

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ СПЕЦИАЛЬНОГО КЛАССА РЕКУРРЕНТНЫХ СООТНОШЕНИЙ В ЦЕЛЯХ АНАЛИЗА РЕКУРСИВНЫХ АЛГОРИТМОВ

© 2008 В.А. Головешкин¹, М.В. Ульянов²

В статье предлагается аналитическое решение для специального класса нелинейных рекуррентных соотношений. Исследуемые рекуррентные соотношения характерны для функций трудоемкости рекурсивных алгоритмов, разработанных методом декомпозиции и обладающих линейной трудоемкостью объединения полученных решений. Аналитические решения получены для двух подклассов, возникающих при теоретическом рассмотрении исследуемого класса рекуррентных соотношений. Результаты могут быть использованы для получения явных функций трудоемкости рекурсивных алгоритмов, декомпозирующих решаемую задачу с линейной трудоемкостью объединения результатов.

Ключевые слова: рекурсивное соотношение, рекурсивный алгоритм, метод декомпозиции, метод вычислений, функция трудоемкости.

1. Предварительные замечания

Проблема рационального выбора того или иного алгоритма решения некоторой задачи в заданном сегменте длин входа может быть решена на основе исследования и сравнительного анализа ресурсных характеристик претендующих алгоритмов. Одной из наиболее часто используемых характеристик алгоритма является его временная эффективность. При теоретическом исследовании алгоритма аналогом временной эффективности является функция трудоемкости, под которой далее, следуя [1], будем понимать число базовых операций принятой модели вычислений, задаваемых алгоритмом на конкретном входе. При этом очевидный интерес представляет

¹Головешкин Василий Адамович (nikshevolog@yandex.ru), кафедра высшей математики Московского государственного университета приборостроения и информатики, 107475, Россия, г. Москва, ул. Стромынка, 20.

²Ульянов Михаил Васильевич (muljanov@mail.ru), кафедра прикладной математики и моделирования систем Московского государственного университета печати, 127550, Россия, г. Москва, ул. Прянишникова, 2а.

задача получения функции трудоемкости в явном виде (аналитическое решение) на основе теоретического анализа алгоритма для входов фиксированной длины. Эта задача актуальна и для рекурсивных алгоритмов, разработанных методом декомпозиции. Цель настоящей статьи — предложить аналитическое решение этой задачи в одном классе рекуррентных соотношений, описывающих функцию трудоемкости рекурсивных декомпозирующих алгоритмов.

Основная идея метода декомпозиции состоит в сведении исходной задачи к ряду более простых задач с понижением их размерности [2, 3], при этом под размерностью задачи понимается некоторая мера длины входа алгоритма, решающего данную задачу. Общая схема метода предполагает выполнение следующей последовательности шагов:

- 1) Шаг разделения задачи. На этом шаге выбирается способ разделения задачи на некоторое число подзадач меньшей размерности.
- 2) Шаг решения полученных подзадач. Это рекурсивный шаг — каждая подзадача рассматривается как задача определенной размерности, и выполняется ее разделение на собственные подзадачи до тех пор, пока рекурсивно не будет получена такая размерность, при которой решение может быть получено непосредственно.
- 3) Шаг останова рекурсии. На этом шаге выполняется непосредственное решение полученных подзадач для малых размерностей.
- 4) Шаг объединения решений. На этом шаге полученные решения подзадач меньшей размерности при возврате в точку рекурсивного вызова объединяются в решение задачи текущей размерности. Подъем вверх по цепочке рекурсивных возвратов позволяет на верхнем уровне дерева рекурсии получить решение исходной задачи.

На основе этой схемы можно получить общий вид рекуррентного соотношения, которое задает функцию трудоемкости такого рекурсивного алгоритма. Пусть n — размерность решаемой задачи, тогда если в рекурсивном алгоритме, при решении задачи размерности n происходит такое ее разделение, которое приводит к необходимости решения a подзадач размерностью n/b , и b является делителем n , то функция трудоемкости такого алгоритма $f_A(n)$ имеет вид [3]:

$$f_A(n) = a \cdot f_A(n/b) + d(n) + U(n), \quad (1.1)$$

где $d(n)$ — трудоемкость фрагмента алгоритма, выполняющего разделение задачи на подзадачи, $U(n)$ — трудоемкость фрагмента алгоритма, объединяющего полученные решения подзадач размерностью n/b в решение задачи размерности n . При этом на шаге останова рекурсии для некоторой малой размерности задачи, т.е. при $n = n_0$, возможно ее прямое (не рекурсивное) решение. Обозначая трудоемкость получения этого прямого решения через $f_A(n_0)$ и вводя обозначение $g(n) = d(n) + U(n)$, получаем, с учетом (1.1),

