

Creating Knowledge in the Age of Digital Information

Robert L. Constable
Dean of the Faculty of
Computing & Information Science
Cornell University



June 16, 2009

Computer Science Department

Ben Guion University

Introduction – The Key Idea

Why is **computing and information science** relevant to nearly every academic discipline?

Why is it key to solving the world's hardest technical and social problems?

This talk will answer these questions, taking an historical approach, starting with **computer science**, the core subject of the computing and information sciences.

The Evolving Character of Computer Science

Computer Science is a field created by **impatient immigrants**. They came first from logic, then mathematics, electrical engineering, linguistics, psychology, economics, neuroscience, and other fields.

They formed CS departments in the US circa 1965.

The Evolving Character of Computer Science

The immigrants asked: **How can we compute faster, better?**

- with **numbers**, strings, formulas, sentences, pictures, movies, **models** of neurons, of the atmosphere, of materials, of the world, etc.

- whether using (five) mainframe computers or a billion **cell phones**.

CS Reacts to Established Sciences

Physics: discovery of the fundamental laws of Nature.

CS: discovery of the fundamental laws of computing.

Fundamental Laws of Computing

Alan Turing in 1936 gave us some fundamental theory: universal machines, unsolvable problems, like the Halting Problem, computable real numbers, and there was much more to come from him during WWII, at Manchester building computers, and stimulating AI in the 1950's.

More Fundamental Theory

Russell and Church gave us type theory, and typed programming languages; the HOL type theory, widely used today in formal methods, is a direct descendent.

Church gave us Church's Thesis (also called the Church/Turing Thesis with variants for total functions, partial functions, and oracular machines).

Nachum Dershowitz will discuss proving this thesis in his talk.

More Fundamental Theory

Hartmanis, Stearns, Rabin: Computational complexity of algorithms.

First reaction to this from mathematicians: asymptotic complexity is the wrong idea, we'll straighten you all out when we have time.

History: Led to the $P=NP$ problem, one of the seven Millennium problems.

More Fundamental Theory

McCarthy, Hoare, Scott gave us Programming Logics in the 70's which are still being used and extended today.

The idea of Mathematics as a Programming Language has a long and diverse history involving mathematicians, logicians, and computer scientists, with slogans such as Propositions-as-types and Proofs-as-programs.

CS Reacts to Other Sciences

Biology: study of all “creatures great and small”.

CS: study of all automata finite and infinite.

Biology: study of living systems, energy, reproduction, information processing.

CS: study of computing systems, communications, fault tolerance, adaptation, correctness, evolution, etc.

The Audacities

Physics: we'll make energy as
does the sun!

Biology: we'll make life!

CS: we'll make thinking machines!

The Realities

Physics: how about lasers and the Big Bang.

Biology: how about DNA and comparative genomics.

CS: how about the Internet/Web and Crypto.

CS Reacts to Information Technology

With the rise of “billion dollar industries” around PC’s, operating systems, databases, graphics, compilers, pda’s, the Internet, the Web, search, and so forth, we get this definition of computer science.

CS is the science base for IT.

Plan for the Remainder of the Talk

1. Reveal the **key driving idea** for near universal relevance by examples
2. Briefly examine **impact on university structures**
3. Historical Perspective and **Conclusion**

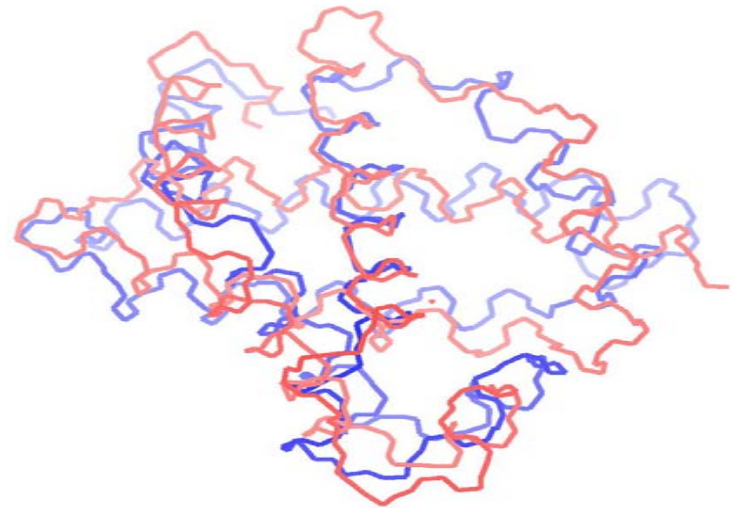
Relevance of CIS ideas and methods by examples

- Computational Biology (Life Sciences)
- Astronomy (Physical Sciences)
- Social Networking (Social Sciences)
- Computational Archeology (Humanities)
- Digital Age Mathematics
- CS- Formal Methods (Information Sciences)

Bioinformatics – Geometric Structure

Ron Elber notes high degree of structural similarity is often observed in proteins with diverse sequences and in different species (below noise level – 15 percent sequence identity).

