we have substituted x with Traitorix, y with p and z with Rome, then, we have all these facts in the knowledge base.

We have hostile Rome, we have potion p, we have Gaul Traitorix, and we also have sells Traitorix p Rome; all these facts were there in the knowledge base, when we applied this rule, and we have deduced criminal Traitorix. Now, this is forward chaining. This is where we started from the existing facts and rules in the knowledge base and deduced that Traitorix is a criminal. We can do the reverse also- we can start with criminal Traitorix and then try to figure out, that how to deduce whether this is true or not. For that, we will start with criminal Traitorix, and then work backwards. Let us see how that is going to work. We are going to start by saying that okay, my initial goal is to deduce that criminal Traitorix. Then we start with this rule- we start with this rule, because this is the rule which has criminal Traitorix on the right hand side.

(Refer Slide Time: 00:35:01)



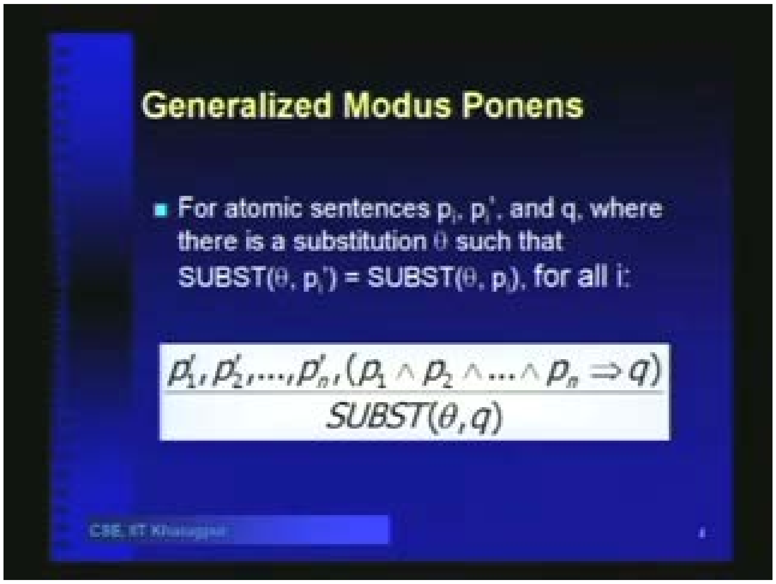
We will now start unifying with the right hand side. So, with the right hand side, if we unify this, then we will see that we can get criminal Traitorix, provided that we can find a Gaul x, a potion y; provided that we can find a potion y and a nation z, such that Traitorix has sold a potion to that nation. What we need, therefore, if we look at this rule, then, what we need in order to deduce that criminal Traitorix, is that we need Gaul Traitorix, we need some potion y, we need some hostile z and we need sells xyz, and this is an and, because we need all of these, we need to solve of these. Now, we have 4 sub-goals- 1 is Gaul Traitorix, 1 is potion y, 1 is hostile z and sells xyz. (Student speaking). Yes, sells Traitorix xyz- that is right, it should be sells Traitorix xyz, right?

But, these sub-goals are not independent, because they have this binding which has to be honored. For example, the potion y that you instantiate y with, should also be the 1 which has been sold. If we instantiate this potion y with some p and instantiate this y with some q, then Traitorix is not a criminal, because the 1 which has been sold and the 1 that you have here are not the same. There is a binding which is there between these. Then, once you have this, we will again move backwards, okay, this is a solved node, because this is given to us, this is there in the knowledge base, so this is solved. We do not need to do anything about this. But for potion y, we need to do something- we need a potion y, which has been sold to this thing.

What we will do is, we will start from this rule now, that sells Traitorix y Rome can be had, provided that there is that for all y potion y and owns Rome y. If we have to have this, then again, we have 2 sub-goals: 1 which says potion y, which actually we already have, and we need owns Rome y. Now, we are going to use this, that there exists y, potion y and owns Rome y, and we will replace use existential elimination to replace the y with a p. The moment we have that, we will have owns Rome p, and these 2 are going to have potion p, and recall that the moment that we substituted Rome here. We will now try to see whether we can instantiate this also with the same thing, and we indeed find, that in the knowledge base, we have hostile Rome, so this is going to be hostile Rome. With that now, we have arrived at the entire set of ground terms.

So, this is solved, this is solved, this is solved and also this is solved. Therefore, we have completed deducing that Traitorix is a criminal. We will study backward chaining in a lot more detail, when we study a language called prolog, which will start after a few lectures. I will start with prolog- that is a logic programming language, and there, you will see that this backward chaining is the predominant way of doing the computation. Let us now see, that what are important formalisms that are necessary to create a deductive engine out of what we have just now studied? We need the generalized modus ponens. What does the generalized modus ponens give us?

