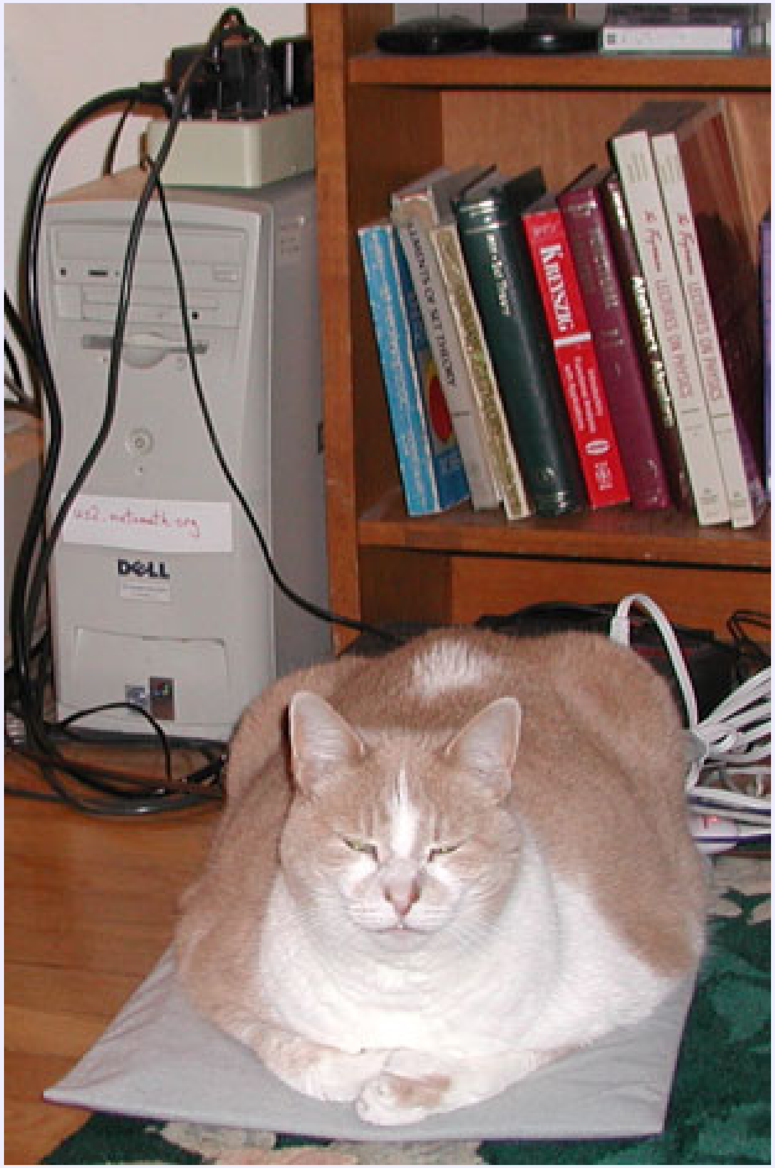
The Metamath Proof Language



Metamath development server

Norman Megill

May 9, 2014

Overview of Metamath

- Very simple language: substitution is the only basic rule

- Very small verifier ($\approx$300 lines code)

- Fast proof verification (6 sec for $\approx$18000 proofs)

- All axioms (including logic) are specified by user

- Formal proofs are complete and transparent, with no hidden implicit steps

Goals

Simplest possible framework that can express and verify (essentially) all of mathematics with absolute rigor

Permanent archive of hand-crafted formal proofs

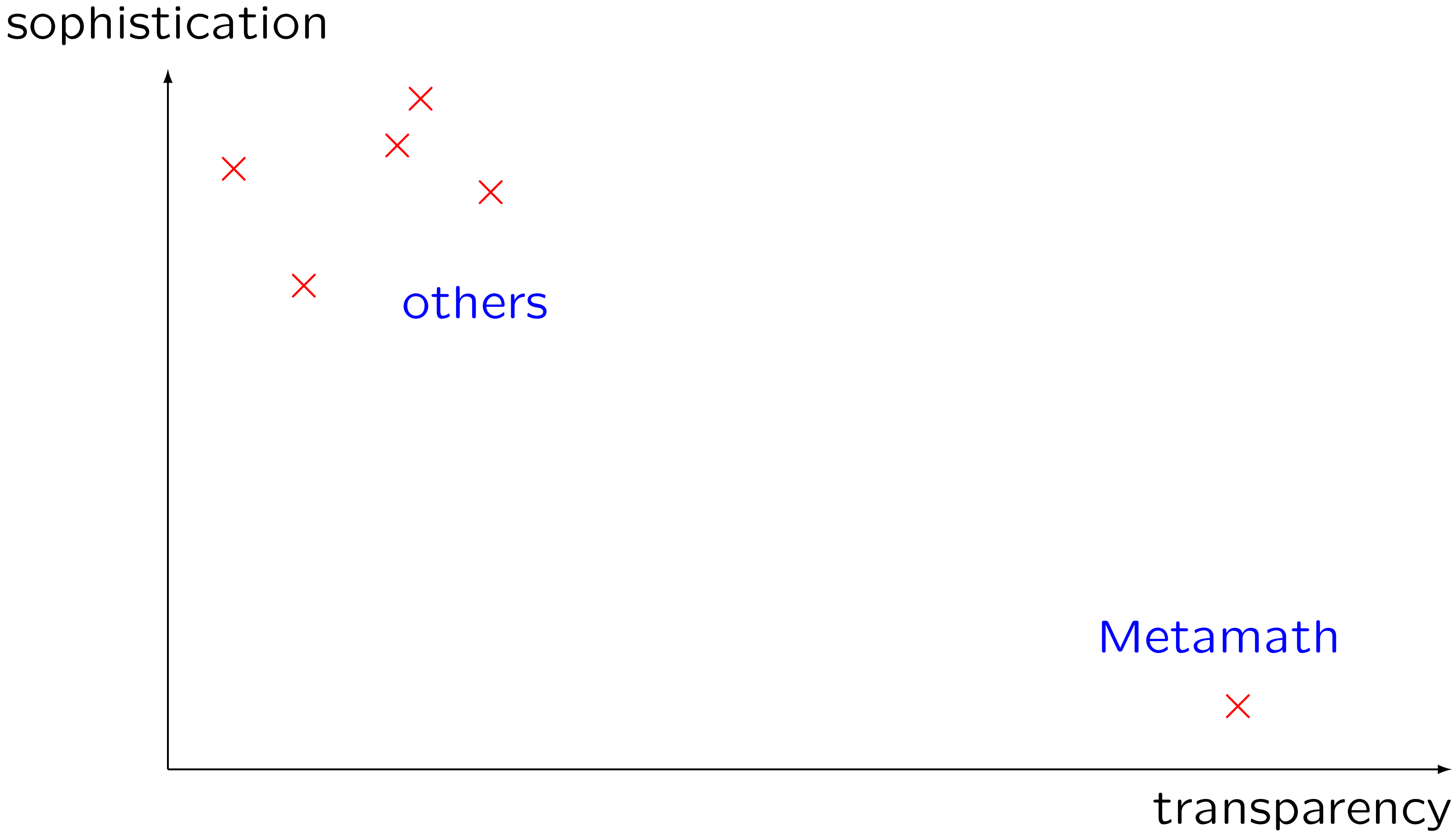
Elimination of uncertainty of proof correctness

