УДК 519.7

А.В. АНИСИМОВ

КОДИРОВАНИЕ ДЕРЕВЬЕВ С ПОМОЩЬЮ ЛИНЕЙНЫХ РЕКУРРЕНТНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Аннотация. Предлагается унифицированное кодирование упорядоченных бинарных деревьев с числовыми метками в вершинах с помощью линейных форм соседних членов линейных рекуррентных последовательностей вида $P_{n+2} = a_{n+2}P_{n+1} + P_n$, где $P_1 = P_2 = 1$; $a_3, a_4, \ldots$ — натуральные числа. Процедуры кодирования/декодирования просты в реализации и используют рекурсивную технику прямого обхода дерева способом перебора в глубину. Дан краткий обзор возможных применений такого кодирования для задач обработки деревьев и криптографических преобразований.

Ключевые слова: бинарные деревья, кодирование деревьев, линейные рекуррентные последовательности, числа Фибоначчи.

ВВЕДЕНИЕ

Современные технологии передачи, хранения и защиты информации в основном ориентированы на числовые представления данных. В то же время в своем большинстве информация имеет сложноструктурированную природу. Поэтому универсальное числовое кодирование таких структур представляет одну из актуальных проблем современного этапа информационно-коммуникационных технологий. Предложенное в данной работе универсальное кодирование упорядоченных деревьев натуральными числами направлено на решение этой проблемы.

Деревья являются одной из важнейших структур данных, широко применяемой в многочисленных компьютерных алгоритмах. Поэтому их числовое кодирование представляет значительный интерес. В настоящее время такой интерес также мотивируется интенсивными исследованиями в области химии и биологии для нахождения подобных молекулярных и таксонометрических структур, использующих иерархические модели представления.

Хорошо известно кодирование неориентированных деревьев с использованием последовательностей числовых меток вершин. Одним из первых подобное кодирование применил Прюфер [1] для доказательства теоремы Кэли о количестве неориентированных деревьев с n вершинами. Впоследствии появилось много улучшенных алгоритмов подобного типа [2–4]. Кодирование Прюфера предполагает, что все метки вершин разные, а сам код дерева состоит из последовательности всех меток, упорядоченных специальным образом.

Теоретический интерес представляет взаимно-однозначное соответствие между неупорядоченными корневыми деревьями и натуральными числами, основанное на разложении чисел в произведение простых чисел (Matula–Goebel bijection) [5, 6].

Для упорядоченных корневых деревьев (ordinal trees) существует множество способов их представления в виде специальных числовых последовательностей. К ним относятся представления с использованием уровневых чисел листьев, перечисляемых слева направо, разнообразные последовательности меток, отражающие порядок обхода вершины (pre-order tree traversal), скобочные 0–1 строки бинарных деревьев. Обзор методов генерации деревьев, использующих их представления в виде числовых последовательностей, дан в книге Д. Кнута [7].

Нас интересует представление дерева в виде одного числа. Очевидно, с использованием нумерации Кантора любую последовательность чисел можно вза-

© А.В. Анисимов, 2017

имно-однозначно представить одним числом. Поэтому теоретически любая числовая последовательность, кодирующая структуру дерева, может трактоваться как одно число. Но такие числа не отражают иерархической структуры деревьев, поэтому бесполезно ими манипулировать.

Другой путь числового кодирования деревьев основан на сопоставлении им порядковых позиционных номеров в алгоритмах перечисления. Такие функции явно представлены в [8–11]. Подобное кодирование вычислительно достаточно сложно и также не отражает структурных свойств деревьев.

Также отметим, что стандартная техника приписывания вершинам поддеревьев номера подсчетов используется при решении проблем изоморфизма деревьев и других подобных проблем [12–16]. Такие отображения деревьев в числа не являются кодированиями, так как не имеют обратных функций.

В настоящей статье предлагается новое числовое кодирование для бинарных деревьев с числовыми метками в вершинах. Кодирование задает биективное отображение деревьев в множество натуральных чисел и осуществляется с помощью линейных бинарных форм от соседних членов линейных рекуррентных последовательностей. В частности, таковым является кодирование с помощью линейных форм Фибоначчи. Предлагаемое кодирование согласуется с иерархической структурой дерева в том смысле, что код дерева строится по кодам сыновей корневой вершины. Процедуры кодирования/декодирования просты в реализации и используют стандартную рекурсивную технику прямого обхода дерева способом перебора в глубину (pre-oder depth-first tree traversal).

Предлагаемое кодирование деревьев натуральными числами дает унифицированный подход для решения многих практических задач, связанных с поиском, передачей и обработкой информации в иерархических системах представления данных. Отметим также, что произвольные деревья легко моделируются с помощью бинарных деревьев. Одним из возможных алгоритмов подобного моделирования является преобразование Кнута — построение по входному дереву так называемого left-child right-sibling бинарного дерева. В свою очередь, графовые структуры моделируются с помощью накрывающих деревьев. Поэтому возможно применение данного подхода для кодирования графов. Предлагаемое кодирование оптимально в отношении битовой длины кода к числу вершин при кодировании всех деревьев, а также для плотных деревьев, у которых число вершин приближается к числу 2^k, где k — глубина дерева.

Под бинарным деревом понимаем корневую упорядоченную иерархическую структуру, в которой каждая не концевая вершина имеет либо двух потомков, либо одного левого или одного правого потомка.

В упрощенном варианте кодирования предлагаемый подход легко модифицируется для кодирования корневых бинарных неупорядоченных деревьев.

ЛИНЕЙНЫЕ ФОРМЫ ЧИСЛОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

В компьютерных приложениях форма представления чисел имеет важное значение 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 представляется перспективной тритовая арифметика. Существуют аддитивные формы представления чисел. Наиболее известная из них — представление чисел в виде сумм чисел Фибоначчи. В работах [17–19] мы развиваем иерархический подход представления чисел в двухбазисных системах. Стандартные классические степенные и многие другие системы счисления являются частным случаем

такого способа задания чисел. В этих работах мы выводим двухбазисное представление чисел, используя простые исходные компоненты: две последовательности чисел (базисы) и линейные двухаргументные формы от соседних членов этих последовательностей. Рекурсивная декомпозиция произвольного числа в линейные бинарные формы определяет форму задания числа.

