On Derandomizing Tests for Certain Polynomial Identities

Manindra Agrawal
Dept of CSE, IIT Kanpur 208016, India
email: manindra@iitk.ac.in

Abstract

We extract a paradigm for derandomizing tests for polynomial identities from the recent AKS primality testing algorithm. We then discuss its possible application to other tests.

1 Introduction

Polynomial identity testing has been in news recently due to two major results proved last year: Kabanets and Impagliazzo [3] proved that any derandomization of a randomized algorithm for the problem results in a lower bound on arithmetic circuits, while Agrawal, Kayal, and Saxena [2] gave a complete derandomization of the test for a specific identity resulting in a deterministic polynomial-time algorithm for primality testing. Viewed optimistically, these results suggest that lower bounds on arithmetic circuits might be easier to obtain than we believe. In this paper, we try to identify a possible way of doing this. We first cast the primality testing algorithm of [2] as constructing a pseudo-random generator against the randomized test for an identity. From that we extract a paradigm for derandomization of a randomized test for polynomial identities. We then discuss an application of this paradigm to testing existence of perfect matchings in a bipartite graph. We also list some interesting consequences of our view of the primality testing algorithm, including a computationally efficient characterization of prime numbers, and an extremely short and simple (though not so efficient) algorithm for primality testing.

2 Definitions

We use Z_n to denote the ring of numbers modulo n, F_p to denote the field of numbers modulo p. For a ring R, $R[X]$ is the ring of polynomials with coefficients in R. For Z_n, polynomial equation $P(X) = P'(X) \pmod{Q(X), n}$ means that $P(X) - P'(X)$ leaves zero remainder when coefficients are reduced modulo n and powers of X are reduced modulo $Q(X)$.

For any two numbers n and r such that $(r, n) = 1$ (i.e., the numbers are relatively prime), $o_r(n)$ equals the order of n modulo r. This number always divides $\phi(r)$, the Euler's totient function.

3 A Characterization of Prime Numbers

The primality testing algorithm of [2] is based on the following partial characterization of prime numbers:

Lemma 3.1 *[2]*

- *If n is prime then for every a and r:*

  $$\textit{Prime}(X, a) = (X+a)^n - X^n - a = 0 \pmod{X^r - 1, n}.$$

- *If for some r such that $o_r(n) > 4\log^2 n$, and for every a, $1 \leq a \leq 2\sqrt{\phi(r)}\log n$:*

  $$\textit{Prime}(X, a) = 0 \pmod{X^r - 1, n},$$

  then either every prime divisor of n is less then r, or n is a prime power.

To turn it into a complete characterization, we need to remove the possibilities that n has small divisors and that n is a non-trivial power of a prime. The first one of these can be eliminated by testing if

$$\text{Prime}(X, 1) = 0 \pmod{X^r, n}.$$

This follows from the following lemma:

Lemma 3.2 *If $\textit{Prime}(X, 1) = 0 \pmod{X^r, n}$ then either n is prime or every prime divisor of n is at least r.*

Proof. If $n < r$ then we have

$$\begin{aligned} \text{Prime}(X, 1) \pmod{X^r, n} &= (X+1)^n - X^n - 1 \pmod{X^r, n} \\ &= (X+1)^n - X^n - 1 \pmod{n}. \end{aligned}$$

And we know that $\text{Prime}(X, 1) = 0 \pmod{n}$ iff n is prime.

Now suppose that $n \geq r$. Let prime p divide n with $p < r$. Then,

$$\begin{aligned}
&\text{Prime}(X,1) \pmod{X^r, n} \\
&= (X+1)^n - X^n - 1 \pmod{X^r, n} \\
&= (X+1)^n - 1 \pmod{X^r, n} \\
&= \sum_{j=1}^{r-1} \binom{n}{j} X^j \pmod{n}.
\end{aligned}$$

Since $p < r$, the coefficient of X^p is $\binom{n}{p}$ in the above equation. It is easy to see that $\binom{n}{p} = \pm\frac{n}{p} \neq 0 \pmod{n}$. ■

To eliminate the possibility of n being a non-trivial prime power, we now simply need to test for a few more a's:

