Background Notes on Alan Turing: Computing Pioneer, War Hero and Gay Icon

Simon Ritchie

Algorithms

Before the invention of computing machinery, a Computer was a human who carried out computations. In the early nineteenth century French mathematicians came up the idea of breaking a computation down into simple steps expressed as an ***algorithm***. Since the computers only had to understand how to do each step, they didn't need too much training and were cheaper to hire. It was left to more skilled people to work out the algorithm. Organisations like insurance companies ran teams of computers, sitting at a desk and following algorithms.

An algorithm typically contains loops of repeated instructions and tests, such as:

1. Start with a number
2. perform a calculation on the number
3. take the result
4. if the result is less than 1, stop, otherwise go back to step 2 and repeat the calculation on the result that you just got

The computer runs the algorithm until the resulting number is less than 1 (or until it's time to go home). For a given starting number, the algorithm either halts with a result or loops forever.

A modern computer program is an expression of an algorithm.

Cambridge mathematician Charles Babbage promoted this approach in Britain. In 1837 he designed (but failed to build) the Analytical Engine, a programmable mechanical computer. Countess Ada Lovelace (daughter of the poet Lord Byron) wrote software for the proposed machine.

Babbage never managed to build his Analytical engine, but there is now a working version in the London Science Museum.

In 1854, during the Crimean War, Babbage figured out how to break the Vigenere cipher. It was used by the Russians and thought to be unbreakable. Babbage never published this work and it could well be that the British used it in secret to break Russian communications. As we will see, this pattern is familiar.

Telephony

In the early 20th century, telephony and radio were the hot technologies.

1891 Almon Brown Strowger invents an electromagnetic automatic call router, the Strowger Telephone Exchange.

1904 John Ambrose Fleming invents the Diode Valve and creates the new field of electronics.

1906 valves used in telephony.

By the 1930's the British telephone network was run by The Post Office (The GPO). The international telephone network was also largely British owned, linking together the greatest empire in the world.

The Post Office had its own training system including sponsoring bright employees to study at University. It ran a research facility at Dollis Hill.

Telephony research produced simple logic mechanisms. Initially these were purely electrical, using devices like relays, but later used electronics.

Much of the research behind modern computing and networking comes out of telephony. By the 1960s the international telephone network potentially allowed any telephone to dial any other telephone, anywhere in the world. Before the Internet was created, it was called "the biggest machine in the world".

Analogue Computers

Before automatic digital computer there were analogue computers. These were mechanical or electro-mechanical devices designed for a particular purpose. They were used for tasks such as controlling the firing of large guns. By the 1970s digital computers were cheap and powerful enough to replace them.

In the late 1930s Turing was inspired by the Liverpool Tide Predicting Machine to design an analogue computer to solve a mathematical problem – finding the first few thousand zeroes of the Riemann zeta function, in the hope of disproving a conjecture about prime numbers. He never finished this project, it became a casualty of the War.

Proof by Contradiction

Proof by contradiction is commonly used by mathematicians to prove that some conjecture is wrong. To understand Turing's proof of undecidability later, you need to understand this.

An alibi is a very simple example: somebody is accused of committing a crime in Coventry. The accused argues that he is innocent because he can prove that at that time he was in Leicester. To be guilty he would need to be in Coventry and Leicester at the same time, which is a contradiction. The initial assertion, that he committed the crime, must be false.

Formally, proof by contradiction works like this:

- Start with some assertion that you want to prove wrong.
- Produce a logical set of deductions from it
- Get to a contradiction

If the steps of the deduction are done correctly, then the initial assertion must be wrong.

You can find some mathematical examples here: http://www.math-cs.gordon.edu/courses/mat231/notes/proof-contradiction.pdf

The Foundations of Mathematics

In 1910 Alfred North Whitehead and Bertrand Russell (then a mathematician) published Principia Mathematica, an attempt to put mathematics on a rigorous foundation using set theory and symbolic logic. (Pages 1-379 of volume 1 led up to a proof that 1 + 1 = 2.)

Symbolic Logic allowed mathematical assertions to be expressed formally, and amenable to being manipulated by an algorithm.

