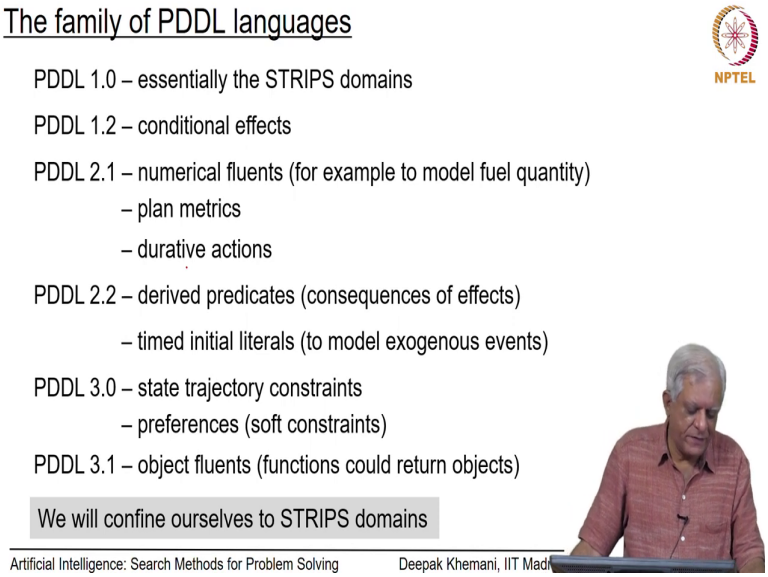
Artificial Intelligence: Search Methods for Problem Solving
Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

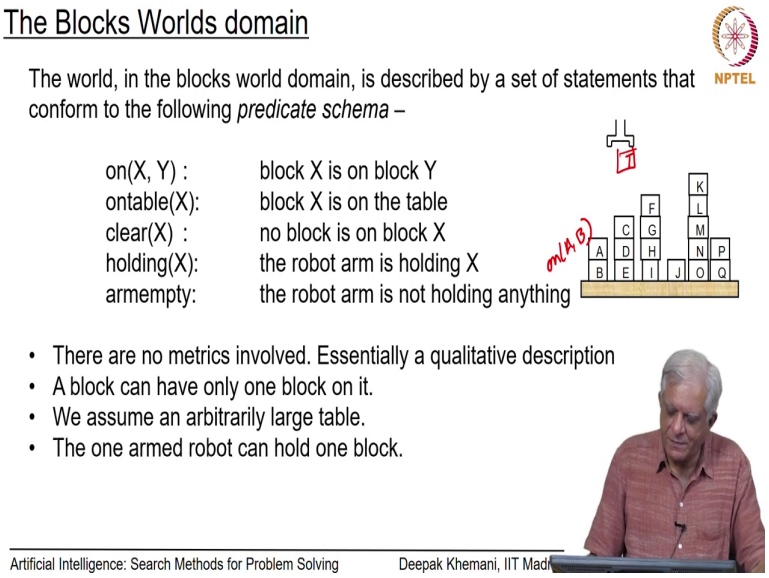
Chapter – 07 and 10
A First Course in Artificial Intelligence
Lecture – 59
The Blocks World Domain

(Refer Slide Time: 00:15)



So welcome back we have just started the study of the area of planning or automated planning or domain in an independent planning is this called and we saw that you know you can model the domains with different degrees of expressivity and then we said that we will start off by looking at the simplest domains which are the STRIPS domains.

(Refer Slide Time: 00:38)



And so, let us do that and we will do that by describing the simplest of all possible worlds and it is been a very popular world for people to write algorithms for planning in and it is called The Blocks World Domain which is kind of depicted on the diagram on the right hand side here.

So, on in this blocks world domain there is a table on which there are blocks which are stacked upon each other and essentially you can assume that all the blocks are of the same size and only one block can be stacked on unlock.

And there is no limitation of what is the size of the table and there are no coordinates, there are no weights, there are no metrics at all. There is simply a world in which blocks are stuck on top of other blocks and there is a robot with a one arm which can manipulate these blocks.

So, this simple domain is called the blocks world domain and the STRIPS language started off by describing this domain. So, the world is described by a set of statements that conform to a language or a set of predicates that we have chosen to describe the world and the predicates are as follows; it says that a predicate on X Y. So, on X Y would be true, if block X were to be on block Y.

So, in the diagram on the right you can see that block A is on block B. So, you can express that by saying on A, B. So, we can describe the world; so we will have a whole set of predicates, instances of this predicates A is on B C is on D F is on F is on G and so on and so forth. Then, we can have a predicate which says that some block is on the table.

