КИБЕРНЕТИКА

А.В. АНИСИМОВ

УДК 519.72

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В СМЕШАННОМ БАЗИСЕ (2, 3)

Ключевые слова: *двоичная система счисления, тритовая система счисления, префиксное кодирование, модулярная арифметика.*

ВВЕДЕНИЕ

В компьютерных приложениях повсеместно доминирует двоичная система счисления. Также хорошо известны троичная и основанная на ней тритовая системы счисления. В настоящей работе вводится и исследуется смешанная система, базирующаяся на рядах степеней чисел два и три. Представление чисел в такой системе счисления получается как частный случай рекурсивного применения разложения чисел в линейные формы в базисе ортогональных числовых последовательностей. Указывается новое универсальное префиксное кодирование чисел с помощью такого представления, обладающее повышенной устойчивостью к ошибкам передачи. Рассматриваются вычислительные аспекты (2,3)-представления чисел.

Пусть $U = \{u_1, u_2, u_3, \ldots\}$ и $V = \{v_1, v_2, v_3, \ldots\}$ — две бесконечные последовательности натуральных чисел. Предполагаем также, что хотя бы одна из них (U или V) неограниченна. Последовательности U и V называем ортогональными, если для любого номера n выполняется условие НОД$(u_n, v_n) = 1$. В этом случае используем обозначения $u_n \perp v_n$ и $U \perp V$.

В дальнейшем все числа предполагаем целыми или натуральными.

Представление натурального числа x в виде линейной комбинации

$$x = au_n + bv_n \tag{1}$$

называем линейной формой в базисе (U, V).

Линейную форму (1) называем положительно определенной, если $x > 0$ и $a \geq 0$, $b \geq 0$.

Индекс n в представлении (1) называем рангом линейного представления.

Известно, что если $u_n \perp v_n$, то любое натуральное число x представимо в виде линейной комбинации вида (1). В настоящей работе нас интересуют только положительно определенные формы, поэтому в дальнейшем, по умолчанию, предполагаем задание натуральных чисел в виде формулы (1) положительно определенным.

Представление (1) называем каноническим, если $0 \leq a < v_n$.

Очевидно, что, применяя тождественное преобразование $x = (a - kv_n)u_n + (b + ku_n)v_n$, всегда можно добиться каноничности представления (1).

Среди всех представлений числа x вида (1) нас интересуют представления максимального ранга. Очевидно, что число x имеет единственное каноническое пред-

ставление максимального ранга. Заметим, что если (1) — представление максимального ранга, то $x < u_{n+1}v_{n+1}$.

Ортогональные линейные формы максимального ранга при рекурсивном применении позволяют строить разнообразные смешанные системы счисления.

Обычное счисление по основанию степеней базового числа M получается рекурсивным применением разложения чисел в максимальные линейные формы по степеням числа M. Точнее, пусть M — базис системы счисления.

Рассмотрим две последовательности: $U = \{M, M^2, M^3\}$ и $V = \{1, 1, 1, \ldots\}$.

Пусть x — натуральное число, отличное от нуля. Положим $x_0 = x$. Существует представление максимального ранга

$$x = a_1 M^{n_1} + x_1 \cdot 1. \tag{2}$$

Среди представлений x вида (2) выбираем представление с максимальным значением a_1. В силу максимальности ранга n_1 выполняются неравенства $a_1 < M$ и $x_1 < M^{n_1}$. Затем аналогично раскладываем x_1 в линейную форму, $x_1 = a_2 M^{n_2} + x_2 \cdot 1$, $a_2 < M$, $n_2 < n_1$. Повторяем процедуру разложения для x_2 и следующих остаточных чисел. В итоге получаем традиционное степенное представление x в системе счисления с основанием M, $x = a_1 M^{n_1} + a_2 M^{n_2} + \ldots + a_m$.

Отметим, что существует неограниченное количество возможных базисных ортогональных последовательностей U и V с различными интересными свойствами. Для решения тех или иных задач можно подбирать соответствующие U и V. В [1] изучались общие свойства представления чисел линейными формами. В работах [2–4] исследовались линейные формы Фибоначчи, т.е. случай, когда ортогональный базис строился с помощью пар последовательных чисел Фибоначчи $u_n = F_{n-1}, v_n = F_n$.

(a,b)-ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

Пусть a и b — взаимно простые натуральные числа. Рассмотрим последовательности $U = \{a, a^2, a^3, \ldots\}$ и $V = \{b, b, b, \ldots\}$.

Пусть x — число, которое не делится на a и не делится на $b, x > b$. Рассмотрим каноническое представление x в базисе (U, V) посредством линейной формы максимального ранга $x = \lambda_1 a^{n_1} + by_1$, $0 < \lambda_1 < b$.

Пусть s_1 — максимальная степень такая, что y_1 делится на b^{s_1}, но не делится на b^{s_1+1}, тогда x получает представление $x = \lambda_1 a^{n_1} + b^{k_1} x_1$, где $0 < \lambda_1 < b$, $n_1 > 0$, $k_1 = s_1 + 1$, $k_1 > 0$, x_1 — остаточное число, взаимно простое с a и b.

Если x_1 допускает положительно определенное разложение в линейную форму в базисе (U, V), то аналогичным образом получаем следующий шаг разложения: $x = \lambda_1 a^{n_1} + b^{k_1}(\lambda_2 a^{n_2} + b^{k_2} x_2)$, где $0 < \lambda_2 < b$, $n_2 > 0$, $k_2 > 0$, x_2 — остаточное число, взаимно простое с a и b.

Многократно применяя разложения остаточных чисел x_i в максимальные линейные формы, получаем представление числа x следующего вида:

$$x = \lambda_1 a^{n_1} + b^{k_1}(\lambda_2 a^{n_2} + b^{k_2}(\lambda_3 a^{n_3} + \ldots + b^{k_{t-1}}(\lambda_t a^{n_t} + b^{k_t} x_{t+1}))\ldots), \tag{3}$$

$0 < \lambda_i < b$, $n_i > 0$, $k_i > 0$, $i = \overline{1, t}$, x_{t+1} — последнее остаточное число, не разложимое в положительно определенную линейную форму.

Задание числа x в виде (3) называем (a, b)-представлением.

(2, 3)-ПРЕДСТАВЛЕНИЕ ЧИСЕЛ

Рассмотрим частный случай представления (3), который получается при выборе в качестве базисных ряда степеней 2 и ряда, состоящего из последовательности констант, равных 3, $U = \{2, 2^2, 2^3, \ldots\}$ и $V = \{3, 3, 3, \ldots\}$. Пусть x — нечетное число, которое не делится на 3. В этом случае коэффициенты λ_i в (3) могут равняться только 1 и представление чисел в такой системе счисления принимает достаточно простой вид.

Рассмотрим (2, 3)-представление чисел более детально. Пусть x — нечетное натуральное число, не делящееся на 3 и отличное от 1. Его можно представить в виде $x = 2^{n_1} + 3y_1$. Здесь n_1 — максимальная степень числа 2 такая, что $x - 2^{n_1}$ делится на 3 и $x - 2^{n_1} > 0$.

Как и в общем случае, выделяя из y_1 все делители 3, получаем представление x в виде $x = 2^{n_1} + 3^{k_1} x_1$, где x_1 — нечетное число, не делящееся на 3, и $k_1 > 0$.

Заметим, что среди нечетных чисел, не делящихся на 3, только 1 не имеет положительно определенного (2,3)-представления. Поэтому если $x_1 \neq 1$, то переходим к следующему шагу представления остаточного числа x_1 в виде (2,3)-линейной формы максимального ранга $x = 2^{n_1} + 3^{k_1}(2^{n_2} + 3y_2)$.

