Computational Thinking

Jeannette M. Wing
President’s Professor of Computer Science and Department Head
Computer Science Department
Carnegie Mellon University

OurCS Workshop
Carnegie Mellon University
4 March 2011
My Grand Vision

- 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.
  - 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, ...

Computing is the **Automation of Abstractions**

**Abstractions**

**Automation**

**Computational Thinking** focuses on the process of abstraction
- choosing the right abstractions
- operating in terms of multiple layers of abstraction simultaneously
- defining the relationships between layers

guided by the following concerns...

*as in Mathematics*
Measures of a “Good” Abstraction in C.T.

- Efficiency
  - How fast?
  - How much space?
  - How much power?

- Correctness
  - Does it do the right thing?
    - Does the program compute the right answer?
  - Does it do anything?
    - Does the program eventually produce an answer? [Halting Problem]

- -ilities
  - Simplicity and elegance
  - Usability
  - Modifiability
  - Maintainability
  - Cost
  - ...

as in Engineering

NEW
Computational Thinking, Philosophically

• 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
Sample Classes of Computational Abstractions

- Algorithms
  - E.g., mergesort, binary search, string matching, clustering
- Data Structures
  - E.g., sequences, tables, trees, graphs, networks
- State Machines
  - E.g., finite automata, Turing machines
- Languages
  - E.g., regular expressions, ..., VDM, Z, ..., ML, Haskell, ..., Java, Perl
- Logics and semantics
  - E.g., Hoare triples, temporal logic, modal logics, lambda calculus
- Heuristics
  - E.g., A* (best-first graph search), caching
- Control Structures
  - Parallel/sequential composition, iteration, recursion
- Communication
  - E.g., synchronous/asynchronous, broadcast/P2P, RPC, shared memory/message-passing
- Architectures
  - E.g., layered, hierarchical, pipeline, blackboard, feedback loop, client-server, parallel, distributed
- ...
Examples of Computational Thinking in Other Disciplines
One Discipline, Many Computational Methods
Computational Thinking in Biology

- Shotgun **algorithm** expedites sequencing of human genome
- DNA sequences are strings in a **language**
- **Boolean networks** approximate dynamics of biological networks
- Cells as a self-regulatory system are like **electronic circuits**
- **Process calculi** model interactions among molecules
- **Statecharts** used in developmental genetics
- Protein kinetics can be modeled as **computational processes**
- **Robot Adam** discovers role of 12 genes in yeast
- **PageRank algorithm** inspires ecological food web
Model Checking Primer

Finite State Machine model M

Temporal Logic property $\Phi$

$\Phi = \text{AG } p$

$\text{AF } p, \text{EG } p, \text{EF } p$

Model Checker

yes  counterexample

$\Phi$ is falsified here.
Model Checking in Biology

Goal: Predict Rate of Folding of Proteins

1. Finite State Machine $M$ represents 3-residue protein

<table>
<thead>
<tr>
<th>001 (2)</th>
<th>011 (-1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>000 (0)</td>
<td>101 (2)</td>
</tr>
<tr>
<td></td>
<td>110 (2)</td>
</tr>
</tbody>
</table>

1'. BDD efficiently represents $M$

2. Temporal Logic Formula $\Phi$

   a. Will the protein end up in a particular configuration?
   b. Will the second residue fold before the first one?
   c. Will the protein fold within $t$ ms?
   d. What is the probability that (c)?
   e. Does the state $s$ have $k$ folded residues and have energy $c$?

Method easily handles proteins up to 76 residues.

Model checking can explore state spaces as large as $2^{76} \approx 10^{23}$, 14 orders of magnitude greater than comparable techniques [LJ07].

Energy Profile for FKBP-12, Computed via Method
One Computational Method, Many Disciplines

Machine Learning has transformed the field of Statistics.
Machine Learning in the Sciences

- Brown dwarfs and fossil galaxies discovery via machine learning, data mining, data federation
- Very large multi-dimensional datasets analysis using KD-trees

**Astronomy**

**Medicine**
- Anti-inflammatory drugs
- Chronic hepatitis
- Mammograms
- Renal and respiratory failure

**Meteorology**
- Tornado formation

**Neurosciences**
- fMRI data analysis to understand language via machine learning

Jeannette M. Wing
Machine Learning Everywhere

Credit Cards

Supermarkets

Entertainment: Shopping, Music, Travel

Sports

Wall Street
NON-FACES

FACES
Question (Kearns): Can a Set of Weak Learners Create a Single Strong One?

