УДК 510.52

РЕГУЛЯРНЫЕ ДЕРЕВЬЯ РЕКУРСИИ: ОПИСАНИЕ И ТЕОРЕТИЧЕСКИЙ АНАЛИЗ

THE REGULAR RECURSIVE TREE:

Головешкин В.А., д-р. техн. наук, проф.,

Пономарёв А.В., кандидат физ.-мат. наук.

Московский государственный университет приборостроения и информатики

Ульянов М.В., д-р. техн. наук, университет проф.

Национальный исследовательский университет – Высшая школа экономики,

Московский государственный университет печати им. Ивана Федорова

Аннотация

Анализ рекурсивных алгоритмов методом подсчета вершин порожденных деревьев предполагает детальное исследование их структур. В связи с этим в статье предложено специальное описание регулярных деревьев, характерных для ряда рекурсивных алгоритмов. Разработан метод, доставляющий аналитическое решение для числа порожденных вершин на каждом уровне такого регулярного дерева, и базирующийся на введенном регулярном описании. Полученные в статье результаты позволяют провести теоретический анализ трудоемкости рекурсивных алгоритмов, порождающих регулярные деревья рекурсии.

Abstract

Recursive algorithm analysis by means of counting generated trees tops refers to detailed study of their structure. Therefore the following article presents a special description of regular trees peculiar to several recursive algorithms. A method based on introduced regular description is devised to provide an analytical solution for a number of generated tops on each level of such regular tree. The results achieved in the article make possible a theoretical analysis of working time for recursive algorithms generating regular recursion trees.

1. Введение

Решение задачи выбора рационального алгоритма в процессе разработки программного обеспечения базируется, как правило, на одной из основных характеристик качества алгоритма — трудоёмкости. Под трудоёмкостью понимается число базовых операций принятой модели вычислений, заданных алгоритмом на конкретном входе [1]. Более значимой оценкой является функция трудоёмкости, аргументом которой является длина входа алгоритма. Содержательно функция трудоёмкости рассматривается в лучшем, худшем и средним случаях [2]. Существующие методы анализа алгоритмов позволяют получить вычислительную сложность алгоритма — асимптотическую оценку функции трудоёмкости в худшем случае, и, в ряде случаев — точную функцию трудоёмкости, позволяющую решать задачу рационального выбора в реальных диапазонах длин входов. Задача получения точной функции трудоёмкости является актуальной как для итерационных, так и для рекурсивных реализаций алгоритмов, в аспекте теоретического анализа которых в настоящее время предложен ряд методов, позволяющих получить аналитические выражения для функций трудоёмкости [1, 2, 3]. Одним из перспективных в этом направлении является метод анализа порождённого дерева рекурсии, наиболее сложным этапом которого является получение аналитических выражений для числа вершин порождённого дерева рекурсии на каждом его уровне [1].

Основная проблема теоретического анализа дерева рекурсии состоит в том, что в общем случае число порождённых вершин является функцией как номера уровня дерева, так и номера порождающей вершины на данном уровне. Более того, в ряде случаев такая функция не задаётся аналитически и структура дерева существенно определяется спецификой конкретного входа алгоритма, что существенно усложняет поставленную задачу. В качестве примера приведём рекурсивный алгоритм решения классической задачи оптимальной упаковки методом динамического программирования, для которого порождённое дерево определяется как объёмом упаковки, так и объемами типов грузов [1]. Тривиальными для этой задачи являются лишь решения для полных m -арных деревьев.

Таким образом, в аспекте исследования рекурсивных алгоритмов очевидный интерес представляет задача теоретического анализа порождённых деревьев рекурсии в случаях, отличных от тривиальных.

