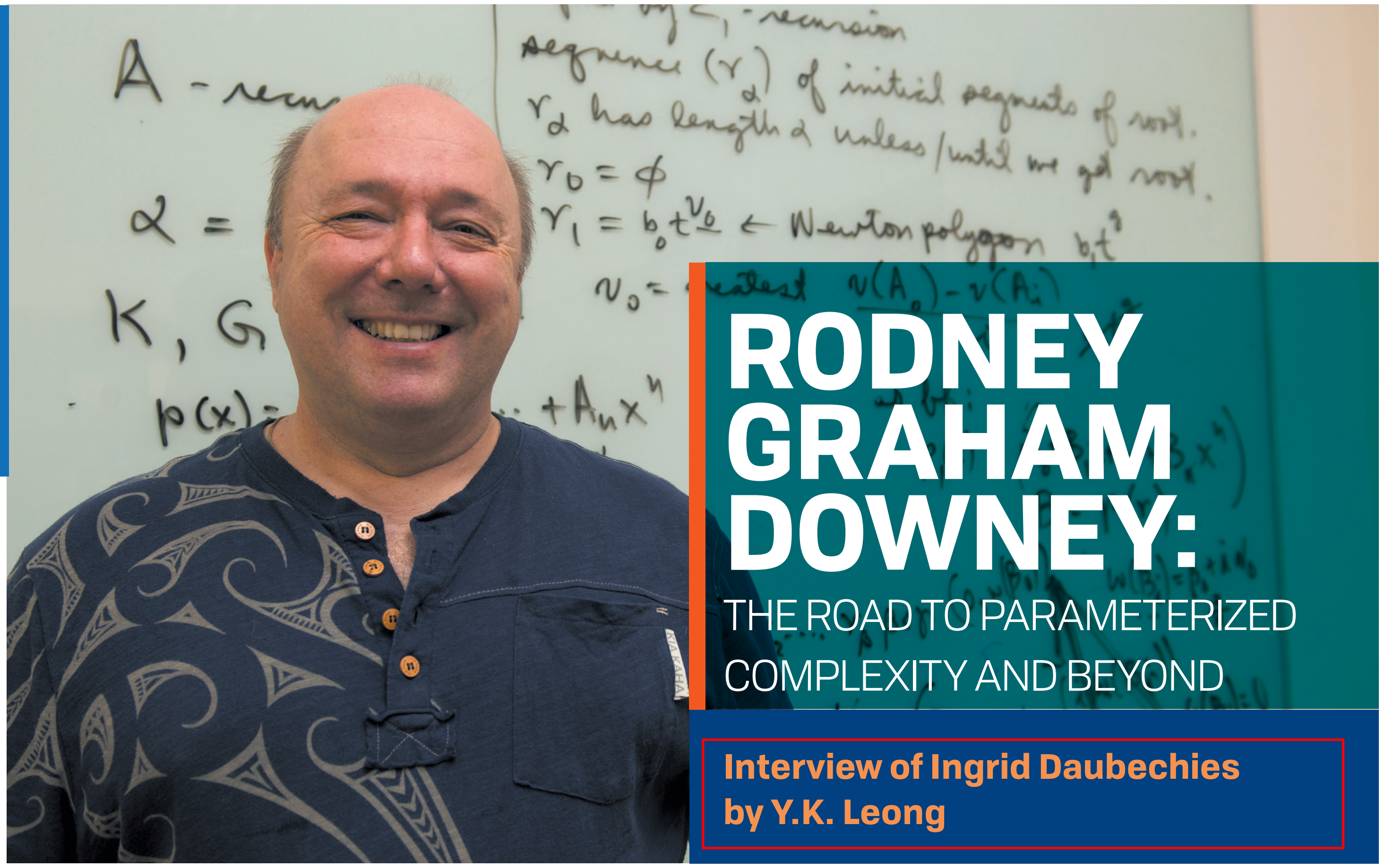

RODNEY GRAHAM DOWNEY:

THE ROAD TO PARAMETERIZED COMPLEXITY AND BEYOND

Interview of Ingrid Daubechies by Y.K. Leong

~~Rodney Graham Downey: The Road to Parameterized Complexity and Beyond~~

Rodney Graham Downey has made fundamental and far-reaching contributions to mathematical logic, effective algebra and theoretical computer science.

Of Scottish descent[4] Downey obtained his BSc from the University of Queensland in Australia and his PhD from Monash University. Subsequently, he taught briefly at the Chisholm Institute of Technology (now part of Monash), then National University of Singapore (NUS) and University of Illinois at Urbana-Champaign. He moved to Victoria University in Wellington in New Zealand in 1986 and was shortly promoted to Reader in 1991. He was given a personal chair in mathematics in 1995 and is reputed to be one of the top three computability theorists in the world.

Downey's research in the theory of computation, and complexity theory has led to deep contributions on the relationship between algebraic and descriptive complexity vs algorithmic complexity. Early in his career, he and his former colleague Michael Ralph Fellows did some ground-breaking and influential work in parameterized complexity. This was initially considered to be a peripheral area of classical and applied computability theory and complexity theory and is now a flourishing area of computer science with regular meetings at Schloss Dagstuhl – Leibniz Center for Informatics. This center is considered to be the informatics (computer science) equivalent of the Oberwolfach Research Institute for Mathematics in Germany. Characteristic of his style of research, Downey moves from one field to another at regular intervals and has also done important work in effective algebra, reverse mathematics and algorithmic information theory. His current research interest is in model theory and computability theory.

