Formalizing exponents 3 and 4 of Fermat's Last Theorem using the proof-assistant 'Isabelle'

DIAMANT symposium 30 November 2007 (14.25-14.45)

Roelof Oosterhuis
Supervisors: Jaap Top & Wim Hesselink

Agenda

- Aim of my research
- Proof assistants → Isabelle
- Fermat's last theorem
  - case n=4
  - case n=3
- Results of my research
  - space factor
  - time factor
- Discussion

Pilot study for a larger ideal

"Formalize and verify by computer a proof of Fermat's Last Theorem."

Prof. dr. Jan Bergstra

(nr 1. of his list of 'ten challenging research problems for computer science')

Aims of my research:

- To give a formal proof of FLT3
- How getting started with 'formalizing mathematics'?
  - How to choose a proof assistant?
  - How getting familiar with such a program?
- How does a formalization work in practice?
  - What problems does one encounter?
  - How much time does it take?
  - How 'doable' are problems like FLT3&4?

Agenda

- Aim of my research
- Proof assistants → Isabelle
- Fermat's last theorem
  - case n=4
  - case n=3
- Results of my research
  - space factor
  - time factor
- Discussion

Formalizing mathematics: what? why?

- Formalizing mathematics = expressing statements and proofs in a usually small and simple formal language with strict rules of grammar and unambiguous semantics
- Opportunities when combined with computer science:
  - -Proof-checking can be automated and can be more reliable
    (only the checking program has to be checked)
  - -Proofs are completely explicit and highly accessible
  - -Opportunities for better online collaboration
    (compare with wikipedia)
  - -Computer can do laborious parts
    (many case distinctions, calculations, ..)

Results in mechanised proving

- Four Colour Theorem (2004).

  By Georges Gonthier using Coq (60,000 lines).

- Prime Number Theorem (2005).

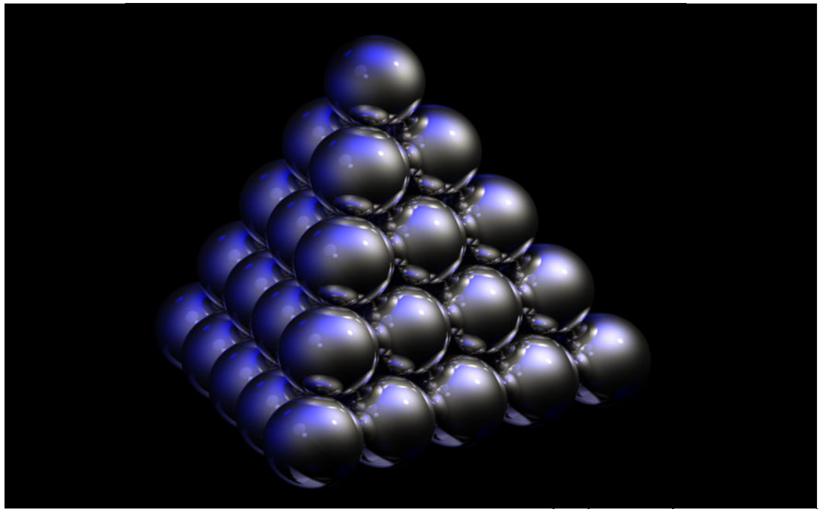
  By Jeremy Avigad + students using Isabelle (30,000 lines)