Exposure of missing steps in informal proofs to any level of detail desired

Non-goals (at this time)

Automated theorem proving

Practical proof-finding assistant for working mathematicians



(Ficticious conceptual chart)

Contributors

David Abernethy
Stefan Allan
Juha Arpiainen
Jonathan Ben-Naim
Gregory Bush
Mario Carneiro
Paul Chapman
Scott Fenton
Jeffrey Hankins
Anthony Hart
David Harvey
Jeremy Henty
Jeff Hoffman
Szymon Jaroszewicz
Wolf Lammen
Gérard Lang
Raph Levien
Frédéric Liné
Roy F. Longton
Jeff Madsen
Rodolfo Medina
Mel L. O'Cat
Jason Orendorff
Josh Purinton
Steve Rodriguez
Andrew Salmon
Alan Sare
Eric Schmidt
David A. Wheeler

Examples of axiom systems expressible with Metamath
(**Blue** means used by the **set.mm** database)

- Intuitionistic, **classical**, paraconsistent, relevance, quantum propositional logics

- Free or **standard** first-order logic with equality; modal and provability logics

- NBG, **ZF**, NF set theory, with **AC**, GCH, **inaccessible** and other large cardinal axioms

Axiom schemes are **exact** logical equivalents to textbook counterparts. All theorems can be instantly traced back to what axioms they use.

What has been accomplished? (1 of 2)

24 of Freek Wiedijk's "Formalizing 100 Theorems" (from ZFC)

$\sqrt{2}$ irrationality	Cantor's theorem
Denumerability of rationals	Sum of a geometric series
Pythagorean theorem	Sum of an arithmetic series
Euler's gen of Fermat's Little Thm	GCD algorithm
Infinitude of primes	Mathematical induction
De Moivre's theorem	Cauchy-Schwarz inequality
Uncountability of reals	Intermediate value theorem
Schroeder-Bernstein thm	Fundamental thm of arithmetic
Binomial theorem	Desargues's theorem
Number of subsets of a set	Triangle inequality
Bezout's theorem	Bertrand's Postulate
Sum of recipr. of triang. numbers	Formula for Pythagorean triples

What has been accomplished? (2 of 2)

Other examples (all proved directly from ZFC axioms)

Hartogs' theorem (without using Axiom of Choice)
Konig's theorem (set theory)
Dedekind-cut construction of reals
Pocklington's theorem (primality test)
Euler's identity $e^{i\pi} = -1$ (and other complex trig and logs)
Cayley's theorem
Bolzano-Weierstrass theorem
Heine-Borel theorem
Banach fixed point theorem
Baire's category theorem
Uniform boundedness principle (Banach-Steinhaus theorem)
Riesz representation theorem

Theorem bpos 13828

Description: Bertrand's postulate: there is a prime between N and $2N$ for every positive integer N. This proof follows Erdős's method, for the most part, but with some refinements due to Shigenori Tochiori to save us some calculations of large primes. See http://en.wikipedia.org/wiki/Proof_of_Bertrand's_postulate for an overview of the proof strategy. (Contributed by Mario Carneiro, 14-Mar-2014.)

Assertion

Ref	Expression
bpos	$\vdash (N \in \mathbb{N} \to \exists p \in \mathbb{P} (N < p \wedge p \leq (2 \cdot N)))$

