國立臺灣大學理學院數學系

碩士論文

Department of Mathematics

College of Science

National Taiwan University

Master Thesis

多項式時間確定型質數判定演算法的研究

On the AKS Algorithm

曾膺任

Ying-Jen Tseng

指導教授：陳其誠 博士

Advisor: Ki-Seng Tan, Ph.D.

中華民國 105 年 1 月

January 2016

目　錄

誌謝

我要感謝指導教授陳其誠老師給我這個研究方向，讓我從中體會到數學理論在質數判定這個重要的問題裡起了什麼樣的作用。在和老師討論論文裡各種的細節的時候，老師總是很有耐心地替我解說，幫我重拾起許多重要的代數觀念，使我獲益良多。老師除了教導我們數學，平日也時常與我們聊天，關心我們的日常生活。老師看似總以一種幽默態度面對周遭，跟在老師身邊，讓我不只學習數學，更多的時候，也許在潛移默化中學到老師面對人生的那股豁達精神。回顧碩班三年多來和老師相處的時光，我覺得自己很幸運能成為老師的學生，這些年來我成長了很多，不管是在數學還是其他方面，我由衷的感謝陳其誠老師！

我也要感謝這幾年在台大認識的好朋友，建鑫，啟樺，俊飛，昶凱，偉碩，黃瑞，楷倫，還記得大家同住男一那段充滿歡笑的回憶嗎？感謝家榮在我碩一的時候提供房間讓我入住，那裡的環境和你讓我有一種家的感覺，到現在還會想起。感謝為淵那陣子的深夜電影約會，讓我學術之餘有紓壓的管道，也感謝你和芳如這幾年的陪伴。感謝金緯帶我進入數學的世界，過去每每和你討論完總是充滿鬥志，你是數學路上的好同志。感謝勇賢在碩班時找我討論修課科目，一起準備考試。你們每一位參與著我的生活，就像拼圖一樣，拼出了我這幾年多采多姿的人生，我感謝你們！

最後我要感謝我的父母，是你們無怨無悔把我帶大，重視教育，賺錢供我讀書，默默地支持我的選擇，當我受挫時在旁鼓勵我，安慰我。你們用一雙手撐起這個溫暖的家庭，到我長大點才發現原來這並不容易。你們永遠是我心靈最深處的依靠，謝謝，我愛你們！

中文摘要

本文研究由 M. Agrawal, N. Kayal and N. Saxena 提出的第一個多項式時間確定型的質數判定演算法，經過 H. Lenstra Jr.等人的建議修改後的版本”PRIMES is in P”(2004)，並補充了一些原文裡證明細節。

關鍵詞：質數；演算法；多項式時間；確定型；質數判定

英文摘要

We take a exposition at the paper “PRIMES is in P” by M. Agrawal, N. Kayal and N. Saxena (2004), in which they used Lenstra's idea and made a revision of their earlier version. We also present some details in the proof.

Key Words: prime number; algorithm; polynomial time; deterministic; primality test

ON THE AKS ALGORITHM

Ying-Jen Tseng

ABSTRACT. We investigate the AKS algorithm which determines whether a number is prime in polynomial time.

1. INTRODUCTION

In August 2002, M. Agrawal, N. Kayal and N. Saxena proposed an unconditional, deterministic and polynomial time primality test. It is now known as AKS test or AKS algorithm. Prior to then, several efficient primality test had been founded. Miller test (proposed in 1975, deterministic and polynomial time assuming the Extended Riemann Hypothesis); Rabin-Miller test(proposed in 1980, unconditional but in randomised polynomial time); Soloray-Strassen test(proposed in 1977, in randomised polynomial time); Adleman-Pomerance-Rumely test(proposed in 1983, deterministic and in $(\log n)^{O(\log\log\log n)}$ time); Goldwasser-Kilian test(proposed in 1986, in randomised expected polynomial time); Atkin-Adleman-Huang test (proposed in 1992, also in randomised polynomial time).

The AKS test finally achieved the desired polynomial runtime requirement using only fully proved facts. It not only settled the theoretical issue of primality test but also stunned the world with its simplicity. Many of the previous algorithms used deeper result while the AKS algorithm utilised simpler tools and acquire more efficient runtime condition. Soon after the AKS algorithm proposed, some variant algorithms had also been founded, two of them are in [LEN05] and [BER03], both proving primality in better asymptotic running time.

