Automated Axiom Schemas with E

John Hester

Department of Mathematics, University of Florida, Gainesville, Florida, USA
https://people.clas.ufl.edu/hesterj/
hesterj@ufl.edu

Abstract. I introduce an approach for automated reasoning in first order set theories that are not finitely axiomatizable, such as ZFC, and describe its implementation alongside the automated theorem proving software E. I then compare the results of proof search in the class based set theory NBG with those of ZFC.

Keywords: ZFC · ATP · E.

1 Introduction

Historically, automated reasoning in first order set theories has faced a fundamental problem in the axiomatizations. Some theories such as ZFC widely considered as candidates for the foundations of mathematics are not finitely axiomatizable. Axiom schemas such as the schema of comprehension and the schema of replacement in ZFC are infinite, and so cannot be entirely incorporated in to the prover at the beginning of a proof search. Indeed, there is no finite axiomatization of ZFC [1].

As an alternative, I have programmed an extension to the automated theorem prover E [2] that generates instances of parameter free replacement and comprehension from well formed formulas of ZFC that are passed to it, when eligible, and adds them to the proof state while the prover is running. This allows directly reasoning in ZFC, avoiding the problems of reasoning in other theories. By using a fair algorithm for selecting replacement and comprehension instances, every possible such instance will eventually be generated given infinite time and resources. This means that refutational completeness will be preserved as long as every possible comprehension and replacement instance is eventually fed to the prover.

1.1 ZFC^o

ZFC^o, or parameter free ZFC, is an alternative axiomatization of ZFC where the schemas of comprehension and replacement have been replaced by their parameter free counterparts, and the rest of the axioms remain the same. ZFC^o is equivalent to ZFC as every instance of the full axioms of comprehension and replacement can be derived in a finite number of steps in ZFC^o [3].

Parameter Free Schema of Comprehension: Let $\phi(x)$ be any formula in the language of ZFC with a single free variable x, and let y be some variable not in ϕ. Then

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

Parameter Free Schema of Replacement: For every formula $\phi(x, y)$ of the language of ZFC,

$$\forall x \exists y \forall y' (\phi(x, y') \leftrightarrow y' = y) \rightarrow$$
$$\forall a \exists b \forall y (y \in b \leftrightarrow \exists x \in a \ \phi(x, y)).$$

2 Implementation of Axiom Schemas as Inference Rules

The schemas of parameter free replacement and parameter free comprehension can be interpreted as the below inference rules, where wff is an abbreviation for well formed formula.

$$\frac{\phi(x) \text{ is a wff} \qquad y \text{ does not occur in } \phi(x)}{\forall a \exists y \forall x (x \in y \leftrightarrow x \in a \wedge \phi(x))}$$

$$\frac{\phi(x, y) \text{ is a wff}}{\forall x \exists y \forall y' (\phi(x, y') \leftrightarrow y' = y) \rightarrow \forall a \exists b \forall y (y \in b \leftrightarrow \exists x \in a \ \phi(x, y)).}$$

