Alan Turing and the Turing Award Winners

A Short Journey Through the History of Computer Science

Luis Lamb, 22 June 2012

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Slides by Luis C. Lamb

.inf
INSTITUTO DE INFORMÁTICA
UFRGS

Alan Mathison Turing

A.M. Turing, 1951

Turing by Stephen Kettle, 2007

Assumptions

- I assume knowlege of Computing as a Science.
- I shall not talk about computing before Turing: Leibniz, Babbage, Boole, Gödel...
- I shall not detail theorems or algorithms.
- I shall apologize for omissions at the end of this presentation.
- Comprehensive information about Turing can be found at http://www.mathcomp.leeds.ac.uk/turing2012/
- The full version of this talk is available upon request.

Alan Mathison Turing

- Born 23 June 1912: 2 Warrington Crescent, Maida Vale, London W9

Google maps

Alan Mathison Turing: short biography

- 1922: Attends Hazlehurst Preparatory School
- ’26: Sherborne School Dorset

- ’31: King’s College Cambridge, Maths (graduates in ‘34).
- ’35: Elected to Fellowship of King’s College Cambridge
- ’36: Publishes *“On Computable Numbers, with an Application to the Entscheindungsproblem”*, Journal of the London Math. Soc.
- ’38: PhD Princeton (viva on 21 June) : “Systems of Logic Based on Ordinals”, supervised by Alonzo Church.
- Letter to Philipp Hall: *“I hope Hitler will not have invaded England before I come back.”*

- ’39 Joins Bletchley Park: designs the “Bombe”.
- ’40: First Bombes are fully operational
- ’41: Breaks the German Naval Enigma.
- ’42-44: Several contibutions to war effort on codebreaking; secure speech devices; computing.
- ’45: Automatic Computing Engine (ACE) Computer.

Bletchley Park Snapshots

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

.inf
UFRGS

Alan Mathison Turing: short biography

- 1945: ***Officer*** *of the Most Excellent Order of the* ***British Empire*** (OBE)
- ’46-47: Lectures on Computer Design, London.
- ’47: Makes reference to Computer intelligence.
- ’47: Describes neural computation.
- 23 Aug. 47: Records a Marathon time of 2h46min3s.
- ’48: Appointed Reader in Mathematics, U of Manchester.
- ’49: Preliminary ideas on program verification.

- ’50: “Computing Machinery and Intelligence”, *Mind, Oct. 1950: Turing Test.*

- ’51: FRS (Fellow of the Royal Society)
- ’52: “The Chemical Basis of Morphogenesis”, Philosophical Transactions of the Royal Society (see Newton, Darwin...).

- ’52: Convicted for gross indecency.
- 7 June 1954: Dies from cyanide poisoning.

Alan Turing: Short Biography

1946

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

A.M. Turing's key contributions

- Investigated Fundamental problems in Logic.

- Turing solved Hilbert's famous "decision problem: Entscheidungsproblem".
- Defined the notion of "computable numbers" and of "computability".
- Defined an abstract computing machine, now referred to as "the universal Turing Machine".
- Turing referred to it as the "*a*(utomatic)-machine" in [0].

- Computable number: "...may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means" [0].

The Turing Machine

- The Turing machine is provided with an (infinite) tape divided into squares.
- It reads/writes one square at a time.
- It reads, writes or erases symbols according to an algorithm.
- The scanner may change its mechanical state and can remember previously read symbols.
- *"Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In §8 I examine certain arguments which would seem to prove the contrary. [...] In particular, it is shown (§11) that the Hilbertian Entscheidungsproblem can have no solution."[0]*

A.M. Turing's key contributions

- Computers are based on Turing Machines: von Neumann constructed his first models based on Turing's [1].