In this thesis, we take a exposition at the paper "PRIMES is in P" [AKS04] by M. Agrawal, N. Kayal and N. Saxena (2004), in which they used Lenstra's idea and made a revision of their earlier version [AKS02]. Some of the structure and arrangement of the content are from [RC05] and [AG05].

We begin with an observation which is the idea that the AKS algorithm is based on.

2. Basic Idea

We shall let Z_n denote the ring $\mathbb{Z}/n\mathbb{Z}$.

Lemma 2.0.1. *Let a be an integer, n be a positive integer, $n \geq 2$, and $(a,n) = 1$. Then n is a prime if and only if*

$$(x+a)^n \equiv x^n + a \pmod{n}. \tag{1}$$

Proof. The coefficient of x^i in $(x+a)^n - (x^n + a)$ is $\binom{n}{i}$. If n is a prime, then $\binom{n}{i} = 0 \pmod{n}$. If n is composite and q is a prime factor of n such that $q^k || n$, then since

$$\binom{n}{q} = \frac{n(n-1)\cdots(n-q+1)}{q!}$$

and $q^k || n(n-1)\cdots(n-q+1)$, q^k can not divide $\binom{n}{q}$, whence n can not divide $\binom{n}{q}$. The coefficient of x^q is $a^{n-q}\binom{n}{q}$, which can not be divided by n since $(a,n) = 1$. So $(x+a)^n - (x^n + a)$ is not identically zero over Z_n. $\square$

The above criterion is not efficient enough to be polynomial-time: to verify (1) directly, one needs to compute all terms of the left-hand side and the computation takes $O(n)$ time. A probable way of solving the problem is to modulo a polynomial $f(x)$ on both side of (1). In particular, if (1) is satisfied, then the congruence

$$(x+a)^n \equiv x^n + a \pmod{x^r - 1, n} \tag{2}$$

will also be satisfied. If the degree r is not so large (bounded by an polynomial function of $\log n$), then we can check (2) quickly (in polynomial time). However, while (2) is necessary for n to be prime, it is not sufficient. And it seems that this is the main difficulty to overcome if one wants a fast algorithm that derived from criterion (1). It turns out M. Agrawal, N. Kayal and N. Saxena managed to resolve this kind of difficulty: they can restore the characterization by verifying (2) for every a up to a certain point, if (2) is satisfied for all of these a, then n must be a prime power, which can be detected efficiently from the very beginning. The degree r is also appropriately chosen to assure each (2) can be verified in polynomial time, hence the total run time of their algorithm is polynomial time. Now we state the algorithm as pseudo code in the next section, after which is its correctness proof followed by analysis of time complexity.

3. The algorithm

Input: integer $n > 1$.

1. If n is a perfect power, return COMPOSITE.

2. Find the least integer r such that the order of n in Z_r^* exceeding $\log^2 n$.

3. If $a|n$ for some $2 \leq a \leq \sqrt{\phi(r)}\log n$, return COMPOSITE.

4. For $2 \leq a \leq \sqrt{\phi(r)}\log n$, if $(x+a)^n \neq x^n + a \pmod{x^r - 1, n}$, return COMPOSITE.

5. Return PRIME.

4. Correctness of the Algorithm

Theorem 1. *The algorithm returns PRIME if and only if n is prime.*

The proof of Theorem 1 is split into two parts: Lemma 4.0.2 and Theorem 2, dealing with the case which returns COMPOSITE and PRIME, respectively.

Lemma 4.0.2. *If the algorithm returns COMPOSITE, then n is composite.*

Proof. If the algorithm returns COMPOSITE from step 1 or 3, then clearly n is composite. Otherwise, COMPOSITE is returned from step 4, then by Theorem 1, n cannot be prime, thus n is also composite. $\square$

The case that returns PRIME requires more efforts, and it serves as the main criterion in the algorithm. Some authors call this criterion "AKS Theorem". Before proceeding on, we need some lemmas.

In the rest of the context, let R denote the ring $Z_p[x]/(x^r-1)$.

