Primality Testing: From Fermat's Theorem to the AKS Algorithm

Aman K. Foujdar
Ashoka University
aman.foujdar_ug2023@ashoka.edu.in

October 10, 2024

Abstract

This article is an exposition of the Agrawal–Kayal–Saxena (AKS) Primality Test. I will first lay the groundwork, using illustrations and examples, necessary to understand the 'main' proof. I will then present the proof of the 'main' AKS Theorem 3.1 and an implementation using Sagemath.

1 Introduction

Primality testing, a fundamental aspect of number theory and cryptography, determines whether a given number is prime. It has been the basis of many widespread encryption and decryption systems, including the famous RSA encryption.

There are several methods to determine the primality of a number. We divide them into two major categories: probabilistic and deterministic. The Miller–Rabin test for primality testing uses Fermat's little theorem and is probabilistic in nature. It identifies prime numbers by first choosing a random number a to produce the expected output. Meanwhile, the Agrawal–Kayal–Saxena (AKS) Primality Test is deterministic in nature.

The AKS primality testing algorithm, introduced in 2002, revolutionized the field by providing the first polynomial-time deterministic algorithm for primality testing. This algorithm guarantees a definitive answer. Therefore, it maintains the accuracy standard required for high-stakes applications in cryptography. The AKS algorithm relies on the congruence properties of cyclotomic polynomials.

The objective of this article is to explain the background material required to understand the AKS algorithm. I will prove only the 'main' part of the algorithm, and illustrate the ideas by means of examples. I also provide the Sagemath code to implement the algorithm.

Our primary references for this article are the seminal paper of Agrawal, Kayal, and Saxena [2] and the comprehensive book of Rempe–Gillen and Waldecker [3]. The book [3] provides a useful introduction to Fermat's Theorem and the Miller–Rabin test.

2 Understanding P, NP, and AKS

In this section, I will establish the notation and theorems we need to understand the main AKS theorem.

2.1 Complexity of an Algorithm

Problems that can be solved by an algorithm are known as decidable problems. It is often challenging, and sometimes even impossible, to develop an algorithm that solves a given mathematical problem.

Even if we have an algorithm to solve a decidable (or computable) problem, we must also consider its efficiency. The complexity of an algorithm is a function that gives us a measure of its 'efficiency' in terms of the amount of data the algorithm processes. We will use the concept of asymptotic growth to measure the efficiency.

The asymptotic growth of a function $f(n)$ can be described using the Big O notation.

Definition 2.1 (Big O notation)**.** *We say $f(n) = O(g(n))$ if there exist positive constants c and n_0 such that for all $n \geq n_0, |f(n)| \leq c|g(n)|$.*

2.2 P and NP Class

An algorithm is considered efficient if its running time function $s(n)$ is polynomial, that is, $s(n) = O(n^k)$ for some positive integer k. The class of decision problems for which efficient algorithms exist is called **P** or *Polynomial Time*. Problems in P can be solved by a deterministic machine in polynomial time. This means that the time required increases at most as some fixed power of n.

NP stands for *Non-deterministic Polynomial Time*. This class comprises decision problems that can be solved by a non-deterministic machine in polynomial time. It includes problems where, although finding a solution may be difficult, verifying a given solution can be done in polynomial time.

For those interested in a more detailed understanding of these concepts, numerous resources provide thorough explanations beyond the scope of this paper. Readers are encouraged to consult these resources for further study.

Remark 2.2. In practical applications, time complexities beyond n^6 are generally considered impractical for large inputs. This is because such high-order polynomial time complexities can lead to long computation times, making them unsuitable for most real-world scenarios.

2.3 Setup for AKS

In this section, I will introduce some key mathematical concepts and theorems needed for the 'main' proof. It includes the order of a number modulo n, Euler's Totient function, the generalization of Fermat's Little Theorem for polynomials, the notion of irreducibility, and cyclotomic polynomials.

Definition 2.3. ***(Order modulo n).*** *Let n be a positive integer and let a be an integer such that $\gcd(a, n) = 1$. The **order of** a **modulo** n, denoted by $ord_n(a)$, is defined as the smallest positive integer d such that*

$$a^d \equiv 1 \pmod{n}.$$

In other words, d is the smallest positive integer for which the congruence relation holds true.

Example 2.4. *Let's find the order of $a = 3$ modulo $n = 7$.*

According to the definition, we need to find the smallest positive integer d such that:

$$3^d \equiv 1 \pmod{7}.$$

Let's compute the powers of 3 modulo 7:

$$\begin{aligned}
d = 1 : &\quad 3^1 \equiv 3 \pmod 7,\\
d = 2 : &\quad 3^2 = 9 \equiv 2 \pmod 7,\\
d = 3 : &\quad 3^3 = 27 \equiv 6 \pmod 7,\\
d = 4 : &\quad 3^4 = 81 \equiv 4 \pmod 7,\\
d = 5 : &\quad 3^5 = 243 \equiv 5 \pmod 7,\\
d = 6 : &\quad 3^6 = 729 \equiv 1 \pmod 7.
\end{aligned}$$

We see that $3^6 \equiv 1 \pmod 7$, and there is no smaller positive integer for which this is true. Therefore, the order of 3 modulo 7 is $ord_7(3) = 6$.

