The Quintessential Questions of Computer Science

November 14, 2007
Motivation for this Talk

• Inform people, especially students, of the intriguing open scientific questions lying at the heart of the field of computer science
Bill Gates Roundtable with CS Faculty at Columbia University 10/13/05
Forty Years of Computer Science

• What is the biggest impact that computer science has had on the world in the past forty years?

• Typical answer: the Internet with its associated global information infrastructure and applications
Forty Years of Programming Languages: The 10 most popular programming languages in 1967

• Algol 60
• APL
• Basic
• BCPL
• COBOL
• Fortran IV
• Lisp 1.5
• PL/I
• Simula 67
• SNOBOL 4
The 10 most popular programming languages in 2007

- Java
- C
- Visual Basic
- C++
- PHP
- Perl
- C#
- Python
- JavaScript
- Ruby
Question 1

• How do we determine the difficulty of a problem?
The Classes $P$ and $NP$

• A problem is in $P$ if it can be solved in polynomial time by a deterministic Turing machine.

  Example: Does a set of $n$ positive and negative integers have a nonempty subset whose sum is positive?

  \[
  \{ -2, 7, -3, 14, -10, 15 \}
  \]

• A problem is in $NP$ if it can be solved in polynomial time by a nondeterministic Turing machine.

  Example: Does a set of $n$ positive and negative integers have a nonempty subset whose sum is zero?

  \[
  \{ -2, 7, -3, 14, -10, 15 \}
  \]
The *P* vs. *NP* Problem

• Does *P* = *NP*?

• Informally: Are there any problems for which a computer can verify a given solution quickly but cannot find the solution quickly?

• Note: This is one of the Clay Mathematics Institute Millennium Prize Problems. The first person solving this problem will be awarded one million US dollars by the CMI (http://www.claymath.org/millennium).

Stephen Cook
The Complexity of Theorem Proving Procedures

L. A. Levin
Universal Sorting Problems
Another Interesting Problem: Integer Factorization

• Problem: Given an $n$-bit integer, find all of its prime factors.

• Best-known deterministic algorithm has time complexity $\exp(O(n^{1/3} \log^{2/3} n))$.

• Open Problem: Can this problem be done in deterministic polynomial time?
Question 2

• How do we model the behavior of complex systems that we would like to simulate?

Large software systems

Human cell
Ion Trap Quantum Computer
Problem: Given a composite $n$-bit integer, find a nontrivial factor.

A quantum computer can solve this problem in $O(n^3)$ operations.

Open Problem: Can this problem be solved in deterministic polynomial time on a classical computer?
Programming Languages and Compilers for Quantum Computers

1. Quantum Computer
2. Mathematical Model
3. Computational Formulation
4. Compiler
Quantum Computer Compiler

Mathematical Model: Quantum mechanics, unitary operators, tensor products

Computational Formulation: Quantum bits, gates, and circuits

QCC: QIR, QASM

Target QPOL

Physical System: Laser pulses applied to ions in traps

EPR Pair Creation

Quantum Circuit Model

QIR → QASM

QPOL

Machine Instructions → Physical Device

\[ |0,y\rangle + \frac{(-1)^y|1,y\rangle}{\sqrt{2}}, \text{ for } x, y \in \{0, 1\} \]

\[ |x\rangle \xrightarrow{H} \]

\[ |y\rangle \]

K. Svore, A. Aho, A. Cross, I. Chuang, I. Markov

A Layered Software Architecture for Quantum Computing Design Tools

Design Flow with Fault Tolerance and Error Correction

Mathematical Model: Quantum mechanics, unitary operators, tensor products

Computational Formulation: Quantum bits, gates, and circuits

QCC: QIR, QASM

Software: QPOL

Physical System: Laser pulses applied to ions in traps

Fault Tolerance and Error Correction (QEC)

| a ⟩
| b ⟩

QEC

Moves

Moves
Question 3

• How do we build a scalable trustworthy information infrastructure?
Demand for Trustworthy Systems

• 36 million Americans have had their identities stolen since 2003

• 155 million personal records have been compromised since 2005

• 28 million veterans had their Social Security numbers stolen from laptops
Demand for Trustworthy Systems
Protection from Malware

• Internet malware
  – worms, viruses, spyware and Internet-cracking tools
  – worms override program control to execute malcode

• Internet worms
  – Morris '88, Code Red II '01, Nimda '01, Slapper '02, Blaster '03, MS-SQL Slammer '03, Sasser '04
  – automatic propagation

• Internet crackers
  – “j00 got h4x0r3d!!”

• After breaking in, malware will
  – create backdoors, install root kits (conceal malcode existence), join a botnet, generate spam

---

Gaurav S. Kc

Defending Software Against Process-Subversion Attacks
PhD Dissertation, Columbia University, 2005
Question 4

• Is there a scientific basis for making reliable software?
How Can We Make Reliable Software?

