The Equational Theories Project: Advancing Collaborative Mathematical Research at Scale

Matthew Bolan, Joachim Breitner, Jose Brox, Mario Carneiro, Martin Dvořák, Andrés Goens, Aaron Hill, Harald Husum, Zoltan Kocsis, Bruno Le Floch, Lorenzo Luccioli, Douglas McNeil, Alex Meiburg, Pietro Monticone, Giovanni Paolini, Marco Petracci, Bernhard Reinke, David Renshaw, Marcus Rossel, Cody Roux, Jérémy Scanvic, Shreyas Srinivas, Anand Rao Tadipatri, **Terence Tao**, Vlad Tsyrklevich, Daniel Weber, Fan Zheng

September 4, 2025

(Tao's) Motivation

- Traditional research: small teams (1–5 experts), hard to scale
- Larger-scale collaborations face verification challenges
- Public contributions often infeasible due to complexity
- AI tools: useful, but can hallucinate → need verification

Proof Assistants as Enablers

- Proof assistants (e.g. Lean) allow modular verification
- Contributions can be verified automatically
- Enables participation from:
  - Professional mathematicians
  - General public
  - Automated tools (ATPs, AI)
- Prior successes: formalization of known results (e.g. PFR)

Towards New Mathematics

- Polymath projects: precedent for online collaboration
- Bottleneck: human moderators verifying contributions
- Adding proof assistants removes this barrier
- Goal: not just formalizing existing results, but exploring *new* mathematics collaboratively

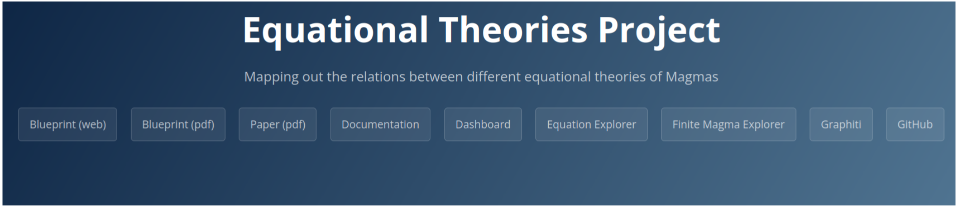
Exploring Classes of Problems

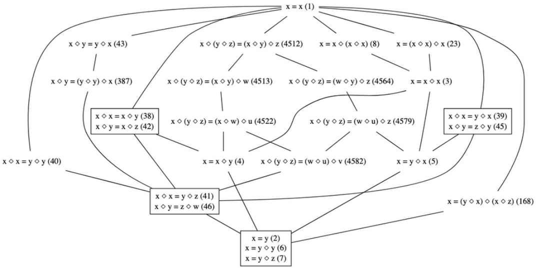
- Aim: explore *families* of mathematical problems at once
- Modular, repetitive tasks $\rightarrow$ well-suited for:
  - Crowdsourcing
  - Automated tools
- Benefits: large datasets for benchmarking, faster intuition
- Analogy: Busy Beaver Challenge, GIMPS

The Equational Theories Project (ETP)

- Pilot project testing this paradigm
- Inspired by MathOverflow question + Mastodon discussion
- Launched Sept 2024
- Goal: Determine complete implication graph of magma laws

The Equational Theories Project (ETP)



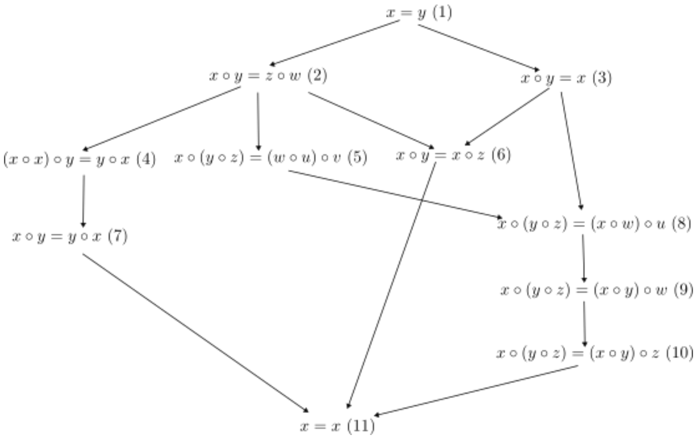


Collaboration Model

- Infrastructure:
  - Lean proof assistant
  - GitHub repository
  - Zulip chat for discussion
- Contributions:
  - Human participants (expert + public)
  - Automated tools (ATPs, AI)
- Key: modular, verifiable pieces