Все рассматриваемые в данной статье числа принадлежат натуральному ряду, обозначенному N. Поэтому далее в тексте это предполагается по умолчанию.

Предположим, что заданы две бесконечные нумерованные последовательности чисел: $U = u_1, u_2, \ldots$ и $V = v_1, v_2, \ldots$ Выражение вида $au_n + bv_n$, где $u_n \in U, v_n \in V$, называем линейной (U, V)-формой.

Представление числа x в виде

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

называем линейным (U, V)-представлением.

Если для всех индексов n НОД$(u_n, v_n) = 1$, то последовательности U и V называем ортогональными, и для этого случая используем обозначение $U \perp V$.

Из алгоритма Евклида нахождения НОД двух чисел следует, что если $U \perp V$, то для любого x существует представление (1). Интерес представляет случай, когда в (1) коэффициенты a и b — неотрицательные числа.

Линейную форму (1) называем положительно-определенной, если $a > 0, b \geq 0$. Для положительно-определенных форм число n в (1) называем рангом представления.

Среди всех возможных положительно-определенных представлений числа x (1) максимального ранга выделяем два специальных вида. В первом случае из всех возможных представлений (1) коэффициент a — минимальный (b — максимальный). Во втором случае коэффициент a — максимальный (b — минимальный). В первом случае представление (1) максимального ранга называем левым каноническим, а во втором — правым каноническим.

Используя тождественное преобразование

$$x = (a \pm kv_n)\, u_n + (b \mp ku_n)v_n,$$

представление (1) максимального ранга всегда можно привести к левому или правому каноническому виду. Отсюда следует, что в левом каноническом представлении выполняются неравенства $0 < a \leq v_n$, а в правом каноническом представлении — неравенства $0 < a$ и $0 \leq b \leq u_n$. Очевидно, если $U \perp V$, то левое (правое) каноническое представление является единственным.

Если для любого натурального числа x существует положительно-определенное представление (1), то пару (U, V) называем двухбазисной системой представления чисел (системой счисления) с базисами U и V.

Формула (1) для разложения в максимальные формы при рекурсивном применении к промежуточным коэффициентам a и b фактически определяет структуру преобразования произвольного числа в ориентированное бинарное дерево. Такое дерево называем линейным (U, V)-деревом числа x и обозначаем его $\mathrm{T}_{U,V}(x)$. Общая рекурсивная структура дерева $\mathrm{T}_{U,V}(x)$ изображена на рис. 1.

Следует отметить, что если

$$u_n < v_n \quad \text{и} \quad \frac{v_{n+1}}{u_n} < c, \tag{2}$$

где c — константа, то, как легко доказать, глубина линейного дерева $\mathrm{T}_{U,V}(x)$ ограничена величиной $0(\log \log x)$.

Во многих случаях выбирается одна последовательность чисел $U = u_1, u_2, \ldots$ Вторая последовательность V задается сдвигом U на одну позицию влево,

$v_n = u_{n+1}$, $n = 1, 2, \ldots$ В этом случае рассматривается задание чисел линейными формами вида $au_n + bu_{n+1}$.

Так, линейные формы Фибоначчи задаются линейными выражениями вида $aF_n + bF_{n+1}$, $a > 0, b \geq 0$, где F_n обозначает n-е число Фибоначчи.

При рассмотрении линейных деревьев для случаев, когда второй базис определяется сдвигом основной базисной последовательности U, используем обозначение $T_U(x)$.

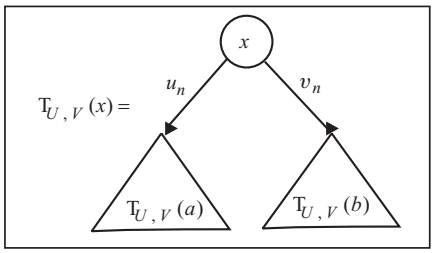

Рис. 1. Линейное дерево $\mathrm{T}_{U,V}(x)$

ЛИНЕЙНЫЕ РЕКУРРЕНТНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ И ЛИНЕЙНЫЕ ФОРМЫ

Как было показано, двухбазисные системы счисления позволяют осуществлять сюръективные отображения натуральных чисел в бинарные деревья. Наша задача — осуществить обратное построение. Необходимо подобрать такие базисы U и V, при которых для произвольного бинарного дерева T, можно однозначно найти такое число x, что $\mathrm{T}_{U,V}(x) = \mathrm{T}$. Тогда x будет числовым кодом дерева T. Предложим решение данной проблемы для базисов, задаваемых соседними членами линейных рекуррентных последовательностей.

Пусть

$$A = a_3, a_4, \ldots \tag{3}$$

является бесконечной последовательностью натуральных чисел, $a_i > 0$. Нумерация членов последовательности, начиная с третьего номера, выбрана для удобства записи рекуррентных соотношений.

Последовательность $\mathcal{P} = P_1, P_2, \ldots$ определяется следующим образом:

$$P_1 = P_2 = 1, \; P_n = a_n P_{n-1} + P_{n-2}, \; n > 2. \tag{4}$$

Рассмотрим положительно-определенные линейные формы вида

$$x = aP_n + bP_{n+1}, \; a > 0, \; b \geq 0. \tag{5}$$

Отметим следующие простые свойства представления (5).

1. Числа P_n и P_{n+1} взаимно просты.

2. Очевидно, что для любого числа x существует положительно-определенное представление вида (5) ранга 2, $x = xP_2$. Значит, последовательность (4) является системой счисления.

3. Если для числа x существует положительно-определенное представление ранга n, то в силу соотношений (4) для x существуют положительно-определенные представления меньших рангов $n-k$, $0 < k \leq n-2$.

Условия максимальности канонической линейной формы даны в следующей теореме.

