

Interview

An Interview with Dana Scott

*ACM Fellow and A.M. Turing Award recipient
Dana Scott reflects on his career.*

ACM FELLOW DANA STEWART SCOTT, the recipient jointly with Michael Rabin of the 1976 A.M. Turing Award for the concept of nondeterministic finite automata, has made seminal contributions spanning computing science, mathematics, philosophy, automata theory, modal logic, model theory, set theory, and the theory of programming languages. After receiving a B.A. in mathematics from the University of California, Berkeley, in 1950, and a Ph.D. from Princeton University in 1958, he held faculty positions at the University of Chicago, UC Berkeley, and at Stanford, Princeton, Oxford, and Carnegie Mellon Universities. He retired as University Professor from CMU in 2003.

The distinguished theoretical computer scientist Gordon Plotkin conducted a series of four oral history interviews of Scott between November 2020 and February 2021. The interviews, the transcripts and videos of which are online,^a cover primarily the period leading up to the 1976 ACM A.M. Turing Award. Presented here is a condensed and highly edited version, which includes some additional post-interview material provided by Scott.

—Len Shustek

What was your early background?

I was born in 1932 in Berkeley, CA,



where I am now in retirement. We lived on a farm near Susanville when I started first grade in a one-room schoolhouse. We later moved to Stockton, where I still remember hearing on December 7, 1941, “Extra, extra, read all about it: Pearl Harbor bombed.”

As a student I learned to play the clarinet and had some piano lessons. I became interested in how instruments made their sounds, so the band teacher gave me a book, *Science of Musical Sounds*, which I found fascinating. It had lots of math, which inspired me to learn some calculus on my own from a book my uncle had. When we moved to Sacramento, I discovered Helmholtz’s book on acoustics and music theory

My high school math teachers in Sacramento were very inspiring, so I was convinced that when I got to college I would major in mathematics.

in the dusty California State Library, and that led to my study of logarithms and Fourier series. I did a small project on acoustics for the Westinghouse Talent Search when I was a senior, and got an honorable mention. Music has been very important to me all my life. And lucky for me my wife (a pianist), our daughter (a cellist), and our granddaughter (a violinist) are professional-grade classical musicians.

My high school math teachers in Sacramento were very inspiring, so I was convinced that when I got the opportunity to go to college I would major in mathematics. Neither my father nor my mother had gone to college, and I was lucky to get a small scholarship to

^a See <https://bit.ly/3xZncVg>

be able to go to the University of California, Berkeley. I think I was only the second in our immediate family to get a college degree.

How did you get involved with mathematical logic at Berkeley?

As a freshman I signed up for an introductory logic course taught by the very Germanic and cigarette-ash covered Paul Marhenke, chairman of the philosophy department. It was an easy course, and I liked logic very much. I signed up for more courses in my sophomore year with Professor Benson Mates of the philosophy department, whom I became very close to. He introduced me to the work of Alfred Tarski, a leader in logic worldwide who had escaped Jewish persecution in Poland and had become a professor at Berkeley in Mathematics.

In my first two years, in order to help support myself I worked in the UC library in the periodicals room. There I discovered the *Journal of Symbolic Logic* and could not understand a thing in any of the articles—except for one on truth-tables by a Jan Kalicki, who had escaped Poland at the end of the war before the communists took over. He had been able to come to Berkeley in 1951 at Tarski's invitation. When I signed up for the Theory of Equations course from him, he was amazed and pleased I had read his paper. We became very good friends, and wrote some joint papers on equational theories, which Tarski had begun to investigate. Our theories were “complete” in the sense that they cannot be extended further without collapsing so that all equations become derivable. Tarski was very interested in what we were able to do.

Very tragically, the next year Kalicki was killed in an automobile accident. That was a very great blow to me because he was a very warm friend and a mentor. By this time I had been introduced to the Tarski circle, and I went on in my junior year to take part in Tarski's teaching both in courses and in seminars. He was an absolutely amazing lecturer, and a great motivator. As many mathematicians can, he could lecture completely without any notes.

Raphael and Julia Robinson were close friends of Tarski. Julia did her thesis under Tarski, but you cannot imagine the difference for women between

I did some programming on the computer von Neumann built at the Institute for Advanced Study. The university soon discovered it is very expensive to keep a machine like that running.

