Arithmetic in Metamath
Case Study: Bertrand's Postulate

MARIO CARNEIRO

24 JULY 2015

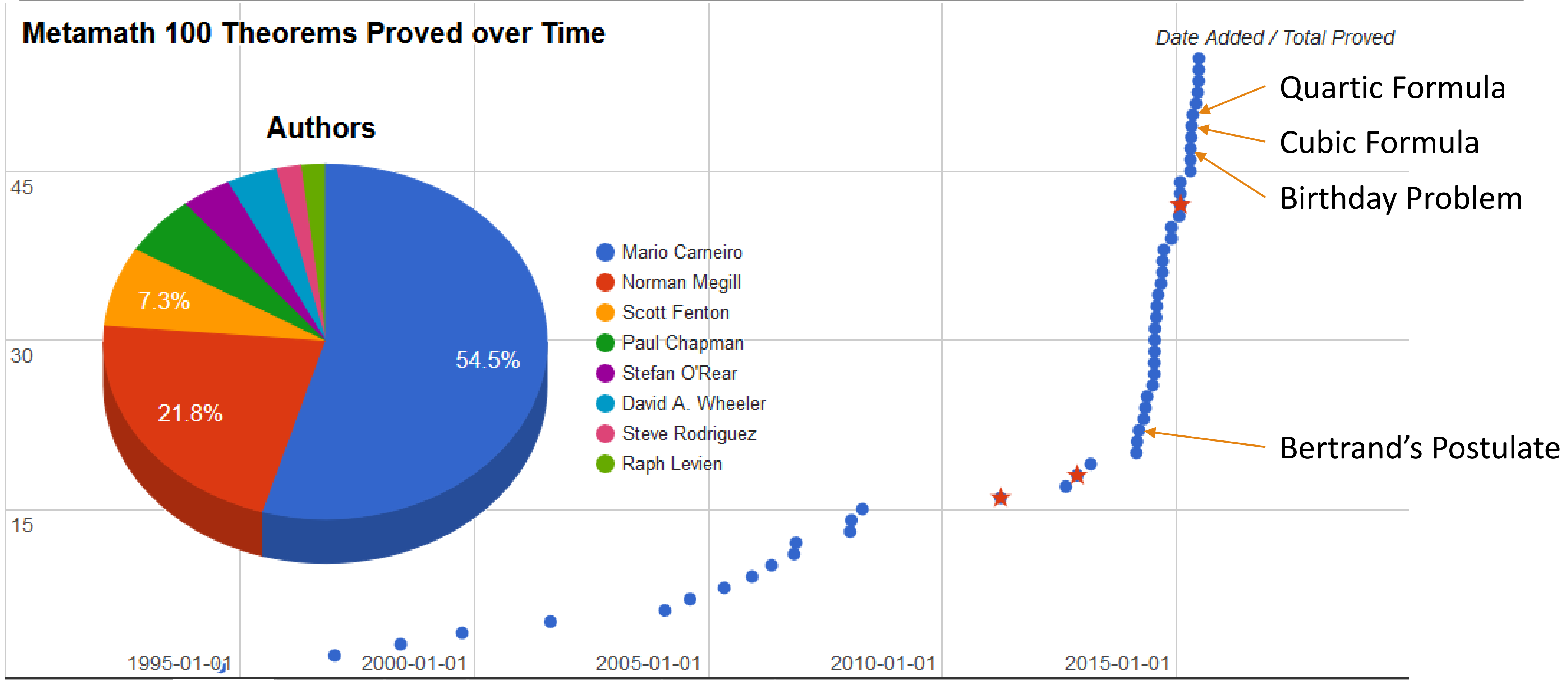
What is Metamath?

- A computer language for representing mathematical proofs
  - The Metamath spec is two pages, one verifier exists in ≈300 lines of Python
  - Eight independent verifiers exist in eight different languages
  - Two proof assistants (MM-PA and mmj2) with another (smm) in development
- A project to formalize modern mathematics from a simple foundation
- Four major databases
  - ZFC set theory (set.mm)
    - Over 25000 proofs, 500K lines, 24M file
  - HOL type theory (hol.mm)
  - Intuitionistic logic (iset.mm)
  - NF set theory (nf.mm)
    - Including Specker’s proof of $\neg \mathrm{AC}$

Metamath 100

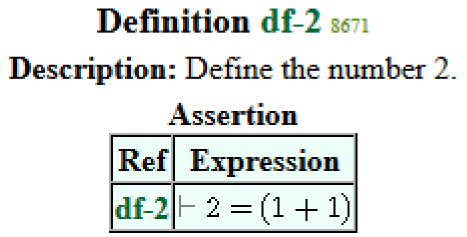
- A project to prove the “Formalizing 100 Theorems” list tracked by Freek Wiedijk
- Currently 55 theorems proven (one short of Isabelle)
- New since last year: (in chronological order)
  - Divergence of Harmonic series
  - Lagrange subgroup theorem
  - Number of Combinations
  - Divisibility by 3
  - Lagrange four-square thm
  - Factor and Remainder thms
  - Basel problem
  - Divergence of inverse primes
  - Fundamental thm of Calculus
  - Mean value theorem
  - Fundamental thm of Algebra
  - Sum of angles in a triangle
  - Solutions to Pell’s equation
  - Liouville’s theorem
  - Sylow’s theorem
  - Wilson’s theorem
  - Erdős-Szekeres theorem
  - Derangements formula
  - Leibniz’ series for π
  - Konigsberg Bridge problem
  - Birthday problem
  - Ramsey’s theorem
  - Solution to a Cubic
  - Solution to a Quartic
  - GCH implies AC (Wednesday)
  - Ptolemy’s theorem
  - Law of Cosines
  - Quadratic reciprocity
  - Sums of two squares
  - Arithmetic/Geometric means

Metamath 100



How does it work?

