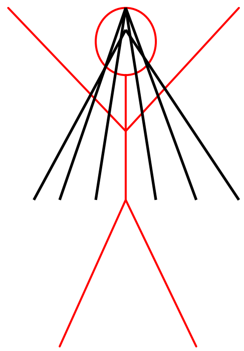
Computational Thinking
and
CS@CMU



Jeannette M. Wing

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

Grand Vision for the Field

- Computational thinking will be a fundamental skill used by everyone in the world by the middle of the 21^{st} Century.
  - Just like reading, writing, and arithmetic.
  - Imagine every child knowing how to think like a computer scientist!

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
  - Game Theory
    - CT is influencing Economics
      - Electronic marketplaces, ad placement, multi-agent systems, security, and networking
  - Nanocomputing
    - CT is influencing Chemistry
      - Molecular-scale computing based on reconfigurable fabric makes the chemistry easier.
  - 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

- Conceptualizing, not programming
  - Computer science is not just computer programming
- Fundamental, not rote skill
  - A skill needed by everyone 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
- Ideas, not artifacts
- It's for everyone
  - C.T. will be a reality when it is so integral to human endeavors that it disappears as an explicit philosophy.

For More

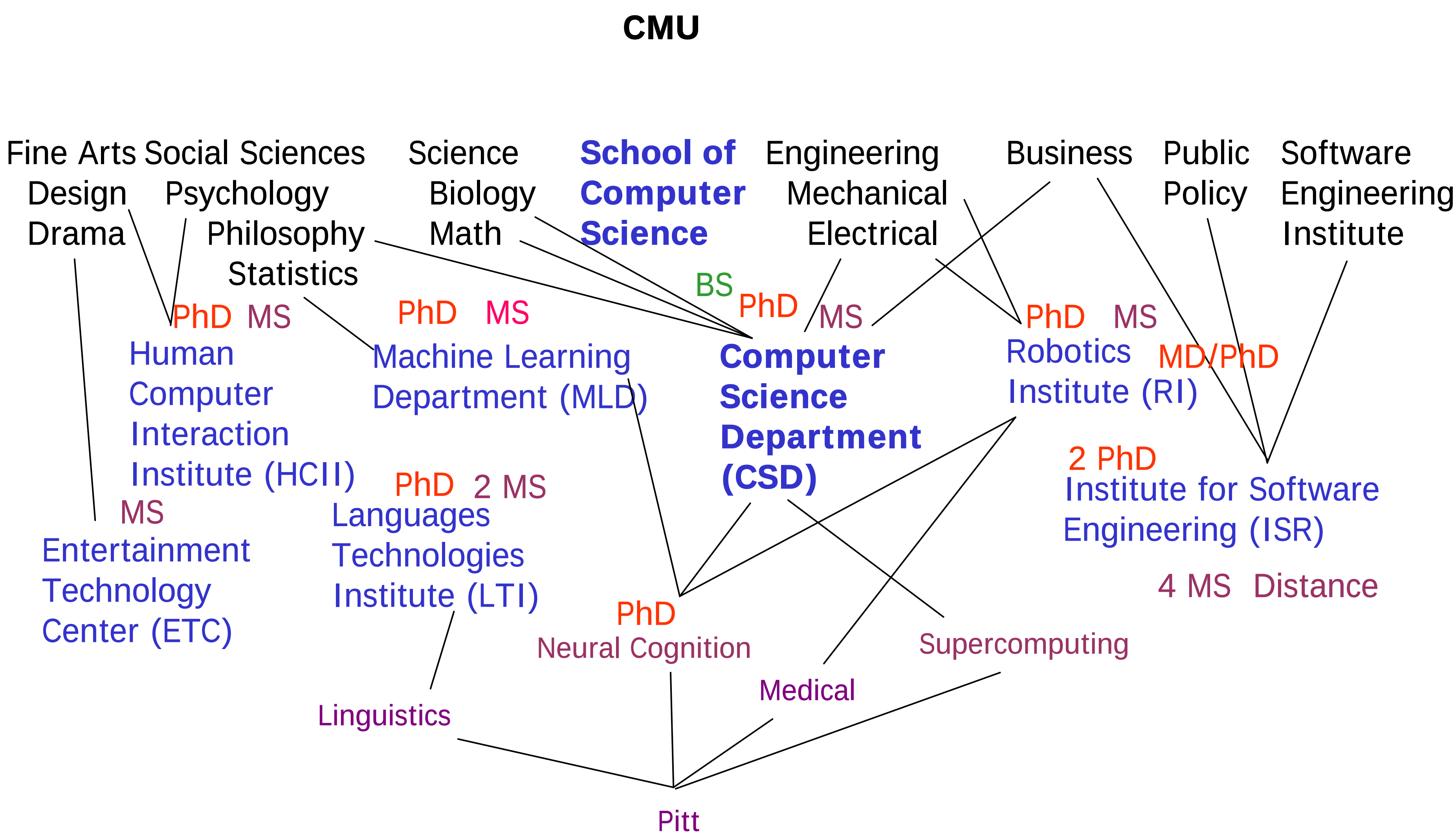
- “Computational Thinking,” Jeannette M. Wing, *CACM* Viewpoint, March 2006, pp. 33-35.
- http://www.cs.cmu.edu/~wing/

