

УДК 681.142.2

**ВЗАИМНО-РЕКУРРЕНТНЫЕ ФОРМУЛЫ
ДЛЯ ПЕРЕЧИСЛЕНИЯ РАЗБИЕНИЙ ПРЯМОУГОЛЬНИКА¹**

А. М. Магомедов*, Т. А. Магомедов**, С. А. Лавренченко***

** Дагестанский государственный университет, г. Махачкала, Россия**** Uber, г. Амстердам, Нидерланды***** Российский государственный университет туризма и сервиса, г. Москва, Россия*

Задача перечисления разбиений прямоугольника заданных целочисленных размеров $h \times w$ на прямоугольники 1×2 рассматривалась рядом авторов в связи с вопросами термодинамики потоков жидкости и проблемой перечисления совершенных паросочетаний плоского графа за полиномиальное время. В данной работе методами теории графов разработан алгоритм, компьютерное воплощение которого способно генерировать систему взаимно-рекуррентных формул для искомого перечисления разбиений прямоугольника произвольных целочисленных размеров. Решены вопросы инициализации рекуррентных формул; показано, что решение задачи организации вычислений в соответствии с системой формул, сгенерированных компьютером, сводится к топологической сортировке ациклического орграфа; сформулированы открытые задачи.

Ключевые слова: *перечисление разбиений прямоугольника, рекуррентные формулы.*

DOI 10.17223/20710410/46/9

**MUTUALLY-RECURSIVE FORMULAS FOR ENUMERATING
PARTITIONS OF THE RECTANGLE**

A. M. Magomedov*, T. A. Magomedov**, S. A. Lawrencenko***

** Dagestan State University, Makhachkala, Russia**** Uber, Amsterdam, Netherlands***** Russian State University of Tourism and Service, Moscow, Russia*

E-mail: magomedtagir1@yandex.ru, tagir.magomedov@gmail.com,
lawrencenko@hotmail.com

The paper deals with the problem of enumerating the (complete) partitions of a given $h \times w$ rectangle into 1×2 rectangles (dominos). An algorithm is given to find a system of mutually recursive formulas enumerating all such partitions of a given rectangle of an arbitrary size. The idea is as follows. In fact, the process of finding all partitions is equivalent to the process of growing a binary tree T with vertices representing partial partitions of the given rectangle. In the former process, a descriptor means the border between the part already partitioned and the remaining part of the rectangle. Let $\varphi(v)$ be the number of extensions of a partial partition v to a complete partition of the rectangle. The tree-growing process terminates as soon as the descriptor of each pendant vertex of T coincides (up to symmetry) with the descriptor

¹Работа выполнена при финансовой поддержке Дагестанского государственного университета и Отдела математики и информатики ДНЦ РАН.

of some non-pendant vertex of T . The algorithm for generating recurrence relations for calculating the number of complete partitions is based on the following: the sibling vertex y , or z , of a vertex x in the growing tree corresponds to the partial partition obtained by horizontal or vertical placement of a domino at the left upper corner of the not-yet-partitioned part of the partial partition corresponding to x . The desired recurrence relations are written upon the completion of the tree-growing process, on the base of the equation: $\varphi(x) = \varphi(y) + \varphi(z)$. The main result of this paper is an algorithm for computer generation of recurrence relations, using only the operation of integer addition. Almost all previously known formulas for solutions for the problem contain floating-point operations, which require a long computation time and significant computer resources.

Keywords: *enumeration of rectangle partitions, recurrent formulas.*

Введение

Ф о р м у л и р о в к а з а д а ч и и и з в е с т н ы е р е з у л ь т а т ы

В комбинаторике под *перечислением* понимается подсчёт количества или непосредственный перебор и построение всех объектов заданного типа [1].

Прямоугольник M размеров w и h будем обозначать $M(h \times w)$, количество различных способов разбиения $M(h \times w)$ на 1×2 -прямоугольники (плитки) — через $f(h, w)$, при этом каждая плитка расположена горизонтально или вертикально. Требуется найти рекуррентные формулы для $f(h, w)$ и выполнить вычисления.

Задача перечисления разбиений прямоугольника заданных размеров на плитки хорошо известна. Она сводится к задаче подсчёта совершенных паросочетаний в планарном сеточном графе и допускает решение за полиномиальное время [2, 3].

Впервые задача была рассмотрена в связи с вопросами термодинамики потоков жидкости в работах [4, 5], где получена формула, которая в обозначениях данной работы имеет вид

$$f(h, w) = 2^{wh/2} \prod_{j=1}^w \prod_{k=1}^h \left(\cos^2 \frac{\pi j}{w+1} + \cos^2 \frac{\pi k}{h+1} \right)^{1/4}. \quad (1)$$

В [6, с. 92–93] указано, что формула (1) выведена независимо авторами [4] и [5]. В [7, с. 6] утверждается, что независимо эту же формулу получил Д. Кнут. В [7] рассмотрены рациональные производящие функции для задачи при $w \leq 8$. Отметим, что в таблице для $f(h, w)$, $h \leq 12$, $w \leq 8$, построенной в [7], в качестве значений $f(10, 8)$, $f(11, 8)$ и $f(12, 8)$ указаны 1031151240, 8940739821 и 82741005789 соответственно, тогда как их истинные значения, вычисленные по методу, предложенному в настоящей работе, равны соответственно 1031151241, 8940739824, 82741005829 и совпадают со значениями, приведёнными в [8], а также со значениями, вычисленными компьютерной программой по формуле (1).

