PRIMALITÉ THÉORIQUE ET PRIMALITÉ PRATIQUE OU AKS VS. ECPP

F. MORAIN

1. Introduction

Le but de cette note est de présenter l'algorithme de primalité déterministe en temps polynomial dû à Agrawal, Kayal et Saxena [2] et de commenter son application à la preuve de primalité pratique. Les ouvrages traitant de ces sujets sont nombreux. Citons l'incontournable [16], le vaste [7], et le récent [8].

2. Primalité et complexité

Depuis l'invention de la théorie de la complexité, le problème PRIME (un nombre donné N est-il premier ?) a servi de cas d'école dans l'établissement de la hiérarchie des classes de complexité. PRIME a été démontré être dans $\mathbf{NP} \cap \mathbf{coNP}$ par Pratt [15], en utilisant la notion de certificat. Plus tard, Adleman et Huang ont montré que PRIME était dans $\mathbf{RP}$, en utilisant des courbes hyperelliptiques de genre 2 [1].

Le premier algorithme déterministe de primalité est dû à Miller [14]. Si une hypothèse de Riemann est vraie, alors le temps de calcul pour prouver que N est premier est $O((\log N)^5)$. L'algorithme AKS quant à lui ne repose sur aucune hypothèse, et a un temps de calcul $\tilde{O}((\log N)^{12})$. Le problème PRIME est donc dans la classe de complexité $\mathbf{P}$.

3. L'algorithme de Miller

L'article original est [14]. On consultera avec profit [12, 13].

Le théorème d'Euler nous dit que si N est un nombre premier et a premier avec N, alors

$$a^{(N-1)/2} \equiv \left(\frac{a}{N}\right) \bmod N \tag{1}$$

où $\left(\frac{a}{N}\right)$ désigne le symbole de Jacobi. Le nombre a étant fixé, un nombre composé N vérifiant (1) est appelé *nombre pseudopremier d'Euler en base a*. On pose

$$A_N = \{a \in (\mathbb{Z}/N\mathbb{Z})^*, a^{(N-1)/2} \equiv \left(\frac{a}{N}\right) \bmod N\}.$$

Si N est premier, alors $A_N = (\mathbb{Z}/N\mathbb{Z})^*$. Par contre, Lehmer a montré [10], que si N est composé, A_N est un sous-groupe propre de $(\mathbb{Z}/N\mathbb{Z})^*$.

Le test de composition de Solovay et Strassen pour le nombre N consiste à choisir des valeurs aléatoires de a et de tester si (1) est vérifié.

Miller est allé plus loin. Si une hypothèse de Riemann est vraie, alors le plus petit élément de $(\mathbb{Z}/N\mathbb{Z})^*$ qui n'est pas dans A_N est plus petit que $c(\log N)^2$. Bach [4] a montré que $c = 2$ était suffisant. L'algorithme est alors :

Date: 4 octobre 2002.

fonction MILLER(N)
1. **pour** $a = 2$ **à** $2 \cdot (\log N)^2$ **faire**
   si l'équation (1) n'est pas satisfaite **alors** retourner **faux**;
2. retourner **vrai**.

Remarquons qu'on peut réduire le nombre de calculs à faire en exploitant la structure de groupe de A_N, ce qui conduit à ne considérer que des a premiers.

4. L'ALGORITHME AKS

L'article original est [2]. Dan Bernstein en a écrit une version courte [6] dont nous nous inspirons ici pour la démonstration du théorème.

4.1. Le théorème fondamental.

Théorème 4.1. *Soient N un entier, s un entier positif, r un nombre premier et q le plus grand facteur premier de $r-1$. On suppose que*

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

On suppose encore que N n'a pas de facteur premier $\leqslant s$ et que $N^{(r-1)/q} \bmod r \notin \{0,1\}$. Finalement, on suppose également que $(X-b)^N = X^N - b$ dans $\mathbb{Z}/N\mathbb{Z}[X]/(X^r-1)$ pour tout $1 \leqslant b \leqslant s$. Alors N est une puissance de nombre premier.