Lemma 3.3 *If* $\text{Prime}(X,a) = 0 \pmod{X-1, n}$ *for every* a, $1 \leq a \leq \log^4 n$, *and every prime divisor of* n *is greater than* $4\log^4 n$, *then* n *is square free.*

Proof. We have $\text{Prime}(X,a) = 0 \pmod{X-1, n}$ for every a, $1 \leq a \leq \log^4 n$. This gives:

$$(1+a)^n = 1 + a^n \pmod{n}$$

for $1 \leq a \leq \log^4 n$. Expanding the RHS of the equation we get:

$$(1+a)^n = 1 + a \pmod{n}$$

for $1 \leq a \leq \log^4 n$.

Now suppose that n is not square-free. Let p^2 divide n for some prime p. Then:

$$k^n = k \pmod{p^2}$$

for $1 \leq k \leq \log^4 n$. Since $p > 4\log^4 n$, $(k,p) = 1$ for $k \leq \log^4 n$. Therefore,

$$k^{n-1} = 1 \pmod{p^2}$$

for $1 \leq k \leq \log^4 n$. As the order of multiplicative group $Z^*_{p^2}$ is $\phi(p^2) = p(p-1)$, we get:

$$k^{(n-1,p(p-1))} = 1 \pmod{p^2}$$

for $1 \leq k \leq \log^4 n$. As p cannot divide $n-1$, we get:

$$k^{p-1} = 1 \pmod{p^2}$$

for $1 \leq k \leq \log^4 n$.

Notice that there exist at most p roots of the equation $x^p = x \pmod{p^2}$ in $Z^*_{p^2}$: if u is a root, then $(u+kp)^p = u^p \pmod{p^2} = u \pmod{p^2}$ and so $u + kp$ cannot be a root for $k \neq 0 \pmod{p}$; this implies that every root must lie in a different residue class modulo p.

Finally, we observe that if $k^p = k \pmod{p^2}$ for $1 \leq k \leq log^4 n$, then there exist more than p roots of the equation $x^p = x \pmod{p^2}$ thus arriving at a contradiction. Let $q_1, q_2, \ldots, q_t$ be all primes less than $\log^4 p < \log^4 n$. By Chebyshev's estimate, we know that $t \geq \frac{\log^4 p}{24 \log\log p}$. The product of a collection of up to $\ell = \frac{\log p}{2\log\log p}$ of these primes is less than $(\log^4 p)^\ell = p^2$. Therefore, all such products will be distinct modulo p^2. Let $m_1, m_2, \ldots, m_T$ be the numbers formed by such products. We have:

$$m_j^p = m_j \pmod{p^2},$$

for $1 \leq j \leq T$, since all the prime factors of m_j satisfy the equation. The number T is at least

$$\begin{aligned}
\binom{t}{\ell} &> \left(\frac{t}{\ell}\right)^\ell \\
&\geq \left(\frac{\log^3 p}{12}\right)^{\frac{\log p}{2\log\log p}} \\
&= p^{\frac{3}{2} - \frac{\log 12}{2\log\log p}}.
\end{aligned}$$

Since $n \geq p^2 \geq 16\log^8 n$, p is greater than 2^{16}. Therefore, $T > p$. ■

Putting together the above lemmas, we get the following characterization of primes.

Corollary 3.4 n *is prime if and only if for any* $r > 4\log^4 n$ *such that* $o_r(n) > 4\log^2 n$: $\text{Prime}(X,1) = 0 \pmod{X^r, n}$ *and for every* a, $1 \leq a \leq 4\log^4 n$: $\text{Prime}(X,a) = 0 \pmod{X^r - 1, n}$.

Proof. It was shown in [2] that there always exists an $r \leq 16\log^5 n$ such that $o_r(n) > 4\log^2 n$. Their proof also yields that there exists such an r with $4\log^4 n < r \leq 16\log^5 n$. Lemma 3.1 shows that if $\text{Prime}(X,a) = 0 \pmod{X^r - 1, n}$ for such an r and for every $1 \leq a \leq 2\sqrt{\phi(r)}\log n < 4\log^4 n$, then n is either a prime power or has divisors less than r. Lemma 3.2 rules out divisors less than r and Lemma 3.3 rules out n being a prime power. Hence n must be prime when all the conditions are satisfied. ■

4 Converting to a Single Identity

Although Corollary 3.4 provides a characterization of prime numbers, it is a bit unwieldy due to presence of multiple identities. Notice that all the identities are very similar—the only difference being the value of a used. We can easily transform them into a single identity.

