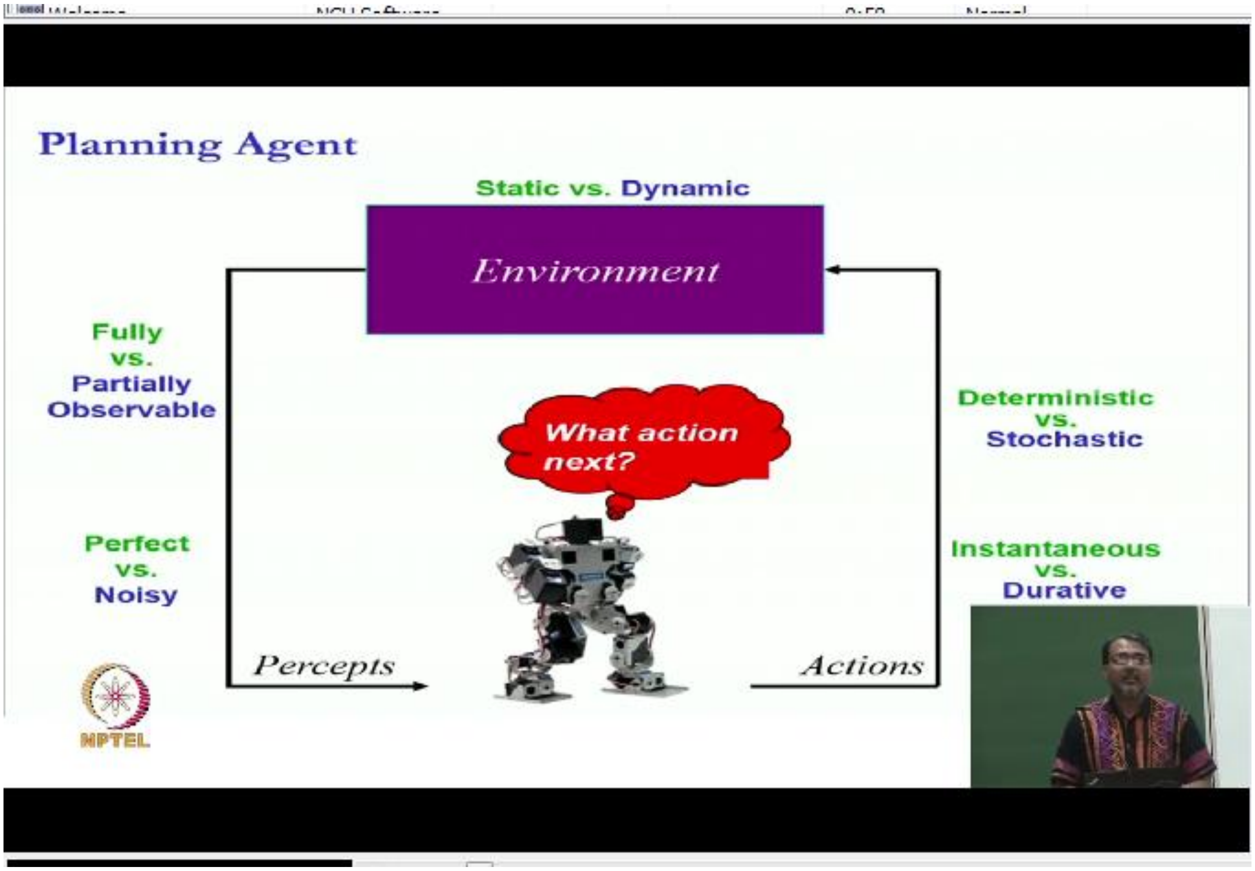
Artificial Intelligence
Prof. Mausam
Department of Computer Science and Engineering
Indian Institute of Science Education and Research-Delhi

Lecture-70
Markov Decision Processes:
Definition
PART-1

So, now is the time we can get started on the next topic, which builds upon what we have done in these 2 lectures. And this is a very important topic for the field of AI today because a lot of the decision making problems happen with uncertainty but for the long time, this is a long sequential process here. And those problems are modelled as a Markov decision process right.

(Refer Slide Time: 00:56)


