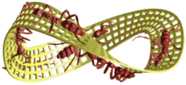
Производящие функции



(дополнительная лекция по ДМ)

Непейвода А. Н.

21 мая 2022

Мастер–теорема («разделяй и властвуй»)

Пусть функция $f(n)$ описывается следующим рекуррентным соотношением:

$$f(n) = a \cdot f(\frac{n}{b}) + g(n)$$

Тогда сложность $f(n)$ можно оценить следующим образом:

- если $g(n) = O(n^c)$, где $c < \log_b a$, тогда $f(n) = \Theta(n^{\log_b a})$;
- если $g(n) = \Theta(n^{\log_b a})$, тогда $f(n) = \Theta(n^{\log_b a} \cdot \log n)$;
- если $g(n) = O(n^c)$, где $c > \log_b a$, причём асимптотически верно, что для некоторой величины $t < 1$ $a \cdot g(\frac{n}{b}) \leqslant t \cdot g(n)$, тогда $f(n) = \Theta(g(n))$.

Сочетание условий $c > \log_b a$ и $a \cdot g(\frac{n}{b}) \leqslant t \cdot g(n)$ выполняется почти всегда.

Экспоненциальные рекуррентности

Классика жанра анализа программ: алгоритм полиномиален по размеру графа, а граф экспоненциален от входных данных. Переход к оценке по размеру входа порождает рекуррентные соотношения уже другой формы, асимптотику которых, однако, также можно оценить с помощью мастер-теоремы.

- Пример оценки:

$$f(n) = 3 \cdot f(n-1) + 2^n$$

Как поступить с двумя рекурсивными вызовами: $f(n) = a_1 \cdot f(n-k_1) + a_2 \cdot f(n-k_2) + g(n)$?

Экспоненциальные рекуррентности

- Частный случай:

$$f(n) = 3 \cdot f(n-1) + 2^n$$

- Общий случай:

$$f(n) = a \cdot f(n-1) + b^n$$

Хак для первой задачи: вводим функцию g такую, что $g(2^n) = f(n)$. Попробуйте взять основание, отличное от 2, и посмотреть, поменяется ли что-то.

Как поступить с двумя рекурсивными вызовами: $f(n) = a_1 \cdot f(n-k_1) + a_2 \cdot f(n-k_2) + g(n)$?

Рабочий пример

Дан недетерминированный конечный автомат $\mathscr{A}$ без ε-переходов над $\{a, b\}$. Известно, что максимальное количество дуг, выходящих из состояния $\mathscr{A}$, равно k (k может быть больше 2, т.к. $\mathscr{A}$ — НКА).

- Найти множество слов длины (меньшей или равной) n;
- Найти количество слов длины (меньшей или равной) n.

- Тупой рекурсивный алгоритм по длине слова – $O(???)$;
- Разбиение на две подзадачи — $O(???)$;
- «Разделяй и властвуй» — $O(???)$.

Рабочий пример

Дан недетерминированный конечный автомат $\mathscr{A}$ без ε-переходов над $\{a, b\}$. Известно, что максимальное количество дуг, выходящих из состояния $\mathscr{A}$, равно k (k может быть больше 2, т.к. $\mathscr{A}$ — НКА).

- Найти множество слов длины (меньшей или равной) n;
- Найти количество слов длины (меньшей или равной) n.

- Тупой рекурсивный алгоритм по длине слова – $O(k^n)$;
- Разбиение на две подзадачи — $O(max(2, \sqrt{k})^n)$;
- «Разделяй и властвуй» — $O(2^n)$. Не зависит от k.

Оценка $O(2^n)$ грубая (предполагает, что $\mathscr{A}$ порождает почти все возможные слова). Более точная оценка связана с решением задачи номер 2.

Комбинаторные задачи

- Сколько существует слов длины 5 в алфавите $\{a, b\}$, не содержащих подстроку ab?
- Сколько существует слов длины n в алфавите $\{a, b\}$, не содержащих подстроку abb? Построить соответствующее рекуррентное соотношение.

Подсказка: рекурсия по количеству слов с соответствующими суффиксами.

Упражнение: слова в алфавите $\{a, b\}$, не содержащие подстроки $aabb$.

Комбинаторные задачи

- Сколько существует слов длины 5 в алфавите $\{a, b\}$, не содержащих подстроку ab?
- Сколько существует слов длины n в алфавите $\{a, b\}$, не содержащих подстроку abb? Построить соответствующее рекуррентное соотношение.