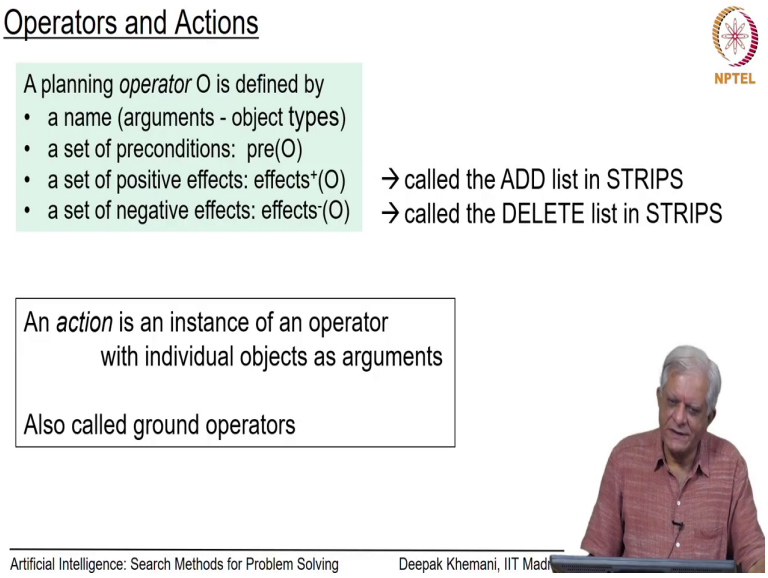
So, for example, we can see that B is on the table, E is on the table, I is on the table and so on. We can say that some blocks is clear which means that there is nothing on top of that block.

So, for example, block A is clear because there is no other block sitting on top of that block. The implication of a block being clear primarily is that you can either pick up that and either the arm can lift that block or you can take another block and put it on top of that. So, that would be a necessary conditions for those kind of actions.

Then, there may be a situation where robot arm may be holding X where X is the block or it may be the case that the arm is empty as is there in the current situations. But on the other hand, if the robot arm were to pick up some block. So, let us say this block is called T, then we would say holding T essentially.

So, as I said there are no metrics involved essentially it is a qualitative description of the world. A block can have only one block on top of it, we assume that we have an arbitrary large table. So, there is no space constraint in that sense and that the one arm robot can pick up one block at a time and so on.

(Refer Slide Time: 03:56)



So, what are the actions and the operators that we can describe in the domain? Now, in general an operator is a schema for an action essentially. Just like the predicate was a schema for statements like on X, Y was a schema for on A, B; likewise we have operators which are schemas for actions and actions will be instances of operators.

So, we can describe; we describe a planning world by describing a set of operators and an operator in a planning world is described and operator O is defined by a name to start with; so what is the name of that operator.

So, when you are talking about the 8 puzzle; we had said you know operators can be left, right, down or up, but in general we can choose any name. One name that you will choose in the blocks world domain is called pickup essentially. So, that is the name of the action or the

operator and it may specify as to what are the arguments and it may also specify what are the types of those arguments. So, for example, you can pick up only a block first.

Then, an operator has to have a set of preconditions; these preconditions are descriptors in terms of the language that we have been defining, the predicates that we have chosen. And the implication of a precondition is that if the preconditions for the operator are true in a given state, then that operator will be applicable essentially. Then, if you execute that operator or execute the action which is an instance of that operator; it may have some positive effects.

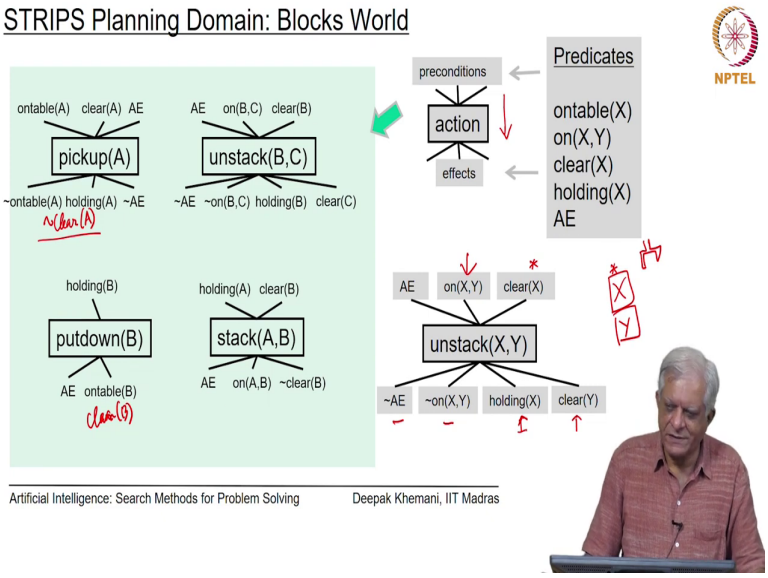
So, which will be given as a set of effects; in the older STRIPS language it was called as an add list, but nowadays we tend to call it as a positive effect essentially, but the idea is still the same. So, for example, a pickup action may have a positive effect that you are holding that object essentially. Actions also have negative effects that something which was true; will no longer be true this was called as a delete list in STRIPS.