2. Постановка задачи

В соответствии с методом анализа трудоёмкости на основе подсчёта вершин порождённого дерева рекурсии [1, стр. 181-185.] будем использовать далее следующие обозначения:

— n — глубина дерева рекурсии, т. е. число его уровней, при этом корень дерева не нумеруется, а имеет специальный индекс Root, а нумерация далее начинается с нуля;

— S — формальное описание структуры порожденного дерева рекурсии;

— $R_A(n,S)$ — суммарное число вершин по всем уровням дерева рекурсии глубины n, имеющего структуру с описанием S;

— $R_V(n,S)$ — суммарное число внутренних вершин дерева;

— $R_L(n,S)$ — число листьев дерева рекурсии, т. е. число вершин на уровне n.

Настоящая статья преследует две цели:

1. Предложить формальное описание дерева рекурсии — S с регулярной структурой, характерной для порождающих их рекурсивных алгоритмов, в том числе для алгоритмов, реализующих метод динамического программирования.

2. Разработать метод решения, позволяющий по данному формальному описанию структуры дерева S и заданному номеру уровня j получить аналитическое решение для числа вершин дерева на данном уровне — $R(j,S)$. Таким образом, если данный вход алгоритма порож-

дает дерево глубины n со структурой S, то основные характеристики дерева, необходимые для анализа трудоёмкости рекурсивного алгоритма вычисляются в виде

$$\begin{aligned} R_A(n,S) &= \sum_{j=0}^{n} R(j,S) + 1; \\ R_V(n,S) &= \sum_{j=0}^{n-1} R(j,S) + 1; \\ R_L(n,S) &= R(n,S). \end{aligned} \tag{1}$$

Заметим, что дополнительная единица учитывает вершину корня дерева рекурсии.

3. Формальное описание структуры регулярного дерева рекурсии

Начнём формирование описания структуры с рассмотрения полного m-арнного дерева. Очевидно, что в этом случае на каждом уровне (кроме уровня листьев) каждая вершина порождает ровно m новых вершин, и дерево полностью самоподобно, начиная с любой вершины. Поскольку особенности ряда рекурсивных алгоритмов состоят в том, что вершины одного уровня порождают различное число вершин следующего уровня, не превышающее однако некоторого наперёд заданного значения, авторы предлагают следующее формальное описание регулярного дерева в виде упорядоченного m-мерного кортежа (строки S) с отсортированными по не возрастанию значениями компонент

$$\begin{aligned} &S = (k_1; k_2; \ldots; k_j; \ldots; k_m), j = \overline{1,m}; \\ &k_1 \geq k_2 \geq \ldots \geq k_j \geq \ldots \geq k_m. \end{aligned} \tag{2}$$

Порождение дерева, со структурой, описываемой строкой S происходит по следующему алгоритму, использующему специальную нумерацию вершин в уровне и понятие типа вершины. На уровне корня дерева вершина **Root** порождает m вершин уровня $n = 0$ с нумерацией $1, 2, \ldots, m$. Присвоим каждой вершине характеристику, которую будем называть «тип вершины» — число из сегмента $[1, \ldots, m]$. Будем говорить далее, что если вершина имеет тип "j", то она порождает k_j вершин на следующем уровне дерева. На уровне $n = 0$ порождённые корнем вершины получают тип, равный собственному номеру, т. о. вершина с номером j имеет тип "j", и, следовательно порождает k_j вершин на уровне 1. Далее, на следующих уровнях, порождённые вершины нумеруются относительно родительской вершины начиная с номера m, справа налево в порядке убывания номеров, и получают типы, соответствующие полученным номерам. Таким образом, вершины, непосредственно порождённые из вершины с типом "j" (таких вершин k_j) описываются следующей строкой

$$(0; 0; \ldots; k_l; k_{l+1}; \ldots; k_m), l = m - k_j + 1. \tag{3}$$

Предложенное формальное описание в виде строки S (2) задаёт регулярное дерево, т. е. дерево, подчиняющееся единой на всех уровнях закономерности порождения вершин, заданной типом вершины в соответствии с её нумерацией, и способом порождения, указанным в формуле (3). Отметим, что в случае если $k_1 = m$, то корень имеет тип "1" и строка S имеет вид $S = \left(m; k_2; \ldots; k_j, \ldots; k_m\right)$, и описывает регулярное неполное (усечённое) m-арное дерево.

