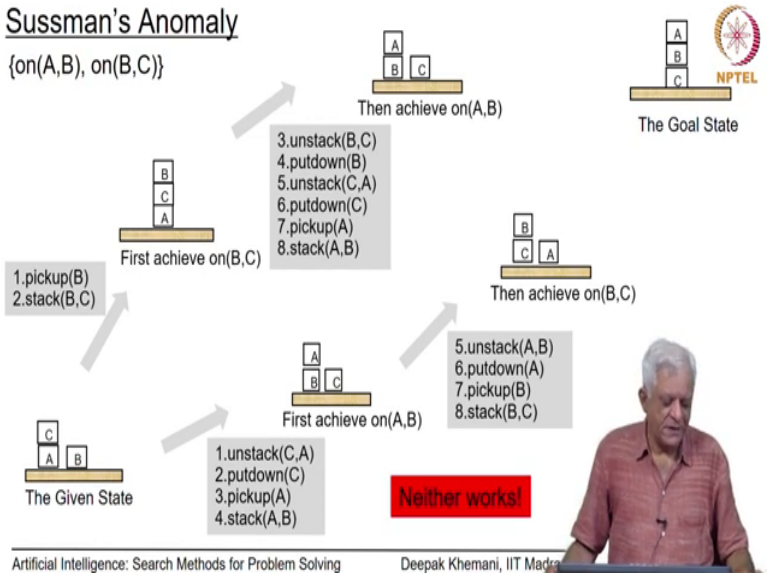
Artificial Intelligence: Search Methods for Problem Solving
Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Chapter – 07 and 10
A First Course in Artificial Intelligence
Lecture – 63
Plan Space Planning (PSP)

(Refer Slide Time: 00:14)

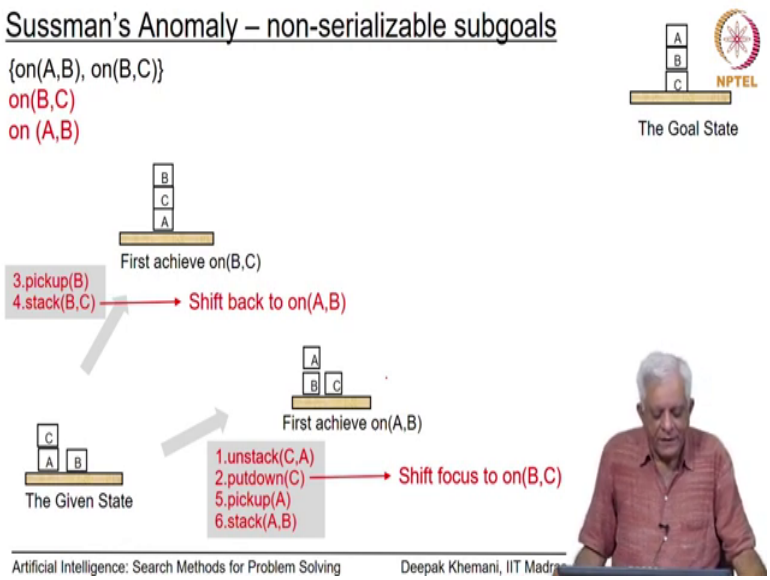


Welcome back. Let us continue study of planning algorithms. Last time, we looked at goal stack planning. And which is an algorithm, which breaks up a goal into sub goals and I used to solve them in serial order. And then, we encountered this problem that there may be some goals which cannot be broken down into serializable sub goals.

And a very simple and elegant example of that was Sussman's anomaly which we saw here. Where the initial state was three blocks; C on A and B on the table and the goal state was to get A on B on C.

And whichever order of the two goals that we prefer or choose that either on B, C or on A, B, whichever we chose first, it turned out that we did not; by the time, we solve the second one we could not achieve the final goal and moreover, the plans were much longer. We saw that the optimal plan in Sussman's anomaly would have been a six-step plan like this, but the algorithm was unable to find it.