- Julian Bigelow (von Neumann's chief engineer, 1971 interview) [3]:

- ***"Turing's [universal] machine does not sound much like a modern computer today, but nevertheless it was. It was the germinal idea... So... [von Neumann] saw... that [ENIAC] was just the first step, and that great improvements would come".***

A.M. Turing's key contributions: WWII

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

.inf
UFRGS

A.M. Turing's key contributions: WWII

Slides by Luis C. Lamb

A.M. Turing's key contributions: WWII

Slides by Luis C. Lamb

A.M. Turing's key contributions

- Church to Kleene letter (1935): Gödel "... regarded thoroughly unsatisfactory" Church's proposal to use lambda-definability as a definition of effective calculability... It seems that only after Turing's formulation appeared did Gödel accept Church's thesis. S.C. Kleene [2].

- Recall that: "computer" in 1935 was a human being (also "computor").

- "Computing machine": small calculating machines.
- Later: electronic/digital computer, and, finally, "computer".

A.M. Turing's key contributions

- Created both Computer Science and
- Artificial Intelligence including: Machine Learning, Neural (supervised) Learning, Genetic Algorithms (genetical or evolutionary search).

- Influenced biological resarch (pattern and form in living organisms), philosophical logic and the philosophy of mind, cognitive science.

- Turing was a brilliant cryptologist: Sir Harry Hinsley (1993, official historian of the British Secret Service):

***The work of Turing and his colleagues shortened the war in at least two years* [3].**

A.M. Turing's key contributions

- **Sydney Brenner (Nobel Prize Winner 2002)**: *"Biological research is in crisis, and in Alan Turing's work there is much to guide us". **Nature, 461, Feb. 2012.***[4]

- *"Three of Turing's papers are relevant to biology".*

- *"The most interesting connection with biology, in my view, is Turing's most important paper: 'On computable numbers with an application to the Entscheindungsproblem'.*

- *"Biologists ask only three questions of a living organism: how does it work? How it is built? And how did it get that way? They are problems embodied in the classical fields of physiology, embryology and evolution. And **at the core of everything are the tapes containing the descriptions to build these special Turing machines."***

The A.M. Turing Award

- Awarded yearly by the Association for Computing Machinery (the world's premier Computer Science Society). http:www.acm.org

- First awarded in 1966, "is given for major contributions of lasting importance to computing."

- *The A.M. Turing Award, sometimes referred to as the "Nobel Prize" of Computing, was named in honor of Alan Mathison Turing (1912–1954), a British mathematician and computer scientist. He made fundamental advances in computer architecture, algorithms, formalization of computing, and artificial intelligence. Turing was also instrumental in British code-breaking work during World War II.The award includes a $250,000 reward prize.*

Evolution of Computer Science: from 1950

- First computers were hard to programme: machine language is highly non-intuitive and error-prone, except for Turing or von Neumann.
- There was a huge demand for fast “calculating” machines, specially in the defense industry.
- Turing, von Neumann, Wilkes, Wilkinson, Flowers, Bigelow, Eckert, Mauchly: directly (or indirectly) involved in defense projects.
- Much of Turing’s work was classified until 1970’s or even 2012! (“Report on the appliction of probability to cryptography”and “Paper on statistics of repetitions”).
- The above requirements/facts influenced computing research.

How computer science evolved?

- The first computers were very hard to programme.
- This led to fundamental research on programming languages, techniques and compiler design
- In the 1950s, FORTRAN was developed at IBM; von Neumann did not see the need for high level languages.
- In the 1960s, a US-European effort led the development of ALGOL, probably the most influential computer language in history.
- At the same time, declarative languages such as LISP were also developed. They targeted AI applications.

The Early Winners of the Turing Award

- The first Turing Awards were given to:

1966 (Alan Perlis, Carnegie Mellon U.): programming/compiling techniques.

1967 (Maurice Wilkes, U. Cambridge): the implementation of a computer with internally stored programs; program libraries; fully operational stored-program computer (EDSAC: Electronic Delay Storage Automatic Computer).

1968: (Richard Hamming, Bell Labs): numerical methods, coding systems, error-detecting/correcting techniques.

