Computational Thinking and Thinking About Computing

Jeannette M. Wing

President's Professor of Computer Science
Carnegie Mellon University

Outline

- Computational Thinking
  - A Vision for our Field
- Thinking about Computing
  - Drivers of our Field
  - 5 Deep Questions in Computing

Computational Thinking

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.
  - In research: scientists, engineers, …, historians, artists
  - In education: K-12 students and teachers, undergrads, …

J.M. Wing, "Computational Thinking," *CACM* Viewpoint, March 2006, pp. 33-35.
Paper off CISE AC website; paper and talks off http://www.cs.cmu.edu/~wing/

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!

Simple Daily Examples

- Looking up a name in an alphabetically sorted list
  - Linear: start at the top
  - Binary search: start in the middle
- Standing in line at a bank, supermarket, customs & immigration
  - Performance analysis of task scheduling
- Putting things in your child's knapsack for the day
  - Pre-fetching and caching
- Taking your kids to soccer, gymnastics, and swim practice
  - Traveling salesman (with more constraints)
- Cooking a gourmet meal
  - Parallel processing: You don't want the meat to get cold while you're cooking the vegetables.
- Cleaning out your garage
  - Keeping only what you need vs. throwing out stuff when you run out of space.
- Storing away your child's Lego pieces scattered on the LR floor
  - Using hashing (e.g., by shape, by color)
- Doing laundry, getting food at a buffet
  - Pipelining the wash, dry, and iron stages; plates, salad, entrée, dessert stations
- Even in grade school, we learn algorithms (long division, factoring, GCD, …) and abstract data types (sets, tables, …).

The First A to Computational Thinking

- Abstractions are our "mental" tools

- The abstraction process includes
  - Choosing the right abstractions
  - Operating simultaneously at multiple layers of abstraction
  - Defining the relationships the between layers

The Second A to Computational Thinking

- The power of our “mental” tools is amplified by our “metal” tools.

- Automation is mechanizing our abstractions, abstraction layers, and their relationships
  - Mechanization is possible due to precise and exacting notations and models
  - There is some “computer” below (human or machine, virtual or physical)

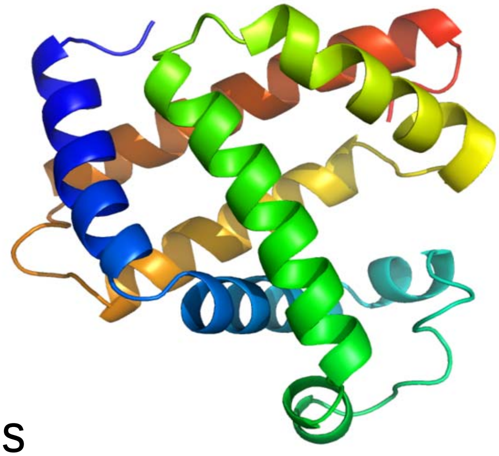
Two A's to C.T. Combined

- Computing is the automation of our abstractions
    - They give us the audacity and ability to scale.

- Computational thinking
    - choosing the right abstractions, etc.
    - choosing the right "computer" for the task

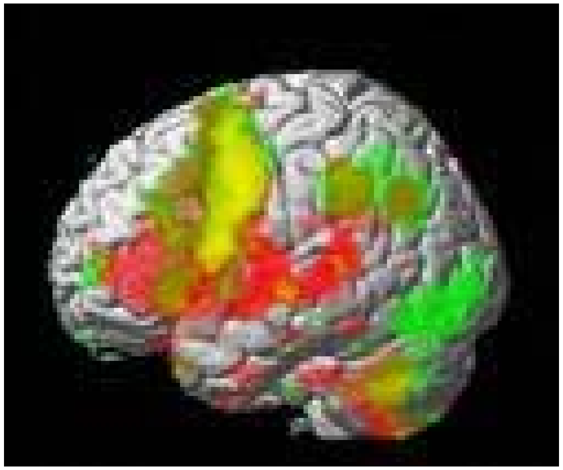
Research Implications

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



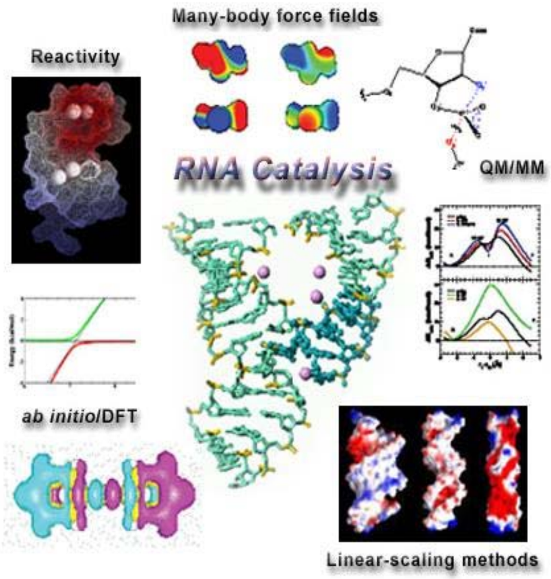
Brain Science

- 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

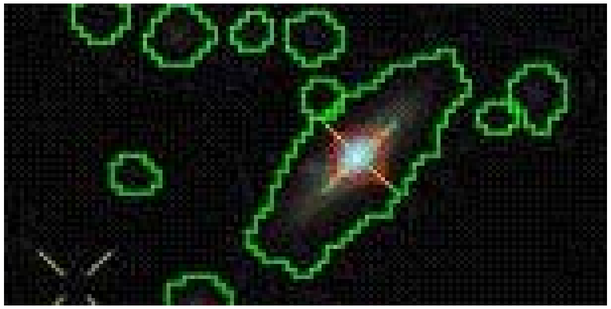


[York, Minnesota]

Geology

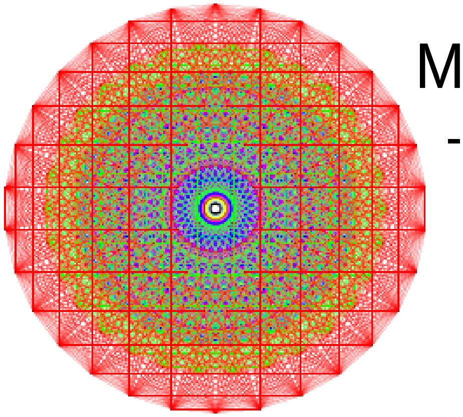
- Modeling the earth’s surface to the sun, from the inner core to the surface
- Abstraction boundaries and hierarchies of complexity model the earth and our atmosphere