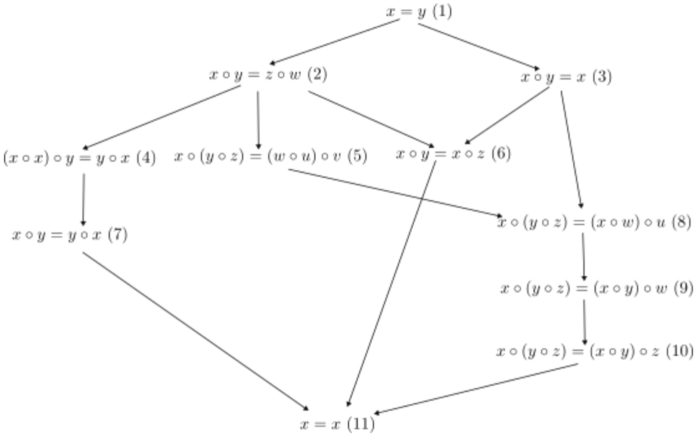
Background: Magmas and Laws

- A **magma**: a set with a binary operation $\circ$
- Equational axioms: equalities built from $\circ$ and variables
- Examples:
  - Commutativity: $x \circ y = y \circ x$
  - Associativity: $(x \circ y) \circ z = x \circ (y \circ z)$
  - Singleton: $x = y$
- Identity axiom $e \circ x = x$ excluded (involves constant e)



The Goal

- Up to trivial equivalences, there are 4694 different laws with ≤ 4 applications of $\circ$
- Define the implication graph by drawing an edge $E_n \models E_m$ if every magma obeying E_n also obeys E_n
- Goal: Determine the entire Hasse diagram
- There are $4694(4694-1) = 22\,028\,942$ implications to prove or disprove



Example entailment proof

Proposition 1 Equation4 implies Equation7.

Proof: Suppose that G obeys Equation4, thus

$$(x \circ x) \circ y = y \circ x \qquad (1)$$

for all $x, y \in G$. Specializing to $y = x \circ x$, we conclude

$$(x \circ x) \circ (x \circ x) = (x \circ x) \circ x$$

and hence by another application of (1) we see that $x \circ x$ is idempotent:

$$(x \circ x) \circ (x \circ x) = x \circ x. \qquad (2)$$

Now, replacing x by $x \circ x$ in (1) and then using (2), we see that

$$(x \circ x) \circ y = y \circ (x \circ x),$$

so in particular $x \circ x$ commutes with $y \circ y$:

$$(x \circ x) \circ (y \circ y) = (y \circ y) \circ (x \circ x). \qquad (3)$$

Also, from two applications (1) one has

$$(x \circ x) \circ (y \circ y) = (y \circ y) \circ x = x \circ y.$$

Thus (3) simplifies to $x \circ y = y \circ x$, which is Equation7. $\square$

The Goal

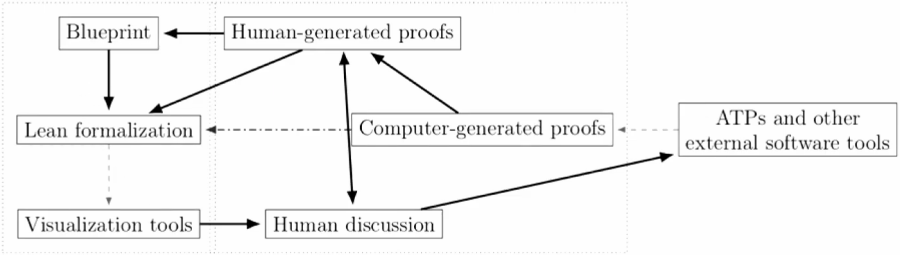
- Many of the implications or anti-implications are easy to work out by hand. But this does not scale.
- Knowing some parts of the graph can help determine others, for example by transitivity.
- The graph also has a *duality symmetry* under the involution from $(x, y) \mapsto x \circ y$ to $(x, y) \mapsto y \circ x$.

