On *plenitudinous platonism*
Research Proposal
24.711 Topics in Philosophical Logic

Alexander H. Patel
`alexanderpatel@college.harvard.edu`

May 24, 2017

> "The demand is not to be denied: every jump must be barred from our deductions. That it is so hard to satisfy must be set down to the tediousness of proceeding step by step. Every proof which is even a little complicated threatens to become inordinately long. And moreover, the excessive variety of logical forms that has gone into the shaping of our language makes it difficult to isolate a set of modes of inference which is both sufficient to cope with all cases and easy to take in at a glance. To minimize these drawbacks, I invented my concept writing. [*Begriffsschrift*]" (Gottlob Frege, *Foundations of Arithmetic*, p. VI)

Contents

1 Draft Note

This document is structured as a proposal for a research project, the output of which will be a final term paper for 24.711. My intention is to take an interdisciplinary approach and rebut an argument in the philosophy of mathematics using the tools of formal verification from computer science. Formal verification is the proving or disproving of the correctness of algorithms underlying the behavior of a software or hardware system.

I will be working as a research assistant from June 5th until August 27th for Professor Margo Seltzer in the Harvard Computer Science department. The topic of her group's research is the synthesis (automatic programming) and verification of operating system kernel components. Because my 24.711 project involves research and programming within the same topic, my intention is to develop the final paper in tandem with my work in this lab because I do not have experience yet with the appropriate software tools. I expect this paper will evolve into my writing sample for PhD applications in philosophy and lead into my undergraduate thesis in philosophy and mathematics (joint concentration).

I plan to meet with Vann once a month, beginning at the start of June and ending before September 10th when the paper is due (in total four times). The objective of these meetings is to present my research findings and solicit direction for the project.

2 Research Topic

Platonism in the philosophy of mathematics defends that mathematical objects are human-independent abstract entities. *plenitudinous platonism* (hereby PP, also known as *full-blooded platonism*), first proposed by Mark Balaguer in *Platonism and Anti-Platonism in Mathematics*, is a version of mathematical platonism that holds that any mathematical object that is logically possible, necessarily exists. As a consequence, according to this doctrine every consistent theory of mathematics refers necessarily to set of existent abstract entities [1].

Balaguer argues in his book that PP is the only formulation that resolves both of Paul Benacceraf's criticims of platonism: that mathematical knowledge is impossible because mathematical objects are causally inert, and that platonism requires a commitment to unique mathematical objects and these objects are provably non-unique. Because in PP every consistent mathematical theory describes a part of the mathematical universe, our beliefs about mathematical objects consistute knowledge of those objects and so the epistemic gap is bridged. Second, Balaguer argues that PP resolves Benacerraf's non-uniqueness objection because it avoids the standard platonistic commitment to the uniqueness of mathematical objects because (all consistent theories manifest) [5].

Balauger offers in his book a partial formulation of PP in second-order modal logic:

$$(\exists x)(Mx)\&(Y)[\Diamond(\exists x)(Mx\&Yx) \rightarrow (\exists x)(Mx\&Yx)] \tag{1}$$

Here, x is a first-order variable, Y is a second-order variable, and Mx means 'x is a mathematical object'.

In "Just What is Full-Blooded Platonism?", Greg Restall argues that Balaguer's second-order formulation of PP yields a contradiction. From (1), he argues that one can derive:

$$p \supset \neg\Diamond(\exists x)(Mx\&\neg p) \tag{2}$$

Informally, (2) states that if proposition p is true then it is impossible for there to be any mathematical objects and for $\neg p$ to be true. This result is devastating for Balaguer's project, according to Restall. For example, if p is 'Cherry took a nap today' and Cherry did take a nap today, then we would conclude that it is not possible that there are both mathematical objects and that Cherry did not nap today, which is an undesirable result[2].

Restall argues that the formalization of PP ought then to be expressed in a third-order modal logic so as to restrict the scope of Y to mathematical properties alone. According to Restall, this third-order formalization yields similar undesirable results.

3 Research Objective

The objective of this research project first is to write a computer program formal verification in a logic programming language of Restall's objections against formalized PP.

By constructing this computer program so that it may be embedded in my term paper through transpilation (translation) into LaTeX, my second intention is to present a philosophy paper the contents of which are verified to be sound by virtue of having successfully compiled its LaTeX source code.

4 Research Output

The output will be a 15-20 page term paper due on September 10th, 2017. Any additional source code that is required to generate or compile this paper will be submitted in a public code repository.

5 Project Components

Modal Logic Programming Languages

There are many programming languages and software tools for formal verification and both automated theorem proving, which require no human interaction other than the specification of a goal, and assisted theorem proving, which assists a human user by "filling in the inferential blanks", per se..

Such tools are used widely in the technology industry for tasks such as verification of hardware correctness, cryptographic systems, and digital circuits. Such languages include Coq, Isabelle, HOL ("Higher-Order Logic"), and ACL2. Most of these tools are written in ML or Prolog.

I have included here a list of such tools that may serve useful for verifying the language used by Restall and Balaguer; the utility of a language ultimately will be found in the fruitfulness and efficiency of the logic proofs I am able to verify with it.