Distinct variable group: N, p

Proof of Theorem bpos

Step	Hyp	Ref	Expression
1		nnre 7758	..3 $\vdash (N \in \mathbb{N} \to N \in \mathbb{R})$
2		2re 7809	...4 $\vdash 2 \in \mathbb{R}$
3		6nn 7861	....5 $\vdash 6 \in \mathbb{N}$
4	3	nnnn0i 7996	...4 $\vdash 6 \in \mathbb{N}_0$
5		reexpcl 8716	...4 $\vdash ((2 \in \mathbb{R} \wedge 6 \in \mathbb{N}_0) \to (2\uparrow6) \in \mathbb{R})$
6	2, 4, 5	mp2	Closure of exponentiation of reals.
7		lelttric 7278	..3 $\vdash ((N \in \mathbb{R} \wedge (2\uparrow6) \in \mathbb{R}) \to (N \leq (2\uparrow6) \vee (2\uparrow6) < N))$
8	1, 6, 7	sylancl 720	.2 $\vdash (N \in \mathbb{N} \to (N \leq (2\uparrow6) \vee (2\uparrow6) < N))$
9		bpos1 13819	..3 $\vdash ((N \in \mathbb{N} \wedge N \leq (2\uparrow6)) \to \exists p \in \mathbb{P} (N < p \wedge p \leq (2 \cdot N)))$
10		eqid 2075	.....6 $\vdash (n \in \mathbb{N} \mapsto ((((\surd`2) \cdot ((x \in \mathbb{R}^+ \mapsto ((\log`x) / x))`(\surd`n))) + ((9 / 4) \cdot ((x \in \mathbb{R}^+ \mapsto ((\log`x) / x))`(n / 2)))) + ((\log`2) / (\surd`(2 \cdot n))))) = (n \in \mathbb{N} \mapsto ((((\surd`2) \cdot ((x \in \mathbb{R}^+ \mapsto ((\log`x) / x))`(\surd`n))) + ((9 / 4) \cdot ((x \in \mathbb{R}^+ \mapsto ((\log`x) / x))`(n / 2)))) + ((\log`2) / (\surd`(2 \cdot n)))))$
11		eqid 2075	.....6 $\vdash (x \in \mathbb{R}^+ \mapsto ((\log`x) / x)) = (x \in \mathbb{R}^+ \mapsto ((\log`x) / x))$
[illegible]		[illegible]	[illegible]

Ghilbert

- Ghilbert and Metamath are sister languages. It's easy to convert between them.
- Modularization: Proofs are organized into files which are imported and exported into other files.
- Online Editor. Proofs can be edited online and have LaTeX typesetting.

Go to: ghilbert-app.appspot.com/wiki/tutorial/overview

Sum of an Arithmetic Series edit

Proof

$\sum_{x=0}^{y} x = \frac{y(y+1)}{2} \to \sum_{x=0}^{y+1} x = \frac{y(y+1)}{2} + (y+1)$		Detach the last number in a sum: y + 1
	$\to \sum_{x=0}^{y+1} x = \frac{(y+2)(y+1)}{2}$	Distributive Property
	$\sum_{x=0}^{A} x = \frac{A(A+1)}{2}$	Induction

Ghilbert

The Quadratic Equation

edit

Proofs can be edited directly on the website

The quadratic equation gives two possible solutions to a second-order polynomial equation. This proof begins with the assumption that solutions to the equation exists and that the constant a is not 0. If the value of a were 0, the equation would be linear not quadratic.

Proof

$$ax^2 + bx + c = 0$$ Starting Hypothesis

$$x^2 + \frac{bx}{a} = -\frac{c}{a}$$ Subtract C, Divide A

$$x^2 + \frac{bx}{a} + \left(\frac{b}{2a}\right)^2 = \left(\frac{b}{2a}\right)^2 - \frac{c}{a}$$ Complete the Square

$$\left(x + \frac{b}{2a}\right)^2 = \frac{b^2 - 4ac}{(2a)^2}$$ Factor the polynomial

$$x + \frac{b}{2a} = \frac{\sqrt{b^2 - 4ac}}{2a} \lor x + \frac{b}{2a} = -\frac{\sqrt{b^2 - 4ac}}{2a}$$ Two solutions when inverting a square

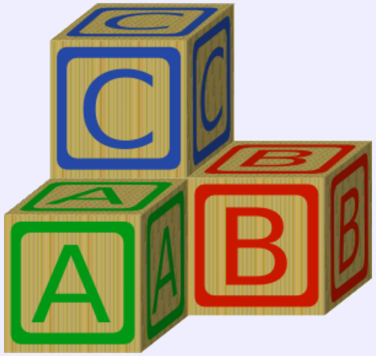
$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \lor x = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$ Subtract B/2A from both sides

Proofs are organized hierarchically.

Click on any step to see how it was derived.

▸ Notation Help ▸ Context

This and other interesting proofs are available at: ghilbert-app.appspot.com/wiki/tutorial/sampler



The Metamath language

Metamath language syntax

Syntax elements: *symbols* (math symbols), *labels* (statement identifiers), and 11 language keywords: `$c` `$v` `$f` `$e` `$d` `$a` `$p` `$.` `$=` `${` `$}`

Constant declaration:	`$c` *symbols* `$.`
Variable declaration:	`$v` *symbols* `$.`
Variable-type assignment:	*label* `$f` *symbols* `$.`
Logical hypothesis:	*label* `$e` *symbols* `$.`
Distinct variable proviso:	`$d` *symbols* `$.`
Axiom scheme:	*label* `$a` *symbols* `$.`
Theorem scheme and its proof:	*label* `$p` *symbols* `$=` *labels* `$.`
Delimit scope of `$f`, `$d`, `$e`:	`${` . . . `$}`

Complete specification is in *Metamath* book, pp. 92–95

(1)	$(\mathscr{A} \to ((\mathscr{A} \to \mathscr{A}) \to \mathscr{A})) \to ((\mathscr{A} \to (\mathscr{A} \to \mathscr{A})) \to (\mathscr{A} \to \mathscr{A}))$	$(L2)$
(2)	$(\mathscr{A} \to ((\mathscr{A} \to \mathscr{A}) \to \mathscr{A}))$	$(L1)$
(3)	$((\mathscr{A} \to (\mathscr{A} \to \mathscr{A})) \to (\mathscr{A} \to \mathscr{A}))$	(1), (2) *MP*
(4)	$(\mathscr{A} \to (\mathscr{A} \to \mathscr{A}))$	$(L1)$
(5)	$(\mathscr{A} \to \mathscr{A})$	(3), (4) *MP*.