CT in Other Sciences, Math, and Engineering



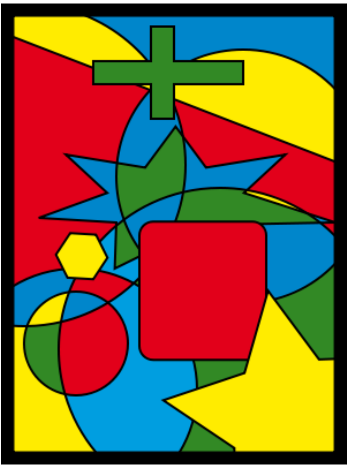
Astronomy

- 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

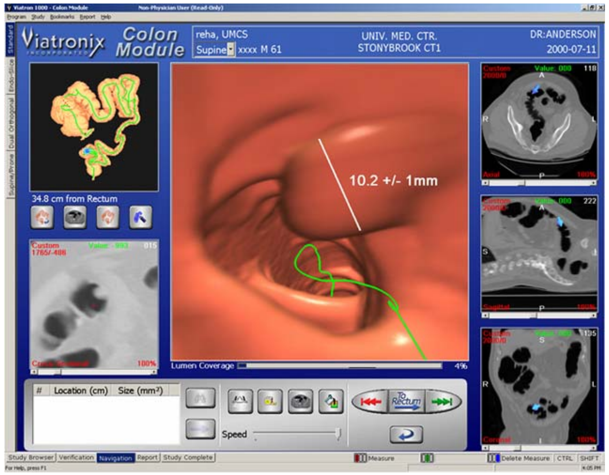
Social Sciences

- 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

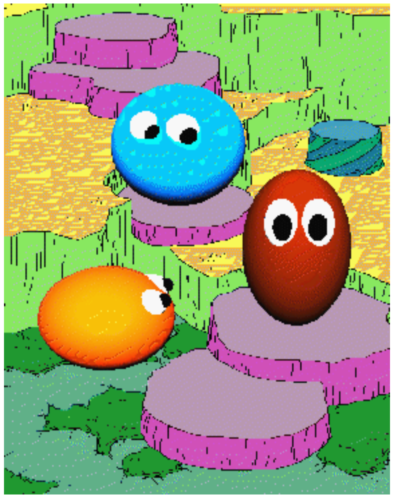


Law

- 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

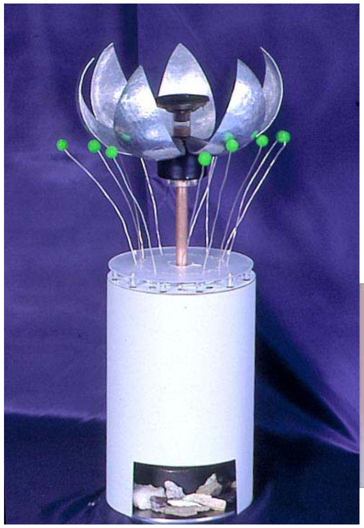
Entertainment



- 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

Jeannette M. Wing

Educational Implications

Pre-K to Grey

- 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
- Post-graduate
  - Executive and continuing education, senior citizens
  - Teachers, not just students

Question and Challenge to Community

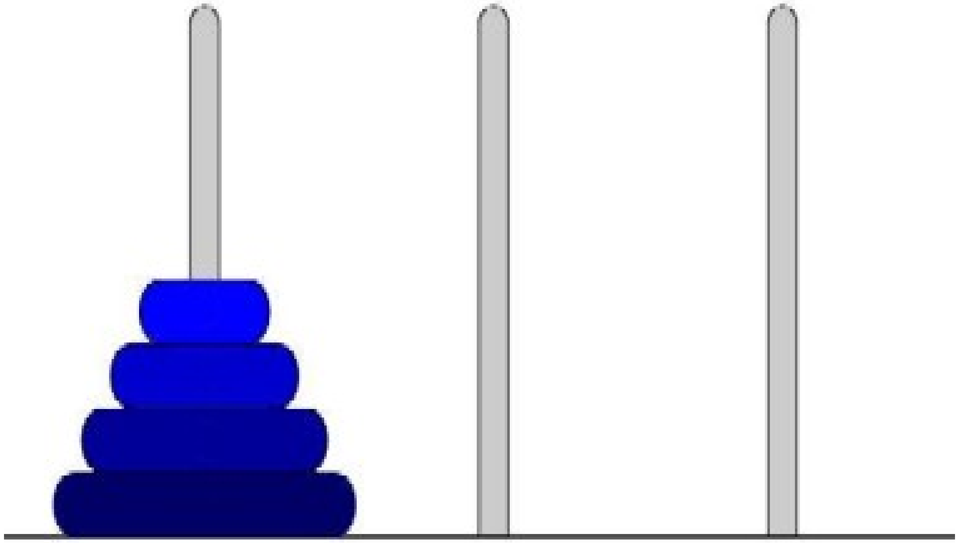
What are effective ways of learning (teaching) computational thinking by (to) children?

- What concepts can students best learn when? What should we teach when? What is our analogy to numbers in K, algebra in 7, and calculus in 12?

- We uniquely also should ask how best to integrate The Computer with learning and teaching the concepts.

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.

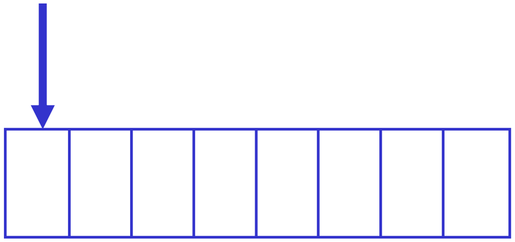


Data Abstraction and Representation

stack

queue

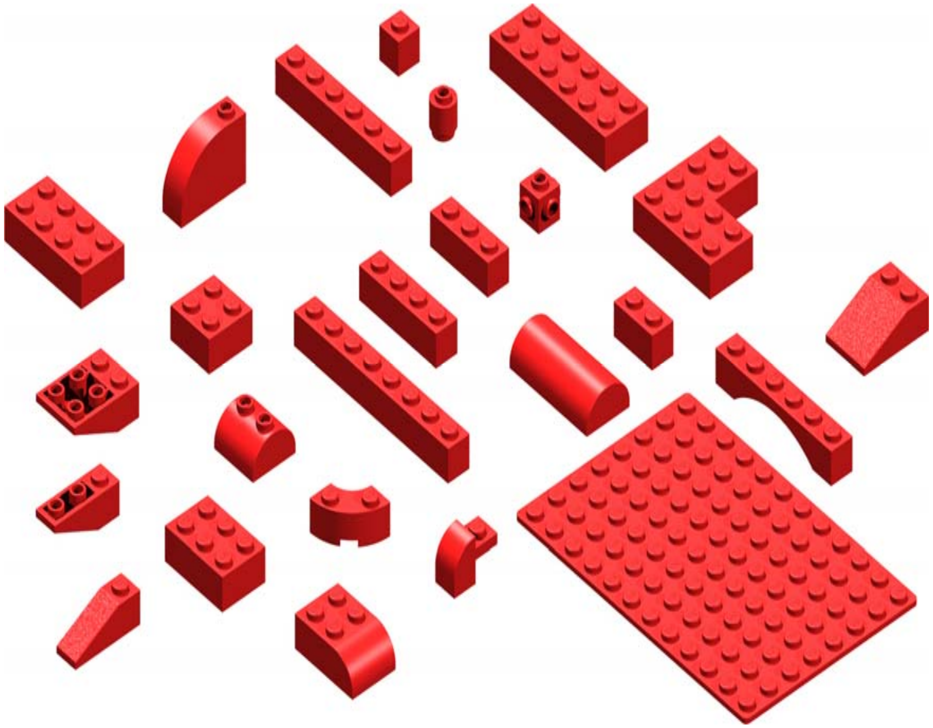
tree
(upside down)



