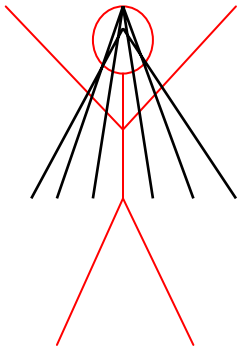
Computational Thinking



Jeannette M. Wing

President's Professor of Computer Science and Department Head
Computer Science Department
Carnegie Mellon University

My Grand Vision for the Field

- Computational thinking will be a fundamental skill used by everyone in the world by the middle of the 21st Century.
  - Just like reading, writing, and arithmetic.
  - Imagine every child knowing how to think like a computer scientist!
  - Incestuous: Computing and computers will enable the spread of computational thinking.

Computational Thinking

- C.T. enables what one human being cannot do alone
  - For solving problems
  - For designing systems
  - For understanding the power and limits of human and machine intelligence

The Two A's of Computational Thinking

- Abstraction
  - C.T. is operating in terms of multiple layers of abstraction simultaneously
  - C.T. is defining the relationships the between layers
- Automation
  - C.T. is thinking in terms of mechanizing the abstraction layers and their relationships
    - Mechanization is possible due to precise and exacting notations and models
  - There is some "machine" below (human or computer, virtual or physical)
- They give us the ability and audacity to scale.

Examples of Computational Thinking

- How difficult is this problem and how best can I solve it?
  - Theoretical computer science gives precise meaning to these and related questions and their answers.
- C.T. is thinking recursively.
- C.T. is reformulating a seemingly difficult problem into one which we know how to solve.
  - Reduction, embedding, transformation, simulation
- C.T. is choosing an appropriate representation or modeling the relevant aspects of a problem to make it tractable.
- C.T. is interpreting code as data and data as code.
- C.T. is using abstraction and decomposition in tackling a large complex task.
- C.T. is judging a system's design for its simplicity and elegance.
- C.T. is type checking, as a generalization of dimensional analysis.
- C.T. is prevention, detection, and recovery from worst-case scenarios through redundancy, damage containment, and error correction.
- C.T. is modularizing something in anticipation of multiple users and prefetching and caching in anticipation of future use.
- C.T. is calling gridlock deadlock and avoiding race conditions when synchronizing meetings.
- C.T. is using the difficulty of solving hard AI problems to foil computing agents.
- C.T. is taking an approach to solving problems, designing systems, and understanding human behavior that draws on concepts fundamental to computer science.

Please tell me your favorite examples of computational thinking!

Evidence of Computational Thinking's Influence

- Computational thinking, in particular, machine learning has revolutionized Statistics
  - Statistics departments in the US are hiring computer scientists
  - Schools of computer science in the US are starting or embracing existing Statistics departments
- Computational thinking is our current big bet in Biology
  - Algorithms and data structures, computational abstractions and methods will inform biology.

- Computational thinking in other disciplines (more examples later)
  - Game Theory
    - CT is influencing Economics
  - Nanocomputing
    - CT is influencing Chemistry
  - Quantum computing
    - CT is influencing Physics

Analogy

The boldness of my vision: Computational thinking is not just for other scientists, it's for *everyone*.

- Ubiquitous computing was yesterday's dream, today's reality
- Computational thinking is today's dream, tomorrow's reality

Computational Thinking: What It Is and Is Not

- Conceptualizing, not programming
  - Computer science is not just computer programming
- Fundamental, not rote skill
  - A skill every human being needs to know to function in modern society
  - Rote: mechanical. Need to solve the AI Grand Challenge of making computers “think” like humans. Save that for the second half of this century!
- A way that humans, not computers think
  - Humans are clever and creative
  - Computers are dull and boring

Computational Thinking: What It Is and Is Not

- Complements and combines mathematical and engineering thinking
  - C.T. draws on math as its foundations
    - But we are constrained by the physics of the underlying machine
  - C.T. draws on engineering since our systems interact with the real world
    - But we can build virtual worlds unconstrained by physical reality
- Ideas, not artifacts
  - It's not just the software and hardware that touch our daily lives, it will be the computational concepts we use to approach living.
- It's for everyone, everywhere
  - C.T. will be a reality when it is so integral to human endeavors that it disappears as an explicit philosophy.

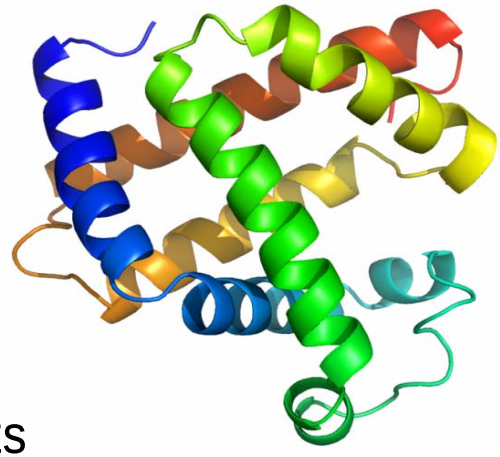
Two Messages for the General Public

- Intellectually challenging and engaging scientific problems in computer science remain to be understood and solved.
  - Limited only by our curiosity and creativity

- One can major in computer science and do anything.
  - Just like English, political science, or mathematics

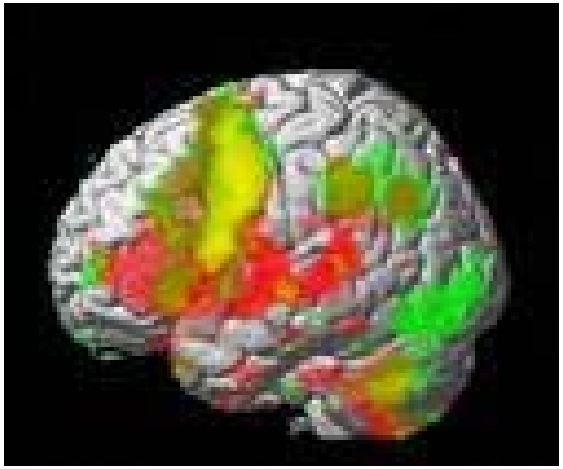
Research Implications

CT in Other Sciences, Math, and Engineering



Biology

- Shotgun algorithm expedites sequencing of human genome
- DNA sequences are strings in a language
- Protein structures can be modeled as knots
- Protein kinetics can be modeled as computational processes
- Cells as a self-regulatory system are like electronic circuits



