Turing vs. Bourbaki

Teaching Mathematics to CS Majors

...and to everybody else

Klaus Sutner
Carnegie Mellon University

Outline

1. CS Curriculum
2. The Framework
3. The Backbone
4. Support Structure

A Brief History of Information

- Language: 50,000
- Writing: 5,000
- Printing: 500
- **Computing**: 50
- Web 2.0: 5

The Challenge

Obvious Problem: the exponential speedup.

Obvious Solution:

Push career-long learning and

focus the durable intellectual core of the discipline.

Looking Forward

Extra care is necessary to identify important trends early and adjust the curriculum accordingly (on fairly rare occasions).

- Formal methods: proving programs correct.
- Parallel computation: executing algorithms on multiple CPUs/cores.

100-Level: Core I

- 15-151: Mathematical Foundations of CS
- 15-122: Foundations of Imperative Programming
- 15-150: Foundations of Functional Programming

200-Level: Core II

- 15-210: Sequential and Parallel Algorithms
- 15-213: Introduction to Computer Systems
- 15-251: Great Theoretical Ideas in CS
- 15-451: Algorithm Design and Analysis

Algorithms and Complexity Electives

- 15-354: Computational Discrete Mathematics
- 15-355: Modern Computer Algebra
- 15-453: Formal Languages, Automata, and Computability
- 15-455: Undergraduate Complexity Theory
- 21-301: Combinatorics
- 21-484: Graph Theory

Logics and Languages Electives

- 15-312: Foundations of Programming Languages
- 15-317: Constructive Logic
- 15-414: Bug Catching: Automated Program Verification
- 15-424: Foundations of Cyber-Physical Systems
- 21-300: Basic Logic
- 80-311: Undecidability and Incompleteness

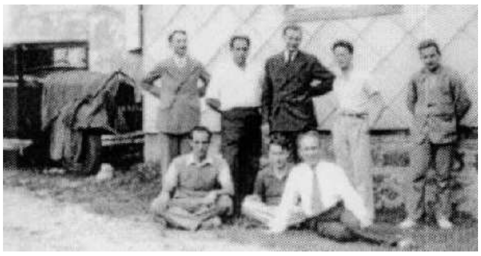
Principles

- Keep the core as small as ever possible.
- Provide multiple paths through the electives.
- Encourage interdisciplinary work across the university.
- Encourage research, independent studies, senior theses.

Incidentally, we are failing miserably on the last item.

Bourbaki, Charles Denis Sauter

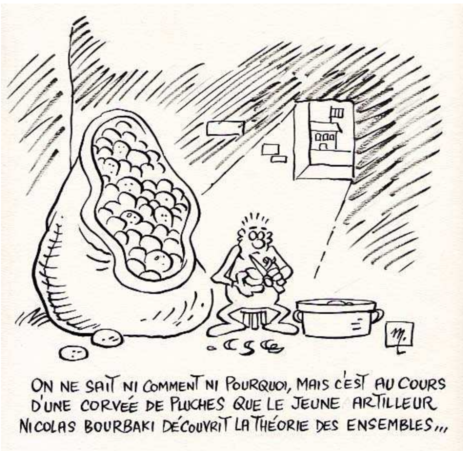
Bourbaki, Nicolas



Bourbaki Volumes

I Set theory
II Algebra
III Topology
IV Functions of one real variable
V Topological vector spaces
VI Integration
VII Commutative algebra
VIII Lie groups

In the Beginning



Bourbaki Idiosyncrasies

- Presentation strictly linear, no external references.
- Problem solving is secondary to axiomatics.
- Algorithmic content is off-topic.
- Combinatorial structure is non-essential.
- Logic is treated minimally.
- Applications nowhere in sight.
- And (cela va sans dire) no pictures.

Introductory Calculus Course, 1970's

- The natural numbers form a monoid, described by the Peano axioms.
- $\mathbb{N}$ can be extended to a commutative group, $\mathbb{Z}$.
- Lo and behold, $\mathbb{Z}$ carries a ring structure.
- Rings are nice, but fields are better: localize to get $\mathbb{Q}$.
- Rationals are great, but there are lots of gaps: the Cauchy completion has none: voila $\mathbb{R}$.
- ...
- Now let's prove Stokes' theorem on C^1 hypersurfaces in n-space.

