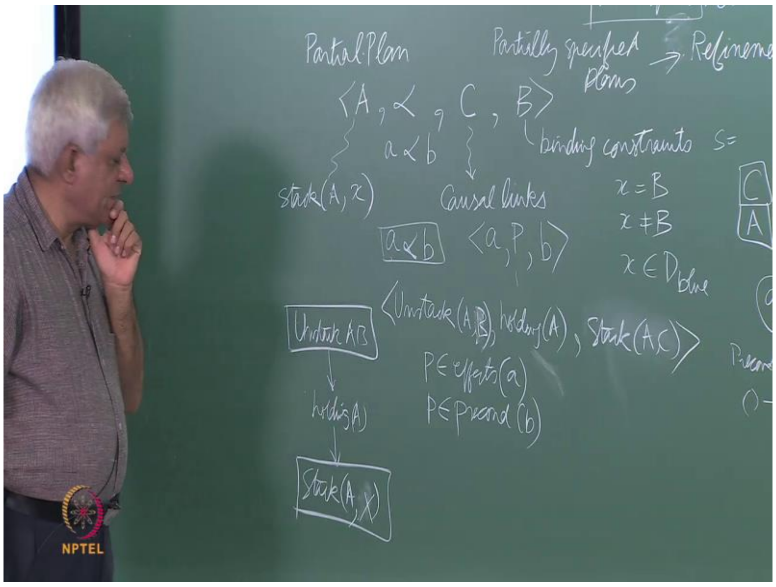
Artificial Intelligence

Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 36
Non linear planning

So, in the last class we looked at goal stack planning and one of the observations we made was that, it essentially does backward listening, and it takes a set of goals to solve, serializes them and tries to solve them one by one essentially. And then we saw that like suspense showed that there are certain problems where you cannot serialize the sub goals and solve them independently and arrive at the final solution; you may have to do extra work to, finally, come to this thing.

(Refer Slide Time: 00:47)



So, today we want to look at an approach which is called nonlinear planning and it is named and it is known by many other names. So, lot of people have looked at it also known as partial order planning which kind of stands for the fact that the plan may be a partial order of actions not necessarily a sequence of actions. So, far we have said in the simplified world that a plan is a sequence of actions; we do action a 1, then we do action a 2 and then we do action a 3 and so on. But this allows us to have actions, plans or recognized plans as a structure which is a partial order, which means that you are not committing to the order of actions essentially.

So, to take an example, supposing you have to wear shoes for going out or something, then you have four actions to be done which is, for example, wear left sock, wear right sock, wear left shoe and wear right shoe essentially. Now it is not necessary that you choose a particular sequence of these four actions. The only thing that you have to do is to wear the left sock before you wear the left shoe and wear the right sock before you wear the right shoes. So, as long as you satisfy this constraint of this order, then the rest can be done in any order. So, in that sense, this kind of planning is also known as least commitment planning.

And by this we mean that in the process of planning, you do not commit too much essentially; you only commit and we will see what are the kind of commitments you have to make as little as possible as little as necessary to solve the current whatever sub-problem that you are solving. This is also known as plan space planning, and again we are familiar with this notion. We have talked about solution space planning; we saw, for example, that when you want to solve SAT like problems or TSP like problems, you construct candidate solutions and do some perturbation in them or we also saw that when we are doing branch and bound with TSP that you work with partial solutions.

That you initially start with a set of all possible to us and then gradually partition them into subsets eventually homing on to towards you are looking at. So, this is in that flavor; it is in the space of possible solutions essentially. So, plan space planning basically works with partially specified plans, and there is a refinement operator step. And there is a refinement step which refines plans in the sense that you specify them more and more essentially. Now a partially specified plan or the partial plan is denoted as a four topple which is a set of actions or operators A which are part of the plan.

So, we are not saying anything about where they lie in the plan; all you are saying is that these are actions as part of the plan. And these actions may be partially instantiated which means instead of saying, for example, unstake a from b, you might say something like unstake x from b which means that you basically want to achieve clear b, for example, and if there is something on b, then you want to unstake it essentially or you might also say unstake a on x, for example. So, partially specified actions may be allowed essentially, then there is an ordering relation which basically says that action a happen before action b.