Теорема 1. Представление (5) является левоканонической формой максимального ранга тогда и только тогда, когда выполняются условия

$$\frac{b}{a_{n+2}} < a < P_{n+1}. \tag{6}$$

Доказательство. Необходимость. Пусть (5) — максимальная левоканоническая линейная форма. Тогда из условия левоканоничности следует $a < P_{n+1}$. Из (4) вытекает равенство

$$x = a(P_{n+2} - a_{n+2}P_{n+1}) + bP_{n+1} = (b - a_{n+2}a)\,P_{n+1} + aP_{n+2}.$$

Таким образом, в максимальной линейной форме (5) ранга n выполняется условие $b - a_{n+2}a < 0$, что соответствует условию (6).

Достаточность. Предположим, что выполняются неравенства (6) и для числа x, заданного в форме (5), существует положительно-определенное представление ранга большего, чем n. Тогда для x существует положительно-определенное линейное представление ранга $n+1$:

$$x = dP_{n+1} + cP_{n+2},\ d > 0.$$

Поскольку $x = (d + kP_{n+2})\,P_{n+1} + (c - kP_{n+1})\,P_{n+2}$, то, не нарушая общности, можно считать, что $c < P_{n+1}$. Получаем

$$x = aP_n + bP_{n+1} = dP_{n+1} + c\,(a_{n+2}P_{n+1} + P_n) = cP_n + (d + a_{n+2}c)\,P_{n+1}.$$

Отсюда следует, что $a \equiv c \bmod P_{n+1}$, а значит, $a = c$. Это означает, что $b = d + a_{n+2}a$. Данный факт противоречит неравенству $a_{n+2}a > b$.

Теорема доказана.

Теорема 2. Пусть задана последовательность (4). Разложение числа x в левоканоническую линейную форму (5) максимального ранга требует применения $O(\log x)$ арифметических операций.

Доказательство. Рассмотрим следующие тождественные преобразования.

$$\text{A: Если } x = aP_m + bP_{m+1},\ a > sP_{m+1},\ s \ge 1,$$

$$\text{то } x = (a - sP_{m+1})\,P_m + (b + sP_m)\,P_{m+1}.$$

$$\text{B: Если } x = aP_m + bP_{m+1},\ b \ge a_{m+2}a,\ \text{то}$$

$$x = (b - a_{m+2}a)\,P_{m+1} + aP_{m+2}.$$

В преобразовании А в качестве s можно выбрать величину $\left\lfloor \dfrac{a}{P_{m+1}} \right\rfloor$. Преобразование А увеличивает второй коэффициент в форме (5), а преобразование В увеличивает ранг. Стартовым представлением является $x = xP_2$. Затем последовательно применяются преобразования А и В до тех пор, пока не выполнится условие максимальности линейной формы (6).

На каждом шаге применения преобразований А или В выполняется ограниченное число простых арифметических операций. Число применений преобразования В определяется максимальным рангом n представления числа x. Из (5) следует неравенство $(a+b)\,P_n \le x$, откуда следует неравенство $P_n < x$.

Учитывая, что для n-го числа Фибоначчи F выполняются соотношения

$$F_n \le P_n,\ F_n \approx \frac{\varphi^n}{\sqrt{5}},\ \varphi \approx \frac{1+\sqrt{5}}{2},$$

получаем неравенство

$$n < \frac{\log_2 x}{\log_2 \varphi} + c,\ n < 1.44 \log_2 x + c,$$

где c — небольшая вычислимая константа.

Теорема доказана.

Богатым источником последовательностей (4) являются числители и знаменатели цепных дробей $a_0 + \dfrac{1}{a_1 + \dfrac{1}{a_2 + \ldots}}$. Как известно, числители и знаменатели цепных дробей (кроме первых двух членов) задаются рекуррентными соотношениями вида (4). В частности, для генерации базовой последовательности (3) можно использовать известный алгоритм разложения в цепную дробь квадратичных иррациональностей вида $\sqrt{n}$. Отметим, что в силу периодичности цепных дробей квадратичных иррациональностей (теорема Лагранжа) выполняется условие ограниченности последовательности (3). Заметим также, что при использовании цепных

дробей в качестве генераторов для (3) значения чисел a_i в основном невелики. Согласно теореме Гаусса–Кузьмина вероятность события $a_i > k$ меньше $\dfrac{1}{k^2 \ln 2}$ [20].

Для чисел, задаваемых линейными формами вида (5), возможно построение оптимальных линейных деревьев специального вида, которое незначительно отличается от общей структуры дерева, изображенного на рис. 1. При задании чисел линейными формами рекуррентных последовательностей (4) под линейным деревом понимаем дерево, получаемое следующим образом.

Пусть $\mathcal{P}$ — последовательность (4) и $x = aP_n + bP_{n+1}$ задается максимальной левоканонической формой. Тогда дерево $T_{\mathcal{P}}(x)$ определяется рекурсивно следующим образом.

Алгоритм *Integer_to_Tree*

1. Если $x = 1$, то дерево $T_{\mathcal{P}}(x)$ состоит из одной изолированной вершины.

2. Если $a > 0,\ b > 0$, то корневая вершина с числом x, где $x = aP_n + bP_{n+1}$ — максимальное левоканоническое представление для x, имеет левое поддерево $T_{\mathcal{P}}\left(a - \left\lfloor \dfrac{b}{a_{n+2}} \right\rfloor\right)$ и правое поддерево $T_{\mathcal{P}}(b)$.

3. Если $b = 0$, n — нечетно, $x \neq 1$, то вершина с кодом x имеет только одного правого сына, который является корнем дерева $T_{\mathcal{P}}(a)$.

4. Если $b = 0$, n — четно, $x \neq 1$, то вершина с кодом x имеет только одного левого сына, который является корнем дерева $T_{\mathcal{P}}(a)$.