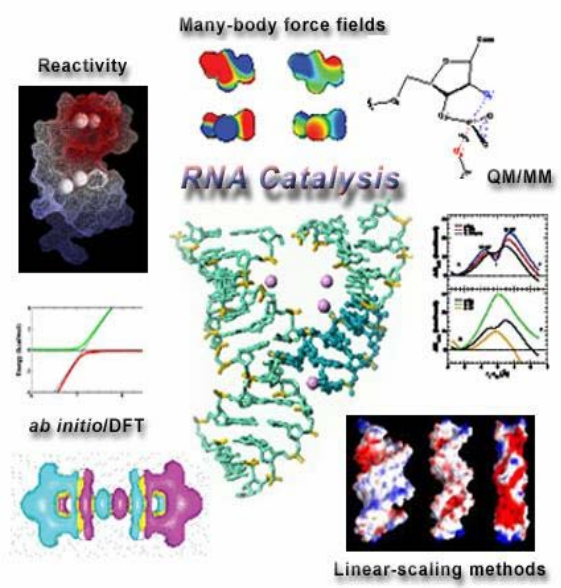
Brain Science

- Modeling the brain as a computer
- Vision as a feedback loop
- Analyzing fMRI data with machine learning

CT in Other Sciences, Math, and Engineering

Chemistry [Madden, Fellow of Royal Society of Edinburgh]

- Atomistic calculations are used to explore chemical phenomena
- Optimization and searching algorithms identify best chemicals for improving reaction conditions to improve yields



[York, Minnesota]

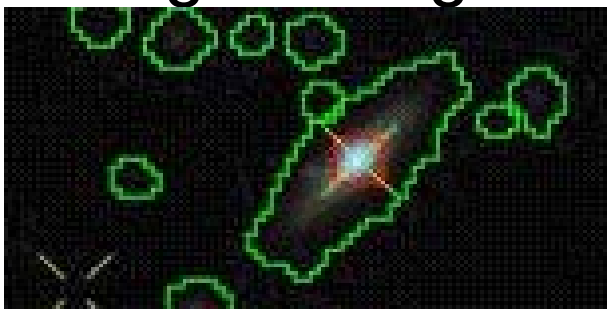
Geology

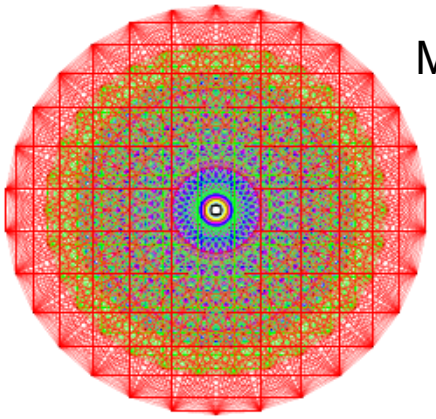
- "The Earth is an analogue computer."
  [Boulton, Edinburgh]
- Abstraction boundaries and hierarchies of complexity model the earth and our atmosphere

CT in Other Sciences, Math, and Engineering

Astronomy

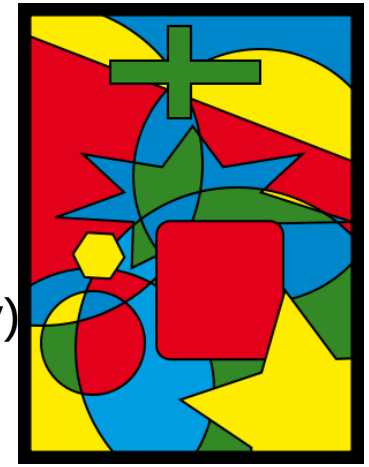
- Sloan Digital Sky Server brings a telescope to every child
- KD-trees help astronomers analyze very large multi-dimensional datasets





Mathematics

- Discovering E8 Lie Group:
  18 mathematicians, 4 years and 77 hours of supercomputer time (200 billion numbers).
  Profound implications for physics (string theory)
- Four-color theorem proof



Engineering (electrical, civil,mechanical, aero&astro, ...)

- Calculating higher order terms implies more precision, which implies reducing weight, waste, costs in fabrication
- Boeing 777 tested via computer simulation alone, not in a wind tunnel

CT for Society

Economics

- Automated mechanism design underlies electronic commerce, e.g., ad placement, on-line auctions, kidney exchange
- MIT PhDs in CS are quants on Wall Street

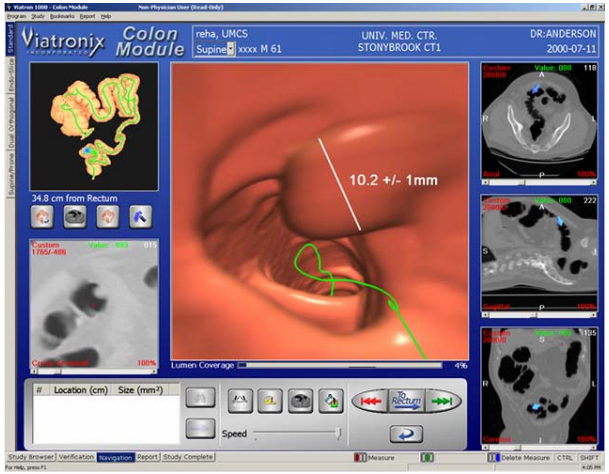
Social Sciences

- Social networks explain phenomena such as MySpace, YouTube
- Statistical machine learning is used for recommendation and reputation services, e.g., Netflix, affinity card

CT for Society

Medicine

- Robotic surgery
- Electronic health records require privacy technologies
- Scientific visualization enables virtual colonoscopy

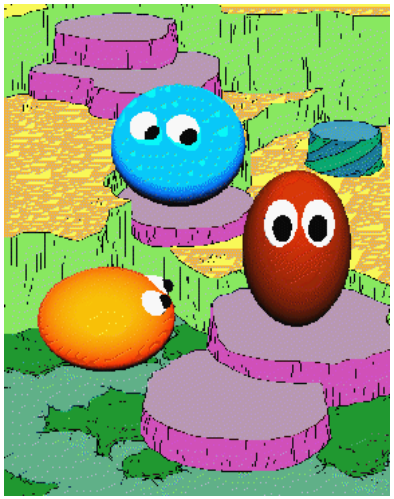


Law

- Stanford CL approaches include AI, temporal logic, state machines, process algebras, petri nets
- POIROT Project on fraud investigation is creating a detailed ontology of European law
- Sherlock Project on crime scene investigation

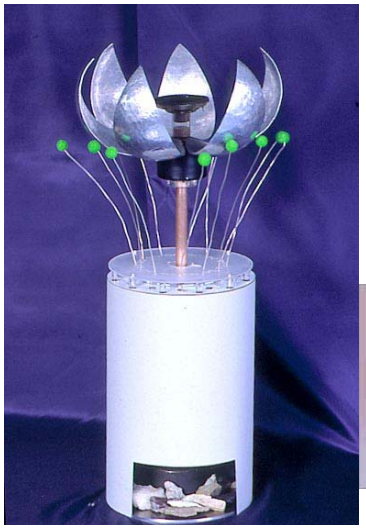
CT for Society

Entertainment



- Games
- Movies
  - Dreamworks uses HP data center to render *Shrek* and *Madagascar*
  - Lucas Films uses 2000-node data center produce *Pirates of the Caribbean.*

Arts



- Art (e.g., Robotticelli)
- Drama
- Music
- Photography

Sports

- Lance Armstrong's cycling computer tracks man and machine statistics
- Synergy Sports analyzes digital videos NBA games

Educational Implications

Curricula Development

- Universities should start with their freshmen-level intro courses.
  - Teach “Ways to Think Like a Computer Scientist” not just “Intro to <programming *langage du jour*>”
- Engage national and international organizations to reform curricula, in particular K-12.
  - ACM, CSTA, CRA, etc.
- It needs to be a collective effort.

