News from India: Primes is in P[1]

Jaikumar Radhakrishnan <jaikumar@tcs.tifr.res.in>
Tata Institute of Fundamental Research, Mumbai

Kavitha Telikepalli <kavitha@mpi-sb.mpg.de>
Max-Planck Institut für Informatik, Saarbrücken

V Vinay <vinay@csa.iisc.ernet.in>
Indian Institute of Science, Bangalore

On 4 August 2002, an email message from Manindra Agrawal, Neeraj Kayal and Nitin Saxena left the computers of the Indian Institute of Technology, Kanpur. The subject line read "Primes is in P", and attached to the message was a paper [AKS] with the same title. This made News.

The paper presented a polynomial time algorithm for recognizing prime numbers, solving a longstanding open problem in Complexity Theory, and passing a milestone in the centuries-old journey towards understanding prime numbers.

In this article, we describe the revised algorithm due to Agrawal, Kayal and Saxena and present a proof of its correctness; this algorithm builds on an earlier improvement due to H.W. Lenstra, Jr.

We want a polynomial-time method to determine if a given number n is prime, that is, a method that terminates after performing $O((\log n)^c)$ steps of computation. The starting point of the new test for primality is the following.

Proposition 1 *(a) If n is prime, then $(X-a)^n = X^n - a \pmod n$.*

(b) If $\gcd(a,n) = 1$ and n is composite, then $(X-a)^n \neq X^n - a \pmod n$.

Proof: (Sketch) (a) If n is prime $\binom{n}{i} = 0 \pmod n$ for $i = 1, 2, \ldots, n-1$ and $a^n = a \pmod n$. (b) If n is composite and p is a prime factor of n, then the coefficient of X^p in $(X-a)^n$, is $\binom{n}{p}(-a)^{n-p} \neq 0 \pmod n$. □

This proposition gives us the following algorithm.

If $(X-1)^n = X^n - 1 \pmod n$, then n is prime, otherwise it is composite.

Figure 1: A primality testing algorithm

[1] *A shorter version of this write-up, based on the original paper [AKS] was written for the EATCS bulletin. The present write-up and its earlier versions are available at:* www.tcs.tifr.res.in/∼jaikumar/mypage.html.

This algorithm classifies numbers correctly as prime and composite; unfortunately, it cannot be implemented efficiently. There are two difficulties. First, the straightforward method for computing the polynomial $(X-1)^n$, requires $n-1$ multiplications, and we are allowing ourselves only $O((\log n)^c)$ time. This is not a serious problem. It is well-known that one can compute powers more efficiently by repeated squaring (see Figure 2). Interestingly, the use of repeated squaring

> If n is a k-bit number, then for $i = 0, 1, 2, \ldots, k-1$, compute $b_i = (X-1)^{2^i} \pmod n$ by repeated squaring, starting from $b_0 = X - 1$. Let $n = \sum_{j=0}^{k-1} \epsilon_i 2^i$, $\epsilon_i \in \{0, 1\}$ be the binary expansion of n. Then, $(X-1)^n = \prod_{i=0}^{k} b_i^{\epsilon_i}$.

Figure 2: Powering by repeated squaring

for computing powers seems to have originated in India, but in the absence of email, it took some time for the word to get around[2].

The second problem with the algorithm of Figure 1, and this is more serious, is that the polynomial $(X-1)^n$ has potentially $n+1$ coefficients, and computing such a polynomial even by the repeated squaring, is not feasible in $O((\log n)^c)$ steps. The key idea in the new primality test is to perform computations modulo a polynomial of small degree. This way, the number of coefficients in the polynomial stays small.

Using this idea, Agrawal, Kayal and Saxena propose the algorithm displayed in Figure 3. To implement Step 2, we try all values of r, starting from 2, one after the other. If at any stage we discover a non-trivial divisor of n, we declare that n is composite. We will show, using elementary arguments, that for all large n, the number r in Step 2, can be chosen to be $O((\log n)^5)$. Assuming this, it is easy to check that this algorithm runs in polynomial time; with some care this algorithm can be implemented using only $O((\log n)^{10.5}(\log\log n)^c)$ bit operations.

We first show that this algorithm is correct and then verify that Step 2 can be implemented as described above.