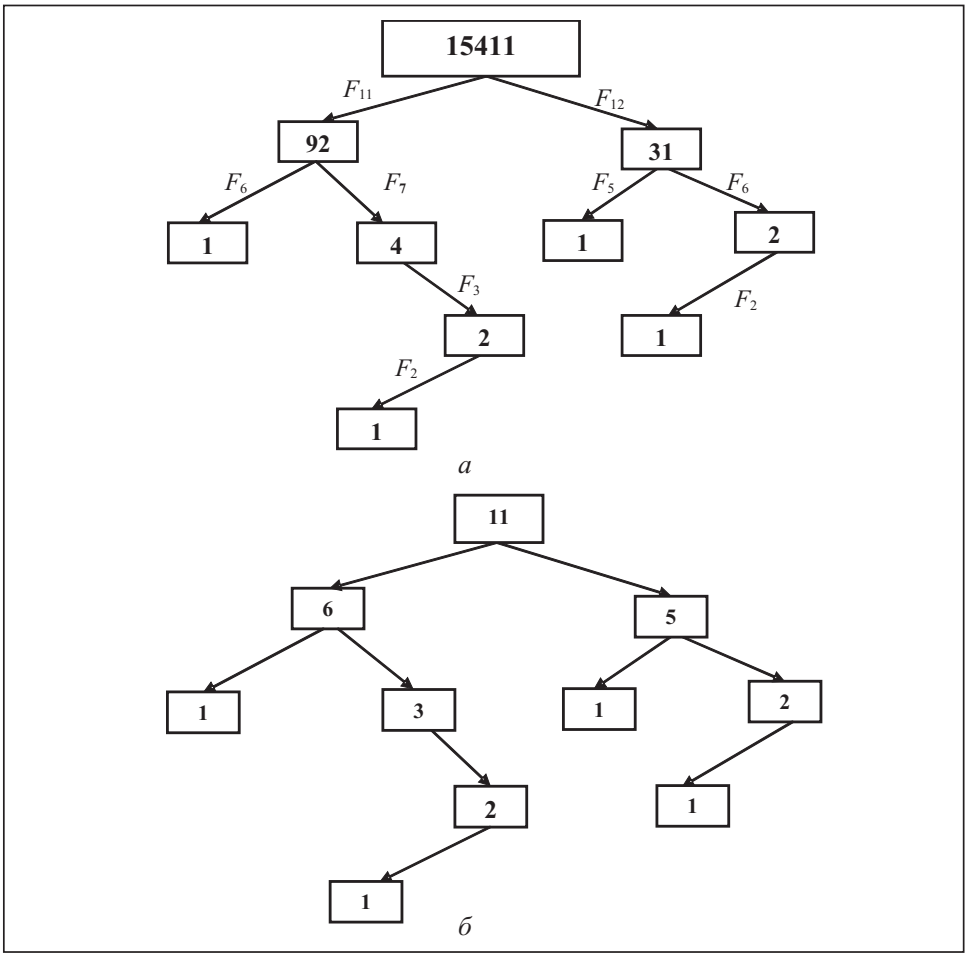

Рис. 2. Процесс конструирования линейного дерева Фибоначчи для числа 15411 (*а*); в вершинах линейного дерева сохраняются ранги соответствующих максимальных разложений (*б*)

Решение задачи кодирования деревьев наиболее просто решается при выборе в качестве (4) ряда Фибоначчи $F_1 = 1, F_2 = 1, F_3 = 2, F_4 = 3, \ldots$ В этом случае все $a_i = 1$.

На рис. 2 дан пример построения линейного дерева Фибоначчи для числа 15411. Пояснения к рис. 2, *а*:

$$15411 = 123F_{11} + 31F_{12}, 92 = 123 - 31 = 5F_6 + 4F_7, 4 = 2F_3, 1 = 5 - 4,$$

$$31 = 3F_5 + 2F_6,\ 1 = 3 - 2,\ 2 = 2F_2.$$

Отметим, что если известны числа $a - \left\lfloor \frac{b}{a_{n+2}} \right\rfloor$, b, n и последовательность $\mathcal{P}$, то можно легко восстановить число $x = aP_n + bP_{n+1}$. Поэтому, зная только структуру дерева $T_{\mathcal{P}}(x)$ и ранги n, запоминаемые в вершинах, можно однозначно восстановить число x в корневой вершине. Так, линейным деревом Фибоначчи числа 15411 будет дерево, где в вершинах сохраняются ранги соответствующих максимальных разложений (рис. 2, *б*).

КОДИРОВАНИЕ ДЕРЕВЬЕВ ЛИНЕЙНЫМИ ФОРМАМИ

При рассмотрении деревьев используем следующие обозначения.

Если v — вершина дерева T, то $v \bullet left$ — ее левый непосредственный потомок, а $v \bullet right$ — соответствующий правый потомок; T_v — поддерево с корнем v; T_v^{left}, T_v^{right} — поддеревья дерева T с корневыми вершинами $v \bullet left$ и $v \bullet right$ соответственно; *root* — корневая вершина дерева T. В данном разделе предполагается отсутствие меток в вершинах.

Пусть задана последовательность (4), a и b — произвольные числа; n — наименьшее число такое, что $\left(a + \frac{b}{a_{n+2}} \right) < P_{n+1}$. Тогда

$$\left(a + \left\lfloor \frac{b}{a_{n+2}} \right\rfloor \right) P_n + bP_{n+1} \tag{7}$$

является максимальной левоканонической линейной формой. Действительно, для коэффициентов этой формы очевидно выполняется условие теоремы 2, $\frac{b}{a_{n+2}} < a + \left\lfloor \frac{b}{a_{n+2}} \right\rfloor < P_{n+1}$. Поэтому если a и b интерпретируются как коды поддеревьев T_v^{left} и T_v^{right}, связанных с корневой вершиной v, то код дерева T_v можно задать с использованием формулы (7).

Кодирование дерева T осуществляется рекурсивным подъемом по дереву от листьев к корню. При этом вычисляется метка текущей вершины v, которая является кодом поддерева T_v. Листьям дерева T присваивается начальная метка 1.

При декодировании после разложения числа-кода текущей вершины v в левоканоническую линейную форму для нахождения кодов поддеревьев T_v^{left} и T_v^{right} следует соответственно из первого коэффициента вычесть второй, деленный на a_{n+2}, а кодом T_v^{right} считать второй коэффициент.

Для различия направленности одиночных ребер, исходящих из вершин, будем использовать признаки четности соответствующих индексов n. Код aP_n при четных n интерпретируется как переход к единственному левому потомку соответствующей вершины. При нечетных n такое значение кода означает переход к правому потомку.