Computational Thinking

~~Computer Science~~ at Carnegie Mellon

CS@CMU

Computing at Carnegie Mellon



Computational Thinking at Carnegie Mellon

- Computational and applied mathematics
- Computational biology
- Computational chemistry
- Computational design
- Computational economics
- Computational finance
- Computational linguistics
- Computational mechanics
- Computational neuroscience
- Computational photography
- Computational physics
- Computational and statistical learning

- Algorithms, combinatorics, and optimization (joint between CS, math, business)
- Computation, organizations, and society
- Computer-aided language learning (CS and modern languages)
- Computer music
- Electrical and computer engineering
- Electronic commerce (CS and business)
- Entertainment technology (CS and drama)
- Human-computer interaction (CS, design, and psychology)
- Language technologies (CS and linguistics)
- Logic and computation (CS and philosophy)
- Pure and applied logic (CS, math, and philosophy)
- Robotics (CS, electrical and computer engineering, and mechanical engineering)

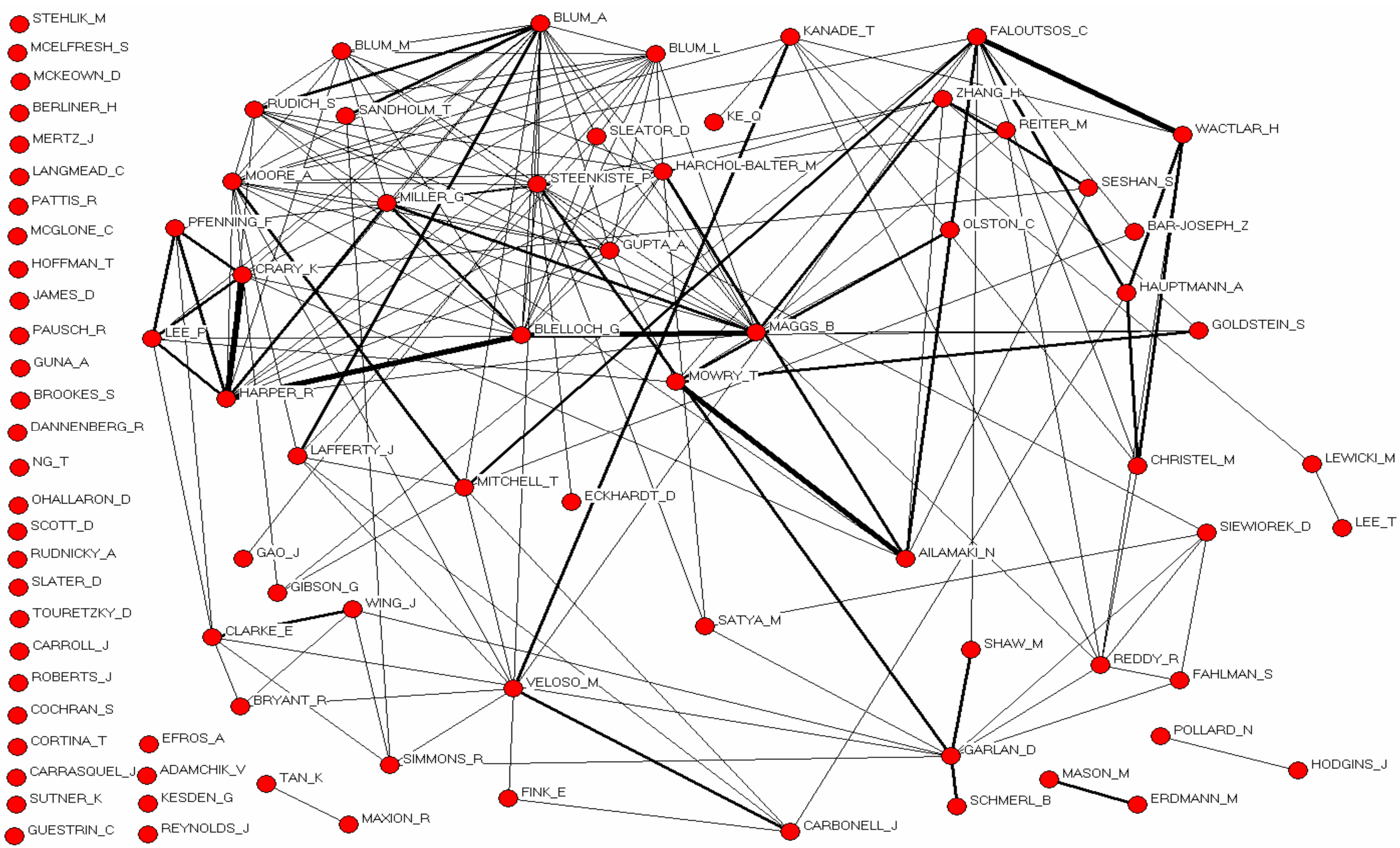
SCS Numbers at a Glance

- 215 faculty
- 213 courses on the books
- 540 bachelors students
  - including a handful of HCI double majors
- 235 masters students across 11 programs
- 400 doctoral students across 9 programs

CS@CMU Distinguishing Characteristics

- Research Style
  - High quality, high impact
  - Collaborative, interdisciplinary
  - We build things. For real users. Systems ⇔ Theory
  - We think big.
- Leadership in Education
  - PhD program: “Research from Day One,” BF, etc.
  - Undergrad program: challenging and unique curriculum, devoted faculty
  - Elite 5th Year MS
- Women in Computer Science
  - Twice the national average in BS degrees
- Supportive Culture
  - Reasonable Person Principle
  - Collective responsibility
  - Presume success
- Organizational Structure