(Refer Slide Time: 00:35:51)



That if we have atomic sentences like these, pi, pi dash and q, where there is a substitution theta, such that if you apply that substitution, then pi dash and pi becomes equal for all i- okay, let us understand what this means. This is the left hand side of a rule, right, p 1 and p 2 and pn- these are all predicates, they have their arguments. p 1 dash, p 2 dash, pn dash, are also predicates which are there in the knowledge base. We know that these are given- so, we are given that p 1 dash, p 2 dash and pn dash are all given to us, and we are given this rule, that p 1 and p 2… pn implies q.

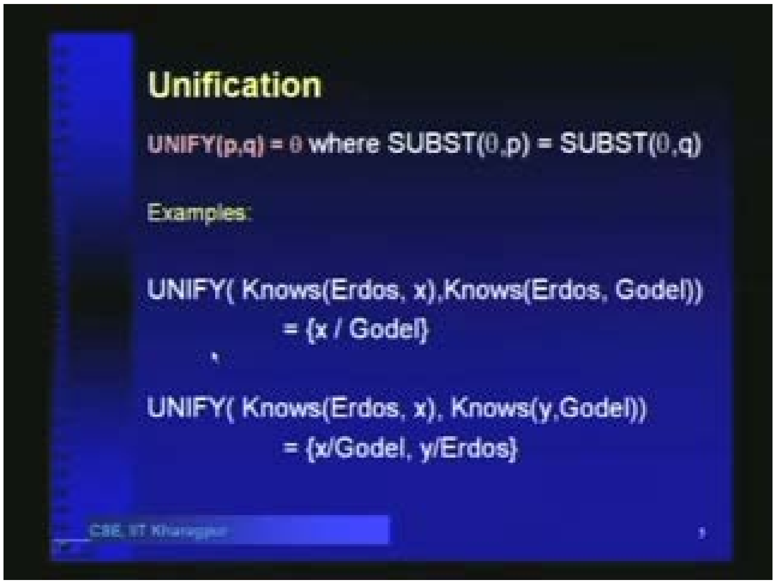
We are trying to see whether we can infer q from this, whether we can deduce q from this, and we can deduce q, provided that the left side of the rule unifies with what we have, like, what we were doing just now is, we were trying to see, that whether we were able to see the- text please. We were seeing that whether we find a Gaul, a potion, a hostile nation which unifies with x, y and z, such that sells xyz is also given in the knowledge base; when we have that, then we can deduce this. So, when we have a Gaul, a potion, a hostile nation and sells xyz, like we deduced from forward chaining, we had potion p, hostile Rome, Gaul Traitorix and sells Traitorix p Rome, then we could deduce criminal x. Therefore, these are our p 1 dash, p 2 dash p, 3 dash, etc., and these are our p one, p 2, p 3, etc. If they can unify with each other; in other words, if there is a substation of the variables such that the 2 becomes equal, like, for example, if we substitute x with Traitorix, then it unifies with Gaul Traitorix.

If we substitute y with p, then it unifies with potion p; if we substitute z with Rome, then it unifies with hostile Rome, right? So, the complete substitution that we have is, x with Traitorix and y with p and z with Rome, so that whole substitution is theta. If we looked here, if you have such a substitution theta, which is able to unify pi dash with pi for all i, which means p 1 dash with p one, p 2 dash with p 2, and so on, then we will deduce q also with that substitution, because recall that when we make a substitution on the left hand side of the rule, we have to make it also on the right hand side of the rule.

So, that gives us: we are able to deduce q with the same substitution. Is this clear? Is it clear? Yes or no? Fine. (Student speaking). Yes, in the goal, yes, in the- when we made the substitution of x by Traitorix here, we also have to make it here, this also has to be replaced by Traitorix, because you see, this is your whole formula, so if you make a substation of any variable with a ground term, you have to do it uniformly on the formula, because whatever is the x here, is also the x here, that is why we have this substitution of q also by theta.What is the important thing here? The important thing here is to deduce- how do we do this unification?

Because this unification of p 1 with p 1 dash, p 2 with p 2 dash, and pn with pn dash, is what helps us in using the rule. We have to be able to formulize a mechanism for doing this unification. Let us look at some examples of this unification. The first 1 is, let us say- so, we define this function unify p q as theta, where substitution of p with that theta- now, what do I mean by this? p is a formula, theta is a substitution; what kind of substitution is theta?