В [8] с помощью порождающих функций получены рекуррентные формулы для $f(h, w)$ при $w = 2, 3$ и 4 и утверждается, что не составляет труда с привлечением компьютерной программы (на языке APL) распространить идею вывода формул на случай $w > 4$. При этом сами формулы не приведены, а таблица значений $f(h, w)$ для $h = 2, 3, \dots, 30$, $w = 0, 1, \dots, 9$ заполнена лишь частично. Более точно, **ни для какого** $w > 4$ таблица не заполнена полностью **для всех** значений $h = 2, 3, \dots, 30$. Так, для $w = 7$ все ячейки таблицы ниже строки $h = 21$ пусты, для $w = 9$ оставлены пустыми все ячейки ниже строки, соответствующей $h = 14$ и т. д. Подчеркнём, что все приведённые в таблице значения согласуются с результатами наших вычислений.

Операции с плавающей запятой затрудняют точные вычисления по формуле (1). Так, например, для прямоугольника $M(8 \times 16)$ результатом вычисления по формуле (1), выполненного С#-программой с применением вещественных переменных двойной точности, является значение 540061286536919, тогда как точное значение $f(8, 18)$ равно 540061286536921. Этот пример показывает малоприспособность формулы (1) для точных вычислений.

Трудности с выполнением точных вычислений послужили мотивацией для построения систем рекуррентных формул, использующих лишь операции сложения целых чисел, более точно — для построения алгоритма компьютерной генерации таких формул.

Различным аспектам рассматриваемой задачи посвящена большая часть главы 7 работ [9, 10]. Для небольших w и h задача была предложена на VI Всероссийской олимпиаде школьников по информатике [11–13]. В [13] для решения задачи предложен метод динамического программирования; там же показано (см. также п. 1.1 данной работы), что при $w = 2$ значения $f(h, w)$ задаются числами Фибоначчи:

$$f(1, 2) = 1, f(2, 2) = 2; f(h, 2) = f(h - 1, 2) + f(h - 2, 2) \text{ при } h > 2;$$

рассмотрены также следующие простые случаи: а) $f(h, w) = 0$, если и только если произведение $w \cdot h$ нечётно; б) $f(h, 1) = 1$ при чётном h .

При очевидной симметрии $f(w, h) = f(h, w)$ один из параметров w и h удобно принимать за основной. В качестве такового примем w , тогда при каждом фиксированном w искомые рекуррентные формулы будут зависеть лишь от переменной h , которая может принимать при вычислениях произвольно большие значения.

В з а и м н о - р е к у р р е н т н ы е ф о р м у л ы

Нам понадобятся некоторые понятия, относящиеся к рекуррентным формулам. Если для каждой из m последовательностей

$$F[i] = \{F_{i1}, F_{i2}, F_{i3}, \dots\}, \quad i = 0, 1, \dots, m - 1,$$

заданы формальные (т.е. без анализа корректности) формулы, выражающие F_{ij} в виде линейной комбинации элементов других последовательностей и элементов F_{ik} , $k \neq j$, с коэффициентами, зависящими только от i , то такие формулы будем называть *взаимно-рекуррентными формулами (в.р.ф.)*.

Под отрезком $[a..b]$, $a < b$, последовательности будем понимать набор её элементов с индексами $a, a+1, \dots, b$. Пусть q — заданное целое положительное число. Упорядоченную систему в.р.ф. будем называть *согласованной системой взаимно-рекуррентных формул с шагом q* (и обозначать q -в.р.ф.), если j -й элемент F_{ij} каждой последовательности $F[i]$ представлен в виде линейной комбинации элементов отрезка $[j - q + 1..j]$ каждой из последовательностей $F[0], \dots, F[i]$ (подразумевается, что коэффициенты линейной комбинации зависят только от i). Выполнение свойства согласования необходимо для организации вычислений $f(w, h)$ по q -в.р.ф., при этом уменьшение шага q является одним из факторов повышения эффективности процесса вычислений. В контексте, где значение q несущественно, будем применять термин *согласование в.р.ф.* (без указания q).

В [10, гл. 7] приведено подробное обсуждение задачи для случаев $w = 2$ и 3: для случая $w = 2$ с применением производящих функций получена известная рекуррентная формула Фибоначчи, а для $w = 3$ — система в.р.ф., состоящая из двух формул. Однако при небольших значениях w (2 и 3, как в [10], или 4 и 5, как в [14]) из рассмотрения выпадают такие существенные аспекты проблемы, как:

- 1) согласование в.р.ф.; другими словами, приведение в.р.ф. к виду, допускающему выполнение вычислений: после инициализации некоторого «порогового» количества q начальных элементов последовательности для каждой формулы в.р.ф. найти элементы с номером $q + 1$: сначала по первой формуле в.р.ф., затем — по второй формуле и так далее;
- 2) выяснение зависимости порогового значения q от w .

В данной работе: 1) построен алгоритм генерации в.р.ф. для произвольного w (из соображений обзорности в.р.ф. при компьютерной реализации алгоритма мы ограничились значениями $w \leq 13$); 2) в качестве вспомогательной задачи решена задача согласования в интерпретации задачи упорядочения вершин ациклического ориентированного графа; 3) рассмотрены вопросы инициализации в.р.ф.; 4) выполнены вычисления.

Отметим, что после генерации в.р.ф., их согласования и инициализации выполнение вычислений со сверхбольшими целыми числами не представило затруднений, так как язык программирования C# [15], на основе которого подготовлено программное обеспечение, включает соответствующий класс BigInteger.

Вывод в.р.ф. «вручную», как это осуществлено в [10] для малых значений w , представляется малореальным уже при значениях w , близких к 10, и вовсе невыполнимым при больших w . Это видно из следующего утверждения, сформулированного с привлечением программного обеспечения (через $Q(w)$ обозначено количество формул в в.р.ф.).

Утверждение 1. Для $w = 2, \dots, 13$ значения $Q(w)$ задаются табл. 1.

Т а б л и ц а 1

w	2	3	4	5	6	7	8	9	10	11	12	13
$Q(w)$	1	2	3	8	10	33	38	142	158	653	709	3050

1. Генерация в.р.ф.