Выделяя из y_2 максимальную степень числа 3, на которое делится y_2, получаем представление x в следующем виде:

$$x = 2^{n_1} + 3^{k_1}(2^{n_2} + 3^{k_2} x_2).$$

Затем раскладываем в линейную (2, 3)-форму остаточное число x_2.

Повторяя процедуру необходимое число раз, получаем представление числа x следующего структурного вида:

$$x = 2^{n_1} + 3^{k_1}(2^{n_2} + 3^{n_2}(2^{n_3} + 3^{k_3}(\ldots(2^{n_t} + 3^{k_t}))\ldots)). \tag{4}$$

Пример 1. $20092009 = 2^{24} + 3(2^{20} + 3(2^{13} + 3^2(2^{10} + 3^2(2^3 + 3^2))))$.

Задание (4) обладает многими интересными свойствами.

Назовем блоком (i-м блоком) выражение, выделенное из (4), вида

$$2^{n_i} + 3^{k_i} 2^{n_{i+1}},\ i = \overline{1, t}\ .$$

Эта часть получена на i-м шаге (2,3)-разложения остаточного числа $x_{i-1} = 2^{n_i} + 3^{k_i} x_i$ и $(i+1)$-го шага соответствующего разложения остаточного числа x_i. Для удобства полагаем $n_{t+1} = 0$ и $x_t = 1$. Разложение (4) определяет последовательность блоков:

$$2^{n_1} + 3^{k_1} 2^{n_2},\ 2^{n_2} + 3^{k_2} 2^{n_3}, \ldots, 2^{n_{t-1}} + 3^{k_{t-1}} 2^{n_t},\ 2^{n_t} + 3^{k_t}. \tag{5}$$

Пусть $\overline{n}_i$ означает максимальную степень двойки в традиционном двоичном представлении остаточного числа x_{i-1} разложения (4). Блок $2^{n_i} + 3^{k_i} 2^{n_{i+1}}$ из (5) назовем максимальным, если $n_i = \overline{n}_i$.

Теорема 1. Для любого i-го блока разложения (4), $i = \overline{1, t}$, выполняются следующие свойства:

a) либо $n_i = \overline{n}_i$, либо $n_i = \overline{n}_i - 1$;

b) выполняется неравенство

$$k_i(\log_2 3 - 1) - \log_2 3 < n_i - \overline{n}_{i+1} - k_i; \tag{6}$$

c) если блок максимальный, то выполняется неравенство

$$k_i(\log_2 3 - 1) < n_i - \overline{n}_{i+1} - k_i; \tag{7}$$

d) если блок не максимальный, то выполняется неравенство

$$n_i - \overline{n}_{i+1} - k_i < k_i(\log_2 3 - 1) + 1.$$

Доказательство. Пункт a) легко вытекает из следующего замечания. Для любого натурального числа s, $s > 0$, $2^s \bmod 3 \neq 2^{s-1} \bmod 3$. Поэтому либо $x_i \equiv 2^{\overline{n}_i} \bmod 3$, либо $x_i \equiv 2^{\overline{n}_i - 1} \bmod 3$. Это определяет n_i согласно a).

Перейдем к доказательству свойства b). Для числа x_{i-1} должно выполняться неравенство $x_{i-1} < 2^{\overline{n}_i + 1}$. Учитывая свойство a), получаем неравенство $\overline{n}_i + 1 \le n_i + 2$. Поэтому выполняется неравенство $2^{n_i} + 3^{k_i} 2^{\overline{n}_{i+1}} < 2^{n_i + 2}$. Отсюда следует $3^{k_i} < 3 \cdot 2^{n_i - \overline{n}_{i+1}}$, $k_i \log_2 3 - \log_2 3 < n_i - \overline{n}_{i+1}$, $k_i(\log_2 3 - 1) - \log_2 3 < n_i - \overline{n}_{i+1} - k_i$. Неравенство b) доказано.

В случае максимальности блока условие b) теоремы 1 можно заменить на более сильное ограничение на k_i в зависимости от параметров блока. В самом деле, из условия максимальности следует, что $2^{n_i} + 3^{k_i} 2^{\overline{n}_{i+1}} < 2^{n_i + 1}$. Отсюда получаем неравенство (7), $k_i \log_2 3 - k_i < n_i - \overline{n}_{i+1} - k_i$.

Перейдем к свойству d). Если блок не максимальный, то выполняются неравенства $2^{n_i} < 3^{k_i} x_i < 3^{k_i} 2^{\overline{n}_{i+1} + 1}$. Отсюда следует необходимое неравенство d) $n_i - \overline{n}_{i+1} - k_i < k_i(\log_2 3 - 1) + 1$.

Обозначим $\Delta_i = n_i - \overline{n}_{i+1} - k_i$. Величину Δ_i называем весом блока.

Следствие 1. Выполняется неравенство $\Delta_i \ge 0$, $i = \overline{1, t}$.

Доказательство. Рассмотрим левую часть неравенства (6). При $k_i = 1$ выражение $k_i(\log_2 3 - 1) - \log_2 3$ принимает минимальное значение, равное -1. Учитывая, что Δ_i — целое число и неравенство (6) строгое, получаем необходимое неравенство $\Delta_i \ge 0$.

Следствие 2. Если i-й блок максимальный, то $\Delta_i > 0$.

Доказательство. Минимальное значение левой части неравенства (7) равно $\log_2 3 - 1 \approx 0{,}584$. Поэтому для максимальных блоков $\Delta_i > 0$.

Следующая теорема устанавливает свойства блоков нулевого веса.

Теорема 2. Если i-й блок $2^{n_t} + 3^{k_i} 2^{n_{i+1}}$ разложения (4) имеет нулевой вес, $\Delta_i = 0$, то выполняются следующие условия:

a) $n_i = \overline{n}_i - 1$;

b) либо $k_i = 1$, либо $k_i = 2$.

Доказательство. Свойство a) непосредственно вытекает из следствия 2.

Рассмотрим левую часть неравенства (6). При $k_i > 2$ соответствующее выражение $k_i(\log_2 3 - 1) - \log_2 3 \ge 2\log_2 3 - 3 \approx 0{,}168 > 0$. Поэтому если $\Delta_i = 0$, то либо $k_i = 1$, либо $k_i = 2$. Это доказывает свойство b).

Для дальнейшего нам понадобятся некоторые свойства блоков веса 1. Максимальный блок веса 1 назовем критическим.

Теорема 3. Критический блок имеет структуру вида $2^{\overline{n}_i} + 3 \cdot 2^{\overline{n}_i - 2}$ либо $2^{\overline{n}_i} + 3 \cdot 2^{\overline{n}_i - 3}$, $\overline{n}_{i+1} = \overline{n}_i - 2$.

Доказательство. Предположим, что i-й блок разложения (4) $2^{n_i} + 3^{k_i} \cdot 2^{n_{i+1}}$ критический. Тогда $n_i = \overline{n}_i$ и выполняется неравенство $2^{n_i} + 3^{k_i} \cdot 2^{\overline{n}_{i+1}} < 2^{n_i + 1}$. Соответствующее неравенство (7) для степеней и веса блока задает ограничение на k_i, $k_i(\log_2 3 - 1) < n_i - \overline{n}_{i+1} - k_i$. Получаем $1 \le k_i < 1/(\log_2 3 - 1) < 2$.

Следовательно, $k_i = 1$ и $\overline{n}_{i+1} = \overline{n}_i - 2$. Так как либо $n_{i+1} = \overline{n}_{i+1}$, либо $n_{i+1} = \overline{n}_{i+1} - 1$, то получаем условие теоремы.