The proof of correctness

It is easy to verify, using Proposition 1, that if n is prime, this algorithm will never declare that it is composite. So, we only need to argue that composite

[2]Knuth [K, p. 461] says: The method is quite ancient; it appeared before 200 B.C. in Pingala's Hindu classic Chandah-sutra [see B. Datta and A.N. Singh, *History of Hindu Mathematics* **2** (Lahore: Motilal Banarsi Das, 1935), 76]. There seems to be no other reference to this method outside of India during the next 1000 years, but a clear discussion of how to compute 2^n efficiently for arbitrary n was given by al-Uqlidisi of Damascus in A.D. 952; see *The Arithmetic of al-Uqlidisi* by A.S. Saidan (Dordrecht: D. Reidel, 1975), 341–342, where the general ideas are illustrated for $n = 51$.

Input: An integer $n \geq 2$.

Step 1: If n is of the form a^b, for integers $a, b \geq 2$, then n is composite.

Step 2: Choose r so that the order of n modulo r is at least $4(\log n)^2 + 2$. Let $\ell = \lfloor 2\sqrt{r} \log n \rfloor + 1$.

Step 3: For $a = 2, 3, \ldots, \ell$, if a divides n, then n is composite.

Step 4: For $a = 1, 2, \ldots, \ell$, if $(X - a)^n \neq X^n - a \pmod{X^r - 1, n}$, then n is composite.

Step 5: If n has not been declared composite by the earlier steps, then n is prime.

Figure 3: The primality testing algorithm

numbers are not declared prime. Compare Step 4 to the inefficient primality test of Figure 1. The main difference is that we are now performing the computations modulo $X^r - 1$. The main danger in this is that even if $(X - a)^n \neq X^n - a \pmod{n}$, it could be that $(X - a)^n = X^n - a \pmod{X^r - 1, n}$. To compensate for this, we now verify the identity for ℓ different values of a, instead of trying just one value, namely 1. Agrawal, Kayal and Saxena show that this is adequate compensation. To see this, let us assume the opposite and show that this leads to a contradiction.

Assumption: n is a composite number and the algorithm of Figure 3 declares that it is prime.

Because the number n passes all tests in Step 4, we know that

$$\text{for } a = 1, 2, \ldots, \ell,\ (X - a)^n = X^n - a \pmod{X^r - 1, n}. \tag{1}$$

Note that in the above identity we can replace the n in $(\text{mod } X^r - 1, n)$ by any divisor of n. Let p be a prime divisor of n. Then, we have

$$\text{for } a = 1, 2, \ldots, \ell,\ (X - a)^n = X^n - a \pmod{X^r - 1, p}. \tag{2}$$

Since p is prime, we always have (see Proposition 1(a))

$$\text{for } a = 1, 2, \ldots, \ell,\ (X - a)^p = X^p - a \pmod{X^r - 1, p}. \tag{3}$$

We thus see that the numbers n and p satisfy similar identities in (2), (3). Such numbers are called *introspective numbers* by Agrawal, Kayal and Saxena. The next claim shows that introspective numbers can be multiplied to produced more introspective numbers.

Claim 1 *Suppose*

$$\begin{aligned}(X-a)^{m_1} &= X^{m_1}-a \pmod{X^r-1,p} \text{ and}\\ (X-a)^{m_2} &= X^{m_2}-a \pmod{X^r-1,p}.\end{aligned}$$

Then, $(X-a)^{m_1 m_2} = X^{m_1 m_2}-a \pmod{X^r-1,p}$.

Proof: The second assumption says that $(X-a)^{m_2}-(X^{m_2}-a) = (X^r-1)g(X) \pmod{p}$, for some polynomial $g(X)$. By substituting X^{m_1} for X in this identity, we get

$$(X^{m_1}-a)^{m_2}-(X^{m_1 m_2}-a) = (X^{m_1 r}-1)g(X^{m_1}) \pmod{p}.$$

Since X^r-1 divides $X^{m_1 r}-1$, this shows that $(X^{m_1}-a)^{m_2} = X^{m_1 m_2}-a \pmod{X^r-1,p}$. Using this and the first assumption, we obtain

$$(X-a)^{m_1 m_2} = (X^{m_1}-a)^{m_2} = X^{m_1 m_2}-a \pmod{X^r-1,p}.$$

□

Now starting from (2) and (3), and repeatedly applying the above claim, we see that for each m of the form $p^i n^j$, $(i,j \geq 0)$, we have $(X-a)^m = X^m - a \pmod{X^r-1,p}$, for $a = 1,2,\ldots,\ell$. (The case $i,j=0$ corresponds to $m=1$, and is trivially true.)