Textbook example: Hamilton, *Logic for Mathematicians* (1988), p. 32

Proof of Theorem id1

Step	Hyp	Ref	Expression
1		ax-2 5	..3 $\vdash ((\varphi \to ((\varphi \to \varphi) \to \varphi)) \to ((\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi)))$
2		ax-1 4	..3 $\vdash (\varphi \to ((\varphi \to \varphi) \to \varphi))$
3	1, 2	ax-mp 7	.2 $\vdash ((\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi))$
4		ax-1 4	.2 $\vdash (\varphi \to (\varphi \to \varphi))$
5	3, 4	ax-mp 7	1 $\vdash (\varphi \to \varphi)$

Metamath's web page display of id1 proof

Example - Intuitionistic implicational calculus (1 of 2)

`$c |- wff ( ) -> $.`	Declare 5 constants
`$v ph ps ch $.`	Declare 3 variables (φ, ψ, χ)
`wph $f wff ph $.`	Establish variable type for φ
`wps $f wff ps $.`	Establish variable type for ψ
`wch $f wff ch $.`	Establish variable type for χ
`wi $a wff ( ph -> ps ) $.`	Syntax builder for implication

Two axiom schemes and rule of modus ponens:

```
ax-1 $a |- ( ph -> ( ps -> ph ) ) $.
ax-2 $a |- ( ( ph -> ( ps -> ch ) )
                    -> ( ( ph -> ps ) -> ( ph -> ch ) ) ) $.
${
  maj $e |- ( ph -> ps ) $.
  min $e |- ph $.
  ax-mp $a |- ps $.
$}
```

Example - Intuitionistic implicational calculus (2 of 2)

Theorem scheme: Identity law

```
id1 $p |- ( ph -> ph ) $=
 wph wph wph wi wi wph wph wi wph wph wph wi wph wi wi
 wph wph wph wi wi wph wph wi wi wph wph wph wi wph ax-2
 wph wph wph wi ax-1 ax-mp wph wph ax-1 ax-mp $.
```

Logic step actions and resulting proof steps:

Push `ax-2` `|- ( ( ph -> ( ( ph -> ph ) -> ph ) ) -> ( ( ph -> ( ph -> ph ) ) -> ( ph -> ph ) ) )`

Push `ax-1` `|- ( ph -> ( ( ph -> ph ) -> ph ) )`

Pop `maj`, pop `min`, push `ax-mp`

`|- ( ( ph -> ( ph -> ph ) ) -> ( ph -> ph ) )`

Push `ax-1` `|- ( ph -> ( ph -> ph ) )`

Pop `maj`, pop `min`, push `ax-mp`

`|- ( ph -> ph )`

“Hidden” hypotheses for substitution assignments to variables in `$a` and `$p` statements

ax-1 showing all hypotheses (pops 2 from stack, pushes 1):

```
wph $f wff ph $.
wps $f wff ps $.
ax-1 $a |- ( ph -> ( ps -> ph ) ) $.
```

ax-mp showing all hypotheses (pops 4 from stack, pushes 1):

```
wph $f wff ph $.
wps $f wff ps $.
min $e |- ph $.
maj $e |- ( ph -> ps ) $.
ax-mp $a |- ps $.
```

Syntax-building steps for substitution assignments

Theorem scheme: Identity law

```
id1 $p |- ( ph -> ph ) $=
 wph wph wph wi wi wph wph wi wph wph wph wi wph wi wi
 wph wph wph wi wi wph wph wi wi wph wph wph wi wph ax-2
 wph wph wph wi ax-1 ax-mp wph wph ax-1 ax-mp $.
```

```
MM> show proof id1 /all /lemmon
    ...
31 wph              $f wff ph
32 wph              $f wff ph
33 wph              $f wff ph
34 32,33 wi         $a wff ( ph -> ph )
35 31,34 ax-1       $a |- ( ph -> ( ( ph -> ph ) -> ph ) )
    ...
```

Why explicit syntax-building steps?

Theorem scheme: Identity law

```
id1 $p |- ( ph -> ph ) $=
 wph wph wph wi wi wph wph wi wph wph wph wi wph wi wi
 wph wph wph wi wi wph wph wi wi wph wph wph wi wph ax-2
 wph wph wph wi ax-1 ax-mp wph wph ax-1 ax-mp $.
```

Only the logic steps "ax-2 ax-1 ax-mp ax-1 ax-mp" are needed theoretically (and by some verifiers e.g. Metamath Solitaire)

Advantages of explicit syntax-building steps:

- Faster verification (no unification needed)
- Simpler verifier (no unification algorithm needed)

Disadvantage:

- Verbose proofs

Compressed proofs

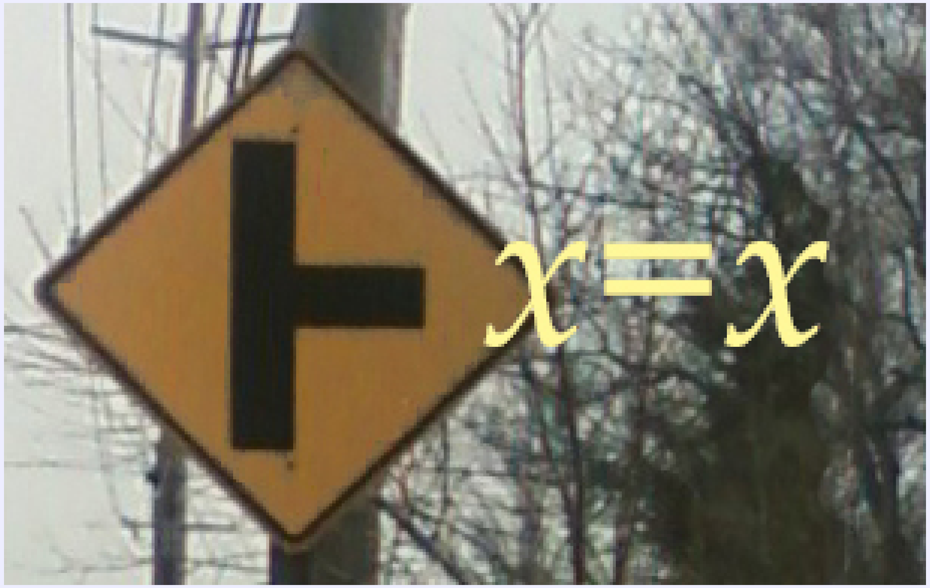
Identity law with **compressed proof**

```
id1 $p |- ( ph -> ph ) $=
 ( wi ax-2 ax-1 ax-mp ) AAABZBZFAFABBGFBAFACAFDEAADE $.
```

