SHE DOES ME GOOD
How Logic Doesn't Let You Down
(cracking the proof code)

João Marcos

LoLITA / DIMAp, UFRN, BR

Semana da Matemática UnB

Workshop de Matemática Aplicada

6 Feb 2015

How many of your proofs are correct?

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

How many of your proofs are correct?

How does **proof writing** compare to **software development**?
Some anecdotal evidence:

- on average, a programmer introduces 1.5 bugs per line while typing
- about one bug per hundred lines of computer code ships to market without detection
- one in each three math papers contain mistakes

How many of your proofs are correct?

How does **proof writing** compare to **software development**?
Some anecdotal evidence:

- on average, a programmer introduces 1.5 bugs per line while typing
- about one bug per hundred lines of computer code ships to market without detection
- one in each three math papers contain mistakes

Not a big issue? [Doron Zeilberger 1998, Opinion 91]

> "Most mathematical papers are leaves in the web of knowledge, that no one reads, or will ever use to prove something else."

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?
An optimistic note: (analogies by M. Feigenbaum)

- computers as 'bubble chambers'
- machines to help 'creating intuition'

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

Do *structured proofs* deliver the right amount
of precision ?

> "...the tiniest proof at the beginning of the Theory of Sets would already require several hundreds of signs for its complete formalization... formalized mathematics cannot in practice be written down in full... We shall therefore very quickly abandon formalized mathematics."
> — N. Bourbaki 1968

Indeed: [A. Matthias 2002]
Just to expand the definition of the number '1' fully
in terms of Bourbaki primitives requires over 4 trillion symbols.

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

Do *structured proofs* deliver the right amount
of precision, rigour ?

$*54{\cdot}43.\quad \vdash :. \alpha, \beta \in 1 . \supset : \alpha \cap \beta = \Lambda . \equiv . \alpha \cup \beta \in 2$

Dem.

$\vdash . *54{\cdot}26 . \supset \vdash :. \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . x \neq y .$

$[*51{\cdot}231] \qquad \equiv . \iota`x \cap \iota`y = \Lambda .$

$[*13{\cdot}12] \qquad \equiv . \alpha \cap \beta = \Lambda \qquad (1)$

$\vdash . (1) . *11{\cdot}11{\cdot}35 . \supset$

$\quad \vdash :. (\exists x, y) . \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . \alpha \cap \beta = \Lambda \qquad (2)$

$\vdash . (2) . *11{\cdot}54 . *52{\cdot}1 . \supset \vdash . \mathrm{Prop}$

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

✱54.43: "From this proposition it will follow, when arithmetical addition has been defined, that 1+1=2." —Volume I, 1st edition, page 379 (page 362 in 2nd edition; page 360 in abridged version). (The proof is actually completed in Volume II, 1st edition, page 86, accompanied by the comment, "The above proposition is occasionally useful.")

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

Do *structured proofs* deliver the right amount
of precision, rigour and beauty?

> "Dirichlet alone, not I, nor Cauchy, nor Gauss knows what a completely rigorous mathematical proof is. Rather we learn it first from him. When Gauss says that he has proved something, it is very clear; when Cauchy says it, one can wager as much pro as con; when Dirichlet says it, it is certain..."
> — Carl Jacobi, quoted by Schubring

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

Do *structured proofs* deliver the right amount
of precision, rigour and beauty?

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

Do *structured proofs* deliver the right amount
of precision, rigour and beauty?

Some criticisms on structured proofs: [L. Lamport 1993]

- “They are too complicated.”
- “They don’t explain why the proof works.”
- “A proof should be great literature.”

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

Do *structured proofs* deliver the right amount
of precision, rigour and beauty?

To be really convincing, shouldn't proofs be *human-surveyable*?

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

Do *structured proofs* deliver the right amount
of precision, rigour and beauty?

To be really convincing, shouldn't proofs be *human-surveyable*?
Yet recall, for instance:

- F. Almgren's 'Big Paper' in geometric measure theory
— the preprint is 1728 pages long, written for longer than a decade
- D. Gorenstein's announcement, in 1983, that the classification of finite simple groups had been completed
— a missing gap in the treatment of the class of 'quasithin' groups was not filled until 2001, with a 1,221-page proof by M. Aschbacher & S. Smith

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

Do *structured proofs* deliver the right amount
of precision, rigour and beauty?

To be really convincing, shouldn't proofs be *human-surveyable*?

How many of your proofs are correct?

How does **proof writing** compare to **software development**?

Does *experimental mathematics* give you just the worst of both worlds?

Do *structured proofs* deliver the right amount
of precision, rigour and beauty?

To be really convincing, shouldn't proofs be *human-surveyable*?

Are theoreticians in need of techniques of **proof engineering**?

Can computers assist you in doing math?

"[...] intellectual activity consists mainly of various kinds of search."
— A. Turing 1948 (report), *Intelligent Machinery* (report)

Can computers assist you in doing math?

"[...] intellectual activity consists mainly of various kinds of search."
— A. Turing 1948 (report), *Intelligent Machinery* (report)

Computer-assisted proofs:

A mathematical proof that has been
at least partially generated by computer.

Can computers assist you in doing math?

> "[...] intellectual activity consists mainly of various kinds of search."
> — A. Turing 1948 (report), *Intelligent Machinery* (report)

Computer-assisted proofs:

A mathematical proof that has been
at least partially generated by computer.