1950 and today. Only men were allowed in the Berkeley Faculty Club, so when Tarski suggested at a lunch there were many problems to be solved about rational arithmetic, Julia was not present. Raphael relayed the suggestion to her later, and that is how her famous thesis eventually evolved in which she showed the undecidability of the theory of the field of rational numbers.

I was then learning about formal theories, and I found the little book³ on mathematical logic by Paul Rosenbloom. One of the chapters was on Haskell Curry's combinators and Alonzo Church's lambda calculus. After spending a lot of time trying to puzzle out how combinators combine and what they do to reproduce one another through their equations, I spent a night having nightmares about these combinators! I dreamed there were gigantic combinators coming to bite me—or to do something terrible. I don't know whether that nightmare about combinators was what cemented my interest in lambda calculus, but that is exactly how it started.

How did you get to Princeton as a graduate student?

In 1954, I had become Tarski's graduate student at Berkeley. He had earlier hired me as an assistant to correct page proofs of translations of his early writings from German into English. It was a horribly boring job, and I got very lazy about it. Tarski understandably became

very angry and fired me, which caused a break between us. But one of my other professors who heard about my difficulties said, “Why don't you think of going somewhere else? Norman Steenrod is visiting from Princeton—go and interview with him.” I did, and, through his kind recommendation, I was able to go to Princeton the subsequent year.

I was of course very interested to meet and to be directed by Alonzo Church. But his much earlier logistic system based on untyped lambda calculus, which he thought was going to solve problems in Frege's system leading to the Russell paradox, had been shown to be inconsistent. That disappointment, I think, was the reason he never lectured about it, and never discussed it with any of his students while I was at Princeton. It is really too bad that I never had a chance to discuss modeling untyped lambda calculus with him after my discovery in 1969.

By the way, Alan Turing had been a Ph.D. student of Church before WWII, and he insisted that Turing formulate all of his work on transfinite computation in terms of lambda calculus. In one of his letters Turing said how much he hated to do that, but it was what he had to do to finish his Ph.D. I do not think they ever had a close personal relationship, and Church never spoke about Turing while I was a graduate student.

Church did not influence my choice of a thesis topic. His kind of direction was to discuss with his students what area of research they wanted to do, and then just let them do it. As I very unkindly like to say, Church mainly corrected the spelling in my thesis. The problem had come from Tarski much earlier.

In the summer of 1958 at Princeton, after my Ph.D., I did some programming on the computer John von Neumann had built at the Institute for Advanced Study. Von Neumann was already ill in the hospital when I got to Princeton, so I never had a chance to meet him. After he died, Princeton got the computer because the IAS never wanted to have anything to do with engineering. Hale Trotter and I were hired by Professor Forman Acton to do a project on the computer, and we chose solving the Pentominoes jigsaw-like geometric puzzle. Inspired by the backtracking algorithms that had been developed

at Berkeley by Dick and Emma Lehmer, we were able to completely do one version of the puzzle.

The university soon discovered it is very expensive to keep a machine like that running. The electrostatic tubes were much affected by humidity—Princeton can be very humid—so the best time to work on the von Neumann machine was 3 A.M. I think by the fall they had had enough and closed down the computer.

As a graduate student I became very closely associated with Georg Kreisel, a visitor at the IAS who had been invited by Kurt Gödel. I found both of them to be very hypochondriac. They loved to talk for hours in German on the phone, because that was an excellent way of not spreading germs! Through Kreisel I learned a lot about Gödel's thinking, and though I met him, I did not become close to Gödel until I was a faculty member at Princeton in the early 1970s.

Princeton was very exciting place to be because so many mathematicians were there or came as visitors. Of course the faculty was very, very strong, but looking back I wish I had taken more advantage of what I could have learned at Princeton.

How did you start working with Michael Rabin?

I became very good friends with my fellow graduate student under Church, Michael Rabin. In 1957, the two of us were chosen for a summer internship at the IBM Yorktown Heights Research Center, which was housed in a rented estate before they built the very fancy building they have today.

We didn't know what to do. We decided to think about automata from the point of view of model theory and structures. There were several recent papers by people we knew on automata, referenced in our later *IBM Journal* paper, that had sparked our interest in the subject. I don't remember how we thought of doing nondeterministic automata, except maybe that we kept coming into problems where it was difficult to create the states to control various kinds of decisions.