So, we will specify order explicitly as supposed to implicitly in the planning that we have been doing so far. So, the planning that we have been doing so far can be seen as a state space planning. And so if you remember states space planning, so you combined selection and placement of actions; of course, this is more true of power state space planning and backward state space planning and little bit lets true for goal stack planning. But in forward state space planning and in backward state space planning, a forward state space planning you says this is s; this is by a 1. I go to some explain and then I choose a 2 and I will go to s double prime and so on.

In forward state space planning, the twin task of choosing actions and placing actions in the plan are done simultaneously. They are sort of so closely coupled that you cannot distinguish between the fact of choosing and scheduling them, because the process of looking for action says that I am looking for the first action, then I am looking for the second action, then I am looking for the third action. So, both the things go together. Likewise in backward state space planning essentially; you are choosing the last action, then the second last action then the third last action and so on essentially.

In nonlinear planning or partial order planning, we do these two things separately. We sort of distinguish between the fact that an action is necessary for necessary part of the plan, but we do not necessarily say where the action should be, okay. So, for example, if you are planning a trip from here to Mandy, then you might say, okay, our necessary part of the plan is to somehow get from let us say Chennai station to Delhi station; let us say you are going by train. So, you have decided what the action is, but you have not stated when that action will take place in your plan in a sequence of actions that.

You will say that somewhere in my plan, there must be this action of catching a train from Chennai to Delhi. So, this is the flavor of partial order planning or plan space planning or least commitment planning is that you have said that, okay, I need to take a train from Chennai to Delhi, but I have not said at which point if because eventually my actions will be sequences of actions at which point will I actually be taking this train essentially. Likewise, another action might be booking a train ticket essentially. Now I could do it at any time before the journey; assuming, of course, that we get tickets, but this is not always the case but still.

So, this thing about combining the selection and the placement of action is forced in state space planning. Because you are working with states in plan space planning, we are working with the notion of these partial plans. And we can say that these two things are dependent of each other. Of course, eventually we will have to impose certain order on the actions, but when we are talking about plans, we say these are the actions in the plan and some actions may have certain ordering which you will state in the ordering relation. And this basically says that a happen before b and so on and so forth.

Then we have a casual relation; let me just use the terms c here which I will come to in a moment and then we have a set of binding constraints. So, what do I mean by this? I might say at some point that an action stack a on to x. So, I will use this lower case for a variable just for a moment is part of my plan. I might even say that this action comes before something and happens after something and so on and so forth which I will say here, but at some point I might say that x equal to b here. If I add this extra bit of information, then I have added something more to the plan which says that this action must put a on to b essentially or I could even say x not equal to b.

I could add any kind of a constraint binding constraint to variable essentially or I could say that x belongs to some domain of let us say block set I have; let say blue block set or something like that. I could do all this sort of things; I could add extra constraints which tell you how the variables are bound essentially. Here we are talking about casual links, and the general idea is that this is a triple where you have an action a; you have a predicate p, and you have an action b. And what you are saying here is that a produces a predicate p and b consumes a predicate b predicate p essentially.

So, for example, p could be holding a, and what is there action which might do that? It could be something like unstack a from y, and there might be another action like put down a. This forms a triple and we see this as a causal link between two actions a and b; in this case unstack a from something and put down a. And we say that unstack a from something is providing a predicate or producing a predicate holding a because that is a effect. So, p is a P belongs to effects of a and p belongs to the pre conditions of b. So, there is a predicate like this if you have expressed the predicate like this as such.

So, all these are sets, sets of actions, sets of ordering links, set of causal links and sets of binding constraints, and this partial plan is represented by a portable like this. And if you have asserted a triple here which says that a p b, we are saying that in our plan a produces a predicate p and that p is consumed by b essentially. So, obviously, the moment I add such a causal link, I must also add an action that a happen before b and you can say that this either happens implicitly or you must do it explicitly. So, this is a structure of a partial plan; as you can see in this partial plan structure, I may specify something; I may not specify other things.

So, I may have let us say six actions and I may have ordering relations only between a few of them and causal links between a few of them and binding constraints for some of them essentially. This is in keeping with this notion of least commitment planning, and what we are saying here is that we will add only as much to the plan as it is necessary for solving the sub problem we are trying to solve. So, if you go back to suspense anomaly, in goal stack planning, you are committing to solving one sub goal first and then the second sub goal. And then you are committing to ordering them in the particular way that you are doing; we do not want to do that. We will try to do add constraints only when necessary. So, in that sense its least commitment.

