Artificial Intelligence

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

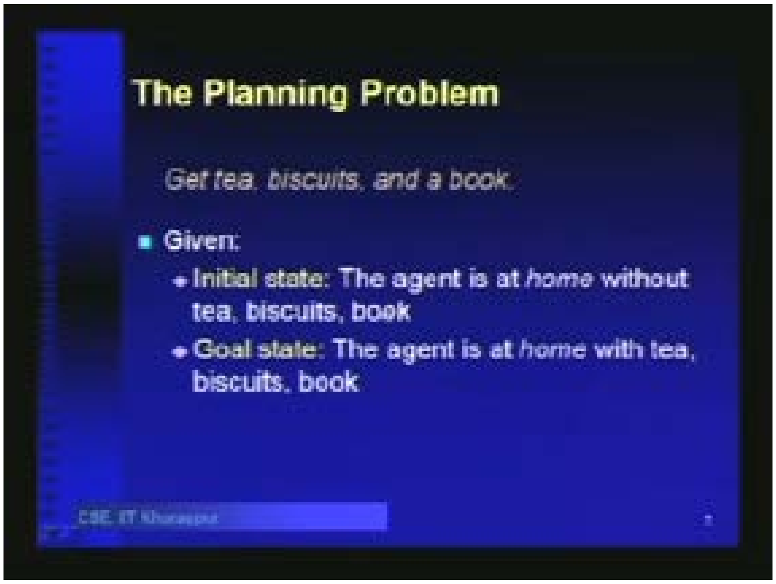
Lecture No- 18
Partial Order Planning

In the last lecture, we had discussed the planning problem and seen how it is related to general searching- how we can represent plans and how we can represent solutions to the plans, in terms of these things.

(Refer Slide Time: 01:09)

We will briefly recap what we had seen; just the outline of that, and then, we will start studying different kinds of planning strategies, out of which we will mainly study 3 different strategies. The first is partial order planning. Partial order planning is an algorithm which was developed sometime back, and then, we will study some more recent algorithms, namely an algorithm called graph plan and an algorithm called SAT plan. These have evolved in the 90s, and have rejuvenated the notion of how planning problems are solved. In this topic, let us quickly recap what we had seen before. A planning problem is something like this, where we want to reach a certain goal and typically, in a planning problem, the goal that we want to reach can be split up into possibly independent sub-goals.

(Refer Slide Time: 02:29)

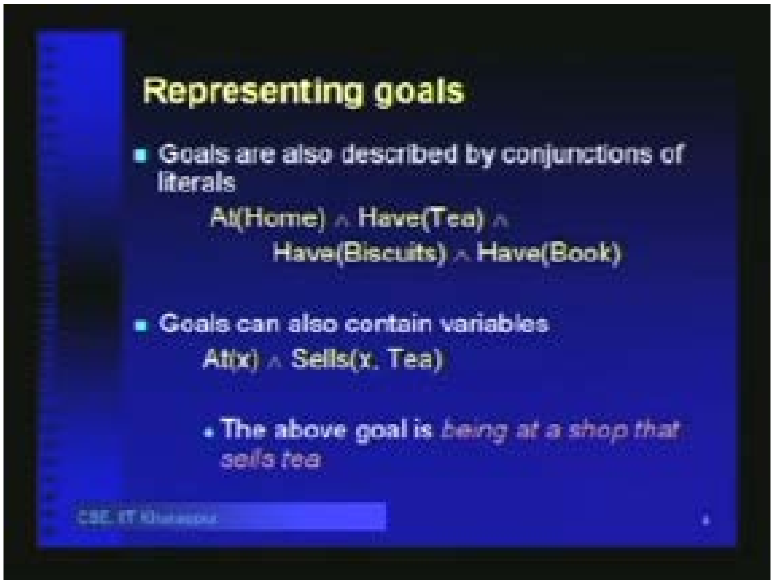


For example, here, we have get tea biscuits and a book, and these could be independent, but it could also be the case that certain kinds of actions can generate both of them, in which case, there will be an implicit dependency between them. So, the representation is in the form of an initial state and a goal state, and we can represent the states in variants of first order logic, represent the goals in a similar way.

(Refer Slide Time: 02:35)

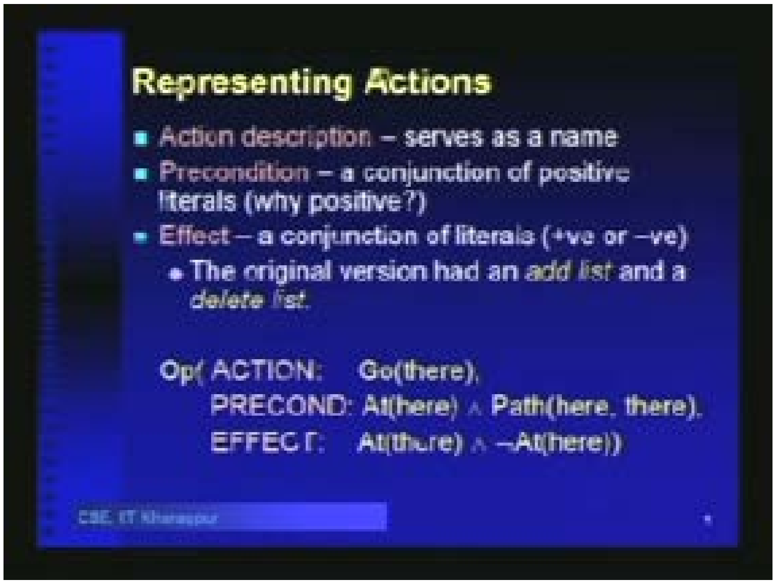
As we had seen, that there are certain constraints that we have on these states, etc. Like for example, in the language strips, the states are represented by conjunctions of function free ground literals. The functions are not allowed, whereas lateral languages like adl will support that. Then, we had seen that we can represent goals also as a conjunction of literals; more recent languages also support disjunctions of literals and goals can also contain variables.