  - Expanding Universe Model
  - Lack of rigid admin boundaries

CSD Collaboration Network [Carley 2004]



What Do We Do?

SCS's Research Enterprise

New, emerging areas (e.g., CS + X):
- Computational biology
- Computational astrophysics
- Nanocomputing
- Foundations of privacy
- Computing technology and society
- …

RI

AI: robotics, vision

LTI

AI: natural language processing, speech

CSD

New, emerging areas in Theory:
- game theory
- …

New, emerging areas in AI:
- optimization
- coaching
- …

Theory: algorithms, complexity, semantics
Systems: computer architecture, O/S, distributed systems, networking, databases, performance modeling, graphics, programming languages, formal methods
AI: planning, learning, search, cognition, computational neuroscience

New, emerging areas in Systems:
- pervasive computing
- trustworthy computing
- post-Moore's Law computers
- …

Systems: software engineering, public policy, e-commerce

ISR

Systems: human-computer interfaces

HCII

AI: machine learning

MLD

What We Do: Research in CSD

- Algorithms and Complexity
- Artificial Intelligence
- Computational Molecular Biology
- Computational Neuroscience
- Computer Architecture
- Databases
- Formal Methods
- Graphics
- *Human Computation*
- Human-Computer Interaction
- Large-Scale Distributed Systems
- Machine Learning
- Mobile and Pervasive Computing
- Networking
- Principles of Programming
- Robotics
- Scientific Computing
- Security
- Software Engineering
- Technology and Society
- Vision, Speech, and Natural Languages

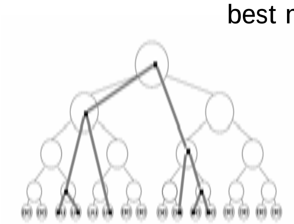
Three Mosaics

Some Highlights

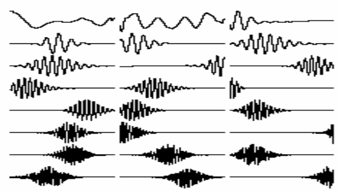
Quantifiable Results

Using deconvolution to correctly identify 15% more cycling genes in yeast cells when compared to using observed values alone [Bar-Joseph]

A new search algorithm to solve the k-nearest neighbor problem with a 10-fold speedup over the best metric-tree algorithm [Moore]