Planning Agent

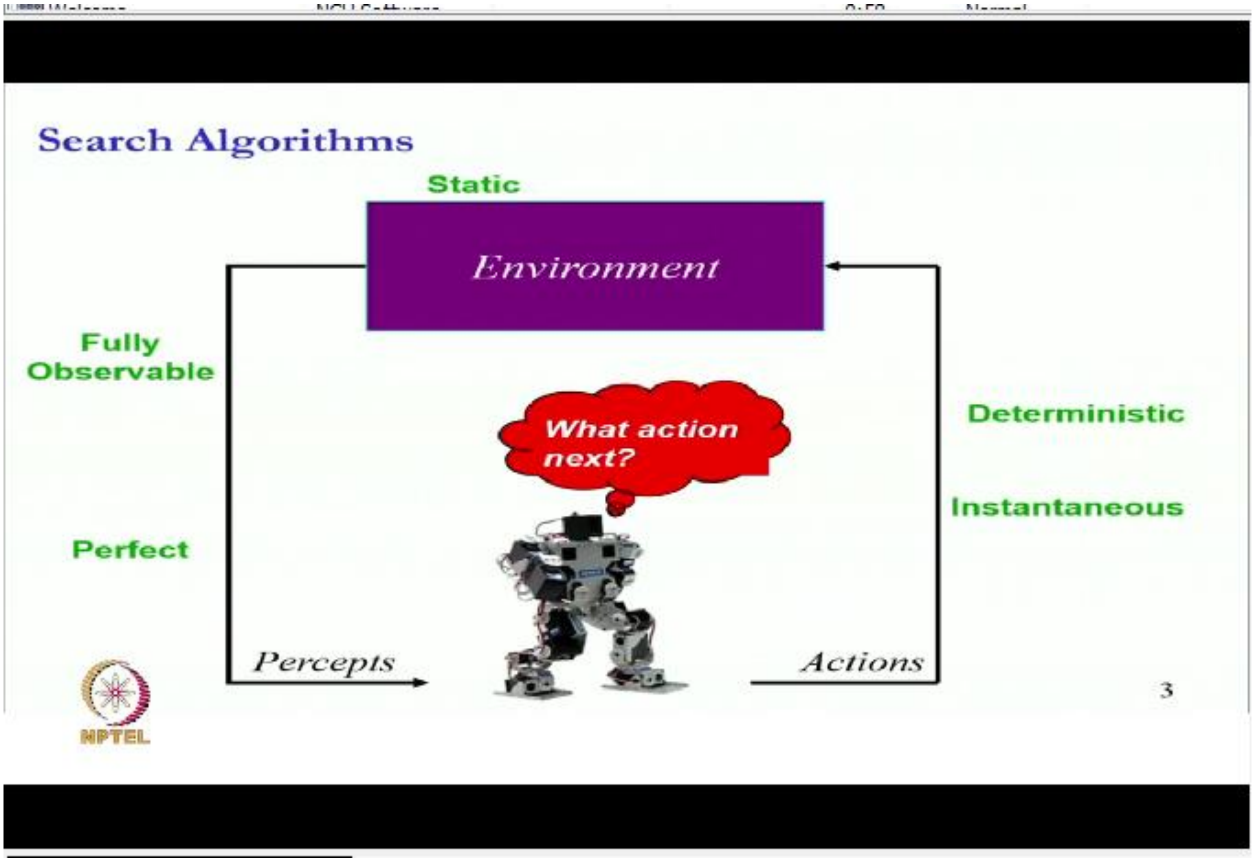
Static vs. Dynamic
Environment
Fully vs. Partially Observable
Perfect vs. Noisy
What action next?
Deterministic vs. Stochastic
Instantaneous vs. Durative
Percepts
Actions
NPTEL

So, if you think about building an AI agent, the AI agent has to interact with the environment, it can observe the environment, it can make changes to the environment. Then there are many properties of the environment and so on and the agent that we have discussed right even in the

PEAS framework and so on so forth, which make the problem harder or easy like does the environment change by itself or am I the only one changing it, is the model is then I have not completely fully observable or only partially observable.

When I observe it, do I make perfect observations or do I make noisy observations right, think about sensors that can go wrong. Is the environment deterministic, is my action deterministic or does my action have different outcomes with different probabilities or my actions instantaneous or my actions durative, etc., etc., etc. There are many, many such properties. And if everything is doing nice and happy.

(Refer Slide Time: 01:54)



Search Algorithms

Static

Environment

Fully Observable

Perfect

What action next?