Answer: Yes, by *Boosting* Algorithms (e.g., [FS99])
Computational Thinking in the Sciences and Beyond
CT in Other Sciences

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

Physics
- Adiabatic quantum computing: How quickly is convergence?
- Genetic algorithms discover laws of physics.

Geosciences
- Abstractions for Sky, Sea, Ice, Land, Life, People, etc.
  - Hierarchical, composable, modular, traceability, allowing multiple projections along any dimension, data element, or query
- Well-defined interfaces
CT in Math and Engineering

- 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

Mathematics

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
- Internet marketplace requires revisiting Nash equilibria model
- Use intractability for voting schemes to circumvent impossibility results

Law
- Inventions discovered through automated search are patentable
- 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

Healthcare
- Algorithmic medicine
- Software design principles and debugging applied to prescriptions of painkillers
- ONC SHARP Program, NSF Smart Health and Wellness Program, NITRD Senior Steering Group on Health IT
CT for Society

**Archaeology**
- eHeritage Project, Microsoft Research Asia
- Digital Forma Urbis Romae Project, Stanford
- Cathedral Saint Pierre, Columbia

**Journalism**
- Crowd sourcing as a new way of getting news tips from sources
- Algorithmic approach to validate credibility of sources
- Digital Media and Learning Initiative, MacArthur Foundation

**Humanities**
- Digging into Data Challenge: What could you do with a million books?
  Nat’l Endowment for the Humanities (US), JISC (UK), SSHRC (Canada)
- Music, English, Art, Design, Photography, ...
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
Education Implications for K-12

Question and Challenge for the Computing Community:

What is an effective way of learning (teaching) computational thinking by (to) K-12?

- What concepts can students (educators) best learn (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 teaching the concepts.

Computer scientists are now working with educators and cognitive learning scientists to address these questions.
Computational Thinking in Daily Life
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, ...).
Getting Morning Coffee at the Cafeteria

- coffee
- soda
- straws, stirrers, milk
- cups
- sugar, creamers
- napkins
- lids
Getting Morning Coffee at the Cafeteria

- coffee
- soda
- straws, stirrers, milk
- cups
- sugar, creamers
- napkins
- lids
Getting Morning Coffee at the Cafeteria

Especially Inefficient With Two or More Persons...
Better: Think Computationally—Pipelining!

coffee  
soda  
sugar, creamers  
napkins

straws, stirrers, milk  
cups  
lids
Computational Thinking at NSF
$48M FY08 to $100M in FY11 Budget Request

Computational Thinking for Science and Engineering

Cyber-Enabled Discovery and Innovation (CDI)

CONTACTS

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Phone</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eduardo Misawa</td>
<td><a href="mailto:cdi@nsf.gov">cdi@nsf.gov</a></td>
<td>(703) 292-8080</td>
<td></td>
</tr>
<tr>
<td>Thomas Russell</td>
<td><a href="mailto:cdi@nsf.gov">cdi@nsf.gov</a></td>
<td>(703) 292-8080</td>
<td></td>
</tr>
<tr>
<td>Kenneth Whang</td>
<td><a href="mailto:cdi@nsf.gov">cdi@nsf.gov</a></td>
<td>(703) 292-8080</td>
<td></td>
</tr>
</tbody>
</table>

Drs. Misawa, Russell, and Whang are being assisted by a multidisciplinary team of Program Officers drawn from throughout NSF. CDI team members include: Kiletwala, Andrew G. (SBE/OAD), William A. M. (SBE/OAD), Dan Lubin (OD/OPP), Manish Parashar (OD/OCI), David Rockcliffe (BIO/MCB), Nigel Sharp (MPS/AST), Carl Taylor (BIO/DBI), Rita Teutonico (SBE/OAD), Susan Winter (OD/OCI), William Wiseman (OD/OPP), and Eva Zanzerka (GEO/EAR).

