Math-Net.Ru

Общероссийский математический портал

И. С. Сергеев, О построении схем для перехода между полиномиальными и нормальными базисами конечных полей, *Дискрет. матем.*, 2007, том 19, выпуск 3, 89–101

DOI: <https://doi.org/10.4213/dm968>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 34.229.45.116

26 апреля 2023 г., 16:56:14

О построении схем для перехода между полиномиальными и нормальными базисами конечных полей

© 2007 г. И. С. Сергеев

Показано, что переход между нормальным и полиномиальным базисом поля $GF(p^n)$ может быть выполнен схемой над $GF(p)$ со сложностью $O(n^{1,806})$ и глубиной $O(\log n)$.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, проект 05-01-00994, программой Президента Российской Федерации поддержки ведущих научных школ, грант НШ 5400.2006.1, и программой фундаментальных исследований Отделения математических наук РАН «Алгебраические и комбинаторные методы математической кибернетики», проект «Синтез и сложность управляющих систем».

1. Введение

При реализации операций в конечном поле $GF(p^n)$ используются два основных представления элементов поля: полиномиальное, в котором элементы поля рассматриваются как многочлены степени не выше $n - 1$, а операции производятся по модулю некоторого неприводимого над $GF(p)$ многочлена $m_n(t)$ степени n , и нормальное, когда элементы поля рассматриваются как линейные комбинации над $GF(p)$ с базисными элементами $\alpha, \alpha^p, \alpha^{p^2}, \dots, \alpha^{p^{n-1}}$, где α — порождающий элемент (или генератор) базиса.

Полиномиальное представление соответствует разложению элементов поля в базисе $A = \{1, \alpha, \alpha^2, \dots, \alpha^{n-1}\}$, называемом полиномиальным базисом, где α — корень $m_n(t)$. Например, можно выбрать α соответствующим одночлену t в явной полиномиальной записи. Чтобы элемент α одновременно являлся порождающим для нормального базиса $B = \{\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_{n-1}\}$, где $\alpha_i = \alpha^{p^i}$, в качестве $m_n(t)$ нужно взять неприводимый многочлен степени n , корни которого $\alpha, \alpha^p, \alpha^{p^2}, \dots, \alpha^{p^{n-1}}$ линейно независимы над $GF(p)$. Такая пара базисов (A, B) может быть построена в произвольном конечном поле. Необходимое обоснование с подробным изложением теории конечных полей содержится в [1, 2].

Сложение в обоих типах базисов выполняется в общем случае с одинаковой сложностью n операций над $GF(p)$. Для умножения в полиномиальном базисе известна оценка Шёнхаге и Штрассена $O(n \log n \log \log n)$ (см., например, гл. 8 в [3]), а также ряд более практичных методов с меньшим, чем n^2 , порядком сложности. Умножение в нормальных базисах выполняется алгоритмом Месси–Омура со сложностью $O(n^3)$ в общем случае

(см., например, [4]). В так называемых оптимальных нормальных базисах и вообще в гауссовых нормальных базисах эта оценка понижается до $O(n^2)$, однако такие базисы существуют не во всех полях. В [5] было предложено обобщение гауссовых нормальных базисов, однако и такие базисы заведомо не существуют в полях $GF(2^{8n})$. Кроме того, пока, по-видимому, не доказано, что обобщенные гауссовы нормальные базисы имеют низкую сложность (в тех полях, где они существуют) в общем случае.

Преимуществом нормальных базисов является быстрое выполнение операции Фробениуса (возведение в степень вида p^k), не требующее затрат при схемной реализации. Для этой операции в полиномиальных базисах известен метод Брента–Кунга [6] с оценкой сложности, зависящей от алгоритма матричного умножения и на сегодняшний день составляющей $O(n^{1.667})$ (см. [7]).

В свете сказанного имеет значение сложность перехода от одного базиса к другому (под переходом понимается соответствующее преобразование координат элемента поля). Основной мотивировкой перехода от нормального к полиномиальному базису является ускорение операции умножения, а обратно — операции возведения в степень (см. также [4]). В общем случае такой переход может быть выполнен методом Лупанова для линейных операторов со сложностью $O(n^2/\log_p n)$ (см. [8]). В [4] показано, что для оптимальных нормальных базисов справедлива оценка $O(n \log n)$. Оценка сложности перехода фактически является оценкой сложности умножения в произвольном нормальном базисе.

В [9] показано, что переход от нормального базиса к полиномиальному можно выполнить со сложностью $O(n^{1.815})$. Там же построен вероятностный алгоритм обратного перехода с такой же оценкой сложности.

В разделе 2 будет описан базовый метод Брента–Кунга [6]. С его помощью в разделах 3 и 4 будут построены схемы перехода соответственно к нормальному и к полиномиальному базису сложности $O(n^{1.834})$ и глубины $O(\log n)$. В разделе 5 будет показано, что оценку сложности этих схем можно снизить до $O(n^{1.806})$. Для схемы перехода к полиномиальному базису эта оценка фактически установлена в [9] (см. раздел 6.1). В разделе 6 будут рассмотрены некоторые приложения.

