Technical Report No. 2006-511
UNIVERSALITY IN COMPUTATION:
SOME QUOTES OF INTEREST*

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

The quotes, beginning on page 2 of this document, and 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. The vast majority were collected on the afternoon of February 28, 2005 from books on the author’s bookshelf. More were added later. These quotes should be useful to those intending to read:

http://www.cs.queensu.ca/Parallel/technical.html#Technical

Also available as:

http://pluton.ijs.si/ marjan/parnum05/

In summary, the statements on the coming pages say, without exception or caveat, 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.”


“[Logical computing machines] can do anything that could be described as ‘rule of thumb’ or ‘purely mechanical’.”


“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.”


“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.”

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


“[...] 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.”


“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 [...]”


“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.”


“[..] 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. [...]”

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

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

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

“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.”

“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.”

“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.”
“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.”


“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.”


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


“[...] 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!”


“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.”

“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.”


“[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.”


“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.”


“[…] by the Church Thesis […] all functions intuitively computable algorithmically are computable by Turing Machines [...]”


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

“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.”

“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.”
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

“ [...] any algorithmic problem for which we can find an algorithm that can be pro-
grammed 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 re-
quire 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,

“A Turing machine can do everything that a real computer can do.”
M. Sipser, *Introduction to the Theory of Computation*, PWS Publishing Company, Boston,
Massachusetts, 1997, p. 125.
“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.”

“The Turing principle
(for virtual-reality generators rendering each other)
It is possible to build a virtual-reality generator whose repertoire includes every physically possible virtual-reality generator.”

“[...] 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.”

“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.”
“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.”

“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.”

“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.”

“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.”

“[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 discovered by the British mathematician Alan [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 device can operate. The reason a Turing machine is called universal is that it can be used to simulate any other computer, albeit rather slowly.”
“[...] 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.”


“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.”


“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.”


“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.”


“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.”

“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.”


“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.”


“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.”


“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

“Church-Turing Thesis: Anything that is intuitively computable can be computed by a Turing Machine.”

“But theoretically, anything one computer is capable of doing, another computer is also capable of doing. Given the right instructions, and sufficient memory, the computer found in your fridge could, for example, simulate Microsoft Windows. The fact that it might be ridiculous to waste time using the embedded computer in your fridge to do anything other than what it was designed for is irrelevant - the point is that it obeys the same model of computation as every other computer and can therefore - by hook or by crook - eventually achieve the same result.

This notion is based on what is now called the Church-Turing thesis (dating back to 1936), a hypothesis about calculation devices, such as electronic computers. Alan Turing and Alonzo Church introduced the formalization of an algorithm and a ‘purely mechanical’ model (which assumes that sufficient time and storage space are available). This leads us to the notion of a universal computer, on which all modern computers are based.”


“If we take a step further and couple this to the Church-Turing thesis of the universal computing machine, what this could imply is that any piece of the Universe can simulate, in a more or less efficient manner, any other piece of the Universe.”


“Fortunately we remind ourselves of Turing’s notion of the universal computer, a computer that can simulate any other.”

“For example, we discussed how your fridge microprocessor could be programmed to run Microsoft Windows, then we described Moore’s logic, that computers are becoming faster and smaller. Therefore, one day, a single atom may be able to simulate fully what a present day PC can do.
   This leads to the fascinating possibility that every little constituent of our Universe may be able to simulate any other, given enough time and energy.”

“The Church-Turing thesis is simply stated.

Everything computable is computable by a Turing machine.

The Church-Turing thesis has stood the test of time, capturing computation models Turing could not have conceived of, including digital computation, probabilistic, parallel and quantum computers and the Internet. The thesis has become accepted doctrine in computer science and the ACM has named its highest honor after Turing.”
   L. Fortnow, The enduring legacy of the Turing machine.
   http://ubiquity.acm.org/article.cfm?id=1921573

“So yes Virginia, the Earth is round, man has walked on the moon, Elvis is dead and everything computable is computable by a Turing machine.”
   L. Fortnow, The enduring legacy of the Turing machine.
   http://ubiquity.acm.org/article.cfm?id=1921573

“Everything computable is computable by a Turing machine.”