Oxygen Transport Proteins



Leghemoglobin in Plants

Yet Bigger Tomatoes



Elber/Tanksley Discovery

Chromosome 2



fw 2.1, 2.2, 2.3

TG608
TG189

CT205

TG554

TG493
TG266
TG469

TG463
CT9

TG337
TG48

TG34
TG91
TG167

TG151

TG59

TG154



stuffer



ovate



Se 2.1

Elber/Tanksley Discovery - continued



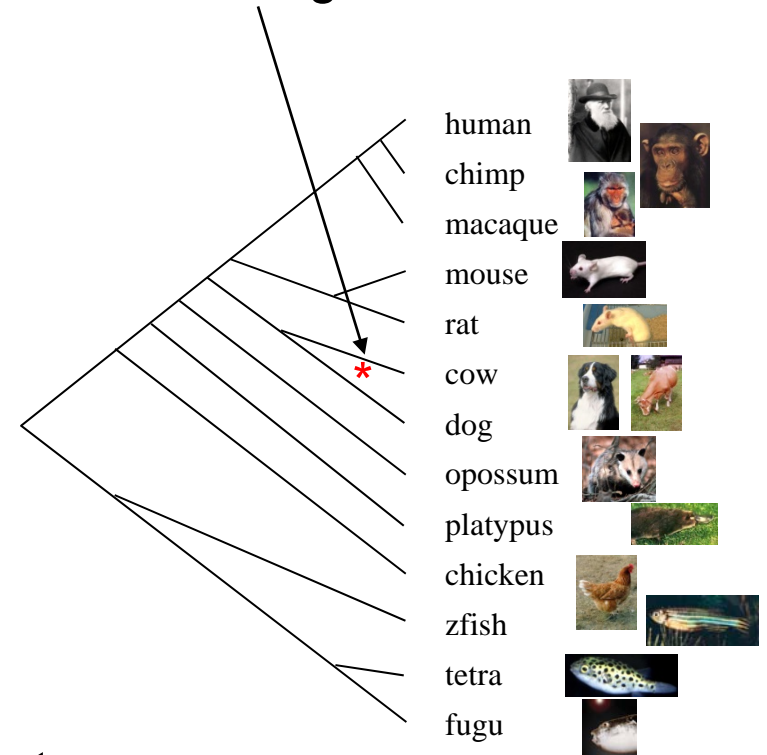
Human Ras p21

- ♦ Molecular switch based on GTP hydrolysis
- ♦ Cellular growth control and cancer
- ♦ Ras oncogene: single point mutations at positions Gly12 or Gly61

Comparative Genomics from Gill Bejerano



How did some of our relatives go back?