Adapt Course Material for Other Audiences

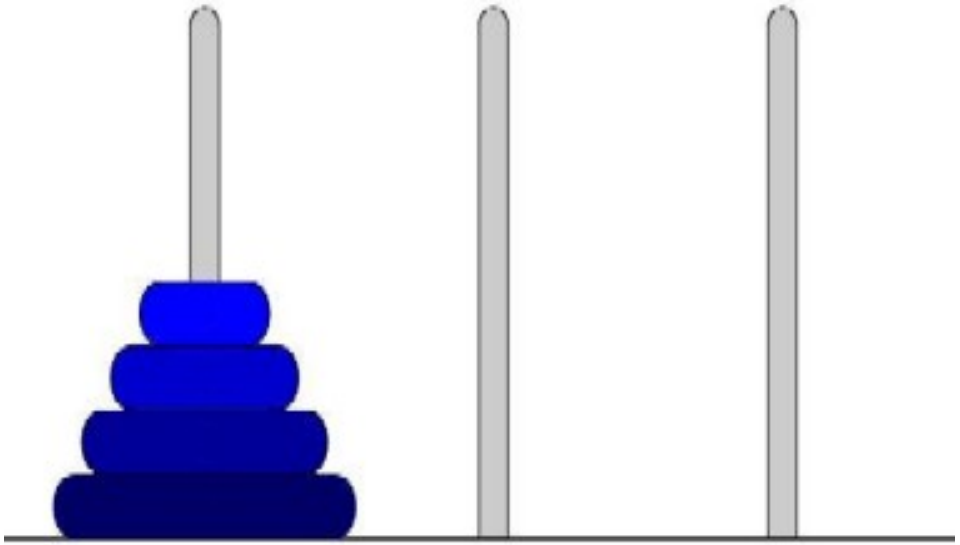
- K-6, 7-9, 10-12
- Undergraduate courses
  - Freshmen year
    - "Ways to Think Like a Computer Scientist" aka Principles of Computing
  - Upper-level courses
- Graduate-level courses
  - Computational arts and sciences
    - E.g., entertainment technology, computational linguistics, …, computational finance, …, computational biology, computational astrophysics
- Students and teachers
  - CS4All: Carnegie Mellon outreach to high school teachers, counselors, and guidance counselors

Ways To Think Like A Computer Scientist

- Freshmen year course
- Suitable for non-majors, but inspirational for majors
- Lessons
  - Thinking Recursively
  - Thinking Abstractly
  - Thinking Ahead (caching, pre-fetching...)
  - Thinking Procedurally
  - Thinking Logically
  - Thinking Concurrently
  - ...
- Problem sets: pencil-and-paper thought exercises, programming exercises

Recursion: Towers of Hanoi

Goal: Transfer the entire tower to one of the other pegs, moving only one disk at a time and never a larger one onto a smaller.

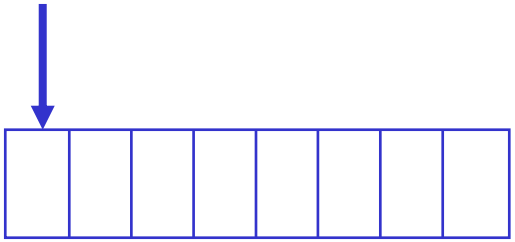


Data Abstraction and Representation

stack

queue

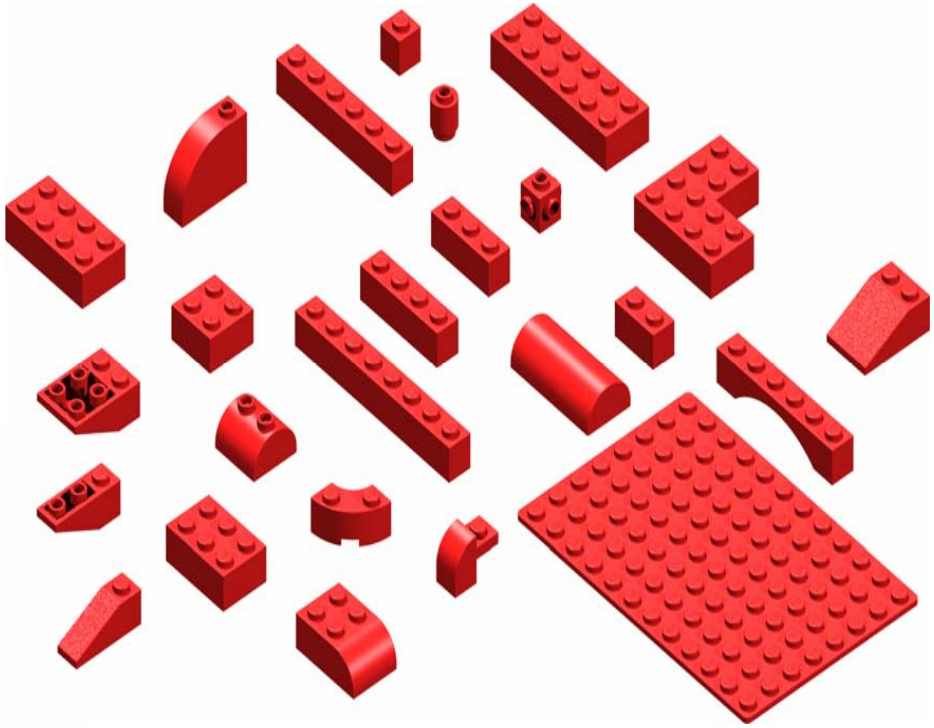
tree
(upside down)