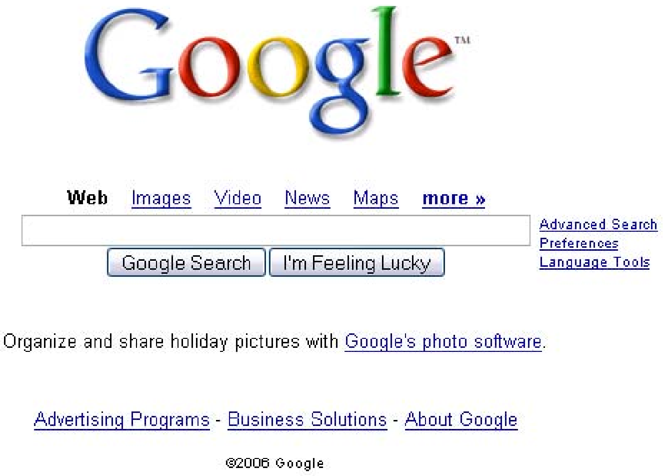
array and pointer

representation invariant

Composition and Decomposition



Sorting and Search

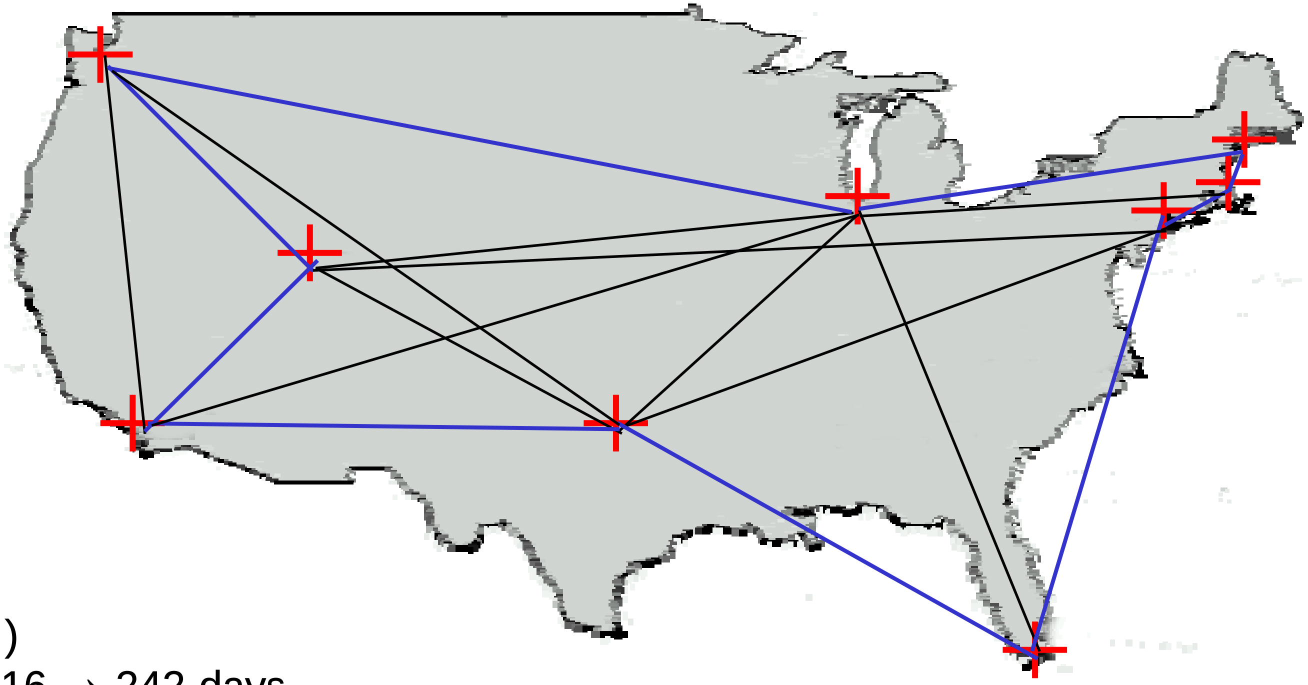


Organize and share holiday pictures with Google's photo software.

