UPDATE ON PRIMALITY TESTING

SERGEI V. KONYAGIN AND CARL POMERANCE

Abstract. We discuss recent developments in the field of primality testing since the appearance [10] of our joint paper *On primes recognizable in deterministic polynomial time.*

1. Background

The subject of primality testing concerns the creation and analysis of efficient algorithms for deciding whether a given integer $n > 1$ is prime or composite. This subject is closely related to, but distinct from, factoring. Some algorithms, such as trial division, can accomplish both tasks, but the most efficient methods are tailored to one or the other.

From a practical point of view, the story of primality testing is a simple one. In real-world applications one does not require mathematical certitude, a tiny possibility of error being acceptable, so various random algorithms that have been known for decades and are easy to implement may be used. An example, commonly known as the Miller–Rabin test, runs in $O((\log n)^{2+\epsilon})$ bit operations (using fast arithmetic subroutines) and almost certainly returns a correct verdict on the primality of a given input n, with the bonus that a composite verdict is mathematically correct. Even the simple base-2 Fermat congruence $2^{n-1} \equiv 1 \pmod{n}$ when applied to a large *random* input n almost certainly steers one right (a number n for which the congruence holds is almost certainly prime, a number n for which it does not hold is definitely composite). Indeed, as shown by Erdős [6], composite numbers n satisfying $2^{n-1} \equiv 1 \pmod{n}$ are much scarcer than primes. For more details on these and similar tests see [5] and the references there.

But a problem as fundamental as deciding primality cries out for a thorough mathematical analysis. Here too, where no possibility of error is to be tolerated, there is both a theoretical and practical side. The practical primality tester has specific numbers n in mind that are to be tested, and wishes to implement an algorithm that will give a completely correct answer. It is possible for such an algorithm to use randomness, where coins are flipped (figuratively), but there is no doubt in the output, the only issue being the running time of the algorithm. A simple but illustrative example is that of finding a quadratic nonresidue for a given prime p. This is an integer k such that the congruence $x^2 \equiv k \pmod{p}$ has no integral solutions. We know that for an odd prime p exactly $(p-1)/2$ choices for k in $\{1, \dots, p-1\}$ are quadratic nonresidues. Moreover, via either Euler's criterion or the law of quadratic reciprocity for Jacobi symbols, it is possible to decide quickly (and deterministically) if a candidate k works or not. So a random and quick method to find a quadratic nonresidue k is to choose randomly from $\{1, \dots, p-1\}$ until one is found. This simple algorithm runs in expected polynomial time. Remarkably, without assuming an unproved hypothesis (the Extended Riemann Hypothesis), we know no deterministic method for finding a quadratic nonresidue that runs in polynomial time.

Long before our article was published, we had the Adleman–Huang test [1], a random primality test with running time expected to be polynomial (and, as opposed to the Miller–Rabin test, there is no doubt in the output). Based on the arithmetic of Jacobian varieties of hyperelliptic curves of genus 2 (and also on elliptic curves), it is a very difficult result, requiring an entire volume for its analysis. Other tests, based on elliptic curves (practical improvements of the Goldwasser–Kilian test) are not theoretically complete, but stand as excellent practical primality proving algorithms for those who do not wish to have any possibility of error. Again, see [5] for more on this.

And this brings us to the last holdout of the theorist: a deterministic primality test that runs in polynomial time. Such a test has long been known (in fact, a version of the Miller–Rabin test), but it relies on the Extended Riemann Hypothesis in a similar way as the quadratic nonresidue problem mentioned above. Withot any unproved hypothesis, we had the APR test [2] with complexity $O((\log n)^{c \log\log\log n})$, tantalizingly close to being polynomial. We also had an interesting result of Pintz, Steiger, and Szemerédi that presented a set of primes, with counting function to x of about $x^{2/3}$, which could be recognized in deterministic polynomial time. These primes p were characterized by $p-1$ being divisible by a very large power of 3.

In our paper we showed that any prime p can be proved prime by a deterministic algorithm in polynomial time, provided we have a fully-factored divisor $d > p^{\epsilon}$ of $p-1$. Furthermore, a simple procedure identifies more than $x^{1-\epsilon}$ primes p up to x with such a fully-factored divisor in $p-1$, and so we have many primes that are recognizable in polynomial time. The case when $d > p^{1/2+\epsilon}$ was done earlier by Fürer [9] (we only learned of this paper recently), and the case when $p-1$ itself is fully factored was rediscovered by Fellows and Koblitz [8].

Our paper had one practical component for those interested in implementing a primality test. The so-called "$n-1$ test" of Brillhart, Lehmer, and Selfridge requires a fully-factored divisor of $p-1$ larger than $p^{1/3}$. Our paper was able to reduce the exponent 1/3 in this practical test to 3/10. (For positive exponents smaller than 3/10 our algorithm still has polynomial complexity, but it is not so practical.) More recently, an analog of this improvement was accomplished for the "$n+1$ test", see [5], though it is no longer deterministic.