A new data structure for representing n-vertex unlabeled graphs using O(n) bits and supporting adjacency and degree queries in constant time [Blelloch]



A new spike representation of auditory signals resulting in a coding 3x more efficient than MP3 [Lewicki]

Applied Math Models

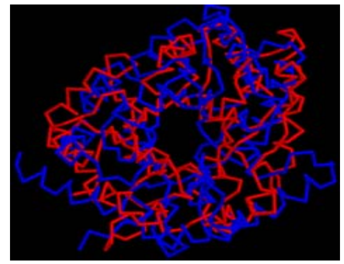
Game theory

Automated mechanism design applicable to divorce settlements and tie-breaking rules [Sandholm]

Near optimal on-line auctions [A. Blum]

Predict Internet stability wrt congestion control if end-points act selfishly [Seshan]

Knot theory



A new topological approach to detecting protein similarity leading to a representation of proteins by line weavings [Erdmann]

Metric spaces

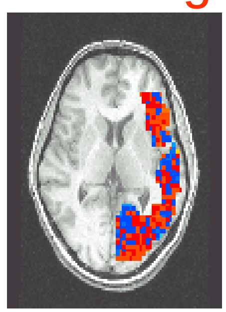
Complexity of metric spaces and applications to TSP-like problems, networking, web-page clustering [Gupta]

Understanding Intelligence

Micro

Theory of consciousness [M. Blum, Rudich, A. Blum]

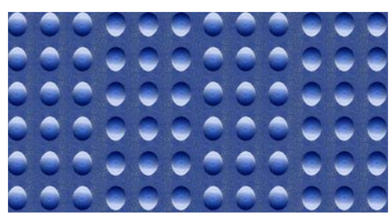
Use fMRI data to construct better cognitive models [Mitchell]



Neural representation of space (Where am I?) in rats (and robots) [Touretzky]

CMU

Role of feedback from higher visual areas on early (V1 and V2) areas by studying awake behaving monkeys [T.S. Lee]



Macro

Multi-agent (robots and humans) planning and learning [Veloso]

Multi-participant (robots and humans) dialog and conversation [Rudnicky]

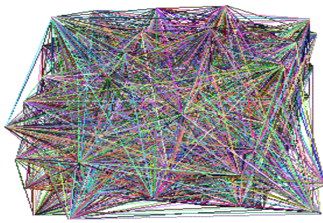
CAPTCHA, ESP: Using hard AI problems to solve crypto [M. Blum]

Some Common Themes

Data-Driven

Lots of data

Texture synthesis uses Shannon's N-grams info theoretic technique to quilt radishes, rocks, and yogurt. [Efros]

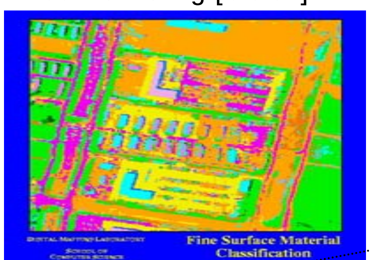


Fractals and power laws to model sensor data, network graphs, multimedia data, protein interactions. [Faloutsos]

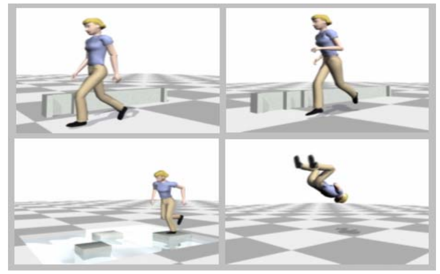
Use server logs from content-delivery networks to estimate interdomain Web traffic flow. [Maggs]

Manage distributed dynamic data applied to Web monitoring [Olston]

Anomaly detection based on real network data [Maxion, Tan]



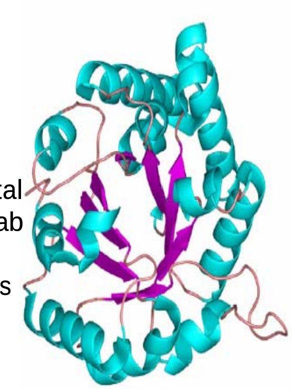
Multispectral imagery
+ photogrammetric knowledge
+ large-scale databases = digital maps
[Cochran, McKeown]



Use motion capture data to render human behavior efficiently [Hodgins]