A computer may surely be useful!

- exploring mathematical phenomena
- searching for relevant information in databases of mathematical facts
- verifying correctness of proofs
- assisting the production of formally verified math
- discovering new theorems

Can computers assist you in doing math?

"[...] intellectual activity consists mainly of various kinds of search."
— A. Turing 1948 (report), *Intelligent Machinery* (report)

Computer-assisted proofs:

A mathematical proof that has been
at least partially generated by computer.

A computer may surely be useful!

Can computers assist you in doing math?

"[...] intellectual activity consists mainly of various kinds of search."
— A. Turing 1948 (report), *Intelligent Machinery* (report)

Computer-assisted proofs:

A mathematical proof that has been
at least partially generated by computer.

A computer may surely be useful!
What are the risks?
In **informal proof**, mistakes arise from: [J. Avigad & J. Harrison 2014]

- gaps in the reasoning
- appeal to faulty intuitions
- imprecise definitions
- misapplied background facts
- and fiddly special cases or side conditions the author failed to check.

How are more reliable: Computers or humans? [D. Mackenzie 2001]
What to learn from computer programmers?

Can computers assist you in doing math?

"[...] intellectual activity consists mainly of various kinds of search."
— A. Turing 1948 (report), *Intelligent Machinery* (report)

Computer-assisted proofs:

A mathematical proof that has been
at least partially generated by computer.

A computer may surely be useful!

What are the risks?

Calculemus

- **Exhaustive checking**: Connect-Four, Rubik's Cube, Four-Color Theorem
- **Model generation**: Tarski High School Algebra Problem
- **Proof Generation**: Robbins Algebras
- **Formal verification**: Flyspeck Project
- **Decidability**: Presburger Arithmetic, Gröbner basis algorithms, Theory of Real Closed Fields

Can computers assist you in doing math?

"[...] intellectual activity consists mainly of various kinds of search."
— A. Turing 1948 (report), *Intelligent Machinery* (report)

Computer-assisted proofs:

A mathematical proof that has been
at least partially generated by computer.

A computer may surely be useful!

What are the risks?

Calculemus

- **Exhaustive checking**: Connect-Four, Rubik's Cube, Four-Color Theorem
- **Model generation**: Tarski High School Algebra Problem
- **Proof Generation**: Robbins Algebras
- **Formal verification**: Flyspeck Project
- **Decidability**: Presburger Arithmetic, Gröbner basis algorithms, Theory of Real Closed Fields

Can computers assist you in doing math?

"[...] intellectual activity consists mainly of various kinds of search."
— A. Turing 1948 (report), *Intelligent Machinery* (report)

Computer-assisted proofs:

A mathematical proof that has been
at least partially generated by computer.

A computer may surely be useful!
What are the risks?
Calculemus

- **Exhaustive checking**: Connect-Four, Rubik's Cube, Four-Color Theorem
- **Model generation**: Tarski High School Algebra Problem
- **Proof Generation**: Robbins Algebras
- **Formal verification**: Flyspeck Project
- **Decidability**: Presburger Arithmetic, Gröbner basis algorithms, Theory of Real Closed Fields

Formal correctness assessed only modulo an underlying axiomatic framework.

Can computers assist you in doing math?

"[...] intellectual activity consists mainly of various kinds of search."
— A. Turing 1948 (report), *Intelligent Machinery* (report)

Computer-assisted proofs:

A mathematical proof that has been
at least partially generated by computer.

A computer may surely be useful!

What are the risks?

Calculemus

- **Exhaustive checking**: Connect-Four, Rubik's Cube, Four-Color Theorem
- **Model generation**: Tarski High School Algebra Problem
- **Proof Generation**: Robbins Algebras
- **Formal verification**: Flyspeck Project
- **Decidability**: Presburger Arithmetic, Gröbner basis algorithms, Theory of Real Closed Fields

Peeking into the future: A new role for referees!

Exhaustive checking: Four-Color Theorem

Exhaustive checking: Four-Color Theorem

It all starts with a puzzle by Francis Guthrie, in 1852.

Exhaustive checking: Four-Color Theorem

It all starts with a puzzle by Francis Guthrie, in 1852.

Milestones:

An influential wrong proof lingers for 11 years. [A. Kempe 1879]

Overall strategy: an *induction* on the reduction of map configurations (and their dual graphs)

Exhaustive checking: Four-Color Theorem

It all starts with a puzzle by Francis Guthrie, in 1852.

Milestones:

An influential wrong proof lingers for 11 years. [A. Kempe 1879]

Overall strategy: an *induction* on the reduction of map configurations (and their dual graphs)

[K. Appel and W. Haken 1976]

Suppose, by absurd, a map needed five colors.

Possible configurations are divided into 1,936 *minimal* such maps.

Show that each configuration can be reduced into a *smaller* configuration which also needs five colours.

Note: The **reduction** was made by a computer!

Exhaustive checking: Four-Color Theorem

It all starts with a puzzle by Francis Guthrie, in 1852.

Milestones:

An influential wrong proof lingers for 11 years. [A. Kempe 1879]

[K. Appel and W. Haken 1976]

Exhaustive checking: Four-Color Theorem

It all starts with a puzzle by Francis Guthrie, in 1852.