Например формальное описание $S = (3;\, 3;\, 3)$ задаёт полное тернарное дерево, а описание $S = (3;\, 3;\, 2)$ — регулярное неполное тернарное дерево, вид которого приведён на рисунке 1. В вершинах дерева указаны их номера, совпадающие, в соответствии с описанным правилом нумерации, с присвоенными им типами. Так, из всех вершин с типом "3" порождаются на следующем уровне две вершины, т. к. в строке формального описания $S = (3;\, 3;\, 2)$ значение $k_3 = 2$. Заметим также, что уровень корня дерева имеет индекс **Root,** и не нумеруется, следующие уровни дерева имеют номера 0, 1 и т. д. Отметим, что введённый тип вершины соответствует её номеру в строке формального описания $S = \left(k_1; k_2; \ldots; k_j; \ldots; k_m\right)$, и из вершины с типом "$j$" порождается k_j вершин, строка описания которых указана в формуле (3).

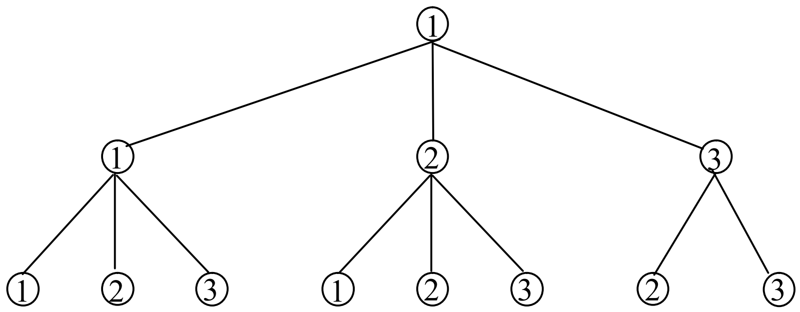

Рис. 1. Фрагмент дерева, заданного описанием $S = (3;\, 3;\, 2)$ *с указанием типов вершин.*

4. Метод аналитического решения задачи о числе вершин регулярного дерева

Не теряя общности, сформулируем задачу подсчёта числа вершин, как задачу определения числа листьев дерева глубины n, порождённого описанием $S = \left(k_1; k_2; \ldots; k_j; \ldots; k_m\right)$, т. е. числа вершин регулярного дерева на уровне n — уровне листьев дерева. Во введённых обозначениях мы решаем задачу определения $R(j, S)$ при $j = n$, т. е. задачу получения аналитического решения для $R(n, S) = R_L(n, S)$.

Введем в рассмотрение характеристику $F(j)$ вершины, имеющей тип "j". Содержательно $F(j)$ есть число типов вершин, порождающих тип "j" по описанию S. Значение $F(j)$ вычисляется по формуле

$$F(j) = \max_{1 \le i \le m} \left\{\, i : m - k_i + 1 \le j \,\right\}. \tag{4}$$

Пусть $w_{n,j}$ есть число вершин, имеющих тип "j" на n-ом уровне дерева. Введём в рассмотрение вектор $W_n = \left\{w_{n,1}; w_{n,2}; \ldots; w_{n,m}\right\}$ размерности m, таким образом, компоненты вектора W_n содержат число вершин каждого типа на уровне n дерева рекурсии. Тогда, с учетом введённой характеристики $F(j)$, для вычисления значений $w_{n,j}$ получаем систему линейных рекуррентных соотношений

$$\begin{cases} w_{\mathrm{Root},1} = 1, \quad w_{\mathrm{Root},j} = 0,\ j = \overline{2,m}, \quad n = \mathrm{Root}; \\ w_{0,j} = 1,\ j = \overline{1,m}, \quad n = 0; \\ w_{n+1,j} = \sum\limits_{i=1}^{F(j)} w_{n,i}, \quad n \ge 1. \end{cases} \tag{5}$$

Заметим, что в соответствии с (5) $W_0 = \{1; 1; \ldots; 1\}$, а если $k_1 = m$, то $W_{\mathrm{Root}} = \{1; 0; \ldots; 0\}$.

