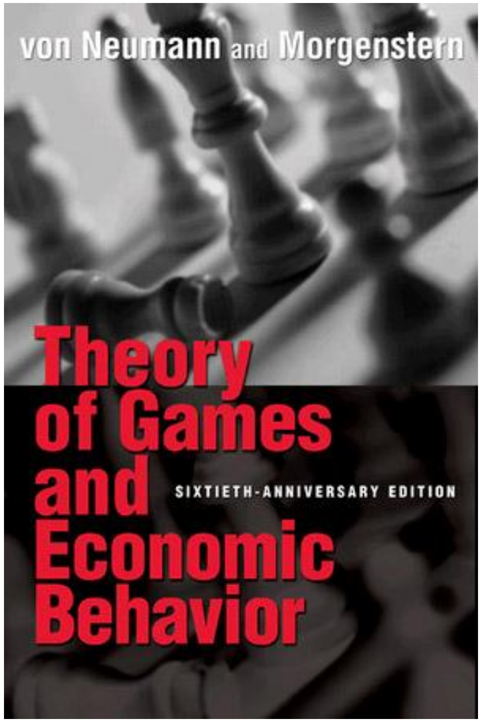
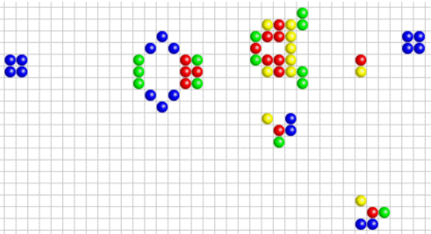
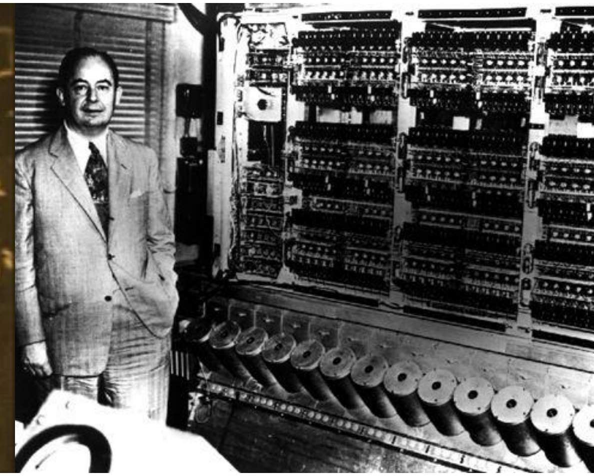
Historical Perspectives

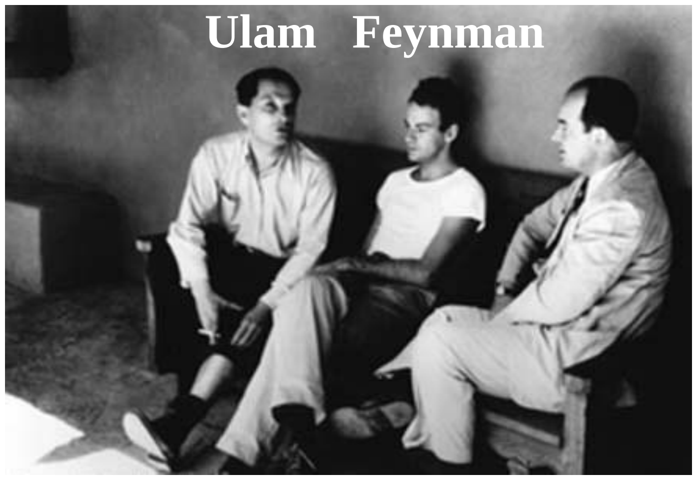
John von Neumann (1903-1957)

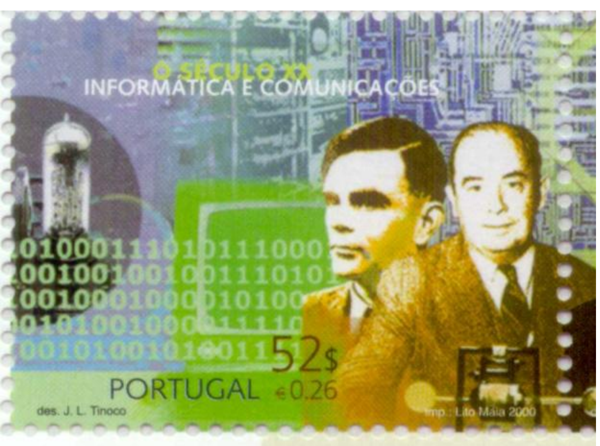
- Contributed to set theory, functional analysis, quantum mechanics, ergodic theory, economics, geometry, hydrodynamics, statistics, analysis, measure theory, ballistics, meteorology, …
- Invented game theory (used in Cold War)
- Re-axiomatized set theory
- Principal member of Manhattan Project
- Helped design the hydrogen / fusion bomb
- Pioneered modern computer science
- Originated the “stored program”
- “von Neumann architecture” and “bottleneck”
- Helped design & build the EDVAC computer
- Created the field of cellular automata
- Investigated self-replication
- Invented merge sort



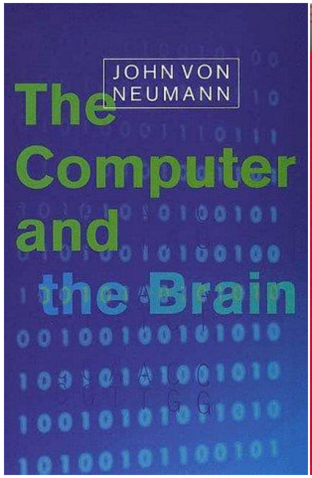


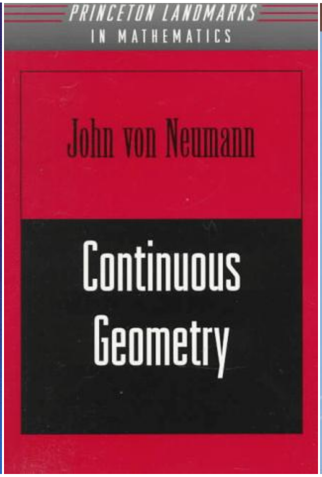


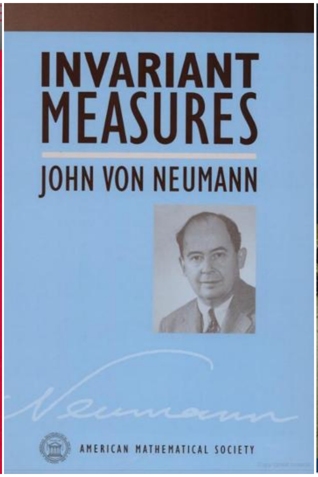


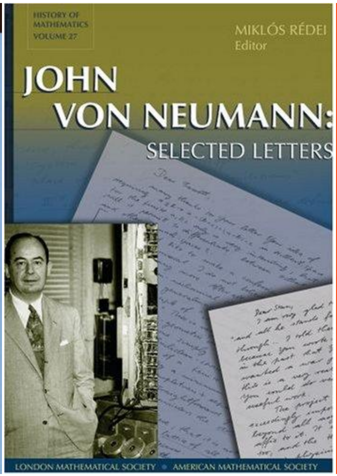


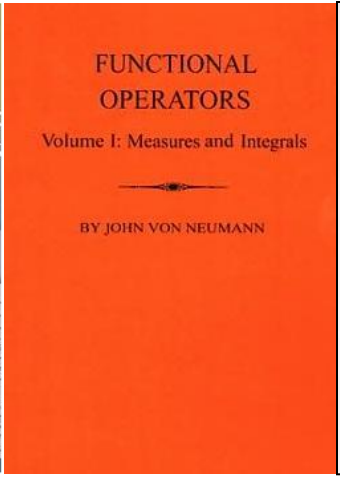
"Most mathematicians prove what they can; von Neumann proves what he wants."

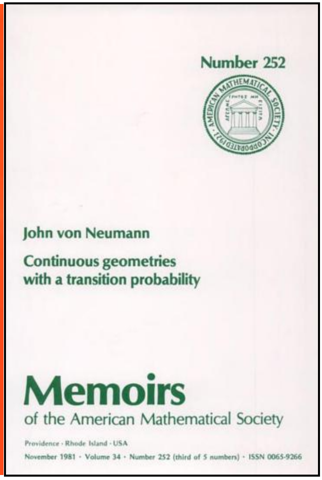

JOHN VON NEUMANN
The Computer and the Brain

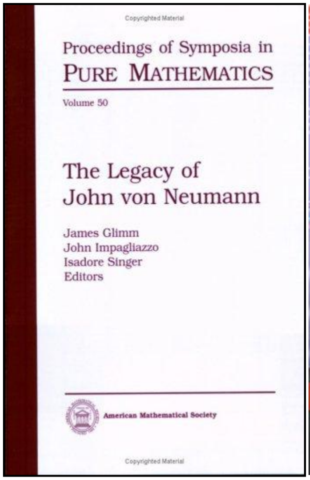

PRINCETON LANDMARKS IN MATHEMATICS
John von Neumann
Continuous Geometry


INVARIANT MEASURES
JOHN VON NEUMANN
AMERICAN MATHEMATICAL SOCIETY

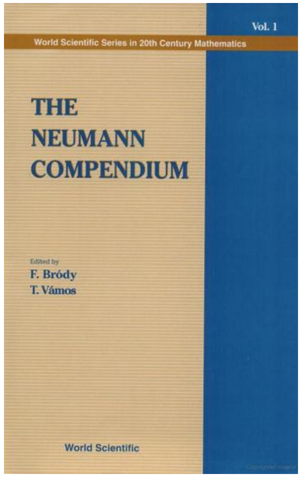

HISTORY OF MATHEMATICS VOLUME 27
MIKLÓS RÉDEI Editor
JOHN VON NEUMANN: SELECTED LETTERS
LONDON MATHEMATICAL SOCIETY • AMERICAN MATHEMATICAL SOCIETY

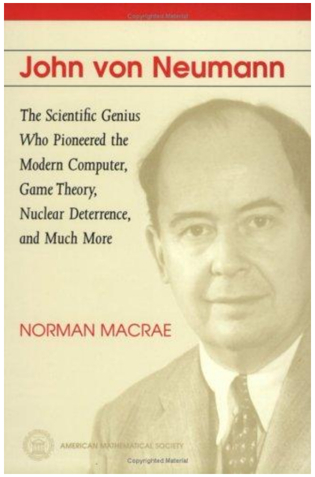

FUNCTIONAL OPERATORS
Volume I: Measures and Integrals
BY JOHN VON NEUMANN

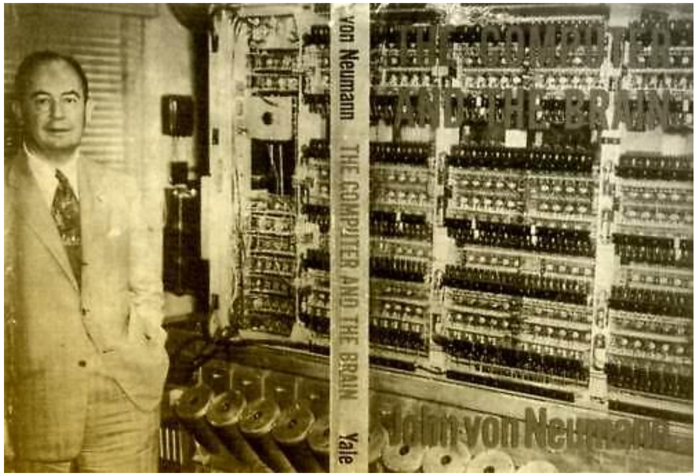

Number 252
John von Neumann
Continuous geometries with a transition probability
Memoirs of the American Mathematical Society
Providence · Rhode Island · USA
November 1981 · Volume 34 · Number 252 (third of 5 numbers) · ISSN 0065-9266


Proceedings of Symposia in PURE MATHEMATICS
Volume 50
The Legacy of John von Neumann
James Glimm
John Impagliazzo
Isadore Singer
Editors
American Mathematical Society

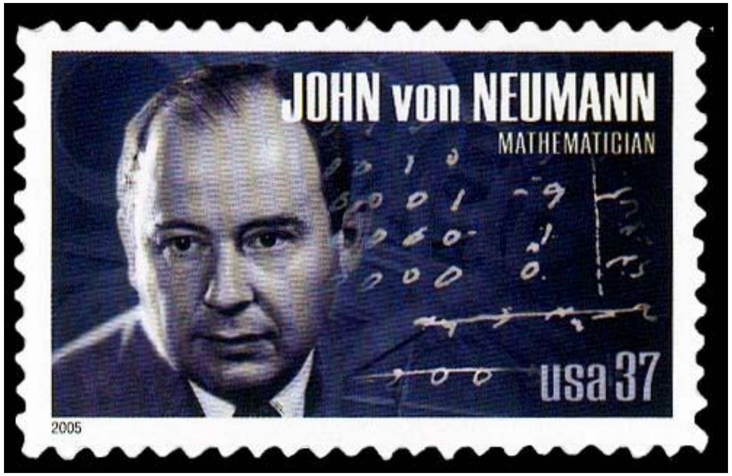

The Computer
from Pascal to von Neumann
Herman H. Goldstine

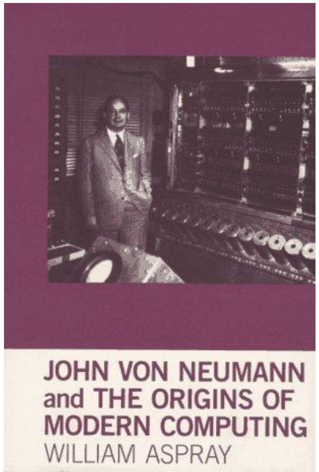

Vol. 1
World Scientific Series in 20th Century Mathematics
THE NEUMANN COMPENDIUM
Edited by
F. Bródy
T. Vámos
World Scientific


John von Neumann
The Scientific Genius Who Pioneered the Modern Computer, Game Theory, Nuclear Deterrence, and Much More
NORMAN MACRAE
AMERICAN MATHEMATICAL SOCIETY


von Neumann
THE COMPUTER AND THE BRAIN
Yale
John von Neumann

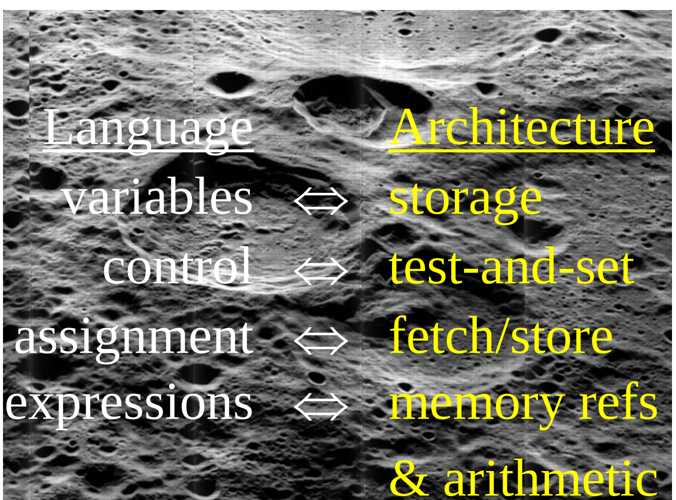

FELADÓ
LEVELEZŐLAP
MAGYAR POSTA
NEUMANN JÁNOS SZÁMÍTÓGÉP-TUDOMÁNYI TÁRSASÁG
INTERKOMPUTO BUDAPEST 1982
KOMPUTO-SCIENCA SOCIETO JOHANO NEUMANN

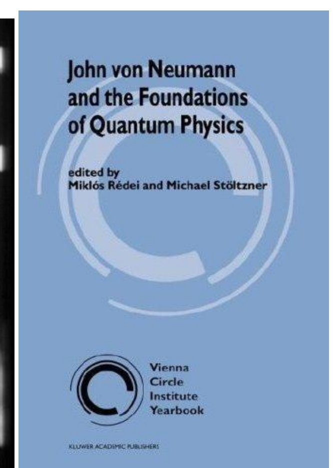

JOHN von NEUMANN
MATHEMATICIAN
usa 37
2005

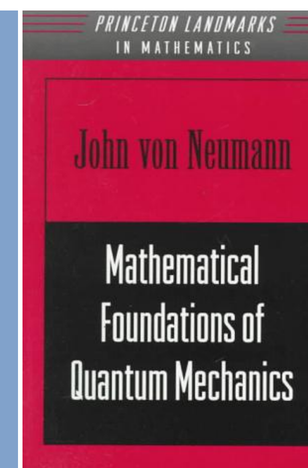

JOHN VON NEUMANN and THE ORIGINS OF MODERN COMPUTING
WILLIAM ASPRAY

von Neumann's Legacy

- Re-axiomatized set theory to address Russell's paradox
- Independently proved Godel's second incompleteness theorem: aximomatic systems are unable to prove their own consistency.
- Addressed Hilbert's 6th problem: axiomatized quantum mechanics using Hilbert spaces.
- Developed the game-theory based Mutually-Assured Destruction (MAD) strategic equilibrium policy – still in effect today!
- von Neumann regular rings, von Neumann bicommutant theorem, von Neumann entropy, von Neumann programming languages



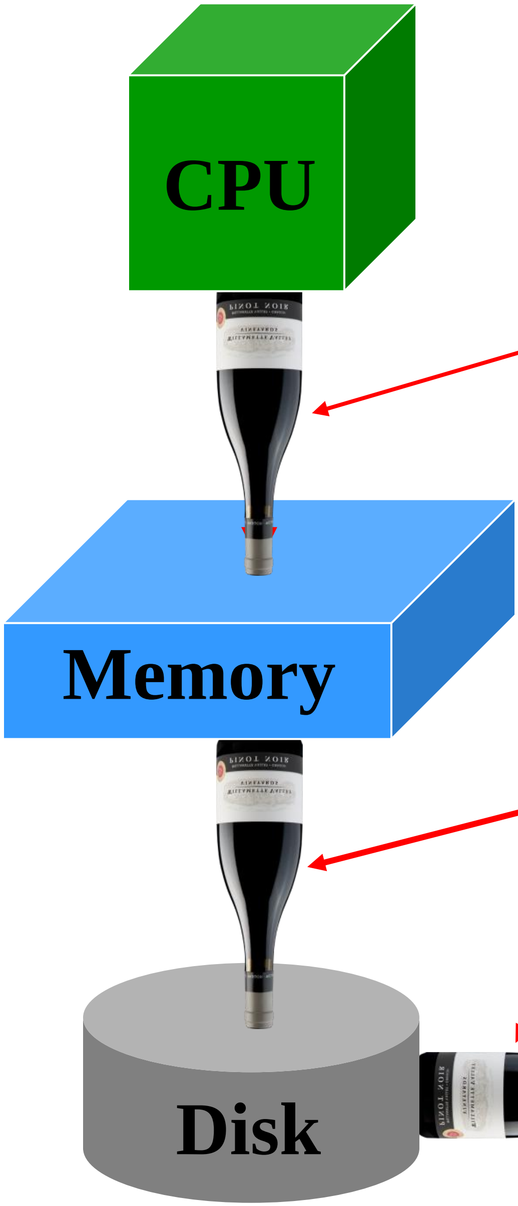




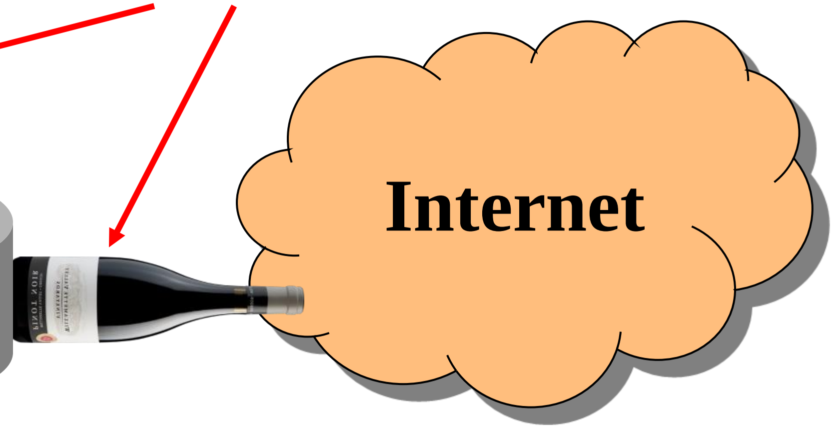
Von Neumann Architecture

"Surely there must be a less primitive way of making big changes in the store than by pushing vast numbers of words back and forth through the **von Neumann bottleneck**. Not only is this tube a literal bottleneck for the data traffic of a problem, but, more importantly, it is an intellectual bottleneck that has kept us tied to word-at-a-time thinking instead of encouraging us to think in terms of the larger conceptual units of the task at hand. Thus programming is basically planning and detailing the enormous traffic of words through the Von Neumann bottleneck, and much of that traffic concerns not significant data itself, but where to find it."

