Notes of binius

Jade Xie

2024 年 5 月 15 日

目录

1 TO-DO

- 多项式 $X^2 + X_{\iota-2} \cdot X + 1$ 在 $\mathcal{T}_{\iota-1}[X_{\iota-1}]$ 上是不可约的（见 [?]Thm. 1）.
- 两个大数相乘并作算法复杂度的分析.

2 介绍

3 Background

3.1 Polynomials

3.2 Binary Towers

3.2.1 子域与扩张

在具体深入 binius 论文 [?] 2.3 Binary Towers 细节之前，先给出数学上关于域扩张的知识.

首先看看什么是扩域. 下面参考《抽象代数》[?] 中的描述. 设 E 是域，F 是其**子域**（即 $F \subset E$ 且 F 按照 E 中的运算成为域，二者乘法单位元同一），则称 E 是 F 的**扩张**或**扩域**（**extension field**），记为 E/F. 理解一下，意思是域 F 是域 E 的子域就构成一个扩域 E/F，F 是 E 的子域表示的意思是保持 F 中的两个二元运算，它们在两个域中是同样的两个二元运算，并且两个域的乘法单位元是同一个. 扩域也可以用线性空间的角度来看，由 E 是 F 的扩域，特别可知 E 是 F 上的**线性空间**.

定义 1. 如果一个代数系统 $(V; +, \cdot; \mathbb{P})$ 满足下列性质，那么就称为数域 $\mathbb{P}$ 上的一个**线性空间**.

(1) 向量加法的交换律：$\forall \alpha, \beta \in V, \alpha + \beta = \beta + \alpha$.

(2) 向量加法的结合律：$\forall \alpha, \beta, \gamma \in V, (\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$.

(3) 向量加法有零元：$\exists \theta \in V, \forall \alpha \in V, \alpha + \theta = \alpha$.

(4) 向量加法有负元：$\forall \alpha \in V, \exists \alpha\prime \in V, \alpha + \alpha\prime = \theta$.

(5) 标量乘法对向量加法有分配律：$\forall \alpha, \beta \in V, \forall k \in \mathbb{P}, k \cdot (\alpha + \beta) = k \cdot \alpha + k \cdot \beta$.

(6) 标量乘法对域加法有分配律：$\forall \alpha \in V, \forall k, l \in \mathbb{P}, (k + l) \cdot \alpha = k \cdot \alpha + k \cdot \alpha$.

(7) 标量乘法与标量的域乘法相容：$\forall\alpha \in V, \forall k, l \in \mathbb{P}, (kl)\cdot\alpha = k\cdot(l\cdot\alpha)$.

(8) 标量乘法有单位元：$\forall\alpha \in V, 1\cdot\alpha = \alpha$.

满足以上八条性质便可称其为线性空间，在不引起混淆的情况下也可记为 $V.k\cdot\alpha$ 也可沿用几何空间中向量数乘的习惯为 $k\alpha$.

根据线性空间的定义，这里 E/F，E 是 F 的线性空间，也就是说：

(1) E 是一个加法阿贝尔群.

(2) F 中的元素与 E 中的元素之间有（数乘）运算且满足：对任意 $c, c\prime \in F$，α，$\alpha\prime \in E$ 有

- (i) $c\alpha \in E$
- (ii) $c(\alpha + \alpha\prime) = c\alpha + c\alpha\prime$
- (iii) $(c + c\prime)\alpha = c\alpha + c\prime\alpha$
- (iv) $(cc\prime)\alpha = c(c\prime\alpha)$
- (v) $1\cdot\alpha = \alpha$

扩域 E 作为 F 上的线性空间，其维数称为扩张次数，记为 $[E:F]$；此线性空间的基称为扩张 E/F 的基，或 E 的 F-基. 当 $[E:F]$ 有限时，称 E/F 为有限扩张. 而 $[E:F]=1$ 意味着 $E=F$. 这里扩域 E 作为 F 上的线性空间，还可以这样来表述：设 $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ 是 E/F 的基，则对于 E 中的任意一个元素 $v \in E$，都可以表示为

$$v = \sum_j \omega_j\alpha_j = \omega_0\alpha_0 + \omega_1\alpha_1 + \cdots + \omega_n\alpha_n, \omega_j \in F, j = 0, 1, \ldots, n.$$