And essentially it says that something that was true before the action was executed, is no longer true. So, for example, if block is on the table and you say pick up the block; then it is no longer on the table essentially.

So, you can see that an operator essentially gives you a complete description of when that operator can be applied, what should be the conditions for which it should be applied and what will be the conditions after the operator has been applied.

So, essentially it takes you from one states to another state. As I said, an action is an instance of an operator where you substitute the variables with real objects. So, instead of X, you might say block A and so on and these are also sometimes called as ground operators; ground in the sense that they do not have any variables in them.

(Refer Slide Time: 07:03)



So, what are the operators that have been traditionally defined in the blocks world domain? Now, remember that the predicates that we defined for the blocks world domain were on table X, on X, Y, clear X, holding X or Arm Empty. Now, remember that an operator has to have a certain structure. On the top, we can see that an action which is an instance of an operator would have to have a certain preconditions and certain effects essentially.

So, in some sense there is an arrow of time which is flowing like this that if the flicks are true, then you can do that action and then the effects will become true as the case may be positive or negative effects. An instance of an operator in the blocks world is called unstack X from Y.

So, essentially what you are saying here is that X is on Y. So, this must be true that is given here; X is clear, that is described here that there is nothing on top of X and the robot arm is empty and that is described by the abbreviation AE; AE stands for Arm Empty.

So, if these three conditions are true that the arm is free or empty, that X is on Y and there is nothing on top of X; then you can implement, you can execute this action which is called unstack X from Y.

And the effect of that would be first of all that you are holding; you would be holding X, then you would have made Y clear. The delete effect or the negative effect would be that X is no longer on y because you have picked it up from there and also the arm is no longer empty.

So, this operator completely described this; for this action or the action schema when you want to unstack one block from another block. So, an instance of that would be something like unstack B from C. So, it must be true that B must be on C and there will be nothing on top of B and so on and so forth. So, in the original STRIPS domain; there were four operators, so let us quickly look at the actions pertaining to those four operators.

The STRIPS language distinguish between pickup and unstuck; in the sense that unstack could be done only from another block whereas, pickup had to be done from the table. So, if you are saying pick up A; a precondition is that A must be on the table and then of course, A must be clear and the arm must be empty.

And when you pick it up, you end up holding that object A; it is no longer on the table and the arm is no longer empty. So, you can see it is similar to unstack except that it takes only one argument.

The opposite of pickup is put down; if you are holding B for example, you can put it down and at the end of that action, Arm Empty would be true and on table B would be true. Now, I have not added in this case; for example, I have not added the negative effect of clear A.

I mean that is in this particular world, it does not matter so much because it being a very simple world. When you pick up A, then you do not know whether to call it clear or not, but when you put it down because once you pick up an object, you are only going to put it down or you are going to stack it onto something else and then of course, it will become clear A.

So, if you if it is clear before the action and if it is clear after the action; then maybe you do not need to delete that predicate and save some amount of work for your planner essentially. So, it is for this reason that the put down action does not say; it does not have an effect called clear B.

So, we are assuming here is that we did not delete clear B at whenever we picked up B; so we do not need to add it again essentially. But this does not mean that that all the preconditions would be in the delete list.

For example, if I had said pick up a red block and define a different action which says the color is part of a description, then the fact that the color of that block is red would be a precondition and it would remain effect after that. So, you do not necessarily have to delete everything which is in the pre condition. So, there may be things in the precondition which may remain true in the post conditions or the effects.

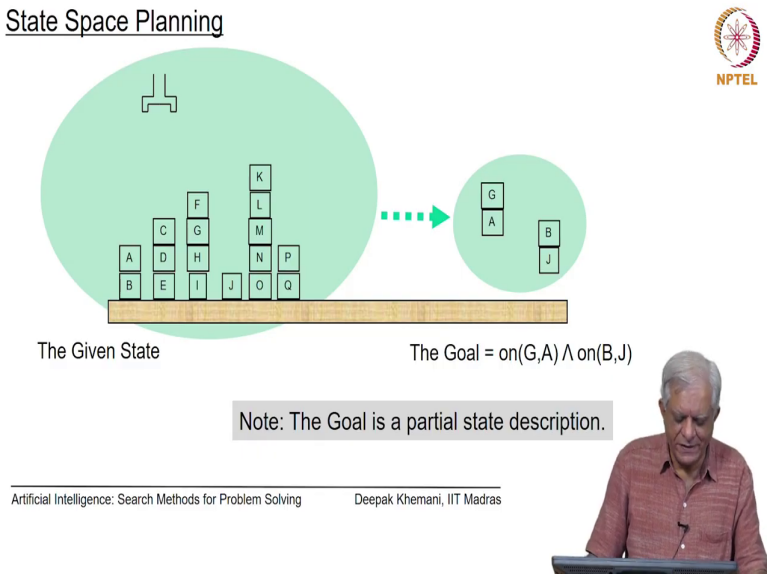
Then, we have the opposite of unstack is stack; so, you can stack A on B provided you are holding A and there is nothing on top of B and the effect of stack would be that the arm would become empty; A would end up being on B and B would not be clear.