His research output totals more than 280 (single or joint author) research papers in leading journals and conference proceedings. He has written 5 books: Parameterized Complexity (with Michael Fellows), Algorithmic Randomness and Complexity (with Denis Hirschfeldt), Fundamentals of Parameterized Complexity (with Michael Fellows), Minimal Weak Truth Table Degrees and Computably Enumerable Turing Degrees (with Keng Meng Ng and Reed Solomon), and A Hierarchy of Turing Degrees: A Transfinite Hierarchy of Lowness Notions in the Computably Enumerable Degrees, Unifying Classes and Natural Definability (with Noam Greenberg).

In addition, he has also co-authored and co-edited 14 books that are special issues of journal publications, or proceedings of workshops and conferences, and has served on the editorial boards of leading journals in logic and computer science, notably Bulletin of Symbolic Logic, Journal of Symbolic Logic, Theory of Computing Systems, Archive for Mathematical Logic and Computability. He currently edits 5 journals.

Downey's scientific contributions have been nationally recognized in New Zealand with numerous awards and honors, notably the Hamilton Award for Science, New Zealand Association of Scientists Research Medal for the best New Zealand based scientist under 40, Hector Medal (Royal Society of New Zealand), Fellowship of the Royal Society of New Zealand and the Rutherford Medal (the premier award of the New Zealand Royal Society worth $100,000).

The international recognition of Downey's work is clearly reflected by invitations to speak at the International Congress of Mathematicians, International Congress of Logic, Methodology and Philosophy of Science, and Association for Symbolic Logic (Gödel Lecture) and by his election as fellows of the following foreign scientific bodies: Institute of Combinatorics and its Applications, Association for Computing Machinery (becoming the second ACM Fellow in New Zealand), Isaac Newton Institute for Mathematical Sciences, American Mathematical Society, Australian Mathematical Society and Institute for Mathematical Sciences (NUS).

He has twice won the Shoenfield Prize (Association for Symbolic Logic) for articles on his work on randomness, and for his book on algorithmic randomness and complexity, Nerode Prize (European Association for Theoretical Computer Science) and Humboldt Research Prize Award (60,000 Euros).

Though his passion in mathematical research is all-consuming, Downey has always been an avid sportsman. Besides having been, in his younger days, a volleyball state player, a rugby forward player at school, and a squash player of high standard, he is currently a keen tennis player while still maintaining his lifelong passion in surfing. Perhaps what is not so well-known is that he is heavily involved in Scottish country dancing in which he is a qualified teacher and choreographer with 5 books of dances.

Downey has a long association with the National University of Singapore (NUS), dating back to 1983-1985 when he took up his first permanent academic post, as a lecturer at NUS. However, he left NUS for a short visiting position at the University of Illinois at Urbana-Champaign and then moved to Victoria University of Wellington in New Zealand, which is geographically and culturally closest to his home country of Australia. In spite of the relative academic isolation of New Zealand during the pre-internet period, the move proved to be decisive – soon afterwards, he found in Michael Fellows a collaborator with whom he would make his first significant impact in complexity theory. From the beginning of his research career, he had already shown a restless streak in research collaboration. He would seek, on his own local as well as overseas grants to travel to the United States, especially to Cornell and Chicago. His gregarious nature soon established a routine of regular overseas visits to the United States, Europe and Singapore. Downey's attachment to Singapore is evident from the fact that he visits Singapore very often, practically every year (before the recent pandemic). Since 1993 he has been invited to NUS and NTU (Nanyang Technological University) as a visiting professor and as a member and fellow of the Institute for Mathematical Sciences (IMS) a number of times. He has collaborated with faculty members of NUS and NTU such as Chi Tat Chong, Frank Stephan, Yue Yang, Guohua Wu and Keng Meng Ng.

As a celebration of Downey's research contributions, his friends and collaborators within Singapore and without organised a workshop "Aspects of Computation" on parametric complexity, algorithmic randomness, classical computability theory and computable structures and reverse mathematics at IMS from 21 August – 15 September 2017. It was during his visit for this program that Y.K. Leong interviewed him on 12 September 2017. The following is an edited and enhanced version of the interview in which he traced the path he took from his early attraction to logic in high school in Queensland, Australia to the pinnacles of logic in the remote capital city of New Zealand. We also get a rare glimpse of the views of an unusually prolific mind on mathematical logic and theoretical computer science and of a not so glamorous side of mathematical research.

Acknowledgement. Y.K. Leong would like thank Von Ping Yap of the Department of Statistics and Applied Probability, National University of Singapore for preparing a raw draft of the transcript of the interview.

IMPRINTS I In an autobiographical note, you have written that your interest in logic came in early during your high school years. Interest in logic at such an early age seems to be rather unusual for those who are gifted in mathematics. Most of them would be attracted to number theory, combinatorics, geometry and things of a more computationally concrete nature. In retrospect, do you see any indication in your early childhood of any tendency towards a logical or algorithmic attitude or approach in your thinking process?