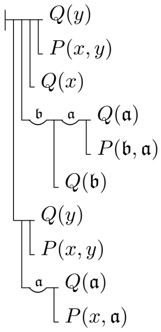
Carnap's Notes of Frege

In a mathematical lecture two things are always mixed together:

1. the pure inferences,
2. the commentary on them.

This mixture has the potential to negatively influence mathematical rigor. In conceptual notation, *assuming a complete understanding of it*, words are superfluous.

Begriffsschrift

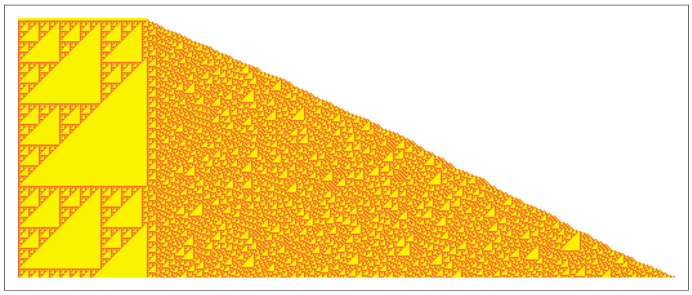


Turing, Alan

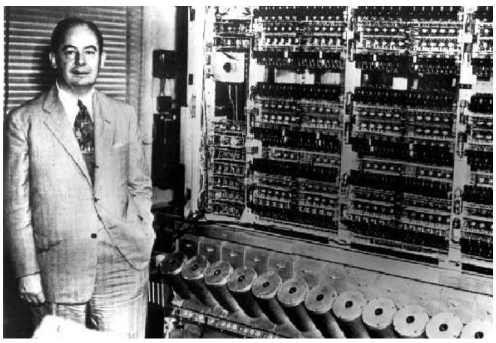
Alan Turing

> I expect that digital computing machines will eventually stimulate a considerable interest in symbolic logic . . . The language in which one communicates with these machines . . . forms a sort of symbolic logic.
>
> A. Turing, address to London Mathematical Society, 1947

A $(3, 2)$ Turing Machine



Computation



CS Students

Typically have great facility with programming–but not with the fundamental mathematical machinery.

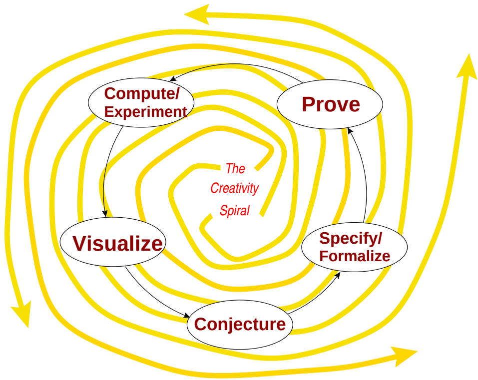
- Exploit this facility to enhance their understanding of math.
- Exploit understanding of math to improve their ability to solve algorithmic problems.

This means in particular: reason about programs.

The Golden Opportunity

> The standard of correctness and completeness necessary to get a computer program to work at all is a couple of orders of magnitude higher than the mathematical community's standard of valid proofs.
>
> Bill Thurston

Buchberger's Spiral



. . . and everybody else

> The computer has already started doing to mathematics what the telescope and microscope did to astronomy and biology. In the future not all mathematicians will care about absolute certainty, since there will be so many exciting new facts to discover: mathematical pulsars and quasars that will make the Mandelbrot set seem like a mere Galilean moon.
>
> D. Zeilberger

More Seriously

> Generally, computer science, that no-nonsense child of logic, will exert growing influence on our thinking about the languages by which we express our vision of mathematics.
>
> Y. Manin

Even More Seriously

> I cant see how else it will go. I think the process will be first accepted by some small subset, then it will grow, and eventually it will become a really standard thing. The next step is when it will start to be taught at math grad schools, and then the next step is when it will be taught at the undergraduate level. That may take tens of years, I dont know, but I dont see what else could happen.
>
> Vladimir Voevodsky

Astonishing connections between Martin-Löf type theory and classical homotopy theory.

Bourbaki Upside Down

31

- Problem solving central motivation.
- Algorithmic content is crucial.
- Combinatorial structure is essential.
- Logic is the foundation.
- Applications are ubiquitous.
- Visualization is used extensively.
- Programming is fully integrated.

Ancient History

32

First major attempt: ModMath in 1996.

