FILOMAT (Niš) 9:3 (1995), 961–965
Algebra, Logic & Discrete Mathematics
Niš, April 14–16, 1995.

GENERATION OF PRIMITIVE BINARY POLYNOMIALS

Miodrag Živković

ABSTRACT. Binary linear recurrent sequences with the primitive characteristic polynomial are used as a good approximation of the random sequence.

A known algorithm is implemented in a program for generation of primitive binary polynomials of degree < 5000 with the given number of terms.

An account is given of problems solved during the program developement. Practically hardest among them is the problem of obtaining the factorizations of the numbers $2^n - 1$.

Let F_q denote the finite field of order $q = p^n$, where p is prime and $n \geq 1$. The multiplicative group F_q^* of nonzero elements of F_q is cyclic and a generator of F_q^* is called a primitive element. A monic irreducible polynomial whose roots are primitive elements is called a primitive polynomial.

It is known [2] that the binary (over $GF(2)$) sequence $\{a_n\}_{n\geq 0}$ satisfying the linear recurrent relation $a_{k+n} = \sum_{i=0}^{n-1} a_{k+i} f_i$ possesses good statistical properties if its characteristic polynomial $f = \sum_{i=0}^{n} f_i x^i$ is primitive. For example, the period length of such sequence (so called m–sequence) is $N = 2^n - 1$, the difference between the number of ones and zeros across the period is exactly one, and each n–tuple from $\{0,1\}^n$ except $\mathbf{0}$ appears exactly once in one period. The two sequences $\{a_{n+r}\}$ and $\{a_{n+s}\}$, $0 \leq r < s \leq N$, are mutually orthogonal, i.e., the equality

$$\sum_{n=0}^{N} (-1)^{a_{n+r}} (-1)^{a_{n+s}} = \begin{cases} N+1, & \text{for } 0 \leq r = s \leq N \\ -1, & \text{for } 0 \leq r < s \leq N. \end{cases}$$

holds. m–sequences are used for obtaining uniformly distributed random numbers [9]. Another field where m–sequences are widely used is cryptology [6]. Quality of m–sequences grows with n, and therefore there is a need to obtain primitive polynomials of degree as large as possible.

Supported by Grant 0401B of RFNS through Math. Inst. SANU.

There are several published tables of primitive binary polynomials. Watson [11] gives for $n < 100$ one primitive of degree n, and Stahnke [8] lists for each $n \leq 168$ a primitive with a minimum number of nonzero coefficients (trinomial or pentanomial). Zierler and Brillhart [13,14] extended this work by listing all primitive and irreducible trinomials of degree $n \leq 1000$, with the period for some for which the factorization of $2^n - 1$ is known. Rodemich and Rumsey [7] have listed all primitive trinomials of degree M_j, $12 \leq j \leq 17$ (here M_j denotes the jth Mersenne exponent, the prime for which $2^{M_j} - 1$ is also prime). The list has been extended by Zierler [12], Kurita and Matsumoto [4] and Heringa, Blöte and Compagner [3] up to $M_{23} = 11213$, $M_{28} = 86243$ and $M_{31} = 216091$ correspondingly. One primitive pentanomial of each degree M_j, $8 \leq j \leq 27$ is also listed in [4]. For those $n < 5000$, for which the factorization of $2^n - 1$ is known, in [Ž13,Ž14] the first primitive trinomial (if such exists) and a randomly generated primitive 5– and 7–nomial of degree n in $GF(2)$ are given.

In this paper we give some characteristics of the algorithm for generation of primitive binary polynomials which is used to assemble the tables in [Ž13,Ž14].

Generation of primitive polynomials is performed by testing primitivity of the sequence of trial polynomials from the given set. Here we deal with the set of polynomials f of degree n with t terms, for given n and odd t, with the constraint $f(0) = 1$. The number t is usually small, to enable simple calculation of the corresponding linear recurrent sequence. The sequence of trial polynomials is formed using the linear recurrent sequence of order 127 as a source of random numbers.

The primitivity test of a given polynomial f is effectively performed using the following set of conditions [5, Th 3.18]

$$f(0) = f(1) = 1, \tag{1}$$

$$\min\{k \mid f \mid x^{2^k} - x\} = n, \tag{2}$$

$$\text{for all prime } p \mid 2^n - 1 \qquad f \nmid x^{(2^n-1)/p} - 1. \tag{3}$$

The condition (1) eliminates polynomials divisible by x and $x + 1$. As t is odd for trial polynomials, this condition is automatically fulfilled.

The polynomial $x^{2^k} - x$ is equal to the product of all irreducible polynomials of degree dividing k [5, Th 3.20], and therefore the irreducible polynomials satisfy the conditions (1) and (2). The inverse is not true: a polynomial

equal to the product of different irreducible polynomials of degrees dividing n satisfy (1) and (2) (and it is not irreducible). The number of irreducible polynomials of degree n equals to [5, Th 3.25]

$$\frac{1}{n}\sum_{d|n}\mu(d)2^{n/d},$$

where μ is the Moebius function, defined by

$$\mu(n)=\begin{cases}1 & \text{if } n=1\\ (-1)^r & \text{if } n \text{ is the product } r \text{ distinct primes}\\ 0 & \text{if } n \text{ is divisible by the square of a prime.}\end{cases}$$

To check if f satisfy (2), it is necessary to calculate n residues $x^{2^k}-x \pmod f$. Squaring in $GF(2)$ is simple, because $(a+b)^2=a^2+b^2$. After each squaring, a residue is calculated from the division of a polynomial of degree at most $2n-2$ by f. The fact that f is sparse is used to perform division more efficiently. The total number of elementary (in$GF(2)$) operations needed to test the condition (2) is bounded by $O(tn^2)$, which is not small, having in mind the number of polynomials that need to be checked. The problem is solved in the usual way (see for example [3]): the condition (2) is modified by previously checking the conditions

(2a) $$(f, x^{2^k}-x)=1, \quad 2\le k\le 12,$$

(2b) $$(f, f')=1,$$

where (f,g) is the greatest common divisor of f and g, which is computed by the Euclidean algorithm. This makes the complete test (2) more complicated, but for the large part of trial polynomials (85% according to the estimate from [3]) it is ended by (2a). The condition (2b) eliminates the polynomials divisible by the square of a polynomial.