©2006 Google

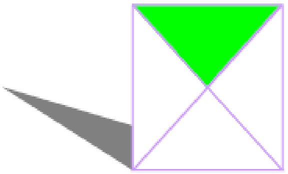
Intractability: Traveling Salesman

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

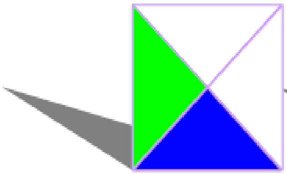


O(n!)
n = 16 → 242 days
n = 25 → 5x10^15 centuries

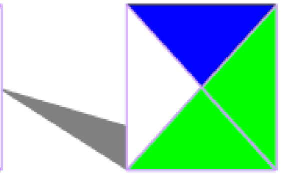
Undecidability: Tiling



1

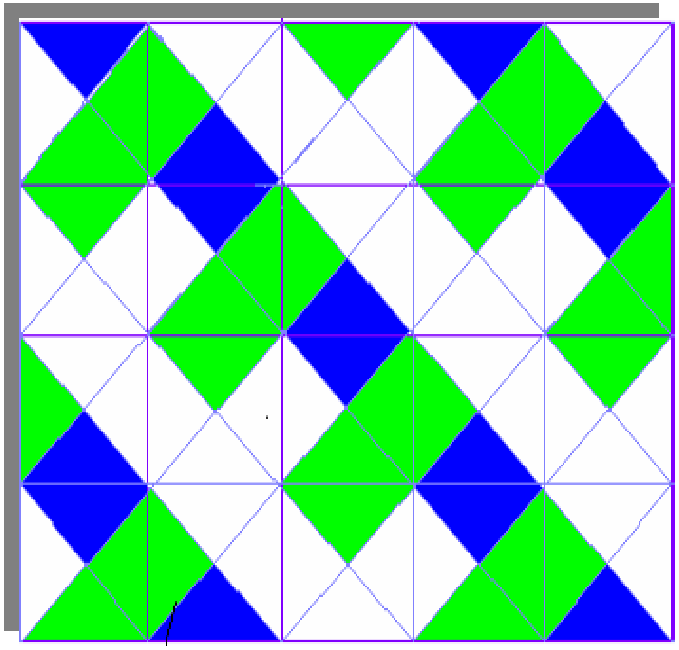


2



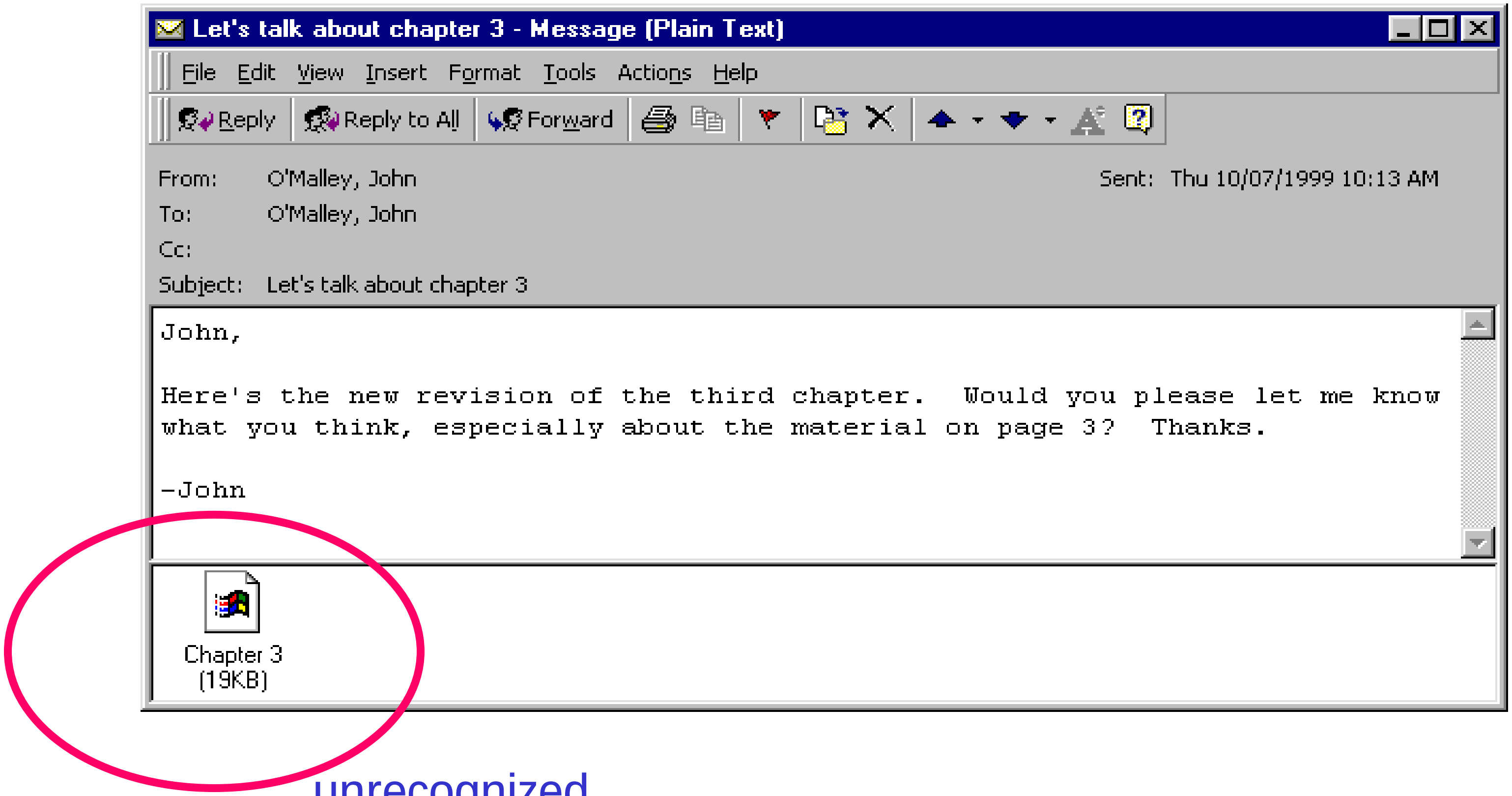
3

Can we tile the entire plane Z^2?



Example from David Harel

Data as Code and Code as Data


Let's talk about chapter 3 - Message (Plain Text)
File Edit View Insert Format Tools Actions Help
Reply
Reply to All
Forward

From: O'Malley, John

Sent: Thu 10/07/1999 10:13 AM

To: O'Malley, John

Cc:

Subject: Let's talk about chapter 3

John,

Here's the new revision of the third chapter. Would you please let me know what you think, especially about the material on page 3? Thanks.

-John

Chapter 3
(19KB)

unrecognized
email attachment

Recursion Revisited

The Y operator

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

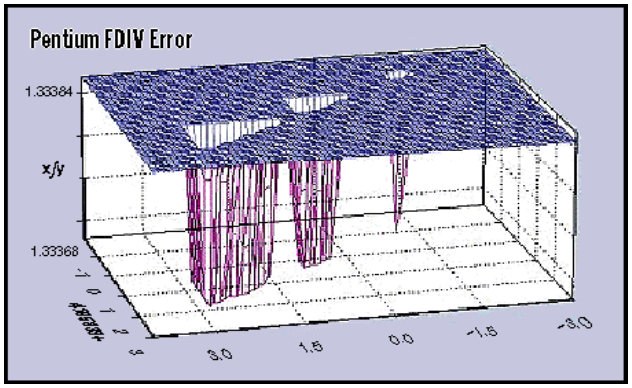
which satisfies the following equation

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

and is the basis of recursion in Computer Science.

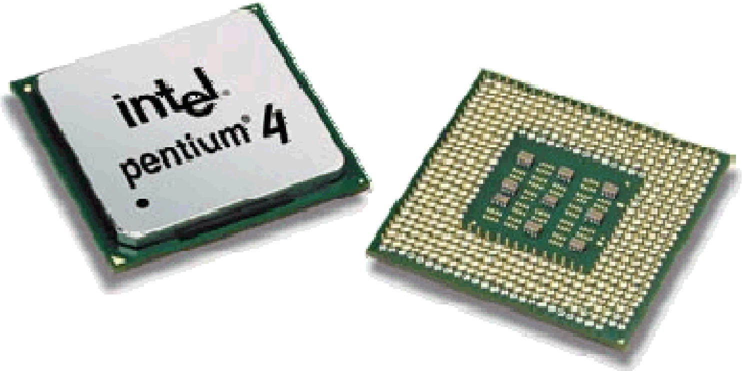
Y is the *fixed point* combinator in lambda calculus.

Correctness: Avoiding Bugs to Save Money and Lives

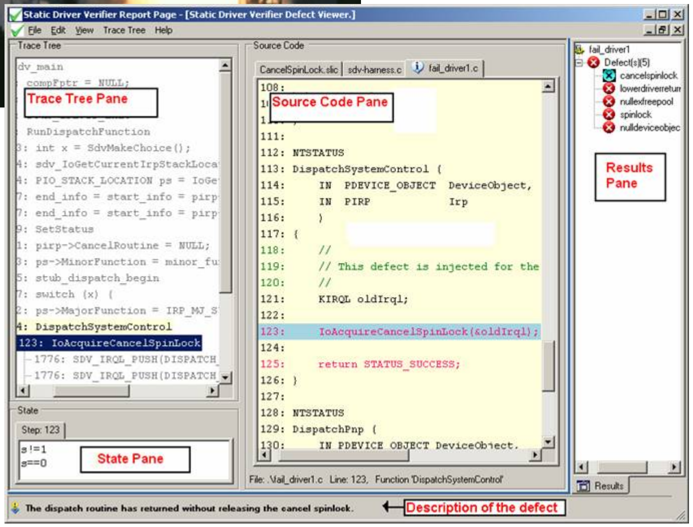


Intel Pentium FPU error

Ariane 5 failure



Now Intel uses formal verification.



```
dv_main
  compFptr = NULL;

  RunDispatchFunction
3: int x = SdvMakeChoice();
4: sdv_IoGetCurrentIrpStackLoca
4: PIO_STACK_LOCATION ps = IoGe
7: end_info = start_info = pirp
7: end_info = start_info = pirp
9: SetStatus
1: pirp->CancelRoutine = NULL;
3: ps->MinorFunction = minor_fu
5: stub_dispatch_begin
7: switch (x) {
2: ps->MajorFunction = IRP_MJ_S
4: DispatchSystemControl
123: IoAcquireCancelSpinLock
 -1776: SDV_IRQL_PUSH(DISPATCH_
 -1776: SDV_IRQL_PUSH(DISPATCH_
```

```
s!=1
s==0
```