Sparse data

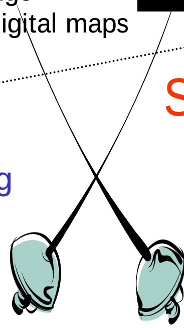
With sparse experimental data (to minimize wet lab cost), determine 3-D structures and dynamics of nucleic acids and proteins [Langmead]



Making use of labeled and unlabeled data [A. Blum, Lafferty]

Use precomputed data-driven deformable object simulation for computer animation, video games, reality based modeling, manufacturing and tissue simulation. [James]

Secret Weapons



Machine learning

For many, many things [Bar-Joseph, A. Blum, Efros, Lafferty, Langmead, Lewicki, Mitchell, Moore, Sandholm, Veloso]

Probability and statistics

SYNC: Scheduling Your Network Connections [Harchol-Balter]

Distributed inference in sensor networks [Guestrin]

Generalized Chernoff bounds on event probabilities for graphical models [Lafferty]

Type theory

$$\frac{\Delta;\Gamma \vdash M : \diamond A \qquad \Delta;\boxed{x{:}A} \vdash N \div B}{\Delta;\Gamma \vdash \mathsf{letposs}\ x{:}A\ \mathsf{be}\ M\ \mathsf{in}\ N \div B}$$

ConCert: grid computing [Crary, Harper, Lee, Pfenning]

Separation logic

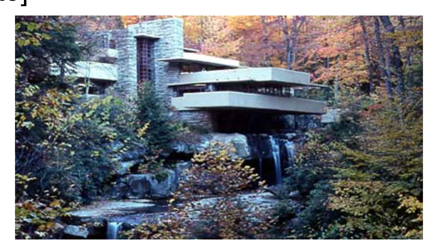
For concurrency [Brookes, Reynolds]

Model checking

For hybrid systems, software, and security [Bryant, Clarke, Wing]

Software architecture

Rainbow: runtime adaptation of self-managing systems [Garlan, Schmerl, Steenkiste]



Economics, decision theory

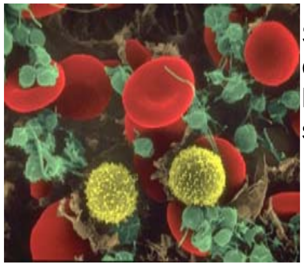
Value-driven software engineering [Shaw]

E-commerce, voting, auctions [Sandholm]

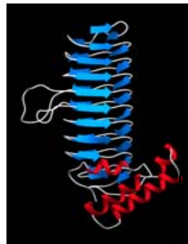
Some Big and Wild Projects

Interdisciplinary/Collaborative (Surprises)

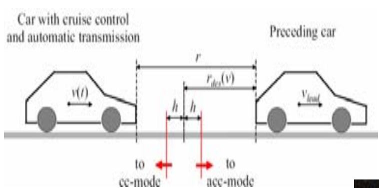
Outside CSD



Simulating blood flow, with computational fluid dynamicists, hemorheologists, ..., transplant surgeons [Miller, Blelloch]

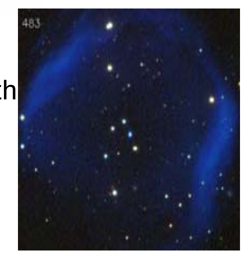


Use of convolution integrals for modeling super-secondary structures in proteins, with Pitt biologist [Carbonell]

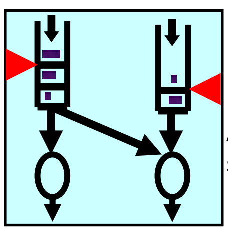


Safety of adaptive cruise control, with ECE; of insulin pump, with Chem Eng [Clarke]

Astronomy databases, with astrophysicists [Ailamaki]



Tablet PCs in 15-100, with HP and MS [Guna]



Analysis of multiserver systems, with Tepper [Harchol-Balter]

Within CSD

Theorem-proving cell phones [Pfenning, Reiter]

Model checking Proof-Carrying Code

C CMU[Clarke, P. Lee]

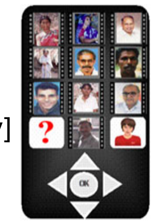
Large/Integrative Systems

Macro

100 Mbps to 100 million homes [Zhang]

Internet-Suspend-Resume on campus [O'Hallaron, Satya]

PCtvt [Reddy]



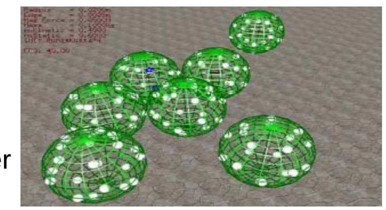
Informedia->Caremedia-> Quality of Life Technology Institute: care of elderly and chronically disabled and ill [Christel, Gao, Hauptmann, Ng, Wactlar]

Humanoid robots [Hodgins, Kanade]

RADAR/CALO [Carbonell, Fahlman, Fink, Moore, Rudnicky, Siewiorek, Veloso]

Micro

Claytronics: Synthetic reality through programmable matter [Goldstein, Mowry]



Slashdot/Wired: Past and Future?

