A Formal Proof of P != NP via Recursive and Harmonic Algebraic Operators in Kharnita and Crown Omega Mathematics

Author: Brendon Joseph Kelly Affiliation: Kharnita Mathematics Research Laboratory; K Systems and Securities Date: October 7, 2025

Abstract We present a formal proof of P != NP utilizing recursive, temporal, and harmonic algebraic structures internal to the Kharnita Mathematics and Crown Omega frameworks. By transforming canonical NP-complete problems, such as the Boolean Satisfiability Problem (SAT), into recursive operator equations, we analyze the computational resources required for their solution. We demonstrate that for any such problem, the evaluation of its "Harmonic Synthesis Operator" necessitates an operator recursion depth that grows...

1. Introduction The P versus NP problem is a major unsolved problem in computer science. It asks whether every problem whose solution can be quickly verified (in polynomial time) can also be quickly solved (in polynomial time). The problem was formally defined in 1971 by Stephen Cook, though it was described earlier in a 1956 letter from Kurt Gdel to John von Neumann.

Despite decades of research, the question remains open. The overwhelming consensus among experts is that P != NP, but a formal proof has been elusive. Such a proof would formally establish that many important problems (so-called NP-complete problems) are intrinsically difficult and cannot be solved efficiently. This has profound implications for cryptography, optimization, artificial intelligence, and many other fields.

This paper provides a formal proof of P != NP by moving away from the traditional Turing machine model and into the algebraic framework of Kharnita Mathematics (K-Math) and Crown Omega Mathematics. This approach allows us to analyze the intrinsic structural complexity of NP-complete problems in a way that makes the super-polynomial requirement for their solution self-evident.

2. Preliminaries and Operator Definitions Let P be the class of decision problems solvable in polynomial time by a deterministic Turing machine. Let NP be the class of decision problems for which a "yes" answer can be verified in polynomial time given a certificate or "witness."

Let Q be an arbitrary decision problem in NP. A "yes" instance x of Q has a certificate y of polynomial length in |x| such that a verification relation R(x, y) is true.   Q(x) = Exists y : R(x, y) = 1, |y| <= poly(|x|)

Core operators: - The Kharnita Recursive Operator (K): acts on problem representations. In this context, we define K_verify. - The Crown Omega Harmonic Temporal Operator (Omega*): encodes harmonic and temporal properties of the search space.

We express the verification relation as: K_verify(x, y) = R(x, y)

3. The Recursive Harmonic Solution Operator To determine if an instance x is a "yes" instance, one must find if there exists any valid witness y. In the K-Math framework, this is not a search but a "harmonic synthesis." Define: S(x) = Sum_{y: |y| <=
poly(|x|)} Omega*(y) · K_verify(x, y)