Подсказка: рекурсия по количеству слов с соответствующими суффиксами.

Построенная рекуррентность для задачи 2: $g(n+1) = g(n) + g(n-1) + 1$.

Упражнение: слова в алфавите $\{a, b\}$, не содержащие подстроки $aabb$.

Формальные степенные ряды

$\mathbb{K}$ — какое-нибудь (числовое) поле. Существует взаимно-однозначное соответствие между списками элементов из $\mathbb{K}$ и многочленами с коэффициентами из $\mathbb{K}$. Т.е. представим списки чисел как «многочлены» (с бесконечным числом коэффициентов):

$$\sum_{n=0}^{\infty} a_n x^n$$

Приятная неожиданность

- Покоэффициентное сложение, коммутативное и ассоциативное.
- Умножение, коммутативное и ассоциативное.

Формальные степенные ряды

Производящая функция последовательности $\{a_n\}$ — это формальный степенной ряд:

$$\sum_{n=0}^{\infty} a_n x^n$$

Приятная неожиданность

- Покоэффициентное сложение, коммутативное и ассоциативное.
- Умножение, коммутативное и ассоциативное.
- Расширение алгебры многочленов! Частное и производная.

Значения, принимаемые x — не численные, а комбинаторные.

Краткое представление ряда

Пусть $P(x)$ — полином степени n. Рассмотрим «обращённый» полином $P^R(x) = P(\frac{1}{x}) \cdot x^n$ (тот же самый, но упорядоченный в перевёрнутом порядке).

Что получится, если формально разделить в столбик 1 на $(-k \cdot x + 1)^R$?

Получаем способ свёртки формального ряда (списка) в частное обратных полиномов.

Алгоритмы свёртки

- ПФ для свёртки $\frac{1}{1-k\cdot x}$ — это $\sum_{i=0}^{\infty} k^i x^i$.
- Умножение на n-ую степень x — это сдвиг на n.
- ПФ для свёртки $\frac{P(x)}{Q(x)}$, где $Q(x)$ — многочлен без кратных и мнимых корней, получается с помощью разложения в сумму простых дробей со знаменателями вида $1 - k \cdot x$.

Упражнение: превратить рекуррентность для чисел Фибоначчи в свёртку вида $\frac{1}{P(x)}$ (или подсмотреть в интернете), а потом разделить 1^R на $P(x)^R$ в столбик.

Теорема о рациональных ПФ

Квазиполином — это сумма вида $\sum \beta_i \cdot x^{q_i} \cdot {k_i}^{\alpha_i \cdot x}$.

Теорема

Следующие утверждения относительно последовательности $\{a_n\}$ эквивалентны:

- ПФ $\{a_n\}$ является рациональной (может быть представлена в виде частного полиномов $P(x)/Q(x)$), где $Q(x) = 1 - c_1 \cdot x - \cdots - c_k \cdot x^k$;
- $f(i) = a_i$ описывается квазиполиномом, основания экспонент которого определяются корнями $Q(x)$;
- $\{a_n\}$ описывается линейной рекуррентностью $a_n = c_1 \cdot a_{n-1} + c_2 \cdot a_{n-2} + \cdots + c_k \cdot a_{n-k}$.

Случаи кратных и мнимых корней $Q(x)$ не рассматриваем!

Алгоритм приведения к дроби

Ищем краткое представление для $G = \sum_{j \geqslant 0} a_j x^j$.

Строим уравнения на коэффициенты производящей функции и умножаем на подходящие степени x:

$$
\begin{array}{ll}
a_k = c_1 \cdot a_{k-1} + c_2 \cdot a_{k-2} + \cdots + c_k \cdot a_0 & (\cdot x^0) \\
a_{k+1} = c_1 \cdot a_k + c_2 \cdot a_{k-1} + \cdots + c_k \cdot a_1 & (\cdot x^1) \\
\ldots & \\
a_n = c_1 \cdot a_{n-1} + c_2 \cdot a_{n-2} + \cdots + c_k \cdot a_{n-k} & (\cdot x^{n-k}) \\
\ldots &
\end{array}
$$

Алгоритм приведения к дроби

Ищем краткое представление для $G = \sum_{j \geqslant 0} a_j x^j$.

