УДК 519.72

А.В. АНИСИМОВ

ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В ДВУХБАЗИСНЫХ СИСТЕМАХ СЧИСЛЕНИЯ

Ключевые слова: *системы счисления, кодирование чисел, деревья, линейные формы, рекурсия, префиксные коды.*

ВВЕДЕНИЕ

Форма представления чисел играет важную роль в компьютерной математике и во многом определяет эффективность алгоритмов обработки информации. Наиболее известны традиционные представления чисел в виде разложения по степеням заданного базисного числа. Архитектура современных вычислительных устройств в основном базируется на двоичной арифметике. Также представляет интерес тритовая система счисления, использующая в качестве базиса число 3. В настоящее время для задач сжатия информации активно изучается представление чисел в виде сумм чисел Фибоначчи [1–4]. В данной статье развивается подход представления чисел в двухбазисных системах. Стандартные классические степенные, а также многие другие известные системы счисления являются частным случаем такого способа задания чисел.

Представим двухбазисное представление чисел с использованием простых исходных компонентов: двух бесконечных последовательностей чисел (базисы) и линейных двухаргументных форм от чисел в этих последовательностях. Рекурсивная декомпозиция произвольного числа в линейные бинарные формы в выбранных базисах определяет вид задания числа.

Вычислительной мотивацией двухбазисного представления чисел служит поиск базисных последовательностей, для элементов которых определенные вычисления выполняются эффективно. Таким образом, вычисления для произвольных аргументов в некоторых удачно выбранных базисах могут быть сведены к более простым вычислениям.

Очевидны применения двухбазисных систем в криптографии. В этом случае базисные последовательности генерируются по выбранным секретным параметрам. Число кодируется индексами ключевых последовательностей, возникающих при рекурсивной декомпозиции. Также возможны применения двухбазисных систем при построении самосинхронизирующихся префиксных кодов. При соответствующем выборе базисных последовательностей возможно конструирование вычислительных архитектур, которые реализуют двухбазисную арифметику.

ДВУХБАЗИСНЫЕ СИСТЕМЫ СЧИСЛЕНИЯ

Все рассматриваемые в настоящей статье числа принадлежат натуральному ряду. Далее в тексте это предполагается по умолчанию.

Пусть $U = u_1, u_2, \ldots$ и $V = v_1, v_2, \ldots$ — две бесконечные последовательности чисел. Назовем их ортогональными, если НОД $(u_i, v_i) = 1$ для всех $i = 1, 2, \ldots$ В этом случае используем обозначение $u_i \perp v_i$ и $U \perp V$.

В некоторых случаях целесообразно добавлять в U и V начальную пару u_0 и соответственно v_0 не обязательно взаимно простых чисел. Более того, допускается случай, когда $u_0 = 0$. Обозначим $\mathbb{N}$ множество натуральных чисел с исключенным нулем. Пусть $x \in \mathbb{N}$.

Представление вида

$$x = \lambda_1 u_n + \lambda_2 v_n, \tag{1}$$

где $u_n \in U$, $v_n \in V$, назовем линейной (U,V)-формой, а индекс n — рангом.

Из алгоритма Евклида нахождения НОД двух чисел следует, что если $U \perp V$, то для любого x существует представление вида (1). Следует отметить случай, когда λ_1 и λ_2 — неотрицательные числа. Представление (1) назовем положительно определенным, если $\lambda_1 > 0$ и $\lambda_2 \geq 0$. Здесь рассматриваются только положительно определенные формы максимального ранга *n*. Поэтому в дальнейшем по умолчанию предполагаем, что все рассматриваемые линейные формы положительно определенные и имеют максимальный ранг.

Среди всех возможных представлений (1) фиксированного ранга *n* выделим форму, где коэффициент λ_1 максимальный. Такое представление назовем каноническим. Используя тождественное преобразование $x = (\lambda_1 + kv_n)u_n + (\lambda_2 - ku_n)v_n$, представление (1) всегда можно привести к каноническому виду. Поэтому в максимальном каноническом (U,V)-линейном представлении $\lambda_2 \leq u_n$. Очевидно, каноническое (U,V)-линейное представление максимального ранга единственное. Отметим, что если для произведения чисел *u* и *v* выполняется неравенство $uv \leq x$, то для x всегда существует положительно-определенное представление вида $x = \lambda_1 u + \lambda_2 v$.

Действительно, пусть $x = au - bv$, где $a > 0$, $b > 0$. Тогда $x = (a - kv)u + (ku - b)v$. Выберем k такое, что $ku > b$ и $(k-1)u < b$. В этом случае выполняется неравенство $0 < ku - b < u$. При таком выборе k должно выполняться неравенство $a - kv > 0$. В противном случае выполнялось бы неравенство $x < uv$, что противоречит исходному предположению.

Заметим также, что для заданных *u* и *v* описанное выше преобразование позволяет получить для x положительно определенную форму или убедиться, что ее не существует. Из этого следует, что если (1) задает каноническое представление максимального ранга для x, то $x < u_{n+1}v_{n+1}$.

Предположим, что задана каноническая форма (1) максимального ранга. В зависимости от решаемых задач можно рекурсивно применять к обоим коэффициентам λ_1 и λ_2 или только к одному из них аналогичный процесс разложения в канонические (U,V)-линейные формы максимального ранга до тех пор, пока такое разложение возможно.

Обозначим S подмножество натуральных чисел такое, что для каждого числа x из S существует по меньшей мере одна линейная форма (1). Описанная рекурсивная процедура разложения коэффициентов в соответствующие линейные формы задает единственное представление чисел из S. Таким образом, для S пара (U,V) может рассматриваться как двухбазисная система счисления.