- A theorem or axiom has a list of hypotheses and a conclusion, which are sequences of constant and variable symbols
  - i.e. φ is a variable, $\rightarrow$ is a constant
- Definitions are the same as axioms
  - A separate program uses a simple "checklist" to ensure definitions are conservative
  - 2 is a constant symbol; its definition is given by the definition/axiom $2 = 1 + 1$

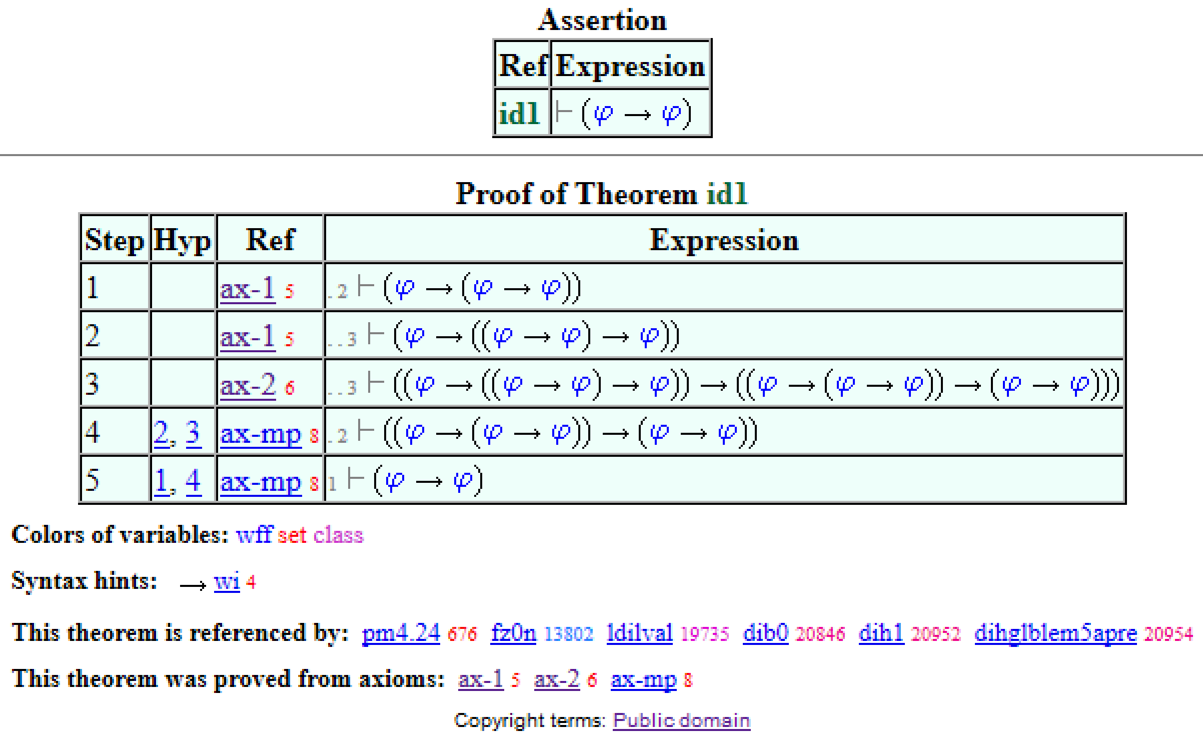


Definition df-2 8671

Description: Define the number 2.

Assertion

Ref	Expression
df-2	$\vdash 2 = (1 + 1)$



Assertion

Ref	Expression
id1	$\vdash (\varphi \rightarrow \varphi)$

Proof of Theorem id1

Step	Hyp	Ref	Expression
1		ax-1 5	.2 $\vdash (\varphi \rightarrow (\varphi \rightarrow \varphi))$
2		ax-1 5	..3 $\vdash (\varphi \rightarrow ((\varphi \rightarrow \varphi) \rightarrow \varphi))$
3		ax-2 6	..3 $\vdash ((\varphi \rightarrow ((\varphi \rightarrow \varphi) \rightarrow \varphi)) \rightarrow ((\varphi \rightarrow (\varphi \rightarrow \varphi)) \rightarrow (\varphi \rightarrow \varphi)))$
4	2, 3	ax-mp 8	.2 $\vdash ((\varphi \rightarrow (\varphi \rightarrow \varphi)) \rightarrow (\varphi \rightarrow \varphi))$
5	1, 4	ax-mp 8	1 $\vdash (\varphi \rightarrow \varphi)$

