Artificial Intelligence:

First Order Logic with Equality

Prof. Deepak Khemani

Department of Computer Science and Engineering

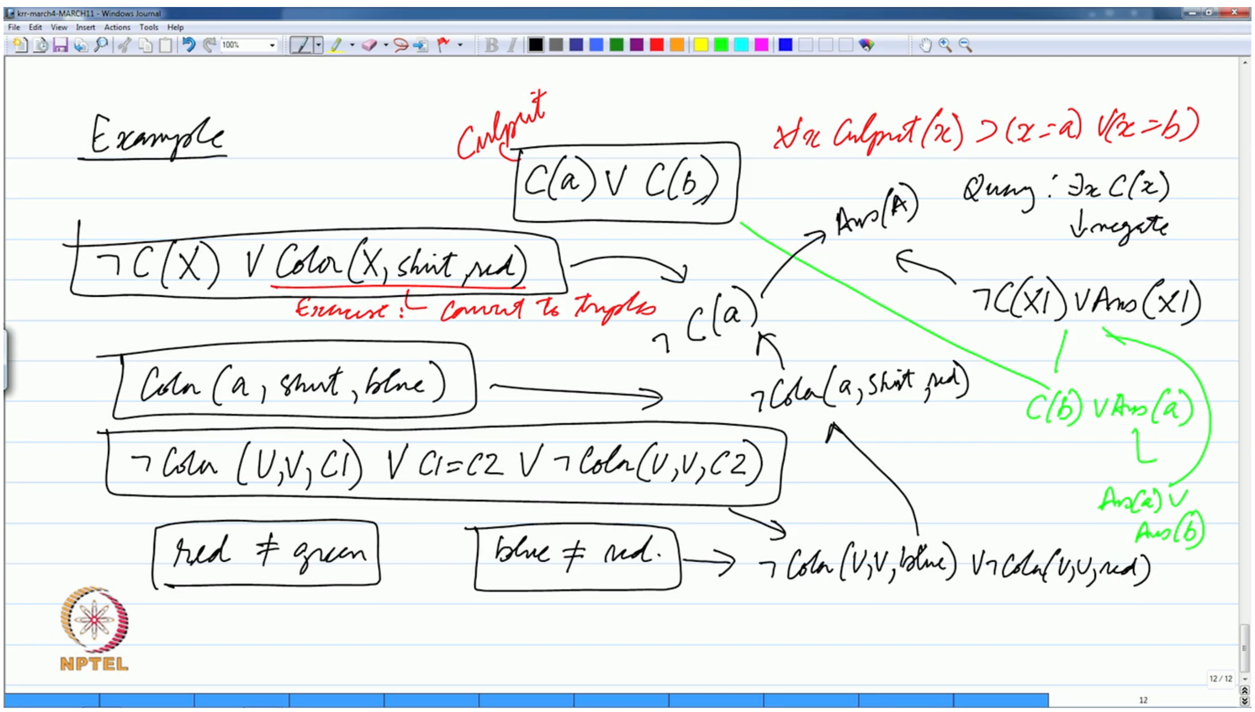
Indian Institute of Technology, Madras

Module – 07

Lecture - 04

From this and this I will get Answer here.

Refer Slide Time: 0.32)



So the black television terminated by pinpointing to me that A is the culprit which is like logically follows from that because you have said either A or B is the culprit wearing red shirt where B is wearing blue shirt. And then we have some information about the colour. So we can infact infer that he is the culprit. But we could have two definitions essentially; one derivation I would not know who is the culprit but I would still say that the query we are interested in is this that Is there somebody who is a culprit? You can say that statement is true but depending on the strategy you follow you may or may not identify the answer. So in the next class we will look at this equality in a little bit more detail. In particular I will ask a question that given a is equal to b and b is equal to c, Is a is equal to c. how do we prove? If somebody says that given a is equal to b and if somebody says that b is equal to c, how do we prove that a is equal to c. And equality as we will see in next class is a special kind of predicate which carries forward some hidden knowledge which we will need to add to the system if we are able to show this.