array and pointer

representation invariant

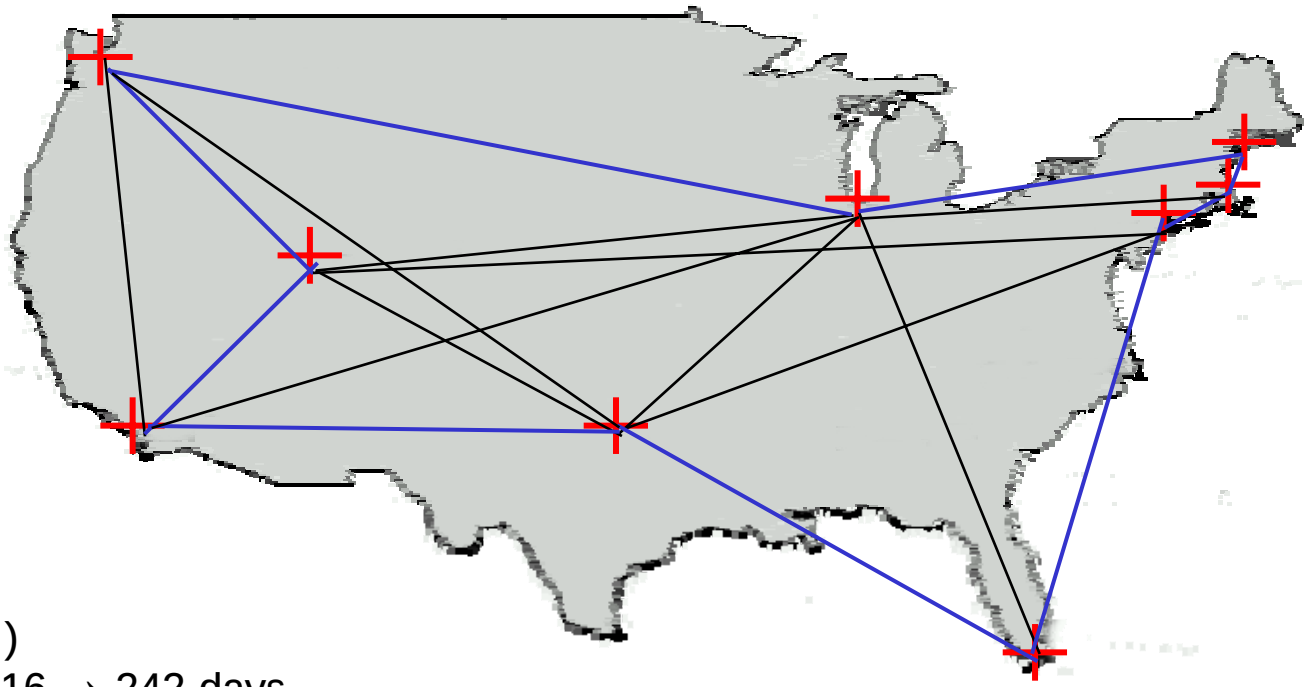
Composition and Decomposition



Sorting and Search

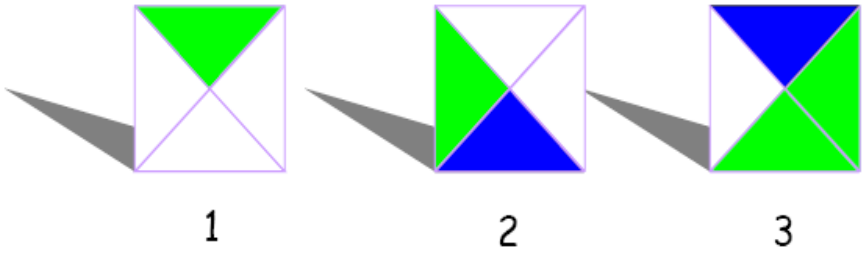
Intractability: Traveling Salesman

Problem: A traveling salesperson needs to visit *n* cities.
Is there a route of at most *d* in length?