(Refer Slide Time: 00:41:10)

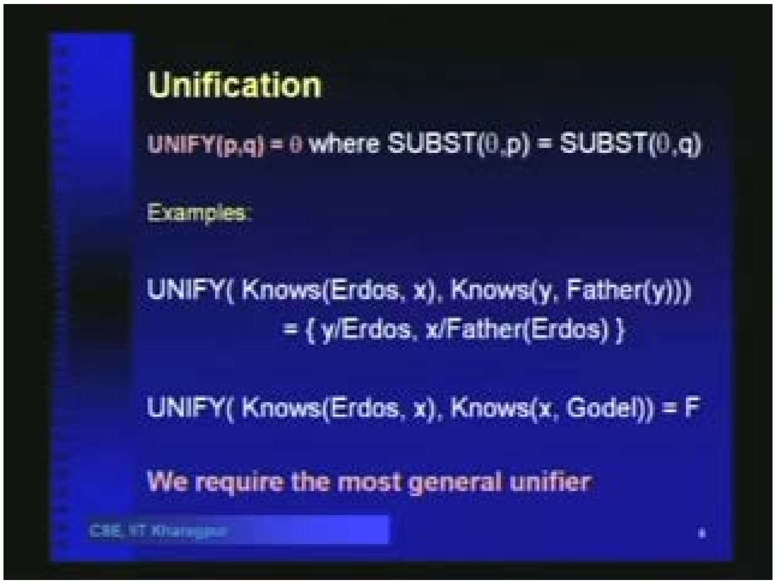


Say x by something, y by something, z by something, like that, right. When we apply that substitution on p, and when we apply that same substation on q, we should get the same predicate. If we are able to get the same formula by doing the substitution, then we say that p and q are unified by the substitution theta. Is it clear? p is a predicate, q is a predicate and the same substitution is able to unify the 2 or make them equal. Let us see some examples. Suppose we have knows Erdos x is 1 predicate, knows Erdos x. Erdos is the name of a very famous mathematician, and we have knows Erdos Godel, right? Then, the unification here is x replaced with Godel, because if you replace the variable x with Godel, then these 2- this p and this q- will become identical.

Now, this example is inspired from the Erdos number. Are you familiar with what is the Erdos number? (Student speaking). Yes. So, the Erdos numbers were very interesting, because Erdos wrote lots of papers, and there are several numbers which indicate how many co-authors he had and how many people published a paper with a co-author of Erdos. So, 1 of the Erdos numbers is direct co-authors of Erdos, another Erdos number is the number of people or the number of publication where a person's Erdos number is number of publications in which that person was a co-author of a co-author of Erdos. And that number becomes- actually, if we go to 1 or 2 more levels, the number becomes really staggering.

Anyway, so let us return to our example. Let us say that we have knows Erdos x and knows y Godel, right, then, the unifier is x replaced with Godel and y replaced with Erdos. And see this theta- the substitution is actually replacing 1 variable of this predicate, another variable of this predicate. But after that substitution is done, then, these 2 becomes identical, because the left hand side will become knows Erdos Godel; the right hand side will also become knows Erdos Godel.

(Refer Slide Time: 00:44:50)