ROD G. DOWNEY D I think it's hard to recall my early childhood, to be honest. My parents were not well off and I know I went to 7 primary schools, so I guess I had "no fixed address" until grades 6 and 7, the last two years of primary

[4] Actually, of mixed descent. It also includes English, Irish, Welsh, Nordic, and Chinese.

school. I know at primary school, I was kind of turned on to math by a particular teacher I had there (a guy called Harry Seldon who, I remember, put up Pascal's triangle), going "Wow, look at that! That's really interesting." And I remember, at primary school I was reading the high school book on Euclidean geometry and I thought that was pretty interesting. I think the interest in logic would have come in secondary school because logic was offered as a subject at high school. And I just thought it was interesting; so I did what I wanted to do: Math I, Math II and Logic. I think I was the only person probably ever to do this combination of subjects. It was kind of fun really, you know, the problems of induction by Hume and modal logic and stuff.

When I went to university, I was planning to do chemistry, but when I got through first year chemistry, that was enough for me. I thought, you know, actually math looks a lot more interesting, especially when I was told I probably wouldn't be any good at it [mathematics] by my academic adviser. I had been pretty wild in my first year of university and had done almost no study.

But I don't know. Young mathematicians are often attracted to fundamental issues and there is no doubt that logic is concerned with fundamental issues, and in mathematics, it kind of feels good to study it [logic]. You feel like you're dealing with the fundamentals really.

I But very few schools offer logic as a subject, isn't it?

D: Correct. It's no longer offered in Queensland, but back in those days they had [such] a curriculum. Funnily enough, logic was put in schools for those people who were not good at mathematics. If you're good, you could do like mechanics and things like that. We had a small number of options and "science" students did 6 subjects: Maths I, Maths II (mechanics), English, Chemistry, Physics and one other subject, usually "tech drawing" for male students and biology for female. A forgotten age. Logic was a maths alternative subject in the "humanities" or "commercial" streams and usually there was no crossover. The idea is that non-science students would not need, for example, calculus, but logic would somehow prepare you for reasoning in life. [Added December 2020: A bit of logic in our dealing with Covid-19 would have not gone astray.]

I Many Australians would have chosen to go to England or the United States for their graduate studies in the 1970s and 1980s. Why did you choose to go to Monash University even if logic was done there?

D Well, again, I think there were two reasons. Firstly, it was a different age. In some sense the reason I went to university at all was that Gough Whitlam had been voted into power in Australia, and universities had become free for all. I was very lucky and because of my family's modest means, I also got lots of financial assistance. But also I think there's a lot more opportunity to go overseas now than there was then (although this could be my ignorance in the 1970's). I applied for standard scholarships, like the Rhodes Scholarship and the Commonwealth Scholarship and things like that. I got interviews for the Shell Scholarship and the Rhodes Scholarship, but didn't get them. (Shell did make me a job offer, though!) I don't know why I applied, as I had no especial desire to have them. I just heard they were prestigious scholarships, but I had no particular reason [to apply for them]. And I think, more importantly, I was kind of naive in the ways of all things academic because, I mean, coming from a background where nobody in the family had ever gone beyond 10th grade, I had no idea why wouldn't Australia be just as good as anywhere else, you know? Then in my final year at university, I did get an interest in advanced mathematical logic (particularly computability related issues) studying things like the Eklof-Fisher work[5] on the decision procedures for abelian groups. So I kind of liked logic. The only people who did any computability theory anywhere in Australia were at Monash. So then I got a scholarship to go down to Monash to do a PhD. It's kind of funny that I've seen a lot of people who have got PhDs from the so-called power universities and who weren't actually that good in the long run. I speculate that perhaps this is because there's a vested interest in those universities to get people through PhDs. Of course, there are tremendous people that come through, but I'm just thinking about my own university [Victoria University of Wellington]. We have some people like that and we have three world-class mathematicians, Geoff Whittle, Robert Goldblatt and, I guess, myself (talking about "older members"; we also have a bunch of talented younger people. We definitely have a world class logic group.). Geoff Whittle's work was spoken about at a recent International Congress of Mathematicians – very, very difficult high-powered brilliant work in matroid theory. And Rob Goldblatt is extremely well known in logic and topos theory. Geoff Whittle got his PhD in Tasmania, and Rob Goldblatt got his PhD at Victoria University of Wellington. So he [Goldblatt] didn't even ever leave the university that he started from, which I don't think is really a good idea. None of these people did their PhD's at "power" universities. So I think somehow if you do your PhD away from the major centers, you may not get all of the ideas, but on the other hand, it's all your work and you really learn to work independently. And coming from that environment, you can still keep working as you have learned how to work independently, whereas I've seen some people who maybe can work at Harvard or somewhere, but when they're not in such rich environments it's much more difficult for them.

I Was John Crossley your supervisor?

D Yes. He was a supervisor in a kind of the British sense, which was rather hands-off in many ways. But he did have a lot of good visitors and an extremely good library, a personal library of preprints and things, which were important in those days. Because of that, I learned to do independent research. And Chris Ash was down at Monash, which was very lucky for me. Chris Ash was an extremely good logician and we spoke a lot. He committed suicide at a young age, sadly[6].