Deterministic

Instantaneous

Percepts

Actions

3

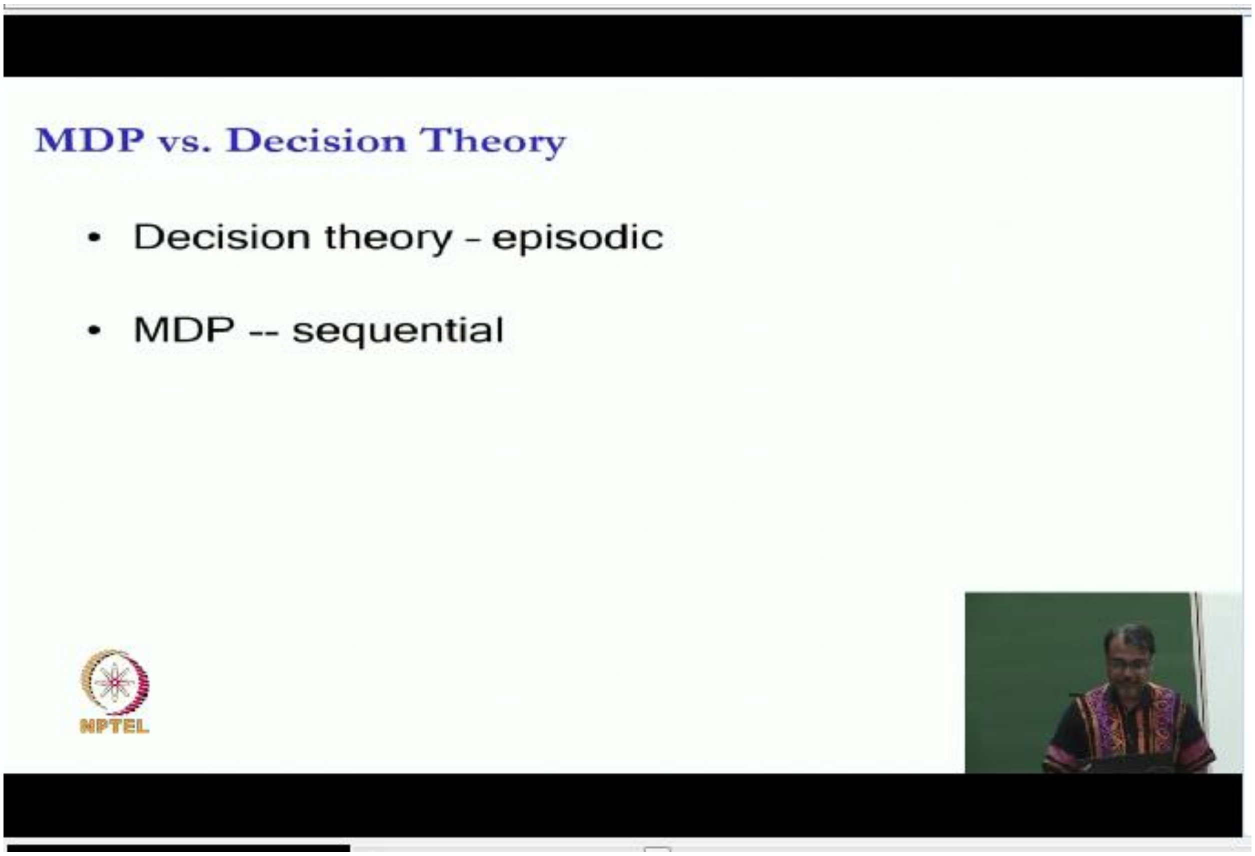
Then this is what we did in the first part of the course. We said ah environment I am not going to change think 15 puzzle, only I will be the one changing it, I will know exactly that configuration, I will know perfectly. I will my actions are deterministic because they are in the formal problem solving world and so on. But of course, if you actually wanted to make a real bot, which is playing 15 puzzle.

Bot as in robot, which has, you know, hand clasp and it can actually pick up a piece and move it and so on, so forth. And it observes the numbers using its vision, lots of uncertainties gonna show up. The people have chess playing competitions where the arm is playing the chess and then they have to deal with a lot of uncertainty. A piece may fall down, you know, they did not put it on the right square.

They put it slightly here, slightly there, lots of things can go wrong, in the real physical world. So the most difficult problem to solve is a dynamic environment, partially observable, noisy sensors, actions take time, actions are stochastic blah, blah, blah, and we will not get to the very full blown problem, right. But the one problem that I want to work on now is when we make one change, that everything else is the same.

We have a fully observable world, the world is perfectly observable, etc., but our actions have stochastic outcomes okay. And this is a very fundamental model because a lot of other models which I am not talking about will can be converted into this formalism like lot of deterministic problems can be converted into logic in the same way a lot of probabilistic volumes can be converted into a Markov decision process.

(Refer Slide Time: 03:44)

