INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 19 (2012) 487–491
DOI: 10.1111/j.1475-3995.2012.00848.x

A celebration of Alan Turing's achievements in the year of his centenary

Edward Hermann Haeusler

Department of Informatics, Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil
E-mail: hermann@inf.puc-rio.br; edward.haeusler@gmail.com

Received 08 February 2012; accepted 08 February 2012

Abstract

In this paper, we review some of the main achievements and contributions of Alan Turing to Computer Science and related areas.

Keywords: history of OR; computing science; computational complexity; complexity theory

Alan Mathison Turing was one of the greatest minds of the last century. He created one of the most intuitive frameworks for defining computable notions, culminating with the design of the electronic computer. He was born on June 23, 1912, in London, and we celebrate this year the centenary of his birth.

Alan Turing obtained his diploma in Mathematics at King's College in Cambridge in 1934, where he was elected Fellow in the following year. Turing got his PhD in algebra and number theory at Princeton University in 1938. He had a part-time appointment at the Government Code and Cypher School (UK) from 1938 to 1939. This appointment became full-time from 1940 to 1945 and he led the cryptanalysis team based at Bletchley Park (UK) that broke the ENIGMA cyphering code used by the German Army during the Second World War. Next, he joined the National Physics Laboratory (UK) as the designer and leader of the ACE computer project from 1945 to 1947. After a sabbatical year (1947–1948) in Cambridge, he joined the University of Manchester, as a Reader in the Mathematics Department, where he was appointed "Deputy Director of the Royal Society Computing Machine Laboratory." There Turing designed the programming system of the first commercially available computer, the *Ferranti Mark I*.[1] He was elected Fellow of the Royal Society in 1951. On his last days, Alan Turing was involved with the mathematical definition of evolving biological systems and morphogenesis.

[1]Important observation: the *Ferranti Mark 1* should not be confused with the MARK I computer. The former was the first commercially available general-purpose computer and was designed at Manchester University, UK. The latter was a joint project from Harvard University, Cambridge, USA.

Published by Blackwell Publishing, 9600 Garsington Road, Oxford, OX4 2DQ, UK and 350 Main St, Malden, MA 02148, USA.

Almost every human being does not spend a day without using, directly or not, a computer program. If one makes use of mobile phones, DVD players, television sets, or vending machines, or if this person travels in cars, aircrafts, trains, and other transportation vehicles, most of the tasks involved in their operation is performed by a computer program. Without the help of a computer, imaging diagnosis by means of magnetic resonance would not be possible. Tasks so distinct as flying an aircraft and recognizing a smile on a scene are coded along the same basic principles, and could run on the same processor. In 2003, a technologically updated car already had *circa* 50 micro-controller/processors in its electronic design (Grimm, 2003). In this car, tasks responsible for providing safety, comfort, and functionality are distributed over these processors and they communicate with each other by message exchanging, which makes this car a computer network itself.

Algorithms exist since human beings attempted to reproduce certain daily needs, such as teaching friends to reach places already visited or to recognize footprints of different animals. Knowing some algorithms is nowadays an important part of human knowledge. Our children normally spend many school years learning important algorithms, developed during the last two or three millennia. The ability to correctly perform the four basic arithmetic operations is among the most important pieces of knowledge acquired by the human race. Iterative methods to evaluate surfaces have existed since the time of Archimedes of Syracuse, not to mention the Sumerians and ancient Egypt. The essence of an algorithm is its universality, in the sense that it can be performed by any human being with very common abilities.

In 1936, Alan Turing showed to the academic community that it is possible to define a machine with the ability of performing the very execution of any machine of its own kind. He presented a formal mechanism to compute transformations of strings of symbols into strings of symbols on a character-oriented base, and formalized these transformations using this mechanism itself. Turing proved, in this constructive way, that it is possible to build a universal machine, i.e., one that performs the task of executing any other machine of the same kind. A programmable task is, after Turing, any task that can be expressed by a machine of this kind. Afterwards, Turing's creation was named the Turing Machine. This model opened the path to engineers to build programmable mechanisms. At this time Turing was heavily influenced by Hilbert's program on the consistency of mathematics, but it was surely his mechanical view of nature that made his machine-based computational model so successful. The importance of Turing's work on the definition of a symbolic mechanical based computing artifact can be appreciated by the fact that the Turing machine is a formalism so robust, that it is still used nowadays as a fundamental tool in computational complexity.