Definition 2.5. ***(Set of Numbers Coprime to** n**).*** *The set $cp(n)$ denotes the collection of all integers from 1 to $n-1$ that are coprime to n. It is defined as:*

$$cp(n) = \{k \in \mathbb{Z} \mid 1 \leq k < n \text{ and } \gcd(k, n) = 1\}.$$

In other words, $cp(n)$ includes all integers k that are less than n and have no common factors with n other than 1.

Definition 2.6. ***(Euler's Totient Function).*** *Euler's Totient function, denoted by $\phi(n)$, is defined as the number of positive integers up to n that are co-prime to n. Simply put, $\phi(n)$ is the number of elements of the set $cp(n)$.*

Lemma 2.7. *The middle binomial coefficient $\binom{2n}{n}$ grows at least exponentially, that is,*

$$\binom{2n}{n} \geq 2^n$$

Proof. We will use induction and the following properties of binomial coefficients:

$$\binom{n+l}{k} \geq \binom{n}{k} \quad \text{and} \quad \binom{n+l}{k+l} \geq \binom{n}{k}.$$

Base Case:
For $n = 0$, we have:

$$\binom{2 \cdot 0}{0} = \binom{0}{0} = 1 \quad \text{and} \quad 2^0 = 1$$

Thus, the inequality $\binom{2n}{n} \geq 2^n$ holds for $n = 0$.
Inductive Step:
Assume the inequality holds for some $n \in \mathbb{N}$, that is,

$$\binom{2n}{n} \geq 2^n$$

We need to show that the inequality holds for $(n+1) \in \mathbb{N}$

$$\binom{2(n+1)}{n+1} = \binom{2n+2}{n+1}$$

Using Pascal's identity of binomial coefficients and properties mentioned above, we have:

$$\binom{2n+2}{n+1} = \binom{2n+1}{n} + \binom{2n+1}{n+1}$$

$$\binom{2n+2}{n+1} \geq \binom{2n}{n} + \binom{2n}{n}$$

By the induction hypothesis:

$$\binom{2n}{n} \geq 2^n$$

So:

$$\binom{2n+2}{n+1} \geq 2^n + 2^n = 2 \cdot 2^n = 2^{n+1}$$

Thus, we have shown that:

$$\binom{2(n+1)}{n+1} \geq 2^{n+1}$$

This completes the induction step and proves that $\binom{2n}{n} \geq 2^n$ for all $n \in \mathbb{N}$. □

Remark 2.8. Pascal's identity states that:

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

Definition 2.9. *Let $f(x)$ be a non-constant rational polynomial, that is, $f(x) \in \mathbb{Q}[x]$. The polynomial $f(x)$ is said to be* ***irreducible*** *over $\mathbb{Q}$ if it does not have any non-trivial divisors over $\mathbb{Q}$. $f(x)$ is irreducible over $\mathbb{Q}$ if, for any polynomials $g(x)$ and $h(x)$ in $\mathbb{Q}[x]$, whenever $f(x) = g(x) \cdot h(x)$, either $g(x)$ or $h(x)$ must be a constant polynomial.*

Given my assumption that the reader is already familiar with Fermat's Little Theorem, I will now focus on generalizing Fermat's theorem to polynomials. Readers are encouraged to consult several comprehensive resources that provide detailed proofs and explanations of Fermat's Little Theorem.

Remark 2.10. **Binomial Theorem** states that

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

Remark 2.11. **Fermat's Little Theorem** states that if p is a prime number and a is an integer not divisible by p, then

$$a^p \equiv a \pmod{p}.$$

Lemma 2.12. ***(Divisors of binomial coefficients).***
If n is prime and $1 \leq k \leq n-1$, then

$$\binom{n}{k} \equiv 0 \pmod{n}.$$

Visualizing Lemma 2.12 using Pascal's Triangle:
Consider Pascal's Triangle for $n = 5$. Here is how the binomial coefficients will look.

$$\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1
\end{array}$$

In this triangle, notice that for $1 \leq k \leq 4$ (that is, $k = 1, 2, 3, 4$), $\binom{5}{k}$ is divisible by 5. Specifically, $\binom{5}{1} = 5$, $\binom{5}{2} = 10$, $\binom{5}{3} = 10$, and $\binom{5}{4} = 5$, all of which are divisible by 5.

