Московский государственный технический университет
имени Н.Э. Баумана

Факультет «Фундаментальные науки»
Кафедра «Математическое моделирование»

А.Н. Канатников

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Конспект лекций

Для студентов кафедры ИУ9

Москва
2010

11. РЕКУРСИВНЫЕ ФУНКЦИИ

Рекурсивные функции — некоторый класс функций одного или нескольких натуральных аргументов, которые можно получить из некоторого исходного набора функций с помощью определенных операций. Понятие „рекурсивная функция“ — удобное математическое описание класса вычислимых (в том или ином смысле) функций, поскольку дается в привычных в математике терминах операций. В то же время процедура построения рекурсивной функции с помощью последовательности операций сродни построению формулы в том или ином формальном исчислении. Это родство выражают словом „конструктивный“. Конструктивный объект отличается как раз тем, что известно, как он получен в результате конечной процедуры применения простейших операций. Поэтому с точки зрения формализации понятия „алгоритм“ рекурсивные функции ничуть не хуже абстрактных машин или нормальных алгорифмов.

Хотя нормальные алгорифмы и машины Тьюринга строились в произвольном алфавите, с принципиальной точки зрения выбор алфавита не является существенным, а любой алгоритм можно трансформировать в алгоритм с двухбуквенным алфавитом (это строго проверено в рамках нормальных алгорифмов). Это и понятно: множество слов в данном алфавите счетно, так что все слова можно перенумеровать, а любую словарную функцию рассматривать как функцию натурального аргумента, которая связывает не сами слова, а их номера. Это было понятно давно, и с самого начала теория алгоритмов строилась на базе двухбуквенного алфавита. В частности, в рамках рекурсивных функций рассматриваются не произвольные словарные функции, а словарные функции в двухбуквенном алфавите, которые легко интерпретируются как функции натурального аргумента.

Различают ***рекурсивные функции***, определенные для любых комбинаций значений аргументов, (т.е. область определения есть $\mathbb{N}^n$), и ***частично рекурсивные функции***, область определения которых составляет лишь часть множества $\mathbb{N}^n$.

Начнем с множества исходных (простейших) рекурсивных функций.

11.1. Примитивно рекурсивные функции

Выделим простейшие функции натурального аргумента:

1) инкремент $f(x) = x^+ = x + 1$;

2) константа нуль $0(x) = 0$;

3) проективная функция $I^n_m(x_1, x_2, \ldots, x_n) = x_m$, $1 \leqslant m \leqslant n$.

Теперь определим простейшие операции над функциями:

1° суперпозиция $f^m(g^n_1(x_1, x_2, \ldots, x_n), \ldots, g^n_m(x_1, x_2, \ldots, x_n))$ (здесь верхний индекс в обозначении функции указывает на ее арность);

2° примитивная рекурсия, которая из функций f^n и g^{n+2} строит новую функцию $h^{n+1} = R(f^n, g^{n+2})$ в соответствии с равенствами:

$$h^{n+1}(x_1, x_2, \ldots, x_n, 0) = f^n(x_1, x_2, \ldots, x_n),$$
$$h^{n+1}(x_1, x_2, \ldots, x_n, y + 1) = g^{n+2}(x_1, x_2, \ldots, x_n, y, h^{n+1}(x_1, x_2, \ldots, x_n, y)).$$

Мы можем дать индуктивное определение примитивно рекурсивной функции.

1. Простейшие функции примитивно рекурсивные.

2. Если функции f^m, g^n_1, g^n_2, …, g^n_m примитивно рекурсивные, то их суперпозиция — примитивно рекурсивная функция.

3. Если функции f^n и g^{n+2} примитивно рекурсивные, то функция $R(f^n, g^{n+2})$ примитивно рекурсивная.

Отметим, что примитивно рекурсивных функций счетное множество. Действительно, простейших функций счетное множество (за счет проективных функций разной арности). С помощью одной операции из простейших мы получаем опять счетное множество. Следовательно, функций, получаемых двумя операциями счетное множество. В результате имеем счетное семейство счетных множеств, которое, как известно, счетно.