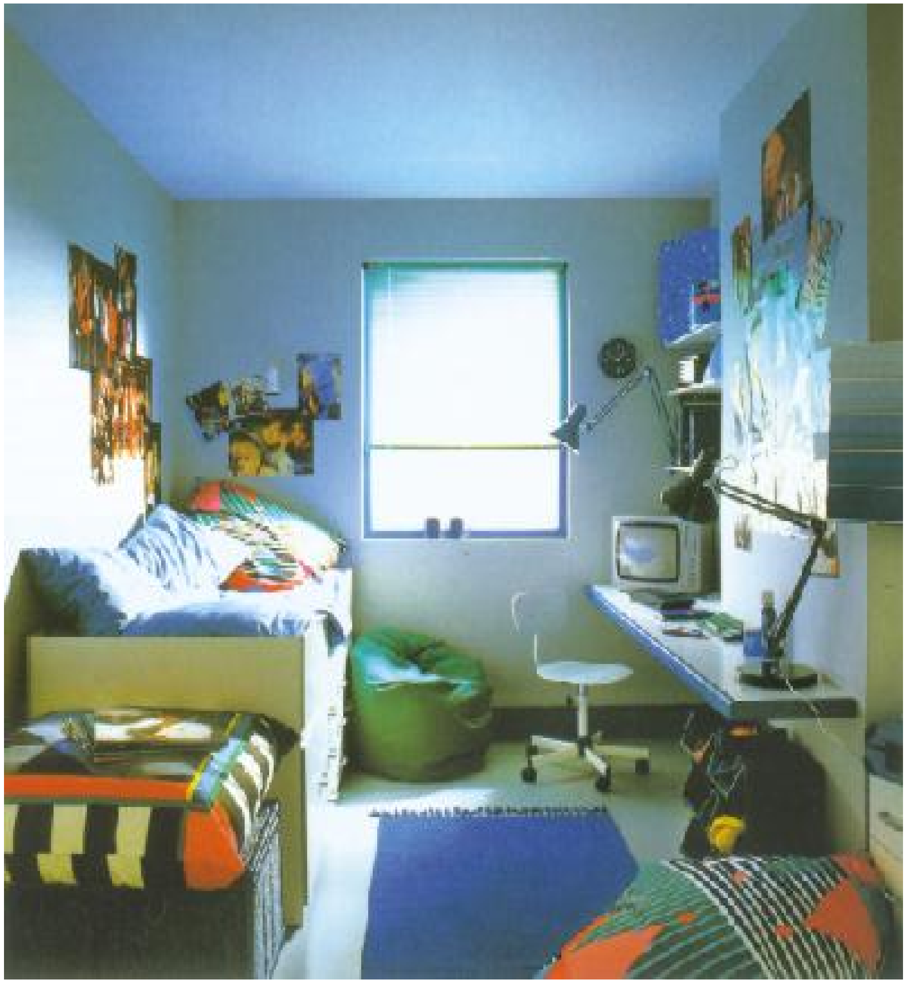
```
108:
111:
112: NTSTATUS
113: DispatchSystemControl (
114:     IN  PDEVICE_OBJECT  DeviceObject,
115:     IN  PIRP            Irp
116:     )
117: {
118:     //
119:     // This defect is injected for the
120:     //
121:     KIRQL oldIrql;
122:
123:     IoAcquireCancelSpinLock(&oldIrql);
124:
125:     return STATUS_SUCCESS;
126: }
127:
128: NTSTATUS
129: DispatchPnp (
130:     IN PDEVICE OBJECT DeviceObject.
```

File: .\fail_driver1.c Line: 123, Function 'DispatchSystemControl'

The dispatch routine has returned without releasing the cancel spinlock.

Now Microsoft uses formal verification.