Lemma 4.1 *Fix any $r > 0$ and any $\ell > 0$. Then,*

$$\textit{Prime}(X,1) = 0 \pmod{(X-a)^r - 1, n} \textit{ for } 0 \le a \le \ell - 1 \quad (1)$$

if and only if

$$\textit{Prime}(X,a) = 0 \pmod{X^r - 1, n} \textit{ for } 1 \le a \le \ell. \quad (2)$$

Proof. The proof is by induction on ℓ. When $\ell = 1$, equation (1) is:

$$\text{Prime}(X,1) = 0 \pmod{X^r - 1, n}.$$

This is identical to equation (2) for $a = 1$.

Now suppose the equivalence holds for $\ell - 1$. So we have:

$$\begin{aligned} \text{Prime}(X,1) &= 0 \pmod{(X-a)^r - 1, n} \text{ for } 0 \le a \le \ell - 2 \\ &\text{iff} \\ \text{Prime}(X,a) &= 0 \pmod{X^r - 1, n} \text{ for } 1 \le a \le \ell - 1. \end{aligned}$$

To prove the forward direction, assume that

$$\text{Prime}(X,1) = 0 \pmod{(X-a)^r - 1, n} \text{ for } 0 \le a \le \ell - 1.$$

Then, for $a = \ell - 1$, we have:

$$(X+1)^n = X^n + 1 \pmod{(X - \ell + 1)^r - 1, n}.$$

Substituting $X + \ell - 1$ for X, we get:

$$(X+\ell)^n = (X + \ell - 1)^n + 1 \pmod{X^r - 1, n}.$$

Using the equivalence for $\ell - 1$, we can replace the RHS of the last equation by $X^n + \ell - 1$. Thus, we get equation (2) for $a = \ell$.

To prove the other direction, assume that

$$\text{Prime}(X,a) = 0 \pmod{X^r - 1, n} \text{ for } 1 \le a \le \ell.$$

Then, for $a = \ell$:

$$\begin{aligned} (X+\ell)^n &= X^n + \ell \pmod{X^r - 1, n} \\ &= (X + \ell - 1)^n + 1 \pmod{X^r - 1, n}. \end{aligned}$$

Now substituting $X - \ell + 1$ for X, we get equation (1) for $a = \ell - 1$. ∎

Corollary 4.2 *n is prime if and only if for any $r > 4\log^4 n$ such that $o_r(n) > 4\log^2 n$: Prime$(X,1) = 0 \pmod{Q(X), n}$ for every $Q(X) \in \{X^r, (X-a)^r - 1 \mid 1 \le a \le 4\log^4 n\}$.*

Proof. Follows directly from Corollary 3.4 and above lemma. ∎

We can simplify it further by using the fact (proved in [2]) that there always exists an r, $r \le 16\log^5 n$ such that $o_r(n) > 4\log^2 n$.

Corollary 4.3 *n is prime if and only if Prime$(X,1) = 0 \pmod{Q(X), n}$ where*

$$Q(X) = X^{16\log^5 n} \cdot \prod_{r=1}^{16\log^5 n} \prod_{a=1}^{4\log^4 n} ((X-a)^r - 1).$$

Proof. Follows from the fact above and the (trivial) observation that $f(X)$ is divisible by $g(X)$ implies it is divisible by all factors of $g(X)$. ∎

Corollary 4.3 gives rise to a two line algorithm for testing primality:

```
Input:  integer n > 1.
1.  Compute
        Q(X) = X^(16 log^5 n) . ∏_{r=1}^{16 log^5 n} ∏_{a=1}^{4 log^4 n} ((X-a)^r - 1).
2.  Output PRIME iff
        (X+1)^n = X^n + 1 (mod Q(X), n).
```

Of course, the time complexity of this algorithm is very high: $O(\log^{17} n)$!