In 1928 David Hilbert proposed a programme of research to prove that mathematics:

- is complete: every correctly posed mathematical statement can be proved or disproved
- is consistent (contains no contradictions)
- is decidable: there is an algorithm that can prove or disprove any mathematical assertion that can be expressed in symbolic logic notation

Kurt Godel proved the first two conjectures wrong: formalised mathematics is either incomplete (not every statement can be proved either true or false) or it contains inconsistencies, or both.

This left Hilbert's Entscheidungsproblem (the decision problem): is there an algorithm that can prove or disprove any mathematical assertion?

Hilbert believed that this was so, and started a programme of research to find the algorithm. Other mathematicians were sceptical.

If somebody suggests that something is always true, it's only necessary to find one case where it's not true to prove it wrong.

Mathematicians knew by hard experience that it's easy to design an algorithm that loops endlessly, and ***very*** hard to design one that is guaranteed to halt. Is it possible to create an algorithm A that will check if an algorithm B will halt? This is The Halting Problem. It can be expressed formally, so according to Hilbert, an algorithm must exist that can solve it. Conversely, if it can be proved that such an algorithm can't exist, then Hilbert's entire programme goes up in smoke.

In 1936 Alan Turing published "On Computable Numbers with an application to the Entscheidungsproblem", proving that the Halting Problem, and therefore the decision problem, was impossible to solve. Not bad for an unknown 24-year old.

To do this, Turing proposed a very simple machine that could replace a human computer. The machine would have an unlimited paper tape, divided into squares on which it could write a digit. The machine worked in binary, so each square could be blank or contain a 1 or a 0. It could move along the tape in steps, square by square, and read the digit from whichever square it was on. It could also erase the square, leaving it blank. It had a simple controller based on the kind of mechanisms used in telephony. As well as the working data, the algorithm that the machine was obeying could be stored on the paper tape.

Turing argued that this machine could do anything that a human computer could do by following an algorithm, and it could mimic any more complicated computing machine, so anything it couldn't do, couldn't be done.

He went on to prove by contradiction that the Halting Problem was not solvable:

- Initial assertion: there is an algorithm A that can decide whether an algorithm B will halt when it's run.
- Write an algorithm C that is just like A, except that instead of halting and declaring that B will stop, it enters an endless loop.
- Run algorithm C over itself
- If C will stop, it will run forever – a contradiction

Unfortunately, Professor Alonzo Church of Princeton University published a proof of the same conclusion a couple of weeks ahead of Turing. Church's proof used more conventional mathematics. Turing's argument was much more direct and robust.

Turing went on to show that the systems used in the two arguments were equivalent.

At the time, both results were fairly obscure. There were maybe a dozen people in the world who understood them. The deeper significance of Turing's paper only emerged much later.

King's College sent Turing to do research at Princeton where he met Einstein, Church and John von Neumann.

(von Neumann used Turing's ideas when he worked on the first electronic stored-program computer in 1945.)

We now understand that Turing's work is the basis of all modern computing and the underlying theory of computation. All modern computers and all the computers that we can currently conceive of are limited versions of a Turing Machine. The processor chip is the controller and the (limited) memory chips replace the (unlimited) paper tape. The program that the computer is running is the algorithm. A typical processor chip is vastly more complicated than Turing's simple controller, but both are equivalent – each can do anything that the other can do.

Babbage's 1837 machine was a mechanical example of the same principle.

Tring was later made a Fellow of the Royal Society for this work.

The Enigma Machine and the Typex

The Enigma machine was originally sold as a encryption device for commercial organisations such as banks. It demonstrated in 1932 at the congress of the International Postal Union.

The British took the ideas, spotted a major flaw and produced the Typex, which worked in a similar manner but without the flaw.

The German Government bought the Enigma design and developed it.

The flaw was that the Enigma always transformed a letter into a different letter, never itself. This gives a cryptographer clues about the original message.

Until 1939, both sides though that each other's system were unbreakable, then the British discovered that Polish cryptographers had broken various versions of Enigma using mathematics and a device called the Bomba. They exploited the same flaw that the British had seen in 1920 but the crucial breakthrough was to view the problem as mathematical, not linguistic.

When Germany invaded Poland, the Polish cryptographers escaped to Britain and worked at Bletchley Park through the War.

