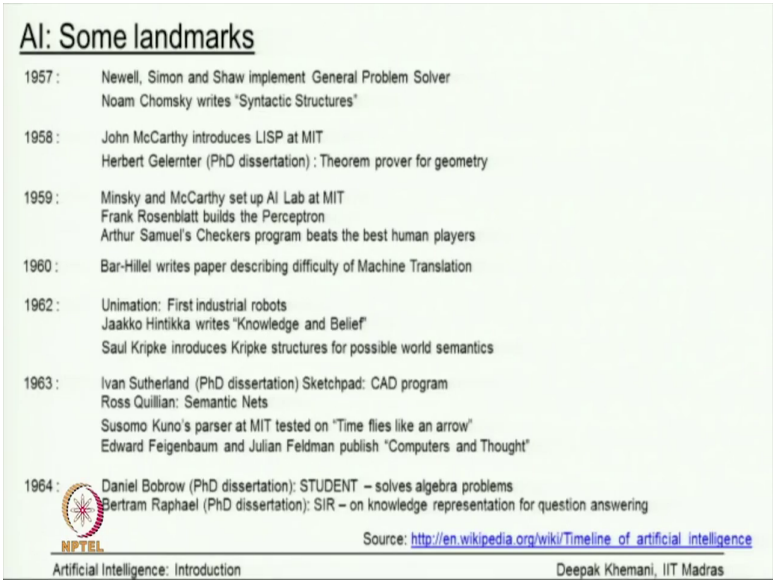
Artificial Intelligence: Search Methods for Problem Solving

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

Lecture – 14
A Brief History of AI

(Refer Slide Time: 00:15)

(Refer Slide Time: 00:24)



So let us move on to general AI and look at what has happened since the Dartmouth conference, before we come back to the main question again essentially. So, we will just quickly learn through some of these things; we have already mentioned the general problem solver by Simon and Newell, then Noam Chomsky wrote his book on Syntactic Structures which was very influential in natural language linguistics.

McCarthy introduces list Gelernter wrote a program for proving theorems in geometry; this was part of his PhD works. So, you will see a list of people whose PhD works appears in landmarks in AI; so it is a kind of a motivation for some of the PhD students sitting here essentially. So, Minsky and McCarthy set up the AI lab in 59; Rosenblatt build us Perceptron, the Perceptron was a single layer neural network which had this learning capacity.

So, the neural networks or programs which are basically perceiving this idea of learning or training as they called it; you show a system a sequence of patterns and it will learn to recognize that pattern. Samuel's programs beat the best players.

People also started talking about machine translation around that time; but one of the early reports were Bar Hillel said that machine translation is not something that we can do so easily, which is actually quite true. In this early 60s the first industrial robot started coming; Hintikka wrote his influential book on Knowledge and Belief, which talks about how can you formally reason about what people know.

So, I know that, you know that something you know this kind of a thing. So, Kripke introduced a formal model for this kind of reasoning called Kripke structures. There was progress in CAD, Sutherland; Quillian wrote the program for Semantic Nets. Kuno's parser on MIT was tested with this very well-known sentence time flies like an arrow essentially.

So, I would urge you to look at this sentence and try to pass it. Now the thing about us humans is that, we always quickly home into one parts of the sentence; because we are like Kant said predisposed to certain ideas about this world essentially. So, when we hear a sentence like time flies like an arrow, we do not even think that it could have any other meaning than that time is going passed very quickly essentially; but for a machine which is not biased by such predisposed ideas, there could be other meanings as well.

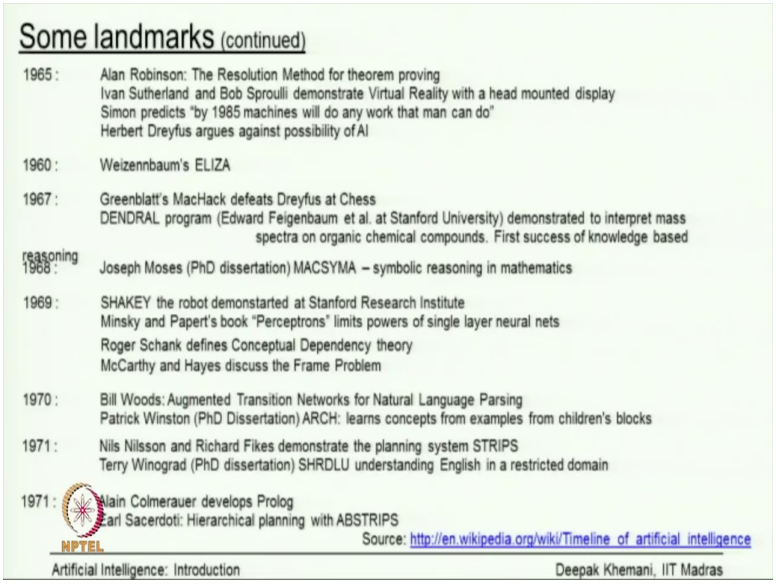
So, this is an ambiguous sentence; if you look at it from the fundamental point of view, it could have other meanings and words could have other meanings. So, one of the problems in natural language processing is that, the language, our languages are so rich that we can say the same thing in many different ways.