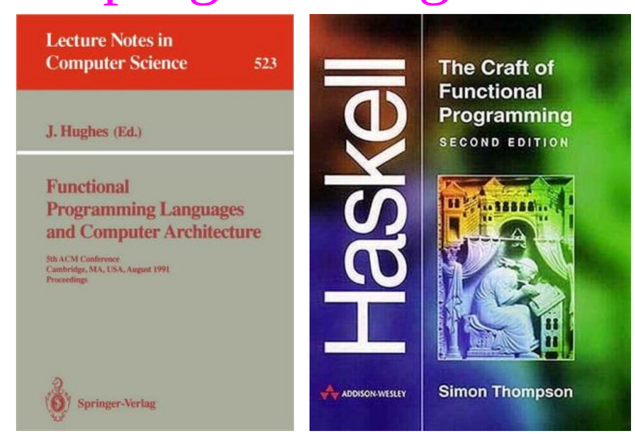
- John Backus, 1977 ACM Turing Award lecture



More bottlenecks



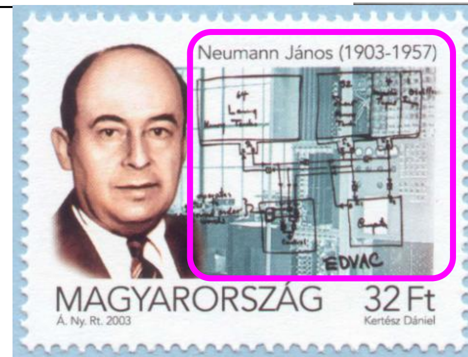
Functional programming



First Draft of a Report on the EDVAC

by

John von Neumann



Contract No. W–670–ORD–4926

Between the

United States Army Ordnance Department

and the

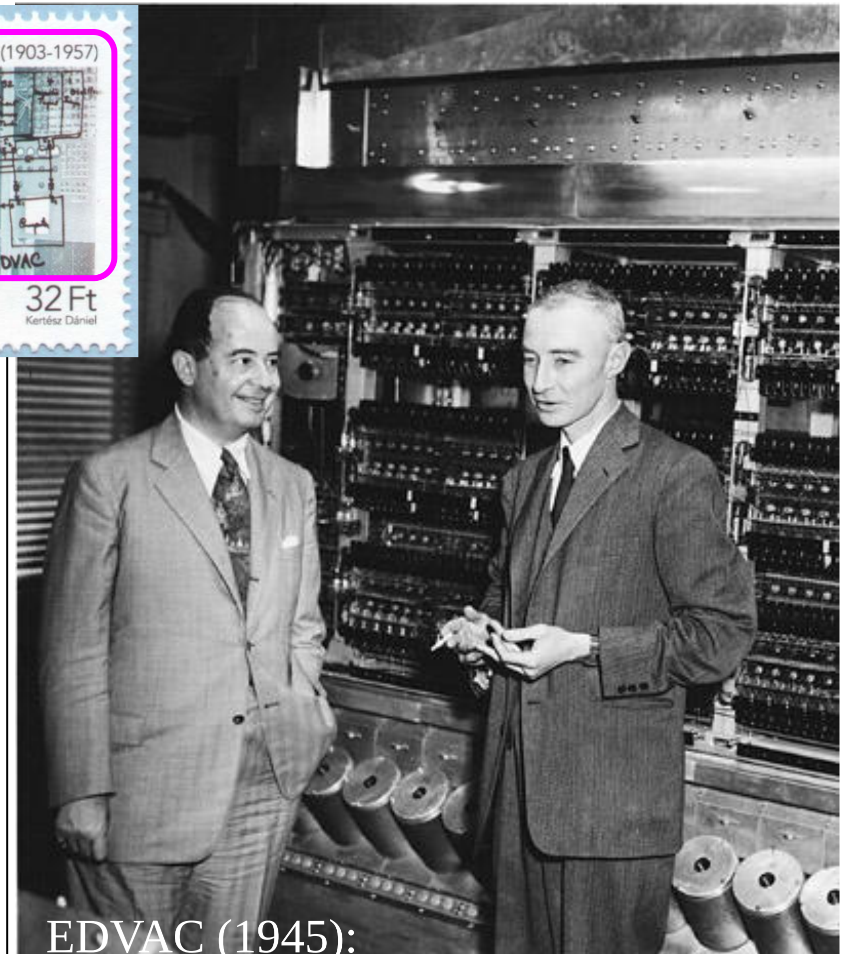
University of Pennsylvania

Moore School of Electrical Engineering
University of Pennsylvania

June 30, 1945

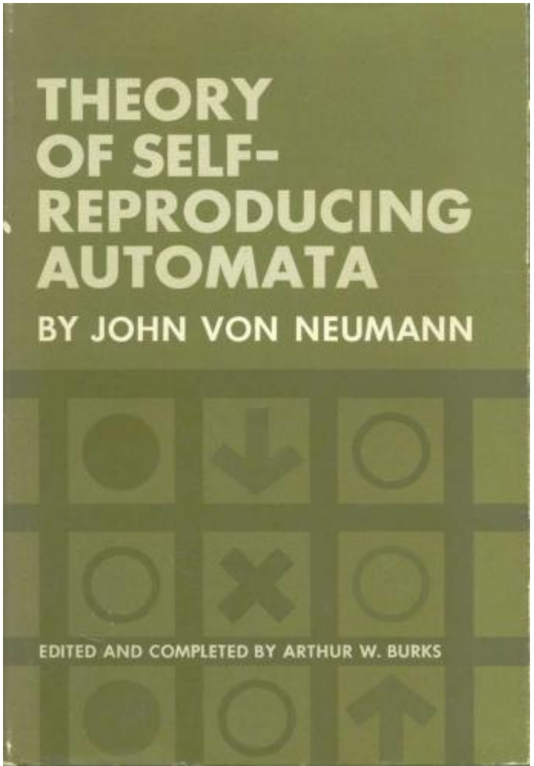
This is an exact copy of the original typescript draft as obtained from the University of Pennsylvania Moore School Library except that a large number of typographical errors have been corrected and the forward references that von Neumann had not filled in are provided where possible. Missing references, mainly to unwritten Sections after 15.0, are indicated by empty {}. All added material, mainly forward references, is enclosed in { }. The text and figures have been reset using TEX in order to improve readability. However, the original manuscript layout has been adhered to very closely. For a more "modern" interpretation of the von Neumann design see M. D. Godfrey and D. F. Hendry, "The Computer as von Neumann Planned It," *IEEE Annals of the History of Computing*, vol. 15 no. 1, 1993.

Michael D. Godfrey, Information Systems Laboratory, Electrical Engineering Department
Stanford University, Stanford, California, November 1992



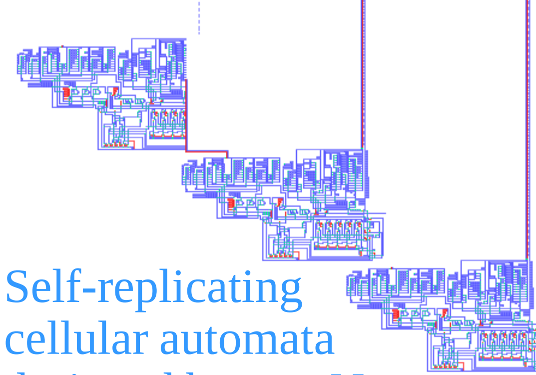
EDVAC (1945):

- 1024 words (44-bits) – 5.5KB
- 864 microsec / add (1157 / sec)
- 2900 microsec / multiply (345/sec)
- Magnetic tape (no disk), oscilloscope
- 6,000 vacuum tubes
- 56,000 Watts of power
- 17,300 lbs (7.9 tons), 490 sqft
- 30 people to operate
- Cost: $500,000

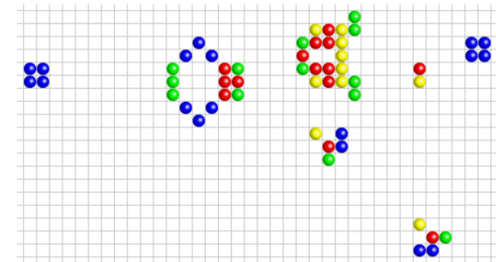


Self-Replication

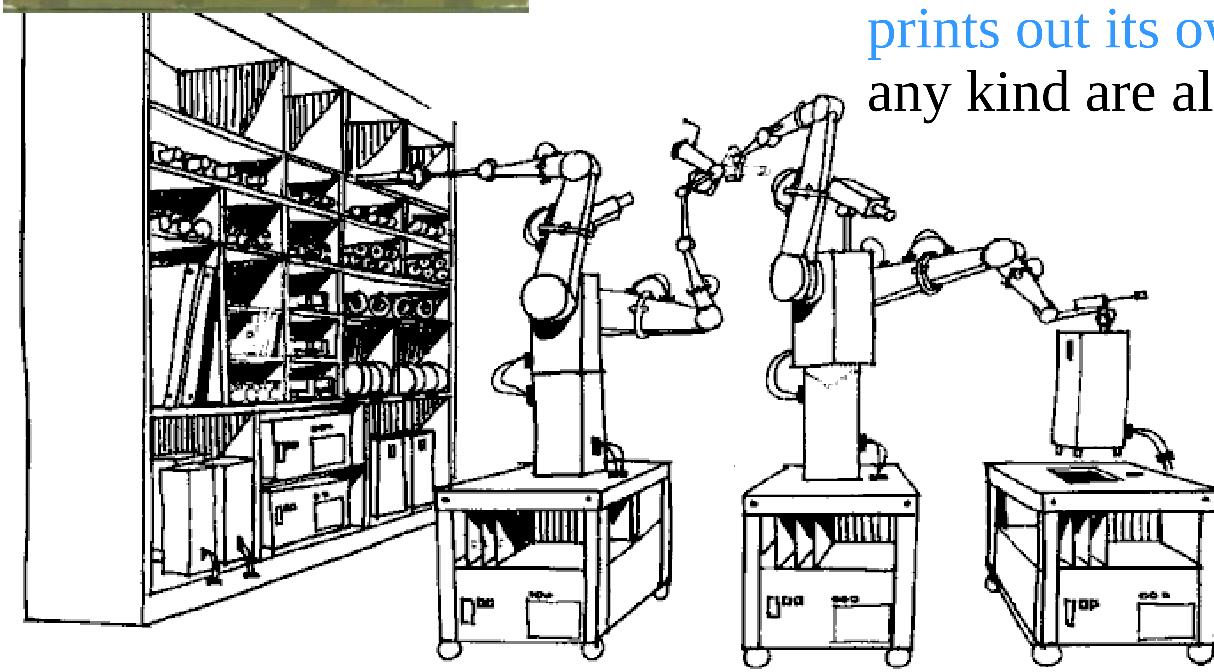
- Biology / DNA
- Nanotechnology
- Computer viruses
- Space exploration
- Memetics / memes
- "Gray goo"



Self-replicating cellular automata designed by von Neumann



Problem (extra credit): write a program that prints out its own source code (no inputs of any kind are allowed).



US005764518A

United States Patent [19]

Collins

[11] **Patent Number: 5,764,518**

[45] **Date of Patent: Jun. 9, 1998**

[54] **SELF REPRODUCING FUNDAMENTAL FABRICATING MACHINE SYSTEM**

[76] Inventor: **Charles M. Collins**, 10800 Oak Wilds Ct., Burke, Va. 22015

[21] Appl. No.: **757,005**

[22] Filed: **Nov. 25, 1996**

Related U.S. Application Data

[63] Continuation-in-part of Ser. No. 364,926, Dec. 28, 1994, Pat. No. 5,659,477.

[51] **Int. Cl.**[6] **G06F 19/00**

[52] **U.S. Cl.** **364/468.01**; 364/468.24

[58] **Field of Search** 364/468.23, 468.22, 364/468.19, 468.01, 468.2, 468.21, 468.24, 474.21, 478.01, 478.03, 478.05, 478.06, 478.13–478.18, 424.028, 424.027, 424.07; 180/168, 8.1–8.7; 104/88.03, 88.04, 88.02; 901/6–8, 1; 318/568.12, 587; 395/80, 82, 901

[56] **References Cited**

U.S. PATENT DOCUMENTS

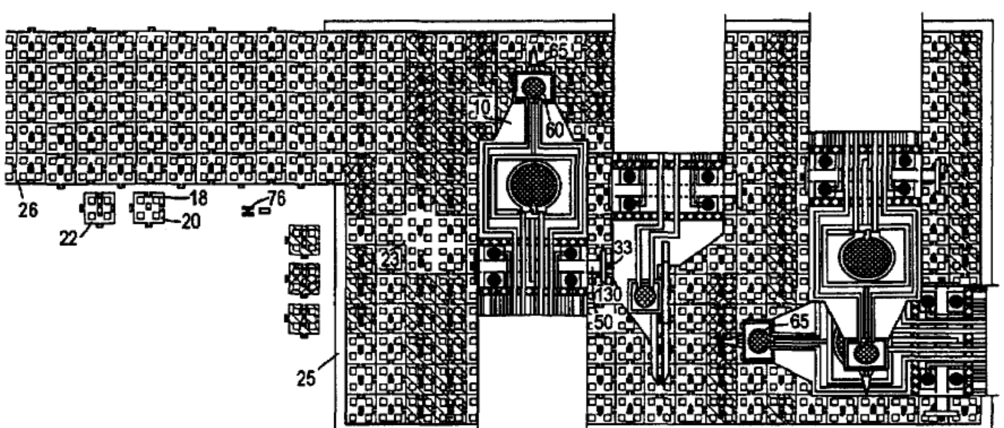
4,621,410	11/1986	Williamson	364/468.22 X
4,669,168	6/1987	Tamura et al.	901/7 X
4,870,592	9/1989	Lampi et al.	364/468.19
4,964,062	10/1990	Ubhayakar et al.	901/39 X
5,084,829	1/1992	Kato	901/7 X
5,145,130	9/1992	Purves	901/1 X
5,150,288	9/1992	Imai et al.	364/468.23 X
5,214,588	5/1993	Kaneko et al.	364/468.2

Primary Examiner—Joseph Ruggiero
Attorney, Agent, or Firm—Henry G. Kohlmann

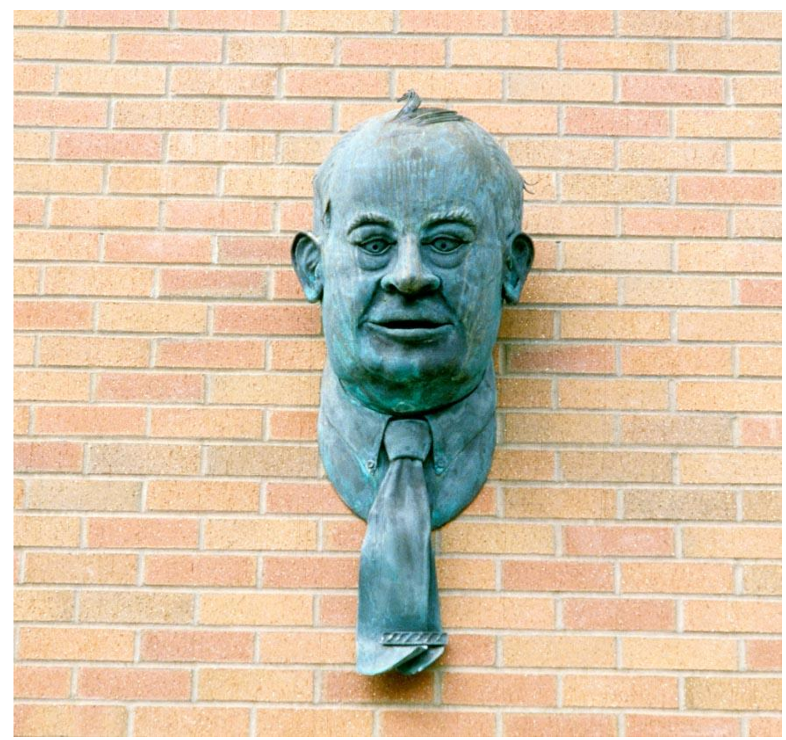
[57] **ABSTRACT**

A system of units for constructing or replicating a means (**10.10.10**p) including means of diverse materials consisting of a plurality of pieces (**20.22.23. 156–165**) having at least one indicia (**18**) thereon for detection thereof, at least one adjoining means functioning according to instructions of a computer program of a processor means for adjoining in any predetermined relation with other of the plurality of the pieces (**20. 22. 23. 156–165**), and the processor means (**30. 120. 166. 167**) having the computer program instructions being responsive to detection of the at least one indicia to provide for arranging the other of the plurality of the pieces in the predetermined relation for controlling the fabrication means in assembling a given number of the plurality of the pieces in the predetermined relation to comprise a produced fabrication means (**10,10,10**p) are selected from a group consisting of a puzzle piece system, a construction system, a hot knife system, a holed piece system.

75 Claims, 30 Drawing Sheets

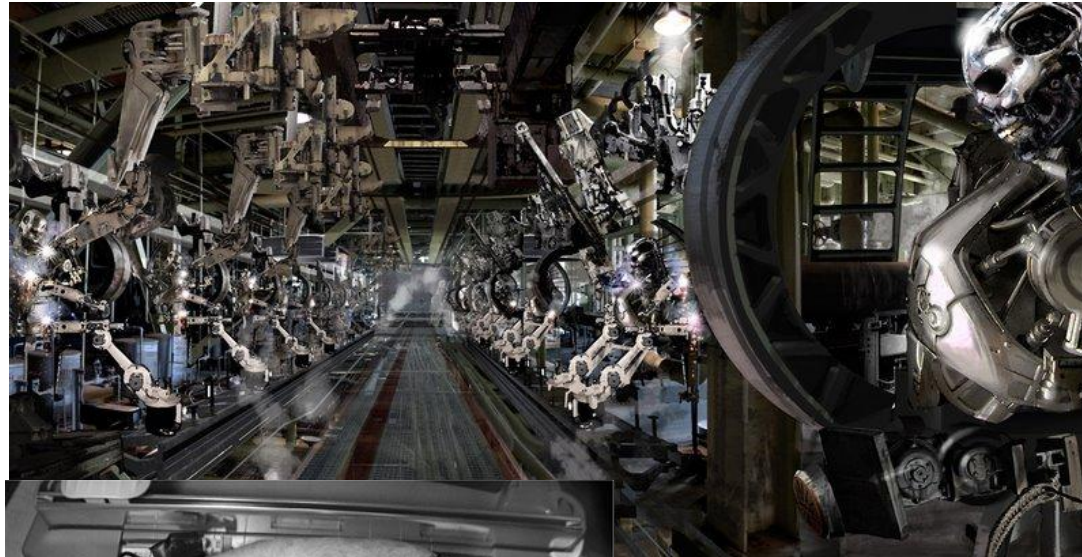


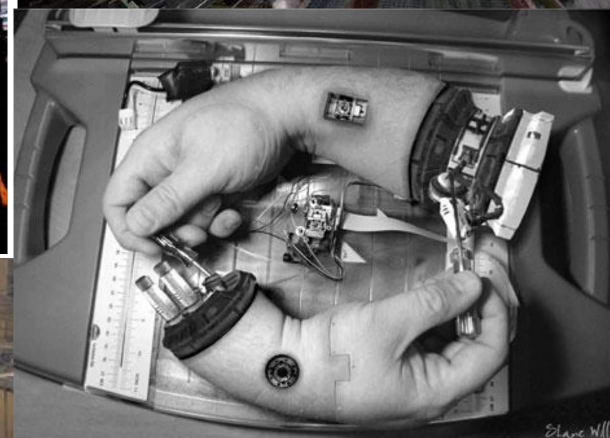
"In mathematics you don't understand things. You just get used to them."
– John von Neumann