2. Метод Брента–Кунга

Рассмотрим задачу вычисления x^{p^k} , где $x \in GF(p^n)$, в полиномиальном базисе поля. Оценка сложности $O(n^2/\log_p n)$ может быть получена методом Лупанова [8] (так как операция Фробениуса является линейным преобразованием). В [6] был предложен метод, имеющий меньший порядок сложности. Опишем его применительно к данной задаче. Пусть $x = f(t)$ в полиномиальном представлении, $\deg f < n$. Ясно, что

$$\begin{aligned} f^{p^k}(t) \bmod m_n(t) &= f(t^{p^k}) \bmod m_n(t) = f(t^{p^k \bmod m_n(t)}) \bmod m_n(t) \\ &= f(\xi(t)) \bmod m_n(t), \end{aligned}$$

где

$$\xi(t) = t^{p^k} \bmod m_n(t).$$

Пусть $n \leq rq$. Запишем $f(t)$ в виде

$$f(t) = f_0(t) + f_1(t)t^r + \dots + f_{q-1}(t)t^{(q-1)r}, \quad (1)$$

где $\deg f_i < r$. Пусть

$$f_i(t) = f_{i,0} + f_{i,1}t + \dots + f_{i,r-1}t^{r-1}, \quad i = 0, 1, \dots, q-1.$$

Алгоритм 1. 1. Многочлены $\xi(t), \xi^2(t), \dots, \xi^{r-1}(t); \xi^r(t), \xi^{2r}(t), \dots, \xi^{(q-1)r}(t)$ по модулю $m_n(t)$ считаются вычисленными заранее.

2. Вычисляются многочлены

$$f_i(\xi(t)) = f_{i,0} + f_{i,1}\xi(t) + \dots + f_{i,r-1}\xi^{r-1}(t).$$

Заметим, что справедливо равенство

$$\begin{bmatrix} f_0(\xi(t)) \\ f_1(\xi(t)) \\ \vdots \\ f_{q-1}(\xi(t)) \end{bmatrix}_{q \times n} = \begin{bmatrix} f_0(t) \\ f_1(t) \\ \vdots \\ f_{q-1}(t) \end{bmatrix}_{q \times r} \begin{bmatrix} 1 \\ \xi(t) \\ \vdots \\ \xi^{r-1}(t) \end{bmatrix}_{r \times n}, \quad (2)$$

где в строчках матриц выписаны коэффициенты соответствующих многочленов, а снизу указаны размерности. Сложность шага оценивается как сложность умножения $q \times r$ матрицы на $r \times n$ матрицу.

3. Вычисляются произведения $f_i(\xi(t))\xi^{ir}(t)$. Сложность шага оценивается как $(q-1)M(n)$, где $M(n)$ — сложность умножения многочленов степени $n-1$ схемой глубины $O(\log n)$.
4. Результаты предыдущего шага (многочлены степени $2n-2$) складываются покомпонентно со сложностью, не большей чем $2nq$.
5. Окончательно, результат приводится по модулю $m_n(t)$ со сложностью $2M(n) + n$ (приведение многочлена степени $2n-1$ по модулю фиксированного многочлена сводится к двум умножениям и одному сложению многочленов степени $n-1$, см., например, гл. 9 в [3]).

2.1. Оценка сложности алгоритма 1

Выберем $r, q \sim \sqrt{n}$. Сложность шагов 3–5 оценивается как

$$O(qM(n)) = O(n^{1.5} \log n \log \log n).$$

Сложность шага 2 есть $O(n^w)$, где w — экспонента умножения матрицы размера $\sqrt{n} \times \sqrt{n}$ на матрицу размера $\sqrt{n} \times n$. Мы используем определение достижимой экспоненты, под которой понимается любое w , для которого существует алгоритм сложности $O(n^w)$. Абсолютная экспонента ω определяется как нижняя грань для достижимых экспонент, алгоритм сложности $O(n^\omega)$ при этом может и не существовать (см. гл. 12 в [3]). В [7] показано, что можно выбрать $w < 1,667$. Таким образом, алгоритм имеет сложность $O(n^{1,667})$. Глубина алгоритма составляет $O(\log n)$, поскольку логарифмическую глубину имеют методы умножения Шёнхаге и Штрассена (для многочленов), а известные методы матричного умножения допускают реализацию схемами того же порядка сложности и логарифмической глубины (см., например, гл. 12 в [10]). Заметим, что метод из книги [10], опубликованной в 1994 г., применим ко всем более поздним алгоритмам матричного умножения, которые здесь цитируются.

3. Переход к нормальному базису

Рассмотрим задачу преобразования координат элемента $x \in GF(p^n)$ из полиномиального базиса $A = \{1, t, t^2, \dots, t^{n-1}\}$ в нормальный базис

$$B = \{\alpha_0 = t, \alpha_1 = t^p, \alpha_2 = t^{p^2}, \dots, \alpha_{n-1} = t^{p^{n-1}}\}.$$

Пусть $n \leq ms$.

Алгоритм 2. 1. Считаем, что известны частичные базисные разложения (относительно первых m элементов базиса B) для $1, t, t^2, \dots, t^{n-1}$, то есть

$$t^i = t_{i,0}\alpha_0 + t_{i,1}\alpha_1 + \dots + t_{i,m-1}\alpha_{m-1} + \beta_i,$$

где $\beta_i \in \langle \alpha_m, \dots, \alpha_{n-1} \rangle$, $i = 0, 1, \dots, m-1$.

2. При помощи алгоритма 1 вычисляются $x^{p^m}, x^{p^{2m}}, \dots, x^{p^{m(s-1)}}$ в полиномиальном базисе со сложностью $O(sn^w)$.

3. Вычисляются частичные разложения для $x^{p^{mi}}$ подстановкой известных разложений для $1, t, t^2, \dots, t^{n-1}$. Справедливо равенство

$$\begin{bmatrix} x \\ x^{p^m} \\ \vdots \\ x^{p^{m(s-1)}} \end{bmatrix}_{s \times m}^B = \begin{bmatrix} x \\ x^{p^m} \\ \vdots \\ x^{p^{m(s-1)}} \end{bmatrix}_{s \times n}^A \begin{bmatrix} 1 \\ t \\ \vdots \\ t^{n-1} \end{bmatrix}_{n \times m}^B,$$

где в строчках матриц выписаны коэффициенты в разложениях по базисам, указанным в верхних индексах матриц (для нормального базиса B имеется в виду частичное разложение).