Следствие 3. В последовательности блоков разложения (4) два критических блока не могут соседствовать друг с другом.

Доказательство. Предположим два соседних блока: i-й и $(i+1)$-й, в разложении (4) критические. Тогда в силу условия теоремы 3 $\bar{n}_{i+1} = n_i - 2$ и соответствующий фрагмент частичной суммы x_i в представлении (4) имеет вид

$$2^{n_i} + 3 \cdot 2^{n_i-2} + 9 \cdot 2^{n_i-4} = 37 \cdot 2^{n_i-4} \text{ либо } 2^{n_i} + 3 \cdot 2^{n_i-2} + 9 \cdot 2^{n_i-5} = 65 \cdot 2^{n_i-5},$$

но $37 \cdot 2^{n_i-4} > 65 \cdot 2^{n_i-5} > 2^{n_i+1}$. Это противоречит условию максимальности i-го блока.

Назовем удалением между блоками величину $n_i - n_j$, $j < i$. Оценим удаление между соседними критическими блоками.

Теорема 4. Удаление между соседними критическими блоками не менее 4.

Доказательство. Пусть i-й блок критичен, имеет вид $2^{n_i} + 3 \cdot 2^{n_i-2}$ и не является последним критическим блоком. За ним следует блок $2^{n_i-2} + 3^{k_{i+1}} \cdot 2^{n_{i+2}}$. Соответствующий фрагмент частичной суммы числа x_{i-1} равен $2^{n_i} + 3 \cdot 2^{n_i-2} + {} + 3^{k_{i+1}+1} \cdot 2^{n_{i+2}}$. Так как критический блок максимален, то выполняется неравенство $2^{n_i} + 3 \cdot 2^{n_i-2} + 9 \cdot 3^{k_{i+1}-1} \cdot 2^{n_{i+2}} < 2^{n_i+1}$. Получаем $9 \cdot 3^{k_{i+1}-1} < 2^{n_i-n_{i+2}-2}$.

Отсюда следует неравенство $3^{k_{i+1}-1} < 2^{n_i-n_{i+2}-5}$. Учитывая, что $k_{i+1} \geq 1$, получаем $n_i - n_{i+2} > 5$. Следовательно, $n_i - n_{i+2} \geq 6$.

Рассмотрим второй возможный вариант.

Пусть критический блок имеет вид $2^{n_i} + 3 \cdot 2^{n_i-3}$ и не является последним критическим блоком. За ним следует блок $2^{n_i-3} + 3^{k_{i+1}} \cdot 2^{n_{i+2}}$. Выполняются неравенства $2^{n_i} + 3 \cdot 2^{n_i-3} + 9 \cdot 3^{k_{i+1}-1} \cdot 2^{n_{i+2}} < 2^{n_i+1}$, $9 \cdot 3^{k_i-1} < 5 \cdot 2^{n_i-n_{i+2}-3}$. Отсюда следует, что $n_i - n_{i+2} - 3 > 0$ и $n_i - n_{i+2} \geq 4$. Так как показатель степени 2 в ближайшем соседнем критическом блоке не может превышать n_{i+2}, то отсюда следует условие теоремы.

Обозначаем τ_{1k} количество критических блоков.

Следствие 5. Выполняется неравенство $\tau_{1k} < \dfrac{\log_2 x}{4} + 1$.

Доказательство. Критический блок не может соседствовать с критическим блоком. Следовательно, учитывая теорему 4, удаление последнего критического блока от первого не превышает величины $n_1 - (\tau_{1k} - 1)4$. Значит, $n_1 - 4\tau_{1k} + 4 > 0$, $\tau_{1k} < \dfrac{\log_2 x}{4} + 1$.

Можно дать другую оценку для критических блоков. Общее количество блоков равно t. Так как критические блоки не могут находится рядом друг с другом, то $\tau_{1k} \leq \dfrac{t}{2}$. Объединяя результат следствия 4 с этим фактом, получаем оценку.

Следствие 6. Выполняется неравенство $\tau_{1k} < \min\left(\dfrac{\log_2 x}{4} + 1,\ t/2 \right)$.

ПРЕФИКСНОЕ (2,3)-КОДИРОВАНИЕ ЧИСЕЛ

Кодом называем взаимно однозначное отображение натуральных чисел в последовательности в алфавите $\{0, 1\}$.

Код называется префиксным, если код никакого числа не может быть началом (префиксом) никакого другого кода числа. Иными словами, битовый код $f(x)$ явля-

ется префиксным, если в последовательности $f(x_1)f(x_2)$ после просмотра кода $f(x_1)$ можно однозначно определить начало кода $f(x_2)$.

Проблема построения оптимальных префиксных кодов для натуральных чисел хорошо известна и достаточно детально исследована. Эта проблема возникает в задачах оптимального символьного кодирования (коды Хаффмана и их обобщения), в конструкциях деревьев поиска (деревья поиска Бентли–Яо), в проблемах передачи потоковых сообщений разной длины. Впервые асимптотически оптимальный префиксный код для натуральных чисел был предложен В.И. Левенштейном [5]. Аналогичный код, ω-код, немного позднее был построен П. Элиасом [6], который также предложил несколько других классов монотонных кодов с большей скоростью роста. Другие префиксные коды с подобными или слегка улучшенными свойствами были предложены в [7–12].

Во всех этих исследованиях, в основном, эксплуатировалась идея рекурсивного логарифмического спуска — строилась последовательность указателей на длину длины последующей последовательности. Если есть исходная битовая последовательность, задающая двоичное представление числа x, то слева к ней приписывают двоичное представление значения длины этой последовательности. Процесс рекурсивно повторяется, пока не приходим к короткой строке, например длины три бита. Перед этими тремя битами вставляется как префикс последовательность, состоящая из k единиц и заканчивающаяся нулем. Здесь k — количество рекурсивных вызовов префиксного приписывания значений длин. Битовая длина такого кода равна $\log^*(x) = \log_2 x + \log_2 \log_2 x + \log_2 \log_2 \log_2 x + \dots$.

Основным недостатком подобного представления чисел является абсолютная неустойчивость к ошибкам при передаче сообщений. Если ошибка возникает внутри логарифмической башни префиксов, то теряется всякая возможность восстановить исходное число или хотя бы фрагмент, в котором произошла ошибка. Более того, ошибка в передаче одного числа может привести к искажению всего потока передачи данных. Поэтому ведутся поиски других менее оптимальных кодов, но более устойчивых к ошибкам. К таким кодам относятся коды с выделенным сигнализатором конца кода. Из них наиболее известно фибаначчиево представление чисел.

Любое число x можно однозначно представить как сумму некоторых чисел Фибоначчи, $x = \sum_{i=0}^{m} d_i \cdot F_i$, $d_i \in \{0, 1\}$, $d_m = 1$, F_i — i-е число Фибоначчи. Бинарная последовательность $(d_m d_{m-1} \dots d_0)_F$ задает представление Фибоначчи числа x. Особенностью такого задания является невозможность расположения двух единиц в смежных позициях. Поэтому представление x в виде $d_0 d_1 \dots d_m 1$ с дополнительной единицей, приписываемой к старшему разряду, задает префиксное кодирование чисел. Наличие двух рядом находящихся единиц сигнализирует о конце кода. Длина такого кода примерно в полтора раза превышает $\log_2 x$. Существенное улучшение характеристик фибоначчиева кодирования получено в [13], где использовались обобщенные числа Фибоначчи.

(2,3)-разложение (4) определяет новое префиксное кодирование чисел, обладающее повышенной устойчивостью к искажениям при потоковой передаче информации.