5 A Paradigm for Derandomization

Corollary 4.2 fits exactly in the framework of the randomized algorithm for identity testing given in [1]. Their algorithm for testing a polynomial identity is as follows:

> Let $P(Z_1, Z_2, \ldots, Z_m)$ be a polynomial over field F_p of degree d_i in Z_i specified by an arithmetic circuit. Let $\hat{P}(X) = P(X^{D_0}, X^{D_1}, \ldots, X^{D_{m-1}})$ where $D_k = \prod_{i=1}^{k}(d_i + 1)$ for $1 \le k \le m$, and $D_0 = 1$. For any r such that $o_r(p) \ge \log D_m$, and for a randomly chosen polynomial $T(X)$ of degree $\log D_m$, output ZERO iff $\hat{P}(X) = 0 \pmod{(T(X))^r - 1, p}$.

It is shown there that the above algorithm succeeds with probability at least $1 - \frac{1}{\log D_m}$ when the input polynomial is not identically zero. Corollary 4.2 has exactly the same form except for some minor differences (like identity being tested over ring Z_n instead of field F_p). The reason why it yields a deterministic algorithm is that the sample space for random polynomial $T(X)$ has been reduced to a

polynomial sized subset: $T(X)$ takes the value $X - a$ for a small number is a's. (Again, there is a minor difference in that Corollary 4.2 also requires to test modulo X^r.) So the primality test of [2] can be viewed as derandomizing the randomized primality testing of [1] in a precise way.

This also suggests the following paradigm for derandomizing identity tests.

> Let $P(Z_1, Z_2, \ldots, Z_m)$ be a polynomial over ring R of degree d_i in Z_i specified by an arithmetic circuit. Let $\hat{P}(X) = P(X^{D_0}, X^{D_1}, \ldots, X^{D_{m-1}})$ where $D_k = \prod_{i=1}^{k}(d_i+1)$ for $1 \leq k \leq m$, and $D_0 = 1$. Construct a small sample space for $Q(X)$—a polynomial of degree bounded by a polynomial in $\log D_m$ and m. Show that $\hat{P}$ is zero iff it is zero modulo $Q(X)$ for every $Q(X)$ in the sample space.

It is easy to observe (see, e.g., [1]) that $P(Z_1, \ldots, Z_m)$ is zero iff $\hat{P}(X)$ is zero. And since there are only a few low degree polynomials in the sample space, identity $P(Z_1, \ldots, Z_m)$ can be efficiently deterministically tested.

Can this paradigm be used to derandomize tests for some other identities? We examine one such identity: for testing bipartite matching. Let $G = (U, V, E)$ be a bipartite graph with $|U| = |V| = n$. Define $n \times n$ matrix $M = [m_{ij}]$ as: $m_{ij} = \epsilon_{ij} \cdot X_{ij}$ with $\epsilon_{ij} = 1$ if edge $(i, j) \in E$, 0 otherwise. It was shown by Lovasz [5] that G has a perfect matching iff $\det(M) \neq 0$. Using this characterization, a simple randomized NC algorithm for matching can be derived since $\det(M)$ is an n^2-variate multi-linear polynomial.

Let us try to apply our paradigm to this algorithm. First, we convert this to a univariate identity by making the substitution $X_{ij} \mapsto X^{2^{n^3 i+j}}$. This preserves the characterization. Let $\hat{M}$ be the resulting matrix. Now, instead of choosing a random small degree polynomial $Q(X)$, we choose $Q(X)$ from the set of polynomials $\{X^r - 1 \mid 1 \leq r \leq n^6\}$. Notice that with this choice of $Q(X)$, $\hat{M} \pmod{Q(X)}$ can be evaluated in NC: if $Q(X) = X^r - 1$, then first compute $u_{ij} = 2^{n^3 i+j} \pmod{r}$ for $1 \leq i, j \leq n$ (this can be done by an NC^1 circuit), then construct matrix $\tilde{M} = [\tilde{m}_{ij}]$ with $\tilde{m}_{ij} = X^{u_{ij}}$, and evaluate $\det(\tilde{M})$ (this can now be done in NC since the determinant is a univariate degree $< n^7$ polynomial), and finally reduce the resulting polynomial modulo $X^r - 1$. Therefore, the entire computation, modulo each $Q(X)$ in the sample space can be done in NC.