To be clear, a nondeterministic automaton is not a probabilistic automaton. When it is making a transition from one state to the next, it instead has many choices that it can make

rather than a specific one. Success in accepting the tape it is reading means that there is some path through the choices that eventually results in success. You don't have to worry about the paths that don't work out, since you only have to find one successful one.

For the proof that a nondeterministic automaton accepts the same set of strings that a deterministic automaton does, you take the power-set of all the states and treat those as new states, and define a transition function on sets of states. Of course, the number of states goes up exponentially. Nondeterministic automata are sometimes better to work with because they require so many fewer states.

At the end of the summer we both went to a logic conference at Cornell University, where practically anybody who was anybody in logic was present. Rabin and I reported on our work^b on automata, and then we prepared a paper,⁴ which was submitted the next academic year. Many people agreed that the proofs were nice and clean, but I don't recall any special enthusiasm about the approach at that time.

Why did you go to Chicago, and then to Berkeley?

In 1957–1958, my last academic year at Princeton, I met Paul Halmos, who was visiting from the University of Chicago. We got to know each other through his interest in algebraic logic, and it was his influence that got me an invitation to come to Chicago as a non-tenured instructor.

^b For which they subsequently received the Turing Award in 1976.

On our summer job, Michael Rabin and I decided to think about automata from the point of view of model theory and structures.

Early on at Chicago I met Stanley Tennenbaum. The University had the terrible idea that bright high school students should come to college in their junior year, but young people are not really ready to come into an adult environment like college. The result was that Stanley took part in many things and met many people, but never finished his undergraduate degree.

Stanley was a big influence on me, and we did a lot of work together. He came up with a tidy proof for Emil Post's problem in recursive function theory by realizing that there are no computable non-standard models satisfying the laws of first-order arithmetic; that is still known today as "Tennenbaum's Theorem." But much of our work together was never published, because our notes were lost when Stanley's car was broken into and a box with all of our papers was stolen. I am sure it was dumped in 10 minutes when the thieves saw what it was, and so it never saw the light of day.

By the end of my two-year appointment at Chicago I had had a full rapprochement with Tarski, so in the summer of 1960 I returned to Berkeley with a fellowship to the newly established Miller Institute. There was enormous activity there on ultraproducts^c and their connection to large cardinals.^d In a paper published in 1961,⁹ I proved that measurable cardinals contradicted Gödel's statement that all sets are constructible. It turned out coincidentally that a very brilliant young logician in Prague, Petr Vopěnka, found the same proof at the very same time.

Rabin came to Berkeley for a one-year sabbatical, and we had a very good time working together again. He discovered a new proof of Trakhtenbrot's Theorem that the set of first-order sentences true in finite structures is not axiomatizable, not enumerable. We had some other results together, but, alas, that was my last working with him because he went to Jerusalem and then to Harvard.

The Belgian numerical analyst René De Vogelaere was there and was a very enthusiastic proselytizer for the use of ALGOL for programming. There was quite a lot of activity—seminars and things like that—trying to understand

^c See <https://bit.ly/3HzaZLh>

^d See <https://bit.ly/3ba0xOb>



Association for Computing Machinery
Advancing Computing as a Science & Profession

ACM Student Research Competition

Attention:
**Undergraduate and Graduate
Computing Students**

The ACM Student Research Competition (SRC) offers a unique forum for undergraduate and graduate students to present their original research before a panel of judges and attendees at well-known ACM-sponsored and co-sponsored conferences. The SRC is an internationally recognized venue enabling students to earn many tangible and intangible rewards from participating:

- **Awards:** cash prizes, medals, and ACM student memberships
- **Prestige:** Grand Finalists receive a monetary award and a Grand Finalist certificate that can be framed and displayed
- **Visibility:** meet with researchers in their field of interest and make important connections
- **Experience:** sharpen communication, visual, organizational, and presentation skills

Learn more:

<https://src.acm.org>

what might be opening up in the design of computer languages.

There were many other logic visitors coming and going to Berkeley at that time. John Myhill from Stanford and I wrote a paper about “Ordinal Definability,” showing that hereditarily ordinal-definable sets form a model of set theory that satisfies the axiom of choice. It’s not surprising that Gödel said, “Oh, I thought of that years ago. But my proof with the constructible sets was so much more important, I never told anyone about it.” Oh well, that’s what happens.