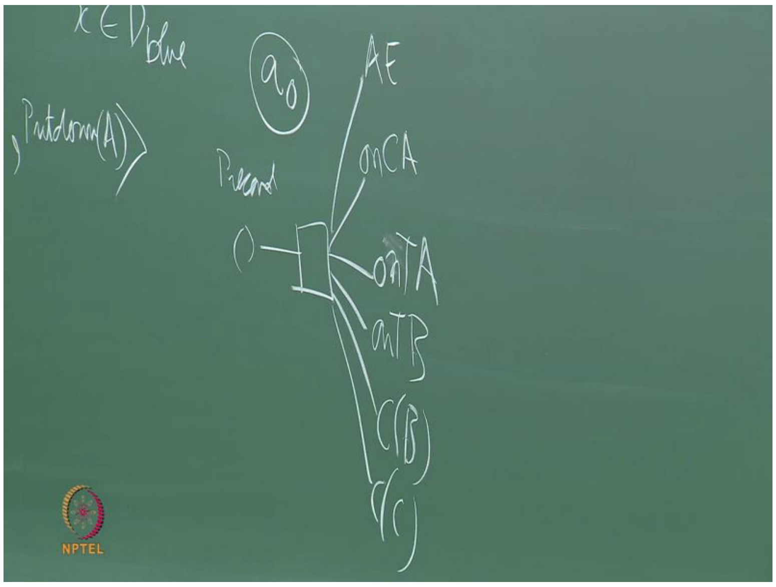
A word about plan space planning; so we are not working in the space of states anymore. We will start with some initial plan and we will apply the refinement step to get new plans out of that. The search space contains only plans or partial plans and partial plans are described by this portable essentially. So, eventually what we want to do is look at a partial plan, choose a refinement step and there may be more than one refinement step. So, some of you have to select one; come up with a new partial plan and do this repeatedly. Choose the refinement step; come up with the new plan and so on. So, one question that one might ask is when will one terminate this.

Now one thing that I would like it to sort of ponder over and think about is the following that even if I have a small state space; let us say I have these three blocks like in suspense anomaly a b and c. There is only a finite number of ways in which you can arrange this c blocks. So, the set of states is finite essentially, but even for such a domain, the set of possible plans is infinite; maybe we will come to that, but I want you to think about this a little bit, okay. So, what is the initial plan? The initial plan this we will call as pie zero; we will always have an initial plan. And the initial plan has two actions a 0 and a infinity.

It has one ordering constraint which says that a 0 happens before a infinity, and it has nothing else. This will always be our initial plan. Remember, keep in mind that states do not exist as far as this planning algorithm is considered; we are only looking at plans or partial plans essentially. So, now the objects that we are manipulating are only plans essentially. So, of course, one question will arise us to when do we terminate because we can no longer say that apply a goal test function to a state, because we do not have state anymore in our this thing. We will have only plans essentially.

Now what are these actions? A 0 has no preconditions and its effects is equal to I just use this notation to start it. So, of course, every planning problem will have a different a 0 action and a 0 action simply produces the start state essentially its effect. When I say this, it is basically a set of predicates, okay.

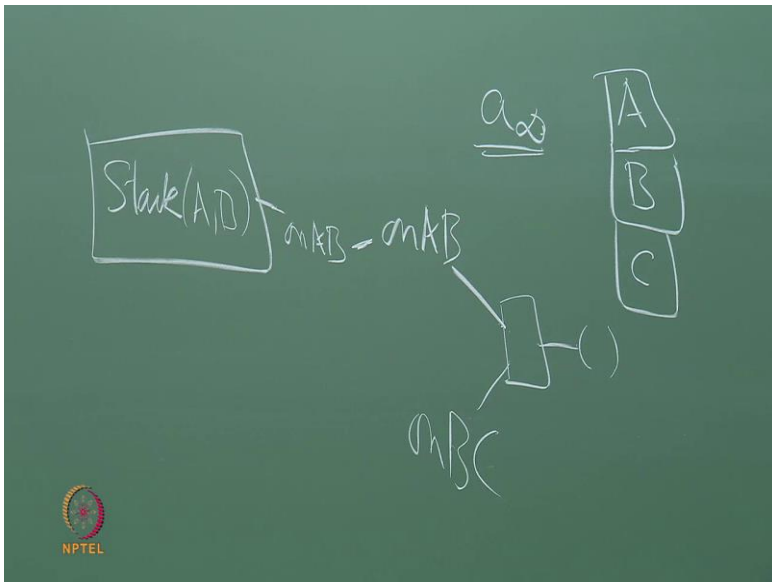
(Refer Slide Time: 18:50)



