statistics on digital libraries of mathematics

Freek Wiedijk

Radboud University Nijmegen

MKM 2008

CICM 2008, Birmingham, UK

2008 06 28, 12 : 00

http://www.cs.ru.nl/~freek/notes/stats.pdf

http://www.cs.ru.nl/~freek/talks/stats.pdf

- **text**
  - paper
    journals, textbooks
  - electrons
    archives, websites
- **code**
  - mathematical tables
  - software
    numerical, computer algebra, subject specific
  - formalizations

80 out of 100 theorems

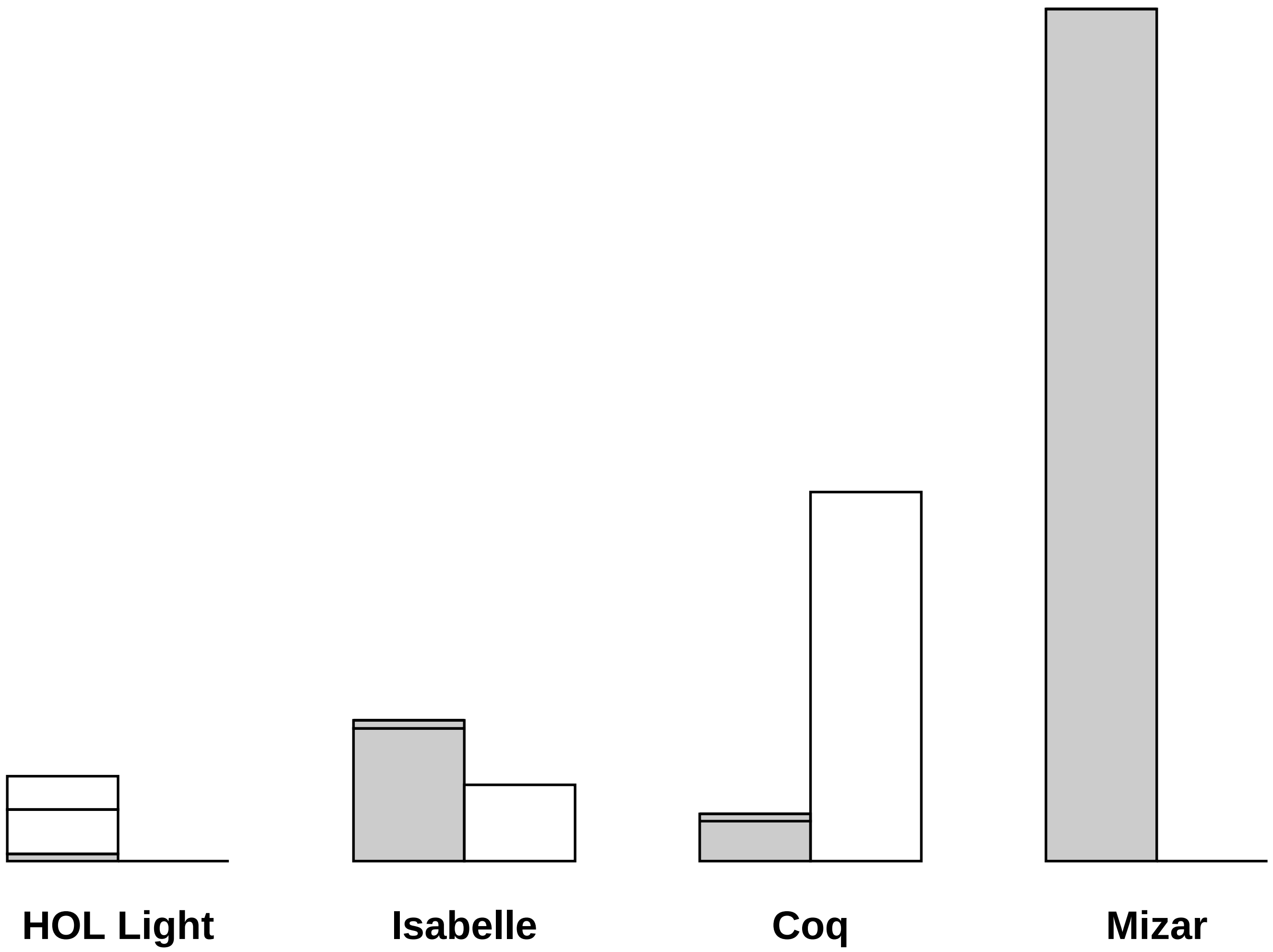
1.	The Irrationality of the Square Root of 2	≥ 17
2.	Fundamental Theorem of Algebra	4
3.	The Denumerability of the Rational Numbers	6
4.	Pythagorean Theorem	6
5.	Prime Number Theorem	2
6.	Gödel's Incompleteness Theorem	3
7.	Law of Quadratic Reciprocity	4
8.	The Impossibility of Trisecting the Angle and Doubling the Cube	1
9.	The Area of a Circle	1
10.	Euler's Generalization of Fermat's Little Theorem	4
11.	The Infinitude of Primes	6
12.	The Independence of the Parallel Postulate	0
13.	Polyhedron Formula	1
	...	