Proof. Recall:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Multiplying both sides by the denominator of RHS:

$$k!(n-k)!\binom{n}{k} = n!. \tag{1}$$

Following the expansion of n! we know that n is a divisor of the RHS of equation 1 and hence also divides the LHS. Since n is prime number and each of the numbers $1, 2, \ldots, k$ is less than n, it follows that n does not divide $k!$. For the same reason, n does not divide $(n-k)!$. So, it follows that n must divide the final term in the product on the RHS which proves our point that $n \mid \binom{n}{k}$. □

Theorem 2.13. ***(Fermat for polynomials).*** *Let p be a prime number. Then*

$$(P(X))^p \equiv P(X^p) \pmod{p} \tag{2}$$

for all polynomials P with integer coefficients.

Remark 2.14. It is important to note that polynomial congruence is defined in terms of congruence of their coefficients, rather than the values of the polynomials. Thus, this generalization of the Fermat's theorem require a separate proof.

Proof. **Base Case:** If $x = 0$, then P is constant. This holds true from Fermat's little theorem.
Induction Hypothesis: Suppose the theorem hold true for all integer polynomials of degree $x = k$.
Inductive Step: We need to show that the claim holds for the polynomial P of degree $x = k+1$. Let

$$P = aX^{k+1} + Q(X) \tag{3}$$

where $Q(X)$ is a polynomial of degree at most k and there exists some $a \in \mathbb{Z}$.

We can now expand on $P(X)^p$ using the binomial theorem (Remark 2.10).

$$(P(X))^p = (aX^{k+1} + Q(X))^p \tag{4}$$

$$= (aX^{k+1})^p + \left(\sum_{n=1}^{p-1} \binom{p}{n} (aX^{k+1})^n (Q(X))^{p-n} \right) + (Q(X))^p. \tag{5}$$

We know that

$$(aX^{k+1})^p = a^p (X^{k+1})^p = a^p X^{p(k+1)} \equiv a(X^p)^{k+1} \pmod{p},$$

using Fermat's Theorem (Remark 2.11). By lemma 2.12, every binomial coefficients $\binom{p}{k}$ in the summation of equation 5 is divisible by p, which implies that the sum is congruent to zero modulo p. And $(Q(X))^p \equiv Q(X^p) \pmod{p}$ holds true by the induction hypothesis. □

Definition 2.15. ***(Congruence Modulo*** n ***and*** $H(x)$***).***

Let $n \geq 2$ be a positive integer, and let $H(x)$ be a non-constant polynomial whose leading coefficient is coprime to n. We say that two polynomials $P(x)$ and $Q(x)$ are congruent modulo n and $H(x)$, denoted as:

$$P(x) \equiv Q(x) \pmod{(n, H(x))}$$

if $P(x)$ and $Q(x)$ yield the same remainder when divided by $H(x)$ with coefficients reduced modulo n.

Example 2.16. *Let $n = 5$ and $H(x) = x^2 + 1$. Consider the polynomials $P(x) = 3x^2 + 4x + 2$ and $Q(x) = 8x^2 + 9x + 7$.*

Polynomial Divisions:

$$P(x) = (x^2 + 1) \cdot 3 + (4x - 1)$$
$$Q(x) = (x^2 + 1) \cdot 8 + (9x - 1)$$

Remainders:

- *For $P(x)$, the remainder is $4x - 1$.*
- *For $Q(x)$, the remainder is $9x - 1$.*

$$4x - 1 \equiv 4x + 4 \pmod{5}$$
$$9x - 1 \equiv 4x + 4 \pmod{5}$$

Since both remainders are congruent modulo 5, we have:

$$P(x) \equiv Q(x) \pmod{(5, x^2 + 1)}$$

Theorem 2.17. ***(Number of Polynomial Zeros Modulo** p **and** H**)*** *Let p be a prime number and let H be a polynomial that is irreducible modulo p. Furthermore, let P be a polynomial of degree $d \geq 0$ modulo p. Then P has at most d polynomial zeros that are pairwise not congruent modulo p and H.*

Lemma 2.18. *Let P be a polynomial. If $P \not\equiv 0 \pmod{n}$, then P has a decomposition*

$$P \equiv (X - a_1) \cdots (X - a_m) \cdot Q \pmod{n}.$$

Here, $m \geq 0$, the numbers $a_1, \ldots, a_m$ range from 0 to $n - 1$, and Q is a polynomial that has no zeros modulo n. If n is prime, then the numbers $a_1, \ldots, a_m$ are unique up to reordering. In other words, if

$$P \equiv (X - b_1) \cdots (X - b_k) \cdot R \pmod{n}$$

is another such decomposition, with $b_1, \ldots, b_k$ in $\{0, \ldots, n - 1\}$, then $m = k$ and the b_j agree with the a_i up to their ordering.

