Summary:
Calculability, Physics, and the Final Limits

Calculability

What can be calculated by a mechanical (algorithmic) procedure in principle

K. F. Goedel (1906-1978)

(Second) Incompleteness Theorem: Every sufficiently strong, computably axiomatized, consistent formal system for arithmetic leaves some arithmetic questions undecidable.

The second theorem goes further: such a system cannot establish its own consistency using only its own methods, assuming it is in fact consistent.

A. M. Turing (1912-1954)

Halting Problem: There is no computer program that can correctly determine, for every possible program and every possible input, whether that program will eventually stop running or continue forever.

Calculability

A. F. Church (1903-1995)

Church-Turing Thesis

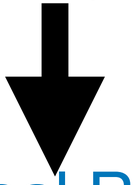
Anything that can be computed by any mechanical, algorithmic procedure can be computed by a Turing machine.

Extended Church-Turing Thesis

Any realistic model of computation can be simulated by a Turing machine with at most **polynomial(*)** overhead in time and space.

Or:

No physically realizable computer can solve computational problems fundamentally faster than a Turing machine, except possibly by a polynomial factor.



Physical Reality

Computers must obey physical laws (classical/quantum)

(*) we need the notion of efficiency —> complexity theory.

Complexity (e.g. Time)

Hyp:
Every elementary calculation requires a fixed amount of time

SUM
$x_1 + x_2 + x_3 + \ldots x_N$ requires O(N) steps.

MATRIXMULTIPLICATION
Multiply two NxN matrices: requires $O(N^3)$ time. Can we improve it?

MATRIXMULTIPLICATION

Year	Discoverer(s)	Exponent	Improvement over ($O(n^3)$)
1969	Volker Strassen	2.8074	First sub-cubic algorithm
1978	Victor Pan	2.796	Improved tensor methods
1979	Don Coppersmith & Shmuel Winograd	2.496	Landmark improvement
1987	Don Coppersmith & Shmuel Winograd	2.376	Refined construction
2010	Virginia Vassilevska Williams	2.3737	New analysis of Coppersmith–Winograd
2012	François Le Gall	2.3728639	Record at the time
2014	Virginia Vassilevska Williams, Josh Alman (independent refinements followed)	~2.37286	Tiny improvements
2020	Josh Alman & Virginia Vassilevska Williams	2.3728596	Current best published exponent

No exact lower bound known …

SPACE Complexity improvement example

GRAPHREACHIBILITY

Can you get from one point in a network to another by following the connections?

Year	**Result**	**Space**	**Contributor(s)**
1960s	DFS uses stack	(O(n)) space	classical
1970	Savitch's algorithm for reachability	(O(\log^2 n)) nondeterministic→deterministic conversion	Walter Savitch
2004	Reingold's theorem (undirected connectivity in log space)	(O(\log n)) deterministic	Omer Reingold

Physically Realizable Computers

Must obey physical laws (classical/quantum)

Computers are physical systems

Question: what is the “cost” of computation?
How much **energy** do we need?

Entropy in Thermodynamics and Information

In physics, entropy measures how many microscopic configurations of a physical system correspond to the same macroscopic description. If a gas can be arranged in many different ways while still looking like the same temperature and pressure, that system has high entropy.