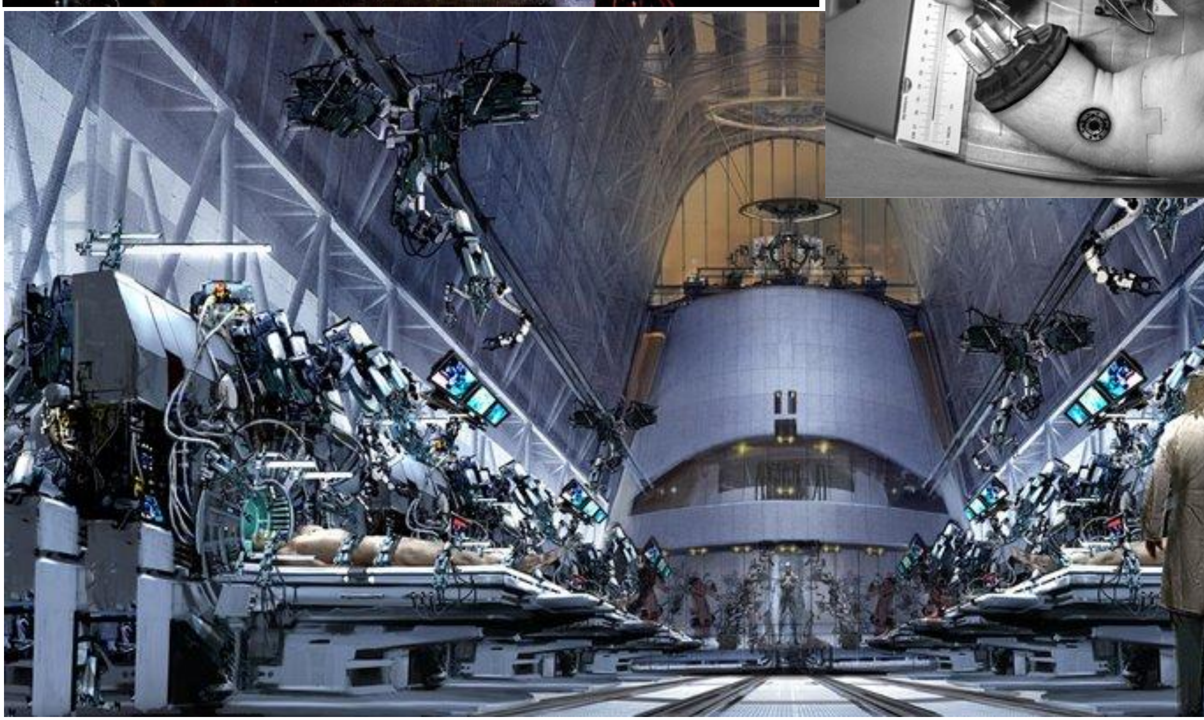



ROB-F

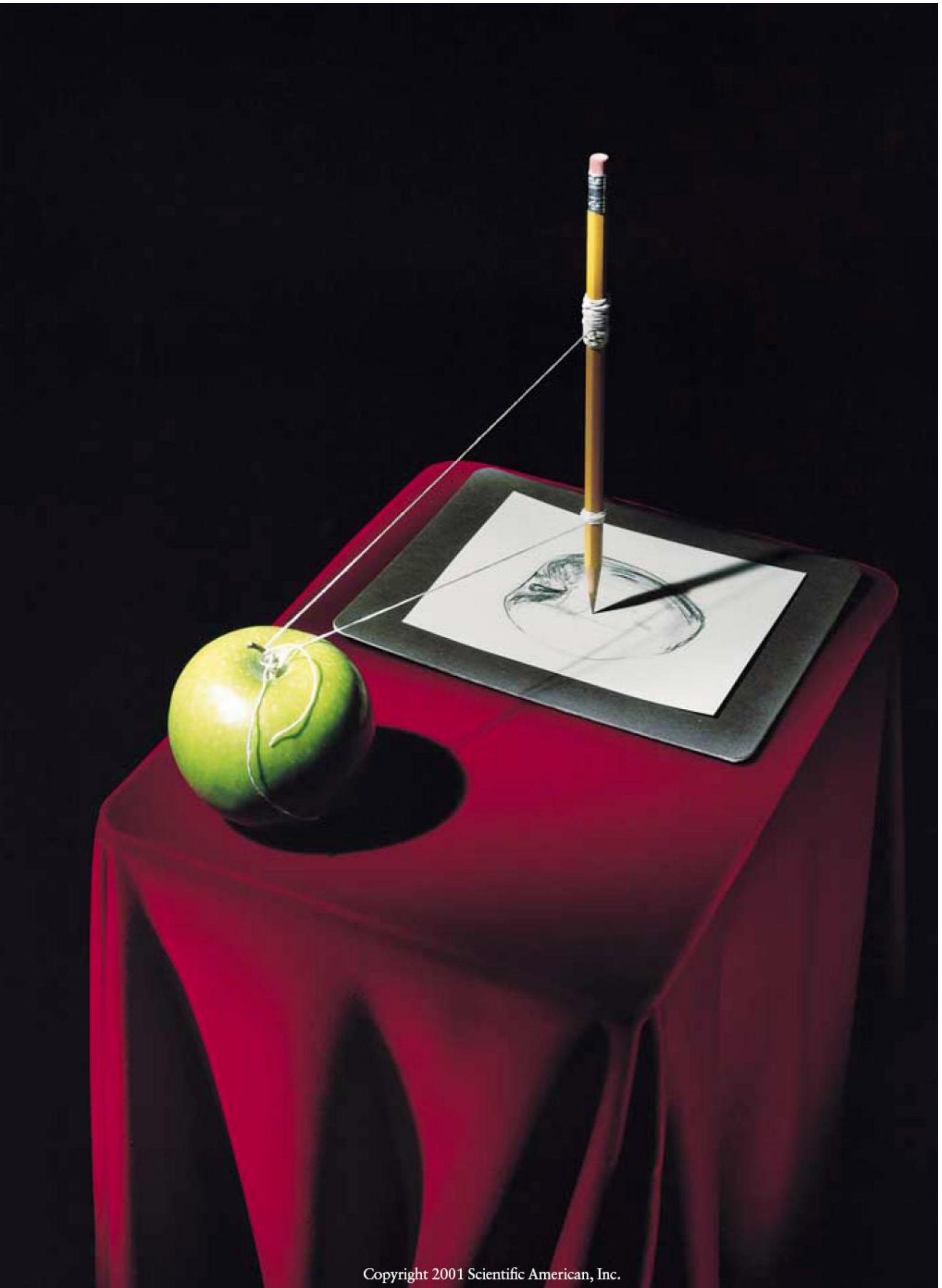

FANUC








TERMINATOR
THE SARAH CONNOR CHRONICLES
FOX



Go Forth and Replicate

Birds do it, bees do it, but could machines do it? New computer simulations suggest that the answer is yes

Apples beget apples, but can machines beget machines? Today it takes an elaborate manufacturing apparatus to build even a simple machine. Could we endow an artificial device with the ability to multiply on its own? Self-replication has long been considered one of the fundamental properties separating the living from the nonliving. Historically our limited understanding of how biological reproduction works has given it an aura of mystery and made it seem unlikely that it would ever be done by a man-made object. It is reported that when René Descartes averred to Queen Christina of Sweden that animals were just another form of mechanical automata, Her Majesty pointed to a clock and said, "See to it that it produces offspring."

The problem of machine self-replication moved from philosophy into the realm of science and engineering in the late 1940s with the work of eminent mathematician and physicist John von Neumann. Some researchers have actually constructed physical replicators. Forty years ago, for example, geneticist Lionel Penrose and his son, Roger (the famous physicist), built small assemblies of plywood that exhibited a simple form of self-replication [see "Self-Reproducing Machines," by Lionel Penrose; SCIENTIFIC AMERICAN, June 1959]. But self-replication has proved to be so difficult that most researchers study it with the conceptual tool that von Neumann developed: two-dimensional cellular automata.

Implemented on a computer, cellular automata can simulate a huge variety of self-replicators in what amount to austere universes with different laws of physics from our own. Such models free researchers from having to worry about logistical issues such as energy and physical construction so that they can focus on the fundamental questions of information flow. How is a living being able to replicate unaided, whereas mechanical objects must be constructed by humans? How does replication at the level of an organism emerge from the numerous interactions in tissues, cells and molecules? How did Darwinian evolution give rise to self-replicating organisms?

The emerging answers have inspired the development of self-repairing silicon chips [*see box on page 40*] and autocatalyzing molecules [see "Synthetic Self-Replicating Molecules," by Julius Rebek, Jr.; SCIENTIFIC AMERICAN, July 1994]. And this may be just the beginning. Researchers in the field of nanotechnology have long proposed that self-replication will be crucial to manu-

By Moshe Sipper and James A. Reggia

Photoillustrations by David Emmite

Her Majesty pointed to a clock and said, "See to it that it produces offspring."

facturing molecular-scale machines, and proponents of space exploration see a macroscopic version of the process as a way to colonize planets using in situ materials. Recent advances have given credence to these futuristic-sounding ideas. As with other scientific disciplines, including genetics, nuclear energy and chemistry, those of us who study self-replication face the twofold challenge of creating replicating machines and avoiding dystopian predictions of devices running amok. The knowledge we gain will help us separate good technologies from destructive ones.

Playing Life

SCIENCE-FICTION STORIES often depict cybernetic self-replication as a natural development of current technology, but they gloss over the profound problem it poses: how to avoid an infinite regress. A system might try to build a clone using a blueprint—that is, a self-description. Yet the self-description is part of the machine, is it not? If so, what describes the description? And what describes the description of the description? Self-replication in this case would be like asking an architect to make a perfect blueprint of his or her own studio. The blueprint would have to contain a miniature version of the blueprint, which would contain a miniature version of the blueprint and so on. Without this information, a construction crew would be unable to re-create the studio fully; there would be a blank space where the blueprint had been.

Von Neumann's great insight was an explanation of how to break out of the infinite regress. He realized that the self-description could be used in two distinct ways: first, as the instructions whose interpretation leads to the construction of an identical copy of the device; next, as data to be copied, uninterpreted, and attached to the newly created child so that it too possesses the ability to self-replicate. With this two-step process, the self-description need not contain a description of itself. In the architectural analogy, the blueprint would include a plan for building a photocopy machine. Once the new studio and the photocopier were built, the construction crew would simply run off a copy of the blueprint and put it into the new studio.

Living cells use their self-description, which biologists call the genotype, in exactly these two ways: transcription (DNA is copied mostly uninterpreted to form mRNA) and translation (mRNA is interpreted to build proteins). Von Neumann made this transcription-translation distinction several years before molecular biologists did, and his work has been crucial in understanding self-replication in nature.

To prove these ideas, von Neumann and mathematician Stanislaw M. Ulam came up with the idea of cellular automata. A cellular-automata simulation involves a chessboardlike grid of squares, or cells, each of which is either empty or occupied by one of several possible components. At discrete intervals of time, each cell looks at itself and its neighbors and decides whether to metamorphose into a different component. In making this decision, the cell follows relatively simple rules, which are the same for all cells. These rules constitute the basic physics of the cellular-automata world. All decisions and actions take place locally; cells do not know directly what is happening outside their immediate neighborhood.

The apparent simplicity of cellular automata is deceptive; it does not imply ease of design or poverty of behavior. The most famous automata, John Horton Conway's Game of Life, produces amazingly intricate patterns. Many questions about the dynamic behavior of cellular automata are formally unsolvable. To see how a pattern will unfold, you need to simulate it fully [see Mathematical Games, by Martin Gardner; SCIENTIFIC AMERICAN, October 1970 and February 1971; and "The Ultimate in Anty-Particles," by Ian Stewart, July 1994]. In its own way, a cellular-automata model can be just as complex as the real world.

Copy Machines

WITHIN CELLULAR AUTOMATA, self-replication occurs when a group of components—a "machine"—goes through a sequence of steps to construct a nearby duplicate of itself. Von Neumann's machine was based on a universal constructor, a machine that, given the appropriate instructions, could create any pattern. The constructor consisted of numerous types of components spread over tens of thousands of cells and required a book-length manuscript to be specified. It has still not been simulated in its entirety, let alone actually built, on account of its complexity. A constructor would be even more complicated in the Game of Life because the functions performed by single cells in von Neumann's model—such as transmission of signals and generation of new components—have to be performed by composite structures in Life.

Going to the other extreme, it is easy to find trivial examples of self-replication. For example, suppose a cellular automata has only one type of component, labeled +, and that each cell follows only a single rule: if exactly one of the four neighboring cells contains a +, then the cell becomes a +; otherwise it becomes vacant. With this rule, a single + grows into four more +'s, each of which grows likewise, and so forth.

Such weedlike proliferation does not shed much light on the principles of replication, because there is no significant machine. Of course, that invites the question of how you would tell a "significant" machine from a trivially prolific automata. No one has yet devised a satisfactory answer. What is clear, however, is that the replicating structure must in some sense be complex. For example, it must consist of multiple, diverse components whose interactions collectively bring about replication—the proverbial "whole must be greater than the sum of the parts." The existence of multiple distinct components permits a self-description to be stored within the replicating structure.

In the years since von Neumann's seminal work, many researchers have probed the domain between the complex and the trivial, developing replicators that require fewer components, less space or simpler rules. A major step forward was taken in 1984 when Christopher G. Langton, then at the University of Michigan, observed that looplike storage devices—which had formed modules of earlier self-replicating machines—could be programmed to replicate on their own. These devices typically consist of two pieces: the loop itself, which is a string of components that circulate around a rectangle, and a construction arm, which protrudes from a corner of the rectangle into the surrounding space. The circulating components constitute a recipe for the loop—for example, "go three squares ahead, then turn left." When this recipe reaches the construction arm, the automata rules make a copy of it. One copy continues around the loop; the other goes down the arm, where it is interpreted as instructions.

By giving up the requirement of universal construction, which was central to von Neumann's approach, Langton showed that a replicator could be constructed from just seven unique components occupying only 86 cells. Even smaller and simpler self-replicating loops have been devised by one of us (Reggia) and our colleagues [*see box on next page*]. Because they have multiple interacting components and include a self-description, they are not trivial. Intriguingly, asymmetry plays an unexpected role: the rules governing replication are often simpler when the components are not rotationally symmetric than when they are.

Emergent Replication

ALL THESE SELF-REPLICATING structures have been designed through ingenuity and much trial and error. This process is arduous and often frustrating; a small change to one of the rules results in an entirely different global behavior, most likely the disintegration of the structure in question. But recent work has gone beyond the direct-design approach. Instead of tailoring the rules to suit a particular type of structure, researchers have experimented with various sets of rules, filled the cellular-automata grid with a "primordial soup" of randomly selected components and checked whether self-replicators emerged spontaneously.

In 1997 Hui-Hsien Chou, now at Iowa State University, and Reggia noticed that as long as the initial density of the free-floating components was above a certain threshold, small self-replicating loops reliably appeared. Loops that collided underwent annihilation, so there was an ongoing process of death as well as birth. Over time, loops proliferated, grew in size and evolved through mutations triggered by debris from past collisions. Although the automata rules were deterministic, these mutations were effectively random,

THE AUTHORS

MOSHE SIPPER and *JAMES A. REGGIA* share a long-standing interest in how complex systems can self-organize. Sipper is a senior lecturer in the department of computer science at Ben-Gurion University in Israel and a visiting researcher at the Logic Systems Laboratory of the Swiss Federal Institute of Technology in Lausanne. He is interested mainly in bio-inspired computational paradigms such as evolutionary computation, self-replicating systems and cellular computing. Reggia is a professor of computer science and neurology, working in the Institute for Advanced Computer Studies at the University of Maryland. In addition to studying self-replication, he conducts research on computational models of the brain and its disorders, such as stroke.

because the system was complex and the components started in random locations.

Such loops are intended as abstract machines and not as simulacra of anything biological, but it is interesting to compare them with biomolecular structures. A loop loosely resembles circular DNA in bacteria, and the construction arm acts as the enzyme that catalyzes DNA replication. More important, replicating loops illustrate how complex global behaviors can arise from simple local interactions. For example, components move around a loop even though the rules say nothing about movement; what is actually happening is that individual cells are coming alive, dying or metamorphosing in such a way that a pattern is eliminated from one position and reconstructed elsewhere—a process that we perceive as motion. In short, cellular automata act locally but appear to think globally. Much the same is true of molecular biology.

In a recent computational experiment, Jason Lohn, now at the NASA Ames Research Center, and Reggia experimented not with different structures but with different sets of rules. Starting with an arbitrary block of four components, they found they could determine a set of rules that made the block self-replicate. They discovered these rules via a genetic algorithm, an automated process that simulates Darwinian evolution.

The most challenging aspect of this work was the definition of the so-called fitness function—the criteria by which sets of rules were judged, thus separating good solutions from bad ones and driving the evolutionary process toward rule sets that facilitated replication. You cannot simply assign high fitness to those sets of rules that cause a structure to replicate, because none of the initial rule sets is likely to allow for replication. The solution was to devise a fitness function composed of a weighted sum of three measures: a growth measure (the extent to which each component type generates an increasing supply of that component), a relative position measure (the extent to which neighboring components stay together) and a replicant measure (a function of the number of actual replicators present). With the right fitness function, evolution can turn rule sets that are sterile into ones that are fecund; the process usually takes 150 or so generations.

Self-replicating structures discovered in this fashion work in a fundamentally different way than self-replicating loops do. For example, they move and deposit copies along the way—unlike replicating loops, which are essentially static. And although these newly discovered replicators consist of multiple, locally interacting components, they do not have an identifiable self-description—there is no obvious genome. The ability to replicate without a self-description may be relevant to questions about how the earliest biological

Continued on page 43

BUILD YOUR OWN REPLICATOR

SIMULATING A SMALL self-replicating loop using an ordinary chess set is a good way to get an intuitive sense of how these systems work. This particular cellular-automata model has four different types of components: pawns, knights, bishops and rooks. The machine initially comprises four pawns, a knight and a bishop. It has two parts: the loop itself, which consists of a two-by-two square, and a construction arm, which sticks out to the right.

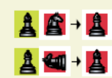
The knight and bishop represent the self-description: the knight, whose orientation is significant, determines which direction to grow, while the bishop tags along and determines how long the side of the loop should be. The pawns are fillers that define the rest of the shape of the loop, and the rook is a transient signal to guide the growth of a new construction arm.

As time progresses, the knight and bishop circulate counterclockwise around the loop. Whenever they encounter the arm, one copy goes out the arm while the original continues around the loop.

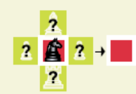
HOW TO PLAY: You will need two chessboards: one to represent the current configuration, the other to show the next configuration. For each round, look at each square of the current configuration, consult the rules and place the appropriate piece in the corresponding square on the other board. Each piece metamorphoses depending on its identity and that of the four squares immediately to the left, to the right, above and below. When you have reviewed each square and set up the next configuration, the round is over. Clear the first board and repeat. Because the rules are complicated, it takes a bit of patience at first. You can also view the simulation at lslwww.epfl.ch/chess

The direction in which a knight faces is significant. In the drawings here, we use standard chess conventions to indicate the orientation of the knight: the horse's muzzle points forward. If no rule explicitly applies, the contents of the square stay the same. Squares on the edge should be treated as if they have adjacent empty squares off the board. *—M.S. and J.A.R.*

KNIGHT



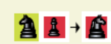
IF THERE is a bishop just behind or to the left of the knight, replace the knight with another bishop.



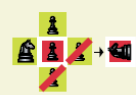
OTHERWISE, if at least one of the neighboring squares is occupied, remove the knight and leave the square empty.

PAWN

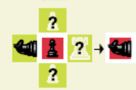
IF THERE is a neighboring knight, replace the pawn with a knight with a certain orientation, as follows:



IF A NEIGHBORING knight is facing away from the pawn, the new knight faces the opposite way.

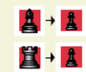


OTHERWISE, if there is exactly one neighboring pawn, the new knight faces that pawn.



OTHERWISE the new knight faces in the same direction as the neighboring knight.

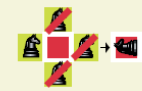
BISHOP OR ROOK



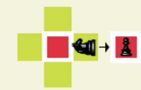
REPLACE IT with a pawn.

EMPTY SQUARE

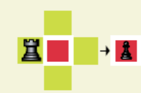
IF THERE are two neighboring knights and either faces the empty square, fill the square with a rook.



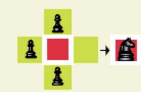
IF THERE is only one neighboring knight and it faces the square, fill the square with a knight rotated 90 degrees counterclockwise.



IF THERE is a neighboring knight and its left side faces the square, and the other neighbors are empty, fill the square with a pawn.

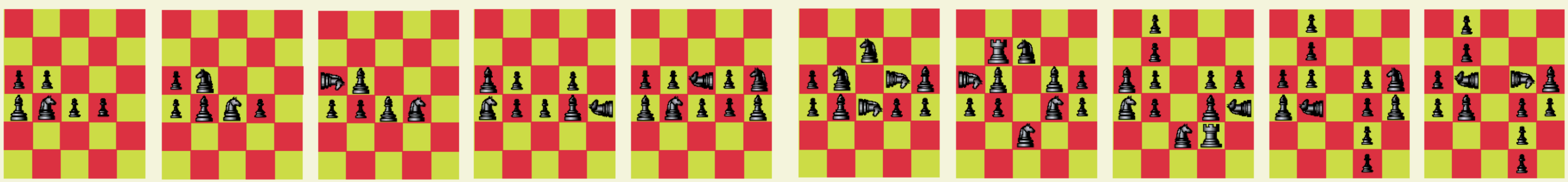


IF THERE is a neighboring rook, and the other neighbors are empty, fill the square with a pawn.



IF THERE are three neighboring pawns, fill the square with a knight facing the fourth, empty neighbor.

STAGES OF REPLICATION



INITIALLY, the self-description, or "genome"—a knight followed by a bishop—is poised at the start of the construction arm.

1 The knight and bishop move counterclockwise around the loop. A clone of the knight heads out the arm.

2 The original knight-bishop pair continues to circulate. The bishop is cloned and follows the new knight out the arm.

3 The knight triggers the formation of two corners of the child loop. The bishop tags along, completing the gene transfer.

4 The knight forges the remaining corner of the child loop. The loops are connected by the construction arm and a knight-errant.

5 The knight-errant moves up to endow the parent with a new arm. A similar process, one step delayed, begins for the child loop.

6 The knight-errant, together with the original knight-bishop pair, conjures up a rook. Meanwhile the old arm is erased.

7 The rook kills the knight and generates the new, upward arm. Another rook prepares to do the same for the child.

8 At last the two loops are separate and whole. The self-descriptions continue to circulate, but otherwise all is calm.

9 The parent prepares to give birth again. In the following step, the child too will begin to replicate.

ROBOT, HEAL THYSELF

Computers that fix themselves are the first application of artificial self-replication

LAUSANNE, SWITZERLAND—Not many researchers encourage the wanton destruction of equipment in their labs. Daniel Mange, however, likes it when visitors walk up to one of his inventions and press the button marked KILL. The lights on the panel go out; a small box full of circuitry is toast. Early in May his team unveiled its latest contraption at a science festival here—a wall-size digital clock whose components you can zap at will—and told the public: Give it your best shot. See if you can crash the system.

The goal of Mange and his team is to instill electronic circuits with the ability to take a lickin' and keep on tickin'—just like living things. Flesh-and-blood creatures might not be so good at calculating π to the millionth digit, but they can get through the day without someone pressing Ctrl-Alt-Del. Combining the precision of digital hardware with the resilience of biological wetware is a leading challenge for modern electronics.

