Oral History of John McCarthy

Interviewed by:
Nils Nilsson

Edited by:
Dag Spicer

Recorded: September 12, 2007
Mountain View, California

CHM Reference number: X4205.2008

Nils Nilsson: Today [September 12, 2007] we'll be having a conversation with John McCarthy, Professor Emeritus in the Computer Science Department at Stanford University. McCarthy received the Association for Computing Machinery's Turing Award in 1971 for his major contributions to the field of Artificial Intelligence. McCarthy coined the term Artificial Intelligence in his 1955 proposal for the 1956 Dartmouth Conference. McCarthy received his B.S. in Mathematics for the California Institute of Technology in 1948 and his Ph.D. in Mathematics from Princeton University in 1951. McCarthy is well known for his efforts to give computers common sense, based on the use of mathematical logic and various extensions to it. He's made major contributions towards what he calls "Human Level Artificial Intelligence." McCarthy has also done extensive work in the mathematical theory of computation, especially proving the correctness of programs. He invented the LISP programming language, and made the key proposals and suggestions that led to timesharing. In addition to the Turing Award, McCarthy received the first Research of Excellence Award of the International Joint Conference on Artificial Intelligence in 1985, the Kyoto Prize of the Inamori Foundation in 1988, the National Medal of Science in 1990, as well as several other honors. He's a member of the Academy of Arts and Sciences, the National Academy of Engineering, and the National Academy of Sciences. Perhaps we can begin, John, by talking about what got you interested in science and mathematics in the first place.

John McCarthy: Well, I was interested from the time I was a kid.

Nilsson: In high school?

McCarthy: No, even before that.

Nilsson: Even before high school. Somebody told me about a book called the "100,000 Whys." Is that something that had an influence?

McCarthy: Yeah. That was a Russian book, and because of my parents' radical associations, in fact they were Communists, these translations of Russian children's things were popular, and a "100,000 Whys" is a pretty good book, a popular science book for children, and I read recently that it had been translated into Chinese, and some Chinese child prodigy had learned his science from that.

Nilsson: That book, is it still in print, do you think?

McCarthy: What?

Nilsson: Is it still in print?

McCarthy: Evidently, in China at least.

Nilsson: In China at least, if you read Chinese. Well, so then in high school, before high school, and in high school, you had an interest in science, and then you went on to Caltech?

McCarthy: Yeah.

Nilsson: And majored in mathematics there. Any interesting stories about Caltech? No?

McCarthy: No, there are, but I can't...

Nilsson: Well, you went to the symposium, called the Hickson Symposium. Do you want to say something about that?

McCarthy: Yeah. This was right after I graduated from Caltech. I had some time out in the army, but my first year in graduate work in math was at Caltech. I moved to Princeton second year. But anyway, in September of 1948, there was this symposium on cerebral mechanisms and behavior, and it had various famous psychologists, including Carl Ashley [ph?], and another one who worked with apes whose name I'm forgetting now.

Nilsson: Kohler.

McCarthy: What?

Nilsson: Kohler?

McCarthy: Yeah, Wolfgang Kohler.

Nilsson: Wolfgang Kohler.

McCarthy: And they compared the computer and the brain, and many years later, when I got the Kyoto Prize, and they asked for a retrospective lecture, I went and looked at the proceedings of that Symposium, and I had supposed that artificial intelligence had been discussed there, that is the use of computers to behave intelligently, but at least there was no hint of it in the proceedings, but it was what gave me the idea.

Nilsson: So it played a role for you in getting you interested in the use of computers to imitate intelligence, even though at the Symposium nothing much was said about that.

McCarthy: Well, we haggle about the word imitate. Using computers, making computers be intelligent, since we relate intelligence to the ability to do certain things, and independently of whether what does it is human, or a Martian, or a mechanism.

Nilsson: Well, that brings us a little bit farther forward, but I remember you saying some things about being interested in the general science of intelligence, independently of how it's mechanized.

McCarthy: Yes. I can't say that I got very far with that.

Nilsson: There's certain principles about what it would mean for there to be a science of intelligence.

McCarthy: Yes.

Nilsson: Now at Caltech, did you have any contact with computers? These were during the late 1940s.

McCarthy: First of all, there was no computer there. The first computers were built in 1948 and 1949 in England and in the US, and I did attend a few lectures on programming for a computer that was not yet built, and later contemplating, when I actually did learn to program in 1955, I could see that I had acquired some misconceptions in 1948, and if I had been given the opportunity to try to run my 1948 program, it wouldn't have worked.

Nilsson: Well, that kind of programming was, what, all in machine language, right?

McCarthy: Yes, I think so.

Nilsson: Because it was certainly before Fortran.

McCarthy: Oh, yeah, but before there were actual translators, people would write in a language that wasn't a machine language, and then would hand translate it to machine language. The Russians had a big system for that.

Nilsson: And then so after one year of graduate work at Caltech, you decided to go to Princeton. How come you just didn't decide to continue at Caltech?

McCarthy: Princeton was the best place for mathematics.

Nilsson: Let's see, there were various famous mathematicians there, von Neumann, for example, but he was at the Institute.

McCarthy: That's correct.

Nilsson: And then did you get to start working right away on a Ph.D., Ph.D. research?

McCarthy: Yeah. When I arrived at Princeton in September of 1949, the Department Chairman, Solomon Lefschetz said to me, "What are you going to do your thesis on?" and I said, "I don't know. I'll think about it." And so I immediately got myself a notebook, and started thinking about ideas for a thesis, and before the year was up, I had decided on a topic, and this made me relatively popular, because the usual thing is that the students pester the professors to suggest topics.

Nilsson: So you knew about your topic, but didn't have anything to do with computers?

