How the physical Church-Turing thesis changed the concept of machine

Gilles Dowek

In search of a definition

Fundamental concept of machine
Never been completely defined

- Computers, robots, networks are machines
- Machines are physical objects
- Machines (in informatics) are computing devices
- ...

But room for many concepts

This talk

The concept of machine broader and broader

Two factors

- new types of machines developed (or imagined)
- investigation of the limits of computation when performed by machine

The "original" concept of machine

In general: tool that uses a source of energy to achieve some goal

In informatics: tool that uses a source of energy to compute (process information expressed in language)

Processing information purpose of the machine or means to achieve other goals

(original concept continued:) Putting physic at a distance

Maxwell's equations useful to build transistors, not to build adders with transistors

In principle:
Adder: physical system where electromagnetic field evolves
Solve Maxwell's equations to explain why the result is 10
But limit conditions too complex

Instead: levels of abstraction: atoms, transistors, Boolean gates...
A pillar of computational thinking
Puts physics at a distance (involved in lower levels, not upper ones)

I. Various machines

This concept of machine challenged by

- quantum computers
- relativistic computers
- DNA computers
- distributed computers
- analog computers

Quantum computers

Physical time put at a distance

Replaced by logical time (Joinet)

Independence challenged by quantum computing

Feynman: simulating a quantum evolution on a classical computer takes more time than the evolution itself

Superposition

Superposition of two machines brings physics back in

Superposed machines: trapped ions rather than laptops

Built / used

A computer is built

The superposition principle is used

Relativistic computers

Different physical phenomenon: dilatation of time
Duration of a process different for different observers
Infinite for one and finite for another (observer falling in black hole)

With an infinite amount of time, more functions computed than with a finite one (halting problem, *Entscheidungsproblem*)

Machine computing forever outside the black hole not a laptop

Built / used

Dilatation of time used rather than built

DNA computers

Example: solve a Hamiltonian path problem
Different: DNA object, not phenomenon
DNA built by humans

But humans build DNA as animals

DNA bio-physical object used to compute, rather than built

Distributed computers

Networks: machines that connect computers
Computation in nodes not in edges

Connecting computers: more computations
Communication protocols make sense on network, not on computer
The computational power is in the network

Distributed computers

Pushed further: swarm computing
Unstructured network of objects with limited computational power
Network Turing complete, although no node is

Objects either built (tiny computers) or not: ants, birds...
Network itself hardly built: unstructured

Analog computers

By definition: a physical phenomenon used to compute
But, border of informatics (weak computational power, simulating simulated similar)

Different with Turing complete analog computers (Bournez)
Binary adder
Solving a differential equation to get the result 10
Relativize the process of successive construction (kept physics at a distance)

Broader and broader

From computers, built like clocks built from gears and springs

In several steps (semiconductors, transistors, Boolean gates...)
keeping physics at a distance

To using physical phenomena and objects

II. The physical Church-Turing thesis

Another factor that led to broaden the concept of machine

Simple but difficult to explain: result of tortuous historical process
Algorithm description languages before computers

Original thesis: no algorithm description language can express an algorithm computing a non recursive function

After computers were built: no machine can execute an algorithm computing a non recursive function

Predicates "algorithm description language" and "machine" not applied to constant but to quantified variable
Precise definition of "algorithm description language" or "machine"

Gandy's definition

To prove the physical Church-Turing thesis

Quite conservative: mechanical (built) device

But properties of device ("principles of mechanism") paved the way for evolution of the concept of machine

A principle of mechanism

In a machine, element of bounded size: bounded state space
(information has bounded density)

Participates to definition of concept of machine
In some systems information has bounded density, in others not
Only the former can be called "machines"

Bekenstein: information density bounded in all physical systems

All physical systems happen *a posteriori* to be "machines"
according to Gandy's definition

III. The contemporary concept of machine

New types of machines and physical Church-Turing thesis

From a clock-like device, built with gears and springs to (almost) any physical system

- not only how a machine can be built also which physical phenomena and object can be used to compute
- relativize the need to put physics at a distance
- how a physical system can evolve: no physical system can turn green or red according to the termination of Turing machine encoded in its initial state
- relativize difference between object built by humans and those that are not, just like Oklo

Almost?

Body freely falling in vacuum: $y = t^2$
Square or square root?
Depends on the interaction protocol

- Choose time, measure distance: square
- Choose distance, measure time: square root

Not just a physical system, but a physical system equipped with a protocol interpreting the evolution of the system