Ok so we are looking at resolution method and we want to focus a little bit on equality. So remember that a term i is equal to term j is a formula. And obviously the negation of it is a formula once it is a formula. But if I give you an example which says that a is equal to be whatever a and b are, some constants in your domain and b is equal to c, then this is true that a is equal to c. now we know from our understanding of equality that if it is indeed the case that a is equal to b and b is equal to c then a is equal to c. but can we derive it? Is there a proof for this? Now you can say that if I gave you three clauses or three sentences, a is equal to b… there is no way you can arrive at a is equal to c unless you do something about equality essentially.

So we need some knowledge of equality. So what is this knowledge? We will call this equality axioms. These are some true statements that we will add to any knowledge base if we are building a knowledge base which uses equality. What are these exioms? These are I am sure you are familiar with these, these are properties of equality relation that is

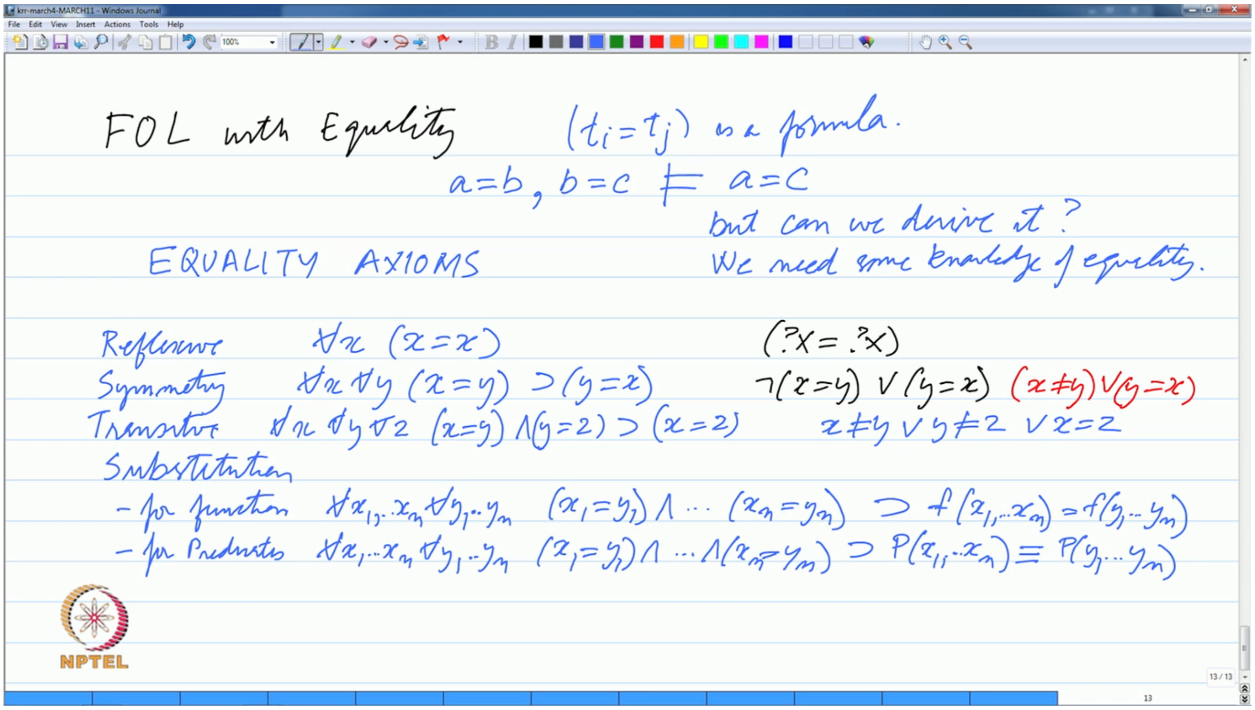
Reflexive: for all x x is equal to x. in clause form you will replace this by saying. Then the other property.