Milestones:

An influential wrong proof lingers for 11 years. [A. Kempe 1879]

[K. Appel and W. Haken 1976]

[N. Robertson, D. Sanders, P. Seymour and R. Thomas 1996]

Simpler proof involving *only* 633 configurations.

Exhaustive checking: Four-Color Theorem

It all starts with a puzzle by Francis Guthrie, in 1852.

Milestones:

An influential wrong proof lingers for 11 years. [A. Kempe 1879]

[K. Appel and W. Haken 1976]

[N. Robertson, D. Sanders, P. Seymour and R. Thomas 1996]

Exhaustive checking: Four-Color Theorem

It all starts with a puzzle by Francis Guthrie, in 1852.

Milestones:

An influential wrong proof lingers for 11 years. [A. Kempe 1879]

[K. Appel and W. Haken 1976]

[N. Robertson, D. Sanders, P. Seymour and R. Thomas 1996]

Georges Gonthier (*Microsoft Research Cambridge*) enters the scene.

Exhaustive checking: Four-Color Theorem

It all starts with a puzzle by Francis Guthrie, in 1852.

Milestones:

An influential wrong proof lingers for 11 years. [A. Kempe 1879]

[K. Appel and W. Haken 1976]

[N. Robertson, D. Sanders, P. Seymour and R. Thomas 1996]

Georges Gonthier (*Microsoft Research Cambridge*) enters the scene.
A **formal program proof** should be code that:

- describes *what* the machine should do
- and also *why* it should be doing it
  (i.e. contain a computer-checked proof of correctness)

Exhaustive checking: Four-Color Theorem

It all starts with a puzzle by Francis Guthrie, in 1852.

Milestones:

An influential wrong proof lingers for 11 years. [A. Kempe 1879]

[K. Appel and W. Haken 1976]

[N. Robertson, D. Sanders, P. Seymour and R. Thomas 1996]

Georges Gonthier (*Microsoft Research Cambridge*) enters the scene.

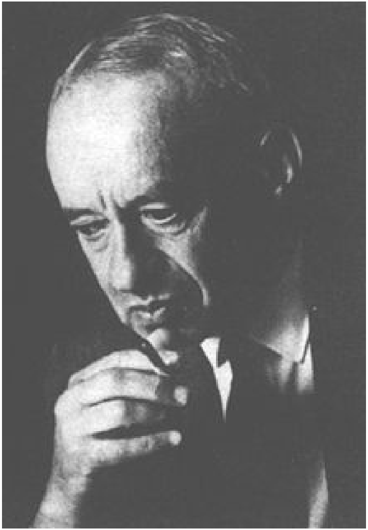
A **formal program proof** should be code that:

- describes *what* the machine should do
- and also *why* it should be doing it
  (i.e. contain a computer-checked proof of correctness)

`> Compilation succeeds.` [G. Gonthier 2005]

uses Coq

Model generation: Tarski's High School Identities



Model generation: Tarski's High School Identities

Consider the following **identities**:

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot 1 = x$
6. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

Model generation: Tarski's High School Identities

Consider the following **identities**:

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot 1 = x$
6. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

Facts:

- these **characterize** precisely the equational theory of $\widehat{\mathbb{N}} = \langle N, \overline{+}, \overline{\cdot}, \overline{1} \rangle$
- there is a **decision procedure** for this theory

Model generation: Tarski's High School Identities

Consider the following **identities** (**HSI**): [R. Dedekind 1888]

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot 1 = x$
6. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
7. $1^x = 1$
8. $x^1 = x$
9. $(x \cdot y)^z = x^z \cdot y^z$
10. $x^{(y+z)} = x^y \cdot x^z$
11. $(x^y)^z = x^{(y \cdot z)}$

Model generation: Tarski's High School Identities

Consider the following **identities** (**HSI**): [R. Dedekind 1888]

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot 1 = x$
6. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
7. $1^x = 1$
8. $x^1 = x$
9. $(x \cdot y)^z = x^z \cdot y^z$
10. $x^{(y+z)} = x^y \cdot x^z$
11. $(x^y)^z = x^{(y \cdot z)}$

A Computer Science perspective. Let

$$\# ::= \bigcirc \mid \#^{++} \mid +(\#,\#) \mid \cdot(\#,\#) \mid \uparrow(\#,\#)$$

Take $\bigcirc$ and $++$ as primitive, as in **PA**. [G. Peano 1889]
Then define, recursively:

[b(+)]	$+(x,\bigcirc) = x$	[b(·)]	$\cdot(x,\bigcirc) = \bigcirc$	[b(↑)]	$\uparrow(x,\bigcirc) = \bigcirc^{++}$
[r(+)]	$+(x,y^{++}) = (+(x,y))^{++}$	[r(·)]	$\cdot(x,y^{++}) = +(\cdot(x,y),x)$	[r(↑)]	$\uparrow(x,y^{++}) = \cdot(\uparrow(x,y),x)$

Model generation: Tarski's High School Identities

Consider the following **identities** (**HSI**): [R. Dedekind 1888]

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot 1 = x$
6. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
7. $1^x = 1$
8. $x^1 = x$
9. $(x \cdot y)^z = x^z \cdot y^z$
10. $x^{(y+z)} = x^y \cdot x^z$
11. $(x^y)^z = x^{(y \cdot z)}$

A Computer Science perspective. Let

$$\# ::= \bigcirc \mid \#^{++} \mid +(\#,\#) \mid \cdot(\#,\#) \mid \uparrow(\#,\#)$$

Take $\bigcirc$ and $++$ as primitive, as in **PA**. [G. Peano 1889]
Then define, recursively:

[b(+)]	$+(x, \bigcirc) = x$	[b(·)]	$\cdot(x, \bigcirc) = \bigcirc$	[b(↑)]	$\uparrow(x, \bigcirc) = \bigcirc^{++}$
[r(+)]	$+(x, y^{++}) = (+(x, y))^{++}$	[r(·)]	$\cdot(x, y^{++}) = +(\cdot(x, y), x)$	[r(↑)]	$\uparrow(x, y^{++}) = \cdot(\uparrow(x, y), x)$

Now *prove* the above identities by (structural) induction!