Colors of variables: wff set class

Syntax hints: $\rightarrow$ wi 4

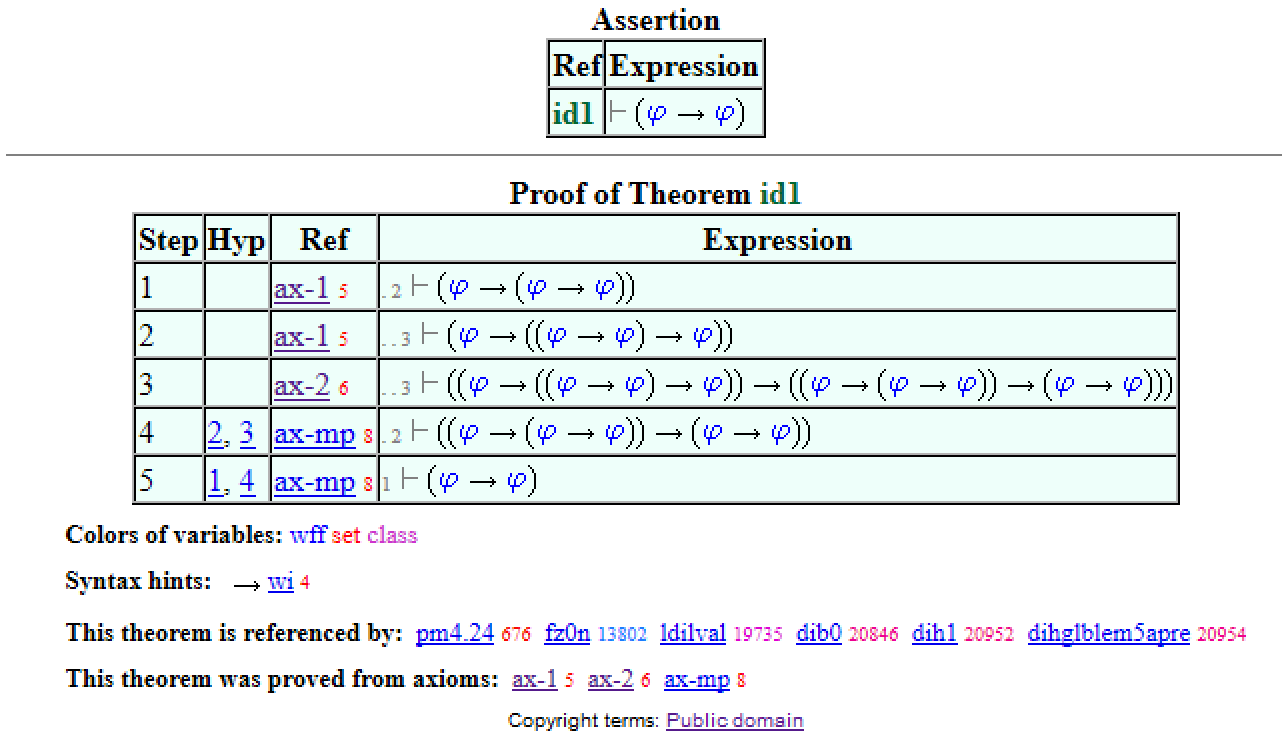
This theorem is referenced by: pm4.24 676 fz0n 13802 ldilval 19735 dib0 20846 dih1 20952 dihglblem5apre 20954

This theorem was proved from axioms: ax-1 5 ax-2 6 ax-mp 8

Copyright terms: Public domain

How does it work?

- Each step is a *direct* substitution for the variables in a previous theorem or axiom, possibly with hypotheses
- No symbol is meaningful until it is given a definition, so a number like 999 will be a syntax error unless it is defined
  - and even then it may not necessarily be defined to mean $10^3 - 1$
- It is possible to define arbitrary syntax with multiple variables like $(\varphi \to \psi)$
  - Ambiguity is not allowed: $\varphi \to \psi$ is not valid because $\varphi \to \psi \to \chi$ has two parse trees
  - Prefix syntax like $\to\varphi\psi$ are always valid



Assertion

Ref	Expression
id1	$\vdash (\varphi \to \varphi)$

Proof of Theorem id1

Step	Hyp	Ref	Expression
1		ax-1 5	.2 $\vdash (\varphi \to (\varphi \to \varphi))$
2		ax-1 5	..3 $\vdash (\varphi \to ((\varphi \to \varphi) \to \varphi))$
3		ax-2 6	..3 $\vdash ((\varphi \to ((\varphi \to \varphi) \to \varphi)) \to ((\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi)))$
4	2, 3	ax-mp 8	.2 $\vdash ((\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi))$
5	1, 4	ax-mp 8	1 $\vdash (\varphi \to \varphi)$

Colors of variables: wff set class

Syntax hints: $\to$ wi 4

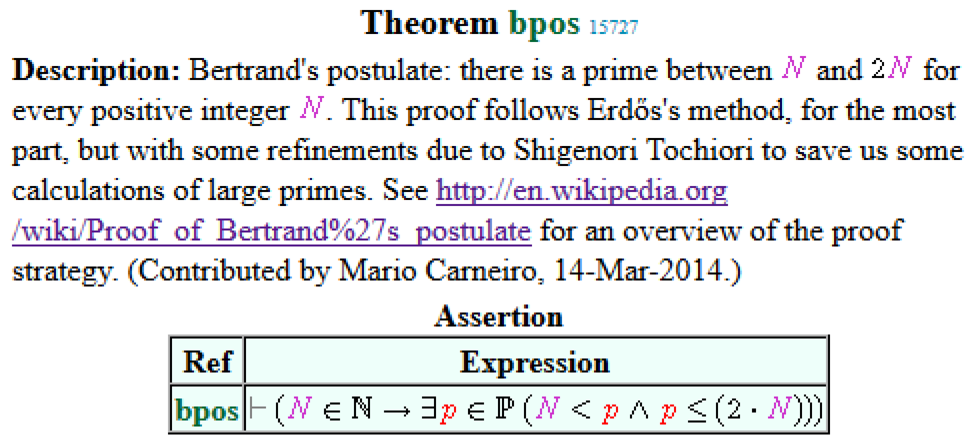
This theorem is referenced by: pm4.24 676 fz0n 13802 ldilval 19735 dib0 20846 dih1 20952 dihglblem5apre 20954