Let t be the order of the subgroup G of Z_r^*, generated by p and n taken modulo r. Consider the list $L = (p^i n^j : 0 \leq i,j \leq \lfloor\sqrt{t}\rfloor)$. Note that all elements in this list are at most $n^{2\sqrt{t}}$. Each element of this list taken modulo r resides in the subgroup G generated by p and n inside Z_r^*. This list has $(\lfloor\sqrt{t}\rfloor+1)^2 > t = |G|$ elements. Thus, we have two numbers in the list that are congruent modulo r. Let these numbers be $m_1 = p^{i_1} n^{j_1}$ and $m_2 = p^{i_2} n^{j_2} = m_1 + kr$, where $(i_1,j_1) \neq (i_2,j_2)$. From now on we will concentrate on just m_1 and m_2. Since $X^r = 1 \pmod{X^r-1}$, we have $(X-a)^{m_2} = X^{m_1+kr}-a = X^{m_1}-a = (X-a)^{m_1} \pmod{X^r-1,p}$. That is,

$$\text{for } a = 1,2,\ldots,\ell,\ (X-a)^{m_1} = (X-a)^{m_2} \pmod{X^r-1,p}. \qquad (4)$$

Claim 2 $m_1 = m_2$.

We will prove this claim below. Let us first complete the proof of correctness by assuming this claim. From this claim and the definition of m_1 and m_2, we see that $p^{i_1} n^{j_1} = p^{i_2} n^{j_2}$. Since $(i_1,j_1) \neq (i_2,j_2)$ and p is prime, this implies that n is a power of p. That is $n = p^s$ for some s. If $s \geq 2$, Step 1 of the algorithm would already have declared that n is composite. This contradicts our assumption that the algorithm declares that n is prime. On the other hand, if $s = 1$, then n is prime, contradicting our assumption that n is composite. We have proved that the algorithm is correct assuming Claim 2.

Proof of Claim 2: We will use the following elementary fact: *in a field, a non-zero polynomial of degree d has at most d roots.* To see the connection between this fact and the claim we are trying to prove, note that (4) just says that the polynomial $b(Z) = Z^{m_1} - Z^{m_2}$ has several roots, namely, $X - a$ for $a = 1, 2, \ldots, \ell$. If we could somehow conclude from this that $b(Z)$ has more roots than its degree, namely $\max\{m_1, m_2\}$, in some field, we can then infer that $b(Z)$ is the zero polynomial, implying $m_1 = m_2$. We, thus, have to arrange two things. First, we need to move to a field instead of the ring $\mathbb{F}_p[X]/(X^r - 1)$. Second, we need to show that $b(Z)$ has more roots than $\max\{m_1, m_2\}$.

Moving to a filed: Let η be a primitive r-th root of unity. Then, by (4), we have

$$\text{for } a = 1, 2, \ldots, \ell, \ (\eta - a)^{m_1} = (\eta - a)^{m_2}. \tag{5}$$

in the field $\mathbb{F}_p(\eta)$, that is, $\eta - a$ is a root of the polynomial $b(Z) = Z^{m_1} - Z^{m_2}$. Note that if e_1 and e_2 are oots of $b(Z)$, then $e_1 e_2$ is also a root. Thus, each element of the form $\prod_{a=1}^{\ell}(\eta - a)^{\alpha_a}$ (for non-negative integers α_a) is a root of $b(Z)$; in particular, for $\ell' \stackrel{\text{def}}{=} \lfloor 2\sqrt{t}\log n \rfloor + 1 \leq \ell$ (the last inequality holds because $t \leq r - 1$), each element of the set

$$S = \left\{ \prod_{a=1}^{\ell'} (\eta - a)^{\alpha_a} : \alpha_a \in \{0, 1\} \right\}$$

is a root. We will argue (based on the choice of r in Step 2) that S has $2^{\ell'}$ elements. Thus, the equation $b(Z) = Z^{m_1} - Z^{m_2}$ has at least $2^{\ell'}$ roots in the field $\mathbb{F}_p(\eta)$. Now, $m_1, m_2 \leq n^{2\lfloor\sqrt{t}\rfloor}$ and $2^{\ell'} > n^{2\sqrt{t}}$. This implies that $b(Z)$ is the zero polynomial, that is, $m_1 = m_2$, establishing Claim 2.

