

# AT THE BORDERLINE OF NUMBER THEORY AND COMPUTATIONAL COMPLEXITY THEORY<sup>1</sup>

Seminar presentation by

Robert McNaughton, Prof. Emeritus

Computer Science Department

Troy, NY 12180-3590

518-274-4452

28 January 2003

Last August, M.Agrawal, N.Kayal and N.Saxena of IIT Kanpur put forth a deterministic polynomial-time algorithm for deciding whether a given positive integer is a prime (see [1]). There had been an active pursuit of such an algorithm for about a quarter of a century, and so the discovery was momentous. These notes are for a one-hour seminar on the AKS paper: a non-expert seminar on the rudiments of the AKS algorithm and a part of the proof of its correctness and efficiency.

**Theorem 1.** There is an exponential-time algorithm to compute, for a given positive integer  $n$ , the prime decomposition of  $n$ . (The running time is  $O(n^2)$ .)

This fact is well known. It is natural to ask whether there exists a polynomial-time algorithm to do this job. The AKS algorithm does only part of it; unfortunately, no deterministic polynomial-time algorithm is known for factoring an integer. In particular, the execution of the AKS algorithm, upon discovering that  $n$  is composite, leaves us in the dark about how to factor it.

A, K and S begin their paper with the following theorem:

**Theorem 2.** If  $n$  is odd,  $n \geq 3$ ,  $c \geq 1$  and  $(c, n) = 1$  then  $n$  is prime iff, in the domain of the polynomials over  $x$ ,

$$(x - c)^n \equiv x^n - c \pmod{n}$$

*Proof:* The coefficient of the  $(i + 1)^{st}$  term in the binomial expansion of  $(x - c)^n$  is

$$n(n - 1) \cdots (n - i + 1)/i!$$

If  $n$  is prime then every term except the first and the last in this expansion is divisible by  $n$ . The first term  $x^n$  occurs on both the left and right sides of the congruence. This

---

<sup>1</sup>This seminar presentation is dedicated to the endearing memory of the late Edith Luchins, an RPI professor of Mathematics. She would so often provide knowledge for students and professors in Computer Science with problems in algebra and number theory, involving ideas similar to those discussed here.

leaves  $-c^n$  on the left and  $-c$  on the right. Since  $(c, n) = 1$ , we get  $c^n \equiv c \pmod{n}$  by Fermat's Little theorem, which establishes the congruence.

On the other hand, if  $n$  is composite let  $i_0$  be the smallest prime divisor of  $n$ . For all  $i < i_0$  the  $(i + 1)^{st}$  coefficient is divisible by  $n$ . But the  $(i_0 + 1)^{st}$  coefficient is not divisible by  $n$ . Hence the congruence does not hold, completing the proof.

This theorem gives us another exponential-time algorithm for primality, which (unlike the AKS algorithm) also yields a prime divisor of  $n$  when  $n$  is composite. Where  $i_0$  is as in the proof of Theorem 1, for all  $i < i_0$ , the  $(i + 1)^{st}$  coefficient is divisible by  $n$ . But the  $(i_0 + 1)^{st}$  coefficient is not. So the smallest prime divisor can be ascertained from the binomial expansion.

We can then take  $n' = n/i_0$  and find the smallest prime divisor of  $n'$ . By repeated application the whole prime decomposition of  $n$  can be computed in exponential time.

The train of thought leading to the AKS algorithm develops this idea in the context of the ring of the polynomials over the integers with one variable  $x$ . It considers the ideal in this ring with two generators  $n$  and  $x^r - 1$ , where  $r$  will be a prime number much less than  $n$ . Two polynomials  $P_1$  and  $P_2$  are congruent modulo this ideal if

$$P_1 = P_2 + nQ_1 + (x^r - 1)Q_2$$

for some polynomials  $Q_1$  and  $Q_2$ .