Symmetry: for all x for all y x equal to y. we could have used equivalence here but it suffices to use implication because its anyway universally quantified variable. So when you say x equal to y implies y equal to x, it could have been y equal to x implies x is equal to y. how do we write this in clause form? NOT of x is equal to y or y is equal to x which we will also write as x not equal to y OR y is equal to x. both are equivalent ways of writing this.

And then Transitive, equality is transitive and that is something we need for the problem we started with. If a is equal to b and be is equal to c then transitivity tells us that a is equal to c. for all x for all y for all z , x is equal to y and y is equal to z then it implies x is equal to z. which we can write as x not equal to y or y not equal to z or x not equal to z.

Then we have some Substitution properties. So for function in its general form, we will state it as follows that for all x1, so this is also a way of writing the set of for all statements. So when I say for all x1 to xn, it is essentially a shortform for saying for all x1 for all x2 upto for all xn. For all y1 to yn, x1 is equal to y1 and upto xn is equal to yn implies f x1 xn equal to f y1 yn. If I have this equal given to me tha x1 is equal to y1 and x2 is equal to y2 and so on and xn is equal to yn then f of x1 x2 upto xn would be equal to f of y1 y2 upto yn essentially. Notice that we are using equality there because f of x1 is a term there, f is a function. Likewise for predicates, the same or similar statement, if P is a predicate on those n variables.

(Refer Slide Time: 10:10)



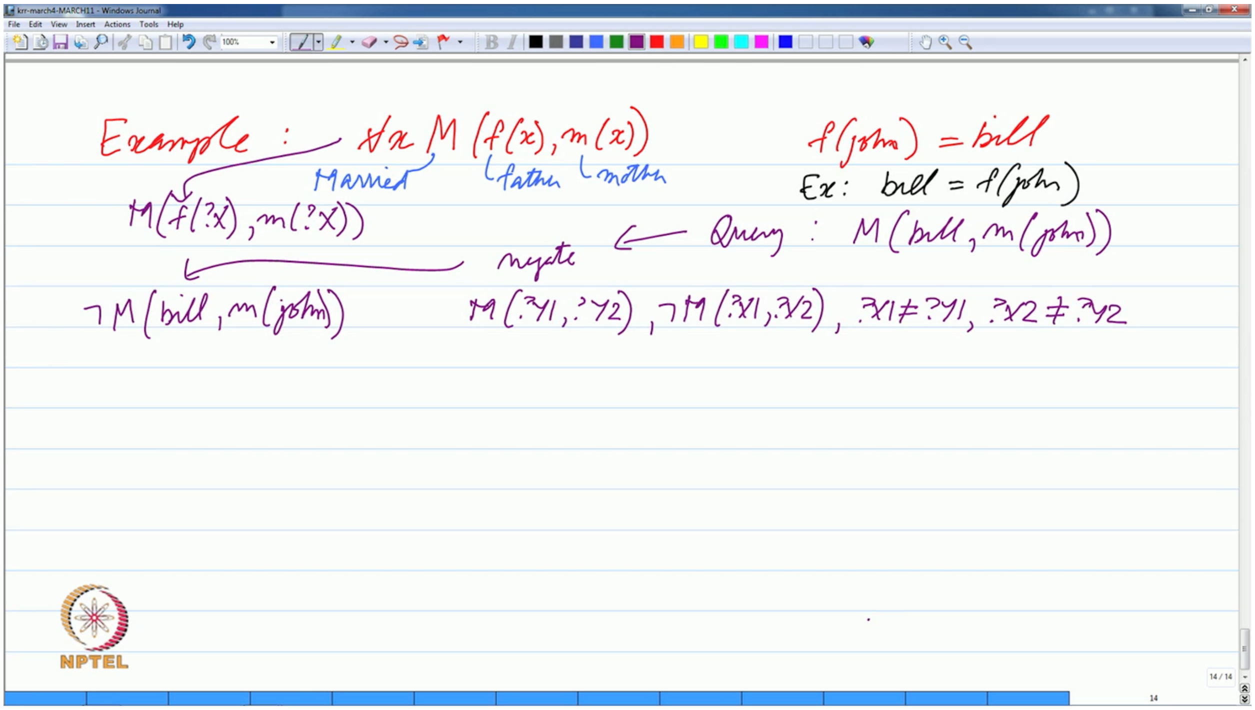
Which as an exercise I will ask you to show is can be written as not x1 is equal to not y1, not xn not equal to not yn. I am using comma here,I could have used disjunct here. So you will notice the clause form that I have written and you should verify that this is the case. I have taken only one side of the equivalence, when I said P of x1 to xn is equivalent to P of y1 to yn, I have taken in this clause form only one side of it. So this clause form should be read as if x1 is equal to y1, if x2 is equal to y2 and xn is equal to yn and P of x1 up to xn is true then P of y1 y2 upto yn is true. So this whole thing on the left hand side you can put in left hand side of the implication and whatever remains is the right hand side of the implication.