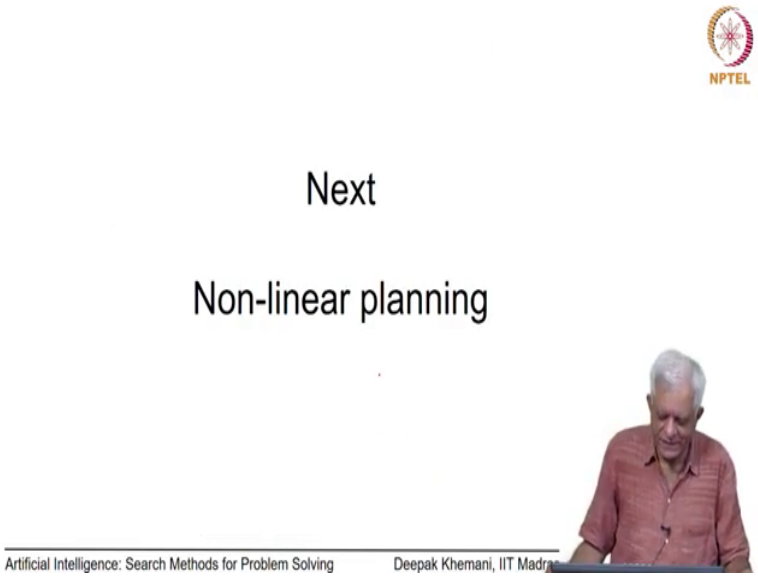
(Refer Slide Time: 01:33)



Ah Then, we observed that if it had started off by searching for one sub goal which is on A, B and then, half way shifted its attention to on B, C solved on B, C and then, came back to on A, B; then it would have found the sixth step plan. But the algorithm does not allow that. It

says the algorithm goal stack planning says solve one sub goal and after you have finished solving it, solve the next sub goal; whereas, that we have seen does not result in the optimal plan.

(Refer Slide Time: 02:08)



So, our next approach is to look at an approach of planning which is called non-linear planning. It is got many names non-linear planning, plan space planning that is what we will start by calling it. It is also called partial order planning, also called least commitment planning. We will look at these aspects as to why all this different names are there as we go along.

(Refer Slide Time: 02:33)

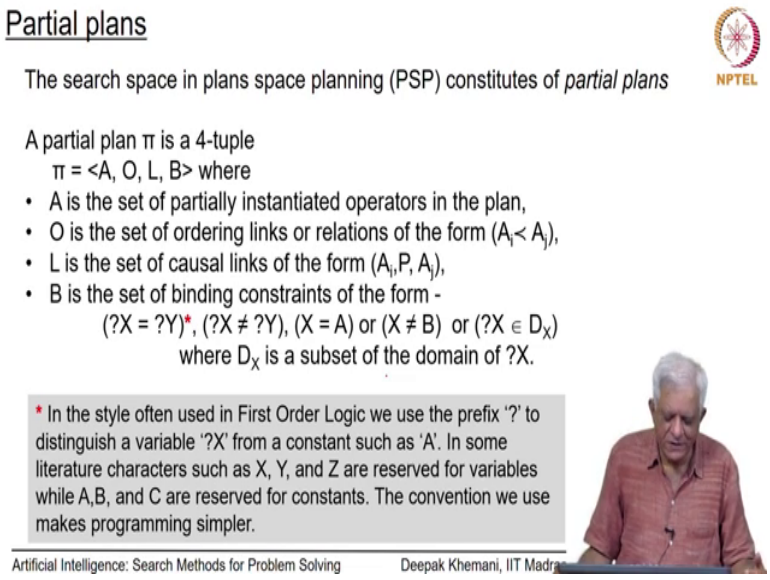
So, let us start with the notion of plan space planning. So, far our planning approaches have reason with states essentially. So, you had the start state, you had the goal description which amounted to a set of states and we had operators which took you from state to state.

So, it was basically looking at a state and a goal description and the algorithm terminated, if the states satisfies the goal description. In some sense, that the goal predicates or the goals were elements of the final state; otherwise, it searches over the otherwise the algorithm searched over the state space looking for actions to add to the plan.

And the plan was kind of constructed as a byproduct that as you search the state space, you started assembling the plan depending on which algorithm either from the first action or the last action, but eventually it was a kind of a byproduct of the plan.

The alternate view that we want to look at now consider is that the why not search in the space of all possible plans essentially. And see if that gives us more flexibility in terms of the kind of plans that we can discover, we will call this approach as plan space planning and let us start off by describing it first.

(Refer Slide Time: 04:00)



So, the search space in plans space planning consist of what are called as partial plans. A partial plan; so, it is a well-defined structure, it is got importance of its own in plan space planning. In fact, we only talk about partial plans in plans space planning, we never talk of states at all.

Of course, we do talk of predicates that are used to define the start state and the goal state. But never do we represent the state as a explicit object in our program; whereas, as some

people would say partial plan is a first class object in the planning approach that we are following now and it can be passed on as parameter and manipulated and things like that.

