Technical Report No. 2006-511
SOME QUOTES OF INTEREST*

Selim G. Akl
School of Computing
Queen's University
Kingston, Ontario, Canada K7L 3N6

April 3, 2006

The quotes on pages 2 to 12 of this document, gleaned from the computer science, logic, physics, psychology, philosophy of the mind, and general science literature, provide clear, unambiguous, and unqualified definitions of the Church-Turing thesis, the principle of simulation, and universal computation. They were collected on February 28, 2005. They should be useful to those intending to read:

S.G. Akl, The myth of universal computation, Technical Report No. 2005-492, School of Computing, Queen's University, Kingston, Ontario, Canada.
http://www.cs.queensu.ca/Parallel/technical.html#Technical

Also available as:

Akl, S.G., The myth of universal computation, in: Parallel Numerics, Trobec, R., Zinterhof, P., Vajteršic, M., and Uhl, A., Eds., Part 2, Systems and Simulation, University of Salzburg, Austria and Jožef Stefan Institute, Ljubljana, Slovenia, 2005, pp. 211–236.
http://pluton.ijs.si/ marjan/parnum05/

In summary, what the statements on pages 2 to 12 are saying is the following: It is possible to build, *once and for all*, a certain device–the **Universal Computer**–henceforth capable of performing *any* computation that *any* other physically realizable device can perform. This claim is *disproved* in Technical Report No. 2005-492: **No finite and fixed device can be universal**.

Please also see Technical Report No. 2006-508 for a generalization of this result:

S.G. Akl, Even accelerating machines are not universal, Technical Report No. 2006-508, School of Computing, Queen's University, Kingston, Ontario, Canada.
http://www.cs.queensu.ca/Parallel/technical.html#Technical

*This research was supported by the Natural Sciences and Engineering Research Council of Canada.

****

"[The] 'computable numbers' include all numbers which would naturally be regarded as computable."

A.M. Turing, On computable numbers with an application to the Entscheidungsproblem, *Proceedings of the London mathematical Society*, Ser. 2, Vol. 42, 1936, p. 249.
****

****

"[Logical computing machines] can do anything that could be described as 'rule of thumb' or 'purely mechanical'."

A.M. Turing, Intelligent machinery, Report, National Physics Laboratory, 1948. Reprinted in: B. Meltzer and D. Michie, Eds., *Machine Intelligence* 5, Edinburgh University Press, Edinburgh, Scotland, 1969, p. 7.
****

****

"The reader will find it incredible, at first sight, that some of these sets of simple operations could give rise to the full range of possible computations. [...] As we will see, it is possible to execute the most elaborate possible computation procedures with Turing machines whose fixed structures contain only dozens of parts. [...] Accepting Turing's thesis, we conclude that the universal machine can simulate any effective process of symbol-manipulation, be it mathematical or anything else; it is a completely general instruction-obeying mechanism."

M.L. Minsky, *Computation: Finite and Infinite Machines*, Prentice-Hall, Englewood Cliffs, New Jersey, 1967, p. 112; p. 128; p. 145.
****

****

"It can also be shown that any computation that can be performed on a modern-day digital computer can be described by means of a Turing machine. Thus if one ever found a procedure that fitted the intuitive notions, but could not be described by means of a Turing machine, it would indeed be of an unusual nature since it could not possibly be programmed for any existing computer."

J.E. Hopcroft and J.D. Ullman, *Formal Languages and their Relations to Automata*, Addison-Wesley, Reading, Massachusetts, 1969, p. 80.
****

****

"[...] anything computable is Turing-machine computable."

D.C. Dennett, *Brainstorms: Philosophical Essays on Mind and Psychology*, Harvester, Brighton, 1978, p. 83.
****

****

"[...] Church's thesis: The computing power of the Turing machine represents a fundamental limit on the capability of realizable computing devices."

P.J. Denning, J.B. Dennis, and J.E. Qualitz, *Machines, Languages, and Computation*, Prentice-Hall, Englewood Cliffs, New Jersey, 1978, p. 2.

****

****

"[...] the Turing machine is equivalent in computing power to the digital computer as we know it today and also to all the most general mathematical notions of computation."

J. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, Massachusetts, 1979, p. 147.

****

****

"As an important part of the evidence for Church's thesis we show how the operations of the ordinary sort of high-speed digital computer can be simulated by a Turing machine [...]."

G.S. Boulos and R.C. Jeffrey, *Computability and Logic*, Cambridge University Press, Cambridge, 1980, p. 54.

****

****

"Although the elementary operations of the Turing machine are restricted, iterations of the operations enable the machine to carry out any well-defined computation on discrete symbols."

J.A. Fodor, The Mind-Body Problem, *Scientific American*, January 1981, p. 130.

