

有限域 $GF(p)$ 上 n 次模多项式的自动选取

赵 俊 朱红东

(宁波师范学院, 宁波315211) (浙江林学院)

摘 要 设 p 为一素数, $n(\geq 2)$ 为一正整数。从有限域 $GF(p)$ 构造新的有限域 $GF(p^n)$ 时, 关键的一步在于选取一个 $GF(p)$ 上的 n 次不可约多项式作为模多项式 $\varphi(\theta)$ 。利用多项式的移位相除法给出了可用以自动选取 $\varphi(\theta)$ 的一种顺序检测算法。

关键词 有限域; 模; 多项式; 因式分解法; 移位相除法; 顺序检测算法
中图分类号 TP301.6

众所周知, 对任意一个素数 p , 模 p 剩余类集 $GF(p) = \{0, 1, 2, \dots, p\} (p \triangleq p-1)$ 关于模 p 加法和乘法构成一个 p 元有限域。从 $GF(p)$ 出发, 对任意一个不小于 2 的正整数 n , 又都可以构造出一个 $R \triangleq p^n$ 元的有限域 $GF(R)$ 。它的构造步骤如下^[1]:

- (1) 选取一个 $GF(p)$ 上的 n 次不可约多项式 $\varphi(x)$ 。
- (2) 选取一个非 $GF(p)$ 中的元素 θ , 令 θ 为 $\varphi(x)$ 的一个根。
- (3) 记 $GF(R)$ 的元素形如:

$$a_0\theta^N + a_1\theta^{N-1} + \dots + a_{N-1}\theta + a_N \quad (1)$$

式中: $N \triangleq n-1$, $a_i \in GF(p)$, $i=0, 1, 2, \dots, N$ 。这就是说, $GF(R)$ 中的元素是 $GF(p)$ 上关于 θ 的不超过 N 次的多项式。

- (4) 定义 $GF(R)$ 上的加法和乘法分别为: $GF(p)$ 上多项式的模 $\varphi(\theta)$ 加法和乘法。

(5) 把 $GF(R)$ 中的元素(多项式)按系数的自然顺序及关于 θ 的升幂来排列后, 依次标记以代号 $0, 1, 2, \dots, R-1$, 得 $GF(R) = \{0, 1, 2, \dots, R-1\}$ 。

有限域理论在诸如正交试验设计、通讯信号编码、多值逻辑设计等领域中都十分有用^[2~4]。为使模 $\varphi(\theta)$ 的运算能付诸计算机实施, 上述 $GF(R)$ 构造过程中的第(1)步是关键所在。在本文中, 我们利用多项式的移位相除法给出了可用以自动选取模多项式 $\varphi(\theta)$ 的一种顺序检测算法。其基本思想是: 在 $GF(p)$ 上所有可能的 n 次多项式中, 按字典序顺序地逐一判定其不可约性, 直到获得一个不可约多项式为止, 将其取为模多项式 $\varphi(\theta)$ 。

1 多项式的移位相除法

在实际选取模多项式过程中, 求多项式相除的余式是一项极大量的工作。对此, 我们设计了一个较快速而简单的移位相除算法。其特点是: 由多项式系数的移位来实现多项式的

收稿日期: 1995-12-02

相除。

设被除式为

$$p(\theta) = a_0\theta^s + a_1\theta^{s-1} + \dots + a_{s-1}\theta + a_s \quad (0 \leq s \leq 2N),$$

除式为

$$q(\theta) = b_0\theta^t + b_1\theta^{t-1} + \dots + b_{t-1}\theta + b_t,$$

且分别用数组 $[a_0, a_1, a_2, \dots, a_s]$ 和 $[b_0, b_1, b_2, \dots, b_t]$ 表示之, 则移位相除的算法可描述如下:

- (1) 如果 $s < t$, 则结束; 否则执行(2)。
- (2) $a_0/b_0 \Rightarrow \text{temp}$ 。
- (3) 让 i 从 1 到 s 做(4)循环(b_t 不足时用 0 补足), 否则执行(5)。
- (4) $a_i - b_i * \text{temp} \Rightarrow a_{i-1}$ 。
- (5) a_s 置零, $s-1 \Rightarrow s$, 返回(1)。