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

Evolution of Computer Science: 1960s

- Early awards: AI and Numerical Analysis.
- 1969: Marvin Minsky (MIT): For advancing the field of Artificial Intelligence: neural networks, automata theory, symbolic mathematics.
- 1970: James Wilkinson (National Physical Lab, England): numerical analysis to facilitiate the use of the high-speed digital computer, [...] recognition for his work in computations in linear algebra and "backward" error analysis.
- 1971: John McCarthy (Stanford): Lisp, AI, Theory.

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

Early Programming Languages Research: 1960s

ALGOL greatly influenced other languages; it was the standard for algorithm description used in academic works for decades.

-- Backus, Naur, Perlis, Dijkstra, Hoare, Wirth all worked on ALGOL.

Alan Perlis: *"The meetings were exhausting, interminable, and exhilarating. One became aggravated when one's good ideas were discarded along with the bad ones of others. Nevertheless, diligence persisted during the entire period. The chemistry of the 13 was excellent.*" ALGOL meeting, early 1960.

FORTRAN is an imperative programming language used in scientific computing.
-- John Backus (1978 Turing award winner, led the FORTRAN team at IBM, in the 1950s).
-- Later, Backus defended declarative languages.

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

Early Programming/Algorithms Research

1972: Edsger **Dijkstra** (Eindhoven U, NL): ALGOL; science of programming; graph algorithms, distributed computing.

1974: Donald E. **Knuth** (Stanford): analysis of algorithms; design of computer languages; "The Art of Computer Programming": the most famous/influential set of scientific books in Computing.

1977: John **Backus** (IBM): Design of programming languages.

1978: Robert **Floyd** (Stanford): reliable software; theory of parsing, semantics of programming languages; verification and synthesis; analysis of algorithms.

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

1975: Alan Newell and Herbert Simon

- Numerous contributions to AI, problem solving, psychology of cognition; decision making, list processing before LISP.
- **Simon is the only person to have won both the Turing Award and the Nobel Prize in Economic Sciences**.

- AI programs (Newell, Simon, Shaw): the Logic Theorist (1956: proved 38 theorems of Whitehead/Russell Principia Mathematica).
- General Problem Solver (1957): theorems, geometric problems and chess playing. Separated knowledge from strategy.

http://diva.library.cmu.edu/Newell/biography.html

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

Programming as a Science Pioneers

- 1977: Kenneth **Iverson** (IBM): programming languages and mathematical notation, APL; programming language theory and practice. Financial and actuarial applications.

- 1978: **Sir Tony Hoare** (Oxford): *Quicksort*; fundamental contributions to the definition and design of programming languages; axiomatic semantics; CSP.

- 1984 Niklaus **Wirth** (ETH): Euler, ALGOL-W, MODULA, PASCAL.

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

Early Theory of Computing Winners

- 1976: Michael **Rabin** (Harvard, Hebrew U.), Dana **Scott** (Oxford): "*Finite Automata and Their Decision Problems*"; both supervised by Church. Non-determinism.

- 1982: Stephen **Cook** (Toronto): *"The Complexity of Theorem Proving Procedures"*. SATtisfiability of propositional formulae is NP-complete.

- 1991: Robin **Milner** (Edinburgh): implemented the Logic of Computable Functions; ML language; Calculus of Communicating Processes; full abstraction.

.inf UFRGS

Early Theory of Computing Winners

- 1985: Richard **Karp** (Berkeley): Proposed the stardard way for proving NP-completeness.

- 1986: John **Hopcroft** (Cornell), Robert **Tarjan** (AT&T Bell): algorithms; data structures. Hopcroft is co-author of: *"Introduction to Automata Theory Languages and Computation".*

- 1993: Juris **Hartmanis** (Cornell), Richard **Stearns** (SUNY): how much time and memory are needed to perform different computations? They named the field *computational complexity.*

.inf UFRGS

Theory and Cryptography Winners

- 1995 Manuel **Blum** (Berkeley): computational complexity theory; application to cryptography and program checking. Creator of CAPTCHA (with von Ahn).
- 2000 Andrew Chi-Chih **Yao** (Princeton): pseudorandom number generation, cryptography, communication complexity.
- 2002 Ron **Rivest** (MIT), Adi **Shamir** (Weizmann Inst. of Science), Leonard **Adleman** (USC): for making public-key cryptography useful in practice: the **RSA** algorithm.