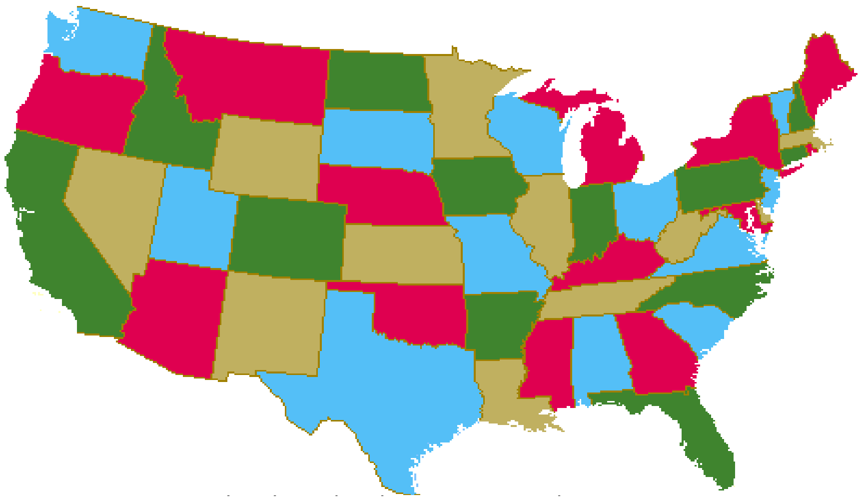
$$\lim_{x \to \infty} \frac{\pi(x)}{x/\log(x)} = 1$$

- Flyspeck Project (not finished).

  By several research groups / proof assistants.

- Large part of undergraduate mathematics





Proof assistants: Coq

```
Theorem main_thm: forall (n p : nat), n * n = double (p * p) ->  p = 0.
intros n; pattern n; apply lt_wf_ind; clear n.
intros n H p H0.
case (eq_nat_dec n 0); intros H1.
generalize H0; rewrite H1; case p; auto; intros; discriminate.
assert (H2: even n).
apply even_is_even_times_even.
apply double_even; rewrite H0; rewrite double_div2; auto.
assert (H3: even p).
apply even_is_even_times_even.
rewrite <- (double_inv (double (div2 n * div2 n)) (p * p)).
apply double_even; rewrite double_div2; auto.
rewrite main_thm_aux; auto.
assert (H4: div2 p = 0).
apply (H (div2 n)).
apply lt_div2; apply neq_O_lt; auto.
apply double_inv; apply double_inv; (repeat rewrite main_thm_aux); auto.
rewrite (even_double p); auto; rewrite H4; auto.
Qed.
```

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

Proof assistants: Mizar

```
theorem
  sqrt 2 is irrational
proof
 assume sqrt 2 is rational;
 then consider i being Integer, n being Nat such that
W1: n<>0 and
W2: sqrt 2=i/n and
W3: for i1 being Integer, n1 being Nat st n1<>0 & sqrt 2=i1/n1 holds n<=n1
      by RAT_1:25;
A5: i=sqrt 2*n by W1,XCMPLX_1:88,W2;
C: sqrt 2>=0 & n>0 by W1,NAT_1:19,SQUARE_1:93;
 then i>=0 by A5,REAL_2:121;
 then reconsider m = i as Nat by INT_1:16;
A6: m*m = n*n*(sqrt 2*sqrt 2) by A5
   .= n*n*(sqrt 2)^2 by SQUARE_1:def 3
   .= 2*(n*n) by SQUARE_1:def 4;
 then 2 divides m*m by NAT_1:def 3;
 then 2 divides m by INT_2:44,NEWTON:98;
 then consider m1 being Nat such that
W4: m=2*m1 by NAT_1:def 3;
 m1*m1*2*2 = m1*(m1*2)*2
     .= 2*(n*n) by W4,A6,XCMPLX_1:4;
 then 2*(m1*m1) = n*n by XCMPLX_1:5;
```

Proof assistants: HOL-Light

```
val lemma = Q.prove
(`!m n. (m**2 = 2 * n**2) ==> (m=0) /\ (n=0)`,
 completeInduct_on `m` THEN NTAC 2 STRIP_TAC THEN
  `?k. m = 2*k` by PROVE_TAC[EVEN_DOUBLE,EXP_2,EVEN_MULT,EVEN_EXISTS]
                THEN VAR_EQ_TAC THEN
  `?p. n = 2*p` by PROVE_TAC[EVEN_DOUBLE,EXP_2,EVEN_MULT,EVEN_EXISTS,EXP2_LEM]
                THEN VAR_EQ_TAC THEN
  `k**2 = 2*(p**2)` by PROVE_TAC [EXP2_LEM] THEN
  `(k=0) \/ k < 2*k` by numLib.ARITH_TAC
 THENL [FULL_SIMP_TAC arith_ss [EXP_2],
        PROVE_TAC [MULT_EQ_0, DECIDE (Term `~(2 = 0n)`)]]);
```

