An Introduction to the AKS Primality Test

Andreas Klappenecker

September 4, 2002

A prime p is a positive integer which is divisible by exactly two positive integers, namely by 1 and p. An integer $n > 1$ is called composite if it is not a prime. A fundamental question is:

How can we tell whether an integer $n > 1$ is prime or not?

Manindra Agrawal, Neeraj Kayal, and Nitin Saxena from IIT Kanpur proposed a new algorithmic solution to this question in August 2002 [1]. Unlike previous solutions, their algorithm produces the correct answer in polynomial time. The purpose of these lecture notes is to give a short overview of this primality test, and to provide a guide to the related literature.

Algorithm 1 (Agrawal, Kayal, Saxena)

Input: An integer $n > 1$.
if n is a power **then output** `composite` **fi**;
$r := 2$;
while $(r < n)$ **do**
   if $\gcd(r, n) \neq 1$ **then output** `composite` **fi**;
   if r is prime **then**
      $q :=$ largest prime factor of $r - 1$;
      if $(q \geq 4\sqrt{r}\log n)$ and $(n^{(r-1)/q} \not\equiv 1 \bmod r)$ **then break fi**;
   fi;
   $r := r + 1$;
od;
for $a = 1$ to $2\sqrt{r}\log n$ **do**
   if $(x - a)^n \not\equiv (x^n - a) \bmod (x^r - 1, n)$ **then output** `composite` **fi**;
od;
output `prime`;

Overview. Algorithm 1 can be divided into three parts.

S_1. The first step, *line 0*, determines whether the number n is of the form $n = m^d$, for some positive integers m and d, with $d > 1$. This amounts to check whether $\lfloor n^{1/k} \rfloor^k = n$ for some k in the range $2 \leq k \leq \log n$.

S_2. The second step, *lines 1-9*, determines whether n has a small prime divisor. The while loop is executed until a small prime $r \in O(\log^6 n)$ is found such that $r - 1$ has a large prime divisor q, which divides the multiplicative order of n modulo r.

S_3. The last step checks whether the relation $(x - a)^n \equiv x^n - a$ modulo $(x^r - 1, n)$ holds for various a's. This step is the crucial part of this method. Unfortunately, it is also the most time consuming one.

The last step is motivated by the following observation.

Lemma 1 *Let a, n be some positive integers such that $\gcd(a, n) = 1$, then n is a prime if and only if the relation $(x - a)^n \equiv x^n - a \bmod n$ holds.*

Proof. Suppose that n is a prime. Recall that $(x - a)^n = \sum_{i=0}^{n} \binom{n}{i} (-a)^{n-i} x^i$. For $0 < i < n$, $\binom{n}{i} \equiv 0 \bmod n$, since n is a prime. The coefficient of x^n is $\binom{n}{n}(-a)^0 = 1$. Notice that $(-a)^{n-1} \equiv 1 \bmod n$ by Fermat's little theorem, whence the coefficient of x^0 is $\binom{n}{0}(-a)^n \equiv -a \bmod n$. It follows that $(x - a)^n \equiv x^n - a \bmod n$.

Suppose now that n is composite. Let p be a prime dividing n, and $n = p^k m$ with $\gcd(p, m) = 1$. Repeatedly applying the known identity $\binom{c}{d} \equiv \binom{\lfloor c/p \rfloor}{\lfloor d/p \rfloor}\binom{c \bmod p}{d \bmod p} \bmod p$ yields $\binom{n}{p^k} \equiv \binom{m}{1} \equiv m \not\equiv 0 \bmod p$, thus $\binom{n}{p^k} \not\equiv 0 \bmod n$. Since $\gcd(a, n) \neq 1$, $(-a)^{n-p^k} \not\equiv 0 \bmod n$. Thus, the coefficient of x^{p^k} is nonzero mod n. It follows that $(x - a)^n \not\equiv x^n - a \bmod n$ for composite n. $\square$

Checking $(x - a)^n \equiv x^n - a \bmod n$ would be too time consuming. Therefore, the third step rather checks whether $(x - a)^n \equiv x^n - a \bmod (x^r - 1, n)$ holds. This is of course more efficient, since the polynomials are then of degree less than the small prime r.