Remark 2.19. The proof of Theorem 2.17 is beyond the scope of this paper. Lemma 2.18 can be easily proven with basic induction. For comprehensive proofs, refer to [3].

Definition 2.20. ***(Cyclotomic Polynomial).*** *The* ***cyclotomic polynomial*** $\Phi_n(x)$ *is defined as the unique monic polynomial with integer coefficients whose roots are precisely the primitive n-th roots of unity. It is given by:*

$$\Phi_n(x) = \prod_{\substack{1 \leq k \leq n \\ \gcd(k,n)=1}} \left(x - e^{2\pi i k/n} \right),$$

where the product is over all integers k such that $1 \leq k \leq n$ and $\gcd(k, n) = 1$.

Think of the cyclotomic polynomials this way: similar to how a natural number can be broken down into its prime factors, a polynomial can be decomposed into a product of irreducible polynomials over $\mathbb{Q}$, which are the polynomial analogs of prime factors. These irreducible/minimal polynomials are the product of the primitive roots of the same order.

Example 2.21. *Let $n = 4$. The primitive fourth roots of unity are the complex numbers $e^{2\pi i/4} = i$ and $e^{-2\pi i/4} = -i$. The cyclotomic polynomial $\Phi_4(x)$ is:*

$$\Phi_4(x) = (x - i)(x + i).$$

Expanding this product:

$$\Phi_4(x) = x^2 - i^2 = x^2 - (-1) = x^2 + 1.$$

So, the cyclotomic polynomial for $n = 4$ is:

$$\Phi_4(x) = x^2 + 1.$$

3 The Theorem of AKS

This section will present the Agrawal-Kayal-Saxena (AKS) algorithm and offer a proof of the 'main' AKS theorem. Additionally, I will include a working code implementation of the AKS algorithm using SageMath.

3.1 The AKS Algorithm

Below is the original AKS algorithm designed to determine whether a given integer $n > 1$ is prime or composite.

```
Input: integer $n > 1$.

    1. If ($n = a^b$ for $a \in \mathbb{N}$ and $b > 1$), output COMPOSITE.
    2. Find the smallest $r$ such that $o_r(n) > \log^2 n$.
    3. If $1 < (a, n) < n$ for some $a \leq r$, output COMPOSITE.
    4. If $n \leq r$, output PRIME.
    5. For $a = 1$ to $\left\lfloor \sqrt{\varphi(r)} \log n \right\rfloor$ do
           if $((X + a)^n \not\equiv X^n + a \pmod{X^r - 1, n})$, output COMPOSITE;
    6. Output PRIME;
```

3.2 The Core Theorem

Theorem 3.1. ***(Theorem of Agrawal, Kayal, and Saxena).***
Let $r \in \mathrm{cp}(n)$ be a prime number with $\mathrm{ord}_r(n) > 4(\log n)^2$. Also set $Q := X^r - 1$. If n is not a power of p, then there are at most r polynomials of the form $P = X + a$, with $a \in \{0, \ldots, p-1\}$, that satisfy

$$(P(X))^n \equiv P(X^n) \pmod{p, Q}. \tag{6}$$

I will first prove some subsidiary theorems and lemmas that are required for the proof of the core theorem.

Lemma 3.2. *Let $r \in \mathbb{N}$ and let $Q := X^r - 1$, $m \in \mathbb{N}$, and $n \geq 2$. Then:*

(a) $Q(X^m) = X^{rm} - 1 \equiv 0 \pmod{Q}$; and

(b) if P is a polynomial such that $P \equiv 0 \pmod{n, Q}$, then also $P(X^m) \equiv 0 \pmod{n, Q}$.

Proof. Let's start with the first statement. It says that $X^r - 1$ is a divisor of $X^{rm} - 1$. This can be understood by the following simple calculation:

$$X^{rm} - 1 = (X^r - 1)\left(X^{(m-1)r} + X^{(m-2)r} + \cdots + X^r + 1\right).$$

(This expression can be derived using polynomial long division.)

Now, for the second part. Suppose we have a polynomial P and an integer n such that P is divisible by $Q = X^r - 1$ modulo n. This means there is another polynomial R such that:

$$P \equiv R \cdot Q \pmod{n}.$$

Now, let's substitute X with X^m in this congruence:

$$P(X^m) \equiv R(X^m) \cdot Q(X^m) \pmod{n}.$$

This shows that $Q(X^m)$ divides $P(X^m)$ modulo n.