Now we conjecture the following:

Conjecture. *Graph G has a perfect matching iff for some $Q(X) \in \{X^r - 1 \mid 1 \leq r \leq n^6\}$, $\det(\hat{M}) \neq 0 \pmod{Q(X)}$.*

In an ongoing work (with S. Biswas, V. Pandey, and R. Verma) we are able to show the following. Let Π be the set of all matchings of graph G expressed as permutations of $[1, n]$. Let $\mathrm{sgn}(\cdot)$ be the sign function of permutations of $[1, n]$. Define polynomials $S_k(Y, Z)$ as:

$$S_k(Y, Z) = \sum_{\pi \in \Pi} \mathrm{sgn}(\pi) \cdot \left(\sum_{i=1}^{n} Y^i Z^{\pi(i)} \right)^k .$$

We can prove that:

Lemma 5.1 *If $\det(\hat{M}) = 0 \pmod{X^r - 1}$ for every r, $1 \leq r \leq n^6$, then $S_k(Y, Z) = 0$ for every k, $0 \leq k \leq \binom{n}{2}$.*

So if it can be shown that, G has a perfect matching implies $S_k(Y, Z) \neq 0$ for some $k \leq \binom{n}{2}$, then we are done. Our experiments (though not very extensive) suggest that this is indeed the case. We can also prove this for several special classes of graphs. For example, if G is a complete bipartite graph, then $S_k(Y, Z) = 0$ for $k < \binom{n}{2}$ and $S_{\binom{n}{2}}(Y, Z) \neq 0$. For non-complete graphs, the smallest value of k for which $S_k(Y, Z) \neq 0$ is observed to be less than $\binom{n}{2}$.

The paradigm can similarly be applied to other special identities, e.g., identity for testing equivalence of read-once branching programs.

6 Future Work

Improving the time complexity of the primality testing algorithm remains a major problem. Recently, Lenstra and Pomerance [4] have brought down the time complexity to $O^{\sim}(\log^6 n)$. A conjecture given at the end of [2] improves this to $O^{\sim}(\log^3 n)$. However, as recently observed by Lenstra and Pomerance, this conjecture is unlikely to be true.

For the paradigm for derandomization that we have identified, much work needs to be done to clarify its utility. Our way of derandomization is different from the one given in [3] where derandomization is done making use of a hard function for arithmetic circuits—this follows the usual methodology of deriving pseudo-random generators from hard functions initiated by Nisan and Wigderson [6]. One interesting question here is to see if our paradigm can also be put in this way. Specifically, can one show that the small sample space for $Q(X)$ can be derived using a hard function? And conversely, if a small sample space for $Q(X)$ derandomizes all the identities, then can one construct a hard function from such a sample space?

Acknowledgment

I thank Hendrik Lenstra for pointing out Lemma 3.2 and Corollary 4.3.

References

[1] M. Agrawal and S. Biswas. Primality and identity testing via chinese remaindering. In *Proceedings of Annual IEEE Symposium on Foundations of Computer Science*, pages 202–209, 1999.

[2] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. Preprint (http://www.cse.iitk.ac.in/news/primality_v3.ps), February 2003.

[3] V. Kabanets and R. Impagliazzo. Derandomizing polyonmial identity test means proving circuit lower bounds. to appear in STOC 2003; TR02-055 (http://www.eccc.uni-trier.de/eccc-local/Lists/TR-2002.html), 2002.

[4] H. W. Lenstra, Jr. and C. Pomerance. Primality testing with gaussian periods. Private communication, March 2003.

[5] L. Lovasz. On determinants, matchings, and random algorithms. In L. Budach, editor, *Fundamentals of Computing Theory*. Akademia-Verlag, 1979.

[6] N. Nisan and A. Wigderson. Hardness vs. randomness. *J. Comput. Sys. Sci.*, 49(2):149–167, 1994.