McCarthy: That's correct. In a vague sense it did, in that I got interested in the notion of an integrating factor in differential equations, and the question that I asked myself is, what is an approximate integrating factor? And then what this led to was a considering that the gradient of a solution to a differential equation, first order differential equation, belongs to two spaces. One is the space of gradients, and the other is the space of a vector satisfying certain algebraic equations, and that if one projected back and forth between those two spaces, according to a theorem of von Neumann, the process would converge to the gradient of a solution. That was my thesis, figuring out what those operators were. So, in some principle, it could have been a method for solving differential equations, but I couldn't, at that time, show anything about how fast it converged. It was maybe 30 years later before I was able to give an example, and show that it converged very, very slowly.

Nilsson: So the problems stayed with you for those 30 years, and then you went back to it.

McCarthy: Yeah, I thought about it from time to time.

Nilsson: Now, you mentioned von Neumann in connection with doing a theorem that your idea made use of. Did you talk with von Neumann about all of this while you were there?

McCarthy: I only had one interview with von Neumann, in which I told him some idea that I had about artificial intelligence, representing a brain and its environment by finite automaton, and he encouraged me. He said, “Write it up. Write it up.” But afterwards, after my conversation with him, I came to the conclusion that it was in fact a bad idea, because one couldn’t represent knowledge in terms of the structure of the finite automaton, which was to represent the brain, so I didn’t write it up. Somebody else re-invented it, and published it, and, yes, it was a bad idea in his hands also.

Nilsson: It was a bad idea then, and still was.

McCarthy: Yeah.

Nilsson: Yeah, well, it sounds vaguely like some of the things that Brooks did with his finite state machines controlling these little creatures.

McCarthy: I don’t know about that.

Nilsson: So you didn’t do anything then about artificial intelligence, except for that, and you stayed on after getting your Ph.D., you stayed on at Princeton for a while?

McCarthy: Two years, as an instructor.

Nilsson: And now, let’s see. Then, you had told me previously you spent some time at Bell Labs, a summer some time around there?

McCarthy: The summer of 1952, yeah.

Nilsson: And there you met Shannon?

McCarthy: Yep.

Nilsson: And you said that, let’s see, that you and Shannon maybe collaborated on doing a volume of papers that you thought would be...

McCarthy: Collecting a volume of papers, yes.

Nilsson: And that volume was automata studies?

McCarthy: Yes.

Nilsson: But let's see, you said you weren't too happy about what all the papers that were submitted?

McCarthy: Well, I was hoping that people would write about artificial intelligence, and one or two did, but a number of them wrote just about automatons, and the most famous initial paper on automata, which was Stephen Kleene's appeared in that volume. So I really didn't have any right to complain.

Nilsson: It had a famous paper in it. But now, the phrase "Artificial Intelligence," you had proposed later. What phrase were you using to describe your interest in those days in this field that would become artificial intelligence?

McCarthy: Now, I didn't have a phrase, but when I wanted to have this Dartmouth summer workshop, the idea was that people would come for the summer, and work on artificial intelligence. I had to call it something, so I called it "Artificial Intelligence," and I had a vague feeling that I'd heard the phrase before, but in all these years I have never been able to track it down.

Nilsson: Now before the Dartmouth Conference, you spent some time as an assistant professor at Stanford, in Mathematics?

McCarthy: Yep.

Nilsson: But anything going on with your-- did you pursue any AI interests during those days?

McCarthy: Certainly, I didn't write anything. I wrote on differential equations, and analysis, and Stanford decided they'd keep two out of their three acting assistant professors, and I was the third.

Nilsson: So then Dartmouth was the winner on that one.

McCarthy: Perhaps it won. Perhaps it lost.

Nilsson: So you went to Dartmouth as an assistant professor of Mathematics, and it was there at Dartmouth that, was it you and Shannon, and some others decided to propose this Dartmouth summer school?

McCarthy: Well, of course they didn't come up to Dartmouth. I went down to MIT, where Shannon had moved to, and the other person who was instrumental was Nathaniel Rochester of IBM, who was the head of their information processing department.

Nilsson: You had met him earlier?

McCarthy: I met him because MIT, sorry, IBM decided to give an IBM 704, and the idea was that it would be one third to MIT, one third to other New England colleges, and one third IBM would keep for its own use, and I was the Dartmouth representative.

Nilsson: Of part of the other New England colleges?

McCarthy: Yep.

Nilsson: And so that's how you met Rochester?

McCarthy: Well, he came up to Dartmouth as part of his tour, so to speak.

Nilsson: And so how did this idea about having this summer workshop, how did that originate?

McCarthy: Well, I had heard about these Defense Department workshops, and so I thought if one could get a lot of smart people, one could make substantial progress in a summer. It turned out to be very much over optimistic.

Nilsson: So people came for various periods of time, right?

McCarthy: Yep.

Nilsson: Were all of them who participated ever there all at once?

McCarthy: Almost certainly not.

Nilsson: So they were there for certain periods of time, and what kinds of things did some of the people do there?

McCarthy: Well, do there is one thing, report on is another. The people who were furthest ahead at that time were Newell and Simon, who came for just a few days, and I got the idea of Lisp processing from them, but did not at all like the language that they used.

Nilsson: That was IPL [Information Processing Language]?

McCarthy: That was IPL. I thought Fortran was much better, and so now one of the things that was done there was that Minsky came up with the idea that a theorem proving program should only try to prove things that were true in a model, and the example given was plain geometry. So you would have a diagram, and you would only try to prove things that were true in the diagram, and he wrote a little article about this, and Rochester decided that that was a good thing to do at IBM, and got a new Ph.D., Herbert Gelernter, to work on it, and had me as a consultant, and so I had these ideas about using Fortran for Lisp processing, and Gelernter, and his colleague Carl Gerberich developed it further to this Fortran Lisp processing language.