Simulating trees blowing in the wind [James]

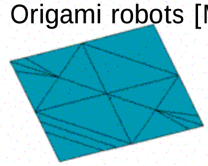
Two robot hands shaking hands [Pollard]

Robot team building a 4-beam structure in space [Simmons]

Unmanned aerial vehicles [Kanade, Ke, Veloso]

Query-by-Humming [Dannenberg]

Origami robots [Mason]



Building virtual worlds [Pausch]

3-D Quake [Seshan]

Educational Portfolio in SCS at CMU

- 7+5 Ph.D. programs
  - Major: Computer Science, Robotics, Language Technologies, Human-Computer Interaction, Software Engineering, Computational Statistics, Computers, Organizations, and Society
  - Joint/Special: Algorithms, Complexity, and Optimization, Computational Biology, Neural Basis of Cognition, Pure and Applied Logic, Ph.D./M.D. with Univ. of Pittsburgh, Neural Computation
- 11 M.S. programs
  - Professional, e.g., Software Engineering, Information Technology, Human-Computer Interaction, Entertainment Technology
  - Academic, e.g., Robotics, Language Technologies, Fifth-Year Master's in Computer Science
- 1 B.S. program
  - Taught primarily by CSD faculty

Computational Thinking in Our Education

- Graduate: See previous slide
- Undergraduate courses
  - 15-251 Great Theoretical Ideas in Computer Science
    - Audience: freshmen majors
    - Topics: recursion, number theory, probabilistic methods, algebraic structures, graphs, matching, finite automata, Turing machines, Big-O, diagonalization, proof, reduction, complexity
  - 15-105 Principles of Computation
    - Audience: freshmen non-majors
    - Topics: algorithms, Big-O, data structures, invariants, programming language paradigms, induction, intractabilitiy, computability, Turing machines, pipelining, distributed computing, operating systems, automated computation, game trees, artificial intelligence
- Outreach
  - CS4All
    - Summer program for high school CS, science, and math teachers, guidance counselors
  - Women@SCS Roadshow
    - Promoting Computer Science K-12 and college students

