News from New Zealand

by

C. S. Calude

Department of Computer Science, University of Auckland
Auckland, New Zealand
cristian@cs.auckland.ac.nz

1 Scientific and Community News

The latest CDMTCS research reports are (http://www.cs.auckland.ac.nz/staff-cgi-bin/mjd/secondcgi.pl):

403 U. Speidel. A Forward-Parsing Randomness Test Based on the Expected Codeword Length of T-codes 05/2011

404. M.J. Dinneen, Y.-B. Kim and R. Nicolescu. An Adaptive Algorithm for P System Synchronization 05/2011

405. A.A. Abbott, M. Bechmann, C.S. Calude, and A. Sebald. A Nuclear Magnetic Resonance Implementation of a Classical Deutsch-Jozsa Algorithm 05/2011

406. K. Tadaki. A Computational Complexity-Theoretic Elaboration of Weak Truth-Table Reducibility 07/2011

2 A Dialogue with Juris Hartmanis about Complexity

Professor Juris Hartmanis, http://en.wikipedia.org/wiki/Juris_Hartmanis*, a Turing Award Winner, is the Walter R. Read Professor of Engineering at Cornell University. He is a pioneer, founder and a major contributor to the area of computational complexity.*

Professor Hartmanis eminent career includes also a strong service component: he served in numerous important committees (Turing Award Committee, Gödel Prize Committee, Waterman Award Committee); he was director of NSF's Directorate for Computer and Information Science and Engineering. Professor Hartmanis has been honoured with many awards and prizes. He was elected a member of the National Academy of Engineering and Latvian Academy of Sciences, and a fellow of the American Academy of Arts and Sciences, the Association for Computing Machinery, New York State Academy of Sciences, and the American Association for the Advancement of Science. He has an Honorary Doctor of Humane Letters from the University of Missouri at Kansas City and a Dr.h.c. from University of Dortmund, Germany.

Cristian Calude: Your early studies took you from Latvia to Marburg in Germany and then to the University of Missouri-Kansas City and Caltech in the USA. You studied physics and mathematics. Could you reminisce about those years?

Juris Hartmanis: My education took place in three different countries on two continents and in three different languages. It all started in Riga Latvia where I was born into a prominent Latvian family and had a happy childhood. My father was a high-ranking officer in the Latvian army and later chief of staff of the Latvian army. In Riga I attended the French Lycee and spent happy summers on our country estate. This all ended in summer of 1940 when the Soviet Union occupied the Baltic countries, Estonia, Latvia and Lithuania. The Russians arrested my father and our county estate was nationalized. The Russian occupation was indeed horrible: not only were high ranking military officers and government officials arrested, tens of thousands of people were deported to Siberian Gulags. In school we learned about the great achievements of the Soviet Union.

In the summer of 1941 Germany attacked Russia and in a matter of weeks the Soviets were driven out of Latvia and the German occupation started. Our county estate was returned and was instrumental in easing food shortages during the four year German occupation. The French Lycee became a public school and French was replaced by German as the obligatory foreign language.

For the summer of 1944 the family again moved to our county estate, in Western Latvia. By this time, the war was going very badly for Germany and by late

Fall much of Latvia was in Soviet hands and all land roots out of Western Latvia were cut off. Our family was lucky to be given a chance to leave Latvia by ship for Germany. In late October 1944 we landed in Danzig (Gdansk, Poland today) and found refuge in Marburg, a small German university town. I attended a German high school for a short time: air raid alarm interrupted the school year that ended in May 1945 when the Americans occupied Marburg. In 1947 I finished the Latvian High School in the Hanau displaced person camp and enrolled in the University of Marburg to study physics. Intellectually these were very stimulating times, but Germany at this time was a very sad place struggling to recover from the horrible devastation of the war. We took the first chance to leave Germany for United States and landed in Kansas City, Mo, in late 1949. The University of Kansas City (now University of Missouri-Kansas City) judged, to my surprise, that I had the equivalent of a Bachelor's Degree and gave me a fellowship for graduate work. Since they had no graduate program in physics I studied mathematics and earned a Masters degree in one year, in only three years of academic study.

In 1951 I applied to CalTech for PhD study in mathematics or physics. They admitted me to the mathematics program judging that I looked like an applied mathematician (with never having taken a course in applied mathematics!). CalTech was and is a superb school and I truly enjoyed my graduate work and Southern California. I earned my PhD in mathematics in four years in 1955 with a dissertation in lattice theory under Professor R. P. Dilworth.