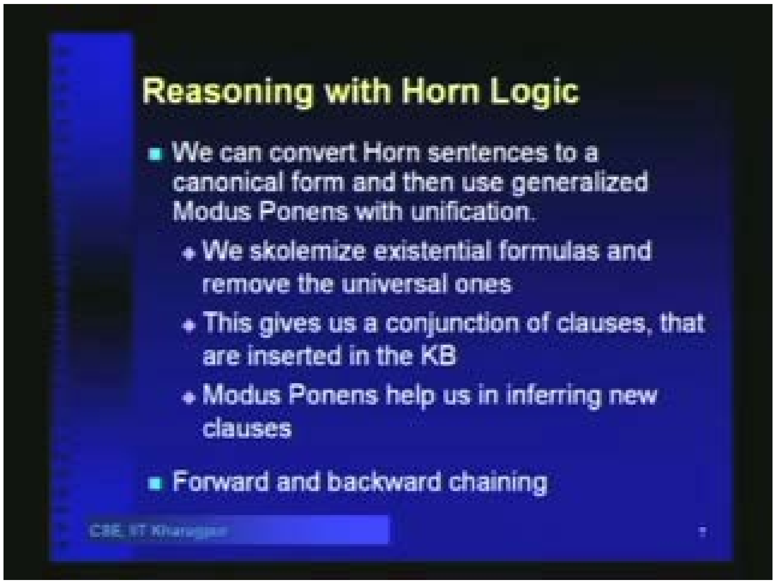


Now, let us get to a few little more complicated example. Suppose we know, we have knows Erdos x and knows y father of y- so, every person knows the person's father, right? Now, here, the unifier is y with Erdos, and then this is more interesting- x with the father of Erdos. Now, see the- what is this father? It is a function, right? Recall that in first order logic, we can have functions also as arguments of predicates. Then, this side will become Erdos, father of Erdos, and this side will also become Erdos and father of Erdos. Then, when does unification fail? It fails in cases like this, that you have knows Erdos x and knows x Godel; now, this x is a common variable here, and now there is no way that you can unify these 2 or there is no substitution of x which will make this predicate and this predicate identical. So, the unifier is empty.

Now, the important thing is that we require the most general unifier. And why do we require the most general unifier? And what is the meaning of the most general unifier? The most general unifier means that when we replace a variable with a ground term, then we are effectively replacing- we are creating a special instance of that rule, and deducing a fact for a very specific person. Like if we have for all x likes x ice cream, and we replace x with Einstein and put likes Einstein ice cream, well, that is fine, but it also- we must also remember that this is just 1 of the many special cases of, for all x likes x ice cream, because there are every other person who also likes ice cream. The fact that likes Einstein ice cream is fine, but if that is not what we require for our proof, then, we could have- we could generate a lot of people liking ice cream without actually using the fact.

When we replace something with a unifier- when we unify something, we will have to see that we unify in the most unrestricted way. The more ground terms that you bring into it, you will find that it is the the right hand side will also have more instantiations and it will be useful for a more restricted set of cases. During some tutorial, I will talk in more detail about the most general unifier and how to derive the most general unifier. And we will also see, that when we use the most general unifier, then the number of steps in your deduction will go down. Did you get an idea about what is meant by the most general unifier? We will study this in more depth when we study prolog.

(Refer Slide Time: 00:48:35)



This is a logic which is a which is a fragment of first order logic- it is called horn logic. And what is a horn logic? Do you remember that when we are studying propositional calculus, I briefly mentioned what are horn sentences? Horn sentences are one, where on the left hand side of the rule, you can have a conjunction of many predicates, and on the right hand side you have a single goal. Now, we do not allow or of predicates on the left hand side or the right hand side. The similar thing applies for first order logic also, and that fragment of first order logic is called horn logic. And in horn logic, the advantage is that you can always repeatedly use modus ponens to unify with the left hand side and deduce the right hand side. So, what we can do is, we can convert the horn sentences to a canonical form, and then use generalized modus ponens with unification.

We can convert them to implicational form, some left hand side implies right hand side, and then use this modus ponens consistently. So, we will skolemize existential formulas and remove the universal ones. Skolemization means that when you have there exists of some variable, replace that variable with some constant which is not there in the knowledge base, as we did for the potion, and as we did for that replacing x by man. Then this gives us a conjunction of clauses that are inserted in the knowledge base and modus ponens helps us in inferring new clauses. And as we have seen, that there are 2 ways of doing this: 1 is forward chaining, another is backward chaining. So, what we had here right now was actually a case of- what we had in our deduction here was indeed in horn logic.

So, if you recall that we started with this one, we skolemized this out and obtain potion p and owns Rome p and thereafter, we kept on using modus ponens. We unified the left hand side of this rule with this, and thereby, we were able to deduce the right hand side.

Now, the idea is that whatever you deduce on the right hand side, we go back into your knowledge base. You will see whether you are able to unify the left hand side with what you have in the knowledge base. In this case, we have potion p and we have owns Rome p; we have unified that with the left hand side, deduce the right hand side, this goes back into the knowledge base. And then, we have hostile Rome, Gaul Traitorix and potion p in the knowledge base, and these 4 together, instantiate the left hand side of this rule and we are able to deduce criminal x which we have here.

In this way, if we have horn logic, we can continue applying modus ponens to add new and new newer and newer facts into the knowledge base, either through forward chaining or through backward chaining, and we are able to deduce the fact. But we will see that this reasoning is not complete, in the sense that there are formulas in first order logic which can be deduced, but cannot be deduced with this kind of inferencing. So, generalized modus ponens is not a complete reasoning procedure. And then, in the next lecture, we will study a mechanism which is a complete resolution procedure or a complete proof mechanism, right? We will define what I mean by complete, and we will study that in the next lecture.