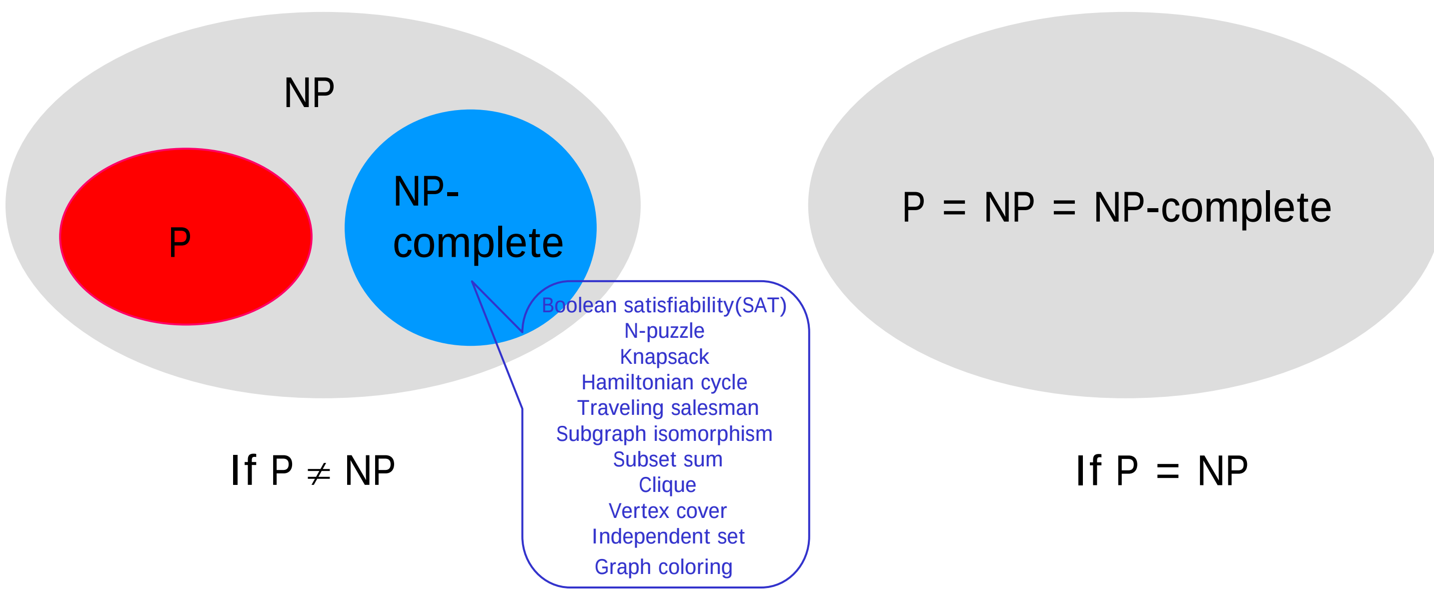
Looking Beyond

Deep Questions in Computer Science

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

The $1M Question: Does P = NP?

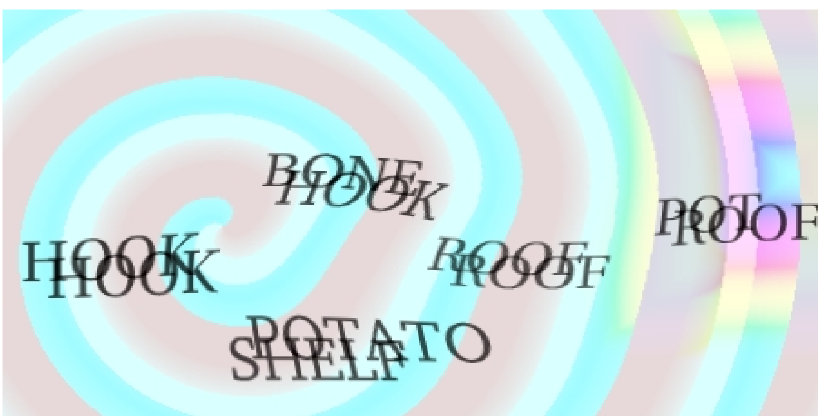
- The most important open problem in theoretical computer science. The Clay Institute of mathematics offers **one million** dollar prize for solution!
  - http://www.claymath.org/Millennium_Prize_Problems/



What is Computable?

- What is are the power and limits of computation?

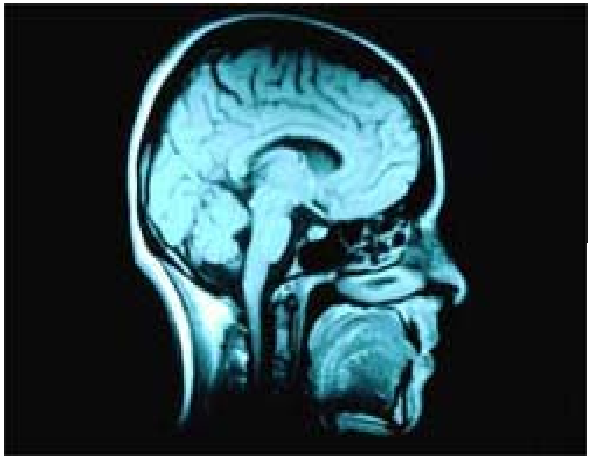
- What is computable when one considers The Computer as the combination of Human and Machine?