Если разложение в максимальные линейные формы применяется только к мультипликативным коэффициентам элементов из V, то рассматриваем последовательность U как основной базис, а V — как вспомогательный. В этом случае процесс декомпозиции описывается последовательностью остаточных чисел:

$$x_0 = x = \lambda_1 u_{n_1} + x_1 v_{n_1},\ x_1 = \lambda_2 u_{n_2} + x_2 v_{n_2}, \ldots, x_{t-1} = \lambda_t u_{n_t} + x_{t+1} v_{n_t}. \qquad (2)$$

В (2) каждое остаточное число $x_i = \lambda_{i+1} u_{n_{i+1}} + x_{i+1} v_{n_{i+1}}$ задается (U,V)-линейной формой максимального ранга, $i = 0, 1, 2, \ldots, t-1$, $u_{n_{i+1}} \in U$, $v_{n_{i+1}} \in V$. Число x_{t+1} либо равно единице, либо оно не представимо в виде (1).

Последовательность (2) можно переписать в следующем структурном виде:

$$x = \lambda_1 u_{n_1} + v_{n_1}(\lambda_2 u_{n_2} + v_{n_2}(\ldots(\lambda_t u_{n_t} + x_{t+1} v_{n_t})\ldots)).$$

Развивая теорию двухбазисных числовых систем, необходимо решать следующие задачи.

1. Как эффективно генерировать базисы U и V для решения заданного класса задач?

2. Если заданы базисы U и V, то какова эффективность алгоритма разложения числа в максимальную (U,V)-линейную форму?

3. Каковы соотношения между параметрами максимальной линейной формы?

4. Какие задачи информационных технологий можно решать с помощью двухбазисных систем счисления?

При разложении в линейные формы (1) важное значение имеет оценка коэффициентов λ_1 и λ_2, поскольку это определяет многие свойства рекурсивного разложения.

Пусть $U = u_1, u_2, \ldots$ и $v_n = v_1, v_2, \ldots$ — две ортогональные последовательности и число x задано разложением (1).

Теорема 1. Предположим, что $v_n > u_n$, $n = 1, 2, \ldots$, и существует предел $\lim\limits_{n\to\infty} \dfrac{v_{n+1}}{u_n}$. Тогда существует константа c такая, что $\lambda_1 + \lambda_2 < c\sqrt{x}$.

Доказательство. Пусть $x = au_n + bv_n$. Исходя из того, что (1) — линейная форма максимального ранга, следует $x < u_{n+1}v_{n+1}$. Из условия теоремы получаем неравенства

$$x < v_{n+1}^2, \quad \sqrt{x} < v_{n+1}, \quad \frac{x}{v_{n+1}} < \sqrt{x}.$$

Это позволяет оценить сумму $\lambda_1 + \lambda_2$ следующим образом:

$$u_n(\lambda_1 + \lambda_2) < x, \quad \lambda_1 + \lambda_2 < \frac{x}{u_n} = \frac{x}{v_{n+1}} \cdot \frac{v_{n+1}}{u_n}.$$

Исходя из наличия предела для $\dfrac{v_{n+1}}{u_n}$ следует существование константы c, ограничивающей отношение $\dfrac{v_{n+1}}{u_n}$. Отсюда имеем $\lambda_1 + \lambda_2 < c\sqrt{x}$.

Теорема доказана.

Последовательное рекурсивное разложение числа x в максимальные (U,V)-линейные формы по коэффициентам при U и V определяет иерархическую структуру — линейное дерево числа x. На рис. 1 $T\lambda_1$ и $T\lambda_2$ — линейные деревья чисел λ_1 и λ_2. Листья дерева соответствуют числам, которые не разложимы в (U,V)-линейные формы. Здесь вершина допускает в наличии одного сына, если соответствующий коэффициент равен нулю.

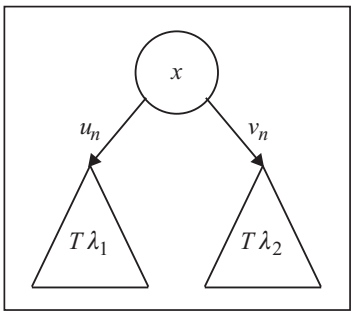

Рис. 1. Линейное дерево для числа $x = \lambda_1 u_n + \lambda_2 v_n$

Следствие 1. При выполнении условий теоремы 1 глубина линейного дерева разложения числа x ограничена величиной $O(\log\log(x))$.

Доказательство. Из теоремы 1 следует, что коэффициенты λ_1 и λ_2, соответствующие первому уровню дерева, не превышают $c\sqrt{x}$. Коэффициенты на втором уровне не превышают $c\sqrt{c\sqrt{x}} = c^{1+1/2}x^{1/4}$. Коэффициенты, соответствующие i-му уровню, не превышают $c^{1+1/2+\ldots+1/2^i}x^{1/2^i}$. Отсюда выводим утверждение следствия.

Во многих случаях для определения двухбазисной системы счисления достаточно задания одного базиса $U = u_1, u_2, \ldots$ Последовательность V определяется сдвигом базиса U на одну позицию, $v_i = u_{i+1}$. Предполагается, что выполняется требование НОД$(u_i, u_{i+1}) = 1$, $i = 1, 2, \ldots$ В этом случае представление (1) переписывается в виде $x = \lambda_1 u_n + \lambda_2 v_{n+1}$.

Для выполнения условий теоремы 1 достаточно, чтобы последовательность U была монотонно возрастающей и существовал предел $\lim\limits_{n\to\infty} \dfrac{u_{n+1}}{u_n}$.