- **Communication**: Shannon [1948] used error detecting and correcting codes for reliable communication over noisy channels

- **Hardware**: von Neumann [1956] used redundancy to create reliable systems from unreliable components

- **Software**: Is there a scientific basis for making reliable software?
Volume of Software and Defects

• World uses hundreds of billions of lines of software
  – 5 million programmers worldwide
  – average programmer generates 5,000 new lines of code annually
  – embedded base: hundreds of billions of lines of software

• Number of embedded defects
  – defect densities: 10 to 10,000 defects/million lines of code
  – total number of defects in embedded base: $5 \times 10^6$ to $50 \times 10^9$

Alfred V. Aho

Software and the Future of Programming Languages

*Science*, February 27, 2004, pp. 1331-1333
# IEEE Spectrum Software Hall of Shame

<table>
<thead>
<tr>
<th>Year</th>
<th>Company</th>
<th>Costs in US $</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>UK Inland Revenue</td>
<td>Software errors contribute to $3.45 billion tax-credit overpayment</td>
</tr>
<tr>
<td>2004</td>
<td>J Sainsbury PLC [UK]</td>
<td>Supply chain management system abandoned after deployment costing $527M</td>
</tr>
<tr>
<td>2002</td>
<td>CIGNA Corp</td>
<td>Problems with CRM system contribute to $445M loss</td>
</tr>
<tr>
<td>1997</td>
<td>U. S. Internal Revenue Service</td>
<td>Tax modernization effort cancelled after $4 billion is spent</td>
</tr>
<tr>
<td>1994</td>
<td>U. S. Federal Aviation Administration</td>
<td>Advanced Automation System canceled after $2.6 billion is spent</td>
</tr>
</tbody>
</table>

The Software Development Process

• Specification
  – Define system functionality and constraints

• Validation
  – Ensure specification meets customer needs
  – “Are we building the right product?”

• Development
  – Produce software

• Verification and testing
  – Ensure the software does what the specification calls for
  – “Are we building the product right?”

• Maintenance
  – Evolve the software to meet changing customer needs

• Quality plan
  – Ensure product meets user needs
Where is the Time Spent?

• 1/3 planning
• 1/6 coding
• 1/4 component test and early system test
• 1/4 system test, all components in hand

“In examining conventionally scheduled projects, I have found that few allowed one-half of the projected schedule for testing, but that most did indeed spend half of the actual schedule for that purpose.”

Why Do Software Projects Fail?

• Unrealistic or unarticulated project goals
• Inaccurate estimates of needed resources
• Badly defined system requirements
• Poor reporting of the project’s status
• Unmanaged risks
• Poor communication among customers, developers, and users
• Use of immature technology
• Inability to handle the project’s complexity
• Sloppy development practices
• Poor project management
• Stakeholder politics
• Commercial pressures

Ingredients for Making Reliable Software

• Good people/management/communication

• Good requirements/modeling/prototyping

• Sound software engineering practices

• Use of mature technology

• Thorough testing

• Verification tools
  – model checkers
  – theorem-proving static analyzers
catch

**Verification Status**

**Call Waiting**

Bellcore reference: cw.pdf
Database: ref.,spec.

Description:

When the subscriber is busy, an incoming call triggers a cw tone at the subscriber and a normal ringtone at the caller. The cw tone can be repeated at most once, after a 10 sec interval (9..30sec) (repeat not implemented in the PathStar code).

The incoming call can be 'acknowledged' by the subscriber with a flash. After the flash, the incoming call is either on hold or connected.

If the subscriber goes onhook without having acknowledged the waiting call -- the subscriber receives ringtone, until pickup, or onhook of the waiting call.

[1] A flash with a held party and no connected party should restore a connection with the held party.
Modeling feature behavior

Every path through feature graph defines a system requirement and hence a check to be made.
Modeling Requirements with Linear Temporal Logic

Example:
“When the subscriber goes offhook, dialtone is generated.”

A failure to satisfy the property:

\[ <> \text{ eventually,} \]

\[ \land \text{ and} \]

\[ X \text{ thereafter, no dialtone is} \]

\[ U \text{ generated until the next onhook} \]

LTL formula:

\[ <> \left( \text{offhook} \land X \left( \neg \text{dialtone} U \text{ onhook} \right) \right) \]
FeaVer Verification Process

- Source code
- Abstractions
- Properties

- Convert formulae to test automata
- Extract model from source code
- Generate error traces in source language
- Compile executable model checker -- one per property
- Schedule and execute checks using model checking engine spin

Gerard Holzmann
FeaVer Software Verification System
But the open problem remains

Is there a scientific basis for making reliable software?
Can we construct computer systems that have human-like attributes such as emotion or intelligence?

Cogito, ergo sum.
An Easier Question, Perhaps

• Can a deterministic program generate random output?

• BBP algorithm can compute the $n$ th bit of pi without having to compute the first $n – 1$ bits.

• But it is not known whether the digits of pi are random.

David H. Bailey, Peter B. Borwein and Simon Plouffe
On the Rapid Computation of Various Polylogarithmic Constants
Marriage with Robots?

“My forecast is that around 2050, the state of Massachusetts will be the first jurisdiction to legalize marriages with robots.”

David Levy
AI researcher
University of Maastricht, Netherlands
LiveScience, October 12, 2007
Bill Gates

Moore’s Law for number of transistors on a chip
Bill Gates

Moore’s Law for power consumption

![Graph showing Moore's Law for power consumption](source: Intel)
Bill Gates’s Question

• How do we extend Moore’s Law?

• Are multicore architectures the answer?
Summary

1. How do we determine the difficulty of a problem?

2. How do we model the behavior of complex systems that we would like to simulate?

3. How do we build a trustworthy information infrastructure?

4. Is there a scientific basis for making reliable software?

5. Can we construct computer systems that have human-like attributes such as emotion or intelligence?

6. How do we extend Moore’s Law?