So, here we have said that because we are putting A on B; B was clear to start with, but it is no longer true after the action essentially ok. So, these are the basic operators from the STRIPS planning world and you can define your own operators in any problem domain that you are working.

As a exercise, I would encourage you to try to modify these operators or modify or extend these operators to a multi armed robot situation where there is more than one arm and we will

also extend the planning domain to say that you can do concurrent actions that both the arms; for example, can put up pick up objects at the same time. So, you could do this as a small exercise that for example, if you have two arms; then what are the actions that you can define.

(Refer Slide Time: 13:26)



Now, given the planning domains state space planning essentially says that you are given a state and you are have some goal description and you have to find the sequence of actions which will take you from the given state or the start state to the goal state essentially.

Now, one thing you should observe that here on the left hand side; the given state is completely described and that is a necessary condition that we have said that the world is completely observable which means that we must know the entire state essentially. But the goal description does not have to be complete.

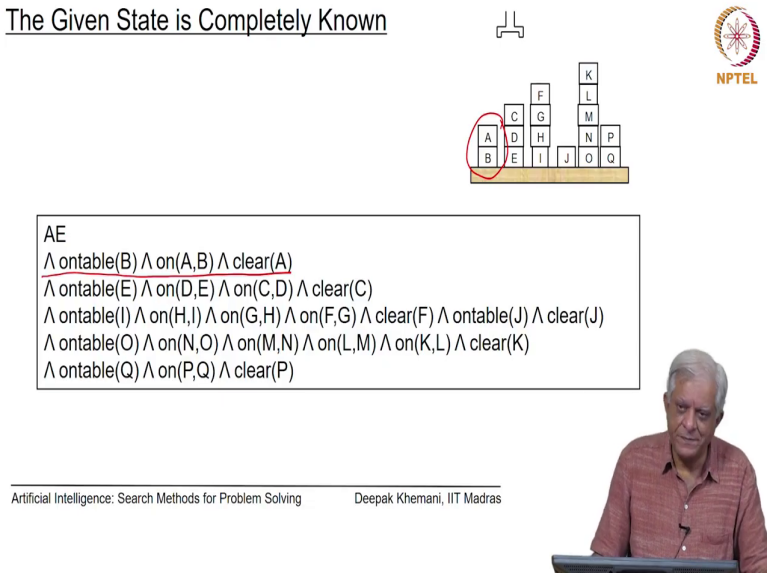
We are not going to say that this is what is going to be true in the entire goal state all we are saying in this small example here is that G block G must be on block A. We do not care where block A is we do not care what is on top of block G all that does not matter, the only thing we care about is that block G must be on block A and also that block B must be on block J.

So, our goal description is only these two predicates on G, A and on B, J and we do not care about the rest of the world. And this is something which is characteristic of planning; it is also characteristic of planning in the true real world essentially. So, you are there for example, watching this video at this point of time and you are waiting for it to get over and you want to go to the canteen and have a dosa.

Then, your goal is simply that you should be in the canteen eating a dosa; you do not care about what other students are doing, what other people are doing, what the waiters are doing; are there more people in the canteen. You do not care about the complete state, you only care about part of a state which you are interested in and that is called your goal description. Your goal description is you are in the canteen and you are eating a dosa.

So, that is one characteristic of planning problems; that was true in the earlier state space search also we said that we said that we will describe the goal in some way. If you describe only part of a state, then it is equivalent to describing a set of states because the other objects could be satisfying other predicates and all of them would be part of the group; all of them would be fit to call a goal state.

(Refer Slide Time: 15:51)



The given state is of course, completely known; so the given state that is shown here pictorially on the top right is shown in the box, as a set of statements in which all the predicates describe the state.

So, on table B says B is on the table, then on A, B; says that A is on B; I am reading the first line here and clear A's shows says that A is clear. So, these three statements on table B, on A, B, clear A are describing this part of the domain. Likewise, other statements describe other part of the state and we have a complete description of the state.