

# Revisiting finite-visit computations

*Steven Lindell, Haverford College*

[slindell@haverford.edu](mailto: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.

# 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!**

# Definitions & Ideas

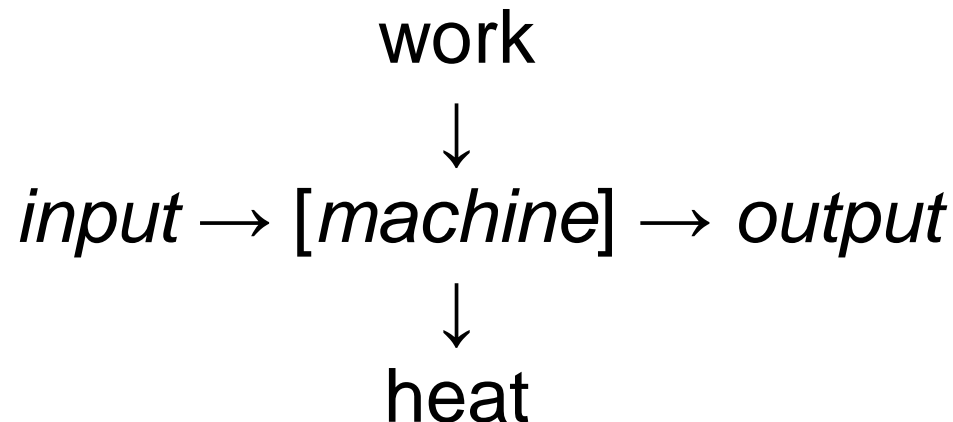
*Information:* giving shape, form, or essential character to something (like the soul to the body)

*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.**

# 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***:



# 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.*

# Constraints

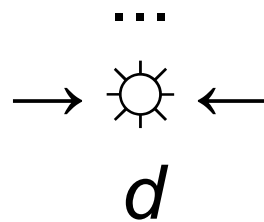
- No I/O: computing *in situ* (like Turing machines) on graphs (must pad input for additional work space).
- *Abstract* data types must become *concrete* data structures

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.

# 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!

# 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

# 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

# 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