Частичное разложение $x^{p^{mi}}$ определяет m координат из полного разложения $x = x_0\alpha_0 + \dots + x_{n-1}\alpha_{n-1}$ в базисе B , а именно,

$$x^{p^{mi}} = x_{n-mi}\alpha_0 + x_{n-mi+1}\alpha_1 + \dots + x_{n-m(i-1)-1}\alpha_{m-1} + \dots$$

Здесь для $j \notin [0, n-1]$ полагаем $x_j \equiv x_{j \bmod n}$. Так как все $x^{p^{mi}}$ вычислены, получено полное разложение x в нормальном базисе. Сложность шага оценивается как сложность умножения $s \times n$ матрицы на $n \times m$ матрицу.

3.1. Оценка сложности алгоритма 2

Сложность шага 3 оценим как сложность s умножений матрицы размера $s \times m$ на $m \times m$ матрицу. Известно (см., например, [7]), что умножение $m^h \times m$ матрицы на $m \times m$ матрицу, где $h \leq 0,294$, выполняется со сложностью $O(m^{2+\varepsilon})$ при любом $\varepsilon > 0$. Положим $s \sim n^{1-w/2}$, $m \sim n^{w/2}$. При $w = 1,667$ выполнено условие $s = o(m^{0,294})$, поэтому сложность обоих шагов и алгоритма в целом оценивается как $O(n^{1+w/2}) \sim O(n^{1,834})$. В отношении глубины справедливо то же замечание, что и для алгоритма 1.

4. Переход к полиномиальному базису

Рассмотрим задачу перехода от нормального представления элемента поля

$$x = x_0\alpha_0 + \dots + x_{n-1}\alpha_{n-1}$$

к полиномиальному. Пусть $n \leq ms$. Справедливо равенство

$$x = \gamma_0 + \gamma_1^{p^m} + \dots + \gamma_{s-1}^{p^{m(s-1)}},$$

где

$$\gamma_i = x_{mi}\alpha_0 + \dots + x_{mi+m-1}\alpha_{m-1}.$$

Если $j \geq n$, полагаем $x_j \equiv 0$.

Алгоритм 3. 1. Считаем, что для $\alpha_0, \alpha_1, \dots, \alpha_{m-1}$ представления в полиномиальном базисе известны.

2. По заданным представлениям для первых m элементов нормального базиса вычисляются полиномиальные представления для γ_i ,

$$\begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \dots \\ \gamma_{s-1} \end{bmatrix}_{s \times n}^A = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \dots \\ \gamma_{s-1} \end{bmatrix}_{s \times m}^B \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \dots \\ \alpha_{m-1} \end{bmatrix}_{m \times n}^A$$

в прежних обозначениях. Сложность шага оценивается как сложность умножения $s \times m$ матрицы на $m \times n$ матрицу.

3. Вычисляются $\gamma_i^{p^{mi}}$ алгоритмом 1 со сложностью $O(sn^w)$ и складываются покомпонентно со сложностью, не превосходящей ns .

Оценивание сложности производится в точности как в алгоритме 2. Таким образом, показано, что переход между базисами может выполняться схемой над $GF(p)$ со сложностью $O(n^{1,834})$ и глубиной $O(\log n)$.

5. Уточнение оценки сложности

В этом параграфе будет рассмотрен модифицированный способ реализации шага 2 в алгоритме 2 и шага 3 в алгоритме 3, который позволит для обоих алгоритмов получить оценку сложности $O(n^{1,806})$. Для этого нам потребуется распространить метод Brenta–Кунга из раздела 2 на решение следующей задачи.

Пусть заданы элементы $x_1, x_2, \dots, x_l \in GF(p^n)$ своими представлениями в полиномиальном базисе поля и натуральные параметры s и m . Требуется вычислить все $x_i^{p^{jm}}$, $j = 1, \dots, s$, также в полиномиальном базисе. Иначе говоря, рассматривается одновременная реализация s последовательных степеней операции Фробениуса $x \rightarrow x^{p^m}$ на l элементах поля.

Введем обозначение $\xi_j(t) = t^{p^{jm}} \bmod m_n(t)$. Пусть элементу x_i соответствует многочлен $g_i(t)$. Представим $g_i(t)$ в виде (1) (см. раздел 2). Матричное тождество (2) (см.

алгоритм 1), соответствующее вычислению $x_i^{p^{jm}}$, то есть композиции многочленов $g_i(\xi_j)$, запишем сокращенно в виде $X_{i,j} = Y_i C_j$. Тогда

$$\begin{bmatrix} X_{1,1} & \dots & X_{1,s} \\ \vdots & \ddots & \vdots \\ X_{l,1} & \dots & X_{l,s} \end{bmatrix}_{lq \times sn} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_l \end{bmatrix}_{lq \times r} [C_1 \dots C_s]_{r \times sn},$$

где $n \leq rq$.

Все $x_i^{p^{jm}}$ вычисляются при помощи ls алгоритмов 1, у которых шаг 2 выполняется совместно посредством умножения $lq \times r$ матрицы на $r \times sn$ матрицу, как записано выше. Для остальных шагов, которые выполняются независимо, оценка сложности есть $O(ls q M(n))$.

Если $l \leq n$, то параметры q и r можно выбрать таким образом, что $lq \sim r \sim \sqrt{ln}$. Пусть $l \sim n^\lambda$ и $s \sim n^\sigma$, где $\lambda, \sigma \in [0, 1]$. Тогда сложность совмещенного шага 2 оценивается как сложность умножения матриц $n^{(1+\lambda)/2} \times n^{(1+\lambda)/2}$ и $n^{(1+\lambda)/2} \times n^{1+\sigma}$. Введя обозначение $\tau = 2(1+\sigma)/(1+\lambda)$, получаем для этого шага оценку сложности $O(n^{(1+\lambda)w_\tau/2})$, где w_τ — экспонента умножения матриц размера $n \times n$ и $n \times n^\tau$. Экспонента дополнительной сложности имеет вид $1 + \sigma + (1+\lambda)/2 = (1+\lambda)(1+\tau)/2$ и не превосходит $(1+\lambda)w_\tau/2$ в силу очевидного неравенства $w_\tau \geq \tau + 1$.