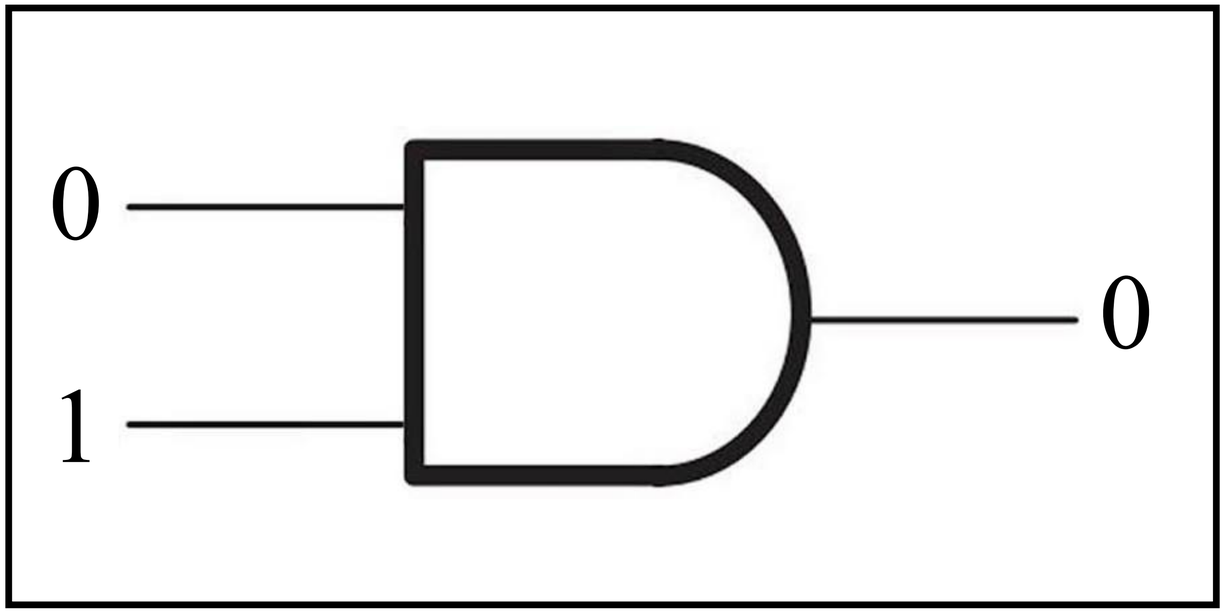
$$S = -k_B \sum_i p_i \ln p_i$$

In information theory, (C. Shannon) entropy measures uncertainty. If you are unsure which message you will receive, entropy quantifies how many different messages are possible and how evenly they are distributed.

$$H = -\sum_i p_i \log_2 p_i$$

Besides a factor, the two definitions are the same: $S = k_B \log_2 H$

Information loss and Entropy



Example: the **AND gate** looses information, since it is not always possible to reconstruct the inputs given the knowledge of the output: **IRREVERSIBILIY**.

Information loss and Entropy

After performing the AND operation and seeing only the output, there is **uncertainty** about which input pair was actually used:

$$H(X,Y|Z) = \sum_z P(Z=z)\left[-\sum_{xy} P(x,y|z)\log_2 P(x,y|z)\right]$$

$$= \frac{1}{4}\left[-1\cdot\log_2 1\right] + \frac{3}{4}\left[-3\left(\frac{1}{3}\log_2\frac{1}{3}\right)\right] =$$

$$= 0 + \frac{3}{4}\log_2 3 \simeq 1.89$$

- If the result is Z=1, we know for sure that X=Y=1
- If the result is Z=0, we have 3 possible inputs over a total of 4 (3/4)

Physically, if we “forget” the inputs, we should have a loss of entropy from the system to the environment.
We forget 2 bits: $\Delta S = 2k_B \ln 2$

Can we compute with irreversible gates?

Toffoli Gate

CCNOT: $(a, b, c) \rightarrow (a, b, c \oplus (a \wedge b))$

a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

YES: for example Toffoli Gate + NOT = Functional completeness

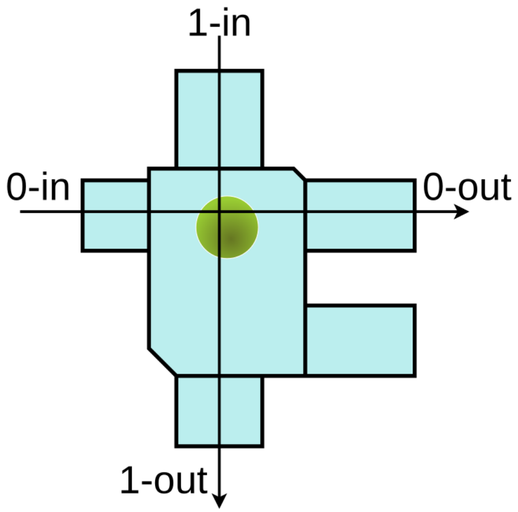
Reversible computation: no dissipation!