Démonstration : il existe un facteur premier p de N tel que $p^{(r-1)/q} \bmod r \notin \{0,1\}$, car sinon cela contredirait l'hypothèse sur N lui-même. L'ordre de p modulo r est plus grand que q par construction.

On considère maintenant un facteur irréductible $h(X)$ du r-ième polynôme cyclotomique $\Phi_r(X) = X^{r-1} + X^{r-2} + \cdots + 1$.

Lemme 4.1. *Le degré de h est au moins q.*

Démonstration : soit d le degré de $h(X)$. D'après la théorie des corps finis, on sait que $h(X) \mid X^{p^d} - X$. Comme $h(X) \mid X^r - 1$, on en déduit que

$$h(X) \mid \mathrm{pgcd}(X^r - 1, X^{p^d} - X) = X^{\mathrm{pgcd}(r, p^d-1)} - 1$$

d'après un calcul classique. Si $d < q$, alors $h(X) \mid X-1$, donc lui est égal, mais $\Phi_r(1) = r \not\equiv 0 \bmod p$. $\square$

Soit $F = \mathbb{F}_p[X]/(h(X))$, qui est un corps fini. On note G le sous-groupe de F^* engendré par les $(X-k)^e$ pour $1 \leqslant k \leqslant s$:

$$G = \left\{\prod (X-1)^{e_1}(X-2)^{e_2}\cdots(X-s)^{e_s} \bmod h(X)\right\}.$$

Lemme 4.2. *Le groupe G est cyclique et a au moins $\binom{q+s-1}{s}$ éléments.*

Démonstration : G est cyclique puisque sous-groupe d'un groupe cyclique. D'autre part, G contient au moins tous les $(X-1)^{e_1}(X-2)^{e_2}\cdots(X-s)^{e_s}$ avec

$$e_1 + e_2 + \cdots + e_s \leqslant q - 1 < \deg(h),$$

ces polynômes étant tous distincts. En effet $X-a$ est distinct de $X-b$ pour $a, b \leqslant s$ car les diviseurs de N plus petits que s ont été éliminés. $\square$

Il nous faut alors trouver un générateur de G, soit $(X-1)^{e_1}(X-2)^{e_2}\cdots(X-s)^{e_s} \bmod h(X)$. On pose $g(X) = (X-1)^{e_1}(X-2)^{e_2}\cdots(X-s)^{e_s}$ dans $\mathbb{F}_p[X]$. L'ordre de g modulo h est le cardinal de G, puisque celui-ci est cyclique.

Lemme 4.3. *Le polynôme g vérifie $g(X)^N = g(X^N) \bmod (X^r - 1, N)$.*

Démonstration : par hypothèse, $(X-b)^N = X^N - b \bmod (X^r - 1, N)$ pour $1 \leqslant b \leqslant s$ et par suite :

$$g(X)^N \equiv ((X-1)^{e_1}(X-2)^{e_2} \cdots (X-s)^{e_s})^N$$
$$\equiv (X^N-1)^{e_1}(X^N-2)^{e_2} \cdots (X^N-s)^{e_s} = g(X^N). \square$$

On définit un ensemble d'entiers T par

$$T = \{e, g(X)^e \equiv g(X^e) \bmod (X^r - 1, p)\}.$$

On vient de voir que $N \in T$. Comme p est premier, il est lui aussi dans T, tout comme 1.

Lemme 4.4. *L'ensemble T est un monoïde multiplicatif.*

Démonstration : si $g(X)^f = g(X^f) \bmod (X^r - 1, p)$, alors

$$g(X^e)^f = g(X^{ef}) \bmod (X^{er} - 1, p)$$

ce qui implique que $g(X^e)^f = g(X^{ef}) \bmod (X^r - 1, p)$ car $X^r - 1 \mid X^{er} - 1$. D'autre part, $g(X)^{ef} = (g(X)^e)^f = g(X^e)^f = g(X^{ef})$. $\square$

