AKS素数测试

维基百科，自由的百科全书

AKS素数测试（又被称为Agrawal–Kayal–Saxena素数测试和Cyclotomic AKS test）是一个决定型素数测试算法 ，由三个来自Indian Institute of Technology Kanpur的计算机科学家，Manindra Agrawal、Neeraj Kayal和Nitin Saxena，在2002年8月6日发表于一篇题为素数属于P的论文。[1]作者们因此获得了许多奖项，包含了2006年的哥德尔奖和2006年的Fulkerson Prize。这个算法可以在多项式时间之内，决定一个给定整数是素数或者合数。

目录

重要性

AKS最关键的重要性在于它是第一个被发表的一般的、多项式的、确定性的和无仰赖的素数判定算法。先前的算法至多达到了其中三点，但从未达到全部四个。

- AKS算法可以被用于检测任何一般的给定数字是否为素数。很多已知的高速判定算法只适用于满足特定条件的素数。例如，卢卡斯-莱默检验法仅对梅森素数适用，而Pépin测试仅对费马数适用。
- 算法的最长运行时间可以被表为一个目标数字长度的多项式。ECPP和APR能够判断一个给定数字是否为素数，但无法对所有输入给出多项式时间范围。
- 算法可以确定性地判断一个给定数字是否为素数。随机测试算法，例如米勒-拉宾检验和Baillie–PSW，可以在多项式时间内对给定数字进行校验，但只能给出概率性的结果。
- AKS算法并未“仰赖”任何未证明猜想。一个反例是确定性米勒检验：该算法可以在多项式时间内对所有输入给出确定性结果，但其正确性却基于尚未被证明的广义黎曼猜想。

概念

AKS 素数测试主要是基于以下定理：整数n (≥ 2)是素数，当且仅当

$$(x-a)^{n} \equiv (x^{n}-a) \pmod{n} \qquad (1)$$

这个同余多项式对所有与n互素的整数a均成立。 这个定理是费马小定理的一般化，并且可以简单的使用二项式定理跟二项式系数的这个特征:

$$\binom{n}{k} \equiv 0 \pmod{n}$$，對任何 $0<k<n$，若且唯若 n 是質數

来证明出此定理。

虽然说关系式 (1) 基本上构成了整个素数测试，但是验证花费的时间却是指数时间。因此，为了减少计算复杂度, AKS改为使用以下的同余多项式：

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

这个多项式与

存在多项式 f 與 g，令:$(x-a)^n - (x^n - a) = nf + (x^r - 1)g \qquad (3)$

意义是等同的。

这个同余式可以在多项式时间之内检查完毕。这里我们要注意所有的素数必定满足此条件式 (令 g = 0 则 (3) 等于 (1)，因此符合 n 必定是素数)。 然而，有一些合数也会满足这个条件式。有关AKS正确性的证明包含了推导出存在一个够小的r以及一个够小的整数集合A，令如果此同余式对所有A里面的整数都满足，则n必定为素数。

历史以及运算时间

在上文引用的论文的第一版本中，作者们证明了算法的渐近时间为Õ$(\log^{12}(n))$。换言之，算法使用少于n的数字长度的十二次方乘以一个（数字长度的）多重对数。但是，论文证明的时间上界却过于宽松；事实上，一个被普遍相信的关于索菲热尔曼素数分布的假设如果为真，则会立即将最坏情况减至Õ$(\log^{6}(n))$。

在这一发现后的几个月中，新的变体陆续出现（Lenstra 2002, Pomerance 2002, Berrizbeitia 2003, Cheng 2003, Bernstein 2003a/b, Lenstra和Pomerance 2003）并依次提高了算法的速度（以改进幅度为序）。由于这些变体的出现，Crandall和Papadopoulos在其科学论文“AKS-类素数测试的实现”（2003年三月发表）中将其称为算法的“AKS-类”。