(Refer Slide Time: 03:18)



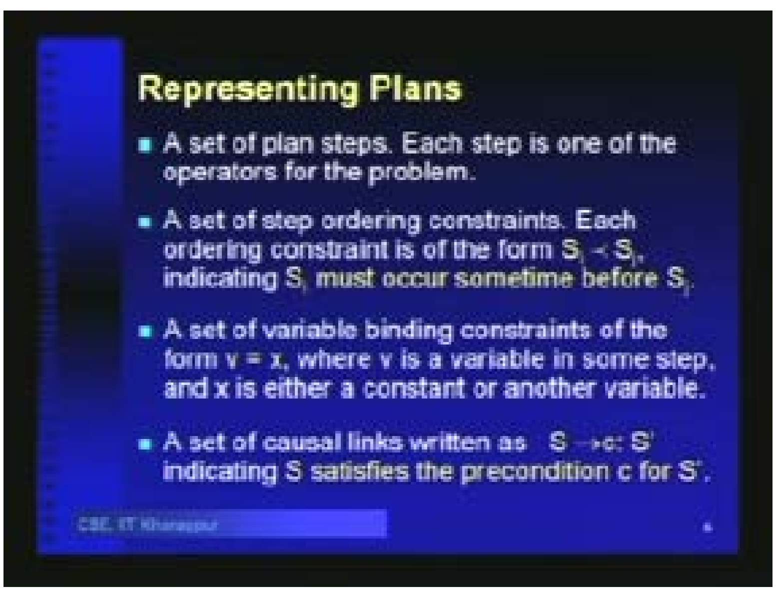
Then, we had seen that the other input to the planning problem is a given set of actions. Each action has a pre-condition and effect; the pre-condition must be satisfied for the action to be applicable, and if you apply the action, then a certain other facts can get added to the knowledge base or detected from the knowledge base. So, that is the effect of the action, right?

(Refer Slide Time: 03:45)



And then, we had seen that a plan is represented as a set of plan steps, and there are there is some causal ordering and also some general ordering between the steps of the planning.

(Refer Slide Time: 04:41)

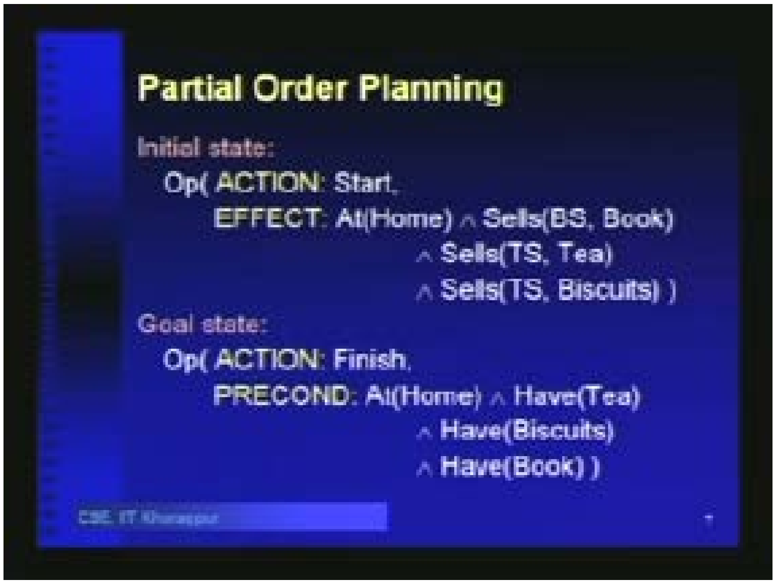


So, these are this and In addition, there is there are some bindings, because the actions themselves can have variables. For example, we can have at x, and then the variable x has to be bound to something which you already have achieved. Suppose you have achieved at tea stall, then, you can use tea stall to bind at x, which could be the pre-condition for sells x something, right? And then, we have a set of causal links, which are introduced when the final state S dash has a pre-condition c, and the action actually satisfies that pre-condition, right? So, these are the causal links, right?

So, what we will do is, we will study the partial order planning algorithm through 1 example, namely, the example of where we have to fetch the tea, the biscuits, and the book. So, the initial state so All this modeling that I am showing here is in strips. The initial step is the start action; recall that in the start action, we do not have any pre-condition, but we can have a set of effects. In this initial state, the action name is start and we have the following effect that we have at home- we are given that BS stands for book stall we are given that book stall sells books and tea stall sells tea and tea stall also sells biscuits

This is what is given to us, and we represent that as the effects of the start action. All that is initially given to you are part of the start action; they are part of the effects of the start action, and in addition to this, we have certain other actions which are given to you. The goal state is 1 where, in a goal state, we will not have any effect; we will only have the pre-condition. And what is the pre-condition? that we We will have the pre-condition that we will have is we are at home, we have tea, have biscuits and have book, right? This is represented by the finish action. For those who have just joined us, what we are doing is, we are studying the partial order planning problem.

(Refer Slide Time: 07:11)

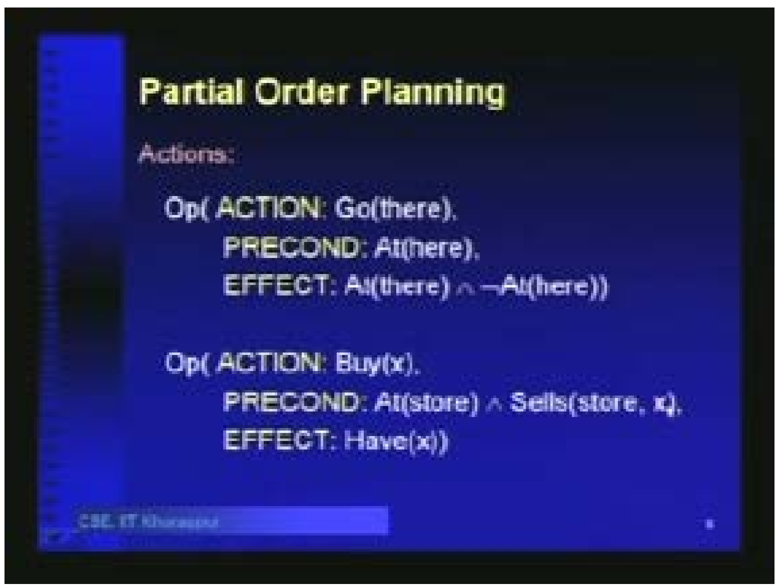


The partial order planning is an algorithm which has been developed quite sometime back, and we are modeling the problem of fetching tea biscuits and book as the example over which we will demonstrate the partial order planning algorithm. So, we have just about stated, by describing the start action and the finish action; the start action has the initial set of propositions, as the effect and the finish action has whatever goal that we want to reach. That is the pre-condition of the finish action. Now, in addition to these 2 steps in the plan, any partial order planning algorithm will start with an initial plan, which consists of the start action and the finish action.