The Goal

- In general, the problem is known to be *undecidable*
- ATPs can decide many specific implications, but are not guaranteed to succeed in finite time
- A few specific laws are well studied: e.g. $x = y(z(x(yz)))$ characterizes abelian groups under subtraction. But the vast majority of laws had no established literature.
- Collaboration needed to resolve all 22 028 942 implications
- Formal verification needed to *verify* all 22 028 942 implications

Formalization Workflow

- Partially run like a regular Lean project + blueprint
- Automation + human contributions integrated via GitHub
- Zulip chat used for coordination and discussion



Methods

- Positive Implications ($E \models E'$):
  - Hand proofs
  - Rewriting rules
  - Duality & preorder structure
  - ATPs like `Vampire`, `Z3`
- Negative Implications ($E \not\models E'$):
  - Syntactic invariants
  - Small finite magmas
  - Linear / Translation-Invariant Models
  - Greedy construction methods
  - Twisting Semigroup Models
  - Ad-hoc

Zulip highlights

Equational > 1648 !=> 206

Bernhard Reinke 2:46 PM

Hello, I think I have a counterexample that shows that 1648 does not imply 206. The magma is supported on the integers and defined as follows

$$x \diamond y = x - sign(y - x)$$

In other words, $x \diamond x = x$, and $x \diamond y = x + 1$ if $x > y$, and $x \diamond y = x - 1$ if $x < y$.

Now this satisfies 1648: it is clear if $x = y$, otherwise consider the case $x > y$, then

$$(x \diamond y) \diamond ((x \diamond y) \diamond y) = (x+1) \diamond ((x+1) \diamond y) = (x+1) \diamond (x+2) = x$$

the case $x < y$ is analogous. But this magma does not satisfy 206:

indeed, for $x = 0$, $y = -1$ we have

$$(x \diamond (x \diamond y)) \diamond y = (0 \diamond 1) \diamond -1 = -1 \diamond -1 = -1 \neq 0 = x$$

🎉 4

Zulip highlights

Equational > **1648 !=> 206** OCT 14, 2024

Shreyas Srinivas EDITED 3:40 PM

Here's a long flow-of-thought proof. I think this can be golfed down significantly if `split` allowed naming the hypothesis and I used more mathlib defs:

```
import Mathlib.Tactic

class Magma (α : Type _) where
  op : α → α → α

infix:65 " ◇ " => Magma.op

abbrev Equation1648 (G: Type _) [Magma G] := ∀ x y : G, x = (x ◇ y) ◇ ((x ◇ y) ◇ y)

abbrev Equation206 (G: Type _) [Magma G] := ∀ x y : G, x = (x ◇ (x ◇ y)) ◇ y


def sign (x : ℤ) :=
  if x = 0 then 0 else if x < 0 then -1 else 1

theorem Equation1648_not_implies_Equation206 : ∃ (G: Type) (_: Magma G), Equation1648 G ∧ ¬
  let instMagmaInt : Magma ℤ := {
    op := fun x y => x - sign (y -x)
  }
  use ℤ, instMagmaInt
  simp [Equation1648, Equation206]
  constructor
  · intro x y
```

Zulip highlights

Equational > Refutations using Z3 SEP 27, 2024

Daniel Weber EDITED 7:33 PM

Using Z3 I was able to refute `Equation4283[x ∘ (x ∘ y) = x ∘ (y ∘ x)] => Equation4358[x ∘ (y ∘ z) = x ∘ (z ∘ y)]`:

x	0	1	2	3	4	5	6
0	1	1	1	4	1	6	1
1	1	1	3	1	1	1	1
2	1	5	1	1	5	1	5
3	1	1	1	1	1	1	1
4	1	1	3	1	1	1	1
5	1	1	1	1	1	1	1
6	1	1	3	1	1	1	1

With `x = 0, y = 1, z = 2`

Zulip highlights

Equational > **1076 !=> 3** — OCT 14, 2024

Daniel Weber said: — 7:08 AM

> Only the rules are important, right? Once we have them I think it should be immediate how R' should be defined, if we restrict it to only differ from R in $R'(a, b, c)$ and $R'(c, \_, \_), R'(\_, c, \_)$ (and any values we have to set to satisfy old rules should be possible to simply convert to new rules)

This is false - in 1692 there's the rule $a \diamond a = x, b \diamond a = a, a \diamond b = c \to a \diamond x = c$, I need to think for a bit how to account for that

EDITED — 8:17 AM
It succeeded on 118, 124, 476, 503, 677, 707, 713*, 883, 906, 1112, 1113, 1289*, 1447. The * are equations it seemingly succeeded on, but couldn't prove that all rules are preserved. I'll try to run it on the remaining equations with more time

Terence Tao — 4:11 PM
Thats a pretty good success rate!