Finally, from the first part of the lemma, we know that Q divides $Q(X^m)$. Therefore, Q also divides $P(X^m)$ modulo n, which is what we needed to prove. □

Definition 3.3. *For the remainder of this paper, let $n \geq 2$ be a positive integer, and let p be a prime divisor of n. Fix some $r \in \mathbb{N}$ and define $Q := X^r - 1$. Let $\mathcal{P}$ denote the set of all integer polynomials P (of any degree) that satisfy the congruence (6).*

Lemma 3.4. *Let P be a polynomial, and let m_1 and m_2 be natural numbers such that*

$$(P(X))^{m_1} \equiv P(X^{m_1}) \pmod{p, Q}$$

and

$$(P(X))^{m_2} \equiv P(X^{m_2}) \pmod{p, Q}.$$

If we set $m := m_1 \cdot m_2$, then it also holds that

$$(P(X))^m \equiv P(X^m) \pmod{p, Q}. \tag{7}$$

Proof. For our proof, we are given that:

$$(P(X))^{m_1} \equiv P(X^{m_1}) \pmod{p, Q}.$$

We can use Lemma 3.2 (part (b)) to extend this result. We substitute X with X^{m_2} in the congruence:

$$(P(X^{m_2}))^{m_1} \equiv P((X^{m_2})^{m_1}) \pmod{p, Q}.$$

This simplifies to:

$$(P(X^{m_2}))^{m_1} \equiv P(X^{m_1 \cdot m_2}) \pmod{p, Q}.$$

Since furthermore $(P(X))^{m_2} \equiv P(X^{m_2})$, we conclude that:

$$(P(X))^{m_1 \cdot m_2} = ((P(X))^{m_2})^{m_1} \equiv (P(X^{m_2}))^{m_1} \equiv P(X^{m_1 \cdot m_2}) \pmod{p, Q},$$

as claimed.

□

Corollary 3.5. *Let $P \in \mathcal{P}$. Then the congruence (7) holds for every number m of the form $m = n^i \cdot p^j$, where $i, j \geq 0$.*

Lemma 3.6. *Let $P_1, P_2 \in \mathcal{P}$. Then $P_1 \cdot P_2 \in \mathcal{P}$.*

Proof. From the definition of $\mathcal{P}$, it follows immediately that

$$(P_1 \cdot P_2)(X^n) = P_1(X^n) \cdot P_2(X^n) \equiv (P_1(X))^n \cdot (P_2(X))^n = ((P_1 \cdot P_2)(X))^n \pmod{p, Q}.$$

□

Definition 3.7. *From here onwards, let l denote the number of elements $a \in \mathbb{N}_0$ with $a \leq p-1$ for which the polynomial $X + a$ belongs to $\mathcal{P}$. We define H as an irreducible factor of Q modulo p, A as the number of elements of $\mathcal{P}$ that are pairwise distinct modulo p and H, and t as the number of polynomials of the form $X^{n^i \cdot p^j}$, where $i, j \geq 0$, that are pairwise distinct modulo p and H.*

Lemma 3.8. *Let P_1 and P_2 be polynomials in P whose degree is smaller than t. If $P_1 \equiv P_2 \pmod{p, H}$, then also $P_1 \equiv P_2 \pmod{p}$.*

Proof. By Corollary 3.5 and the given hypothesis, we have the following congruence relation:

$$P_1(X^m) \equiv (P_1(X))^m \equiv (P_2(X))^m \equiv P_2(X^m) \pmod{p, H}$$

for every integer m of the form $m = n^i \cdot p^j$. This implies that, for every such m, the polynomial X^m is a zero of the polynomial

$$T := P_1(Y) - P_2(Y) \pmod{p, H}.$$

According to the definition of t, the polynomial T must have at least t distinct polynomial zeros modulo p and H. However, by our hypothesis, the degree of T is less than t.

Theorem 2.17 states that a polynomial with degree less than t cannot have t distinct zeros modulo p and H. Therefore, the only way for T to satisfy this condition is if T is identically zero modulo p. This concludes the proof.

□

Theorem 3.9. ***(Lower bound on A in terms of t and l.)*** *Remember, A is the number of elements of $\mathcal{P}$ that are pairwise distinct modulo p and H. The number A satisfies*

$$A \geq \binom{t+l-1}{t-1}.$$

Proof. Let $k < t$, and let $a_1, a_2, \ldots, a_k \in \{0, 1, \ldots, p-1\}$ such that the polynomials $X + a_1, \ldots, X + a_k$ are in $\mathcal{P}$. According to Lemma 3.6, the product of these polynomials, denoted T, is also in $\mathcal{P}$ and has a degree less than t.

By Lemma 2.18, the polynomials $X + a_1, \ldots, X + a_k$ are uniquely determined by the polynomial T, up to reordering.