And there was another guy called John Stillwell. His PhD was actually in recursion theory but he later changed [his area] and he's now well-known for his expository works in the history of mathematics and in topology. I learned topology from him. I thought that was extremely interesting. I can't say that not doing my PhD in one of those major places hurt me in any mathematical sense, but it can hurt you in other senses. I'm now old enough to have been on more appointment committees than I ever want to be, but you see people go, "Oh, look at that person. They've got an MIT PhD." And I think, "Nay, so what?" I mean, let's see what they do in the years after their PhD, you know. I think we're too easily blinded by this vision of a hierarchy in our own minds, but this is still true around the world. I mean, it's very rare to go to an Ivy league school and find professors who weren't also educated at Ivy league schools. So "placement entropy" seems to operate, as well as an "old boy" (or at least "old person" network). I think I saw this with my eldest son [Carlton] too. He's did his PhD at Carnegie Mellon in machine learning, and it's an extremely good machine learning department that just has connections everywhere. This made his post-PhD path easier, I believe. So I guess that matters.

I After your PhD, you spent a few short stints in Australia, United States and Singapore. You were, in fact, in Singapore for nearly three years before you moved permanently to Victoria University of Wellington in New Zealand after a year at the University of Illinois at Urbana-Champaign. It is understandable for you to leave Singapore, but to choose to move permanently to New Zealand rather than to the United States seems to defy logic. Is there any specific reason for this move?

D Well, first and foremost, there's more to life than academics. The job situation was very tight in the eighties. All of the jobs had been filled up by people during the Sputnik years. And then as now, Australia didn't really employ logicians except in computer science. I was offered a job at NUS. I thought, "Why not?" And so I came over here and I found NUS a remarkably pleasant place to be in. I did lots of research there. I enjoyed working with [Chong] Chi Tat and Yang Yue and people like that. It was kind of fun. But there is more to life than the academics, you know. My wife and I really wanted to bring our family up in the kind of environment that we remembered being brought up in. And that meant Australia and perhaps New Zealand. We had been both to New Zealand. So we knew it was a reasonable place to go to. And when I went to New Zealand, it was going to be for four years, and then we'd go back to Australia. But here I am, still in New Zealand and now a citizen, because Wellington's a lovely place to be in. And you know, it's turned out to be a great department too.

I Why not back to Australia?

D Well, I didn't get offered a job, did I? It was pretty simple. By the time I could've got a job in Australia, I didn't want to move. Because we already had kids and I was already senior and the whole thought of moving, you know, ... I had opportunities later to possibly move to chairs and things like that but I didn't take them. We knew we didn't want to live in the US either. I have lots of friends who are Americans. I like visiting the place, but I don't think I'd like to live there. We wanted our kids to just go down to the beach and do those things that Australians and Kiwis do. It's just, you know, play rugby, cricket and all that kind of stuff.

I You have a personal chair in New Zealand. They are quite generous to you in terms of being given the perfect liberty of doing almost anything you like, isn't it?

D Oh, in terms of teaching, there's never been any expectation. The only thing I had to do when I got there as a junior person was that they needed someone to set up a discrete math program. And I basically set up the discrete math program at Victoria [University of Wellington]. And after that, when I was doing more advanced courses, I could teach whatever I like. And so I remember at one stage, I thought, "Well, it could be fun to learn some computer science." So I taught complexity theory for a few years and that's where probably parameterised complexity came from and

[5] Paul C. Eklof and Edward R. Fisher, "The elementary theory of abelian groups", Annals of Mathematical Logic 4 (1972) 115 – 171

[6] Christopher John Ash (1945-95)

meeting Mike Fellows, of course[7] . This academic freedom was a kind of nice thing. Of course I always did my service teaching and I think that it is important that senior faculty are involved in, for example, first year teaching.

There were few constraints on me because I was always really productive and no one would ever have questioned that I wasn't working hard. And the other thing is, of course, in the early days, I always found that small trips to places and then working hard afterwards was definitely a way to be research productive. So I'd go and visit Richard Shore at Cornell or Carl Jockusch at Urbana-Champaign or someone like that. I learned a lot from Richard, Carl, Ted Slaman and others who changed my thinking about mathematics. And I'd work there for a while over a few weeks and then come back and look at the details. I don't necessarily find living in someone's pocket overly productive ... How often do you see people working with people in the same department over a sustained period? It's kind of rare actually. (I guess Hardy and Littlewood spring to mind.)

So the only difficulty in the early days was actually getting overseas because there was zero research money available. There was no granting agency (in New Zealand). So I used to beg these people. "Oh, look, if I can give you three talks here and three talks there, can you come up with, you know, a few hundred dollars to help pay for my airfare and things?" It all kind of worked out in the end. And now, of course, in New Zealand there's the Marsden Fund, which was a great initiative and funds blue sky research. Two good things happened to me: first, the internet came to be. I think the internet's made a big change for researchers away from the central places. If you go back probably a hundred years, any decent German university probably had more mathematicians than there were in the US, certainly at Harvard or something like that. Then there was a movement to the [United] States because of the War and things. So only then did US mathematics flower. And the internet came in in the early nineties and it was great because, instead of taking two months to write a letter to somebody and get a reply, it was kind of instantaneous. You can write a letter and you get an answer and you go, okay, and that kind of thing. So I could bug people really quickly rather than slowly. And the Marsden Fund has been very generous. I've been lucky enough to get lots of Marsden grants, and I've had great postdocs down through the years. I have had 22 of them, and many now occupy positions at leading Universities (like Chicago, Berkeley, etc), so I was lucky to have such brilliant people work with me when they were young.