Proof assistants: Isabelle

```
proof (induct n rule: infinite-descent)
  fix n assume ¬ ?Q n
  then obtain m where n0: n>0 and mn: n^2 = 2*m^2 by auto
  hence 2 dvd n^2 by (simp add: dvd-def)
  hence 2 dvd n by (simp add: two-is-prime prime-dvd-power-two)
  then obtain p where pn: n = 2*p by (auto simp add: dvd-def)
  with mn have m^2 = 2*p^2 by (simp add: nat-number ring-simps)
  moreover have m < n
  proof (rule ccontr)
    assume ¬ m < n
    hence m-le-n: m^2 ≥ n^2 by (simp add: power-mono)
    from n0 have n^2 > 0 by auto
    with mn have n^2 > m^2 by auto
    with m-le-n show False by auto
  qed
  moreover have m > 0
  proof (rule ccontr, simp)
    assume m = 0
    with mn n0 show False by (auto simp add: power2-eq-square)
  qed
```

Isabelle characteristics

- Readable i/o
- Good documentation
- Several logics
- Two input formats (Isabelle & Isar)
- Isabelle's automation is good at:
  - Logical reasoning
  - Calculating with equalities, like

$$x^3 + y^3 = (x + y)(x^2 - xy + y^2)$$

  (at least if you know the commands)
  - *Not:* calculating with inequalities (perhaps no prover is), like

$$x \geq 1,\ y > 0 \Rightarrow xy \geq y$$

Agenda

- Aim of my research
- Proof assistants → Isabelle
- Fermat's last theorem
  - case n=4
  - case n=3
- Results of my research
  - space factor
  - time factor
- Discussion

Fermat's Last Theorem

Conjecture (Fermat, 1601-1665): If $n > 2$, then
$$x^n + y^n = z^n$$
does not have a solution for $x, y, z \in \mathbb{Z}_{>0}$.

- Fermat proved the case $n = 4$
  (using infinite descent and Euclid's construction of Pythagorean triples)
- Euler (1707-1783) proved the case $n = 3$
  (two versions, both contain errors)
- Wiles proved (in 1995) the case n is a prime ≥ 5
  (after several smaller and bigger repairs)

This completed the proof of FLT:
$$x^{pm} + y^{pm} = z^{pm} \iff (x^m)^p + (y^m)^p = (z^m)^p$$

The easy case n=4: informal proof

$$\exists x, y, z \in \mathbb{Z}_{\neq 0} : \; x^4 + y^4 = z^4$$

$$\Downarrow$$

$$\exists a, b, c \in \mathbb{Z}_{>0} : \; a^4 + b^4 = c^2, \;\; \gcd(a,b) = 1, \; a \text{ odd}$$

$$\Downarrow$$

$$\exists u, v \in \mathbb{Z}_{>0} : \; a^2 = u^2 - v^2, \; b^2 = 2uv, \; c = u^2 + v^2, \;\; \gcd(u,v) = 1$$

$$\Downarrow$$

$$\exists k, l \in \mathbb{Z}_{>0} : \; a = k^2 - l^2, \; v = 2kl, \; u = k^2 + l^2, \;\; \gcd(k,l) = 1$$

$$\text{and } \exists m \in \mathbb{Z}_{>0} : \; m = b/2, \; \text{hence } m^2 = uv/2 = kl(k^2 + l^2)$$

$$\Downarrow$$

$$\exists \alpha, \beta, \gamma \in \mathbb{Z}_{>0} : \; k = \alpha^2, \; l = \beta^2, \; k^2 + l^2 = \gamma^2$$