Таким образом, общая сложность вычислений с точностью до логарифмической составляющей оценивается как $O(n^{(1+\lambda)w_\tau/2})$. В [7] показано, что w_τ не превосходит минимума по $\theta \in [0, 1]$, $q \in \mathbb{N}$ величины

$$\frac{1}{(1-\theta) \log q} (2 + \tau) \log(q + 2) + \theta \log \theta + ((1+\tau)(1-\theta)) \log((1+\tau)(1-\theta)) + (1 + \tau\theta) \log(1 + \tau\theta) - (2 + \tau) \log(2 + \tau). \quad (3)$$

Следующая лемма мотивируется тем, что оценка сложности умножения матриц размера $\sqrt{n} \times \sqrt{n}$ и $\sqrt{n} \times n^{1+\sigma}$, вытекающая из (3), хуже, чем аналогичная оценка для умножения матриц размера $n^{(1+\sigma)/2} \times n^{(1+\sigma)/2}$ и $n^{(1+\sigma)/2} \times n$.

Лемма 1. Пусть $\delta > 0$ и элемент $x \in GF(p^n)$ задан в полиномиальном базисе. Пусть $s \sim n^\sigma$, где $\sigma \in [0, 1]$, и $\tau = 2/(1+\sigma)$. Тогда вычисление всех $x^{p^{jm}}$, $j = 1, \dots, s$, может быть выполнено схемой над $GF(p)$ сложности $O(n^{\delta+w_\tau/\tau})$ и глубины $O(\delta^{-1} \log n)$.

Доказательство. Пусть $k \in \mathbb{N}$ и $s \leq s_1 \dots s_k$, где все $s_i \sim \sqrt[k]{s}$. Введем обозначения $u_i = s_1 \dots s_i$, $n_i = s_{i+1} \dots s_k$ и $m_i = m n_i$, положим $n_k = 1$. Рассмотрим следующий порядок вычислений.

На первом шаге вычисляются все $x^{p^{jm_1}}$, $j = 1, \dots, s_1$.

Пусть $2 \leq i \leq k$. После $(i-1)$ -го шага вычислены все $x^{p^{jm_{i-1}}}$, $j = 1, \dots, u_{i-1}$. На i -м шаге вычисляется результат применения операций Фробениуса со степенями p^{jn_i} , $j = 1, \dots, s_i$, ко всему набору. Получаем все $x^{p^{jm_i}}$, $j = 1, \dots, u_i$.

В частности, после k -го шага известны все $x^{p^{jm}}$, $j = 1, \dots, s$.

Пусть каждый шаг выполняется алгоритмом 1, модифицированным так, как указано в начале этого параграфа. Как показано выше, сложность i -го шага определяется умножением $n^{(k+(i-1)\sigma)/2k} \times n^{(k+(i-1)\sigma)/2k}$ матрицы на $n^{(k+(i-1)\sigma)/2k} \times n^{1+\sigma/k}$ матрицу. Сложность этой операции оценивается как сложность $n^{(k-i+1)\sigma/(k(1+\sigma))+\sigma/k}$ умножений матриц с размерами $n^{(k+(i-1)\sigma)/2k}$, $n^{(k+(i-1)\sigma)/2k}$ и $n^{\tau(k+(i-1)\sigma)/2k}$. Для экспоненты матричного

умножения справедлива оценка

$$\begin{aligned} w_\tau \left(\frac{1}{2} + \frac{(i-1)\sigma}{2k} \right) + \frac{(k-i+1)\sigma}{k(1+\sigma)} + \frac{\sigma}{k} &= \frac{w_\tau}{2} + \frac{\sigma}{2k}((i-1)w_\tau + (k-i+1)\tau) + \frac{\sigma}{k} \\ &= \frac{w_\tau}{\tau} - \frac{\sigma}{2k}(k-i+1)(w_\tau - \tau) + \frac{\sigma}{k} \\ &\leq \frac{w_\tau}{\tau} + \frac{\sigma}{2k} - \frac{(k-i)\sigma}{2k}, \end{aligned}$$

где последний переход следует из неравенства $w_\tau \geq \tau + 1$.

Суммарная сложность вычислений по всем шагам оценивается с использованием формулы для суммы геометрической прогрессии со знаменателем $n^{-\sigma/2k}$ как $O(n^{w_\tau/\tau + O(1/k)} / (1 - n^{-\sigma/2k}))$. Окончательно утверждение леммы следует из выбора $k \sim c/\delta$, где c — подходящая константа.

Лемма 2. Пусть элементы $x_0, \dots, x_{s-1} \in GF(p^n)$ заданы в полиномиальном базисе, $\delta > 0$ и $s \sim n^\sigma$, где $\sigma \in [0, 1]$. Пусть $\tau = 2/(1 + \sigma)$. Тогда сумма $\sum x_j^{p^{jm}}$ может быть вычислена схемой над $GF(p)$ сложности $O(n^{\delta + w_\tau/\tau})$ и глубины $O(\delta^{-1} \log n)$.