Обозначим последовательность из n нулей через 0^n, а n единиц — 1^n. Если $n = 0$, то условимся считать $0^0 = 0$. Длину битовой последовательности x обозначим $|x|$. Пусть число x не делится на 2 и на 3. Рассмотрим для x разложение (4).

Код $\mathbf{C}^+$. Зададим следующее кодирование чисел с помощью символов 0 и 1. Кодирование чисел выполняется поблочно. Кодом блока $2^{n_i} + 3^{k_i} \cdot 2^{n_{i+1}}$ является последовательность $0^{\Delta_i + 1} 1^{k_i}$, где Δ_i — вес блока, $\Delta_i = n_i - \overline{n}_{i+1} - k_i$.

Определим $C^+(x) = c_1 c_2 \ldots c_t \#$, c_i — код i-го блока $i = \overline{1, t}$, # — специальная константная последовательность символов.

Особая последовательность битов # задает сигнализатор конца слова. Ключевое неравенство (6) накладывает ограничения на смежные последовательности нулей и единиц в $C^+(x)$. Поэтому если задать последовательность из 0 и 1, при которой нарушается соотношение (6), то эта последовательность может интерпретироваться как символ конца слова. Например, не может быть кода блока вида 0111. В противном случае $k_i = 3$. Вес такого блока 0. Это противоречит условию теоремы 2.

Положим # = 0111. Восстановление блока по его коду происходит следующим образом. Если известны значения $\Delta_i = n_i - \overline{n}_{i+1} - k_i$, k_i, остаточное число x_i и, следовательно, $\overline{n}_{i+1}$, то можно восстановить значение n_i, $n_i = \overline{n}_{i+1} + k_i + \Delta_i$, и затем вычислить $\overline{n}_i$ как максимальную степень числа двойки в двоичном представлении числа $x_{i-1} = 2^{n_i} + 3^{k_i} x_i$.

Восстановление числа x по его коду $C^+(x)$ проводится последовательным просмотром кода справа налево, начиная от # , выделением блоков и восстановлением соответствующих параметров блока. Значение веса блока определяется как количество нулей между двумя единицами, ограничивающими эту последовательность, минус 1. Восстанавливаются остаточные числа в порядке $x_t = 1, x_{t-1}, x_{t-2}, \ldots, x_0 = x$.

Очевидно, код $C^+(x)$ является префиксным. Длина кода $C^+(x)$ равна сумме длин кодов блоков плюс длина последовательности # , равная 4:

$$|C^+(x)| = \sum_{i=1}^{t} \Delta_i + \sum_{i=1}^{t} k_i + \sum_{i=1}^{t} 1 + 4 = \sum_{i=1}^{t} (n_i - \overline{n}_{i+1} - k_i) + \sum_{i=1}^{t} k_i + t + 4 =$$

$$= \overline{n}_1 + \sum_{i=1}^{t} (n_i - \overline{n}_i) + t + 4.$$

Пусть t_M — количество максимальных блоков в разложении (4).

Учитывая, что $\overline{n}_1 = \lfloor \log_2 x \rfloor$ и что $n_i - \overline{n}_i = -1$ для немаксимальных блоков и $n_i - \overline{n}_i = 0$ — для максимальных, получаем точную оценку для длины $C^+(x)$, $|C^+(x)| = \lfloor \log_2 x \rfloor + t_M + 4$.

Значащей длиной кода $C^+(x)$ назовем длину кода без учета константной величины, равной длине последовательности #, т.е. значащая длина равна $|C^+(x)|$–4. Ясно, что значащая длина асимптотически совпадает с фактической длиной кода.

Из разложения (4) следует $3^{k_1 + k_2 + \ldots k_t} < x$ и, значит, $k_1 + k_2 + \ldots k_t < \log_3 x$. Величина t зависит от степеней числа 3 в представлении (4). Чем больше степени 3 в (4), тем меньше количество t.

Легко показать, что существуют числа, не содержащие максимальных блоков. Конструирование таких чисел выполняется индуктивно с помощью следующих построений.

Пусть m — произвольное натуральное число и x — нечетное, не делящееся на три, $2^n < 3^m x < 2^{n+1}$. Рассмотрим число $y = 2^n + 3^m x$. Тогда (2, 3)-разложение y имеет вид $2^n + 3^m$ ((2, 3)-*разложение числа x*). Причем первый блок (2, 3)-разложения числа y немаксимальный. Это вытекает из следующих фактов.

Выполняются неравенства: $2^{n+1} < 2^n + 3^m x < 3 \cdot 2^n < 2^{n+2}$. Так как $y \equiv 2^n \bmod 3$ и $y > 2^{n+1}$, то первый блок (2, 3)-разложения числа y немаксимальный.

Таким образом, если число x в своем (2, 3)-разложении не имеет максимальных блоков, то и y не имеет их. Более того если x не имеет нулевых блоков, то за счет варьирования степени m можно добиться, что и y не будет иметь нулевых блоков.

Итак, начиная от числа 3^{m_t}, последовательно применяя указанную выше конструкцию, можно получить возрастающую цепочку блоков, каждый из которых немаксимален.

Пример 2. Для числа 10507 (2,3)-разложение имеет вид $2^{12} + 3(2^{10} + 3(2^7 + 3^5))$. Остаточные числа равны соответственно $x_0 = 10507, x_1 = 2137, x_2 = 371$. Максимальные степени в двоичном разложении остаточных чисел равны соответственно 13, 11, 8. Все блоки немаксимальные. Веса блоков: $\Delta_1 = 0, \Delta_2 = 1, \Delta_3 = 2$. Значение кода $C^+(10507) = 01001000111110111$. Значащая длина кода $C^+(10507)$ равна 13.

Для числа x, не имеющего в (2,3)-разложении максимальных блоков, длина кода $C^+(x)$ равна $\lfloor \log_2 x \rfloor + 4$.

Очевидно, что также не представляет труда построить числа, (2, 3)-разложение которых состоит только из максимальных блоков. В этом случае $t = t_M$.

Отсюда следует, что значащая длина кода $C^+(x)$ может колебаться в пределах от $\log_2 x$ до $\log_2 x + \log_3 x$.

Интересной особенностью кода $C^+(x)$ является выполнение числовых соотношений между соседними вхождениями последовательностей нулей и единиц — выполнение неравенства (6) для немаксимальных блоков и неравенства (7) для максимальных. Эти неравенства можно рассматривать как контролирующие соотношения, связывающие любые соседние вхождения подпоследовательностей 0 и 1. Если неравенство (6) не выполняется и подпоследовательность не совпадает с #, то, значит, в окрестности этого места произошла ошибка при передаче кода блока. Это существенно увеличивает помехоустойчивость (2, 3)-кодов и делает их более гибкими относительно обнаружения ошибок по сравнению с кодированием Фибоначчи.

Возможен другой модифицированный вариант префиксного (2, 3)-кодирования, связанный с введением дополнительных указателей, вместо увеличения на единицу количества нулей, кодирующих веса блоков. В этом варианте кодирования на некоторых числах возможно уменьшение длины кода, в отдельных случаях она может стать даже меньше длины двоичного представления исходного числа.

Код С. Кодом блока $2^{n_i} + 3^{k_i} \cdot 2^{n_{i+1}}$ является последовательность $0^{\Delta_i} 1^{k_i}$, где Δ_i — вес блока, $\Delta_i = n_i - \overline{n}_{i+1} - k_i$.

Определим $C(x) = c_1 \ldots c_t \# \beta$, где c_i — код i-го блока, # = 011111.