Computer Systems Winners

- 1983: **Ken Thompson, Dennis Ritchie** (Bell Labs): operating systems theory; implementation of UNIX (on a spare DEC minicomputer).

- 1987: **John Cocke** (IBM): design and theory of compilers; architecture of large systems; reduced instruction set computers (RISC, IBM 801).

- 1990: **Fernando Corbató** (MIT): time-sharing and resource-sharing computer systems, CTSS (Compatible Time Sharing System) for IBM 7090/94 and Multics.

.inf UFRGS

The IBM 801 Memo

Name | Internal Zip | City & State 1976 801

H.G.K., Jan 19, 77

To: File

IBM

IBM CONFIDENTIAL

From Date: October 8, 1976
Name & Tie/Ext.: George Radin/862-1682
Title/Dept. Name: Mgr./801 Experimental Minicomputer Project
Internal Zip/City, State: Computer Sciences Department
or U.S. mail address: RES - Yorktown Heights, N. Y.

Subject: 801 Minicomputer Project

Reference:

The 801 minicomputer project in Research attempts to achieve large factors of improvement in job cost/performance over any existing or planned system, independent of technology advances. It relies solely on innovations in machine design, architecture, operating system and compilers, using out-of-catalogue hardware components and standard IBM I/O devices.

The attached paper is designed to explain these innovations and thus to help readers of the architecture and specification documents understand why some of the seemingly strange features were included. It can also stand alone as an overview of the first phase of the project.

.inf UFRGS

Modern Programming Languages

- 2006: **Frances Allen** (IBM): optimizing compiler techniques; automatic parallel execution. *"A Catalog of Optimizing Transformations,":* first systematization of optimizing transformations.

- 2008: **Barbara Liskov** (MIT): programming language and system desig; data abstraction, fault tolerance, and distributed computing; developed CLU (inflenced C++, C# and Python).

.inf UFRGS

Modern Programming Languages

- 2001: **Ole-Johan Dahl** (U. Oslo) and **Kristen Nygaard** (Oslo, Aahrus): object-oriented programming; discrete event simulation language Simula I and Simula 67 (inspired Smalltalk).

-- modularity, objects, classes, inheritance with virtual quantities.
-- Software: built in layers of abstraction.

- 2005: **Peter Naur** (Copenhagen): programming languages; ALGOL 60, compiler design.

-- Programmed the EDSAC at Cambridge.
-- ALGOL 60 Report.
-- Opposed to Dijkstra and Wirth's structured programming.
-- "Programming as Theory Building": starting point the Agile Software Development.

.inf UFRGS

Database research winners

- 1973: Charles **Bachman** (Honeywell), created the Integrated Data Store (IDS); first builders of DBMS

- 1981: Edward **"Tedd" Codd** (IBM): relational data model: based on First-order Logic. IBM refused to implement the relational model. **Virtually all databases today are based on his model,** including Oracle's which was called "Relational Software, Inc."... and the rest is history...

- 1998: **Jim Gray** (Microsoft): contributions to database and transaction processing research and technical leadership in system implementation. Key figure in IBM System R.

.inf UFRGS

Interactive/Personal Computing and Graphics

- 1988: Ivan **Sutherland** (Sutherland, Sproull and Associates): first computer graphical user interface (GUI); visionary contributions to computer graphics: Sketchpad.

- 1997 Douglas **Engelbart** (Bootstrap Institute & Bootstrap Alliance (now DEI)): vision of the future of interactive computing technological innovations include the mouse, hypertext, and the split screen interface.