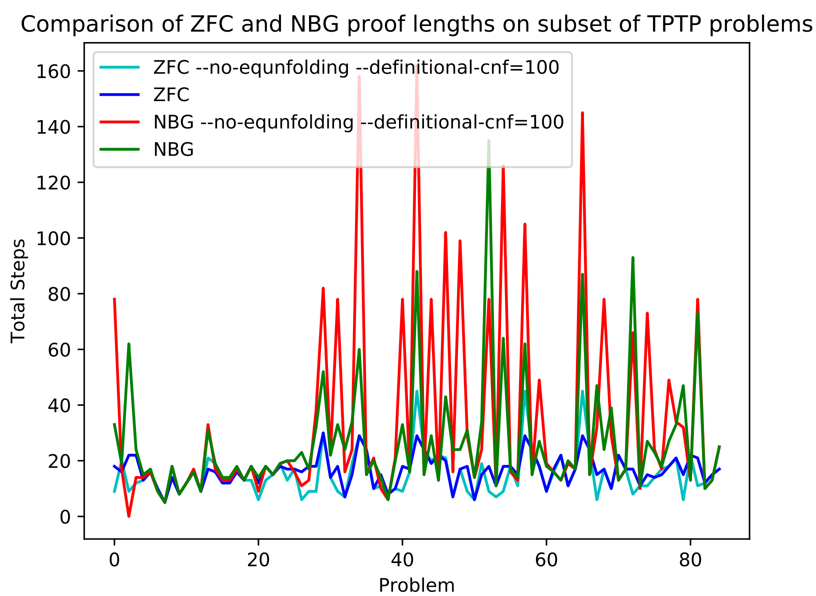
2.1 Fragmentary Approach

The approach is to generate the parameter free comprehension and parameter free replacement instances corresponding to every eligible clause generated in the proof search, then add them to the proof state. In both cases, the axiom schemas of parameter free replacement and parameter free comprehension are replaced by inference rules that take an input clause and return the corresponding replacement or comprehension instance if possible. This is easy to check, as if there is one free variable, you know there is a corresponding comprehension instance, and if there are two variables you know there are corresponding replacement inferences.

This is done by adding the clauses generated by the schema inference rules to the `tmp_store` of the proof state, which imitates the process by which new clauses are added to the collection of unprocessed clauses during a normal E proof search. While this produces ZFC proofs and benefits from the internal guidance in E by applying inference rules to the desirable clauses selected by the given clause algorithm, it has a serious downside as this will only produce a fragment of ZFC. Only applying the inference rule to the clauses generated in proof search will mean that there are many clauses and formulas that are never generated and so will never have their corresponding replacement and comprehension instances added to the proof state.

The `SET` directory of the TPTP library contains a large number of set theory problems, many of them in the language of NBG [4]. In order to compare the approaches and merits of NBG and ZFC^o, I have taken 124 of the NBG problems and corresponding definitions, and transformed them in to the language of ZFC. This mostly entails removing predicates from the NBG statements that assert certain objects are sets, as this is unnecessary in the language of ZFC, so the corresponding ZFC problems are simpler to express. From a theoretical point of view, since NBG is a conservative extension of ZFC, and ZFC^o is equivalent to ZFC, for the chosen problems every proof that is found in NBG should have a corresponding proof in ZFC^o. In all of the proof attempts described here I took the fragmentary approach described in the previous section, so full equivalence is lost.

Often, it turns out that the proofs in ZFC are shorter than corresponding proofs in NBG, sometimes much shorter. This seems to be due to the fact the axiomatization of ZFC removes the need to verify that some objects of interest are sets. Below is a graph comparing the proof lengths of $TPTP$ problems using the `--auto` mode of E in ZFC and NBG both with and without the definitional options, for the problems which at least one version of ZFC and NBG could find solutions.



3 Future Work

As there are many possibilities for comprehension and replacement instances to be added to the state, this problem can be compared to the issue of theorem proving with very large axiom lists. It would be interesting to use the approaches taken in research on automated theorem proving on large theories. In addition, machine learning approaches such as those found in Deep Network Guided Proof Search [5] and ENIGMA [6] could provide increased performance by selecting only the axiom instances that are necessary. The value of this cannot be understated as the approaches presented here add many unprocessed clauses to the state that are not necessarily helpful.

The Mizar project uses Tarski-Groethendieck set theory as its foundation, which is itself an extension of ZFC. Formal proofs available through Mizar could provide an invaluable source of training data for an automated theorem prover implementing Tarski-Groethendieck set theory in a way very similar to what is described in this paper.

4 Conclusion

ZFC theorem proving on general problems seemed to be very comparable to that of NBG in success rates, but also provided much shorter proofs in some situations. This suggests that with improved guidance functions and the full schemas of comprehension and replacement, ZFC based automated proof attempts could yield more successes than NBG. In particular, ZFC seemed to have better performance on deeper problems that dealt with more complex predicates.

5 Acknowledgements

I would like to thank Douglas Cenzer, Josef Urban, and Martin Schulz for helpful conversations that led to this experiment.

References

[1] Hillary Putnam. *Nonstandard Models and Kripke's Proof of the Gödel Theorem*. 2000.

[2] Stephan Schulz. *System Description: E 1.8, Proceedings of the 19th LPAR*. 2013.

[3] Ralf Schindler, Philipp Schlicht. *ZFC without parameters (A note on a question of Kai Wehmeier)*.

[4] G Sutcliffe. *The TPTP Problem Library and Associated Infrastructure*. 2017.

[5] Sarah Loos, Geoffrey Irving, Christian Szegedy. *Deep Network Guided Proof Search*. 2017.

[6] Jan Jakubuv, Josef Urban. *ENIGMA: Efficient Learning-based Inference Guiding Machine*. 2017.