Corollaire 4.1. *Le groupe T contient tous les produits $N^i p^j$.*

Fin de la démonstration du théorème : considérons tous les produits $N^i p^j$ avec $0 \leqslant i, j \leqslant \lfloor\sqrt{r}\rfloor$. Il y a $(\lfloor\sqrt{r}\rfloor + 1)^2 > r$ tels nombres. Il existe donc deux couples distincts (i, j) et (k, ℓ) tels que $N^i p^j \equiv N^k p^\ell \bmod r$. Posons $t = N^i p^j$ et $u = N^k p^\ell$. Par construction :

$$g(X^t) = g(X^u) \bmod (X^r - 1)$$

ou encore

$$g(X)^t = g(X)^u \bmod (X^r - 1)$$

c'est-à-dire encore $g(X)^t = g(X)^u \bmod h(X)$, ce qui veut dire que $t \equiv u \bmod \operatorname{ord}(g)$. Mais t et u sont bornés par $N^{2\lfloor\sqrt{r}\rfloor} < \binom{q+s-1}{s} \leqslant \operatorname{ord}(g)$. Par suite, $t = u$ et donc $N^{i-k} = p^{j-\ell}$. $\square$

4.2. **Le choix de r et s.** Examinons la condition (2). Les auteurs minorent brutalement

$$\binom{q+s-1}{s} > \left(\frac{q}{s}\right)^s$$

et imposent ensuite $q \geqslant 2s$, puis dans la foulée :

$$2^s \geqslant N^{2\lfloor\sqrt{r}\rfloor}$$

ce qui donne une solution au problème avec

$$\frac{r-1}{2} \geqslant q = 2s \geqslant 4\lfloor\sqrt{r}\rfloor \log_2 N.$$

Remarquons que cela implique de fait $N > r \geqslant 64(\log_2 N)^2$, ce qui veut dire que $N \geqslant 11689$ et que 89* ne pourra être prouvé premier par cette méthode.

Il reste à s'assurer de l'existence d'un nombre premier r convenable. Pour cela, les auteurs utilisent des résultats de Fouvry [9] ainsi que de Baker et Harman [5] sur la taille du plus grand facteur premier de $P - 1$ quand P est premier :

Proposition 4.1. *Il existe deux constantes c_1 et c_2 telles qu'il existe un nombre premier r dans l'intervalle $[c_1(\log N)^6, c_2(\log N)^6]$ tel que $r-1$ ait un facteur premier $q \geqslant 4\sqrt{r}\log_2 N$ et pour lequel l'ordre de N modulo r soit divisible par q.*

*private joke

4.3. **L'algorithme.** Donnons maintenant l'algorithme associé aux choix précédants :

fonction AKS(N)
1. **si** $N = a^b$ **alors** retourner **faux** ;
2. $r := 2$;
3. **tantque** $r < N$ **faire**
4.   **si** r est premier **alors**
5.   **si** $r \mid N$ **alors** retourner **faux** ;
6.   trouver q le plus grand facteur premier de $r-1$;
7.   **si** $q \geqslant 4\sqrt{r}\log_2 N$ **et** $N^{(r-1)/q} \not\equiv 1 \bmod r$ **alors** sortir de la boucle ;
8. $r := r + 1$;
9. **pour** $a = 1$ **à** $2\sqrt{r}\log_2 N$ **faire**
10.   **si** $(X-a)^N \not\equiv X^N - a \bmod (X^r - 1, N)$ **alors** retourner **faux**.
11. retourner **vrai**.

Remarques :
- On calcule bien sûr une fois pour toutes $X^N \bmod (X^r - 1, N)$.
- Une des particularités intéressantes de l'algorithme est qu'on peut précalculer une table de r convenables pour une taille de N donnée. Il ne reste plus qu'à vérifier la condition sur $N^{(r-1)/q}$ pour les nombres de la liste.

4.4. **Analyse de l'algorithme.** Nous utilisons ci-dessous la notation $\tilde{O}$ qui permet de ne pas prendre en compte les facteurs logarithmiques qui pourraient apparaître.