We have to pay a price for this gain in efficiency. For composite n, it might now happen that $(x - a)^n \equiv x^n - a \bmod (x^r - 1, n)$ holds, although $(x - a)^n \not\equiv x^n - a \bmod n$. However, checking sufficiently many distinct a's allows to rule out such anomalies. The choice of the prime r guarantees that we can find a suitable a in the range $1 \leq a \leq 2\sqrt{r} \log n$.

Correctness. We proceed to show that the algorithm is correct. If the input is prime, then it follows from our preceding discussion that the output of Algorithm 1 is `prime`. The difficulty rests in showing that a composite number n cannot pass through the tests in steps S_1, S_2, and S_3 without producing the desired output `composite`.

Assume that an input n has passed the tests in the steps S_1, S_2, and S_3. In other words, the algorithm did not report that the number is composite. If the break statement is never executed, then $\gcd(r,n) = 1$ for all r in the range $2 \leq r < n$, which means that n is a prime.

Therefore, we may assume that the while loop terminated early by the break statement in line 6. This means that the algorithm found a prime r such that the largest prime factor q of $r-1$ satisfies $q \geq 2s$, with $s = 2\lfloor\sqrt{r}\rfloor \log n$. We have $n^{(r-1)/q} \not\equiv 1 \bmod r$ as a consequence of the test in line 6. Since $\gcd(r,n) = 1$, we also have $n^{(r-1)/q} \not\equiv 0 \bmod r$. Notice that n does not have any prime factors of size smaller than $r \geq q \geq s$, since such factors would have been detected by the test in line 3. Finally, we observe that passing step S_3 means that the input n fails all s tests in the for loop in lines 10–12. In other words, the number n satisfies $(x-a)^n \equiv (x^n - a) \bmod (x^r - 1, n)$ for all a in the range $1 \leq a \leq s$.

The particular choice of $s = 2\lfloor\sqrt{r}\rfloor \log n$ as an upper bound on the for loop and $q \geq 2s$ implies that equation (1) in the following theorem is satisfied, since $\binom{q+s-1}{s} > \left(\frac{q}{s}\right)^s \geq 2^s$, and the choice of s implies $2^s = n^{2\lfloor\sqrt{r}\rfloor}$. Thus, all hypotheses of the following theorem are satisfied.

Theorem A. *Let n, s be positive integers, $n > 1$. Assume that n is not a power. Let r be a prime and denote by q the largest prime factor of $r-1$. Suppose that n does not have a prime factor less than or equal to s, that*

$$\binom{q+s-1}{s} > n^{2\lfloor\sqrt{r}\rfloor}, \tag{1}$$

that $n^{(r-1)/q} \bmod r \not\in \{0,1\}$, and that $(x-a)^n \equiv x^n - a \bmod (x^r - 1, n)$ for all $1 \leq a \leq s$. Then n has to be a prime.

Remark. It should be noted that a suitable prime r will be found with certainty. In fact, the while loop will be iterated at most $r \in O(\log^6 n)$ times. We will discuss complexity issues later.

Proof of Theorem A. Let p denote a prime factor of n satisfying the condition $p^{(r-1)/q} \bmod r \not\in \{0,1\}$. Such a prime has to exists, for otherwise all prime factors p of n would satisfy $p^{(r-1)/q} \bmod r \in \{0, 1\}$, therefore n – as a product of these primes – would have to satisfy $n^{(r-1)/q} \bmod r \in \{0, 1\}$, contradicting the hypothesis of the theorem.