Model generation: Tarski's High School Identities

Consider the following **identities** (**HSI**): [R. Dedekind 1888]

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot 1 = x$
6. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
7. $1^x = 1$
8. $x^1 = x$
9. $(x \cdot y)^z = x^z \cdot y^z$
10. $x^{(y+z)} = x^y \cdot x^z$
11. $(x^y)^z = x^{(y \cdot z)}$

Model generation: Tarski's High School Identities

Consider the following **identities** (**HSI**): [R. Dedekind 1888]

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot 1 = x$
6. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
7. $1^x = 1$
8. $x^1 = x$
9. $(x \cdot y)^z = x^z \cdot y^z$
10. $x^{(y+z)} = x^y \cdot x^z$
11. $(x^y)^z = x^{(y \cdot z)}$

'Natural' questions: [A. Tarski 1969]

- do the above characterize the equational theory of $\mathbb{N} = \langle N, \overline{+}, \bar{\cdot}, \overline{\uparrow}, \overline{1} \rangle$?
- is there a decision procedure for this theory?

Model generation: Tarski's High School Identities

Consider the following **identities** (**HSI**): [R. Dedekind 1888]

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot 1 = x$
6. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
7. $1^x = 1$
8. $x^1 = x$
9. $(x \cdot y)^z = x^z \cdot y^z$
10. $x^{(y+z)} = x^y \cdot x^z$
11. $(x^y)^z = x^{(y \cdot z)}$

'Natural' questions:

- do the above characterize the equational theory of $\mathbb{N} = \langle N, \overline{+}, \bar{\cdot}, \overline{\uparrow}, \overline{1} \rangle$?
  NO!
- is there a decision procedure for this theory?
  YES! [A. Macintyre 1981]

Model generation: Tarski's High School Identities

Consider the following **identities** (**HSI**): [R. Dedekind 1888]

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot 1 = x$
6. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
7. $1^x = 1$
8. $x^1 = x$
9. $(x \cdot y)^z = x^z \cdot y^z$
10. $x^{(y+z)} = x^y \cdot x^z$
11. $(x^y)^z = x^{(y \cdot z)}$

Model generation: Tarski's High School Identities

Consider the following **identities** (**HSI**): [R. Dedekind 1888]

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot 1 = x$
6. $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
7. $1^x = 1$
8. $x^1 = x$
9. $(x \cdot y)^z = x^z \cdot y^z$
10. $x^{(y+z)} = x^y \cdot x^z$
11. $(x^y)^z = x^{(y \cdot z)}$

Here is an *unprovable true identity*, $W(x, y)$: [A. Wilkie 1980–81]

$$(A^y + B^y)^x \cdot (C^x + D^x)^y = (A^x + B^x)^y \cdot (C^y + D^y)^x$$

where $A = 1 + x$, $B = 1 + x + x \cdot x$, $C = 1 + x \cdot x \cdot x$, $D = 1 + x \cdot x + x \cdot x \cdot x \cdot x$.

Model generation: Tarski's High School Identities

Here is an *unprovable true identity*, $W(x, y)$: [A. Wilkie 1980–81]

$$(A^y + B^y)^x \cdot (C^x + D^x)^y = (A^x + B^x)^y \cdot (C^y + D^y)^x$$

where $A = 1 + x$, $B = 1 + x + x \cdot x$, $C = 1 + x \cdot x \cdot x$, $D = 1 + x \cdot x + x \cdot x \cdot x \cdot x$.

Exercise:

Let $E = 1 - x + x \cdot x$ and check that $W(x, y)$ is true by factoring:

$$C \text{ as } A \cdot E, \text{ and } D \text{ as } B \cdot E$$

The trouble with **HSI**:

Inability to manipulate polynomials with negative coefficients!

Model generation: Tarski's High School Identities

Here is an *unprovable true identity*, $W(x, y)$: [A. Wilkie 1980–81]

$$(A^y + B^y)^x \cdot (C^x + D^x)^y = (A^x + B^x)^y \cdot (C^y + D^y)^x$$

where $A = 1 + x$, $B = 1 + x + x \cdot x$, $C = 1 + x \cdot x \cdot x$, $D = 1 + x \cdot x + x \cdot x \cdot x \cdot x$.

Model generation: Tarski's High School Identities

Here is an *unprovable true identity*, $W(x, y)$: [A. Wilkie 1980–81]

$$(A^y + B^y)^x \cdot (C^x + D^x)^y = (A^x + B^x)^y \cdot (C^y + D^y)^x$$

where $A = 1 + x$, $B = 1 + x + x \cdot x$, $C = 1 + x \cdot x \cdot x$, $D = 1 + x \cdot x + x \cdot x \cdot x \cdot x$.