上述(4)循环执行一遍, 表示被除式移位一次, 即实行了一次相除。(1)中出现的 $s < t$, 表示不能再除, 此时数组 $[a_0, a_1, a_2, \dots, a_s]$ 所代表的多项式即为余式。

移位相除算法中(2)和(4)两步使用了 $GF(p)$ 上的除法和减法。它们可以通过搜索 $GF(p)$ 的整数模 p 乘法表和加法表来进行: 设整数模 p 的乘法表和加法表分别用两个二维数组

$$\text{int mul} = p[P+1][P+1], \text{add} = p[P+1][P+1] \quad (2)$$

表示之, 则为计算 a/b 和 $a-b$, 只需搜索相应运算表的第 b 行($b=0, 1, 2, \dots, P$)即可。当 $\text{mul}-p[b][j]=a$ 时, $a/b=j$; 当 $\text{add}-p[b][j]=a$ 时, $a-b=j$ 。

2 $GF(p)$ 上多项式的不可约性判定

设 $g(\theta)$ 是 $GF(p)$ 上的任一 $n (\geq 2)$ 次多项式, 为了判定 $g(\theta)$ 在 $GF(p)$ 上的不可约性, 可设其首系数为 1, 即

$$g(\theta) = \theta^n + b_1\theta^{n-1} + \dots + b_{n-1}\theta + b_n \quad (3)$$

式中, $b_i \in GF(p)$, $i=1, 2, \dots, n$ 。由于次数不小于 2 的不可约多项式的零次项必不为零, 故 $GF(p)$ 上共有 $P(P+1)^{n-1}$ 个可供选择的候选模多项式。我们把它们的系数按字典序排成一个矩阵

$$B_n = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 2 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & P & \dots & P & P \end{bmatrix}$$

其中各行首列元素均为 1。这样, 在选取模多项式时只需逐行扫描矩阵 B_n , 并判定对应多项式的不可约性即可。

根据多项式的因式分解定理可知^[6], 对 $GF(p)$ 上任意一个形如(3)的多项式, 必存在一个正整数 k , 使得 $g(\theta)$ 可唯一地表示为

$$g(\theta) = p_1^{i_1}(\theta)p_2^{i_2}(\theta)\dots p_h^{i_h}(\theta) \quad (4)$$

式中, $p_i(\theta)$ 是 $GF(p)$ 上首系数为 1 的次数为 n_i 的不可约多项式, i_i 和 n_i 均为正整数, $j=$

$1, 2, \dots, k$, 而且有

$$i_1 n_1 + i_2 n_2 + \dots + i_k n_k = n \quad (5)$$

由此可得到下面的关于多项式不可约性的判定定理。

定理 设 $g(\theta)$ 为 $GF(p)$ 上首系数为 1 的 $n (\geq 2)$ 次多项式, 如果 $g(\theta)$ 不能被 $GF(p)$ 上任何首系数为 1 次数不超过 $[n/2]$ 的不可约多项式整除, 则 $g(\theta)$ 在 $GF(p)$ 上是不可约的。

证明 假若 $g(\theta)$ 在 $GF(p)$ 上可约, 则有 (4) 式成立, 且其中的 k 必不小于 2。由 $k \geq 2$ 及 (5) 式知, 一定存在某个正整数 $n_{j_0} \leq [n/2] (1 \leq j_0 \leq k)$ 。这样, $g(\theta)$ 将被 $GF(p)$ 上的一个 n_{j_0} 次首系数为 1 的不可约多项式 $p_{j_0}(\theta)$ 整除。此与已知条件相矛盾, 故知 $g(\theta)$ 在 $GF(p)$ 上不可约。定理证毕。

由上述判定定理可知, 为判定 $g(\theta)$ 的不可约性, 只需顺序地检测 $g(\theta)$ 能否被首系数为 1 的次数不超过 $[n/2]$ 的不可约多项式整除即可。为此, 我们把 $GF(p)$ 上首系数为 1 的次数不超过 $[n/2]$ 的所有多项式分次数将其系数按字典序排成 $[n/2]$ 个矩阵

$$B_1 \triangleq \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ \dots & \dots \\ 1 & P \end{bmatrix}_{P \times 2}, \quad B_2 \triangleq \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 2 \\ \dots & \dots & \dots \\ 1 & P & P \end{bmatrix}_{P(P+1) \times 3}, \quad \dots, \\ B_{[n/2]} \triangleq \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 2 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & P & \dots & P & P \end{bmatrix}_{P(P+1)^{[n/2]-1} \times ([n/2]+1)}$$

