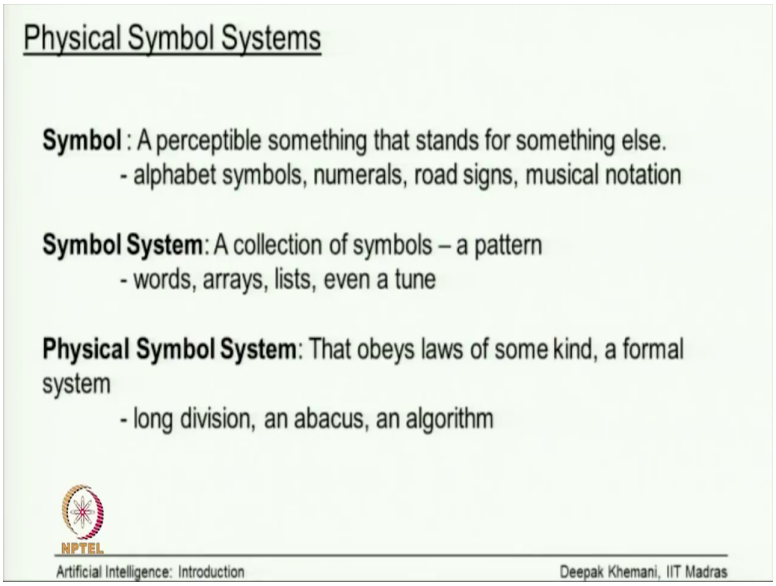
Artificial Intelligence: Search Methods for Problem Solving
Prof. Deepak Khemani
Department of Computer Science & Engineering
Indian Institute of Technology, Madras

Lecture – 13
Introduction (2013)
The Chess Saga

So, one of the things they talk about was this architecture for AI, what do you need for AI.

(Refer Slide Time: 00:25)



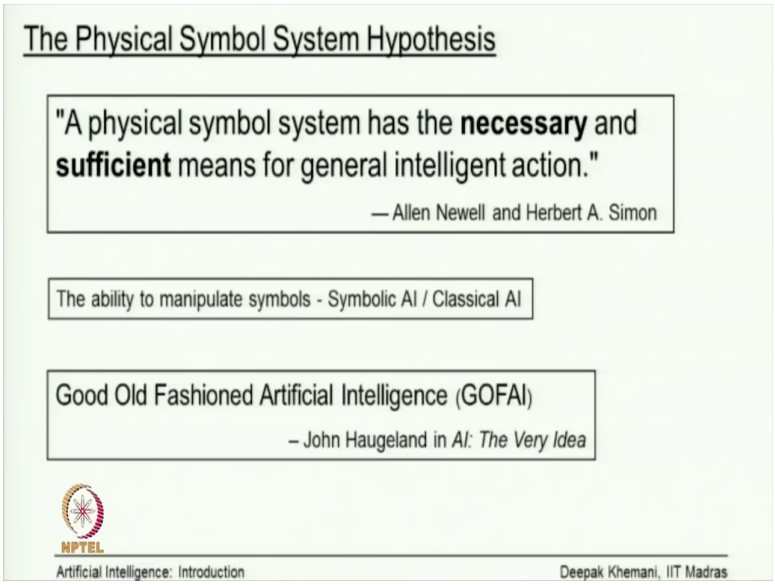
So, you have this idea of the physical symbol systems and a symbol is something as for as we are concerned a perceptible something which stands for something else. So, a symbol stands for something else. Essentially if you write the numeral 7, it stands for the number 7. Of

course, it is not the number 7, it just stands for the number 7 and we could have in a different script we could have written it differently.

A symbol system is a collection of symbols. So, for example, a data structure or English language word or even a musical tune essentially. So, you put them together you have a symbol system. So, you have an alphabet which is made of symbols and then you put together things of from that alphabet and you have a symbol system and a physical symbol system is something which obeys laws which are like the laws of physics essentially ok.

So, in some times if they if you can manipulate them using well defined laws or rules then they are physical in that sense they are physical in the sense that they can be manipulated according to this law. So, anything you can use algorithms or the procedure for long division for example, and so on and so, forth.

(Refer Slide Time: 01:30)



The important statement that they made is known as the physical symbol system hypothesis. It says that a physical symbol system has a necessary and sufficient means to generate intelligent action.

So, what they are saying that all you need in our terminology; all you need to build intelligent systems is the ability to create data structures and write algorithms which will operate upon those data structures you did nothing else essentially that is the basic infrastructure you need.