Битовая последовательность # задает сигнализатор конца слова. Не может быть кода блока вида 011111. В противном случае $k_i = 5$. Вес такого блока либо 0, либо 1. Неравенство (6) дает соотношение $4\log_2 3 - 5 < 1$, но $\log_2 3 > 1{,}5$, поэтому $4\log_2 3 > 6$. Получаем противоречие.

Слово β — дополнительная интерпретирующая последовательность из 0 и 1.

Последовательность символов в β служит для различения ситуаций, когда $\Delta_i = 0$ либо $\Delta_i = 1$, так как в обоих случаях Δ_i кодируется одним нулем.

Назовем i-й блок двойственным, если либо $\Delta_i = 0$, либо $\Delta_i = 1$ и при этом $k_i = 1$ или $k_i = 2$. Только двойственные блоки нуждаются в дополнительной интерпретации.

Для удобства рассмотрения последний условный $(t+1)$-й блок, состоящий из одной единицы, считаем максимальным.

Если i-й блок двойственный, то полагаем $\varepsilon_i = 0$, если $\Delta_i = 0$, и $\varepsilon_i = 1$, если $\Delta_i = 1$.

Пусть $\gamma = \varepsilon_1 \varepsilon_2 \ldots \varepsilon_r$ — битовая сигнальная последовательность для всех двойственных блоков, встречающихся в (5) в порядке прямого просмотра слева направо.

Определим $\beta = \gamma^{\mathrm{T}}$, где γ^{T} — зеркальное обращение последовательности γ.

Восстановление числа x по его коду $C(x)$ проводится последовательным просмотром символов кода, начиная от # справа налево, выделением блоков и восстановлением соответствующих параметров блока, при необходимости обращаясь к последовательности символов указателей β. При этом восстанавливаются остаточные числа в порядке $x_t, x_{t-1}, \ldots, x_0 = x$.

Выделение кода блока происходит выделением последовательностей $0^{\Delta_i} 1^{k_i}$ в контекстном окружении: слева 1, справа 0. Восстановление блока по его коду проводится аналогично рассмотренному выше случаю кода $C^+(x)$. При обнаружении двойственного блока просматривается текущий символ интерпретирующей последовательности β. Определяется правильное значение Δ_i и маркер просмотра последовательности β сдвигается на одну позицию вправо. Просмотр β происходит слева направо.

Начало просмотра $C(x)$ определяется положением маркера просмотра кодов блоков на последнем символе перед #, маркером просмотра последовательности β на первом символе после # и значением остаточного числа, равным 1. Конец β определяется при переходе к началу кода $C(x)$.

Длину кода $C(x)$ без учета длины константы # будем называть значащей длиной.

Пример 3. Пусть $x = 1427$. (2-3)-разложение числа 1427 имеет следующий вид: $1427 = 2^9 + 3(2^7 + 3(2^5 + 3^3))$.

Остаточные числа равны: $x_0 = 1427, x_1 = 305, x_2 = 59, x_3 = 1$. Соответственно значения $\lfloor \log_2 x_i \rfloor$ равны: $\bar{n}_1 = 10, \bar{n}_2 = 8, \bar{n}_3 = 5, \bar{n}_4 = 0$.

Имеем три блока: $2^9 + 3^1 2^7, 2^7 + 3^1 2^5, 2^5 + 3^3$. Веса блоков равны: $\Delta_1 = 0$, $\Delta_2 = 1, \Delta_3 = 2$. Коды блоков равны соответственно: 01, 01, 00111. В интерпретации нуждаются первые два блока. $C(1427) = 01\ 01\ 00111\ 011111\ 10$, $\lfloor \log_2 1427 \rfloor = 10$. Значащая длина кода $C(1427)$ равна 11.

Пример 4. (2,3)-представление числа 157741 имеет следующий вид:

$$157741 = 2^{16} + 3^3 (2^{10} + 3(2^9 + 3(2^5 + 3^2 (2^2 + 3)))).$$

В данном представлении пять блоков:

$$2^{16} + 3^3 2^{10},\ \ 2^{10} + 3^1 2^9,\ \ 2^9 + 3^1 2^5,\ \ 2^5 + 3^2 2^2,\ \ 2^2 + 3.$$

Соответствующие остаточные числа равны: $x_0 = 157741$, $x_1 = 3415$, $x_2 = 797$, $x_3 = 95, x_4 = 7, x_5 = 1$. Им соответствуют максимальные степени 2, входящие в двоичные представления чисел x_i, $i = \overline{0,4}$: $\bar{n}_1 = 17, \bar{n}_2 = 11, \bar{n}_3 = 9, \bar{n}_4 = 6, \bar{n}_5 = 2$. Веса блоков соответственно равны: $16-11-3=2$, $10-9-1=0$, $9-6-1=2$, $5-2-2=1$, $2-1=1$.

Следовательно, коды блоков имеют вид 00111, 01, 001, 011, 01. Вес второго блока нулевой, а веса, четвертого и пятого равны 1. Эти блоки двойственные и нуждаются в интерпретации. Интерпретирующая последовательность β имеет вид $\beta = 110$. Разделительный символ # = 011111, поэтому код равен $C(157741) = 00111\ 01\ 001\ 011\ 01\ 011111\ 110$.

Двоичная запись числа 157741 занимает 18 битов, значащая длинна (2,3)-кода для этого числа равна 18.

Теорема 5. Код $C(x)$ является префиксным.

Доказательство. Рассмотрим код $C(x)=c_1c_2\ldots c_t\#\beta$. При просмотре последовательности $C(x)$ слева направо однозначно определяется позиция #. Количество дополнительных битов в β определяется обратным просмотром справа налево кода $c_1c_2\ldots c_t$ и количеством вхождений подпоследовательностей …10110 … или … 1010 …, с изолированным символом 0, нуждающимся в интерпретации. Каждому такому вхождению в $c_1c_2\ldots c_t$, согласно построению β, сопоставляется соответствующий интерпретирующий символ 0 или 1 в β. Если двойственных блоков веса 0 или 1 не имеется, то $\beta=\varnothing$ и конец кода $C(x)$ определяется последовательностью #.

Оценим длину кода $C(x)$. Пусть τ_0 — количество блоков нулевого веса, τ_{1k} — количество критических блоков.

Теорема 6. Длина кода $C(x)$ не превышает $\log_2 x+\tau_0+\tau_{1k}+6$.

Доказательство. Рассмотрим код $C(x)=c_1c_2\ldots c_t\#\beta$. Код i-го блока $0^{\Delta_i}1^{k_i}$, $i=\overline{1,t}$, содержит $n_i-\bar{n}_{i+1}-k_i$ нулей и k_i единиц. Дополнительно, за счет кодирования нулевых весов блоков символом 0, введено τ_0 нулей. Следовательно, битовая длина части $c_1c_2\ldots c_t$ равна

$$\tau_0+\sum_{i=1}^{t}(n_i-\bar{n}_{i+1}-k_i)+\sum_{i=1}^{t}k_i=\tau_0+\bar{n}_1+\sum_{i=1}^{t}(n_i-\bar{n}_i). \qquad (8)$$

Длина последовательности β равна сумме количества двойственных блоков веса 0 и веса 1. Учитывая, что блоки веса 0 немаксимальные, то для каждого символа в β, интерпретирующего нулевой блок или немаксимальный блок веса 1, в сумме $\sum_{i=1}^{t}(n_i-\bar{n}_i)$ из (8), найдется соответствующая отрицательная величина $n_i-\bar{n}_i=-1$.

Длина последовательности # равна 6. Поэтому получаем, что суммарная длина кода $C(x)$ не превышает $\bar{n}_1+\tau_0+\tau_{1k}+6$. Так как $\bar{n}_1=\lfloor\log_2 x\rfloor$, то получаем условие теоремы 6.