Why did you go to Stanford?

In 1963, John Myhill decided to leave Stanford, which created an opening there. At the same time there was some unhappiness at Berkeley because with all the new appointments the mathematics department was branching out, and there was a certain amount of animosity toward Tarski because he had pushed the development of logic so much. I decided I wanted to get away from that.

I had often worked summers at Stanford with Patrick Suppes, whose course I had taken while at Berkeley as an undergraduate in the 1950s and who became a close friend and mentor. He was always interested in how logic could be applied in mathematical psychology, and in 1958 we wrote a joint paper¹¹ relating measurement theory to probability theory. My contributions were on the side of model theory, not on the side of applications to psychology.

Solomon Feferman, one of Tarski’s former Ph.D. students whom I had kept in close contact with, joined the Stanford faculty. Then in the early 1960s, Georg Kreisel began regular vis-

its to Stanford, and it was actually he who convinced me to make the move from Berkeley.

At Stanford mathematicians like George Forsythe were beginning to feel that computer science should have some independence. And the mathematics department, which was heavily oriented toward classical applied mathematics, was happy to give away numerical analysis to help form the new department. Don Knuth moved over from Caltech. John McCarthy came from MIT and started up his AI lab. I was not directly involved in the computer science department, but I knew everybody who was there.

Paul Cohen, who had come to Stanford from the University of Chicago, invented “forcing”^e to give functions some properties just on the basis of a small amount of information. He saw he could extend that not just to functions over the integers but to the transfinite realm as well, and he used it to prove the Continuum Hypothesis^f is independent of set theory axioms.

Everyone was completely amazed by what he was doing. Robert Solovay at Berkeley visited Stanford often to hear about Cohen’s ideas, and in conversations with the logic group observed that these forcing notions are giving Boolean values to statements about set theory. Over the 1966 New Year’s break I thought to myself: “Wait. Why don’t I start with a suitable Boolean algebra in the first place, and then use that to interpret set theory and prove the independence of the continuum hypothesis in a different way?” There were a lot of details to work out, but it eventually led to my paper⁶ that won the 1972 Leroy P. Steele prize.

Work on Programming Language Theory

During a 1968–1969 sabbatical from Stanford at the University of Amsterdam I got to know Aad van Wijngaarden, who was very big in the development of ALGOL. Apparently, there were a lot of battles among people who had many different ideas about how ALGOL should develop, but van Wijngaarden had the final say on the definition of ALGOL 68.

^e See <https://bit.ly/39ubN7K>

^f See <https://bit.ly/3N92fwI>

**At Berkeley, the
numerical analyst
René De Vogelaere
was an enthusiastic
proselytizer for
ALGOL.**

That summer I attended the IFIP Working Group 2.2 on the formal description of programming. After hearing the many questions about language design, and after meeting Christopher Strachey and seeing his approach, I felt very attracted to thinking about computer languages.²

While at Amsterdam I had decided to move back to Princeton, but because of these new interests I made a special plea to the philosophy department to have my first semester in the fall of 1969 be taken on leave so I could visit Oxford and work with Strachey.

I had long experience with recursive function theory, and at Stanford I had lectured on automata theory. Many people had written about operators on spaces of recursive functions. When I found out that Strachey was depending so much in a very formal way on Church's type-free lambda calculus, I told him: "It would be much better if you thought about typed operators." And that's how I came to write the paper on Logic of Computable Functions (LCF)³ in order to try to convince Strachey it would be better to use an approach having a simple mathematical foundation that could be expanded.

He was very pleased, and he immediately adopted thinking in that way. He had lots of experience utilizing lambda calculus in discussing properties of programs, but this provided a model based on principles of recursive function theory that were well understood, so he didn't have to think of it as an abstract formalistic trick. But then in the late fall of 1969 I discovered how to use the LCF ideas to model the type-free theory.

How did you get back to Princeton?

A difficulty at Stanford was that the mathematics department was very oriented toward classical analysis and was not really interested in the development of logic. I was split between mathematics and philosophy, and I felt that logic was not especially welcome in mathematics. Donald Davidson, the philosopher and long-term professor who left in 1968 for Princeton, came by

The contributions I made over my career were very much motivated by my teaching, and my great luck was having really excellent students.

for a visit in Amsterdam and recruited me to come to Princeton.