意思就是 E 中的所有元素都可以用 F 中的元素通过 E 的 F-基进行线性表出. 不将基显式地写出来，也可以将 E 看作是 F 上的向量空间 (vector space，其实向量空间就是线性空间 [?])，例如上述的 v 就能用一个向量来表示

$$v = (\omega_0, \omega_1, \omega_2, \ldots, \omega_n)$$

定义 2. *(1)* 设 E/F 为域的扩张，$\alpha \in E$，称 α 是 F 上的**代数元素**（*algebraic element*）是指：α 是 F 上的某多项式 $f(x) \in F[x]$ 的根，即 $f(\alpha) = 0$，也就是说存在正整数 n 和不全为 0 的 c_0，c_1，$\cdots$，$c_n \in F$ 使得 $c_n a^n + \cdots + c_1 a + c_0 = 0$（称 $f(x)$ 是 α 的化零多项式）.

(2) 如果 E 中所有元素都是 F 上的代数元素，则称 E/F 是**代数扩张**（*algebraic extension*）. 非代数元素为**超越**元素，非代数扩张为**超越扩张**（*transcendent extension*）.

(3) 若复数 α 是 $\mathbb{Q}$ 上的代数元素，则称 α 为代数数，否则称为超越数.

在域扩张中，有很重要的一个定理，是域扩张的有力工具，定理如下.

定理 1 (单代数扩张)**.** 设 F 是任一域，$p(x) \in F[x]$ 是任一个 $n(>1)$ 次不可约多项式，则存在 F 的 n 次单扩张 $E = F(\alpha)$，且 α 是 $p(x)$ 的根. 事实上，商环 $E = F[x]/(p(x))$ 为域. 视同构 $F \simeq \bar{F}$ 为相等（对 $b \in F$ 视为 $b = \bar{b}$），则 E 是 F 的 n 次扩域，$\alpha = \bar{x}$ 是 $p(X)$ 的根，且

$$E = F(\alpha) = \{b_0 + b_1\alpha + \cdots + b_{n-1}\alpha^{n-1} | b_0, b_1, \ldots, b_{n-1} \in F\},$$

*(*这里 $\overline{g(x)}$ 表示 $g(x)$ 的模 $(p(x))$ 同余类，$\bar{F} = \{\bar{b} : b \in F\}$*).*

例 1 (复数域的引入). 设 $F=\mathbb{R}$ 是实数域, $p(X)=X^2+1$ 不可约, $n=2$. 则商环

$$E=\mathbb{R}[x]/(x^2+1)=\{\bar{a}+\bar{b}\bar{x}|a,b\in\mathbb{R}\}=\{a+bi|a,b\in\mathbb{R}\}$$

(其中 $i=\bar{x}$, 且对实数 b 记 $\bar{b}=b$). 于是 $0=x^2+1=\bar{x}^2+1=i^2+1$, $i^2=-1$, 故常记 $i=\sqrt{-1}$, $E=\{a+bi\}$ 就是复数域 $\mathbb{C}$. 这是引入复数域的最严格途径.

论文 [?] 中的域扩张构造就是按照这个定理思路进行扩张的. 先选定第一个域 $\mathcal{T}_0:=\mathbb{F}_2$（第一个域也可以选择大一些的域 $\mathbb{F}_{2^{2^k}}$，例如 $\mathbb{F}_{16}$，最后计算结果都可以递归到第一个域 $\mathbb{F}_{2^{2^k}}$ 中的计算，可以用查表的方法直接得到结果 [?]），接着基于 $\mathbb{F}_2$ 进行单代数扩张，考虑在 $\mathbb{F}_2$ 上的不可约多项式 $p(x)=x^2+x+1$，即 $p(x)$ 在 $\mathbb{F}_2$ 上找不到一个数 x_0 使得 $p(x_0)=x_0^2+x_0+1=0$，但是在需要扩的域上 $\mathbb{F}_{2^2}$ 上能够找到一个根，记这个根为 $X_0\in\mathbb{F}_{2^2}$，即 $X_0^2+X_0+1=0$ 在 $\mathbb{F}_{2^2}$ 上成立. 设 $\mathcal{T}_1:=\mathbb{F}_2[X_0]/(X_0^2+X_0+1)$ 为 $\mathbb{F}_2$ 的扩域. 考虑到能够进行递归的域扩张，对于 $\forall\iota>1$，

$$\mathcal{T}_\iota:=\mathcal{T}_{\iota-1}[X_{\iota-1}]/(X_{\iota-1}^2+X_{\iota-2}\cdot X_{\iota-1}+1),\qquad X_{\iota-2}\in\mathcal{T}_{\iota-1},X_{\iota-1}\in\mathcal{T}_\iota. \tag{1}$$