- 2009 Charles **Thacker** (Microsoft): first modern PC: the Alto at Xerox PARC; Ethernet, multiprocessor workstations, and tablet personal computers.

.inf UFRGS

The first workstation: Xerox Alto

.inf UFRGS

Sketchpad: 1963

-- Sketchpad: A Man-Machine Graphical Communication System
-- Graphic User Interface;
-- object oriented programming;
-- Ivan Sutherland demonstrated that computer graphics could be used for both artistic and technical purposes; novel method of human-computer interaction.

Software Engineering, Architecture.

1989: William **Kahan** (Berkeley): IEEE 754-1985 standard for floating-point computation.

1999: Fred **Brooks** (UNC): software engineering, computer architecture, operating systems. Coined the term “computer architecture”. Created OS/360, led IBM 360 project (compatibility).

"The most important single decision I ever made was to change the IBM 360 series from a 6-bit byte to an 8-bit byte, thereby enabling the use of lowercase letters. That change propagated everywhere.

Computer Systems, Networks

1992 Butler **Lampson** (Parc-DEC-Microsoft): Xerox 9700 laser printer, first what-you-see-is-what-you-get (WYSIWYG) text editor, ALTO operating system, tablet PC.

2003 Alan **Kay** (PARC-Disney-Viewpoints): object-oriented programming and window-based GUI design; Smalltalk; ALTO (1974): Influenced by Engelbart.
-- Key ideas from PARC were adopted by Apple Macintosh (1984).

2004: **Vint Cerf** (MCI), Robert **Kahn** (CNRI): Internet's basic communications protocols, TCP/IP; leadership in networking; both founded the ISOC (Internet Society).

Software Engineering, Verification

- 1996: Amir **Pnueli** (Weizmann): temporal logic in computer science; model checking.

- 2007: Edmund **Clarke** (CMU), E. Allen **Emerson** (UT Austin), Joseph **Sifakis** (Verimag): for developing Model-Checking into a highly effective verification technology.

- Model checking: now widely used to verify designs for integrated circuits, computer networks, and software (e.g. Airbus, Intel and Microsoft).

Recent AI Winners: Back to Turing?

- 1994: Edward **Feigenbaum** (Stanford) and Raj **Reddy** (CMU): large scale, commercial AI systems, expert systems, robotics.

- 2010: Leslie **Valiant** (Harvard): PAC learning; computational complexity; parallel/distributed computing;

- **2011: Judea Pearl** (UCLA): probabilistic/Bayesian reasoning and learning in AI.

.inf UFRGS

Turing Award by Research Field I

- Artificial Intelligence: 1969 (Minsky), 1971 (McCarthy), 1975 (Newell, Simon), 1994 (Feigenbaum and Reddy), 2010 (Valiant), 2011 (Pearl).

- Computer Architecture/Operating Systems/Computer Systems: 1967 (Wilkes), 1983 (Thompson, Ritchie), 1987 (Cocke), 1990 (Corbató), 1992 (Lampson), 1999 (Brooks), 2009 (Thacker).

- Numerical Methods/Analysis: 1968 (Hamming), 1970 (Wilkinson), 1989 (Kahan).

- Programming Languages/Programming Techniques/Compilers: 1966 (Perlis), 1967 (Wilkes), 1972 (Dijkstra), 1974 (Knuth), 1977 (Backus), 1978 (Floyd), 1979 (Iverson), 1980 (Hoare), 1984 (Wirth), 1987 (Cocke), 1991 (Milner), 2001 (Dahl and Nygaard), 2003 (Kay), 2005 (Naur), 2006 (Allen), 2008 (Liskov).

- Cryptology: 1995 (Blum), 2000 (Yao), 2002 (Rivest, Shamir, Adleman)

- Databases: 1973 (Bachman), 1981 (Codd), 1998 (Gray).