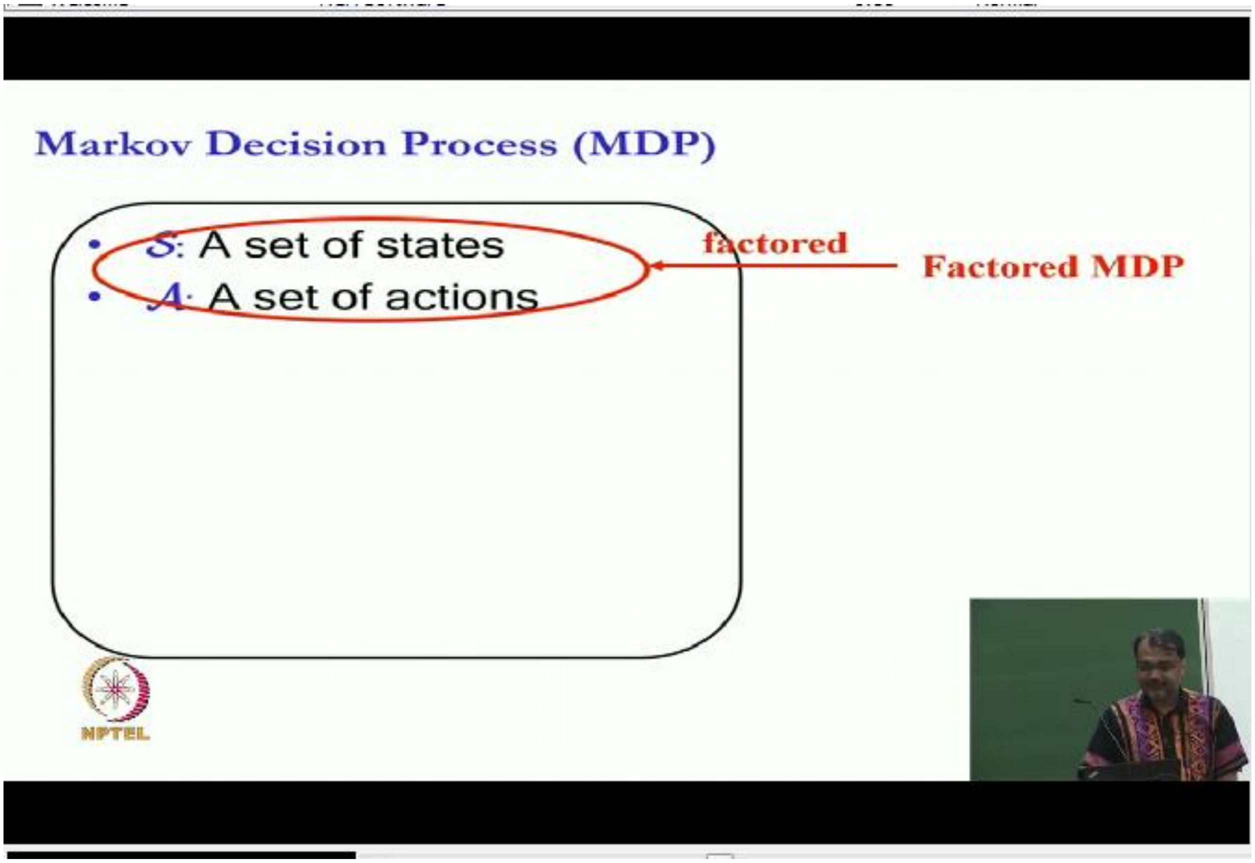


MDP vs. Decision Theory

- Decision theory – episodic
- MDP -- sequential

What is the difference between what we have done in the last 2 lectures and now the difference is that in decision theory, up until now, we have only done one action which plan do I make. Now we are going to do a long sequence of actions, right. So that is the major change. And if we as we build these things, we have to always start building some atomic agent world right. So, let us in the past we had a set of states and set of actions and deterministic transitions. So Markov decision process will be defined in the same way. But we will have stochastic transitions.

(Refer Slide Time: 04:23)



Markov Decision Process (MDP)

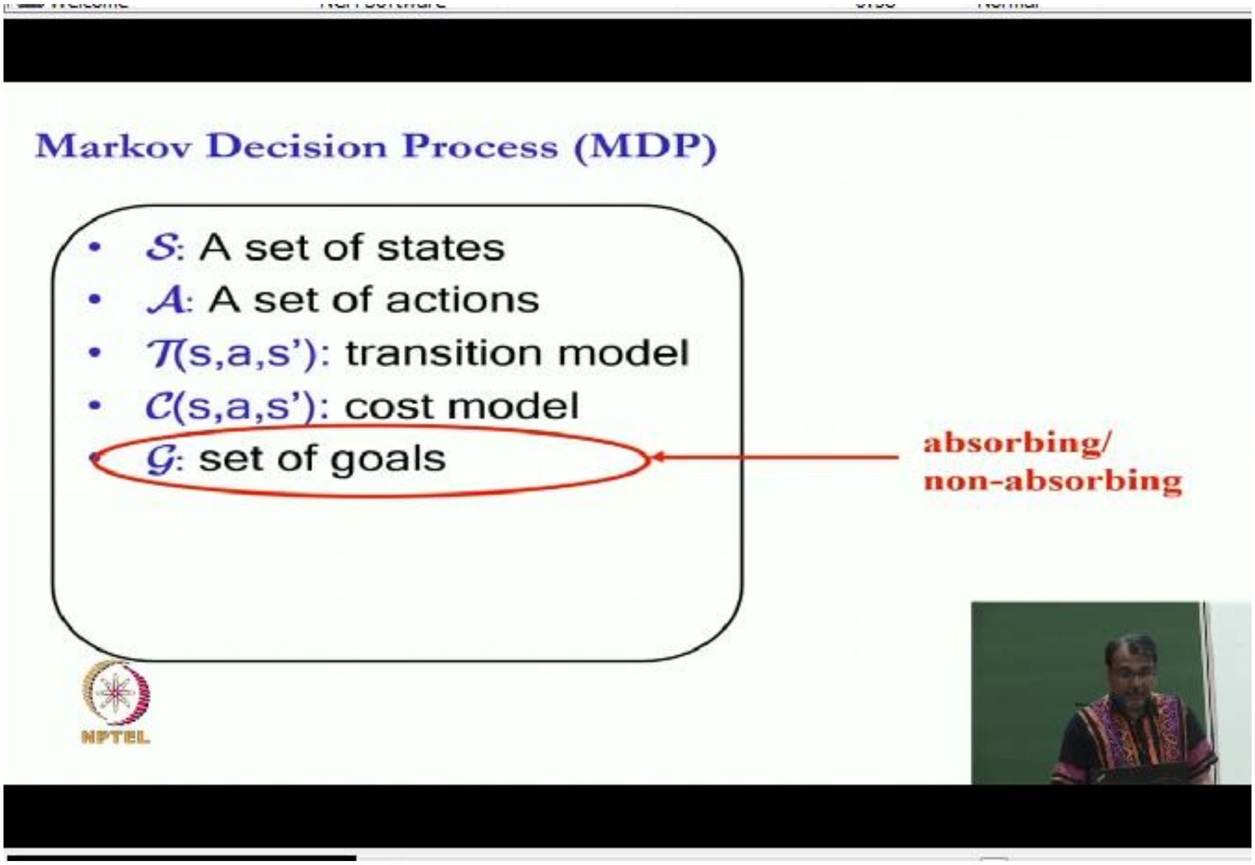
- $\mathcal{S}$: A set of states
- $\mathcal{A}$: A set of actions

factored

Factored MDP

So, we will have a set of states, in this lecture we will use S to denote it curly S will have a set of actions A. Now, all that we learned about factoring a state into state variables etc., will still hold right. So my states could be given a set of state variables, true false, false, true and that makes a state and that is what is called a factored MDP. In fact, actions could be factored to in a first order representation which we will not talk about it. But whatever we did or did not do in the search world applies in the probabilistic volumes.