But also conversely something we say can have many different meanings as well essentially know; which is of course very useful for politicians, because they say something and then they say that this is what I meant or you are quoting me out of context or something like that essentially.

So, this sentence time flies like an arrow, I would urge you to look at other meanings or look up the web and try to see what other parts structures. So, when you say parts like; parts is we mean you know subject, object and you know noun phrase, verb phrase and this kind of structure essentially.

So, it is time adjective or is it a verb you know, you can look at some of those options essentially. A very influential book called Computers and Thought by Feigenbaum around that time. There was programs to do, you know we saw a program for geometry, then we saw program for algebra by Bobrow, a program to answer questions for people; you can put all these things together and build a system which can you know teach somebody let us say algebra or geometry or something like that.

(Refer Slide Time: 04:54)



In 65 Alan Robinson introduced this very this method called the resolution method for theorem proving, which revolutionizes this whole idea of theorem proving. So, by theorem proving we mean the kind of thing that LT was doing, logic theorems was doing; that given a set of axioms or given a set of premises, what are the things that you can prove from there essentially. And there were other developments; Herbert Simon predicted in 1965 that by 1985 machines will do any work that man can do essentially.

So, this very optimistic people; we have seen that Alan Turing also said that by 2 thousand the Turing test would be passed essentially, none of that has really happened. So, at the same time Herbert Dreyfus argues again AI; we have seen Eliza earlier.

In 1967 Dreyfus was bea10 by a chess program at chess; also there was this program called Dendral which was one of the early successes of AI, which could do chemical analysis for us. So, we will not go into the details now, but later on in the course we will see what Dendral did; but it was it operated at the level of an expert chemistry.

Then Moses theories work on symbolic reasoning in AI which is now so common essentially; that everybody uses symbolic integration, packages, you know MATLAB, maxima and all these kind of packages. Shakey the first robot appeared in SRI Stanford Research Institute; Minsky and Papert wrote this book on Perceptrons, we mentioned perceptrons, they are single layer neutron. What Minsky and Papert showed in 1969 was that, perceptron was limited to recognizing only certain kinds of patterns essentially.

And the kind of patterns that the perceptron could recognize what patterns that were linearly separable; which means that if you were to plot them in some space, let say 2 dimensional or 3 dimensional space or let say 2 dimensional space, then you could draw a straight line which would say one side of the line is class a and the other side of the line is class b. So, such patterns are called linearly separable patterns. And what Minsky and Papert showed was that, that is all the perceptron could do.

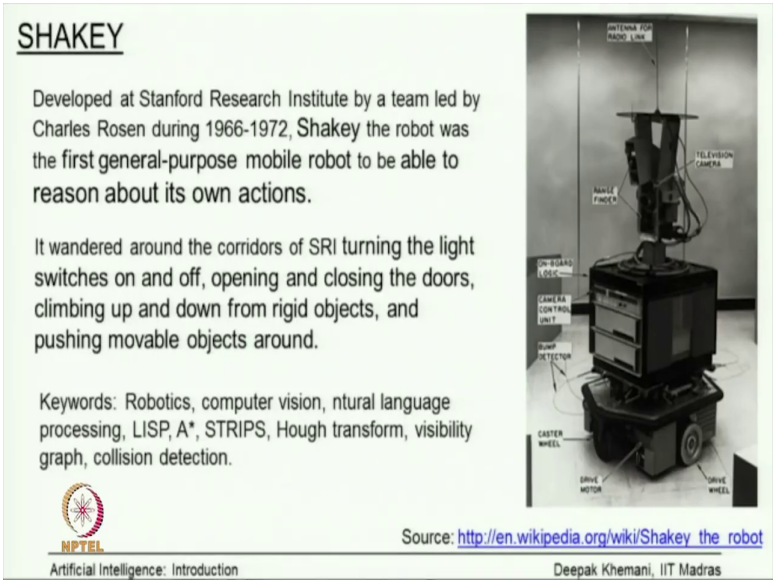
And it is it is said that this skilled the research in neural network for quite a while; it was not a till the mid eighties that people started getting interested in neural network again. That is when they realized that multi layered neural networks have do not have this limitation; that they are not limited to finding only linear separators, so but that is what happen in 69 essentially. So, Schank talked about his conceptual dependency theory; McCarthy talked about the frame problem essentially.

