

УДК 519.19

АЛГОРИТМЫ ГЕНЕРАЦИИ КОРНЕВЫХ ДЕРЕВЬЕВ НА ОСНОВЕ ПРОЦЕДУРЫ ПОЛНОГО РАЗБИЕНИЯ

В. В. Кручинин

Деревья являются одним из важнейших классов комбинаторных множеств, которые достаточно хорошо изучены. Однако можно предложить ряд алгоритмов генерации классов корневых деревьев, основанных на процедуре полного разбиения. Впервые подобную процедуру предложил Н. Я. Виленкин как подсчет процессов последовательных разбиений [1]. Р. Стенли предложил использовать эту процедуру для полного разбиения конечного множества S , в результате выполнения которой получается некоторое корневое дерево [2]. Основная идея данного доклада заключается в том, что эту процедуру можно модифицировать, используя вместо конечного множества S натуральное число n . Тогда, имея некоторый алгоритм $R(n)$ представления натурального числа n в виде суммы, в общем случае, неотрицательных чисел λ_i , строится непомеченное n -узловое корневое дерево. Алгоритм построения дерева следующий.

- 1) Создаем корень дерева r и пару $\langle n, r \rangle$ заносим в стек.
- 2) Если стек пуст, то завершаем процедуру.
- 3) Вынимаем из стека очередную пару $\langle k, z \rangle$. Если $k = 1$, то данный узел становится листом, и переходим на шаг 2, иначе, в соответствии с алгоритмом R , получаем представление $\pi : [\lambda_1 + \lambda_2 + \dots + \lambda_m]$, такое, что $\sum_{i=1}^m \lambda_i = k - 1$, и переходим на шаг 4.
- 4) Создаем m узлов $\{s_i\}_{i=1}^m$, присоединяем их в качестве сыновей к узлу z в соответствии с порядком, заданным в представлении $\{\lambda_i\}_{i=1}^m$. Все пары $\langle \lambda_i, s_i \rangle$ записываем в стек. Переходим на шаг 3.

Рассмотрим свойства предложенного алгоритма полного разбиения.

1. Если задан $R(n)$ и $\lambda_i \geq 1$ для всех представлений $\pi \in P_{n-1}$, то предложенный алгоритм всегда остановится.

Это утверждение основано на условии, что в дереве с n узлами имеется корень, следовательно, число $n - 1$ является количеством узлов во всех поддеревьях сыновей корня. Алгоритм R получает некоторое представление $\lambda_1 + \lambda_2 + \dots + \lambda_m = n - 1$, где λ_i определяет число узлов в поддереве i -го сына и $1 \leq \lambda_i \leq n - 1$. Тем самым, последовательно применяя алгоритм R к числам, меньшим чем n , получим останов. Поскольку алгоритм строит корневое дерево и всегда останавливается, то он всегда строит дерево.

2. Число деревьев для заданного числа узлов равно

$$K(n) = \sum_{\pi \in P_{n-1}} \prod_{i=1}^k K(\lambda_i), \quad (1)$$

где $\pi = \{\lambda_1 + \lambda_2 + \dots + \lambda_k = n - 1\}$; P_{n-1} — множество представлений числа $n - 1$ в соответствии с алгоритмом R .

Применим этот алгоритм для генерации n -узловых корневых упорядоченных деревьев. Поскольку в таких деревьях порядок следования сыновей в узле важен и сумма чисел узлов в поддеревьях сыновей равна $n - 1$, то для генерации таких деревьев можно использовать процедуру полного разбиения, основанную на генерации композиций

натурального числа n . Тогда

$$K(n+1) = \sum_{\pi \in S_n} \prod_{i=1}^k K(\lambda_i), \quad (2)$$

где $\pi = \{\lambda_1 + \lambda_2 + \dots + \lambda_k = n\}$; S_n — множество композиций числа n . Известно также [2], что $K(n+1) = C_n$, где C_n — число Каталана. Тогда получим следующее тождество для чисел Каталана:

$$C_n = \sum_{\pi \in S_n} \prod_{i=1}^k C_{\lambda_i-1}.$$

Из выражения (2) следует, что каждая композиция порождает некоторое подмножество корневых упорядоченных деревьев. Для построения эффективного генератора [3] необходимо знать число деревьев для каждой композиции $\pi \in S_n$. Обозначим через $T(i)$ число корневых упорядоченных деревьев для i -й композиции в лексикографическом упорядочении композиций. С использованием свойств композиций было получено выражение для функции $T(i)$:

$$T(i) = \begin{cases} 1, & i = 0, \\ T(i - 2^m)^{\frac{4l+2}{l+2}}, & i > 0. \end{cases} \quad (3)$$

где $m = \lfloor \log_2(i) \rfloor$, $l = \begin{cases} m - \lfloor \log_2(2^{m+1} - i - 1) \rfloor - 1, & 0 \leq i < 2^{m+1} - 1, \\ m, & i = 2^{m+1} - 1. \end{cases}$

На рис. 1 изображен график функции $T(i)$. Свойства этой функции таковы:

- 1) $\sum_{i=0}^{2^m-1} T(i) = C_m$, где C_m — число Каталана.
- 2) $T(2^m - 1) = C_{m-1}$.
- 3) В точках $i = 2^m - 1$ функция $T(i)$ будет иметь локальные максимумы.