4:54 PM
When you are done with your run, can you update `Conjectures.lean` with all the implications that this procedure will likely resolve (presumably in the negative), so that it is

Zulip highlights

Equational > FINITE: The Lean+Duper implications DEC 5, 2024

Amir Livne Bar-on EDITED 9:31 PM

Looking at the Duper proofs of 1167=>X in https://leanprover.zulipchat.com/#narrow/channel/458659-Equational/topic/Austin.20pairs/near/481257624, many of them go through 4689 (spelled "eq2693" in DuperConjectures1.lean). This was transitively reduced to the conjecture 1167=>4615.

Here's a proof for that one:

Equation 1167 is $L_y L_{z \diamond Sy} = I$, while equation 4615 is $L_{Sx} = L_{z \diamond x}$. For x s that are squares $x = Sa$ we have $L_{z \diamond x} = L_a^{-1}$, which is independent of z so 4615 holds.

But every element in 1167 is a square: $Sx = L_x x$, so $x = L_{z \diamond Sx} Sx$ for any z. We can take for example $a = L_{Sx \diamond Sx} Sx$ for the above.

👍 Terence Tao, Michael Bucko

Blueprint proofs

For particularly hard or intricate proofs, the blueprint was used to write a Latex version of the proof first

Now we seek to enlarge a partial solution. We first make an easy observation:

Proposition 18.13. (Enlarging L_0')✓

> *Suppose one has a partial solution in which $L_0'x$ is undefined for some $x \in N$. Then one can extend the partial solution so that $L_0'x$ is now defined.*

Proof ▾

By axiom (i''), $L_0'(R_0')^n x$ is undefined for every integer n. Let $d = E_m$ be a generator of SM that does not appear as a component of any index of any of the generators e_a appearing anywhere in the partial solution; such a d exists due to the finiteness hypotheses. We set $L_0'x := e_d$, and then extend by Equation 6, Equation 7, thus

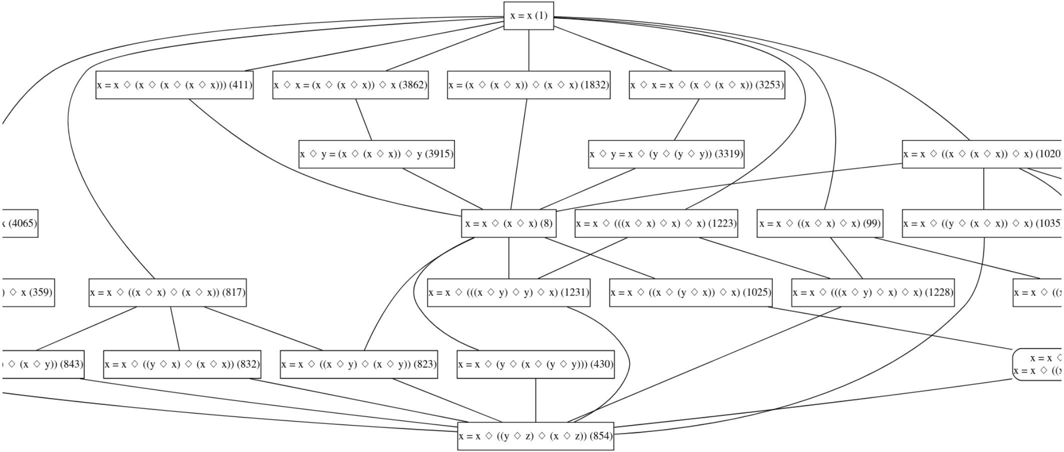
$$L_0'(R_0')^n x := (R_0')^n e_d$$

and

$$L_0'(R_0')^n e_d := (R_0')^{n-1} x.$$

Because of the new nature of d, no collisions in the partial function L_0 are created by

Visualization tools: Graphiti



Visualization tools: Equation Explorer

Equation Details

Back to List

Equation854[x = x ◇ ((y ◇ z) ◇ (x ◇ z))]

(Dual equation: Equation2712[x = ((y ◇ x) ◇ (y ◇ z)) ◇ x])
(Visualize implies and implied by of the equation, or see 1, 2, 3 graph edges away)
(Size of smallest non-trivial magma: 2 (Explore))

☑ Hide equivalent equations ☑ Treat conjectures as unknown ☐ Display the finite graph ☐ Show only explicit proofs

This equation implies (=>):