На основании известного метода решения линейных рекуррентных соотношений [1, глава 1] будем искать решения соотношения (5) вида

$$H_n = \lambda^n H_0,\ H_0 = \{h_{0,1}; h_{0,2}; \ldots; h_{0,m}\},\ H_n = \{h_{n,1}; h_{n,2}; \ldots; h_{n,m}\}, \tag{6}$$

где H_n, H_0 — некоторые вектора размерности m, а значения λ определяются из условия равенства нулю определителя следующей системы уравнений, рассматриваемой относительно неизвестных $h_{0,j}$.

$$\left\{ \lambda \cdot h_{0,j} = \sum_{i=1}^{F(j)} h_{0,i}, \quad \forall j = \overline{1,m}. \right. \tag{7}$$

Искомое решение нашей задачи — вектор W_n получается линейной комбинацией найденных решений (6) с учётом начальных условий из (5).

Сформулируем этапы предложенного метода для решения рекуррентного соотношения (5) в случае, когда число собственных векторов H_0 равно m.

1. Вычисление характеристик типов вершин $F(j)$ по формуле (4) на основе предложенного формального описания регулярного дерева рекурсии в виде строки S (2).

2. Построение системы линейных рекуррентных соотношений (5).

3. Получение решений вида

$$H_n = \lambda^n H_0.$$

3.1. Построение системы уравнений (7).

3.2. Определение значений λ из условия равенства нулю определителя системы (7).

3.3. Нахождение значений собственных векторов H_0 для каждого значения λ.

4. Построение решений исходной системы (5) — W_n как линейной комбинации полученных решений для H_0 на основе определения неизвестных коэффициентов линейной комбинации по значениям начальных условий, заданных вектором W_0.

5. Подсчёт числа вершин на уровне n исследуемого дерева рекурсии

$$R(n,S)=\sum_{j=1}^{m} w_{n,j}\ . \tag{8}$$

6. Поскольку полученное аналитическое решение для W_n справедливо для любого значения n, то формула (8) доставляет решение и для $R(j,S)\ \forall j=\overline{0,n}$, что даёт возможность определения полного числа порожденных вершин на всех уровнях дерева глубины n и числа внутренних вершин в соответствии с формулой (1).

Особо отметим, что в случае, когда число собственных векторов H_0 меньше, чем m, необходимо использовать известные [1, 3, 4] специальные методы получения решений.

5. Примеры анализа регулярных деревьев рекурсии

Ниже мы приводим примеры анализа регулярных деревьев как для случая, когда число собственных векторов H_0 равно m, так и для случая, когда это число меньше, чем m.

а) Вначале рассмотрим пример дерева, заданного описанием $S=(3;\,3;\,2)$, вид которого приведён на рисунке 1. Заранее укажем, что в этом случае число собственных векторов H_0 равно m. Мы подробно иллюстрируем метод, непосредственно указывая номера и содержание этапов предложенного метода решения.