(Refer Slide Time: 05:04)



Markov Decision Process (MDP)

- $\mathcal{S}$: A set of states
- $\mathcal{A}$: A set of actions
- $\mathcal{T}$(s,a,s'): transition model
- $\mathcal{C}$(s,a,s'): cost model
- $\mathcal{G}$: set of goals

absorbing/
non-absorbing

Then we will have a transition function. The transition function says that if I am in state S that my agent is in state S and the agent decides to take an action A, it may reach S prime with some probability, it may reach certain different S double prime with a different probability the probability function is given by T s a s prime. S is the current state, s prime is the next state. Then I will give you a cost model right, actions can take costs.

So, I will say if I take an action a in state s and we state s prime. This is the amount of cost that I will pay. Now, you guys have done probability, can you guess which kinds of processes do we typically call Markov processes. Sukrith yes, it is independent of the previous, it means what happens in the future is completely dependent on the current state and does not depend on anything else that has happened in the past.

This would be a first order Markov assumption. Second order means 2 and so on, so we will work on first. So now notice what has happened, the transition function has been defined only on the current state. And what happens afterwards, the cost function also has only been defined on the current state. How we arrived at s has nothing to do with the transition function or the cost function. Therefore, it is a Markov process, yes.

We have our state s and that is a deterministic and we have an action a. So, why are not we saying that s dash will be the final state. I mean, if do not s I taken action a then I should reach s that why is it with certain probability right. What is your name. Arshad says why do I have why, why is not my world deterministic. So now we have given this example many times, so think about the word in the physical environment.

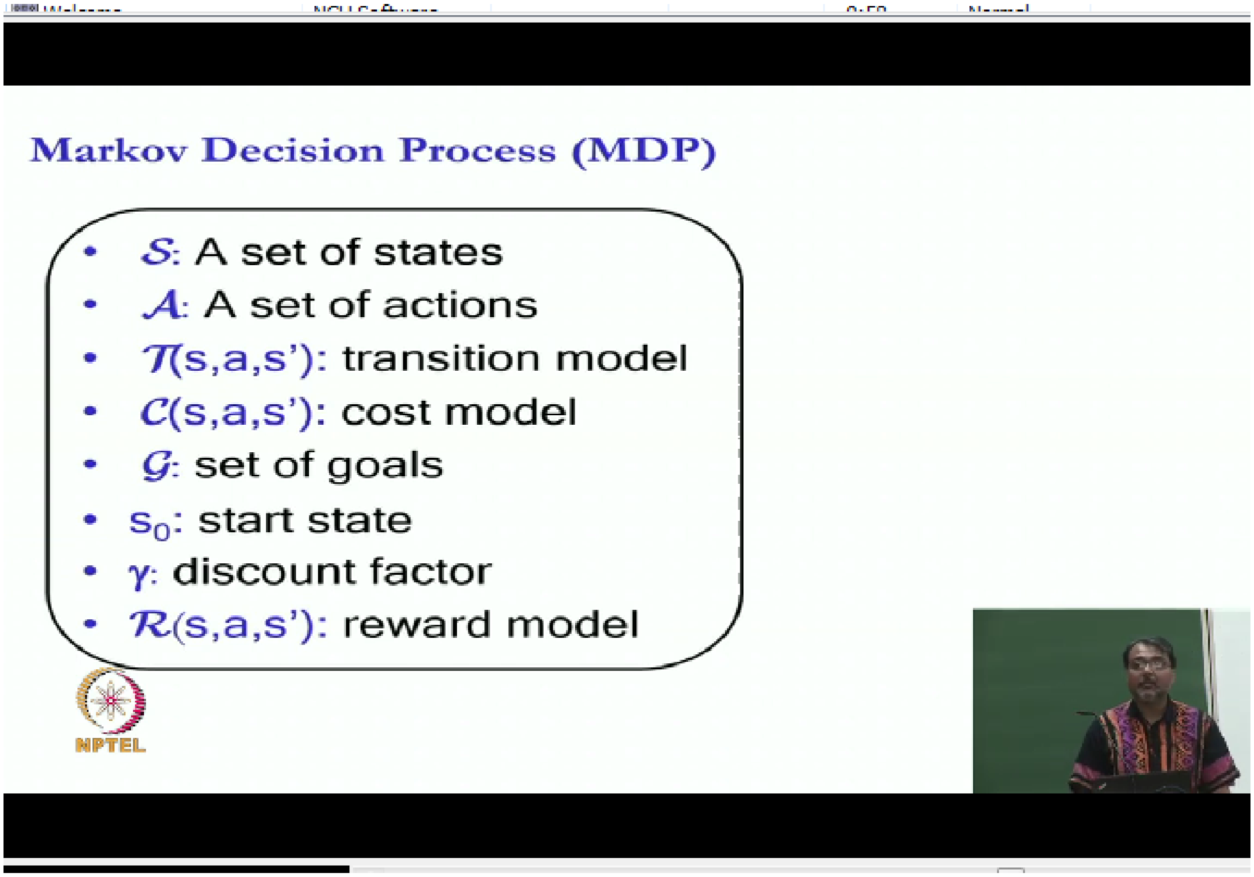
Let us say it is trying to play chess. It knows the board, let us say it knows the board perfectly for now. It decides to pick up the bishop and moves it to a different square. Now what can go wrong when it tries to do this. It is an arm. It is a physical arm, it is goes close to the board, it tries to pick up the thing and when it is slowly trying to grasp the bishop what can happen, it may miss it. For example, so it is about what we know at the time of taking an action.