This theorem was proved from axioms: ax-1 5 ax-2 6 ax-mp 8

Copyright terms: Public domain

Bertrand's Postulate

- There is a prime between n and $2n$
- Most proofs, like Erdős's, start with: "Assume that $n > 4000$."
  - That's a lot of base cases!
- These base cases are addressed with the sequence $2,3,5,7,13,23,43,83,163,317,631,1259,2503$ which (we claim) contains only primes
  - How to prove a number is prime?
    - Trial division
    - Pocklington's theorem (thank you Mizar)
- Need a good way to handle large arithmetic calculations



Theorem bpos 15727

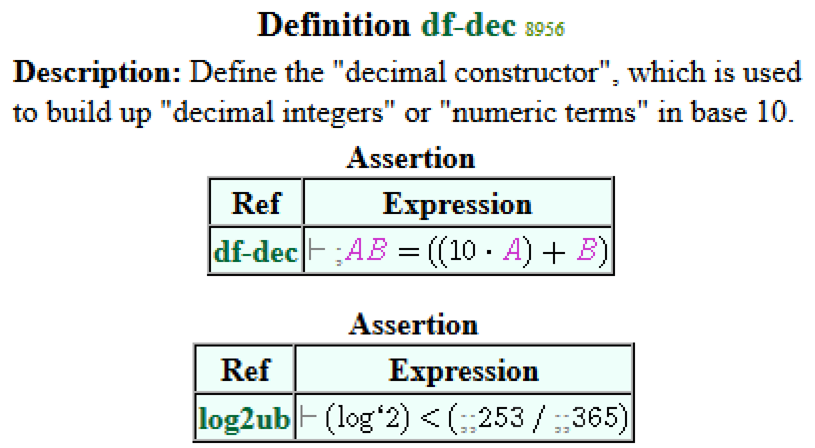
Description: Bertrand's postulate: there is a prime between N and $2N$ for every positive integer N. This proof follows Erdős's method, for the most part, but with some refinements due to Shigenori Tochiori to save us some calculations of large primes. See http://en.wikipedia.org/wiki/Proof_of_Bertrand%27s_postulate for an overview of the proof strategy. (Contributed by Mario Carneiro, 14-Mar-2014.)

Assertion

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

The decimal operator

- Define $;AB = 10A + B$
  - Ex: $;13 = 10 \cdot 1 + 3$,
    $;;269 = 10 \cdot (10 \cdot 2 + 6) + 9$
  - Base 10, not base 4
- Structure of a decimal term is as a tree of “low digit” – “higher digits” nodes
  - Technically allows “nonstandard” constructions such as $;2;69 = 89$ but these are not used in the algorithm
- Ten has two representations – the symbol 10 and $;1\,0$



Definition df-dec 8956

Description: Define the "decimal constructor", which is used to build up "decimal integers" or "numeric terms" in base 10.

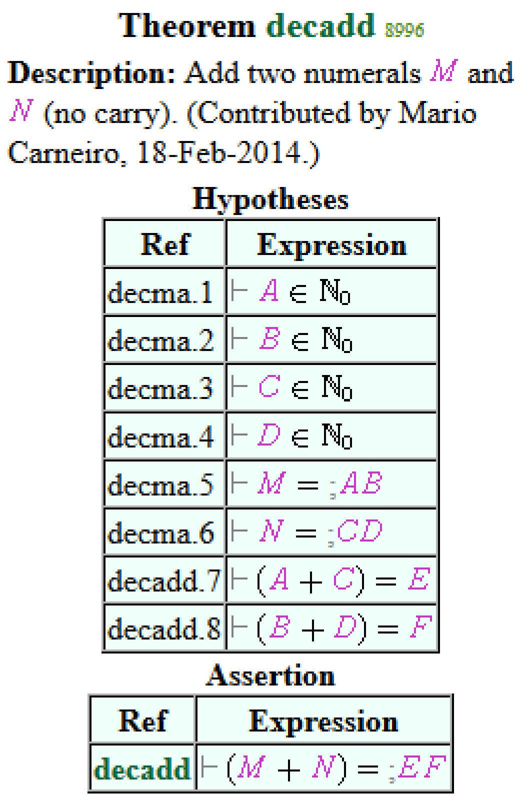
Assertion

Ref	Expression
df-dec	$\vdash ;AB = ((10 \cdot A) + B)$

Assertion

Ref	Expression
log2ub	$\vdash (\log‘2) < (;;253 / ;;365)$

Building blocks



Theorem decadd 8996

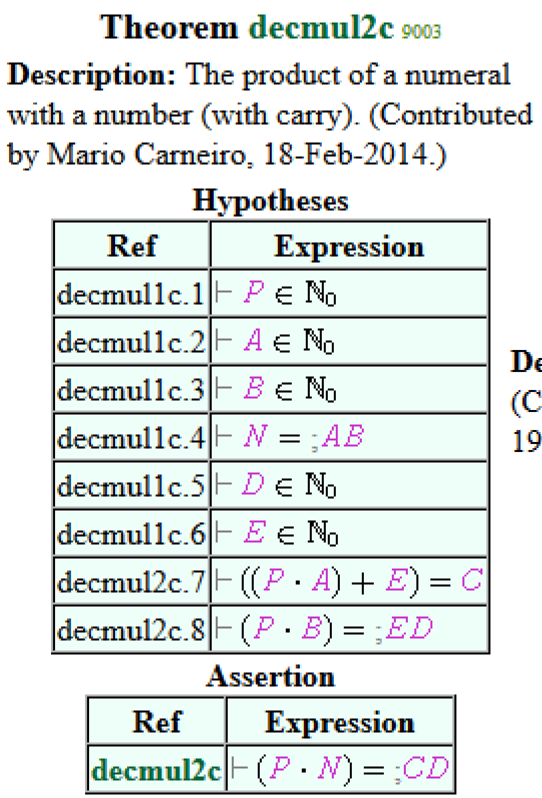
Description: Add two numerals M and N (no carry). (Contributed by Mario Carneiro, 18-Feb-2014.)