****

****

"[...] as primitive as Turing machines seem to be, attempts to strengthen them seem not to have any effect. [...] Thus any computation that can be carried out on the fancier type of machine can actually be carried out on a Turing machine of the standard variety. [...] *any* way of formalizing the idea of a 'computational procedure' or an 'algorithm' is equivalent to the idea of a Turing Machine. [...]

It is theoretically possible, however, that Church's Thesis could be overthrown at some future date, if someone were to propose an alternative model of computation that was publicly acceptable as fulfilling the requirement of 'finite labor at each step' and yet was provably capable of carrying out computations that cannot be carried out by any Turing machine. No one considers this likely. "

H.R. Lewis and C.H. Papadimitriou, *Elements of the Theory of Computation*, Prentice Hall, Englewood Cliffs, New Jersey, 1981, p. 168-169; p. 223.

****

****

"[...] for every algorithm there is a mu-recursive function over the non-negative integers whose computation effectively does the work of the algorithm."

R. McNaughton, *Elementary Computability, Formal Languages, and Automata*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982, p. 148.

****

****

"[...] any algorithm for computing on numbers can be carried out by a program of [the programming language] S."

M.D. Davis and E.J. Weyuker, *Computability, Complexity, and Languages*, Academic Press, New York, 1983, p. 54.

****

****

"Church maintains that any algorithmic procedure, described informally, can be formalized as a Turing Machine."

G.J Tourlakis, *Computability*, Reston, Reston, Virginia, 1984, p. 280.

****

****

"At its logical base every digital computer embodies one of these paper and pencil devices invented by the British mathematician A.M. Turing. The machines mark off the limit of computability."

J.E. Hopcroft, Turing Machines, *Scientific American*, Vol. 250, No. 5, May 1984, p. 86.

****

****

"Its significance for the theory of computing is fundamental: given a large but finite amount of time, the Turing machine is capable of any computation that can be done by any modern digital computer, no matter how powerful."

J.E. Hopcroft, Turing Machines, *Scientific American*, Vol. 250, No. 5, May 1984, p. 86.

****

****

"I can now state the physical version of the Church-Turing principle:'Every finitely realizable physical system can be perfectly simulated by a universal model computing machine operating by finite means'. This formulation is both better defined and more physical than Turing's own way of expressing it."

D. Deutsch, Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer, *Proceedings of the Royal Society*, Series A, 400, 1985, p. 99.

****

****

"It is believed that there are no functions that can be defined by humans, whose calculation can be described by any well defined algorithm that people can be taught to perform, that cannot be computed by Turing machines."

D.I.A. Cohen, *Introduction to Computer Theory*, John Wiley, New York, 1986, p. 790.

****

****

"Turing showed that his very simple machine [...] can specify the steps required for the solution of any problem that can be solved by instructions, explicitly stated rules, or procedures."

R.L. Gregory, *The Oxford Companion of the Mind*, Oxford University Press, Oxford, 1987, p. 784.

****

****

"[...] Church's thesis [...] implies that our intuitive notions of algorithm are captured by the computable functions.

D. Wood, *Theory of Computation*, Harper and Row, New York, 1987, p. 359.

****

****

"[...] if we have shown that a problem can (or cannot) be solved by any TM, we can deduce that the same problem can (or cannot) be solved by existing mathematical computation model nor by any conceivable computing mechanism. The lesson is: Do not try to solve mechanically what cannot be solved by TMs!"

D. Mandrioli and C. Ghezzi, *Theoretical Foundations of Computer Science*, John Wiley, New York, 1987, p. 152.

****

****

"The interesting thing about a universal Turing machine is that, for any well-defined computational procedure whatever, a universal Turing machine is capable of simulating a machine that will execute those procedures. It does this by reproducing exactly the input/output behaviour of the machine being simulated."

P.M. Churchland, *Matter and Consciousness*, MIT Press, Cambridge, Massachusetts, 1988, p. 105.

****

****

"Once it is appreciated that one can make Turing machines which perform arithmetic or simple logical operations, it becomes easier to imagine how they can be made to perform more complicated tasks of an algorithmic nature. After one has played with such things for a while, one is easily reassured that a machine of this type can indeed be made to perform *any mechanical operation whatever*! Mathematically, it becomes reasonable to *define* a mechanical operation to be one that can be carried out by such a machine. The noun 'algorithm' and the adjectives 'computable', 'recursive', and 'effective' are all used by mathematicians to denote the mechanical operations that can be performed by theoretical machines of this type–the Turing machines. So long as a procedure is sufficiently clear-cut and mechanical, then it is reasonable to believe that a Turing machine can indeed be found to perform it."

R. Penrose, *The Emperor's New Mind*, Oxford University Press, New York, 1989.
****

****

"[Turing's] results entail something remarkable, namely that a standard digital computer, given only the right program, a large enough memory and sufficient time, can compute any rule-governed input-output function. That is, it can display any systematic pattern of responses to the environment whatsoever."