Доказательство. Выберем $k \in \mathbb{N}$ и $s_i \sim \sqrt[k]{s}$ так, что $s \leq s_1 \dots s_k$. Введем обозначения $u_i = s_1 \dots s_i$, $n_i = s_{i+1} \dots s_k$ и $m_i = tu_i$, где $n_k = 1$. Положим $x_{0,j} = x_j$, если $j < s$, и $x_{0,j} = 0$ в противном случае. Ясно, что

$$\sum_{j=0}^{s-1} x_j^{p^{jm}} = \sum_{j=0}^{n_0-1} x_{0,j}^{p^{jm}} = \sum_{j=0}^{n_1-1} x_{1,j}^{p^{jm_1}} = \dots = \sum_{j=0}^{n_i-1} x_{i,j}^{p^{jm_i}} = \dots = \sum_{j=0}^{n_{k-1}-1} x_{k-1,j}^{p^{jm_{k-1}}},$$

где $x_{i,j} = x_{i-1,js_i} + x_{i-1,js_i+1}^{p^{m_{i-1}}} + \dots + x_{i-1,js_i+s_i-1}^{p^{(s_i-1)m_{i-1}}}$.

Рассмотрим алгоритм, в котором на каждом i -м шаге, $1 \leq i \leq k$, вычисляются все $x_{i,j}$, $j = 0, 1, \dots, n_i - 1$. Результат последнего шага, $x_{k,0}$, совпадает с искомой суммой.

На i -м шаге выполняются s_i операций Фробениуса $x \rightarrow x^{p^{m_{i-1}}}$ на соответствующих наборах x_{i-1,js_i+l} , $j = 0, 1, \dots, n_i - 1$, где $l = 0, 1, \dots, s_i - 1$. Число сложений на всех шагах равно в точности $s - 1$, что дает вклад $O(sn)$ в сложность алгоритма.

Таким образом, если воспользоваться алгоритмом из начала раздела, то сложность i -го шага определяется $n^{\sigma/k}$ умножениями $n^{(k+(k-i)\sigma)/2k} \times n^{(k+(k-i)\sigma)/2k}$ и $n^{(k+(k-i)\sigma)/2k} \times n$ матриц.

Заключительное рассуждение повторяет завершение доказательства леммы 1 (сложность i -го шага здесь оценивается так же, как и сложность $(k-i+1)$ -го шага в лемме 1).

Замечание 1. Если порядок глубины не является принципиальным, то в обеих леммах можно выбрать $k = \lceil \log_2 s \rceil$ и все $s_i = 2$. Оценка для сложности примет вид $O(n^{w_\tau/\tau} + n^{1/\tau} M(n))$, а для глубины — $O(\sigma \log^2 n)$.

Теорема 1. Переход между соответствующими нормальным и полиномиальным базисами поля $GF(p^n)$ может быть реализован схемой над $GF(p)$ сложности $O(n^{1,806})$ и глубины $O(\log n)$.

Доказательство. Как следует из доказанных лемм, порядок сложности алгоритмов 2 и 3 определяется умножением матриц размера $n^\sigma \times n^{1-\sigma}$ и $n^{1-\sigma} \times n$ (на шагах 3 и 2 соответственно), и умножением матриц размера $n^{(1+\sigma-O(\delta))/2} \times n^{(1+\sigma-O(\delta))/2}$ и $n^{(1+\sigma-O(\delta))/2} \times n^{1+O(\delta)}$ (на шагах 2 и 3), где $\sigma \in [0, 1]$.

Положим $\sigma = 0,195$. Так как $\sigma < 0,294(1 - \sigma)$, первое умножение выполняется со сложностью $O(n^{2-\sigma+\varepsilon})$. Также справедливо неравенство $w_\tau/\tau < 1,806$, что следует из оценки (3) при выборе $\theta = 0,02$ и $q = 8$. Выбирая ε и δ достаточно малыми, получаем для обоих умножений оценку $O(n^{1,806})$.

Следствие 1. Умножение в нормальном базисе поля $GF(p^n)$ может выполняться схемой над $GF(p)$ сложности $O(n^{1,806})$ и глубины $O(\log n)$.

Отметим, что константы под знаком O в этих оценках очень велики, во-первых, из-за большого (порядка тысячи) числа итераций и во-вторых, из-за больших мультипликативных констант в оценках для матричных умножений.

6. Замечания

6.1. О методе Калтофена–Шаупа

Алгоритм 3 перехода к полиномиальному базису изложен в работе [9] с той разницей, что шаг 3 реализуется в схеме Горнера с m последовательно выполняемыми операциями Фробениуса. Оценка сложности получается такой же, что и в разделе 4, однако оценка глубины имеет вид $O(m \log n)$ и является полиномиальной относительно n .

Авторами статьи [9] также доказано существование алгоритма, имеющего сложность $O(n^{1,815})$ (с учетом результата [7] эта оценка уточняется до $O(n^{1,806})$). Доказательство основано на построении алгоритма такой сложности для в определенном смысле двойственной операции и не является конструктивным. Конструктивное доказательство содержится в разделе 5. В известных автору работах, опубликованных после 1998 г. и посвященных реализации операций в конечных полях, этот результат из [9] не упоминается, что отчасти объясняется отсутствием аналогичного алгоритма для перехода к нормальному базису.

6.2. О вычислениях в произвольном полиномиальном базисе

Переход между произвольными полиномиальными базисами поля $GF(p^n)$ выполняется методом Брента–Кунга не сложнее, чем за $O(n^{1,667})$ операций. Действительно, пусть $x \in GF(p^n)$ представляется многочленом $f(\beta)$ в полиномиальном базисе A_β с генератором β . Далее, пусть β представляется многочленом $\xi(\alpha)$ в полиномиальном базисе A_α с генератором α , корнем неприводимого многочлена $m_n(t)$. В таком случае $x = f(\xi(\alpha))$, то есть в базисе A_α элемент x представляется многочленом $f(\xi(t)) \bmod m_n(t)$. Этот многочлен можно вычислить алгоритмом 1.

В частности, со сложностью $O(n^{1,667})$ и глубиной $O(\log n)$ можно перейти от произвольного полиномиального базиса к полиномиальному базису, генератор которого также порождает нормальный базис поля, и обратно. Таким образом, из теоремы 1 автоматически вытекает следующий результат.