I **You once mentioned there was kind of a fighting spirit that spurred you to do well in your undergraduate studies. Do you think that this is indicative of if not characteristic of a kind of contrarian spirit that drives you in your early research work in areas that were not considered to be promising?**

D I did see the questions beforehand and my wife said "absolutely yes" to this. [Laughs] My view in mathematics is kind of weird in the sense that I like to work on what I like to work on. And if something's interesting, then usually I find that if you follow your intuition, it probably will be interesting. And you know, I'm not one of these people that have, like, the reef fish effect with what's the current "research target" that the reef fish are following. This came, for example, with parameterized complexity. I mean, we used to get very discouraging reviews of that in the early days when people would go, "This seems to be a very small thing." (Even reviews saying, in effect, "if this is important why aren't people from e.g. Berkeley doing it?") And now there're millions of euros going into the research in Europe and things like that because it's turned out to be useful. People are using parameterized complexity methodology now in Australia for trying to solve deafness in Aboriginal children. So it's kind of shocking to me actually. I think you just got to follow your intuition rather than trends. I do what I think is interesting and also I get bored after about a decade; I want to do something else.

I **Although one often refers to "logic and set theory" as if they were inseparable twins, there is no distinctive field that combines logic and set theory in the way that fields like algebraic geometry, differential geometry, topology, algebra, algebraic geometry, etc combine the language and tools of two different fields. Do you see any signs that logic and set theory are drifting further from each other, logic towards more applications like applicable disciplines like computer science, and set theory towards rarefied universes of its own, with its own questions?**

D It's a very long question. Well, I'm not a set theorist and I don't actually know that much about set theory, to be honest. I mean, my understanding of pure set theory is that they have their own internal questions that have driven them like "What is the real value of the continuum?" But, of course, there are areas like descriptive set theory, which have been around since

[7] This meeting and subsequent development of parameterized complexity is reported in "The birth and early years of parameterized complexity,"The Multivariate Algorithmic Revolution and Beyond, Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday, Lecture Notes in Computer Science, Vol. 7370 (ed. Bodlaender, Downey, Fomin and Marx) Springer-Verlag LNCS 7370, 2012, 17-38.

Lebesgue[8] and Luzin[9] , and which are going through a tremendous development in the last few years with initiatives like Borel[10] reducibility where you're trying to understand the relative complexity of problems from analysis and things like that. This is a very flourishing area and it's actually interacting with the rest of logic and the rest of recursion theory.

This is an area that I'm working in myself currently too, in a kind of modest way, in a project trying to give a measure of complexity to discontinuous functions on the reals. So a function is continuous if and only if it's computable relative to an oracle. If I give you some kind of oracle, which tells you how to take rational balls to rational balls, which is a countable collection of things, so you can code them up with an oracle. Then if I know that oracle, then relative to that oracle, I actually have a computable function on the reals. The trouble is that it's more difficult to do this with discontinuous functions. So there's a nice problem: How do you extend a notion of complexity to discontinuous functions or how do you measure the complexity of a discontinuous function? And this is a really nice project because you're kind of climbing your way up through what's called the Baire hierarchy. This is some work with Linda Westrick and Adam Day. Amazingly enough, some of the fine reducibilities that we looked at turned out to correlate to things that people had studied classically, like there was a hierarchy called the Bourgain hierarchy, which we knew nothing about. And it turned out that viewing things computationally turned out to be identical to viewing things the way that people viewed them classically. The Bourgain hierarchy kind of classifies Baire functions. It classifies Baire class 1 functions according to how much they wiggle.

I **So you're going back to the roots?**

D Yeah. Logic does lots of those things. I remember talking to algebraists (who want to know what logic can do for them) and saying. "Well, we prove, for example, that the isomorphism problem for torsion-free abelian groups is what's called "analytic complete"." Now, what does that mean? Well, people study invariants. They want to know what's the dimension of a vector space. There are the Ulm invariants for abelian groups and things like that. Well, logic tells you how not to do things. Proving this isomorphism problem is analytic complete essentially shows that there's no way of assigning invariants of torsion-free abelian groups. That is, no "invariants" can simplify the problem below to make it easier than the trivial invariants of "isomorphism type". So in other words, there can't be any invariants by showing that it's as hard as any other isomorphism problem, which is kind of a nice result. It's using logic to answer a question you might have asked, which is a classical question.

I **It seems to me that in logic you sort of refer to the negative in a certain way. You cannot do this. You cannot do that.**

D Well, there are lots of examples of that. But there're always positive aspects to negative answers; not that I suggest that my work is like Turing's, but we witness in Turing's work on the "Entscheidungsproblem" ["decision problem"], the key idea is a compiler ultimately. Look at compilers all around the world today.