And we will also be given a set of actions that we can apply and these actions, for this problem, are as follows: we have the action, go there, right? Now, there is a variable, so, whenever we have apply this action on a state, there will have to be bound to something; some place which is there in the knowledge base at that point of time. The pre-condition is at here, and the effect is that we are no longer at here and we are at there, so this there and here, are variables. I could have written x and y also. The second action that is given to us is buy x. How do we buy x? We can buy x if we are at a store and sells store x. Now, again, store is just a name that I have used to make it friendly.

I mean, you could use at y and sells y x; I mean; this is not a specific store that I am talking about; it is just a variable at y and sells y x. This is the pre-condition, and the effect is that have x, so the buy action requires that you are at the place which sells x and the effect of the action is that you have x. If you wrote an action called steal x, then you could also about write action steal x, then, at store sells store x and have x. You could make the problem more complex by adding things like whether you have money or not right.

(Refer Slide Time: 09:03)



That is where the difference between buy x and steal x will come. So, if you have have money as the pre-condition, then, the action is buy x, and in the pre-condition for the steal x, you need not have the have money predicate. Before we go into the algorithm, let me quickly take an example; take the example that we have at hand and see how partial order planning will work. What we are going to do is, we will start with the start action and the finish action; at the start action- text please- we will write start- this is the action. So, the graph that I am drawing is s; the plan that we write strips in the form of- the plan: recall that we had given a structure of a plan in strips.

Let me quickly go back- I think I have it in the previous lecture. Anyway, it is just the initial set of actions and the initial set of steps- the set of links and the set of orderings. We have initially, the start action, and we have the finish action here- these are the 2 steps which will always be there in the plan and the effect of the start action is following: we have at home and so, this is what is given as the effect of the start action, and what we need to achieve is the pre-condition of the finish, which says the things that we have to have is have book, have tea, have biscuits, and last but not the least we have to be back at home, so, not to get lost after buying all these things, right?

Now, let us see that how does the partial order planning algorithm- so, this is the initial plan and the initial plan will also contain 1 ordering which we will indicate by a dotted line between the start and the finish. The initial ordering link will be between the start and the finish. This is the initial plan. Now, what we will do in every step is, we will examine that which of the pre-conditions of the steps that has not been achieved. This 1 does not have any pre-conditions; this 1 does have pre-conditions and namely, these are the pre-conditions and none of these have been achieved; only at home has been achieved, but we still do not know whether it is directly from here.

Have book, have tea, have biscuits- they have not been achieved, and we will say that at home is achieved only when there is a causal link from somebody who is producing at home to this step, right? As long as we do not have the causal link, we do not know who is going to give us this at home, so we will consider that it is not yet achieved. Now, we start with have book and then, we examine the set of actions that are given to us and see which of these actions can produce have book.

Buy book can produce have book, because if we look at the set of actions here, out of these 2 actions, have is there in the effect of buy, so we try buy. But then here, we have to bind x; we have to bind x to what? To book, so it is not just applying the action, but also binding x to book. We put that action here, so let us put that step buy book. Basically, what is happening here is that we have substitute x by book; that gives us this plan, this step. And then, we can say that this have book is an effect of buy book, so we have a causal link here. And what is the pre-condition of buy book? The pre-condition of buy book is at x and sells x book.

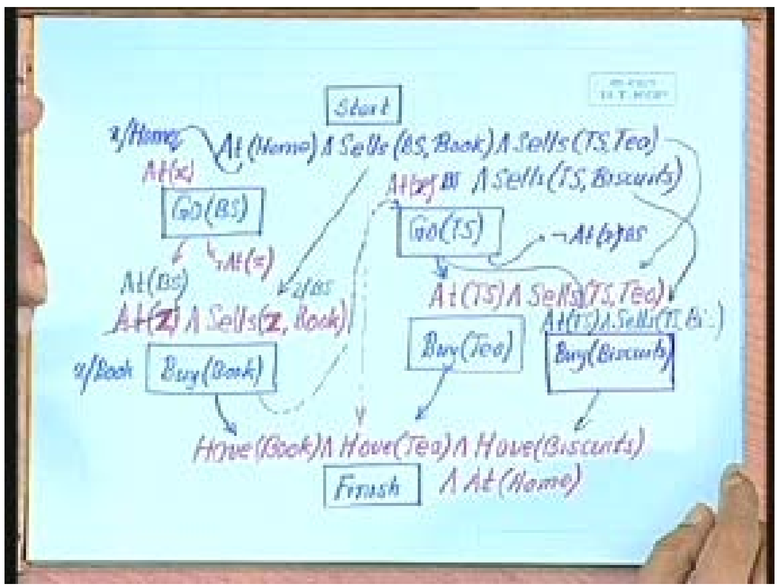
Actually, let me call this z, because not to confuse with this x; this x was the other argument- this x was the other argument- it was this argument, because this argument will get replaced by this. So, we need a z to bind to this and we need 2 steps, which is going to give us that at z and sell z book. Now, we see that the start action has sells book, store book, so we can replace this z by book store. What we will have is, we have this causal link, but z replaced by bookstore, but because this is bound like this, so this z will also become bookstore. So, we will require- this is going to get replaced by at book store, right?

Now, the moment we put this causal link, we are saying that okay, this sells is being given by this; this pre-condition is being satisfied by this. So, this is the causal link, then we can say that this has been achieved; so now, it has been achieved and this 1 is yet to be achieved, because we do not have this at. Then, we see that who can give us at of something? Go can give us; go has its effect at of something, right? So, we look at the go action and we put go of bookstore here, and this go of book store can achieve this at of bookstore, right? But remember that it also produces as a side effect; not of at, of wherever it was originated, so, not of at of say, let us call it x.

And the pre-condition of this is that at x. Now, let us see who can give us at x; another go could have given us at x and also this can give us at x, right? Let us say that we put this causal link to achieve this, and so, for this, the x gets replaced by home, which means that the side effect of this 1 is not of at home. Now, see, in this whole sequence, I have specifically chosen the ones which are going to work in general. The planner can also choose other actions, like for example, when you have this go of bookstore, it can try putting another go before this.