Dana Scott, Marko Petkovsek, KS

Heavily based on *Mathematica*.

Abysmal failure.

"Why do I have to learn another stupid language?"

Ancient History

32

First major attempt: ModMath in 1996.

Dana Scott, Marko Petkovsek, KS

Heavily based on *Mathematica*.

Abysmal failure.

"Why do I have to learn another stupid language?"

The Central Disconnect?

33

Computer algebra is perfect to sharpen one's intuition, generate examples, counterexamples, perform arithmetic drudgery, visualize data.

It does not help much with constructing proofs which appear to be the central object of apprehension for most of the students.

Alas, theorem provers are currently too unwieldy to be used early on. I later tried a version of Ed Clarke's Analytica prover, with very mixed results.

The Problem With Proofs

34

> It became dramatically clear how much proofs depend on the audience. We prove things in a social context and address them to a certain audience.
>
> W. Thurston

> I do still believe that rigor is a relative notion, not an absolute one. It depends on the background readers have and are expected to use in their judgment.
>
> R. Thom

> A proof only becomes a proof after the social act of "accepting it as a proof".
>
> Y. Manin

Solution?

35

Start at a very low lever of granularity, using a highly verbose style on fairly simple problems.

Exploit Thurston's observation to buy into the high level of precision (point at compilers).

Gradually relax the required level of detail (like higher level reasoning about programs) and amp up the level of difficulty.

Keep fingers crossed at all times.

Dream Machine

Support for Experimental Mathematics

- Document-centric user interface: compute, visualize, store, document.
- Interactive, expressive prototyping language.
- General compute engine to manipulate data.
- Reasonably large algorithm base.
- Fast code to generate sizable structures.
- Seamlessly integrated system between different domains.
- Strong support for graphics.
- Hooks for a proof checker/theorem prover.

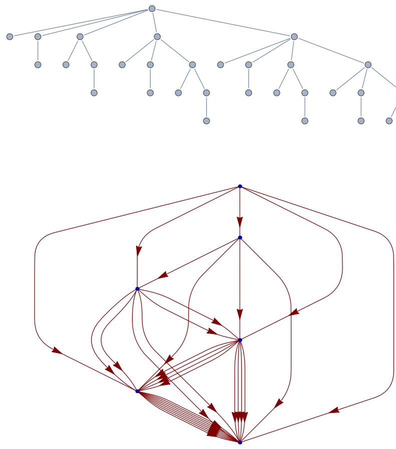
Example: Setty

CDM employs the hypothetical programming language F++ (Frege-plus-plus) to get students to think about sets in a more relaxed manner.

Adam Blank, PhD student at CMU, implemented a little language Setty for 15-151 where computations in hereditarily finite sets can be carried out quite easily.

For example, one can easily implement the unpairing functions for Kuratowski pairs or implement a primality tester based on von Neumann ordinals.

von Neumann 5



MFCS Goals

- Become familiar with what a proof is and is not
- Be able to make well-structured, thought-out arguments in response to mathematical questions
- Get an understanding for how mathematics is fundamentally a part of computer science
- Learn basic problem solving in mathematical topics like functions, graphs, combinatorics, and probability
- Learn how to identify the “right technique” to solve a problem

Flipped Classroom

CMU has 2 classrooms for collaborative work (Y shaped tables, lots of computers, wall screens).

Workshops where student solve problems in groups of 3, one TA for 9 students.

One side-effect: listening in on student conversation provides much better feedback that quizzes, midterms, finals, even office hours.

Very popular among students, much less so with faculty. Essentially unsustainable.

MFCS Topics

Superficially similar to any standard introductory discrete math course, but quite different under the cover.

Foundations

Is set theory really the right way to start?

I used to think the answer was an uneqivocal Yes, but I'm not sure anymore.

Great Theoretical Ideas in CS

Builds directly on MFCS, amps up the depth and difficulty of the material.

Homework problems are notoriously hard and hugely time-consuming.

Perceived as a hazing experience by some students.

Standard top-down format: lectures, recitations, office hours.