PROGRAM GUIDELINES

Solicitation 10-506

SYNOPSIS

Cyber-Enabled Discovery and Innovation (CDI) is NSF’s bold five-year initiative to create revolutionary science and engineering research outcomes made possible by innovations and advances in computational thinking. Computational thinking is defined comprehensively to encompass computational concepts, methods, models, algorithms, and tools. Applied in challenging science and engineering research and education contexts, computational thinking promises profound impact on the nation’s competitiveness and scientific and engineering knowledge. Collectively,
Range of Disciplines in CDI Awards

- Aerospace engineering
- Astrophysics and cosmology
- Atmospheric sciences
- Biochemistry
- Biomaterials
- Biophysics
- Chemical engineering
- Civil engineering
- Communications science and engineering
- Computer science
- Cosmology
- Ecosystems
- Genomics
- Geosciences
- Linguistics
- Materials engineering
- Mathematics
- Mechanical engineering
- Molecular biology
- Nanocomputing
- Neuroscience
- Proteomics
- Robotics
- Social sciences
- Statistics
- Statistical physics
- Sustainability
- ...

... advances via Computational Thinking
“to develop competencies in computational thinking”
Computational Thinking in Education and Beyond
CMU and Other Colleges and Universities

• CMU: Redesign of Intro Courses
  – “15-110: Principles of Computer Science. An introduction to computer science, based on the principles of computational thinking. Many taking this course will be nonmajors, but we will also use it as the entry point for any entering student with limited programming experience.” [Bryant, Stehlik, Sutner, Introductory Computer Science Education at Carnegie Mellon University: A Deans’ Perspective, CMU-CS-10-140, August 2010]

C.T. in Education: National Efforts

Computational Thinking

Computing Community

CSTA
SIGCSE
CRA-E
ACM-Ed
National Academies
College Board
K-12
AP
NSF
CE21
Congress
Hill Event 2009
Computer Science Education Week
Computer Science Education Act 2010

Microsoft
Industry
Google
CS4HS
Exploring Computational Thinking Website
CMU and Other Colleges and Universities

• CMU: Redesign of Intro Courses
  – “15-110: Principles of Computer Science. An introduction to computer science, based on the principles of computational thinking. Many taking this course will be nonmajors, but we will also use it as the entry point for any entering student with limited programming experience.” [Bryant, Stehlik, Sutner, Introductory Computer Science Education at Carnegie Mellon University: A Deans’ Perspective, CMU-CS-10-140, August 2010]

The Computer Science and Informatics panel said “Computational thinking is influencing all disciplines....”

par Jeannette M. Wing
Academic Organizations

• College Board [http://csprinciples.org](http://csprinciples.org)
  – With NSF support, revision of CS AP courses
  – Five universities pilots this year: UNC-Charlotte, UC Berkeley, Metropolitan Sate College of Denver, UC San Diego, and University of Washington
  – More schools—high schools, community colleges, and universities—to participate next year.

• National Academies Computer Science and Telecommunications Board
Computer Science Organizations

- SIGCSE sessions
- ACM Educational Council discussions
- CSTA: http://www.csta.acm.org/
  - *Computational Thinking Resource Set: A Problem-Solving Tool for Every Classroom*
- CRA-E
  - Includes recommendations for computational thinking courses for non-majors.
Congress

- Event on the Hill: May 29, 2009
  - Sponsored by ACM, CRA, CSTA, IEEE, Microsoft, NCWIT, NSF, and SWE, called for putting the “C” (computer science) into “STEM.”

- Computer Science Education Week:
  http://www.csedweek.org/
  - Sponsored by ABI, ACM, BHEF, CRA, CSTA, Dot Diva, Google, Globaloria, Intel, Microsoft, NCWIT, NSF, SAS, and Upsilon Pi Epsilon.

- Computer Science Education Act
  - On July 30, 2010 Rep. Jared Polis (D-CO) introduced the Computer Science Education Act (H.R.5929) to strengthen K-12 computer science education.
Industry Support

• CS4HS
  – Initiated in 2006 at CMU, with support from Google and later Microsoft,
  – 2007, spread to UCLA and UW
  – By 2010, under the auspices of Google, CS4HS spread to 20 schools in the US and 14 in Europe, the Middle East, and Africa.