общий вид рекуррентного соотношения для функции трудоемкости алгоритмов, разработанных методом декомпозиции в случае, если b является делителем b [3]:

$$\begin{cases} f_A(n_0), & n = n_0; \\ f_A(n) = a \cdot f_A(n/b) + d(n) + U(n), & n > n_0. \end{cases} \quad (1.2)$$

Отметим, что получение аналитического решения соотношения (1.2) осложняется тем, что размерность решаемой задачи должна, очевидно, быть целой. Поэтому в общем случае вместо n/b в качестве аргумента функции в (1.2) должна фигурировать целая часть частного с округлением вниз или вверх, т.е. $\lfloor n/b \rfloor$ или $\lceil n/b \rceil$.

Изучение рекуррентных соотношений вида (1.2) привело к получению ряда результатов: как аналитических решений для некоторых частных случаев, например для $b = 2$ и линейной функции $g(n)$ [4], так и асимптотических оценок функции $f_A(n)$ при $n \rightarrow \infty$. Применение основной теоремы о рекуррентных соотношениях (J.L. Bentley, Dorothea Haken, J.B. Saxe, 1980, [2]) позволяет получить асимптотическое поведение функции $f_A(n)$ при $n \rightarrow \infty$ для достаточно широкого класса функций $g(n)$ при произвольных a и b . Формулировка теоремы в обозначениях из [2] (T — обозначение временной сложности алгоритма) имеет вид.

Теорема. Пусть $a \geq 1$ и $b > 1$ — константы, $g(n)$ — известная функция, $T(n)$ определено при неотрицательных значениях n формулой

$$T(n) = aT(\lfloor n/b \rfloor) + g(n), \quad (1.3)$$

где под $\lfloor n/b \rfloor$ понимается $\lfloor n/b \rfloor$ или $\lceil n/b \rceil$, тогда:

- 1) если $g(n) = O(n^{\log_b a - \epsilon})$ для некоторого $\epsilon > 0$, то $T(n) = \Theta(n^{\log_b a})$;
- 2) если $g(n) = O(n^{\log_b a})$, то $T(n) = \Theta(n^{\log_b a} \log_b n)$;
- 3) если найдутся константы $c > 0$ и $\epsilon > 0$ такие, что при достаточно больших n выполнено $g(n) > cn^{\log_b a + \epsilon}$ и найдется положительная константа $d > 1$ такая, что при достаточно больших n выполнено $ag(n/b) \leq dg(n)$, то $T(n) = \Theta(g(n))$.

Очевидно, что для функций $g(n)$, принадлежащих к определенным классам, могут быть получены не только асимптотические оценки, но и аналитические решения. Одно из таких аналитических решений для специального класса рекуррентных соотношений и предлагается в настоящей статье.

2. Постановка задачи

Рассмотрим специальный класс рекуррентных соотношений вида (1.2) с линейной функцией $g(n)$, описывающих функцию трудоемкости алгоритмов, разработанных методом декомпозиции задачи размерности n на p подзадач размерностью $\lfloor n/k \rfloor$, где p, k — целые положительные числа: $p \geq 2$,

$k \geq 2$. Поскольку под $[n/k]$ понимается $\lfloor n/k \rfloor$ или $\lceil n/k \rceil$, то в рамках изучаемого специального класса рассматриваются два подкласса рекуррентных соотношений, а именно:

I. Подкласс рекуррентных соотношений вида

$$\begin{cases} T(0) = C, & n = 0; \\ T(n) = pT(\lfloor n/k \rfloor) + An + B, & n \geq 1. \end{cases} \quad (2.1)$$

II. Подкласс рекуррентных соотношений вида

$$\begin{cases} T(0) = 0, & n = 0; & T(1) = C, & n = 1; \\ T(n) = pT(\lceil n/k \rceil) + An + B, & n > 1. \end{cases} \quad (2.2)$$

Соотношения (2.1) и (2.2) являются частными случаями (1.3) с функцией $g(n) = An + B$ и описывают функцию трудоемкости рекурсивного алгоритма, использующего метод декомпозиции с линейной трудоемкостью разделения и объединения решений. Отметим, что такие соотношения характерны для функции трудоемкости алгоритмов решения задачи сортировки, поиска выпуклого охватывающего контура и ряда других задач [2, 3].