Electronics engineers have been working on fault-tolerant circuits ever since there were electronics engineers [see "Redundancy in Computers," by William H. Pierce; SCIENTIFIC AMERICAN, February 1964]. Computer modems would still be dribbling data at 1200 baud if it weren't for error detection and correction. In many applications, simple quality-control checks, such as extra data bits, suffice. More complex systems provide entire backup computers. The space shuttle, for example, has five processors. Four of them perform the same calculations; the fifth checks whether they agree and pulls the plug on any dissenter.

The problem with these systems, though, is that they rely on centralized control. What if that control unit goes bad?

Nature has solved that problem through radical decentralization. Cells in the body are all basically identical; each takes on a specialized task, performs it autonomously and, in the event of infection or failure, commits hara-kiri so that its tasks can be taken up by new cells. These are the attributes that Mange, a professor at the Swiss Federal Institute of Technology here, and others have sought since 1993 to emulate in circuitry, as part of the "Embryonics" (embryonic electronics) project.

One of their earlier inventions, the MICTREE (microinstruction tree) artificial cell, consisted of a simple processor and four bits of data storage. The cell is contained in a plastic box roughly the size of a pack of Post-its. Electrical contacts run along the sides so that cells can be snapped together like Legos. As in cellular automata, the models used to study the theory of self-replication, the MICTREE cells are connected only to their immediate neighbors. The communication burden on each cell is thus independent of the total number of cells. The system, in other words, is easily scalable—unlike many parallel-computing architectures.

Cells follow the instructions in their "genome," a program written in a subset of the Pascal computer language. Like their biological antecedents, the cells all contain the exact same genome and execute part of it based on their position within the array, which each cell calculates relative to its neighbors. Wasteful though it may seem, this redundancy allows the array to withstand the loss of any cell. Whenever someone presses the KILL button on a cell, that cell shuts down, and its left and right neighbors become directly connected. The right neighbor recalculates its position and starts executing the deceased's program. Its tasks, in turn, are taken up by the next cell to the right, and so on, until a cell designated as a spare is pressed into service.

Writing programs for any parallel processor is tricky, but the MICTREE array requires an especially unconventional approach. Instead of giving explicit instructions, the programmer must devise simple rules out of which the desired function will emerge. Being Swiss, Mange demonstrates by building a superreliable stopwatch. Displaying minutes and seconds requires four cells in a row, one for each digit. The genome allows for two cell types: a counter from zero to nine and a counter from zero to five. An oscillator feeds one pulse per second into the rightmost cell. After 10 pulses, this cell cycles back to zero and sends a pulse to the cell on its left, and so on down the line. The watch takes up part of an array of 12 cells; when you kill one, the clock transplants itself one cell over and carries on. Obviously, though, there is a limit to its resilience: the whole thing will fail after, at most, eight kills.

The prototype MICTREE cells are hardwired, so their processing power cannot be tailored to a specific application. In a finished product, cells would instead be implemented on a field-programmable gate array, a grid of electronic components that can be reconfigured on the fly [see "Configurable Computing," by John Villasenor and William H. Mangione-Smith; SCIENTIFIC AMERICAN, June 1997]. Mange's team is now custom-designing a gate array, known as MUXTREE (multiplexer tree), that is optimized for artificial cells. In the biological metaphor, the components of this array are the "molecules" that constitute a cell. Each consists of a logic gate, a data bit and a string of configuration bits that determines the function of this gate.

Building a cell out of such molecules offers not only flexibility but also extra endurance. Each molecule contains two copies of the gate and three of the storage bit. If the two gates ever give different results, the molecule kills itself for the greater good of the cell. As a last gasp, the molecule sends its data bit (preserved by the triplicate storage) and configuration to its right neighbor, which does the same, and the process continues until the rightmost molecule transfers its data to a spare. This second level of fault tolerance prevents a single error from wiping out an entire cell.

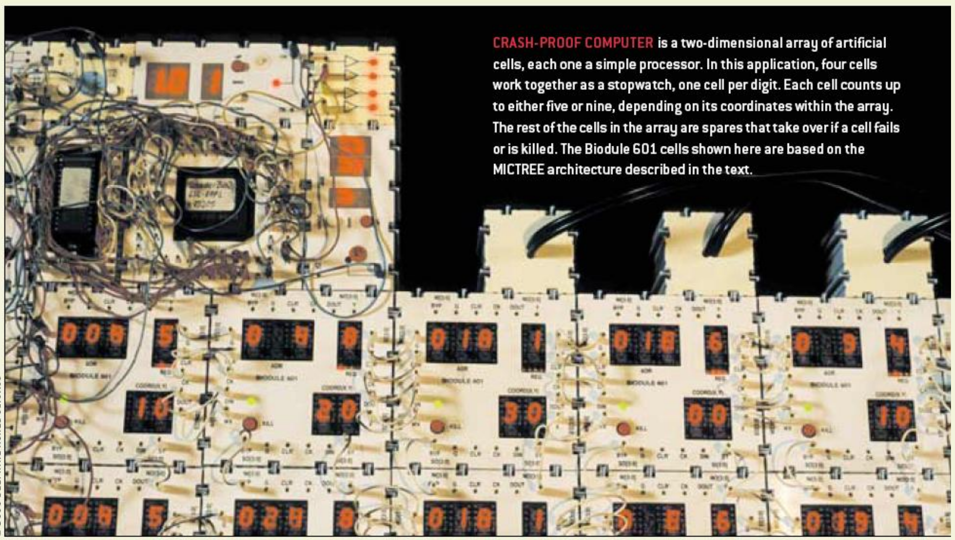
A total of 2,000 molecules, divided into four 20-by-25 cells, make up the BioWall—the giant digital clock that Mange's team has just put on display. Each molecule is enclosed in a small box and includes a KILL button and an LED display. Some molecules are configured to perform computations; others serve as pixels in the clock display. Making liberal use of the KILL buttons, I did my utmost to crash the system, something I'm usually quite good at. But the plucky clock just wouldn't submit. The clock display did start to look funny—numerals bent over as their pixels shifted to the right—but at least it was still legible, unlike most faulty electronic signs.

That said, the system did suffer from display glitches, which Mange attributed mainly to timing problems. Although the processing power is decentralized, the cells still rely on a central oscillator to coordinate their communications; sometimes they fall out of sync. Another Embryonics team, led by Andy Tyrrell of the University of York in England, has been studying making the cells asynchronous, like their biological counterparts. Cells would generate handshaking signals to orchestrate data transfers. The present system is also unable to catch certain types of error, including damaged configuration strings. Tyrrell's team has proposed adding watchdog molecules—an immune system—that would monitor the configurations (and one another) for defects.

Although these systems demand an awful lot of overhead, so do other fault-tolerance technologies. "While Embryonics appears to be heavy on redundancy, it actually is not that bad when compared to other systems," Tyrrell argues. Moreover, MUXTREE should be easier to scale down to the nano level; the "molecules" are simple enough to really be molecules. Says Mange, "We are preparing for the situation where electronics will be at the same scale as biology."

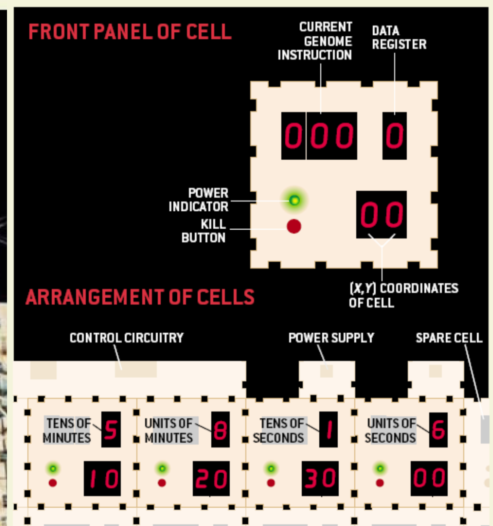
On a philosophical level, Embryonics comes very close to the dream of building a self-replicating machine. It may not be quite as dramatic as a robot that can go down to Radio Shack, pull parts off the racks, and take them home to resolder a connection or build a loving mate. But the effect is much the same. Letting machines determine their own destiny—whether reconfiguring themselves on a silicon chip or reprogramming themselves using a neural network or genetic algorithm—sounds scary, but perhaps we should be gratified that machines are becoming more like us: imperfect, fallible but stubbornly resourceful.

—*George Musser, imperfect but resourceful staff editor and writer*



CRASH-PROOF COMPUTER is a two-dimensional array of artificial cells, each one a simple processor. In this application, four cells work together as a stopwatch, one cell per digit. Each cell counts up to either five or nine, depending on its coordinates within the array. The rest of the cells in the array are spares that take over if a cell fails or is killed. The Biodule 601 cells shown here are based on the MICTREE architecture described in the text.

© 1999 DELPHINE AURES *Eurelios*



Continued from page 39

replicators originated. In a sense, researchers are seeing a continuum between nonliving and living structures.

Many researchers have tried other computational models besides the traditional cellular automata. In asynchronous cellular automata, cells are not updated in concert; in nonuniform cellular automata, the rules can vary from cell to cell. Another approach altogether is Core War [see Computer Recreations, by A. K. Dewdney; SCIENTIFIC AMERICAN, May 1984] and its successors, such as ecologist Thomas S. Ray's Tierra system. In these simulations the "organisms" are computer programs that vie for processor time and memory. Ray has observed the emergence of "parasites" that co-opt the self-replication code of other organisms.

Getting Real

SO WHAT GOOD are these machines? Von Neumann's universal constructor can compute in addition to replicating, but it is an impractical beast. A major advance has been the development of simple yet useful replicators. In 1995 Gianluca Tempesti of the Swiss Federal Institute of Technology in Lausanne simplified the loop self-description so it could be interlaced with a small program—in this case, one that would spell the acronym of his lab, "LSL." His insight was to create automata rules that allow loops to replicate in two stages. First the loop, like Langton's loop, makes a copy of itself. Once finished, the daughter loop sends a signal back to its parent, at which point the parent sends the instructions for writing out the letters.

Drawing letters was just a demonstration. The following year Jean-Yves Perrier, Jacques Zahnd and one of us (Sipper) designed a self-replicating loop with universal computational capabilities—that is, with the computational power of a universal Turing machine, a highly simplified but fully capable computer. This loop has two "tapes," or long strings of components, one for the program and the other for data. The loops can execute an arbitrary program in addition to self-replicating. In a sense, they are as complex as the computer that simulates them. Their main limitation is that the program is copied unchanged from parent to child, so that all loops carry out the same set of instructions.

In 1998 Chou and Reggia swept away this limitation. They showed how self-replicating loops carrying distinct information, rather than a cloned program, can be used to solve a problem known as satisfiability. The loops can be used to determine whether the variables in a logical expression can be assigned values such that the entire expression evaluates to "true." This problem is NP-complete—in other words, it belongs to the family of nasty puzzles, including the famous traveling-salesman problem, for which there is no known efficient solution. In Chou and Reggia's cellular-automata universe, each replicator received a different partial solution. During replication, the solutions mutated, and replicators with promising solutions were allowed to proliferate while those with failed solutions died out.

Although various teams have created cellular automata in electronic hardware, such systems are probably too wasteful for practical applications; automata were never really intended to be implemented directly. Their purpose is to illuminate the underlying principles of replication and, by doing so, inspire more concrete efforts. The loops provide a new paradigm for designing a parallel computer from either transistors or chemicals [see "Computing with DNA," by Leonard M. Adleman; SCIENTIFIC AMERICAN, August 1998].

In 1980 a NASA team led by Robert Freitas, Jr., proposed planting a factory on the moon that would replicate itself, using local lunar materials, to populate a large area exponentially. Indeed, a similar probe could colonize the entire galaxy, as physicist Frank J. Tipler of Tulane University has argued. In the nearer term, computer scientists and engineers have experimented with the automated design of robots [see "Dawn of a New Species?" by George Musser; SCIENTIFIC AMERICAN, November 2000]. Although these systems are not truly self-replicating—the offspring are much simpler than the parent—they are a first step toward fulfilling the queen of Sweden's request.

In a sense, researchers are seeing a continuum between nonliving and living structures.

Should physical self-replicating machines become practical, they and related technologies will raise difficult issues, including the *Terminator* film scenario in which artificial creatures outcompete natural ones. We prefer the more optimistic, and more probable, scenario that replicators will be harnessed to the benefit of humanity [see "Will Robots Inherit the Earth?" by Marvin Minsky; SCIENTIFIC AMERICAN, October 1994]. The key will be taking the advice of 14th-century English philosopher William of Ockham: *entia non sunt multiplicanda praeter necessitatem*—entities are not to be multiplied beyond necessity. SA

MORE TO EXPLORE

Simple Systems That Exhibit Self-Directed Replication. J. Reggia, S. Armentrout, H. Chou and Y. Peng in *Science*, Vol. 259, No. 5099, pages 1282–1287; February 26, 1993.

Emergence of Self-Replicating Structures in a Cellular Automata Space. H. Chou and J. Reggia in *Physica D*, Vol. 110, Nos. 3–4, pages 252–272; December 15, 1997.

Special Issue: Von Neumann's Legacy: On Self-Replication. Edited by M. Sipper, G. Tempesti, D. Mange and E. Sanchez in *Artificial Life*, Vol. 4, No. 3; Summer 1998.

Towards Robust Integrated Circuits: The Embryonics Approach. D. Mange, M. Sipper, A. Stauffer and G. Tempesti in *Proceedings of the IEEE*, Vol. 88, No. 4, pages 516–541; April 2000.

Moshe Sipper's Web page on artificial self-replication is at **lslwww.epfl.ch/~moshes/selfrep/**

Animations of self-replicating loops can be found at **necsi.org/postdocs/sayama/sdsr/java/**

For John von Neumann's universal constructor, see **alife.santafe.edu/alife/topics/jvn/jvn.html**

John von Neumann Institut für Computing

John von Neumann Institute for Computing (NIC)

The John von Neumann Institute for Computing (NIC) is a joint foundation of Forschungszentrum Jülich and Deutsches Elektronen-Synchrotron DESY to support supercomputer-aided scientific research and development. Since April 2006, the GSI Helmholtzzentrum für Schwerionenforschung joined NIC as a contract partner. NIC takes over the functions and tasks of the High Performance Computer Centre (HLRZ) established in 1987 and continues this centre's successful work in the field of supercomputing and its applications.

- **Provision of supercomputer capacity** for projects in science, research and industry in the fields of modelling and computer simulation including their methods.
  Research proposals can be submitted by German scientists and by partners in the EU projects DEISA and I3HP.
  There is also an Offer to the New Member States and candidate countries of the European Union.

  The supercomputers with the required information technology infrastructure (software, data storage, networks) are operated by the Jülich Supercomputing Centre (JSC) in Jülich and by the Centre for Parallel Computing at DESY in Zeuthen.
- **Supercomputer-oriented research and development** in selected fields of physics and other sciences, especially in elementary-particle physics, by research groups for supercomputing applications.
- **Education and training in the fields of scientific computing** by symposia, workshops, summer schools, seminars, courses, and guest programs for scientists and students.

S.Hoefler-Thierfeldt@fz-juelich.de, 01-Jul-2008
URL: <http://www.fz-juelich.de/nic/Allgemeines/Allgemeines-e.html>

HOME
ABOUT
MEMBERSHIP
PUBLICATIONS
MEETINGS
COMMUNITIES
AWARDS
- INFORMS Prizes and Awards
- INFORMS Subdivision Prizes
- INFORMS Fellows
- INFORMS Scholarships
- Non-INFORMS Prizes
EDUCATION
CAREER
VOLUNTEERS
RESOURCES

YOU ARE HERE: AWARDS > - INFORMS Prizes and Awards > John von Neumann Theory Prize

John von Neumann Theory Prize

A prize is awarded annually to a scholar (or scholars in the case of joint work) who has made fundamental, sustained contributions to theory in operations research and the management sciences. The award is given each year at the National Meeting if there is a suitable recipient. Although the prize is normally given to a single individual, in the case of accumulated joint work, the recipients can be multiple individuals. The award is $5,000, a medallion and a citation.

- **Who Was John von Neumann?**
  Read about the life and legacy of John von Neumann
- **Application Process**
  View information about eligibility, procedures and deadlines
- **Past Winners**
  View information about all past winners of this prize

Most Recent Winner

2008: **Frank P. Kelly**

Return to the INFORMS Awards Page

ANNOUNCEMENTS

INFORMS Prizes and Awards: 2008 Winners

Subdivision Prizes and Awards: 2008 Winners

INFORMS Fellows: Class of 2008

Recognition of INFORMS members: View the news

What's New: View new information about prizes and awards

E-mail:
titkarsag@njszt.hu

Neumann János Számítógép-tudományi Társaság

H-1054 Budapest, Báthori u. 16. Tel.: (+36-1) 472-2730
ECDL Iroda: 472-2710 F.: 472-2728

Introducing the John von Neumann Computer Society

Ability, pride and creativity of our compatriots are for Hungary the fundament of progress and the only spring-board into the future.
(Count István Széchenyi 1842)

Fields of activity:

As a significant professional body and learned society in the Hungarian IT community, the John von Neumann Computer Society (NJSZT) is dedicated to preserving values that can be included in today's knowledge-based society as well as to setting new directions that meet the requirements of the age and to actively forming the IS world of the future. The primal activities of our Society are IT support, ECDL (European Computer Driving Licence) Hungary, Hungarian Smart Card Forum, Organization of International and National Conferences.

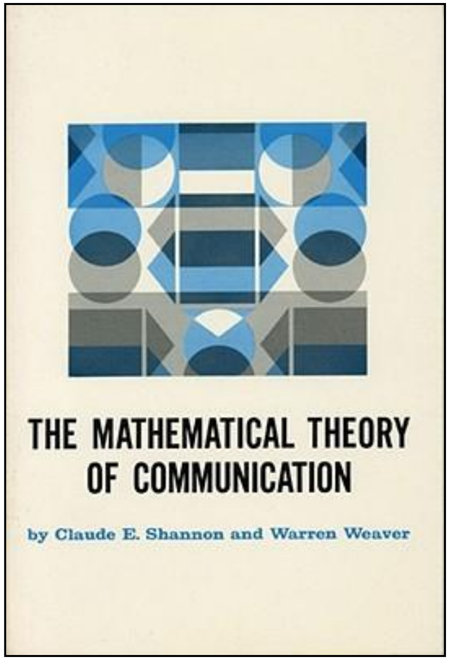
John von Neumann Computer Society
Neumann János Számítógép-tudományi Társaság

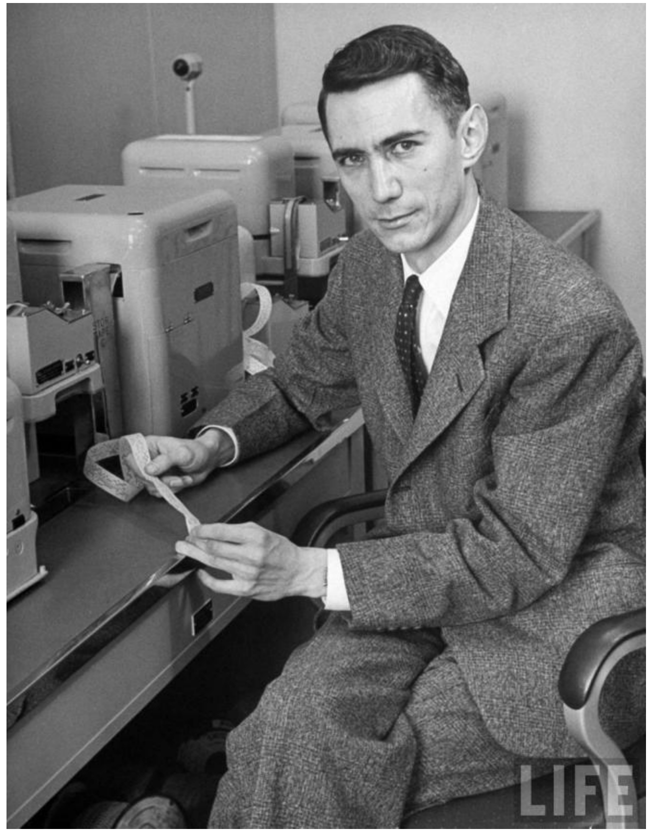
Address :	**Báthori u. 16. 1054 Budapest Hungary**
Mailing address :	**P.O.B. 201 1364 Budapest Hungary**
Telephone :	**(36-1) 472-2730**
Fax. :	**(36-1) 472-2739**
E-mail :	secretariat@njszt.hu
WWW :	www.njszt.hu
President :	**Prof. Gábor PÉCELI**
Telephone :	**(36-1) 463-2057**
Fax.:	**(36-1) 463-3580**
E-mail:	peceli@mit.bme.hu