Nilsson: So they augmented Fortran with some Lisp processing facilities?

McCarthy: No, they didn't do that.

Nilsson: They didn't do that.

McCarthy: What they did was they made some Fortran functions for Lisp processing, but the overall structure was Fortran, and in 1958, when I spent a summer, a second summer, at IBM, then I decided to write a program for differentiation, and it became clear that the natural way to write it involved recursion, and Fortran did not allow for that. So therefore when I returned to MIT in the fall of 1958, it was clear to me that [we] needed a new programming language.

Nilsson: Well, there are two interesting things about that. One is the Lisp processing. What was it about Lisp processing that recommended itself for applications you wanted to pursue in artificial intelligence? What is it about Lisp that...

McCarthy: Well, you have data of variable size, and the key thing is that you can match the structure of the data by using lists. For example, a sum can be represented of plus followed by the summands, and so you immediately know that a list structure represents a sum without having to go parsing each time, and that has made for greater power.

Nilsson: And things that are a little more connected with artificial intelligence than arithmetic might be, I suppose, representing say a theorem in geometry could be done as a list, or representing the axioms?

McCarthy: Or more generally any logical sentence also has a structure. It's conjunction or disjunction, or a universal quantification.

Nilsson: And then recursion. Why did you think recursion was going to be important?

McCarthy: Well, if you take the rule given in a calculus text as to how to do formal differentiation, then you have the rule that the derivative sum is the sum of the derivatives, which means that if you want to do the derivative and the expression turns out to be a sum, then you have to find the summands, and call a differentiation program on the summands. So that's where it comes in.

Nilsson: It naturally suggested itself. And now, in connection with the development of the Lisp programming language, was it the case that you had earlier written a paper on computing with symbolic expressions? Did the paper come after Lisp, or did it precede Lisp?

McCarthy: Well, in the middle. Namely, we started on Lisp in the fall of 1958. I had to write a quarterly progress report for the Research Laboratory of Electronics that would come out as a report in January 1959, and so I wrote about Lisp in that, and I wanted to show that you could have a universal function, and it was much more transparent in structure than a universal Turing machine. So I wrote out this eval function just as a kind of a mathematical exercise, more to show mathematicians that this was good for this purpose. Mathematicians paid no attention.

Nilsson: That was the germ of the paper?

McCarthy: But one of the programmers, Steve Russell, did pay attention. He said, "Yes, if I program this eval in machine language, it will be an interpreter for Lisp."

Nilsson: There you have it.

McCarthy: I was a little skeptical of that, but he did it, and fine.

Nilsson: So that was the first-- was that the first instance of being able to run a Lisp program, then?

McCarthy: Well, what we did was we were going to have a compiler, but the compiler was clearly going to take quite some time. So what we did was we had a bunch of conventions, and hand compiled into a symbolic assembly program.

Nilsson: Well, here we are at MIT, and you had been at Dartmouth. What was the transition? You could have stayed at Dartmouth, I suppose. How did it happen that you decided to go down to MIT?

McCarthy: Well, John Kemeny, who was the head of the Math Department at Dartmouth, and who'd hired me, arranged for me to get a Sloan Foundation Fellowship, and I decided to take that Fellowship at MIT, and then I double-crossed him by not returning to Dartmouth, but staying at MIT.

Nilsson: And that's when you met Marvin Minsky?

McCarthy: No.

Nilsson: You had met him at the workshop, that's right.

McCarthy: No. Much earlier than that.

Nilsson: Oh.

McCarthy: Minsky and I were graduate students in math at Princeton at the same time.

Nilsson: Oh, I see. Now, he had done something on neural networks.

McCarthy: That's right. He built one at Harvard.

Nilsson: But then you had of course worked with him at that Dartmouth summer conference, and then you actually got involved with an AI project at MIT, once you ended up at MIT?

McCarthy: Well, we started an AI project at MIT. We met with-- we had a conversation and decided that we'd like an AI project, and who should be coming down the hall but Jerry Wiesner, who was Head of the

Research Laboratory of Electronics. We said, “We want an AI project.” He said, “What do you need?” and I said, “We need a room and a secretary, and two programmers, and a key punch.” And he said, “And how about six graduate students?”

Nilsson: He had some that he could arrange for you to meet?

McCarthy: He had committed himself to support six graduate students in mathematics on a big grant that-- a great big joint services contract, and he didn’t know what he was going to do with those six graduate students.

Nilsson: He could unload them on you.

McCarthy: That’s right.

Nilsson: And, wow, these six graduate students, they did then come and did they do some work on your AI project?

McCarthy: Yes, they did.

Nilsson: What sorts of programs did they write?

McCarthy: Well, a couple of them were working on a compiler for Lisp. I’m not exactly sure what the others did for us. They wrote their Ph.D. theses in pure mathematical logic, ignoring my ideas about proving programs correct. But then at least two of them later went into the field of proving programs correct.

Nilsson: You had mentioned one time to me in conversation about Chess endgames. Was that one of those graduate students?

McCarthy: Oh, yeah. Paul Abrahams wrote a program for three move mates, and used my proposal for the alpha beta heuristic.

Nilsson: The alpha beta heuristic, which lots of people claim they invented.

McCarthy: Well, Knuth wrote a big article about it, in which he discussed who did what, and the complication was that there was a weak form of it and a strong form of it, and he was never able to establish who exactly did what, but both Arthur Samuel, for his checker program, and Newell and Simon for their chess program, had some form of alpha beta in it. It may have been the strong form, and my original write up on it, well, maybe I wouldn’t even say write up-- propaganda about it, had a complication to it where I considered optimistic and pessimistic evaluations.