So how do we now may be as an exercise I would ask you to show that if a is equal to b and b is equal to c then a is equal to c. all you have to do is use some of these axioms along with what is given to you and the negation of a is not equal to c. we will instead take another example which is from the book Reckman and Levis.

And this example says that the following is given to you. For all x, I will just use M and I will explain this. These lower case letters are functions so f stands for father and m stands for mother and this upper case M is a predicate and It stands for Married. So what this statement is saying, that everyone's parents are married. So for all x father of x is married to mother of x. lets take that as given. And we are also given a statement which says that father of john is bill. I will give you an exercise here that instead of this If I had given you the other way round you must show the same thing that we are trying to show. And you can see that if you use a symmetry axiom you can convert one form to another. Now this needs a little bit of practice so I am just leaving it as a small exercise to you. Whats given to us are these two formulae in red so this will get converted into married father X, mother X, and that's given to us which is father john and the query is to show that this statement is true. Married bill to mother of john. So we are given that bill is john's father and we are given that everybody's parents are married so all we have to show is bill is married to mother of john. That's the formula essentially.

So we negate this and as before as a clause. Not married bill mother of john and we use the substitution of predicates to produce a formula which says that

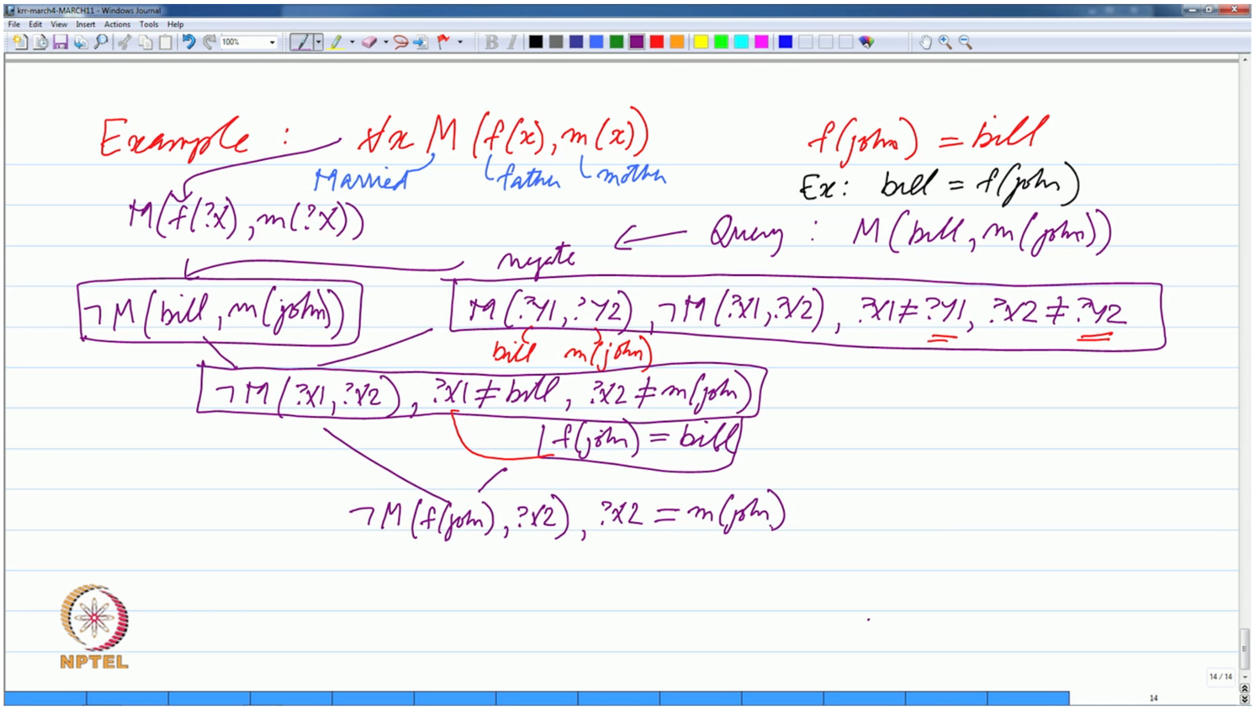
(Refer Slide Time: 16:44)