Задача состоит в получении аналитического решения для функции $T(n)$ в каждом подклассе, т.е. в записи функции $T(n)$ в явном виде как функции аргумента n , который содержательно представляет собой длину входа исследуемого алгоритма.

3. Аналитическое решение рекуррентных соотношений подкласса I

Поскольку в правой части соотношения (2.1) аргументом функции является целая часть от деления аргумента n на значение $k - \lfloor n/k \rfloor$, введем в рассмотрение представление числа в системе счисления по основанию k

$$n = a_m k^m + a_{m-1} k^{m-1} + \dots + a_1 k + a_0, \quad (3.1)$$

где a_j : $0 \leq a_j < k$ — целые числа (цифры в системе счисления по основанию k). В начале рассмотрим ряд частных случаев значений аргумента n в рекуррентном соотношении (2.1). Прежде всего отметим, что если $0 \leq n < k$, то из (2.1) непосредственно следует

$$T(n) = pT(0) + An + B = pC + An + B.$$

3.1. Случай $n = k^m$ и $p \neq k$

В этом случае соотношение (2.1) преобразуется к виду

$$T(k^m) = pT(k^{m-1}) + Ak^m + B,$$

и, в частности, при $m = 0, 1, 2$ имеем

$$\begin{aligned} T(1) &= pC + A + B, \\ T(k) &= pT(1) + Ak + B = p^2C + A(p + k) + B(p + 1), \\ T(k^2) &= pT(k) + Ak^2 + B = p^3C + A(p^2 + kp + k^2) + \\ &\quad + B(p^2 + p + 1) = p^3C + A\frac{p^3 - k^3}{p - k} + B\frac{p^3 - 1}{p - 1}. \end{aligned}$$

По индукции положим

$$T(k^m) = p^{m+1}C + A\frac{p^{m+1} - k^{m+1}}{p - k} + B\frac{p^{m+1} - 1}{p - 1}, \quad (3.2)$$

тогда

$$\begin{aligned} T(k^{m+1}) &= p \left[p^{m+1}C + A\frac{p^{m+1} - k^{m+1}}{p - k} + B\frac{p^{m+1} - 1}{p - 1} \right] + \\ &\quad + Ak^{m+1} + B = p^{m+2}C + A\frac{p^{m+2} - k^{m+2}}{p - k} + B\frac{p^{m+2} - 1}{p - 1}, \end{aligned}$$

следовательно, полученное соотношение верно для всех неотрицательных значений m .

Для перехода к исходному аргументу n заметим, что из $n = k^m$ следует $m = \log_k n$. Подстановка $m = \log_k n$ в (3.2) дает аналитическое решение рассматриваемого случая

$$T(n) = p^{1+\log_k n}C + A\frac{p^{1+\log_k n} - k^{1+\log_k n}}{p - k} + B\frac{p^{1+\log_k n} - 1}{p - 1}. \quad (3.3)$$

3.2. Случай $n = k^m$ и $p = k$

В этом случае соотношение (2.1) имеет вид

$$T(k^m) = kT(k^{m-1}) + Ak^m + B,$$

в частности $T(1) = kC + A + B$ и $T(k) = kT(1) + Ak + B = k^2C + 2Ak + B(k + 1)$.

По индукции положим

$$T(k^m) = k^{m+1}C + (m + 1) \cdot k^m A + B\frac{k^{m+1} - 1}{k - 1}, \quad (3.4)$$

тогда

$$\begin{aligned} T(k^{m+1}) &= k \left[k^{m+1}C + (m + 1) \cdot k^m A + B\frac{k^{m+1} - 1}{k - 1} \right] + Ak^{m+1} + B = \\ &= k^{m+2}C + (m + 2) \cdot k^{m+1} A + B\frac{k^{m+2} - 1}{k - 1}, \end{aligned}$$

что доказывает предположение индукции.

Подстановка $n = k^m$ и $m = \log_k n$ в (3.4) дает аналитическое решение

$$T(n) = Ckn + An(\log_k n + 1) + B\frac{kn - 1}{k - 1}. \quad (3.5)$$

Заметим, что при $k = 2$ получаем $T(n) = \Theta(n \log_2 n)$, что, очевидно, согласуется с результатом основной теоремы о рекуррентных соотношениях (см. случай 2 теоремы).