On the *nonderivability* of the exotic identity $W(x, y)$:
Induction on the length of a supposed derivation of $W(x, y)$ from **HSI**.
(*proof-theoretical* argument)

Model generation: Tarski's High School Identities

Here is an *unprovable true identity*, $W(x, y)$: [A. Wilkie 1980–81]

$$(A^y + B^y)^x \cdot (C^x + D^x)^y = (A^x + B^x)^y \cdot (C^y + D^y)^x$$

where $A = 1 + x$, $B = 1 + x + x \cdot x$, $C = 1 + x \cdot x \cdot x$, $D = 1 + x \cdot x + x \cdot x \cdot x \cdot x$.

On the *nonderivability* of the exotic identity $W(x, y)$:
Induction on the length of a supposed derivation of $W(x, y)$ from **HSI**.
(*proof-theoretical* argument)

Finding *actual counterexamples* to $W(x, y)$: (*model-theoretical* argument)

Model generation: Tarski's High School Identities

Here is an *unprovable true identity*, $W(x, y)$: [A. Wilkie 1980–81]

$$(A^y + B^y)^x \cdot (C^x + D^x)^y = (A^x + B^x)^y \cdot (C^y + D^y)^x$$

where $A = 1 + x$, $B = 1 + x + x \cdot x$, $C = 1 + x \cdot x \cdot x$, $D = 1 + x \cdot x + x \cdot x \cdot x \cdot x$.

On the *nonderivability* of the exotic identity $W(x, y)$:
Induction on the length of a supposed derivation of $W(x, y)$ from **HSI**.
(*proof-theoretical* argument)

Finding *actual counterexamples* to $W(x, y)$: (*model-theoretical* argument)
From 59 elements in [R. Gurevič 1985]
to 12 elements on [S. Burris & K. Yeats 2001],
but with at least 11 elements [J. Zhang 2005].

(note intensive use of computers!)

Model generation: Tarski's High School Identities

Here is an *unprovable true identity*, $W(x,y)$: [A. Wilkie 1980–81]

$$(A^y + B^y)^x \cdot (C^x + D^x)^y = (A^x + B^x)^y \cdot (C^y + D^y)^x$$

where $A = 1 + x$, $B = 1 + x + x \cdot x$, $C = 1 + x \cdot x \cdot x$, $D = 1 + x \cdot x + x \cdot x \cdot x \cdot x$.

On the *nonderivability* of the exotic identity $W(x,y)$:
Induction on the length of a supposed derivation of $W(x,y)$ from **HSI**.
(*proof-theoretical* argument)

Finding *actual counterexamples* to $W(x,y)$: (*model-theoretical* argument)
From 59 elements in [R. Gurevič 1985]
to 12 elements on [S. Burris & K. Yeats 2001],
but with at least 11 elements [J. Zhang 2005].
(note intensive use of computers!)
Further:
Non-finite axiomatizability of $\mathbb{N} = \langle N, \overline{+}, \bar{\cdot}, \overline{\uparrow}, \overline{1} \rangle$ [R. Gurevič 1990]

Model generation: Tarski's High School Identities

Here is an *unprovable true identity*, $W(x,y)$: [A. Wilkie 1980–81]

$$(A^y + B^y)^x \cdot (C^x + D^x)^y = (A^x + B^x)^y \cdot (C^y + D^y)^x$$

where $A = 1 + x$, $B = 1 + x + x \cdot x$, $C = 1 + x \cdot x \cdot x$, $D = 1 + x \cdot x + x \cdot x \cdot x \cdot x$.

On the *nonderivability* of the exotic identity $W(x,y)$:
Induction on the length of a supposed derivation of $W(x,y)$ from **HSI**.
(*proof-theoretical* argument)

Finding *actual counterexamples* to $W(x,y)$: (*model-theoretical* argument)
From 59 elements in [R. Gurevič 1985]
to 12 elements on [S. Burris & K. Yeats 2001],
but with at least 11 elements [J. Zhang 2005].
(note intensive use of computers!)
Further:
Non-finite axiomatizability of $\mathbb{N} = \langle N, \overline{+}, \bar{\cdot}, \bar{\uparrow}, \bar{1} \rangle$ [R. Gurevič 1990]

(*bonus:* connections to the theory of type isomorphisms in lambda calculi)

Proof generation: Robbins Algebras are Boolean

Proof generation: Robbins Algebras are Boolean

You know, of course, what a **Boolean Algebra** is!

A complemented distributive lattice $\langle B, \sqcap, \sqcup, - \rangle$.

Proof generation: Robbins Algebras are Boolean

You know, of course, what a **Boolean Algebra** is!

A complemented distributive lattice $\langle B, \sqcap, \sqcup, - \rangle$.

But can you really *recognize* one when you see it?

Proof generation: Robbins Algebras are Boolean

You know, of course, what a **Boolean Algebra** is!

A complemented distributive lattice $\langle B, \sqcap, \sqcup, - \rangle$.

But can you really *recognize* one when you see it?