Псевдокод функции кодирования дерева *treetointeger* имеет следующий вид:

Function *treetointeger* (var v: vertex of a tree T): integer;
var a, b: integer;
begin
1. If v is a leaf then $treetointeger \leftarrow 1$;
2. If v has two sons $v_{\bullet} left$ and $v_{\bullet} right$ then do the following:
   2.1. $a \leftarrow treetointeger(v_{\bullet} left)$; $b \leftarrow treetointeger(v_{\bullet} right)$;
   2.2. Find the smallest integer n such that $a + \dfrac{b}{a_{n+2}} < P_{n+1}$;
   2.3. $treetointeger \leftarrow \left(a + \left\lfloor \dfrac{b}{a_{n+2}} \right\rfloor \right) P_n + bP_{n+1}$;
3. If v has only one left son $v_{\bullet} left$ then do the following:
   3.1. $a \leftarrow treetointeger(v_{\bullet} left)$;
   3.2. If $a \neq 1$ then $treetointeger \leftarrow aP_n$, where n is the smallest even integer such that $a < P_{n+1}$;
   3.3. If $a = 1$ then $treetointeger \leftarrow P_4$;
4. If v has only one right son $v_{\bullet} right$ then do the following:
   4.1. $a \leftarrow treetointeger(v_{\bullet} right)$;
   4.2. If $a \neq 1$ then $treetointeger \leftarrow aP_n$, where n is the smallest odd integer such that $a < P_{n+1}$;
   4.3. If $a = 1$ then $treetointeger \leftarrow P_3$;
end.

Алгоритм кодирования дерева T имеет следующий вид.

Алгоритм *Tree_ to_ Integer*
Вход: дерево T, последовательность $\mathcal{P}$;
Результат: код дерева T;
begin
результат $\leftarrow treetointeger(root)$;
end.

Согласно построению выполняется условие $\dfrac{b}{a_{n+2}} < a + \left\lfloor \dfrac{b}{a_{n+2}} \right\rfloor < P_{n+1}$.

В силу теоремы 1 код каждой вершины задается левоканонической максимальной формой от кодов a и b соответствующих сыновей.

На рис. 3 дан пример кодирования дерева с использованием линейных форм Фибоначчи. На рис. 3, *б* вершине v_8 сопоставлено в качестве кода число $7 = 2F_3 + F_4$. Вершине v_5 сопоставлено кодовое значение $163 = (2+7)F_6 + 7F_7$. Первое число Фибоначчи, большее 9, равно $F_7 = 13$. Вершине v_2 сопоставлено число $3F_5 = 15$, $F_5 = 5$. Это первое число Фибоначчи с нечетным номером такое, что $3 < F_6$. Форма $3F_5$ является левоканонической и имеет ранг 5.

Декодирование числа-кода осуществляется динамическим конструированием дерева T от корня к листьям. На первом шаге формируется стартовый корень дерева, с которым связывается число-код. На каждом шаге алгоритма декодирования по текущей паре (v, c), где v — текущая сформированная вершина, а c — ее кодовое значение, выполняется разложение $c = aP_n + bP_{n+1}$ в максимальную левоканоническую форму. Коэффициентами этой формы определяются соответствующие направленные потомки вершины v и значения их текущих кодов. Величина $a - \left\lfloor \dfrac{b}{a_{n+2}} \right\rfloor$ задает код левого поддерева, а b — код правого поддерева. Формальная запись алгоритма декодирования не представляет труда, поэтому здесь опускается.

Следует отметить, что предложенная процедура *Tree_ to_ Integer* легко модифицируется для кодирования корневых бинарных неориентированных деревьев. В этом случае нет необходимости делать различие между левым и правым сыновьями.