Implies	Does not imply	Unknown
Equation1[x = x] Try This! Show Proof	**Equation2[x = y] (+ 1495 equiv.)** Try This! Show Proof	None
Equation8[x = x ◇ (x ◇ x)] Try This! Show Proof	**Equation3[x = x ◇ x]** Try This! Show Proof	
Equation99[x = x ◇ ((x ◇ x) ◇ x)] Try This! Show Proof	**Equation4[x = x ◇ y] (+ 70 equiv.)** Try This! Show Proof	
Equation101[x = x ◇ ((x ◇ y) ◇ x)] (+ 1 equiv.) Try This! Show Proof	**Equation5[x = y ◇ x] (+ 70 equiv.)** Try This! Show Proof	
Equation359[x ◇ x = (x ◇ x) ◇ x] Try This! Show Proof	**Equation9[x = x ◇ (x ◇ y)] (+ 8 equiv.)** Try This! Show Proof	
Equation378[x ◇ y = (x ◇ y) ◇ y] Try This! Show Proof	**Equation10[x = x ◇ (y ◇ x)] (+ 5 equiv.)** Try This! Show Proof	

This equation is implied by (<=):

Implied by	Not implied by	Unknown by
Equation2[x = y] (+ 1495 equiv.) Try This! Show Proof	**Equation1[x = x]** Try This! Show Proof	None
Equation4[x = x ◇ y] (+ 70 equiv.) Try This! Show Proof	**Equation3[x = x ◇ x]** Try This! Show Proof	
Equation433[x = x ◇ (y ◇ (x ◇ (z ◇ y)))] Try This! Show Proof	**Equation5[x = y ◇ x] (+ 70 equiv.)** Try This! Show Proof	

Visualization tools: Finite Magma Explorer

Operation table:

```
0 1 2
1 2 0
0 1 1
```

Process

Satisfies:

- 1 - x = x
- 151 - x = (x ◇ x) ◇ (x ◇ x)
- 2441 - x = (x ◇ ((x ◇ x) ◇ x)) ◇ x

Refutes:

- 2 - x = y with {"x":0,"y":1}
- 3 - x = x ◇ x with {"x":1}
- 4 - x = x ◇ y with {"x":0,"y":1}
- 5 - x = y ◇ x with {"x":0,"y":1}
- 6 - x = y ◇ y with {"x":0,"y":1}
- 7 - x = y ◇ z with {"x":0,"y":0,"z":1}

Filter: number or expression

All implications refuted by this magma are already known.

Dashboard

The implication graph is **100.00000%** complete.

An implication is considered *explicitly true* or *explicitly false* if we have a proof of the corresponding proposition formalised in Lean. It is *implicitly true* or *implicitly false* if the proposition can be derived by taking the reflexive transitive closure of explicitly proven implications.

Our current counts of implications in each of those categories are:

explicitly true	implicitly true	explicitly false	implicitly false	no proof
10,657	8,167,622	586,925	13,268,432	0

The *no proof* column above represents work that we still need to do. Among the *no proof* implications, we have the following conjecture counts:

explicitly true	implicitly true	explicitly false	implicitly false	no conjecture
0	0	0	0	0

The implication graph is **100.00000%** complete if we include conjectures.

Dashboard

Finite graph

Some implications are true specifically only for finite magmas.

The finite implication graph is **99.99999%** complete.

explicitly true	implicitly true	explicitly false	implicitly false	no proof
10,750	8,168,349	586,220	13,268,315	2

The finite implication graph is **99.99999%** complete if we include conjectures.

explicitly true	implicitly true	explicitly false	implicitly false	no conjecture
0	0	0	0	2

Finite implication graph

- Recall: $E_n \models E_m$ if every magma obeying E_n also obeys E_n
- Let $E_n \models_{fin} E_m$ if every *finite* magma obeying E_n also obeys E_n
- This is not the same, as there are some cases where every $E_m \models E_n$ counterexample is infinite

The remaining open question

Theorem (Open problem)

Does the law E_{677}*:* $x = y \circ (x \circ ((y \circ x) \circ y))$ *imply the law* E_{255}*:* $x = ((x \circ x) \circ x) \circ x$ *for finite magmas?*

LLMs need not apply

- The ETP made extensive use of “good old-fashioned AI” in the form of ATPs
- But LLMs made only modest contributions:
  - Building visualization tools
  - code completion
  - guessing a rewriting system for a specific law from similar examples
- The implication graph appears to have some structure, such that a neural network may be able to predict it with high accuracy given a portion of the graph. This is still speculative however, and we did not use ML for this during the project.

The End

https://teorth.github.io/equational_theories