I **But he didn't use the word compiler at that time.**

D It was a conceptual compiler. The idea that you can have a single interpreter which does the work rather than designing a machine for each computational purpose. But there are also lots of positive mathematical applications of logic, such as new proofs of various results using methods from logic. For example, work by Niel Lutz (Jack Lutz's son). They're using computable methods to analyse Hausdorff dimension. So what happened was, I think, there was a divergence in logic while it was being developed, particularly computability theory. But now there's a big movement towards pushing it back into the areas from whence it came, towards understanding the computational content of classical mathematics.

There is also a reason that logic is intertwined with computer science. People like Moshe Vardi[11] observe that it is the calculus of computing. But I was never drawn to proof theory.

I **Set theory seems to be going the other way round.**

D I don't feel competent really to comment. For descriptive set theory, that's certainly not true [that it's going the other way]. Descriptive set theory, which is a certain subarea of set theory, has never left analysis. It's always been intertwined with analysis. And I guess also, [Saharon] Shelah's work on classification (that's all about mathematics) that tries to answer questions like "When do theories have invariants?" in a different way. His view is that (this is my view of his view; you've got to read that monstrous book of his)[12] either a theory should

[8] Henri Léon Lebesgue (1875-1941)
[9] Nikolai Nikolaevich Luzin (1883-1950)
[10] Félix Édouard Justin Émile Borel (1871-1956-)
[11] See "Interview with Moshe Vardi: Dare to Know" Imprints Issue 32, July-December 2018, 12-20
[12] Classification theory and the number of nonisomorphic models", Vol 92 of Studies in Logic and the Foundations of Mathematics. North Holland Publishing Co., Amsterdam, second edition, 1990

have a small number of models with a certain collection of invariants, and here they are, or they just have too many models to have invariants. Then they basically look like trees and there are just too many trees. So you can't hope to have invariants. This is called the Dichotomy Theorem.

I Other than the famous P = NP problem, what are some of the central problems of computability and complexity? Do you see any prospect of the P = NP problem being solved within the next 10 years? (There are actually two questions here.)

D Well, I think it's fair to say that most areas of mathematics have their own internally generated problems. And they gain importance when people can't solve them. Sometimes it's not even clear why they're important; it's just that people tried to solve them. Witness Fermat's last theorem. The fact that it's solved kind of damaged the research. I mean, it has developed Kummer's theory of ideals, all kinds of wonderful things, but now it's being solved, right? Of course, computability and complexity has had many, many of their own problems. I can name some of the things: Vaught's conjecture, Martin's conjecture, and these are internally generated problems, which give insight into the areas of their study. And most areas that I've kind of moved into or looked into are full of these things: the classification problem, the invariants problem. There're all kinds of problems like these. But are there other really profound things in logic which can transcend mathematics like the P versus NP question? Probably not. It's hard to say. I think that history decides these things.

Anyway, there are a lot of problems in complexity theory which seem just as hard as the P versus NP problem. For instance consider the P versus BPP problem, where BPP stands for "bounded probabilistic polynomial time". BPP are more or less those things which have probabilistic algorithms. And the famous example of something in that class was primality testing, and that was known to be BPP by using the Solovay-Strassen[13] algorithm, for example, which is a probabilistic algorithm with one-sided error. And it either says no, the thing is not prime, or yes, it may be prime with some error like 1/2. But if you then do it a thousand times independently, the probability that you have an error believing that it is prime, when it's not prime is like very, very low. However, this problem was famously proved to be in polynomial time (by a very bad algorithm in terms of running times). That is, a BPP problem was re-classified in to P. And now it is thought that everything in that class BPP is in polynomial time. When I was young, we thought BPP did not equal P, but now there's a belief that actually BPP = P: That is, everything that has these randomized algorithms can actually be de-randomized. We have no idea how to do prove this and it's kind of an important problem because, we do have BPP algorithms for lots of things. 1

Polynomial identity testing (PIT) is a good example. The problem is specified by being given a polynomial in a whole bunch of variables over a finite field. The question I want to answer is, "Is the polynomial uniformly zero?" For example take z3z1z2 – z2z3z1 , which is always zero no matter what input I give it. There is a randomized algorithm for PIT but the algorithm is kind of dumb. You just put in random values from the field, assuming the field is large enough. And if you get zero, you go, "yes". If you don't get zero, you go, "no". "No" is definitely correct. If you say yes, then it is probably correct because the chance of guessing a zero is very low as zeroes for nontrivial polynomials are very sparse. And why is this an important problem? Because there are lots of things which can be transformed into PIT using an efficient reduction. Thus to solve our problem we transform it into a polynomial and solve polynomial identity testing. (Examples of this methodology can be found in my book Fundamentals of Parameterized Complexity with Fellows.) We have absolutely no clue how to derandomize PIT. So that's a nice problem.

There're lots of problems like this that we have that we're not even close to solving. And it would be hard to believe that all use the same trick that P versus NP does. But, on the other hand, you know, there's another one (my favorite): "Is W[1] = FPT?"[14]

But actually, if you want a really important problem [for practical computing], I say to myself, what is the goal of computer science? What is the goal of computability theory? And I think the answer is: to understand computation. And so I think that the biggest problem that I'd like to try and solve is the following. I don't know how many mathematicians out there would know that, in fact, Sat Solvers (solve the satisfiability problem for propositional logic, something which is a fundamental NP-complete problem) are pretty efficient in industrial problems. So what you do is -- it's a very weird thing -- you take an industrial problem and you use an efficient reduction to convert it into an instance of satisfiability. That's NP complete: satisfiability. And then you run it through a Sat Solver, something which has been algorithmically engineered to solve this. Now, we know that theoretically it won't work, but for lots of natural problems appearing in real life, it works very well indeed. You know, NASA uses it for their independent robots and all that kind of stuff. We have no idea why it works well on such problems. I mean, we honestly don't know why. What is it about the topology (or something) of the "real world" which makes seemingly intractable problems tractable? If we could understand this it would revolutionize algorithm design.