Выбор базисов U и V предполагает их конструктивную генерацию. Существует большой выбор интересных последовательностей в качестве конструктивных базисов. В этой связи интерес представляет каталогизированная база данных OEIS (On-Line Encyclopedia of Integer Seguences), которую также часто называют

в честь ее основателя SLOANE [5], насчитывающая более 200 тысяч последовательностей, накопленных в научных исследованиях.

ПРИМЕРЫ ДВУХБАЗИСНОГО ПРЕДСТАВЛЕНИЯ ЧИСЕЛ

1. Степенные системы счисления. Основной базис U задается последовательностью степеней выбранного базисного числа M, $M>1$, вспомогательный базис V состоит из константной последовательности единиц, $U=0, 1, M, M^2, \ldots$; $V=1, 1, \ldots$

При таком выборе базисов линейное разложение по коэффициентам вспомогательного базиса V совпадает с классической M-ичной системой счисления. Число 0 имеет нулевой ранг. Положительные числа, меньшие M, задаются канонической линейной формой ранга 1 с нулевым коэффициентом λ_2.

2. Аддитивные системы представления чисел. Пусть заданы основной базис $U=u_1, u_2, \ldots$ и вспомогательный базис $V=1, 1, \ldots$, состоящий из единиц. Предположим, что выполняются неравенства $2u_i \geq u_{i+1} > u_i$, $i=1, 2, \ldots$ Тогда из условия максимальности линейных форм разложения следует, что все коэффициенты λ_i в (2) равны единице. В этом случае представление числа x принимает наиболее простой вид:

$$x = u_{n_1} + u_{n_2} + \ldots + u_{n_t} + x_{t+1}, \tag{3}$$

где $n_1 > n_2 > \ldots > n_t$, $0 \leq x_{t+1} < u_1$. В частности, если $u_1 = 1$, то $x_{t+1} = 0$.

3. Код Фибоначчи. Рассмотрим следующий пример. Числа Фибоначчи определяются с помощью рекурсивного соотношения

$$F_{i+1} = F_{i-1} + F_i, \quad i \geq 1, \quad F_0 = F_1 = 1.$$

Основной базис $U=2, 3, 5, \ldots$ задается последовательностью чисел Фибоначчи, начиная с F_2. Дополнительный базис $V=1, 1, \ldots$ состоит из последовательности единиц. Для чисел Фибоначчи выполняется условие: $2F_i > F_{i+1}$. Раскладывая в таком базисе числа в (U, V)-линейные формы, получаем следующий известный результат. Для любого натурального числа x существует его представление в виде суммы некоторых чисел Фибоначчи $x = \sum_{i=1}^{k} d_i F_i$, где $d_i \in \{0, 1\}$, $i=1, \ldots, k-1$, $d_k = 1$.

Поскольку $F_{n-1} + F_n = F_{n+1}$, в последовательности $d_k, d_{k-1}, \ldots, d_1$ нет рядом находящихся единиц. Двоичный код Фибоначчи числа x задается инвертированной последовательностью $d_1, d_2, \ldots, d_k, 1$, в которой приписана дополнительная единица. Наличие двух единиц в конце слова позволяет разделить кодовые слова при их конкатенации.

4. Аддитивное представление числа в системе простых чисел. Любое натуральное число раскладывается в произведение простых чисел. Подобное утверждение справедливо и для представления чисел в виде сумм простых чисел.

Рассмотрим двухбазисную систему следующего вида. Основной базис задается последовательностью простых чисел, $U=2, 3, 5, 7\ldots$, а вспомогательный базис — последовательностью единиц, $V=1, 1, \ldots$

В теории чисел известен постулат Бертрана, сформулированный в 1845 г., согласно которому для любого натурального числа n, $n>3$, существует простое число p, находящееся в интервале $n < p < 2n-2$. Доказательство этого факта впервые получено П. Чебышевым в 1850 г. Впоследствии более сильные формулировки этого утверждения получены Рамануджаном и Эрдешем.

Обозначим через p_i i-е простое число. Из постулата Бертрана следует, что $2p_i > p_{i+1} > p_i$. Следовательно, для любого натурального числа x, $x \neq 0$, существует единственное представление типа (3) в виде суммы простых чисел. В этой формулировке число 1 относим к простым числам.

Согласно оценке количества простых чисел, не превышающих числа x, $\pi(x) \approx x / \ln x$, прямое бинарное представление такого задания числа, аналогич-

ное двоичному коду Фибоначчи, не является оптимальным. Улучшение бинарного представления можно добиться, используя так называемый дельта-подход.

Если $x = p_{i_1} + p_{i_2} + \ldots + p_{i_k}$, $p_{i_j} > p_{i_{j+1}}$, $j = 1, 2, \ldots, k-1$, то для задания x достаточно закодировать k-мерный вектор разностей $(i_1 - i_2, i_2 - i_3, \ldots, i_{k-1} - i_k, i_k)$.

Исходя из этого представления, отметим следующий факт. Существует достаточно эффективное префиксное кодирование чисел, основанное на известной экспериментально проверенной, но не доказанной гипотезе Гольдбаха: любое четное число, большее двух, можно представить в виде суммы двух простых чисел [6].

5. Представление чисел в виде *b*-сумм. Предположим, что вспомогательный базис задается последовательностью повторяющегося одного и того же числа, отличного от единицы.

Пусть $U = 0, u_1, u_2, \ldots$ — основной базис, b — натуральное число, $u = 0$, $v_0 = b$, $b > 1$, $V = b, b, \ldots$ и u_i, $u_i \perp b$, $i = 1, 2, \ldots$ Число b задается линейной формой нулевого ранга. Число b^k последовательно k раз раскладывается в линейные формы нулевого ранга. Разложение числа x в таком базисе позволяет представить x в структурированном виде

$$x = \lambda_1 u_{n_1} + b^{k_1}(\lambda_2 u_{n_2} + b^{k_2}(\ldots(\lambda_t u_{n_t} + x_{t+1} b^{k_t})\ldots)), \tag{4}$$

где либо $x_{t+1} = 1$, либо x_{t+1} — число, не представимое в виде положительно-определенной (U, V)-линейной формы.

