Inconsistent mathematics in Isabelle

Check out this weird logic

David Ripley

davewripley.rocks
Monash University

Table of contents

Section 1

The Weber program

Paradoxes and Inconsistent Mathematics

This project pursues the logic and set theory of Zach Weber's 2021 *Paradoxes and Inconsistent Mathematics*.

A tiny bit of history...

Weber's work takes off from the Australian(ish) greats in inconsistent maths: Sylvan, Mortensen, Meyer, Brady, Priest, Restall, etc.

Scope

The book includes work in set theory, arithmetic, algebra, analysis, topology.

For today:

Just set theory

Two axioms

This is a first-order naive set theory, with two axiom schemas: abstraction and extensionality.

Abstraction

For each variable x and formula ϕ, we have a set abstract term $\{x|\phi\}$.

Abstraction is the schema:

$$t \in \{x|\phi\} \leftrightarrow \phi[x := t]$$

Abstraction vs comprehension

From abstraction follows naive comprehension:

$$\exists y \forall x (x \in y \leftrightarrow \phi)$$

where x can be (typically is) free in ϕ.

This is enough for familiar paradoxes (Russell, Burali-Forti, etc) to kick in.

Extensionality

Extensionality is the single axiom:

$$\forall x \forall y (x = y \leftrightarrow \forall z (z \in x \leftrightarrow z \in y))$$

What's the logic?

Note that so far we have axioms, tied to $\leftrightarrow$, but no logic.

In the early 20th century, set theorists took the logic as given and fussed about paradox-proof set-theoretic axioms.

In the early 21st, we take the set-theoretic axioms as given and fuss about paradox-proof logics.

> *[W]e must, on the one hand, restrict these principles sufficiently to exclude [triviality] and, on the other, take them sufficiently wide to retain all that is valuable in this theory.*

LP set theory

The goal is not just to find any old logic in which these axioms don't trivialize.

LP has long been known as one such.

The aim here is to support sustained (more or less) ordinary reasoning.

Section 2

subDMQ

subDMQ and subDLQ

Weber gives a logic subDLQ axiomatically.

Here I use a logic subDMQ: subDMQ is certainly at least as strong as subDLQ, and if Weber's claims about subDLQ are correct, they are the same logic.

If (as I suspect), Weber has made an error about subDLQ, then subDMQ is a little stronger.

BCK

Start from BCK, with connectives $\Rrightarrow, \otimes$ for implication and conjunction.

$$(\phi \Rrightarrow \psi) \Rrightarrow (\rho \Rrightarrow \phi) \Rrightarrow \rho \Rrightarrow \psi$$
$$(\phi \Rrightarrow \psi \Rrightarrow \rho) \Rrightarrow \psi \Rrightarrow \phi \Rrightarrow \rho$$
$$\phi \Rrightarrow \psi \Rrightarrow \phi$$

$$\phi \Rrightarrow \psi \Rrightarrow \phi \otimes \psi$$
$$(\phi \Rrightarrow \psi \Rrightarrow \rho) \Rrightarrow \phi \otimes \psi \Rrightarrow \rho$$

$$\phi \Rrightarrow \psi, \phi / \psi$$

Entailment

Add an entailment $\to$, and define $\phi \leftrightarrow \psi$ as $(\phi \to \psi) \otimes (\psi \to \phi)$:

$$\phi \to \phi$$
$$(\phi \to \psi) \otimes (\psi \to \rho) \to \phi \to \rho$$
$$(\phi \to \psi) \Rrightarrow \phi \Rrightarrow \psi$$

$$\phi \otimes \psi \to \phi$$
$$\phi \otimes \psi \to \psi \otimes \phi$$
$$\phi \otimes (\psi \otimes \rho) \to (\phi \otimes \psi) \otimes \rho$$

$$(\phi \leftrightarrow \psi) \Rrightarrow (\xi(\phi) \leftrightarrow \xi(\psi))$$

Negation

Add a negation $\neg$:

$$(\phi \to \psi) \to (\neg\psi \to \neg\phi)$$
$$\neg(\phi \Rrightarrow \psi) \Rrightarrow \neg(\phi \to \psi)$$
$$\neg\neg\phi \leftrightarrow \phi$$
$$\phi \otimes \neg\psi \Rrightarrow \neg(\phi \Rrightarrow \psi)$$

Disjunction

Add a disjunction $\vee$:

$$\phi \to \phi \vee \psi$$
$$(\phi \Rrightarrow \rho) \Rrightarrow (\psi \Rrightarrow \rho) \Rrightarrow \phi \vee \psi \Rrightarrow \rho$$

$$\phi \otimes (\psi \vee \rho) \leftrightarrow (\phi \otimes \psi) \vee (\phi \otimes \rho)$$
$$\phi \vee (\psi \otimes \rho) \leftrightarrow (\phi \vee \psi) \otimes (\phi \vee \rho)$$

$$\phi \otimes \psi \leftrightarrow \neg(\neg\phi \vee \neg\psi)$$
$$\phi \vee \psi \leftrightarrow \neg(\neg\phi \otimes \neg\psi)$$

$$\phi \vee \neg\phi$$

Summary

WHEW!