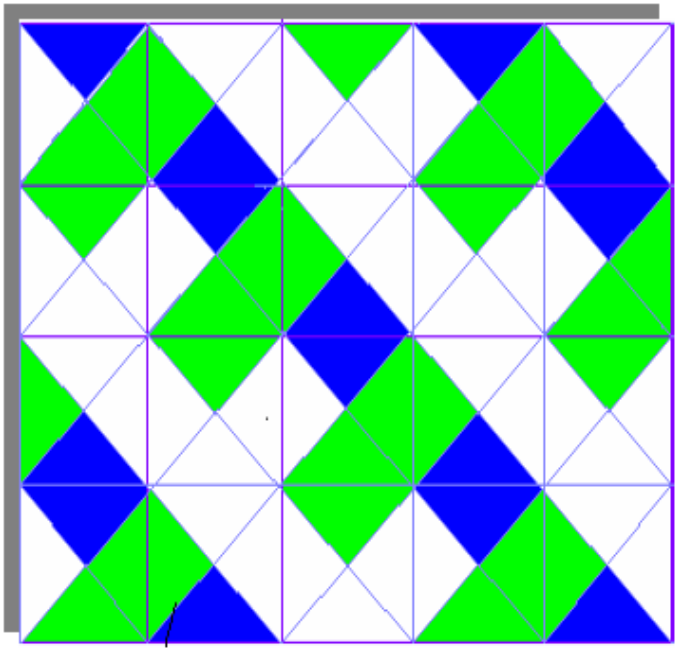


O(n!)
n = 16 → 242 days
n = 25 → 5x10^15 centuries

Undecidability: Tiling

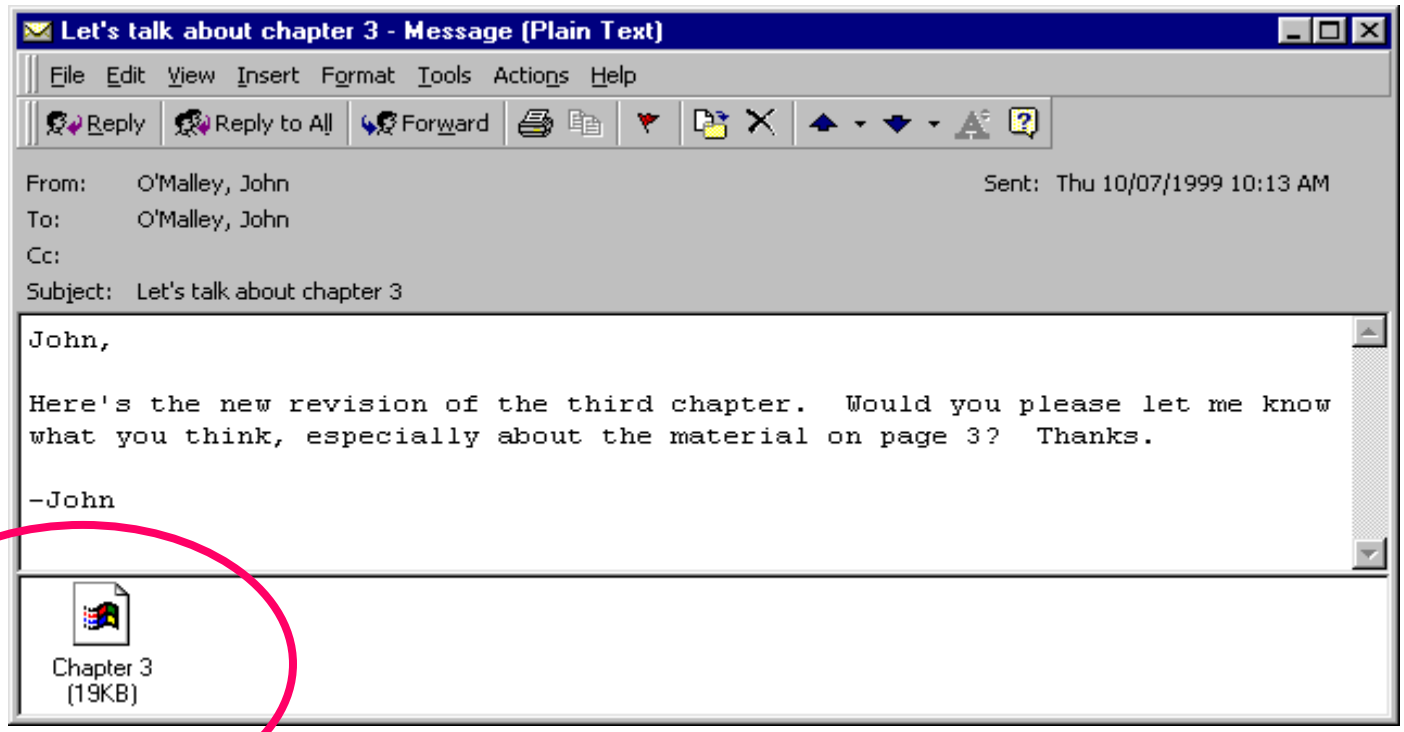


Can we tile the entire plane Z^2?



Example from David Harel

Data as Code and Code as Data



unrecognized
email attachment

Recursion Revisited

The Y operator

$$Y = \lambda\, f.\ (\lambda\, x.\ f\ (x\ x))\ (\lambda\, x.\ f\ (x\ x))$$

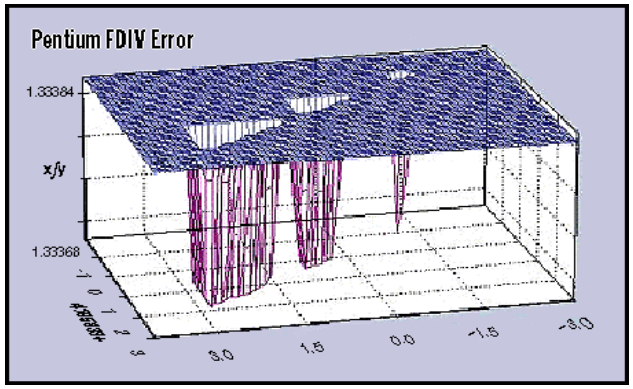
which satisfies the following equation

$$Y\ f = f\ (Y\ f)$$

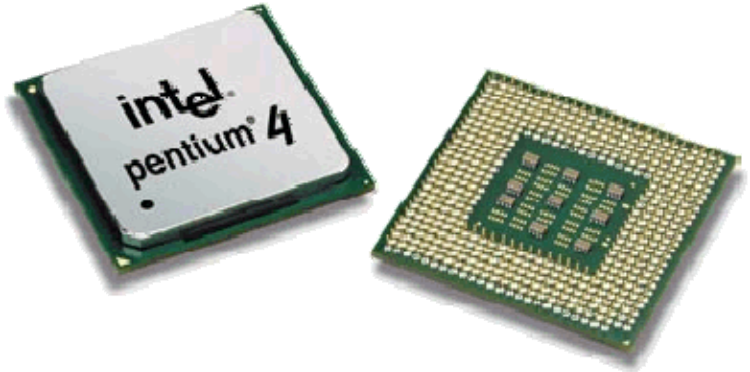
and is the basis of recursion in Computer Science.

Y is the *fixed point* combinator in lambda calculus.

Correctness: Avoiding Bugs to Save Money and Lives

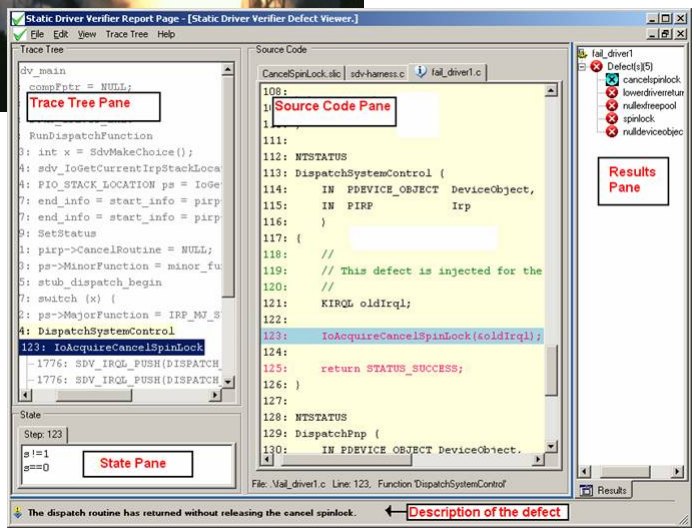


Intel Pentium FPU error



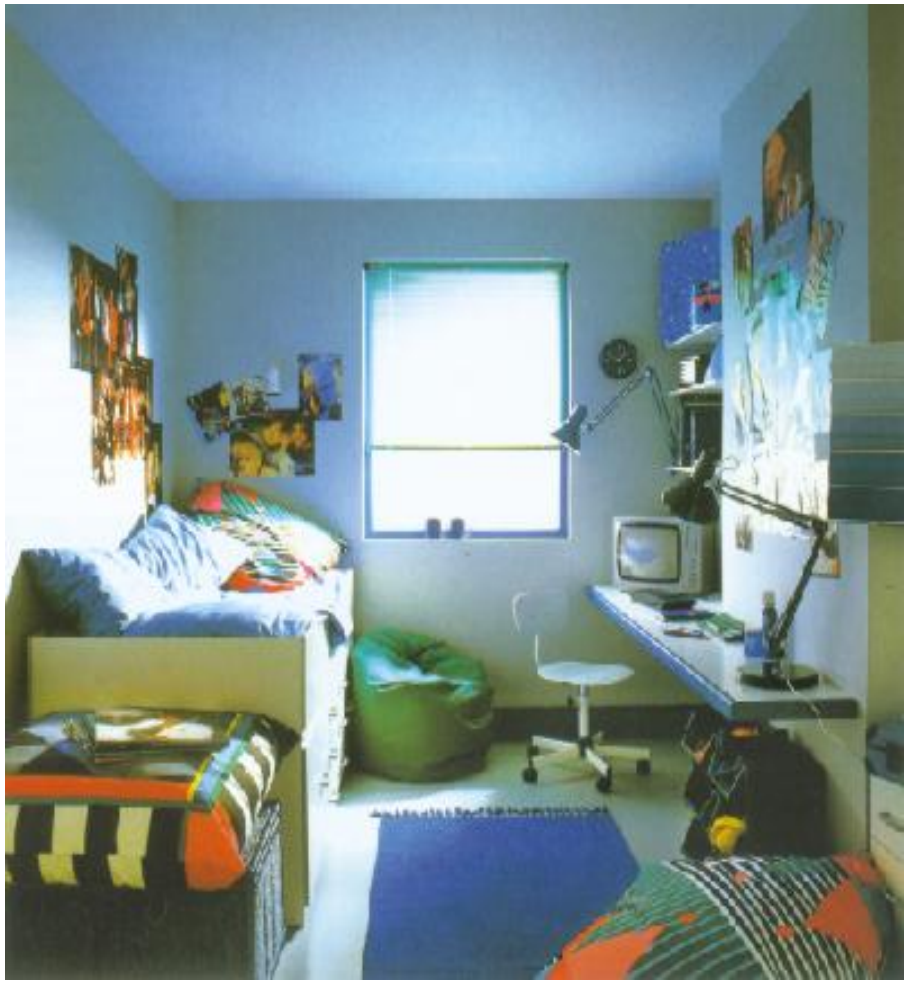
Now Intel uses formal verification.