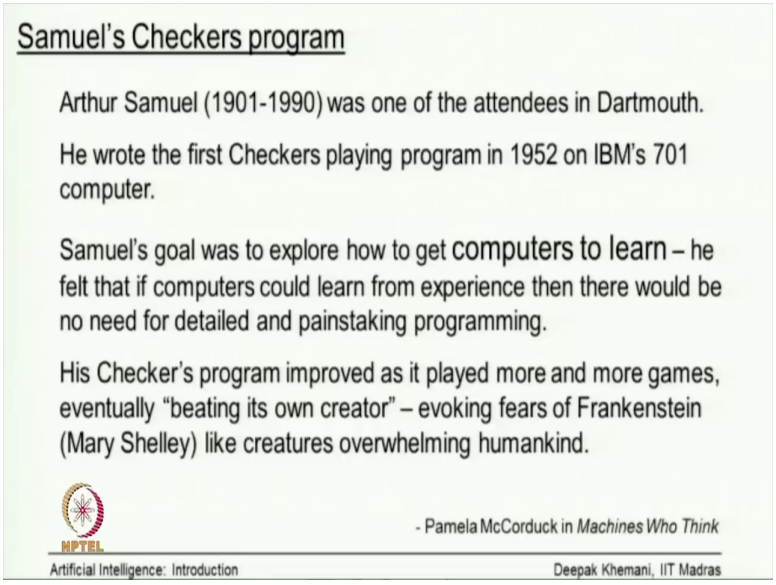
So, unlike for example, Roger Penrose who feels that the human mind a human brain has some kind of physics which is going on which we cannot replicate, they said nothing of the sort if you can do information processing which means if you can operate on symbol systems using well defined algorithms, you can create intelligent behavior.

So, this is known as a symbolic AI or classical AI. Classical AI follows this principle that its a top down design approach to building intelligent systems, that you will create your data structures and you would like your algorithms and you will produce intelligent systems or has Haugeland calls it good old fashioned AI essentially.

So, as opposed to symbolic classical AI, we have what sometimes we call a sub symbolic systems or signal level systems. Systems like the human brain which operates at a level where information is not encoded into symbols. So, if you look at a neural network for example, it is made of many nodes and many edges connecting nodes and edges have values which are numbers essentially your weights as we call them.

And everything that the neural network knows is encoded in terms of those weight essentially, but the weight itself does not stand for anything it does not mean anything to us. In that sense it is not a symbol a symbol should stand for something whereas, if I write a let us say program in which I have a variable called x which stands for let us say the distance from place a to place b, then its some its a symbol because it stand for something. So, symbolic AI or classical AI is concerned with explicit representation and algorithms for working on those representations and that is what we are pursuing here ok.

(Refer Slide Time: 03:47)



So, we had mentioned the Samuel's checkers program he was also one of the participants in this Dartmouth conference and his contribution was this program to play checkers and he wrote it in this IBM 701 machine and as I said a little bit earlier, his goal was to explore learning for computers his idea was that if computers could learn, then there would be no need to do all this painstaking programming.

And when he says programming in a painstaking fashion, he really means that because in those days we did not have this whole set of high level languages that you are so, comfortable with nowadays, you have the assembly language and maybe you I am not sure whether even Fortran was devised at that time LISC was devised a little bit later than the Dartmouth conference.

So, programming was very painstaking in those days and Samuel said that if you can somehow make the machines learn, then we do not have to program them essentially and you know that is a fascinating seductive idea that you build some system and it will learn and become efficient by self and of course, we have a thriving machine learning community today.

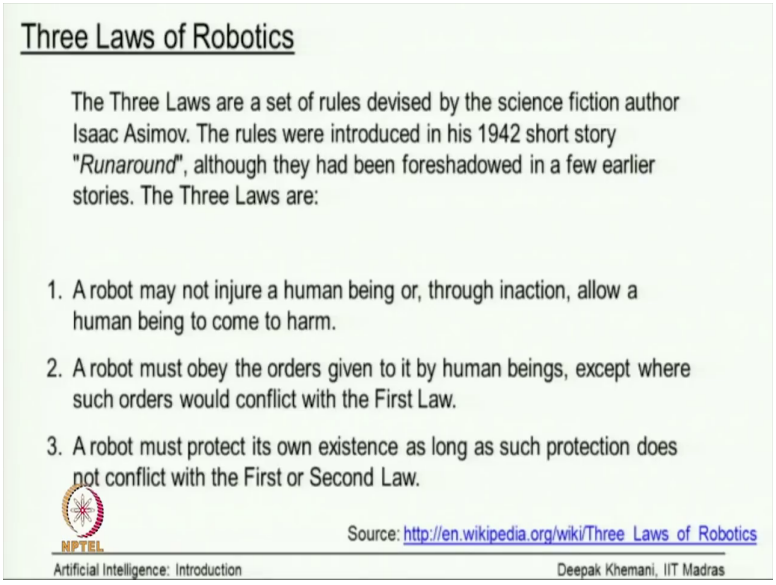
So, his checkers program Samuel was not a great checkers player he was a computer scientist, but the story goes that he wrote this program the program became better and better and better and eventually it beat its own creative as we say.