For example, in the suspense anomaly, we said this is C A B; this is the start stage. So, we have an action; let me draw it here, whose preconditions is the empty set and whose effects are these things on C A on table A on table B, clear B, clear C and A. So, this is a 0 action. As you can see every time you define the planning problem. So, what is the planning problem? A planning problem is a given set of actions. If you are looking at the state space project then the set of states in a start state and a goal state and a set of action that you can use to generate the moves essentially.

So, now, we have said that we are doing away with the notion of states at all. So, instead of the start state, we have a start action and what the start action says is there is an action whose effects are these precondition this start space predicates essentially that somehow produces that.

(Refer Slide Time: 20:33)



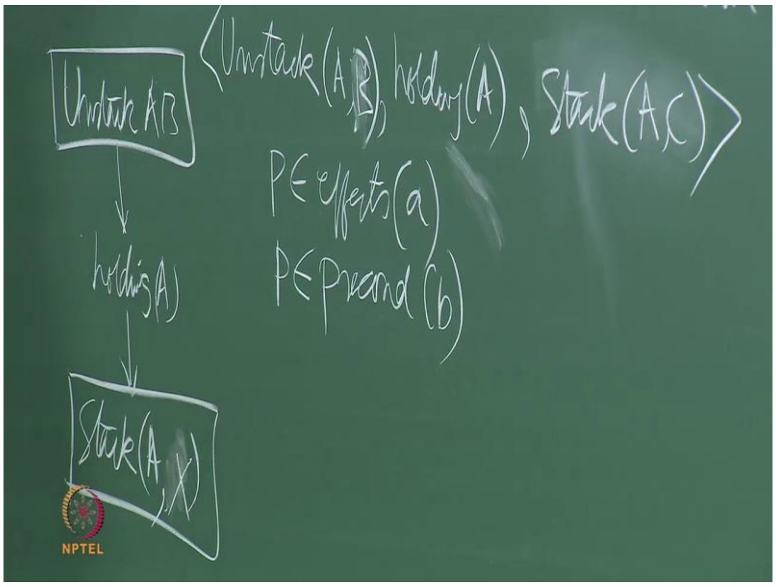
And if our goal is to have A on B on C, then my a infinity is an action which has no effects but whose preconditions are what I want to achieve. So, on AB and BC; this is a infinity. So, this is my starting node in the search space of partial plans, and this is a partial plan which has two actions. One action which produces the start state, other action which consumes the goal state and I have specified anything else except that the start stage happens before the goal stage which we have said here. The start action happens before the goal action or a 0 happens before a infinity.

So, if you now try to remember, when we are talking of TSP and we say that we partially specify we say that we have only one edge specified which will go from Chennai to Bangalore. Then we say that everything else could be anything. So, it was a set of solutions. Likewise, this partial plan pi 0 can be seen to stand for a set of plans in which this is the first action and that are the last action. Now intuitively we can see that we will have to somehow make the connections between these and the kind of connections that we will need to make is that how do we generate the. So, I must have some action.

So, a little bit like backward state space search; some action I must have stack a on b which will have an effect on a b which I will link to this essentially. So, this is like saying that I have specified now one more action in my plan that I must have the stack a b action, and along with specifying this, I am going to specify the few more things. For example, I would say that there is a causal link between stack a b and a infinity and stack a b is producing this predicate on a b which is being consumed by infinity. So, I must establish a causal link.

And one of the things that we would want to do in partial order planning is to somehow add causal links and see that they are not disturbed later essentially and various people have tried various approaches to that essentially. So, this is the causal links which is doing that essentially. So, let me say this.

(Refer Slide Time: 23:44)



So, let say you unstack a from let us call this b here in the different planning problem and then you want to generate this holding a and you want to link it to put down b put down a essentially or instead of putdown a, let me choose a different action which is to stack; let us say a on c. I want to illustrate what I mean by this disruption of causal links essentially. So, what am I doing here? I am saying that I have an action unstack a; it produces holding a which is consumed by stack a onto c or instead of c, you can use a variable x or something like that, it does not matter.