Следствие 6. Если в разложении (4) отсутствуют блоки веса 0 и критические блоки, то длина кода $C(x)$ не превышает $\lfloor\log_2 x\rfloor+6$.

Условие следствия 6 автоматически выполняются, если, например, все $k_i\ge 3$.

Пусть τ_j — количество блоков веса j. Тогда $\sum_{i=1}^{t}\Delta_i=\sum_{j=1}^{k}j\tau_j$, где k — максимальный вес блока. С другой стороны, из определения веса блока получаем $\sum_{i=1}^{t}\Delta_i=\bar{n}_1+\sum_{i=1}^{t}(n_i-\bar{n}_i)-\sum_{i=1}^{t}k_i=\tau_1+\sum_{j=2}^{k}j\tau_j$. Следовательно, $\tau_0+\tau_1=\tau_0+\bar{n}_1+$

$+\sum_{i=1}^{t}(n_i-\bar{n}_i)-\sum_{i=1}^{t}k_i-\sum_{j=2}^{k}j\tau_j$

Учитывая, что блоки нулевого веса немаксимальные, получаем неравенства $\tau_0+\tau_{1k}\le\tau_0+\tau_1<\log_2 x-\sum_{i=1}^{t}k_i-\sum_{j=2}^{k}j\tau_j$. Поэтому значащая длина кода $C(x)$ не превышает $2\log_2 x-\sum_{i=1}^{t}k_i-\sum_{j=2}^{k}j\tau_j$.

При конструировании кода $C(x)$ (как и $C^+(x)$) предполагалось, что число x нечетное и не делится на 3. Для того чтобы распространить такое кодирование на все

числа, поступаем следующим образом. Вводим расширенную кодировку $\widetilde{C}(x)$. Расширение кодировки $C^+(x)$ выполняется аналогичным образом. Первые два бита в $\widetilde{C}(x)$ сигнализируют о делимости на 2 и 3: $00-x$ нечетное и не делится на 3; $01-x$ нечетное и делится на 3; $10-x$ четное и не делится на 3; $11-x$ четное и делится на 3.

Пусть $x=2^n3^k x_1$, где x_1 — нечетное число, которое не делится на 3. Если $n=0$, $k=0$, то $\widetilde{C}(x)=00C(x_1)$, если $n=0, k\neq 0$, то $\widetilde{C}(x)=011^k C(x_1)$; если $n\neq 0$, $k=0$, то $\widetilde{C}(x)=101^n C(x_1)$, если $n\neq 0$, $k\neq 0$, то $\widetilde{C}(x)=110^n 1^k C(x_1)$.

Для очень больших чисел при конструировании кодов блоков можно добиться эффекта дополнительного сжатия. Вместо 0^{Δ_i} в коде $C(x)$ (аналогично и в $C^+(x)$) при больших k_i предлагается рассматривать код $0^{\Delta_i-\lfloor ck_i\rfloor}$, где c — некоторая константа, не превышающая 0,5. Более точно: рассмотрим ключевое неравенство (6) $k_i(\log_2 3-1)-\log_2 3<\Delta_i$.

Пусть c — некоторая константа. Из (6) следуют неравенства

$$k_i(\log_2 3-1-c)-\log_2 3<k_i(\log_2 3-1)-\lfloor ck_1\rfloor-\log_2 3<\Delta_i-\lfloor ck_i\rfloor,$$

которые имеют смысл при условии $k_i(\log_2 3-1-c)-\log_2 3>0$. Это неравенство будет выполняться при условиях

$$0<c<\log_2 3-1,\ k_i>\frac{\log_2 3}{\log_2 3-1-c}. \tag{9}$$

Таким образом, в качестве константы c можно выбрать любое j-е приближение к числу $\log_2 3-1$. В частности, рассмотрим первое приближение $c=0{,}5$. Условие (9) выполняется при $k_i\geq 19$. В коде блока вместо Δ_i, при $k_i\geq 19$, можно рассматривать величину $\Delta_i=\Delta_i-\lfloor 0{,}5k_i\rfloor$. Соответствующий код при $k_i\geq 19$ принимает вид $0^{\Delta_i-\lfloor 0{,}5k_i\rfloor}1^{k_i}$. Происходит экономия на $0{,}5k_i$ битов. При восстановлении числа по его коду значение n_i определяется по формуле $n_i=\overline{n}_{i+1}+k_i+\lfloor 0{,}5k_i\rfloor+\Delta_i$.

Следует отметить, что во многих случаях (например, если в (2,3)-разложении числа x отсутствуют блоки веса 0 и критические блоки, но существуют немаксимальные блоки) кодирование $C(x)$ может иметь длину, меньшую $\log_2 x$.

Для чисел x, у которых в (2,3)-представлении все блоки немаксимальные и отсутствуют двойственные блоки, длина кода $C(x)$ минимальна, $|C(x)|=\lfloor\log_2 x\rfloor-t+6$. Это следует из рассмотрения формулы (8). В этом случае сигнализирующая последовательность β пустая, $\tau_0=0$, $n_i-\overline{n}_i=-1$, $i=\overline{1,t}$.

Пример построения таких чисел проводится рассмотренной выше индуктивной конструкцией построения чисел, не имеющих в (2,3)-представлении максимальных блоков.

Пример 5. (2,3)-разложение числа 753787 имеет следующий вид:

$$753787=2^{18}+3^3(2^{13}+3^3(2^7+3^5)).$$

Все блоки немаксимальные. Веса блоков равны соответственно: $\Delta_1=18-14-3=1$, $\Delta_2=13-8-3=2$, $\Delta_3=7-5=2$, количество блоков равно 3, $C(753787)=0111001110011111\#$, $\lfloor\log_2 753787\rfloor=19$. Значащая длина кода $C(753787)$ равна 16.

Пусть $y = 2^n 3^k x$, где x — нечетное число, не делящееся на 3.

Пусть x имеет (2,3)-разложение (4). Тогда y можно задать позиционной последовательностью пар чисел

$$(y)^1_{2,3} = (n,k)(n_1,k_1)(n_2,k_2)\ldots(n_t,k_t). \tag{10}$$

Если y — нечетное число и не делится на 3, то первая пара чисел имеет вид (0,0).

Из (2,3)-представления (4) путем раскрытия скобок следует другая интересная эквивалентная форма задания чисел.

Любое положительное натуральное число y можно представить в виде суммы произведений некоторых степеней чисел 2 и 3:

$$y = \sum_{i=1}^{t} 2^{n_i} 3^{m_i}. \tag{11}$$

Существует представление (11), при котором выполняются неравенства $n_i > n_{i+1}$, $m_i < m_{i+1}$, $i = \overline{1, t-1}$.

В этой второй форме число y задается последовательностью пар чисел

$$(y)^2_{2,3} = (n_1, m_1)(n_2, m_2)\ldots(n_t, m_t), \quad n_i > n_{i+1}, m_i < m_{i+1}. \tag{12}$$

Выражение вида $2^n 3^m$ по аналогии с битами и тритами можно назвать битритом. Любое задание числа y в форме (11) будем называть битритовым. Битритовое задание чисел может оказаться полезным при выполнении многих арифметических операций. Например, произведение битритовых форм является битритовой формой, а вычисление модульного остатка можно свести к вычислению остатков только от деления стандартных битритов и соответствующего модульного суммирования. Легко выполняется также возведение в битритовую степень.