The ring $R = (\mathbf{Z}/n\mathbf{Z})[x]/(x^r - 1)$ is well-suited for algorithmic purposes, but for the analysis is will be simpler to use a finite field. Let $K = \mathbf{F}_p[x]/(h(x))$, where $h(x)$ denotes an irreducible factor of $(x^r - 1)/(x - 1)$. The finite field K is a homomorphic image of R. This coarser picture will be enough, since the constraints on q ensure that this field is not too small:

Lemma 2 *Let $h(x) \in \mathbf{F}_p[x]$ be an irreducible factor of $(x^r - 1)/(x - 1)$. Then $\deg h(x) \geq q$.*

Proof. Recall that the multiplicative order e of $p \bmod r$ is the smallest exponent e such that $p^e \equiv 1 \bmod r$. Notice that q must divide e. Fermat's little theorem shows that $p^{r-1} \equiv 1 \bmod r$. Hence $r - 1 = eb$ for some integer b. By definition, q divides $r - 1$. If q does not divide e, then q has to divide b, which yields $p^{(r-1)/q} \equiv p^{e(b/q)} \equiv 1 \bmod p$, contradicting our choice of p. Therefore, q divides e, hence $e \geq q$. It is shown in Theorem 2.47 of [5] that the degree of an irreducible factor $h(x)$ of the cyclotomic polynomial $(x^r - 1)/(x - 1)$ coincides with multiplicative order e of $p \bmod r$. □

Let $f_a(x)$ denote the polynomial $x - a$. It follows from our assumptions that$f_a(x^n) \equiv f_a(x)^n \bmod (x^r - 1, p)$ for all a in the range $1 \leq a \leq s$. In addition, we have $f_a(x^p) \equiv f_a(x)^p \bmod p$, since $\gcd(a, p) = 1$. As a consequence, we obtain similar power laws for products of these polynomials. We form the group generated by the polynomials $f_a(x)$. This data structure will allow to assemble the information obtained about the individual polynomials.

Lemma 3 *Let G be the subgroup of the multiplicative group K^* generated by the elements $(x - a)$ with $1 \leq a \leq s$. Then G is a cyclic group which is at least of order $\binom{s+q-1}{s}$.*

Proof. Since K^* is a cyclic group, G must be cyclic as well. Let a, b be distinct integers in the range $1 \leq a, b \leq s$. It cannot happen that the elements $(x - a)$ and $(x - b)$ are equal in K, because this would imply that p is a small prime dividing $|a - b| \leq s$, and n does not have any prime factors $p \leq s$ by assumption. The group G contains at least $\binom{q+s-1}{s}$ elements, since the elements $\prod_{i=1}^{s}(x - i)^{e_i}$ satisfying $e_1 + e_2 + \cdots + e_s \leq q - 1 < \deg h(x)$ are pairwise distinct. □

Lemma 4 *Let $g(x)$ be a generator of the cyclic group G. The set of exponents $\mathcal{E} = \{e \in \mathbf{Z} \,|\, e \geq 1, g(x^e) \equiv g(x)^e \bmod (x^r - 1, p)\}$ is closed under multiplication.*

Proof. Let $e, d \in \mathcal{E}$. Thus, $g(x^e) \equiv g(x)^e \bmod (x^r - 1, p)$. Substituting x^d for x yields $g(x^{ed}) \equiv g(x^d)^e \bmod (x^{dr} - 1, p)$. Since $x^r - 1$ divides $x^{rd} - 1$, we obtain in particular $g(x^{de}) \equiv g(x^d)^e \bmod (x^r - 1, p)$. Therefore, we can derive $g(x)^{de} \equiv (g(x)^d)^e \equiv g(x^d)^e \equiv g(x^{de}) \bmod (x^r - 1, p)$. Thus, $ed \in \mathcal{E}$. □

Lemma 5 *We have $n, p \in \mathcal{E}$, hence $n^i p^j \in \mathcal{E}$.*

Proof. The generator $g(x)$ is of the form $\prod_{i=1}^{s} (x - i)^{e_i}$. Therefore,

$$g(x)^n \equiv \prod_{i=1}^{s} (x - i)^{ne_i} \equiv \prod_{i=1}^{s} (x^n - i)^{e_i} \equiv g(x^n) \bmod (x^r - 1, n),$$

thus this holds in particular modulo $(x^r - 1, p)$. The relation $g(x^p) \equiv g(x)^p \bmod p$ holds for any product of the polynomials $(x - a)$. Therefore, $p, n \in \mathcal{E}$. □