Some authors like to select some of Turing's statements in various articles and presentations to show that he anticipated some ground-breaking ideas, such as quantum computation or connectionist models of intelligence, creating a controversial discussion about the origins of such ideas. This is needless, since his contributions to the development of human knowledge cannot be obscured by these minor discussions. Nobody can omit his name from the history of science as the person responsible for defining computation by both a mechanical and theoretico-formal apparatus. According to Gödel (1965), the father of a purely arithmetical approach to computation, Turing's work provided "a precise and unquestionably adequate definition of the general concept of formal system." In a review of Turing's (1936–1937) "On computable numbers with an application to the Entscheidungsproblem," comparing his own notion of λ-definability, Gödel's notion of recursiveness (via equational calculus) and Turing's effective calculability (via mechanically idealized machines), Church (1937) remarked that Turing's approach "has the advantage of making the identification with effectiveness in the ordinary (not explicitly defined) sense evident immediately."

Besides his theoretical influence on the creation of the computer machine, he designed his own computer (Turing, 1986) and wrote some reports on the art of programming (Turing, 1948). His influence on the EDVAC (Electronic Discrete Variable Automatic Computer) may be overestimated in the history of digital computer design. His work was important for Von Neumann, one of the minds that influenced the EDVAC project. Turing led the automatic computing engine (ACE) team at the National Physical Laboratory, London (UK), and was its main designer (Turing, 1986). The ACE was designed to be an electronic stored-program computer, with a pilot prototype that first ran in 1950. This pilot version was not a stored-program computer, but it was possibly the fastest computer in the world at that time. Its commercial version, the DEUCE (Digital Electronic Universal Computing Engine), was first installed in 1955 and many of them were in service until 1965. He also wrote one of the first programmer's handbooks, pointing out the very essential nature of the "Electronic Computer" as a fast, disciplined, and unintelligent task performer (Turing, 1948, 2012). In Manchester, Turing had the opportunity to contribute to the programming system of a stored-program computer, the MARK I.

Any discussion about Turing's contributions to science has to emphasize his work at Bletchley Park, where he took part in breaking the ENIGMA code used during the Second World War. With the help of spy-provided information about the ENIGMA machine, Turing lead the design of "La Bombe," an electromechanical engine that increased cryptanalysis speed and ultimately broke ENIGMA cyphering. La Bombe ran continuously until 1943, helping the Allies to win the war.

From his many scientific contributions, Turing is known as one of the founders of Computer Science. We cannot forget that the Turing machine was his way to formalize the informal concept of an effective procedure, a definition that he needed in order to show to the academic community his negative answer to the *Entscheidung* problem related to Hilbert's program. He started his career, undoubtedly, with a masterpiece. His PhD thesis, supervised by Alonso Church, was not a successful piece of knowledge if compared to what he did before. However, he started to drive his mind toward a possible way of overcoming the rather intrinsic limits of his own theory and Gödel's incompleteness theorems, and he obviously did not overcome them. With his introduction of oracles (Turing, 1938), he inaugurated the path toward what nowadays is known as hypercomputation, i.e., to consider formalisms and machines incorporating pieces that are able to perform noncomputable tasks.

According to Andrew Hodges (2012), his main biographer, Turing's creations were not only driven by his curiosity about the functioning of the human brain. Turing seemed to wish the functional replication of a healthy human brain. He searched to integrate all the mathematical and scientific knowledge available at his time to overcome every technical obstacle to see how far one can go without having to abandon a mechanicist view of Nature. From 1941, Turing must have had discussions on mechanization of many "intelligent" tasks with his colleagues. In 1945, he moved his mind to a completely radical view that the human mind itself could be embraced by the operations of his Turing machine. What is nowadays known as the Church–Turing thesis has more profound readings. Robin Gandy, former PhD student and thereafter colleague of Turing, has an argument showing that a particular class of physical processes are essentially Turing representable (Gandy, 1980). How far one can extend this class is on the domain of the "Physical Church–Turing thesis." Nowadays there are works considering quantum physical systems, or mechanisms, to extend Gandy's proof (Arrighi and Dowek, 2011). Turing himself, when he published the Turing test (Turing, 1950), was delivering a different kind of extension of the Church–Turing thesis regarding the mental abilities of human beings.