- Isabelle/HOL (`https://isabelle.in.tum.de/`) - an interactive proof assistant written in Standard ML. Isabelle natively contains a library for working with modal logics, it is unclear yet whether this supports higher-order logics (although Isabelle/HOL certainly does this very well) [11].
- Prolog (`http://www.swi-prolog.org/`)- Prolog is a general-purpose logic programming language widely used in industry. It's inference engine is capable of capturing and validating complex relationships and rule-based logical queries.
- MOLTAP (`http://twan.home.fmf.nl/moltap/related-work.html`) - an automated theorem prover for modal logic that can both construct deductions and generate Kripke models are reputations. This library does not natively support higher-order modal logics.
- ModLeanTAP (`http://formal.iti.kit.edu/beckert/modlean/`) - written in Prolog, ModLeanTAP is a tableau calculus for propositional modal logics. It does not appear that this supports higher-order modal logics natively.

Transpilation into LaTeX

Transpilation is the process of translating a computer program from one language to another. Transpiling a formal verification from a logic programming language into LaTeX appears at first glance to entail writing a mapping from strings and syntax rules in the programming language to LaTeX markup. However, proof assistants may condense the longer inferences in a given program or otherwise obfuscate the raw proofs, and so additional work will be needed here.

Isabelle/HOL contains a robust system for outputting proofs constructed within the Isabelle proof assistant to LaTeX. It is doubtful that this library would natively support the typesetting necessary to transpile modal logic proofs, and so such functionality would have to be built on top of the existing transpilation tools. Based on my understanding of the structure of Prolog, writing a transpiler for Prolog proofs is a well-defined and solved problem.

6 Research Questions

1. Is the philosophy of mathematics amenable to further formalization? What kinds of formalizations are needed to add further rigor to the landmark positions of this field?
2. Are existing formal verification tools and proof assistants equipped to express the content of arguments in the philosophy of mathematics?
3. If existing tools are not expressive enough, can they be extended so as to be able to capture the complex inferences and abstractions inherent to the philosophy of mathematics?
4. Can higher-order modal logic claims be expressed in a logic programming language? Can sentences in higher-order modal logic be reduced to sentences in first-order modal

logic or the predicate calculus? How computationally expensive is it to translate these sentences for a theorem prover be useful for?

5. Is it only possible to provide formal verifications of positive claims in higher-order model logic (syntactic proofs in the formal language), or are there tools that can refute arguments by counter-example (semantic proofs via Kripke frames)? The former of these is called *deductive verification* while the latter is called *model checking*.

6. Is Greg Restall correct that the formalization of Plenituduous Platonism presented by Balaguer yields a contradiction? Does Restall's recommendations for amending this formalization also fail, as he claims?

7 Timeline

Date	*Goal*
Early June	Develop research proposal and hone plan with Vann.
Early July	Investigate assistant and automated theorem provers for higher-order modal logic; complete language tutorials and implement example proofs;
Early August	Complete first pass formal verification of Restall's and Balaguer's dispute; figure out how to transpile proofs into LaTeX.
Late August/Early September	Paper draft review with Vann. Final term paper (15-20 pages) due September 10, 2017.

8 References

[1] Balaguer, Mark. *Platonism and Anti-Platonism in Mathematics*. Oxford University Press, New York, 1998.

[2] Restall, Greg. "Just what is Full-Blooded Platonism?" Philosophia Mathematica 11 (1):82–91 (2003).

[3] Orgun, Mehmet A. and Wanli Ma. "An Overview of Temporal and Modal Logic Programming". International Conference on Temporal Logic 1994.
`http://www-cs-faculty.stanford.edu/~uno/abcde.html`

[4] Linsky, Bernard and Edward N. Zalta. "Naturalized Platonism versus Platonized Naturalism". *The Journal of Philosophy*, Vol. 92, No. 10 (Oct., 1995), pp. 525-555
`http://www.jstor.org/stable/2940786`

[5] Colyvan, Mark and Edward N. Zalta. "Mathematics: Truth and Fiction?" Philosophia Mathematica (1999) 7 (3): 336-349.
`https://mally.stanford.edu/Papers/balaguer.pdf`

[6] Avigad, Jeremy. "Understanding, formal verification, and the philosophy of mathematics". Journal of the Indian Council of Philosophical Research, 27, 161-197
`http://repository.cmu.edu/philosophy/616/`

[7] Nipkow et al. "Isabelle/HOL: A Proof Assistant for Higher-Order Logic". Springer-Verlag, Berlin, 2016.
`http://isabelle.in.tum.de/doc/tutorial.pdf`

[8] Lambert, Leigh. "Modal Logic in Computer Science".
`http://www.cs.brandeis.edu/ cs112/cs112-2004/newReadings/ModalLogicInCS.pdf`

[9] Benzmüller et al. "Implementing and Evaluating Provers for First-Order Modal Logics" ECAI 2012.
`https://page.mi.fu-berlin.de/cbenzmueller/papers/C34.pdf`

[10] Wenzel, Makarius. "The Isabelle System Manual". 12 December 2016.
`http://isabelle.in.tum.de/doc/system.pdf`

[11] Isabelle Source Code: Sequents and Modals
`https://github.com/seL4/isabelle/tree/master/src/Sequents`