Ariane 5 failure



Now Microsoft uses formal verification.

Caching



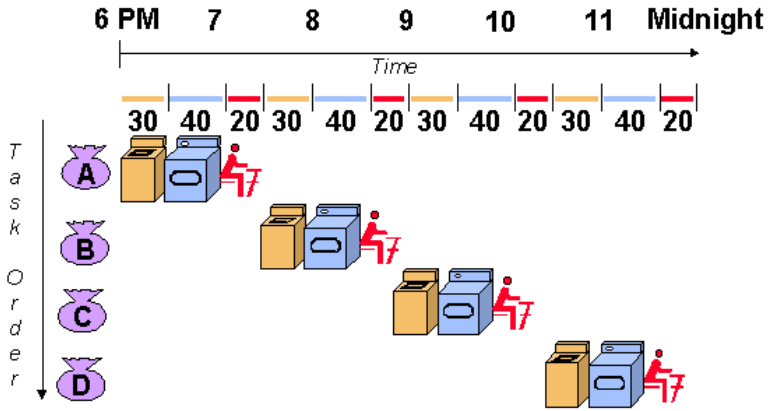
home

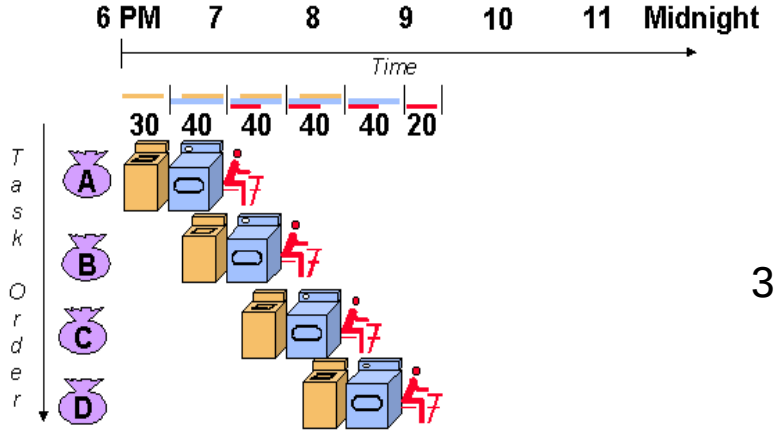
locker

knapsack

Pipelining: Doing Laundry

6 hours to do 4 loads

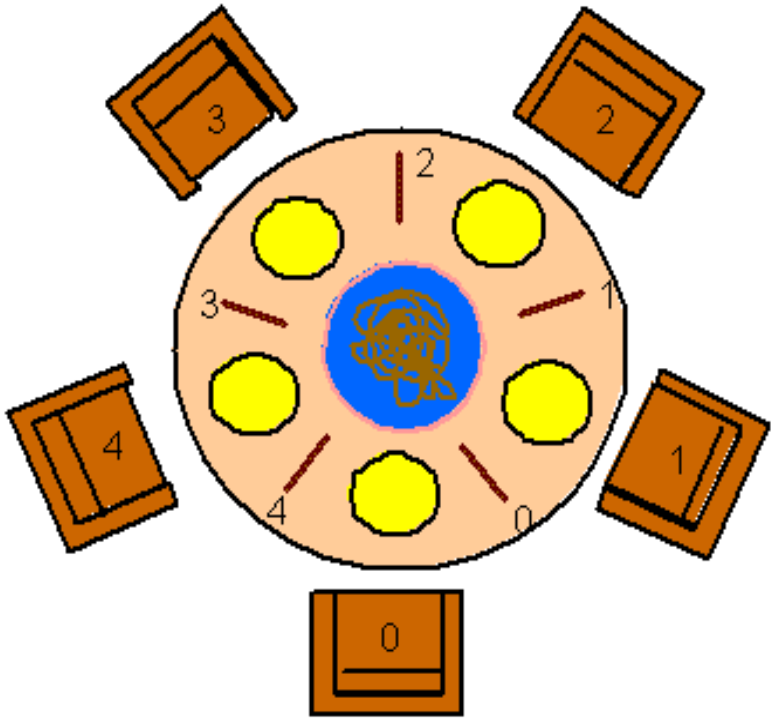




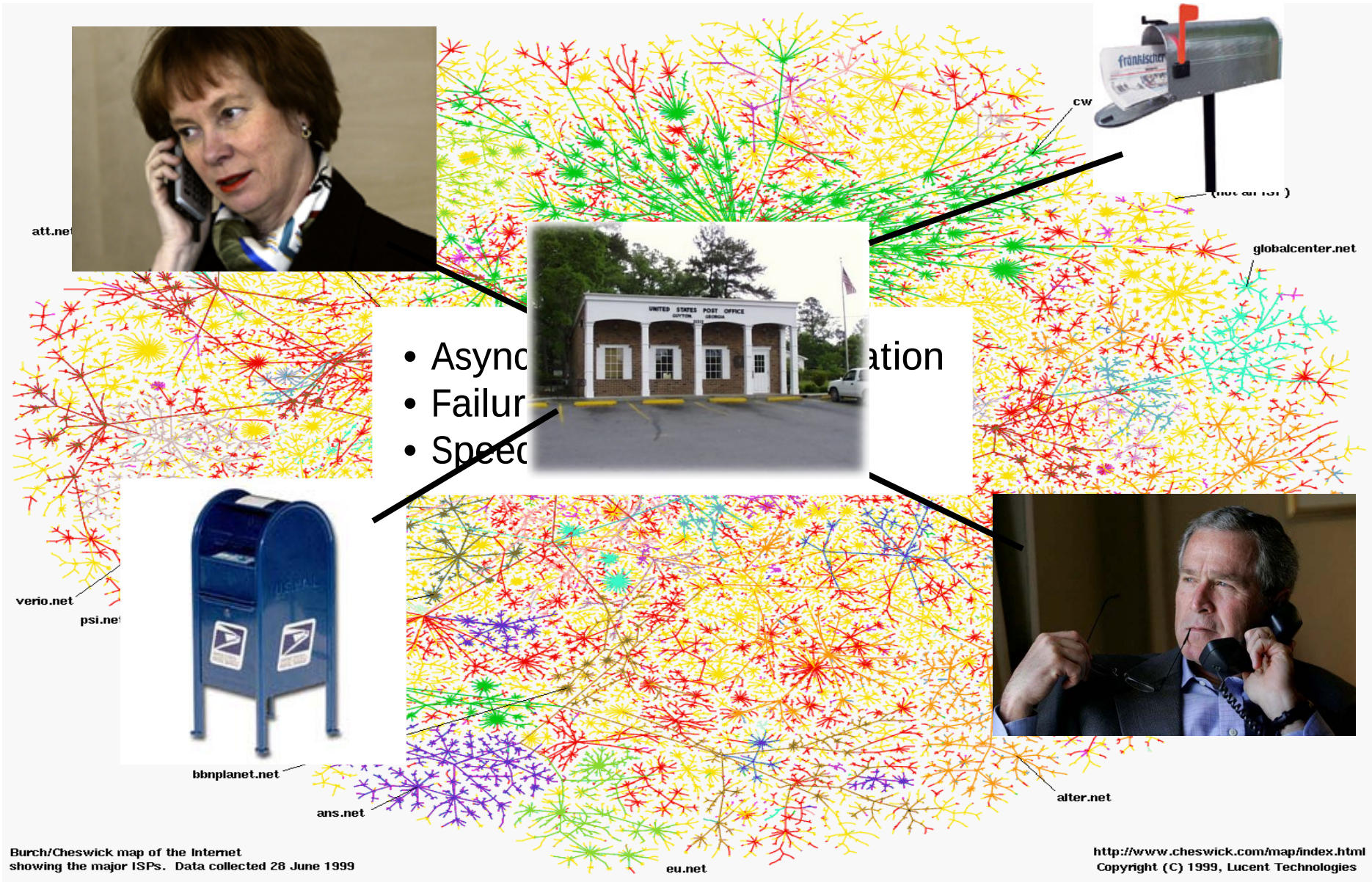
3.3 hours to do 4 loads

Concurrency: Dining Philosophers

Five philosophers sit around a circular table. Each philosopher spends his life alternately thinking and eating. In the centre of the table is a large bowl of spaghetti. A philosopher needs two forks to eat a helping of spaghetti.



Distributed Computing: The Internet