The frame problem says that, if you are representing about a world and if you are reasoning about change in the world; how do you figure out what is not changed essentially? So, if something was true, if my watch is lying on the table here and I look that side for a few minutes and then I look at this again; is the watch still there? Of course, I can see it is still there; but in the reasoning system after sometime, after 2 hours will the watch still be there? Of course, they had a different example about a loaded gun, which we will not get into right now.

So let me see what are the other stuff. So, Nilsson and Fikes demonstrated this planning system called STRIPS, we will see STRIPS later in this course. And Winograd wrote a program called SHRDLU which could do natural language conversation in a well very limited domain, the domain of blocks world. So, it could converse with a person take instructions and do things. So, you should look up SHRDLU on the net and you will probably get a sample conversation.

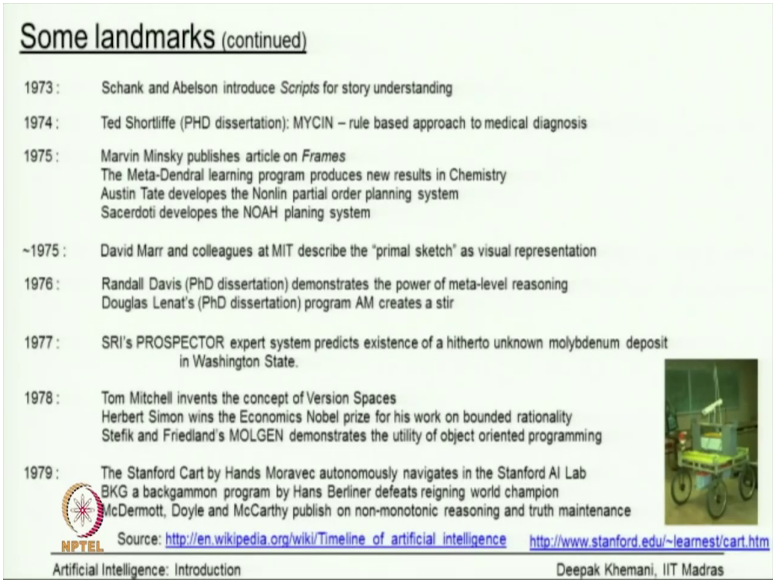
So, you can say things like pick up the green block and put it on top of the red one and it would do that; it will understand what you are saying and do that, in a sense it would generate a plan for doing that essentially, it is quite interesting. Around that time this language prolog was deviced by this French man Colmerauer and some more work on planning for example, ABSTRIPS essentially.

(Refer Slide Time: 09:37)



So, here is a picture of Shakey, this was the first autonomous robot that was built which could take it is own decisions. So, it could wander around the corridors of SRI which is Stanford Research Institute and do some limited things like open doors and plug it iself or getting charged and things like that essentially. And so, Stanford was another hotspot of AI activity; people like Nilsson and many others were there. And some of the names that we associate with Shakey list is, LISP of course came by McCarthy; but algorithms like A star and STRIPS and visibility graph and collision detection they were all discussed at Stanford.

(Refer Slide Time: 10:27)



So, let us move on. So, in early 7 ties, Schank and Abelson introduce a idea of a Scripts, they say they if you want to understand; remember this Schank says that we perceive the world in terms of pre-defined a priori as we call them. What are those a priories? Schank and Abelson say that, these are like scripts; so it is like a movie script that you always follow a script when you are acting. So, if you, so the standard example that they used were restaurant essentially; if you go to a restaurant, you essentially following a script, that this is how typically things are done in a restaurant.