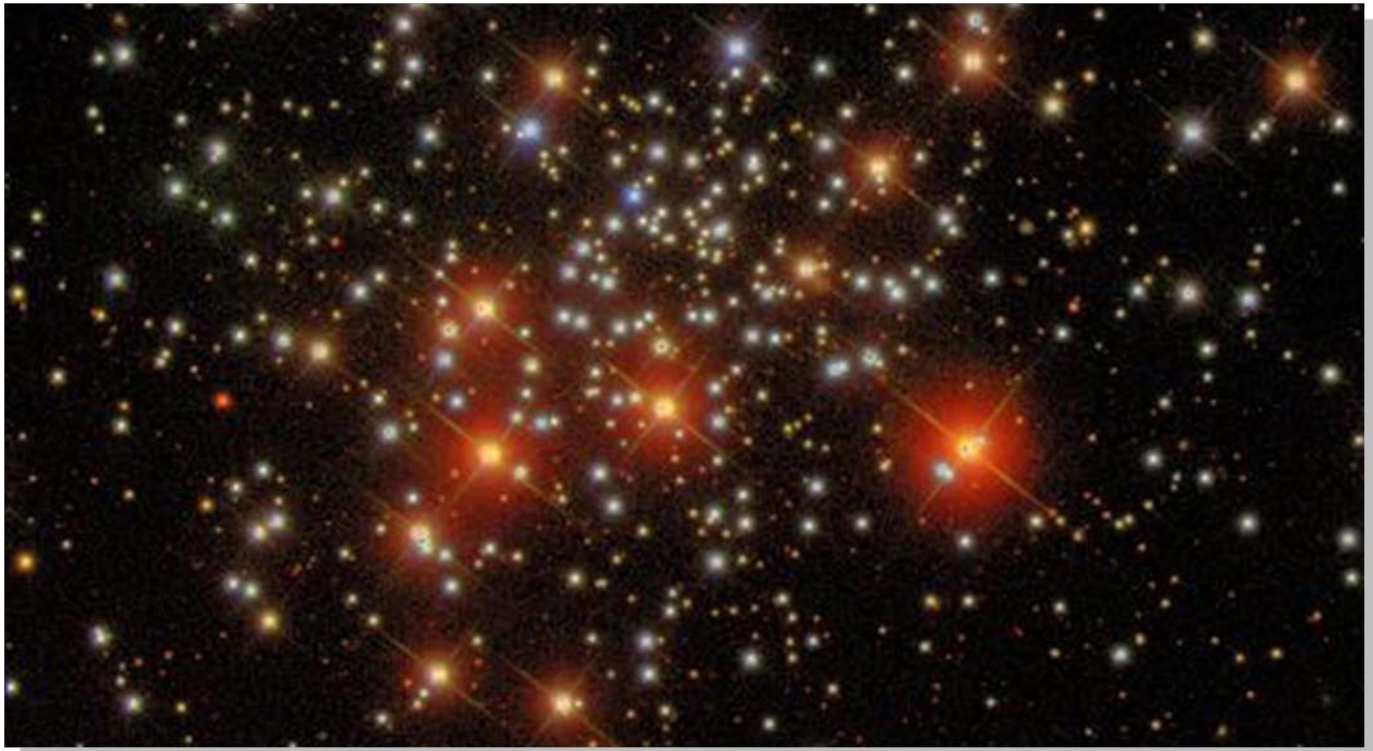


What we now understand

1. Ultraconserved elements exist.
2. They are maintained via strong on-going selection.
3. It is a heterogeneous bunch:
4. Some mediate splicing
5. Some regulate gene expression
6. Some express ncRNAs

National Virtual Observatory

The PC is a telescope for viewing “digital stars”.

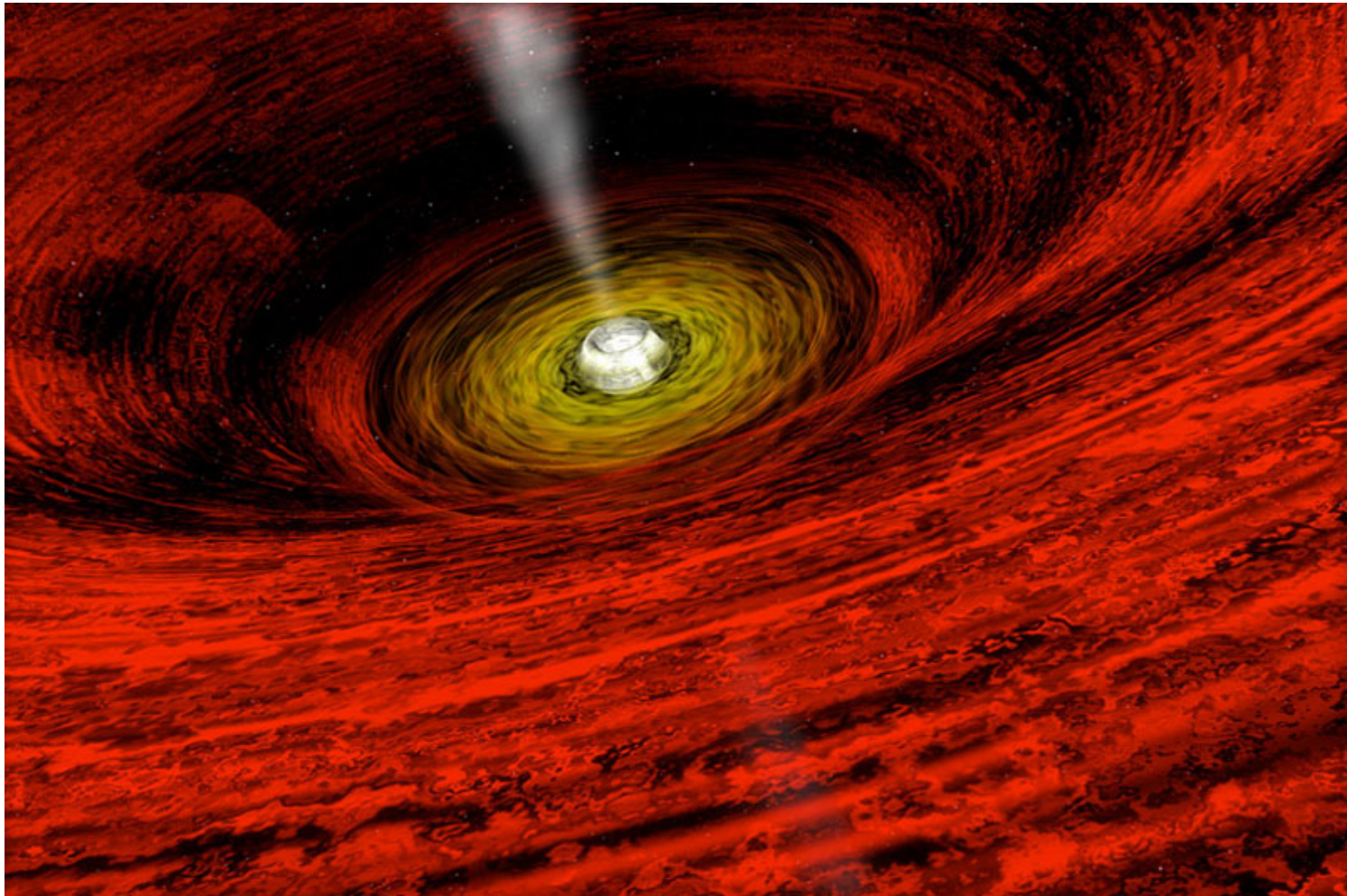


Changing the face of astronomy

The astronomer **Alexander Szalay** has said the work of computer scientists on the NVO has “changed astronomy as we know it”.