Hypotheses

Ref	Expression
decma.1	$\vdash A \in \mathbb{N}_0$
decma.2	$\vdash B \in \mathbb{N}_0$
decma.3	$\vdash C \in \mathbb{N}_0$
decma.4	$\vdash D \in \mathbb{N}_0$
decma.5	$\vdash M = ;AB$
decma.6	$\vdash N = ;CD$
decadd.7	$\vdash (A + C) = E$
decadd.8	$\vdash (B + D) = F$

Assertion

Ref	Expression
decadd	$\vdash (M + N) = ;EF$



Theorem decmul2c 9003

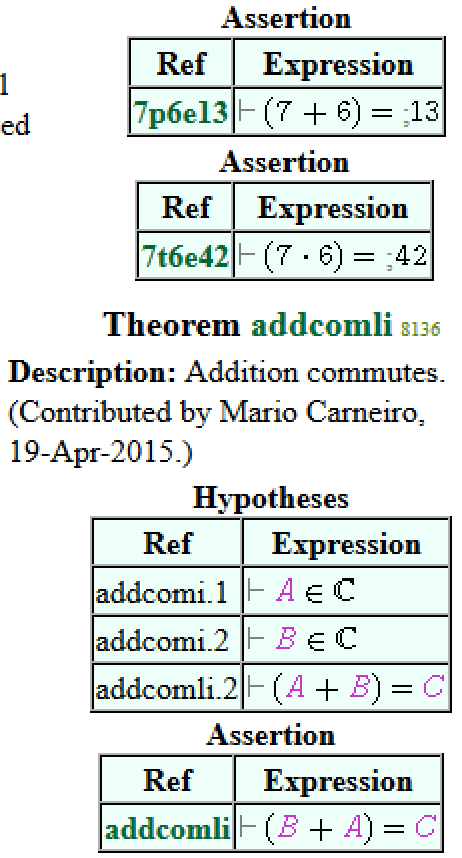
Description: The product of a numeral with a number (with carry). (Contributed by Mario Carneiro, 18-Feb-2014.)

Hypotheses

Ref	Expression
decmul1c.1	$\vdash P \in \mathbb{N}_0$
decmul1c.2	$\vdash A \in \mathbb{N}_0$
decmul1c.3	$\vdash B \in \mathbb{N}_0$
decmul1c.4	$\vdash N = ;AB$
decmul1c.5	$\vdash D \in \mathbb{N}_0$
decmul1c.6	$\vdash E \in \mathbb{N}_0$
decmul2c.7	$\vdash ((P \cdot A) + E) = C$
decmul2c.8	$\vdash (P \cdot B) = ;ED$

Assertion

Ref	Expression
decmul2c	$\vdash (P \cdot N) = ;CD$



Assertion

Ref	Expression
7p6e13	$\vdash (7 + 6) = ;13$

Assertion

Ref	Expression
7t6e42	$\vdash (7 \cdot 6) = ;42$

Theorem addcomli 8136

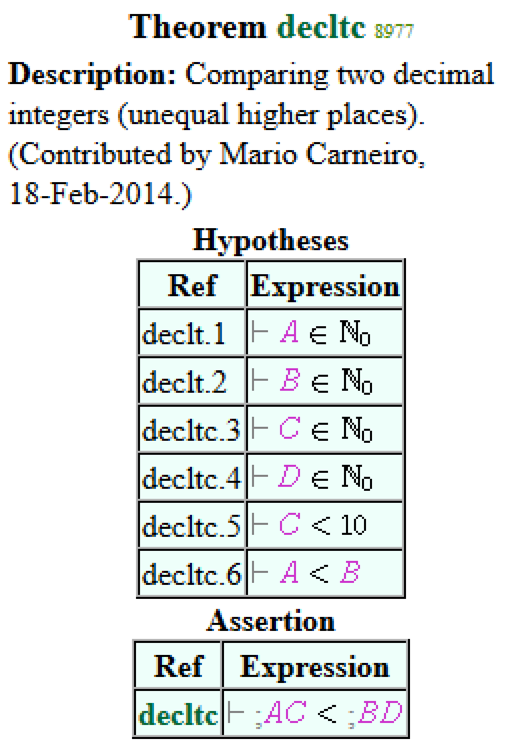
Description: Addition commutes. (Contributed by Mario Carneiro, 19-Apr-2015.)

Hypotheses

Ref	Expression
addcomi.1	$\vdash A \in \mathbb{C}$
addcomi.2	$\vdash B \in \mathbb{C}$
addcomli.2	$\vdash (A + B) = C$

Assertion

Ref	Expression
addcomli	$\vdash (B + A) = C$



Theorem decltc 8977

Description: Comparing two decimal integers (unequal higher places). (Contributed by Mario Carneiro, 18-Feb-2014.)