出于对这些变体和其他回复的回应，论文“素数属于P”稍后被进行了更新，新版本包括了一个AKS算法的正规公式化表述和其正确性证明。（这一版本在Annals of Mathematics上发表。）虽然基本思想没有变化，r却被采用了新方法进行选择，而正确性证明也变得更加紧致有序。与旧证明依赖于许多不同的方法不同，新版本几乎只依赖于有限域上的分圆多项式的特征。新版本同时也优化了时间复杂度的边界到Õ$(\log^{10.5}(n))$。通过筛法获得的其他结果可以将其进一步简化到Õ$(\log^{7.5}(n))$。

在2005年，Carl Pomerance和H. W. Lenstra, Jr.展示了一个AKS的变体，可以在Õ($\log^6$(n))次操作内完成测试（n是被测试数）。对于原算法的Õ($\log^{12}$(n))边界而言，这是一个显著的改进。[2]

算法

整个算法的操作如下：[1]

输入：整数 n > 1

1. 若存在整数a > 0 且b > 1 ，令 n = a^b ；则输出合数
2. 找出最小的 r 令 $\mathrm{ord}_r(n) > \log_2{}^2(n)$.
3. 若 对某些a ⩽ r，1 < gcd(a,n) < n，输出合数。(gcd是指最大公约数)。
4. 若 n ⩽ r, 输出素数。
5. 对 a = 1 到 $\lfloor\sqrt{\varphi(r)}\log(n)\rfloor$的所有数，

如果 $(X+a)^n \neq X^n+a \pmod{X^r - 1, n}$，输出合数。

6. 输出 素数。

这里的 $\mathrm{ord}_r(n)$是n mod r的阶。 另外，这里的log 代表以二为底的对数，$\varphi(r)$则是r的欧拉函数。

下面说明若n是个素数，那么算法总是会返回素数：由于n是素数，步骤1和3永远不会返回合数。步骤5也不会返回合数，因为(2)对所有素数n为真。因此，算法一定会在步骤4或6返回素数。

对应地，如果n是合数，那么算法一定返回合数：如果算法返回素数，那么则一定是从步骤4或6返回。对于前者，因为n ≤ r， n必然有因子a ≤ r符合1 < gcd(a,n) < n，因此会返回合数。剩余的可能性就是步骤6，在文章[1]中，这种情况被证明不会发生，因为在步骤5中检验的多个等式可以确保输出一定是合数。

相关页面

1. ^ 1.0 1.1 1.2 Manindra Agrawal, Neeraj Kayal, Nitin Saxena, "PRIMES is in P (http://www.cse.iitk.ac.in/users/manindra/algebra/primality_v6.pdf)", Annals of Mathematics 160 (2004), no. 2, pp. 781–793.
2. ^ H. W. Lenstra, Jr. and Carl Pomerance, "Primality Testing with Gaussian Periods (http://www.math.dartmouth.edu/~carlp/PDF/complexity12.pdf)", preliminary version July 20, 2005.

外部连接

- 埃里克·韦斯坦因, AKS Primality Test at MathWorld
- R. Crandall, Apple ACG, and J. Papadopoulos (March 18, 2003): On the implementation of AKS-class primality tests (http://developer.apple.com/hardware/ve/pdf/aks3.pdf) (PDF)
- Article by Borneman, containing photos and information about the three Indian scientists (http://www.ams.org/notices/200305/fea-bornemann.pdf) (PDF)
- Andrew Granville: It is easy to determine whether a given integer is prime (http://www.ams.org/bull/2005-42-01/S0273-0979-04-01037-7/home.html)
- The Prime Facts: From Euclid to AKS (http://www.scottaaronson.com/writings/prime.pdf), by Scott Aaronson (PDF)
- The PRIMES is in P little FAQ (http://www.instantlogic.net/publications/PRIMES%20is%20in%20P%20little%20FAQ.htm) by Anton Stiglic
- 2006 Gödel Prize Citation (http://sigact.acm.org/prizes/godel/2006.html)
- 2006 Fulkerson Prize Citation (http://www.ams.org/notices/200611/comm-fulkerson.pdf)
- [1] (http://terrytao.wordpress.com/2009/08/11/the-aks-primality-test/)
- The AKS "PRIMES in P" Algorithm Resource (http://fatphil.org/maths/AKS)

取自“http://zh.wikipedia.org/w/index.php?title=AKS質數測試&oldid=29786869”