Now, consider another set of $k' < t$ elements, such that $b_1, b_2, \ldots, b'_k \in \{0, 1, \ldots, p-1\}$ such that $X + b_1, \ldots, X + b_{k'}$ are also in $\mathcal{P}$. Since k' is less than t, then it follows that either the coefficients a_i and b_j agree up to reordering or the product of these polynomials, T', is not congruent to T modulo p. In the latter case, by Lemma 3.8, T and T' are also not congruent modulo p and H.

We need to determine how many distinct polynomials of degree at most t can be formed from a set of l different polynomials of degree 1 by taking their products. This is equivalent to counting the number of ways, order is irrelevant, to choose up to t elements from l options. In this case, we're allowing for repetition.

The number of possible distinct polynomials is given by the binomial coefficient:

$$\binom{t+l-1}{t-1}.$$

□

Theorem 3.10. *If n is not a power of p, then $A \leq \frac{n^{2\sqrt{t}}}{2}$.*

Proof. We know that if a polynomial $P \in \mathcal{P}$, then the congruence (3.2) holds for for every number m of the form $m = n^i \cdot p^j$, where $i, j \geq 0$. If n is not a power of p, then for different combinations of i and j, the product $n^i \cdot p^j$ will yield different values of m. If we require that $0 \leq i, j \leq \lfloor \sqrt{t} \rfloor$, then there are exactly $(\lfloor \sqrt{t} \rfloor + 1)^2 > t$ such choices.

By the definition of t, we can find two numbers of the form $m = n_i \cdot p_j$, which we will call m_1 and m_2, such that

$$X^{m_1} \equiv X^{m_2} \pmod{p, H}. \tag{8}$$

Assume $m_1 > m_2$. Then we have:

$$m_2 < m_1 \leq (n \cdot p)^{\sqrt{t}} \leq \frac{n^{2\sqrt{t}}}{2}.$$

Here, we used the fact that p is a non-trivial divisor of n, which implies $p \leq \frac{n}{2}$.

For any polynomial $P \in \mathcal{P}$, it follows from the congruence (8) and Corollary 3.5 that:

$$(P(X))^{m_1} \equiv P(X^{m_1}) \equiv P(X^{m_2}) \equiv (P(X))^{m_2} \pmod{p, H}.$$

Therefore, every polynomial $P \in \mathcal{P}$ is a polynomial zero of $R := Y^{m_1} - Y^{m_2}$ modulo p and H. The polynomial R has degree m_1 modulo p, and in particular, $R \not\equiv 0 \pmod{p}$. By Theorem 2.17, the number of polynomial zeros of R modulo p and H is bounded above by m_1. This implies:

$$A \leq m_1 \leq \frac{n^{2\sqrt{t}}}{2}.$$

□

Remark 3.11. Since i and j can independently take any integer value from 0 to $\lfloor t \rfloor$, there are $\lfloor t \rfloor + 1$ possible values for i and the same for j. Therefore, the total number of distinct m values is given by the product of the number of choices for i and j, which is $(\lfloor t \rfloor + 1)^2$.

Corollary 3.12. *If $t > 4(\log n)^2$ and $l \geq t-1$, then n is a power of p.*

Proof. By the hypothesis and Theorem 3.9, we have the following inequality for A:

$$A \geq \binom{t+l-1}{t-1} \geq \binom{2(t-1)}{t-1} \geq 2^{t-1} = \frac{2^t}{2}.$$

using Lemma 2.7.

From the hypothesis, we know that $t > 4(\log n)^2$. It follows that

$$t = \sqrt{t} \cdot \sqrt{t} > 2\sqrt{t} \log n.$$

Thus,

$$\frac{2^t}{2} > \frac{2^{2\sqrt{t}\log n}}{2} = \frac{n^{2\sqrt{t}}}{2}$$

This shows that A is greater than $n^{2\sqrt{t}}/2$. By Theorem 3.10, this implies that n must be a power of p. □

Definition 3.13. *The expression $Q := X^r - 1$ can be factored as*

$$X^r - 1 = (X - 1) \cdot (X^{r-1} + X^{r-2} + \cdots + X + 1).$$

Here, the polynomial

$$K_r := X^{r-1} + X^{r-2} + \cdots + X + 1 \tag{9}$$

*is called the r-**th cyclotomic polynomial**.*

Lemma 3.14. *Let p and r be prime numbers with $p \neq r$, and let H be an irreducible factor (modulo p) of the r-th cyclotomic polynomial K_r. Then*

$$X^r \equiv 1 \pmod{p, H}$$

and

$$X^k \not\equiv 1 \pmod{p, H} \quad \textit{for all } k \in \{1, \ldots, r - 1\}.$$

Proof. We begin by noting that $X^r \equiv 1 \pmod{X^r - 1}$, and since H is a divisor of $X^r - 1$ modulo p, it follows that

$$X^r \equiv 1 \pmod{p, H}.$$

Now, let $k \geq 1$ be the smallest integer such that

$$X^k \equiv 1 \pmod{p, H}.$$

Before we proceed further, we need to show that in this case k divides r, as this will be crucial for the main proof.