Следствие 2. Переход между двумя любыми нормальными или полиномиальными базисами поля $GF(p^n)$ может быть выполнен схемой над $GF(p)$ сложности $O(n^{1,806})$ и глубины $O(\log n)$.

Для полноты картины отметим, что переход между двумя нормальными базисами можно выполнять быстрее, со сложностью порядка $M(n)$. Пусть элемент $x = x'_0\beta_0 + \dots + x'_{n-1}\beta_{n-1}$ задан в нормальном базисе B_β с генератором β , и требуется

найти представление $x = x_0\alpha_0 + \dots + x_{n-1}\alpha_{n-1}$ в нормальном базисе B_α с генератором α . Пусть $\beta = b_0\alpha_0 + \dots + b_{n-1}\alpha_{n-1}$. Легко проверить, что

$$\begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_{n-1} \end{bmatrix}^{B_\alpha} = \begin{bmatrix} b_0 & b_{n-1} & \dots & b_1 \\ b_1 & b_0 & \dots & b_2 \\ \dots & \dots & \dots & \dots \\ b_{n-1} & b_{n-2} & \dots & b_0 \end{bmatrix}_{n \times n} \begin{bmatrix} x'_0 \\ x'_1 \\ \dots \\ x'_{n-1} \end{bmatrix}^{B_\beta}.$$

Матрица перехода является циркулянтной, то есть все ее строки получаются из первой строки последовательными циклическими сдвигами. Умножение на такую матрицу, как известно, сводится к умножению многочленов. А именно, все x_i являются коэффициентами многочлена $(b_{n-1} + b_{n-2}y + \dots + b_0y^{n-1})(x'_{n-1} + x'_{n-2}y + \dots + x'_0y^{n-1}) \bmod y^n - 1$, который вычисляется со сложностью $M(n) + n$ и глубиной $O(\log n)$. Заметим, что такое умножение на самом деле есть циклическая свертка, которая фактически является частью алгоритмов умножения Штрассена и Шёнхаге (см., например, гл. 8 в [3]).

6.3. Об умножении в нормальных базисах

В этом пункте будет рассмотрена еще одна пара алгоритмов перехода между нормальными и полиномиальными базисами, связанная со стандартным методом умножения в нормальных базисах. При определенных ограничениях для этих алгоритмов можно получить лучшую оценку сложности, чем для рассмотренных в основном тексте.

Умножение в нормальном базисе B выполняется стандартным алгоритмом Мессии-Омура (см., например, [2, 4]) со сложностью $O(C_B n)$, где C_B — сложность базиса, понятие, вытекающее из структуры этого алгоритма. Сложность нормального базиса B с генератором α определяется как суммарное число ненулевых коэффициентов в разложении элементов $\alpha\alpha_0, \alpha\alpha_1, \dots, \alpha\alpha_{n-1}$ в базисе B . Непосредственно из определения вытекает, что сложность операции умножения на элемент базиса B не превосходит $2C_B - n$ при глубине $\lceil \log_2 n \rceil + 1$. Известно, что $C_B \geq 2n - 1$ (также см. [2, 4]).

Как следует из теоремы 1, оценки сложности умножения в нормальном базисе стандартным алгоритмом и алгоритмом с использованием перехода к полиномиальному базису находятся в отношении $O(C_B/n^{0,806})$. Далее будет показано, что для базисов низкой сложности это отношение имеет больший порядок при условии, что p не очень велико.

Теорема 2. Переход к нормальному базису B в поле $GF(p^n)$ может быть реализован схемой сложности $O(\sqrt{n}C_B) + O(n^{1,667})$ и глубины $O(\min\{\sqrt{n}, p \log_p n\} \log n)$.

Доказательство. Пусть элемент поля в стандартном базисе представлен многочленом $f(t)$ степени не выше $n - 1$. Пусть $n \leq rq$, где $q, r \sim \sqrt{n}$ и

$$f(t) = f_0(t^r) + t f_1(t^r) + \dots + t^{r-1} f_{r-1}(t^r),$$

где $\deg f_i < q$.

Алгоритм 4. 1. Считаем, что нормальные представления для t^{ir} , $i = 0, 1, \dots, q - 1$, вычислены предварительно.

2. Нормальное представление элементов $f_i(t^r)$ получается умножением $r \times q$ матрицы коэффициентов многочленов f_i на $q \times n$ матрицу представления элементов t^{ir} в нормальном базисе.

3. Далее, пусть $p_0 p^s < r \leq (p_0 + 1)p^s$, где $0 < p_0 < p$ и $s \geq 0$. Для всех $i = (p_0 - 1)p^s, \dots, r - p^s - 1$ вычислим элементы $g_i = t^{p^s} f_{i+p^s}(t^r) + f_i(t^r)$ в нормальном представлении (заметим, что $t^{p^s} = \alpha_s$, поэтому сложность одной такой операции не превосходит C_B) и положим $g_i = f_i(t^r)$ для $i < (p_0 - 1)p^s$ и $i \geq r - p^s$. Справедливо равенство

$$f(t) = g_0 + t g_1 + \dots + t^{r_1-1} g_{r_1-1}, \quad (4)$$

где $r_1 = p_0 p^s$. Повторяем указанную процедуру для формулы (4) и т. д. Для нахождения окончательного результата требуется всего

$$p_0 + (p - 1) \lfloor \log_p(r - 1) \rfloor \sim \min\{\sqrt{n}, p \log_p \sqrt{n}\}$$

таких шагов, на которых выполняется в совокупности $r - 1$ операций, состоящих из умножения на элемент нормального базиса и сложения.