Machine learning applied to large databases has led to new discoveries, e.g. new exotic sources, identification of unidentified sources.

And we can even extrapolate to more complex exotic systems



Other Examples

- Social Sciences

There are laws of social networks, e.g.,
six degrees of separation

- Humanities

Assembling the [map of the city of Rome](#), circa 210 A.D.



Eli Shamir's work is clearly relevant in the humanities.

Consider these famous problems

- The Poincare Conjecture
- The Four Color Theorem

Computers and the Web have fundamentally changed how they were definitively solved.

On November 11, 2002 Perelman posted a proof of the Poincaré Conjecture on the **Cornell arXiv**, Paul Ginsparg's **digital library** of “e-prints.” This posting stimulated the math community to “fill in the details.”

(Paul Ginsparg is an **Information Science** professor in CIS, and Perelman's proof builds on the work of William Thurston, a Field's Medalist who has a joint CIS appointment with Math.)

Digital Age Mathematics – The Poincaré Conjecture

In 2006 the International Mathematical Union (IMU) tried to award **Gregory Perelman** of St. Petersburg its highest honor, the Fields Medal, for solving the **Poincaré Conjecture**, one of the seven Millenium problems. He would not accept.

“If the proof is correct, no other recognition is needed.”

The Poincaré Conjecture - Controversy

Fields Medalist Shing-Tung Yau said in 2006,

“In Perelman’s work, spectacular as it is, many key ideas of the proofs are sketched or outlined, and complete details are often missing.”

The idea of a proof is central to modern mathematics. They have strict forms, like a sonnet. It can now be measured against a new standard, the **complete formal proof** – an idea from Hilbert made precise and implemented by computer scientists.

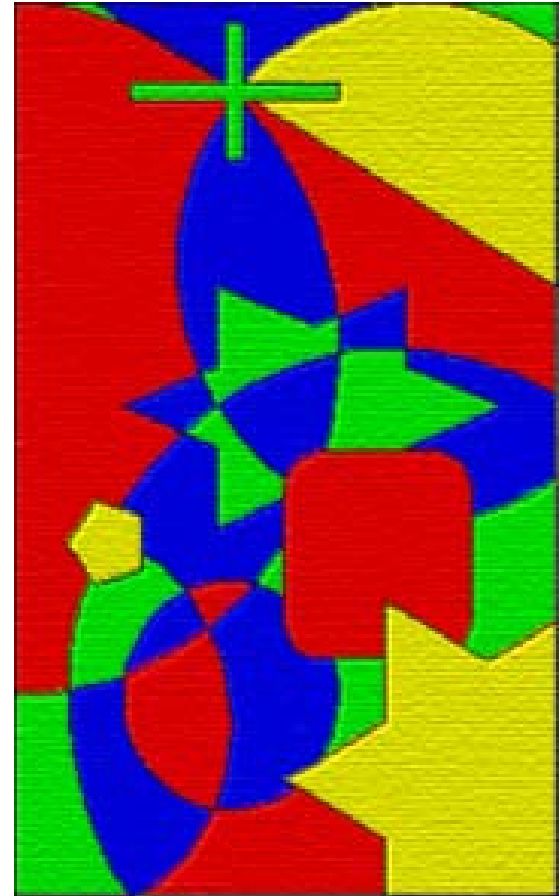
Digital Age Mathematics

One of the most profound contributions of computer science to intellectual history is the demonstration that **computers can implement many high level mental functions.**

(The converse is also profound, the discovery that our **mundane mental functions are extremely difficult to automate**, such as recognizing shape similarity.)

The Four Color Theorem 1976

In 1976 computers helped Appel and Haken prove the 1852 four color conjecture – that any planar map can be colored using four colors so that no two adjacent regions have the same color.



Concerns about the Appel/Haken Proof

The programs used to show that the 1,476 reducible maps could be four colored **were not proved to be correct**, and ran for hundreds of hours.

Formal Proof of the Four Color Theorem

In 2004, **Georges Gonthier** at MSR used the Coq theorem prover, with help from Benjamin Werner, to give a definitive computer checked proof of **the four color theorem**.