В то же время всех функций натурального аргумента континуум. Множество $\mathbb{N}^k$ при любом k является счетным. Множество всех отображений $\mathbb{N}^k$ в множество $\{0,\ 1\}$ есть булеан счетного множества, т.е. континуум. А множество всех функций k аргументов можно представить как множество подмножеств в $\mathbb{N}^{k+1}$ (графиков функций). Значит, не более чем континуум.

Из этих соображений вытекает, что подавляющее большинство функций натурального аргумента не является примитивно рекурсивным. Однако конкретный пример функции, не являющейся примитивно рекурсивной, привести не просто. Здесь та же ситуация, что и с функциями действительного переменного: весь ассортимент функций, которыми мы реально пользуемся — очень малая часть всего многообразия функций.

Замечание 11.1. Проективные функции позволяют из данной примитивно рекурсивной функции получать новые функции простой перестановкой или дублированием аргументов. Действительно, если $f(x_1, x_2)$ примитивно рекурсивна, то примитивно рекурсивными будут $f(x_2, x_1)$, $f(x, x)$, $f(x, 0)$, поскольку

$$\begin{aligned}
f(x_2, x_1) &= f(I_2^2(x_1, x_2), I_1^2(x_1, x_2)),\\
f(x, x) &= f(I_1^2(x, x_2), I_1^2(x, x_2)),\\
f(x, 0) &= f(I_1^2(x, x_2), 0(I_1^2(x, x_2))).
\end{aligned}$$

По этой же причине если функция $f(x_1, x_2, \ldots, x_n)$ является примитивно рекурсивной, то и функция $g(x_1, x_2, \ldots, x_n, x_{n+1}) = f(x_1, x_2, \ldots, x_n)$, полученная добавлением фиктивного аргумента, тоже примитивно рекурсивна:

$$g(x_1, x_2, \ldots, x_n, x_{n+1}) = f(I_1^{n+1}(x_1, \ldots, x_{n+1}), I_2^{n+1}(x_1, \ldots, x_{n+1}), \ldots, I_n^{n+1}(x_1, \ldots, x_{n+1})).$$

Эти правила можно обобщить, введя следующее понятие. Говорим, что функция h^m получена из функций f^m, $g_1^{k_1}$, $\ldots$, $g_m^{k_m}$ с помощью подстановки, если

$$h^m(x_1, x_2, \ldots, x_n) = f^m(g_1^{k_1}(y_1^1, \ldots, y_{k_1}^1), \ldots, g_m^{k_m}(y_1^m, \ldots, y_{k_m}^m)),$$

где y_s^r обозначает одну из переменных x_1, x_2, $\ldots$, x_n. Как сказано, подстановка может быть сведена к суперпозиции с помощью соответствующих проективных функций.

Пример 11.1. Все постоянные функции примитивно рекурсивные. Действительно, функция $0(x) = 0$ простейшая, функция $f(x) = 1$ есть суперпозиция $0(x)^+$, а значит, простейшая. Далее применяя суперпозицию уже построенной функции $f(x) = k$ с функцией-инкрементом, получаем $f(x) = k + 1$. Постоянные функции от нескольких переменных можно рассматривать как функции, полученные из постоянных функций одного переменного добавлением фиктивных аргументов. Отметим, что установленное свойство позволяет формально интерпретировать постоянные функции одного переменного как функции, полученные из „нульарных“ функций добавлением фиктивного аргумента.

Замечание 11.2. Формально с помощью примитивной рекурсии мы можем получить лишь функцию, имеющую не менее двух аргументов (поскольку функция f имеет хотя бы один аргумент и, значит, $n \geqslant 1$). Однако использование постоянных функций позволяет построить следующую рекурсию:

$$\begin{aligned}
h^1(0) &= k,\\
h^1(y^+) &= g^2(y, h^1(y)).
\end{aligned}$$

Здесь k — произвольное натуральное число. Действительно, рассмотрим постоянную функцию $f^1(\xi) = k$ и функцию $g^3(\xi, x_1, x_2) = g^2(x_1, x_2)$. С помощью f^1 и g^3 строим функцию двух переменных:

$$h^2(\xi, 0) = f^1(\xi),$$
$$h^2(\xi, y^+) = g^3(\xi, y, h^2(\xi, y)).$$

После этого получаем нужную функцию: $h^1(y) = h^2(y, y)$ (на самом деле значение h^2 не зависит от первого аргумента). Приведенный пример наталкивает на мысль интерпретировать конкретные числа как нульарные функции, что позволяет ввести рассмотренный пример в рамки определения примитивной рекурсии.