We write  $(x - c)^n \equiv x^n - c \pmod{x^r - 1, n}$  to mean that

$$(x - c)^n = x^n - c + (x^r - 1)Q_1 + nQ_2$$

for some polynomials  $Q_1$  and  $Q_2$ .

All congruence classes of the ideal generated by  $n$  and  $x^r - 1$  are represented by expressions of the form

$$a_0 + a_1x + \dots + a_{r-1}x^{r-1}$$

where for each  $i$ ,  $0 \leq a_i \leq n - 1$ . Multiplication and addition of such forms are easily done. Note that the laws of the ideal do not apply to exponents; e.g.,  $x^n$  cannot be replaced by  $x^0$  even though  $n \equiv 0$ .

**Theorem 3.** For all  $c \geq 1$  and prime  $r$ ,

$$(1) \quad (x - c)^n \equiv x^n - c \pmod{n}$$

implies

$$(2) \quad (x - c)^n \equiv x^n - c \pmod{x^r - 1, n}$$

The converse to this obvious theorem is not valid. There exist  $c, n, r$  for which (2) is true but (1) is false. The achievement of A, K and S is based on the fact that, for any

$n$ , (1) is true for all  $c$  if and only if (2) is true for all but a small set of values of  $c$  and  $r$ . This fact requires a difficult proof not contained in these notes.

Where  $x$  is composite,  $P(x)$  = the greatest prime divisor of  $x$ .

We now present the AKS algorithm, slightly modified for expository purposes. In this paper,  $\log(x)$  means  $\log_2(x)$ . The algorithm works for all inputs  $n$  sufficiently large. For the purpose of these notes, the minimum value for  $n$  will not be specified precisely, because of my failure to understand this matter thoroughly. Suffice it so say that any integer whose primality would be a serious question could be handled by the algorithm.

### The AKS algorithm

```

input integer  $n$ 
if ( $n > 3$  and ( $2|n$  or  $3|n$ )) output COMPOSITE;
if ( $n$  is of the form  $a^b, a > 1, b > 1$ ) output COMPOSITE;
 $r \leftarrow 4$ ;
until [EITHER ( $r = n$ )
      OR (( $r$  is prime) AND ( $P(r - 1) \geq 4\sqrt{r}\log(n)$ )
      AND ( $n^{(r-1)/P(r-1)} \not\equiv 1 \pmod{r}$ ))]
] do
  if ( $\gcd(n, r) \neq 1$ ) output COMPOSITE;
   $r \leftarrow r + 1$  ;
for  $a = 1$  to  $2\sqrt{r}\log(n)$  do
  if ( $(x - a)^n \not\equiv x^n - a \pmod{x^r - 1, n}$ )
    output COMPOSITE;
output PRIME

```

Comment on the algorithm: The statement “ $r$  is prime” and the function “ $P(r - 1)$ ” can be computed by the exponential-time algorithm of Theorem 1, since the

computed values of  $r$  will be bounded by a polynomial function of the logarithm of  $n$ , for almost all values of  $n$ .

Theorems 4 through 7 will justify the algorithm. Unfortunately, Theorems 6 and 7 are well beyond what can be proved here. They, or similar theorems, are proved in the AKS paper.

**Theorem 4.** If  $\lim_{n \rightarrow \infty} r/n = 0$  and  $a < n$  then  $(x - a)^n \pmod{x^{r-1}, n}$  is computable in polynomial time.

To avoid exponential time in this computation,  $(x - a)^n$  is computed by repeated squaring.

**Theorem 5.** It is possible to determine in polynomial time whether there exist positive integers  $c, b > 1$  such that  $n = c^b$ .

*Proof:* There are only  $\log n$  many possible values of  $b$ . For each  $b$ , compute  $\log c = (\log n)/b$  and see if  $c$  is an integer.

DEFINITION: Where  $p$  is a prime,  $o_p(x) =$  the smallest positive integer  $h$  such that

$$x^h \equiv 1 \pmod{p}$$

In words, the order of  $x$  modulo  $p$ .