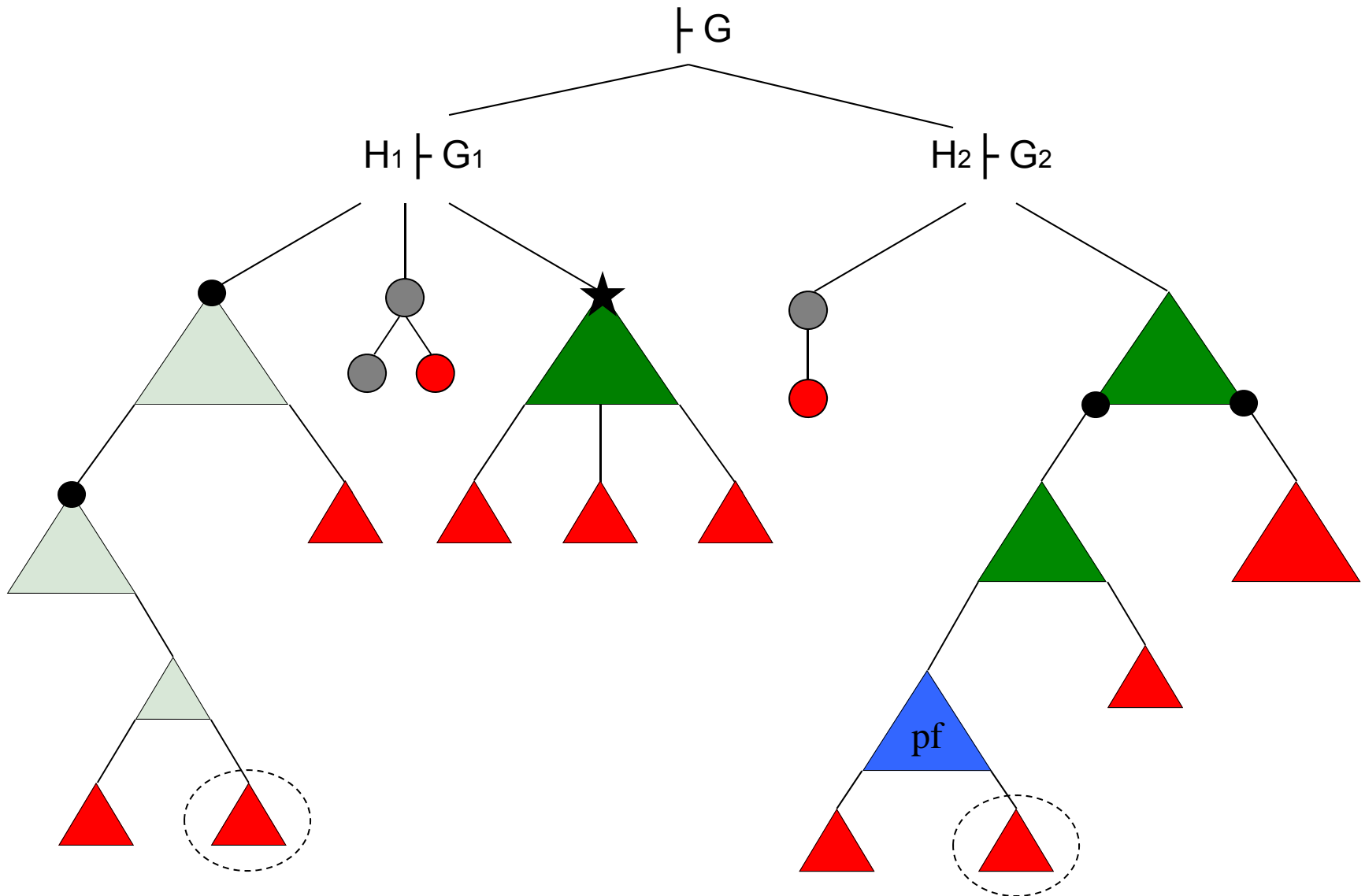
The Nature of Formal Proofs

Formal proofs are elements of a tree-like data structure whose nodes are called **sequents**. They have the form

$$H_1, \dots, H_n \vdash G$$

Where the H_i are propositions called the **hypotheses** and G is the **goal**.