And let us say this arrow stands for a causal link essentially and implicitly we assume that wherever causal link goes and other link follows that this much happen before this. So, it implicitly captures it essentially. Now I have said that my plan is a partial plan. Now supposing I have some other action floating around in my plan which says that stack let us say d 1 to y. So, let me change my example a little bit. Let us say this is producing it is producing holding a, but it is also producing clear b and then I am saying unstack stack x on to b.

So, this also illustrates a fact that when I am considering two actions, they do not necessarily have to be continuous essentially. So, I am not saying that this action happens immediately after this. I am saying this happens after this sometimes, but this action unstack a b; one of the things it produces is clear b and for the stack action x on b, I need to consume this clear b action. Because I can only put something on into b if b is clear. So, I want to emphasize the fact that that they are not contiguous; this could happen at supposing it is eventually you produce a linear plan because it is a one arm robot and it can only do things in a linear fashion.

So, let us say this is the fifth action, and this is the twelfth action. So, I am not saying anything about that; all I am saying is that there is a causal link from this action to this action which means it is producing something which this is consuming. And there is an ordering link which says that this was happened and sometime only later this must have happened. And now somewhere in my partial plan, there is a action floating around called stack d onto y which says that you stack this object d onto some variable y. Now we can see as I say that let us assume this is the fifth action and this is the twelfth action.

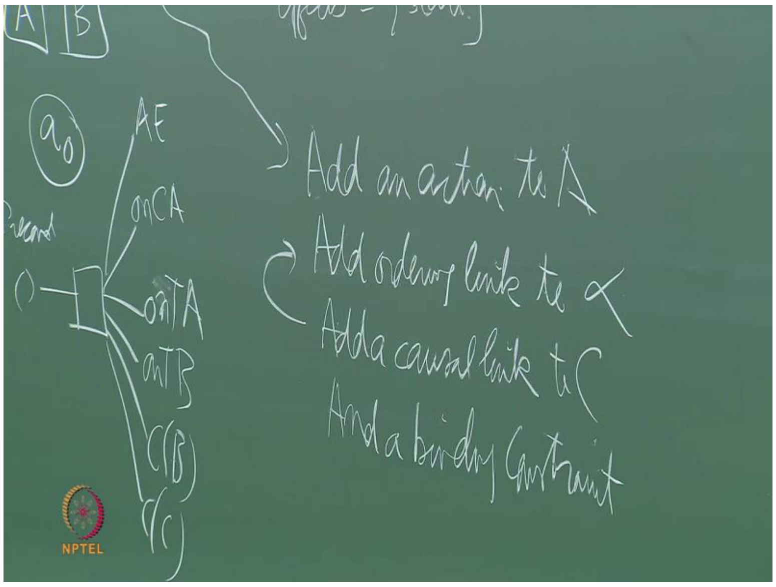
What if this was the seventh action sometime in between and what time this y became equal to b; remember we can do these binding constraints. So, some point I might say stack for some reason that stack d onto b. So, there is a danger that this causal link that I am interested in is going to get disrupted by this action. And my planning process must somehow take this into cognizance and try to do something about it; we will come to that in a moment. So, some of the older planning algorithms, there was an algorithm called tweak one of the first nonlinear planning algorithms. This was written by a guy called Ariston Tate.

So, if you look of Tate and tweak, you will find get some information on the web. They use the term clobbering which modern planning people do not use, and they also use the term declobbering. And I was told that Ariston Tate has one of these online courses on planning which one of my acquaintances was telling me about. So, I am sure if you search for this online course, you will hear lot about tweak and this process essentially. So, what do I mean by clobbering? I mean I am clobbering the casual link. This action threatens to clobber this causal link.

And if it clobbers it, another action might declobber it essentially which means that even if I let say stack d on to b at some later point I might say unstack d from y because eventually I want to stack b on to y. So, I am just trying to give you a flavor of the nature of partial order planning that you keep throwing actions into your partial plan, because you somehow discover that those actions are necessary. Like, for example, deciding that you have to take a train from Chennai to Delhi and then you keep adding more actions. When you add more actions you have to be careful about things like this essentially.

That casual links are not disrupted essentially, then when we talk of links like ordering links; one is to talk about consistency. The ordering links should be consistent, and what do I mean by this? You cannot say that a happen before b and b happens before c and c happens before a, then you have a cycle and then that is not consistent essentially. So, another thing that the planning algorithm will have to do is to know what is the planning algorithm we are talking about.