Caching



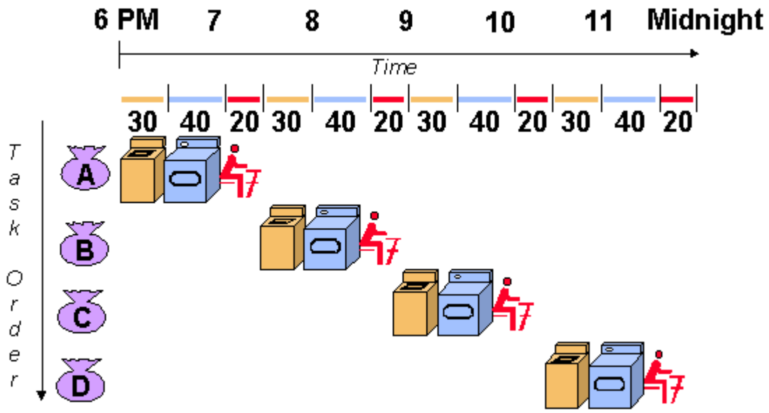
home

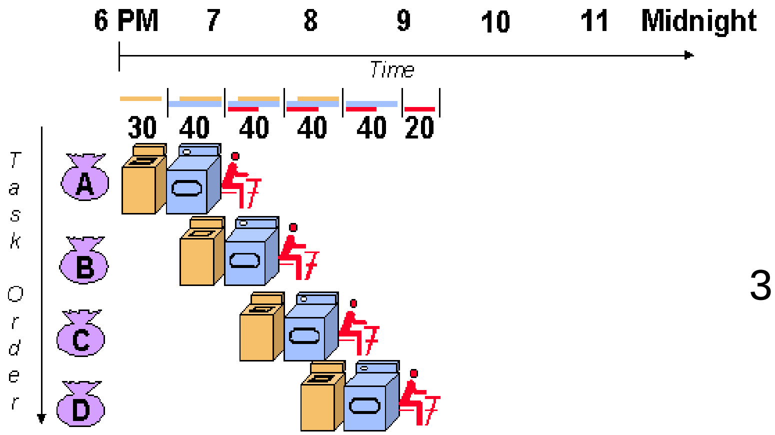
locker

knapsack

Pipelining: Doing Laundry

6 hours to do 4 loads

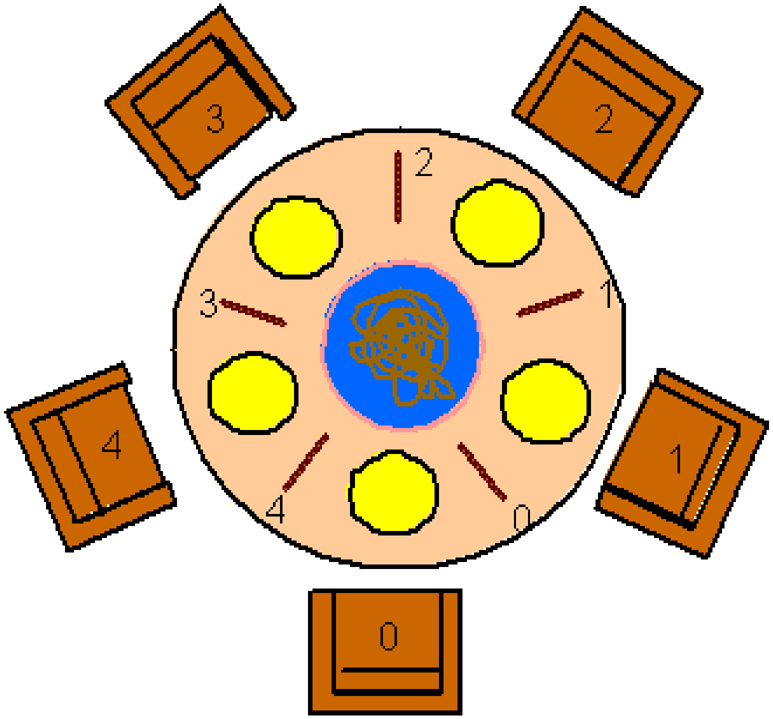




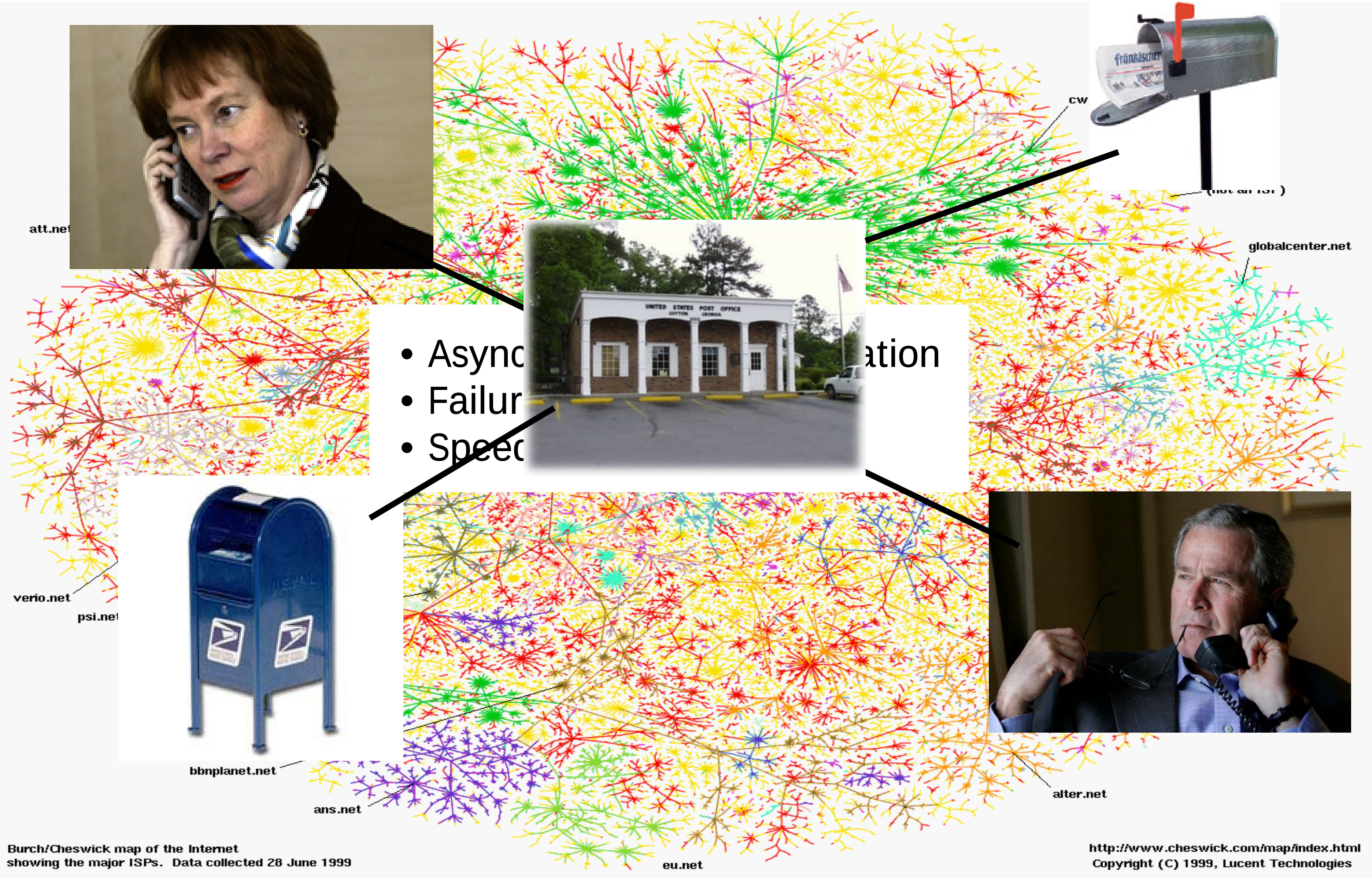
3.3 hours to do 4 loads

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



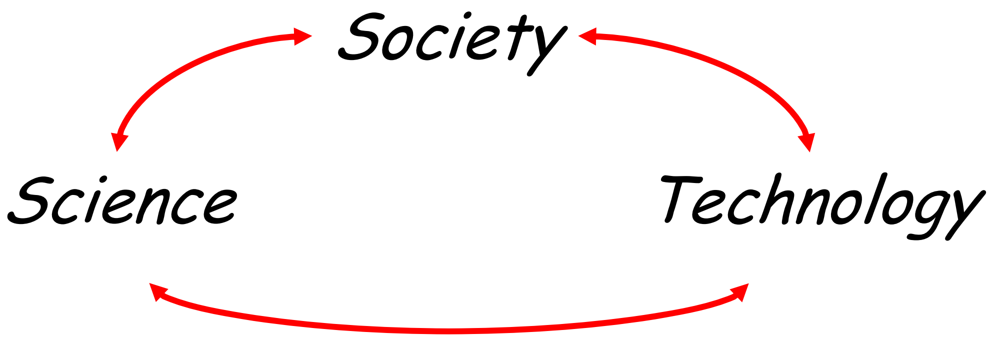
- Async ation
- Failur
- Speec

Burch/Cheswick map of the Internet
showing the major ISPs. Data collected 28 June 1999

http://www.cheswick.com/map/index.html
Copyright (C) 1999, Lucent Technologies

Thinking About Computing

Drivers of Computing



Society

Science

Technology

Technology Trends: Computing Substrates

- Moore’s Law will end in 10-15 years [Gordon Moore 9/18/07]

- Nanocomputing is here.
  - March 2006, IBM researchers build the first complete IC around a single carbon nanotube molecule.

- Biocomputing is here.
  - 1994, Adleman solves 7-point Hamiltonian path problem with DNA computing
  - 2004, Shapiro, Benenson, Gil, Ben-Dor, and Adar of Weizmann Institute announce in *Nature* the construction of a DNA computer

- Quantum is coming?
  - “Quantum Cryptography to Secure Ballots in Swiss Election,” *Network World,* Oct 11 2007

- Bio-Nano-Quantum
  - “Fabrication of Photonic Transfer DNA-Quantum Dot Nanostructures,” Heller, Sullivan and Dehling, Nanotech 2005.
  - “Economical Fabrication of Quantum Dot-Electronics Using Biofunctionalized Protein Nanotubes as Building Blocks, Masui, NSF CAREER award

More Technology Trends

- Devices
  - 2 billion cell phones in the world; RFID tags; sensors everywhere
  - A BMW is "now actually a network of computers" [R. Achatz, Seimens, Economist Oct 11, 2007]
  - Robots in your home

- Information
  - Drowning in data; sensors everywhere; storage is cheap; information overload

- Communication
  - Femto cells—personal base stations
  - Web 3.0 (semantic web)
  - Virtual worlds: Second Life is today's Mosaic

- Brainy machines
  - IBM and EPFL's Blue Brain Project: to create a biologically accurate, functional model of the brain
  - www.numenta.com : software platform for intelligent computing modeled after human neocortex

Users and Society

- Expectations: 24/7 availability, 100% reliability, 100% connectivity, instantaneous response, store anything and everything forever, ...