1. Вычисление $F(j)$ по описанию $S=(3;\,3;\,2)$:

$$F(1)=2,\ F(2)=3,\ F(3)=3\ .$$

2. Построение системы линейных рекуррентных соотношений (5):

$$\begin{cases} w_{0,j}=1,\ j=\overline{1,3}; \\ w_{n+1,1}=w_{n,1}+w_{n,2}; \\ w_{n+1,2}=w_{n,1}+w_{n,2}+w_{n,3}; \\ w_{n+1,3}=w_{n,1}+w_{n,2}+w_{n,3}. \end{cases}$$

3.1. Построение системы уравнений (7):

$$\begin{cases} \lambda h_{0,1}=h_{0,1}+h_{0,2}; \\ \lambda h_{0,2}=h_{0,1}+h_{0,2}+h_{0,3}; \\ \lambda h_{0,3}=h_{0,1}+h_{0,2}+h_{0,3}. \end{cases} \tag{9}$$

3.2. Определение значений λ из условия равенства нулю определителя системы:

$$\begin{vmatrix} 1-\lambda & 1 & 0 \\ 1 & 1-\lambda & 1 \\ 1 & 1 & 1-\lambda \end{vmatrix}=0\,.$$

Вычисляя определитель, получаем уравнение — $\lambda\cdot\left(\lambda^2-3\lambda+1\right)=0$, корни которого равны:

$$\lambda_1=0;\quad \lambda_2=\frac{3-\sqrt{5}}{2};\quad \lambda_3=\frac{3+\sqrt{5}}{2}\,.$$

3.3. Нахождение значений собственных векторов H_0 для каждого значения λ.

Подставляя полученные корни в (9), и решая полученные системы, получим собственные вектора H_0, соответствующие по нумерации корням характеристического уравнения:

$$H_0^{(1)} = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}; \quad H_0^{(2)} = \begin{pmatrix} 1 \\ \frac{1-\sqrt{5}}{2} \\ \frac{1-\sqrt{5}}{2} \end{pmatrix}; \quad H_0^{(3)} = \begin{pmatrix} 1 \\ \frac{1+\sqrt{5}}{2} \\ \frac{1+\sqrt{5}}{2} \end{pmatrix}.$$

4. Построение решений исходной системы (5) — W_n, как линейной комбинации полученных решений для H_0 с начальными условиями W_0:

$$W_n = A \cdot 0^n \cdot H_0^{(1)} + B \cdot \left(\frac{3-\sqrt{5}}{2} \right)^n \cdot H_0^{(2)} + C \cdot \left(\frac{3+\sqrt{5}}{2} \right)^n \cdot H_0^{(3)},$$

где, по определению $0^0 = 1$, а A, B, C — неизвестные коэффициенты линейной комбинации, определяемые по начальным условиям W_0:

$$\begin{cases} A + B + C = 1; \\ -A + \frac{1-\sqrt{5}}{2} B + \frac{1+\sqrt{5}}{2} C = 1; \\ \frac{1-\sqrt{5}}{2} B + \frac{1+\sqrt{5}}{2} C = 1. \end{cases}$$

Решение полученной системы имеет вид:

$$A = 0; \quad B = \frac{\sqrt{5}-1}{2\sqrt{5}}; \quad C = \frac{\sqrt{5}+1}{2\sqrt{5}},$$

что доставляет окончательное решение для числа вершин исследуемого регулярного дерева

$$W_n = \frac{\sqrt{5}-1}{2\sqrt{5}} \cdot \left(\frac{3-\sqrt{5}}{2} \right)^n \cdot H_0^{(2)} + \frac{\sqrt{5}+1}{2\sqrt{5}} \cdot \left(\frac{3+\sqrt{5}}{2} \right)^n \cdot H_0^{(3)}.$$

5. Подсчёт числа вершин на уровне n исследуемого дерева рекурсии.

В соответствии с (8) общее число вершин на уровне n есть сумма числа вершин каждого типа:

$$R(n, S) = \sum_{j=1}^{3} w_{n,j},$$

и, выполняя необходимые преобразования, получаем искомое решение для $S = (3; 3; 2)$:

$$R(n, S) = \frac{7+3\sqrt{5}}{2\sqrt{5}} \cdot \left(\frac{3+\sqrt{5}}{2} \right)^n + \frac{3\sqrt{5}-7}{2\sqrt{5}} \cdot \left(\frac{3-\sqrt{5}}{2} \right)^n.$$

Непосредственной подстановкой можно проверить, что $R(1, S) = 8$, $R(2, S) = 21$.