Now this is an instance of the substitution of predicates and it says that if x1 is equal to y1 and if x2 is equal to y2 and if married x1 x2 then married y1 y2 or if married y1 y2 then married x1 x2. So we can now resolve this with this by saying that, the substitution is this gets bill and this gets mother of john so y1 is here also and y2 is here also. So this part will go away and what will be remaining is not married x1 x2 or x1 not equal to y1 which is bill or x2 not equal to y2 which is mother of john.

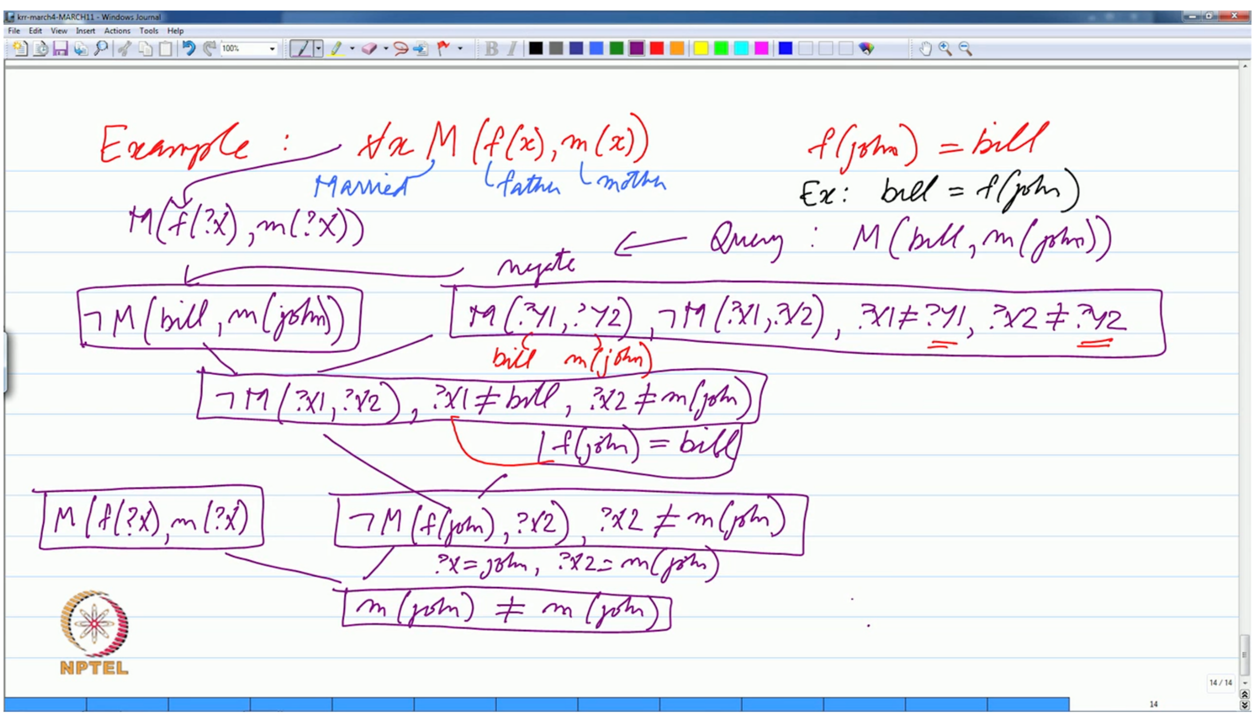
Next I use the other fact which is given to us which is father of john is equal to bill. So let me put them in boxes so there is no confusion. So this father of john is a part of what is given to us. And I can match here x1 to father of john and bill will match bill so this will go away. And what will remain is not married now x1 is father of john, x2 is still a variable and x2 is equal to father of john.