Here is a *parsimonial* axiomatization of Boolean Algebras:
Assume $\sqcup$ commutative and associative, and add
the **Huntington identity**:

$$(\forall x, y)\ -(-x \sqcup y) \sqcup -(-x \sqcup -y) = x$$

[E. V. Huntington 1933]

Proof generation: Robbins Algebras are Boolean

You know, of course, what a **Boolean Algebra** is!

A complemented distributive lattice $\langle B, \sqcap, \sqcup, - \rangle$.

But can you really *recognize* one when you see it?

Here is a *parsimonial* axiomatization of Boolean Algebras:
Assume $\sqcup$ commutative and associative, and add
the **Huntington identity**:

$$(\forall x, y)\ -(-x \sqcup y) \sqcup -(-x \sqcup -y) = x$$

[E. V. Huntington 1933]

What if we added, *instead*, the following 'equivalent' **Robbins identity**?

$$(\forall x, y)\ -(-(x \sqcup y) \sqcup -(x \sqcup -y)) = x$$

[H. Robbins ±1933]

Proof generation: Robbins Algebras are Boolean

Robbins Algebras are born! *Specification:*

$\sqcup$ is commutative

$\sqcup$ is associative

$(\forall x, y)\ -(-(x \sqcup y) \sqcup -(x \sqcup -y)) = x$

Proof generation: Robbins Algebras are Boolean

Robbins Algebras are born! *Specification:*

$\sqcup$ is commutative

$\sqcup$ is associative

$(\forall x, y)\ -(-(x \sqcup y) \sqcup -(x \sqcup -y)) = x$

A cute benchmark problem for provers: *Are Robbins Algebras Boolean*?

Human Provers first!

Some sufficient conditions: [S. Winker 1990, 1992]

$(\forall x)\ --x = x$

$(\exists y)(\forall x)\ x \sqcup y = x$

$(\forall x)\ x \sqcup x = x$

$(\exists x, y)\ x \sqcup y = x$ **[first Winker condition]**

$(\exists x, y)\ -(x \sqcup y) = -x$ **[second Winker condition]**

Proof generation: Robbins Algebras are Boolean

Robbins Algebras are born! *Specification:*

$\sqcup$ is commutative

$\sqcup$ is associative

$(\forall x, y)\ -(-(x \sqcup y) \sqcup -(x \sqcup -y)) = x$

A cute benchmark problem for provers: *Are Robbins Algebras Boolean*?

Human Provers first!

Some sufficient conditions: [S. Winker 1990, 1992]

$(\forall x)\ --x = x$

$(\exists y)(\forall x)\ x \sqcup y = x$

$(\forall x)\ x \sqcup x = x$

$(\exists x, y)\ x \sqcup y = x$ **[first Winker condition]**

$(\exists x, y)\ -(x \sqcup y) = -x$ **[second Winker condition]**

Automated Provers enter the scene! [W. McCune 1997]

What if we can *prove a contradiction* by adding to Robbins Algebras the negation of the second Winker condition?

Proof generation: Robbins Algebras are Boolean

Robbins Algebras are born! *Specification:*

$\sqcup$ is commutative

$\sqcup$ is associative

$(\forall x, y)\ -(-(x \sqcup y) \sqcup -(x \sqcup -y)) = x$

A cute benchmark problem for provers: *Are Robbins Algebras Boolean*?

$(\exists x, y)\ -(x \sqcup y) = -x$ **[second Winker condition]**

Automated Provers enter the scene! [W. McCune 1997]

What if we can *prove a contradiction* by adding to Robbins Algebras the negation of the second Winker condition?

The *Argonne National Laboratory* hits the news:

Proof generation: Robbins Algebras are Boolean

Robbins Algebras are born! *Specification:*

$\sqcup$ is commutative

$\sqcup$ is associative

$(\forall x, y)\ -(-(x \sqcup y) \sqcup -(x \sqcup -y)) = x$

A cute benchmark problem for provers: *Are Robbins Algebras Boolean*?

$(\exists x, y)\ -(x \sqcup y) = -x$ **[second Winker condition]**

Automated Provers enter the scene! [W. McCune 1997]

What if we can *prove a contradiction* by adding to Robbins Algebras the negation of the second Winker condition?

The *Argonne National Laboratory* hits the news:

> "The successful search took about 8 days on an RS/6000 processor and used about 30 megabytes of memory. (For those who have the EQP preprint, the search used basic paramodulation with the super0 restriction on AC unifiers, the pair algorithm with ratio 1, and max-weight 70.)"

McCune used the *automated theorem provers* **EQP** and **Otter**.

Proof generation: Robbins Algebras are Boolean

Robbins Algebras are born! *Specification:*

$\sqcup$ is commutative

$\sqcup$ is associative

$(\forall x, y)\ -(-(x \sqcup y) \sqcup -(x \sqcup -y)) = x$

A cute benchmark problem for provers: *Are Robbins Algebras Boolean*?

$(\exists x, y)\ -(x \sqcup y) = -x$ **[second Winker condition]**

Automated Provers enter the scene! [W. McCune 1997]

What if we can *prove a contradiction* by adding to Robbins Algebras the negation of the second Winker condition?

