Alan Turing and the Turing Award Winners

A Short Journey Through the History of Computer Science

Luis Lamb, 22 June 2012
Alan Mathison Turing

A.M. Turing, 1951

Turing by Stephen Kettle, 2007

Slides by Luis C. Lamb
Assumptions

• I assume knowledge 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
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
- '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 contributions to war effort on codebreaking; secure speech devices; computing.
Alan Mathison Turing: short biography

- 1945: **Officer of the Most Excellent Order of the British Empire** (OBE)
- ’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.


- ’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.
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
A.M. Turing’s key contributions: WWII
A.M. Turing’s key contributions: WWII

Slides by Luis C. Lamb
- 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.

Slides by Luis C. Lamb
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 research (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].
• 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 application 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:
  
  
  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.
Evolution of Computer Science: 1960s

- Early awards: AI and Numerical Analysis.


- 1970: James Wilkinson (National Physical Lab, England): numerical analysis to facilitate the use of the high-speed digital computer, [...] recognition for his work in computations in linear algebra and "backward" error analysis.

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.
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.


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

- General Problem Solver (1957): theorems, geometric problems and chess playing. Separated knowledge from strategy.

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

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.
Early Theory of Computing Winners


• 1991: Robin Milner (Edinburgh): implemented the Logic of Computable Functions; ML language; Calculus of Communicating Processes; full abstraction.
Early Theory of Computing Winners


- 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.
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.

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.
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.
Modern Programming Languages


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

- **2001**: **Ole-Johan Dahl** (U. Oslo) and **Kristen Nygaard** (Oslo, Aarhus): 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.
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.
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.
The first workstation: Xerox Alto
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.

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."
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.

-- 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


- 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.
Turing Award by Research Field I


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


Slides by Luis C. Lamb
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)

Slides by Luis C. Lamb
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 build the state-of-the-art computing technologies?
“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


[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.