▶ RAHELEH JALALI, ONDŘEJ JEŽIL, *Correctness of the AKS primality algorithm in Bounded Arithmetic.*
Department of Computer Science, University of Bath.
E-mail: rahele.jalali@gmail.com.
Faculty of Mathematics and Physics, Charles University.
E-mail: firma.ondra@gmail.com.

We establish the correctness of the AKS primality testing algorithm [1] within a formal mathematical framework known as *bounded arithmetic* [3, 2, 5]. 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, $S^1_2+\text{iWPHP}$, 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 $\text{lcm}(1,\ldots,2n) \geq 2^n$ in a weaker system, S^1_2.
- A verification of the Kung–Sieveking algorithm for polynomial division within VTC^0.

This work investigates the formal proof of the AKS primality test within bounded arithmetic, a framework linking proof complexity and computational classes. While AKS proved PRIMES $\in$ **P**, formally verifying its correctness poses new challenges. Building on prior work in PV_1 and related systems, we show that AKS can be proved correct in T_2^{count}. The proof proceeds via $S^1_2+\text{iWPHP}$ with two algebraic axioms—later shown provable in VTC^0_2, and formalizes key results from number theory and algebra, advancing our understanding of computational mathematics in logical frameworks.

Main result

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

THEOREM 1. *If* $a \in \mathbb{Z}$, $n \in \mathbb{N}$, $n \geq 2$ *and* $\gcd(a,n) = 1$, *then*

$$n \text{ is a prime} \iff (X+a)^n \equiv X^n + a \pmod{n}. \quad (1)$$

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 $\text{ord}_r(n) > \log^2(n)$ and for enough a:

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

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

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. *Let* $n \in \mathbb{N}$*, then there exists* $r \leq \max\{3, \lceil (\log n)^{O(1)} \rceil\}$ *such that* $ord_r(n) > \log^2 n$.

PROOF SKETCH. We use the fact that in S^1_2 that

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

$\dashv$

Second, we need to show that primality is recognized.

THEOREM 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 4 (VTC^0_2). *If* $a \in \mathbb{Z}$*,* $n \in \mathbb{N}$ *a prime,* $p \geq 2$ *and* $\gcd(a, n) = 1$*, then*

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

The provability in turn follows from Jeřábek's formalization of iterated multiplication in VTC^0 [4].

Finally, we show that compositeness is recognized.

THEOREM 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.

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

[2] Samuel R. Buss. Bounded Arithmetic. PhD thesis, Princeton University, 1985.

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

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

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