Lemma 4.0.3. *Let p be a prime number and r be a positive integer co-prime to p. Let $T: R \longrightarrow R$ be defined by $T(f) = f^p$. Then T is injective.*

Proof. Suppose there are u, v belonging to R such that $T(u) = T(v)$. Then

$$0 = T(u) - T(v) = u^p - v^p = (u-v)^p = T(u-v).$$

Let $w = u - v$. It suffices to prove $w = 0$. Write $w = a_0 + a_1x + \cdots + a_{r-1}x^{r-1}$, so that in R,

$$\begin{aligned} 0 &= T(w) \\ &= w^p \\ &= (a_0 + a_1x + \cdots + a_{r-1}x^{r-1})^p \\ &= a_0 + a_1x^p + \cdots + a_{r-1}x^{(r-1)p}. \end{aligned}$$

If $x^{ip} = x^{jp}$ in R for some nonnegative integer i, j, then $r|p(i-j)$, which means $r|i-j$ due to $(r,p) = 1$. Since $0, 1, 2, ..., r-1$ are all distinct modulo r, the terms $1, x^p, x^{2p}, ..., x^{(r-1)p}$ are actually the rearrangement of $1, x, x^2, ..., x^{r-1}$, and hence $a_0 + a_1x^p + \cdots + a_{r-1}x^{(r-1)p} = 0$ implies $a_0 = a_1 = \cdots = a_{r-1} = 0$. That is, $w = 0$. Therefore, T is injective. $\square$

Let $\overline{f(x)} \in R$ denote the residue class of $f(x) \in Z_p[x]$ and let m be a non-negative integer. If $\overline{g(x)} = \overline{f(x)}$, then $g(x) = f(x) + h(x)(x^r - 1)$, hence

$$g(x^m) = f(x^m) + h(x^m)(x^{rm} - 1).$$

Since $x^{mr} - 1$ is divisible by $x^r - 1$, we see that $\overline{g(x^m)} = \overline{f(x^m)}$. In other words, the residue class $\overline{f(x^m)}$ depends only on $\overline{f(x)}$ and is independent of the choice of $f(x)$. Hence, we have a well-defined operator

$$E_m : R \longrightarrow R,\ \ \overline{f(x)} \mapsto \overline{f(x^m)}.$$

In particular, if p is a prime number, then since $f(x^p) = f(x)^p$ for every $f(x) \in Z_p[x]$, we can conclude that for every $\overline{f(x)} \in R$,

$$\overline{f(x^p)} = \overline{f(x)}^p.$$

Theorem 2. (AKS Theorem) *Suppose n is an integer with $n \geq 2$, r is a positive integer with $(r,n) = 1$ and the order of n in Z_r^* is larger than $\log^2 n$. Moreover, assume that*

$$(x+a)^n = x^n + a \pmod{x^r - 1, p} \tag{3}$$

holds for integer a with $0 \leq a \leq \sqrt{\phi(r)}\log n$. If n has a prime factor $p > \sqrt{\phi(r)}\log n$, then $n = p^m$ for some positive integer m. If n has no prime factor in the interval $[1, \sqrt{\phi(r)}\log n]$ and n is not a perfect power, then n is prime.

Proof. Suppose n has a prime factor $p > \sqrt{\phi(r)}\log n$. Denote

$$G = \{g(x) \in Z_p[x] : g(x)^n = g(x^n) \pmod{x^r - 1}\}.$$

By (3), we know that $x + a \in G$, for all $0 \leq a \leq \sqrt{\phi(r)}\log n$. Since G is closed under multiplication, if each e_a is a nonnegative integer, then the product

$$\prod\nolimits_{0 \leq a \leq \sqrt{\phi(r)}\log n} (x+a)^{e_a} \in G.$$

Let $\bar{G} \subset R$ denote the set of all residue classes of $f \in G$ modulo $x^r - 1$. Then

$$\bar{G} = \{\overline{f(x)} \in R \mid \overline{f(x^n)} = \overline{f(x)}^n\}.$$