We can express r in terms of k as follows:

$$r = c \cdot k + r_0,$$

where $c \in \mathbb{Z}$ and $0 \leq r_0 < k$. Substituting this into the equation for X^r, we have

$$1 \equiv X^r = X^{c \cdot k + r_0} = (X^k)^c \cdot X^{r_0} \equiv X^{r_0} \pmod{p, H}.$$

Since $r_0 < k$ and k is the smallest such number, it follows that $r_0 = 0$. Therefore, $r = c \cdot k$, meaning k divides r.

Given that r is prime, k must be either r or 1.

Assume, for contradiction, that $k = 1$. This would imply that $X - 1 \equiv 0 \pmod{p, H}$. Since H is irreducible modulo p, it would follow that $H \equiv X - 1 \pmod{p}$. Consequently, $X - 1$ would be a divisor of K_r modulo p, which implies $K_r(1) \equiv 0 \pmod{p}$.

However, we know that

$$K_r(1) = 1 + 1 + \cdots + 1 = r \not\equiv 0 \pmod{p},$$

which leads to a contradiction. Therefore, $k = r$ as desired. □

Corollary 3.15. *Let r and p be prime numbers with $r \neq p$, and let H be an irreducible factor (modulo p) of the r-th cyclotomic polynomial K_r. Also, let $n \in \mathbb{N}$ be any multiple of p such that* $\gcd(n, r) = 1$. *Then $\mathrm{ord}_r(n) \leq t \leq r$.*

Proof. We know that, t denotes the number of distinct polynomials of the form X^m that are distinct modulo p and H, where m varies over all possible products of powers of n and p.

The order $\mathrm{ord}_r(n)$ is defined as the number of pairwise different elements of the form $m = n^j$ modulo r. By Lemma 3.14, if m_1 and m_2 are distinct powers of n modulo r, then we have $X^{m_1} \not\equiv X^{m_2} \pmod{p, H}$. This proves the first inequality: $t \geq \mathrm{ord}_r(n)$.

Lemma 3.14 also states that there can be at most r polynomials of the form X^m that are different modulo p and H. Hence, the total number t cannot exceed r, confirming that $t \leq r$.

Combining these two results, we conclude that

$$\mathrm{ord}_r(n) \leq t \leq r.$$

□

Proof of Theorem 3.1 (Theorem of Agrawal, Kayal, and Saxena). This proof is as given in [3].

Let n and p be as before, where p is a prime number and n is a multiple of p. Also, let r be a prime number that is coprime to n and satisfies $\mathrm{ord}_r(n) > 4(\log n)^2$. Since r is coprime to n, we have $r \neq p$.

From Corollary 3.15, we know that

$$4(\log n)^2 < t \leq r,$$

where t is the number of distinct polynomials X^m modulo p and H.

Now, let l denote the number of integers $a \in \{0, \ldots, p-1\}$ for which $X + a$ satisfies the condition in 6. Suppose that $l \geq r$. Then, we have $l \geq t \geq t - 1$. By Corollary 3.12, n must be a power of p.

This proves our main theorem. □

Remark 3.16. The original article by Agrawal, Kayal, and Saxena proves a slightly stronger version of Theorem 3.1. In particular, "at most r" in the statement can be replaced by "at most $2\sqrt{r}\log n$".