When we take an action, right, we do not know what is going to happen. But after we have taken the action, we know exactly what happened. This is what the model is assuming. Of course, the chess example is not exactly perfect because it may not exactly know where it is at every point in time. But even if it exactly knew, even if it closed perfectly, you know, there are some things that may not be modeled in the chessboard.

And you know, the piece may get missed, or it starts to lift it and then it lifts it but it falls down and lots of things can happen. Things can never be perfect in a physical world. So the assumption that the model is making is that yes, I know exactly about the state. I know perfectly about it. I know exactly where I am. But once before I take an action, I do not know where I am going to reach. But after I take an action, I know exactly where I reached. We will talk more about this.

Then I may be given a set of goals right. And in fact the model is so, general that it can deal with absorbing goal and non absorbing goal which basically means, you reach this goal you are done or you reach this goal, you got some value from reaching this goal you carry on and now achieve the next goal in life. The model can handle all of that.

(Refer Slide Time: 09:35)



Markov Decision Process (MDP)

- $\mathcal{S}$: A set of states
- $\mathcal{A}$: A set of actions
- $\mathcal{T}$(s,a,s'): transition model
- $\mathcal{C}$(s,a,s'): cost model
- $\mathcal{G}$: set of goals
- s_0: start state
- γ: discount factor
- $\mathcal{R}$(s,a,s'): reward model

You may or may not be given a start state. There is another additional thing that we need to work with called the discount factor gamma. And I will explain what that means. Sometimes you may be given a reward function also. These are the various kinds of inputs, I can give you in a Markov decision process. Of course, not all inputs may be necessarily given in a particular setting.

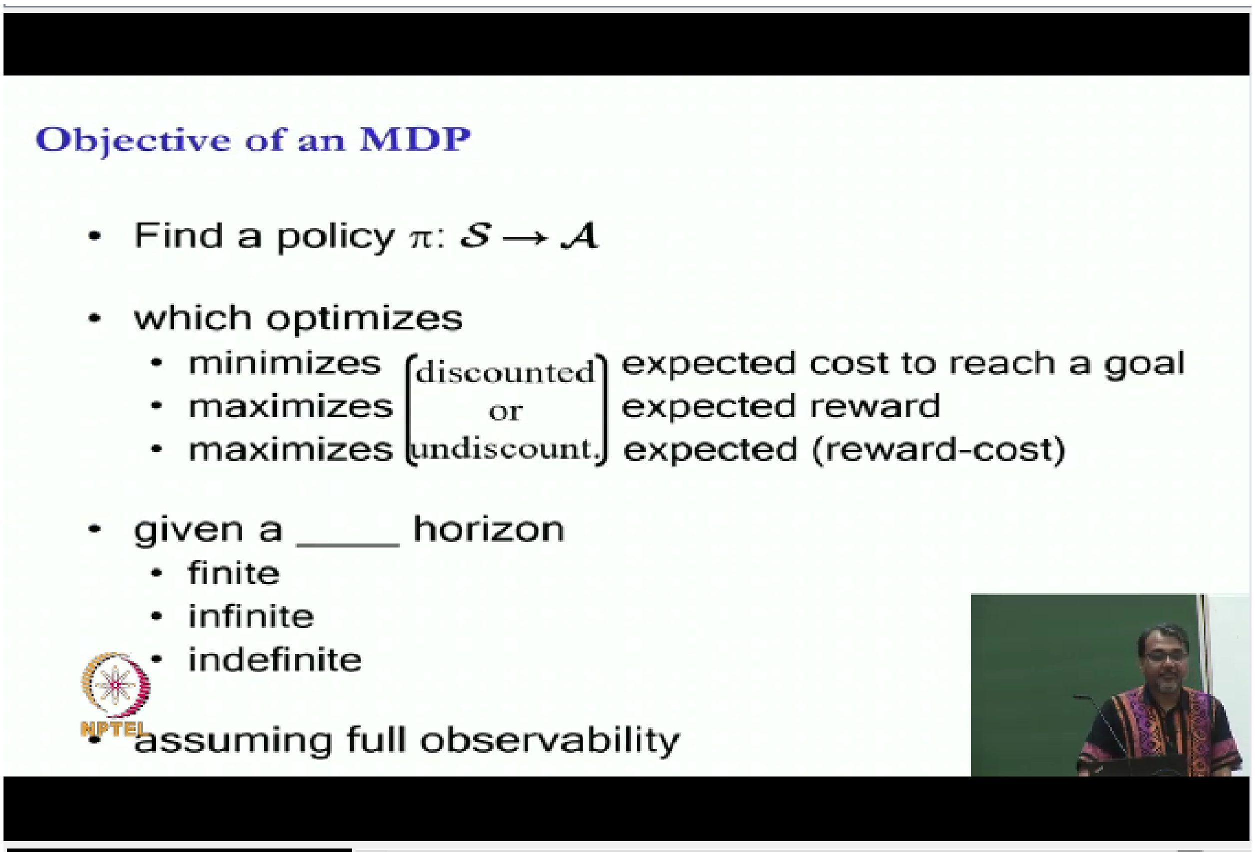
For example, one of the possible inputs that you may be given as this, I do not give you a cost function, but I give you a reward function, your job is to sort of maximize your reward. And I tell you where you are starting from, sometimes I may not tell you where you are starting from, sometimes I will give you costs of actions and goals right and you are saying okay, achieve the goals and reduce the cost.