For each $\overline{f(x)} \in \bar{G}$, denote $v := \overline{f(x^{n/p})}$ and $w := \overline{f(x)}^{n/p}$. Then

$$v^p = \overline{f(x^n)} = \overline{f(x)}^n = w^p.$$

Thus, by Lemma 4.0.3, we have $v = w$. Let m_1 and m_2 be positive integers such that

$$\overline{f(x^{m_1})} = \overline{f(x)}^{m_1} \text{ and } \overline{f(x^{m_2})} = \overline{f(x)}^{m_2}$$

in R. Then there is $q(x) \in Z_p[x]$ satisfying

$$f(x)^{m_2} = f(x^{m_2}) + q(x)(x^r - 1)$$

in $Z_p[x]$. Substitute x with x^{m_1} and get

$$f(x^{m_1})^{m_2} = f(x^{m_1 m_2}) + q(x^{m_1})(x^{m_1 r} - 1)$$

in $Z_p[x]$. Note that $x^r - 1 | x^{m_1 r} - 1$, and hence in R

$$\overline{f(x^{m_1 m_2})} = \overline{f(x^{m_1})}^{m_2} = \overline{f(x)}^{m_1 m_2}.$$

Define

$$I = \{p^i (n/p)^j \mid i, j \geq 0\}.$$

From the above it has been shown for every $m \in I$ and every $g(x) \in G$, $\overline{g(x)}^m = \overline{g(x^m)}$ in R.

Let $Q_r(x)$ be the r^{th} cyclotomic polynomial over the finite field Z_p. Then $Q_r(x) | x^r - 1$ and $Q_r(x)$ factors into irreducible factors of degree $O_r(p)$ [LN86]. Let $h(x)$ be one such irreducible factor and let $\mathbb{F}$ denote $Z_p[x]/(h(x))$, which is a finite extension of Z_p. Every element of $\mathbb{F}$ is the residue class of some $f(x) \in Z_p[x]$ and will be denoted as $\widehat{f(x)}$. Let $\hat{G} \subset \mathbb{F}$ denote the set of all residues classes of polynomials in G modulo $h(x)$.

Let $\hat{I}$ be the set of all residues of numbers in I modulo r and denote $t = |\hat{I}|$. Obviously $\phi(r) \geq t$. An element $\overline{f(x)} \in R$ is the residue class of a unique $f(x) \in Z_p[x]$ with $\deg f(x) < r$ and we define the degree of $\overline{f(x)}$ to be that of $f(x)$. Suppose $\overline{f(x)}, \overline{g(x)} \in G$ are of degree less than t such that $\widehat{f(x)} = \widehat{g(x)}$ in $\mathbb{F}$. Then

$$\widehat{f(x^i)} = \widehat{f(x)^i} = \widehat{f(x)}^i = \widehat{g(x)}^i = \widehat{g(x)^i} = \widehat{g(x^i)}$$

in $\mathbb{F}$, for all $i \in \hat{I}$. Now since $\hat{x}$ (the residue class of x in $\mathbb{F}$) is a primitive rth root of 1, all $\hat{x}^i$, $i \in \hat{I}$, are distinct elements in $\mathbb{F}$. Hence $f(x) = g(x)$ in $Z_p[x]$ (otherwise $h(x) = f(x) - g(x)$ will have more than t roots in $\mathbb{F}$). Thus, we have proved that for any two distinct elements of degree less than t in G will map to different elements in $\hat{G}$.

Since $p > \sqrt{\phi(r)}\log n \geq \sqrt{t}\log n$, the linear polynomials $x, x+1, ..., x+\lambda$, $\lambda := [\sqrt{t}\log n]$, are all distinct in G. Since $Z_p[x]$ is a unique factorisation domain, for distinct sequences $e := e_0, ..., e_\lambda$, the corresponding product

$$f_e := \prod\nolimits_{0 \leq a \leq \lambda} (x+a)^{e_a}$$

are distinct. Because $n^i \in I$, for every $i = 0, ...$, the number t is no less than the order of n in Z_r^*, and hence $t \geq \log^2 n$. Therefore,

$$t \geq \sqrt{t}\log n > \lambda.$$

To have $\deg f_e < t$, we can choose e such that either $e_i < t$ for some fixed i and $e_j = 0$ for all $j \neq i$, or each term $e_j = 0$ or 1 and not all $e_j = 1$. Then we can conclude that there are at least $2^{\lambda+1}$ distinct $f_e \in G$ with $\deg f_e < t$. This implies