1.1. Клетка ветвления и частичное разбиение

Суть решения, предложенного в [10] для случая $w = 2$, заключается в следующем. Закрашенная на рис. 1, *а* клетка (см. ниже определение *клетки ветвления*) разветвляет процесс разбиения на плитки в том смысле, что входит либо в горизонтальную плитку, либо в вертикальную (рис. 1, *б* и *в* соответственно). Поэтому $f(h, 2)$ равно сумме чисел разбиений светлой фигуры рис. 1, *б* и светлой фигуры рис. 1, *в*. Очевидно, что число разбиений светлой фигуры рис. 1, *в* равно числу разбиений светлой фигуры рис. 1, *г*, т. е. $f(h - 2, 2)$. Таким образом, $f(h, 2) = f(h - 1, 2) + f(h - 2, 2)$. Поскольку $f(1, 2) = 1$, $f(2, 2) = 2$, при $w = 2$ получаем рекуррентную формулу Фибоначчи.

Считая *начальной клеткой* горизонтальной плитки её левую клетку, а *начальной клеткой* вертикальной плитки — верхнюю, заметим, что упорядочение клеток исходного прямоугольника по строкам

$$M_{11}, \dots, M_{1w}, \dots, M_{h1}, \dots, M_{hw} \quad (2)$$

индуцирует упорядочение плиток в каждом *полном разбиении*: из двух плиток предшествующей будем считать плитку, чья начальная клетка располагается раньше в упорядочении по строкам; для соответствующего упорядочения плиток в полном разбиении также будем применять термин «по строкам».

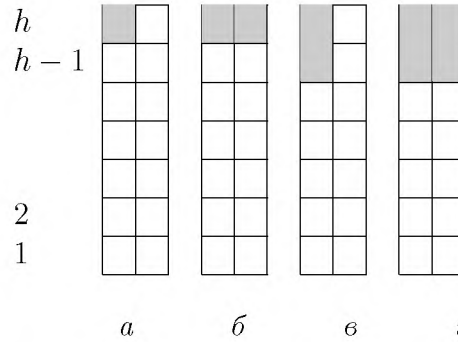


Рис. 1. Рекуррентная формула Фибоначчи, получаемая при $w = 2$ (слева — обозначения высот столбцов)

Понятие *частичного разбиения* (ч.р.) определяется рекурсивно. Пустое покрытие считается частичным разбиением и обозначается μ_{11} ; клетку M_{ij} назовем *клеткой ветвления ч.р.*, если выполнены два условия: 1) M_{ij} является первой непокрытой плиткой клеткой в последовательности (2); 2) клетки M_{ij} , $M_{i,j+1}$ и $M_{i+1,j}$ существуют и не покрыты плитками.

Если μ_{ij} — ч.р. с клеткой ветвления M_{ij} , то наименьшее разбиение, содержащее μ_{ij} и одну из двух пар клеток — $(M_{ij}, M_{i+1,j})$ либо $(M_{ij}, M_{i,j+1})$, — считается частичным разбиением — потомком ч.р. μ_{ij} , если обладает клеткой ветвления или равно полному разбиению.

Другими словами, первая незакрашенная клетка ч.р. называется клеткой ветвления, если её правая и нижняя соседние клетки не закрашены. Для ч.р. удобно называть занятую плитками часть прямоугольника *закрашенной фигурой*, а оставшуюся часть — *светлой фигурой*. Поскольку в перечислении разбиений без ограничения общности можно применять способ укладки плиток по строкам, в любом ч.р. закрашенной фигурой будет верхняя, а светлой — нижняя.

1.2. Громоздкость метода северо-западной клетки

Полный перебор разбиений прямоугольника $M(h \times w)$ естественно представить в виде построения двоичного ориентированного дерева. Название следующего алгоритма 1 связано с местом, занимаемым клеткой ветвления в светлой фигуре ч.р.

Алгоритм 1. Метод северо-западной клетки

- 1: Пустому разбиению μ_{11} сопоставить корневую вершину дерева; клеткой ветвления корневой вершины является M_{11} .
 - 2: Осуществить последовательное рассмотрение и нумерацию листьев уровней $0, 1, \dots$: пока имеется лист дерева, отличный от полного разбиения, для каждого такого листа — ч.р. v — включать клетку ветвления в добавляемую в ч.р. плитку — горизонтальную или вертикальную — и продолжить отдельно каждое из двух полученных разбиений до получения потомка ч.р. v .
-

Двоичное дерево, генерируемое методом северо-западной клетки, будем называть *полным двоичным деревом*. Рассматривая упорядоченность плиток полного разбиения по строкам в качестве хронологической последовательности укладки плиток, получим следующее утверждение.

Утверждение 2. Набор листьев полного двоичного дерева — суть все полные разбиения.

Построение полного двоичного дерева при больших значениях w и h весьма проблематично из-за громоздкости, в этом смысле утверждение 2 имеет лишь ту ценность, что служит основой для исключения (*элиминации*) полного перебора. Элиминации перебора посвящены п. 1.3 и 1.4.

Замечание 1. Там, где это не приводит к неоднозначности, для ч.р. и соответствующей ему вершины полного двоичного дерева будем использовать одно и то же обозначение.

1.3. К л а с с ы э к в и в а л е н т н о с т и

Каждому ч.р. v сопоставим *дескриптор* $(v_1, \dots, v_w)$, где

$$v_k = h - \max\{i : M_{ik} \in v, k = 1, \dots, w\}$$

и индекс $\text{ind}_v = \max\{v_1, \dots, v_w\}$. Вектор $(v_k - \min\{v_1, \dots, v_w\} : k = 1, \dots, w)$ будем называть *нормализованным дескриптором* v . Другими словами, *дескриптор ч.р.* — вектор, образованный высотами столбцов светлой фигуры ч.р., а *индекс ч.р.* — наибольшее значение этих высот.

