Alan Turing: A retrospective - Audio

Alan Turing: The theory of computation

Blaine Price

My name is Blaine Price, I'm a senior lecture in computing at The Open University. I recently spoke to Darrel Ince Emeritus Professor of Computing and editor of the Turing Computation Papers. Turing widely regarded as the father of computer science, how does Professor Ince see his contribution?

Professor Darrel Ince

He invented, I think that's the best word, the Turing Machine. Which was a model of computation which is reflected in every sort of computer we use today. The computer in my iPhone, the computers in a large scientific establishment, the computers that are attached to the hull of a ship which measure temperature for climate researchers. He had a model of computation which lasts to this very day and works for tiny computers up to super computers.

Blaine Price

And can you explain briefly for us what his model of computation is?

Professor Darrel Ince

Well it's based on a tape machine where the computation is carried out by tape moving under a head, the sort of advantage of that it for example simplifies a description of an algorithm. If you look in any book on the theory of computation almost the first thing you meet is a Turing Machine and it was, when he developed it, it was a remarkable piece of futurology. It presaged the architectures in a sense the abstract architectures of computing and it was the Turing Machine and his other work which established an area of computing knows as the theory of computation. And what's staggering is that he was a prime mover when we didn't have any computers, he just conjured it out of the ether and the theory of computation is interesting it's quite a dense subject it's very mathematical but it's led to all sorts of practical applications, let's take an example. There's a sub-branch of the theory of computation called computational complexity, it developed in the 70's, it's in the same sort of flavour as all the other workings in the theory of computation. What it does is that it attempts to discover whether a particular algorithm or computer program is going to be efficient or inefficient. So for example let's take sat nav. You type in your destination and what happens is a computer inside the sat nav which is reflected in the Turing Machine does a calculation looks at all the potential roots and finds the one that is either the shortest or is going to take the least time. And Turing didn't work in the theory of computation complexity however what is certain is that

he established the whole discipline of which that is a sub-discipline and where we can now say yeah that sat nav application is going to run in a certain amount of time.

Blaine Price
You've mentioned the theory of computing one of the more famous problems that every computer scientist is taught about is the halting problem. Can you briefly explain what the halting problem is and how it woks?

Professor Darrel Ince
The halting problem is if you're given a description of computer program you have to decide whether that program finishes running or continues to run forever, so it's an example of a decision problem where you have a yes or no and it's one of the key results in the theory computation. It's easy to formulate and impossible to solve using the Turing Machine and what is important about the halting problem was that inside it there was a formal definition of what a computer is and what a piece of software and what a computer program was highly theoretical piece of work it's not something you put in front of the readers of a newspaper or a magazine but it was another example of his forward looking view of the world.

Blaine Price
You mentioned about some of his work at least being in unpublished manuscripts and today he's of course in computer science noted as one of the founding fathers as you send the noble prize equivalent is the Turing award. But in his time could you describe how he was regarded as an academic because he did a lot of his work and it was unpublished at the time.

Professor Darrel Ince
What's interesting, perhaps not at his time but he was regarded by a fairly narrow community as a leviathan. I think he was much bigger and much broader than that. What is profoundly impressive about his work is that it encompassed both theory and practice. On the one hand he applied logic for instance symbolic logic or predicate calculus what you like, mathematics, tough mathematics to theoretical problems and on the other hand he was involved in solving real life problems were real lives were at stake. You know he was a cryptographer, he used mathematics when he was at Bletchley he was involved in development of computers and I think when I edited the computation papers he was in a sense somebody who was recognised by computer scientists and by mathematicians but not in the wider world as that.

Blaine Price
You mentioned some of the breadth of his work, could you tell us a bit about some of the work he did that doesn't relate closely to computing, like morphogenesis?

Professor Darrel Ince

Morphogenesis is an interesting one and it's the one that he's still not known that well for. What he did was, I think it was 1951, was to describe the way that non uniformity things like stripes, spots, spirals etc. Arise naturally sort of out of homogenous states sometimes referred to as reaction diffusion theory in morphogenesis. And it has it saved as a basic model in theoretical computation biology and to this very day it's seen by some as a pre cursor to chaos theory. So what we're actually talking about is not just a man who does theory and gets his hands dirty in cryptography and computer building but somebody who jumps from one discipline to another.

Blaine Price

Where would you see Turing's enduring legacy?

Professor Darrel Ince

The Turing Machine, that will always be there as a key item. The fact that he articulated the importance of high level abstractions too often we're forced as computer scientists and software engineers into go down into details far too quickly without having some sort of theory. He was a model for researchers in the address theory, the Turing Machine and yet he went down into Bletchley into things like the enigma code. And what Turing is, is a model for doing basic research. He's a model for basic research and then goes out and does it in terms of details which a lot theoreticians tend to avoid a bit. He had a very panoramic view of his subjects he dealt with theory computability, hard ware, game playing, artificial intelligence, neural nets he even presaged the mathematical theory of programming languages and software so we're talking about somebody who when computers where non existent or when they were very, very primitive was talking about stuff which we now regard as mainstream.

Blaine Price

Thank you very much for your time.