- BCK implication/conjunction, with counterexample but not contraposition
- non-weakening entailment, with conjunctive syllogism and contraposition
  - does not residuate anything
  - implies implication, its negation is implied by negated implication
  - supports disjunction intro and conjunction elim
  - is used in set-theoretic axioms
- additive disjunction
- bientailment substitutes, supports commutativity, associativity, de morgan, distribution, double negation
- LEM(!?)

I'll spare you the quantificational stuff; it's largely standard.

Simplicity?

subDMQ does not have the vice of being oversimple.

It is not made for elegance; it's made to do a job:
support mathematical reasoning without crashing and burning on the paradoxes.

Nontriviality?

There is no nontriviality proof for this naive set theory.

If I had to bet, I'd bet it's nontrivial.

What about subDLQ?

The difference between this subDMQ and Weber's subDLQ is:

subDMQ includes axioms

$$(\phi \leftrightarrow \psi) \Rrightarrow (\xi(\phi) \leftrightarrow \xi(\psi))$$
$$x = y \Rrightarrow \phi(x) \rightarrow \phi(y)$$

In subDLQ, these are rules rather than implications.

Deduction theorem?

Weber claims a deduction theorem for subDLQ that would suffice, if true, to prove the extra axioms of subDMQ; on the basis of this, he claims these extra axioms as theorems of subDLQ.

I don't think it's true; the given proof certainly does not go through.

As Weber does make some use of the extra strength, I think subDMQ (which does obey a deduction theorem) is closer to the text, even though it's not the given logic.

Section 3

Computers

The importance of community

How do errors get prevented and fixed in mathematics?

Through community.

We are not quite legion

It's fair to say that the community of folks working in this set theory, or even this set theory and its relatives, is small.

This means it's easier for errors to go undetected and uncorrected.

Intuitions

Plus, this logic is both weird and complex.

So it takes time focusing specifically on this logic to build good intuitions about what's doable in it.

For almost everyone, our intuitions from other logics are deep and persuasive; errors become easy to make.

What logic is it this week?

Finally, the logic is a moving target.

I've changed subDLQ to subDMQ here to fix what I take to be an error.

But very few publications, even from Weber and collaborators, use subDLQ anyhow; the project is one of continual redesign based on experience.

Proof assistants

Proof assistants are pieces of software that can help.

The general idea

There are many different proof assistants: Lean, Coq, Isabelle, Agda, ...

Each of these implements some general logic, and provides some structured way to input proofs in that logic.

The software checks whether what's input is a correct proof, and often provides useful tools for developing proofs as well.

Isabelle examples

```
lemma implB': "(A ⇛ B) ⇛ (B ⇛ C) ⇛ A ⇛ C"
  proof -
    from implC and implB show "(A ⇛ B) ⇛ (B ⇛ C) ⇛ A ⇛ C" ..
  qed
```

```
lemma entl_antecedent_strengthening: "(A → B) ⇛ (A & C → B)"
  proof -
    from entl_impl and entl_contra have "(A → B) ⇛ (¬B → ¬A)" ..
    from this and entl_disj_inl have step1:"(A → B) ⇛ (¬B → ¬A ∨ ¬C)"
      by(rule entl_trans_10)
    from entl_impl and entl_contra have "(¬B → ¬A ∨ ¬C) ⇛ (¬(¬A ∨ ¬C) → ¬¬B)" ..
    from step1 and this have "(A → B) ⇛ (¬(¬A ∨ ¬C) → ¬¬B)" ..
    from this and dne have "(A → B) ⇛ (¬(¬A ∨ ¬C) → B)"
      by(rule entl_trans_10)
    from dm_cnd and this show "(A → B) ⇛ (A & C → B)"
      by(rule entl_trans_01)
  qed
```

Trustworthy

If a proof assistant is implemented correctly, then what it says is a proof really is a proof.

The community to worry about is the community of users of that proof assistant.

Build intuitions

Working in a proof assistant is a great way to build intuitions for an unfamiliar logic.

You quickly develop strategies and identify patterns for getting the computer to accept a proof; this is the needed intuition-building.

Adaptable

When you remove an axiom, the assistant will notice, and complain about everything that you used that axiom to prove.