$$
\begin{aligned}
&a_k = c_1 \cdot a_{k-1} + c_2 \cdot a_{k-2} + \dots + c_k \cdot a_0 && (\cdot x^0) \\
&a_{k+1} = c_1 \cdot a_k + c_2 \cdot a_{k-1} + \dots + c_k \cdot a_1 && (\cdot x^1) \\
&\dots \\
&a_n = c_1 \cdot a_{n-1} + c_2 \cdot a_{n-2} + \dots + c_k \cdot a_{n-k} && (\cdot x^{n-k}) \\
&\dots
\end{aligned}
$$

Теперь складываем все эти уравнения:

$$\sum_{j \geqslant k} a_j x^{j-k} = c_1 \cdot \sum_{j \geqslant k-1} a_j x^{j-k+1} + \dots + c_k \cdot \sum_{j \geqslant 0} a_j x^j$$

Алгоритм приведения к дроби

Ищем краткое представление для $G = \sum_{j\geqslant 0} a_j x^j$.

$$\sum_{j\geqslant k} a_j x^{j-k} = c_1 \cdot \sum_{j\geqslant k-1} a_j x^{j-k+1} + \dots + c_k \cdot \sum_{j\geqslant 0} a_j x^j$$

Чтобы стало возможным подставить G, умножаем сумму на x^k:

$$\sum_{j\geqslant k} a_j x^j = c_1 x \cdot \sum_{j\geqslant k-1} a_j x^j + \dots + c_k x^k \cdot \sum_{j\geqslant 0} a_j x^j$$

Алгоритм приведения к дроби

Ищем краткое представление для $G = \sum_{j \geqslant 0} a_j x^j$.

$$\sum_{j \geqslant k} a_j x^j = c_1 x \cdot \sum_{j \geqslant k-1} a_j x^j + \cdots + c_k x^k \cdot \sum_{j \geqslant 0} a_j x^j$$

Выражаем результат в терминах G и нескольких первых значений a_i:

$$G - \Big(\sum_{j<k} a_j x^j\Big) = c_1 x \cdot \Big(G - \sum_{j<k-1} a_j x^j\Big) + \cdots + c_k x^k \cdot G$$

Итог:

$$G = \frac{\sum_{j<k} a_j x^j - c_1 x \cdot \sum_{j<k-1} a_j x^j - \cdots + c_{k-1} x^{k-1} a_0}{1 - c_1 x - \cdots - c_k x^k}$$

Пример оценки

Оценить скорость роста количества слов длины n, не содержащих abb.

Берём уже известное соотношение и начальные значения g: $g(0) = 1$; $g(1) = 2$; $g(n) = g(n-1) + g(n-2) + 1$. Соотношение имеет дополнительное слагаемое (1), поэтому придётся повторять конструкцию суммы по шагам.

$$\sum_{j \geqslant 2} a_j x^j = x \cdot \sum_{j \geqslant 1} a_j x^j + x^2 \cdot \sum_{j \geqslant 0} a_j x^j + x^2 \cdot \sum_{j \geqslant 0} x^j.$$

Пример оценки

Оценить скорость роста количества слов длины n, не содержащих abb.

Берём уже известное соотношение и начальные значения g: $g(0) = 1$; $g(1) = 2$; $g(n) = g(n-1) + g(n-2) + 1$.

$$\sum_{j\geqslant 2} a_j x^j = x \cdot \sum_{j\geqslant 1} a_j x^j + x^2 \cdot \sum_{j\geqslant 0} a_j x^j + x^2 \cdot \sum_{j\geqslant 0} x^j.$$

Как свернуть последнюю сумму, мы уже знаем, поэтому остальные построения не представляют труда:

$$G - 1 - 2x = x \cdot (G - 1) + x^2 \cdot G + \frac{x^2}{1-x}$$

Пример оценки

Оценить скорость роста количества слов длины n, не содержащих abb.

Берём уже известное соотношение и начальные значения g: $g(0) = 1$; $g(1) = 2$; $g(n) = g(n-1) + g(n-2) + 1$.

$$G - 1 - 2x = x \cdot (G - 1) + x^2 \cdot G + \frac{x^2}{1-x}$$

$$G = \frac{1+x}{1-x-x^2} + \frac{x^2}{(1-x)(1-x-x^2)} = \frac{1}{1-2x+x^3}$$

Пример оценки

Оценить скорость роста количества слов длины n, не содержащих abb.