这里, 相应多项式的总个数为

$$\sum_{k=1}^{[n/2]} P(P+1)^{k-1} = (P+1)^{[n/2]} - 1 \quad (6)$$

我们只需在这些多项式中挑选出所有不可约的作为判定不可约性时用以做除式的多项式即可。这项挑选工作可以按下述方式递推完成。

设 k 为一正整数, 记 $F^{(k)} \triangleq \{\text{矩阵 } B_k \text{ 中各行相应的待判定不可约性的多项式全体}\}$, $FB^{(k)} \triangleq \{F^{(k)} \text{ 中的不可约多项式全体}\}$, $D^{(k)} \triangleq \{\text{在判定 } F^{(k)} \text{ 中多项式的不可约性时用作除式的不可约多项式全体}\}$ 。这里, $F^{(k)}$ 和 $FB^{(k)}$ 中的多项式的次数均为 k , 而 $D^{(k)}$ 中的多项式的次数不会超过 $[k/2]$ 。我们可以下列步骤来获得各 $D^{(k)}$ 集:

(1) 因任何 1 次多项式都是不可约的, 故有

$$D^{(1)} = \emptyset, \quad FB^{(1)} = F^{(1)} \quad (\emptyset \text{ 为空集}) \quad (7)$$

(2) 对于 $k=2, 3, \dots, [n/2]$, 有递推式

$$D^{(k)} = \begin{cases} D^{(k-1)} \cup FB^{(k/2)} & k \text{ 为偶数} \\ D^{(k-1)} & k \text{ 为奇数} \end{cases} \quad (8)$$

(3) 据判定定理知, $F^{(k)}$ 中的一个多项式是不可约的当且仅当它不能被 $D^{(k)}$ 中的任何多项式整除。由此可得到 $FB^{(k)}$, $k=2, 3, \dots, [n/2]$ 。

(4) 令

$$D^{(n)} \triangleq FB^{(1)} \cup FB^{(2)} \cup \dots \cup FB^{(\lceil n/2 \rceil)} \quad (9)$$

则 $D^{(n)}$ 即为在判定 n 次多项式 $g(\theta)$ 的不可约性时用以做除式的所有次数不超过 $\lceil n/2 \rceil$ 的不可约多项式全体的集合, 我们称之为除式集。

3 自动选取模多项式的顺序检测算法

在获得了矩阵 B_n 和除式集 $D^{(n)}$ 后, 我们就可以通过顺序地检测 B_n 中各行所对应的多项式的不可约性来自动选取模多项式了。实现这种顺序检测方法的算法, 我们称之为顺序检测算法。现将它分为 3 个阶段描述如下。

3.1 矩阵 $B_n, B_1, B_2, \dots, B_{\lceil n/2 \rceil}$ 的生成

为了获得矩阵 B_n , 我们先生成一个 $(P+1)^n \times (n+1)$ 阶的矩阵

$$A_n \triangleq \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & P & \dots & P & P \end{bmatrix}$$

其中比 B_n 中仅多了 $(P+1)^{n-1}$ 个最后一列元素为零的行。只要注意到 C 语言中对除法 “/” 和乘法 “*” 的如下两条规定^[6]: 1) “/” 在其两个操作数均为整数时将进行整除运算 (即舍去商的尾数而取整数部分); 2) “*” 和 “/” 两者同时在一个表达式中出现时将按先左后右进行运算, 矩阵 A_n 的元素 a_{ij} ($0 \leq i \leq P^n - 1, 1 \leq j \leq n$) 便可利用下列函数关系式来得到

$$\begin{cases} a_{i0} = 1 \\ a_{ij} = (i - i/p^{n-i+1} \times p^{n-i+1})/p^{n-i} \end{cases} \quad (10)$$