CC: How did you get into computer science? Tell us about some of your early work.

JH: My road to computer science was not very direct. Shortly before my graduation from CalTech, Bob Walker from Cornell visited CalTech and, on the recommendation of Bob Dilworth, offered me and my friend and fellow graduate, John B Johnston, instructorships in mathematics at Cornell (in the 1950s academic careers started at the instructor level). We both accepted and I spent two delightful years in Ithaca. I loved Cornell University, the campus and the Finger Lakes region. I continued my work in lattice theory. During my second year at Cornell Dick Shuey from the GE Research Lab in Schenectady, NY visited Cornell and invited me to visit the GE lab. After a short interview at the lab I was offered a summer job in the newly formed Information Studies Section headed by Dick Shoey. (At that time I could not accept a permanent position at GE since I had already accepted a position as Assistant Professor at Ohio State University to work with Marshal Hall on the lattices of subgroups of groups.)

The summer at the GE Lab was an exciting and productive experience. The Information Studies section was exploring what research should be done to lay a scientific foundation for the emerging information and computing technology. In short, they strived to define and contribute to computer science. Intellectually it was a very intensive and gratifying time for me. By the end of the summer I knew that I was going to dedicate myself to the emerging computer science and that, after nine month at Ohio State University, I will return to the GE Lab as a research scientist. Indeed, I did return the following year to the GE Lab and spent seven happy years in Schenectady doing computer science.

My early work at GE dealt mostly with finite automata and their decomposition into smaller automata and exploitation of the decomposition theory to the state assignment problem for finite automata. At GE I learned about Shannon's theory of information and I was very impressed by it and one of my early papers "The Application of Some Basic Inequalities for Entropy" is a fumbling try to use entropy in formulation of a quantitative theory of computing. It turned out it was Turing's work that was the key to the quantitative theory of computing. Only after Dick Stearns and I studied Turing's work did we have the right tools to define such a theory.

CC: In your Turing-award winning paper with Richard Stearns you introduced the time complexity classes and proved the time hierarchy theorem. Tell us about the cooperation with Richard Stearns.

JH: The Information Studies section at the GE Lab had a tradition of inviting highly gifted graduate students and young faculty for summer jobs. Dick Stearns, a mathematics graduate student at Princeton, came to the Lab for a summer job shortly after I had joined the Lab. Dick was working on a dissertation in game theory at Princeton, but at the Lab he joined me working on the state assignment problem for sequential machines. By the end of the summer we completed our paper "On the State Assignment Problem for Sequential Machines, II". Dick was a highly gifted person and we worked very well together. He returned to Princeton for a year, finished his game theory theses, and, as we had planned, joined that GE Information Studies section. We resumed our collaboration and at the same time we intensively studied material related to computer science.

It is interesting to note that neither of us was familiar with Turing's work on computability (certainly I was not) and we studied material on Turing machines with excitement. We quickly realized that Turing machines may be the right model to explore the complexity of computations. Turing had shown that adding tapes to Turing machines did not change what they could compute. Our problem was to explore how the computational complexity of problems changed with changes of the Turing machine model. We very quickly showed that various changes of the Turing machine had minor effects on the computational complexity and we could quantify the changes. For example, the computation of a multi tape Turing machine, or one with two-dimensional tapes or even with multi-dimensional memory space could all be performed on a one-tape machine in the square of the computation time of the other models. These results assured us that the Turing machine was the right model to study computational complexity. With excitement we defined time-bounded computational complexity classes and investigated their properties. By time-bounded diagonalization we showed that a slight increase in the asymptotic time bound yielded a bigger complexity class. This showed that there are computations with very sharp computational complexity bounds.

Manuel Blum showed in his MIT dissertation that this is not the case for all problems by constructing exotic problems without sharp complexity bounds. We presented our first results on computational complexity in 1964, "Computational Complexity of Recursive Sequences", at the IEEE Annual Symposium on Switching Circuit Theory and Logical design. Quickly other results followed. With P. M. Lewis we investigated tape or memory bounded computational complexity classes, derived hierarchy theorems for tape bonded computations and other interesting results. We also introduced a new version of the Turing machine by separating the work tape from the read-only input tape. This allowed the investigation of the rich class of computations which required little memory. For example, we showed that context-free languages can be recognized on square of $\log(n)$ tape and that any non-regular language require at least $\log\log(n)$ tape (for inputs of length n). By this time, we fully realized that computational complexity was an exciting, rapidly growing and an important part of computer science. After seven happy and productive years in 1965 I left the GE Research Lab for Cornell to chair the newly authorized Computer Science Department.