б) Второй пример в котором регулярное дерево задаётся описанием $S = (3;3;1)$, также приводит к трём различным собственным векторам. В данном случае мы приведём только основные результаты. Значения λ, определённые из условия равенства нулю определителя системы оказываются равными

$$\lambda_1 = 0; \quad \lambda_2 = 1; \quad \lambda_3 = 2.$$

что приводит к следующему решению для числа вершин с соответствующими типами

$$w_{n,1} = 2^n; \quad w_{n,2} = 2^n; \quad w_{n,3} = 2 \cdot 2^n - 1.$$

Выполняя суммирование по всем типам, получаем решение для уровня дерева

$$R(n,S) = 2^{n+2} - 1.$$

в) Следующий пример порождает случай, для которого необходим специальный метод решения. Рассмотрим описание $S = (m;\, m-1;\, m-2;\ \ldots;\, 1)$. Дерево, представленное на рисунке 2, соответствует этому описанию при значении $m = 3$ — $S = (3;\, 2;\, 1)$.

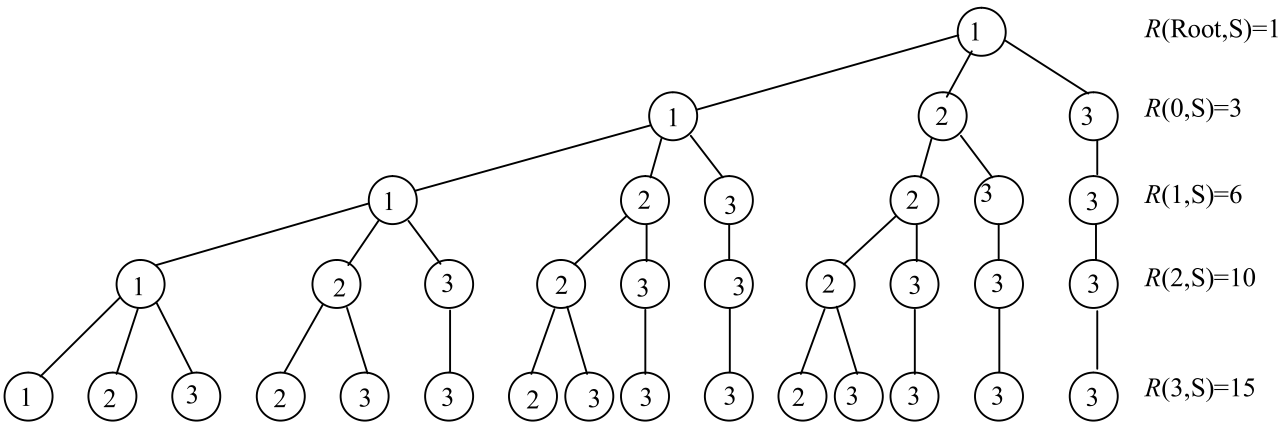

Рис. 2. Фрагмент дерева, заданного описанием $S = (3;\, 2;\, 1)$ с указанием типов вершин.

Начальные этапы аналитического решения для этого случая совпадают с уже описанным методом:

1. Вычисление $F(j)$ по описанию $S = (m;\, m-1;\, m-2;\ \ldots;\, 1)$:

$$F(1) = 1,\ F(2) = 2,\ F(3) = 3, \ldots, F(m) = m.$$

2. Построение системы линейных рекуррентных соотношений (5):

$$\begin{cases} w_{0,j} = 1,\ j = \overline{1,3}; \\ w_{n+1,1} = w_{n,1}; \\ w_{n+1,2} = w_{n,1} + w_{n,2}; \\ \ldots \\ w_{n+1,j} = w_{n,1} + w_{n,2} + \ldots + w_{n,j}; \\ \ldots \\ w_{n+1,m} = w_{n,1} + w_{n,2} + w_{n,3} + \ldots + w_{n,m}; \end{cases} \tag{10}$$