You know this around the time when Babbage was constructing his machine and we have mentioned his collaborator Ada countess of Lovelace who was the daughter of lord Byron and lord Byron had this friend called Peter Shelley. And Peter Shelley is maybe wife was Mary Shelley who wrote this novel called Frankenstein and this novel I do not know whether you have heard about it some of you might have read it is a its about an artificial creature you know people were fascinated with artificial creatures.

So, this novel is about this artificial feature creature called Victor frank Franken Frankenstein who was made by a doctor and eventually it became becomes like a monster who you know destroys a doctors essentially. So, that whole idea that these machines will become smart and you know overpower us and kill us has always been around for a long time.

So, you have seen the movie matrix you know. So, that is one of the films and we will come to matrix again later for a different reason. So, quite an interesting film, but one of the things in matrix is that this machines are trying to control human species essentially. So, you do not always realize it when you are watching the film.

(Refer Slide Time: 06:49)



So, in 1940 sometime Isaac Asimov who was a science fiction writer he introduced this so, called laws of robotics which became very popular it took the fancy of all the people around he wrote it as part of a short story called Runaround which he wrote in 1942.

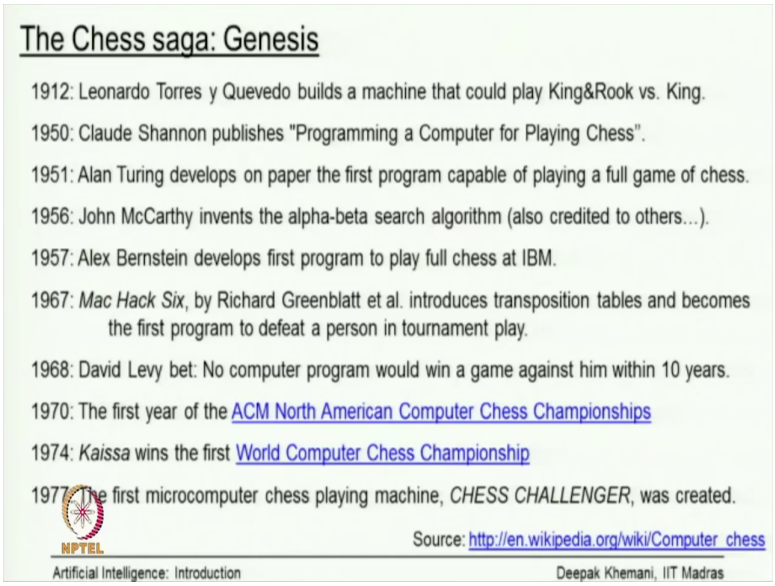
So, there are these three laws that Isaac Asimov said. So, laws in the sense that not like Newton's laws which we discover that the physical world is obeying, but more like laws that human beings make you know legislation by the parliament and so, on essentially. So, for example, our parliament is making a law that they cannot come under RTI essentially. So, this kind of laws essentially.

So, his law was basically this three laws that the robot will not injure a human being that is the first law. You know protecting the human species do a robot must always be built in such a fashion that it will never harm a human being essentially. Secondly, it will obey human beings

at all times unless it violates rule 1. So, if I build a robot and I tell it to harm another person then the robot should not obey it.

And thirdly it must protect its own existence as long as it does not violate the first two laws essentially. So, it you know all this is there in the popular imagination because people are worried about machines overtaking human beings already for 4 500 years they have been looking at talking moving heads which they think are thinking. So, chess as we know for many reasons has been a fascination for computer scientists ever since computers were invented.

(Refer Slide Time: 08:42)



So, let us look at a quick history of chess first and then we look at a history of the rest of the computing world before we come back to this question of what are minds and so, on.