So, if you hear a story about somebody going to restaurant, then you can understand that story if we have the script in your head essentially, so that was the idea of scripts. We have already talked of Dendral, another success was MYCIN; it was a rule based what people called as expert systems and the eighties was the time and people wanted to build lots of expert systems. By expert systems they meant, a system which will capture the knowledge of

an expert; typically in a rule form we will see this later in the course and perform at an expert level essentially. So, MYCIN was the program which did medical diagnosis.

Minsky wrote about his frames advance version of Dendral called Meta Dendral had some learning capabilities; there were some more planning systems, there was more work on visual representation.

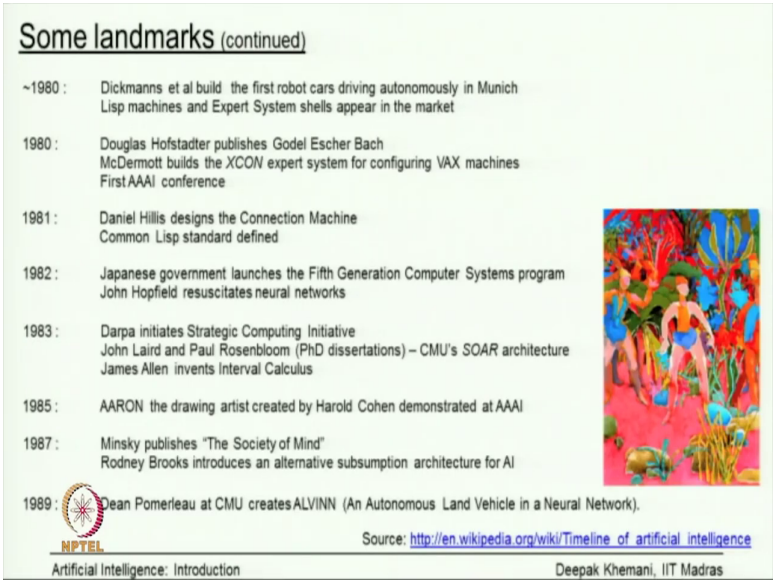
The power of meta level reasoning for example, Randall Davis we see here another PhD thesis. Lenat's program called AM, AM stood for applied mathematician; created quite a stir when it was first published essentially. So, if you look at some of those papers which talk about AM, it was a program which Lenat claimed was doing mathematical discovery; that you give it basic concepts and it will learn new things.

So, concepts like prime numbers, the concept of frame numbers ok; not an algorithm to find prime numbers something. What is the notion that some there is something called prime numbers Lenat claimed his program could learn? And essentially his thing was that it followed the heuristic that go towards extremes of certain kind and one extreme is that, the number of devices that a number can have

So, one extreme is 2 essentially; you cannot you cannot have less than 2 devices, except you know one and it iself, counting one and it iself. So, the smallest number is 2. So, those numbers are interesting. So, Lenat claimed that his program was doing that; but has it of10 happens, people discovered that a lot of things for sort of encoded into his learning program it iself essentially. So, there and that is something that we have to be vary about essentially.

Now, we write a program and we put it our knowledge somehow which is embedded into the program and then we say this system has discovered that knowledge. So, we are talking about expert systems; the system called prospector which was built at SRI for prospecting the national world found some unknown deposits of molybdenum you know in Washington State. So, Mitchell invents the concept of version spaces and so on and so forth; we will skip some of these things.

(Refer Slide Time: 14:10)



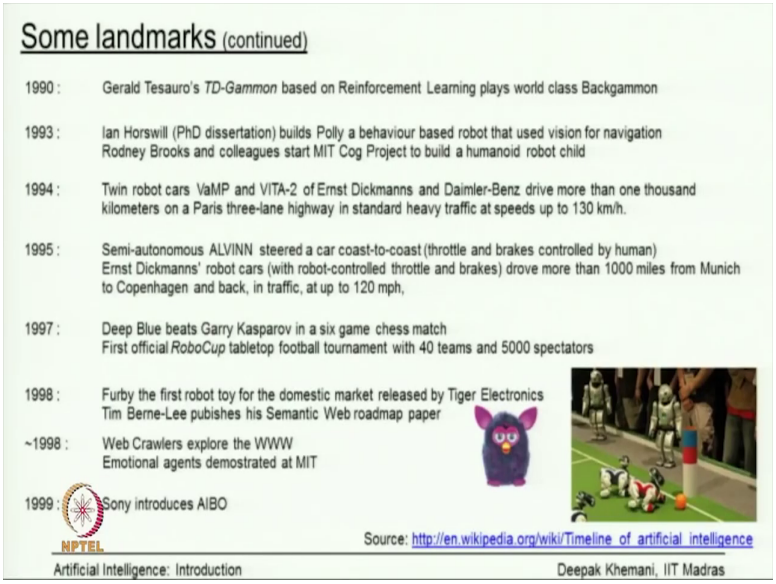
And so, you can read them from the slides or from the Wikipedia page essentially. So, I want to point out to some of these newer things which were happening. So, one new thing was happening around this time was autonomous vehicles essentially. Can you have a car which will drive it iself? So, as early as 1980s, people were building cars which would navigate themselves on the road.