然后将矩阵 A_n 中所有的最后一列元素为零的行全部删去即得到矩阵 B_n 。

用类似的方法可以获得矩阵 $B_1, B_2, \dots, B_{\lceil n/2 \rceil}$ 。

3.2 除式集 $D^{(n)}$ 的产生

由于从 $FB^{(2)}$ 到 $FB^{(\lceil n/2 \rceil)}$ 的各个集合中元素的个数都是事先未知的, 所以我们需 要建立 一个链表组 $FBL[1], FBL[2], \dots, FBL(\lceil n/2 \rceil)$ 分别来存放 $FB^{(1)}, FB^{(2)}, \dots, FB^{(\lceil n/2 \rceil)}$, 还需要建立两个链表 DNL 和 DKL 用于分别存放 $D^{(n)}$ 和 $D^{(k)}$ 。于是根据递推式(8)和(9)式即可按下列步骤获得除式集 DNL。

(1) 将 DNL 和 DKL 均初始化为空表。

(2) 将矩阵 B_1 的各行填入表 $FBL[1]$ 中。

(3) 让 k 从 2 开始到 $\lceil n/2 \rceil$ 做(4)~(6)循环, 否则转到(7)。

(4) 将 $FBL[k-1]$ 连接到 DNL 的链尾。

(5) 若 k 为偶数, 则将 $FBL[k/2]$ 连接到 DKL 的链尾。

(6) 以 DKL 为除式集, 从矩阵 B_k 中顺序地挑选出相应多项式是不可约的行填入表 $FBL[k]$ 中 (其算法见下述 3.3 小节)。

(7) 将最后建立的 $FBL[k]$ (即 $FBL[\lceil n/2 \rceil]$) 连接到 DNL 的链尾。

3.3 n 次模多项式的选取

以 DNL 为除式集, 在矩阵 B_n 中选择第 1 个其相应多项式是不可约的 (即可作为模多项

式的)行的工作,可由如下描述的算法完成。

让 i 从 0 到 $P(P+1)^{n-1}-1$ 重复下列步骤:

(1) 将 B_n 中的第 i 行元素取到一维数组 g 中。

(2) 链表跟踪指针 $trace$ 从 DNL 的链头开始到链尾做(2 a)~(2 c)循环,否则转到(3)。

(2 a) 将 $trace$ 所指向的多项式系数取到一维数组 d 中。

(2 b) 按多项式的移位相除法,用 d 去除 g ,得余式的系数存放于一维数组 r 中。

(2 c) 若 $r \neq 0$ (这里 0 表示其元素全为零的一维数组),则 $trace$ 下移一个位置,返回(2 a);否则 i 增加 1,返回(1)。

(3) 将 g 中的元素复制到一维数组 p 中,算法终止。此时, p 中存放着所求的 $GF(p)$ 上的一个 n 次模多项式的系数。

参 考 文 献

- 1 Muzio J C, Wesselkamper T C. *Multiple-valued Switching Theory*. Bristol and Boston: Adam Hilger Ltd. 1986
- 2 马希文. 正交设计的数学理论. 北京: 人民教育出版社, 1981
- 3 杨义先, 林须端. 编码密码学. 北京: 人民邮电出版社, 1992
- 4 赵小杰, 吴训威. 模代数在多值逻辑系统中的适应范围. 科学通报, 1991, 16(18), 1431~1433
- 5 北京大学数学系几何与代数教研室代数小组. 高等代数. 第 2 版. 北京: 高等教育出版社, 1985
- 6 李书涛. C 语言程序设计教程. 北京: 北京理工大学出版社, 1993

Zhao Jun (Ningbo Normal College, Ningbo 315211, Zhejiang, PRC) and Zhu Hongdong. Automatic Choice of n Degrees Module Polynomials over Finite Field $GF(p)$. *J Zhejiang For Coll*, 1996, 13(2): 215~219

Abstract: Let p be a prime number and $n(\geq 2)$ be a positive integer. A crucial step lies in taking a n degrees irreducible polynomial over $GF(p)$ as module polynomial $\varphi(\theta)$ when a new finite field $GF(p^n)$ is constructed from $GF(p)$. A sequential measurement algorithm which can automatically choose $\varphi(\theta)$ is given, making use of the shifting division for polynomials.

Key words: finite fields; modules(mathematics); polynomials; factorizational methods; shifting division; sequential measurement algorithm