And then, we say that we go to the book store from home, by applying this link; it can come from some other place also- it is not necessarily that it has to go to the book store from home. So, there is an element of search that is there built into this, which I am not explicitly demonstrating here, but you should realize that it is going to try out these different things and we try to achieve that all the unsatisfied pre-conditions. For example, now we have have tea, have biscuits and at home; these 3 are not yet achieved. So, again, we look in a similar way and now we will come into some more detailed problems.

(Refer Slide Time: 30:36)



So, we are going to look at have tea; let us see what happens in have tea- so, we need buy tea just like we ((20:44)) that have book can only be produced by the buy action. Similarly, have tea can only be produced by the buy action, so we are not searching everything; we are narrowing down our search by checking what we have to find. Let us put buy of tea and buy tea will achieve this, but then we need- I am not putting the bindings once again like we did here, so, I am just creating a short cut here and straightaway writing that I have at of tea stall and sells tea stall tea, and this sells tea stall tea is obtained from here and at TS will still have to be obtained.

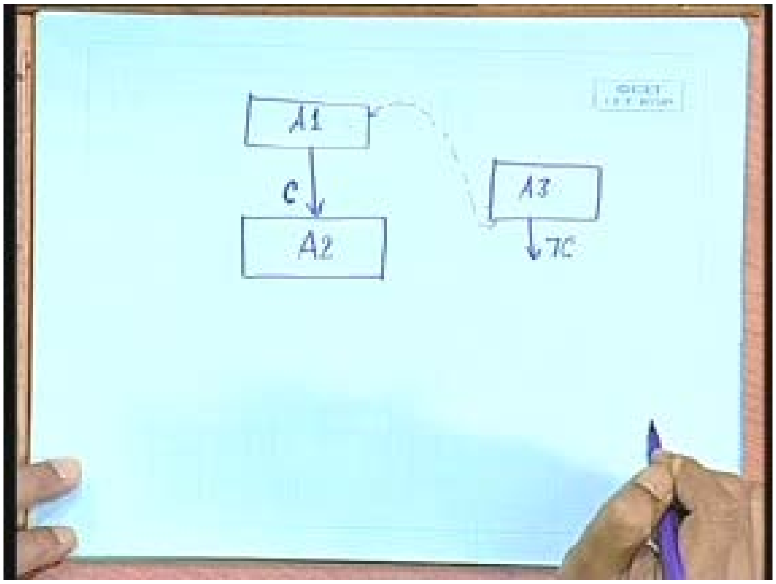
Now, let us see now, we will have an interesting scenario, so again, to have at tea stall, we have to go to tea stall, so, we have go tea stall, because the at is only produced either- it is already there or it is being produced by the go- there is no other way to get an at and at tea stall was not there anywhere. So, to produce that, we have to use go tea stall; that is clear, and that is going to give us this. But now, we also have at this, at the pre-condition of this, we have an at x. Now, can we put a link from at home to at x? (Students speaking).. See, there is a problem here.

We cannot put this here, because if you apply 1 of these- if you apply this action, then, not at home will come into effect. And when you are not at home, then, you cannot use at home again, so, you cannot put because you have this link already here; you cannot put this link here. In other words, this action threatens the pre-condition of the other action and this action threatens the pre-condition of this and this threatens the pre-condition of this, because this has the pre-condition at home and this fellow, if you bind it to home, then it is going to produce not at home.

And not at home is the pre-condition that conflicts with this one. Are you getting what I am saying? Therefore, we have to have some formalism to identify that this kind of threatening is taking place, and we have to serialize them. We have to say so, if you have this problem, then, either we have to do this action before at home, which is not possible because at home is being generated by the start, and we cannot have anything before start. We have to do this after go BS, right, so, we can bind this x with this bookstall. (Students speaking). Let me create a more generic scenario. I have some action; I have some action A, which has the pre-condition of C, which is being generated by some action- let us call this A1, let us call this A2, right?

And then, I have this step, that I have inserted another planning step, which is A3, which produces not C as its effect. Now, if you do A3 between A1 and A2, then, we have a problem. The conclusion is that either we have to do A3 before A1, or we have to do A3 after A2. If you have a cause-effect relationship by virtue of a pre-condition, any conflicting action which produces the negation of the pre-condition cannot be done in between these 2. If this was not producing not C, if it was producing something else, then, there is no problem.

(Refer Slide Time: 26:51)



If you have this kind of scenario, then we will put those additional ordering constraints, so, we will either put an ordering constraint like this, which says that A3 has to be done before A1, or we will not put this and instead put an ordering constraint like this, which says that A3 has to be done after A2. It is by that reasoning that we decide that we will have to do this after this, go BS otherwise, we will have a problem, but then, it is still not enough, because this go is going to produce as a side effect not of at x. Suppose I bind this to BS; then, it will produce not of at BS. Now, if you do it between these 2, then again, we will have a problem, because to buy the book, we have to be at BS.

This is the case that I was showing that this produces at BS, which is being used by this, which is a pre-condition of this and this action produces not of at BS. We have to postpone it further and therefore, we have to postpone it right up to here, so, we will put an ordering link like this, which goes from buy book to go TS and we will substitute x by BS.

Now, what do we have? See, what is the plan that is forming up out this? If you just follow the causal links and you do a topological ordering based on the causal links, then you will see that what we are having here is that we first go to the book stall, then we buy the book, then we go to tea stall, then we buy the tea. Now, we have still not done anything about buying the biscuits, and the space is already cluttered, so, let me try to fit in what we do about buying biscuits. So, we will put the action buy biscuits here.

This buy biscuits is going to give me the have biscuits. Now, what does buy biscuits require? It requires at tea stall and sells tea stall biscuits. We will have at tea stall and sells tea stall, and then biscuits. This at tea stall is achieved by this. So, we can have the link from this to this- sells tea stall biscuits is already given here, so we can have that. After this, we have achieved everything except at home. Let me redraw some part of this, so that we can proceed to that, because again, we will have some problems when we try to insert the at, so, we have to go home, but go home from where?

