Formalization of the Primality Algorithm

Raheleh Jalali, Ondřej Ježil

Abstract

We establish the correctness of the AKS primality testing algorithm within a formal mathematical framework known as *bounded arithmetic*. Specifically, we prove its correctness within the theory T_2^{count}, which corresponds to the first-order consequences of another well-known theory, VTC^0, when expanded with an additional mathematical function (which we call VTC^0_2).

Our approach follows two key steps:

1. Intermediate Proof in a Simpler System: We first show that the AKS algorithm works within a weaker arithmetic system, S_2^1 + iWPHP, but with two extra mathematical assumptions:
   - A generalized version of Fermat's Little Theorem.
   - A principle that ensures certain polynomial roots in finite fields can be mapped to small numbers in a structured way.
2. Final Proof in VTC^0_2: We then show that these two extra assumptions can themselves be proved within VTC^0_2, completing the proof.

To achieve this, we also develop new formalizations of key number-theoretic and algebraic results, including:

- Legendre's Formula, combinatorial number systems, and cyclotomic polynomials over finite fields, all within a framework called PV_1.
- A proof of the inequality $\text{lcm}(1, \ldots, 2n) \geqslant 2^n$ in a weaker system, S_2^1.
- A verification of the Kung–Sieveking algorithm for polynomial division within VTC^0.

1 Introduction

This work explores the feasibility of formally proving the correctness of the AKS primality testing algorithm within bounded arithmetic, a framework in proof complexity that studies the strength of formal proofs relative to computational complexity classes. In 2002, Agrawal, Kayal, and Saxena introduced the AKS algorithm [1], marking a historic breakthrough: it was the first deterministic polynomial-time primality test that relied on no unproven assumptions. This established the result that PRIMES $\in \mathbf{P}$, meaning primality testing is inherently feasible from a computational perspective. However, the formal proof

of the AKS algorithm's correctness introduces its own complexity, raising a fundamental question: How feasible is a proof of this result within a formal logical system?

The study of such formal proofs has deep roots in bounded arithmetic, which are in depth treated in [6, 3]. A growing body of research has formalized complexity-theoretic results within PV_1 and its extensions, including the work of [8, 7, 5, 2, 4]. Notably, Jeřábek previously proved the correctness of the Rabin–Miller primality test within the theory $S_2^2 + \mathrm{iWPHP(PV)} + \mathrm{PHP(PV)}$, demonstrating that probabilistic primality tests can be formally verified in restricted arithmetic. Building upon this tradition, the present work establishes that the AKS algorithm's correctness can be proved within the bounded arithmetic theory T_2^{count}.

The proof strategy follows a two-step approach: first, the correctness of AKS is established in S_2^1+iWPHP with two additional algebraic axioms: a generalized Fermat's Little Theorem and an injectivity principle for polynomial roots. Then, these axioms are shown to be provable within VTC_2^0, completing the proof. Along the way, the study formalizes important mathematical results, including Legendre's theorem, the existence of cyclotomic extensions over finite fields, a bound on least common multiples, and the Kung–Sieveking algorithm for polynomial division. These results not only strengthen our understanding of the formal proof complexity of primality testing but also contribute to the broader goal of characterizing computational mathematics within logical systems.

2 Main result

Let us start with a theorem, which is a generalization of Fermat's Little Theorem.

Theorem 2.1. If $a \in \mathbb{Z}$, $n \in \mathbb{N}$, $n \geqslant 2$ and $\gcd(a, n) = 1$, then

$$n \text{ is a prime } \iff (X + a)^n \equiv X^n + a \,(\mathrm{mod}\ n). \tag{1}$$

This suggests a basic primality test: given an input n, pick a and check if the congruence holds. However, this requires evaluating n coefficients, leading to a runtime of $\Omega(n)$ in the worst case. To improve efficiency, we can reduce the number of coefficients by evaluating both sides of (1) modulo a polynomial of the form $X^r - 1$, where r is a suitably small value. In other words: If we find r such that $\mathrm{ord}_r(n) > \log^2(n)$ and for enough a:

$$(X + a)^n \equiv X^n + a \,(\mathrm{mod}\ n, X^r - 1),$$

then n is a power of a prime. The proof mostly involves elementary results about finite fields.

2.1 Proof of Correctness of the AKS Algorithm

The proof of correctness comprises three parts. First, we need to prove the existence of r.

Theorem 2.2. Let $n \in \mathbb{N}$, then there exists $r \leqslant \max\{3, \lceil (\log n)^{O(1)} \rceil\}$ such that $\mathrm{ord}_r(n) > \log^2 n$.

Proof sketch. We use the fact that in S^1_2 that

$$\mathrm{lcm}(1, \ldots, m) \geqslant 2^{\lfloor m/2 \rfloor}.$$

□

Second, we need to show that primality is recognized.

Theorem 2.3. If n is a prime then the AKS algorithm outputs `PRIME`.

This follows immediately from generalized Fermat's theorem. Moreover, we show in VTC^0_2:

Theorem 2.4 (VTC^0_2)**.** If $a \in \mathbb{Z}$, $n \in \mathbb{N}$ a prime, $p \geqslant 2$ and $\gcd(a, n) = 1$, then

$$n \text{ is a prime } \implies (X + a)^n \equiv X^n + a \,(\mathrm{mod}\ n, X^r - 1).$$

The provability in turn follows from Jeřábek's formalization of iterated multiplication in VTC^0 [5].

Finally, we show that compositeness is recognized.

Theorem 2.5. If the AKS algorithm outputs `PRIME` on n, then n is a prime.

The proof comprises several lemmas formalizing various algebraic and number theoretic notions.

References

[1] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p. *Annals of mathematics*, pages 781–793, 2004.

[2] Lijie Chen, Zhenjian Lu, Igor C Oliveira, Hanlin Ren, and Rahul Santhanam. Polynomial-time pseudodeterministic construction of primes. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1261–1270. IEEE, 2023.

[3] Stephen Cook and Phuong Nguyen. *Logical foundations of proof complexity*, volume 11. Cambridge University Press Cambridge, 2010.

[4] Azza Gaysin. Proof complexity of universal algebra in a csp dichotomy proof. *arXiv preprint arXiv:2403.06704*, 2024.

[5] Emil Jeřábek. Iterated multiplication in VTC^0. *Archive for Mathematical Logic*, 61(5):705–767, 2022.

[6] Jan Krajíček. *Bounded arithmetic, propositional logic and complexity theory*, volume 60. Cambridge University Press, 1995.

[7] Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Annals of Pure and Applied Logic*, 171(2):102735, 2020.

[8] Ján Pich. Logical strength of complexity theory and a formalization of the pcp theorem in bounded arithmetic. *Logical Methods in Computer Science*, 11, 2015.