3.3. Случай $n = ak^m$, $1 \leq a < k$ и $p \neq k$

В этом случае соотношение (2.1) принимает вид

$$T(ak^m) = pT(ak^{m-1}) + Aak^m + B.$$

Полагая по индукции

$$T(ak^m) = p^{m+1}C + Aa \frac{k^{m+1} - p^{m+1}}{k - p} + B \frac{p^{m+1} - 1}{p - 1}, \quad (3.6)$$

получаем

$$\begin{aligned} T(ak^{m+1}) &= p \left[p^{m+1}C + Aa \frac{k^{m+1} - p^{m+1}}{k - p} + B \frac{p^{m+1} - 1}{p - 1} \right] + \\ &+ Aak^{m+1} + B = p^{m+2}C + Aa \frac{k^{m+2} - p^{m+2}}{k - p} + B \frac{p^{m+2} - 1}{p - 1}, \end{aligned}$$

что доказывает правильность полученного соотношения для всех значений $m \geq 0$.

Для перехода в данном случае к исходному аргументу n заметим, что из $n = ak^m$ следует $m = \lfloor \log_k n \rfloor$, и $a = \frac{n}{k^m} = k^{\log_k n - m}$, подстановка в (3.6) дает аналитическое решение

$$T(ak^m) = p^{m+1}C + A \frac{n}{k^m} \frac{k^{m+1} - p^{m+1}}{k - p} + B \frac{p^{m+1} - 1}{p - 1}. \quad (3.7)$$

3.4. Случай $n = ak^m$, $0 \leq a < k$ и $p = k$

Опуская предварительные выкладки, полагаем по индукции

$$T(ak^m) = k^{m+1}C + (m+1) \cdot k^m Aa + B \frac{k^{m+1} - 1}{k - 1}, \quad (3.8)$$

тогда

$$\begin{aligned} T(ak^{m+1}) &= k \left[k^{m+1}C + (m+1)k^{m+1}Aa + B \frac{k^{m+1} - 1}{k - 1} \right] + \\ &+ Aak^{m+1} + B = k^{m+2}C + (m+2) \cdot k^{m+1}Aa + B \frac{k^{m+2} - 1}{k - 1}, \end{aligned}$$

что доказывает предположение индукции.

Подстановка $a = \frac{n}{k^m}$ в (3.8) с учетом того, что $m = \lfloor \log_k n \rfloor$, дает аналитическое решение:

$$T(ak^m) = k^{m+1}C + An(m+1) + B \frac{k^{m+1} - 1}{k - 1}. \quad (3.9)$$

3.5. Случай $n = Sk + a$, $0 \leq a < k$, $S > 0$

Заметим, что в этом случае соотношение (2.1) преобразуется к виду

$$T(n) = pT(S) + A(Sk + a) + B = T(Sk) + Aa,$$

далее, полагая, что аргумент функции T имеет вид Sk^m , при $m = 1, 2, 3$, получаем

$$\begin{aligned} T(Sk) &= pT(S) + ASk + B, \\ T(Sk^2) &= p[pT(S) + ASk + B] + ASk^2 + B = \\ &= p^2T(S) + ASk(p+k) + B(p+1), \\ T(Sk^3) &= p^3T(S) + ASk(p^2 + pk + k^2) + B(p^2 + p + 1), \end{aligned}$$

по индукции при $p \neq k$ положим, что

$$T(Sk^m) = p^mT(S) + ASk \frac{p^m - k^m}{p - k} + B \frac{p^m - 1}{p - 1}, \quad (3.10)$$

и предположение доказывается аналогично предыдущим случаям.

Если $p = k$, то предположение индукции о том, что

$$T(Sk^m) = k^{m+1}T(S) + ASk^m m + B \frac{k^{m+1} - 1}{k - 1}, \quad (3.11)$$

также доказываются аналогично ранее рассмотренным случаям.