So you can immediately carry over whatever it's not complaining about; no need to re-prove.

```
axiomatization
  where entlI:          "A ⟶ A"
    and entl_cs:        "(A ⟶ B) & (B ⟶ C) ⟶ (A ⟶ C)"
    and entl_contra:    "(A ⟶ B) ⟶ ¬B ⟶ ¬A"
    and entl_cel:       "A & B ⟶ A"
    and entl_conj_comm: "A & B ⟶ B & A"
    and entl_conj_ass:  "A & (B & C) ⟶ (A & B) & C"
    and entl_disj_inl:  "A ⟶ A ∨ B"
    and entl_disj_inr:  "B ⟶ A ∨ B"
    and lem:            "A ∨ ¬A"
    and dn_bi:          "¬¬A ⟷ A"
    and dm_dnc_bi:      "A ∨ B ⟷ ¬(¬A & ¬B)"
    and dm_cnd_bi:      "A & B ⟷ ¬(¬A ∨ ¬B)"
    and dist_cd_bi:     "A & (B ∨ C) ⟷ (A & B) ∨ (A & C)"
    and dist_dc_bi:     "A ∨ (B & C) ⟷ (A ∨ B) & (A ∨ C)"
    and dm_ans_bi:      "∀ P ⟷ ¬∃(%x. ¬P(x))"
    and dm_sna_bi:      "∃ P ⟷ ¬∀(%x. ¬P(x))"
    and entl_ui:        "∀ P ⟶ P(t)"
    and all_disj:       "∀(%x. A ∨ P(x)) ⟶ A ∨ ∀ P"
    and entl_impl:      "(A ⟶ B) ⟹ A ⟹ B"
    and nimpl_nentl:    "¬(A ⟹ B) ⟹ ¬(A ⟶ B)"
    and impl_cex:       "A & ¬B ⟹ ¬(A ⟹ B)"
    and implB:          "(A ⟹ B) ⟹ (C ⟹ A) ⟹ C ⟹ B"
    and implC:          "(A ⟹ B ⟹ C) ⟹ B ⟹ A ⟹ C"
    and implK:          "A ⟹ B ⟹ A"
    and impl_disj_left: "(A ⟹ C) ⟹ (B ⟹ C) ⟹ A ∨ B ⟹ C"
    and impl_conj_in:   "A ⟹ B ⟹ A & B"
    and conj_import:    "(A ⟹ B ⟹ C) ⟹ A & B ⟹ C"
```

```
axiomatization
  where entlI:             "A → A"
    (* and entl_cs:            "(A → B) & (B → C) → (A → C)" *)
    and entl_contra:       "(A → B) → ¬B → ¬A"
    and entl_cel:          "A & B → A"
    and entl_conj_comm:    "A & B → B & A"
    and entl_conj_ass:     "A & (B & C) → (A & B) & C"
    and entl_disj_inl:     "A → A ∨ B"
    and entl_disj_inr:     "B → A ∨ B"
    and lem:               "A ∨ ¬A"
    and dn_bi:             "¬¬A ↔ A"
    and dm_dnc_bi:         "A ∨ B ↔ ¬(¬A & ¬B)"
    and dm_cnd_bi:         "A & B ↔ ¬(¬A ∨ ¬B)"
    and dist_cd_bi:        "A & (B ∨ C) ↔ (A & B) ∨ (A & C)"
    and dist_dc_bi:        "A ∨ (B & C) ↔ (A ∨ B) & (A ∨ C)"
    and dm_ans_bi:         "∀ P ↔ ¬∃(%x. ¬P(x))"
    and dm_sna_bi:         "∃ P ↔ ¬∀(%x. ¬P(x))"
    and entl_ui:           "∀ P → P(t)"
    and all_disj:          "∀(%x. A ∨ P(x)) → A ∨ ∀ P"
    and entl_impl:         "(A → B) ⇛ A ⇛ B"
    and nimpl_nentl:       "¬(A ⇛ B) ⇛ ¬(A → B)"
    and impl_cex:          "A & ¬B ⇛ ¬(A ⇛ B)"
    and implB:             "(A ⇛ B) ⇛ (C ⇛ A) ⇛ C ⇛ B"
    and implC:             "(A ⇛ B ⇛ C) ⇛ B ⇛ A ⇛ C"
    and implK:             "A ⇛ B ⇛ A"
    and impl_disj_left:    "(A ⇛ C) ⇛ (B ⇛ C) ⇛ A ∨ B ⇛ C"
    and impl_conj_in:      "A ⇛ B ⇛ A & B"
    and conj_import:       "(A ⇛ B ⇛ C) ⇛ A & B ⇛ C"
```

Automatable

Proof assistants can also be automated.

Since the proofs that they check are just structured pieces of text, they can be produced by programs.

Two roles for automation:

- Automate away all the boring stuff proof assistants require of us
- Enlist automation in finding proofs

Section 4

Where things stand

Logical claims

Weber presents subDLQ first, and takes time to explore the logic and prove many useful lemmas.