The numerical complexity of finding the factorization of 2^n-1 is very large. This makes it unreasonable to include the factorization as a part of the primitivity check. Even more, it is unreasonable to compute these factorizations at all, because all those of them that are known can be found in [1,10] (which is an output of the famous Cunningham project). Therefore the factorizations from [1,10] are input in a data base using a special program. The process is not straightforward, because in cited references there

are actually four tables, containing (not always complete) factorizations of the numbers
(A) $2^{2k-1}-1,\ k \leq 600$,
(B) $2^{2k-1}+1,\ k \leq 600$,
(C) $2^{4k-2}+1 = LM, L = 2^{2k-1}-2^k+1,\ M = 2^{2k-1}+2^k+1,\ k \leq 600$,
(D) $2^{4k+1}+1,\ k \leq 300$.
The first step is to list all $n < 4800$ for which all the prime factors of 2^n-1 can be found. For some values of n, the number 2^n-1 has simple algebraic factors. These algebraic factors are then further split in the algebraic factors or their prime factors are taken from one of the tables A, B, C or D. The program for updating the factorization data base tests automatically the factors during the input. It uses algebraic factors and the factors that are already in the data base. This is useful for example if the factorization of $2^{2n}-1$ is to be input when the factorization of 2^n-1 is already input.

The efficiency of the primitivity check depends also on the order in which the prime factors of 2^n-1 are used in (3): the check is carried out for the small factors p first, because according to [5, Th 3.5] the probability that (3) is not satisfied is greater for small than for large prime factors.

The number of primitive polynomials of degree n is $\phi(N)/n$ [5, Th 3.5] (here $\phi(n)$ is Euler's function, showing the number of integers i with $1 \leq i \leq n$ that are relatively prime to n). Therefore a randomly chosen binary polynomial is primitive with the probability α/n where

$$\alpha = (1-2^{-n}) \prod_{p|N} (1-\frac{1}{p}).$$

The complexity of this primitivity check is $O(ktn^2)$, where k is the number if different prime factors of 2^n-1. Taking into account the "density" of primitive polynomials, an upper bound for the complexity of generation of one primitive polynomial is roughly estimated by $O(ktn^3)$. The program, when running on a PC with the 80486 microprocessor on 66MHz, gives one primitive polynomial of degree 500 (1000) after about 2 min (20 min).

References

[1] J. Brillhart, D. H. Lehmer, J. L. Selfridge, B. Tuckerman, S. S. Wagstaf, Jr, *Factorization of $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ up to high powers*, 2nd ed., Contemp. Math., vol. 22, Amer. Math. Soc., Providence, RI, 1988.

[2] S. W. Golomb, *Shift Register Sequences*, Holden–Day, San Francisco, 1967.

[3] J. R. Heringa, H. W. Blöte and A. Compagner, *New Primitive trinomials of Mersenne–exponent degrees for random number generation*, International Journal of Modern Physics C **3** (1992), 561–564.

[4] Y. Kurita, M. Matsumoto, *Primitive t-nomials ($t = 3, 5$) over $GF(2)$ whose degree is a Mersenne exponent ≤ 44497*, Math. Comp. **56** (1991), 817–821.
[5] R. Lidl, H. Niederreiter, *Finite Fields*, Encyclopedia Math. Appl., Vol.20, Addison–Wesley, Reading, Mass., 1983.
[6] W. Meier and O. Staffelbach, *Fast correlation attacks on certain stream ciphers*, J. Cryptology **1** (1989), 159–176.
[7] E. R. Rodemich, H. Rumsey, Jr., *Primitive trinomials of high degree*, Math. Comp. **22** (1968), 863–865.
[8] W. Stahnke, *Primitive binary polynomials*, Math. Comp. **27** (1973), 977–980.
[9] R. C. Tausworthe, *Random numbers generated by linear recurrence modulo two*, Math. Comp. **19** (1965), 201–209.
[10] S. S. Wagstaf, Jr., *Update 2.6 to the Second Edition of Factorization of $b^n \pm 1$,*, 1993.
[11] E. J. Watson, *Primitive polynomials (mod 2)*, Math. Comp. **16** (1962), 368–369.
[12] N. Zierler, *Primitive trinomials whose degree is a Mersenne exponent*, Inform. Control **15** (1969), 67–69.
[13] N. Zierler, J. Brillhart, *On Primitive trinomials* (mod 2), Inform. Control **13** (1968), 541–554.
[14] N. Zierler, J. Brillhart, *On primitive trinomials* (mod 2), *II*, Inform. Control **14** (1969), 566–569.
[15] M. Živković, *A Table of Primitive Binary Polynomials*, Math. Comp. **92** (1993), 1368–1369.
[16] M. Živković, *Table of Primitive Binary Polynomials*, II, Math. Comp. **93** (1993), 368–372.

Matematički fakultet, Studentski trg 11, 11000 Belgrade, Yugoslavia