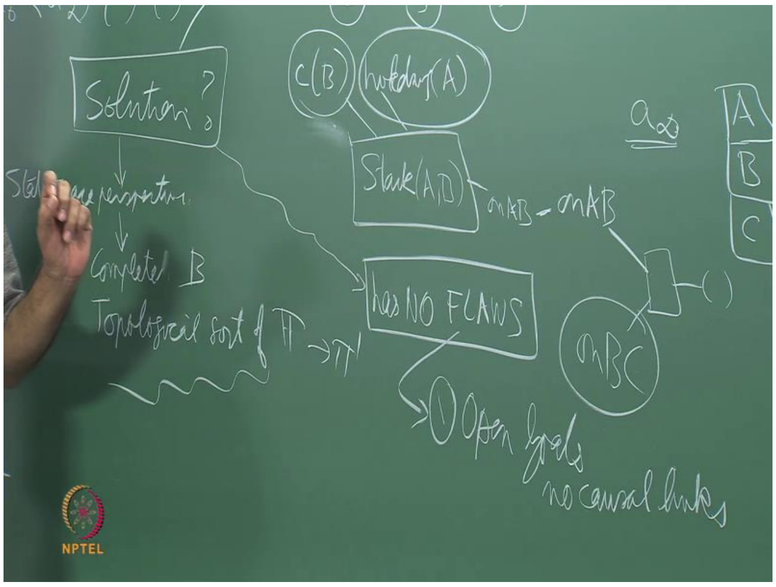
(Refer Slide Time: 30:55)



We are talking about a set of refinement steps and the refinement steps are of four kind, add an action to the set a, add ordering link to this set of ordering links, add the casual link to see. Anyway it says whenever you add this, you have to also add one here or add a constraint. So, what we are saying is that we will start with a partial plan; we will start with this partial plan always an action a 0 which produces the start state as it is showed here and an action a infinity which consumes the goal state as we have shown there. And then we want to fill in more and more stuff; how do we fill in? We have a series of refinements steps.

What do refinement steps do? They may add an action; for example, I said here that you must add this action stack a and b, then I must add a causal link between this and a infinity and an ordering link between this and a infinity. So, I could choose any one of these refinement operators, and that is where the search will come into play. And eventually, I need to refine the plan more and more which means specify the plan more and more. Initially, when I have only a 0 and a infinity, I could have an infinite number of plans which would fit into this and which would still be a solution.

(Refer Slide Time: 32:50)



So, what is a solution? When do we terminate? One way to say it is that, okay, if you want to decide whether a partial plan is a solution, then you could at look at it from the state space perspective, which means you completely specify it. Complete; that means a slight b here; they do not any variables in the plan. Insensate all variables to something, produce a linear order or in other words do a topological sort, because given a partial order, you can always convert it to a consistent linear order. And that process as I am sure you know is called topological sorting and this topological sort of pie will give me some pie prime.

And then I will just use the old mechanism of progressing from the start state, applying the actions one by one, because now I have sequence of actions. I will apply the first action, then the second action, then the third action and so on and check whether the last state that I get is the goal state or not. Of course, I could do that; the travel is the topological sorts could be many. So, do I check for one sort or do I do for all. So, if you go back to this shoe wearing problem which no doubt you have encountered at some point.

You could first wear both the socks and then the two shoes, then both the sox could wear in any order and then the two socks two shoes in any order or you could first wear the left socks, then the left shoe, then the right socks, then the right shoe; all these minimizations of the partial plan that we talked about but we did not draw are valid plans. They are valid ways of wearing a pair of shoes; do I need to look at all those socks and then do that? That would be too painful. Instead the partial planning community has come up with a different test for a solution plan and we say that as well plan has no flaws. Of course, we need to clarify what do we mean by this essentially.

So, what are we interested in? We are interested in a way of looking at a partial plan and saying whether it is a solution plan or not, it is a solution or not. And implicitly, what we mean is this that if this was a solution, then I could take any ordering of the actions and that would any consistent ordering by which you mean that if there is an ordering relation in the partial plan, it must be respected in the solution plan in the linear plan, and that is called topological sort. I can do with any topological sort and that will be a valid plan.