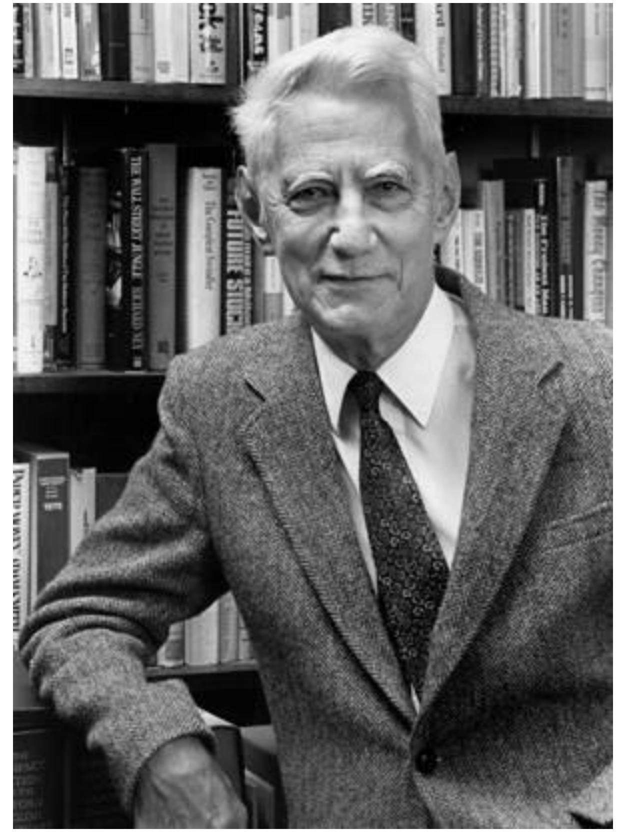
Historical Perspectives

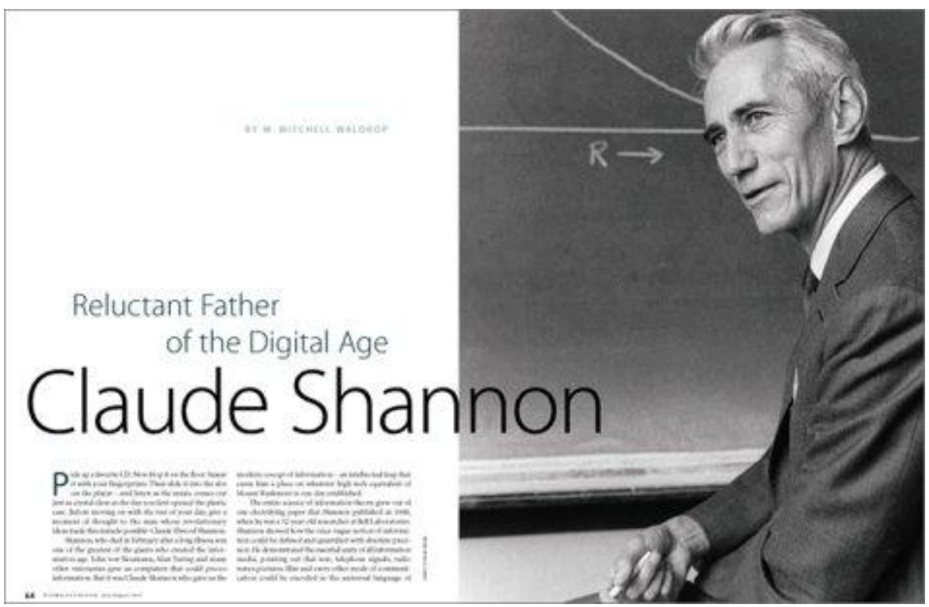
Claude Shannon (1916-2001)

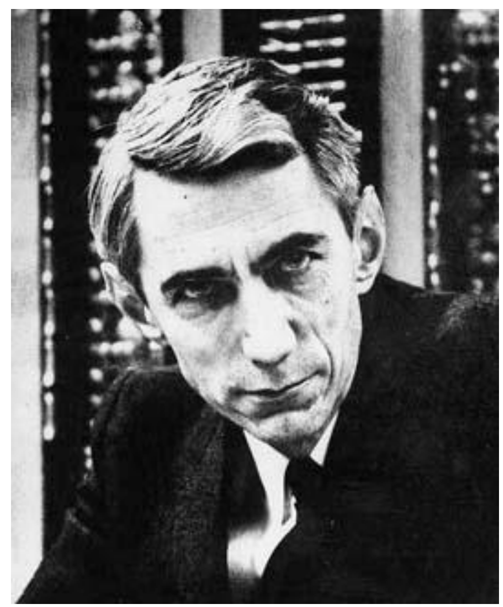
- Invented electrical digital circuits (1937)
- Founded information theory (1948)
- Introduced sampling theory, coined term “bit”
- Contributed to genetics, cryptography
- Joined Institute for Advanced Study (1940)
  Influenced by Turing, von Neumann, Einstein
- Originated information entropy, Nyquist–Shannon, sampling theorem, Shannon-Hartley theorem, Shannon switching game, Shannon-Fano coding, Shannon’s source coding theorem, Shannon limit, Shannon decomposition / expansion, Shannon #
- Other hobbies & inventions: juggling, unicycling, computer chess, rockets, motorized pogo stick, flame-throwers, Rubik's cube solver, wearable computer, mathematical gambling, stock markets
- “AT&T Shannon Labs”

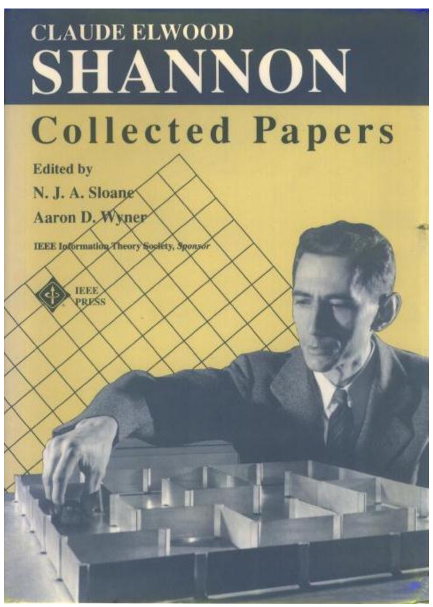



LIFE




BY M. MITCHELL WALDROP
Reluctant Father
of the Digital Age
Claude Shannon
R →




CLAUDE ELWOOD
SHANNON
Collected Papers
Edited by
N. J. A. Sloane
Aaron D. Wyner
IEEE Information Theory Society, Sponsor
IEEE
PRESS

A SYMBOLIC ANALYSIS OF RELAY AND SWITCHING CIRCUITS

by

Claude Elwood Shannon
B.S., University of Michigan
1936

Submitted in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
from the
Massachusetts Institute of Technology
1940

Signature of Author____________________________

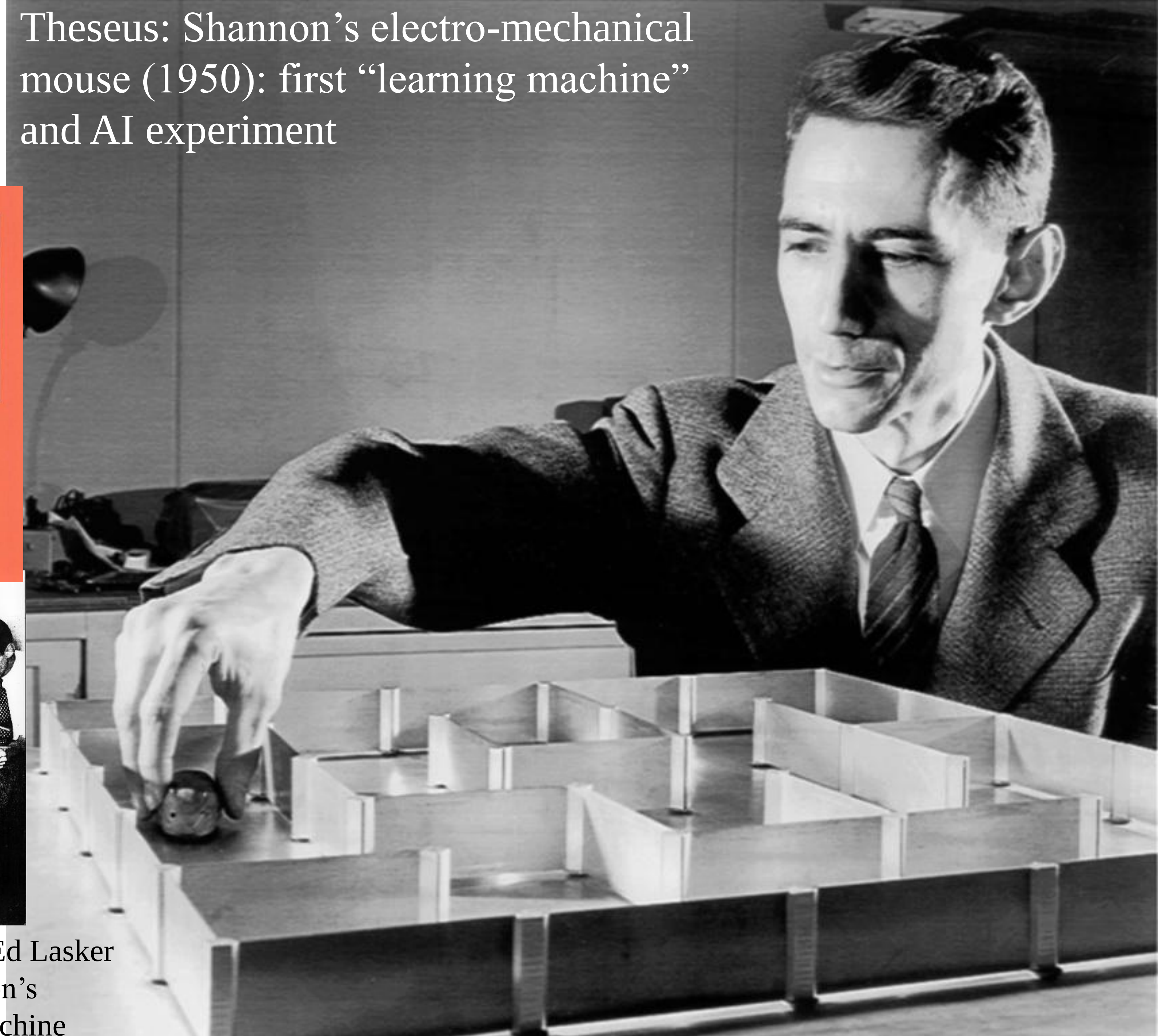
Department of Electrical Engineering, August 10, 1937

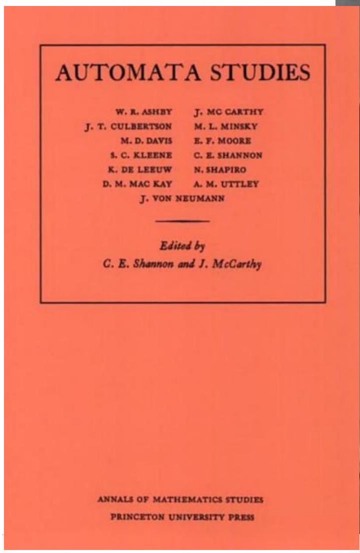
Signature of Professor
in Charge of Research____________________________

Signature of Chairman of Department
Committee on Graduate Students____________________

TABLE OF CONTENTS

Theseus: Shannon’s electro-mechanical mouse (1950): first “learning machine” and AI experiment





AUTOMATA STUDIES

W. R. ASHBY — J. MC CARTHY
J. T. CULBERTSON — M. L. MINSKY
M. D. DAVIS — E. F. MOORE
S. C. KLEENE — C. E. SHANNON
K. DE LEEUW — N. SHAPIRO
D. M. MAC KAY — A. M. UTTLEY
J. VON NEUMANN

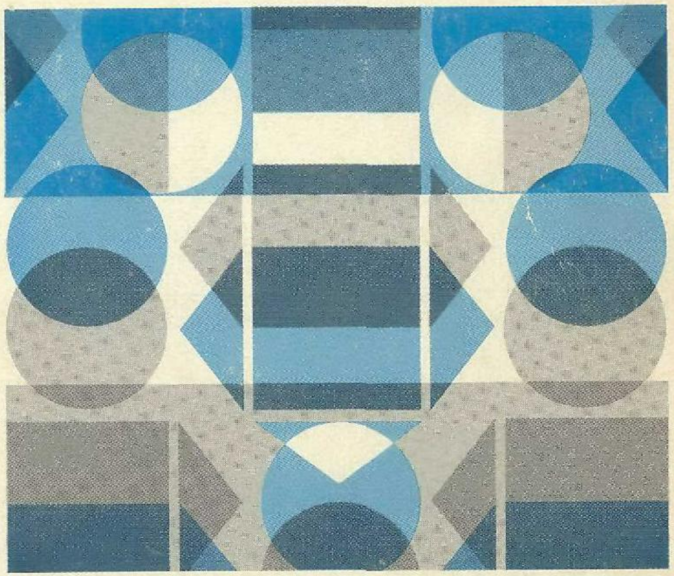
Edited by
C. E. Shannon and J. McCarthy

ANNALS OF MATHEMATICS STUDIES
PRINCETON UNIVERSITY PRESS

Chess champion Ed Lasker looking at Shannon’s chess-playing machine

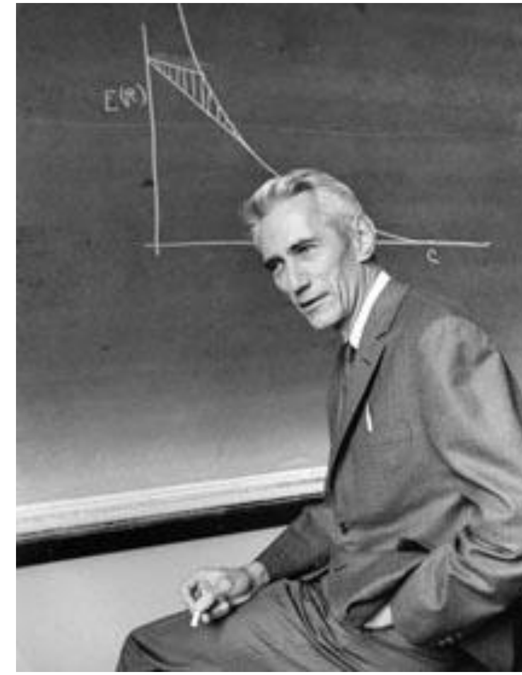
Shannon's home study room

Shannon's On/Off machine



THE MATHEMATICAL THEORY OF COMMUNICATION

by Claude E. Shannon and Warren Weaver



Eighth paperback printing, 1980

Originally published in a clothbound edition, 1949.

Manufactured in the United States of America.
Library of Congress Catalog Card No. 49-11922.

ISBN 0-252-72548-4

Introduction

The recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning;* that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

[1] Nyquist, H., "Certain Factors Affecting Telegraph Speed," *Bell System Technical Journal,* April 1924, p. 324; "Certain Topics in Telegraph Transmission Theory," *A.I.E.E. Trans.,* v. 47, April 1928, p. 617.
[2] Hartley, R. V. L., "Transmission of Information," *Bell System Technical Journal,* July 1928, p. 535.

If the number of messages in the set is finite then this number or any monotonic function of this number can be regarded as a measure of the information produced when one message is chosen from the set, all choices being equally likely. As was pointed out by Hartley the most natural choice is the logarithmic function. Although this definition must be generalized considerably when we consider the influence of the statistics of the message and when we have a continuous range of messages, we will in all cases use an essentially logarithmic measure.

The logarithmic measure is more convenient for various reasons:

1. It is practically more useful. Parameters of engineering importance such as time, bandwidth, number of relays, etc., tend to vary linearly with the logarithm of the number of possibilities. For example, adding one relay to a group doubles the number of possible states of the relays. It adds 1 to the base 2 logarithm of this number. Doubling the time roughly squares the number of possible messages, or doubles the logarithm, etc.

2. It is nearer to our intuitive feeling as to the proper measure. This is closely related to (1) since we intuitively measure entities by linear comparison with common standards. One feels, for example, that two punched cards should have twice the capacity of one for information storage, and two identical channels twice the capacity of one for transmitting information.

3. It is mathematically more suitable. Many of the limiting operations are simple in terms of the logarithm but would require clumsy restatement in terms of the number of possibilities.

The choice of a logarithmic base corresponds to the choice of a unit for measuring information. If the base 2 is used the resulting units may be called binary digits, or more briefly *bits,* a word suggested by J. W. Tukey. A device with two stable positions, such as a relay or a flip-flop circuit, can store one bit of information. N such devices can store N bits, since the total number of possible states is 2^N and $\log_2 2^N = N$. If the base 10 is used the units may be called decimal digits. Since

$$\begin{aligned}\log_2 M &= \log_{10} M/\log_{10} 2\\ &= 3.32 \log_{10} M,\end{aligned}$$

I

Discrete Noiseless Systems

1. The Discrete Noiseless Channel

Teletype and telegraphy are two simple examples of a discrete channel for transmitting information. Generally, a discrete channel will mean a system whereby a sequence of choices from a finite set of elementary symbols $S_1 \cdots S_n$ can be transmitted from one point to another. Each of the symbols S_i is assumed to have a certain duration in time t_i seconds (not necessarily the same for different S_i, for example the dots and dashes in telegraphy). It is not required that all possible sequences of the S_i be capable of transmission on the system; certain sequences only may be allowed. These will be possible signals for the channel. Thus in telegraphy suppose the symbols are: (1) A dot, consisting of line closure for a unit of time and then line open for a unit of time; (2) A dash, consisting of three time units of closure and one unit open; (3) A letter space consisting of, say, three units of line open; (4) A word space of six units of line open. We might place the restriction on allowable sequences that no spaces follow each other (for if two letter spaces are adjacent, they are identical with a word space). The question we now consider is how one can measure the capacity of such a channel to transmit information.

In the teletype case where all symbols are of the same duration, and any sequence of the 32 symbols is allowed, the answer is easy. Each symbol represents five bits of information. If the system

transmits n symbols per second it is natural to say that the channel has a capacity of $5n$ bits per second. This does not mean that the teletype channel will always be transmitting information at this rate — this is the maximum possible rate and whether or not the actual rate reaches this maximum depends on the source of information which feeds the channel, as will appear later.

In the more general case with different lengths of symbols and constraints on the allowed sequences, we make the following definition: The capacity C of a discrete channel is given by

$$C = \lim_{T \to \infty} \frac{\log N(T)}{T}$$

where $N(T)$ is the number of allowed signals of duration T.

It is easily seen that in the teletype case this reduces to the previous result. It can be shown that the limit in question will exist as a finite number in most cases of interest. Suppose all sequences of the symbols $S_1, \cdots, S_n$ are allowed and these symbols have durations $t_1, \cdots, t_n$. What is the channel capacity? If $N(t)$ represents the number of sequences of duration t we have

$$N(t) = N(t - t_1) + N(t - t_2) + \cdots + N(t - t_n).$$

The total number is equal to the sum of the numbers of sequences ending in $S_1, S_2, \cdots, S_n$ and these are $N(t - t_1)$, $N(t - t_2)$, $\cdots, N(t - t_n)$, respectively. According to a well-known result in finite differences, $N(t)$ is the asymptotic for large t to AX_0^t where A is constant and X_0 is the largest real solution of the characteristic equation:

$$X^{-t_1} + X^{-t_2} + \cdots + X^{-t_n} = 1$$

and therefore

$$C = \lim_{T \to \infty} \frac{\log AX_0^T}{T} = \log X_0.$$

In case there are restrictions on allowed sequences we may still often obtain a difference equation of this type and find C from the characteristic equation. In the telegraphy case mentioned above

$$N(t) = N(t-2) + N(t-4) + N(t-5) + N(t-7) + N(t-8) + N(t-10)$$

a decimal digit is about $3\frac{1}{3}$ bits. A digit wheel on a desk computing machine has ten stable positions and therefore has a storage capacity of one decimal digit. In analytical work where integration and differentiation are involved the base e is sometimes useful. The resulting units of information will be called natural units. Change from the base a to base b merely requires multiplication by $\log_b a$.

By a communication system we will mean a system of the type indicated schematically in Fig. 1. It consists of essentially five parts:

1. An *information source* which produces a message or sequence of messages to be communicated to the receiving terminal. The message may be of various types: (a) A sequence of letters as in a telegraph or teletype system; (b) A single function of time $f(t)$ as in radio or telephony; (c) A function of time and other variables as in black and white television — here the message may be thought of as a function $f(x, y, t)$ of two space coordinates and time, the light intensity at point (x, y) and time t on a pickup tube plate; (d) Two or more functions of time, say $f(t)$, $g(t)$, $h(t)$ — this is the case in "three-dimensional" sound transmission or if the system is intended to service several individual channels in multiplex; (e) Several functions of several variables — in color television the message consists of three functions $f(x, y, t)$, $g(x, y, t)$, $h(x, y, t)$ defined in a three-dimensional continuum — we may also think of these three functions as components of a vector field defined in the region — similarly, several black and white television sources would produce "messages" consisting of a number of functions of three variables; (f) Various combinations also occur, for example in television with an associated audio channel.

2. A *transmitter* which operates on the message in some way to produce a signal suitable for transmission over the channel. In telephony this operation consists merely of changing sound pressure into a proportional electrical current. In telegraphy we have an encoding operation which produces a sequence of dots, dashes and spaces on the channel corresponding to the message. In a multiplex PCM system the different speech functions must be sampled, compressed, quantized and encoded, and finally interleaved properly to construct the signal. Vocoder systems, television and frequency modulation are other examples of complex operations applied to the message to obtain the signal.

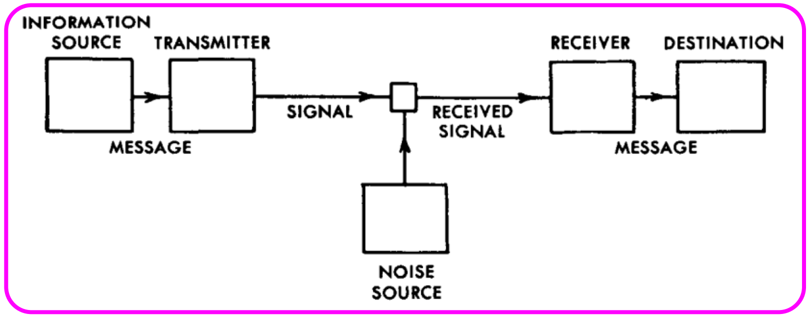


Fig. 1. — Schematic diagram of a general communication system.