We can now conclude the proof of Theorem A. Consider the products $n^i p^j$ with $0 \leq i, j \leq \lfloor \sqrt{r} \rfloor$. There are $(1 + \lfloor \sqrt{r} \rfloor)^2 > r$ such numbers. Thus, by the pigeonhole principle, we must have distinct (i, j) and (k, ℓ) such that $n^i p^j \equiv n^k p^\ell \bmod r$. Let $u = n^i p^j$ and $t = n^k p^\ell$. By construction, $g(x^u) \equiv g(x^t) \bmod (x^r - 1, p)$, hence $g(x)^u \equiv g(x)^t \bmod (x^r - 1, p)$. It follows that $g(x)^t = g(x)^u$ in the field K. This means that $t \equiv u \bmod |G|$. However, $n^{2\lfloor \sqrt{r} \rfloor} < \binom{q+s-1}{s} \leq |G|$. Therefore, t and u must be equal, hence $n^{i-k} = p^{j-\ell}$. It is not possible that i equals k, since this would force $j = \ell$. Therefore, n is of the form $n = p^m$. Since n is not a power, we must have $m = 1$, hence p is a prime. This concludes the proof of Theorem A. □

Complexity. We give a rough complexity estimate to show that the runtime of Algorithm 1 is bounded by polynomial in the number of digits of n. The existence of a suitable small prime r is a consequence of results from analytic number theory:

Lemma 6 *There exist two real constants c_1, c_2 such that there is a prime r in the range $c_1(\log n)^6 \leq r \leq c_2(\log n)^6$, which satisfies the following property: $r - 1$ has a prime factor $q \geq 4\lfloor \sqrt{r} \rfloor \log n$ and q divides the multiplicative order of $n \bmod r$.*

This lemma is a consequence of a result by Fouvry [4], see Lemma 4.2 in [1] for a proof.

Proposition 7 *The runtime of Algorithm 1 is polynomial in the number of digits of n.*

Proof. The runtime is determined by the third step, since this is the most time consuming part of Algorithm 1. Calculating $(x-a)^n \bmod (x^r-1, n)$ with a square-and-multiply method requires $O(\log n)$ multiplications of polynomials of degree less than r, with coefficients in $\mathbf{Z}/n\mathbf{Z}$. The multiplication of two such polynomials requires $O(r^2)$ operations in the ring $\mathbf{Z}/n\mathbf{Z}$. Therefore, the for loop requires $O(2\sqrt{r}\log n \cdot r^2 \log n)$ ring operations. A classical multiplication in $\mathbf{Z}/n\mathbf{Z}$ requires $O((\log n)^2)$ additions. Assuming $r \in O((\log n)^6)$, we get a total complexity estimate of $O((\log n)^{19})$. □

Notes. We followed in our exposition the seminal paper [1], taking advantage of the expositions given by Bernstein [2], and Morain [6]. All three papers are highly recommended for further study. The book by Crandall and Pomerance [3] is an excellent source for known primality tests.

References

[1] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. Preprint, IIT Kanpur, August 2002.

[2] D. Bernstein. An exposition of the Agrawal-Kayal-Saxena primality-proving theorem. Preprint, University of Illinois at Chicago, August 2002.

[3] R. Crandall and C. Pomerance. *Prime Numbers: A Computational Perspective.* Springer Verlag, New York, 2001.

[4] E. Fouvry. Theoreme de Brun-Titchmarche; application au theoreme de Fermant. *Invent. Math.*, 79:383–407, 1985.

[5] R. Lidl and H. Niederreiter. *Finite Fields.* Cambridge University Press, 2nd edition, 1997.

[6] F. Morain. Primalité théorique et primalité pratique ou AKS vs. ECPP. Preprint, Laboratoire d'Informatique de l'École Polytechnique, August 2002.

Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, `klappi@cs.tamu.edu`