Предположим, что $b \perp M$. Зададим базисы $U = 0, 1, M, M^2, \ldots$; $V = b, b, \ldots$ В этом случае в (4) $x_{t+1} < bM$.

6. Цепные дроби. Значительный интерес представляет использование в качестве базисов последовательности числителей и (или) знаменателей подходящих дробей, возникающих при разложении чисел в цепные дроби. В частности, можно использовать хорошо известную эффективную генерацию подходящих дробей квадратичных иррациональностей вида $\sqrt{k}$.

7. Линейные формы Фибоначчи. Пусть U задается рядом Фибоначчи 1, 1, 2, 3, 5 … Линейная форма Фибоначчи ранга n определяется как линейная форма вида

$$x = aF_n + bF_{n+1}. \tag{5}$$

Для линейных форм Фибоначчи существуют быстрые алгоритмы, позволяющие получать максимальные формы. Основой этих алгоритмов служат следующие преобразования.

1. Если $x = aF_k + bF_{k+1}$ и $b \geq a$, то $x = (b-a)F_{k+1} + aF_{k+2}$.
2. Если $a \geq F_{k+1}$, то $x = (a - F_{k+1})F_k + (b + F_k)F_{k+1}$.

Преобразование п. 2 увеличивает коэффициент b, а преобразование п. 1 увеличивает ранг. Отсюда следует утверждение.

Если (5) — максимальная линейная форма Фибоначчи для числа x, $a > 0$, $b \geq 0$, то $b < a < F_{n+1}$. Количество арифметических операций в алгоритме построения максимальной линейной формы Фибоначчи ограничено величиной $O(\log x)$.

Для чисел Фибоначчи существует предел $\lim\limits_{x \to \infty} \dfrac{F_{n+1}}{F_n} = \dfrac{\sqrt{5}+1}{2}$. Поэтому разложение чисел в линейные формы Фибоначчи относится к случаю, задаваемому условиями теоремы 1 и следствием 1. На рис. 2 изображено линейное дерево Фибоначчи для числа 15411.

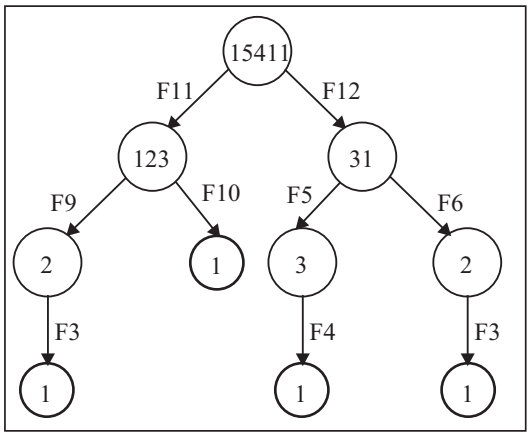

Рис. 2. Линейное дерево Фибоначчи для числа 15411

Линейные формы Фибоначи были введены в [7] и изучались в [8–10].

ПАРАЛЛЕЛЬНОЕ МОДУЛЬНОЕ ЭКСПОНЕНЦИРОВАНИЕ

Операция модульного возведения в степень $a^x = \mod m$ для больших чисел является одной из основных процедур многих известных криптосистем. Время ее выполнения определяет эффективность соответствующих алгоритмов. На сегодняшний день ведутся активные исследования, направленные на ускорение модульного экспоненцирования. Обзор существующих результатов в этой области выходит за рамки данной работы. Двухбазисные системы счисления позволяют предложить эффективные унифицированные средства для увеличения быстродействия возведения в степень за счет сведения к возведению в специально выбранные степени.

Предположим, что в базисных последовательностях U и V возведение в степени u_n и v_n выполняется более просто по сравнению с произвольной степенью.

Если $x = \lambda_1 u_n + \lambda_2 v_n$, то

$$a^x = \mod n = (a^{u_n})^{\lambda_1} (a^{v_n})^{\lambda_2} \mod n. \tag{6}$$

Разлагая рекурсивным способом одновременно λ_1 и λ_2 по базисам (U, V) получаем рекурсивно-параллельный алгоритм вычисления модулярного возведения в степень. Формула (6) определяет двухпроходную структуру алгоритма типа бинарное дерево.

В первой фазе выполняется прохождение вершин от корня к листьям. В каждой вершине происходят однотипные действия: прием числа от вершины-отца, разложение этого числа в максимальную (U,V)-линейную форму, модулярное возведение в степени, определяемые ранговыми числами соответствующих базисов, передача коэффициентов линейного разложения вершинам-сыновьям.

Во второй фазе идет сборка результата от листьев к корню дерева. В каждой вершине выполняется модулярное умножение чисел, получаемых от вершин-сыновей, и результат передается вершине-отцу. Вторую фазу можно выполнять без прохождения по дереву. Результаты, полученные в листьях, в произвольном порядке подаются на узел модулярного умножения, где все они перемножаются по заданному модулю.

Подобная архитектура алгоритма допускает многообразие реализаций как на программном, так и на аппаратном уровнях. Существует достаточно много вариантов выбора базисных последовательностей U и V для рассматриваемой задачи. Считается, что возведение в квадрат является более простой (быстрой) операцией по сравнению с умножением двух чисел. Если x — k-значное число, заданное в базисе степеней числа b, $x = (x_{k-1} \ldots x_1 x_0)_b$, то $x^2 = \sum_{i=0}^{k-1} x_i^2 + 2\sum_{i=0}^{k-2} \sum_{j=i+1}^{k-1} x_i x_j b^{i+j}$.