proof assistants

only five systems seriously used for mathematics

HOL	**HOL Light**	69
HOL	**ProofPower**	42
	Isabelle	40
	Coq	39
	Mizar	45

library sizes



content of the files

formalization $\approx$ long chain of **lemmas**

lemma $\approx$

label $+$

statement $+$

proof

```
C (* ========================================================================= *)
C (* Binary expansions as a bijection between numbers and finite sets.         *)
C (* ========================================================================= *)
B
T let LT_POW2_REFL = prove
T  (`!n. n < 2 EXP n`,
P   INDUCT_TAC THEN REWRITE_TAC[EXP] THEN TRY(POP_ASSUM MP_TAC) THEN ARITH_TAC);;
B
T let BINARY_INDUCT = prove
T  (`!P. P 0 /\ (!n. P n ==> P(2 * n) /\ P(2 * n + 1)) ==> !n. P n`,
P   GEN_TAC THEN STRIP_TAC THEN MATCH_MP_TAC num_WF THEN GEN_TAC THEN
P   STRIP_ASSUME_TAC(ARITH_RULE
P    `n = 0 \/ n DIV 2 < n /\ (n = 2 * n DIV 2 \/ n = 2 * n DIV 2 + 1)`) THEN
P   ASM_MESON_TAC[]);;
B
T let BOUNDED_FINITE = prove
T  (`!s. (!x:num. x IN s ==> x <= n) ==> FINITE s`,
P   REPEAT STRIP_TAC THEN MATCH_MP_TAC FINITE_SUBSET THEN EXISTS_TAC `0..n` THEN
P   ASM_SIMP_TAC[SUBSET; IN_NUMSEG; FINITE_NUMSEG; LE_0]);;
B
T let EVEN_NSUM = prove
T  (`!s. FINITE s /\ (!i. i IN s ==> EVEN(f i)) ==> EVEN(nsum s f)`,
P   REWRITE_TAC[GSYM IMP_IMP] THEN MATCH_MP_TAC FINITE_INDUCT_STRONG THEN
```

```
C (*  Title:      HOL/Quadratic_Reciprocity/Gauss.thy
C     ID:         $Id: Int2.thy,v 1.12 2007/06/11 09:06:07 chaieb Exp $
C     Authors:    Jeremy Avigad, David Gray, and Adam Kramer
C *)
B
E header {*Integers: Divisibility and Congruences*}
B
S theory Int2 imports Finite2 WilsonRuss begin
B
D definition
D   MultInv :: "int => int => int" where
D   "MultInv p x = x ^ nat (p - 2)"
B
B
E subsection {* Useful lemmas about dvd and powers *}
B
T lemma zpower_zdvd_prop1:
T   "0 < n \<Longrightarrow> p dvd y \<Longrightarrow> p dvd ((y::int) ^ n)"
P   by (induct n) (auto simp add: zdvd_zmult zdvd_zmult2 [of p y])
B
T lemma zdvd_bounds: "n dvd m ==> m \<le> (0::int) | n \<le> m"
P proof -
P   assume "n dvd m"
```