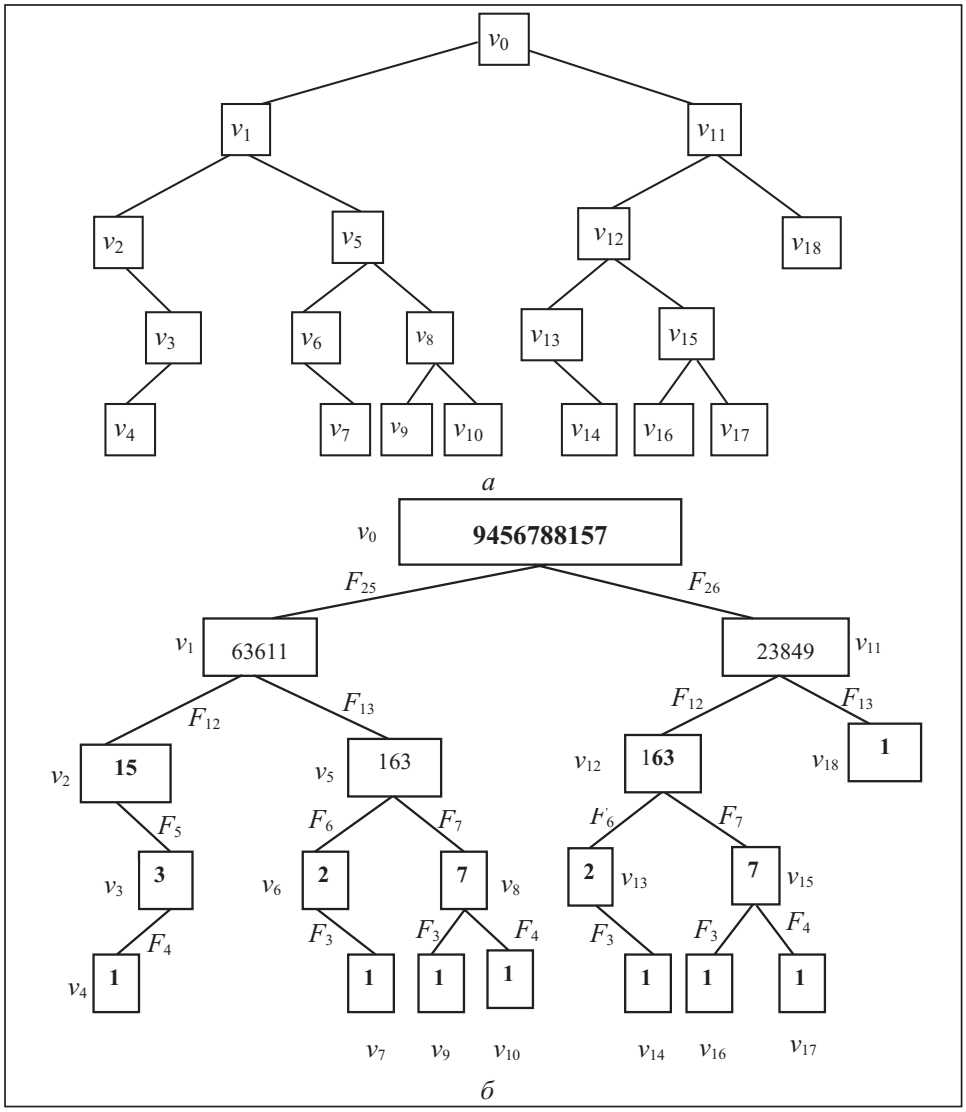

Рис. 3. Исходное дерево для построения кода (*а*); пример вычисления кода дерева с использованием линейных форм Фибоначчи (*б*)

Кодирование выполняется следующим образом.

1. Если вершина v является корневой вершиной деревьев T_1 и T_2 с кодами a и b соответственно, то кодом дерева T_v служит число $x = \left(a + \left\lfloor \frac{b}{a_{n+2}} \right\rfloor \right) P_n + bP_{n+1}$, где n — наименьший номер такой, что $a + \left\lfloor \frac{b}{a_{n+2}} \right\rfloor < P_{n+1}$.

2. Если вершина v имеет одного сына w и кодом дерева T_w является число a, то кодом T_v является число aP_n, где n — наименьшее число такое, что $a < P_{n+1}$.

3. Кодом листовой вершины служит число 1.

Декодирование выполняется очевидным образом.

При декодировании по коду $x = cP_n + dP_{n+1}$ вершины v однозначно восстанавливаются числа $a = c - \left\lfloor \frac{d}{a_{n+2}} \right\rfloor$ и $b = d$, которые являются кодами соответствующих поддеревьев.

Заметим, что в случае неориентированных деревьев код вершины v может быть получен двумя способами:

$$x_1 = \left(a + \left\lfloor \frac{b}{a_{n+2}} \right\rfloor \right) P_n + b P_{n+1} \text{ и } x_2 = \left(b + \left\lfloor \frac{a}{a_{m+2}} \right\rfloor \right) P_m + a P_{m+1},$$

где m — наименьшее число такое, что $\left(b + \left\lfloor \frac{a}{a_{m+2}} \right\rfloor \right) < P_{m+1}$.

При декодировании однозначно восстанавливается пара чисел a и b.

КОДИРОВАНИЕ ДЕРЕВЬЕВ С МЕТКАМИ В ВЕРШИНАХ

Кодирование деревьев с числовыми метками в вершинах сводится к двойному кодированию. Сначала находится код дерева T_v, затем кодируется пара $(\text{код}(\mathrm{T}_v), k_v)$, где k_v — заданная числовая метка вершины v. Таким образом, задача сводится к кодированию пар натуральных чисел.

Известная формула Кантора биективного кодирования числовых пар имеет вид

$$(a, b) \leftrightarrow \frac{(a+b)(a+b+1)}{2} + b.$$

Мы предлагаем более простой вариант кодирования пар натуральных чисел с помощью линейных рекуррентных последовательностей.

Пусть $\mathcal{P}$ — последовательность чисел, задаваемых линейными рекуррентными соотношениями вида (4). Паре (a, b) ставим в соответствии число c:

$$(a, b) \leftrightarrow c = (a+b)P_n + bP_{n+1}, \tag{8}$$

где n — такой номер, при котором выполняется условие $a + b < P_{n+1}$. Очевидно, что линейная форма $(a+b)P_n + bP_{n+1}$ — максимальная. Поэтому, раскладывая c в линейную форму (8), находим a и b.

ОПТИМАЛЬНОСТЬ КОДИРОВАНИЯ

Рассмотрим длину кода дерева, задаваемого алгоритмом *Tree_to_Integer*.

Для линейной формы, которая определяет код текущей вершины, число n выбиралось таким образом, чтобы выполнялись неравенства $P_n \le a + \left\lfloor \frac{b}{a_{n+2}} \right\rfloor < P_{n+1}$.

Это означает, что при построении кода при переходе от непосредственных потомков к вершине-отцу значение кода возрастает по квадратичному закону. Следовательно, если глубина дерева равна k, то битовая длина кодового числа будет величиной порядка 2^k.

Обозначим $S(k)$ количество всех возможных упорядоченных бинарных деревьев глубины, не превышающей числа k, $k \ge 1$. Возможны три случая: корневая вершина имеет одного левого потомка, одного правого потомка или двух потомков. Деревья с корнями в вершинах-потомках имеют глубину, не превышающую $k-1$. Поэтому количество различных деревьев глубины, не превышающей k, определяется рекуррентной формулой

$$S(k) = 2S(k-1) + S^2(k-1) = (S(k-1)+1)^2 - 1.$$

Отсюда получаем $S(k) = 2^{2^k} - 1$. Если использовать в подсчете корневую вершину, рассматриваемую как дерево глубины 0, то получим формулу $S(k) = 2^{2^k}$.

Это означает, что невозможно построить биективное отображение в целые числа всех бинарных деревьев глубины, не превышающей k, и при этом ограни-

читься только числами с битовой длиной, меньшей 2^k. Поэтому предлагаемое числовое представление оптимально при кодировании всех деревьев или отдельных плотных деревьев, у которых число вершин приближается к величине 2^k. Однако во многих приложениях возникают неплотные деревья. В этом случае возможно использовать частичное кодирование поддеревьев с помощью модифицированной процедуры *Tree_to_Integer*.

Кодом дерева будет последовательность чисел, которые являются базовыми кодами поддеревьев, выделяемых из основного дерева. Выбор этих поддеревьев зависит от стратегии кодирования и определяется выбором внутренних вершин, в которых прерывается основное кодирование. Детальное описание процедур кодирования и декодирования для такой ситуации выходит за пределы настоящей работы.