Вычисление x^2 требует $\frac{1}{2}k(k+1)$ одноцифровых умножений в базисе b по сравнению с k^2 таких умножений при вычислении произведения произвольных k-значных чисел. Поэтому для больших чисел длиною в несколько тысяч бит возведение в квадрат выполняется быстрее обычного умножения на 25–50%. Поэтому в базисах U и V будем в основном использовать степени числа 2.

Наиболее очевидным является использование базисов $u_n = 2^n$, $v_n = 2^n + 1$. Пусть

$$x = a2^k + b(2^k + 1). \tag{7}$$

Представление (7) задает быстрый способ вычисления a, b и n по двоичному заданию числа x. Действительно, (7) можно переписать в виде $x = (a+b)2^k + b$.

Следовательно, достаточно найти максимальное представление $x = q2^k + b$, $q > b$, $a = q - b$.

Если

$$x = 2^n + a_{n-1}2^{n-1} + \ldots + a_0 = 2^k(2^{n-k} + a_{n-1}2^{n-k-1} + a_k) + a_{k-1}2^{k-1} + \ldots + a_0,$$

то в качестве k достаточно выбрать число $\left\lfloor \frac{n-1}{2} \right\rfloor$. В этом случае

$$q = 2^{n-k} + a_{n-1}2^{n-k-1} + \ldots + a_k,\ b = a_{k-1}2^{k-1} + \ldots + a_0,\ a = q - b.$$

Данное решение легко переносится на случай сверхбольших экспонент, задаваемых в степенном базисе с основой $b = 2^M$.

Одним из возможных вариантов соответствующего выбора последовательностей U и V является также выбор базовой последовательности $U = u_0, u_1, \ldots$, члены которой удовлетворяют рекуррентному соотношению $u_{n+1} = 2^M u_{n-1} + u_n$, где M — заданная константа. Члены последовательности V задаются сдвигом на один шаг последовательности U, $v_n = u_{n+1}$, $u_0 = u_1 = 1$. Возможны другие варианты выбора U, когда, например, $u_{n+1} = u_{n-1} + 2^M u_n$. Компьютерные эксперименты показали высокую эффективность предлагаемого подхода [11].

Отметим, что во всех рассматриваемых случаях согласно следствию 1 глубина дерева алгоритма параллельного возведения в степень ограничена величиной $O(\log \log x)$. Число вершин в дереве составляет $O(\log x)$. Это позволяет рассматривать возможность эффективной реализации предлагаемых параллельных алгоритмов на аппаратном уровне.

КОДИРОВАНИЕ ДЕРЕВЬЕВ

Деревья являются одной из наиболее часто используемых алгоритмических структур данных. Поэтому во многих приложениях возникают задачи их оптимального хранения и передачи.

Разложение чисел в линейные формы определяет соответствующие бинарные деревья. Возможен обратный процесс. Если имеется структура типа бинарного дерева, то возможно взаимно однозначное отображение такого дерева в число, получаемое с помощью обратного процесса — построения максимальных линейных форм (1) по коэффициентам λ_1 и λ_2. Для этого необходимо по числам a и b, соответствующим кодам левого и правого поддеревьев, построить число $x = au_n + bv_n$, которое является кодом вершины-отца, при этом линейное представление $au_n + bv_n$ является максимальной (U,V)-линейной формой для x. Декодирование кода дерева осуществляется разложением числа-кода в максимальные линейные формы.

Решение данной задачи является для многих базисов U и V эффективным, в частности для базисов, задаваемых числами Фибоначчи. В этом случае, если $a > b$, достаточно выбрать первое F_n такое, что $a < F_n$. Более сложное кодирование, использующее двойное разложение в максимальные линейные формы, применимо для отмеченных деревьев, в вершинах которых хранится информация (числа).

Возмодно также применение двухбазисных систем, связанных со структурой типа бинарное дерево. Пусть число x задано в форме (1). Рекурсивное кодирование $C(x)$ числа x с помощью индексов разложения в (U,V)-линейные формы, эквивалентное построению линейного дерева (U,V)-разложения числа x, определяется следующим образом:

$$C(x) = (C(\lambda_1))n(C(\lambda_2)).$$

Такое кодирование может быть использовано в криптографических преобразованиях. Последовательности U и V генерируются по секретному ключу.

В принципе U и V могут быть произвольными последовательностями взаимно простых чисел, при этом хотя бы одна из них должна возрастать (не обязательно монотонно).

Рассмотрим частный случай. Ключ задается числом k. Пусть P_n / Q_n — последовательность подходящих дробей разложения в цепную дробь числа $\sqrt{k}$. Последовательности U и V задаются числителями $u_i = P_i$, $v_i = P_{i+1}$ или знаменателями Q_i и Q_{i+1}. В кодовом слове фигурируют только индексы последовательностей U и V и коды остаточных чисел, соответствующих листьям дерева. Поскольку существует бесконечно много вариантов выбора k, то только по индексам восстановить исходное число без знания U и V не представляется возможным.

Нетрудно модифицировать систему таким способом, чтобы U и V после каждого кодирования меняли числа с соответствующими использующимися в кодировании индексами. В этом случае повторное кодирование одного и того же числа будет давать разные кодовые слова.

УНИВЕРСАЛЬНЫЕ ПРЕФИКСНЫЕ КОДИРОВАНИЯ ЧИСЕЛ

Префиксные (префиксно-свободные) коды представляют значительный интерес в связи с их активным использованием во многих приложениях, особенно для сжатия и передачи информации. Наиболее известны представители таких кодов: код Хаффмана [12], коды Эллиаса [13], код Левенштейна [14] и классы кодов, которые используют кодирование Фибоначчи [1–4]. Современный обзор таких кодов можно найти в работе [15].