Current Topic List

- Introduction
- Proofs
- Induction
- Iteration
- Counting I
- Counting II
- Probability I
- Probability II
- Graphs I
- Graphs II
- Games I
- Games II
- Number Theory I
- Number Theory II
- Groups
- Fields and Polynomials
- Coding Theory
- Computability
- Finite State Machines
- Finite State Machines
- Lambda Calculus
- Set Theory
- Asymptotics
- Complexity I
- Complexity II
- Complexity III
- Approximation Algorithms
- Secrets of the Universe

Full Disclosure

This course lives and dies with the TAs: huge grading burden. The TAs are all undergraduates.

The reward: they get to design most of the homeworks. Central motto: why should the next class have it any better than I?

Burn-out is not uncommon, but candidates elbow each other out of the way to become a 251-TA.

Computational Discrete Math (CDM)

In a nutshell, the main idea behind this course is that the development of the digital computer, together with the theory of computation, is one of the most important development in mathematics in the 20th century. Consequently, this course takes a fresh look at some of the standard concepts of discrete mathematics (relations, functions, logic, graphs, algebra, automata), with strong and consistent emphasis on computation and algorithms.

CDM

Unlike MFCS and GTI, this course is optional (constrained elective). Enrollment drops to about 10% of the class.

As a consequence, can push much further, much faster.

For example, correctness proof for Safra's determinization algorithm for ω-automata.

Universality


```
 0:   copy   C  R  1            // R = C
 1:   write  R  p  x   2        // R[0] = x
 2:   read   C  p  I   3        // I = C[p]
 3:   pop    I  r  4            // r = pop(I)
 4:   zero   I 13  5            // if( I == 0 ) halt
 5:   pop    I  p  6            // p = pop(I)
 6:   read   R  r  x   7        // x = R[r]
 7:   zero   I  8  9            // if( I != 0 ) goto 9
 8:   inc    x 12               // x++; goto 12
 9:   zero   x 10 11            // if( x != 0 ) goto 11
10:   pop    I  p  2            // p = pop(I)
11:   dec    x 12 12            // x--
12:   write  R  r  x   2        // R[r] = x; goto 2
13:   halt
```

Logic

> **5.43**
>
> *But in fact all propositions of logic say the same thing, to wit nothing.*
>
> *L. Wittgenstein*

Algorithms around Propositional Logic

- Normal forms (negation, conjunctive, disjunctive)
- Boolean matrices
- Davis-Putnam-Logeman-Lovelace

Apply to solve combinatorial problems.

Long Formulae

$$
\begin{aligned}
&(\neg p_{11} \lor p_{32} \land p_{23}) \land (\neg p_{12} \lor p_{31} \land p_{33} \land p_{24}) \land \\
&(\neg p_{13} \lor p_{32} \land p_{34} \land p_{21} \land p_{25}) \land (\neg p_{14} \lor p_{33} \land p_{35} \land p_{22} \land p_{26}) \land \\
&(\neg p_{15} \lor p_{34} \land p_{36} \land p_{23} \land p_{27}) \land (\neg p_{16} \lor p_{35} \land p_{37} \land p_{24} \land p_{28}) \land \\
&(\neg p_{17} \lor p_{36} \land p_{38} \land p_{25}) \land \\
&(\neg p_{18} \lor p_{37} \land p_{26}) \land (\neg p_{21} \lor p_{42} \land p_{13} \land p_{33}) \land \\
&(\neg p_{22} \lor p_{41} \land p_{43} \land p_{14} \land p_{34}) \land \\
&(\neg p_{23} \lor p_{42} \land p_{44} \land p_{11} \land p_{15} \land p_{31} \land p_{35}) \land \\
&(\neg p_{24} \lor p_{43} \land p_{45} \land p_{12} \land p_{16} \land p_{32} \land p_{36}) \land \\
&(\neg p_{25} \lor p_{44} \land p_{46} \land p_{13} \land p_{17} \land p_{33} \land p_{37}) \land \\
&(\neg p_{26} \lor p_{45} \land p_{47} \land p_{14} \land p_{18} \land p_{34} \land p_{38}) \land \\
&(\neg p_{27} \lor p_{46} \land p_{48} \land p_{15} \land p_{35}) \land \\
&(\neg p_{28} \lor p_{47} \land p_{16} \land p_{36}) \land (\neg p_{31} \lor p_{12} \land p_{52} \land p_{23} \land p_{43}) \land \\
&(\neg p_{32} \lor p_{11} \land p_{13} \land p_{51} \land p_{53} \land p_{24} \land p_{44}) \land \\
&(\neg p_{33} \lor p_{12} \land p_{14} \land p_{52} \land p_{54} \land p_{21} \land p_{25} \land p_{41} \land p_{45}) \land \\
&(\neg p_{34} \lor p_{13} \land p_{15} \land p_{53} \land p_{55} \land p_{22} \land p_{26} \land p_{42} \land p_{46}) \land \\
&(\neg p_{35} \lor p_{14} \land p_{16} \land p_{54} \land p_{56} \land p_{23} \land p_{27} \land p_{43} \land p_{47}) \land \\
&(\neg p_{36} \lor p_{15} \land p_{17} \land p_{55} \land p_{57} \land p_{24} \land p_{28} \land p_{44} \land p_{48}) \land \\
&(\neg p_{37} \lor p_{16} \land p_{18} \land p_{56} \land p_{58} \land p_{25} \land p_{45}) \land \\
&(\neg p_{38} \lor p_{17} \land p_{57} \land p_{26} \land p_{46}) \land (\neg p_{41} \lor p_{22} \land p_{62} \land p_{33} \land p_{53}) \land \\
&(\neg p_{42} \lor p_{21} \land p_{23} \land p_{61} \land p_{63} \land p_{34} \land p_{54}) \land \\
&(\neg p_{43} \lor p_{22} \land p_{24} \land p_{62} \land p_{64} \land p_{31} \land p_{35} \land p_{51} \land p_{55}) \land \\
&(\neg p_{44} \lor p_{23} \land p_{25} \land p_{63} \land p_{65} \land p_{32} \land p_{36} \land p_{52} \land p_{56}) \land \ldots
\end{aligned}
$$