对于多项式 $X_{\iota-1}^2+X_{\iota-2}\cdot X_{\iota-1}+1$，将 $X_{\iota-1}$ 看作自变量 X，多项式 $X^2+X_{\iota-2}\cdot X+1$ 在 $\mathcal{T}_{\iota-1}[X_{\iota-1}]$ 上是不可约的（见 [?]Thm. 1）. 下面写一些具体的例子来理解这个域扩张的过程。从 $\mathbb{F}_2$ 开始:

$$\mathcal{T}_0:=\mathbb{F}_2=\{0,1\}$$

对 $\mathbb{F}_2$ 进行扩张，引入新的元素 X_0，$\mathcal{T}_1:=\mathbb{F}_2[X_0]/(X_0^2+X_0+1)$，$\mathcal{T}_1$ 的 $\mathcal{T}_0$-基为 $1,X_0$，根据扩域的线性空间角度理解，$\mathcal{T}_1$ 中的元素都能写成 $a+bX_0(a,b\in\mathbb{T}_{\iota-1})$ 的形式，则

$$\mathcal{T}_1:=\mathbb{F}_2[X_0]/(X_0^2+X_0+1)=\{0,1,X_0,1+X_0\}.$$

实际 $\mathcal{T}_1\cong\mathbb{F}_{2^2}$. 接着扩域在 $\mathcal{T}_1$ 上扩域

$$\mathcal{T}_2:=\mathcal{T}_1[X_1]/(X_1^2+X_0X_1+1)$$

要计算 $\mathcal{T}_2$ 中的元素，可以用一个表格来计算，见表**??**，因为 $\mathcal{T}_2$ 中的元素都可以用 $\mathcal{T}_2$ 的 $\mathcal{T}_2$-基为 $\{1,X_1\}$ 线性表出，即写成 $a+bX_1,a,b\in\mathcal{T}_1$. 继续扩域的过程是类似的. 那么最后形成一个扩张塔 $\mathcal{T}_0\subset\mathcal{T}_1\subset\cdots\mathcal{T}_\iota$. 那么

表 1: $\mathcal{T}_2$ 中的元素

a/bX_1	0	$1\cdot X_1$	$X_0\cdot X_1$	$(1+X_0)\cdot X_1$
0	0	X_1	X_0X_1	$X_1+X_0X_1$
1	1	$1+X_1$	$1+X_0X_1$	$1+X_1+X_0X_1$
X_0	X_0	X_0+X_1	$X_0+X_0X_1$	$X_0+X_1+X_0X_1$
$1+X_0$	$1+X_0$	$1+X_0+X_1$	$1+X_0+X_0X_1$	$1+X_0+X_1+X_0X_1$

最后得到一个相同的环，$\forall\iota\geq 0$，

$$\mathcal{T}_\iota=\mathbb{F}_2[X_0,\cdots,X_{\iota-1}]/(X_0^2+X_0+1,\cdots,X_{\iota-1}^2+X_{\iota-2}\cdot X_{\iota-1}+1).$$

这里的表示意思是可以从 $\mathbb{F}_2$ 直接扩到 $\mathcal{T}_\iota$，且 $\mathbb{F}_2(X_0,\ldots,X_{\iota-1})\cong\mathbb{F}_2[X_0,\cdots,X_{\iota-1}]/(X_0^2+X_0+1,\ldots,X_{\iota-1}^2+X_{\iota-2}\cdot X_{\iota-1}+1)$.