Двухбазисное представление чисел позволяет выявить широкий спектр новых префиксных кодов. Для этого рассмотрим двухбазисные системы с ограничениями на коэффициенты λ_1 и λ_2 в (1). Пусть R_1 и R_2 — заданные подмножества натуральных чисел. В представлении (1) необходимо, чтобы $\lambda_1 \in R_1$ и $\lambda_2 \in R_2$. Здесь (U,V)-линейные формы с такими ограничениями определяют специфические представления чисел, которые используют свойства множеств R_1 и R_2.

Частный случай такого задания чисел представляет интерес. Пусть p — простое число и g — образующий элемент мультипликативной группы Z_p^* вычетов по модулю p. Рассмотрим (U,V)-разложения натуральных чисел, где U — бесконечная последовательность степеней числа g, расширенная нулем, $U = 0, g, g^2, g^3, \ldots$, а V — константная последовательность, определяемая числом p, $V = p, p, \ldots$ Ограничения на коэффициенты задаются множествами $R_1 = \{1\}$ и $R_2 = \mathbb{N}$. Иными словами, рассматриваются только представления вида $g^n + \lambda p$. Заметим, что λp последовательно раскладывается в число $\lambda' p^k$, $k \geq 1$, где λ' не делится на p. Предположим, что $x > g^{p-1}$ и НОД$(x, p) = 1$. Учитывая, что g — образующий элемент группы Z_p^*, получаем максимальную степень n_1 такую, что

$$x = g^{n_1} + p^{b_1} x_1, \tag{8}$$

где $x_1 > 0$ и x_1 взаимно простое с p.

Для чисел, меньших g^{p-1}, такого представления может не быть. Рассмотрим, например, (2,5)-систему. Число 11 не представимо в виде (8), так как наименьшая степень n для числа 2 такая, что $2^n \equiv 11 \bmod 5$ равна 4, $2^4 = 16$.

Рекурсивно применяя процесс разложения коэффициентов при p в формы вида (8), получаем следующее разложение:

$$x = x_1 = g^{n_1} + p^{k_1} x_2,$$

$$x_2 = g^{n_2} + p^{k_2} x_3, \ldots, x_t = g^{n_t} + p^{k_t} x_{t+1}. \tag{9}$$

В (9) число x_{t+1} не представимо в виде (8).

Структурная форма для x имеет вид

$$x = g^{n_1} + p^{k_1}(g^{n_2} + p^{k_2}(\dots(g^{n_t} + p^{k_t}x_{t+1}))\dots).$$

Обозначим $\overline{n}_i$ максимальную степень g в степенном разложении x по базису g,

$$x_i = a_{\overline{n}_i} g^{\overline{n}_i} + a_{\overline{n}_i - 1} g^{\overline{n}_i - 1} + \dots + a_0, \tag{10}$$

$$0 \le a_j < g,\ j = 0, \dots, \overline{n}_i,\ i = 1, 2, \dots, t+1.$$

Из (10) следует выполнение неравенства $x_i < g^{\overline{n}_i + 1}$. Для значения n_i имеем $p-1$ возможностей:

$$n_i = \overline{n}_i,\ n_i = \overline{n}_i - 1, \dots,\ n_i = \overline{n}_i - (p-2).$$

Отсюда следует выполнение неравенства

$$g^{n_i} + p^{k_i} g^{\overline{n}_{i+1}} < x < g^{\overline{n}_i + p - 1}. \tag{11}$$

В логарифмическом виде из (11) после упрощений получаем неравенство

$$k_i \log_g p + \overline{n}_{i+1} < n_i + \log_g (g^{p-1} - 1), \tag{12}$$

которое можно переписать в виде

$$k_i \log_g p - k_i < n_i - \overline{n}_{i+1} - k_i + \log_g (g^{p-1} - 1). \tag{13}$$

Так как $\log_g (g^{p-1} - 1) < p - 1$, то справедливо неравенство

$$k_i \log_g p - k_i < n_i - \overline{n}_{i+1} - k_i + p - 1. \tag{14}$$

Минимальное значение левой части неравенства (14) равно $\log_g p - 1$ при $k_i = 1$.

Пусть c — наибольшее натуральное число такое, что $g^{c+1} < p$. Из (14) следуют неравенства

$$0 < \log_g p - c - 1,$$

$$0 < \log_g p - c - 1 < n_i - \overline{n}_{i+1} - k_i + p - c - 1. \tag{15}$$

Положим $\Delta_i = \overline{n}_i - \overline{n}_{i+1} - k_i + p - c - 1$. Из (15) следует, что $\Delta_i > 0$. Числа Δ_i и k_i удовлетворяют неравенству

$$k_i \log_g p - k_c - c < \Delta_i. \tag{16}$$

Пары (Δ_i, k_i), $i = 1, 2, \dots, t$, называем блоками. Число x однозначно определяется последним остаточным числом x_{t+1} и последовательностью блоков

$$x \equiv (\Delta_1, k_1)(\Delta_2, k_2)\dots(\Delta_t, k_t)(x_{t+1}). \tag{17}$$

Восстановление числа x по (17) происходит в обратном порядке, начиная с x_{t+1}. Общий шаг восстановления x выполняется следующим образом. По числу x_{i+1} вычисляется $\overline{n}_{i+1}$. Затем по значениям $\Delta_i, k_i, \overline{n}_{i+1}, p$ и c находится число n_i. Остаточное число x_i вычисляется по формуле

$$x_i = g^{n_i} + p^{k_i} x_{i+1},\ i = t, t-1, \dots, 1.$$

Если x — произвольное число, отличное от нуля и единицы, то оно представляется в виде $x = g^{n_0} p^{k_0} x_1$, где x_1 — число, взаимно простое с g и p. Таким образом, произвольное натуральное число x однозначно задается последовательностью

$$(x_{t+1})(n_0, k_0)(\Delta_1, k_1)(\Delta_2, k_2)\dots(\Delta_t, k_t). \tag{18}$$