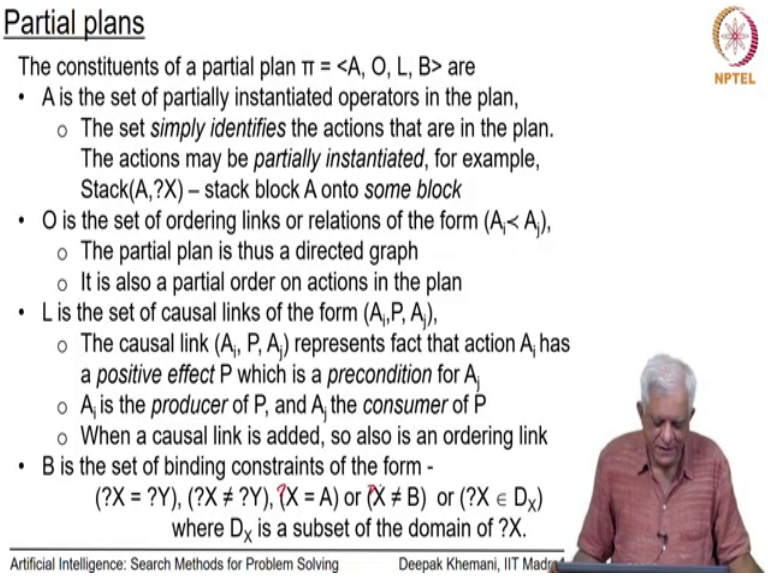
So, it a partial plan is a four tuple, which is made up of four sets which we will call as A, O, L and B; where, A is a set of partially instantiated operators in the plan. So, we are simply specifying as to what are the actions that go into the plan.

We separately assert as to which actions come before which other actions and that is specified by something called ordering links. So, here we have an example of A i being before A j or A i preceding A j and this is said explicitly in the partial plan. Then, there is a set of causal links and the causal links are as follows that there are two actions A i, P and A j.

And we say that A i produces P and its consumed by A j, we will just look at this in a little bit more detail in the moment. And finally, there is a set B which is a set of binding constraints which are also formed like X is not equal to Y, X equal to Y or X not equal to A or X equal to B and so on essentially. I should have added question mark X here because I want to use the convention that anything before anything which has a question mark as the prefix is a variable and anything which does not is a constant.

Because they are two kinds of things that one deals within logical kind of a frame work. There are some people who say that ok, now let us just take the convention that X, Y, Z are variables and A, B, C are constants. But then, if you have to write a program, you would have to take care of that explicitly; whereas, adopting a naming convention like this is better from the programming perspective.

(Refer Slide Time: 06:51)



So, just let us expand upon the definition of partial plan. The constituents as we said are first is a set A, which is a partially instantiated operators. This set simply identifies as to what are the actions in the plan and the actions may be only partially instantiated.

Remember that we had said that actions are instances of operators have variables or operators are like schemers. Whereas, actions are specific actions which talk about specific objects on manipulating specific objects, but the partially instantiated object like stack A on X says that stack block A on to some block.

And the partial planning algorithm allows you to do that and the plan representation allows you to assert that such an action exist. Now, O is the set of ordering links as we said it says

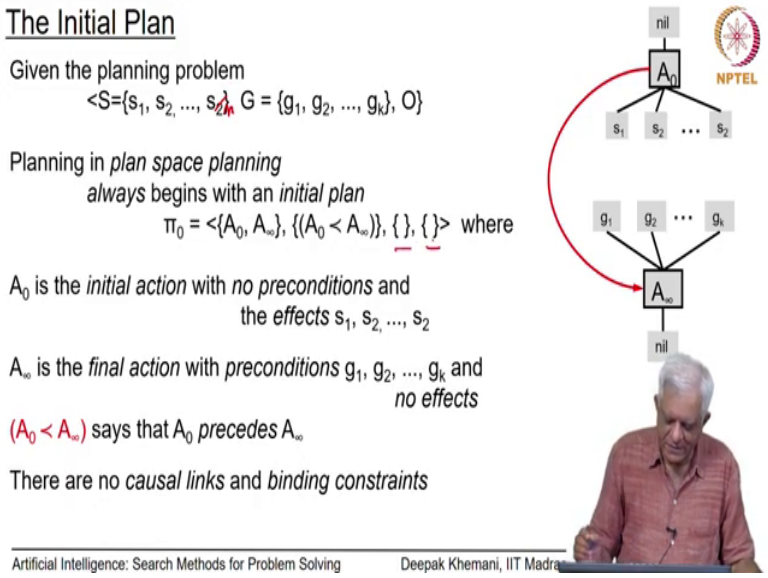
that A i precedes A j in this case. So, that converts a partial plan into a directed graph and we can also see the plan now as a partial ordering on the actions in the plan.