3. The *channel* is merely the medium used to transmit the signal from transmitter to receiver. It may be a pair of wires, a coaxial cable, a band of radio frequencies, a beam of light, etc. During transmission, or at one of the terminals, the signal may be perturbed by noise. This is indicated schematically in Fig. 1 by the noise source acting on the transmitted signal to produce the received signal.

4. The *receiver* ordinarily performs the inverse operation of that done by the transmitter, reconstructing the message from the signal.

5. The *destination* is the person (or thing) for whom the message is intended.

We wish to consider certain general problems involving communication systems. To do this it is first necessary to represent the various elements involved as mathematical entities, suitably idealized from their physical counterparts. We may roughly classify communication systems into three main categories: discrete, continuous and mixed. By a discrete system we will mean one in which both the message and the signal are a sequence of discrete symbols. A typical case is telegraphy where the message is a sequence of letters and the signal a sequence of dots, dashes and spaces. A continuous system is one in which the

Suppose we have a set of possible events whose probabilities of occurrence are $p_1, p_2, \cdots, p_n$. These probabilities are known but that is all we know concerning which event will occur. Can we find a measure of how much "choice" is involved in the selection of the event or of how uncertain we are of the outcome?

If there is such a measure, say $H(p_1, p_2, \cdots, p_n)$, it is reasonable to require of it the following properties:

1. H should be continuous in the p_i.
2. If all the p_i are equal, $p_i = \frac{1}{n}$, then H should be a monotonic increasing function of n. With equally likely events there is more choice, or uncertainty, when there are more possible events.
3. If a choice be broken down into two successive choices, the original H should be the weighted sum of the individual values of H. The meaning of this is illustrated in Fig. 6. At the left we have three possibilities $p_1 = \frac{1}{2}$, $p_2 = \frac{1}{3}$, $p_3 = \frac{1}{6}$. On the right we first choose between two possibilities each with probability $\frac{1}{2}$, and if the second occurs make another choice with probabilities $\frac{2}{3}$, $\frac{1}{3}$. The final results have the same probabilities as before. We require, in this special case, that

$$H(\tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{6}) = H(\tfrac{1}{2}, \tfrac{1}{2}) + \tfrac{1}{2} H(\tfrac{2}{3}, \tfrac{1}{3}).$$

The coefficient $\frac{1}{2}$ is the weighting factor introduced because this second choice only occurs half the time.

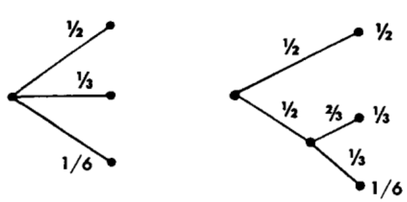


Fig. 6. — Decomposition of a choice from three possibilities.

In Appendix 2, the following result is established:

Theorem 2: The only H satisfying the three above assumptions is of the form:

$$H = -K \sum_{i=1}^{n} p_i \log p_i$$

where K is a positive constant.

This theorem, and the assumptions required for its proof, are in no way necessary for the present theory. It is given chiefly to lend a certain plausibility to some of our later definitions. The real justification of these definitions, however, will reside in their implications.

Quantities of the form $H = -\Sigma\, p_i \log p_i$ (the constant K merely amounts to a choice of a unit of measure) play a central role in information theory as measures of information, choice and uncertainty. The form of H will be recognized as that of entropy

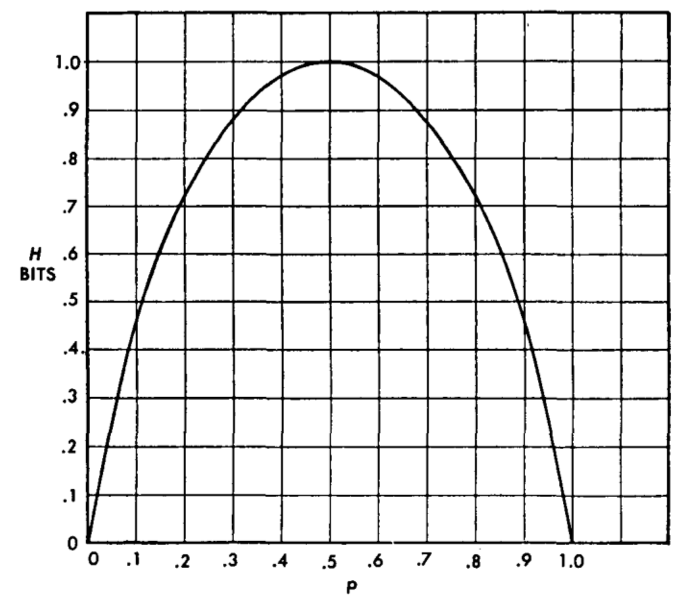


Fig. 7. — Entropy in the case of two possibilities with probabilities p and $(1-p)$.

as defined in certain formulations of statistical mechanics[8] where p_i is the probability of a system being in cell i of its phase space.

[8] See, for example, R. C. Tolman, *Principles of Statistical Mechanics*, Oxford, Clarendon, 1938.

quence of symbols x_i; and let β be the state of the transducer, which produces, in its output, blocks of symbols y_j. The combined system can be represented by the "product state space" of pairs (α, β). Two points in the space (α_1, β_1) and (α_2, β_2), are connected by a line if α_1 can produce an x which changes β_1 to β_2, and this line is given the probability of that x in this case. The line is labeled with the block of y_1 symbols produced by the transducer. The entropy of the output can be calculated as the weighted sum over the states. If we sum first on β each resulting term is less than or equal to the corresponding term for α, hence the entropy is not increased. If the transducer is non-singular let its output be connected to the inverse transducer. If H'_1, H'_2 and H'_3 are the output entropies of the source, the first and second transducers respectively, then $H'_1 \geq H'_2 \geq H'_3 = H'_1$ and therefore $H'_1 = H'_2$.

Suppose we have a system of constraints on possible sequences of the type which can be represented by a linear graph as in Fig. 2. If probabilities $p_{ij}^{(s)}$ were assigned to the various lines connecting state i to state j this would become a source. There is one particular assignment which maximizes the resulting entropy (see Appendix 4).

Theorem 8: Let the system of constraints considered as a channel have a capacity $C = \log W$. If we assign

$$p_{ij}^{(s)} = \frac{B_j}{B_i} W^{-l_{ij}^{(s)}}$$

where $l_{ij}^{(s)}$ is the duration of the s^{th} symbol leading from state i to state j and the B_i satisfy

$$B_i = \sum_{s,j} B_j W^{-l_{ij}^{(s)}}$$

then H is maximized and equal to C.

By proper assignment of the transition probabilities the entropy of symbols on a channel can be maximized at the channel capacity.

9. The Fundamental Theorem for a Noiseless Channel

We will now justify our interpretation of H as the rate of generating information by proving that H determines the channel capacity required with most efficient coding.

Theorem 9: Let a source have entropy H (bits per symbol) and a channel have a capacity C (bits per second). Then it is possible to encode the output of the source in such a way as to transmit at the average rate $\frac{C}{H} - \epsilon$ symbols per second over the channel where ϵ is arbitrarily small. It is not possible to transmit at an average rate greater than $\frac{C}{H}$.

The converse part of the theorem, that $\frac{C}{H}$ cannot be exceeded, may be proved by noting that the entropy of the channel input per second is equal to that of the source, since the transmitter must be non-singular, and also this entropy cannot exceed the channel capacity. Hence $H' \leq C$ and the number of symbols per second $= H'/H \leq C/H$.

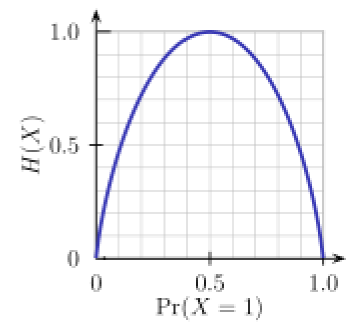
The first part of the theorem will be proved in two different ways. The first method is to consider the set of all sequences of N symbols produced by the source. For N large we can divide these into two groups, one containing less than $2^{(H+\eta)N}$ members and the second containing less than 2^{RN} members (where R is the logarithm of the number of different symbols) and having a total probability less than μ. As N increases η and μ approach zero. The number of signals of duration T in the channel is greater than $2^{(C-\theta)T}$ with θ small when T is large. If we choose

$$T = \left(\frac{H}{C} + \lambda\right) N$$

then there will be a sufficient number of sequences of channel symbols for the high probability group when N and T are sufficiently large (however small λ) and also some additional ones. The high probability group is coded in an arbitrary one-to-one way into this set. The remaining sequences are represented by larger sequences, starting and ending with one of the sequences not used for the high probability group. This special sequence acts as a start and stop signal for a different code. In between a sufficient time is allowed to give enough different sequences for all the low probability messages. This will require

Entropy and Randomness

- Entropy measures the expected "uncertainly" (or "surprise") associated with a random variable.
- Entropy quantifies the "information content" and represents a lower bound on the best possible lossless compression.
- Ex: a random fair coin has entropy of 1 bit.
  A biased coin has lower entropy than fair coin.
  A two-headed coin has zero entropy.



- The string 00000000000000… has zero entropy.
- English text has entropy rate of 0.6 to 1.5 bits per letter.

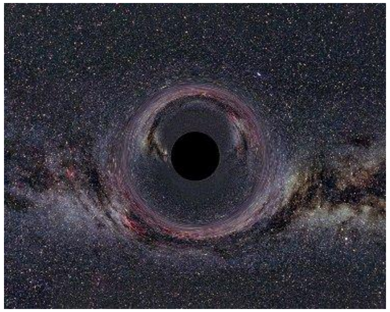
Q: How do you simulate a fair coin with a biased coin of unknown but fixed bias?

A [von Neumann]: Look at pairs of flips. HT and TH both occur with equal probability of p(1-p), and ignore HH and TT pairs.

Entropy and Randomness

- Information entropy is an analogue of thermodynamic entropy in physics / statistical mechanics, and von Neumann entropy in quantum mechanics.

- Second law of thermodynamics: entropy of an isolated system can not decrease over time.
- Entropy as "disorder" or "chaos".
- Entropy as the "arrow of time".
- "Heat death of the universe" / black holes



- Quantum computing uses a quantum information theory to generalize classical information theory.

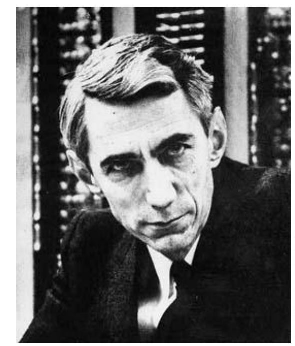
Theorem: String compressibility decreases as entropy increases.

Theorem: Most strings are not (losslessly) compressible.

Corollary: Most strings are random!

"My greatest concern was what to call it. I thought of calling it 'information', but the word was overly used, so I decided to call it 'uncertainty'. When I discussed it with John von Neumann, he had a better idea. Von Neumann told me, 'You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name, so it already has a name. In the second place, and more important, nobody knows what entropy really is, so in a debate you will always have the advantage.' "

- Claude Shannon on his conversation with John von Neumann regarding what name to give to the "measure of uncertainty" or attenuation in phone-line signals (1949)



Extra credit: watch this video!

Awards and Honors
- Claude E. Shannon Award
  - Shannon Award Nomination Form
  - 1993 Shannon Lecture (ps)
  - 1993 Shannon Lecture (pdf)
  - 1994 Shannon Lecture (ps)
  - 1994 Shannon Lecture (pdf)
  - 1995 Shannon Lecture (ps)
  - 1995 Shannon Lecture (pdf)
  - 1996 Shannon Lecture (pdf)
  - 1997 Shannon Lecture (pdf)
  - 2007 Shannon Lecture (pdf)
- Aaron D. Wyner Distinguished Service Award
- Information Theory Paper Award
- ComSoc & IT Joint Paper Award
- Chapter of the Year Award
- Golden Jubilee Paper Awards
- Golden Jubilee Awards for Technological Innovation
- IEEE Fellows
- ISIT Student Paper Award

Board of Governors
Committees
Society Chapters

Claude E. Shannon Award

— filed under: awards

The Claude E. Shannon Award of the IT Society has been instituted to honor consistent and profound contributions to the field of information theory. Each Shannon Award winner is expected to present a Shannon Lecture at the following IEEE International Symposium on Information Theory. Transcripts of some of the lectures are available on-line.

Starting for the 2010 Award, the Shannon Award Committee has decided to issue an open call for nominations, preferably using the nomination form. Although anyone may make a nomination, the Committee retains the responsibility of assuring that a suitable slate of candidates is nominated, and may itself generate nominations. Nominations and optional letters of endorsement must be submitted by March 1 to the current President of the IEEE Information Theory Society.

The first Shannon Lecturer was Claude Shannon himself followed by:

- David S. Slepian (1974)
- Robert M. Fano (1976)
- Peter Elias (1977)
- Mark S. Pinsker (1978)
- J. Wolfowitz (1979)
- W. Wesley Peterson (1981)
- Irving S. Reed (1982)
- Robert Gallager (1983)
- Solomon W. Golomb (1985)
- William L. Root (1986)
- James L. Massey (1988)

NEWS
- Postdoctoral position in computational biology
- Call For Papers: Special Issue on Cognitive Wireless Networks
- Research Fellow Position: Satellite Data Communications
- 2010 Claude E. Shannon Award
- 2009 IEEE Fellows

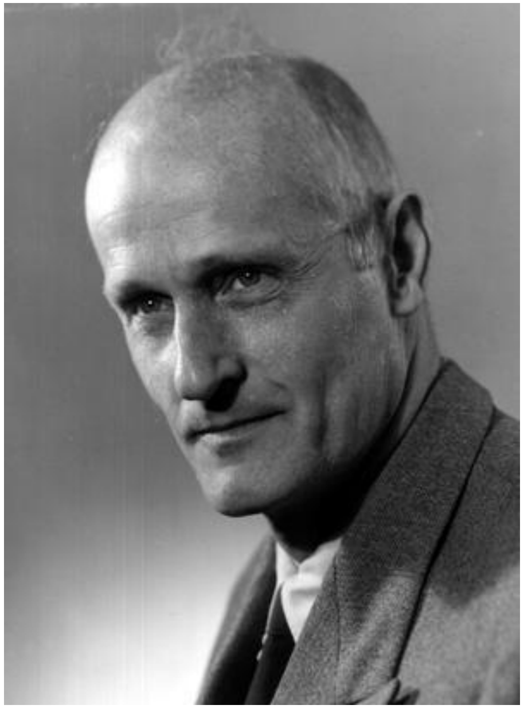
View All

UPCOMING EVENTS
- Sun Oct 11 — ITW 2009, Taormina
- Tue Oct 13 — BoG Meeting, ITW Taormina 2009
- Mon Dec 14 — Twelfth IMA International Conference on Cryptography and Coding (IMACCC)

View All

cs.IT updates on arXiv.org
Robust THP Transceiver Designs

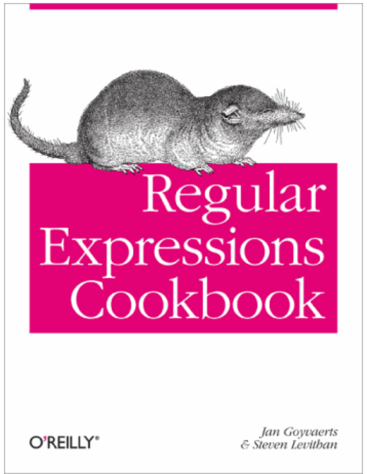
Historical Perspectives

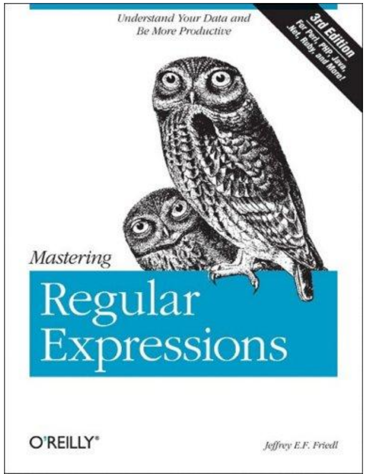


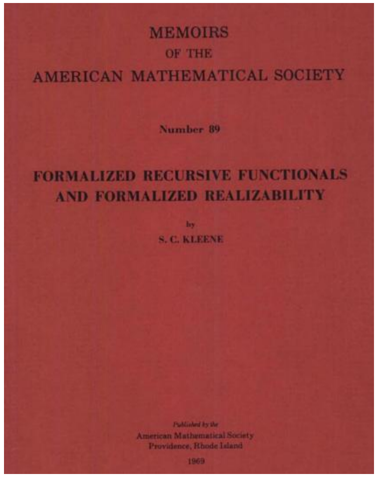
Stephen Kleene (1909-1994)

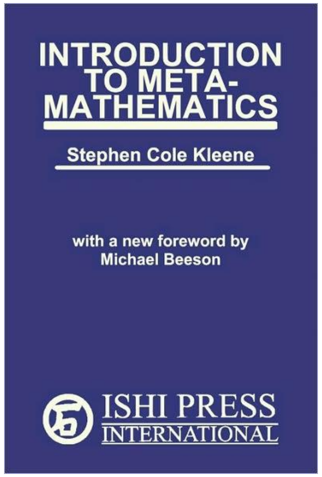
- Founded recursive function theory
- Pioneered theoretical computer science
- Student of Alonzo Church; was at the Institute for Advanced Study (1940)
- Invented regular expressions
- Kleene star / closure, Kleene algebra, Kleene recursion theorem, Kleene fixed point theorem, Kleene-Rosser paradox

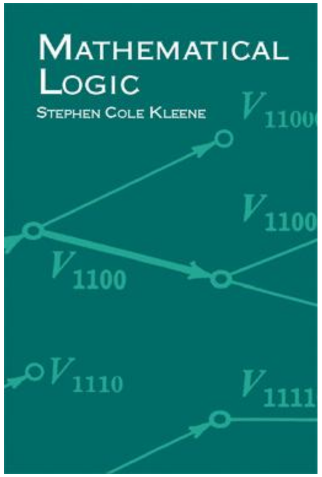
"Kleeneliness is next to Gödeliness"









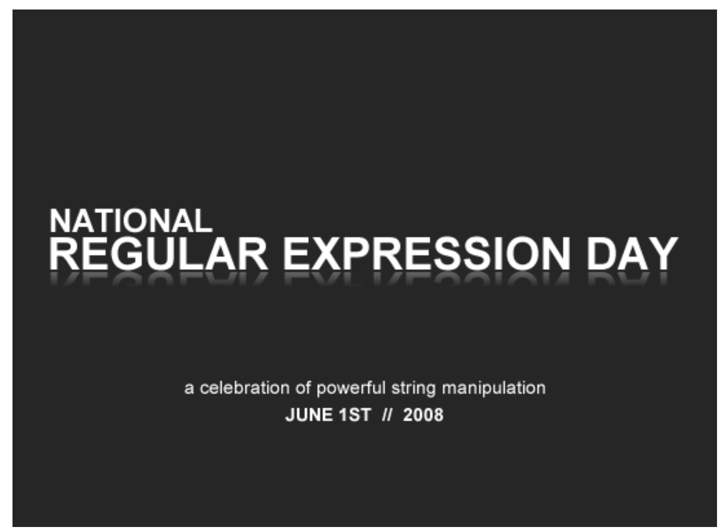


RegEx

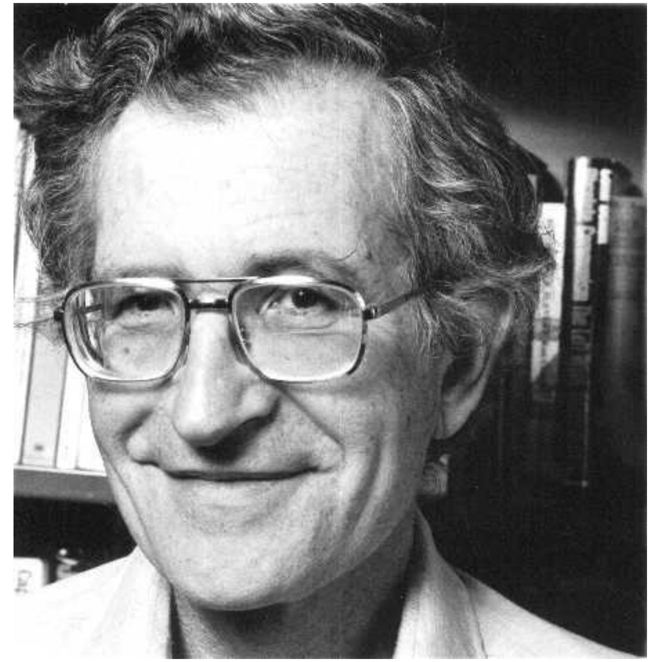
Regular Expression

/h[a4@](([c<]((k)|(\|<)))|((k)|(\|<))|(x))\s+\
((d)|([t\+]h))[3ea4@]\s+p[l1][a4@]n[3e][t\+]/i