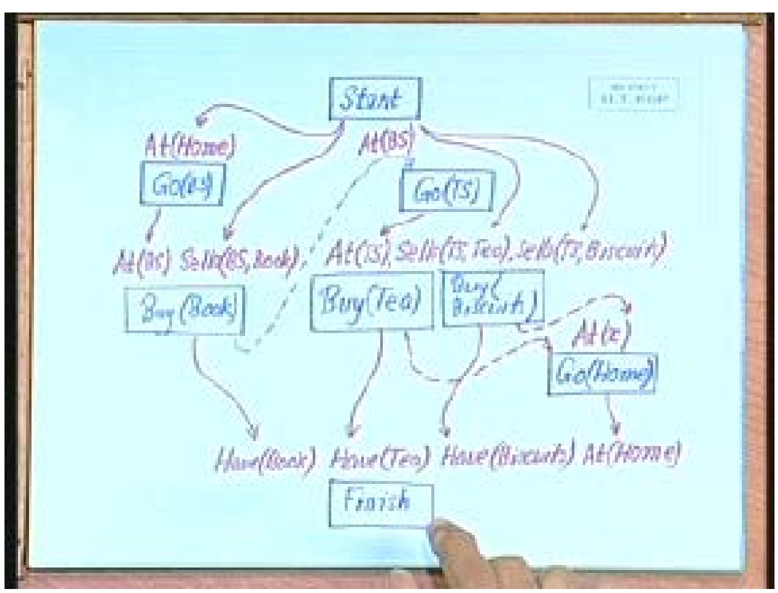
If we go home from book stall, then, we will not have bought the tea and biscuits. Again, those orderings will come into play, so, let me just clean up this part a little bit and then we will go into that. What we have here is the plan which we have so far is- we have start, then we have go book stall, we have buy book, and we have finish here. Then, we have go tea stall, where we buy tea, buy biscuits and we needed at home here, which is being given by this causal link. This produces at BS and we needed sells BS book, which is given from here. Then, we wanted at TS sells TS tea sells TS- this is from here and the remaining thing are from start.

That achieves all this, and we needed have book here, which comes from here; we needed have tea, which comes from here; we needed have biscuits, they are from here; and we need at home, which is not yet achieved. And we also show that in addition to these links, we have the ordering link, which goes from here to here. Because the go action threatens the at BS, so that is why we had to put this before this or after this. Now, having done this, let us now see how we achieve at home; if we want to achieve at home from here, then, the problem will be that this is going to produce not at home. There is a conflict between this causal link and this so, and since we cannot put it before the start, so therefore, we have to somehow go home.

We have applied the new step, which is go home and that go home will achieve this. Now, when we go home, so, we must be at x some place, so, we have to decide that this at of x that we have here- who will give as this? Again, can we have at home? No, because this is going to conflict with this one, which produces not at home. It cannot be at book stall, because of, again, this conflict that this requires at book stall and if we use this first, then, we will not be add book stall- see this here, when we bound this thing, then, we had actually put at book stall; the at x of here was replaced by book stall.

Therefore, we cannot have that also. Then, we try at tea stall, but then again, at tea stall is threatening this link. So, we can have at tea stall, but because of this link which threatens this, we have to have a causal link from this ordering link. Sorry, from this to this and also an ordering link from this to this and these 2 are because both of these require at tea stall. It threatens this link; it also threatens the link to the at tea stall of this.

(Refer Slide Time: 40:17)



This is where partial order planning will stop. This is the thing that it will produce; this does not give us a total ordering among the actions. For example, the sequence in which you buy tea and biscuits; this sequence is not given by this plan, so you have the freedom. You either use this action first and then this action, or this action followed by this. The relative ordering between these 2 actions is not given, so you are free to use any 1 of them. That is why we call it the partial order planning, because it gives you the set of actions that you need to take and it gives you as partial order, describing the sequence in which the actions have to be taken.

(Students speaking). Yes. Suppose before getting this go book stall; suppose we had done this, we had done up to this part- let us say, then, to achieve this at tea stall, you could have used the go tea stall first and then gone from to this also. But what you are probably trying to say is that the partial order does not capture that you could have gone to- you could have done it the other way also. Yes, we could have put that flexibility also, but then what we would have to do is, when you do this ordering, then you have to maintain both the edges.

So, either we do this like this, or we do it like this and keep both the edges, and then you have to say that its mutex between these 2 edges, both edges should not be there; only 1 of them has to be there. That would have captured the whole gamut of the solution space, but that again, is a combinatorial problem. Because you will end up having a plan where there are a set of mutex pairs of edges, so, this could have been done before this and then resolving all of those, we will again have another complexity. So, what we attempt to do is, we fix 1 of the- we try 1 of the orderings first.

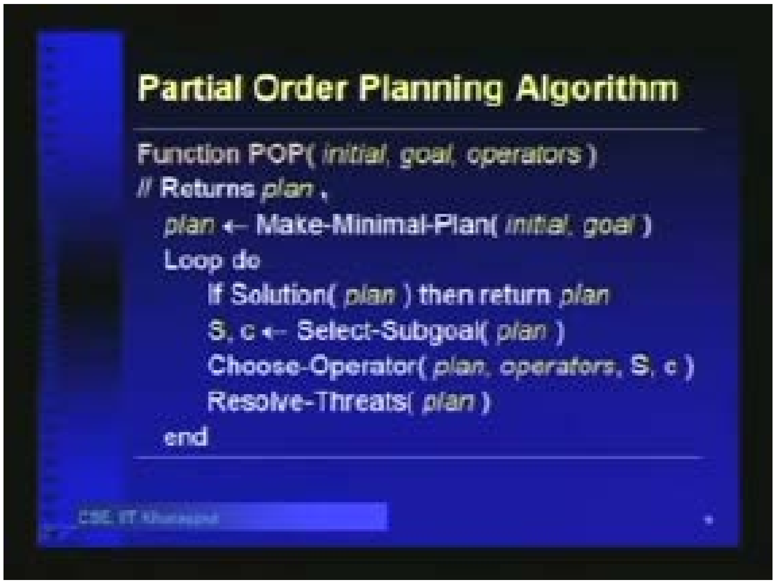
We say that okay, let us say, try deferring this, so, we put go TS after buy book and not the other way around. If by doing this, we are unable to produce the plan, then, we will backtrack and change; that is where we are taking care of the combinatorial problem which is underlying this; we try 1 after the other. As a result, the final plan that we have- it has some partial order between independent actions, but once 1 is threatening the other, we just forcibly serialize it. They mean 1 order. (Students speaking). After you have this plan, then, see this- the partial order planning algorithm guarantee that this graph that you have does not have any cycles.