$$\text{hence } \gamma^2 = \alpha^4 + \beta^4, \;\; \gcd(\alpha, \beta) = 1, \; \alpha \text{ or } \beta \text{ odd}, \; \gamma < c$$

```
  qed
  -- "show the solution is smaller"
  moreover have "γ^2 < c^2"
  proof -
    from gamma2 klavu have "γ^2 ≤ abs u" by simp
    also have "... ≤ (abs u)^2" by (rule power2_ge_self)
    also have "... ≤ u^2" by (simp add: abs_power2_distrib)
    also have "... < u^2 + v^2"
    proof -
      from uv0 have v2non0: "0 ≠ v^2"
        by (auto simp add: power2_eq_square zero_le_power2)
      have "0 ≤ v^2" by (rule zero_le_power2)
      with v2non0 have "0 < v^2" by (auto simp add: less_int_def)
      thus ?thesis by auto
    qed
    also with uvabc have "... ≤ abs(c)" by auto
-1:**  Fermat4.thy       (Isar script XS:isabelle/s Scripting)--L505--80%--------------
```

```
proof (prove): step 234

fixed variables: a, b, c, u = u, v = v, k = k, l = l, m = m, α = α, β = β, γ = γ
prems:
  a ^ 4 + b ^ 4 = c²
  a * b * c ≠ 0
  a ∈ zOdd
  zgcd (a, b) = 1
  a² = u² - v² ∧ b² = 2 * u * v ∧ ¦c¦ = u² + v² ∧ zgcd (u, v) = 1
  a = k² - l² ∧ v = 2 * k * l ∧ ¦u¦ = k² + l²
  zgcd (k, l) = 1
  b = 2 * m
  ¦k¦ = α² ∧ ¦l¦ = β² ∧ ¦k² + l²¦ = γ²

using this:
  k² + l² = γ²
  a = k² - l² ∧ v = 2 * k * l ∧ ¦u¦ = k² + l²

goal (have, 1 subgoal):
 1. γ² ≤ ¦u¦
```

Example of Isabelle interaction

The tricky case n=3…

SATURDAY, JUNE 04, 2005

Euler's Mistake

For those who are not familiar with Fermat's Last

In Leonhard Euler's original proof for Ferm
anyone interested in seeing the details of t
genius like Euler was capable of getting th
Harold M. Edwards's book, *Fermat's Last*

The mistake came when Euler tried to pro
Euler attempted to prove this lemma using,

Lemma: Given that there exist p,q with the following properties:
(a) gcd(p,q)=1
(b) p,q have opposite parities
(c) $p^2 + 3q^2$ is a cube

(7) Which combined with step (1) gives us:

$$p^2 + 3q^2 = [a^3 - 9ab^2]^2 + 3(3a^2b - 3b^3)^2$$

(8) Which means that we could define a,b such that:

$p = a^3 - 9ab^2$.
$q = 3a^2b - 3b^3$.
gcd(a,b)=1 [since otherwise, any common factor would divide p ar

QED

IT

POSTED BY LARRY FREEMAN AT 9:00 PM

2002)

1998) **11 COMMENTS:**

Source: http://www.fermatslasttheorem.blogspot.com (25-11-'07)

Agenda

- Aim of my research
- Proof assistants → Isabelle
- Fermat's last theorem
  - case n=4
  - case n=3
- Results of my research
  - space factor
  - time factor
- Discussion