3.6. Случай $n = ak^m + bk^{m-1}$, $0 \leq a < k$, $0 \leq b < k$ и $p = k$

Предварительно введем следующие обозначения чисел. Исходное число имеет представление $n = a_mk^m + a_{m-1}k^{m-1} + \dots + a_1k + a_0$, в соответствие которому поставим числа n_j , содержательно обозначающие замену нулями последних справа разрядов от 0-го до j -го в k -ичном представлении числа n . В частности,

$$\begin{aligned} n_0 &= a_mk^m + a_{m-1}k^{m-1} + \dots + a_1k, \\ n_1 &= a_mk^m + a_{m-1}k^{m-1} + \dots + a_2k^2, \end{aligned}$$

при этом $T(n) = T(n_0) + Aa_0$ (см. случай 3.5).

Представим число $n = ak^m + bk^{m-1}$ в виде $n = k^{m-1}(ak + b) = Sk^{m-1}$, и на основании формулы (3.10), полагая $S = ak + b$, получим

$$T(n) = p^{m-1}T(ak + b) + A(ak + b)k \frac{p^{m-1} - k^{m-1}}{p - k} + B \frac{p^{m-1} - 1}{p - 1},$$

поскольку $T(ak + b) = T(ak) + Ab$ (см. случай 3.5), то

$$\begin{aligned} T(n) &= p^{m-1}T(ak) + A(ak)k \frac{p^{m-1} - k^{m-1}}{p - k} + B \frac{p^{m-1} - 1}{p - 1} + \\ &+ Abp^{m-1} + Abk \frac{p^{m-1} - k^{m-1}}{p - k} = T(ak^m) + Ab \frac{p^m - k^m}{p - k}. \end{aligned} \quad (3.12)$$

3.7. Случай $n = ak^m + bk^{m-1}$, $0 \leq a < k$, $0 \leq b < k$ и $p = k$

Рассуждая аналогично случаю 3.6 и используя формулу (3.11), имеем

$$T(n) = T(k^{m-1}(ak + b)) = k^mT(ak + b) + A(ak + b)k^{m-1}(m - 1) + B \frac{k^m - 1}{k - 1},$$

что после преобразований дает

$$T(n) = k^m T(ak) + A(ak)k^{m-1}(m-1) + B \frac{k^m - 1}{k-1} + k^m Ab + Abk^{m-1}(m-1),$$

и окончательно

$$T(n) = T(ak^m) + Abk^{m-1}m. \quad (3.13)$$

3.8. Общее решение для подкласса I

Переходя к введенным обозначениям для чисел n_j , на основе формулы (3.12) в предположении, что $p \neq k$, последовательно получаем

$$\begin{aligned} T(n) &= T(n_0) + Aa_0, \\ T(n_0) &= T(n_1) + Aa_1 \frac{p^2 - k^2}{p - k}, \\ T(n_1) &= T(n_2) + Aa_2 \frac{p^3 - k^3}{p - k}, \dots, \end{aligned}$$

и, объединяя полученные выражения, имеем общее аналитическое решение при $p \neq k$

$$T(n) = T(n_{m-1}) + A \sum_{j=0}^{m-1} a_j \frac{p^{j+1} - k^{j+1}}{p - k}, \quad (3.14)$$

заметим, что $n_{m-1} = a_m k^m$, а для $T(a_m k^m)$ уже получено явное аналитическое выражение (см. формулу 3.6), а значения a_j вычисляются на основе представления числа n в соответствии с формулой (3.1).

Для случая $p = k$ используем формулу (3.13) для вычисления $T(n_j)$, в результате имеем аналитическое решение:

$$T(n) = T(n_{m-1}) + A \sum_{j=0}^{m-1} a_j (j+1) \cdot k^j, \quad (3.15)$$

а явное выражение для $T(a_m k^m)$ в этом случае задается формулой (3.7). Тем самым формулы (3.14) и (3.6) доставляют общее аналитическое решение рекуррентных соотношений, принадлежащих подклассу I для случая $p \neq k$, а формулы (3.15) и (3.7) — для случая $p = k$.