That is, the algorithm will check for that and we will only add link, provided it does not have produce a cycle. And so therefore, if you do a topological ordering on this, that will give you 1 sequence of execution of the actions that will achieve the desired goal. If you and topological ordering is not unique, so, each topological ordering of these actions is going to give you a different plan, a different sequence of actions that achieves the goal. (Students speaking). Well, you might end up finding a plan like that; you might end up finding a plan where you first go to the tea stall, buy tea, then go to the book stall, buy book, then go to the tea stall, buy biscuits and then go home.

As of now, we are not even considering the problem of finding the optimal plan; we are just satisfied by finding any plan. But yes, these optimizations has been studied and in the later algorithms that I will talk about, they will talk about finding the shortest length plan, the minimum number of actions, etc. More recently, people have also talked about planning with temporal goals and planning with costs associated with the actions.

So that you find out a plan which minimizes the total cost. We will now quickly look at the pseudo code of the partial order planning algorithm- the pseudo code is directly what we have described just now. This is the partial order planning algorithm; it returns a plan- a plan is what? It is the graph like this, which has a set of steps, a partial order between the steps which is given in terms of the causal links, and the causal links and the orderings. And the set of planning steps that is the final set of solution that is given.

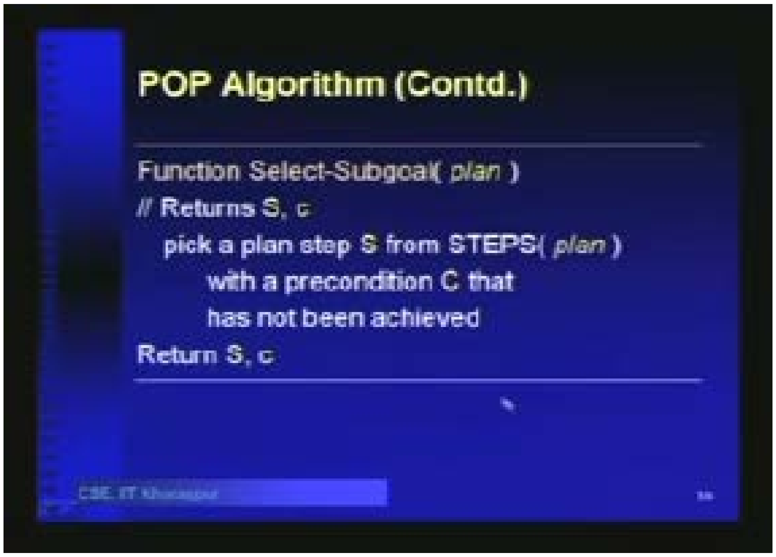
(Refer Slide Time: 43:24)



The first step is calling this- make minimal plan with the initial and the goal, so, that is the start state and the goal state. This is the start plan that we then- we check if the plan that we have here is a solution. When is it a solution? When all the pre-conditions of all the actions have been achieved. All the pre-conditions of all the steps in the plan have been achieved, then return plan. Otherwise, we select a sub-goal, which means it is a pre-condition of some step which as not yet been achieved. That is a sub-goal. And then, we choose an operator which achieves that sub-goal. Now, operator can be of 2 types- it could be some existing step in the plan, like for example, when we looked at buy biscuits, we found that this had already achieved the at TS.

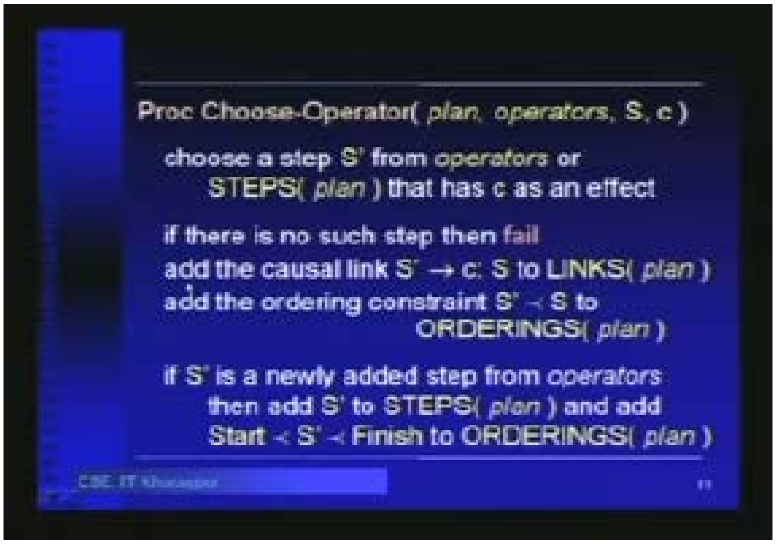
That was already achieved as part of another step, so, we did not have to add any more steps, like go to tea stall again. And the operator could also be a new action that comes in as in additional step into the plan. And then, after we choose the operator, the final step is to resolve threats which will lead to addition of certain ordering links. Now, let us look at what is a select sub-goal, pick a plan step S from steps plan with a pre-condition C that has not been achieved and return C. So, that is simple.

(Refer Slide Time: 45:23)



Then, choose operator is choose a step S dash from operators or steps of plan. Steps of plan is the existing steps that are already there in the plan that has C as an effect. If there is no such step, then we fail and we backtrack to some previous choice point. Choice points could be points where we decided previous orderings, it could be points where we choose previous actions.

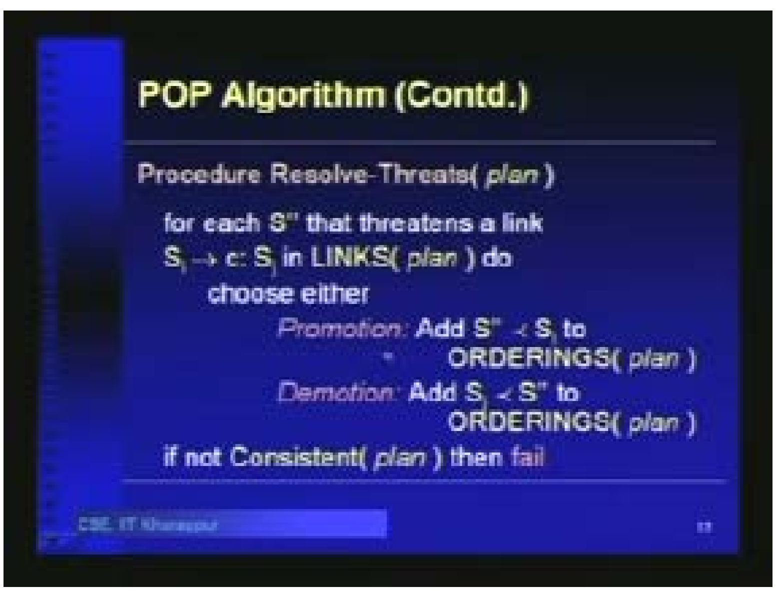
(Refer Slide Time: 46:13)