Specification is in Appendix B of *Metamath* book

Advantages:

- 85% proof size reduction on average (7$\times$ smaller)
- 6$\times$ faster verification (reading compressed format directly)
- set.mm size breakdown: 8.5MB for proofs, 16.3MB total



Predicate calculus with equality

Classical propositional calculus

We will implicitly assume predicate calculus axioms include:

Axiom *Simp*	**ax-1**	$\vdash (\varphi \to (\psi \to \varphi))$
Axiom *Frege*	**ax-2**	$\vdash ((\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)))$
Axiom *Transp*	**ax-3**	$\vdash ((\neg \varphi \to \neg \psi) \to (\psi \to \varphi))$
Rule of Modus Ponens	**ax-mp**	$\vdash \varphi \quad \& \quad \vdash (\varphi \to \psi) \quad \Rightarrow \quad \vdash \psi$

Axiom schemes for classical propositional calculus (Łukasiewicz's system, called P_2 by Church)

Variables vs. metavariables

Elements of **actual** first-order logic (for set theory):

- Fixed set of individual variables: v_1, v_2, v_3,...
- Wffs (well-formed formulas) constructed from variables connected by $=$ and $\in$, which are then used to build up larger wffs connected with $\rightarrow$, $\neg$, $\forall$ (e.g. $(v_1 = v_3 \rightarrow \neg\forall v_2\ v_2 \in v_4)$).
- There are **no** wff variables

Elements of **Metamath** (set.mm database):

- Individual metavariables x, y,... ranging over v_1, v_2, v_3,...
- Wff metavariables φ, ψ,... ranging over wffs such as $v_2 \in v_4$ and $(v_1 = v_3 \rightarrow \neg\forall v_2\ v_2 \in v_4)$
- $x = y$, $x \in y$, $\neg\ \varphi$, $(\varphi \rightarrow \psi)$, and $\forall x \varphi$ are wff schemes
- Actual variables $v_1, v_2, \ldots$ are **never** mentioned explicitly

Simple schemes and simple metalogic

Simple scheme - An axiom scheme or theorem scheme containing only:

1. Wff metavariables φ, ψ,. . . with no arguments
2. Individual metavariables x, y,. . .
3. Provisos of the form "where x and y are distinct"
4. Provisos of the form "where x does not occur in φ"

Proof using simple metalogic - A proof in which each step is a simple scheme—either a direct substitution into an axiom scheme (inheriting any provisos) or an inference rule applied to previous steps.

Proofs: logic vs. simple metalogic

In a **standard first-order logic proof**, each step is a *single instance* of an axiom scheme (or rule applied to previous steps) using $v_1, v_2, \ldots$. There are no provisos associated with any step (or the final theorem). All variables are "distinct" by definition.

In **simple metalogic**, each proof step is itself a *scheme* using x, y,... and φ, ψ,... and possible distinct-variable provisos

Predicate calculus (with equality) in Metamath

The Metamath language (simple schemes) does not have "free variable" and "proper substitution" as built-in primitives. Traditional predicate calculus cannot be represented directly.

Tarski's system S2 (1965) (with predicates $=$ and $\in$) is **equivalent** but has only simple schemes for its axioms.

Axiom of Quantified Implication	$\vdash (\forall x(\varphi \rightarrow \psi) \rightarrow (\forall x\varphi \rightarrow \forall x\psi))$
Rule of Generalization	$\vdash \varphi \quad \Rightarrow \quad \vdash \forall x\varphi$
Axiom of Equality (1)	$\vdash (x = y \rightarrow (x = z \rightarrow y = z))$
Axiom of Existence	$\vdash \neg \forall x \neg x = y$, where x is distinct from y
Axiom of Equality (2)	$\vdash (x = y \rightarrow (x \in z \rightarrow y \in z))$
Axiom of Equality (3)	$\vdash (x = y \rightarrow (z \in x \rightarrow z \in y))$
Axiom of Quantifier Introduction	$\vdash (\varphi \rightarrow \forall x\varphi)$, where x does not occur in φ

Tarski's system S2

Example of proof as intended by Tarski's system S2:

1	$\vdash (v_2 \in v_1 \to (v_2 = v_1 \to v_2 \in v_1))$	ax-1
2	$\vdash \forall v_1(v_2 \in v_1 \to (v_2 = v_1 \to v_2 \in v_1))$	$1,$ ax-gen
3	$\vdash (\forall v_1(v_2 \in v_1 \to (v_2 = v_1 \to v_2 \in v_1))$ $\to (\forall v_1 v_2 \in v_1 \to \forall v_1(v_2 = v_1 \to v_2 \in v_1)))$	ax-5
4	$\vdash (\forall v_1 v_2 \in v_1 \to \forall v_1(v_2 = v_1 \to v_2 \in v_1))$	$2, 3,$ ax-mp

Proof using simple metalogic (Metamath):

1	$\vdash (\varphi \to (\psi \to \varphi))$	ax-1
2	$\vdash \forall x(\varphi \to (\psi \to \varphi))$	$1,$ ax-gen
3	$\vdash (\forall x(\varphi \to (\psi \to \varphi)) \to (\forall x\varphi \to \forall x(\psi \to \varphi)))$	ax-5
4	$\vdash (\forall x\varphi \to \forall x(\psi \to \varphi))$	$2, 3,$ ax-mp

Metalogical completeness

A set of axiom schemes is **metalogically complete** when all valid simple schemes are provable with simple metalogic.

Example: System P_2 of classical propositional calculus is metalogically complete.

Problem: Tarski's system S2, while *logically* complete, is not *metalogically* complete.

Example: $\vdash (x = y \to (\forall y \varphi \to \forall x (x = y \to \varphi)))$ (ax-11 in set.mm) can only be proved in S2 by induction on formula length of φ

Solution: Extend Tarski's S2 with additional (though logically redundant) simple schemes.