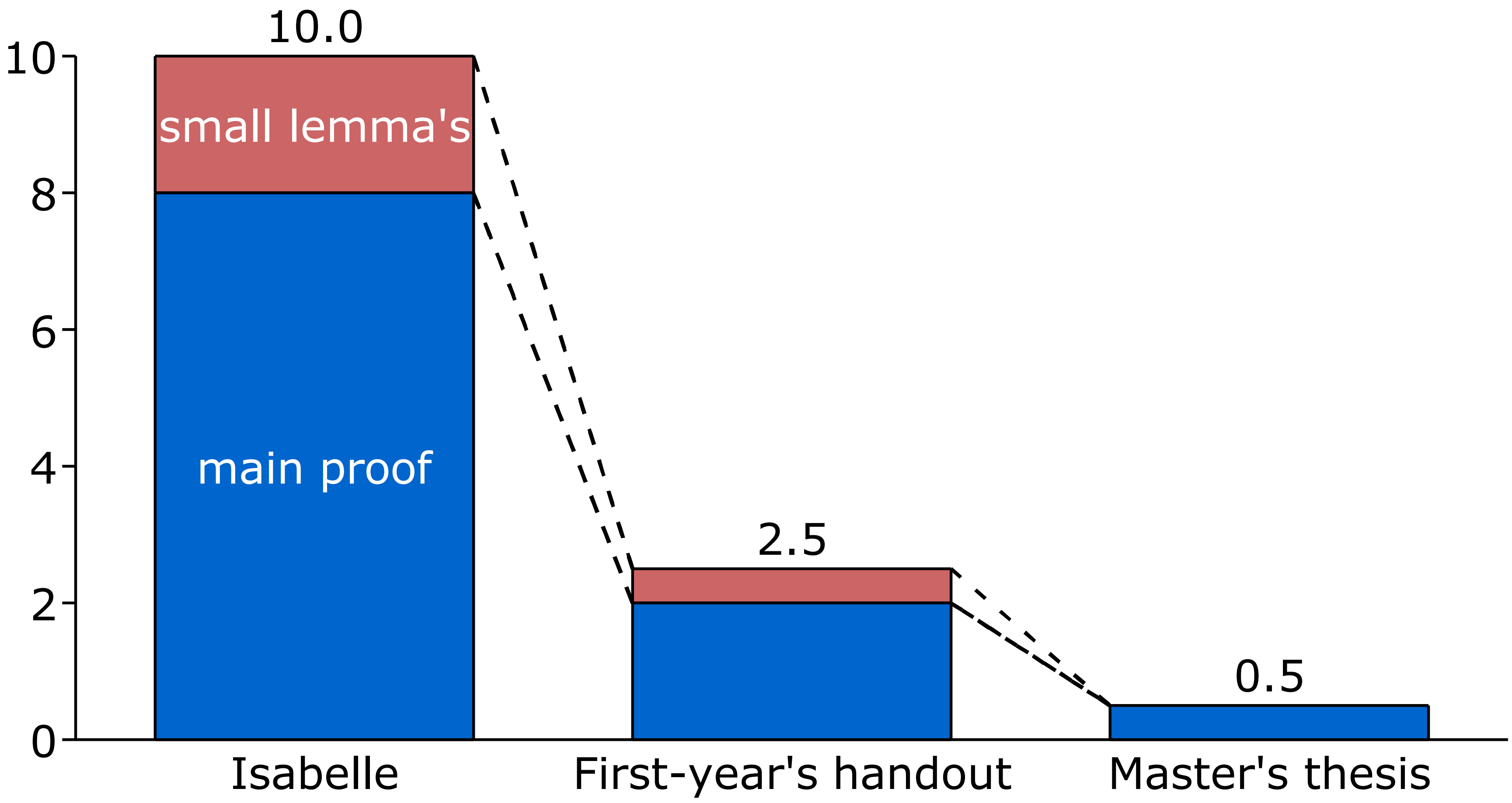
Results (1)

- Formal proof of FLT4: 1000 lines
- Formal proof of FLT3: 2500 (extra) lines
- Formal proof of Lagrange's Four-square Theorem *("any natural number can be written as the sum of 4 squares")*: 500 lines, 10 hours

How much 'more' work involves this, compared with 'informal' mathematics?