Nilsson: Now for people who are viewing this video, who might not know what the alpha beta heuristic is, is there a way to describe it in lay terms, what it is that it does?

McCarthy: Sure. The simplest way to describe it would be to use the blackboard, but I don't suppose you want to do that.

Nilsson: You'll have to speak visually.

McCarthy: All right. Here's the idea.

Nilsson: Or people could Google it I suppose.

McCarthy: I'm sure they could.

Nilsson: And look it up on Wikipedia.

McCarthy: Yeah. But the basic idea is this. Suppose I consider making a move, and then I consider that if I make this move, the opponent can capture my queen. Then I normally don't consider what else the opponent could do to me, because capturing the queen is a disaster sufficient that I shouldn't make the move that leads up to it.

Nilsson: Whereas a more naïve search process might look at all of these possible moves, right?

McCarthy: Well, the well defined minimax procedure would indeed do that, and the minimax procedure was built into the first chess programs.

Nilsson: So as I mentioned in the introduction, you had lots of other interests, of course, besides artificial intelligence, theory of computation being one. You've already said something about that, and the invention of Lisp. You also had these ideas about timesharing. Can you say something about how you decided to propose some ideas for that, for timesharing?

McCarthy: Well, for AI purposes, what one wants to be doing is to sitting at a terminal at the computer, and interacting with your intelligence program, so you can see what it does and make it better. So given the expense of computers, and also the fact that you spend most of your time sitting there thinking about what the computer has done, the obvious thing is timesharing, or at least it was obvious to me. So in 1957, when I came to MIT on this Fellowship, I proposed a simple form of it, but later it turned out that IBM, in connection with Boeing's desire to connect a wind tunnel to a computer, had invented or offered a system that had interrupts, which occurred to me that was really, was going to work, that was what was really needed to be able to do timesharing. So we begged that from IBM, and eventually got it, since MIT was in a position where it didn't pay for anything for that IBM 704. It begged, and it always took IBM a little while before they decided in fact to give it to us.

Nilsson: And so then, after they gave it to you, then somebody implemented-- you helped somebody implement an actual timesharing system?

McCarthy: Yeah. It was complicated. An assistant professor by the name of Herbert Teager was going to implement it, but his ideas got very elaborate, so the person who actually did implement it on the 704 was Fernando Corbató, and also I was a consultant at Bolt Beranek and Newman [BBN], and I talked to

Ed Fredkin who worked there, and I said, “Of course that PDP-1 computer is too small.” BBN was getting the very first PDP-1 computer, and he said, “No, it’s not.” And he devised what hardware mods would be required for the PDP-1, for timesharing, and persuaded his friend Ben Gurley who was the chief engineer at DEC, put them in, and then [Ed] Fredkin went off and I was left with the timesharing project at BBN. I got somebody to program it.

Nilsson: On the PDP-1?

McCarthy: On the PDP-1, and it was ready in September of 1962, which was the time I left MIT to go to Stanford now for the second time, and BBN didn’t use it because there were so few users of the PDP-1 that it was still convenient to let somebody sign up for exclusive use of the machine.

Nilsson: Now, with regard to timesharing, of course now computers are a lot less expensive, and people can afford to have individual ones, and so but still in all, even on these individual personal computers, there are versions of timesharing, right?

McCarthy: Well, what turns out-- when personal computers were first invented, then people said, “Oh, we can get rid of this timesharing stuff,” but then it then it turns out that, if you want the computer to be printing at the same time you’re doing something else, or doing any other kind of task, you want timesharing anyway, so all of the basic software for timesharing is in modern operating systems, whether they be [from] Microsoft or Apple, except that at least with regard to the windows, these features were sort of dragged in, and were not done very well.

Nilsson: Of course with Apple, based on Unix, they were already there.

McCarthy: Yeah, once Apple did base it on Unix.

Nilsson: Once they did it.

McCarthy: Yep.

Nilsson: And what about at MIT? So you were leaving MIT at about that time, but Corbató and others carried on with the timesharing, with was it Multics? Did that get started?

McCarthy: Well, what happened was that one of the people who was at BBN was J.C.R. Licklider, and I got him enthusiastic about timesharing, and then he went to ARPA and undertook to have a timesharing project at MIT that he wanted me to be head of, and he was a little bit surprised when he had discovered that I had disappeared.

Nilsson: You were back at Stanford.

McCarthy: Yeah, and so...

Nilsson: So he had to find somebody else.

McCarthy: Yep.

Nilsson: And was that Corbató?

John McCarthy: Well, MIT was rather hierarchical, so they put a professor of electrical engineering, Robert Fano, in charge, and Corbató under him, because Corbató was maybe just still an assistant professor, or something like that. But it was Corbató who did the design and the actual work. Fano presided.

Nilsson: Anything else to say about MIT days before we move on to Stanford? You leaving MIT, heading to Stanford.

McCarthy: Well, there was a big study that I ended up as-- I wasn't first chairman, but I ended up as chairman of it that produced a report that said MIT's next computer should be timeshared. At MIT, I had made a big mistake. What I should have done was underestimated the cost of it. Instead I was completely conservative, and overestimated the cost of what it would be.

Nilsson: And that scared them?

McCarthy: That scared them into stalling.

Nilsson: I see.

McCarthy: So they wanted a market study, and I thought that a market study on the desirability of timesharing was like conducting a market study among ditch diggers as to whether steam shovels were a good idea. So with Forsyth at Stanford called me up, and asked me to come to Stanford, I thought to turn him off by saying, "I'd been to Stanford before. I'd only come as a full professor," and he said, "I think I can arrange that," which surprised me very much, since MIT had just made me an associate professor.