During the last years of his life, Alan Turing dedicated his thoughts to the mathematical view of Intelligence and Life (Turing, 1950, 1952). One of his last works was on mathematical models of life. This might be one of the first serious works on mathematical biology, as it is known today. Once again, his mind avoided being driven by the most popular scientific problems of his day.

Sometimes, particularly in the formal sciences, some mental constructions and arguments are as important as the conclusions they make possible. Turing's first impacting result is a mathematical construction so nice that it has inspired some theorems on the theory of computation and metamathematics. Turing showed how to write down a logical first-order formula from every Turing machine, such that this formula is satisfiable if, and only if, the respective Turing machine halts. The inspiration for the Trakhtenbrot (1950) theorem on the undecidability of the finite first-order logic validity may be seen as a consequence of Turing's proof, by observing that the models for formulas related to halting programs are always finite. Cook's reduction of propositional logic satisfiability to polynomially bounded nondeterministic Turing machines computations might have been inspired also by the original Turing technique.

Turing died younger than most of his contemporaries, as a consequence of the intolerance of the society in which he lived. Because of his homosexuality, he was criminally prosecuted in 1952. He was sentenced to either receive estrogen injections or go to jail. His security clearance was withdrawn. Alan Turing was found dead at his flat in Wilmslow (UK), on June 7, 1954. His death was officially ruled as a suicide, committed by eating a cyanide-poisoned apple.

For those born after the 1980s, it is hard to imagine a lifestyle without the computer, which is ubiquitous in our lives. Even the most intolerant human being has to acknowledge Alan Turing for building the bridge to where no one dreamed to go before. For operational researchers and optimizers, we must say that there would hardly be any developments, applications, or applied work in these areas without the modern computers that provide effective problem solving methods, developed on the solid grounds established by Turing's ideas.

The computer science community is not only celebrating the centenary of Turing's birth in 2012. This is also a celebration of the origins of computers and computer science. There is a lot to celebrate, from the mathematical proof of the existence of a programming machine to the ACE design, passing by the project of "La Bombe." This worldwide celebration of Turing's achievements can be followed at the official site http://www.turingcentenary.eu/.

References

Arrighi, P., Dowek, G., 2011. The physical Church-Turing thesis and the principles of quantum theory. *CoRR*, abs/1102.1612.

Church, A., 1937. Review of "On computable...". *Journal of Symbolic Logic* 2, 42–43.

Gandy, R., 1980. Church's thesis and principles for mechanisms. In Keisler, H.J., Barwise, J., and Kunen, K. (eds) *The Kleene Symposium*, Vol. 101. Elsevier, Amsterdam pp. 123–148.

Gödel, K., 1965. Gödel's postscriptum to his lecture notes of 1934. In Davis, M. (ed.) *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions*, Dover Publishing, Mineola, NY, pp. 71–73.

Grimm, K., 2003. Software technology in an automotive company: Major challenges. Proceedings of the 25th International Conference on Software Engineering, May 2003, IEEE Computer Society, Washington, pp. 498–503.

Hodges, A., 2012. Alan Turing. Available at http://www.turing.org.uk/ (accessed 7 February, 2012).

Trakhtenbrot, B.A., 1950. The impossibility of an algorithm for the decision problem for finite domains. *Doklady Akademii Nauk SSSR* 70, 569–572.

Turing, A.M., 1936–1937. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* 42, 230–265.

Turing, A.M.,1938. Systems of logic defined by ordinals. PhD thesis, Princeton University.

Turing, A.M., 1948. Intelligent machinery. Technical Report, National Physical Laboratory, London.

Turing, A.M., 1950. Computing machinery and intelligence. *Mind* 59, 433–460.

Turing, A.M., 1952. The chemical basis of morphogenesis. *Philosophical Transactions of the British Royal Society* 237, 37–72.

Turing, A.M., 1986. Proposal for development in the Mathematics Division of an automatic computing engine (ACE). In Carpenter, B.E. and Doran, R.W. (eds), *A.M. Turing's ACE Report of 1946 and Other Papers*, MIT Press, Cambridge, MA, pp. 20–105.

Turing, A.M., 2012. Programmers' handbook for Manchester electronic computer. Technical Report, University of Manchester Computing Laboratory, 1950. Available at http://www.AlanTuring.net/programmers_handbook (accessed 6 February, 2012).