P.M. Churchland and P.S. Churchland, Could a Machine Think, *Scientific American*, January 1990, p. 26.
****

****

"Astonishingly, Turing was able to show that any procedure that can be computed at all can be computed by a Turing machine. [...] Despite their simple organization, Turing machines are, in principle, as powerful as any other mode of organizing computing systems."

K. Sterelny, *The Representational Theory of Mind*, Basil Blackwell, Oxford, 1990, p. 238.
****

****

"[...] by the Church Thesis [...] all functions intuitively computable algorithmically are computable by Turing Machines [...]."

P. Cousot, Models and logics for proving programs, in: J. van Leeuwen, Ed., *Handbook of Theoretical Computer Science*, Vol. B, Elsevier, Amsterdam, 1990, p. 924.
****

****

"The limits of Turing machines, according to the Church-Turing thesis, also describe the theoretical limits of all computers."

R.P. McArthur, *From Logic to Computing*, Wadsworth, Belmont, California, 1991, p. 401.

****

****

"Turing had proven–and this is probably his greatest contribution–that his Universal Turing machine can compute any function that any computer, with any architecture, can compute."

D.C. Dennett, *Consciousness Explained*, Little, Brown, Boston, 1991, p. 215.

****

****

"Every algorithm can be implemented by a Turing machine. [...] By their rendering the notion of algorithm precise and unambiguous, Turing machines afford researchers in the theory of computability with a valuable tool for determining whether given tasks can be successfully completed, or for proving that they cannot be completely solved by any physical machine [...] regardless of how much memory or computing time we may be willing to provide."

O. Omidvar, *Progress in Neural Networks*, Vol. 1, Ablex, Norwood, New Jersey, 1991, pp. 132–133.

****

****

"Turing's analysis of what is involved in computation [...] seems so general that it is hard to imagine some other method which falls outside the scope of his description [...] so [...] anything which can be computed can be computed by a Turing machine."

S. Abramsky, D.M. Gabbay, T.S.E. Maibaum, Eds., *Handbook of Logic in Computer Science*, Vol. 1, Clarendon Press Oxford, 1992, p. 123.

****

****

"[...] any algorithmic problem for which we can find an algorithm that can be programmed in some programming language, *any* language, running on some computer, *any* computer, even one that has not been built yet but *can* be built, and even one that will require unbounded amounts of time and memory space for ever-larger inputs, is also solvable by a Turing machine."

D. Harel, *Algorithmics: The Spirit of Computing*, Addison-Wesley, Reading, Massachusetts, 1992, p. 233.

****

****

“The Turing principle

(for physical computers simulating each other)

It is possible to build a universal computer: a machine that can be programmed to perform any computation that any other physical object can perform.”

D. Deutsch, *The Fabric of Reality*, Penguin Books, London, England, 1997, p. 134.
****

****

“The Turing principle

(for virtual-reality generators rendering each other)

It is possible to build a virtual-reality generator whose repertoire includes that of every other physically possible virtual-reality generator. .”

D. Deutsch, *The Fabric of Reality*, Penguin Books, London, England, 1997, p . 134.
****

****

“The Turing principle

It is possible to build a virtual-reality generator whose repertoire includes every physically possible environment.”

D. Deutsch, *The Fabric of Reality*, Penguin Books, London, England, 1997, p. 135.
****

****

“[...] even the lowliest of today’s home computers can be programmed to solve any problem, or render any environment, that our most powerful computers can, provided only that it is given additional memory, allowed to run for long enough, and given appropriate hardware for displaying its results.”

D. Deutsch, *The Fabric of Reality*, Penguin Books, London, England, 1997, p. 194.
****

****

“But I have already mentioned the significance of computational universality – the fact that a single physically possible computer can, given enough time and memory, perform any computation that any other physically possible computer can perform.”

D. Deutsch, *The Fabric of Reality*, Penguin Books, London, England, 1997, pp. 195–196.
****

****

"But the point of universality is that it should be possible to program a single machine, specified once and for all, to perform any possible computation, or render any physically possible environment."

D. Deutsch, *The Fabric of Reality*, Penguin Books, London, England, 1997, p. 210.

****

****

"[The Turing principle] is on par with the principle of conservation of energy and the other laws of thermodynamics: that is, it is a constraint that, to the best of our knowledge, all other theories conform to."

D. Deutsch, *The Fabric of Reality*, Penguin Books, London, England, 1997, p. 345.

****

****

"A universal virtual-reality generator is physically possible. Such a machine is able to render any physically possible environment, as well as certain hypothetical and abstract entities, to any desired accuracy."