Остаточные числа в разложении (9) убывают в геометрической прогрессии:

$$x_{i+1} = \frac{x_i - g^{n_i}}{p^{k_i}} < \frac{x_i}{p^{k_i}}. \tag{19}$$

Из (19) следует неравенство

$$\sum k_i \log_2 p < \log_2 x. \tag{20}$$

Так как $k_i \geq 1$, то для числа блоков t получаем грубую оценку:

$$t < \frac{\log_2 x}{\log_2 p}. \tag{21}$$

Префиксные (g, p)-коды. Пусть $\{0,1\}^*$ — множество всех слов в алфавите $\{0,1\}$. Если $u \in \{0,1\}^*$, то $|u|$ обозначает длину слова u, u^k — последовательная конкатенация k раз слова u. Последовательности вида $0^n 1^k$ называем 01-группами.

Последовательность (18) в бинарном алфавите $\{0,1\}$ кодируется следующим образом. Число x_{t+1} удовлетворяет неравенству $x_{t+1} < g^{p-1}$. Поэтому для задания всех чисел, для которых невозможно представление (7), резервируется константное число бит. Это число не превышает $(p-1)\log_2 g$. Пара (n_0, k_0) кодируется 01-группой $0^{n_0+1}1^{k_0+1}$. Пара (Δ_1, k_1) кодируется 01-группой $0^{\Delta_i}1^{k_i}$.

Таким образом, (g, p)-представление чисел задает инъективное отображение $\mathbb{C}_{g,p}$ множества $\mathbb{N}$ в $\{0,1\}^*$,

$$\mathbb{C}_{g,p}(x) = \text{код}(x_{t+1})0^{n_0+1}1^{k_0+1}0^{\Delta_1}1^{k_1}\ldots 0^{\Delta_t}1^{k_t}. \tag{22}$$

Здесь код (x_{t+1}) обозначает кодирование числа x_{t+1}.

Пусть M — множество данных и $U \in \{0,1\}^*$, $\mathbb{C}: M \to U$ есть биективное отображение M на U, определяющее кодирование элементов из M словами из U. Элементы из U называем кодовыми словами, а U — кодом для M.

Кодирование $\mathbb{C}$ называется однозначно декодируемым (uniquely decodable), если по конкатенации кодовых слов $c(m_1)\,c(m_2)\ldots c(m_k)$ можно однозначно восстановить коды $c(m_i)$, $i = 1, \ldots, k$. Код U называется префиксно-свободным (префиксным), если никакое слово из U не может быть началом (префиксом) любого другого слова из U.

Отображение $\mathbb{C}_{g,p}$ не всегда является однозначно декодируемым. Неравенство (15) позволяет определить разделители между кодовыми словами.

Пусть k — наименьшее число такое, что $k \log_g p - k > c+1$. Согласно (16) блока $(1, k)$ не существует. Это означает, что в бинарном задании кодирования не существует кода блока вида $\# = 01^k$. Поэтому можно приписывать 01^k в конце слова $\mathbb{C}_{g,p}(x)$. Тем самым распознается окончание кодового слова.

Определим код $\mathbb{C}_{g,p} = \{\mathbb{C}_{g,p}(x)\# \mid x \in \mathbb{N}\}$, котрый является префиксно-свободным. Оценим длину кодового слова для числа $x = 2^{n_0}3^{k_0}x_1$.

Из представления (22) следует

$$|\mathbb{C}_{g,p}(x)| = |\text{код}(x_{t+1})| + n_0 + k_0 + 2 + \sum_{i=1}^{t}(n_i - \bar{n}_{i+1}) - \sum_{i=1}^{t} k_i + \sum_{i=1}^{t} k_i + t(p-1-c) =$$

$$= \bar{n}_1 + (|\text{код}(x_{t+1})| - \bar{n}_{t+1}) + t(p-c-1) + n_0 + k_0 + 2 + \sum_{i=1}^{t}(n_i - \bar{n}_i).$$

Заметим, что $n_i - \bar{n}_i \leq 0$. Учитывая, что

$$\bar{n}_i = \lfloor \log_g x_i \rfloor,\ x_{t+1} < g^{p-1},\ t < \frac{\log_2 x}{\log_2 p},$$

$$\log_g x = n_0 \log_2 g + k_0 \log_2 p + \log_2 x_1,$$

получаем следующее утверждение.

Утверждение 1. Длина кодового слова $|\mathbb{C}_{g,p}(x)\#|$ не превышает $c_1 + c_2 \log_2 x$, где c_1 и c_2 — соответствующие константы.

Подмножество кодовых слов U из $\{0,1\}^*$ называется универсальным, если для любого перечислимого множества данных M и любого распределения вероятностей P, определенного на M, приписывание данных в порядке убывания вероятностей словам из U в порядке возрастания длин дает среднюю ожидаемую длину кодового слова, ограниченную величиной $c_1 + c_2 H(P)$, где c_1 и c_2 — константы, а $H(P)$ обозначает энтропию распределения.

Понятие универсального кодирования было введено П. Элиасом [13]. Это понятие отражает свойство перечислимого множества слов быть почти оптимальным кодом для произвольного источника данных с любым распределением вероятностей.