I'm assuming P doesn't equal NP; of course, it would be revolutionary if P equals NP with an algorithm that is efficient. You know, modern banking security would be destroyed, and many computational tasks would become wildly more efficient. But I actually don't believe that P=NP with an efficient algorithm. (If P=NP via an algorithm which took polynomial time with an exponent or constant such as the number of atoms in the universe, it would not really be of much use.) Thus, if we could really understand what it is about the universe that enabled things that theoretically shouldn't work but does work most of the time, then that insight would enable us to more efficiently design things. This is a really hard problem, but I think it's the big challenge. Computing practice generating theoretical questions in discrete mathematics is an analogue to the fact that classical physics generated a lot of classical mathematics. Computing is a great source of wonderful problems in mathematics. Suppose a hundred years from now, we prove P ≠ NP is independent of the axioms of set theory, this problem of explaining practical tractability would be still a very important problem to solve. And I think that's a tremendously good problem. So, in other words, invent a theory of complexity that explains the actual behaviour of algorithms in real life. And we lack that completely. I mean, an average case doesn't work, a generic case doesn't work. Parameterized complexity explains things some of the time. There're all kinds of different approaches, but we don't have any approach in genuine mathematics. It could make a difference to mankind or something like that.

I Could there be any sort of limitations in the human brain that actually prevents certain problems from being solved?

D Almost certainly. But I think we won't know. I mean, if you take the "hard AI" view that we're just machines and therefore the incompleteness theorem must apply to us, then there are things that we won't be able to solve. The question also involves AI. I am aware that machine learning can solve hard problems like the game of Go better than humans. But also important problems like protein folding with far greater accuracy and we don't know how it does it! I wonder if we might get to the same state in mathematics.

I Do you think we're hardwired for logic?

D I would have thought so down through the years, but having seen recent events in politics, I'm not completely sure I'd agree with that. I thought the world was becoming more logical until recently. I mean, logic goes back a long way, but it was still an achievement. Throughout history, mathematicians regard Greek mathematics with awe. I mean, there was a good reason for that: it was an amazing achievement, really. Euclid's Elements is an amazing book. There is a reason that that horrible machine learning used on the net targets emotional responses rather than reason. We are emotional beings.

I Is there such a thing as evolutionary logic?

D Evolutionary logic ... what does that mean?

I I mean, we talk about evolutionary psychology, evolutionary biology.

D There are things called non-monotonic logic that studies the logics of things evolving with time. They use this in computer science quite a lot. There's something called linear logic, which is actually a kind of "capitalist" logic. You kind of only have a certain amount of things to spend. And every time you use something, you spend some of these. Non-monotonic logics are, generally speaking, a kind of logic of belief. You believe things and you'd make deductions, but then as things change, you update your beliefs and then maybe things you thought of before were no longer true. And, of course, that's very similar to the kind of logic that we use for the scientific method.

I Some call it fuzzy logic.

D Fuzzy logic, yeah. I remember one of my lecturers, when I was an undergraduate, said to me: "You know, fuzzy logic, that's an example of mathematical pornography." which I thought was hilarious at the time. And it turned out that fuzzy logic is actually used in these circuits in Japan, you know, these logic circuits that dry your washings and all that. They're using fuzzy logic. So who knows, I mean, there we go. Who knows what's important?

I How do you choose the problems to work on? Do you prefer to work on hard problems?

D Well, I just follow my intuition. If I look at something that looks good or doable, I'll give it a go. There are certain areas that I don't particularly want to work on because they're not to my taste. But sometimes you're driven by the need to find problems for students to do. That's about all, but usually I get bored after a while. Then I want to do something new. I like that initial learning when you get into a new area, when you kind of learn what's going on in that area. I like the challenge. So I had my parameterized complexity decade, and then I got into randomness. And now I'm more interested in

[13] Robert Martin Solovay, Volker Strassen

[14] W[1] = FPT, a central problem of parametrized computability

model theory and computable analysis, also online algorithms, and who knows what. Maybe I've only got a few years left, I suppose.

I Which result of yours surprises even you yourself?

D Well, there are lots of things that I've worked on. The answers have surprised me. I mean, there are lots of things which I thought were true turned out to be false. Lots of things that [I thought] were false turned out to be true. I think the thing which really did surprise me was in parameterized complexity. The stuff we [Mike Fellows and I] began working on years ago really has turned out to be actually useful and we wouldn't have anticipated that. I know we didn't anticipate it in the early days because I have old correspondence from Mike on where we were going, "Oh, do we think this would ever be useful?" And it turned out to be quite useful, both in terms of the mathematics they're now using, for example, the techniques in low dimensional topology and providing algorithms much better than the currently known algorithms in, for example, computational biology. So those were great surprises to me. You know, my wife was surprised too. She said, "You did something useful." I said, "Well, it wasn't my fault, [Laughs] I didn't try."