Nilsson: So then we'll continue with Stanford days.

END OF TAPE 1

Nilsson: So before we go on and talk a little bit about Stanford, let's conclude some of the discussion you had about timesharing. You had mentioned some things earlier about it.

McCarthy: Well, I've written up my recollections about it, and they're on my web pages, and also my January 1959 memo proposing it, but it got into the air, so to speak, very quickly, because basically speaking, the general idea can be expressed in just a few sentences. So the Dartmouth people got it from me, of which they have acknowledged, and DEC got it from me, which they have acknowledged, and this book by Alan Newell and maybe Gordon Bell, I think. But still nobody's written it up properly as to, you know, talk to everybody who was involved and said what we did, and who did what.

Nilsson: So there's an invitation. Somebody ought to go out and look at that in a very scholarly way, and see who did what.

McCarthy: Should have, unless too many of the people who were involved are no longer functional, but...

Nilsson: Well, while we're on the subject of timesharing, apparently IBM, even though they provided the 704 that MIT used, they themselves didn't take to timesharing?

McCarthy: Yes and no, or maybe one should say no and yes. Namely, the person who was in charge of IBM's next big computer, the 360, who was Gene Amdahl, decided that he didn't need to put any hardware into the 360 that was dedicated to timesharing. Now, MIT and IBM got into a dispute, and there were two aspects to this dispute. The first one was over money, namely whether IBM should pay MIT some large sum concerning the invention of core memory, which had been invented by Jay Forrester at MIT, and that eventually got resolved after the heads of the two organizations resigned from each other's board of directors. The other was that after I left MIT, the committee that was in charge of timesharing came up with a very elaborate system involving pages and segments, which eventually led to Multics, which was-- and IBM would not make the modifications to the 360 that were required for this, and so Corbató negotiated with General Electric, which would. And then at the last minute, IBM said, or somebody said to IBM, or Watson said actually, "Give them anything they want," and so IBM came up with this 360 model 67, which did have all of that. But MIT had already decided to go with General Electric.

Nilsson: Who was making a computer at the time?

McCarthy: Yeah, General Electric was a competitor in the large computer game. But IBM did make this 360 model 67, and did make their own timesharing system with it, but it was complicated and slow, in my opinion, because of the ideas that they took from the Multics proposal. Multics was never a success either. It had rings of protection and so forth, and it had more elaborate protection than anyone has ever found it desirable to make sense.

Nilsson: Well, that's interesting that IBM was a little slow on that, and it was also the case, wasn't it, that even after Gelernter did his work on geometry theorem proving, and maybe there's some other work related to AI at IBM, IBM soon after lost interest in artificial intelligence.

McCarthy: It was more than losing interest. It was development of actual hostility. There were two aspects of it. One aspect was a sort of sales aspect. I forget when, but maybe already as early as 1955, IBM stopped using the word "computer," and talked about data processing machines, because these would be less frightening to their customer base. And then in 1959, somehow they decided that artificial intelligence was a bad idea, and discontinued it.

Nilsson: So no more artificial intelligence work in the IBM laboratories until much later.

McCarthy: Yeah.

Nilsson: Okay.

McCarthy: IBM artificial intelligence has had its ups and downs at IBM.

Nilsson: So now here we are at Stanford, and Forsyth was able to get a full professorship position for you at Stanford. This was in the Math department at the time, is that right?

McCarthy: Well, the intention was to have a computer science department as soon as there were enough faculty, so it was a computer science division of the Math department. And so I came in the fall of 1962. The computer science department was in fact established in February of 1965.

Nilsson: 1965. And so what were some of the very early things that you were doing at Stanford? You had a PDP-1 there, did you?

McCarthy: DEC gave me one. And then [Patrick] Suppes was interested in machine aided instruction and I convinced him that timesharing was the way to go, and he arranged for us to get a joint grant from NSF that allowed greatly beefing up that PDP-1.

Nilsson: Yeah, and then you put a timesharing system on it.

McCarthy: And then we developed a timesharing system for it.

Nilsson: And so, was there some early artificial intelligence work that you did on that system?

McCarthy: Well, yes. On a game playing, we did a game of Kala, and were able to establish that if they started with three stones in each pit, it was a win for the first player. We proved that rigorously. We also, the programs indicated that it was a win for the first player with four, five, and six, but we didn't have an actual proof of it. The PDP-1 was not powerful enough to do that.

Nilsson: Was there any work on chess at that time?

McCarthy: Not at Stanford.

Nilsson: Not at Stanford.

McCarthy: Or rather actually, I started on a chess program when I was at MIT, and I taught the first course at MIT in programming for freshmen, and three of the people who finished that course took up my start of a chess program, and wrote a chess program for the IBM computers in Fortran. And that was Alan Kotok's senior thesis, so he started out on at as a sophomore, and finished it.

Nilsson: Now, your interest in artificial intelligence of course, which got really going in pretty good shape along with the Dartmouth Conference, you had written a paper, which you gave in 1958 at Teddington.

McCarthy: Yeah.

Nilsson: Can you say a little bit about that?

McCarthy: That was [about] programs with common sense. That was the first set of ideas that I had in which I retained confidence that they were on the right track. As I told you, I didn't like the ideas of evolving automata that I discussed with von Neumann, and in the automata studies I had a paper inversion of functions defined by Turing machines, which basically had the same effect. It didn't really permit direct representation of common sense information.