A Picture of Proof Structure



Analyzing Proof Structure

key insights ●

clever step ★

filled in by machine

humans ignore ▲

humans need ○

experts need ★

routine ▲

learners need ○

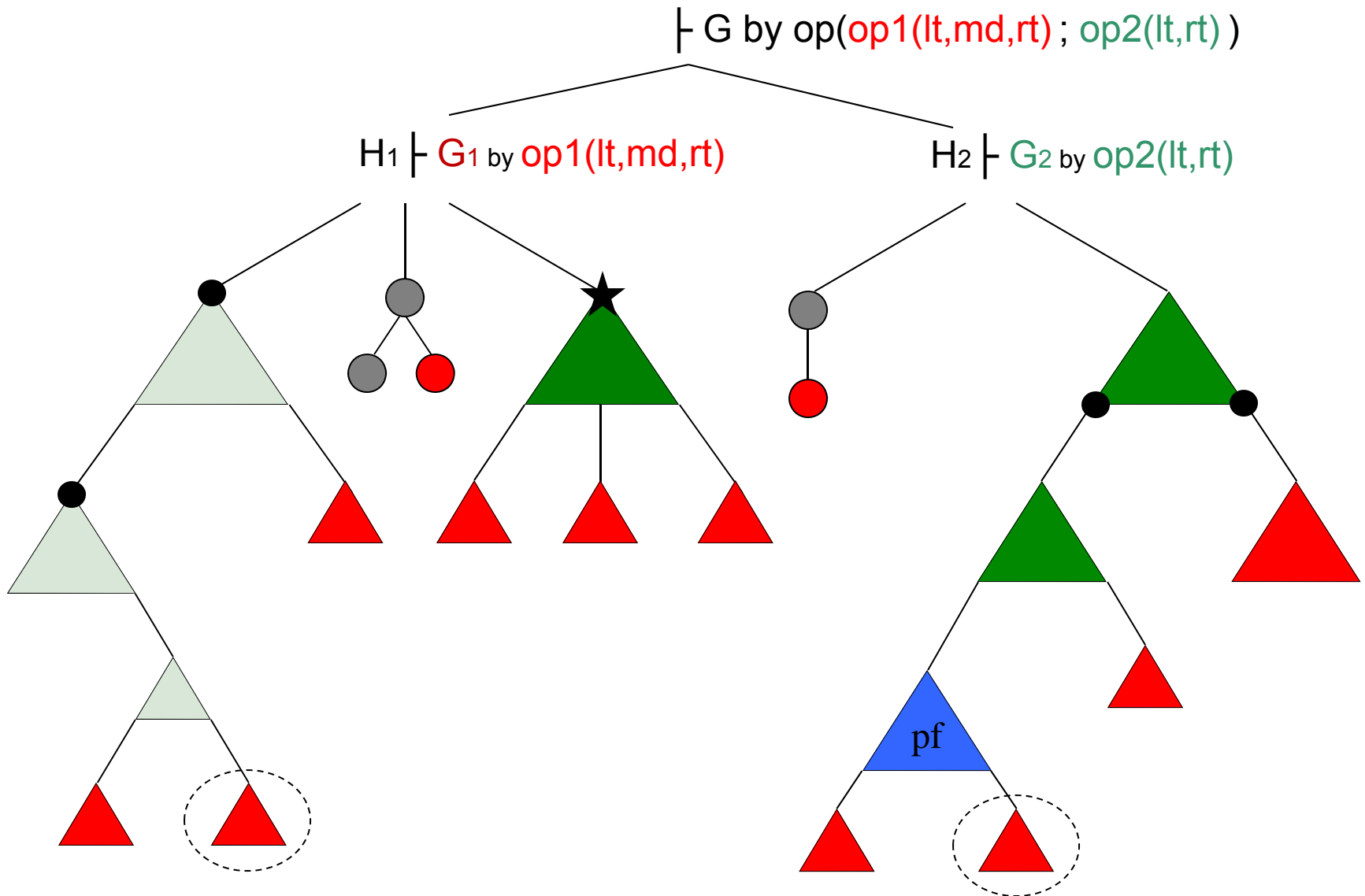
obvious ▲

trivial ▲

well known ●

minor variant of pf ▲

A Picture of Proof Structure with Extracts



The Extract is a Graph Coloring Algorithm

The widely used Coq theorem prover is constructive, and it is possible to read the theorem as saying

Given any planar graph, we can find a coloring of it using at most four colors.

Gonthier chose not to make the proof fully constructive because he wanted to remain faithful to the original. Had he made it constructive, the extract would be coloring algorithm.

Formal Methods

- Some code extracted from proofs is very useful, compare graph coloring to distributed computing, say **n-coloring** to **authentication** and **consensus**, or to a **verified compiler**, or
- Compare the Fundamental Theorem of Algebra to the verification of smart-card domain separation.
- At Intel hundreds of engineers use formal methods every day, and many programmers at Microsoft do as well.

Formal Methods in Systems

Distributed computing protocols are especially hard to “get right”, this difficulty has cost companies and governments billions of dollars.

My colleague Mark Bickford and I use the Nuprl system to extract protocols that are **correct by construction**. For example, we produce **consensus protocols** this way – from Nuprl to Java with a verified compiler.

We aspire to be able to handle protocol stacks of the kind Danny Dolev will discuss, and our methods are much like those of Amir Pnueli.

Formal Methods in Systems

Specifying distributed computing tasks is not like specifying traditional mathematical problems. Specifying **fault tolerant consensus protocols** is especially tricky because there is a result (even a constructive result) saying that such consensus can't be achieved!

But the Google file system and the Chubby lock service depend on the Paxos consensus protocol working properly. What to do?

Jeff Ullman's method probably depends on Paxos.

Uses of Formal Methods and Applied Logic

- Formal methods arose in the quest for **certain knowledge**.
- They are useful in checking for errors (**model checking**).
- They have been used to refine conjectures and solve **open problems** in mathematics.
- They are used to design and **verify distributed protocols**.
- They precisely **connect layers of abstraction** across many orders of magnitude, allowing us to control massive complexity.

The Question

Why is **computing and information science** relevant to nearly every academic discipline?

Why is it key to solving the world's hardest technical and social problems?