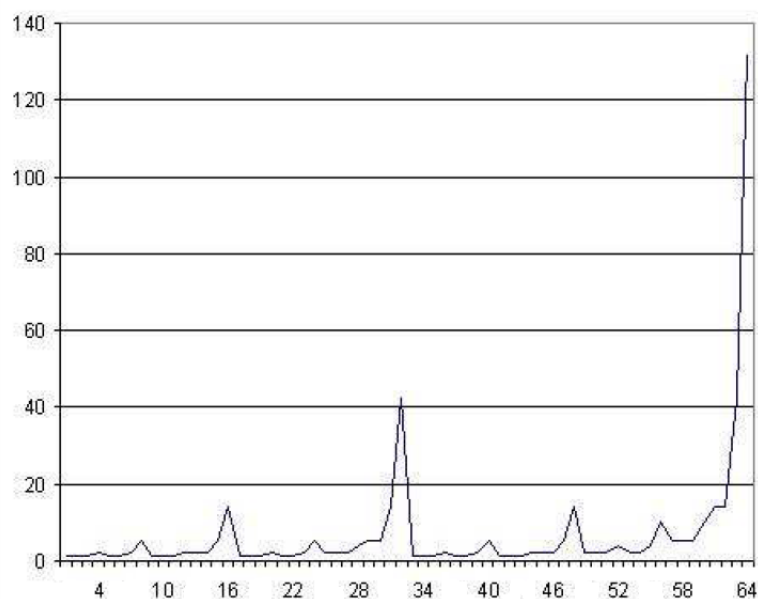


Рис. 1. График функции $T(i)$

В докладе будут подробно рассмотрены алгоритмы генерации корневых деревьев с использованием процедуры полного разбиения, основанной на разложениях, композициях и разбиениях натурального числа n .

ЛИТЕРАТУРА

1. Виленкин Н. Я. Комбинаторика. М.: Наука, 1969.
2. Стенли Р. Перечислительная комбинаторика. Деревья, производящие функции и симметрические функции. М.: Мир, 2005.
3. Кручинин В. В. Методы построения алгоритмов генерации и нумерации комбинаторных объектов на основе деревьев И/ИЛИ. Томск: Изд-во «В-Спектр», 2007.

УДК 519.175.1

АЛГОРИТМЫ ПРОВЕРКИ ИЗОМОРФИЗМА ГРАФОВ НА ОСНОВЕ ИХ ПОСЛЕДОВАТЕЛЬНОЙ СОГЛАСОВАННОЙ ДЕРЕГУЛЯРИЗАЦИИ

И. В. Широков, А. В. Пролубников

В задаче проверки изоморфизма графов (задача ИГ) даны два обыкновенных графа с одинаковым числом вершин и ребер. Необходимо ответить на вопрос, существует ли такое биективное отображение (изоморфизм) множества вершин одного графа на множество вершин второго, которое сохраняло бы смежность соответствующих вершин?

По причине неопределенности своего положения в иерархии теории сложности, задача ИГ имеет большое теоретическое значение, а также часто возникает и в приложениях. Алгоритмы решения задачи ИГ используются при решении многих прикладных задач: в задачах распознавания образов, в протоколах доказательства с нулевым разглашением; к задаче ИГ может быть сведена и вычислительно-эффективно решена задача дешифрования шифра двойной перестановки [1]. Поскольку для вычислительно сложных случаев задачи ИГ не разработано полиномиальных алгоритмов решения, возможно построение криптографических схем, основанных на вычислительной сложности решения для них задачи ИГ, например, как в [2].

Под последовательной согласованной дерегуляризацией пары графов мы понимаем такое последовательное изменение элементов матриц смежности графов (весов вершин и ребер), вследствие которого понижается мощность групп автоморфизмов графов при сохранении равенства некоторых вычисляемых в ходе дерегуляризации инвариантных характеристик графов. В докладе рассматриваются полиномиальные схемы алгоритмов для задачи ИГ, основанные на таком подходе и в качестве инварианта использующие элементы обратной матрицы к модифицированной матрице смежности.

Предложенные алгоритмы протестированы на библиотеке задач [3]. Не найдено примеров неправильного решения или невозможности нахождения решения представленными алгоритмами задачи ИГ для деревьев, случайных, планарных графов, регулярных k -мерных сеток ($k \leq 4$) и некоторых других классов графов. Установлено, что алгоритм решает и те задачи изоморфизма графов из библиотеки, на которых время работы алгоритмов Ullman и NAUTY — наиболее эффективных алгоритмов решения общего случая задачи ИГ — становится экспоненциальным при некоторой нумерации вершин [4]. Кроме этого, задача ИГ успешно решается предложенными алгоритмами и для сильно регулярных графов из наиболее обширной их библиотеки [5] — эти графы представляют собой класс, для которого задача проверки изоморфизма графов име-