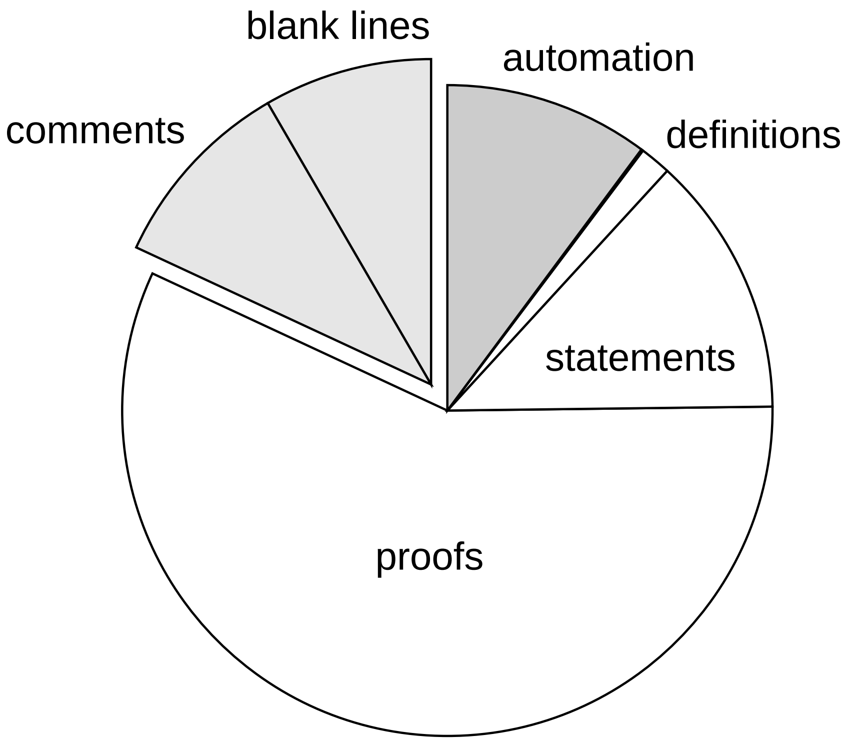
```
C (************************************************************************)
C (*  v      *   The Coq Proof Assistant  /  The Coq Development Team     *)
C (* <O___,, * CNRS-Ecole Polytechnique-INRIA Futurs-Universite Paris Sud *)
C (*   \VV/  **************************************************************)
C (*    //   *      This file is distributed under the terms of the       *)
C (*         *       GNU Lesser General Public License Version 2.1        *)
C (************************************************************************)
C (*i     $Id: Decidable.v 5920 2004-07-16 20:01:26Z herbelin $    i*)
B
E (** Properties of decidable propositions *)
B
D Definition decidable (P:Prop) := P \/ ~ P.
B
T Theorem dec_not_not : forall P:Prop, decidable P -> (~ P -> False) -> P.
P unfold decidable in |- *; tauto.
P Qed.
B
T Theorem dec_True : decidable True.
P unfold decidable in |- *; auto.
P Qed.
B
T Theorem dec_False : decidable False.
P unfold decidable, not in |- *; auto.
```

```
C :: Non negative real numbers. Part II
C ::  by Andrzej Trybulec
C ::
C :: Received March 7, 1998
C :: Copyright (c) 1998 Association of Mizar Users
B
S environ
B
S  vocabularies ARYTM_2, BOOLE, ORDINAL2, ARYTM_3, ARYTM_1;
S  notations TARSKI, SUBSET_1, ARYTM_3, ARYTM_2;
S  constructors ARYTM_2;
S  requirements SUBSET;
S  theorems ARYTM_2;
B
S begin
B
L reserve x,y,z for Element of REAL+;
B
T theorem Th1:
T   x + y = y implies x = {}
P proof reconsider o = {} as Element of REAL+ by ARYTM_2:21;
P   assume x + y = y;
P   then x + y = y + o by ARYTM_2:def 8;
```

11 kinds of lines

B	blank lines
C	comments
E	documentation
S	modules: imports and sectioning
L	contexts
D	definitions
N	notation
H	automation: directives
X	automation: program code
T	theorem statements
P	proofs