- Classes: young to old, able and disabled, rich and poor, literate and illiterate, …

- Numbers: individual → cliques → acquaintances → social networks → cultures → populations

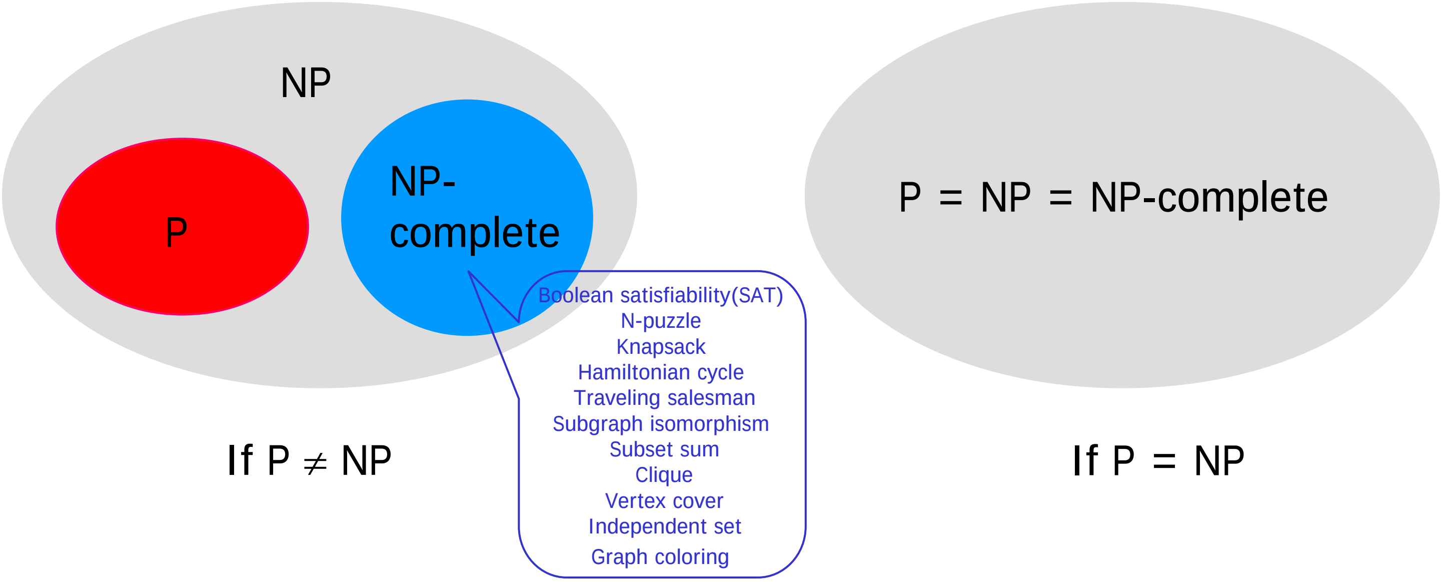
- The Internet/Web is a great equalizer.
  - What about privacy? Anonymity to accountability
  - When will it stop being free?
  - Will it continue to be self-regulating?

5 Deep Questions in Computing

- P = NP ?
- What is computable?
- What is intelligence?
- What is information?
- (How) can we build complex systems simply?

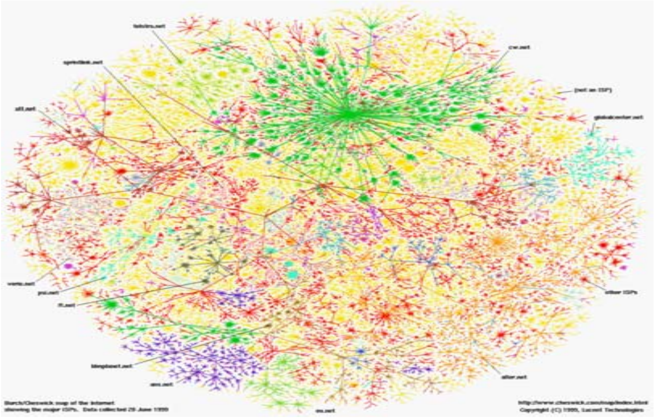
The $1M Question: Does P = NP?

- 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/

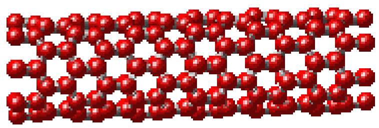


What is Computable?

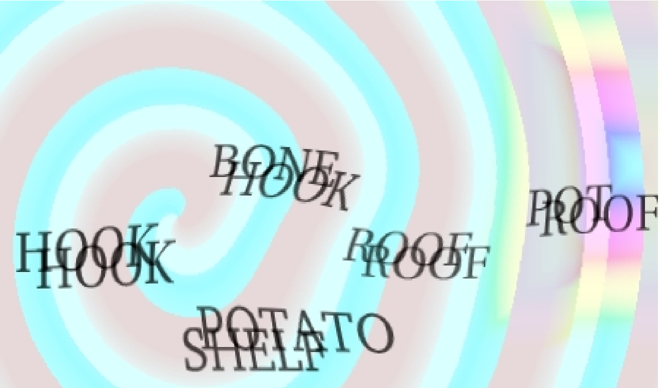
- What are the power and limits of computation?
- What is a computer?



- Not just a PC anymore: The Internet, server farms, supercomputers, multi-cores, …, nano, bio, quantum, etc.



- What is the power of computing, by machine and human together?

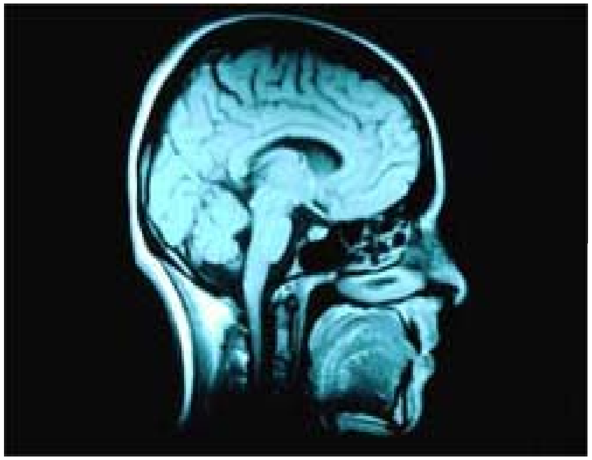


The ESP Game

beta

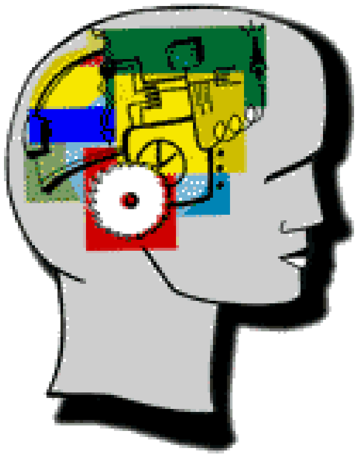
As seen on CNN and newspapers around the world!

What is Intelligence?



Human and Machine

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



ORTS OPINION

e to those who nee
day.