So, every time we introduce this ordering link, we introduce some more order into the plan. The causal link is something that I had skimmed over in the last slide, but here is the complete definition. We say that a tuple made up of A i which is an action P which is a goal or a predicate and A j; which is another action represents the fact that action A i has a positive effect P, which is a precondition for action A j.

So, we also say that A i produces P and A j consumes P. A i is the producer of P and A j is the consumer of P. Whenever we will add a causal link like this into the plan, we would also add at corresponding ordering link; because of the fact that time is owen into this connection.

A i must happen first it must produce P and then, A j will use that as a precondition. So, we also add an ordering link whenever we add a causal link. And as we said B is the set of binding constraints of the form, which are which basically say that variables are equal or not equal or they come from certain domain and things like that.

(Refer Slide Time: 09:28)



Now, when we are doing plans space planning which is also called partial order planning, we start with the given problem as before. So, we have observed this that a planning problem is defined by three things; one is the initial state which must be completely described, one is the whole description which may be partial in nature and the third thing is the set of operators that are available to you for making the plans.

So, this should have been n here. So, planning in plans plan space always begins with something which we call as an initial plan and it is always the same structure. So, pi 0 has two actions all the time which we call as A 0 and A infinity. And it has a ordering constraint which says that A 0 must happen A infinity at it and it has no causal links and it has no binding constraints.

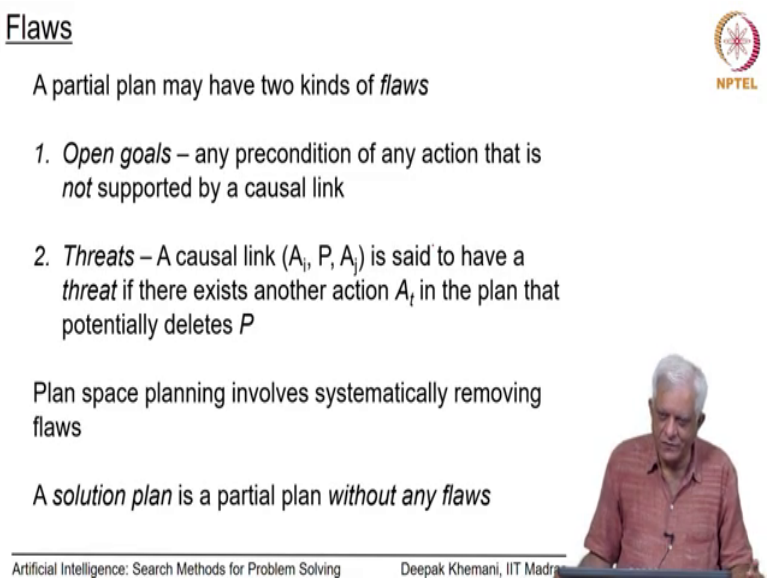
So, A 0 is the initial action which is always present in any partial plan. And it is an action which has it is a very special action which is devised only for partial planning or plan space planning. It has no preconditions and its effects are the goal the predicates of the start state.

In some sense, it is producing the start state essentially. Thus, so, in some sense that it says that this is where your actions must begin must begin. A infinity is the final or the last action; its preconditions are the goal states and it has no effects. So, A 0 produces the start state or the start goals or the start predicates and A infinity consumes them and as we said there is one ordering link here, this says that A 0 must happen before A infinity.

This is the initial plan which the plans space planning algorithm always begins with. The only thing that changes as you define new problems is that the elements s 1, s 2 and so on and g 1, g 2 and so on.

They change as we move to a different problem; but otherwise, the structure is still the same. The initial plan has got two actions; A 0, A infinity and assertion that A 0 must happen before A infinity. There are no causal links and binding constraints as we observed. These two sets are empty.

(Refer Slide Time: 12:12)



Now, there is a notion of something called a flaw in a partial plan. So, a partial plan is called a partial plan because in some sense, it is not a complete plan and if it becomes a complete plan then we call it as solution plan. A partial plan may have two kinds of flaws essentially.