3.1. Построение системы уравнений (7):

$$\begin{cases} \lambda h_{0,1} = h_{0,1}; \\ \lambda h_{0,2} = h_{0,1} + h_{0,2}; \\ \lambda h_{0,3} = h_{0,1} + h_{0,2} + h_{0,3}; \\ \dots \\ \lambda h_{0,m} = h_{0,1} + h_{0,2} + h_{0,3} + \ldots + h_{0,m}. \end{cases} \tag{11}$$

3.2. Определение значений λ из условия равенства нулю определителя системы:

Вычисляя определитель системы (11), получаем уравнение — $(1-\lambda)^m = 0$, и, следовательно корень $\lambda = 1$ имеет кратность m. В данной задаче число собственных векторов меньше, чем m, и мы используем специальный метод, который в данном случае предусматривает непосредственное решение системы (10).

4. Непосредственное решение системы (10).

4.1. Решим рекуррентное соотношение для $j = 1$:

$$\begin{cases} w_{0,1} = 1; \\ w_{n+1,1} = w_{n,1}. \end{cases} \Rightarrow \forall n \; w_{n,1} = 1.$$

4.2. На основании полученного решения имеем рекуррентное соотношение для $j = 2$:

$$\begin{cases} w_{0,2} = 1; \\ w_{n+1,2} = w_{n,2} + 1. \end{cases} \Rightarrow \forall n \; w_{n,2} = n + 1.$$

4.3. С учетом этого решения рекуррентное соотношение для $j = 3$ имеет вид

$$\begin{cases} w_{0,3} = 1; \\ w_{n+1,3} = w_{n,3} + (n+1) + 1. \end{cases} \tag{12}$$

Ищем частное решение в виде $w^*_{n,3} = An^2 + Bn + C$, и, подставляя его в (12), получаем

$$2An + A + B = n + 2, \quad \Rightarrow \quad A = \frac{1}{2}, B = \frac{3}{2},$$

начальное условие доставляет решение $C = 1$, и окончательно

$$w_{n,3} = \frac{1}{2}n^2 + \frac{3}{2}n + 1 = \frac{(n+1)(n+2)}{2}.$$

4.4. Заметим, что $w_{n,1} = 1 = C_n^0$, $w_{n,2} = n + 1 = C_{n+1}^1$, $w_{n,3} = C_{n+2}^2$. Докажем по индукции, что для произвольного значения m

$$w_{n,m} = C_{n+m-1}^{m-1}. \tag{13}$$

Соотношение (13) верно при $m = 1, m = 2, m = 3$. Предположение индукции состоит в том, что соотношение $w_{n,k} = C_{n+k-1}^{k-1}$ верно для всех $k \le m$. Тогда на основе продолжения системы линейных рекуррентных соотношений (10) для $m + 1$ получаем рекуррентное соотношение

$$\begin{aligned} w_{n+1,m+1} &= w_{n,1} + w_{n,2} + w_{n,3} + \ldots + w_{n,m} + w_{n,m+1} = \\ &= C_n^0 + C_{n+1}^1 + C_{n+2}^2 + \ldots + C_{n+m-1}^{m-1} + w_{n,m+1}. \end{aligned} \tag{14}$$

Заметим, что $C_n^0 = C_{n+1}^0$, $C_n^k + C_n^{k-1} = C_{n+1}^k$ [3], тогда

$$C_n^0 + C_{n+1}^1 + C_{n+2}^2 + \ldots + C_{n+m-1}^{m-1} = C_{n+2}^1 + C_{n+2}^2 + \ldots + C_{n+m-1}^{m-1} = C_{n+m}^{m-1}.$$

Покажем, что

$$w_{n,m+1} = C_{n+m}^m$$

является решением рекуррентного по n соотношения (14). Подставляя это предполагаемое решение в (14) получаем

$$w_{n+1,m+1} = C_{n+m}^{m-1} + C_{n+m}^m = C_{n+m+1}^m,$$

что соответствует (13) с подстановкой соответствующих индексов, таким образом, решение системы (10) $\forall m \geq 1, \forall n \geq 1$ имеет вид

$$w_{n,m} = C_{n+m-1}^{m-1},$$

что и доказывает предположение индукции (13).