Nilsson: So you wanted to be able to represent common sense information in some kind of language, which was declarative, and you had thought about the use of logic. So do you want to say a little bit more about those ideas? They were described in your 1958 paper, but have been ideas you've been pursuing ever since, actually, right?

McCarthy: Well, that's correct. And basically mathematical logic as invented by, well, Frege, I would say, in 1879. Of course the formal logic goes all the way back to Aristotle, but Frege's system was the first one that was sufficiently expressive, and that's good enough, in my opinion, for common sense. One needs some extension for non-monotonic reasoning, and I think one also reason-- needs an extension for context as objects, but it's good enough. The logicians, however, have not been interested in pursuing common sense.

Nilsson: Now, that particular school of artificial intelligence… some have called it the logicist's school, and there's been lots of controversy about it. I recently looked at an article again by Drew McDermott criticizing even though he started out being interested in logic, criticizing the approach, so what...

McCarthy: Is that a recent article, or his old article?

Nilsson: It was his old article.

McCarthy: M'hmm.

Nilsson: Anything to say about the pros and cons? I mean, some people say, "Well, look, humans aren't all that logical. A lot of what we do is not deduction, but something else." What would be your defense for logic against arguments of that sort?

McCarthy: Well, in the first place, that's right. A lot of the way people draw conclusions is not logical, but in the second place, people recognize the correctness of logical arguments, and quite apart from anything having to do with computers. If their arguments are incomplete, as far as logic is concerned, people can be persuaded of that, or even that their arguments are wrong. So after all, logic was invented in terms of communication among humans, not specifically for the use of computers. Now, people have had, from time to time, expectations of short term success, and logic is one of those areas that has induced expectations of short term success. Another one is neural nets, and connectionism, and so forth, and short term success has eluded people, and so it looks like human level artificial intelligence is quite likely to be a long term proposition. At least it requires new ideas, and with regard to new ideas, you don't know when somebody's going to come up with it, or not. Now, the current enthusiasm for short term success is promoted by Ray Kurzweil, and I looked at-- he wrote a book called *The Singularity is Near*, and I read his chapter on artificial intelligence, and as far as I'm concerned, he has not provided any sufficient basis for his short term optimism. That chapter is not technical. It merely-- what it mainly consists of is reference to various demos.

Nilsson: So he's not pushing any particular point of view about what's going to lead to short term success in artificial intelligence, right? He just says it's going to happen?

McCarthy: Yeah. And maybe it will, and maybe it won't, but if it does, it won't be due to him. <laughs>

Nilsson: He'll be there to help celebrate, he hopes, because he's also interested in extending his life span.

McCarthy: Yes.

Nilsson: Now, somewhere around the time that Stanford decided to have a computer science department separate from Mathematics, you helped start, or maybe did start the Stanford Artificial Intelligence Laboratory, under the acronym SAIL. Can you say anything a little bit about how that got started?

McCarthy: Well, when I came to Stanford, and I asked Licklider for some money to do work on artificial intelligence, and he gave me a fairly moderate grant, then I began to think about what was required for vision, computer vision, and the dominant idea at that time, which is still rather prominent, is making computer programs to do discrimination. For example, discrimination between the pictures of the letters of the alphabet.

Nilsson: Sometimes called pattern recognition, or something like that.

McCarthy: The pattern recognition slogan is for something a little more systematic than that, I think, but the first things were using various ad hoc features. That is, does it have a hole in it? Well, that provides the beginning of a classification. Does it have something sticking down in it?

Nilsson: Selfridge was doing things like that, right?

McCarthy: I don't know whether he was, or not. I wasn't thinking about him. But anyway, my idea was that you wanted description, not just discrimination, and my argument for that was that if you wanted to program a robot to pick up something, then it didn't just have to discriminate on the whole picture. It had to be able to locate the object, and represent its shape, so I decided to try for robotics, and Licklider was gone by then, but I did apply and got a rather large grant, or contract, with DARPA to start the Stanford Artificial Intelligence Laboratory. In particular, the amount of money required for our very own PDP-6 computer that later got beefed up into a PDP-10, and then eventually to a KL-10, and so forth.

Nilsson: Did you get some students interested in doing this description work of scenes?

McCarthy: Yeah.

Nilsson: And also you had some faculty who were additional faculty involved with artificial intelligence work there?

McCarthy: Yeah, Jerry Feldman, for example, who came as an assistant professor. Yeah.

Nilsson: And then Les Earnest, who wasn't a faculty member, but came to help run the lab.

McCarthy: Yeah, and when I started running the lab, after a while I told Ivan Sutherland that I wanted to hire an executive officer, and he said, "Yes, you should. You're the only one of our contractors who has a perfect record. You have never submitted a quarterly progress report."

Nilsson: At all, or on time?

McCarthy: At all. So he recommended Les.

Nilsson: Because he had known Les from Lincoln?

McCarthy: Yeah. And Les took not only charge of that, but also was the substantial initiator of a number of the projects involving the timeshared applications of the computer, including PUB [ph?], which was the first really document producing thing.

Nilsson: There were a lot of other things besides artificial intelligence that might be regarded, I guess, as scaffolding for all these projects, but probably one was some other work also, right? Graphics?

McCarthy: Yeah, there was at least one thesis on graphics.

Nilsson: And displays, different kinds of displays.

McCarthy: Yeah, m'hmm. And control; that is, controlling the [Stanford] Cart, and the permanent-- another assistant professor who had got his Ph.D. at Stanford, with whom he had as his advisor Raj Reddy, and eventually he disappeared as many of our assistant professors did, when we were ready to renew their appointment as assistant professors. Other schools were willing to make them a higher position.

Nilsson: In other words, other schools wanted to get started in this field, or expand in that field. Raj Reddy went to CMU.