Strachey came to visit me at Princeton in the early 1970s, and we finished our paper¹⁰ on "mathematical semantics," as we called the area then. Later it seemed better to say "denotational semantics" to distinguish it more clearly from the axiomatic semantics that Tony Hoare was promoting, and from the operational semantics that Gordon Plotkin was promoting. I would say today that axiomatic, denotational, operational semantics all meld together, and the question is to take which aspects you want for an analysis or a proof, or for giving the foundations for some kind of implementation. You choose what is appropriate for the thing you want to accomplish.

While at Princeton, Gödel, who was ill, asked me for help in preserving his unpublished notes. One of them was his ontological proof of the existence of God based on modal logic, which I very unfortunately spoke about at a Princeton seminar without his permission. I do not accept the conclusion myself; my feeling is that if you assume what you want to prove, you will eventually prove it. However, owing to my mistake the idea got out and the topic has become very popular recently.¹


Why was there no further collaboration with Christopher Strachey?

In the spring of 1972, I was startled to receive a letter from the Vice Chancellor of Oxford University inviting me to be the first holder of a new chair in Mathematical Logic. It was a very difficult decision to decide to relocate my

family again so soon after moving to Princeton, but the work with Strachey and his group was so potentially productive I decided to accept.

Alas, the writing of an essay for the 1974 Adams Prize^h proved to be a very arduous task³ and Strachey fell ill in early 1975. His untimely death that May was the very sad end of so many chapters for the large number of friends, students, and associates who had been so strongly influenced by him and his ideas.⁸

Do you have a summing up?

The contributions I made over my career were very much motivated by my teaching, and my great luck was having really excellent students,⁷ many of whom became very close personal family friends. It was the inspiration of these students that motivated me most. I have to say I miss that contact very much now in retirement, because once you retire you become sort of a ghost. 

^h See https://en.wikipedia.org/wiki/Adams_Prize

References

1. Benz Müller, C. and Paleo, B.W. Automating Gödel's ontological proof of God's existence with higher-order automated theorem provers. In *ECAI 2014, IOS Press, Frontiers in Artificial Intelligence and Applications 263* (2014), 93–98; <https://bit.ly/3NfriH>
2. de Bakker, W. and Scott, D. A theory of programs: An outline of joint work. IBM seminar Vienna (aug. 1969). In J.W. de Bakker, 25 jaar semantiek, J.W. Klop, J.J.C. Meijer, and J.J.M.M. Rutten, Eds. CWI, 1989, 1.
3. Campbell-Kelly, M. Christopher Strachey, 1916–1975: A biographical note. In *Annals of the History of Computing* 7, 1 (Jan.–Mar. 1985), 19–42.
4. Rabin, M. and Scott, D. Finite automata and their decision problems. *IBM Journal of Research and Development* 3, 2 (Apr. 1959), 114–125.
5. Rosenbloom, P. *The Elements of Mathematical Logic*. Dover Publications, 1950.
6. Scott, D.S. A proof of the independence of the continuum hypothesis. *Mathematical Systems Theory* 1 (1967), 89–111.
7. Scott, D. Mathematics Genealogy Project. <https://bit.ly/3HxZF1W>
8. Scott, D. Some reflections on Strachey and his work. *Higher-Order and Symbolic Computation* 13 (2000), 103–114.
9. Scott, D. Measurable cardinals and constructible sets. *Bulletin de l'Académie Polonaise des Sciences, Série des sciences mathématiques, astronomiques et physiques* 9 (1961), 521–524.
10. Scott, D. and Strachey, C. *Toward a mathematical semantics for computer languages*. In *Proceedings of the Symposium on Computers and Automata*. Microwave Research Institute Symposia Series, vol. 21. Polytechnic Press, Brooklyn, NY, USA, 1971, 19–46.
11. Scott, D. and Suppes, P. Foundational aspects of theories of measurement. *The Journal of Symbolic Logic* 23, 2 (1958), 113–128; <https://bit.ly/3xXcTrS>

Edited by **Len Shustek** (len@shustek.com) Chair Emeritus, Computer History Museum, Mountain View, CA, USA.

Copyright held by author.

^g A 1969 memorandum circulated but unpublished until a 1993 annotated version: D.S. Scott, "A type-theoretical alternative to ISWIM, CUCH, OWHY," *Theoretical Computer Science* 121 (1993), 411–440.