Turing Award by research field II

- Theory: 1972 (Dijkstra), 1976 (Rabin, Scott), 1982 (Cook), 1985 (Karp), 1986 (Hopcroft, Tarjan), 1991 (Milner), 1993 (Hartmanis, Sterns), 1995 (Blum), 1996 (Pnueli), 2000 (Yao), 2007 (Clarke, Emerson, SIfakis), 2010 (Valiant)

- Computational Complexity/Algorithms: 1974 (Knuth), 1982 (Cook), 1985 (Karp), 1986 (Hopcroft, Tarjan), 1995 (Blum), 2000 (Yao), 2010 (Valiant).

- Interactive Computing/Graphics: 1979 (Iverson), 1988 (Sutherland), 1997 (Engelbart), 2009 (Thacker).

- Software Engineering/Verification: 1978 (Floyd), 1980 (Hoare), 1991 (Milner), 1996 (Pnueli), 1999 (Brooks), 2005 (Naur), 2007 (Clarke, Emerson, Sifakis).

- Distributed Computing/Networks: 1972 (Dijkstra), 1983 (Thon Ritchie), 1990 (Corbató), 1991 (Milner), 1992 (Lampson), 2004 Kahn), 2008 (Liskov), 2010 (Valiant)

Turing Awards by Research Field III

Analysis of Algorithms **Artificial Intelligence** Combinatorial Algorithms Compilers
Computational Complexity Computer Architecture Computer Hardware **Cryptography**
Data Structures Databases Education Error Correcting Codes Finite Automata Graphics
Interactive Computing Internet Communications List Processing Numerical Analysis
Numerical Methods Object Oriented Programming Operating Systems Personal Computing
Program Verification Programming
Programming Languages Proof Construction Software **Theory**
Software Engineering Verification of Hardware and Software Models Computer Systems

http://amturing.acm.org/bysubject.cfm?cat=18

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

Afterthoughts

Theoretical Computer Science
Logic, complexity: What do they compute? How do they do it? What they need to compute?

Artificial Intelligence: Can computers think, learn, act, autonomously?

Interactive Computing: How humans interact with computers?

Programming Languages/Software Engineering: How can we safely and easily program them?

Computer Systems: How can one use science to built the state-of-the-art computing technologies?

Afterthoughts

"We can only see a short distance ahead, but we can see plenty there that needs to be done." Alan M. Turing in *"Computing Machinery and Intelligence"*, Mind,1950.

"Our field is still in its embryonic stage. It's great that we haven't been around for 2000 years. We are at a stage where very, very important results occur in front of our eyes." Michael O. Rabin, 1995 in [5].

Bibliography

[0] Alan M. Turing. On Computable Numbers, with an Application to the Entscheindungsproblem. Proc. of the London Mathematical Society. ser. 2, vol. 42, pp. 230-265. (1936). Corrections vol. 43 pp. 544-46, (1937).
[1] B. J. Copeland (ed). The Essential Turing: The ideas that gave birth to the computer age. Oxford Univ. Press (2004).
[2] S.C. Kleene. Origins of Recursive Function Theory. Annals of the History of Computing, 3: 52-67 (1981).
[3] H. Hinsley; A. Stripp. Codebreakers: The Inside Story of Bletchley Park. Oxford Univ. Press (1993).
[4] Sydney Brenner. Life's Code Script. Nature Vol 482 (7386), pp. 461, (2012)
[5] Denis Shasha and Cathy Lazere: Out of Their Minds: The Lives and Discoveries of 15 Great Computer Scientists. Copernicus, 1995
[6] Andrew Hodges. Alan Turing: The Enigma. 1983, Vintage 1992.
[7] Martin Davis (ed). The Undecidable. Raven Press 1965, Dover 2004.
[8] R. Rojas, U. Hashagen (eds). The First Computers: History and Architecutres. MIT Press 2000.
[9] A.M. Turing. Computing Machinery and Intelligence. Mind 49:433-460, 1950.

[10] Photographs of this talk: http://www.acm.org; http://www.computerhistory.org; Wikipedia entries on Turing Award winners and computing history; Bletchley Park pictures by the author.