Пример 11.2. Функция $s(x, y) = x + y$ примитивно рекурсивная. Действительно, из определения суммы в рамках формальной арифметики $x + 0 = x$, $x + y^+ = (x + y)^+$. Из этих равенств заключаем, что $s(x, y)$ есть примитивная рекурсия:

$$s(x, 0) = I_1^2(x, 0), \quad s(x, y^+) = s(x, y)^+ = I_3^3(0, 0, s(x, y)^+).$$

Вторая формула представляет $s(x, y^+)$ как суперпозицию функций $g^3(x, y, z) = z^+ = I_3^3(x, y, z^+)$, 0, 0 и $s(x, y)$.

Пример 11.3. Функция $m(x, y) = xy$ также примитивно рекурсивная. Вспомним из формальной арифметики, что $m(x, 0) = 0$, $m(x, y^+) = m(x, y) + x = s(x, m(x, y))$. Последняя функция имеет два аргумента, а должно быть три. Вводим функцию $s^3(x, y, z) = s(x, z)$. Тогда $m(x, y^+) = s^3(x, y, m(x, y))$. Остается показать, что $s^3(x, y, z)$ является примитивно рекурсивной. Ее можно получить с помощью суперпозиции: $s^3(x, y, z) = s(I_1^3(x, y, z), I_3^3(x, y, z))$.

Теорема 11.1. Если $f^n(x_1, x_2, \ldots, x_n)$ примитивно рекурсивна, то функции

$$s^n(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{x_n} f^n(x_1, x_2, \ldots, x_{n-1}, i) \quad \text{и} \quad m^n(x_1, x_2, \ldots, x_n) = \prod_{i=0}^{x_n} f^n(x_1, x_2, \ldots, x_{n-1}, i)$$

примитивно рекурсивны.

◀ В данном случае

$$s^n(x_1, x_2, \ldots, x_{n-1}, 0) = f^n(x_1, x_2, \ldots, x_{n-1}, 0),$$
$$s^n(x_1, x_2, \ldots, x_{n-1}, k^+) = s^n(x_1, x_2, \ldots, x_{n-1}, k) + f^n(x_1, x_2, \ldots, x_{n-1}, k + 1).$$

Функция $f^n(x_1, x_2, \ldots, 0)$ примитивно рекурсивна как суперпозиция проекций и нулевой функции. Выражение $s^n(x_1, x_2, \ldots, x_{n-1}, k) + f^n(x_1, x_2, \ldots, x_{n-1}, k + 1)$ можно представить в виде $g^{n+1}(x_1, \ldots, x_{n-1}, k, s(x_1, x_2, \ldots, x_{n-1}, k))$, где

$$g^{n+1}(x_1, \ldots, x_{n-1}, k, m) = f^n(x_1, x_2, \ldots, x_{n-1}, k + 1) + m.$$

Последняя функция, очевидно, примитивно рекурсивная.

Доказательство в случае умножения аналогично. ▶

Замечание 11.3. Доказанную теорему можно немного обобщить, а именно: в условиях теоремы для любого k примитивно рекурсивными являются функции

$$s^n(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{x_n - k} f^n(x_1, x_2, \ldots, x_{n-1}, i) \quad \text{и} \quad m^n(x_1, x_2, \ldots, x_n) = \prod_{i=0}^{x_n - k} f^n(x_1, x_2, \ldots, x_{n-1}, i),$$

где предполагается, что сумма нулевая, а произведение равно единице, если нижний предел суммирования больше верхнего. Возьмем, например, сумму. Очевидно, что

$$s^n(x_1, x_2, \ldots, x_n) = \sum_{j=k}^{x_n} f^n(x_1, x_2, \ldots, i - k).$$

Положив

$$g(x_1, x_2, \ldots, x_{n-1}, y) = \begin{cases} f^n(x_1, x_2, \ldots, y-k), & y \geqslant k; \\ 0, & 0 \leqslant y < k, \end{cases}$$

получим

$$s^n(x_1, x_2, \ldots, x_n) = \sum_{j=0}^{x_n} g(x_1, x_2, \ldots, i).$$

Согласно теореме 11.1 эта функция примитивно рекурсивна, если функция g примитивно рекурсивна. Доказать примитивную рекурсивность g — несложное упражнение.