ВОЗМОЖНЫЕ ПРИЛОЖЕНИЯ

В настоящей статье мы даем только краткий набросок возможных приложений предлагаемого метода кодирования.

Сжатие и поиск поддеревьев. Для упорядоченных деревьев существует широкий класс задач, связанных с манипуляциями над поддеревьями: поиск всех поддеревьев, изоморфных заданному, поиск наибольшего общего поддерева, поиск наиболее часто повторяющего поддерева, установление изоморфизма деревьев, сжатие деревьев. Для подобных задач предлагаемый подход сводит задачу для деревьев к задаче над числами-кодами, что унифицирует и упрощает решение. В качестве примера такого класса задач рассмотрим сжатие деревьев более детально.

Обычно сжатие структурированных данных происходит за два этапа. Сжатие данных и сжатие структуры происходит там, где эти данные хранятся. В этом смысле деревья проявляют двойственную структуру. Они часто используются для структурной организации данных, но также сами могут представлять специфическую информацию, например словари, деревья пикселов, XML-деревья, деревья грамматического разбора предложений, хэш-деревья и т.д.

Сжатие деревьев базируется в основном на нахождении одинаковых поддеревьев. Указатели на повторяющиеся поддеревья используются для минимизации избыточной информации. Дерево трансформируется в меньший упорядоченный ациклический граф (DAG). Минимальный такой граф является единственным, и существуют эффективные алгоритмы его построения. (Обзор дан в работах [21, 22].)

Известные алгоритмы компактного представления деревьев основываются на использовании рекурсивной процедуры, которая динамически создает уникальные идентификаторы поддеревьев и поддерживает глобальную поисковую таблицу для них (look-up table). Время выполнения такой процедуры зависит от структуры данных, используемой для поддержки поиска в такой таблице. Прямое применение поиска сравнением дает временную оценку $O(n^2)$, где n — число вершин. Использование сбалансированных деревьев уменьшает общее время до $O(n \log n)$. В [13] с применением варианта корзиночной сортировки (basket sort) удалось сократить время процедуры до $O(n)$. Подобная техника используется также в других задачах сравнения деревьев поиска и графов [14–17].

Кодирование деревьев числовыми линейными формами дает унифицированный подход для первой фазы таких задач. Мы только кодируем дерево вместе с его поддеревьями с помощью линейных форм Фибоначчи. Если поддеревья одинаковые, то и соответствующие числовые коды будут одинаковыми. Проблема сводится к построению числовых кодов и соответствующей корзиночной сортировки.

Отметим также, что предлагаемый подход может применяться к нагруженным деревьям с числовыми метками. Прямое перенесение других известных методов решения подобных задач для деревьев с числовыми метками представляется проблематичным.

Криптография. Возможный сценарий для симметричной криптографии имеет следующий вид. Выбираются три линейные рекуррентные последовательности: $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$. Эти последовательности рассматриваются как секретные ключи системы симметрического шифрования. Например, возможна быстрая генера-

ция $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ из разложений квадратичных иррациональностей $\sqrt{n_1}$, $\sqrt{n_2}$ и $\sqrt{n_3}$ в цепные дроби, где n_1, n_2 и n_3 — секретные ключевые числа. Шифрование происходит за два этапа. На первом этапе число x (послание) преобразуется в дерево $T_{\mathcal{P}_1}(x)$ с помощью разложения x в линейные формы, используя последовательность $\mathcal{P}_1$. Отметим, что на этом этапе не обязательно строить положительно-определенные максимальные формы. Возможно недетерминированное использование разложений в левоканонические линейные формы произвольного ранга, выбираемые недетерминировано. Дерево $T_{\mathcal{P}_1}(x)$ в своих корневых вершинах поддеревьев хранит соответствующие метки-ранги разложения в левоканонические формы чисел, которые раскладываются в линейной форме.

На втором этапе дерево $T_{\mathcal{P}_1}(x)$ кодируется одним числом y с помощью последовательностей $\mathcal{P}_2$ и $\mathcal{P}_3$. Последовательность $\mathcal{P}_2$ используется для кодирования структуры дерева, а $\mathcal{P}_3$ — для кодирования пар (код вершины, метка вершины). Получаем числовой код $y = C_{\mathcal{P}_2,\mathcal{P}_3}(T_{\mathcal{P}_1}(x))$.

При декодировании число y сначала раскладывается в линейное дерево $T_{\mathcal{P}_1}(x)$. Затем по этому дереву составляется число x.

Заметим, что в вершинах дерева $T_{\mathcal{P}_1}(x)$ содержатся только номера-ранги соответствующих линейных разложений. Поэтому можно показать, что для любого числа x существует бесконечно много пар $(x_i, \mathcal{P}_{1i})$ таких, что $T_{\mathcal{P}_1}(x) = T_{\mathcal{P}_{1i}}(x_i)$. Поэтому для всех таких x_i соответствующие коды имеют вид $C_{\mathcal{P}_2,\mathcal{P}_3}(T_{\mathcal{P}_{1i}}(x_i)) = C_{\mathcal{P}_2,\mathcal{P}_3}(T_{\mathcal{P}_1}(x)) = y$. Это означает, что при наблюдении кодов y невозможно получить какую-либо информацию о послании x. Анализ более сильных атак типа CPA (chosen plain-text attack) выходит за рамки данной работы.

Преобразование деревьев в числа могут быть полезны для block-chain технологий при манипуляциях с иерархическими данными. Например, это позволяет в криптовалютах типа Bitcoin или Ethereum использовать иерархические списки блоков транзакций.

ЗАКЛЮЧЕНИЕ

Для задач, связанных с обработкой поддеревьев (сжатие деревьев, поиск, выявление структурных особенностей и т.д.), достаточно использовать простейший вид линейных рекуррентностей, задаваемых числами Фибоначчи. Для класса задач, требующих криптографических преобразований, возможно использование более общих линейных рекуррентных последовательностей, задаваемых секретными ключами. Поэтому здесь рассмотрено общее кодирование линейными рекуррентными последовательностями. Предложенное кодирование приемлемо и для числовых пар, что позволяет выполнять однотипное кодирование деревьев, содержащих в вершинах числовую информацию.