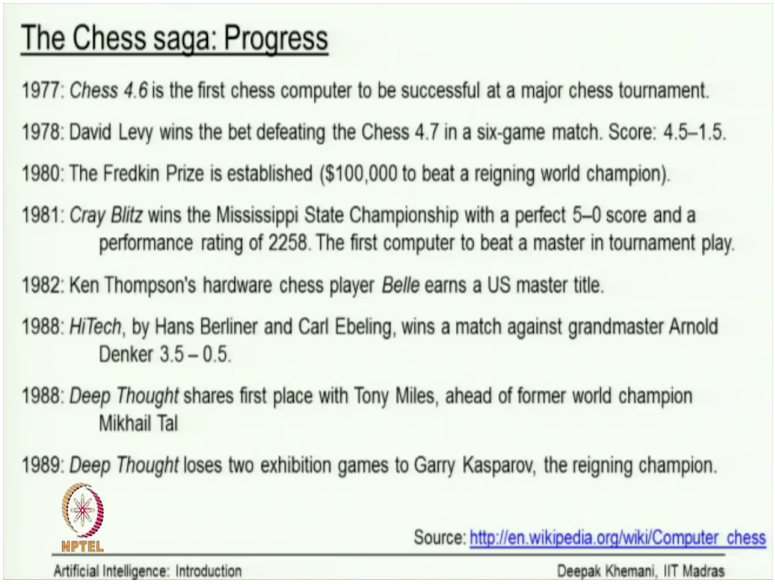
So, you see all these names here and all this all this material is been taking by taken from the Wikipedia. So, you just have to go to Wikipedia, computer chess and you will see this timeline or part of this timeline. So, Claude Shannon first wrote a paper called programming a computer for playing chess, Turing developed a on paper an algorithm for doing it, McCarthy invented the alphabet algorithm that we will study, but other people also said to have invented it including Samuel essentially.

Bernstein Alex Bernstein who worked in IBM was the person to who wrote a first actual program which could play chess complete programmer which could play a complete game of chess.

Then a program called Mac Hack six by Greenblatt was the first program to defeat a person in tournament play essentially and we have already mentioned David Levy who made this bet in 19689 that no machine can beat him in 10 years and in 2008 he wrote this book called love and sex with robots essentially I mean he swung from one end of the capabilities of machine intelligence to the other end that they could be like human companions to us.

In 1970 the American computer chess championship started, in 1947 a program called Kaissa from somewhere in the USSR won the first or maybe it was from Germany sorry world chess championships and a small microcomputer started playing chess in 1977.

(Refer Slide Time: 10:31)



So that was the initial years then in the late 70s they started making progress. So, a program called chess 4.6 played well in a major chess tournament, 1978 David Levy won his bet defeating a program called 4.7. In 1980 a Fredkin Prize was instituted hundred thousand dollars, in 1980 was quite a bit of money for beating a reigning world champion a program called cray blitz.

Now, cray was these you know supercomputers which are built by Seymour cray and they were the fastest machines for a very long period of time the best and the fastest machines were cray machines and this cray blitz was a program running on those machines. So, it wins the Mississippi state championship with a score of 5 0 which means it won all the games.

The performance rating for those of you who are familiar with chess ratings 2 to 5, 8 which is quite good and it was the first program to be the human master. Around that time people

started investing in special purpose hardware to play chess essentially. So, for example, high tech which was also developed around this time in 1988 at CMU had 64 processors, one processor dedicated for each square of the chessboard and that kind of thing. So, Ken Thompson had a hardware chess.

So, people are investing more and more in hardware at the same time. So, then a program called deep thought shared the first place with tony miles ahead of the former world champion not the reigning world champion and Mikhail Tal essentially. So, deep thought is a name which does it ring a bell for anyone?

Student: Hitchhiker.

Its hitchhikers guide to the galaxy essentially. So, the name has been taken from there..

(Refer Slide Time: 12:41)

It lost two games to Garry Kasparov in 1989 and after that the years of Triumph for the machine. In 1992 a microcomputers program one or computer chess championships playing against mainframes and supercomputers a program called chess genius actually defeated Kasparov in a game.

In 1996 a program called deep blue which was also developed at IBM lost to Kasporav, but in 1997 it beat Kasparov in a 6 game match and its creators or inventors were actually awarded the Fredkin prize that we mentioned some time ago, then afterwards it was a series of losses for the human players. Kramnik drew an eight game against a program called Deep Fritz, then a program called Hydra beat Adams 5.5 is to 0.5 which means Adams could only draw one game.

Then Kramnik lost to Deep Fritz later and nowadays it is routine for the human players including Viswanathan Anand from India to constantly take help from machines while preparing their own chess playing strategies. So, here we have mentioned in 2010, Topalov prepares by sparring against supercomputer blue gene essentially.

(Refer Slide Time: 14:20)

Next

A Brief History of AI