Произведение битритов сводится к сложению соответствующих степеней:

$$(n_1, m_1) \times (n_2, m_2) = (n_1 + n_2, m_1 + m_2).$$

При суммировании битритов возможно применение некоторых формальных оптимизирующих преобразований, направленных на уменьшение количества битритов в представлении (12):

$$(n,m)(n,m) = (n+1,m);$$
$$(n,m)(n,m+1) = (n+2,m);$$
$$(n,m)(n+1,m) = (n,m+1);$$
$$(n,m)(n+3,m) = (n,m+2).$$

В то же время традиционным является двоичное задание чисел. Поэтому рассмотрим более детально связь между двоичным и битритовым представлениями чисел.

Заметим следующее. Умножение и деление на 3 фактически не сложнее одной операции сложения двоичных чисел. В самом деле, $3a = 2a + a$, поэтому умножение на 3 требует одного сдвига и одного сложения. Более того, умножение на 9 двоичного числа требует всего трех сдвигов и одного сложения, $9a = 8a + a$. Поэтому при умножении на 3^{2k} необходимо выполнить $3k$ сдвигов влево и k сложений.

Пусть $x = (x_n x_{n-1\ldots} x_0)_2$ — двоичное представление числа x, $x_i \in \{0,1\}$, x_n — старший, x_0 — младший разряд. Если x — переменная величина, то считаем, что x_i — также переменная величина, равная значению i-го разряда x.

Остаток от деления числа x на 3, $x \bmod 3$, находится в зависимости от результата поразрядной суммы, $S = \left(\sum_{i=0}^{n} (-1)^i x_i \right) \bmod 3$. Если $S = 1$ или $S = -2$, то $x \bmod 3 = 1$; если $S = -1$ или $S = 2$, то $x \bmod 3 = 2$; если $S = 0$, то $x \bmod 3 = 0$.

Предположим, что x делится на 3, $x = 3y$, $y = x/3$, где y — некоторое число. Двоичное представление y по заданному числу x находится поразрядно, начиная с младших разрядов, $y = (y_{n-1} y_{n-2 \ldots} y_0)_2$. Пусть c_i означает единицу переноса в i-й разряд. Учитывая, что $3y = 2y + y$, значения y_i находятся по следующей рекуррентной схеме:

$$y_0 = x_0;\ c_1 = 0;$$
$$y_{i+1} = y_i \oplus x_{i+1} \oplus c_{i+1};$$
$$c_{i+2} = 1, \text{ если } y_i + y_{i+1} + c_{i+1} > 1;\ c_{i+2} = 0, \text{ если } y_i + y_{i+1} + c_{i+1} < 2.$$

Здесь $\oplus$ означает сложение по $\bmod 2$, $i = \overline{0, n-1}$.

Определим $x \operatorname{div} 3 = (x - x \bmod 3)/3$. Очевидно, что при схемной реализации все рассмотренные выше операции не сложнее реализации одной операции сложения.

Заметим, что можно предложить аналогичные простые схемы для реализации деления на 9. Таким образом, имея двоичное представление числа x, за время, пропорциональное $\log_2 x$, можно получить (2,3)-разложение типа (10) или (11).

Очевидно также, что по форме (10) или (11) легко перейти к двоичному представлению числа x.

Рассмотрим операцию модулярной редукции для битритов. Воспользуемся слегка модифицированной основной идеей известного метода модулярной редукции Монтгомери [14] , которая сводит поиск модуля к целочисленному делению на удобные числа.

Пусть a, b и n — натуральные взаимно простые числа и $ab \bmod n = r$.

Лемма 1. Существует число t такое, что выполняются следующие условия:

1) $0 \le t < b$; 2) $\dfrac{tn + r}{b}$ — целое число; 3) $\dfrac{tn + r}{b} \equiv a \bmod n$.

Доказательство. В последовательности из b чисел $r, n + r, 2n + r, \ldots, (b-1)n + r$ все числа дают разные остатки при делении на b. Поэтому найдется такое число t, что $tn + r \equiv 0 \bmod b$. Очевидно, что $\dfrac{tn + r}{b} \cdot b \equiv r \bmod n$. Следовательно, $\dfrac{tn + r}{b} \equiv a \bmod n$. Отметим, что $\dfrac{tn + r}{b} < 2n$. Поэтому $a \bmod n = \dfrac{tn + r}{b} - \Delta$, где $\Delta = 0$, если $\dfrac{tn + r}{b} < n$ и $\Delta = n$, если $\dfrac{tn + r}{b} > n$.

Пусть n — нечетное число, не делящееся на 3. Предположим, известно двоичное значение вычета $r_{s,k} = 2^s 3^k \bmod n$, $r_{s,k} = (r_v r_{v-1} \ldots r_0)_2$. Тогда за время, равное времени выполнения одного сложения, можно найти значения $r_{s-1,k} = 2^{s-1} 3^k \bmod n$ и $r_{s,k-1} = 2^s 3^{k-1} \bmod n$.

В действительности, применяя лемму 1, ищем остаток $r_{s-1,k}$ в виде $\dfrac{tn + r_{s,k}}{2}$. Отсюда следует, что t — битовая величина, равная младшему разряду $r_{s,k}$, $t = r_0$. Так как $r_{s,k} < n$ и $t \in \{0, 1\}$, то $\dfrac{tn + r_{s,k}}{2} < n$, поэтому $r_{s-1,k} = \dfrac{r_0 n + r_{s,k}}{2}$.

Повторяя описанную выше процедуру i раз, $i < s$, легко находится значение $r_{s-i,k}$. Аналогичным образом вычет $r_{s,k-1}$ ищем в виде $\frac{tn + r_{s,k}}{3}$. Значение t находим из соотношения $tn + r_{s,k} \equiv 0 \bmod 3$,

$$t = (-(n^{-1} \bmod 3) r_{s,k}) \bmod 3.$$

Заметим, что значение t, как и значение $n^{-1} \bmod 3$, равно либо 1, либо 2. Поэтому нахождение $r_{s,k-1}$ не сложнее одного сложения и не представляет труда.

Выполняя j раз подобные действия, вычисляется значение $r_{s,k-j}$.

При необходимости многократного вычисления операций типа $x \bmod n$ при фиксированном модуле n предлагается накапливать значения остатков $r_{s,k}$ в табличном виде, $A[s,k] = r_{s,k}$. Перед началом вычислений следует определить некоторые стартовые значения матрицы A. В качестве таких значений можно выбрать $2^{2^i} \bmod n$ и $3^{2^j} \bmod n$, затем их попарно модульно перемножить и занести в матрицу A. Максимальные значения i и j определяются из условий вычислений. В процессах, где выполняются серии умножений, например, при модульном возведении в степень в качестве максимальных i и j следует выбрать минимальные значения, при которых $2^{2^i} > n^2$ и $3^{2^j} > n^2$. Это дает $i \approx \log_2 \log_2 n$ и $j \approx \log_2 \log_2 n$. Вычисления стартовых модульных значений степеней выполняется за $\log_2 \log_2 n$ возведений в квадрат и соответствующего вычисления модуля методом Барета или классическим методом [15]. Как видим, объем предвычислений невелик и пропорционален $(\log\log n)^2$. После заполнения стартовых значений матрицы A входные значения выполняемых процедур переводятся в битритовую форму, над ними выполняются соответствующие арифметические операции (сложения, умножения, возведения в квадрат) и вычисляются модульные значения битритов путем обращения к таблице A. Если необходимо вычислить $2^s 3^k \bmod n$, но в таблице A отсутствует соответствующее значение остатка $r_{s,k}$, то в таблице A ищется ближайшие значения s' и k' такие, что $2^{s'} 3^{k'} \bmod n$ определено и $s' > s$, $k' > k$. Затем вышеописанным способом модульного деления вычисляется значение $2^s 3^k \bmod n$. Попутно заполняется таблица A. При выполнении достаточного количества вычислений происходит ускорение вычислений за счет обращения к заполненной справочной таблице.