(Refer Slide Time: 19:33)



Now I can reuse this fact that I have written here that everybody's parents are married. And resolve it with this. So x will match john, and I will get x. so let me write it here, x is equal to john and x2 is equal to mother of john so I will get. I should have had a negation here, so I will get mother of john is not equal to mother of john. Which obviously is not a true statement. But we don't have to rely on our knowledge here. Since we have already put our axioms of equality.

(Refer Slide Time: 21:29)



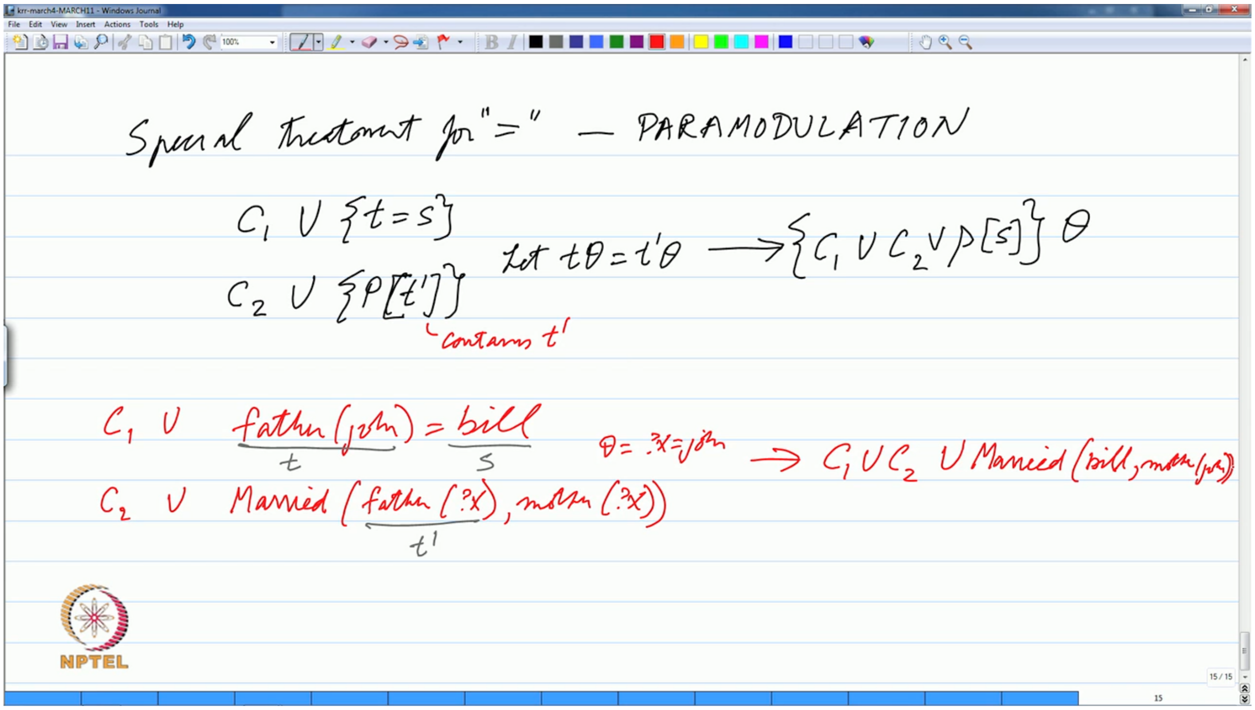
And remember that the reflexive axiom says that X2 is equal to X2, its just a statement of the reflexive axiom that for all x they are the same essentially. So you can see that from this and this you can derive a null essentially. What did we do here? We used the substitution in predicate and we used the reflexive property. And in this exercise that I have give you you will see that you will also need to use the symmetry property essentially. Or may be you can find an alternative proofs. One of the things about proofs is that you can have different proofs essentially.