(C)2006 FTS Conventures - www.ftsconventures.com

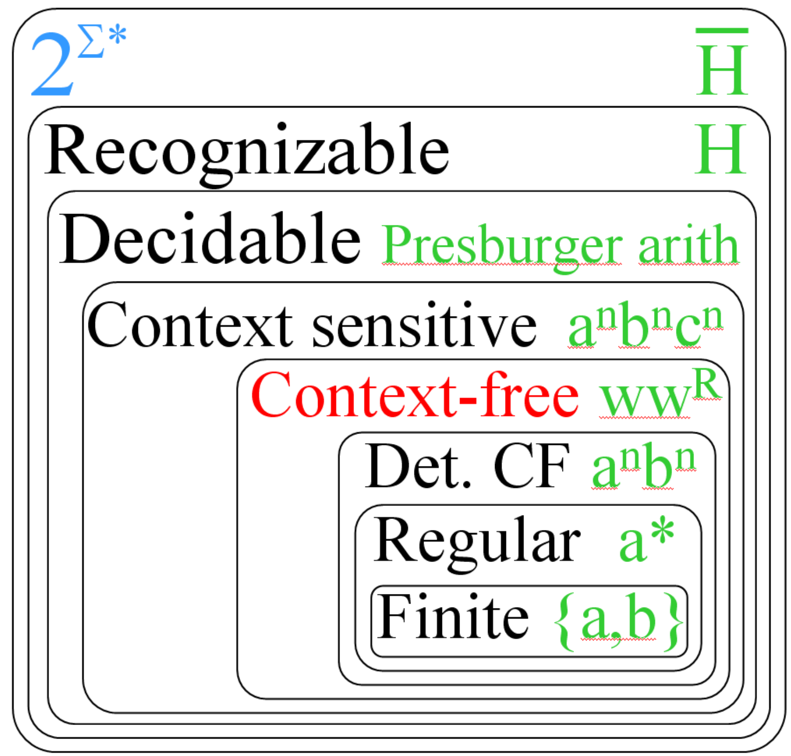


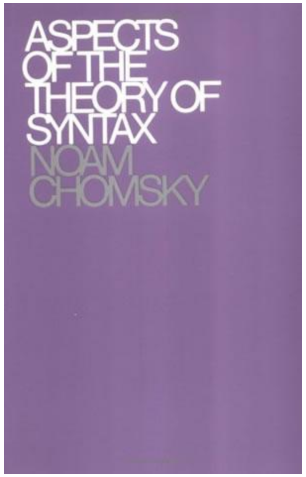
Historical Perspectives

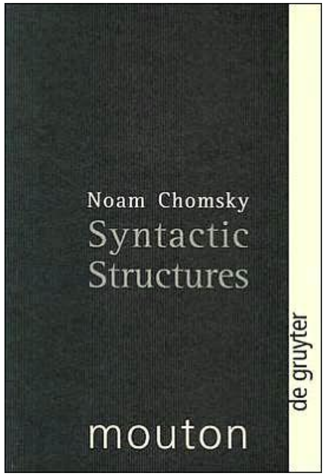


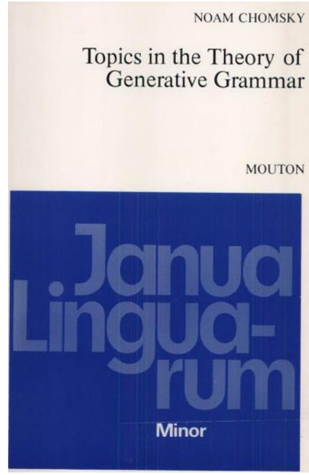
Noam Chomsky (1928-)

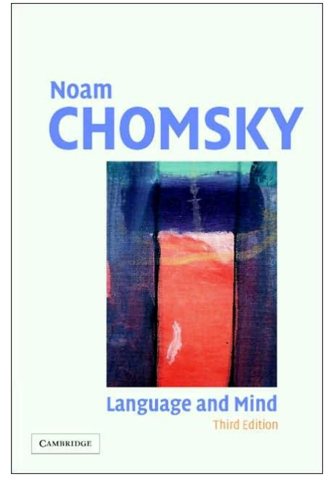
- Linguist, philosopher, cognitive scientist, political activist, dissident, author
- Father of modern linguistics
- Pioneered formal languages
- Developed generative grammars
  Invented context-free grammars
- Defined the Chomsky hierarchy
- Influenced cognitive psychology, philosophy of language and mind
- Chomskyan linguistics, Chomskyan syntax, Chomskyan models
- Critic of U.S. foreign policy
- Most widely cited living scholar
  Eighth most-cited source overall!

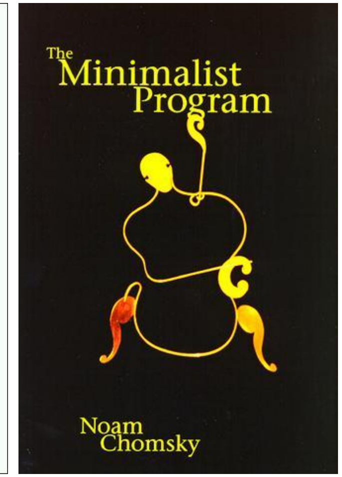


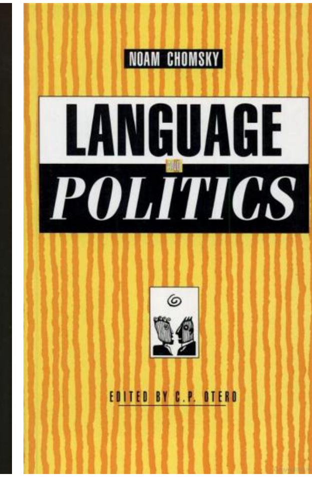

ASPECTS
OF THE
THEORY OF
SYNTAX
NOAM
CHOMSKY

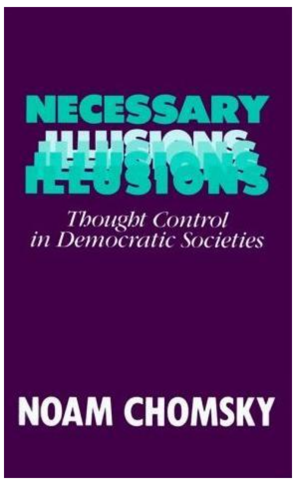

Noam Chomsky
Syntactic
Structures
mouton
de gruyter

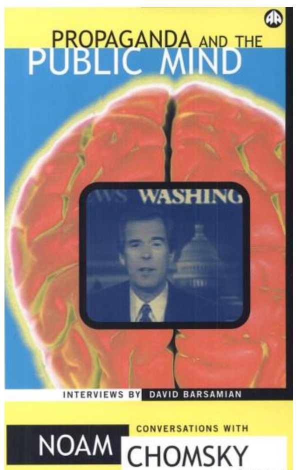

NOAM CHOMSKY
Topics in the Theory of
Generative Grammar
MOUTON
Janua
Lingua-
rum
Minor

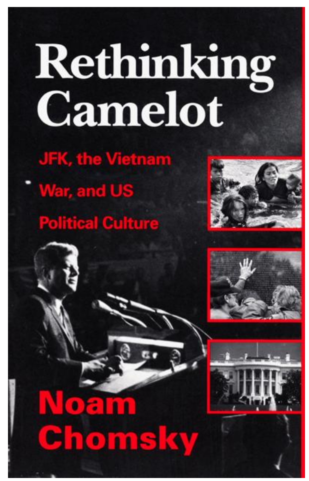

Noam
CHOMSKY
Language and Mind
Third Edition
CAMBRIDGE

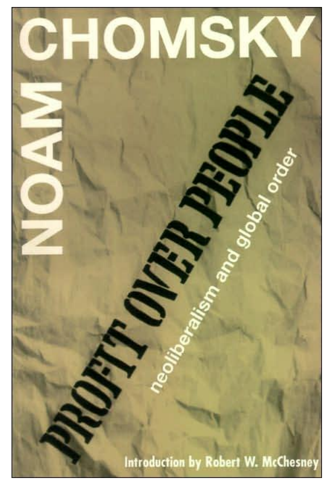

The
Minimalist
Program
Noam
Chomsky

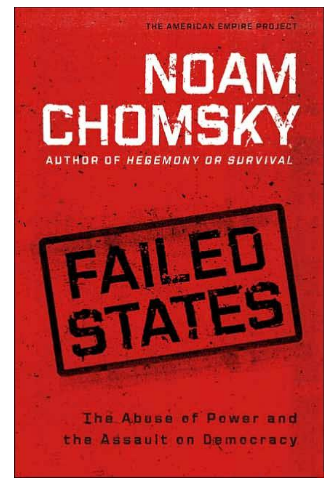

NOAM CHOMSKY
LANGUAGE
POLITICS
EDITED BY C.P. OTERO

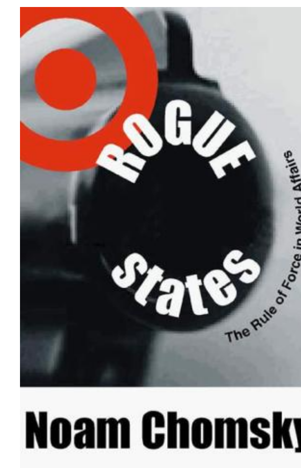

NECESSARY
ILLUSIONS
Thought Control
in Democratic Societies
NOAM CHOMSKY

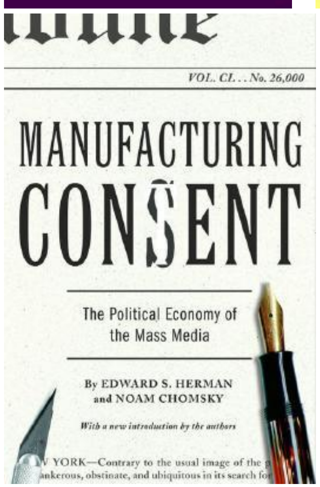

PROPAGANDA AND THE
PUBLIC MIND
WASHING
INTERVIEWS BY DAVID BARSAMIAN
CONVERSATIONS WITH
NOAM CHOMSKY

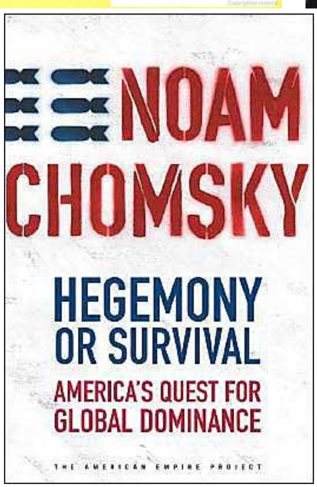

Rethinking
Camelot
JFK, the Vietnam
War, and US
Political Culture
Noam
Chomsky

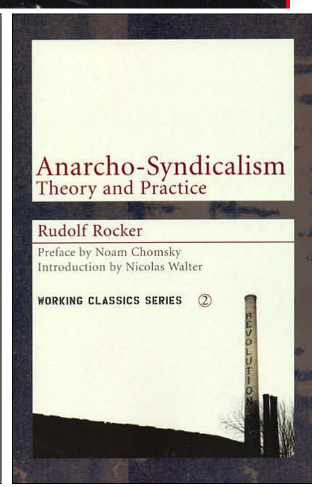

CHOMSKY
NOAM
PROFIT OVER PEOPLE
neoliberalism and global order
Introduction by Robert W. McChesney

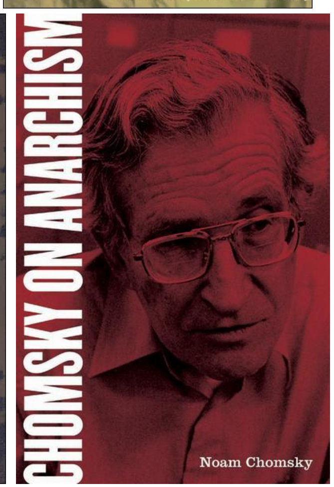

THE AMERICAN EMPIRE PROJECT
NOAM
CHOMSKY
AUTHOR OF HEGEMONY OR SURVIVAL
FAILED
STATES
The Abuse of Power and
the Assault on Democracy

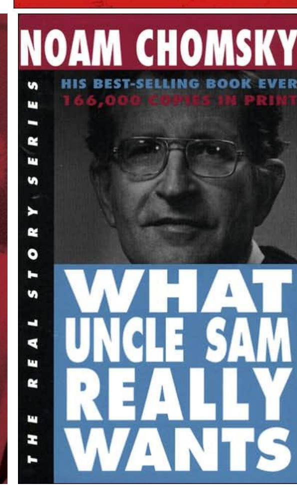

ROGUE
States
The Rule of Force in World Affairs
Noam Chomsky

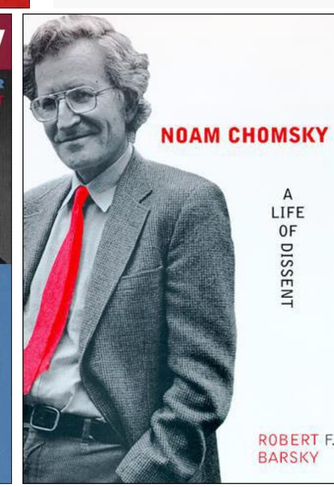

VOL. CI. . . No. 26,000
MANUFACTURING
CONSENT
The Political Economy of
the Mass Media
By EDWARD S. HERMAN
and NOAM CHOMSKY
With a new introduction by the authors


NOAM
CHOMSKY
HEGEMONY
OR SURVIVAL
AMERICA'S QUEST FOR
GLOBAL DOMINANCE
THE AMERICAN EMPIRE PROJECT

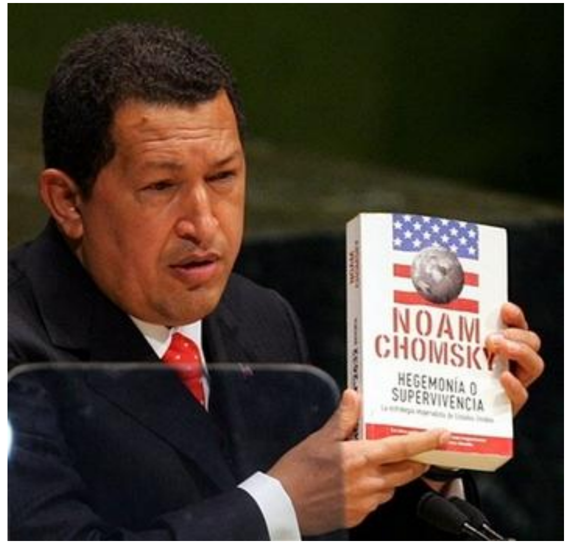

Anarcho-Syndicalism
Theory and Practice
Rudolf Rocker
Preface by Noam Chomsky
Introduction by Nicolas Walter
WORKING CLASSICS SERIES 2


CHOMSKY ON ANARCHISM
Noam Chomsky

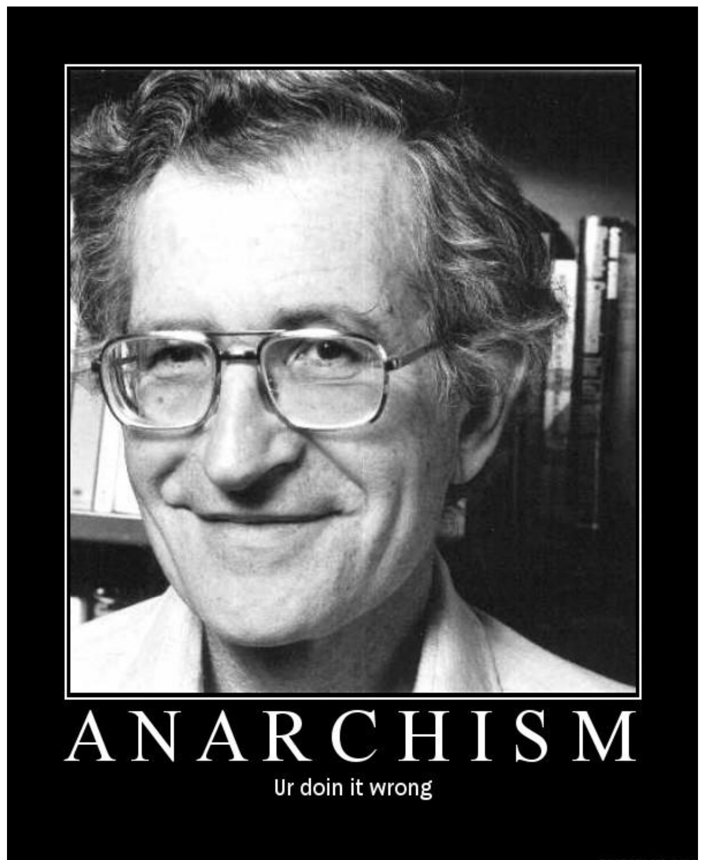

NOAM CHOMSKY
HIS BEST-SELLING BOOK EVER
THE REAL STORY SERIES
WHAT
UNCLE SAM
REALLY
WANTS

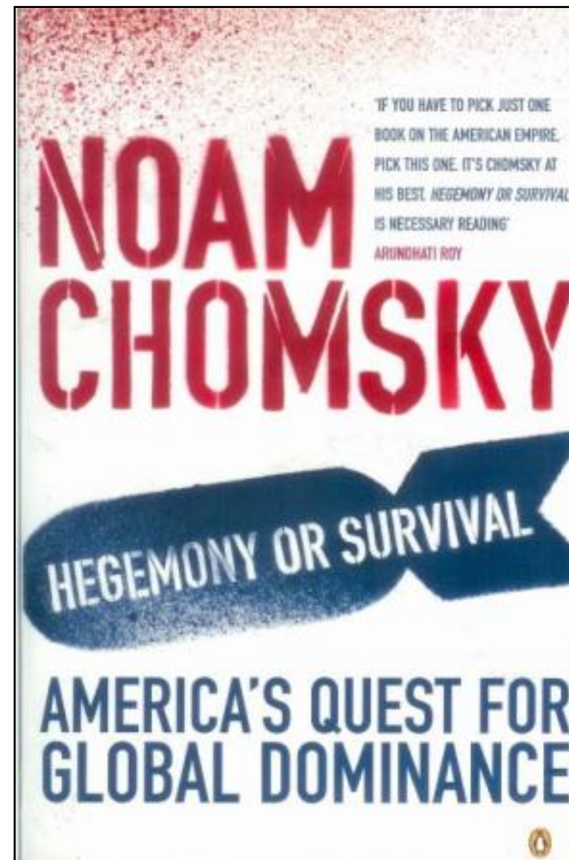

NOAM CHOMSKY
A
LIFE
OF
DISSENT
ROBERT F.
BARSKY


NOAM
CHOMSKY

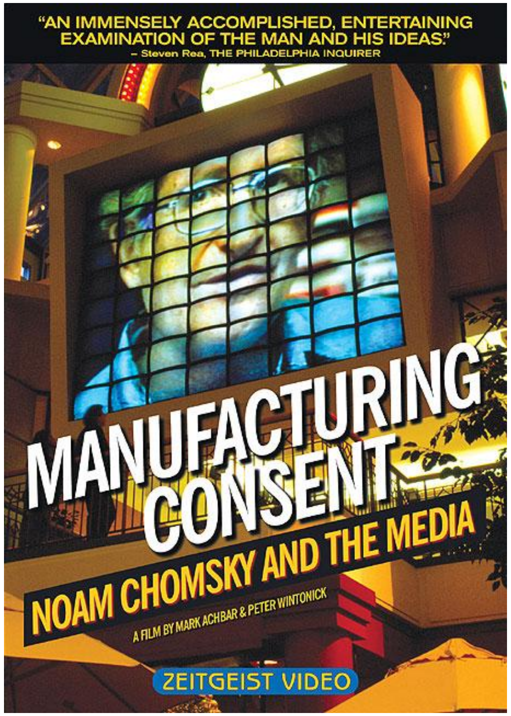

NOAM
CHOMSKY
HEGEMONÍA O
SUPERVIVENCIA

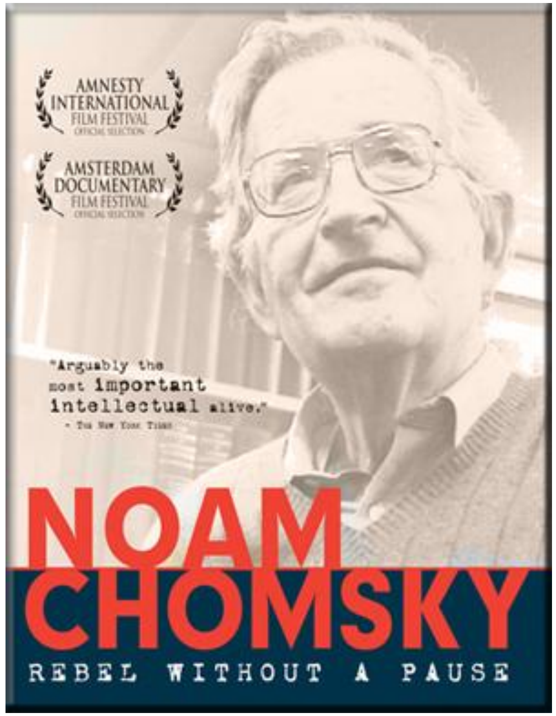

PROF. NOAM CHOMSKY
PROFESSOR OF LINGUISTICS, M.I.T.