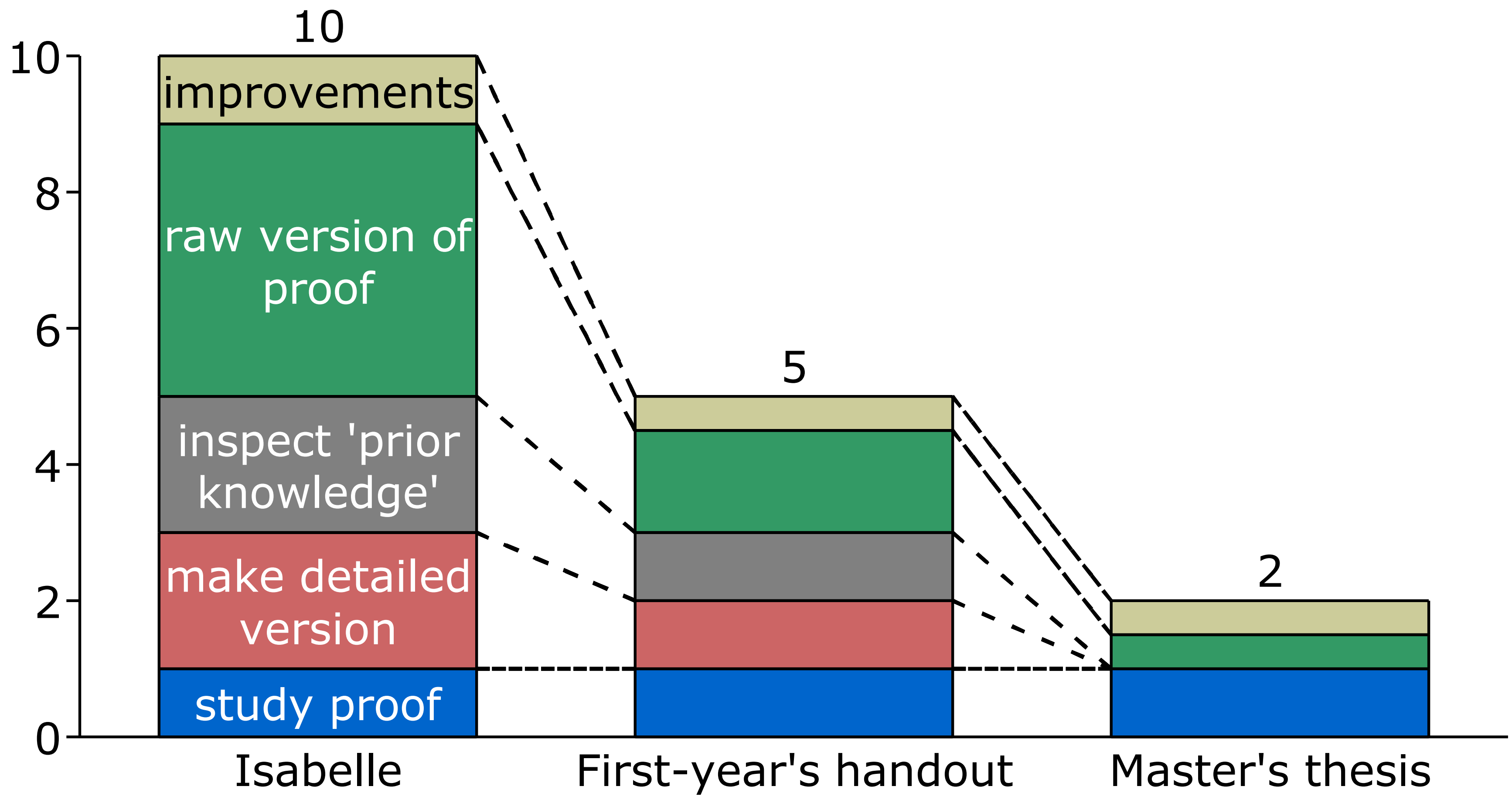
Space factor: 4 - 20

Pages required for a proof of
Lagrange's Four-square theorem



Time factor: 2 - 5

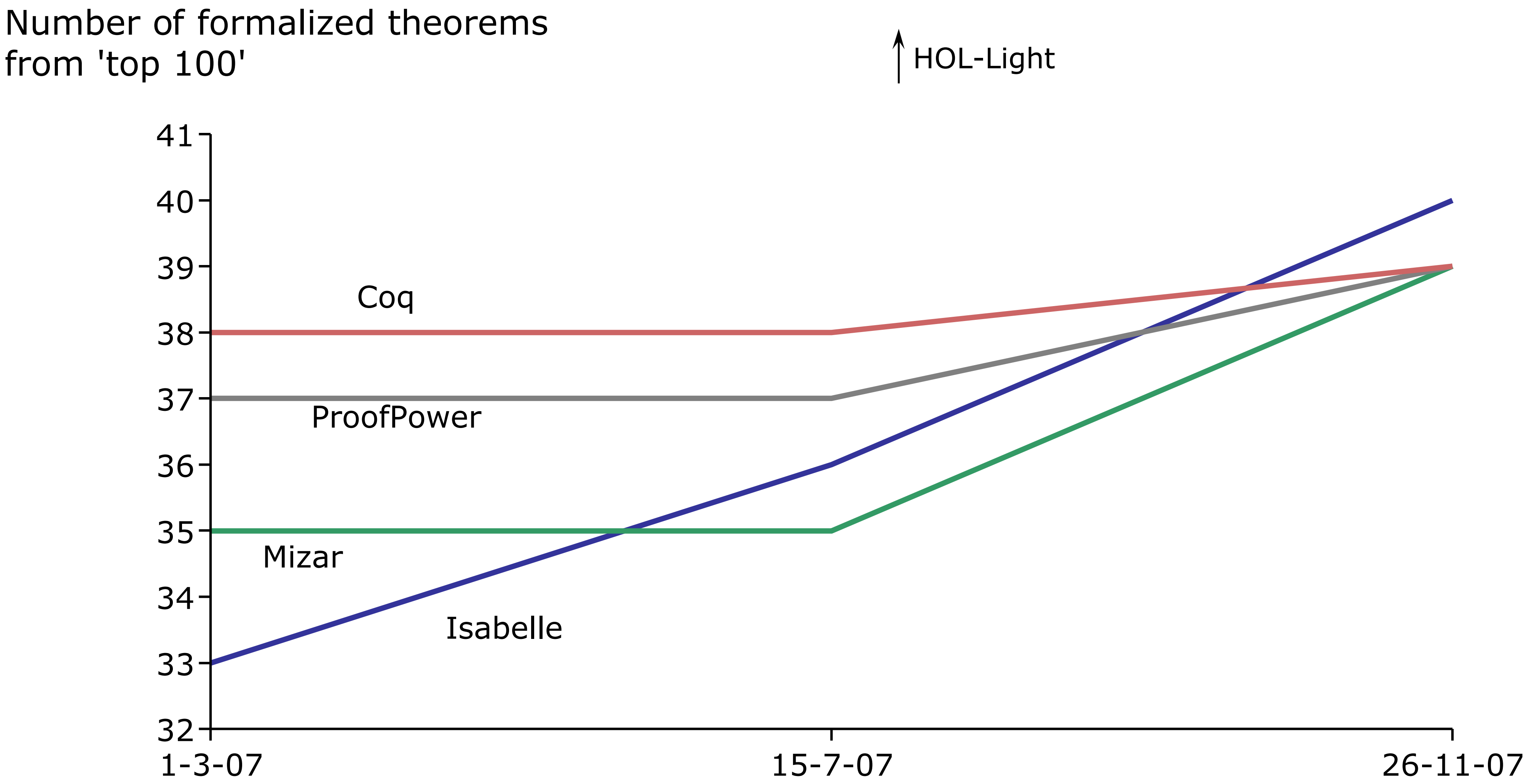
Hours required for producing a proof of
Lagrange’s Four-square theorem



Agenda

- Aim of my research
- Proof assistants → Isabelle
- Fermat's last theorem
  - case n=4
  - case n=3
- Results of my research
  - space factor
  - time factor
- Discussion