Computers heavily used in:

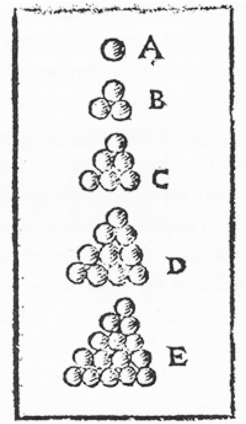
- *finding* the proof
- *parsing* the proof
- *refining* the proof
- *checking* the proof

Formal verification: Kepler Conjecture

Formal verification: Kepler Conjecture

The problem: **Packing spheres**

“No arrangement of equally sized spheres filling space has a greater average density than that of the cubic close packing (face-centered cubic) and hexagonal close packing arrangements.”



Conjecture: density $\frac{\pi}{\sqrt{18}}$ ($\approx 74\%$)

Formal verification: Kepler Conjecture

The problem: **Packing spheres**

Conjecture: density $\frac{\pi}{\sqrt{18}}$ ($\approx 74\%$)

Its **history**, in brief:

- piling cannonballs [1606]: Sir Walter Raleigh & Thomas Harriot
- J. Kepler [1611]: "Strena seu de Nive Sexangula"
- C. F. Gauss [1831]: solution checked for regular lattices
- A. Thue [1890]: two-dimensional analog, density $\frac{\pi}{\sqrt{12}}$ ($\approx 91\%$)
- Hilbert's 18th problem [1900]
- L. Fejes Tóth [1953]: give me enough computational power!
- Wu-Yi Hsiang [1993;2001]: an incomplete geometrical proof...
- Thomas Hales [1998-2005;2006;2014] & Samuel P. Ferguson: proofs with *computational flavor*

Formal verification: Kepler Conjecture

The problem: **Packing spheres**

Conjecture: density $\frac{\pi}{\sqrt{18}}$ ($\approx 74\%$)

Formal verification: Kepler Conjecture

The problem: **Packing spheres**

Conjecture: density $\frac{\pi}{\sqrt{18}}$ ($\approx 74\%$)

On the **Hales-Ferguson** proof:

- 1990s: *Annals of Mathematics* starts to accept computer proofs
- 1998: invited Hales to submit proof
  (300 pages of mathematical argument
  + 40K lines of computer code and 3Gb of data)
- Jan 1999: panel of 12 experts led by Gabor Fejes Tóth,
  conference at IAS Princeton
- 4 years later... **"We're 99% certain it is correct."**

  "The news from the referees is bad, from my perspective. They have not been able to certify the correctness of the proof, and will not be able to certify it in the future, because they have run out of energy to devote to the problem. This is not what I had hoped for..."

  — Robert MacPherson, editor of *Annals*

Formal verification: Kepler Conjecture

The problem: **Packing spheres**

Conjecture: density $\frac{\pi}{\sqrt{18}}$ ($\approx 74\%$)

On the **Hales-Ferguson** proof:

- 1990s: *Annals of Mathematics* starts to accept computer proofs
- 1998: invited Hales to submit proof
  (300 pages of mathematical argument
  + 40K lines of computer code and 3Gb of data)
- Jan 1999: panel of 12 experts led by Gabor Fejes Tóth, conference at IAS Princeton
- 4 years later... **"We're 99% certain it is correct."**
- 100-page mathematical kernel the paper published in 2004 at *Annals*
- computational part published elsewhere [2006]

Formal verification: Kepler Conjecture

The problem: **Packing spheres**

Conjecture: density $\frac{\pi}{\sqrt{18}}$ ($\approx 74\%$)

Formal verification: Kepler Conjecture

The problem: **Packing spheres**

Conjecture: density $\frac{\pi}{\sqrt{18}}$ ($\approx 74\%$)

The **aftermath**:

- *Annals* will no longer try to fully verify correctness of 'math-code'
- Hales gets a position at Pitt
- Hales wins the 2006 AMS Robbins Prize
- [Hales & Ferguson 2006] wins the Fulkerson Prize, in 2009

Formal verification: Kepler Conjecture

The problem: **Packing spheres**

Conjecture: density $\frac{\pi}{\sqrt{18}}$ ($\approx 74\%$)

Formal verification: Kepler Conjecture

The problem: **Packing spheres**

Conjecture: density $\frac{\pi}{\sqrt{18}}$ ($\approx 74\%$)

The ambitious **FlysPecK** Project starts in 2003.

Estimated to take 20 work-years. Completed in 2014.

```
From: <HALES@pitt.edu>
Date: Sun, Aug 10, 2014 at 4:26 PM
Subject: Flyspeck project completion
To: Thomas Hales <hales@pitt.edu>

We are pleased to announce the completion of the Flyspeck project,
which has constructed a formal proof of the Kepler conjecture. The
Kepler conjecture asserts that no packing of congruent balls in
Euclidean 3-space has density greater than the face-centered cubic
packing. It is the oldest problem in discrete geometry. The proof of
the Kepler conjecture was first obtained by Ferguson and Hales in
1998. The proof relies on about 300 pages of text and on a large
number of computer calculations.

The formalization project covers both the text portion of the proof
and the computer calculations. The blueprint for the project appears
in the book "Dense Sphere Packings," published by Cambridge University
Press. The formal proof takes the same general approach as the
original proof, with modifications in the geometric partition of space
that have been suggested by Marchal.
```

Formal verification: Kepler Conjecture

The problem: **Packing spheres**

Conjecture: density $\frac{\pi}{\sqrt{18}}$ ($\approx 74\%$)

The ambitious **FlysPecK** Project starts in 2003.

Estimated to take 20 work-years. Completed in 2014.