McCarthy: That's right.

Nilsson: And eventually became a Dean there. All right, but there was also music, work on the use of computers in music, too, right?

McCarthy: Yeah, well, that was a sort of hanger on, namely, I let them use the computer, but no ARPA money was ever spent on it.

Nilsson: Having a good computer around attracts a lot of people who have ideas that they'd like to explore on a good-sized computer.

McCarthy: Well, that's where I-- I mean, for a while, besides the IBM installation, and some PDP-8s in various specific laboratories, we were the only general purpose computer on campus.

Nilsson: And Knuth used it also, didn't he, for his typesetting work?

McCarthy: That's right.

Nilsson: M'hmm. Well, now, besides the robotics work that was going on, and all of these other things that were using the computers that you had, what sort of work were you doing, or what sort of thoughts did you have at that time that extended your use of logic as a representation language, and moving toward representing the information needed for human level artificial intelligence?

McCarthy: Well, I did improve my representations of common sense, and wrote some papers, but basically the problem of representing common sense knowledge, either in logic or some other way, is still unsolved, or at least incompletely solved, or partially solved.

Nilsson: What about some early attempts who tried, early attempts at the idea of implementing the ideas in your 1958 and other memos on the advice taker? There were some students who tried to implement some of those ideas?

McCarthy: Well, knowledge representation is a whole industry these days.

Nilsson: M'hmm. Well, I guess I was getting at people like...

McCarthy: And it has been for a long time, since the 1970s.

Nilsson: I guess I was getting at people like Black and Cordell Green, and their attempts to actually use the situation calculus that you proposed.

McCarthy: That's right. Well, that was 1969, Cordell Green.

Nilsson: But these were early attempts, but not completely-- they certainly were not, didn't lead to the success that Kurzweil would have liked.

McCarthy: That's true. And every now and then, somebody comes along and says, "Well, you see, this hasn't got us to human level intelligence, so my idea must be better," and then there's a lot of propaganda, it gets some money, and after a while their idea doesn't lead to human level intelligence right away, either.

Nilsson: Well, these various ideas that people have had...

McCarthy: Connectionism is the example that I would use.

Nilsson: Subsumption. <inaudible> networks.

McCarthy: Yes.

Nilsson: Well, nevertheless, do you think that the combination of these ideas might in fact be useful?

McCarthy: I don't know. I mean, I think logic is fundamental.

Nilsson: Now, in terms of extending logic, you're also well known for your work on circumscription, just one way of achieving a non-monotonic logic. Do you want to say some things about that? Non-monotonic logic in general, and circumscription in particular?

McCarthy: Well, first of all, one can observe that people do non-monotonic reasoning. That is, draw conclusions from partial facts that they might not draw, or would not draw, if they had certain additional facts that do not contradict the partial facts. And so the fact is, you have to draw conclusions from partial information. And some people have jumped to the conclusion, oh, maybe a 100 years ago, that that was probabilistic, but it turns out that that's not the only way in which humans come to conclusions. So circumscription is a formalization in mathematical logic of this property, and I wrote two papers on it.

Nilsson: So it essentially involves something like if-- well, there are various, more simpler versions of non-monotonic reasoning than circumscription. So for example, the default logics, and they're related to circumscription.

McCarthy: They overlap, but they're not simpler. The simple one is predicate completion, where you have some predicate, and you say, "So and so implies this predicate," and you say, "So and so, if and only if the predicate." That's predicate completion. And then there's negation as failure. In Prolog, that's simple in Prolog, and that also existed in Microplanner. Both of those date from the early '70s.

Nilsson: Isn't there a use also that the database people make, in which if some particular fact is not in the database, you can assume that it's not true?

McCarthy: Well, that's sort of commonly done. I don't know to what extent. It's qualified. In other words, if you have a database of Stanford students, and you look up a name and it's not in the database, then you conclude that this person is not a Stanford student, so you assume that the database is complete, which you might not always want to do. But, now, as for circumscription, it's based on the idea of minimizing the extent of predicate, allowing certain things to be varied in order to achieve this minimum, and so in that respect, it's very analogous to minimization in mathematics generally.

Nilsson: What are some other things? Did you have other ideas that you suggested that would be important to achieve human level artificial intelligence? Non-monotonic reasoning is one, but you also investigated some other elaborations of logics also.

McCarthy: Some of them are elaborations of logic, or at least one major one is an elaboration of logic. Many of them are things formalized within logic, it doesn't require any extension of the logic. For example, concepts as objects.

Nilsson: You reify the concept.

McCarthy: Yeah. And you can do that within logic. I advocate such reification as a competitor to modal logic. Now, the thing that, as far as I can see, does require an extension is contexts as objects.

Nilsson: So, if you were just sort of to summarize, what do you think a new graduate student working in artificial intelligence might want to work on, in addition to all of the different things that come up, like connectionism, Bayes networks, other neural network techniques, subsumption; what are some of the core-- what do you think a person ought to set about that he could actually, or she, make progress on in, say, a dissertation?

McCarthy: Well, representing particular domains of common sense knowledge is, in my view, the thing that will really give us the clues as to where we need to go towards human level artificial intelligence. That being more important than for the moment, than coming up with new general theories. Now, there's some nice problems like a person who has come up with a bunch of nice problems is Ernie Davis at NYU, and one can mention one of his, namely the Birthday Present Problem, so A and B are thinking about a birthday present for C, and C must not know what the present is in advance, and therefore A and B should not talk about it in the presence of C, and so forth, and to formalize all the necessary facts for that, is something that I'm not sure whether there are any published attempts at that.

Nilsson: So formalizing a particular more or less circumscribed problems, in which common sense is needed, would be a good idea.