Then, add the causal links, otherwise, we add the causal links S dash to S to the LINKS plan. Plan has 1 attribute called links, so, we just add it to that. That is the set of links and because every causal link is also an ordering constant, so, we also add S dash precedes S. If S dash is a newly added step from operators, which means it was not at previous step, then, we add S dash to steps of plan and add this ordering that S dash should be between start and finish. See, this is very important because later on, when you resolve threats, you will know that S dash cannot be done before start and it cannot be done after finish. So, in order to preclude those kinds of scenarios, we are apriori adding some orderings to put S dash between start and finish.

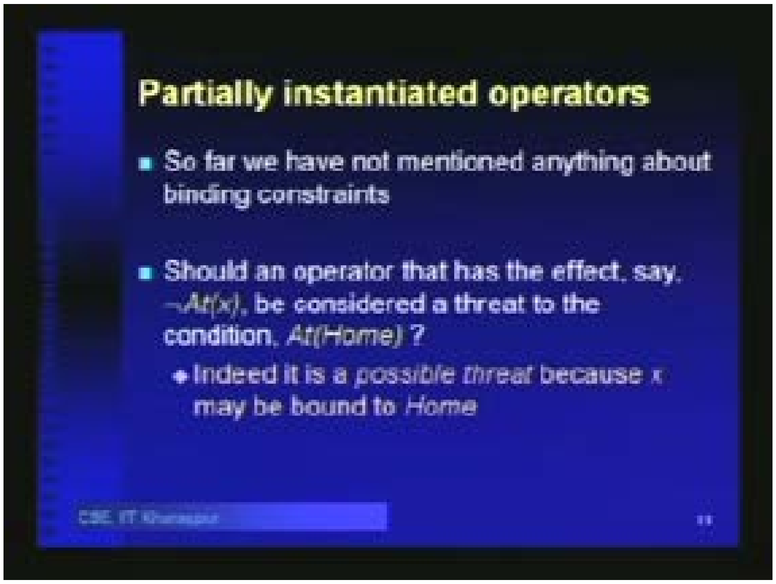
Now, what we do is- resolve threats for each S dash S double dash that threatens a link. So, this is some step which threatens the link, so, why does it threaten? Because this action is producing not of C. If this produces not of C, then, that is going to threaten this transition from Si to Sj. In that case, we either promote, which means that we will decide to do S double dash before Si, or we will demote and decide to do S double dash after Sj. We will choose 1 of these and then, we will check whether the plan is consistent, in the sense that whether we are introducing cycles or not. If we are not introducing cycles, then fine, you just go ahead.

(Refer Slide Time: 47:47)



Now, well, I have already mentioned some things about binding constraints, so this is there in the original slide, just ignore it. Now, here is a question- that suppose that an operator has the effect not at x. Should it be considered a threat to the condition at home?

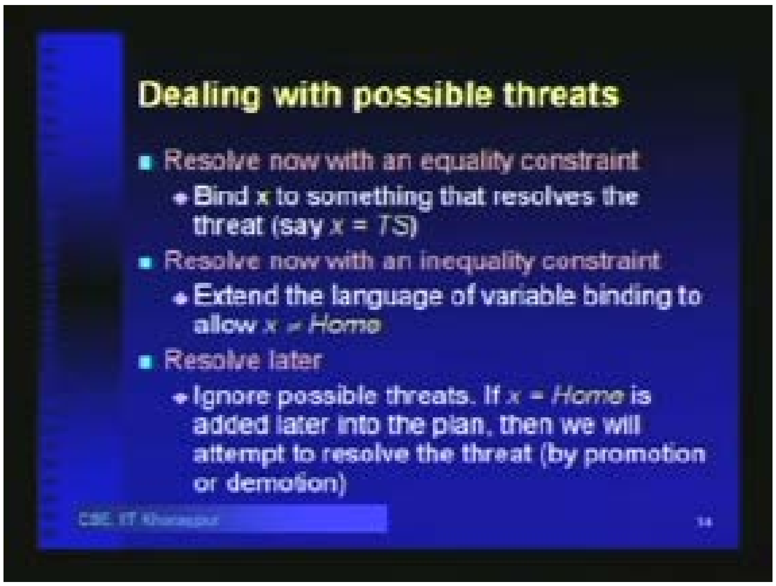
(Refer Slide Time: 50:10)



Now, in this particular example that we had, everything was bound, so, when we use go of x here, because we needed at of BS, we just bound x to BS. Just by looking at what we want to achieve, we can substitute the variable and put the appropriate thing, but we cannot do it always. And why can we not do it always? Because recall that the goal can have a variable. If the goal had have of y and then, you have to put something in order to have of y, you have to buy y, but at that point of time, you still do not know what value it will bind to. Suppose we have a scenario where the operator has the effect of not of at x and x is not yet bound; this is possible and we have a condition at home.

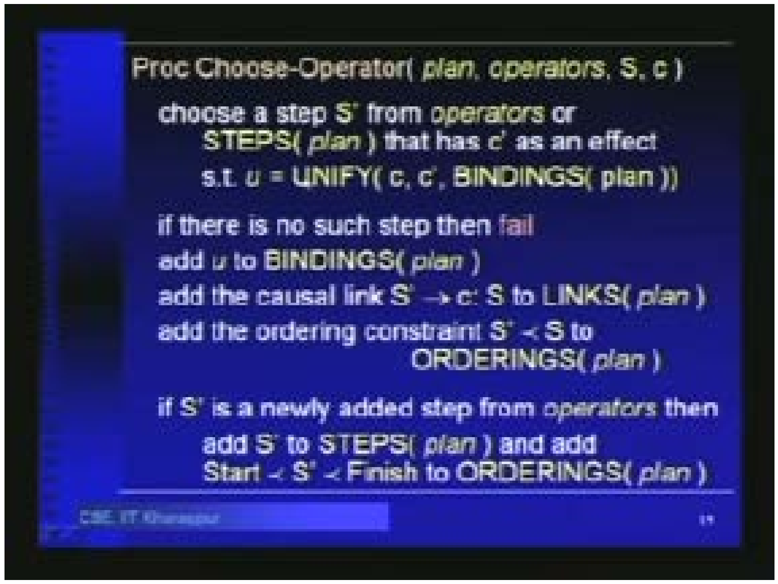
Should we consider this as a threat? This will become a threat if later on, x gets bound to home. If x does not get bound to home, if x gets bound to tea stall, we do not have a problem. This is a possible threat. What do we do with these possible threats? There are 3 ways of dealing with the possible threats- 1 is that we resolve it now, with an equality constraint: we say that bind x to something. If you bind x to something other than home, then, we are safe. That is 1 way of doing it. But bind x to what we do not know- so that is not always a very good idea; you have to do a lot of backtracking. The other thing that we could do is resolve now with an inequality constraint- so, we say, that add a constraint that x cannot get bound to home.