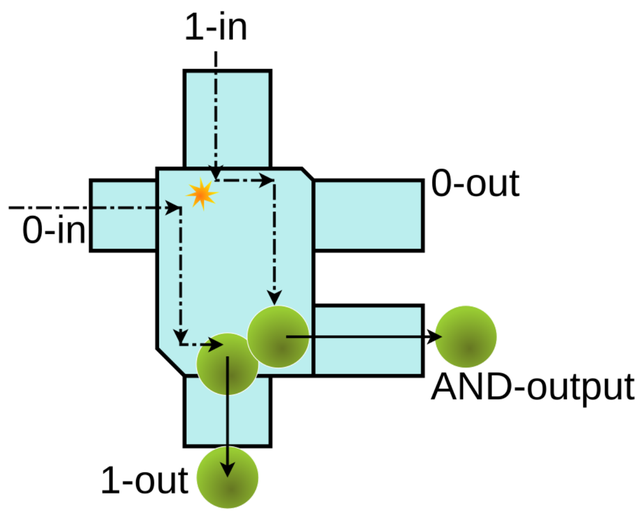
Another example of reversible computation

The Toffoli-Fredkin “Billiard-Computer”

- Fully based on elastic collisions
- No dissipation in principle
- Can realize all gates

AND Gate





from Wikipedia

Edward Fredkin (1934 – 2023)
Tommaso Toffoli (1943-)

Is it really possible to compute without dissipation ?

Landauer's key insight was not that "computation costs energy," but something more precise and more subtle: **only logically irreversible operations necessarily generate heat**.

Rolf W. Landauer
(1927-1999)

We saw already that computation itself can be done without energy change if the processes are reversible. Landauer understood that one unavoidable process requires dissipation: erasure.

The deep idea is that entropy increase is not tied to computation itself, but to the compression of logical possibilities. A reversible computation just rearranges information among physical states without collapsing possibilities. By contrast, resetting a memory cell takes many possible initial states and maps them to one final state, and this reduction in logical information must be compensated by an increase in physical entropy elsewhere.

If you want to re-use a physical system for computation, a minimum entropy change is unavoidable.

Ultimate Limits from Physics (?)

Margolus-Levitin Theorem $$\frac{\text{Number of Operations}}{\text{Time}} \sim \frac{2E}{\pi\hbar}$$

How many operations during the age of the Universe (13.8Gy)?

$$N_{ops} \sim \frac{Et}{\hbar} = \frac{E \times Age_U}{\hbar} \approx \frac{10^{70} \times 10^{17}}{10^{-34}} = 10^{121}$$

Maximum Entropy (Hartle-Hawking): $$S_{max} \sim \frac{k_B c^3}{G\hbar} \cdot \frac{A}{4} \approx k_B \cdot 10^{122}$$

Maximum number of bits in the Universe:

$$N_{bits} = \frac{S}{k_B \ln 2} \approx 10^{122} \approx 10^{109} \text{ Terabytes}$$

Challenge to the ECTT

ECTT: Efficient in physics ⇒ efficient on a Turing machine (up to polynomial overhead)

The Extended Church–Turing Thesis is falsified if there exists a physically realizable model of computation that solves some problems in polynomial resources while requiring super-polynomial time on any Turing machine under any polynomially faithful encoding of the physical system.

Integer Factorization

FACTORIZATION is the problem of taking a whole number and breaking it into its prime factors.

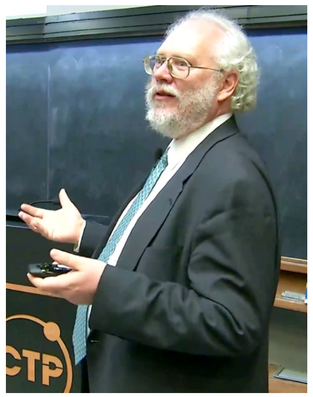
Fundamental in cryptography (e.g. RSA algorithm):
Integer multiplication: easy (503×509=256027)
Factorization: hard (what are the prime factors of 256027 ?)

Best known algorithm: General Number Field Sieve (GNFS) with time complexity

$$\approx exp\left[\mathcal{O}(n^{1/3}(\log n)^{2/3})\right]$$

Shor's Algorithm

Shor's algorithm uses quantum superposition and interference to find hidden periodic structure in numbers. This periodicity can then be converted into the prime factors of a large integer.



Peter W. Shor (1959)

Basic idea:

1. Builds a quantum state that encodes many possible inputs at once
2. Uses a quantum transformation to "reveal" a hidden periodic pattern (a period in modular exponentiation)
3. Extracts that period using a quantum Fourier transform
4. Converts the period into the factors of the number

Polynomial time in the number of digits.

ECTT: Quantum Version

If FACTORING is not in P classically but is in P on a physically realizable quantum computer (as in Shor's algorithm), then the classical ECTT is likely false, and a quantum version becomes the natural replacement:

Any physically realizable model of computation can be efficiently simulated by a quantum Turing machine with at most polynomial overhead.

$$|\Psi\rangle = \sum_i c_i |T_i\rangle$$