СПИСОК ЛИТЕРАТУРЫ

1. Prüfer H. Neuer Beweis cines Satzesuber Permutationen. *Archiv für Mathematik und Physik.* 1918. Vol. 27. P. 142–144.
2. Deo N., Micikevičius P. Prufer-like codes for labeled trees. *Congressus Numerantium.* 2001. Vol. 151. P. 65–73.
3. Deo N., Micikevičius P. A new encoding for labeled trees employing a stack and a queue. *Bulletin of the Institute of Combinatorics and its Applications.* 2002. Vol. 34. P. 77–85.
4. Neville E.N. The codifying of tree structures. *Proceedings of Cambridge Philosophical Society.* 1953. Vol. 49. P. 381–385.
5. Matula D.W. A natural root tree enumeration by prime factorization. *SIAM Review.* 1968. Vol. 10. P. 273.
6. Göbel F. On a 1–1 correspondence between rooted trees and natural numbers. *Journal of Combinatorial Theory.* Ser. B. August. 1980. Vol. 29, Iss. 1. P. 141–143.
7. Knuth D. Generating all trees — History of combinatorial generation. *The Art of Computer Programming.* Upper Saddle River, NJ: Addison–Wesley Professional. 2006. Vol. 4, Fascicle 4. 128 p.

8. Ruskey F., Hu T.C. Generating binary trees lexicographically. *SIAM Journal on Computing.* 1977. Vol. 6, Iss. 4. P. 745–758.
9. Er M.C. Enumerating ordered trees lexicographically. *The Computer Journal.* 1985. Vol. 28, Iss. 5. P. 538–542.
10. Proskurowski A. Binary tree gray codes. *Journal of Algorithms.* 1985. Vol. 6, Iss. 2. P. 225–238.
11. Zaks S. Lexicographic generation of ordered trees. *Theoretical Computer Science.* 1980. Vol. 10, Iss. 1. P. 63–82.
12. Aho A.V., Hopcroft T.E., Ullman F.D. The design and analysis of computer algorithms. Upper Saddle River. NJ: Addison-Wesley, 1974. 470 p.
13. Downey P.J., Sethi R., Tarjan R.E. Variations on the common subexpression problem. *Journal of the ACM.* 1980. Vol. 27, Iss. 4. P. 758–771.
14. Flajolet P., Sipala P., Steyaert F.M. Analytic variations on the common subexpression problem. *Automata, Languages and Programming* (Lecture Notes in Computer Science). 1990. Vol. 443. P. 220–234.
15. Grossi R. On finding common subtrees. *Theoretical Computer Science.* 1993. Vol. 108, Iss. 2. P. 345–356.
16. Dinitz Y., Itai A., Rodeh M. On an algorithm of Zemlyachenko for subtree isomorphism. *Information Processing Letters*. 1999. Vol. 70, Iss. 3. P. 141–146.
17. Anisimov A.V. Two-base numeration systems. *Cybernetics and System Analysis.* 2013. Vol. 49, N 4. P. 501–510.
18. Anisimov A.V. Prefix encoding by means of the (2,3)-representation of numbers. *IEEE Trans. on Information Theory.* 2013. Vol. 59, N 4. P. 2359–2374.
19. Anisimov A.V. Linear Fibonacci forms and parallel algorithms for large numbers. *Lecture Notes in Computer Science.* 1995. Vol. 964. P. 16–20.
20. Khinchin A.Ya. Gauss's problem and Kuzmin's theorem. Chapter 15. In: Continued Fractions. New York: Dover, 1997. P. 71–86.
21. Katajainen J., Mäkinen E. Tree compression and optimization with applications. *International Journal of Foundations of Computer Science*. 1990. Vol. 1, Iss. 4. P. 425-447.
22. Bille P., Gørtz I.L., Landau G.M., Weimann O. Tree compression with top trees. Proceedings 40th International Colloquim on Automata, Languages and Programming, ICALP 2013. Lecture Notes in Computer Science. 7965 (Part 1), (arXiv: 1304.5702. Vol. 1. [cs. DS] 21 Apr. 2013. P. 160–171).

Надійшла до редакції 20.07.2017

А.В. Анісімов

КОДУВАННЯ ДЕРЕВ ЗА ДОПОМОГОЮ ЛІНІЙНИХ РЕКУРЕНТНИХ ПОСЛІДОВНОСТЕЙ

Анотація. Запропоновано уніфіковане кодування упорядкованих бінарних дерев з числовими позначками у вершинах за допомогою лінійних форм сусідніх членів лінійних рекурентних послідовностей вигляду $P_{n+2} = a_{n+2}P_{n+1} + P_n$, де $P_1 = P_2 = 1$; $a_3, a_4, \ldots$ — натуральні числа. Процедури кодування/декодування прості у реалізації і використовують рекурсивну техніку прямого обходу дерева способом перебору в глибину. Надано короткий огляд можливих застосувань такого кодування для задач обробки дерев і криптографічних перетворень.

Ключові слова: бінарні дерева, кодування дерев, лінійні рекурентні послідовності, числа Фібоначчі.

A.V. Anisimov

CODING TREES BY MEANS OF LINEAR RECURRENCE SEQUENCES

Abstract. A unified integer encoding of ordinal binary trees with integer labels in vertices is given. The encoding is based on the use of linear forms depending on two neighboring members of linear recurrences $P_{n+2} = a_{n+2}P_{n+1} + P_n$, where $P_1 = P_2 = 1$; $a_3, a_4, \ldots$ are natural numbers. Encoding and decoding procedures are simple in implementation and use recursive pre-order tree traversal. A brief review of possible applications for subtree processing and cryptographic symmetric encoding is presented.

Keywords: binary trees, encoding trees, linear recurrent sequences, Fibonacci numbers.

Анисимов Анатолий Васильевич,
чл.-кор. НАН Украины, доктор физ.-мат. наук, профессор, декан Киевского национального университета имени Тараса Шевченко, e-mail: ava@unicyb.kiev.ua.