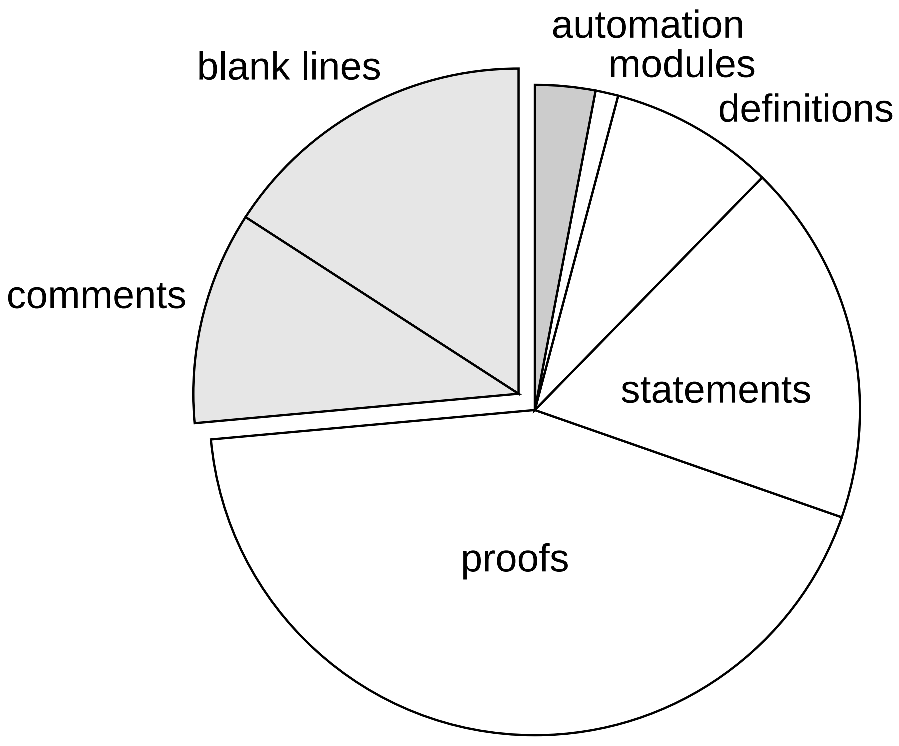
relative sizes: HOL Light



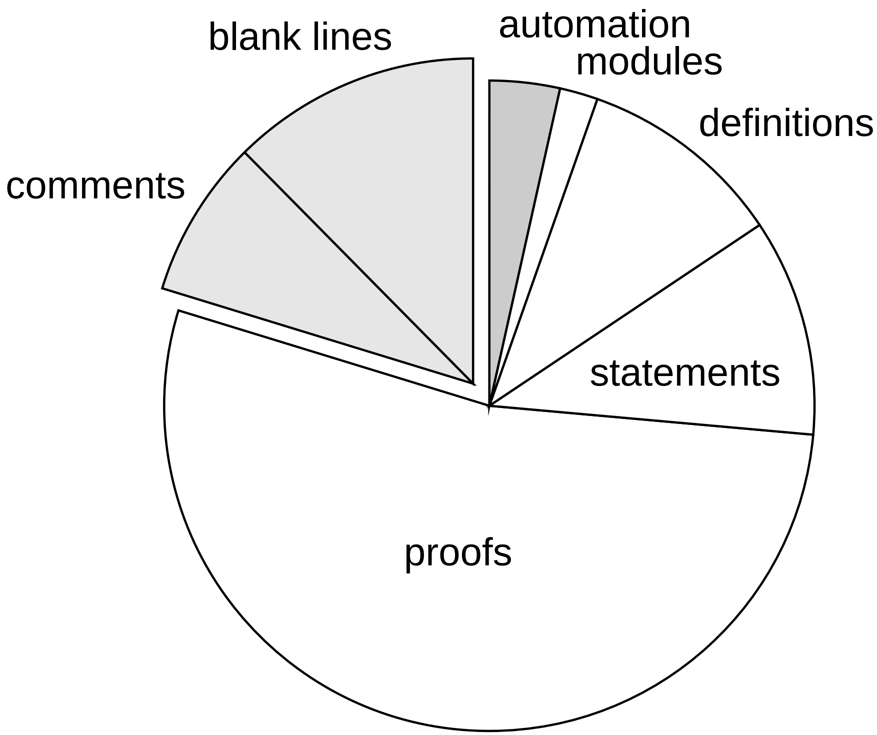
almost no module lines

small definitions part!

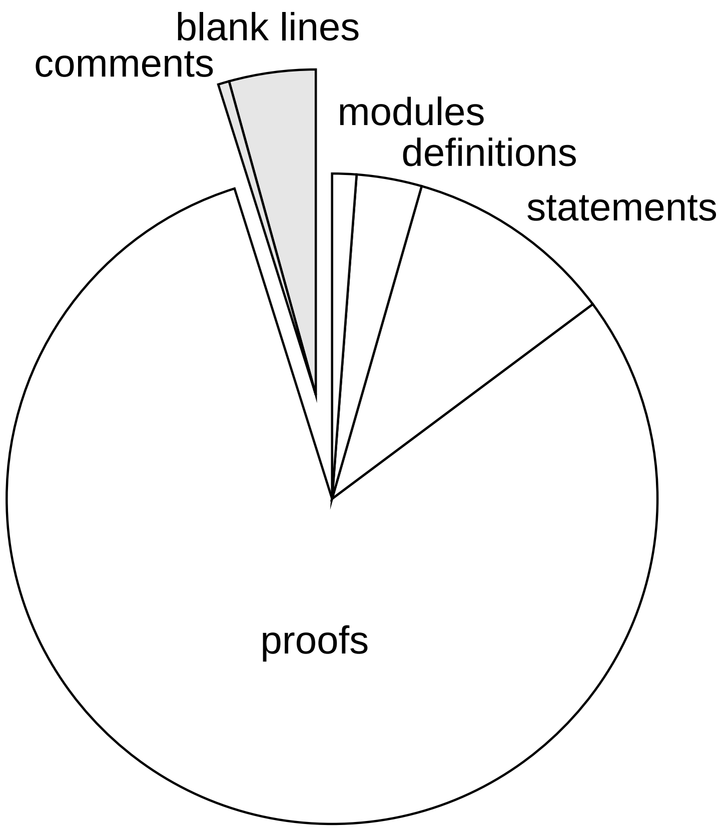
relative sizes: Isabelle



relative sizes: Coq



relative sizes: Mizar



no automation lines

almost no comments

large proofs part!

lessons learned

- formalizations all are very similar

  . . . despite fundamental differences between proof assistants (foundations, interaction styles)

- formalizations consist primarily of proofs

- classification of line types in formalizations

- small definitions are good!

- proof automation is important!