D. Deutsch, *The Fabric of Reality*, Penguin Books, London, England, 1997, p. 348.

****

****

"The advantage of the Turing principle is that it is already, for reasons quite independent of cosmology, regarded as a fundamental principle of nature - admittedly not always in this strong form, but I have argued that the strong form is necessary if the principle is to be integrated into physics."

D. Deutsch, *The Fabric of Reality*, Penguin Books, London, England, 1997, p. 354.

****

****

"[...] the celebrated Church-Turing thesis: The Turing machine and the RAM (as well as any of a number of other models such as lambda calculus or partial recursive functions) are universal models of computation."

B.M. Moret, *The Theory of Computation*, Addison-Wesley, Reading, Massachusetts, 1998, p. 113.

****

****

"[A] Universal computer [...] is not a real device (although it could be) but a set of rules for manipulating strings of ones and zeros. It comes as something of a surprise to learn that only a small set of rules is needed to be able to carry out all possible computations. This result was dicovered by the British mathematician Allen [sic] Turing in 1936. In a real computer many rules, built out of a set of simple rules, are used to increase the speed with which the devide can operate. The reason a Turing machine is called universal is that it can be used to simulate any other computer, albeit rather slowly."

G.J. Milburn, *Schrödinger's Machines*, W.H. Freeman and Company, New York, 1997, p. 157.

****

****

"As far as we know, no device built in the physical universe can have any more computational power than a Turing machine. To put it more precisely, any computation that can be performed by any physical computing device can be performed by any universal computer, as long as the latter has sufficient time and memory."

D. Hillis, *The Pattern on the Stone*, Basic Books, New York, 1998, pp. 63–64.

****

****

"The classical theory of computation is essentially the theory of the universal Turing machine—the most popular mathematical model of computation. Its significance relies on the fact that, given a large but finite amount of time, the universal Turing machine is capable of any computation that can be done by any modern classical digital computer, no matter how powerful."

A. Ekert and C. Macchiavello, An overview of quantum computing, in: *Unconventional Models of Computation*, C.S. Calude, J. Casti, and M.J. Dinneen, Eds., Springer-Verlag, Singapore, 1998, p. 19.

****

****

"It turns out that extremely simple head-and-tape structures (i.e., with few states for the head and few symbols for the tape alphabet [...]) are already capable of doing, in this fashion, anything that can be done by more complex structures."

T. Toffoli, Non-conventional computers, in: *Encyclopedia of Electrical and Electronics Engineering*, J. Webster, Ed., Wiley & Sons, 1998.

****

****

"What is important is that on the basis of Turing's analysis of the notion of computation, it is possible to conclude that anything computable by any algorithmic process can be computed by a Turing machine. So if we can prove that some particular task cannot be accomplished by a Turing machine, we can conclude that no algorithmic process can accomplish that task."

M. Davis, *The Universal Computer*, W.W. Norton, New York, New York, 2000, p. 151.
****

****

"So the shocking thing about this 1936 paper is that first of all [Turing] comes up with the notion of a general-purpose or universal computer, with a machine that's flexible, that can do what any machine can do. One calculating machine that can do any calculation, which is, we now say, a general-purpose computer."

G.J. Chaitin, A century of controversy over the foundations of mathematics, in: *Finite vs Infinite*, C.S. Calude and G. Păun, Eds., Springer-Verlag, London, 2000, p. 87.
****

****

"To say that the Turing machine is a general model of computation is simply to say that any algorithmic procedure that can be carried out at all (by a human, a team of humans, or a computer) can be carried out by a TM. This statement [...] is usually referred to as the Church-Turing thesis."

J. Martin, *Introduction to Languages and the Theory of Computation*, Third Edition, McGraw-Hill, New York, 2003, p. 353.
****

****

"Although the Universal Turing Machine – as its name suggests – is universal, in the sense that it can simulate any specialized Turing machine (i.e., any computational task), the machine cannot be reconfigured and its architecture cannot be changed during operation."

C. Teuscher, Turing's connectionism, in: C. Teuscher, Ed., *Alan Turing: Life and Legacy of a Great Thinker*, Springer-Verlag, Berlin, 2004, pp. 506–507.
****

****

"Anything a real computer can compute, a Turing machine can also compute."

http://en.wikipedia.org/wiki/Turing_machine
****

****

“Universal computer: A computer that is capable of universal computation, which means that given a description of any other computer or program and some data, it can perfectly emulate this second computer or program. Strictly speaking, home PCs are not universal computers because they have only a finite amount of memory. However, in practice, this is usually ignored.”

http://www.daviddarling.info/encyclopedia/U/universal_computer.html
****

****

“Universal computation: Capable of computing anything that can in principle be computed; being equivalent in computing power to a Turing machine or the lambda calculus.”

http://www.daviddarling.info/encyclopedia/U/universal_computation.html
****