

Revisiting finite-visit computations

Steven Lindell, Haverford College
slindell@haverford.edu
June 21, 2004

Abstract: Incorporating matter and energy requirements into physically implemented machines sheds new light on feasible computability. Under certain circumstances, conventional mathematical models may not be asymptotically realizable due to the limited availability of resources required to store data and execute instructions.

Given at rump session of Conference on Computational Complexity 2004 1

Emphasize physical realizability.

The conventional approach is to measure complexity via SPACE & TIME, which are inherently unbounded by their nature. An alternative approach is to count the MATTER & ENERGY that a machine model consumes or requires during the course of a computation. Assert that under the right conditions, we can incorporate feasibility into the very definition of computation, making it an inherent component of the definition.

What is a Computation?

Make a philosophical distinction between:

- intention: an abstract mathematical concept based on notions of information transformation (algorithm)
- extension: a concrete physical object that manipulates matter using energy (computer)

Lead to different methods of study:

- anthropomorphic models: *psychological* comprehension
- building real computers: *physiological* understanding

Want an approach that blends these!

2

Begin by purposefully distinguishing between the abstract concept of a step-by-step method operating on symbolic patterns, and its physical execution. Emphasize that Turing obtained his analysis of machine computability by an understanding of human computation. This led to, but did not determine, the engineering challenges involved in designing real computers. In particular, it does not anticipate all physical limitations that might be encountered.

Definitions & Ideas

Information: giving shape, form, or essential character to something (like the soul to the body)

Transformation: a change of shape, form, or essential character (modifying quality without altering some underlying quantity)

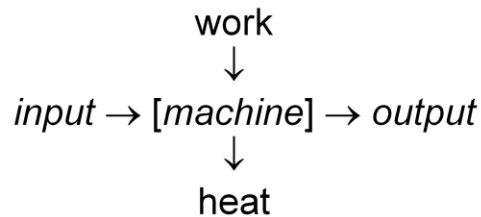
Entropy measures fundamental information content of an ensemble (physical or mathematical) in bits (or symbols): **reversible transformations conserve/preserve information.**

3

Begin with a closer examination of the fundamental notions behind a computation. Can think of patterns as relationships endowing matter with information (e.g. the shape of a zero is a circle and the shape of a one is a line), and methods as steps which utilize energy to modify those patterns (e.g. 0 to 1). The way to measure information quantity is the concept of entropy. By concentrating on deterministic reversible changes we can ensure information is neither created nor destroyed.

Self-powered automaton

Conventional models are open (and isothermal) systems. Not reasonable as increased miniaturization approaches physical limits of memory capacity and processor speed (in one human generation!). Consider ***thermodynamics***:



4

Automaton used to mean a device that operated under its own motive power, not just decisions but also motion. As computers become more powerful, we are going to need to consider the consequences of this power consumption. In particular, the dominant factor is now heat, and will continue to be so even with the next generation of photonic machines. The goal is to accurately model how matter and energy are distributed throughout a computer.

Assumptions

- Represent discrete symbols of information concretely as ***matter***, distributed in *cells* of uniformly bounded density.
- Manipulate by discrete operations which transform using ***energy***; catenated into *steps* of uniformly bounded power.

Definition: A machine is *physically scalable* if it can be built and operated using a fixed technology regardless of the total size of the input and the length of the computation.

Assume **closed** (adiabatic) system. Let size n = mass of the input. Propose studying MATTER(n) and ENERGY(n).

- *Energy is uniformly distributed throughout the matter.*

5

Assumes atomic theory of matter and quantum theory of energy, with upper and lower bounds on density and power. This means that the machine's capability can't change over space and time. Similar to symmetries which govern the laws of our universe: spatial invariance yields conservation of momentum and temporal invariance yields conservation of energy. Emphasize that closure also means system is finite and self-contained. Mass corresponds to SPACE; energy corresponds TIME.

Constraints

- No I/O: computing *in situ* (like Turing machines) on graphs (must pad input for additional work space).
- *Abstract* data types must become *concrete* data structures

Basic premises: **$O(1)$ bits/cell** **$O(1)$ work/step**

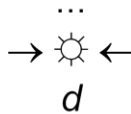
- *In particular, each step is concerned with a fixed amount of information to be processed, and a fixed number of choices for the next move.*
- N.b. Turing was already aware of the asymptotic infeasibility of RAM.

6

Since all physical resources are given (the input supplies both matter & energy), everything is done in place. Think of a sequential finitely controlled head (or even concurrent heads) walking about a tape of symbols drawn from a fixed alphabet, which it can be modify provided it doesn't change the shape of the tape. Note that by fixing the word size (exactly one) we preclude random-access models which actually manipulate unbounded amounts of information, something which Turing knew.

Bounded Degree

Clearly, we should have bounded out-degree at each node, but what about the in-degree d ? Consider entropy loss:



Argument: Information lost upon moving in is $\log_2 d$ bits. This costs kT per bit of energy by the 2nd law [Landauer]. Therefore, in-degree must be bounded for $O(1)$ work per step!

7

Although uniformity considerations would also bound the degree of the tape, this cute argument illustrates how intimately thermodynamics influences what seems like a purely abstract question. Landauer's insight was only achieved from examining the apparent paradox of Maxwell's demon, finally resolved after a century of study. Additional note: edges in our graphs are static, as opposed to the dynamically re-configurable pointer machines of Kolmogorov and Schönhage.

Finite Visit

Spatial arrangement (to **avoid** *overcrowding*):

- symbols lie in (symmetric) *bounded-degree* data structure

Temporal rule (to **avoid** *overheating*):

- modify those symbols in a (reversible) *finite-visit* fashion

Argument:

1. because of non-zero transmission losses, any energy beyond a certain radius is inaccessible
2. within a given radius there is only $O(1)$ energy

8

We saw how physical considerations prevent overcrowding, but a deeper insight is required to see that each cell can only be operated on a fixed number of times. Concentrating on power distribution (instead of overheating), observe that even though the amount of energy increases polynomially with the radius in a finite dimensional space, its availability drops off exponentially with distance. No hot spots! Can replenish indefinitely in a tree.

Conclusion

One Consequence: An *algorithm* (a mathematical notion) may fail to be asymptotically realizable as a *computation* (a physical notion)!

Simple Reason: on a sufficiently large input, it might run out of power and/or overheat -- cannot continue to ignore physical consequences.

Key Idea: work happens in bounded size neighborhoods, with only a limited number of opportunities to make changes to any given neighborhood.

1-D finite-visit = regular [Hennie] 2-D finite-visit = universal

9

This is very surprising and somewhat counter-intuitive. The consequence refers to implementation within a closed system. The reason is basically because a mechanical automaton -- in the classical sense of both words -- refers to a self-powered device obeying physical law. The idea says that local patterns within the machine cannot be modified over and over again. Hennie's conclusion was in 1965. Easy to see a grid can simulate an arbitrary TM provided it is sufficiently large.

Summary

Premise: Incorporate physical resources into mathematical model.

Conclusion: ENERGY must be linearly proportional to MATTER.

Compare Hypercomputation: "beyond the Turing limit"

- hypothetically relies on infinite resolution of continuous-values

Propose Hypocomputation: "beneath the Turing limit"

- recognize nature only admits finite resolution of discrete values

10

Take away: a closed computer, physically uniform with respect to size of the data it is asked to process, necessarily has strong restrictions on what it can and can't compute -- a natural notion of feasibility. So instead of going beyond the Turing limit -- an entire recent issue of TCS was devoted to this topic -- I propose we should stay within the Turing limit because that may be what the rules of the natural world imposes on us.