N IN TO
MAIL OR SAVE
S

PRINT

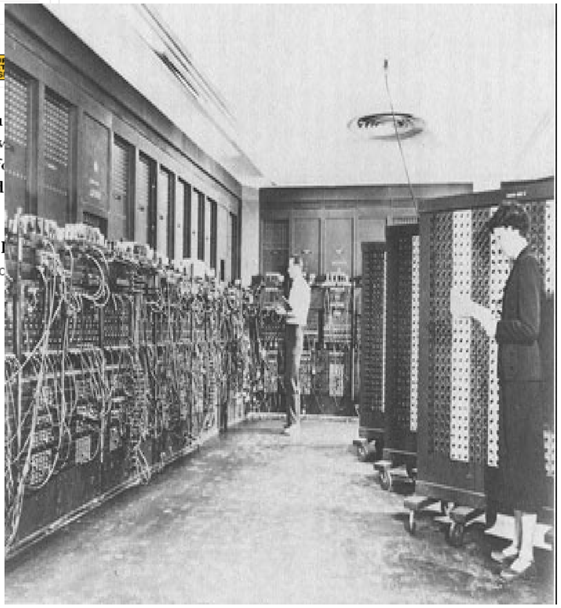
TICLE TOOLS
ONSORED BY
FOR YOUR
ONSIDERAT
MISS SUNS

iped-u
ss softv
ctory f
ne and

when I
lost Game 2 by walking into a checkmate in c

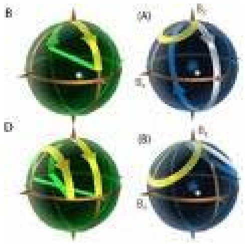
"Computing Versus Human Thinking," **Peter Naur**, Turing Award 2005 Lecture, *CACM*, January 2007.

Human vs. Machine

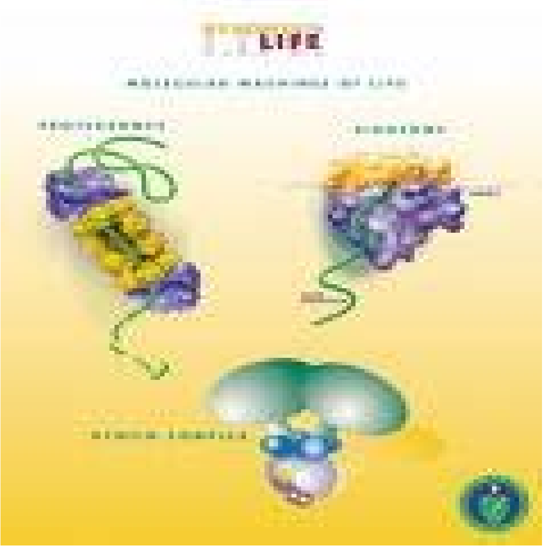


What is Information?

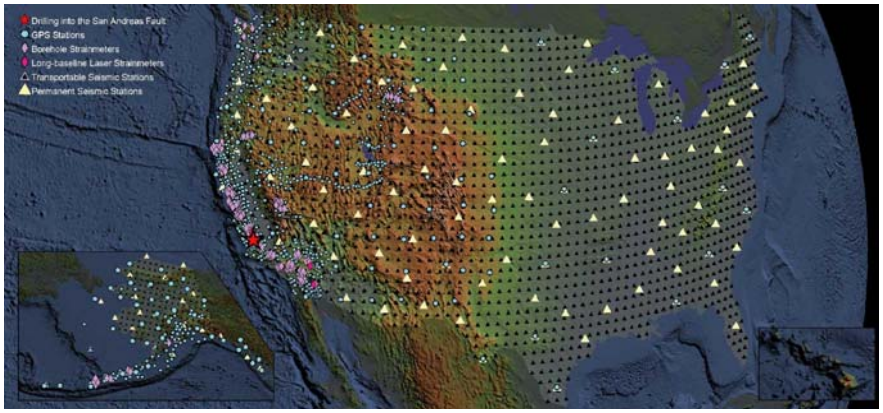
- From nature
  - It’s not just 0’s and 1’s
    - Qubits



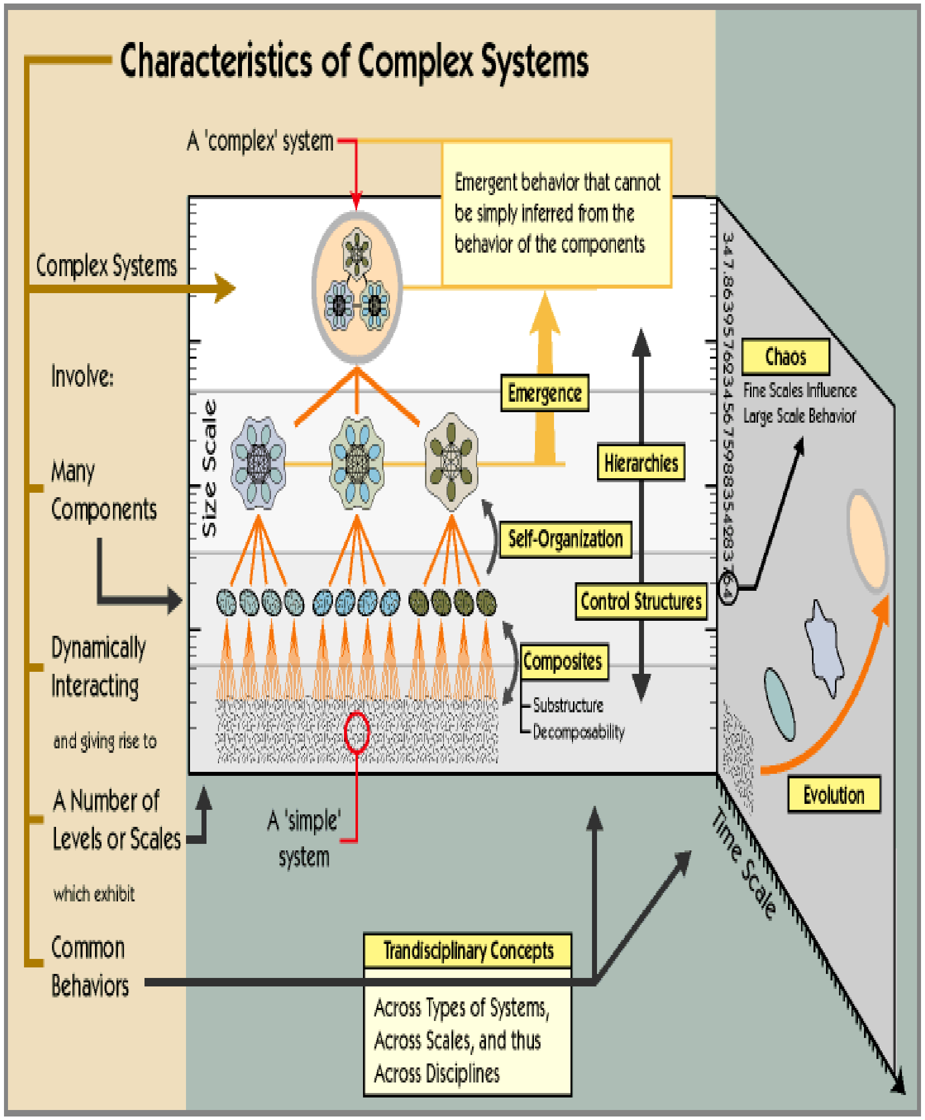
  - “Biology is an information science.”
    - Geology too.
  - Molecules/chemicals are processors of information (computer), carriers ofinformation (storage), and channels of information (communication)



- …To knowledge
  - We are drowning in data. Data is dirt; knowledge is gold.


Drilling into the San Andreas Fault
GPS Stations
Borehole Strainmeters
Long-baseline Laser Strainmeters
Transportable Seismic Stations
Permanent Seismic Stations

(How) Can We Build Complex Systems Simply?



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

Question: Is there a complexity theory for systems as there is for algorithms?

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

Question: 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?

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

Spread the Word!

- Help make computational thinking commonplace
- Help explain the science in computer science

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

Thank you!