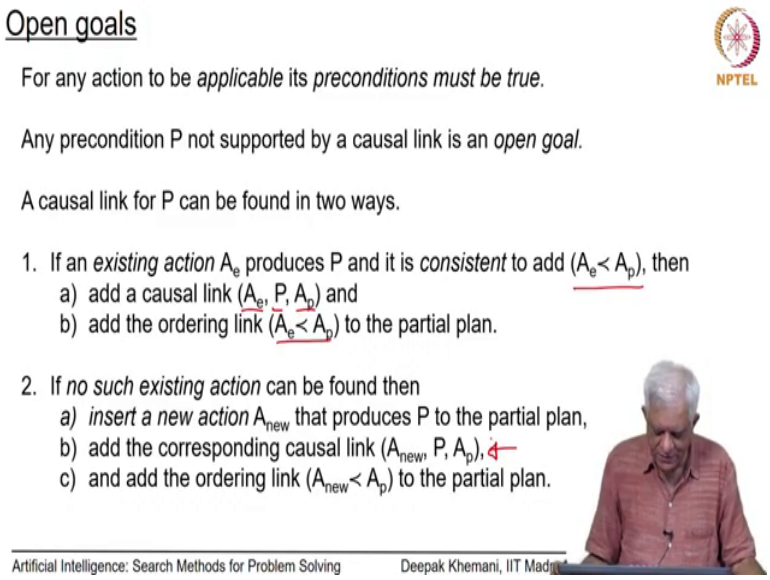
The first kind is called open goals and an open goal is any precondition of any action in the plan that is not supported by a causal link; because you know when you want to execute an action, it must have its preconditions true and we have said that causal links provide the preconditions for actions. So, if there is some precondition of some action which is not supported by a causal link, we call it an open goal.

The other kind of flaw in a plan is as follows that if there is a causal link from action A i to action P j and A i produces P and A j consumes P. We said we say that if this causal link is set to have a threat, if there exist another action A t in the plan such that it can potentially

delete P, which means that A i may produce p, but A t may delete it and therefore, A j will not be able to execute.

We say that A t potentially clobbers the causal link between A i and A j. Now, plan space planning in all its different variations that people have tried is essentially systematically removing the flaws one by one and if you have managed to remove all the flaws, then we say we have a solution plan essentially. So, let us now address this question of how to remove these two kinds of flaws essentially.

(Refer Slide Time: 14:06)



So, the first and the simpler of the flaws is open goals. As we observed for any action to be applicable, its preconditions must be true and the preconditions are provided by a causal link. So, any precondition that is not supported by a causal link is an open goal and we must

somehow cater to that. A causal link can be found in two ways; one is that there is an existing action.

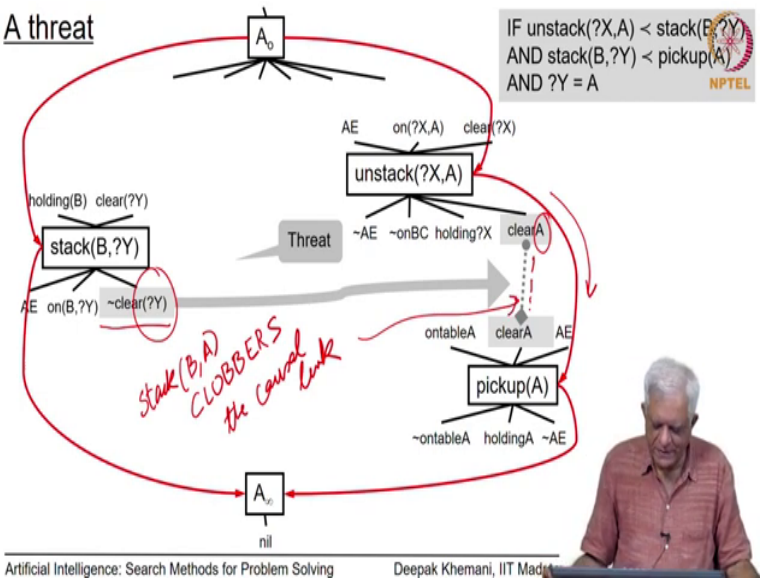
Let us call it e, a existing action A e and that action produces P that we are interested in and it is consistent to add that A i at A e happens before A p. This test of consistency is important because eventually the actions have to happen in some given order. And if you have a ordering which is not consistent with what can be implemented in practice, then you do not have a valid plan.

So, this consistency check is required. Then, the simplest consistency check is that there are no loops essentially in the ordering cycles. So, if an existing action A e produces P, then we can add a causal link to that effect.