Equational Theorem Proving

Famous conjecture from 1933, Robbins' Conjecture:

$$
\begin{aligned}
x &= \overline{\overline{x+y} + \overline{x+\overline{y}}} \\
x + y &= y + x \\
(x+y)+z &= x+(y+z)
\end{aligned}
$$

together imply the double negation property: $\overline{\overline{x}} = x$.

Proven in 1996 by the automatic prover EQP.

1994: Pentium FDIV Bug

Floating Point Error

$4195835.0/3145727.0 = 1.3338204491362410025$ correct
$4195835.0/3145727.0 = 1.3337390689020375894$ pentium

Alternatively

$4195835.0 - 3145727.0 * (4195835.0/3145727.0) = 0$ correct
$4195835.0 - 3145727.0 * (4195835.0/3145727.0) = 256$ pentium

Discovered in October 1994 by number theorist Thomas R. Nicely, doing research in pure math.

Temporal Logic

- $\mathbf{EX}\,\varphi$: for some s' such that $s \to s'$: $\mathcal{A}, s' \models \varphi$.
- $\mathbf{AX}\,\varphi$: for all s' such that $s \to s'$: $\mathcal{A}, s' \models \varphi$.
- $\mathbf{EG}\,\varphi$: there exists a path (s_i) starting at s such that $\mathcal{A}, s_i \models \varphi$ for all i.
- $\mathbf{AG}\,\varphi$: for all paths (s_i) starting at s we have $\mathcal{A}, s_i \models \varphi$ for all i.
- $\mathbf{EF}\,\varphi$: there exists a path (s_i) starting at s such that $\mathcal{A}, s_i \models \varphi$ for some i.

MSO on Infinite Words

W	a	a	b	a	a	a	b	a	a	a	…
x	0	0	1	0	0	0	0	0	0	0	…
X	0	0	1	0	1	0	1	0	0	0	…

Decidability of Presburger arithmetic (program correctness applications)

$$\mathfrak{N}_0 = \langle \mathbb{N}, +, <, 0 \rangle$$

Computing with Polynomials

Cycle index polynomial for dihedral group D_{20}:

$$\frac{1}{40}(x_1^{20} + 10x_2^9 x_1^2 + 11x_2^{10} + 2x_4^5 + 4x_5^4 + 4x_{10}^2 + 8x_{20})$$

