Automated Exploration and Expansion of Large Cardinal Axioms

Pu Justin Scarfy Yang

October 12, 2024

Abstract

This proposal outlines a system for the automated generation and exploration of large cardinal hierarchies, starting from inaccessible cardinals and progressing beyond measurable and supercompact cardinals to discover even larger structures. The system continuously generates new sets of pairwise disjoint axioms, ensures consistency via automated theorem proving, and updates the cardinal hierarchy indefinitely. The project draws on principles of recursive cardinal construction and employs machine learning techniques to optimize the generation of novel large cardinal structures.

1 Introduction

Large cardinals form a crucial part of set theory, serving as the foundation for much of modern mathematical logic. Their study has led to deep insights into the structure of the mathematical universe, particularly in exploring the limits of Zermelo-Fraenkel set theory (ZFC) with the axiom of choice. From **inaccessible cardinals** to **supercompact** and **huge cardinals**, these objects help define the boundaries of mathematical consistency and universality [**?**].

The goal of this project is to create an automated system capable of generating large cardinal structures through recursive processes and continuously updating the cardinal hierarchy by discovering novel cardinal types. Using tools like automated theorem proving, we aim to generate pairwise disjoint sets of axioms that define cardinals beyond current known limits.

2 Mathematical Framework

2.1 Large Cardinals and Pairwise Disjoint Axioms

The study of large cardinals dates back to the work of Gödel and Cohen on the consistency of the continuum hypothesis and the axiom of choice [**?**, **?**]. Large cardinals, such as inaccessible and measurable cardinals, have since played a critical role in set theory, forming a central part of the cumulative hierarchy [?].

Our system begins by generating inaccessible cardinals and recursively moving beyond them to construct **Mahlo cardinals**, **measurable cardinals**, and **supercompact cardinals**. Each generated set of axioms will be pairwise disjoint, ensuring that the sets do not interfere with one another while generating distinct hierarchies of cardinals.

2.2 Recursive Cardinal Generation

Recursive methods for generating large cardinals build on embedding and reflection properties, essential in the hierarchy from measurable cardinals onwards [?]. We will extend this approach by introducing novel cardinal types based on recursive properties:

- Embedding properties for generating new measurable and supercompact cardinals [?].
- Reflection principles extending beyond supercompactness to huge and even larger cardinals [?].
- The system will recursively generate new axioms as it expands the hierarchy of cardinals, using these reflection and embedding techniques.

3 Automated Theorem Proving

Automated theorem proving has seen significant advances in recent years, particularly in formal verification systems like **Coq**, **Lean**, and **Isabelle**. These systems have been instrumental in verifying the consistency of complex mathematical structures, including theorems in number theory and combinatorics [?, ?].

Our project will use these tools to verify that each newly generated set of axioms and its corresponding cardinal hierarchy remains consistent with ZFC and the previously generated axioms. Specifically:

- **Coq** will be used to handle the formalization of axiomatic systems and verify their logical consistency [?].
- **Lean** will allow for interactive exploration of recursive cardinal generation processes, verifying the consistency of the hierarchical structures generated [?].
- **Isabelle** will be employed to ensure the correctness of large cardinal constructions and to test for contradictions in the axiom sets [?].

4 Continuous Expansion and Updating

The central goal of this project is to create a system that can run indefinitely, continuously expanding the hierarchy of large cardinals. This involves:

- Automatically generating new axiom sets and constructing corresponding cardinal structures beyond the known limits of measurable and supercompact cardinals.
- The system will use feedback from the theorem-proving tools to refine and adjust the generated axioms, ensuring consistency as it builds new cardinal types.
- Exploration of **huge cardinals** and **rank-into-rank cardinals** will serve as milestones in the system's recursive expansion, pushing the boundaries of known large cardinal theory [?, ?].

4.1 Machine Learning and Pattern Recognition

Machine learning techniques can optimize the recursive generation of large cardinals by recognizing patterns in the axioms and cardinal hierarchies generated. Drawing inspiration from similar techniques in combinatorics [?]:

- The system will apply **machine learning** to detect recurring patterns in the relationships between axioms and their corresponding cardinal structures.
- Based on these patterns, the system will propose new axiom structures and cardinal definitions to extend the hierarchy further.

5 Computational Requirements

5.1 Cloud Computing Infrastructure

Given the potentially unbounded nature of this project, the system will require extensive computational resources. Cloud platforms such as **Amazon Web Services (AWS)**, **Google Cloud Platform (GCP)**, and **Microsoft Azure** provide scalable virtual machines and persistent data storage, ideal for this project's needs.

The system will rely on:

- **AWS EC2** instances for running the recursive cardinal generation processes.
- **GCP Compute Engine** for parallel processing of theorem-proving tasks.
- Persistent storage solutions such as **AWS S3** or **GCP Cloud Storage** to store results from cardinal exploration.

5.2 High-Performance Computing (HPC)

For complex cardinal constructions and proof verifications, high-performance computing (HPC) clusters will be used. Access to national supercomputing centers or university clusters will provide the necessary computing power to handle the deep recursion involved in generating large cardinal hierarchies.

6 Conclusion

This project seeks to automate the exploration of large cardinal theory, generating, verifying, and expanding cardinal hierarchies beyond the current boundaries of mathematical knowledge. By employing recursive methods, formal verification, and machine learning, we aim to push the limits of set theory and uncover new mathematical structures in the process. The system will run indefinitely, continually updating the hierarchy of large cardinals.

References