Hypotheses

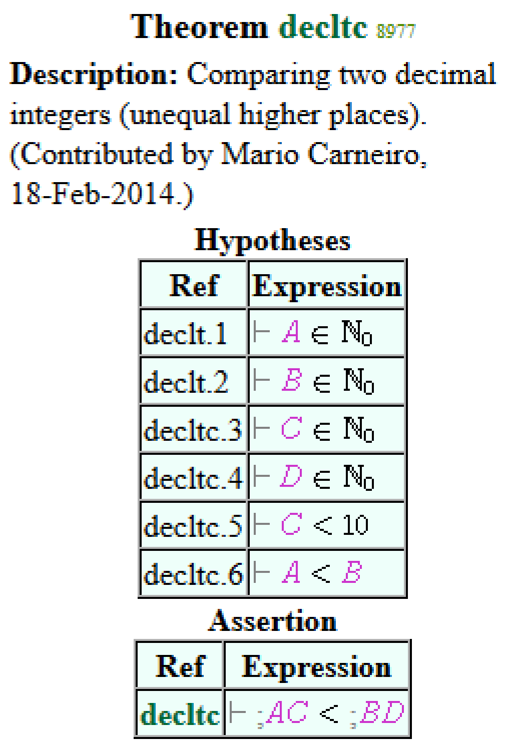
Ref	Expression
declt.1	$\vdash A \in \mathbb{N}_0$
declt.2	$\vdash B \in \mathbb{N}_0$
decltc.3	$\vdash C \in \mathbb{N}_0$
decltc.4	$\vdash D \in \mathbb{N}_0$
decltc.5	$\vdash C < 10$
decltc.6	$\vdash A < B$

Assertion

Ref	Expression
decltc	$\vdash ;AC < ;BD$

Building blocks

- We can recurse over the tree (list) structure of a term
  - The basic algorithms for addition and multiplication can be defined recursively over the structure
  - Algorithms like Karatsuba require splitting the input digit string in half, which cannot be done in one step but can be done with a helper theorem in O(n) steps
  - Asymptotics similar to list operations in Lisp
- Similar techniques can be applied to store many data structures, like graphs (see [konigsberg])



Theorem decltc 8977

Description: Comparing two decimal integers (unequal higher places). (Contributed by Mario Carneiro, 18-Feb-2014.)

Hypotheses

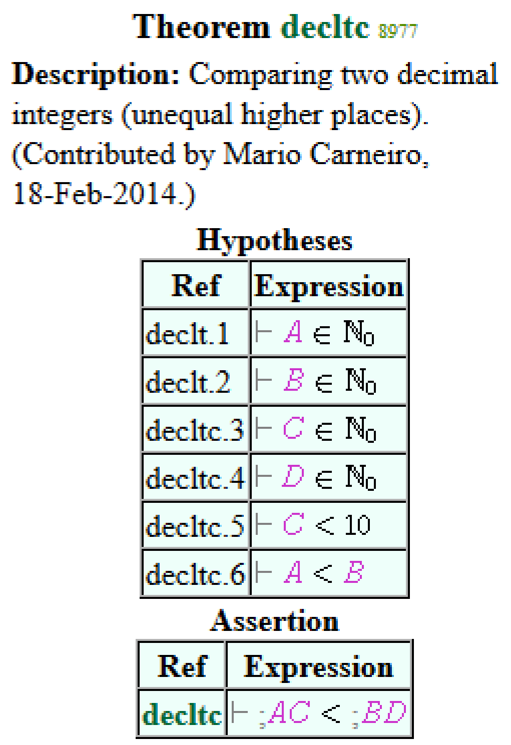
Ref	Expression
declt.1	$\vdash A \in \mathbb{N}_0$
declt.2	$\vdash B \in \mathbb{N}_0$
decltc.3	$\vdash C \in \mathbb{N}_0$
decltc.4	$\vdash D \in \mathbb{N}_0$
decltc.5	$\vdash C < 10$
decltc.6	$\vdash A < B$

Assertion

Ref	Expression
decltc	$\vdash ;AC < ;BD$

The algorithm

- Proofs are context free (always the same steps given the same input assertion)
  - Allows for a simple recursive structure: Decide what theorem to apply based on the form of the goal, and then prove the resulting subgoals
- We can evaluate terms as part of determining the theorem to apply
  - "Evaluate" here means to convert the term ;;123 to the integer 123, or evaluate a multiplication or addition using Mathematica's native operations
  - This allows us to fail quickly if we are asked to prove the unprovable
  - The reverse is also possible, converting 123 to ;;123
- Ex: if the goal is of the form $;AB < ;CD$ and $\mathrm{eval}(A) \neq \mathrm{eval}(B)$, then apply decltc (if the goal is true then the subgoals will be too)



Theorem decltc 8977

Description: Comparing two decimal integers (unequal higher places). (Contributed by Mario Carneiro, 18-Feb-2014.)