Теорема 11.2. Пусть g_1^n, g_2^n, h^n — примитивно рекурсивные функции. Тогда примитивно рекурсивной является функция

$$f^n(x_1, x_2, \ldots, x_n) = \begin{cases} g_1^n(x_1, x_2, \ldots, x_n), & h^n(x_1, x_2, \ldots, x_n) \neq 0; \\ g_2^n(x_1, x_2, \ldots, x_n), & h^n(x_1, x_2, \ldots, x_n) = 0. \end{cases}$$

◄ Доказательство построено на представлении f^n в виде композиции

$$f^n(x_1,x_2,\ldots,x_n) = g_1^n(x_1,x_2,\ldots,x_n)\cdot \mathrm{sg}(h^n(x_1,x_2,\ldots,x_n)) + g_2^n(x_1,x_2,\ldots,x_n)\cdot \overline{\mathrm{sg}}(h^n(x_1,x_2,\ldots,x_n)),$$

в которой функция $\mathrm{sg}(x)$ равна 1 при $x \neq 0$ и 0 при $x = 0$, а функция $\overline{\mathrm{sg}}(x)$ представляет ее дополнение: $\overline{\mathrm{sg}}(x) = 1 - \mathrm{sg}(x)$. Первую можно построить с помощью примитивной рекурсии: $\mathrm{sg}(0) = 0$, $\mathrm{sg}(x^+) = 1$, где постоянную 1 можно рассматривать как функцию двух аргументов x и $\mathrm{sg}(x)$. Аналогично можно построить и дополняющую функцию $\overline{\mathrm{sg}}(x)$: $\overline{\mathrm{sg}}(0) = 0^+$, $\overline{\mathrm{sg}}(x^+) = 0$. ►

Замечание 11.4. Функции $\mathrm{sg}(x)$ и $\overline{\mathrm{sg}}(x)$ можно получить с помощью ***усеченной разности*** $r(x,y) = x \mathbin{\dot{-}} y = \max\{x-y,\ 0\}$. Действительно, $\overline{\mathrm{sg}}(x) = 1 \mathbin{\dot{-}} x$, $\mathrm{sg}(x) = \overline{\mathrm{sg}}(\overline{\mathrm{sg}}(x))$. Докажем, что усеченная разность разность есть примитивно рекурсивная функция.

Имеем $x \mathbin{\dot{-}} 0 = x$ — примитивно рекурсивная функция. Покажем, что $\varphi(x) = x \mathbin{\dot{-}} 1$ — тоже примитивно рекурсивная функция. Опять примитивная рекурсия: $\varphi(0) = 0$ — примитивно рекурсивна (формально), $\varphi(y^+) = y = I_1^2(y, \varphi(y))$. Теперь, возвращаясь к $r(x,y) = x \mathbin{\dot{-}} y$, можем записать $r(x, y^+) = x \mathbin{\dot{-}} y^+ = (x \mathbin{\dot{-}} y) \mathbin{\dot{-}} 1 = \varphi(r(x,y)) = I_3^3(x, y, \varphi(r(x,y)))$. Полагая $g^3(x,y,z) = I_3^3(x,y,\varphi(z))$, получим $r(x,y^+) = g^3(x,y,r(x,y))$, что соответствует определению примитивной рекурсии. #

11.2. Предикаты, простые числа и возвратная рекурсия

Рассмотренные функции $x+y$, xy и т.д. получаются с помощью алгебраических операций. Таким способом однако нельзя получить другие используемые на практике функции, например функция $p(i)$, значением которой является $(i+1)$-е по порядку простое число. Однако и такие функции являются эффективно вычислимыми и естественен вопрос, являются ли они примитивно рекурсивными.

