A Deterministic Polynomial-Time Primality Test Based on Binomial Coefficient Divisibility

Abstract

This paper describes a deterministic primality test based on the divisibility properties of binomial coefficients modulo a candidate integer n. The test checks the condition $\binom{n}{k} \equiv 0 \pmod{n}$ for all k in the range $1 \leq k \leq \lfloor \log_2 n \rfloor$. This criterion is a direct consequence of the polynomial identity $(x+1)^n \equiv x^n + 1 \pmod{n}$, which holds if and only if n is prime. The algorithm uses an efficient recurrence relation, achieves a time complexity of $\widetilde{O}(\log^3 n)$, and correctly identifies composite numbers, including Carmichael numbers.

1 Introduction

Primality testing is a foundational problem in computational number theory. While probabilistic tests are efficient, deterministic tests like the AKS algorithm established that the problem is in P. This work presents a test based on elementary properties of binomial coefficients modulo n, offering a deterministic algorithm with polynomial-time complexity.

2 The Freshman's Dream and Primality

The core idea of the test is rooted in a classic polynomial identity known in finite fields as the "Freshman's Dream."

Theorem 1 (The Freshman's Dream)**.** Let n be a prime number. Then for any integer a in the ring of integers modulo n, the following identity holds:

$$(x+a)^n \equiv x^n + a \pmod{n}.$$

This theorem is a direct application of Fermat's Little Theorem and the divisibility of binomial coefficients by primes. The congruence is to be interpreted as a polynomial congruence, meaning the coefficients of like powers of x are congruent modulo n.

Expanding the left-hand side using the binomial theorem yields:

$$(x+a)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} x^k.$$

The right-hand side is:

$$x^n + a.$$

For the two polynomials to be congruent modulo n, all coefficients for x^k where $1 \leq k \leq n-1$ must be divisible by n. Specifically, for the case $a = 1$, we require:

$$\binom{n}{k} \equiv 0 \pmod{n} \quad \text{for all } k \text{ such that } 1 \leq k \leq n-1. \tag{1}$$

Thus, the Freshman's Dream identity holds for a prime n. Conversely, if Equation 1 holds for all k in the specified range, then the polynomial identity $(x+1)^n \equiv x^n + 1 \pmod{n}$ is true. A fundamental result, underpinning tests like AKS, is that this polynomial identity holds *if and only if* n is prime.

3 The Efficient Primality Criterion

Checking the condition for all k from 1 to $n-1$ is computationally infeasible. The key insight is that for a composite number n, a violation of this identity can be detected by examining only a small number of coefficients.

Theorem 2 (Polynomial Coefficient Criterion)**.** Let $n > 1$ be an integer. Let $L = \lfloor \log_2 n \rfloor$. Then n is prime *if and only if*:

$$\binom{n}{k} \equiv 0 \pmod{n} \quad \text{for all } k \text{ such that } 1 \leq k \leq L.$$

Proof Sketch. ($\Rightarrow$) If n is prime, the Freshman's Dream holds, so Equation 1 is true for all k, which includes the range $1 \leq k \leq L$.

($\Leftarrow$) This is the non-trivial direction. If n is composite, assume for contradiction that $\binom{n}{k} \equiv 0 \pmod{n}$ for all $1 \leq k \leq L$. This would imply that the polynomial congruence $(x+1)^n \equiv x^n + 1 \pmod{n}$ holds for all terms of degree less than or equal to L. However, a result from the proof of the AKS theorem shows that if n is composite, this congruence must fail for some term of degree k, and crucially, that this k will be less than a bound that is $O(\log^c n)$. The specific bound $L = \lfloor \log_2 n \rfloor$ is chosen to be large enough to guarantee this witness k is found. Thus, the assumption leads to a contradiction, proving that for some $k \leq L$, $\binom{n}{k} \not\equiv 0 \pmod{n}$. □

3.1 The Algorithm

The algorithm checks the condition in Theorem 2 efficiently using a recurrence relation.

Algorithm 1 Deterministic Primality Test via Binomial Coefficients

Require: Integer $n > 1$
Ensure: Returns **true** if n is prime, **false** otherwise.

1: $L \leftarrow \lfloor \log_2 n \rfloor$
2: $C \leftarrow 1$ ▷ This will represent $\binom{n}{k}$ for the current k
3: **for** $k \leftarrow 1$ to L **do**
4: $C \leftarrow C \cdot (n-k+1) \cdot \text{inv}(k,n) \mod n$ ▷ Recurrence: $C_k = C_{k-1} \cdot \frac{n-k+1}{k} \mod n$
5: **if** $C \not\equiv 0 \pmod{n}$ **then**
6: **return false**
7: **end if**
8: **end for**
9: **return true**

In line 4, $\text{inv}(k,n)$ denotes the modular inverse of k modulo n, computed using the Extended Euclidean Algorithm.

4 Complexity Analysis

The loop runs $L = O(\log n)$ times. The dominant operation in each iteration is the modular multiplication and the computation of the modular inverse $\text{inv}(k,n)$. Using standard algorithms, these operations on $O(\log n)$-bit numbers take $O(\log^2 n)$ time. The total time complexity is:

$$O(\log n) \times O(\log^2 n) = O(\log^3 n).$$

This can be expressed with soft-O notation as $\widetilde{O}(\log^3 n)$.

5 Conclusion

This primality test is a practical application of the Freshman's Dream identity. By leveraging efficient computation and a logarithmic bound on the coefficients to check, it provides a deterministic polynomial-time algorithm. Its simplicity, relying solely on modular arithmetic, makes it a valuable method for primality testing.