Замечание 2. Если придерживаться способа укладки плиток по строкам, то элементы нормализованного дескриптора любого ч.р. принадлежат множеству $\{0, 1, 2\}$.

Для ч.р. v обозначим через $\varphi(v)$ количество способов достройки v до полного разбиения. Если x и y — (непосредственные) потомки соответствующей вершины v , то

$$\varphi(v) = \varphi(x) + \varphi(y).$$

Очевидно, что $\varphi(v)$ не зависит от способа укладки плиток в закрашенной фигуре и всецело определяется конфигурацией светлой фигуры, т. е. нормализованным дескриптором ч.р. v и ind_v .

Рассмотрим двоичное поддерево, полученное в результате выполнения начальных шагов методом северо-западной клетки. Вершины с дескрипторами $(v_1, \dots, v_w)$, $(v_w, \dots, v_1)$ и $(v_1 + C, \dots, v_w + C)$, где $C = \text{const}$, будем называть *эквивалентными*. Другими словами, две вершины дерева назовём эквивалентными, если их нормализованные дескрипторы совпадают с точностью до симметрии. Выполним разбиение множества всех вершин на классы эквивалентности $F[0], F[1], F[2], \dots$. Количество различных способов достройки ч.р. v (вершины дерева), принадлежащей $F[i]$, до полного разбиения будем называть *перечислением для вершины v* и обозначать $F[i, \text{ind}_v]$ (рис. 2).

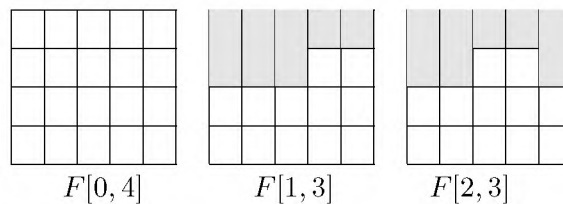


Рис. 2. Обозначение перечисления для вершины дерева — частичного разбиения (включает номер класса эквивалентности и индекс — высоту светлой фигуры)

1.4. Элиминация перебора

Лист v , полученный на некотором шаге метода северо-западной клетки, будем называть *терминальной вершиной*, если в момент создания v уже существует эквивалентная ей вершина v' степени больше единицы; при этом вершину v' будем называть *вторичной терминальной вершиной*. На рис. 3 под каждым ч.р. — вершиной дерева — приведён её дескриптор. Для ч.р., соответствующих вершинам $v_8, v_9, v_{10}, v_{11}, v_{14}$ и v_{15} , сохранены лишь обозначения, рисунки этих ч.р. удалены из соображений экономии места. Для листа v_{12} имеется эквивалентная вершина v_4 степени 3; таким образом, вершина v_{12} является терминальной, а v_4 — вторичной терминальной. Начало поддерева с корнем в вершине v будет таким же, как и начало поддерева с корнем в вершине v' . Это наблюдение служит основой для генерации взаимно-рекуррентных формул.