CC: You created the computer science department at Cornell University and served as its first chair.

JH: During summer of 1965 Professor Bob Walker (the same person who brought me to Cornell) called me and invited me to visit Cornell to discuss the newly authorized Computer Science Department. I visited Cornell and at the end of my visit I was essentially offered a full professorship and the chairmanship of the Computer Science Department. The prospects, support and environment for computer science at Cornell looked so good that a short time later I accepted the offer. I have never regretted this decision.

My goal was to create a first class Computer Science department at Cornell with a great, informal and congenial environment for a cohesive group of scientists who will help define computer science and contribute to its development. We were very fortunate that our early hires, which included John Hopcroft, Bob Constable and David Gries, shared this vision, brought visibility to the department and helped to shape and develop the Department and computer science. I am delighted that they are still active at Cornell. I had a great time helping define computer science education, establish light teaching loads, to find the best possible faculty and create an informal, friendly and cohesive environment.

Besides leading the Department I had a great time doing research with some outstanding students. Jointly we explored the structure of complexity classes, studied the nature of complete problems and our conjectures initiated lively research activity in this area, we struggled with the P and NP problem and explored relativized versions of this and other problems. The role of sparse sets in complexity theory was explored and led to some interesting conjectures that stimulated some very good results. For me it was a real joy to work with my students, most of whom worked in complexity theory and made beautiful contribution to computer science.

CC: You have 20 PhD students and 198 descendants (cf. `http://genealogy.math.ndsu.nodak.edu/id.php?id=10404`), many of whom are prominent researchers themselves. Can you comment on your role as supervisor and mentor?

JH: My move from GE Research Lab in 1965 to head the new Computer Science department at Cornell was a very well timed move. At the Lab I had immersed myself in CS research and had learned a lot and created some computer science results. In particular, computational complexity theory was very well received and started to attract other computer scientists. Cornell provided me with an ideal base to expand my involvement with shaping computer science education, teaching theoretical computer science and, particularly, computational complexity. On the national and international stage I promoted computational complexity theory and actively participated in building the international theoretical computer science and computation complexity community. In all these activities, I was supported and worked with a stream of highly gifted and well-motivated students. I have already discussed some of the topics we explored with my students without explicitly mentioning their names; since all of them wrote fine dissertations it would not be fair to just single out a few without reviewing all their contributions. At the same time, I have to acknowledge that working with my students and seeing them succeed was one of the greatest experiences. They were all great individuals. I met regularly every week with my students for creative arguing and shouting matches about a broad topics in computer science as well as about most recent results from our group or outsiders. I met individually with my students for searching discussions of research. I have suggested very specific problems to some of my students and gladly listened to others who picked their own topics. I believe that creativity is highly individual and must be so understood and that the mentor-student relationship has to be built and mutual respect and even friendship. We also met on the volleyball court and delightful trips to conferences. It is a pity that volleyball has now been replace by hockey.

CC: Please comment your beautiful result regarding trivial theorems in formal systems with arbitrarily long proofs.

JH: I admire deep results in mathematics and computer science, but I am not particularly impressed by just the difficulty of a proof and certainly not just by its length. On the other hand, I had heard a lot of bragging about the difficulty and length of proofs and I started reflecting of how the length of proofs was related to the theorem it proved. After a while, I realized that in any reasonable formal system one can construct very simple sets of theorems (say a regular set) whose length of proofs grows faster than any prescribed recursive function in the length of the theorem. I had, somewhat mischievous fun deriving this result and it has made me even more suspicious of the long proof worshipers.

CC: Tell us about Gödel's lost letter and P = NP.

JH: Some time before June 1989 Dr. Gerhard Heise showed up in my office to discuss some matter in a hand written letter in German from Gödel to von Neumann, unrelated to complexity problems. It did not take me long to realize that in this letter Gödel in essence asked von Neumann about the computational complexity of a NP complete problem about theorem proving.