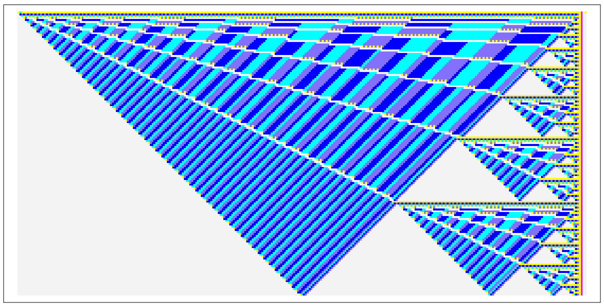
Substituting three colors:

$$\tfrac{1}{40}\big((c_1+c_2+c_3)^{20} + 10(c_1^2+c_2^2+c_3^2)^9 (c_1+c_2+c_3)^2 + 11(c_1^2+c_2^2+c_3^2)^{10} + 2(c_1^4+c_2^4+c_3^4)^5 + 4(c_1^5+c_2^5+c_3^5)^4 + 4(c_1^{10}+c_2^{10}+c_3^{10})^2 + 8(c_1^{20}+c_2^{20}+c_3^{20})\big)$$

Expand . . .

$$c_1^{20} + c_2c_1^{19} + c_3c_1^{19} + 10c_2^2c_1^{18} + 10c_3^2c_1^{18} + 10c_2c_3c_1^{18} + 33c_2^3c_1^{17} + 33c_3^3c_1^{17} + 90c_2c_3^2c_1^{17} +$$
$$90c_2^2c_3c_1^{17} + 145c_2^4c_1^{16} + 145c_3^4c_1^{16} + 489c_2c_3^3c_1^{16} + 774c_2^2c_3^2c_1^{16} + 489c_2^3c_3c_1^{16} + 406c_2^5$$
$$c_1^{15} + 406c_3^5c_1^{15} + 1956c_2c_3^4c_1^{15} + 3912c_2^2c_3^3c_1^{15} + 3912c_2^3c_3^2c_1^{15} + 1956c_2^4c_3c_1^{15} + 1032c_2^6c_1^{14} + 1032c_3^6c_1^{14} +$$
$$5832c_2c_3^5c_1^{14} + 14724c_2^2c_3^4c_1^{14} + 19416c_2^3c_3^3c_1^{14} + 14724c_2^4c_3^2c_1^{14} + 5832c_2^5c_3c_1^{14} + 1980c_2^7$$
$$c_1^{13} + 1980c_3^7c_1^{13} + 13608c_2c_3^6c_1^{13} + 40824c_2^2c_3^5c_1^{13} + 67956c_2^3c_3^4c_1^{13} + 67956c_2^4c_3^3c_1^{13} + 40824c_2^5$$
$$c_3^2c_1^{13} + 13608c_2^6c_3c_1^{13} + 3260c_2^8c_1^{12} + 3260c_3^8c_1^{12} + 25236c_2c_3^7c_1^{12} + 88620c_2^2c_3^6c_1^{12} + 176484c_2^3c_3^5c_1^{12} +$$
$$221110c_2^4c_3^4c_1^{12} + 176484c_2^5c_3^3c_1^{12} + 88620c_2^6c_3^2c_1^{12} + 25236c_2^7c_3c_1^{12} + 4262c_2^9c_1^{11} +$$
$$4262c_3^9c_1^{11} + 37854c_2c_3^8c_1^{11} + 151416c_2^2c_3^7c_1^{11} + 352968c_2^3c_3^6c_1^{11} + 529452c_2^4c_3^5c_1^{11} + 529452c_2^5c_3^4c_1^{11} +$$
$$352968c_2^6c_3^3c_1^{11} + 151416c_2^7c_3^2c_1^{11} + 37854c_2^8c_3c_1^{11} + 4752c_2^{10}c_1^{10} + 4752c_3^{10}c_1^{10} + 46252c_2c_3^9c_1^{10} + 208512c_2^2$$
$$c_1^{10} + 554520c_2^3c_3^7c_1^{10} + 971292c_2^4c_3^6c_1^{10} + 1164342c_2^5c_3^5c_1^{10} + 971292c_2^6c_3^4c_1^{10} + 554520c_2^7c_3^3c_1^{10} + 208512c_2^8c$$
$$c_2^9c_3c_1^{10} + 4262c_2^{11}c_1^9 + 4262c_3^{11}c_1^9 + 46252c_2c_3^{10}c_1^9 + 231260c_2^2c_3^9c_1^9 + 693150c_2^3c_3^8c_1^9 + 1386300c_2^4c_3^7c_1^9 +$$
$$1940568c_2^5c_3^6c_1^9 + 1940568c_2^6c_3^5c_1^9 + 1386300c_2^7c_3^4c_1^9 + 693150c_2^8c_3^3c_1^9 + 231260c_2^9c_3^2c_1^9 +$$
$$46252c_2^{10}c_3c_1^9 + 3260c_2^{12}c_1^8 + 3260c_3^{12}c_1^8 + 37854c_2c_3^{11}c_1^8 + 208512c_2^2c_3^{10}c_1^8 + 693150c_2^3c_3^9c_1^8 +$$
$$1560534c_2^4c_3^8c_1^8 + 2494836c_2^5c_3^7c_1^8 + 2912112c_2^6c_3^6c_1^8 + 2494836c_2^7c_3^5c_1^8 + 1560534c_2^8c_3^4c_1^8 + 693150c_2^9c_3^3c_1^8 +$$
$$37854c_2^{11}c_3c_1^8 + 1980c_2^{13}c_1^7 + 1980c_3^{13}c_1^7 + 25236c_2c_3^{12}c_1^7 + 151416c_2^2c_3^{11}c_1^7 + 554520c_2^3c_3^{10}c_1^7 + \ldots$$