Three essential uses of computation: [J. Avigad & J. Harrison 2014]

- enumerating a class of combinatorial structures called "tame hypermaps"

  uses Isabelle

- using linear-programming methods to establish bounds on a large number of systems of linear constraints
- using interval methods to verify approximately 1,000 nonlinear inequalities that arise in the proof

  use HOL Light

Buggy proofs: On a major screwup by Kurt Gödel

Buggy proofs: On a major screwup by Kurt Gödel

Definition of *screwup*: (*apud* Mark Dominus, in his blog)
A purported proof of a *false* statement that *remains undetected* for a long period, and is actually *built upon* by others as if it were true.

Buggy proofs: On a major screwup by Kurt Gödel

Definition of *screwup*: (*apud* Mark Dominus, in his blog)
A purported proof of a *false* statement that *remains undetected* for a long period, and is actually *built upon* by others as if it were true.

Buggy software vs. buggy theorems.

Buggy proofs: On a major screwup by Kurt Gödel

Definition of *screwup*: (*apud* Mark Dominus, in his blog)
A purported proof of a *false* statement that *remains undetected*
for a long period, and is actually *built upon* by others as if it were true.

Here is a correct (yet surprising) result: [K. Gödel 1933]
The class of sentences of the form

$$\exists^*\forall^n\exists^*\varphi$$

where φ is quantifier-free, is *decidable* if (and only if) $n \leq 2$

Buggy proofs: On a major screwup by Kurt Gödel

Definition of *screwup*: (*apud* Mark Dominus, in his blog)
A purported proof of a *false* statement that *remains undetected*
for a long period, and is actually *built upon* by others as if it were true.

Here is a correct (yet surprising) result: [K. Gödel 1933]
The class of sentences of the form

$$\exists^*\forall^n\exists^*\varphi$$

where φ is quantifier-free, is *decidable* if (and only if) $n \leq 2$,
with an incorrect coda:

> "Zum Schluß möchte ich noch bemerken, daß sich Satz I auch für Formeln, welche das =-Zeichen enthalten, nach demselben Verfahren beweisen läßt."

Buggy proofs: On a major screwup by Kurt Gödel

Definition of *screwup*: (*apud* Mark Dominus, in his blog)
A purported proof of a *false* statement that *remains undetected*
for a long period, and is actually *built upon* by others as if it were true.

Here is a correct (yet surprising) result: [K. Gödel 1933]
The class of sentences of the form

$$\exists^*\forall^n\exists^*\varphi$$

where φ is quantifier-free, is *decidable* if (and only if) $n \leq 2$,
with an incorrect coda:

> "Zum Schluß möchte ich noch bemerken, daß sich Satz I auch für Formeln, welche das =-Zeichen enthalten, nach demselben Verfahren beweisen läßt."

Indeed...

Buggy proofs: On a major screwup by Kurt Gödel

Definition of *screwup*: (*apud* Mark Dominus, in his blog)
A purported proof of a *false* statement that *remains undetected*
for a long period, and is actually *built upon* by others as if it were true.

Here is a correct (yet surprising) result: [K. Gödel 1933]
The class of sentences of the form

$$\exists^*\forall^n\exists^*\varphi$$

where φ is quantifier-free, is *decidable* if (and only if) $n \leq 2$,
with an incorrect coda:

> "Zum Schluß möchte ich noch bemerken, daß sich Satz I auch für Formeln, welche das $=$-Zeichen enthalten, nach demselben Verfahren beweisen läßt."

Indeed...
... the proof is *not extensible* to the language with '$=$' [S. Aanderaa 1960s]

Buggy proofs: On a major screwup by Kurt Gödel

Definition of *screwup*: (*apud* Mark Dominus, in his blog)
A purported proof of a *false* statement that *remains undetected* for a long period, and is actually *built upon* by others as if it were true.

Here is a correct (yet surprising) result: [K. Gödel 1933]
The class of sentences of the form

$$\exists^*\forall^n\exists^*\varphi$$

where φ is quantifier-free, is *decidable* if (and only if) $n \leq 2$,
with an incorrect coda:

> "Zum Schluß möchte ich noch bemerken, daß sich Satz I auch für Formeln, welche das =-Zeichen enthalten, nach demselben Verfahren beweisen läßt."

Indeed...
... the proof is *not extensible* to the language with '$=$' [S. Aanderaa 1960s]
... and as a matter of fact the assertion is *false*! [W. Goldfarb 1983]

Buggy proofs: On a major screwup by Kurt Gödel

Definition of *screwup*: (*apud* Mark Dominus, in his blog)
A purported proof of a *false* statement that *remains undetected*
for a long period, and is actually *built upon* by others as if it were true.

Here is a correct (yet surprising) result: [K. Gödel 1933]
The class of sentences of the form

$$\exists^*\forall^n\exists^*\varphi$$

where φ is quantifier-free, is *decidable* if (and only if) $n \leq 2$,
with an incorrect coda:

> "Zum Schluß möchte ich noch bemerken, daß sich Satz I auch für Formeln, welche das $=$-Zeichen enthalten, nach demselben Verfahren beweisen läßt."

Indeed...

... the proof is *not extensible* to the language with '$=$' [S. Aanderaa 1960s]
... and as a matter of fact the assertion is *false*! [W. Goldfarb 1983]

The Grand Challenge: **Will YOU fare better?**