So, in Munich the first car was done, then expert system shell started appearing in the market; another expert system called XCON for configuring VAX systems was developed. The AAAI conference, the AI conference started, series started. Hillis designed the connection machine which was made, supposed to be a machine of thousands of processors connected together; little bit like our own brains, simple processing units connected together.

So, one interesting thing was this program called AARON developed by professor Harold Cohen who is still around, which could make drawing. So, this drawing which you see there; painting that you see there, it is been created by a program. And when you say created, we mean visualized and drawn; it is not as if somebody has told the program draw a man or draw something, the system has drawn this whole thing it iself.

And if you look up AARON on the web, you will see a whole gallery of his pin, it is paintings essentially. Actually I tried to get AARON to draw the cover for my book; but Professor Cohen said that no it is not active any longer. So, some more work.

(Refer Slide Time: 15:57)



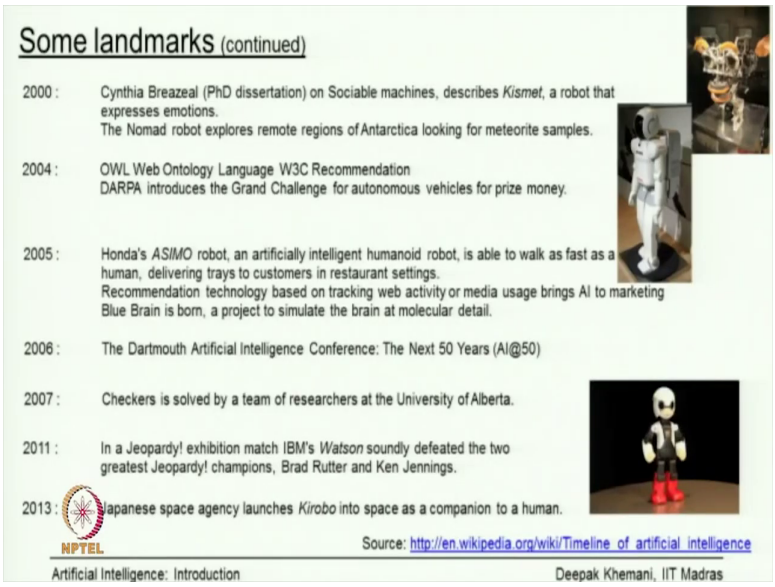
So, in 1990 this program called TD Gammon which played the game of backgammon, using this technique called reinforcement learning; this is a course learning right now in the department, be it the became the world champion in backgammon essentially.

So what else is, so we have these 2 robot cars in 94, 95; if you look at 94 and 95, we have descriptions of robot robotic cars which can navigate themselves. So, there are these cars which one went from Munich to Copenhagen and came back all by it iself essentially; you know driving, breaking, accelerating, avoiding vehicles and so on and so forth and we already seen that 97 Deep Blue beats Kasparov.

The RoboCup tournament started, these are football tournaments for robots. Robotic toys like this Furby that you see there by tiger electronics came into picture. So, these are toys which could, Furby apparently could learn your language; you could teach it how to speak this thing. Sony introduced this AIBO; AIBO is that small dog like robot that you can see in the picture on the bottom, right. So, there is 2 small dog like creatures; they are these Sony AIBO robots, which became very popular.

In this picture they are participating in a RoboCup football tournament. So, you can see an orange colored ball there and you can see some different humanoid robots hanging around in the side line.