В [1], а также в несколько ином представлении в [13] имеется следующее утверждение. Если длина кодового слова для числа x не превышает $c_1 + c_2 \log_2 x$, то такое кодирование всех чисел является универсальным.

Учитывая утверждение 1, получаем следующий факт.

Утверждение 2. Любое (g, p)-представление чисел задает универсальное префиксно-свободное кодирование чисел.

Наиболее простой системой (g, p)-представления чисел является (2,3)-представление, (2,3)-задание чисел совпадает с представлением (4) в системе $U = 0, 1, 2, 2^2, \ldots$ и $V = 3, 3, \ldots$

Рассмотрим пример (2,3)-разложения:

$$20132013 = 3(6710671) =$$

$$= 3(2^{22} + 3(2^{18} + 3(2^{17} + 3(2^{13} + 3(2^{10} + 3(2^9 + 3(2^7 + 3(2^2 + 3^2))))))).$$

Формальный алгоритм (2,3)-разложения имеет следующий вид:

Алгоритм (2,3)-декомпозиция.

Вход: положительное число x.

Результат: массив A пар (n_i, k_i).

1. Представить x в виде $x = 2^{n_0} 3^{k_0} x_1$, где x_1 — нечетное число, которое не делится на 3.

2. $A[0] \leftarrow (n_0, k_0)$, $u \leftarrow x_1$.

3. До тех пор, пока $x \neq 1$, выполнять

3.1. Представить $u = 2^n 3^k v$, где n — максимальная степень такая, что $u \equiv 2^n \bmod 3$ и $u - 2^n > 0$, v — нечетное число, взаимно простое с числом 3.

3.2. $i \leftarrow i + 1$.

3.3. $A[i] \leftarrow (n, k)$, $u \leftarrow v$.

4. Результат A.

Оказалось, что (2,3)-коды обладают многими эффективными качествами [17]. При кодировании чисел C величина $C(x) - \log_2 x$ называется поточечной избыточностью (pointwise redundancy).

Коэффициент поточечной избыточности определяется как

$$\lim_{x \to \infty} \frac{C(x) - \log_2 x}{\log_2 x}.$$

Для (2,3)-кодов среднее ожидаемое значение коэффициента избыточности равно 0,16. Для кодирования Фибоначчи такое значение всегда равно 0,44. (2,3)-коды являются самосинхронизирующимися и обладают свойствами повышенной устойчивости к искажениям.

Подробное изучение (2,3)-представления чисел было начато автором этой работы в [16]. Асимптотические характеристики и свойства (2-3)-кодов изучались [17].

ЗАКЛЮЧЕНИЕ

Как показывает данный обзор, двухбазисное кодирование чисел предлагает широкий спектр средств для решения многих прикладных задач. Двухбазисное задание чисел представляет интерес как многомерное обобщение известных систем счисления.

СПИСОК ЛИТЕРАТУРЫ

1. Apostolico A., Fraenkel A. Robust transmission of unbounded strings using Fibonacci representations // IEEE Trans. form. Theory. — 1987. — **IT-33**, N 2. — P. 238–245.
2. Klein S.T., Ben-Nissan M.K. On the usefulness of Fibonacci compression codes // The Computer J. — 2010. — **53**, N 6. — P. 701–716.
3. Klein S.T. Fast decoding of Fibonacci encoded texts / Proc. Intern., Data Compression Conf. (DCC), 2007 // IEEE Computer Society (USA), 2007. — P. 388.
4. Klein S.T., Ben-Nissan M.K. Using Fibonacci compression codes as alternatives to dense codes / Proc. Intern., Data Compression Conf. (DCC), 2008 // IEEE Computer Society (USA), 2008. — P. 472–481.
5. http:/ocis.org.
6. Fenwick P. Variable length integer codes based on the Goldbach conjecture and other additive codes // IEEE Trans. Inform. Theory. — 2002. — **48**, N 8. — P. 2412–2417.
7. Анисимов А.В., Рындин Я.П., Редько С.Е. Обратное преобразование Фибоначчи // Кибернетика. — 1982. — № 3. — С. 9–11.
8. Анисимов А.В. Линейные формы Фибоначчи и параллельные алгоритмы большой размерности // Кибернетика и системный анализ. — 1995. — № 3. — С. 106–115.
9. Anisimov A.V. Linear Fibonacci forms and parallel algorithms for high dimension arithmetic // Lect. Notes Comput. Sci. — 1995. — **964**. — P. 16–20.
10. Анисимов А.В. Алгоритмічна теорія великих чисел. Модулярна арифметика великих чисел. — Київ: Академперіодика, 2001. — 153 с.
11. Мекуш О.Г. Алогритми модулярної арифметики великих чисел: Автореф. дис. ... канд. фіз.-мат. наук / КНУ імені Тараса Шевченка. — Київ, 2005. — 17 с.
12. Haffman D.A. Method for construction of minimum-redundancy codes // Proc. the IRE. — 1952. — **40**, N 9. — P. 1098–1101.
13. Elias P. Universal codeword sets and representations of the integers // IEEE Trans. Inform. Theory. — 1975. — **21**, N 2. — P. — 194–203.
14. Левенштейн В.И. Об избыточности и замедлении разделимого кодирования натуральных чисел // Пробл. кибернетики. — 1968. — № 20. — С. 173–179.
15. Salomon D. Variable-length codes for data compression. — London: Springer-Verlag London Limited. — 2007. — Sept. — 192 p.
16. Анисимов А.В. Представление чисел в смешанном базисе (2,3) // Кибернетика и системный анализ. — 2009. — № 4. — P. 3–18.
17. Anisimov A. Prefix encoding by means of the (2,3)-representation of numbers // IEEE Trans. Inform. Theory. — 2013. — **59**, N 4. — P. 2359–2374.

Поступила 04.01.2013