I There is a persistent perception, that mathematics is a closet activity with mathematicians working mainly on their own as exemplified by the well-known successes of Andrew Wiles and Yitang Zhang. How much of this perception has changed during the past 10 years?

D I think you can't avoid the fact that mathematics is a social contract. It has its own society and each sub-area of mathematics has a different way of working. And I think there are certain areas of mathematics where people do tend to work a lot by themselves. You could say that with number theory. Yet look, Terry Tao has lots of co-workers and is notably, I think, one of the first to get a Fields medal for lots of joint work. I mean, previous Fields medal winners didn't have much joint work. I think the fact that communication is so easy now surely must encourage people. I know there's this stereotype of the mathematician that sits in his room and doesn't talk to people and indeed I've met such people. But, on the other hand, I've met lots of mathematicians that like working with other people. I love working with people. It's much more fun, and you have that sense of competition, especially with younger people and you also have this to-and-fro flow of ideas. Why wouldn't you want to work with other people? It's bizarre to me that people wouldn't want to work with people. I've had 20-odd postdocs now, and I stay in touch with all of them. I mean, try and keep working with them. So yes, I think it's now changed. Maybe not in some areas, but who knows? Of course, there are some areas of science where you've got to have Sherpas. I mean, you can't do big experimental data without lots of Sherpas to help you with your research and experiments. I think in mathematics, certainly in computability and complexity theory, the number of authors per paper seems to be rising. There is surely a reason that places like Google, Facebook, etc have teams of people working on their problems, as the sum of lots of very smart people is certainly greater than its parts.

I It's not so much with pure mathematics. Maybe in applied mathematics, there is more collaboration.

D Well, in applied mathematics, you've got to have people doing computations and things. As data gets more complex to understand, you need more skills being brought to bear. I'm sure there will always be pure mathematicians who are the lone geniuses that just want to do what they want to do. But I think it's much more fun to work with other people. You've got to find people who care about what you care about too, because there are only so many people in the world that care about Martin-Löf randomness, or other specific things you care about. You should get to know them.

I On a philosophical note, mathematics is something that is sort of drawing necessary conditions. So if you're very, very bright, or let's say you have an infinitely intelligent person, then mathematics is really easy, isn't it?

D Is mathematics easy? I wouldn't know about super bright people, but I don't think I've ever met a mathematician who was successful and who didn't really work hard. When I was at university in my early years, I was pretty lazy because I had this kind of belief that, well, you know, if you kind of knew the axioms and kind of knew how things worked, you could survive (i.e. pass exams). But once you get to doing research, forget it, right? You really have to work hard. I've met people that pretend not to work hard. Well, I think there was an old British thing that you're supposed to pretend that you were just occasionally coming back from the golf club or something like that. But of all the people I know whom I would regard as really good mathematicians, they work hard. There're no two ways about it. You know, it's like if you work hard, you get lucky. You've got to be pretty obsessive. If you want to be good at tennis, you've got to be pretty obsessive about practice. If you want to be good at mathematics, you've got to be pretty obsessive about the problems. And they've got to be in your head all the time. You've got to fill yourself with them.

I This brings me to the next question. You have once written that mathematics is also a social activity and yet the breakthroughs are dependent very much on personal insights which occur only after an intense and almost obsessive preoccupation with a problem. What is your personal recipe for success in mathematics?

D My recipe is just as I would tell a PhD student, that if you really want to succeed in mathematics it's got to be something you can't do it from nine to five. It's got to be something you obsess about. One of the things that always annoy me is that, if I'm doing an administrative role, I wake up in the night thinking about how am I going to get some new course in or something like that. Whereas I should be waking up in the night thinking about a [research] problem that I've been working on. But I don't see how it's possible to achieve in mathematics without real work. I like working for short periods of time with people, but then I like going away and rolling it over in my head. And, you know, there are problems I've been working on for years that I still haven't solved and they come back to haunt me constantly. I don't know how other people work, but this is the way I've always worked: "Think really hard." If I do it too much, too long, I wake up [at night], I start feeling kind of unwell from indigestion and things. Like anything else, you've got to have little breaks. But if you're going to work on something, you

"I THINK YOU CAN'T AVOID THE FACT THAT MATHEMATICS IS A SOCIAL CONTRACT. IT HAS ITS OWN SOCIETY AND EACH SUB-AREA OF MATHEMATICS HAS A DIFFERENT WAY OF WORKING."

really got to work at it. You got to immerse yourself in the problem. It's the only way to go, just like anything else. I mean, if you're going to do something, you might as well try and do it well. I guess I had a really great training from my parents who didn't really care about success but about effort. I mean, they had no idea what I was doing, of course. They didn't really care about anything I did as long as I gave it my best shot. If you succeed, great; if you don't, well, you can't look back and go, "Well, I didn't try." I find mathematics very, very personally rewarding when things work out. It's very personally frustrating when things don't work out. To a student or a young researcher, I say that the most difficult thing in mathematical research is coping with frustration and getting used to the fact that most of the time you lose. The opponent wins and the opponent is more powerful than you are. If you work on hard problems, you will fail most of the time. But, you know, sometimes you get a little insight and that's good.