So, this is sort of the various models I might give you in a Markov decision processes. Now, let us think about what is an objective of such a problem of a Markov decision process. And this is a very interesting question and let us work on it piecemeal. Now, in a search problem when I have a starting state a deterministic world, I need to reach the goal. What is the solution, think depth first search, think breadth first search.

What solution am I outputting. I am outputting a path, a path that goes from start state to the to any goal state okay. In a Markov decision processes when actions can have uncertain probabilistic outcomes, not in my control. What is the output that I am looking for. Is it a path. Is it a sequence of actions. Path is a sequence of actions, yes. So in a state I might do a first action, what can happen, I can reach different states with different probabilities.

Now, what might I do in different states. I might take different actions right, for example, one of my actions could be asked an economist and based on the outcome favorable or unfavorable, I can make the large plant or the smart plant or no plant. So, in a setting where I make, wants to do different actions in different uncertain outcomes is a path the current output, what am I looking for. So you can say I am looking for the conditional graph or there is an alternative way to represent the same thing and that is called a policy.

(Refer Slide Time: 13:02)



Objective of an MDP

- Find a policy $\pi: \mathcal{S} \rightarrow \mathcal{A}$
- which optimizes
  - minimizes $\begin{bmatrix}\text{discounted}\\ \text{or}\\ \text{undiscount.}\end{bmatrix}$ expected cost to reach a goal
  - maximizes expected reward
  - maximizes expected (reward-cost)
- given a _____ horizon
  - finite
  - infinite
  - indefinite
- assuming full observability

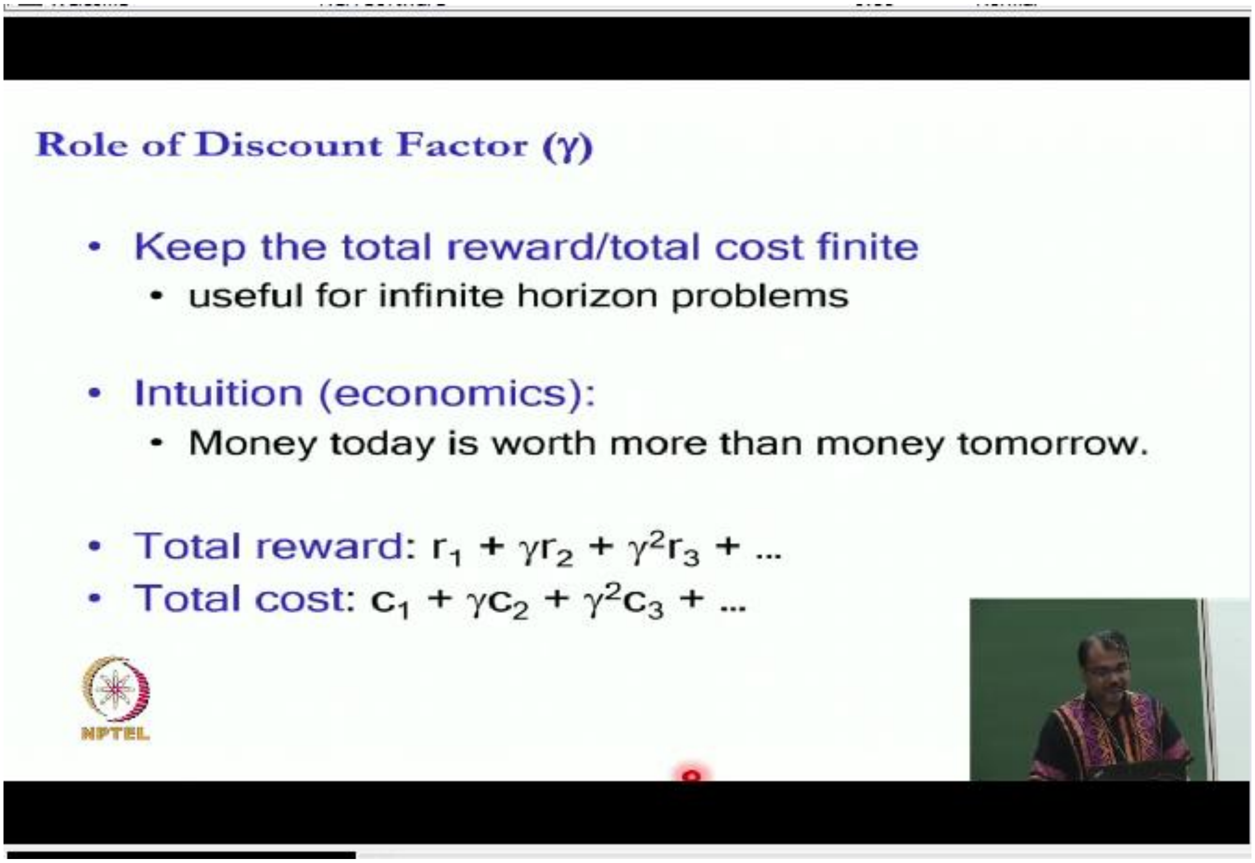
A policy is a mapping from states to actions. It is a table. It is a table where for every state I know which action to do. Is that sufficient for us. Let us think about, I am in a state I can look at my policy table it tells me which action to do, I do this action. If I do this action, I can reach many, many states. Now what I am doing is I am assuming what is called full observability. This full observability to your point says, once I take this action, I know exactly where I land.