Proposition 4.2. *Le temps de calcul de* AKS *est* $O((\log N)^{19})$ *si on utilise des algorithmes classiques de multiplication et* $\tilde{O}((\log N)^{12})$ *si on utilise de la multiplication rapide, aussi bien pour les polynômes que pour les entiers.*

Démonstration : le temps de calcul est largement dominé par le temps passé dans la boucle 9. Pour chaque valeur de a, le calcul de $(X-a)^N \bmod (X^r-1, N)$ demande $O(\log N)$ multiplications $A(X)B(X) \bmod (X^r - 1)$ où $A(X)$ et $B(X)$ ont degré $O(r)$. La réduction ne coûte rien car $X^r - 1$ est creux. Notant $\mathcal{P}(N, r)$ le temps nécessaire à multiplier deux polynômes de degré r à coefficients dans $\mathbb{Z}/N\mathbb{Z}$, le temps de la boucle 9 est donc $O(\sqrt{r}(\log N)(\log N)\mathcal{P}(N, r))$.

Le temps $\mathcal{P}(N, r)$ peut s'exprimer en nombre d'opérations sur des entiers modulo N. Notant $\mathcal{M}(N)$ le temps nécessaire à multiplier deux entiers modulo N, on a $\mathcal{P}(N, r) = r^2\mathcal{M}(N)$ si l'on utilise la méthode de multiplication classique de polynômes. Utiliser la FFT conduit à $\mathcal{P}(N, r) = r \log r \mathcal{M}(N) = \tilde{O}(r\mathcal{M}(N))$. Si une arithmétique naïve dans $\mathbb{Z}/N\mathbb{Z}$ est utilisée, alors $\mathcal{M}(N) = O((\log N)^2)$, sinon, c'est encore avec la FFT $\mathcal{M}(N) = \tilde{O}(\log N)$.

En résumé, si tout est naïf, on obtient

$$O(r^{5/2}(\log N)^4).$$

Si on utilise la FFT partout, cela devient :

$$\tilde{O}(r^{3/2}(\log N)^3).$$

Injecter $r = O((\log N)^6)$ conduit aux résultats voulus. □

Remarque. Dans la pratique, il semble à peu près certain qu'on puisse trouver r de la taille de $O((\log N)^2)$ (ce qui permet de satisfaire la condition combinatoire). Cela conduirait à une complexité $\tilde{O}((\log N)^6)$ dans le meilleur des cas.

4.5. **Qu'en est-il en pratique ?** Le tableau ci-dessous donne les tailles respectives des différentes fonctions de complexité :

N	$2(\log N)^2$	$(\log N)^6$
10^{100}	$1.060380e+05$	$1.490370e+14$
10^{1000}	$1.060380e+07$	$1.490370e+20$
2^{512}	$2.518957e+05$	$1.997894e+15$
2^{1024}	$1.007583e+06$	$1.278652e+17$
2^{10000}	$9.609060e+07$	$1.109054e+23$

L'algorithme de Miller est déjà lent, car il faut calculer de nombreuses exponentielles modulaires.

La quantité $(\log N)^6$ donne une idée de l'ordre de grandeur du degré des polynômes avec lesquels il faut travailler. Sans astuce supplémentaire, il parait difficile d'arriver à s'en sortir. En pratique, il est presque certain qu'on peut trouver un $r = c(\log_2 N)^2$ avec $c \geqslant 64$. Prenons un exemple. Si $N = 2^{512}$, alors le plus optimiste $r = 64(\log_2 N)^2 = 2^{24} > 16 \cdot 10^6$, conduisant à manipuler des polynômes denses de plus de 1 Goctets, ce qui est peut-être envisageable.

Supposons que l'on veuille prouver la primalité de $N = 10^9 + 7$ (qui est premier). E. Thomé a implantanté AKS à l'aide de GMP 4.1, sur un PC à 700 MHz. Avec l'approximation de AKS, on prend $r = 57287$, ce qui conduit à $s = 14340$. Chaque calcul intermédiaire prend 44 secondes, ce qui donne un temps total de plus de 7 jours. Si on utilise directement la condition (2), on peut prendre $(r, q, s) = (3623, 1811, 1785)$, et cela prend 1.67×1785 secondes ou encore 49 minutes. Le meilleur triplet est $(r, q, s) = (359, 179, 4326)$, conduisant à un temps total de 6 minutes et 9 secondes.