So, we are saying A e produces P and that is consumed by A p and we also add the causal, the ordering links in that A e must happen before A p to the partial plan. If there is no such existing action, then we look for a new action. And in this is a way in which new actions are added to the plan essentially.

So, we insert a new action a new and this action also obviously, must produce the predicate P that we are interested in. And we add the corresponding causal link for saying that ok, that open goal that A p had is now supported by this causal link, where a new produces that open that pre condition P. And obviously, we also add the ordering constraint to the partial plan.

(Refer Slide Time: 16:18)



So, let us look at this notion of a threat you know little bit more detail. So, here we have a partial plan which has been constructed so far. We have added two actions here say that unstack something from A and may be that was required. So, that you could clear A, which means maybe there was something on top of A.

And then, you want to clear it and another action pick up A, which follows unstack A; because there is a ordering link between the two essentially. Now, unstack A produces clear A and pick up A consumes clear A. So, there is a causal link which is depicted here in this dashed line essentially.

Now, if there is another action, we introduce at some point and let us say this action says that stack B on to something; stack B on to a variable Y and that means, stack on to something.

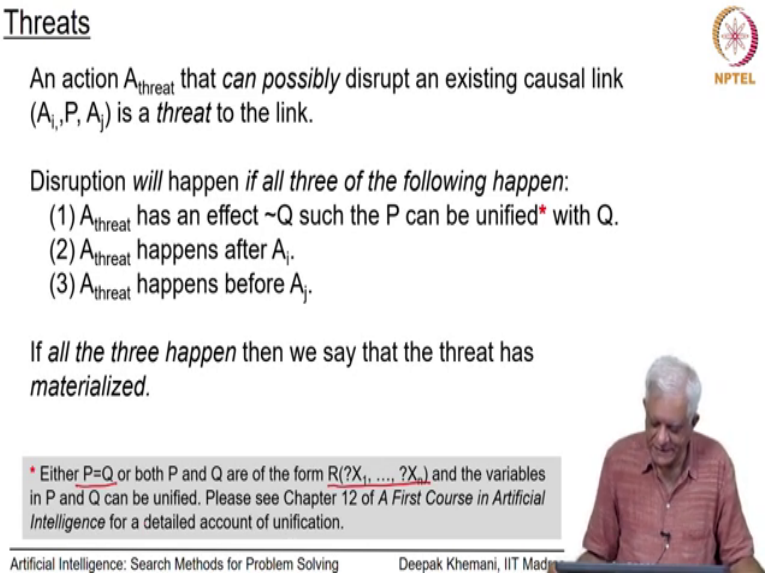
Now, it so turns out that because when you stack B on to something, that something will now become not clear essentially.

Can that something be A? That is the question that we would want to ask. If yes, then that action stack B, Y is a threat to a causal link that we just spoke about. We also say that stack B, Y clobbers the causal link or potentially breaks the causal link essentially. Now, this can happen if there are three things which are true; the first thing is that this action unstack the threatening action stack B must happen after unstack X, A.

So, unstack X, A produces clear A and after that if stack B happens, then it will delete clear A. And because this clear A has to be consumed by pick up A for the threat to materialize, the stack action must happen before that essentially. And thirdly, this variable Y that we have in this action must be bound to the constant A in the causal link essentially.

If all these three things happen, then we say that the threat is materialized. So, we say that is stack B, A CLOBBERS, it is a terminology which was introduced in the 80s the causal link. Which causal link? This is the one that we are talking about essentially and people spoke about de-cobbling and things like that.

(Refer Slide Time: 19:38)



So, let us formalize this notion of a threat. We say that an action A threat that can possibly disrupt an existing causal link given by A i, P, A j; where, A i produces P and A j consumes P. It is a threat. Even if it can possibly do this disruption, it is a threat and we must remove this threat that is the idea of removing flaws from partial plans.

Now, this disruption will actually happen if all the three of the following conditions as we just observed this thing. One is that A threat has an effect not Q such that P and Q can be unified.