Принципиально полное заполнение таблицы A можно вынести в предвычисления. В этом случае матрица A будет содержать количество элементов, пропорциональное $\log_2^2 n$.

В случае многоразрядной арифметики большие числа задаются в системе счисления по основанию 2^M. Обычно на практике выбирают $M = 32$ или $M = 64$. Модулярная арифметика таких чисел хорошо изучена и реализована во всех современных криптографических системах с открытыми ключами.

Вычисление остатка от деления на 3 для больших чисел очевидно. Рассмотрим операцию деления на 3 для больших чисел.

Пусть x задано в двоичной системе счисления, $x = (x_n x_{n-1} \ldots x_0)_2$, c и z — битовые переменные. Определим процедуру $\mathrm{div}3(x, c, z)$.

Procedure $\mathrm{div}3(x, c, z)$

INPUT: $x = (x_n x_{n-1\ldots} x_0)_2$, c, z.

$c \in \{0, 1\}$, $z \in \{0, 1\}$.

OUTPUT: $y = (y_n y_{n-1} \ldots y_0)_2$, c, z.

1. $y_0 \leftarrow x_0 \oplus c \oplus z$.
2. For i from 1 to n do

   $y_i \leftarrow x_i \oplus y_{i-1} \oplus c$.

   If $y_i + y_{i-1} + c > 1$ then $c \leftarrow 1$ else $c \leftarrow 0$
3. $z \leftarrow y_n$.
4. Return $(y = (y_n y_{n-1\ldots} y_0)_2, c, z)$.

Если $x = 3y$, то $\mathrm{div}3(x,0,0)$ выдаст в качестве результата $(y, 0, 0)$.

Пусть x — большое число, заданное в системе счисления по основанию $2M$. $x = x_k 2^{kM} + x_{k-1} 2^{(k-1)M} + \ldots + x_0$, $0 \le x_i < 2^M$.

Числа x_i заданы в двоичном виде, $x_i = (x_{i,(M-1)} x_{i,(M-2)} \ldots x_{i,0})_2$, $i = \overline{0, k}$.

Рассмотрим следующий алгоритм.

Algorithm $M\,\mathrm{div}\,3$

INPUT: $x = (x_k x_{k-1} \ldots x_0)_{2^M}$, c, z.

$c \in \{0, 1\}$, $z \in \{0, 1\}$.

OUTPUT: $y = (y_k y_{k-1} \ldots y_0)_{2^M}$.

1. $c \leftarrow 0$; $z \leftarrow 0$.
2. For i from 0 to k do

   $(y_i, c, z) \leftarrow \mathrm{div}3(x_i, c, z)$.
3. Return $y = (y_k y_{k-1} \ldots y_0)_{2^M}$.

Если $x = 3y$, то $M\mathrm{div}\,3$ при входе $(x, c = 0, z = 0)$ выдает в качестве результата y.

Как видим, деление на 3 не превышает сложности реализации одной операции сложения для многоразрядных чисел.

Рассмотрим базисные ортогональные последовательности $U = \{2^M, 2^{2M}, 2^{3M}, \ldots\}$ и $V = \{3, 3, 3, \ldots\}$. Пусть x — нечетное, не делящееся на 3 число, $(2^M, 3)$-представление в битритовой форме для числа x имеет вид

$$x = \sum_{i=1}^{t-1} 2^{s_i M + \varepsilon_i} 3^{m_i} + 3^{m_t} x_{t+1},$$

где $\varepsilon_i \in \{0, 1\}$, x_{t+1} — остаточное число, которое не имеет положительно определенного $(2^M, 3)$-представления. Заметим, что $x_{t+1} < 3 \cdot 2^M$.

При необходимости многократного выполнения серий умножений предлагается аналогично рассмотренному выше случаю формировать вспомогательную матрицу значений вычетов $r_{s,m} = 2^{sM} 3^m \bmod n$.

При формировании такой таблицы следует использовать лемму 1, т.е. искать вычет $r_{s-1,k}$ по формуле $r_{s-1,k} \equiv \dfrac{tn + r_{s,k}}{2^M}$, где $t = -(n^{-1} \bmod 2^M) r_{s,k} \bmod 2^M$.

ЗАКЛЮЧЕНИЕ

Другим интересным источником битритовых форм служит (3,2)-представление чисел, получаемое аналогично вышерассмотренному случаю (2,3)-разложения чисел. В (3,2)-разложении участвуют базисные последовательности $U = \{3, 3^2, 3^3, \ldots\}$ и $V = \{2, 2, 2, \ldots\}$.

Используя (3, 2)-представления чисел, также можно построить универсальное префиксное кодирование чисел с повышенной устойчивостью к локальным искажениям при передаче информации.

В настоящий момент (2, 3)-кодирование C является единственным известным универсальным префиксным кодированием чисел, которое не всегда увеличивает длину кода по сравнению с двоичным заданием чисел.

(2,3)-представление чисел раскрывает их двумерную геометрическую структуру, что может найти применения в стеганографии и криптографии.

СПИСОК ЛИТЕРАТУРЫ

1. Анисимов А. В. Кодирование данных линейными формами числовых последовательностей // Кибернетика и системный анализ. — 2003. — № 1. — С. 3–15.
2. Анисимов А. В., Редько В. Е., Рындин Я. П. Обратное преобразование Фибоначчи // Там же. — 1995. — № 3. — С. 106–115.
3. Анисимов А. В. Линейные формы Фибоначчи и параллельные алгоритмы большой размерности // Там же. — 1995. — № 3. — С. 106–115.
4. Anisimov A. V. Linear Fibonacci forms and parallel algorithms for high dimension arithmetic // Lecture Note in Comput Sci. — N.Y.: Springer Verlag, 1995. — P. 64–69.
5. Левенштейн В.И. Избыточность и задержка восстановительного кодирования натуральных чисел // Проблемы кибернетики. — 1968. — № 20. — С. 173–179.
6. Elias P. Universal codeword sets and representations of the integers // IEEE Trans. on Inform. Theory. — 1976. — **21(2)**. — P. 194–203.
7. Bentley J. L., Yao A. C. An almost optimal algorithm for unbounded searching // Inform. Process. Letters. — 1976. — **5(3)**. — P. 82–87.
8. Even S., Rodch M. Economical encoding of comma between strings // Com. of the ACM. — 1978. — **21(4)**. — P. 315–317.
9. Stout Q. F. Improved prefix encodings of the natural numbers // IEEE Trans. on Inform. Theory. — 1980. — **26.** — P. 607–609.
10. Yamamoto H. A new recursive universal code of the positive integers // Ibid. — 2000. — **46** . — P. 717–723.
11. Ahlswede R. F., Han T. S., Kobayashi K. Universal coding of integers and unbounded search trees // Ibid. — 1997. — **43**. — P. 669–682.
12. Anisimov A., Reznik Y. A. Construction of prefix codes using linear forms of numerical sequences // Proc. 26th Symposium on Inform. Theory in the Benelux. — Brussels; Belgium, May 19–20, 2005.
13. Apostolico A., Frankel A. S. Robust Transmission of Unbounded Strings Using Fibonacci Representations // IEEE Trans. on Inform. Theory. — 1987. — **IT-33**, N 2. — P. 238–245.
14. Montgomery P.L. Modular multiplication without trial division // Mathematics of Computations. — 1985. — **44**, N 170. — P. 529–521.
15 Кнут Д. Э. Искусство программирования. Т. 2. — М.: Киев: «Вильямс», 2000. — 832 с.

Поступила 02.03.2009