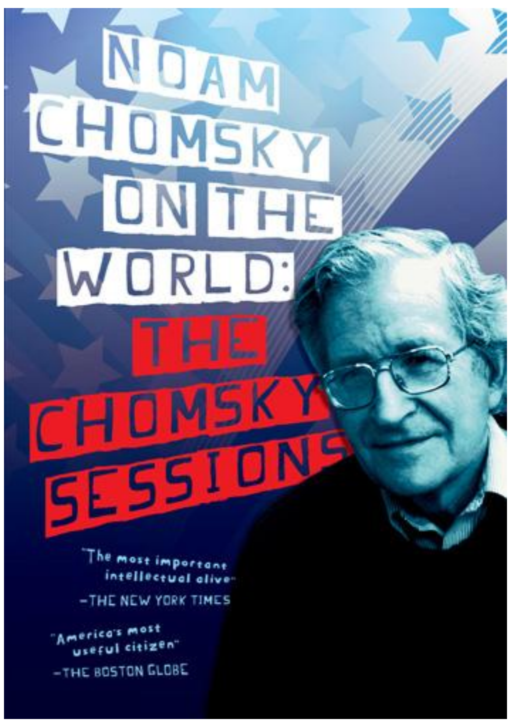

ANARCHISM
Ur doin it wrong
DIY.DESPAIR.COM

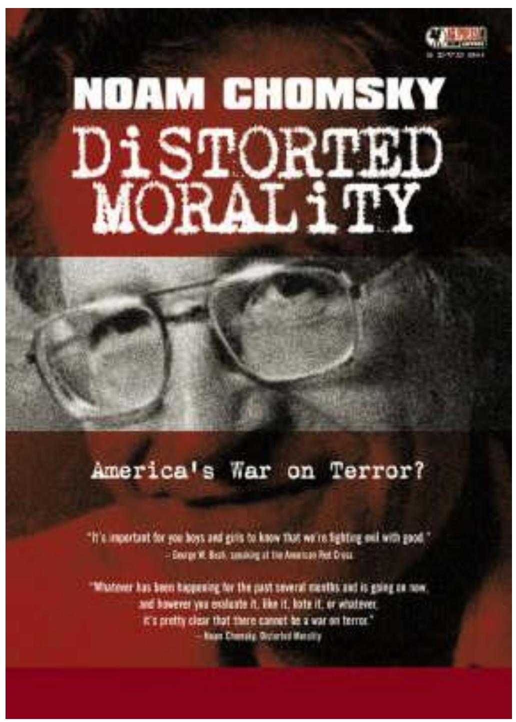

'IF YOU HAVE TO PICK JUST ONE
BOOK ON THE AMERICAN EMPIRE,
PICK THIS ONE. IT'S CHOMSKY AT
HIS BEST. HEGEMONY OR SURVIVAL
IS NECESSARY READING'
ARUNDHATI ROY
NOAM
CHOMSKY
HEGEMONY OR SURVIVAL
AMERICA'S QUEST FOR
GLOBAL DOMINANCE

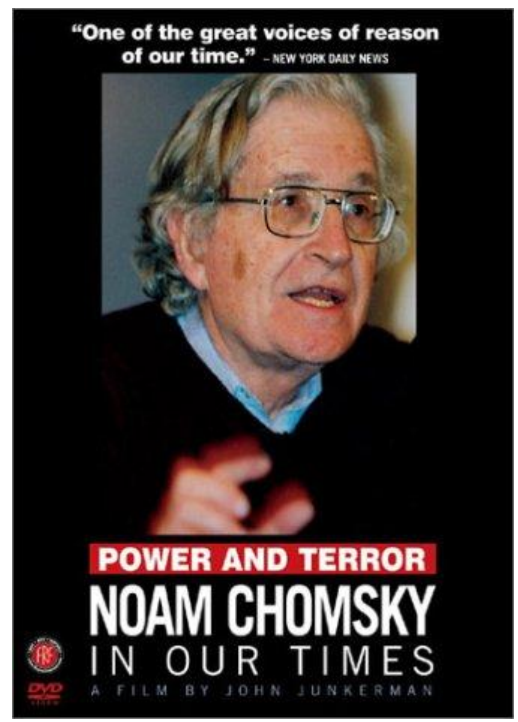
"…I must admit to taking a copy of Noam Chomsky's 'Syntactic Structures' along with me on my honeymoon in 1961 … Here was a marvelous thing: a mathematical theory of language in which I could use as a computer programmer's intuition!"
- **Don Knuth** on Chomsky's influence

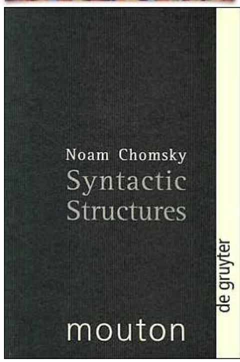


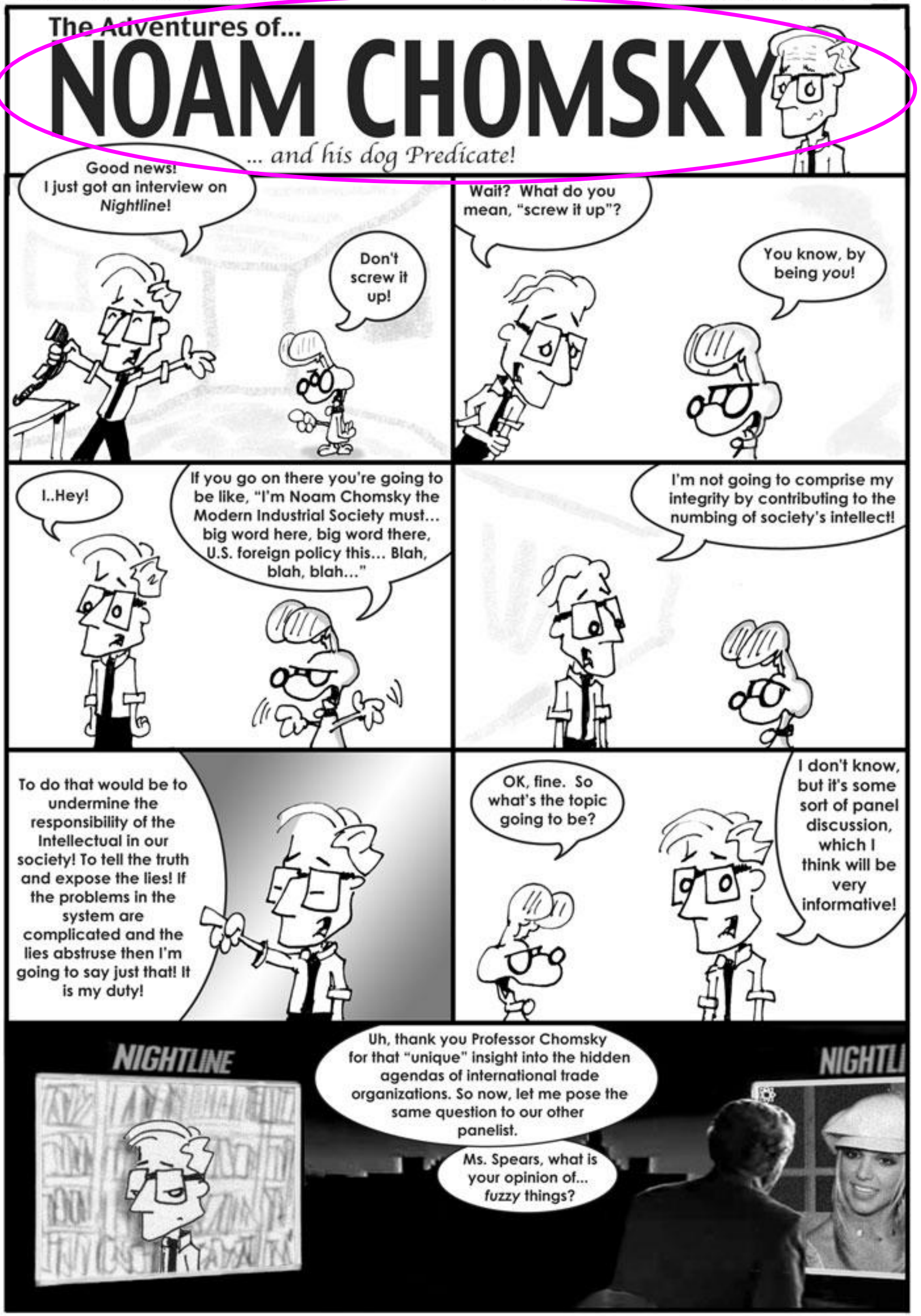




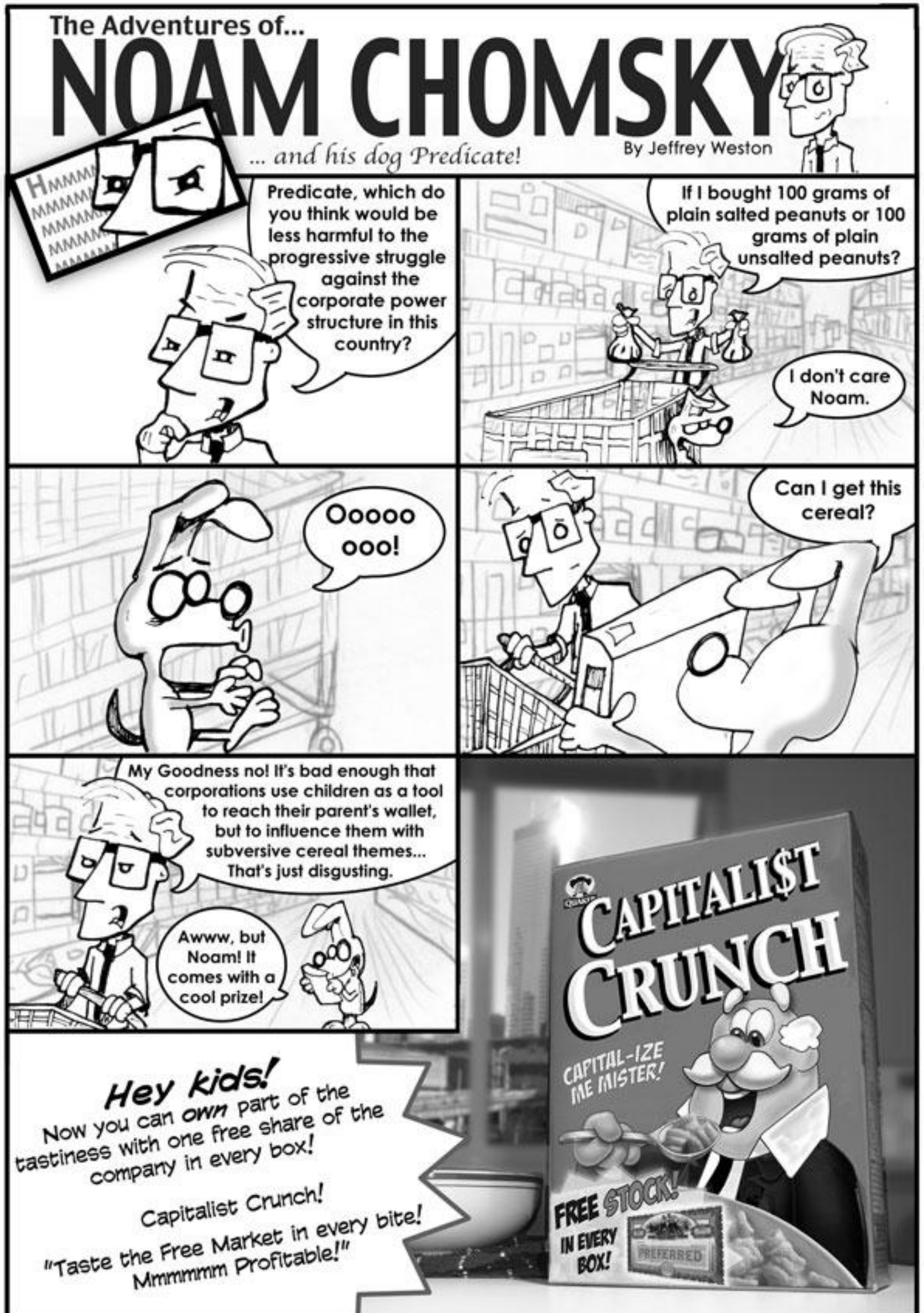







The Adventures of...
NOAM CHOMSKY
... and his dog Predicate!
Good news! I just got an interview on Nightline!
Don't screw it up!
Wait? What do you mean, "screw it up"?
You know, by being you!
I..Hey!
If you go on there you're going to be like, "I'm Noam Chomsky the Modern Industrial Society must... big word here, big word there, U.S. foreign policy this... Blah, blah, blah..."
I'm not going to comprise my integrity by contributing to the numbing of society's intellect!
To do that would be to undermine the responsibility of the Intellectual in our society! To tell the truth and expose the lies! If the problems in the system are complicated and the lies abstruse then I'm going to say just that! It is my duty!
OK, fine. So what's the topic going to be?
I don't know, but it's some sort of panel discussion, which I think will be very informative!
NIGHTLINE
NIGHTLI
Uh, thank you Professor Chomsky for that "unique" insight into the hidden agendas of international trade organizations. So now, let me pose the same question to our other panelist.
Ms. Spears, what is your opinion of... fuzzy things?

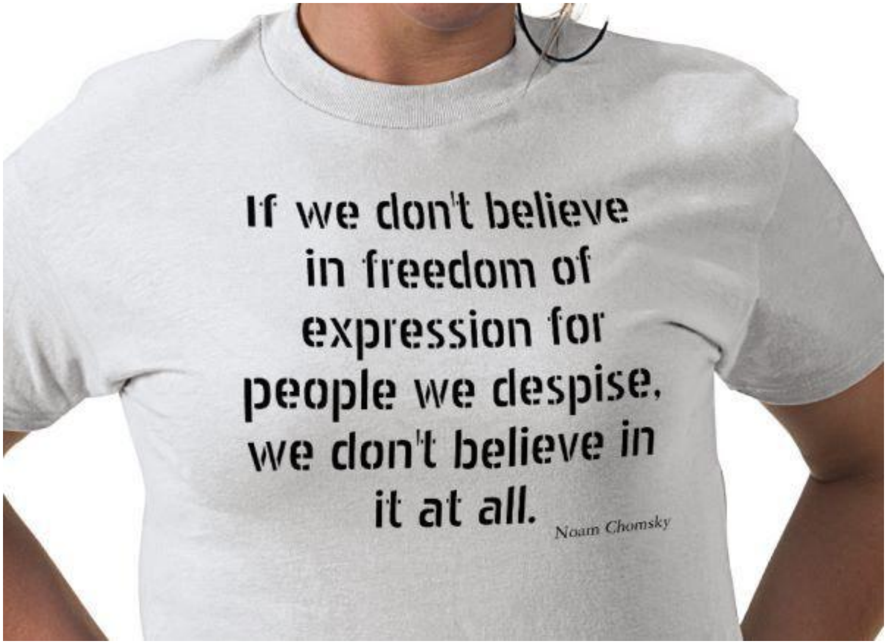
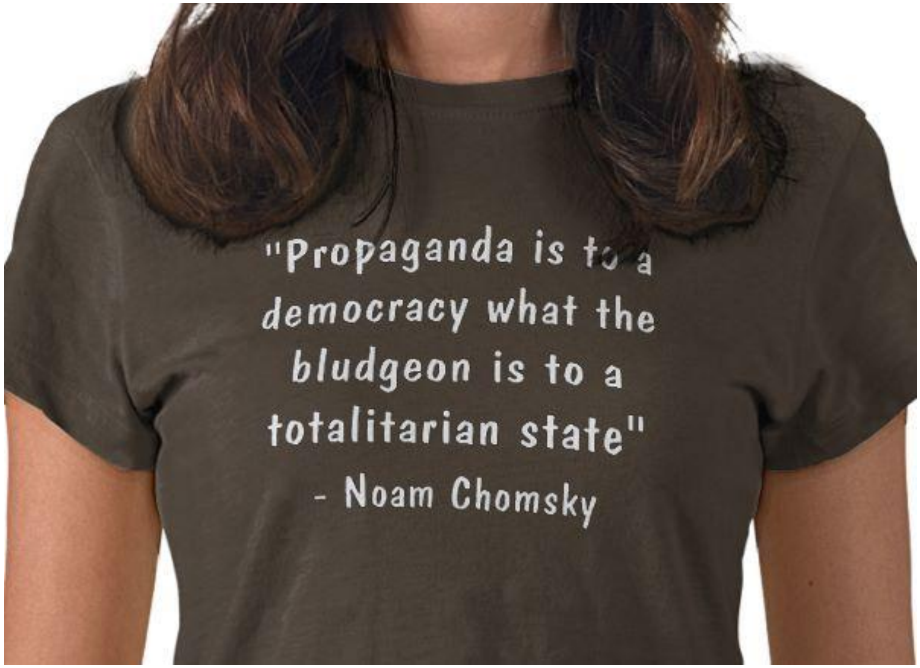

The Adventures of...
NOAM CHOMSKY
... and his dog Predicate!
By Jeffrey Weston
I need a better way to get my message out.
Yeah, market research is saying that The Noam Chomsky Quote of the Day Calendar is giving people head aches.
What can I do?
There is only one option! SELL OUT!
Look Noam, let's be realistic. You're a downer. People don't want to hear about how awful things are all the time!
Positive spin Noam, that's the way to get your message out!
How can you possibly put positive spin on the continuing decay and directed destruction of our basic freedoms?!!
Noam sez...
Everything's Fine!
© CLEARCHANNEL

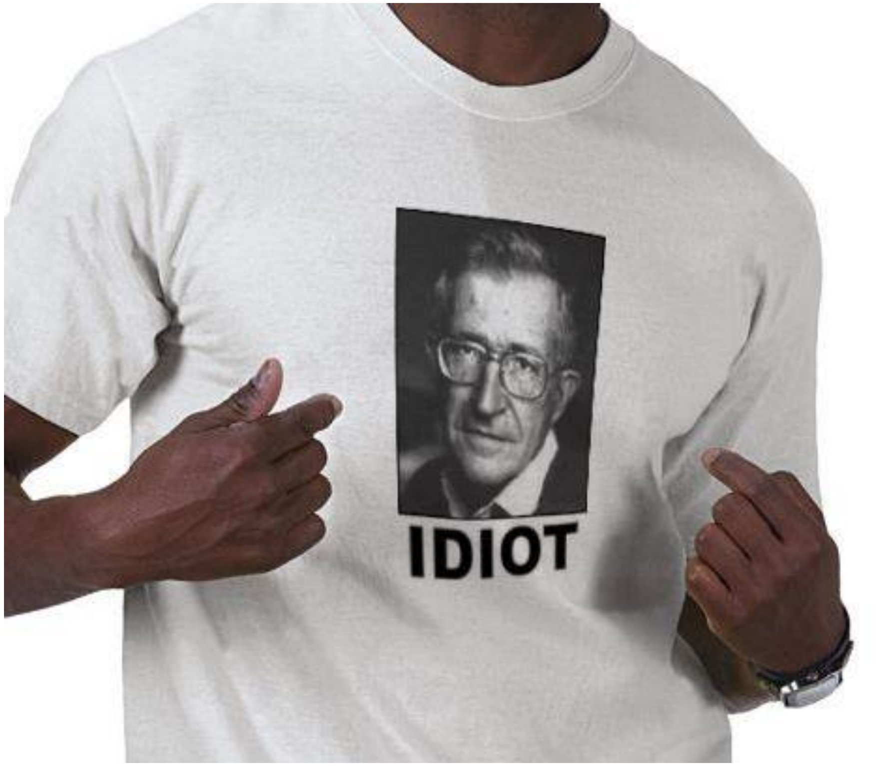


www.PostmodernHaircut.com

www.PostmodernHaircut.com

www.PostmodernHaircut.com


If we don't believe
in freedom of
expression for
people we despise,
we don't believe in
it at all.
Noam Chomsky

"Propaganda is to a
democracy what the
bludgeon is to a
totalitarian state"
- Noam Chomsky

COULD CHOMSKY
BE WRONG?

IDIOT

TURING CENTENARY CONFERENCE

CiE 2012 - How the World Computes

Print this page

Local Arrangements
- Home page
- Contact Info

Scientific Arrangements
- CiE 2012
- CiE Network and Series

External Links
- The Alan Turing Year
- Alan Turing Home Page
- The Turing Digital Archive
- Bletchley Park

University of Cambridge
18 June - 23 June, 2012

CiE 2012 is one of a series of special events, running throughout the Alan Turing Year, celebrating Turing's unique impact on mathematics, computing, computer science, informatics, morphogenesis, philosophy and the wider scientific world. Its central theme is the computability-theoretic concerns underlying the broad spectrum of Turing's interests, and the contemporary research areas founded upon and animated by them. In this sense, **CiE 2012**, held in Cambridge in the week running up to the centenary of Turing's birthday, deals with the essential core of what made Turing's contribution so influential and long-lasting. **CiE 2012** promises to be an event worthy of the remarkable scientific career it commemorates.

Programme Committee: S Barry Cooper (Leeds, **Co-chair**), Anuj Dawar (Cambridge, **Co-chair**)

Organising Committee: Luca Cardelli, S Barry Cooper (Leeds), Ann Copestake, Anuj Dawar (**Chair**), Martin Hyland, Andrew Pitts

News

17.8.09
Andrew Hodges to speak at CiE 2012

22.7.09
Cambridge confirmed for CiE12

31.12.07
Turing Advisory Group founded

Picture of King's College Chapel in Cambridge

Picture of Bletchley Park Bombe rebuild

The Alan Turing Memorial in Sackville Park, Manchester