$$|\hat{G}| \geq 2^{\lambda+1} > 2^{\sqrt{t}\log n} = n^{\sqrt{t}}. \tag{4}$$

Suppose n is not a perfect power of p. Consider the subset

$$J := \{(n/p)^i p^j \mid 0 \leq i, j \leq [\sqrt{t}]\} \subset I.$$

Since n is not a power of p, J contains $([\sqrt{t}]+1)^2 > t$ distinct elements. So there are at least two numbers m_1, m_2 in J with $m_1 > m_2$ such that $m_1 = m_2 \pmod r$. Then $x^{m_1} = x^{m_2} \pmod{x^r - 1}$, and hence $\widehat{x}^{m_1} = \widehat{x}^{m_2}$. Let $\widehat{f(x)} \in \hat{G}$. Then

$$\widehat{f(x)}^{m_1} = \widehat{f(x^{m_1})} = f(\widehat{x^{m_1}}) = f(\widehat{x^{m_2}}) = \widehat{f(x^{m_2})} = \widehat{f(x)}^{m_2}. \tag{5}$$

It follows that $\hat{f}(x)^{m_1} = \hat{f}(x)^{m_2}$ in $\mathbb{F}$. Hence every $\hat{f}(x) \in \hat{G}$ is a root of the equation $Y^{m_1} - Y^{m_2} = 0$ in $\mathbb{F}$. Again, because the number of roots of a polynomial in any extension of the field which its coefficients lie can not exceed its degree, we have $|\hat{G}| \leq m_1$. Clearly

$$m_1 \leq (n/p \cdot p)^{[\sqrt{t}]} \leq n^{\sqrt{t}},$$

hence $|\hat{G}| \leq m_1 \leq n^{\sqrt{t}}$, a contradiction to (4). Therefore n is a power of p. It is clear now that if n has no prime factor $p \leq \sqrt{\phi(r)} \log n$ and n is not a perfect power, then n is prime. $\square$

5. Time Complexity Analysis

We use the notation $\tilde{O}(t(n))$ for $O(t(n) * poly(\log t(n)))$, where $t(n)$ is some function of n. Note that we can perform addition, multiplication and division operations between two m bits number in time $\tilde{O}(m)$ [vzGG99]. Operations on two degree d polynomials with coefficients at most m bits can be done in time $\tilde{O}(dm)$ in a similar way [vzGG99]. In the following, we compute the runtime bound in terms of n and r in the algorithm.

Theorem 3. *The asymptotic time complexity of the algorithm is $\tilde{O}(r^{3/2} \log^3 n)$.*

Proof. The first step of the algorithm can be done with checking every possible exponent in $\tilde{O}(\log^3 n)$ time.

Step 2 can be done by trying successive numbers r that is coprime to n, and test if $n^k \neq 1 \pmod{r}$ for every $k \leq \log^2 n$. For a particular r, this can be done in $\tilde{O}(\log r \log^2 n)$, so it will take $\tilde{O}(r \log^2 n)$ time.

The time taken for step 3 is $\tilde{O}(r^{1/2} \log^2 n)$. In step 4, we have to verify about $\sqrt{\phi(r)} \log n$ equations. To verify each equation, one needs $\log n$ multiplications of degree r polynomials with coefficient of size $O(\log n)$, hence each equation can be verified in $\tilde{O}(r \log^2 n)$. The total time taken for step 4 is therefore $\tilde{O}(r^{3/2} \log^3 n)$.

Summing the above, we get the total time complexity of the algorithm: $\tilde{O}(r^{3/2}log^3 n)$.

$\square$

Up to this point, we have seen that the time needed for the algorithm is $\tilde{O}(r^{3/2}\log^3 n)$. Only if r is bounded by a polynomial of $\log n$ can the algorithm be in polynomial runtime overall. Indeed, this is the case, and we prove this in Theorem 4 using Lemma 5.0.4.

Lemma 5.0.4. *Let $LCM(m)$ denote the lcm of first m numbers. For $m \geq 7$: $LCM(m) \geq 2^m$.*

I heard the following proof from Dr. Yi-Chih Chiu, who was a post doctor research fellow at National Taiwan University when I worked on this thesis, and he mainly used the approach as in [Nai82], with more direct arguments.