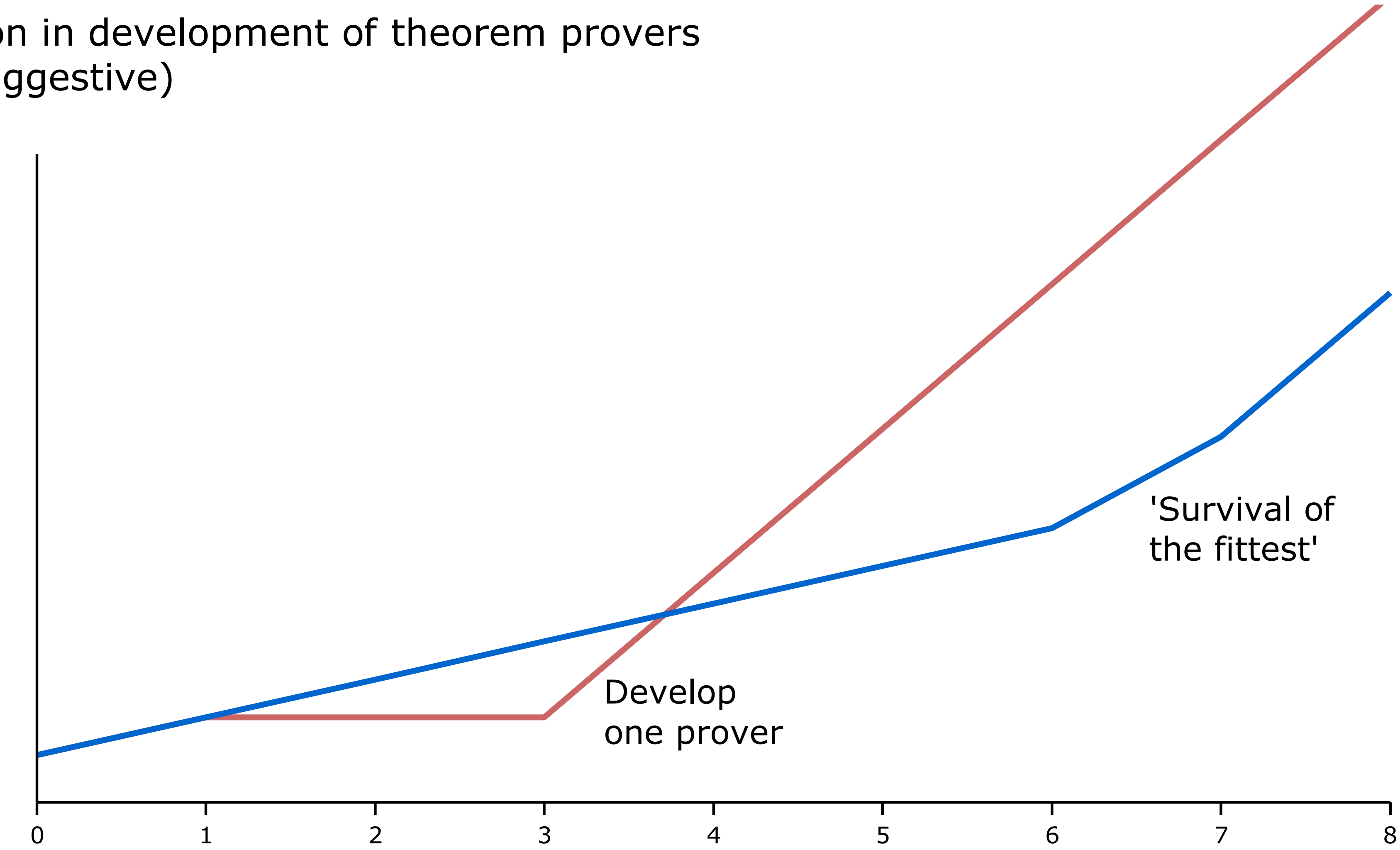
The battle is not over yet...



NB: HOL-Light omitted in graph
Source: Freek Wiedijk, http://www.cs.ru.nl/~freek/100

Collaborate now to speed up development

Progression in development of theorem provers
(a little suggestive)



Collaborate now to prevent doing work twice

- None of the current systems (Mizar, Isabelle, HOL, ProofPower, Coq) is acceptable as the QED system yet

  (F.Wiedijk, ‘The QED manifesto revisited’, http://mizar.org/trybulec65/8.pdf)

- You don’t want to formalize Wiles’ proof twice…

Discussion

- Mechanized theorem proving / proof verification…
  - is not that far away from the usual mathematical work (anymore)
  - will gain an important role in the daily life of mathematical research and education, within a few decades
  - is an accessible research field, for mathematicians as well as for computer scientists
  - requires immediate world wide collaboration
    - to speed up the development
    - to prevent doing proofs twice