how to build a library of formalized mathematics

Freek Wiedijk
Radboud University Nijmegen

MathWiki Workshop
University of Edinburgh

2007 10 31, 11:00
state of the art

top 100

http://www.cs.ru.nl/~freek/100/

google 100 theorems
current systems

• interesting

HOLs
– HOL Light  63
– ProofPower  39
– Isabelle/HOL  36

non-HOLs
– Coq  39
– Mizar  39

• not in the top five
– PVS  15
– NuPRL  12
– ACL2  8
the 20 unformalized theorems

12. The Independence of the Parallel Postulate
16. Insolvability of General Higher Degree Equations
21. Green’s Theorem
24. The Undecidability of the Continuum Hypothesis
28. Pascal’s Hexagon Theorem
29. Feuerbach’s Theorem
33. Fermat’s Last Theorem
41. Puiseux’s Theorem
43. The Isoperimetric Theorem
47. The Central Limit Theorem
48. Dirichlet’s Theorem
50. The Number of Platonic Solids
53. Pi is Transcendental
56. The Hermite-Lindemann Transcendence Theorem
59. The Laws of Large Numbers
62. Fair Games Theorem
67. e is Transcendental
76. Fourier Series
82. Dissection of Cubes
92. Pick’s Theorem
current libraries

- many people, badly organized
  - MML
  - AFP
  - Coq contribs

- one person, well organized
  - John Harrison
  - Georges Gonthier

Mizar
Isabelle/HOL
Coq
HOL Light
Coq
looks do matter

fake problems

• ‘it is too much work’
  
  *de Bruijn factor in space*: about 4 times
  *de Bruijn factor in time*: about 10 times = about 1 week/page
  *all of undergraduate mathematics*: about 140 man-years

  not expensive!

• ‘it is not useful’
  
  – correctness
  – explicitness
  – art

• ‘mathematicians will not want it’
real problems

- insufficient automation
  - computer algebra is much more powerful
  - automation of high school mathematics

\[
x = \frac{i}{n}, \quad n = m + 1 \vdash n! \cdot x = i \cdot m!
\]

\[
\frac{k}{n} \geq 0 \vdash \left| \frac{n - k}{n} - 1 \right| = \frac{k}{n}
\]

\[
n \geq 2, \quad x = \frac{1}{n + 1} \vdash \frac{x}{1 - x} < 1
\]

- no good way to write calculus

  formulas in proof assistants ↔ formulas in a calculus textbook
provocative statement 1

a library that does not code the calculus formula

\[ \sum_{n=\infty}^{\infty} e^{int} \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-ins} f(s) \, ds \]

in a way that is very close to the computer algebra term

\[
\text{sum}(e^{(I*n*t)}/(2*pi)*\text{int}(e^{(-I*n*s)*f(s)},s=-pi..pi),\n\text{n=-infinity..infinity})
\]

will never be widely used
real problems (continued): too unlike real mathematics

- **the look of the proofs**
  
  intros k l H; induction H as [|l H].
  intros; absurd (S k <= k); auto with arith.
  destruct H; auto with arith.

- **constructive mathematics**
  
  - reasoning by cases
    
    \(\text{a quadratic equation will have zero, one, or two roots, depending on the sign of the discriminant}\)

  - extensionality
    
    \(\text{what do you mean: ‘the complex square root is not extensional?’}\)
a library that supports constructive reasoning will never be widely used
... unless the constructivity can be completely ignored by classical users
... but that will not be feasible
portability to the future

idiosyncratic ↔ canonical

• statements
  - HOL
  - FOL + soft types

• proofs
  - declarative proofs
    - Mizar, Isar, Christophe Raffalli, Pierre Corbineau, …
    - Fitch-style natural deduction

  independent of the specifics of the system
portability to the future (continued)

\[
\begin{array}{c}
\frac{1}{0} \ ? \\
\frac{1}{0} = 0? \frac{1}{0} \text{ is an unknown number? } \frac{1}{0} \text{ is a non-denoting term? } \frac{1}{0} \text{ is illegal?}
\end{array}
\]

(I do not like proof terms in my formulas either)
(I like partial logics about as much as I like constructive logics)
none of the existing systems is portable to the future

... so any library of formal mathematics will have to be redone later
it’s a social problem

definitions

four kinds of information in a formal library

– definitions

– statements

– proofs

– tactics / decision procedures

the statements should be what matters

the right definitions?

the right notions
are conceptual advances helpful?

coercions
subtyping
record types
module systems
type universes
canonical structures
binders
induction-recursion
coinduction
partiality

all pretty much irrelevant
why don’t we have a good library of formalized mathematics yet?

what are the main obstacles?

• social?

• engineering?

• mathematical?
obstacles

• social problem
  many people *and* well organized
  how to decide on the definitions?
  how to decide on the names of the theorems?
  how to decide on the structure of the library?

• engineering problem
  good formalization of calculus
  automation of high school mathematics

• mathematical problem
  how to deal with partiality?
building a good library of formal mathematics is a social problem

... the main problem is to keep the library well organized

... after having solved the problem of getting participants in the first place
looking for a solution: the internet

‘benevolent dictatorship’

examples

– Linux

– Wikipedia
a formal library should be **flat**

... consisting of a sequence of ‘articles’

... consisting of a sequence of ‘lemmas’
looking for a solution: traditional mathematics

‘many different variations that still are usable together’

Coq and Isabelle contribs are not like this (not used together)  
John’s and Georges’ libraries are not like this (just one variation)  
Mizar’s MML is very much like this

however ‘articles’ should have two parts: preliminaries / content

– each article owned by someone

– preliminaries point to the articles where the lemmas should go

– content part should stay together
provocative statement 6

a formal library should not just be a ‘sea of lemmas’

... because a proof assistant is not a stateless thing
linking existing proof assistants together is not useful

... for the same reasons that these systems are not portable to the future.
the aim

formalization for communication of mathematics

proof assistants that are visual?