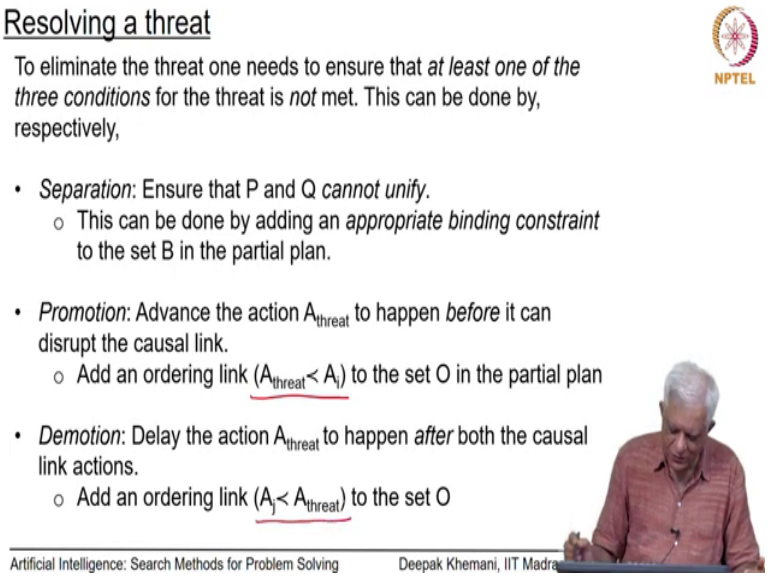
This notation comes from the language of logic and we have been using logic to represent the goals and we will use continuity use the logic. And basically, when we say P and Q can be

unified, we can say that either P and Q are the same or both are of the form of some predicate with some variables.

And the variables in P and Q can be unified rigorously, then we P will become equal to Q. So, by unification essentially we say can we make the two formulas the same, which would mean that P is the link in the causal link and not Q has become equal to not P which means it is broken the link.

As we said a threat happens after A i, A i produces P and A threat deletes it and A threat happens before A i, A j. So, A threat is deleted P and it is no longer available for A j to consume and then, we say that the threat is materialized essentially.

(Refer Slide Time: 21:22)



How do we resolve a threat? Essentially, what we can do is to eliminate one of the preconditions, so to eliminate the threat one needs to ensure let at least one of the three conditions which are required for the threat to materialize is not met essentially.

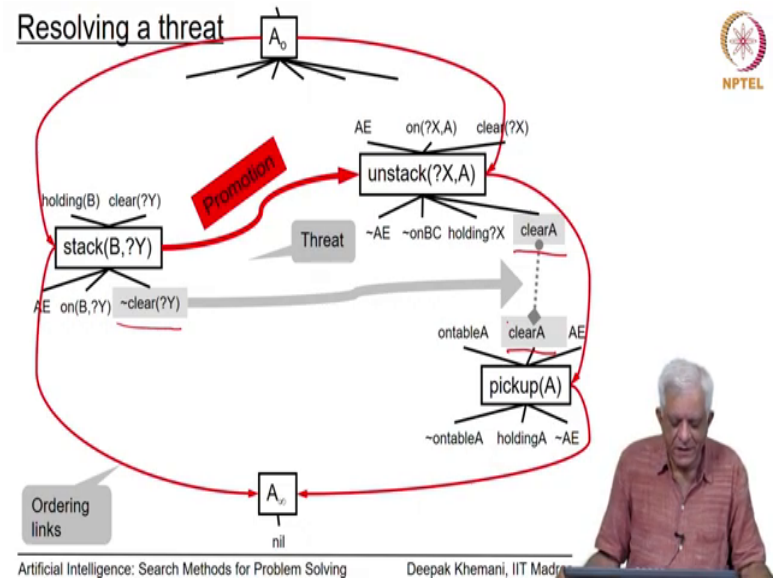
So, this can be done by I am doing one of the three respectively for the three conditions needed for the threat one is called separation and essentially, separation says that ensure that P and Q cannot unify. Then, and this can be done by adding an appropriate binding constraint to this set B in the partial plan.

So, for example, we can somehow say that P is something different from Q. Remember P is the is the link proposition in the causal link and Q is the not of Q is actually the effect of this threatening action or we can move the action A threat to happen before it can disrupt the causal link.

And we call this as promotion, we advance the action A threat and we do this by adding a ordering link which says that A threat must happen before A i to the set of the partial plans. So, you can see that essentially, we are looking at this plan has an explicit representation and we are manipulating that representation and in that sense, a partial plan is a first class object which we can manipulate.