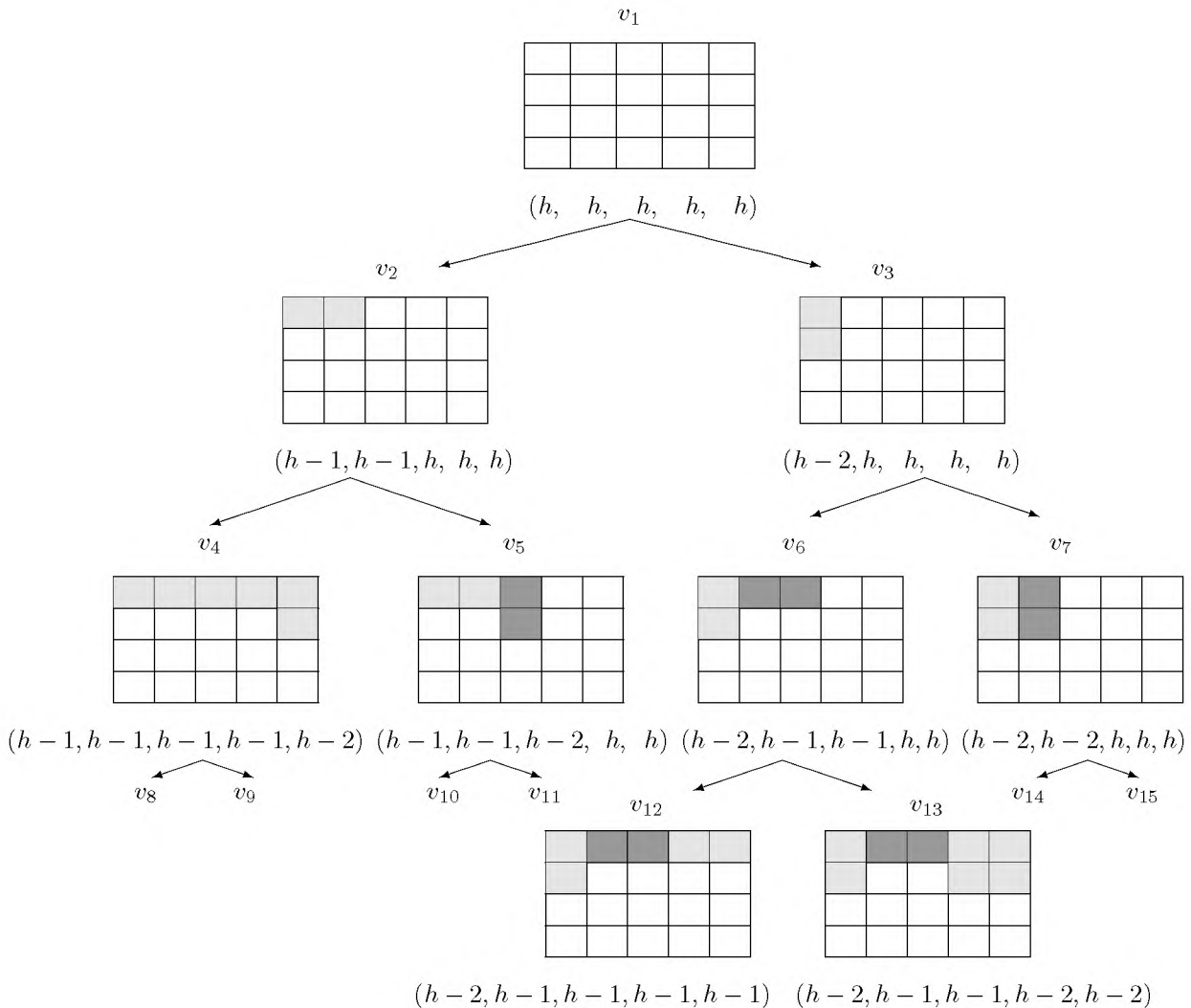


Рис. 3

В качестве критерия прерывания метода северо-западной клетки выберем достижение условия «все листья построенной части дерева являются терминальными вершинами», а построенное до прерывания дерево будем называть *элиминированным двоичным деревом*. Эффективность построения элиминированного двоичного дерева видна из следующего утверждения.

Утверждение 3. Для любой вершины v i -го уровня ($w \leq i$) двоичного дерева найдется такой уровень j , что $i - w \leq j \leq i + w$ и j -й уровень содержит вершину, эквивалентную вершине v .

Доказательство. В самом деле, если вдоль границы (по одну сторону от неё) ч.р. вертикально уложить w плиток, получим ч.р. с тем же нормализованным дескриптором, что и у исходного ч.р. ■

Пусть v — вторичная терминальная вершина элиминированного двоичного дерева. Двоичное поддерево T_v с корнем в вершине v будем называть *терминальным*, если среди его вершин терминальными являются только корень и листья.

Утверждение 4. Перечисление для вторичной терминальной вершины v равно сумме перечислений для листьев терминального поддерева T_v .

Справедливость утверждения 4 вытекает из формулы (2).

Приведём алгоритм 2 перечисления разбиений по терминальным поддеревьям.

Алгоритм 2. Перечисление по терминальным поддеревьям

- 1: Из каждого класса эквивалентности $F[i]$ выбрать по одной вторичной терминальной вершине v .
 - 2: Построить терминальное поддерево T_v и представить перечисление $F[i, \text{ind}_v]$ в виде суммы перечислений для листьев терминального поддерева T_v .
-

Утверждение 5. Алгоритм 2 формирует в.р.ф. для решения задачи перечисления разбиений $M(h \times w)$ за время $O(2^w)$.

Доказательство. По определению терминальной вершины каждый класс эквивалентности $F[i]$ содержит не менее двух вершин; среди них в точности одна вершина терминальная, следовательно, в $F[i]$ всегда найдётся вторичная терминальная вершина. Из утверждения 4 следует, что алгоритм 2 формирует в.р.ф. для решения задачи перечисления разбиений $M(h \times w)$.

Так как построение элиминированного двоичного дерева завершается, как только в каждом ч.р. — листе дерева — все клетки верхних двух строк прямоугольника покрыты плитками, время построения в.р.ф. не превышает $O(2^w)$. ■

2. Вычислительные аспекты

2.1. Структура программного обеспечения

Пусть элиминированное двоичное дерево построено (напомним, что все его листья являются терминальными вершинами). Обозначим искомое множество в.р.ф. через R , а множество терминальных вершин, включающее точно одного произвольного выбранного представителя из каждого класса эквивалентности, — через N . Сначала каждой вершине из N присвоим строковые метки вида

$$F[\text{номер класса эквивалентности, индекс}],$$

затем по алгоритму «Перечисление по терминальным поддеревьям» сгенерируем искомое множество R .

Программное сопровождение рассматриваемой задачи состоит из трёх модулей. В первом реализован алгоритм «Перечисление по терминальным поддеревьям», а также сконструированы нормализованные дескрипторы вершин множества N для передачи второму модулю. Второй модуль выполняет инициализацию начальных значений

в.р.ф. для каждого из нормализованных дескрипторов, вычисленных первым модулем. Третий модуль предназначен для выполнения расчётов в соответствии с в.р.ф., сгенерированными первым модулем, и значениями их начальных элементов, вычисленных вторым модулем; отметим, что расчёты выполняются с использованием арифметики больших целых чисел.

Приведём систему в.р.ф., сгенерированных компьютерной программой для случая $w = 8$:

$$\begin{aligned}
F[00, H] &= F[01, H] + F[15, H] + F[05, H] + F[29, H] + F[19, H] + F[36, H - 1] + F[06, H - 1] + \\
&\quad + F[00, H - 2]; \\
F[01, H] &= F[02, H] + F[14, H] + F[11, H] + F[07, H - 1] + F[26, H - 1] + F[06, H - 1]; \\
F[02, H] &= F[06, H] + F[04, H - 1] + F[10, H - 1] + F[02, H - 1]; \\
F[03, H] &= F[32, H] + F[01, H - 1] + F[20, H - 1] + F[21, H - 1] + F[14, H - 1] + F[03, H - 2]; \\
F[04, H] &= F[17, H] + F[18, H] + F[01, H - 1] + F[20, H - 1] + F[02, H - 1] + F[04, H - 2]; \\
F[05, H] &= F[13, H] + F[17, H - 1] + F[08, H - 1] + F[00, H - 2] + F[10, H - 2]; \\
F[06, H] &= F[00, H - 1] + F[01, H - 1]; \\
F[07, H] &= F[36, H] + F[01, H - 1] + F[20, H - 1]; \\
F[08, H] &= F[06, H] + F[11, H]; \\
F[09, H] &= F[06, H] + F[22, H - 1] + F[16, H - 2] + F[03, H - 2] + F[09, H - 2]; \\
F[10, H] &= F[02, H] + F[05, H]; \\
F[11, H] &= F[10, H - 1] + F[08, H - 1]; \\
F[12, H] &= F[04, H - 1] + F[17, H - 1]; \\
F[13, H] &= F[01, H - 1] + F[00, H - 2] + 2F[10, H - 2] + F[16, H - 2] + F[26, H - 2]; \\
F[14, H] &= F[03, H - 1] + F[16, H - 1] + F[09, H - 1]; \\
F[15, H] &= F[23, H] + F[12, H] + F[25, H - 1] + F[07, H - 1] + F[36, H - 1]; \\
F[16, H] &= F[09, H] + F[13, H] + F[14, H - 1] + F[16, H - 2]; \\
F[17, H] &= F[36, H] + F[10, H - 1] + F[04, H - 2] + F[17, H - 2]; \\
F[18, H] &= F[04, H - 1] + F[04, H - 2] + F[18, H - 2]; \\
F[19, H] &= F[02, H - 1] + F[00, H - 2] + F[10, H - 2]; \\
F[20, H] &= F[04, H - 1] + F[03, H - 1] + F[07, H - 1]; \\
F[21, H] &= F[03, H - 1] + F[28, H - 2] + F[21, H - 2]; \\
F[22, H] &= F[14, H] + F[23, H] + F[29, H]; \\
F[23, H] &= F[24, H - 1] + F[03, H - 1] + F[32, H - 1]; \\
F[24, H] &= F[32, H] + F[30, H] + F[15, H - 1] + F[27, H - 1] + F[31, H - 1] + F[23, H - 1] + \\
&\quad + F[24, H - 2]; \\
F[25, H] &= F[36, H] + F[15, H - 1] + F[27, H - 1]; \\
F[26, H] &= F[06, H] + F[13, H]; \\
F[27, H] &= F[24, H - 1] + F[25, H - 1]; \\
F[28, H] &= F[21, H] + F[31, H] + F[32, H - 1] + F[28, H - 2]; \\
F[29, H] &= F[32, H - 1] + F[09, H - 1] + F[00, H - 2] + F[22, H - 2]; \\
F[30, H] &= F[33, H - 1] + F[35, H - 1] + F[24, H - 2] + F[30, H - 2]; \\
F[31, H] &= F[24, H - 1] + F[28, H - 2] + F[33, H - 2] + F[31, H - 2];
\end{aligned}$$

$$\begin{aligned}
F[32, H] &= F[36, H] + F[28, H - 1] + F[22, H - 1] + F[03, H - 2] + F[24, H - 2] + F[32, H - 2]; \\
F[33, H] &= F[31, H] + F[34, H - 1] + F[30, H - 1] + F[33, H - 2]; \\
F[34, H] &= F[33, H - 1] + F[34, H - 2]; \\
F[35, H] &= F[23, H] + F[30, H - 1] + F[37, H - 1] + F[35, H - 2]; \\
F[36, H] &= F[00, H - 1] + F[20, H - 1] + F[27, H - 1] + F[17, H - 2] + F[32, H - 2] + F[36, H - 2]; \\
F[37, H] &= F[35, H - 1] + F[03, H - 2] + F[37, H - 2].
\end{aligned}$$

Из алгоритма 2 следует, что в формуле с левой частью $F[i, H]$ второй индекс каждого слагаемого в правой части $\leq H$.

Замечание 3. Компьютерные эксперименты дают основания для уточняющего предположения: «вторые индексы суть H , $H - 1$ или $H - 2$ », однако доказать справедливость данного предположения не удалось.

Если в в.р.ф. R присутствует формула с левой частью $F[i, H]$, содержащая в правой части слагаемое $F[j, H]$ с тем же вторым индексом, то будем говорить, что *формула j предшествует формуле i* . Для организации вычислений по в.р.ф. R «сверху вниз» требуется упорядочить R таким образом, чтобы соблюдалось условие: если для пары формул i и j известно, что j предшествует i , то в упорядочении R формула j встречается раньше, нежели формула i . Задачу такого упорядочения будем называть *задачей согласования в.р.ф.* Объясним на примере.

Из фрагмента

$$\begin{aligned}
F[00, H] &= F[01, H] + F[15, H] + F[05, H] + F[29, H] + F[19, H] + F[36, H - 1] + \\
&\quad + F[06, H - 1] + F[00, H - 2]; \\
F[26, H] &= F[06, H] + F[13, H]; \\
F[37, H] &= F[35, H - 1] + F[03, H - 2] + F[37, H - 2]
\end{aligned}$$

видно, что формуле 0 должны предшествовать формулы 1, 5, 15, 19 и 29; у формулы 37 нет предшествующих формул; у формулы 26 две предшествующие формулы — 6 и 13.

Полный список отношений частичного предшествования для в.р.ф. R для случая $w = 8$ приведён в табл. 2. Каждая ячейка начинается с её номера $i = 0, 1, \dots, 37$; номера всех формул, предшествующих в в.р.ф. R формуле i (если таковые имеются), перечислены в i -й ячейке после разделительного знака «:».

Т а б л и ц а 2
Полный список частичных предшествований
для случая $w = 8$

0: 1,15,5,29,19	6:	12:	18:	24: 32,30	31:
1: 2,14,11	7: 36	13:	19:	25: 36	32: 36:
2: 6	8: 6,11	14:	20:	26: 6,13	33: 31:
3: 32	9: 6	15: 23,12	21:	27:	34:
4: 17,18	10: 2,5	16: 9,13	22: 14,23,29	28: 21,31	35: 23
5: 13	11:	17: 36	23:	29:	36:
				30:	37:

Утверждение 6. В в.р.ф. R отсутствуют циклические ссылки.

