# Computation Is Universal, Computers Are Not

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

1. *Computation is universal.* Computation is ubiquitous: To be is to compute. Indeed, computing permeates the Universe and drives it: Every atom, every molecule, every cell, everything, everywhere, at every moment, is performing a computation. Thus, from human cognition, to photosynthesis in plants, to the migration of butterflies, all transformational processes involve a computation.

Does the atom, the molecule, or the cell know that it is computing? We cannot know for certain, and truly we should not care. Whether or not the atom, the molecule, or the cell, or for this matter every constituent of the Universe, knows or does not know that it is manipulating information is no more relevant than whether or not our computers know that they are computing. The important point is that the computational paradigm is a very powerful yet simple model that effectively captures the workings of Nature. By explaining nature's processes as "acquiring, manipulating, and transmitting information", we can better understand the world that surrounds us. Being is computing.

See:

E.B. Davies. Building infinite machines. *British Journal for Philosophy of Science*, 52:671--682, 2001.

J. Durand-Lose. Abstract geometrical computation for black hole computation. Research Report No. 2004-15, Laboratoire de l'Informatique du Parallelisme, Ecole Normale Superieure de Lyon, Lyon, France, April 2004.

K. Kelly. God is the machine. *Wired*, 10(12), 2002.

S. Lloyd. *Programming the Universe.* Knopf, New York, 2006.

S. Lloyd and Y.J. Ng. Black hole computers. *Scientific American*, 291(11):53--61, November 2004.

R. Rucker. *The Lifebox, the Seashell, and the Soul*. Thunder's Mouth Press, New York, 2005.

C. Seife. *Decoding the Universe*. Viking Penguin, New York, 2006.

F.J. Tipler. *The Physics of Immortality: Modern Cosmology, God and the Resurrection of the Dead.* Macmillan, London, 1995.

V. Vedral. *Decoding Reality*. Oxford University Press, Oxford, United Kingdom, 2010.

J.A. Wheeler. Information, physics, quanta: The search for links. In *Proceedings of the 3rd International Symposium on Foundations of Quantum Mechanics in Light of New Technology*, pages 354--368. Tokyo, 1989.

J.A. Wheeler. *At Home in the Universe*. American Institute of Physics Press, Woodbury, New York, 1994.

S. Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, Illinois, 2002.

K. Zuse. Calculating Space. MIT Technical Translation AZT-70-164-GEMIT, Massachusetts Institute of Technology (Project MAC), Cambridge, Mass. 02139, 1970.

2. *Computers are not universal.* Contrary to conventional wisdom (encapsulated by the famous, and now defunct, Church-Turing Thesis), it is not true that a single universal computer can perform, through simulation, any computation that is possible on any another computer. This statement is valid even if the putative universal computer is supplied with an infinite memory, an unlimited amount of time, and the ability to interact with the outside world. It is also true whether the computer is theoretical or practical, sequential or parallel, conventional or unconventional.

How is this result arrived at? We first recall a basic assumption in computer science: Every computer (be it real or hypothetical) performs a finite and fixed number of elementary operations per computational step. This number may be a constant (as on a laptop), or it may be a variable (as with accelerating machines), but it is always finite and fixed once and for all (even if it is a function of time). Furthermore, time is divided into discrete time units, and each computational step takes one time unit.

Here's the main theorem: Given $n$ spatially and temporally connected physical variables, $X1$, $X2$, . . . , $Xn$, where $n$ is a positive integer, there exists a function $Fn(X1, X2, . . . , Xn)$ of these variables, such that no computer can evaluate $Fn$ for any arbitrary $n$, unless it is capable of unboundedly many elementary operations per time unit.

The theorem is proven by exhibiting functions $Fn$ computable in a straightforward manner by a computer capable of at most $n$ elementary operations per time unit (call such a computer $Cn$). However, the latter cannot compute $Fn$+1. While $Cn$+1 can now compute the function $Fn$+1, $Cn$+1 is in turn defeated by $Fn$+2. This continues forever.

See:
http://research.cs.queensu.ca/home/akl/techreports/unconventional.pdf
http://research.cs.queensu.ca/home/akl/techreports/even.pdf
http://research.cs.queensu.ca/home/akl/techreports/time_travel.pdf