The videos mentioned below present a much clearer picture of the Enigma than I can give here in words.

Turing's Bombe and Other Work at Bletchley

Again, the videos give a better explanation than I can here, but in essence, Turing designed an electrical machine called the Bombe (named after the Polish Bomba) that solved the most difficult of the cryptographic problems posed by the Enigma. It was based on the Polish work but had some new ideas.

Enigma wasn't the only encryption device that Bletchley dealt with, and the problem solved by the Bombe was only the most difficult of a whole set. Turing became one of Bletchley's most senior designers, producing clerical procedures, paper devices, electrical and electronic devices to break enemy encryption systems of all sorts. By the end of the War he was acting as a consultant to all departments and so knew intimate details of all the work they did. This contrasted with the strict security imposed on most of the other thousands of Bletchley workers, who knew exactly enough to do their job and no more. He was also sent to the USA as a consultant to the American codebreakers. He was the codebreakers' codebreaker.

Turing also worked with the Post Office electronics research engineer Tommy Flowers and promoted his idea that valve-based electronics could be used to produce reliable and fast decryption systems. Flowers went on to design the Colossus, which Bletchley used to break the Lorentz cypher machine. This was a much more sophisticated device than the Enigma and was used to carry very sensitive high-level German military traffic, including Hitler's own communications.

Turing's own speciality was the Naval Enigma, the most difficult to break. Success with that allowed the British Navy to maintain its dominance of the oceans and eventually to eliminate the U-boat threat. The U-boats were very successful early on at attacking the Atlantic conveys that supplied Britain with food and other materials from the USA and Canada, but they needed to be supplied with fuel and torpedoes. Enigma decrypts told the navy where the supply ships and the U-boats were. They destroyed the supply ships first, and then the U-boats.

Without that success, Britain would have been starved into surrender. The Americans, Canadians and Australians would probably have continued the war in the Pacific but they couldn't have sent troops to invade Italy and France. It would have been left to Russia to invade Europe.

Turing's work alone was said to have shortened the Second World War by several years, and he inspired the whole work of the Bletchley codebreakers.

Bletchley was able to break most Enigma traffic for most of the War, but its work had to be kept secret. If the Germans had discovered that Enigma had been broken, they could easily have made it less vulnerable. So middle-ranking officers were told that the intelligence came from spies in the German military, and they had long experience that information like that couldn't be trusted. In any case, the whole structure of the armed forces had to be changed to make best use of the information.

Another problem was that simply having too much success would also give the game away, so the British had to careful how they used the intelligence.

Winston Churchill had done intelligence work himself in his youth. He was a great supporter of Bletchley and an avid reader of decrypts. He would embarrass his generals by quoting intelligence that they hadn't bothered to read.

It took until about 1941 to reorganise the armed forces to make full use of the Bletchley decrypts, and even then, there were failures caused by commanders not listening.

The British did run many very successful deception operations during the war: conjuring up fantasy armies, building fake targets to distract bombers, persuading the Germans that the V1s and V2s were not landing in the correct place and getting them retargeted. Bletchley was central to these, showing whether the deceptions were working and allowing a change of strategy where necessary.

The most famous is probably Operation Mincemeat. An invasion of Italy was imminent but the British planted information implying that the target was Greece. Bletchley decrypts gave a running commentary on the success of this stunt, showing the information being analysed and moving up the German Intelligence hierarchy all the way to Hitler, and then troops being ordered to move from Italy and to Greece, out of the way of the real invasion.

Operation Mincemeat was the basis of the film The Man Who Never Was.

Post-War Work

After the War, hundreds of trained engineers left Bletchley Park. Commercial Enigma machines were still in circulation and it was useful to maintain the illusion that they were secure, so the engineers weren't allowed to talk about what they had done. This hampered them, but they formed the base of the new high-tech businesses of the 1950s and 1960s.

Turing joined the new National Physical Laboratory and designed the Automatic Computing Equipment (ACE), but the NPL was bureaucratic and progress was slow.

The first modern computer was the American ENIAC, built in 1945 as part of the Manhattan Project, which produced the atomic bomb. John von Neumann was part of the ENIAC design team and passed on Turing's ideas from 1936.