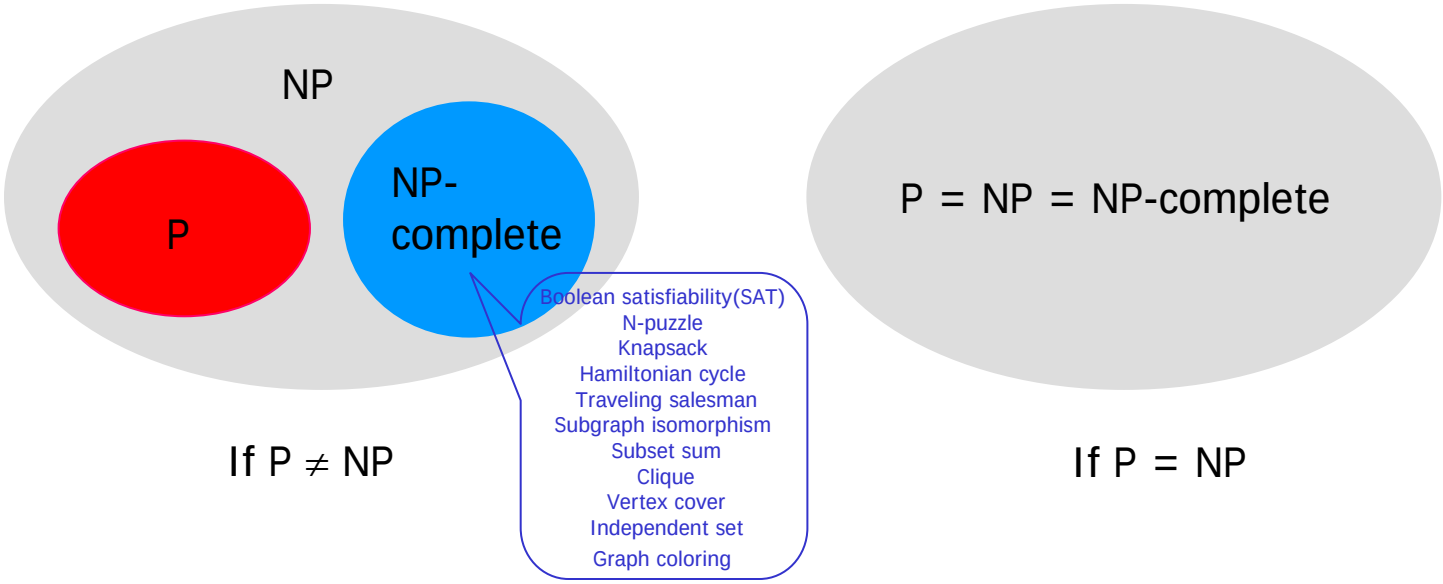


Deep Questions in Computer Science

- Does P = NP ?
- What is computable?
- What is intelligence?
- What is system complexity?

The $1M Question: Does P = NP?

- The most important open problem in theoretical computer science. The Clay Institute of mathematics offers **one million** dollar prize for solution!
  - http://www.claymath.org/Millennium_Prize_Problems/



What is Computable?

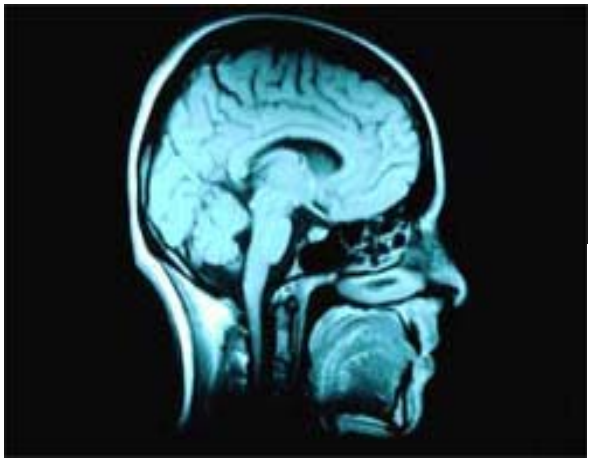
- What is are the power and limits of computation?

- What is computable when one considers The Computer as the combination of Human and Machine?