Теорема 3. *Переход к полиномиальному базису в поле $GF(p^n)$ можно выполнить схемой сложности $O(n^{1,667}) + O(n^{1,5} \log p \log n \log \log n)$ и глубины $O(\sqrt{n} \log p \log n)$.*

Доказательство. Пусть $n \leq r q$ и $q, r \sim \sqrt{n}$. Элемент поля в нормальном базисе представим в виде

$$\gamma_0 + (\gamma_1)^p + (\gamma_2)^{p^2} + \dots + (\gamma_{r-1})^{p^{r-1}},$$

где все γ_i — линейные комбинации базисных элементов α_{jr} , $j = 0, 1, \dots, q - 1$.

Алгоритм 5. 1. Считаем, что известны полиномиальные представления элементов α_{jr} , $j = 0, 1, \dots, q - 1$.

2. Полиномиальное представление элементов γ_i получается умножением $r \times q$ матрицы коэффициентов γ_i в частичном базисе из элементов α_{jr} на $q \times n$ матрицу представления элементов α_{jr} в полиномиальном базисе.

3. Воспользуемся вариантом схемы Горнера. Положим $g_{r-1} = \gamma_{r-1}$. Далее вычисляем $g_{r-2} = \gamma_{r-2} + g_{r-1}^p$ и т. д. по формулам $g_i = \gamma_i + g_{i+1}^p$. Всего для завершения вычислений нужно выполнить последовательно $r - 1$ действий, состоящих из возведения в степень p и сложения со сложностью $O(M(n) \log p)$ каждое (так как возведение в степень p выполняется в аддитивной цепочке длины $O(\log p)$).

Окончательно, если $\log p = O(n^{0,167})$, то рассматриваемое отношение оценок сложности умножения для алгоритма Месси–Омура и алгоритма с переходом к полиномиальному базису принимает вид $O(C_B/n^{0,667})$, когда $C_B = O(n^{1,167})$, и $O(\sqrt{n})$ в противном случае (при $C_B \gtrsim n^{1,306}$ приоритет имеет исходная оценка $O(C_B/n^{0,806})$).

6.4. О реализации всех автоморфизмов Фробениуса

То, что выполнение некоторых операций в нормальном базисе может быть ускорено за счет перехода к полиномиальному базису (и затем, обратно), кажется правдоподобным в свете уже доказанного. Рассмотрим операцию вычисления всех x^{p^i} , $i = 0, 1, \dots, n - 1$, которая в нормальном базисе не требует схемных затрат, и следует ожидать, что при ее реализации в полиномиальном базисе можно извлечь выгоду от перехода к нормальному базису.

Оценку $O(M(n^2) \log n) = O(n^2 \log^2 n \log \log n)$ можно получить методом, изложенным в [11] (см. также гл. 14 в [3]). Заметим, что на самом деле, в [11] рассмотрена более общая операция вычисления всех $x^{p^i} \bmod f$, где многочлен f также является входом алгоритма. В основе метода работы [11] лежит способ быстрого вычисления значений многочлена степени $n - 1$ в n точках, которым определяется глубина метода $O(\log^2 n)$ в общем случае.

Однако в известной степени более простым способом можно получить даже несколько лучшую оценку сложности, а именно, $O(nM(n)) = O(n^2 \log n \log \log n)$ при глубине $O(\log n)$. Действительно, любым методом сложности $O(n^2)$ найдем разложение $x = x_0\alpha_0 + \dots + x_{n-1}\alpha_{n-1}$ в некотором нормальном базисе B . По свойству нормального базиса наборы координат элементов x^{p^i} совпадают с точностью до циклического сдвига. Поэтому для завершения вычислений остается перейти обратно к полиномиальному представлению, то есть вычислить произведение циркулянтной матрицы

$$X = \begin{bmatrix} x \\ x^p \\ \dots \\ x^{p^{n-1}} \end{bmatrix}_{n \times n}^B = \begin{bmatrix} x_0 & x_1 & \dots & x_{n-1} \\ x_{n-1} & x_0 & \dots & x_{n-2} \\ \dots & \dots & \dots & \dots \\ x_1 & x_2 & \dots & x_0 \end{bmatrix}$$

на матрицу перехода C размера $n \times n$. Компоненты вектора-произведения матрицы X на произвольный столбец $(c_0, c_1, \dots, c_{n-1})^T$ матрицы C вычисляются со сложностью $O(M(n))$ и глубиной $O(\log n)$ (см. пункт 6.2), откуда и следуют заявленные оценки. Заметим, что при помощи стандартного рассуждения оценку сложности можно привести к виду $O(n^2 \log n \log \log n / \log \log \log_p n)$, имеющему смысл при $p = o(n)$.

6.5. О проверке линейной независимости нормальной системы

В приложениях часто встречается задача проверить, порождает ли выбранный элемент β нормальный базис в поле $GF(p^n)$, иначе говоря, является ли нормальная система $\{\beta, \beta^p, \dots, \beta^{p^{n-1}}\}$ линейно независимой над $GF(p)$. Обычно предполагается, что β задан в полиномиальном представлении поля.

В [12] предложен алгоритм проверки, для которого, с учетом замечания из [11], справедлива оценка сложности $O(M^2(n) \log n) = O(n^2 \log^3 n \log^2 \log n)$. Глубина имеет порядок $O(\log^2 n)$, однако в методе [12] используется алгоритм Евклида, поэтому его затруднительно реализовать в виде схемы. В [9] предложен вероятностный алгоритм для той же задачи со сложностью (с учетом [7]) $O(n^{1.806})$. Покажем, что проверку линейной независимости нормальной системы можно выполнить схемой сложности $O(n^{1.806})$ и глубины $O(\log n)$.