But I do not want to actually do this process of making linear orders and testing, applying that, check for validity function. I want to look at the partial plan itself and make it this observation essentially, okay. So, it should have no flaws. So, what do we mean by flaws? There are two kinds of flaws; one is called open goals. By open goals, you mean no casual links. So, if you look at my partial plan of three objects which you have to somehow figure out from this stuff on the board, I have one action a 0. I am trying to solve those Sussman anomaly problems; I have one action.

So, this is my starting position and a 0 essentially produce this starting position. I have one action a infinity which has these two preconditions; one of them has a causal link which is stack a on b, but this instead intern has more casual links or more preconditions. So, you must be holding a. Then b must be clear; I think that is all, right. So, if I look at this partial plan of three objects a 0 stack a b and a infinity, I have three open goals or three unsatisfied goals; one is clear b, one is holding a and one is on b c. And I say if I have an open goal in my partial plan which means I have a unsatisfied goal or I have a goal which does not have a causal link, then that is a flaw in my plan essentially.

So, a solution plan must not have open goals. If it has no open goals, then it could be a solution plan. The other kind of flaw is called a threat, and what we saw here was a threat essentially. What is that threat? A threat and action threatens a causal link. So, causal links can have threats from actions essentially. When is this situation a threatening situation? It must satisfy three conditions. So, I will just mention them today and in the next class we will take it up from there.

First thing is that it must be somehow undoing this predicate that the casual link is supporting. Remember that every causal link has a predicate produced by one consumed by the other. If it can somehow produce lot of clear b; so if this action has an effect not clear b, then it could be a threat, why? Because this action stack something onto b; it requires b to be clear, and this action is producing that or clobbering that predicate. So, how do we express this view? We say this that we cannot unify this tear by and by unify for the moment, we will just assume that we cannot assign this value to this variable which means, if y becomes equal to b, then this action could be a threat, but that is only one condition.

There are two more conditions. The other two conditions are that this action happens after this action and before this action. If all these three conditions were to be true which means that I can put y equal to b in my plan? I can add an ordering link between unstack a b and this like this which means it is consistent to add this ordering link. And I can add an ordering link like this which means this action happens before this action. If all these three things happen, then we say that the theta is materialized essentially, and in effect, it will no longer be a valid plan, because once you have put d on to b, you cannot put this x on to b essentially.

So, something has gone wrong with the plan essentially, but this is only a potential threat. It is a potential threat, because my plan is a partial plan; I do not know what this value for y is. I can force it to be not equal to b; for example, I said you can do something like this. I can say do not stack it on to b, then of course, I have removed the threat or I can say force it to happen before this action. Then also I have sort of evaded the threat or I can say force this action to happen after this action essentially; that we will see the algorithms for resolving flaws.

But we have this idea of a threat now and action a or let us say an action c threatens a causal link a p b; if it is consistent that it can produce lot of p if it is consistent that it can happen after a and before b, then it is a potential threat essentially. And of course, you resolve that threat we will have to see that one of those three conditions does not happen essentially. So, we say that partial plan is a solution plan if it has no flaws and by this we mean, it has no open goals like for example, here we have three open goals this one this one and this one. And it must have any threat essentially.

In this example, I have only three open goals. So, may be my step would be to put in the action which will produce this or put in the action which will produce this or put in the action which will produce this. And this is the general flavor that once I have flaws in my plan and the flaw should be either open goal or a threat, I must produce a solution for the flaw which is to say somehow takes care of that flaw essentially. So, the high level algorithm for partial order planning or plan space planning or least commitment planning or nonlinear planning is to start with a empty plan a 0 n p t, which basically is telling you what the start state is and what the goal state is. Keep refining it till there are no flaws left essentially.

Once you have no flaws, you must still have a partial order; you may not have specified a complete order. Like for example, in this shoe wearing example, but that would still be a plan. And by this we mean we take any consistent linearization of those actions, and that action will be a plan in the sense of those plans being a valid plan in the state space perspective that we have seen earlier. So, today we have just specified what do we mean by a partial plan. So, this is four topple made up of a set of actions which may be partially instantiated the set of ordering links which may not be complete, a set of casual links which may just link some actions with other actions and set of binding constraints which says that some variables can takes some values or cannot take some values and so on.

And we have defined what does it mean for it to be a solution plan that it should have no flaws, no open goals and no threats, and task is to keep refining a partial plan till it has no flaws essentially. So, we look at this algorithm in the next class when we meet which is on next Friday.