4. Аналитическое решение рекуррентных соотношений подкласса II

Поскольку в правой части соотношения (2.2) аргументом функции является целая часть с округлением вверх от деления аргумента n на значение $k - \lceil n/k \rceil$, введем в рассмотрение специальное представление числа n по степеням k

$$n = a_m k^m - a_{m-1} k^{m-1} - \dots - a_1 k - a_0, \quad m = \lceil \log_k n \rceil, \quad m \geq 1, \quad (4.1)$$

где a_j : $0 \leq a_j < k$ — целые числа (цифры в системе счисления по основанию k). В начале рассмотрим ряд частных случаев значений аргумента n в рекуррентном соотношении (2.2). Прежде всего, отметим, что если $1 < n \leq k$, то из (2.2) непосредственно следует

$$T(0) = 0; \quad T(1) = C; \quad T(n) = pT(1) + An + B = pC + An + B. \quad (4.2)$$

4.1. Случай $n = ak^m$, $a > 0$ и $p \neq k$

В этом случае соотношение (2.2) принимает вид

$$T(ak^m) = pT(ak^{m-1}) + Aak^m + B.$$

Полагая по индукции

$$T(ak^m) = p^m T(a) + Aka \frac{p^m - k^m}{p - k} + B \frac{p^m - 1}{p - 1}, \quad (4.3)$$

получаем

$$\begin{aligned} T(ak^{m+1}) &= pT(ak^m) + Aak^{m+1} + B = \\ &= p^{m+1}T(a) + Aka \frac{p^{m+1} - p^m}{p - k} + B \frac{p^{m+1} - p}{p - 1} + Aak^{m+1} + B = \\ &= p^{m+1}T(a) + Aka \frac{p^{m+1} - k^{m+1}}{p - k} + B \frac{p^{m+1} - 1}{p - 1}, \end{aligned}$$

что доказывает правильность соотношения (4.3) для всех значений $m \geq 1$.

4.2. Случай $n = ak^m$, $a > 0$ и $p = k$

Рассуждая аналогично случаю 4.1, полагаем по индукции

$$T(ak^m) = k^m T(a) + Ama \cdot k^m + B \frac{k^m - 1}{k - 1}, \quad (4.4)$$

тогда

$$\begin{aligned} T(ak^{m+1}) &= kT(ak^m) + Aak^{m+1} + B = \\ &= k^{m+1}T(a) + Ama \cdot k^{m+1} + B \frac{k^{m+1} - k}{k - 1} + Aak^{m+1} + B = \\ &= k^{m+1}T(a) + A(m+1)a \cdot k^{m+1} + B \frac{k^{m+1} - 1}{k - 1}. \end{aligned}$$

что доказывает предположение индукции.

4.3. Случай $n = ak^m - b$, $a > 0$, $0 \leq b < k$

В этом случае соотношение (2.2) преобразуется к виду

$$T(ak^m - b) = pT(ak^{m-1}) + Aak^m - Ab + B = T(ak^m) - Ab. \quad (4.5)$$

4.4. Случай $n = k^m (ak^l - b)$, $a > 0$, $0 \leq b < k$ и $p \neq k$

Используя соотношение (2.2) и полученные ранее результаты — формулы (4.3) и (4.5), получаем

$$\begin{aligned} T(k^m(ak^l - b)) &= p^m T(ak^l - b) + Ak(ak^l - b) \frac{p^m - k^m}{p - k} + B \frac{p^m - 1}{p - 1} = \\ &= p^m T(ak^l) - p^m Ab + Aka k^l \frac{p^m - k^m}{p - k} + B \frac{p^m - 1}{p - 1} - Akb \frac{p^m - k^m}{p - k} = \\ &= T(ak^{l+m}) - Ab \frac{p^{m+1} - k^{m+1}}{p - k}. \end{aligned} \quad (4.6)$$

4.5. Случай $n = k^m (ak^l - b)$, $a > 0$, $0 \leq b < k$ и $p = k$

Используя соотношение (2.2) и формулы (4.4) и (4.5), получаем

$$\begin{aligned} T(k^m(ak^l - b)) &= k^m T(ak^l - b) + Am(ak^l - b) \cdot k^m + B \frac{k^m - 1}{k - 1} = \\ &= k^m T(ak^l) - k^m Ab + Amak^{l+m} + B \frac{k^m - 1}{k - 1} - Ambk^m = \\ &= T(ak^{l+m}) - Abk^m(m + 1). \end{aligned} \quad (4.7)$$

4.6. Общее решение для подкласса II

Переходя к введенному представлению числа $n = a_m k^m - a_{m-1} k^{m-1} - \dots - a_1 k - a_0$ и используя формулы (4.5) и (4.6) последовательно получаем:

$$\begin{aligned} T(n) &= T(a_m k^m - a_{m-1} k^{m-1} - \dots - a_1 k - a_0) = \\ &= T(a_m k^m - a_{m-1} k^{m-1} - \dots - a_1 k) - Aa_0 = \\ &= T(a_m k^m - a_{m-1} k^{m-1} - \dots - a_2 k^2) - Aa_0 - Aa_1 \frac{p^2 - k^2}{p - k} = \\ &= \dots, \end{aligned}$$