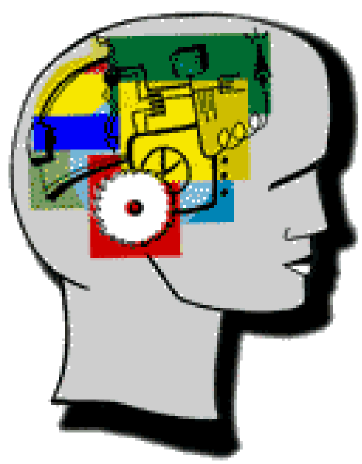
CAPTCHAs

Labeling Images on the Web

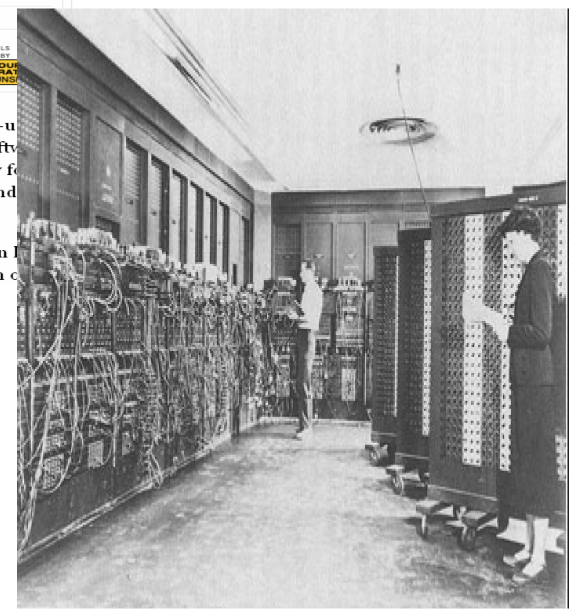
What is Intelligence?



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

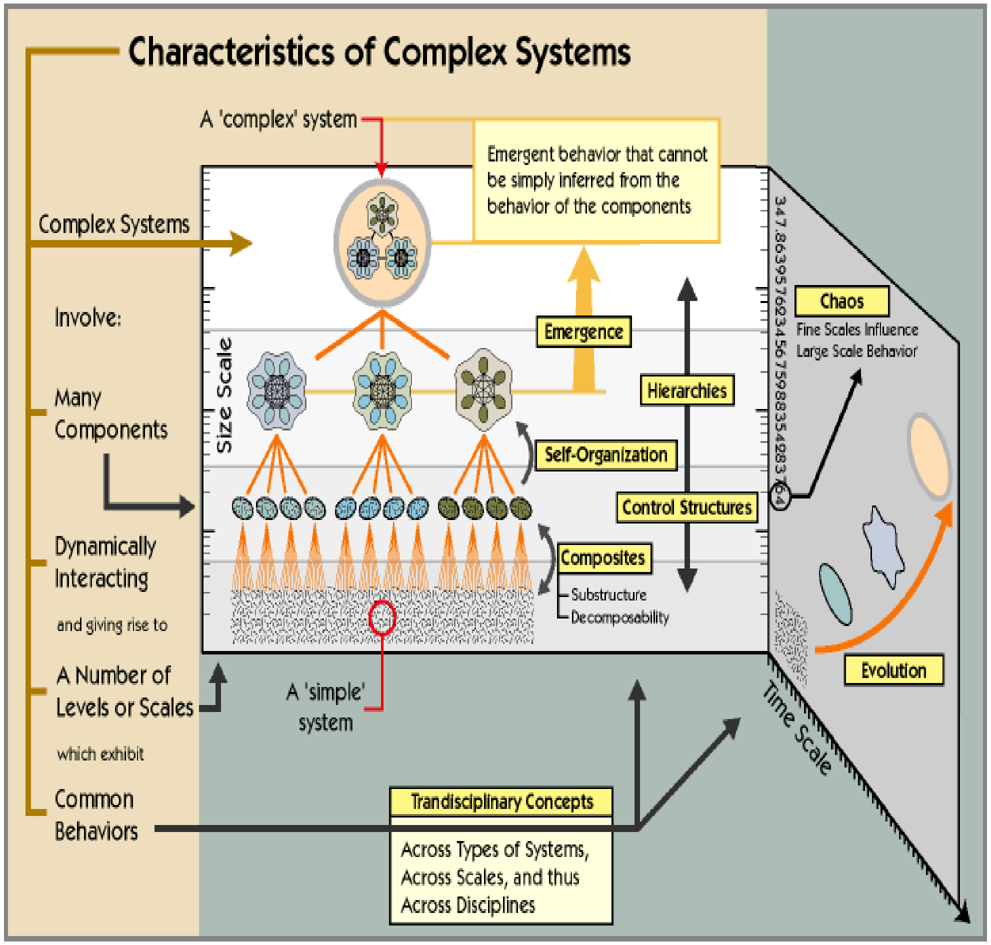
What is System Complexity?

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 21^{st} Century.
- Join us at Carnegie Mellon and the entire computing community toward making computational thinking commonplace.
  - http://www.cs.cmu.edu/computational_thinking.html

Spread the word!

To your fellow faculty, students, researchers, administrators, teachers, parents, principals, guidance counselors, government officials, policy makers, ...