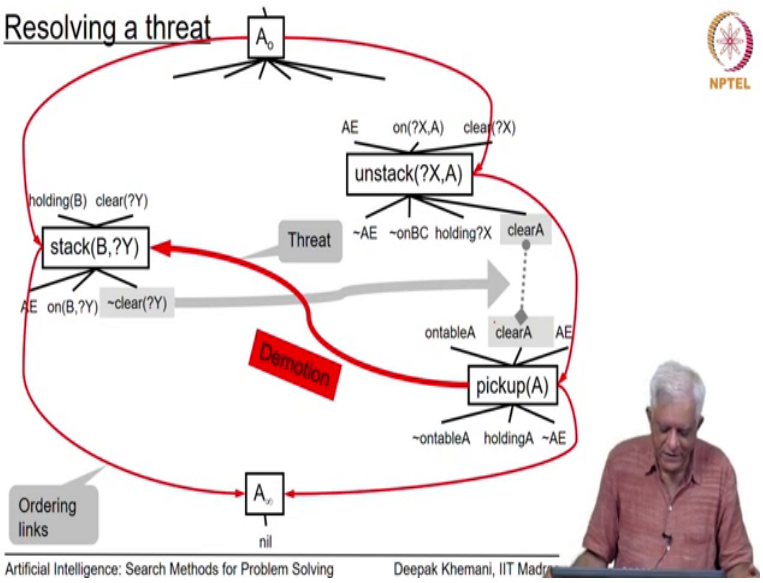
The third thing is called demotion which says that the action A threat should happen after the causal link has been in some sense consummated and we can do that by saying that A j which is a consuming action for the causal link happens before A threat. So, A j has finished consuming P and now, it no longer matters whether A threat happens afterwards and destroys P.

(Refer Slide Time: 23:30)



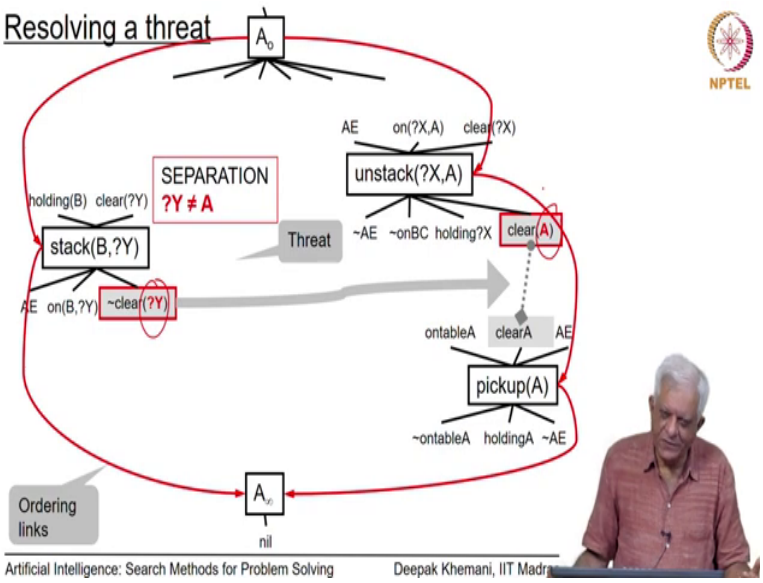
So, here are this example for the threat that we saw. ah We can reduce this threat. Remember that this threat is saying that ‘not clear Y and clear a’ that link is being broken essentially. And we are saying that make sure that stack B, Y happens before unstack X, A; then, the threat will no longer be there and we say this method of resolving the threat is called promotion.

(Refer Slide Time: 24:01)



The other way to do is to say that this action which was consuming the predicate is happens before that the threatening actions, whether it has finished consuming it and the threat no longer materializes.

(Refer Slide Time: 24:18)



And the third thing we can do is separation. We say that this variable Y should not be able to bind to a constant A in which case it would the stack action would remove this clear Y, but that Y would not be A essentially. We have seen this basic algorithm for plan space planning. So, what is algorithm? Just let us do a recap.

The algorithm says that start off is the initial plan which contains the initial action A 0; which produces the start predicates and the final action A infinity which consumes a goal predicates. Keep inspecting this plan, keep looking for a flaw and keep resolving the flaw.

So, there we say there are two kinds of flaws; one is open goals, they can be resolved by either establishing a causal link with an existing action or by adding a new action which will

provide a causal link. The other kind of flaw was a threat and a threat essentially threatens to break a causal link.

And we said that, we can resolve this flaw either by making sure that, this threatening action happens before the causal link comes into effect or after it has been consumed or making sure that it does not break the causal link by saying that a variable is bound in an appropriate manner.

So, now, let us take up an example of this process and see how partial order planning or plan space planning as we have been calling it, finds a plan. We will do that in the next session.