Берём уже известное соотношение и начальные значения g:
$g(0) = 1$; $g(1) = 2$; $g(n) = g(n-1) + g(n-2) + 1$.

$$G = \frac{1+x}{1-x-x^2} + \frac{x^2}{(1-x)(1-x-x^2)} = \frac{1}{1-2x+x^3}$$

Разделив 1 на $1-2x+x^3$ в столбик, мы действительно можем убедиться, что получаются нужные оценки. Заодно из неоднородной рекуррентности $a_n = a_{n-1} + a_{n-2} + 1$ мы построили однородную: $a_n = 2 \cdot a_{n-1} - a_{n-3}$.
Чтобы оценить рост функции g, нужно найти разложение полинома $1-2x+x^3 = (1-q_1 \cdot x)(1-q_2 \cdot x)(1-q_3 \cdot x)$.
Получаем $q_1 = 1$, $q_2 = \frac{\sqrt{5}+1}{2}$, $q_3 = \frac{1-\sqrt{5}}{2}$. Наибольшее значение — второе, оно и определяет асимптотику.

Пример оценки

Оценить скорость роста количества слов длины n, не содержащих abb.

Чтобы оценить рост функции g, нужно найти разложение полинома $1 - 2x + x^3 = (1 - q_1 \cdot x)(1 - q_2 \cdot x)(1 - q_3 \cdot x)$. Получаем $q_1 = 1$, $q_2 = \frac{\sqrt{5}+1}{2}$, $q_3 = \frac{1-\sqrt{5}}{2}$. Наибольшее значение — второе, оно и определяет асимптотику.

Если стоит задача определить *точное* количество слов длины n, тогда придётся разложить G в сумму простых дробей (см. ниже) и найти коэффициенты A_1, A_2, A_3. Понятно, что для оценки асимптотики g это делать не обязательно.

$$\frac{1}{1 - 2x + x^3} = \frac{A_1}{1 - q_1 \cdot x} + \frac{A_2}{1 - q_2 \cdot x} + \frac{A_3}{1 - q_3 \cdot x}.$$

Связь ПФ и М.- Т.

Оценить асимптотику роста n-ого коэффициента производящей функции, заданной следующей рекуррентностью:

$$f(n) = a \cdot f(n-1) + b^{c \cdot n}$$

Вывести из этой оценки хотя бы один частный случай Мастер–Теоремы.

Теорема о регулярных языках

Определение

ПФ языка L — это формальный ряд вида $\sum\limits_{i=1}^{\infty} n_i x^i$, где n_i — это количество слов языка длины ровно i.

Теорема

Если язык L регулярен, тогда его ПФ является рациональной.

Показать, что язык $\{a^{2^n} \mid n \in \mathbb{N}\}$ не является регулярным.

Рабочий приём: стабилизация рекуррентного отрезка ряда.

Показать, что язык $\{a^{[\log_2 n]+3n} \mid n \in \mathbb{N}\}$ нерегулярен.

ПФ для регулярки

Только для однозначных регулярных выражений!

Алгоритм построения ПФ

Строим свёртку ПФ по следующему алгоритму:

1. Любой букве соответствует x;
2. Альтернативе соответствует сложение, конкатенации — умножение;
3. Если s является ПФ для выражения Φ, тогда $\frac{1}{1-s}$ является ПФ для выражения Φ^*.

Далее разворачиваем свёртку в ряд по вышеописанному алгоритму.

Физический смысл: комбинаторные объекты (см. лекции Станкевича в ИТМО по ДМ для 2 курса) + однозначные сочетания этих объектов.

Пример применения

Оценить асимптотику роста количества слов в $\{a, b\}$, не содержащих подслова abb.

Строим однозначное регулярное выражение для искомых слов:

$$b^*(aa^*b)^*a^*$$

Здесь ещё нужно обосновать, почему это выражение однозначное (это почти очевидно) и почему оно описывает требуемый язык (например, по индукции). Дальше останется тупо применить алгоритм порождения ПФ:

$$G(b^*(aa^*b)^*a^*) = \frac{1}{1-x} \cdot \frac{1}{1-x\cdot\frac{1}{1-x}\cdot x} \cdot \frac{1}{1-x} = \frac{1}{1-x} \cdot \frac{1}{1-x-x^2}.$$

Этот путь намного короче первого, если хорошо владеть техникой построения регулярок.