Firing Squad



Student Comments

-
- The instructor could make studying the mathematics of a rock interesting.
- Klaus is like Chuck Norris if Chuck Norris was brilliant and as awesome as his memes. The class is fantastic and I am so glad to be taking it.

Simon Initiative

http://www.cmu.edu/simon

Built on the core principles of learning advanced by CMUs Nobel Laureate and pioneering educator, Dr. Herbert A. Simon, whose work linked cognitive models of learning with computation tools, the Simon Initiative makes the learner its focus and measurably improving learning outcomes its goal.

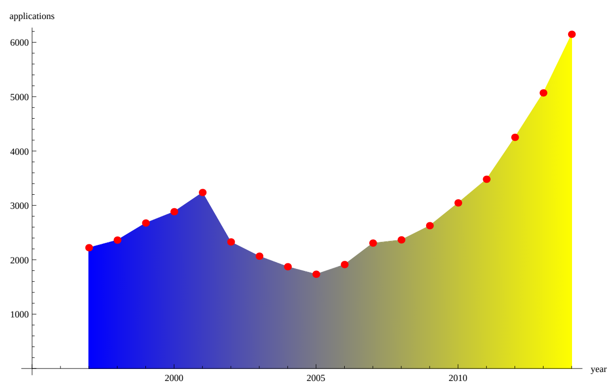
Die Botschaft hör' ich wohl, allein mir fehlt der Glaube.

Goethe

Carl Wieman

- motivational tactics
- build upon prior thinking and knowledge
- introduce at most 7 (seven) new items in one lecture
- design strenuous practice activities
- provide timely and specific feedback

The Onslaught



The Grading Problem

A theory course is not a spectator sport, it requires lots of problem solving and engagement.

Alas, that produces vast numbers of homework solutions to be evaluated and commented.

Traditional solution: hire armies of (mostly undergraduate) TAs. Require lots of feeding and care, results often unimpressive.

Verifications

Judgment based (peer) grading, Adam Blank, PhD student at CMU:

- Students submit structured LaTeX.
- System compiles, collates, generates PDF.
- TAs sample at random, generate small taxonomy of errors.
- Verification: students grade each others work based on given taxonomy, annotate PDF accordingly.
- TAs do quality control.

Alternatively TAs do all the grading using the same system; major speed-up, better fairness.

Preserving Content

> "Universities generate content every day through their courses and seminars. Then they throw it away. There is a certain charm with this approach, but it is not cost effective. Universities operate like renaissance quartets based on live performances. ...Content storage and reuse are also important to test and ameliorate performance and to generate an institutional memory."
>
> Dennis Tsichritzis, "Reengineering the University," CACM June 1999

Preserving Content, II

> "Jesus saves, and so should you."
>
> Dana Scott

Assessment

Force Concept Inventory in mechanics from the 1990's, Halloun and Hestenes (Arizona State).

Huge impact on physics education, Eric Mazur at Harvard, Carl Wieman at Stanford.

Provides a reasonable measure of learning outcomes, of student progress.

Results are not pretty.

A Plea

> We need a concept inventory of discrete math/TCS.

Anecdotal evidence is no longer a sufficient basis for curriculum design.

Even a small project would help, say, a CI for propositional logic.

Thank You.