扩域中一次性添加多个元素: 在对域 F 扩张时，可以向域 F 中陆续添加多个元素 α，β，$\cdots$，γ，得到扩域 $F(\alpha,\beta,\cdots,\gamma)$. 即先向 F 添加 α，再向 $F(\alpha)$ 添加 β，等等. 易知 $F(\alpha,\beta,\cdots,\gamma)$ 即是 α，β，$\cdots$，γ 和 F 的元素多次加减乘除得到的结果集合，是含 F 和 α，β，$\cdots$，γ 的最小域. 例如

$$\mathbb{Q}(\sqrt{2},\sqrt{3})=(\mathbb{Q}(\sqrt{2}))(\sqrt{3})=\{a+b\sqrt{2}+c\sqrt{3}+d\sqrt{6}|a,b,c,d\in\mathbb{Q}\}.$$

那就可以这样来理解 $\mathbb{F}_2(X_0,\ldots,X_{\iota-1}) \cong \mathbb{F}_2[X_0,\cdots,X_{\iota-1}]/(X_0^2+X_0+1,\ldots,X_{\iota-1}^2+X_{\iota-2}\cdot X_{\iota-1}+1)$，是对 $\mathbb{F}_2$ 一次性添加了 ι 个元素 $X_0,\ldots,X_{\iota-1}$，也可以看成逐个去添加元素. 扩张之后形成的域与上述的商环是同构的.

现在考虑在 $\mathcal{T}_\iota$ 中的两个元素 a 和 b 做乘法运算，那么 a 与 b 都能用 $\mathcal{T}_\iota$ 的 $\mathcal{T}_{\iota-1}$-基线性表示，由域扩张构造方程(??) $\mathcal{T}_\iota := \mathcal{T}_{\iota-1}[X_{\iota-1}]/(X_{\iota-1}^2+X_{\iota-2}\cdot X_{\iota-1}+1)$ 知 $\mathcal{T}_\iota/\mathcal{T}_{\iota-1}$ 的基为 $\{1,X_{\iota-1}\}$，则

$$a = a_0 + a_1X_{\iota-1}, \quad a_0,a_1 \in \mathcal{T}_{\iota-1}$$
$$b = b_0 + b_1X_{\iota-1}, \quad b_0,b_1 \in \mathcal{T}_{\iota-1}$$

因此 a 与 b 相乘可以写为:

$$\begin{aligned}
&(a_0+a_1X_{\iota-1})(b_0+b_1X_{\iota-1})\\
=&a_0b_0+(a_0b_1+a_1b_0)X_{\iota-1}+a_1b_1X_{\iota-1}^2\\
&\text{（由于在}\mathcal{T}_\iota\text{中}X_{\iota-1}\text{为不可约多项式}X_{\iota-1}^2+X_{\iota-2}\cdot X_{\iota-1}+1\text{的根，因此}X_{\iota-1}^2+X_{\iota-2}\cdot X_{\iota-1}+1=0\\
&\text{，则}X_{\iota-1}^2=-X_{\iota-2}\cdot X_{\iota-1}-1.\text{在 }\mathbb{F}_{2^k}\text{ 中，特征为 2，则 }1+1=0\text{，因此 }X_{\iota-1}^2=X_{\iota-2}\cdot X_{\iota-1}+1\text{）}\\
=&a_0b_0+a_1b_1+(a_0b_1+a_1b_0+a_1b_1X_{\iota-2})X_{\iota-1}
\end{aligned}$$

这样在 $\mathcal{T}_\iota$ 中比较大的两个数就能有效的转换为在比较小的 $\mathcal{T}_{\iota-1}$ 中的数相乘，这个过程利用了 Karatsuba 算法 [?]，能够比直接在 $\mathcal{T}_\iota$ 中进行计算更有效. 利用算法复杂度分析中的主定理，可以得到这里的乘法复杂度为 $\mathcal{O}(2^{\log 3\cdot\iota})$[?].

参考文献

[1] Benjamin E. Diamond and Jim Posen, *Succinct Arguments over Towers of Binary Fields*, Cryptoglogy ePrint Archive, 2023.

[2] 张贤科, 抽象代数, 2022.

[3] John L. Fan and Christof Paar, *On efficient inversion in tower fields of characteristic two*, Proceddings of IEEE International Symposium on Information Theory, 1997.

[4] 向量空间, Wikipedia.

[5] Doug Wiedemann, An Iterated Quadratic Extension of GF(2), The Fibonacci Quarterly, 1988.

[6] Karatsuba Algorithm, Wikipedia.