3.3 SageMath Implementation of AKS

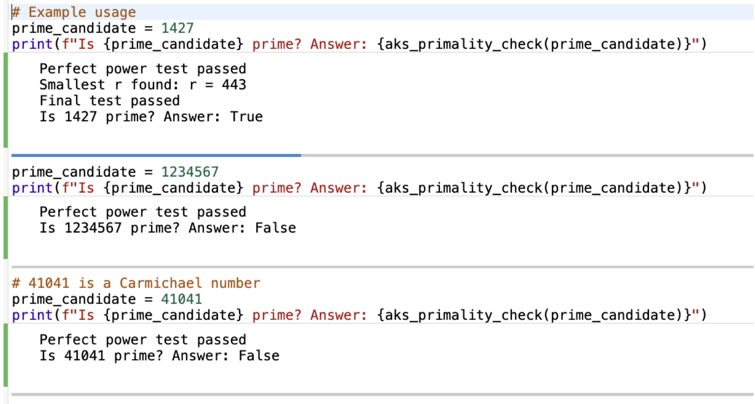
In this section, I will present a working code implementation of the AKS primality test using SageMath.

```
# SageMath necessary imports
from sage.all import *

# Efficient method for exponentiation
def quick_exponentiation(base, exponent):
    """
    Quickly computes base raised to the power of exponent
    """
    result = 1
    while exponent > 0:
        if exponent % 2 == 1:
            result *= base
        base *= base
        exponent //= 2
    return result

# Function to check if a number is a perfect power
def check_perfect_power(num):
    max_exponent = 1 + (len(bin(abs(num))) - 2)
    for exponent in range(2, max_exponent):
        low, high = 0, num
        while low < high - 1:
            mid = (low + high) // 2
            mid_power = quick_exponentiation(mid, exponent)
            if mid_power > num:
                high = mid
            elif mid_power < num:
                low = mid
            else:
                return True  # num is a perfect power: num = mid^exponent
    return False

# Implementation of the AKS Primality Test
def aks_primality_check(num):
    if num == 1:
        return False  # 1 is not prime
    if num == 2:
        return True   # 2 is prime

    # Check if num is a perfect power
    if check_perfect_power(num):
        return False
    print("Perfect power test passed")

    # Determine the smallest r such that ord_r(num) > 4 * (log2(num))^2
    max_order = 4 * (log(num, 2))^2
    for r in range(2, num):
        if gcd(r, num) > 1:
            return False  # num is composite
        multiplicative_order = mod(num, r).multiplicative_order()
        if multiplicative_order > max_order:
            break

    print(f"Smallest r found: r = {r}")
```

```
if r == num:
    return True  # num is prime

# Create a polynomial ring over Z/numZ with a variable x
ZnX.<x> = IntegerModRing(num)[]
# Create the quotient ring by (x^r - 1)
ZnX_quotient.<xbar> = ZnX.quotient(x^r - 1)

#The Freshman Dream test for numbers that might otherwise slip through the
    earlier stages.
limit = int(2 * sqrt(r) * log(num, 2)) + 2
for j in range(1, limit):
    if ZnX_quotient((xbar + j)^num) != ZnX_quotient(xbar^num + j):
        print(f"Freshman dream test failed at j = {j}")
        return False

print("Final test passed")
return True
```

Remark 3.17. For reference, I looked at the SageMath implementation by Jsevillamol [1]. This implementation served as a valuable resource in the development of my program.

I will now check for primality using my program:



```
# Example usage
prime_candidate = 1427
print(f"Is {prime_candidate} prime? Answer: {aks_primality_check(prime_candidate)}")
    Perfect power test passed
    Smallest r found: r = 443
    Final test passed
    Is 1427 prime? Answer: True

prime_candidate = 1234567
print(f"Is {prime_candidate} prime? Answer: {aks_primality_check(prime_candidate)}")
    Perfect power test passed
    Is 1234567 prime? Answer: False

# 41041 is a Carmichael number
prime_candidate = 41041
print(f"Is {prime_candidate} prime? Answer: {aks_primality_check(prime_candidate)}")
    Perfect power test passed
    Is 41041 prime? Answer: False
```

Figure 1: Primality Testing using our algorithm

3.4 Time Complexity and Improvements

Until now, we've focused on proving the main theorem of the AKS Primality Test. However, to establish the correctness of the entire algorithm, it is necessary to explicitly prove each step of the algorithm beyond the main theorem. These proofs are relatively simple and concise compared to the main proof. Interested readers can refer to the original AKS paper [2] or see [3].

The asymptotic time complexity of the AKS algorithm is $\tilde{O}(\log^{21/2} n)$. This indicates that the algorithm is polynomial-time in the size of the input. For a rigorous proof of this time complexity, see [2].

Following the work of Agrawal, Kayal, and Saxena, researchers proposed several improvements to the original algorithm. Notably, Lenstra and Pomerance developed an algorithm with improved running time compared to the original AKS algorithm. While it builds on the same core idea, it uses a different polynomial Q for testing congruences.

Despite these advancements, no deterministic primality test has yet surpassed the efficiency of the Miller–Rabin test. This ongoing gap presents exciting opportunities for further research and development in primality testing.

Acknowledgements

I would like to express my sincere gratitude to Gaurav Bhatnagar, Manjil Saikia, and Sagar Shrivastava for help with writing this article.

References

[1] Jsevillamol. *Implementing the AKS Primality Test*. https://ask.sagemath.org/question/44275/implementing-the-aks-primality-test/. 2019.

[2] Neeraj Kayal Manindra Agrawal and Nitin Saxena. "PRIMES is in P". In: *Annals of Mathematics* 160.2 (2004), pp. 781–793. DOI: 10.4007/annals.2004.160.781. URL: https://www.cse.iitk.ac.in/users/manindra/algebra/primality_v6.pdf.

[3] Lasse Rempe-Gillen and Rebecca Waldecker. *Primality Testing for Beginners*. Vol. 70. Student Mathematical Library. American Mathematical Society, 2014. URL: https://www.ams.org/bookpages/stml-70.