Remarquons que Miller et AKS sont massivement (et trivialement) distribuables. Cela dit, cela n'en fait pas des algorithmes efficaces en pratique.

5. Comparaison avec la concurrence

Il existe sur le marché deux algorithmes de primalité couramment utilisés : les sommes de Jacobi (voir [7] pour une présentation) et ECPP [3]. Le premier est déterministe presque polynomial, le second probabiliste sans doute polynomial (on ne dispose que d'une analyse heuristique en $O((\log N)^{6+\varepsilon})$ présentée dans [11]). ECPP fournit en plus un certificat vérifiable rapidement (en $O((\log N)^4)$), contrairement à tous les autres algorithmes proposés. Ils sont tous les deux distribuables.

ECPP est à même de prouver la primalité de nombres de 512 bits en 1 seconde, de 1024 bits en 1 minute[†] et des nombres de 2^{10000} en un temps "raisonnable" (un mois). Même si on arrivait à faire baisser la valeur de r dans AKS, rien ne dit que l'algorithme serait plus rapide que ECPP pour les nombres de taille raisonnable.

Remerciements. Merci à Y. Gallot pour avoir signalé des erreurs numériques dans la première version de cette note. Merci également à P. Zimmermann pour ses commentaires ; à A. Klappenecker pour avoir détecté des imprécisions et N. Brisebarre pour avoir signalé une erreur de bibliographie.

Références

[1] L. M. Adleman and M.-D. A. Huang. *Primality testing and Abelian varieties over finite fields*, volume 1512 of *Lecture Notes in Math.* Springer–Verlag, 1992.

[2] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. Preprint; available at `http://www.cse.iitk.ac.in/primality.pdf`, August 2002.

[†] avec la version 6.4.5 disponible sur `http://www.lix.polytechnique.fr/Labo/Francois.Morain` tournant sur un Pentium III à 450 MHz.

[3] A. O. L. Atkin and F. Morain. Elliptic curves and primality proving. *Math. Comp.*, 61(203):29–68, July 1993.

[4] E. Bach. Explicit bounds for primality testing and related problems. *Math. Comp.*, 55(191):355–380, July 1990.

[5] R. C. Baker and G. Harman. The Brun-Titschmarsh theorem on average. In *Proceedings of a conference in Honor of Heini Halberstam*, volume 1, pages 39–103, 1996.

[6] D. Bernstein. An exposition of the Agrawal-Kayal-Saxena primality-proving theorem. Preprint; available from `http://cr.yp.to/papers.html#aks`, August 2002.

[7] H. Cohen. *A course in algorithmic algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer–Verlag, 1996. Third printing.

[8] R. Crandall and C. Pomerance. *Primes – A Computational Perspective*. Springer Verlag, 2000.

[9] E. Fouvry. Théorème de Brun-Titschmarsh; application au théorème de Fermat. *Invent. Math.*, 79:383–407, 1985.

[10] D. H. Lehmer. Strong Carmichael numbers. *J. Austral. Math. Soc. Ser. A*, 21:508–510, 1976.

[11] A. K. Lenstra and H. W. Lenstra, Jr. Algorithms in number theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity, chapter 12, pages 674–715. North Holland, 1990.

[12] H. W. Lenstra, Jr. Miller's primality test. *Inform. Process. Lett.*, 8(2):86–88, 1979.

[13] H. W. Lenstra, Jr. Primality testing. In *Computational methods in number theory, Part I*, pages 55–77. Math. Centrum, Amsterdam, 1982.

[14] G. L. Miller. Riemann's hypothesis and tests for primality. In *Proc. 7th STOC*, pages 234–239, 1975.

[15] V. R. Pratt. Every prime has a succint certificate. *SIAM J. Comput.*, 4:214–220, 1975.

[16] P. Ribenboim. *The new book of prime number records*. Springer-Verlag, 1996.

Laboratoire d'Informatique de l'École polytechnique (LIX), F-91128 Palaiseau Cedex, France
E-mail address, F. Morain: `morain@lix.polytechnique.fr`