На основании полученных решений определим общее число вершин на уровне дерева с номером n, суммируя число вершин всех типов на этом уровне. Мы вводим третий аргумент m в функцию числа вершин, т. к. рассматриваем обобщённое описание $S = (m;\, m-1;\, m-2;\; \ldots;\, 1)$ и, используя известные соотношения для биномиальных коэффициентов [3], получаем:

$$R(n,m,S) = \sum_{j=1}^{m} w_{n,j} = \sum_{j=1}^{m} C_{n+j-1}^{j-1} = C_{n+m}^{m-1}. \tag{15}$$

На основе (15) определим общее число вершин дерева с описанием $S = (m;\, m-1;\, m-2;\; \ldots;\, 1)$, суммируя, очевидно, число вершин на всех уровнях, включая корень дерева

$$R_A(n,m,S) = 1 + \sum_{i=0}^{n} R(i,m,S) = \sum_{i=0}^{n} C_{i+m}^{m-1} = C_{n+m+1}^m. \tag{16}$$

Обращаясь к фрагменту дерева, представленному на рисунке 2 с описанием $S = (3;\, 2;\, 1)$, т. е. с $m = 3$, получим общее число вершин на уровне $n = 3$, используя формулу (15) — $R(3,\, 3,\, S) = C_{3+3}^{3-1} = C_6^2 = 15$, и суммарное число вершин на всех уровнях, включая уровень $n = 3$ на основе формулы (16) — $R_A(3,\, 3,\, S) = C_{3+3+1}^3 = C_7^3 = 35$. Отметим, что полученное решение совпадает с результатами анализа порождённого дерева рекурсии для алгоритма решения задачи оптимальной одномерной упаковки методом динамического программирования при регулярной параметризации задачи [1]. При этом результат, приведённый в [1] является частным, по отношению к результатам данной статьи, поскольку доставляет решение только для деревьев с описанием вида $S = (m;\, m-1;\, m-2;\; \ldots;\, 1)$.

6. Заключение

В целях анализа и исследования рекурсивных алгоритмов в статье предложено специальное описание регулярных деревьев рекурсии, характерных для ряда рекурсивных алгоритмов. Такие деревья порождаются, например, алгоритмами, рекурсивно реализующими метод динамического программирования при определенной параметризации, регуляризирующей деревья, порождаемые рекурсивными алгоритмами. На базе этого описания предложен метод решения, доставляющий число порожденных вершин на каждом уровне такого регулярного дерева. Предложенный метод позволяет получить аналитическое решение для числа порожденных вершин по каждому уровню дерева рекурсии, и, следовательно, общее число вершин как функцию глубины дерева.

Полученные результаты могут быть использованы в целях детального теоретического анализа рекурсивных алгоритмов, порождающих регулярные деревья рекурсии, для получения точных функций трудоемкости и емкостной эффективности.

Библиографический список

1. Головешкин В. А., Ульянов М.В. Теория рекурсии для программистов. — М.: ФИЗМАТЛИТ. 2006. 296 с.

2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-ое издание: Пер. с англ. — М.: Издательский дом «Вильямс», 2005. — 1296 с.

3. Грэхем Р. , Кнут Д., Паташник О. Конкретная математика. Основание информатики. Пер. с англ. — М.: Мир. 1998. 703 с.

4. Сачков В. Н. Введение в комбинаторные методы дискретной математики. — 2-ое изд., испр. и доп. — М.: МЦНМО, 2004. — 424 с.