Post-war American computing design was mostly funded by the Military and designed for their purposes. The British company Lyons (of tea house fame) produced the first business computer – Lyons' Electronic Office (LEO). The chief designer went to America to find out about computers, only to find that most of the background work had been done in secret in Britain.

Lyons and Ferranti funded computing research at Manchester University. Turing followed the money and moved to Manchester. He continued to be a consultant to the security services.

Turing had always been interested in the idea of machine intelligence and he was one of the early workers in that field. Apart from inventing the computer in the first place, he's known for The Turing Test, based on his Imitation Game. In this game, a person A sits at a keyboard and screen (in those days probably a teletype), exchanging messages with another person B and an intelligent machine C, bot somewhere else. A can ask B and C questions via the messages. A's task is to figure out which is the human and which the machine on the basis of the answers. If A can't do this, Turing argued that the machine is intelligent.

Call centres now use supposedly artificially-intelligent "chat bots" to answer customer questions. If you've ever been unfortunate enough to interact with one of these, you will know that, seventy years on, Artificial Intelligence research has not produced a device that's anywhere near passing the Turing Test.

Turing's Death

In 1951 British diplomats Guy Burgess and Donald Maclean defected to the Soviet Union. They were both Cambridge graduates, gay and had been spying for the soviets since the 1930s. This caused a huge scandal on both sides of the Atlantic and gay men were suddenly regarded as a security risk. As a gay Cambridge graduate, Turing was in serious danger.

In 1952, Turing was charged with committing homosexual acts. He pleaded guilty and avoided jail, but had to undergo hormone treatment for a year – chemical castration. He also lost his security clearance and with it, an important part of his identity.

Turing was found dead of cyanide poisoning at his house in Manchester in 1954. He had a cyanide-making experiment running and there was a partly-eaten apple by his bed. The coroner concluded that Turing had deliberately spiked the apple with cyanide. He didn't bother to have the apple tested. Verdict suicide.

Turing's mother and others argued that he had just been careless and got cyanide on his hands. She had often chided him about personal hygiene. Also, his probation and the hormone treatment had ended a year earlier, so why now? His death is another enigma.

Turing remained a fairly obscure figure until 1975 when information about Bletchley Park was released under the 30-year rule. He's now regarded as one of the leading scientists of the 20th Century.

Further Reading

Alan Turing: The Enigma by Andrew Hodges pub Hutchinson

The definitive biography of Turing.

The Code Book by Simon Singh pub Fourth Estate

Covers the whole history of cryptography including Enigma and Turing's work.

Colossus: Bletchley Park's Greatest Secret by Paul Gannon, pub Atlantic Books

Covers Lorenz, a more sophisticated cypher than Enigma, and Colossus, the machine invented by Tommy Flowers to break it. One of many Bletchley technologies inspired by Turing's earlier work.

Enigma by Robert Harris pub Hutchinson

Harris' novel describes the working atmosphere at Bletchley. It was made into the film of the same name in 2001.

The Imitation game

This 2014 film is supposedly based on Hodges' book, but unfortunately presents a travesty of Turing's time at Bletchley Park. If you want to know the real story of his life and work, read the book.

Some Useful Videos on Youtube

Analogue computer for controlling naval guns: https://www.youtube.com/watch?v=s1i-dnAH9Y4

The Liverpool Tide Predicting Machine: https://www.youtube.com/watch?v=roNyTlGiz5o

The Turing Machine: https://www.youtube.com/watch?v=dNRDvLACg5Q

The Halting Problem: https://www.youtube.com/watch?v=macM_MtS_w4

Enigma:

https://www.youtube.com/watch?v=mcX7iO_XCFA

https://www.youtube.com/watch?v=G2_Q9FoD-oQ

(Both of these videos are a bit confusing at times, but the two together paint a fairly good picture.)

Professor David Brailsford on the Polish code breakers, Turing and Bletchley Park: https://www.youtube.com/watch?v=kj_7Jc1mS9k

Turing's Bombe: https://www.youtube.com/watch?v=V4V2bpZlqx8

John von Neumann: https://www.youtube.com/watch?v=macM_MtS_w4