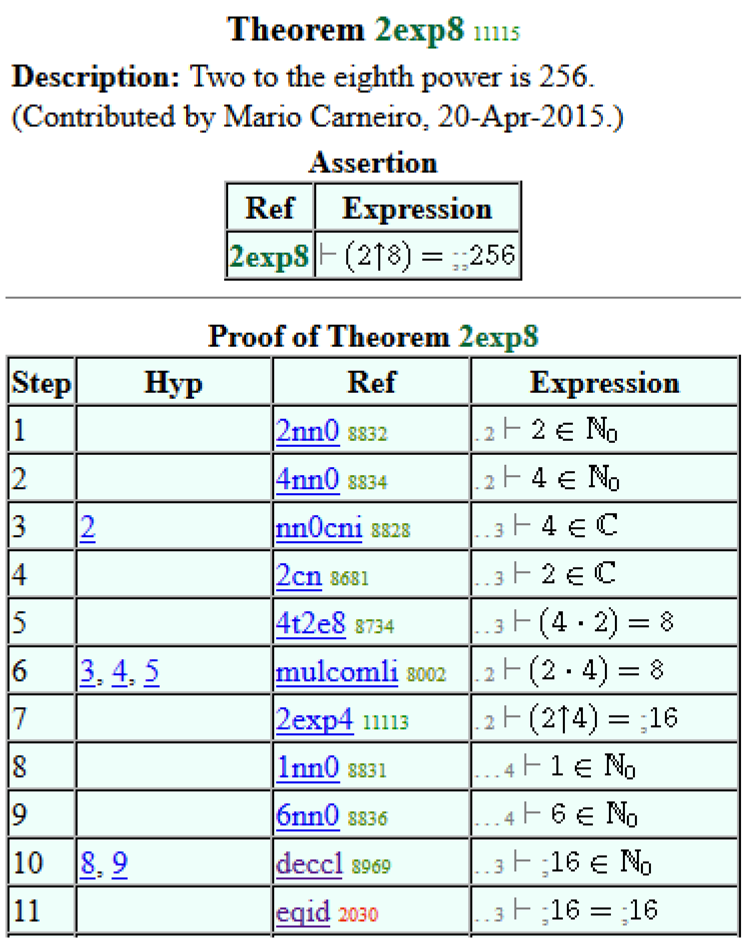
Hypotheses

Ref	Expression
declt.1	$\vdash A \in \mathbb{N}_0$
declt.2	$\vdash B \in \mathbb{N}_0$
decltc.3	$\vdash C \in \mathbb{N}_0$
decltc.4	$\vdash D \in \mathbb{N}_0$
decltc.5	$\vdash C < 10$
decltc.6	$\vdash A < B$

Assertion

Ref	Expression
decltc	$\vdash ;AC < ;BD$

The result



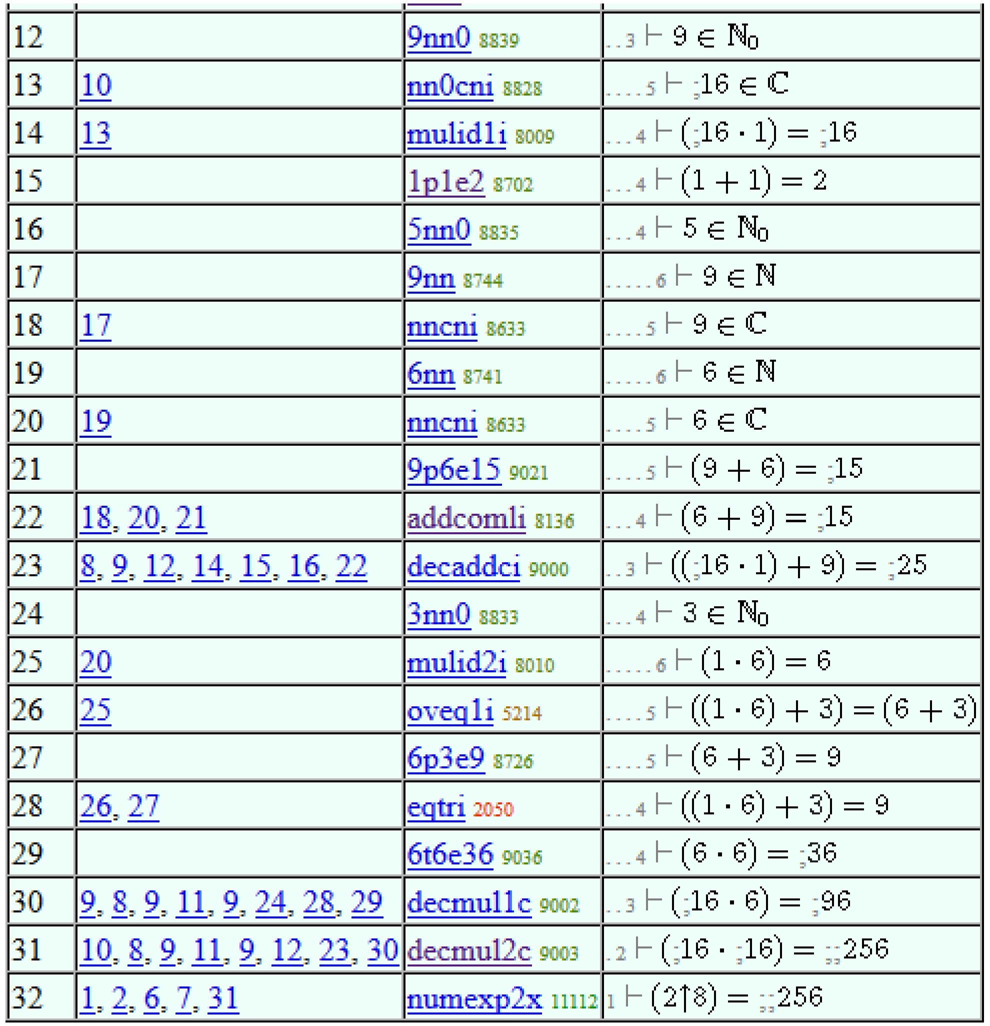
Theorem 2exp8 11115

Description: Two to the eighth power is 256. (Contributed by Mario Carneiro, 20-Apr-2015.)