Metamath's schemes vs. Tarski's system S2

ax-5	$\vdash (\forall x(\varphi \to \psi) \to (\forall x \varphi \to \forall x \psi))$	$\vdash (\forall x(\varphi \to \psi) \to (\forall x \varphi \to \forall x \psi))$
ax-6	$\vdash (\neg \forall x \varphi \to \forall x \neg \forall x \varphi)$	
ax-7	$\vdash (\forall x \forall y \varphi \to \forall y \forall x \varphi)$	
ax-gen	$\vdash \varphi \quad \Rightarrow \quad \vdash \forall x \varphi$	$\vdash \varphi \quad \Rightarrow \quad \vdash \forall x \varphi$
ax-8	$\vdash (x = y \to (x = z \to y = z))$	$\vdash (x = y \to (x = z \to y = z))$
ax-9	$\vdash \neg \forall x \neg x = y$	$\vdash \neg \forall x \neg x = y$, where x is distinct from y
ax-10	$\vdash (\forall x\, x = y \to \forall y\, y = x)$	
ax-11	$\vdash (x = y \to (\forall y \varphi \to \forall x(x = y \to \varphi)))$	
ax-12	$\vdash (\neg \forall z\, z = x \to (\neg \forall z\, z = y \to (x = y \to \forall z\, x = y)))$	
ax-13	$\vdash (x = y \to (x \in z \to y \in z))$	$\vdash (x = y \to (x \in z \to y \in z))$
ax-14	$\vdash (x = y \to (z \in x \to z \in y))$	$\vdash (x = y \to (z \in x \to z \in y))$
ax-17	$\vdash (\varphi \to \forall x \varphi)$, where x does not occur in φ	$\vdash (\varphi \to \forall x \varphi)$, where x does not occur in φ

Metalogical completeness

Theorem. The extended set of axiom schemes ax-1 through ax-17 is **metalogically complete** (Theorem 9.7 in Megill 1995).

Open problem: The (metalogical) **independence** of of these schemes has not been proven, except for ax-9 and ax-11.

- Independence of ax-9 proved by Raph Levien (2005)
- Independence of ax-11 proved by Juha Arpiainen (2006)

Distinct variable provisos

The axiom scheme "($\varphi \to \forall x \varphi$), where x does not occur in φ" is expressed in the Metamath language as

```
${
   $d x ph $.
   ax-17 $a |- ( ph -> A. x ph ) $.
$}
```

Rule: Substitutions inherit distinct variable provisos.

Example: Substitute $y = z$ for φ. Then

($\varphi \to \forall x \varphi$), where x does not occur in φ

becomes

($y = z \to \forall x\ y = z$), where x is distinct from y and z.

Traditional logic notions using Metamath

Traditional logic: "where x is not free in φ"
Metamath: use logical (`$e`) hypothesis $\vdash (\varphi \rightarrow \forall x \varphi)$

Traditional logic: "The proper substitution of y for x in φ"
Metamath: $[y/x]\varphi$, defined $((x = y \rightarrow \neg\varphi) \rightarrow \forall x(x = y \rightarrow \varphi)))$

Traditional logic: "$\varphi(y)$ where y is free for x in $\varphi(x)$"
Metamath: $[y/x]\varphi$

Definitions in Metamath

- Definitions are introduced as axioms (`$a`) and are indistinguishable from axioms to the verifier

- Soundness (eliminability and non-creativity) depends highly on the underlying logic and cannot be automatically checked generally

- In set.mm we require new definitions to be automatically checkable. **All but 3 definitions in set.mm are automatically verifiable with a simple algorithm.**

Definitions for predicate calculus in `set.mm`

Definitions extend wff syntax, and the definiendum (l.h.s.) and definiens (r.h.s.) are connected with the biconditional $\leftrightarrow$.

Examples:

df-an $\quad \vdash ((\varphi \wedge \psi) \;\leftrightarrow\; \neg(\varphi \rightarrow \neg\psi))$