In particular, he claims 32 theorems of subDLQ in his presentation of the logic.

I've got all of these proved from subDLQ in Isabelle.

Set-theoretic claims

In the chapter on set theory, there are about 100 claimed theorems.

I've got proofs of 82 of these in subDMQ in Isabelle; only 4 use the extra strength of subDMQ.

Two I think don't hold even in subDMQ; for one of these I have a weaker substitute.

Ongoing work on the remaining 16.

Next steps

Finish up Weber's set theory chapter.

Build automation.

Do some more inconsistent mathematics!

`https://bitbucket.org/davewripley/subdmq-set-theory` holds the proofs, for anyone to build on; PRs welcome!

Section 5

Metalanguage

A universal logic

Weber aims at a universal logic: a single canon of deductive validity, applicable always and everywhere:

> *[W]hen it comes to pure deductive logic, there are no concessions. Validity is a global property. If an argument form is "valid" in some domains but not others, then it is not valid. The whole point is that logic should work come what may, and we don't know what will come....*
> *In light of all this, if some contradictions are true, then nonparaconsistent logic is invalid, tout court. The same logic must be used everywhere—perhaps most importantly, in developing foundational mathematics. [p 94]*

Object language and metalanguage

Weber thus aims to avoid any separation between object language and metalanguage. He quotes Priest approvingly: "...the whole point of the dialetheic solution to the semantic paradoxes is to get rid of the distinction between object language and meta-language."

Huh?

I find this all deeply mysterious.

The object language has the syntax of a first-order logic with various connectives, plus the predicates $\in$ and $=$. The metalanguage is mathematically-flavoured English.

These are different.

Weber (and Priest and Sylvan before him) do not seem to have taken any steps to resolve this glaring difference: they do not attempt to make their object languages more like natural language, and they do not attempt to write their articles and books in the first-order languages they study.

My best stab:

It does seem like there's some relationship between the kinds of reasoning you find in mathematics journals and classical logic.

I've got no great theory of that relationship (other than that it's obviously not identity).

But maybe we don't need such a theory: maybe the idea is that Weber's own reasoning should bear the same relation, whatever it is, to subDMQ.

That's a bit inchoate, but it's my best guess.

Here's an issue

When working in a proof assistant, on the other hand, we have a clearer question in view.

There really are two formal systems at play, one embedded in the other, and they can be compared.

In the case of Isabelle, I'm working in higher-order intuitionistic logic with definitions; let's call that IHOL.

It's in IHOL that I specify: these are the axioms of the set theory and logic, and these are the rules. And what Isabelle checks is IHOL proofs from those specifications to the claim that such-and-such is provable in the specified set theory.

Why IHOL?

There are two practical reasons not to use subDMQ where Isabelle uses IHOL, plus a reason why it wouldn't matter if we did:

First, while we do have a subDMQ-based arithmetic, I don't know if this proves enough to allow us to do axiomatic proof theory in it. (Cancellation of multiplication, used in Gödel numbering, is afaik still an open question.)

Second, even if we had a well worked-out way for subDMQ to handle this, I don't have the skill or inclination to write a subDMQ-based proof assistant.

Third, even if someone did write a subDMQ-based proof assistant, they would surely write it in something not subDMQ-based, reproducing the issue up a level.

What to make of that

So everything I've done in Isabelle works in IHOL as a metalanguage. Weber would be suspicious of this.

IHOL, for example, includes a contracting conditional that obeys modus ponens. So Weber thinks that IHOL does not meet the standards of deductive validity, since deductive validity cannot allow for any such connective.

Even if results shown in Isabelle are reliable, "Mathematical 'proofs' constructed using reliable but logically invalid steps may be convincing enough arguments, but they are not proofs" (p 89).

Indirection to the rescue

But suppose this is the case: the results shown in Isabelle are reliable, although not proved.

The results shown are of the form "such-and-such has a proof in Weber's theory".

So if we believe that reliably, we believe that such-and-such does have a proof that meets Weber's standards of deductive validity.

We may not have proved that such-and-such has a proof, but demanding that is demanding too much.

Reliability

So is IHOL reliable in its reports in this domain?

I think so:

- we lack any example where it goes wrong,
- no complicated reasoning is involved, and
- we can just extract the Weber-approved proofs from the Isabelle code.

Trusting Isabelle

I think that should be enough for Weber; it's enough for me.

We have many languages and logics in the world, and we should not be scared to use them, where they are helpful.

Section 6

Conclusion

- Inconsistent mathematics is back, baby!
- subDMQ is lovely, if your tastes go baroque.
- Most of Weber's set-theoretic claims have been verified in Isabelle.
- There's more to come, and participation is welcome!
- Don't be deterred by metalanguage mismatch.