Выделим класс целочисленных функций, принимающих лишь значения 0, 1, т.е. отображений множества $\mathbb{N}^k$ в множество $\{0,\ 1\}$. Такие функции являются истинностными функциями предикатов, определенных на множестве натуральных чисел. Не разделяя предикаты и их истинностные функции, будет говорить, что ***n-местный предикат*** — это функция от n натуральных аргументов, принимающая два значения 0, 1. В этом контексте можно говорить о ***примитивно рекурсивных предикатах***. Выясним, как себя ведет условие примитивной рекурсивности при выполнении обычных логических операций.

Теорема 11.3. Если $R_1(x_1, \ldots, x_n)$ и $R_2(x_1, \ldots, x_n)$ — примитивно рекурсивные предикаты, то примитивно рекурсивными являются следующие предикаты:

$$R_1 \vee R_2, \quad R_1 \wedge R_2, \quad R_1 \to R_2, \quad R_1 \sim R_2, \quad \neg R_1.$$

◀ Достаточно выразить эти операции через уже рассмотренные:

$$x \vee y = \mathrm{sg}(x+y), \quad x \wedge y = xy, \quad \neg x = \overline{\mathrm{sg}}(x),$$
$$x \to y = \neg x \vee y = \overline{\mathrm{sg}}(x \dot{-} y), \quad x \sim y = \overline{\mathrm{sg}}((x \dot{-} y) + (y \dot{-} x)). \qquad ▶$$

Предикаты можно получать как результат сравнения целочисленных функций. Если сравниваемые функции примитивно рекурсивны, то получаемый предикат тоже примитивно рекурсивен.

Теорема 11.4. Если функции $f(x_1, x_2, \ldots, x_n)$ и $g(x_1, x_2, \ldots, x_n)$ примитивно рекурсивны, то примитивно рекурсивными являются предикаты

$$f < g, \quad f \leqslant g, \quad f \geqslant g, \quad f < g, \quad f = g, \quad f \neq g.$$

◀ Все эти сравнения либо перестановкой двух функций, либо применением логических операций сводятся к одному сравнению $f > g$. Например, $f \geqslant g \equiv \neg(g > f)$, $(f \neq g) \equiv (f > g) \vee (g > f)$. Сравнение $f > g$ можно выразить через ранее рассмотренные функции: $(f > g) \equiv \mathrm{sg}(f \dot{-} g)$. ▶

С предикатами могут также выполняться кванторные операции.

Теорема 11.5. Если $R(x_1, x_2, \ldots, x_n)$ — примитивно рекурсивный предикат, то примитивно рекурсивны предикаты

$$R_\exists(x_1, x_2, \ldots, x_n) = \exists y(y \leqslant x_n \wedge R(x_1, x_2, \ldots, x_{n-1}, y),$$
$$R_\forall(x_1, x_2, \ldots, x_n) = \forall y(y \leqslant x_n \to R(x_1, x_2, \ldots, x_{n-1}, y)$$

и функция

$$\mu(x_1, x_2, \ldots, x_n) = \min\left\{y\colon ((y < x_n) \wedge R(x_1, x_2, \ldots, y)) \vee (y = x_n)\right\}.$$

◀ Утверждение о предикатах вытекает из представлений

$$R_\exists(x_1, x_2, \ldots, x_n) = \mathrm{sg}\Big(\sum_{i=0}^{x_n} R(x_1, x_2, \ldots, x_{n-1}, i)\Big),$$
$$R_\forall(x_1, x_2, \ldots, x_n) = \prod_{i=0}^{x_n} R(x_1, x_2, \ldots, x_{n-1}, i)$$

и теоремы 11.1. Примитивная рекурсивность функции μ вытекает из представления

$$\mu(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{x_n}\Big((i \neq x_n) \wedge \prod_{j=0}^{i} \neg R(x_1, x_2, \ldots, x_{n-1}, j)\Big).$$