Assertion

Ref	Expression
2exp8	$\vdash (2 \uparrow 8) = ;;256$

Proof of Theorem 2exp8



Step	Hyp	Ref	Expression
1		2nn0 8832	..2 $\vdash 2 \in \mathbb{N}_0$
2		4nn0 8834	..2 $\vdash 4 \in \mathbb{N}_0$
3	2	nn0cni 8828	...3 $\vdash 4 \in \mathbb{C}$
4		2cn 8681	...3 $\vdash 2 \in \mathbb{C}$
5		4t2e8 8734	...3 $\vdash (4 \cdot 2) = 8$
6	3, 4, 5	mulcomli 8002	..2 $\vdash (2 \cdot 4) = 8$
7		2exp4 11113	..2 $\vdash (2 \uparrow 4) = ;16$
8		1nn0 8831	....4 $\vdash 1 \in \mathbb{N}_0$
9		6nn0 8836	....4 $\vdash 6 \in \mathbb{N}_0$
10	8, 9	deccl 8969	...3 $\vdash ;16 \in \mathbb{N}_0$
11		eqid 2030	...3 $\vdash ;16 = ;16$
12		9nn0 8839	...3 $\vdash 9 \in \mathbb{N}_0$
13	10	nn0cni 8828	....5 $\vdash ;16 \in \mathbb{C}$
14	13	mulid1i 8009	....4 $\vdash (;16 \cdot 1) = ;16$
15		1p1e2 8702	....4 $\vdash (1 + 1) = 2$
16		5nn0 8835	....4 $\vdash 5 \in \mathbb{N}_0$
17		9nn 8744	.....6 $\vdash 9 \in \mathbb{N}$
18	17	nncni 8633	....5 $\vdash 9 \in \mathbb{C}$
19		6nn 8741	.....6 $\vdash 6 \in \mathbb{N}$
20	19	nncni 8633	....5 $\vdash 6 \in \mathbb{C}$
21		9p6e15 9021	....5 $\vdash (9 + 6) = ;15$
22	18, 20, 21	addcomli 8136	....4 $\vdash (6 + 9) = ;15$
23	8, 9, 12, 14, 15, 16, 22	decaddci 9000	...3 $\vdash ((;16 \cdot 1) + 9) = ;25$
24		3nn0 8833	....4 $\vdash 3 \in \mathbb{N}_0$
25	20	mulid2i 8010	.....6 $\vdash (1 \cdot 6) = 6$
26	25	oveq1i 5214	....5 $\vdash ((1 \cdot 6) + 3) = (6 + 3)$
27		6p3e9 8726	....5 $\vdash (6 + 3) = 9$
28	26, 27	eqtri 2050	....4 $\vdash ((1 \cdot 6) + 3) = 9$
29		6t6e36 9036	....4 $\vdash (6 \cdot 6) = ;36$
30	9, 8, 9, 11, 9, 24, 28, 29	decmul1c 9002	...3 $\vdash (;16 \cdot 6) = ;96$
31	10, 8, 9, 11, 9, 12, 23, 30	decmul2c 9003	..2 $\vdash (;16 \cdot ;16) = ;;256$
32	1, 2, 6, 7, 31	numexp2x 11112	1 $\vdash (2 \uparrow 8) = ;;256$

```
2exp8 $p |- ( 2 ^ 8 ) = ; ; 2 5 6 $=
  ( c2 c5 cdc c6 c1 c4 c8 2nn0 4nn0 nn0cni c9 1nn0 6nn0 9nn0 cmul co nncni c3
  2cn caddc 4t2e8 mulcomli 2exp4 deccl eqid mulid1i 1p1e2 5nn0 9nn 6nn 9p6e15
  addcomli decaddci 3nn0 oveq1i 6p3e9 eqtri 6t6e36 decmul1c decmul2c numexp2x
  mulid2i ) AABCZDCEDCZFGHIFAGFIJSUAUBUCEDVCDVDKVDEDLMUDZLMVDUEZMNEDBAVDEOPKL
  MNVDVDVEJUFUGUHKDEBCKUIQDUJQZUKULUMEDKDDRVDMLMVFMUNEDOPZRTPDRTPKVHDRTDVGVBU
  OUPUQURUSUTVA $.
```

Limitations

- This is a *limited domain* theorem prover
  - Can prove any true statement using:
    - The symbols $0,1,2,3,4,5,6,7,8,9,10$
    - The operators $+ \quad \cdot \; ; \; <$
    - The operators $\in \mathbb{N}, \in \mathbb{N}_0, \in \mathbb{C}$.
- Additions are possible for handling e.g. exponentiation
- Most (all?) “numerical” theorems (involving only concrete numbers, not variables) can be reduced to addition, multiplication, and strict order of nonnegative integers, so this is not a big restriction

Success stories

- The first version of the arithmetic algorithm was created in Feb 2014
  - Used for the proof of bpos (Bertrand's postulate)
- The second version was made in Apr 2015, concurrent with this paper (http://arxiv.org/abs/1503.02349)
  - Used for the proof of log2ub ($\log 2 < \frac{253}{365}$), a lemma for birthday (the Birthday problem)
- We don't need to be afraid of big numbers anymore (i.e. casual usage when convenient)
  - Used for the cubic and quartic equations, where numbers like $3^3 = 27$ and $4^4 = 256$ appear

Why did it take so long?

- Metamath, like many formal systems, is geared toward abstract math
  - Most abstract math does not need numbers larger than 10
- Computers are comfortable with bigger numbers than humans
- Automation in Metamath is in its infancy
  - This is the first Metamath theorem prover which produces complete proofs
  - Metamath does have a step search of depth up to around 3
  - Much more is planned, and current work on the new smm proof assistant promises “user scripts” like HOL proof programs

Flyspeck

- Flyspeck is Tom Hales’ recently completed project to prove the Kepler Conjecture in HOL Light and Isabelle
- Is it feasible to port Flyspeck to Metamath?
  - Not yet
- A Metamath proof verifies in linear time, but the proofs are longer
  - The length of a Metamath proof is proportional to the running time of a HOL Light proof
- How to optimize for Metamath
  - No searches: you already know the answer!
  - Metamath proves NP-hard problems in polynomial time (cf. Luís “offline oracles”)
  - Spend more time making the proofs shorter
    - Round all numbers to the minimum needed to establish an inequality

Questions