Доказательство. Каждое слагаемое в этих формулах (линейных комбинациях) является целым положительным числом, поэтому наличие формулы вида

$$F[H, j] = \dots + F[H, k] + \dots$$

исключает наличие формулы вида

$$F[H, k] = \dots + F[H, j] + \dots$$

Утверждение доказано. ■

Напомним известное утверждение о топологической сортировке [16, с. 95].

Утверждение 7. Вершины ациклического ориентированного графа G на n вершинах можно таким образом пометить числами из множества $\{0, 1, \dots, n-1\}$, что если в графе G имеется дуга (i, j) , то $i < j$.

Алгоритм топологической сортировки [16, с. 96] состоит в следующем. Сначала произвольная вершина с нулевой полустепенью исхода помечается числом $n-1$ и удаляется из графа вместе с инцидентными дугами. Затем в оставшемся графе произвольная вершина с нулевой полустепенью исхода помечается числом $n-2$. Описанная процедура повторяется, пока не пометим все вершины. Об одном применении топологической сортировки для решения задачи о существовании расписаний специального типа см. в [17].

Утверждение 8. Согласование в.р.ф. R всегда выполнимо.

Доказательство. Рассмотрим орграф $G(R) = (V, E)$, вершины $v_0, v_1, \dots, v_{n-1}$ которого соответствуют в.р.ф. R , дуги — отношению предшествования формул: дуга из вершины v_i в вершину v_j проведена тогда и только тогда, когда i -я формула предшествует j -й. По утверждениям 6 и 7 упорядочение в.р.ф. R в соответствии с топологической сортировкой вершин графа $G(R)$ обеспечивает согласование R . ■

Поскольку топологическая сортировка выполнима за линейное время, получаем следующее

Следствие 1. Согласование в.р.ф. R выполнимо за время $O(2^w)$.

Замечание 4. Задача топологической сортировки текстуально близка к задаче поиска ориентированного гамильтонова пути в каждой компоненте ориентированного графа. В общем случае задача о гамильтоновом пути NP-полна [18, с. 249], но для ациклических ориентированных графов задача об ориентированном гамильтоновом пути разрешима за полиномиальное время [19]. Заметим также, что ориентированный граф имеет цикл в том и только в том случае, когда алгоритм поиска в глубину [20] находит обратную дугу.

Замечание 5 (о сложности процедуры построения согласованных в.р.ф. R).

Процедура состоит в последовательном выполнении алгоритма 2 и алгоритма согласования. Из утверждения 5 и следствия 1 получим, что вычислительная сложность процедуры построения согласованных в.р.ф. R не превышает $O(2^w)$.

2.2. Инициализация в.р.ф.

Как отмечено в п. 2.1, модуль 1 генерирует нормализованные дескрипторы каждой вершины множества N для передачи модулю 2. Модуль 2 вычисляет начальные

значения для всех в.р.ф., составляющих R . Нетривиальным является вопрос о количестве подлежащих вычислению начальных элементов каждой из (согласованных) в.р.ф. Нетрудно видеть связь данного вопроса со значением шага в.р.ф. Компьютерные преобразования подтверждают следующую гипотезу (проверка выполнена для $w \leq 13$), но доказать её не удалось.

Гипотеза 1 (о шаге в.р.ф.). Сгенерированные алгоритмом «Перечисление по терминальным поддеревьям» и согласованные алгоритмом топологической сортировки в.р.ф. являются 3-согласованными взаимно-рекуррентными формулами.

Из гипотезы 1 следует, что для организации вычислений по сгенерированным согласованным в.р.ф. достаточно инициализировать по три начальных значения каждого из $F[i]$. Приведём примеры вычисленных значений $f(h, w)$:

- 1) $f(3, 2) = 3$;
- 2) $f(20, 3) = 413403$ [10, с. 379];
- 3) $f(2, 4) = 5$;
- 4) $f(100, 5) = 4995246427425596587926101947511568142197556312989986399$;
- 5) $f(8, 6) = 167089$ (в [13, с. 53] допущена опечатка: « $f(8, 6) = 4213133$ »);
- 6) $f(30, 7) = 744382189686310539093281$;
- 7) $f(20, 8) = 3547073578562247994$ [13, с. 54];
- 8) $f(100, 8) = 82480872701819841011582029499055748502616126613824529802178700733106822468932478950831981372929$.

ЛИТЕРАТУРА

1. *Stanley R. P.* Enumerative Combinatorics. V.2. Cambridge: Cambridge University Press, 1999. 228 p.
2. *Montroll E. W.* Lattice statistics // Applied Combinatorial Mathematics / ed. E. F. Beckenbach. N.Y.: John Wiley and Sons, 1964. P. 96–143.
3. *Караваяев А. М., Перепечко С. Н.* Задача о димерах на цилиндрах: рекуррентные соотношения и производящие функции // Математическое моделирование. 2014. Т. 26. № 11. С. 18–22.
4. *Kateleyn P. W.* The statistic of dimers on a lattice I: The number of dimer arrangements on quadratic lattice // Physica. 1961. V. 27. P. 1209–1225.
5. *Temperley H. N. V. and Fisher M. E.* Dimer problem in statistical mechanics — an exact result // Phil. Mag. 1961. V. 6. P. 1061–1063.
6. *Matoušek J.* Thirty-three Miniatures: Mathematical and Algorithmic Applications of Linear Algebra: Amer. Math. Soc., 2010. 182 p.
7. *Klarner D. and Pollack J.* Domino tilings of rectangles with fixed width // Discr. Math. 1980. V. 32. P. 45–52.
8. *Read R. C.* A note on tiling rectangles with dominoes // Fib. Q. 1980. V. 18. No. 1. P. 24–27.
9. *Graham R. L., Knuth D. E., and Patashnik O.* Concrete Mathematics. Massachusetts: Addison-Wesley, 1994. 657 p.
10. *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. Основание информатики: пер. с англ. М.: Мир, 1998. 703 с.
11. <http://neerc.ifmo.ru/school/archive/1993-1994/ru-olymp-roj-1994-problems.html> — Олимпиады по информатике. 1994.
12. *Кирюхин В. М.* VI Всероссийская олимпиада школьников по информатике // Информатика и образование. 1994. № 3. С. 47–50.
13. *Волченков С. Г.* Задача «Паркет» // Информатика и образование. 1994. № 3. С. 52–54.