McCarthy: Yeah, but consider the general notion of secrecy. One would presume that if you wanted to do this Birthday Present Problem, then you ought to come up with some general notion of keeping something secret, by for example, not leaving the birthday present lying around, not talking about it, and so forth.

Nilsson: How many of these particular divisions do you think we would have to work on before making some progress toward general common sense? Are we talking about six of them, or 100 of them, or...

McCarthy: 20. Needless to say, if you really had even ten, somebody looking at the ten might get some way of saying, "Well, there really aren't ten, there are only three." And putting it together.

Nilsson: Now, if you look at Cyc as one example, trying to make, represent common sense knowledge, do you think Cyc concludes a lot of these?

McCarthy: Well, it includes a lot. I've asked people to do various things for me in Cyc, and it's never really worked out. I can describe one example, and that was arranging for a dinner, and Cyc was able to say, well, you need a table and you need silverware, but it didn't come up with you only need one table, but you need silverware for each diner. This was quite some time ago, so maybe Cyc is better in that respect, but somehow it's representation of requirements for some action. At least how it's used has not been adequate. He's holding us up.

Nilsson: Five minutes.

McCarthy: Oh, that's <inaudible>.

Nilsson: Now, if you were to pursue that kind of work with Cyc, trying to get it to reason about various things, and made notes of its inadequacies, would that, do you think, be a good way to proceed? Because for every one of these inadequacies, I suppose you could add something to Cyc.

McCarthy: Yes, that's a good idea, provided you don't mean John McCarthy. <laughs>

Nilsson: <laughs> Well, would it be a good idea for students to do something like that, do you think?

McCarthy: The trouble with any specific project is who knows that the student might end up spending all his time on the politics of Cyc, or the sysadmin of Cyc, so the student would have to be or his advisor would have to be capable of getting through those two things at least before the student would be really conceptualizing it.

Nilsson: Do you think that the effort that went into Cyc is worthwhile? That that's the sort of thing that makes progress toward human level AI?

McCarthy: Yeah, I think so.

Nilsson: And if so, should there be more of them that would have perhaps somewhat different points of view about how to proceed and therefore might be...

McCarthy: Yeah. Certainly. A quite different point of view about how to proceed, except that I'm not sure what this different point of view is. But, you know, Lenat made a real good try. I think more recently, in order to get support, he's undertaken to do whatever DARPA wants, and probably what DARPA wants is associated with various current slogans, which may not be the best path towards human level AI.

Nilsson: One thing we haven't touched on, and maybe we could do this sort of to conclude, is your views on progress in learning. It looks like machine learning is an important topic in AI these days, and lots of people are involved in using it for one thing or another, and control, and so on. How about learning? What do you have to say about learning?

John McCarthy: Well, I don't know much about the techniques that have been used, but I have looked at what it learns, and what it learns are the main discriminations, and these are very limited. For example, no learning program would, from experience, learn about secrecy. It might, if you formalize things sufficiently, know some linear function of some things, which give you the probability that some particular thing is secret, so my opinion is that--

END OF TAPE 2

Nilsson: So continuing what you were saying about learning and discrimination.

McCarthy: Well, the stuff that I've read about learning has been mainly discrimination and not learning how to describe things. An example that I'd want to give is learning new concepts like secrecy. But also, I have an example on my web page about learning the three dimensional reality behind a two-dimensional appearance. I have a little puzzle up there on my web page called appearance.html that some people have played with. People who have done something serious with it are Stephen Muggleton [ph] and Ramon Otero [ph] using inductive logic programming but they haven't fully solved it yet.

Nilsson: Inductive logic programming is a variety of learning isn't it? Where one does more than just discrimination?

McCarthy: That's correct.

Nilsson: There's a lot of possibility of learning statements there if one has what they call a bias.

McCarthy: Yeah, because I believe with inductive logic programming you can learn something that has variables in it. At least three variables that are implicitly universally quantified. I don't know whether it goes up to learning the full predicate calculus sentences that can have alternations of quantifiers.

Nilsson: Alright. Well, that's interesting. Are there any further things that you want to say in conclusion of the interview to give cheer to people that are doing work in artificial intelligence? If the goal is not around the corner, are we making progress?

McCarthy: There's something I would want, if you can do it, to stuff it in the middle somewhere. That is, when you were asking about the virtues of logic, the things that the logicians have proved for example, Gödel's Completeness Theorem in which he showed that anything that is true in all models of a given collection of sentences, is provable. That, from the point of view of AI, that's important. Probably more important than his subsequent incompleteness theorem, because nobody is expecting to make programs that can decide arbitrary mathematical sentences.

Nilsson: So although its important to be able to show that something true in all models is provable, the detractors of the use of logic would say "Well, that may well be, but it might not be provable in practice."

McCarthy: Well that's so, but in limited domains, you can deal directly with models and in fact the satisfaction programs do that for a limited part of logic. Mainly, propositional logic. I guess the point I wanted to make there is that for AI, logic is inevitable in the same sense that for physical systems calculus is inevitable. Stuff that in there somewhere.

Nilsson: We'll do that. So in conclusion, I think its worth playing out that McCarthy's websites have an awful lot of information, pointers to papers, pointers to things like the original Dartmouth Proposal, ideas that he's had on topics that might not have found their way in the formal papers yet plus a lot of information about some of John's opinions on matters outside of artificial intelligence. So, highly recommend that you visit. You can Google John McCarthy.

McCarthy: Yeah, problem is that the general systematic way of handling the URLs of faculty members came along after my particular URL was established. Googling John McCarthy will get you there.

Nilsson: Ok well thank you very much.

END OF TAPE 3