(Refer Slide Time: 17:45)



So, you see this picture, it is a robot called Kismet; another PhD dissertation from MIT, a robot that expresses emotion. So, again if you go to the web and look for Kismet; you will see videos of Cynthia talking to this robot and this robot smiling and you know making all kind of human expression, human like expressions in the system.

The language owl web ontology language was standardized; DARPA introduce the grand challenge in prize money. Honda produced this robot called ASIMO which you can see the small knee length robot which could walk around at human pace. So, it is not easy problem to make a tool like a creature walk, keep it is balance and walk and things like that and it became very popular.

Recommendation systems started coming into play like Amazons recommendations; if you go to Amazon or Flipkart and you look at a book, it will all immediately make some

recommendations to you that people like you bought this as well things like that. Collaborative filtering for those of you attended Anand's talk yesterday, he talked about collaborative filtering.

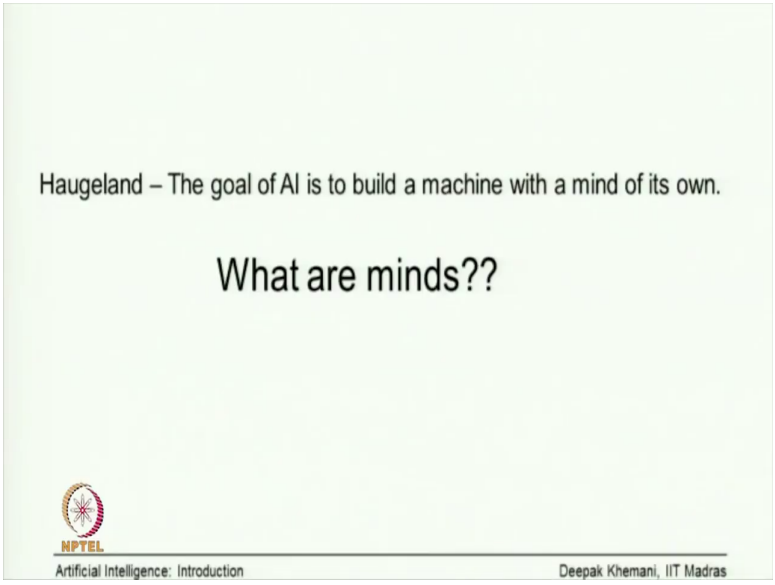
Recommendation systems; when there was another Dartmouth conference in 2006, which said what is AI going to be in the next 50 years essentially. Checkers, the game of checkers was solved. What do we mean by solved? Just as by solved we mean that, if both the players are playing perfectly, we can tell you what the outcome will be essentially. So, we can do that for things like smaller games like cross and noughts or tic tac toe as it is called; but for checkers it was done quite recently using extensive computing essentially.

So, we know that checkers is a drawn game or maybe it is a win for white I do not know. Then much more recently IBM's program called Watson, which participated in this game called Jeopardy which is look up on the web beats the world champions at this game essentially. So, Jeopardy is a program which in which you need lot of general knowledge, geographic knowledge and things like that. So, this program had access to all this knowledge and it could play this game better than human beings essentially.

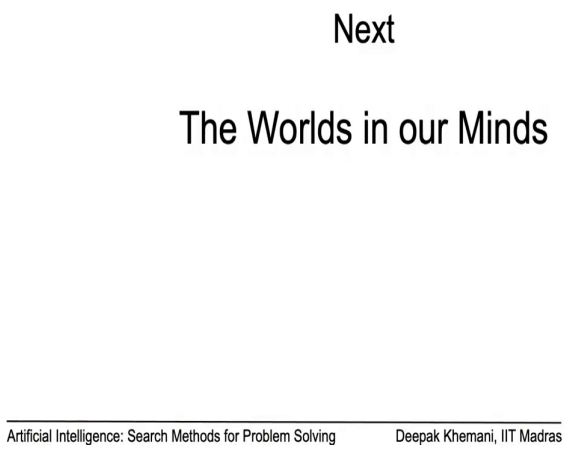
And we mentioned last week this robot called Kirobo, which is what David Levy was talking about; human companions to human beings essentially. So, this robot was sent into space last week by the Japanese space agency and it is meant as a human companion to human astronaut which will follow in November essentially. So, this is a very brief history of what has been happening in the last fifty years.

We will stop here and the next thing that we will follow is come back to this question about; what are minds?

(Refer Slide Time: 21:00)



(Refer Slide Time: 21:06)



So, that will be the last segment of our introduction to this essentially.