Lessons from Examples

- In all of these examples, computers were acting as essential partners, accelerating the work, performing tasks that **no amount of human effort could duplicate – not all people on the planet working together.**
- Many of the key intellectual tasks were automated; the scientists decomposed and assigned the tasks. **They had to understand computational thinking to do it right.**
- Computers enabled access to vast amounts of **digital data, making new relationships “visible,”** and changing the scale.

A Universal Meta-Science?

Logic has been described as a **meta-science** because its results apply to all forms of reasoning and discourse in any rational subject.

Computing on digital information resources allows us to create knowledge **inaccessible** without this new science.

This is characteristics of a **universal meta-science**.


The Key Idea

By computing better and faster on numbers, formulas, theories, and models we have made it possible to **create with computer assistance** new knowledge and more certain knowledge in virtually every field, knowledge that would be inaccessible without advanced computing on vast amounts of digital information.

Basic Facts about the Information Sciences

The impact of the information sciences on the academy will be large, changing how we *create*, *preserve* and *disseminate knowledge* – that's university business, and it increases demand for *computer science* in other related academic disciplines.

Plan of the Talk




1. Reveal key idea by examples
-  2. **Examine the impact on universities**
3. Historical Perspective and Conclusion

How we are responding in North America?

- Creating colleges (faculties) of **CIS**
- Broadening the **I-Schools**
- Broadening the Computing Research Association (**CRA**), now includes **CRA Deans** Committee
- Changing Computer Science education
- Exploring **cross-cutting academic structures** – multidisciplinary and interdisciplinary

College Structures

There are over a dozen **colleges of CIS** and over 20 **I-schools** in North America. Here are three of the top CIS colleges:

- Cornell 
- CMU 
- Georgia Tech 

Comparing Colleges

Cornell



- Computer Science
- Statistics
- Information Science
- Computational Biology
- Computational Science & Engineering
- (Digital Arts)

CMU



- Computer Science
- Machine Learning
- HCI Institute
Language Technologies
- Robotics
- Software Research
- Entertainment
Technologies

Georgia



- Computing Science & Systems
- Interactive & Intelligent Computing
- CSE

Impact of CIS

CIS will impact **every discipline** because it goes to the core of what they do. Perhaps 5% to 7% of the faculty in most disciplines will want to be connected as well to a center of CIS research and teaching.

5% to 7% of faculty
(at Cornell 80 to 110)

Impact of CIS – continued

Students will be increasingly computer savy and demand to know how computational thinking applies to their interests.

The economy will need more **knowledge workers**.

Taken together, this means having **many more faculty trained in CIS** and connected to it academically as well as intellectually.

Plan of the Talk

1. Reveal key idea by examples
2. Examine impact on universities
- ➔ 3. **Historical Perspective and Conclusion**

Historical perspective

The **Industrial Revolution (IR1)** is about: extending muscle power (mass, energy, force, power, space, and time)

The **Information Revolution (IR2)** is about: extending brains (information, intelligent processes, computation, complexity, and networks)

IR1 created colleges of engineering, shaping the **physical sciences**.

IR2 is creating colleges of computing, shaping the **information sciences**.