Dr Heise left me a copy of Gödel's letter which I found very fascinating and I was impressed by Gödel's curiosity about computational complexity of theorem proving. I translated the letter to English and published a note in the EATCS Bulletin, "Gödel, von Neumann and the P=?NP Problem". I spent some time searching for a possible reply from von Neumann but could not find it nor has it been fond since then. Von Neumann was not well at that time and we now have to assume that he never replied to Gödel's letter. I do not know if Gödel raised the same computational complexity problem with anybody else. A complete translation of this letter has been published since then with comments (my note on the letter contained only quotes of the parts relevant to the computational complexity question).

CC: What is "The Real Conjecture of Hartmanis"?

JH: I don't think that there is "The Real Conjecture of Hartmanis". There are several of our conjectures that stimulated a lot of work and others that should have. One of my early conjectures that there are no sparse complete sets for NP, unless P=NP, was verified by Maheney's beautiful dissertation and the sparse sets raised many other interesting questions in computational complexity theory. The Berman Hartmanis conjecture that all NP complete sets are polynomial time isomorphic stimulated a lot of work and some great oracle constructions to twist it either way. My feeling is that this conjecture may not be true, as stated, but it holds for all NP complete sets with a simple padding property which is possessed by all known NP complete sets. The conjecture also holds for NP complete sets defined under less power full reductions. See Manindra Agrawal's The First-Order Isomorphism Theorem, `http://www.cse.iitk.ac.`

in/users/manindra/isomorphism/uniform-ac0-iso.pdf.

The conjecture that I like very much states that: a real-time computable real number is either rational or transcendental, stated differently no irrational algebraic number is real-time computable. If true, this would give an amazingly powerful method to prove numbers transcendental. This conjecture emerged from Sterns and my failure to prove that the square root of 2 is real-time computable.

CC: This conjecture was called "The Real Conjecture of Hartmanis" by Lipton in https://rjlipton.wordpress.com/2009/02/24/a-conjecture-of-hartmanis/. My guess is that Lipton called in this way because he believes it's the most important conjecture you proposed. Does this seem reasonable?

JH: As stated above, I do like this conjecture very much, and it may be mathematically the most profound of our conjectures. At the same time, I am sure that it will be very difficult to prove, should it be true. There seem to be no mathematical results or techniques to attack this problem and in general proving numbers to be transcendental has in many cases been very difficult. Also, it is a conjecture about concepts from two different disciplines and thus not fully appreciated in either one of them. So far I know of only a few attempts to try to resolve this conjecture and so its impact on research in mathematics and computer science has been limited. Contrary to some of our other conjectures that have initiated a lot of good research. It is interesting to note that Steven Cook in his Turing Award lecture discusses our 1965 paper and particularly singles out this conjecture as "intriguing question that still is open to day."

CC: You have lead NSF's Directorate for Computer and Information Science and Engineering...

JH: I accepted the position of Assistant Director of NSF for CISE partially in gratitude for continues NSF research support at Cornell and for the opportunity to help guide the development of computer science in a new capacity. NSF is a great institution that has been and is vital to computer science research. For sake of brevity, I will just say that I truly enjoyed my two years at NSF and I am impressed how well CISE operated. It expanded my horizons about computer science and government research support and I would urge computer scientist to seriously consider serving some time at NSF or other government research organizations. Finally, Washington is a delightful city and I enjoyed it very much.

CC: What are your most preferred results?

JH: I do not have a preferred result. I had a lot of fun doing research and working with very original and interesting people. When I look back and think about specific results I almost always enjoy recalling how they were obtained and how they fit in the development of computer science. Somewhat like a father feels about his sons, I feel about my results, I like them all, some a bit more some a bit less, but all of them were a pleasure to create.

CC: You have been quoted by saying: "It's been a magnificent ride, like sitting in a cockpit and observing a brand new science being created. I am delighted and surprised at what impact computer science is having." ... "When I decided to be a computer scientist, I couldn't imagine the dramatic impact it has had."

JH: Indeed! We all can take great pleasure and pride in what computer science and computer technology have achieved. It is awesome! And I take particular pleasure in the elegance and beauty of computer science.

CC: How do you see complexity theory evolution?

JH: I am delighted to see how computational complexity is evolving and growing. Already in the late 60s and early 70s one could see that computational complexity theory was going to be an essential part of theoretical computer science. Today its relevance to computer science and even other sciences is fully recognized. I am particularly impressed be the widening scope and importance of computational complexity. For example, its relevance to cryptography and the impressive results about interactive proofs and non-approximability results. I am very grateful that I could participate in the founding, shaping and development of computational complexity and computer science.

CC: Many thanks.