Proof. We first prove that $n(n+1)\binom{2n+1}{n} | LCM(2n+1)$. This is true because

$$LCM(2n+1) = \prod_{p^r \leq 2n+1 < p^{r+1}} p^r,$$

while the exponent of p in the prime factorisation of $\binom{2n+1}{n}$ equals

$$\sum_{i \geq 1} ([(2n+1)/p^i] - [n/p^i] - [(n+1)/p^i]) \leq r$$

if $p^r \leq 2n+1 < p^{r+1}$ as each term is 0 or 1. When $p^a || n$ or $p^a || (n+1)$, we can improve the upper bound of the above summation by $r-a$. Hence, the divisibility property follows.

As a consequence, $LCM(2n+1) \geq n(n+1)\binom{2n+1}{n} = n(2n+1)\binom{2n}{n} \geq n \cdot 2^{2n}$. This shows $LCM(m) \geq 2^m$ for odd $m \geq 3$. The case of even m follows from the crude estimation $LCM(m) \geq LCM(m-1)$. $\square$

Theorem 4. *Let $n \geq 3$. In* step 2*, the integer r can be found with $r \leq \lceil \log^5 n \rceil + 1$ or n will be verified composite in this step.*

Proof. Since $n \geq 3$, so $m \geq \lceil \log^5 n \rceil > 10$ and by Lemma 5.0.4,

$$LCM(m) \geq 2^m. \tag{6}$$

Let r_0 be the least number that does not divide the product

$$Q := \prod_{i=1}^{\lfloor \log^2 n \rfloor} (n^i - 1) < n^{\log^4 n} = 2^{\log^5 n}.$$

If l is a prime number and $l^b \| LCM(r_0 - 1)$, then $l^b \le r_0 - 1$, and hence $l^b \mid Q$. This implies

$$LCM(r_0 - 1) \le Q < 2^{\log^5 n}.$$

then we must have

$$r_0 \le \left\lceil \log^5 n \right\rceil + 1,$$

for otherwise, $r_0 - 1 \ge \lceil \log^5 n \rceil + 1$, hence by (6),

$$LCM(r_0 - 1) \ge 2^{\lceil \log^5 n \rceil + 1} > 2^{\log^5 n}$$

a contradiction to the above inequality.

Now, if $(r_0, n) > 1$ then n is composite; otherwise, $(r_0, n) = 1$ and $O_{r_0}(n) > \log^2 n$.

$\square$

From theorems above, the time complexity of the algorithm is $\tilde{O}(r^{3/2} \log^3 n) = \tilde{O}(\log^{21/2} n)$. Using a deep result from analytic number theory in [Fo85], one can show that r may actually be chosen with $r = O(\log^3 n)$, and thus getting a more tight but ineffective bound of the runtime: $\tilde{O}(\log^{7.5} n)$.

References

[AKS04] Agrawal, Manindra; Kayal, Neeraj; Saxena, Nitin, *PRIMES is in P,* Annals of Mathematics **160**, 2(2004), 781-793.

[AKS02] Agrawal, Manindra; Kayal, Neeraj; Saxena, Nitin, *PRIMES is in P,* Preprint, (2002).

[RC05] Crandall, R. and Pomerance, C. *Prime Numbers: A Computational Perspective*, 2nd ed. New York: Springer-Verlag, 2005.

[AG05] Granville, A. *It Is Easy to Determine Whether a Given Integer Is Prime.* Bull. Amer. Math. Soc. **42**, 3-38, 2005.

[Nai82] M.Nair. *On Cheybyshev-type inequalities for primes.* Amer. Math. Monthly, **89**:126-129, 1982.

[Fo85] E. Fouvry. *Theorem de Brun-Titchmarsh; application au theoreme de Fermat.* Invent. Math., **79**:383-407, 1985.

[LN86] R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications.* Cambridge University Press, 1986.

[vzGG99] Joachim von zur Gathen and Jurgen Gerhard. *Modern Computer Algebra.* Cambridge University Press, 1999.

[MJ04] *Problems in Algebraic Number Theory,* 2nd ed. Springer-Verlag, 2004.

[LEN05] H. W. Lenstra Jr. and Carl Pomerance, *Primality testing with Gaussian periods,* preliminary version July 20, 2005.

[BER03] D. Bernstein, *Proving primality in essentially quartic time.* http://cr.yp.to/ntheory.html#quartic

DEPARTMENT OF MATHEMATICS, NATIONAL TAIWAN UNIVERSITY, TAIPEI 10764, TAIWAN

E-mail address: r01221030@ntu.edu.tw