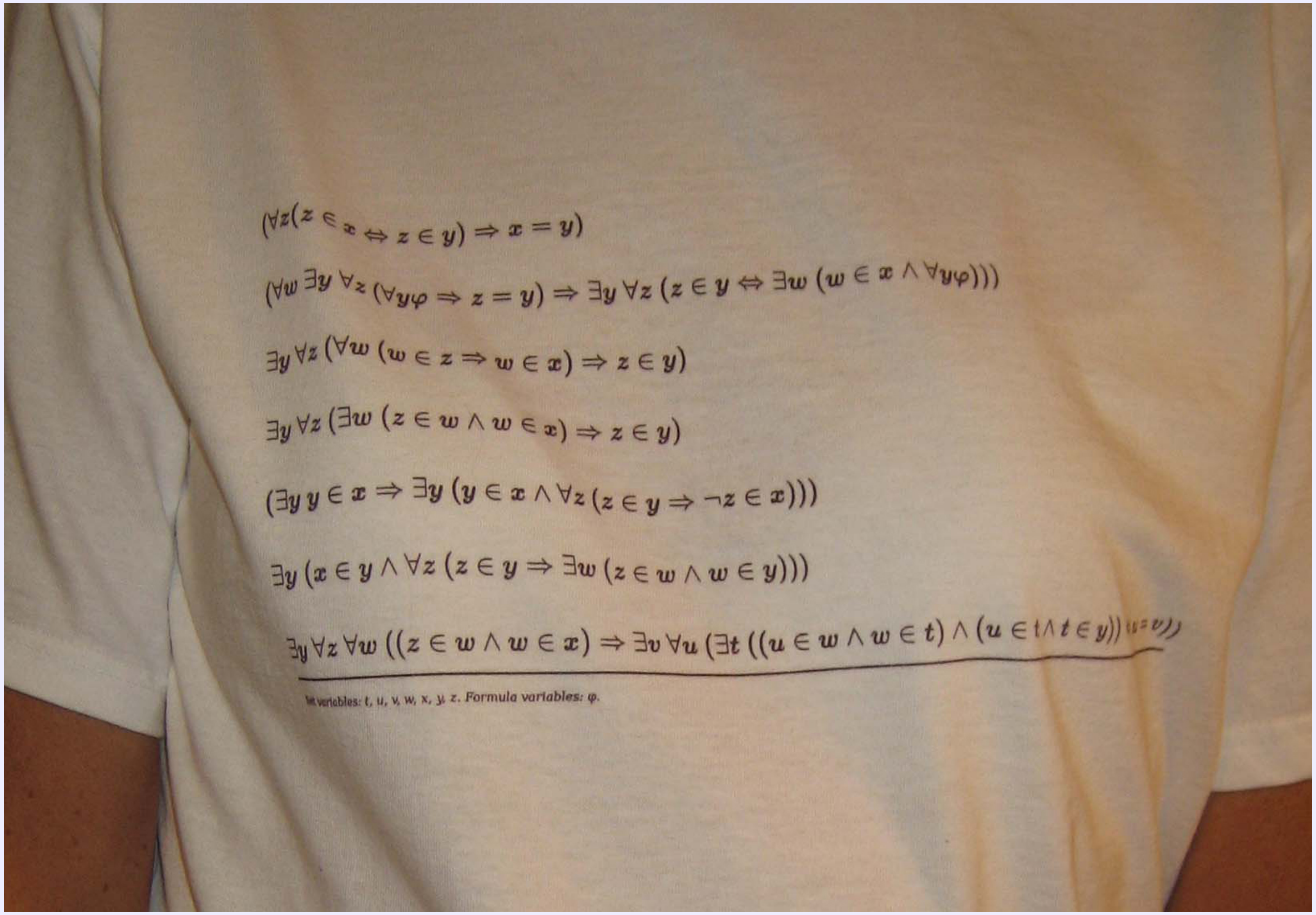
df-ex $\quad \vdash (\exists x \varphi \;\leftrightarrow\; \neg\forall x \neg\varphi)$

df-eu $\quad \vdash (\exists ! x \varphi \;\leftrightarrow\; \exists y \forall x (\varphi \leftrightarrow x = y))$

where x and y are distinct and y does not occur in φ

Any *new* variable on r.h.s. must be distinct from all others.

ZFC set theory



Axiom schemes for ZFC set theory in `set.mm`

All individual metavariables x, y, z,... below are assumed to be mutually distinct		
Axiom of Extensionality	**ax-ext**	$\vdash (\forall z(z \in x \leftrightarrow z \in y) \rightarrow x = y)$
Axiom of Replacement	**ax-rep**	$\vdash (\forall w \exists y \forall z(\forall y \varphi \rightarrow z = y) \rightarrow \exists y \forall z(z \in y \leftrightarrow \exists w(w \in x \wedge \forall y \varphi)))$
Axiom of Power Sets	**ax-pow**	$\vdash \exists y \forall z(\forall w(w \in z \rightarrow w \in x) \rightarrow z \in y)$
Axiom of Union	**ax-un**	$\vdash \exists y \forall z(\exists w(z \in w \wedge w \in x) \rightarrow z \in y)$
Axiom of Regularity	**ax-reg**	$\vdash (\exists y\, y \in x \rightarrow \exists y(y \in x \wedge \forall z(z \in y \rightarrow \neg z \in x)))$
Axiom of Infinity	**ax-inf**	$\vdash \exists y(x \in y \wedge \forall z(z \in y \rightarrow \exists w(z \in w \wedge w \in y)))$
Axiom of Choice	**ax-ac**	$\vdash \exists y \forall z \forall w((z \in w \wedge w \in x) \rightarrow \exists v \forall u(\exists t((u \in w \wedge w \in t) \wedge (u \in t \wedge t \in y)) \leftrightarrow u = v))$

Axioms vs. axiom schemes again

In Metamath, every axiom, theorem, and proof step is a simple scheme

In **standard ZFC set theory**, the Axiom of Extensionality is a **specific axiom** in the language of first-order logic:

$$(\forall v_3(v_3 \in v_1 \leftrightarrow v_3 \in v_2) \rightarrow v_1 = v_2)$$

In **Metamath (set.mm)**, this is stated as an **axiom scheme**:

$$(\forall z\,(z \in x \leftrightarrow z \in y) \rightarrow x = y), \text{ where } x,\ y,\ z \text{ are distinct}$$

Under first-order logic, every instance of this scheme is logically equivalent to the specific axiom

Axiom Scheme of Replacement

In set.mm, Replacement is *automatically* a scheme:

$$(\forall w \exists y \forall z (\forall y \varphi \to z = y) \to \exists y \forall z (z \in y \leftrightarrow \exists w \, (w \in x \land \forall y \varphi))),$$

where x, y, z, w are distinct

By using $\forall y \varphi$ instead of φ, we "protect" it against the case where φ might be substituted with an expression containing y.

Alternately, we could use just φ and add the proviso "where y does not occur in φ." **A matter of taste.**

We can also eliminate **all** provisos:

$$\exists x (\exists y \forall z (\varphi \to z = y) \to \forall z (\forall y \, z \in x \leftrightarrow \exists x (\forall z \, x \in y \land \forall y \varphi)))$$

Class builders

A **class builder** is an expression of the form $\{x \mid \varphi\}$. Let A, B,... be metavariables ranging over class builders. We extend wffs with the following "definitions:"

$$\begin{aligned} y \in \{x \mid \varphi\} &\leftrightarrow [y \,/\, x]\varphi \\ A = B &\leftrightarrow \forall x\,(x \in A \leftrightarrow x \in B) \\ A \in B &\leftrightarrow \exists x\,(x = A \wedge x \in B) \end{aligned}$$

where x does not occur in A or B. Soundness (eliminability, non-creativity) must be proved outside of Metamath, and Metamath treats them (like all definitions) as **axioms**.