(Refer Slide Time: 51:13)



The third is resolve later- do not do anything right now; go ahead with the planning and then, if at some point of time, x gets bound to home, we will try to put in additional constraints to promote or demote this action, so that the threat is resolved. And this third 1 is the 1 which is usually the most POPular thing. If we do that, then the choose operator and the resolve threats will slightly change. Let us see how it will change. First of all, when we choose operator, because we have bindings- now, we have to see what we want and find an operator which can bind with that.

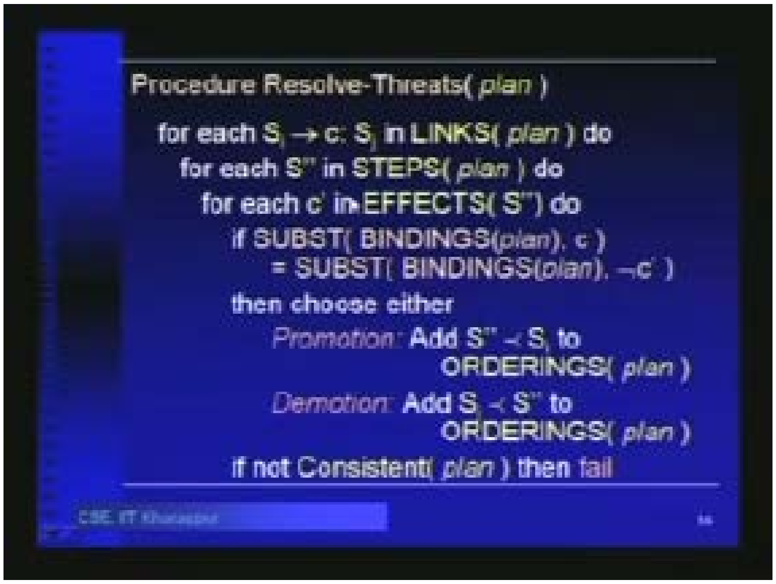
(Refer Slide Time: 52:47)



Now, this is all written in formal languages. What effectively it means is that if you have- coming back to our example, if you had at x here and this is can and you require at BS and this produces at of x, then, basically, we say that we apply this action, provided that we can bind x to BS, which in this case we can. This and this at x and this at BS will unify. Remember unification, in first order logic? We will unify, provided we substitute x with BS. That is simply what it means. In some cases, these predicates can be more complicated. That is why we formally say that the effect of the action and the pre-condition that you require should unify by a mutual substitution of the variables.

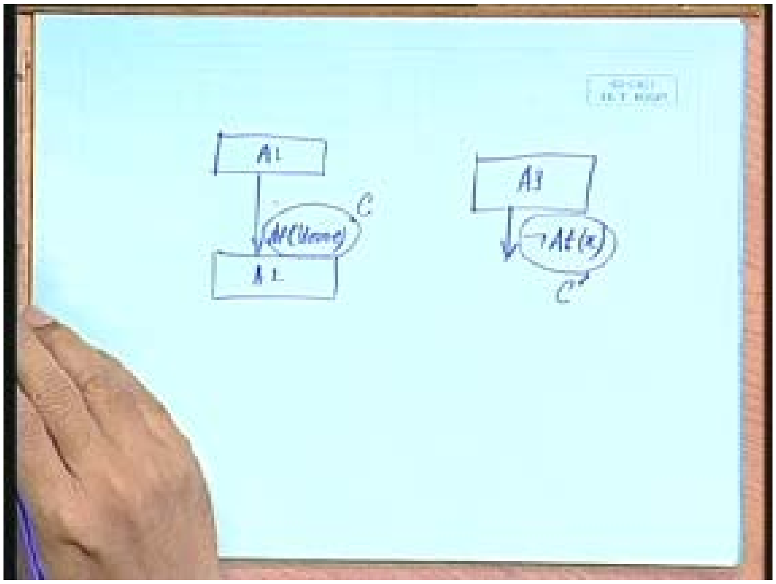
That is what we have here, that you choose the operator such that it unifies with what you want. If it does, then, we add those unifications into bindings. So, bindings is another set that we had in the plan structure, which I had not described before. These bindings will just keep the these bindings, like x is getting replaced by BS and so on; remaining part of this choose operator is similar. And in the resolve threats, what we need to check here is that once you add another link or another step in your plan, you have to check whether the bindings that you have in the plan unifies the effects of the 2 plans. So, if it unifies this C with this not of C dash now- are you with me? (Students speaking). No. You are not with me.

(Refer Slide Time: 53:24)



See, what I mean to say is that we had this at of x, and we have this not of at of home. The problem is if x gets bound to home, because then, this will produce at of home and that one, which produces not of at of home will then threaten this. We had this action A1, which requires, say, at home and we have this A3, which produces not of at x. Our problem is that if x gets bound to home, then, this action will threaten this transition.

(Refer Slide Time: 55:11)



So, what we are doing here is, we are checking whether this binds with the not of this, so this is our C and this is the not of C dash. If these 2 bind, which means, see, what is not of C dash? It is at x. Does at x bind with at home? And yes, it does. Then, we have the problem. That is what we are checking here in this; that whether this C binds with the not of this C, where this unifies with this. The substitution that we have in the bindings of the plan that unifies this C and this not of C dash- if that happens, then, that is the threat; that is the valid threat, so we either promote it or demote it. Now, is it somewhat clear now? (Students speaking). Somewhat, right? I think what we need to do is, we must take 1 day. With this, we come to the end of this lecture.