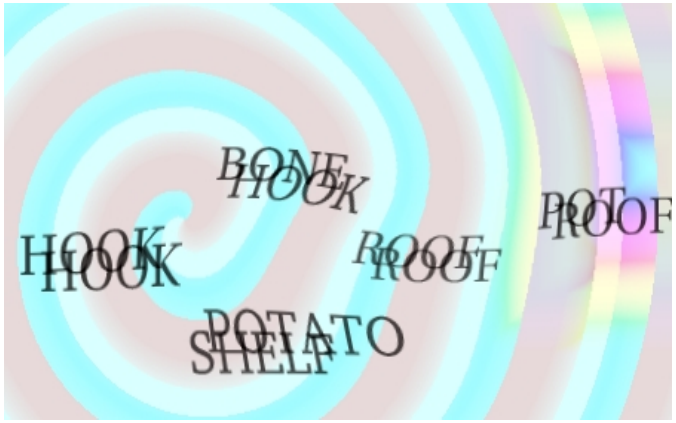
CAPTCHAs

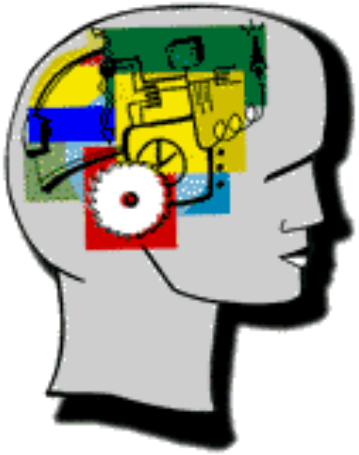
Labeling Images on the Web

What is Intelligence?

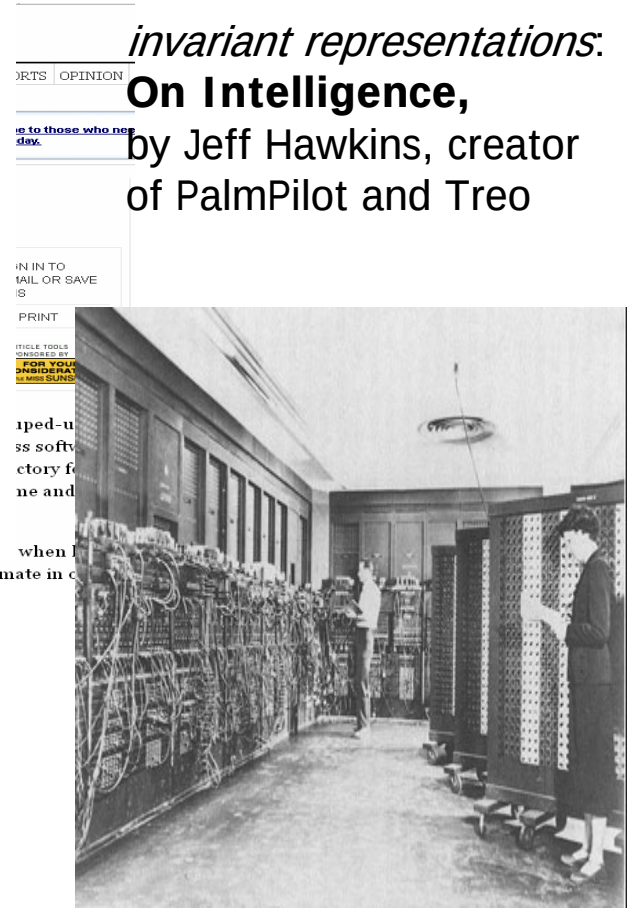


Human and Machine





invariant representations:
On Intelligence,
by Jeff Hawkins, creator of PalmPilot and Treo



"Computing Versus Human Thinking," **Peter Naur**, Turing Award 2005 Lecture, *CACM*, January 2007.

Human vs. Machine

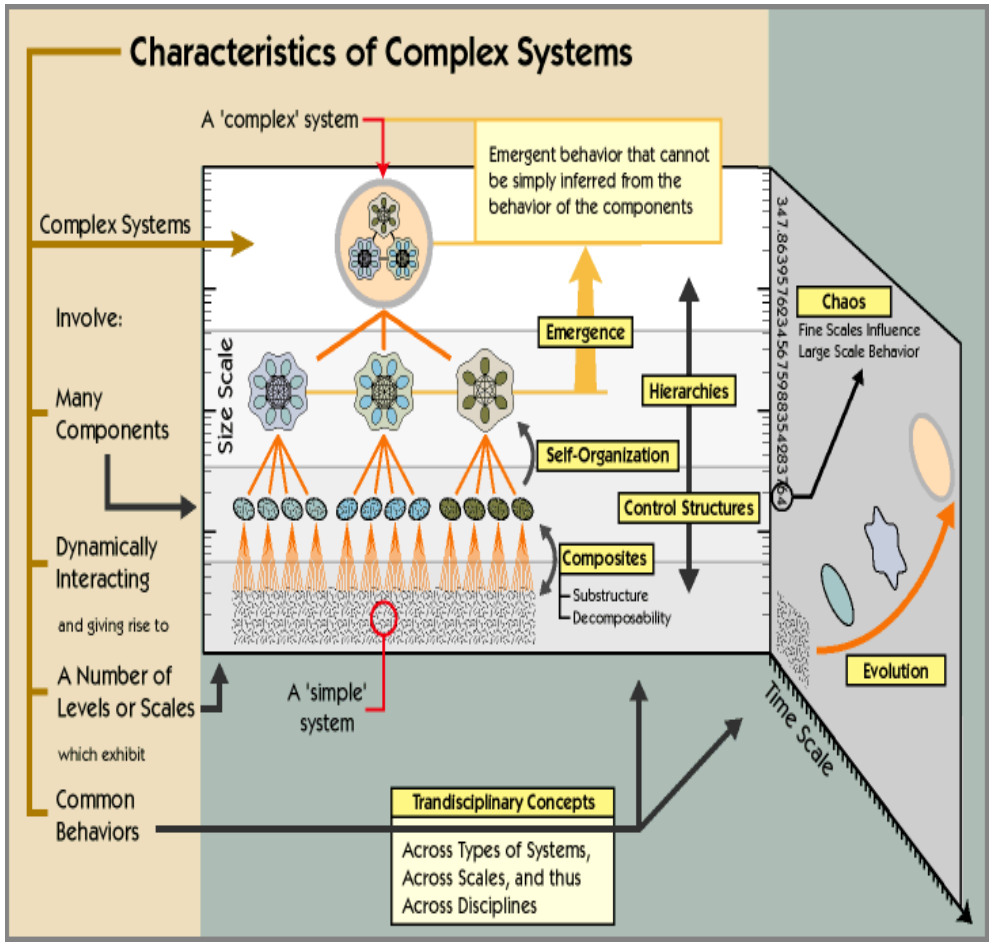
What is System Complexity?

Question 1: Do our systems have to be so complex?

- Can we build systems with simple designs, that are easy to understand, modify, and maintain, yet provide the rich complexity in functionality of systems that we enjoy today?

Further, observe:
- We have complexity classes from theory.
- We build complex systems that do amazing, but often unpredictable, things.

Question 2: Is there a meaning of system complexity that spans the theory and practice of computing?



Grand Vision for Society

- Computational thinking will be a fundamental skill used by everyone in the world by the middle of the 21st Century.
- Join us at Carnegie Mellon and the entire computing community toward making computational thinking commonplace.
  - www.cs.cmu.edu/ct

Spread the word!

To your fellow faculty, students, researchers, administrators, teachers, parents, principals, guidance counselors, school boards, teachers' unions, congressmen, policy makers, ...