• Microsoft Research-Carnegie Mellon Center for Computational Thinking: http://www.cs.cmu.edu/~CompThink/
  – Since 2007, the Center supports both research and educational outreach projects.

  – Wealth of links to further web resources, including lesson plans for K-12 teachers in science and mathematics.
Spread the Word

• Help make computational thinking commonplace!

To fellow faculty, students, researchers, administrators, teachers, parents, principals, guidance counselors, school boards, teachers’ unions, congressmen, policy makers, ...
Thank you!
References (Representative Only)

• Computational Thinking

• Model Checking, Temporal Logic, Binary Decisions Diagrams

• Computational Thinking and Biology
  – Executable Cell Biology, Jasmin Fisher and Thomas A Henzinger, Nature Biotechnology, Vol. 25, No. 11, November 2007. (See paper for many other excellent references.)
References (Representative Only)

- **Machine Learning and Applications**
  - Symbolic Aggregate Approximation, Eamonn Keogh, UC Riverside, [http://www.cs.ucr.edu/~eamonn/SAX.htm](http://www.cs.ucr.edu/~eamonn/SAX.htm) (applications in Medical, Meteorological and many other domains)
  - The Auton Lab, Artur Dubrawski, Jeff Schneider, Andrew Moore, Carnegie Mellon, [http://www.autonlab.org/autonweb/2.html](http://www.autonlab.org/autonweb/2.html) (applications in Astronomy, Finance, Forensics, Medical and many other domains)

- **Computational Thinking and Astronomy**
  - Sloan Digital Sky Survey @Johns Hopkins University, [http://www.sdss.jhu.edu/](http://www.sdss.jhu.edu/)

- **Computational Thinking and Archaeology**
  - See also Marc Levoy’s digital archaeology projects: [http://www-graphics.stanford.edu/~levoy/](http://www-graphics.stanford.edu/~levoy/)
  - See also UK universities: [http://en.wikipedia.org/wiki/Computational_archaeology#Research_groups_and_institutions](http://en.wikipedia.org/wiki/Computational_archaeology#Research_groups_and_institutions)

- **Computational Thinking and Chemistry**

- **Computational Thinking and Economics**
References (Representative Only)

- Michael Kearns, Computational Game Theory, Economics, and Multi-Agent Systems, University of Pennsylvania, [http://www.cis.upenn.edu/~mkearns/#gamepapers](http://www.cis.upenn.edu/~mkearns/#gamepapers)

• Computational Thinking and Journalism

• Computational Thinking and Law
- Burkhard Schafer, Computational Legal Theory, [http://www.law.ed.ac.uk/staff/burkhardschafer_69.aspx](http://www.law.ed.ac.uk/staff/burkhardschafer_69.aspx)

• Computational Thinking and Medicine and Healthcare
- Institute for Computational Medicine, Johns Hopkins University, [http://www.icm.jhu.edu/](http://www.icm.jhu.edu/)
- See also Symbolic Aggregate Approximation, Eamonn Keogh, UC Riverside, [http://www.cs.ucr.edu/~eamonn/SAX.htm](http://www.cs.ucr.edu/~eamonn/SAX.htm)

• Computational Thinking and Meteorology
- See also Symbolic Aggregate Approximation, Eamonn Keogh, UC Riverside, [http://www.cs.ucr.edu/~eamonn/SAX.htm](http://www.cs.ucr.edu/~eamonn/SAX.htm)
References (Representative Only)

• Computational Thinking (especially Machine Learning) and Neuroscience

• Computational Thinking and Sports
  – Lance Armstrong’s cycling computer tracks man and machine statistics, website
Credits

• Copyrighted material used under Fair Use. If you are the copyright holder and believe your material has been used unfairly, or if you have any suggestions, feedback, or support, please contact: jssoleil@nsf.gov

• Except where otherwise indicated, permission is granted to copy, distribute, and/or modify all images in this document under the terms of the GNU Free Documentation license, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation license” (http://commons.wikimedia.org/wiki/Commons:GNU_Free_Documentation_License)
Computational Thinking, in Summary

- Computational thinking is the *thought processes* involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information processing agent. [Cuny, Snyder, Wing 2010, in progress]