Okay so you can see that when we are talking about equality sometimes we have to do a lot of work to prove somethings which are fairly obvious essentially. So here we were given the fact that everybody's parents are married and we have been told that john is the father of bill and we just wanted to show that johns father which is bill is married to john's mother which is mother of john. So she is an unnamed individual here. Which seems to be obvious from the given fact but to prove it we have to go through the whole process of using all these equality axioms.

So sometimes what people have tried is they have suggested shortcuts essentially. We can call it special treatment for equality and this is a method called Paramodulation. So this is another rule that we are adding to the system. So far we have added only resolution rule but now we are adding another rule which is paramodulation. And this rule is as follows that if we have a set of clauses C1 and somewhere a statement which says t equals s, so we are bothered about equality. And we have another set of clauses C2 and we have another set of predicates which involve t prime. So this square bracket we will use to say that contains t prime. Its not that it's a function of t prime it's a function of other things also but amongst other arguments is this variable called t prime. And this square bracket is what we will use to denote that. And then if we can find a substitution. So let some t theta is equal to t prime theta. If we find a substitution that will make the term t match this t prime. Both of them are functions remember but they could have functions inside them. Then we can derive C1 union C2, this is something like we did for resolution, we carry forward C1 and C2 but instead of eliminating the other clause we are replacing with a function of this s. because we know that t is equal to s and for this whole thing we apply theta. So for this new rule which kind of is a shortcut for handling equality is called paramodulation. So I will just illustrate it with an example we just saw.

Which is that you are given father or john equal to bill. And lets say there is something else which we are not bothered about. And then we are given C2 something and this statement. And now theta is equal to x is equal to john and we can now derive whatever that C1 and C2 were UNION I will replace so this corresponds to the term t, this corresponds to the term t prime, this corresponds to the term s, so I can replace it with s and get this statement Married. So I can replace t prime with s which is bill and since I need to apply this substitution theta which is x is equal to john so I must get mother of john.

(Refer Slide Time: 28:14)



So this whole exercise that we did a short while ago and we went through a proof process the same two facts were given to us that everybody's parents are married, father of x is married to mother of x and father of john is bill and we wanted to show that bill is married to mother of john. We can now do it in one step essentially. So this paramodulation as you see is a shortcut for doing many of the things you need to do when you are handling equality. Okay so we will stop here and in the next class we will continue looking at the resolution method. We will try to observe that it can lead to computational difficulties and we will just quickly look at what are the ways we use to address those computational difficulties. You have already seen that it is semi decidable so we still have to be careful about giving the right inputs. If you give something which doesn't follow then the program may never terminate. But then we are all used to a program sometimes going into infinite loop. And its not as if you can somehow intelligently decide whether a sequence of resolution steps is going into a infinite branch because that would amount to being able to solve in some sense the halting problem and we have seen that logic programming is just like any other programming language which means its just like a turing machine. In the next class we will wind up with resolution method and we will look at some of the issue which come , you can handle the complexity part somehow.