2. Derandomization and the AKS algorithm

By far the most important development since our article was the AKS algoritm [3], named for its inventors, Agrawal, Kayal, and Saxena. Their algorithm is deterministic and it distinguishes between primes and composites in polynomial time. Further it does not depend on any unproved hypotheses for its analysis.

Like the algorithm in our paper and in many other approaches to primality testing, the AKS algorithm either recognizes n as composite by a series of simple tests, or if n passes all of these tests, a group is built up that is so large that n is inescapably prime. (For more on this line of thought see [12].)

Two analyses of the AKS algorithm are presented in [3], a more elementary analysis using effective tools and running time $O((\log n)^{10.5+\epsilon})$, and an analysis using ineffective tools and running time $O((\log n)^{7.5+\epsilon})$. Both of these estimates are upper bounds for the true running time, conjectured to be $O((\log n)^{6+\epsilon})$. A version of the AKS algorithm with this running time and effective tools is presented in the preprint [11] and is described in [5].

Unfortunately, the AKS algorithm has not proved to be numerically competitive with previous primality tests. Even a random version with expected running time $O((\log n)^{4+\epsilon})$ (see [4]) is not competitive.

In [3] a conjecture is made that suggests a version of the algorithm has running time $O((\log n)^{3+\epsilon})$. Using a heuristic of Erdős [7] on Carmichael numbers, Lenstra and Pomerance (unpublished) have given a plausibility argument that this AKS conecture is false.

Since we knew already a random polynomial-time algorithm for primality testing, the AKS test might be thought of as a derandomization, even though it bears little resemblance to the Adleman–Huang test. Similarly, the Fürer and Fellows–Koblitz algorithms for proving the primality of a prime p where a large part of $p-1$ is factored are derandomizations of an algorithm of Lucas, as improved by Proth, Pocklington, and Lehmer early in the twentieth century. Our paper as well contains a derandomization (and extension) of the Brillhart, Lehmer, Selfridge $n-1$-test, as mentioned. In [13], Źrałek applied some of the methods of our paper to derandomize a factorization algorithm, namely the $p-1$ method of Pollard. Here, one is expected to find quickly those prime factors p of n which have the additional property that all of the prime factors of $p-1$ are small. (For this reason, some implementers of the RSA cryptosystem use prime factors p, q where both $p-1, q-1$ have large prime factors, so-called safe primes.) The Pollard $p-1$ method uses randomness and Źrałek derandomizes it. In a later paper he again uses similar ideas, this time to factor polynomials over some finite fields $\mathbf{F}_p$.

References

[1] L. M. Adleman and M.-D. A. Huang, *Primality testing and two-dimensional abelian varieties over finite fields*, Lecture Notes in Math. **1512**, Springer-Verlag, Berlin, 1992, 142 pp.

[2] L. M. Adleman, C. Pomerance, and R. S. Rumely, *On distinguishing prime numbers from composite numbers*, Ann. of Math. **117** (1983), 173–206.

[3] M. Agrawal, N. Kayal, and N. Saxena, PRIMES *is in* P, Ann. of Math. **160** (2004), 781–793.

[4] D. Bernstein, *Proving primality in essentially quartic random time*, Math. Comp. **76** (2007), 389–403.

[5] R. Crandall and C. Pomerance, *Prime numbers: a computational perspective*, 2nd ed., Springer, New York, 2005.

[6] P. Erdős, *On almost primes*, Amer. Math. Monthly **57** (1950), 404–407.

[7] P. Erdős, *On pseudoprimes and Carmichael numbers*, Publ. Math. Debrecen **4** (1956), 201–206.

[8] M. R. Fellows and N. Koblitz, *Self-witnessing polynomial-time complexity and prime factorization*, Designs, Codes, and Cryptography **2** (1992), 231–235.

[9] M. Fürer, *Deterministic and Las Vegas primality testing algorithms*, in Proceedings of ICALP 85 (July 1985). Nafplion, Greece. W. Brauer, ed., Lecture Notes in Computer Science **194**, Springer-Verlag, Berlin, 1985, pp. 199–209.

[10] S. V. Konyagin and C. Pomerance, *On primes recognizable in deterministic polynomial time*, in The mathematics of Paul Erdős, R. L. Graham and J. Nešetřil, eds., Springer-Verlag, Berlin, 1997, pp. 176–198.

[11] H. W. Lenstra, jr and Carl Pomerance, *Primality testing with Gaussian periods*, www.math.dartmouth.edu/∼carlp/aks041411.pdf.

[12] C. Pomerance, *Primality testing: variations on a theme of Lucas*, Congressus Numerantium **201** (2010), 301–312.

[13] B. Źrałek, *A deterministic version of Pollard's $p-1$ algorithm*, Math. Comp. **79** (2010), 513–533.

Steklov Institute of Mathematics, 8 Gubkin Street, Moscow 119991, Russia
E-mail address: `konyagin23@gmail.com`

Dartmouth College, Department of Mathematics, Hanover, NH 03755, USA
E-mail address: `carlp@math.dartmouth.edu`