Пусть известно, что некоторый элемент α является генератором нормального базиса B_α в поле $GF(p^n)$. Со сложностью $O(n^{1.806})$ и глубиной $O(\log n)$ можно получить представление $\beta = b_0\alpha_0 + \dots + b_{n-1}\alpha_{n-1}$ в этом базисе. Нетрудно показать (см., например, главу 3 в [2]), что система $\{\beta, \beta^p, \dots, \beta^{p^{n-1}}\}$ является базисом тогда и только тогда, когда обратима циркулянтная матрица

$$\begin{bmatrix} b_0 & b_{n-1} & \dots & b_1 \\ b_1 & b_0 & \dots & b_2 \\ \dots & \dots & \dots & \dots \\ b_{n-1} & b_{n-2} & \dots & b_0 \end{bmatrix}_{n \times n}.$$

Обратимость такой матрицы, в свою очередь, эквивалентна тому, что многочлен

$$b(y) = b_{n-1} + b_{n-2}y + \dots + b_0y^{n-1}$$

обратим по модулю $y^n - 1$ (о свойствах циркулянтных матриц более подробно см. в гл. 2 в [10]). Для обратимости многочлена по модулю другого необходимо и достаточно, чтобы они были взаимно просты.

Чтобы установить, являются ли многочлены, один из которых, $y^n - 1$, фиксирован, взаимно простыми, можно воспользоваться вариантом метода Брента-Кунга. Пусть известно разложение

$$y^n - 1 = \varphi_1^{e_1}(y) \dots \varphi_k^{e_k}(y)$$

на неприводимые множители. Введем обозначение $\deg \varphi_i = d_i$. Очевидно, $\sum_{i=1}^k d_i \leq n$. Пусть $n \leq rq$ и $b(y) = f_0(y) + f_1(y)y^r + \dots + f_{q-1}(y)y^{(q-1)r}$, где $\deg f_i < r$.

Алгоритм 6. 1. Считаем, что все остатки $\psi_{i,j}(y) = y^{ir} \bmod \varphi_j(y)$ вычислены заранее.

2. Операция вычисления остатков от деления многочлена степени $r - 1$ на многочлены с постоянными коэффициентами $\varphi_j(y)$ является линейным $(r \times d)$ -оператором, где $d \leq n$. Таким образом, нахождение всех остатков $\rho_{i,j}(y) = f_i(y) \bmod \varphi_j(y)$ сводится к умножению $q \times r$ матрицы, составленной из коэффициентов многочленов f_i , на $r \times d$ матрицу этого оператора.

3. Вычисляются все произведения $\rho_{i,j}(y)\psi_{i,j}(y)$. Сложность шага можно оценить как $O(qM(n))$, так как вычисление k произведений многочленов степени d_i по порядку не сложнее, чем умножение многочленов степени $n - 1$ (на этом шаге выполняется q таких операций).

4. Поскольку

$$b(y) \bmod \varphi_j(y) = \sum_{i=0}^{q-1} \rho_{i,j}(y)\psi_{i,j}(y) \bmod \varphi_j(y),$$

для завершения вычислений остается при каждом j сложить полученные на предыдущем шаге произведения $\rho_{i,j}(y)\psi_{i,j}(y)$ и привести результат по модулю $\varphi_j(y)$. Сложность этого шага также не превосходит $O(qM(n))$.

Если все остатки, найденные указанным алгоритмом, отличны от нуля, то β является генератором нормального базиса, и не является в противном случае. Выбирая $q, r \sim \sqrt{n}$ и учитывая, что $d \leq n$, получаем оценки сложности $O(n^{1,667})$ и глубины $O(\log n)$ для алгоритма 6, и $O(n^{1,806})$, $O(\log n)$ — для сложности и глубины алгоритма в целом.

Заметим, что при программной реализации и отсутствии ограничения на глубину, для нахождения общего наибольшего делителя двух многочленов, конечно, можно воспользоваться алгоритмом Евклида с оценками сложности $O(n \log^2 n \log \log n)$ и глубины $O(\log^2 n)$, не требующим разложения многочлена $y^n - 1$ на множители. Для случая программной реализации отметим, что в предложенном методе требуется, чтобы один нормальный базис в поле $GF(p^n)$ уже был построен. Для нахождения нормального базиса можно воспользоваться одним из вероятностных алгоритмов [9, 12].

Автор благодарен своему научному руководителю С. Б. Гашкову за постановку задачи и внимание к работе.

Список литературы

1. Лидл Р., Нидеррейтер Х., *Конечные поля*. Мир, Москва, 1988.
2. Jungnickel D., *Finite fields: structure and arithmetics*. Wissenschaftsverlag, Mannheim, 1995.
3. von zur Gathen J., Gerhard J., *Modern computer algebra*. Cambridge Univ. Press, Cambridge, 1999.
4. Болотов А. А., Гашков С. Б., О быстром умножении в нормальных базисах конечных полей. *Дискретная математика* (2001) **13**, №3, 3–31.
5. Feisel S., von zur Gathen J., Shokrollahi M. A., Normal bases via general Gauss periods. *Math. Comput.* (1999) **68**, №225, 271–290.
6. Brent R., Kung H., Fast algorithms for manipulating formal power series. *J. ACM* (1978) **25**, №4, 581–595.
7. Huang X., Pan V., Fast rectangular matrix multiplication and applications. *J. Complexity* (1998) **14**, 257–299.
8. Лупанов О. Б., *Асимптотические оценки сложности управляющих систем*. Изд-во МГУ, Москва, 1984.
9. Kaltofen E., Shoup V., Subquadratic-time factoring of polynomials over finite fields. *Math. Comput.* (1998) **67**, №223, 1179–1197.
10. Bini D., Pan V., *Polynomial and matrix computations*, **1**. Birkhäuser, Boston, 1994.
11. von zur Gathen J., Shoup V., Computing Frobenius maps and factoring polynomials. *Comput. Complexity* (1992) **2**, 187–224.
12. von zur Gathen J., Giesbrecht M., Constructing normal bases in finite fields. *Symb. Comp.* (1990) **10**, 547–570.

Статья поступила 5.06.2006.