I know exactly where it is favorable or unfavorable. Now there is no uncertainty there. So now I know exactly the state I am in because I know exactly what happened before. Because I know exactly the state I am in, I can still again look at the policy table and figure out which action to do and executed. So in the world where there is full observability a policy is a good solution to my problem, because I always am aware of my state completely.

Does that make sense. You are learning this for the first time. So if you have any questions you should ask. Yes, yes that is right. Yes, yes, okay. So, first asked the question do I know the probabilities and for now, we are assuming all the probabilities are known to us for all the states.

So, for now, we are assuming that this is given to you, state action transition cost, all these models are given to you. Now when these models are not given to that problem is called does anybody know, that is called reinforcement learning and we will talk about it, it is part of a course, it is coming right after this set of lectures okay. But for now, we will assume that all this model is given to us.

(Refer Slide Time: 15:17)



Role of Discount Factor (γ)

- Keep the total reward/total cost finite
  - useful for infinite horizon problems
- Intuition (economics):
  - Money today is worth more than money tomorrow.
- Total reward: $r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots$
- Total cost: $c_1 + \gamma c_2 + \gamma^2 c_3 + \ldots$

The next thing we have to answer is in the search world we were minimizing the cost to reach the goal or maximizing the profit. But now, costs to reach the goal makes no sense because their probabilities in the middle. So, we are going to optimize expected costs or expected rewards okay. Last but not the least, how long do I want the agent to move. Is it only allowed 20 steps, 50 steps or fixed number of steps that is called finite horizon okay.

Horizon is the number of steps. If I say you are allowed to go for infinite time you just keep moving in the world forever and ever and ever. That is called infinite horizon. Google is an infinite horizon agent. It is always there. It is expected to be always there, right. It always lives, that is an infinite horizon. Then there is a horizon point indefinite horizon. Indefinite horizon says, you have a goal to achieve.

The goal is absorbing, so once you achieve it, you are done. But I cannot tell you whether you will achieve it in 20 steps or 30 steps. So you keep going, keep going until you achieve the goal. When you achieve the goal you stop, humans are indefinite horizon problems. I do not want to say it, but our goal is to die and do well before dying. So, if you are modeling a problem where there is a goal that is the end of your you know agent hood.

Then that is called an indefinite horizon problem and again a Markov decision process can handle all of this okay, there are different theoretical properties for them, but what we will learn will probably will handle everything. Now, there is one thing that I have not talked about and that is the idea of discounting or undiscounted. So, I am not saying maximize expected reward, I am saying maximize either discounted expected reward or undiscounted.

So, what is this discounting business. Now, suppose I can do action right and that gives you 1 rupee and I can do action left and that gives you 2 rupees and you can keep doing the action right or action left that is your problem for infinite time which action is better right or left intuitively which action looks right, right or left. Right gives you 1 rupee, left gives you 2 rupees. So, which action is better left.

But what is the long term reward for taking right infinite times, come on one times infinity and what is the expected value of taking left infinite times 2 times infinity. So, therefore, we know that left is better than the right. But if we allow the model to work over infinite time, then it leads to unbounded value functions. And if it leads to unbounded value functions that we cannot separate between left and right and that is not good.

So, what we are going to do is that we are going to again use an intuition from economics will say, money today is worth more than the same amount of money tomorrow. Why because I can invest it in a bank and I will get a small return on top of it. So, therefore, money today is r then money tomorrow will be gamma times r, I would have lost the interest of one day on that amount of money.

And that is called a discount factors. So, discount factors are used to keep the total reward or total cost finite. These are specifically useful for infinite horizon problems. The intuition is money today is worth more than the same amount of money tomorrow. And how we compute our long term reward is by saying r 1 is the reward at time step 1, r 2 is the reward at time step 2, but when we are adding, we will add gamma times r 2.

For r 3 we will add gamma squared times r 3 and so on so forth. So now, for the right action my total reward is 1 + gamma + gamma square + gamma cube up to infinity, which is 1 upon 1 - gamma for the left action it is 2 + 2 gamma + 2 gamma square and so on, which is equal to 2 upon 1 - gamma and as long as gamma is not 1, but less than 1. This action will come out to be better than this action, which is what we wanted.

So therefore, the objective of an MDP is to find a policy that map's from states to actions, which optimizes discounted or unknown discounted expected costs or reward or reward minus cost given a finite infinite or indefinite horizon assuming full observability.