Article: *Transforming the Academy*

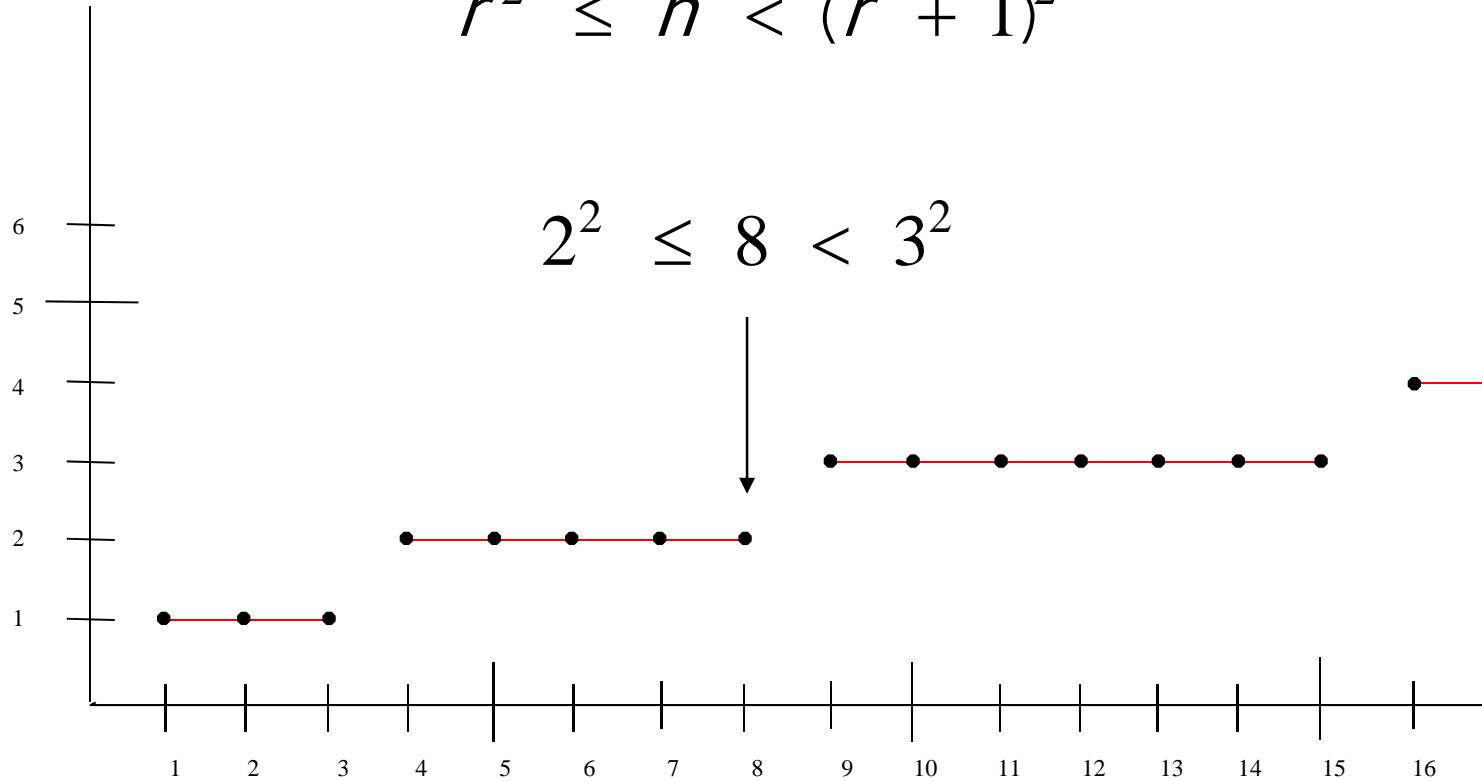
PhysicaPlus Issue 9, January, 2007 Online Magazine of the Israel Physical Society “To grasp the magnitude of the changes we face, it is important to realize that knowledge created with computer assistance goes well beyond classical knowledge formation – arising from computer processing of digital information resources on a scale that could not be achieved by all peoples of the earth acting in concert using all their cognitive powers. Computers already provide sufficient discriminatory powers that scale and speed compensate for their currently limited intelligence as they draw conclusions, make predictions and participate in discoveries.”

THE
END

Integer Square Root

$$r^2 \leq n < (r + 1)^2$$

$$2^2 \leq 8 < 3^2$$



Proof of Root Theorem

$$\forall n : \mathbb{N}. \exists r : \mathbb{N}. r^2 \leq n < (r + 1)^2$$

BY **allR**

$$n : \mathbb{N} \vdash \exists r : \mathbb{N}. r^2 \leq n < (r + 1)^2$$

BY **NatInd 1**

....base case....

$$\vdash \exists r : \mathbb{N}. r^2 \leq 0 < (r + 1)^2$$

BY **existsR [0]** THEN Auto

....induction case

$$i : \mathbb{N}^+, r : \mathbb{N}, r^2 \leq i - 1 < (r + 1)^2$$

$$\vdash \exists r : \mathbb{N}. r^2 \leq i < (r + 1)^2$$

BY **Decide [(r + 1)² ≤ i]** THEN Auto

Proof of Root Theorem (continued)

....Case 1....

$$i : \mathbb{N}^+, r : \mathbb{N}, r^2 \leq i - 1 < (r + 1)^2, (r + 1)^2 \leq i$$

$$\vdash \exists r : \mathbb{N}. r^2 \leq i < (r + 1)^2$$

BY existsR $\lceil \underline{r} + 1 \rceil$ THEN Auto '

....Case 2....

$$i : \mathbb{N}^+, r : \mathbb{N}, r^2 \leq i - 1 < (r + 1)^2, \neg((r + 1)^2 \leq i)$$

$$\vdash \exists r : \mathbb{N}. r^2 \leq i < (r + 1)^2$$

BY existsR $\lceil \underline{r} \rceil$ THEN Auto

Nuprl's Automation (Auto)

Consider what Auto' can figure out

It knows $(r + 1)^2 \leq i$ and

$$(i-1) < (r + 1)^2$$

Thus $i < (r + 1)^2 + 1$

Need to know $i < ((r + 1) + 1)^2$

Is $(r + 1)^2 + 1 \leq (r + 2)^2 = r^2 + 4r + 4$?

$$r^2 + 2r + 2 < r^2 + 4r + 4$$

$$2(r + 1) < 4(r + 1)$$

yes

New Program 2

Here is...

```
fun sqrt  $n$  = if  $n = 0$  then 0
             else let val  $r$  = sqrt ( $n - 1$ )
                  in
                     if  $n < (r + 1)^2$  then  $r$ 
                     else  $r + 1$ 
                  end
```

Conclusion: Only the Beginning

We are in early stages of the **Information Revolution**.

Combining digital information with digital computation is an **explosive mix**. We see machines that help us reason, that automatically seek information on the Web to help make progress on solving a problem.

It will become more clear that CIS is about strengthening the symbiotic relationship between people and machines used **to create and use new knowledge**.