**Theorem 6.** There exist  $c_1, c_2$  such that, for all  $n$  sufficiently large, there exists a prime  $r$

$$c_1(\log n)^6 \leq r \leq c_2(\log n)^6$$

such that either  $\gcd(r, n) > 1$  or else both

$$P(r - 1) \geq 4\sqrt{r} \log n$$

and

$$P(r - 1) | o_r(n)$$

(The latter case implies that

$$n^{o_r(n)/P(r-1)} \not\equiv 1 \pmod{r}.)$$

From this theorem we infer that the algorithm always terminates and that it runs in polynomial time.

**Theorem 7.** If (1)  $n$  is composite, (2)  $r < n$ , (3)  $r$  is prime,

$$(4) P(r - 1) \geq 4\sqrt{r} \log n$$

and

$$(5) n^{o_r(n)/P(r-1)} \not\equiv 1 \pmod{r}$$

then for some  $a \leq 2\sqrt{r} \log n$ ,

$$(x - a)^n \not\equiv x^n - a \pmod{x^r - 1, n}$$

**Theorem 8** For  $n$  sufficiently large, the algorithm always halts in polynomial time, outputting PRIME if  $n$  is prime, and COMPOSITE if  $n$  is composite.

The proof follows from the following three facts, the proof of each of which we leave to the reader:

(1) By Theorem 6, the algorithm always halts in polynomial time, outputting either PRIME or COMPOSITE.

(2) If COMPOSITE is the output then  $n$  is composite. (Use Theorems 2 and 3 for the case that the termination occurs during the “for” loop.)

(3) If  $n$  is composite then, by Theorem 7, COMPOSITE is the output.

The proofs of Theorems 6 and 7 depend heavily on advanced results in number theory, far beyond what might be taught in an elementary course. Let  $\pi(x)$  = the number of primes  $\leq x$ . A well known fact is:

**Lemma 1.** As  $x \rightarrow \infty$ ,  $\pi(x)$  is asymptotic to  $x/\log_e(x)$ .

This fact implies that the primes are increasingly sparse among the integers: as  $x$  gets larger,  $\pi(x)/x$  gets smaller and tends towards zero. But it approaches zero rather slowly. This property of  $\pi(x)$  is at the basis of the following lemma, known as the Brun-Titchmarsh Theorem (see [5]), which in turn is used by A, K and S in their proof of the important fact we call Theorem 6.

**Lemma 2.** There exist  $c > 0$  and  $n_0$  such that for all  $x \geq n_0$ ,

$$|\{p|p \text{ prime}, p \leq x \text{ and } P(p-1) > x^{2/3}\}| \geq cx/\log x$$

Some readers of the AKS paper have commented that the AKS algorithm is easy to program. (I have heard of at least one implementation.) In the paper the authors prove that the asymptotic time complexity of their algorithm is  $O((\log n)^{12})$ . They mention that, almost certainly, practical program runs will do significantly better than this. They describe some current research attempts in number theory which, if successful, may some day enable them to prove that the time complexity of their algorithm is  $O((\log n)^3)$ .

It seems likely that the AKS algorithm will be practically useful in testing the primality of large numbers.

## References

- [1] M. Agrawal, N.Kayal and N.Saxena, “PRIMES is in P,” unpublished but widely distributed paper, IIT Kanpur, August 6, 2002.
- [2] G.L. Miller, “Riemann’s hypothesis and tests for primality,” **J. Comput. Sys. Sci.**, vol. 13 (1976), pp. 300–317.
- [3] M.O. Rabin, “Probabilistic algorithm for testing primality,” **J. Number Theory**, vol. 12 (1980), pp. 128–138.
- [4] R. Lidl and N. Niederreiter, **Introduction to finite fields and their applications**, Cambridge Univ. Press, 1986.
- [5] R.C. Baker and G. Harman, “The Brun-Titchmarsh theorem on average,” in **Proc. Conference in Honor of Heini Halberstam**, Vol. I (1996), pp. 39–103.