14. Магомедов А. М., Магомедов Т. А. Компьютерный вывод рекуррентных формул разбиения прямоугольника // Тез. докл. X Белорусской матем. конф., 3–7 ноября 2008 г. Ч. 4. Минск: Институт математики НАН Беларуси, 2008. С. 44.
15. Albahari J. and Albahari B. C# 5.0 in a Nutshell. The Definitive Reference. 5th ed. O'Reilly Media, 2012. 1064 p.
16. Swamy M. N. and Thulasiraman K. Graphs, Networks and Algorithms. N.Y.: Wiley-Interscience, 1981. 590 p.
17. Магомедов А. М. Цепочечные структуры в задачах о расписаниях // Прикладная дискретная математика. 2016. № 3(33). С. 67–77.
18. Garey M. R. and Jonson D. S. Computers and Intractability. San Francisco: Freeman and Company, 1979. 347 p.
19. Lawler E. L. Combinatorial Optimization: Networks and Matroids. N.Y.: Holt, Rinehart, and Winston, 1976. 384 p.
20. Емеличев В. А., Мельников О. И., Сарванов В. И., Тышкевич Р. И. Лекции по теории графов. М.: Книжный дом «Либроком», 2009. 392 с.

REFERENCES

1. Stanley R. P. Enumerative Combinatorics, vol. 2. Cambridge, Cambridge University Press, 1999. 228 p.
2. Montroll E. W. Lattice statistics. Applied Combinatorial Mathematics (ed. E. F. Beckenbach). N.Y., John Wiley and Sons, 1964, pp. 96–143.
3. Karavaev A. M. and Perepechko S. N. Zadacha o dimerakh na tsilindrakh: rekurrentnye sootnosheniya i proizvodyashchie funktsii [The problem of dimers on a cylinder: recurrence relations and generating functions]. Matematicheskoe Modelirovanie, 2014, vol. 26, no. 11, pp. 18–22. (in Russian)
4. Kotelevy P. W. The statistic of dimers on a lattice I: The number of dimer arrangements on quadratic lattice. Physica, 1961, vol. 27, pp. 1209–1225.
5. Temperley H. N. V. and Fisher M. E. Dimer problem in statistical mechanics — an exact result. Phil. Mag., 1961, vol. 6, pp. 1061–1063.
6. Matoušek J. Thirty-three Miniatures: Mathematical and Algorithmic Applications of Linear Algebra. Amer. Math. Soc., 2010. 182 p.
7. Klarner D. and Pollack J. Domino tilings of rectangles with fixed width. Discr. Math., 1980, vol. 32, pp. 45–52.
8. Read R. C. A note on tiling rectangles with dominoes. Fib. Q., 1980, vol. 18, no. 1, pp. 24–27.
9. Graham R. L., Knuth D. E., and Patashnik O. Concrete Mathematics. Massachusetts, Addison-Wesley, 1994. 657 p.
10. Graham R., Knut D., Patashnik O. Konkretnaja Matematika. Osnovanie Informatiki [Concrete Mathematics]. Moscow, Mir Publ., 1998. 703 p. (in Russian)
11. <http://neerc.ifmo.ru/school/archive/1993-1994/ru-olymp-roi-1994-problems.html> — Olympiads in Informatics, 1994.
12. Kirjuhin V. M. VI Vserossijskaja olimpiada shkol'nikov po informatike [VI all-Russian Olympiad in Informatics]. Informatika i Obrazovanie, 1994, no. 3, pp. 47–50. (in Russian)
13. Volchenkov S. G. Zadacha «Parket» [Task «Tiling»]. Informatika i Obrazovanie, 1994, no. 3, pp. 52–54. (in Russian)
14. Magomedov A. M. and Magomedov T. A. Komp'yuternyj vyvod rekurrentnyh formul razbieniia prjamougol'nika [Computer construction of recurrence formulas for partitions of the rectangle]. Proc. X Belarusian Math. Conf., November 3–7, 2008. Part 4. Minsk, Institut of Mathematics NAS of Belarus, 2008, p. 44. (in Russian)

15. *Albahari J. and Albahari B.* C# 5.0 in a Nutshell. The Definitive Reference. 5th Ed. O'Reilly Media, 2012. 1064 p.
16. *Swamy M. N. and Thulasiraman K.* Graphs, Networks and Algorithms. N.Y., Wiley-Interscience, 1981. 590 p.
17. *Мазомедов А. М.* Цепочечные структуры в задачах о расписаниях [Chain structures in scheduling tasks]. Прикладная Дискретная Математика, 2016, no. 3(33), pp. 67–77. (in Russian)
18. *Garey M. R. and Jonson D. S.* Computers and Intractability. San Francisco, Freeman and Company, 1979. 347 p.
19. *Lawler E. L.* Combinatorial Optimization: Networks and Matroids. N.Y., Holt, Rinehart and Winston, 1976. 384 p.
20. *Emelichev V. A., Mel'nikov O. I., Sarvanov V. I., and Tyshkevich R. I.* Лекции по теории графов [Lectures on Graph Theory]. Moscow, Librokom Publ., 2009. 392 p. (in Russian)