We can prove $x = \{y \mid y \in x\}$ when x and y are distinct, so an individual variable x is a special case of a class expression.

Defining new classes

In definitions extending class syntax, the definiendum (l.h.s.) and definiens (r.h.s.) are connected with equality $=$.

Examples: Universal class, union of a class, maps-to notation

df-v $\quad \vdash \mathrm{V} \ = \ \{x|x = x\}$

df-uni $\quad \vdash \bigcup A \ = \ \{x|\exists y(x \in y \wedge y \in A)\}$
where x and y are distinct and do not occur in A

df-mpt $\quad \vdash (x \in A \mapsto B) \ = \ \{\langle x, y\rangle|(x \in A \wedge y = B)\}$
where x and y are distinct, and y does not occur in A or B

Emulating deductions in a Hilbert-style system (1 of 2)

- Metamath is intended for **Hilbert-style deductive systems** (axiom schemes plus inference rules)

- **Metamath does not have the Deduction Theorem built in** ("$\Delta \cup \{P\} \vdash Q$ implies $\Delta \vdash P \to Q$")**.**

- Alternative: Natural deduction emulation

Emulating deductions in a Hilbert-style system (2 of 2)

Theorem pockthg 13697

Description: The generalized Pocklington's theorem. If $N - 1 = A \cdot B$ where $B < A$, then N is prime if and only if for every prime factor p of A, there is an x such that $x \uparrow (N - 1) = 1 (\mod N)$ and $\gcd(x \uparrow ((N - 1) / p) - 1, N) = 1$. (Contributed by Mario Carneiro, 3-Mar-2014.)

Hypotheses

Ref	Expression
pockthg.1	$\vdash (\varphi \to A \in \mathbb{N})$
pockthg.2	$\vdash (\varphi \to B \in \mathbb{N})$
pockthg.3	$\vdash (\varphi \to B < A)$
pockthg.4	$\vdash (\varphi \to N = ((A \cdot B) + 1))$
pockthg.5	$\vdash (\varphi \to \forall p \in \mathbb{P} (p \parallel A \to \exists x \in \mathbb{Z} (((x \uparrow (N - 1)) \bmod N) = 1$ $\wedge (((x \uparrow ((N - 1) / p)) - 1) \gcd N) = 1)))$

Assertion

Ref	Expression
pockthg	$\vdash (\varphi \to N \in \mathbb{P})$

The End

Thank you!

Supplementary slides

Recursive definitions (1 of 2)

Recursive definitions are hard to eliminate. Instead, we can define a "recursive definition generator" (df-rdg):

$$
\begin{aligned}
\vdash \mathrm{rec}(F, A) \;=\; \bigcup\{f \mid \exists x \in \mathrm{On}\,(f \;\mathrm{Fn}\, x \\
\land \forall y \in x\, f`y = (g \mapsto \mathrm{if}(g = \varnothing, A, \\
\mathrm{if}(\mathrm{Lim}\,\mathrm{dom}\, g, \bigcup \mathrm{ran}\, g, \\
F`(g`\bigcup \mathrm{dom}\, g))))`(f \restriction y))\}, \\
\text{where } x, y, f, g \text{ don't occur in } F \text{ or } A
\end{aligned}
$$

F is the characteristic function, A is the initial value, and $\mathrm{rec}(F, A)$ is a function on the (proper) class of all ordinals.

Recursive definitions (2 of 2)

Ordinal addition is defined with a direct definition (df-oadd):

$$\vdash +_o = (x \in \mathrm{On}, y \in \mathrm{On} \mapsto (\mathrm{rec}((z \in \mathrm{V} \mapsto \mathrm{suc}\, z), x)`y))$$

where x, y, z are distinct

Recursive definition emerges as theorems (oa0, oasuc, oalim):

$$\vdash (A \in \mathrm{On} \to (A +_o \varnothing) = A)$$

$$\vdash ((A \in \mathrm{On} \wedge B \in \mathrm{On}) \to (A +_o \mathrm{suc}\, B) = \mathrm{suc}\,(A +_o B))$$

$$\vdash ((A \in \mathrm{On} \wedge B \in \mathrm{On} \wedge \mathrm{Lim}\, B) \to (A +_o B) = \underline{\bigcup} x \in B (A +_o x)),$$

where x doesn't occur in A or B

Emulating Hilbert's epsilon in ZFC (1 of 2)

The class expression "$\varepsilon x \varphi$" denotes "some x satisfying wff φ."
The **Transfinite Axiom** is a conservative extension of ZFC:

$$\varphi \to [\varepsilon x \varphi / x]\varphi$$

where x is free in φ and $[\ldots/x]\varphi$ denotes proper substitution.

To emulate the transfinite axiom in ZFC, we define two class expressions A and B, where y is does not occur in φ:

$$\begin{aligned} A &= \{x | (\varphi \wedge \forall y([y/x]\varphi \to (\mathrm{rank}`x) \subseteq (\mathrm{rank}`y)))\} \\ B &= \bigcup\{x \in A | \forall y \in A \neg y \, r \, x\} \end{aligned}$$

Theorem (hta in set.mm):

$$r \text{ We } A \to (\varphi \to [B/x]\varphi)$$

Class B emulates Hilbert's epsilon $\varepsilon x \varphi$.

Emulating Hilbert's epsilon in ZFC (2 of 2)

Epsilon-calculus proof	**ZFC proof**
$\vdots$	$\vdots$
$\varphi \to [\varepsilon x \varphi / x]\varphi$	r We $A \to (\varphi \to [B(r)/x]\varphi)$
$\vdots$	$\vdots$
(manipulate $\varepsilon x \varphi$)	(manipulate $B(r)$)
$\vdots$	$\vdots$
($\varepsilon x \varphi$-free result)	r We $A \to$ ($B(r)$-free result)
	$\exists r\, r$ We $A \to$ ($B(r)$-free result)
	($B(r)$-free result)
$\vdots$	$\vdots$

More details:
http://us.metamath.org/downloads/megillaward2005he.pdf