и окончательно имеем

$$T(n) = T(a_m k^m) - A \sum_{j=0}^{m-1} a_j \frac{p^{j+1} - k^{j+1}}{p - k}. \quad (4.8)$$

Заметим, что для $T(a_m k^m)$ уже получено явное аналитическое выражение — формула (4.3), подстановка которого в (4.8) позволяет получить окончательный результат — общее аналитическое решение рассматриваемого подкласса при $p \neq k$:

$$T(n) = p^m T(a_m) + Aka_m \frac{p^m - k^m}{p - k} + B \frac{p^m - 1}{p - 1} - A \sum_{j=0}^{m-1} a_j \frac{p^{j+1} - k^{j+1}}{p - k}. \quad (4.9)$$

Для случая $p = k$ используем формулу (4.7) для вычисления $T(n)$ и формулу (4.4) для вычисления $T(a_m k^m)$, в результате имеем аналитическое

решение при $p = k$:

$$T(n) = k^m T(a_m) + A m a_m k^m + B \frac{k^m - 1}{p - 1} - A \sum_{j=0}^{m-1} a_j (j+1) k^j. \quad (4.10)$$

Значение $T(a_m)$ в формулах (4.9) и (4.10) вычисляется по формуле (4.2):

$$T(a_m) = pC + A a_m + B.$$

Переход к исходному аргументу n в формулах (4.9) и (4.10) производится подстановкой значения $m = \lceil \log_k n \rceil$ и вычислением значений a_j на основе представления n , задаваемого формулой (4.1).

Заключение

В статье получены аналитические решения для частных и общих случаев нелинейных рекуррентных соотношений специального класса, задаваемого формулами (2.1) и (2.2), возникающих при теоретическом анализе трудоемкости рекурсивных алгоритмов, использующих метод декомпозиции.

Полученные результаты могут быть использованы в целях детального теоретического анализа рекурсивных декомпозирующих алгоритмов, имеющих линейную трудоемкость разделения задачи и объединения полученных решений.

Литература

- [1] Ульянов, М.В. Классификация и методы сравнительного анализа вычислительных алгоритмов / М.В. Ульянов. – М.: Издательство физико-математической литературы, 2004. – 212 с.
- [2] Алгоритмы: построение и анализ / Т.Кормен [и др.]; 2-ое изд. пер. с англ. – М.: Издательский дом "Вильямс", 2005. – 1296 с.
- [3] Головешкин, В.А. Теория рекурсии для программистов / В.А. Головешкин, М.В. Ульянов. – М.: ФИЗМАТЛИТ, 2006. – 296 с.
- [4] Грэхем, Р. Конкретная математика. Основание информатики / Р.Грэхем, Д.Кнут, О.Паташник; пер. с англ. – М.: Мир, 1998. – 703 с.

Поступила в редакцию 2/VI/2008;
в окончательном варианте — 2/VI/2008.

AN ANALYTICAL SOLUTION OF SPECIAL CLASS OF RECURRENCE RELATIONS FOR THE PURPOSE TO ANALYSE RECURSIVE ALGORITHMS

© 2008 Goloveshkin V.A.,³ Ulyanov M.V.⁴

In the paper analytical solution for special class of recurrence relations is proposed. Analysed recurrence relations are characterized for the of labour-intensiveness recurrence relations functions processed by decomposition method and has linear labour-intensiveness of union obtained solutions. Analytical solution for two subclasses arising at theoretical examination of investigated class recurrence relations is obtained. The results can be used to receive explicit functions of labour-intensiveness recurrence algorithms decomposing the task with linear labour-intensiveness of results union.

Keywords: *relation recursive, recursive algorithm, decomposition method, calculation model, labour-intensiveness function.*

Paper received 2/VI/2008.

Paper accepted 2/VI/2008.

³Goloveshkin Vasily Adamovich (nikshevolog@yandex.ru), Dept. of Applied Mathematics, Moscow State University of Instrumental-making and Informatics, Moscow, 107475, Russia.

⁴Ulyanov Mikhail Vasilievich (muljanov@mail.ru), Dept. of Applied Mathematics and System Modeling, Moscow State University of Printing Arts, Moscow, 127550, Russia.