$Z_1^m - Z_2^m$ has many roots: We need to argue that the $2^{\ell'}$ products of the form $\prod_{a=1}^{\ell'}(\eta - a)^{\alpha_a}$, $\alpha_a \in \{0, 1\}$, are distinct elements of $\mathbb{F}_p(\eta)$. Each of these elements is obtained by substituting η for X in a polynomial of the form $\prod_{a=1}^{\ell'}(X - a)^{\alpha_a} \in \mathbb{F}_p[X]$. First, are these polynomials distinct in $\mathbb{F}_p[X]$? By Step 3, n (and hence p) has no small small divisors. Thus, each $X - a$, $a = 1, \ldots, \ell'$, is a distinct element of $\mathbb{F}_p[X]$. Since elements of $\mathbb{F}_p[X]$ factorize uniquely into irreducible factors, different products must give rise to different polynomials. Now, we need to show for different $g(X)$ of the form $\prod_{a=1}^{\ell'}(X - a)^{\alpha_a}$, $g(\eta)$ are different in $\mathbb{F}_p(\eta)$. By Claim 1, $g(X)^m = g(X^m) \pmod{X^r - 1, p}$ for each such $g(X)$ of the form $\prod_{a=1}^{\ell'}(X - a)^{\alpha_a}$ and each m of the form $p^i n^j$. It follows that $g(\eta)^m = g(\eta^m)$ in $\mathbb{F}_p(\eta)$ for each such m.

Thus, if if $g_1(X)$ and $g_2(X)$ are of the form $\prod_{a=1}^{\ell'}(X - a)^{\alpha_a}$ and $g_1(\eta) = g_2(\eta)$, then $g_1(\eta^m) = g_2(\eta^m)$; that is, each element of the form η^m ($m = n^i p^j$, $i, j \geq 0$) is a root of the polynomial $g_1(X) - g_2(X) \in \mathbb{F}_p[X]$. Since η is a primitive r-th root

of unity, $\eta^{n^i p^j}$ takes as many distinct values as the number of distinct residues mod r generated by $n^i p^j$. Hence, $g_1(X) - g_2(X)$ has at least t roots in $\mathbb{F}_p(\eta)$. But, $g_1(X)$ and $g_2(X)$ are polynomials of degree at most $\ell' \leq \lfloor 2\sqrt{t}\log n \rfloor + 1$, and $t \geq \mathsf{ord}_r(n) \geq 4(\log n)^2 + 2$. But then, $\ell' < t$, and $g_1(X) - g_2(X)$ must be the zero polynomial, that is $g_1(X) = g_2(X)$ in $\mathbb{F}_p[X]$. Thus, distinct products of the form $\prod_{a=1}^{\ell'}(X-a)^{\alpha_a} \in \mathbb{F}_p[X]$ give distinct elements in $\mathbb{F}_p(\eta)$ when we substitute η for X. That is, S has $2^{\ell'}$ distinct elements, and $b(Z)$ has at least $2^{\ell'}$ roots in the $\mathbb{F}_p(\eta)$. This, finally, establishes that the algorithm is correct.

The existence of a small r in Step 2

We will need a lower bound on the *lcm* of $1, 2, \ldots, R$.

Claim 3 (See [N, V]) *The lcm of the* $1, 2, \ldots, 2k+1$ *is at least* 2^{2k}. *(In fact, it is known [N, V] that for* $R \geq 7$, *the* lcm *of* $1, 2, \ldots, R$ *is at least* 2^R.*)*

Proof: We have,

$$2^{-2k} \geq \int_0^1 [x(1-x)]^k \mathrm{d}x = \sum_{i=0}^{k} \binom{k}{i} \int_0^1 (-1)^i x^{k+i} \mathrm{d}x = \sum_{i=0}^{k} \frac{M_i}{k+i+1} = \frac{M}{L},$$

where the M_i's and M are integers and L is the *lcm* of $k+1, k+2, \ldots, 2k+1$. The integral is clearly positive, so it is at least $1/L$. Thus, $L \leq 2^{2k}$. □

Let us return to Step 2 of the algorithm. Suppose for all r less than some some odd number R, we have that r does not divide n and also $\mathsf{ord}_r(n) \leq T \stackrel{\text{def}}{=} 4(\log n)^2 + 2$. Then, each $r \leq R$ divides

$$\prod_{i=0}^{T}(n^i - 1) \leq n^{T^2}.$$

By the above claim, we have $2^{R-1} \leq n^{T^2}$, that is, $R \leq T^2 \log n + 1$, Thus, there is a number $r = O((\log n)^5)$ with $\mathsf{ord}_r(n) > T$.

References

[AKS] M Agrawal, N Kayal and N Saxena. Primes is in P, `http://www.cse.iitk.ac.in/users/manindra/primality.ps`.

[K] DE Knuth. The Art of Computer Programming, Volume 2, Seminumerical Algorithms, 3rd edition, Addison-Wesley, 1998.

[N] M Nair. On Chebyshev-type inequalities for primes. *American Mathematical Monthly*, 89:126–129, 1982.

[V] V Vinay. Lecture notes on Computational Complexity Theory scribed by PR Subramanya. `http://www.imsc.res.in/~iarcs/elnotes/cc.ps.gz`.