Здесь особенность — наличие переменного x_n под знаком суммы, что не предусматривается в теореме 11.1. Обойти это можно так. Функция

$$\mu_e(x_1, x_2, \ldots, x_n, y) = \sum_{i=0}^{y} f^{n+1}(x_1, x_2, \ldots, x_n, i) = \sum_{i=0}^{y}\Big((i \neq x_n) \wedge \prod_{j=0}^{i} \neg R(x_1, x_2, \ldots, x_{n-1}, j)\Big)$$

примитивно рекурсивна. Значит, и функция $\mu(x_1, x_2, \ldots, x_n) = \mu_e(x_1, x_2, \ldots, x_n, x_n)$ примитивно рекурсивна. ▶

Для кванторных операций будем использовать упрощенные обозначения:

$$\exists(y{\leqslant}x_n)\, R(x_1, x_2, \ldots, x_{n-1}, y), \quad \forall(y{\leqslant}x_n)\, R(x_1, x_2, \ldots, x_{n-1}, y).$$

С помощью этих базовых кванторных операций можно получить другие операции, также приводящие к примитивно рекурсивным предикатам, если исходные предикаты примитивно рекурсивны:

$$\nabla(y{<}x_n)\, R(x_1, x_2, \ldots, x_{n-1}, y), \quad \nabla(k{\leqslant}y{\leqslant}x_n)\, R(x_1, x_2, \ldots, x_{n-1}, y)$$

и другие, отличающиеся типом неравенства (здесь ∇ — один из двух кванторов). Аналогичное обозначение введем для операции минимизации μ:

$$\mu(y{\leqslant}x_n)R(x_1, x_2, \ldots, x_{n-1}, y) = \min\{y\colon (y < x_n \wedge R(x_1, x_2, \ldots, y)) \vee y = x_n\}\,.$$

Как и кванторные операции, операция минимизации допускает вариации с заменой неравенства на нестрогое или на двойное неравенство.

Предикаты позволяют ввести новые примитивно рекурсивные функции, например:

1) $x \,\vdots\, y$ — предикат „x делится на y“;

2) $\mathrm{Pr}(x)$ — предикат „число x простое“;

3) $\pi(i)$ — функция, значением которой является $(i+1)$-е (в порядке возрастания) простое число.

Предикаты $x \,\vdots\, y$ и $\mathrm{Pr}(x)$ можно записать с помощью логических и кванторных операций:

$$(x \,\vdots\, y) \equiv \exists(i{\leqslant}x)\,(x = iy), \quad \mathrm{Pr}(x) \equiv \neg\exists(i{<}x)((i > 1) \wedge (x \,\vdots\, i)).$$

Функцию, перечисляющую простые числа, можно задать с помощью примитивной рекурсии следующим образом:

$$\pi(0) = 2, \quad \pi(x^+) = \mu(\pi(x) < i \leqslant \pi(x)! + 1)\ \mathrm{Pr}(i).$$

Здесь использован элементарный факт, что число $\pi(x^+)$, т.е. первое простое число, следующее за $\pi(x)$, не превышает $\pi(x)! + 1$. Действительно, число $\pi(x)! + 1$ не делится ни на одно из простых чисел $\pi(1)$, …, $\pi(x)$. Значит, либо оно простое, либо у него есть простой делитель, отличающийся от указанных. Поэтому в интервале $[\pi(x) + 1,\ \pi(x)! + 1]$ есть простые числа. Взяв среди таких простых чисел наименьшее, получим $\pi(x^+)$.

В операторе μ, использованном при определении $\pi(x)$, задано сложное условие. Такое условие можно реализовать следующим образом. С помощью предиката $\mathrm{Pr}(x)$ строим функцию двух переменных $g(x, y) = \mu(i \leqslant y)(\mathrm{Pr}(i) \wedge (i > x))$, которая возвращает первое простое число на промежутке $x < i < y$ или число y, если на указанном промежутке нет простого числа. Тогда примитивную рекурсию для функции $\pi(x)$ можно определить так: $\pi(x^+) = g(\pi(x), \pi(x)! + 1)$. Остается убедиться в том, что $x!$ — примитивно рекурсивная функция.

Каждое натуральное (ненулевое) число x можно разложить на простые множители, т.е. представить в виде $x = p_1^{a_1} p_2^{a_2} \ldots p_k^{a_k}$. Такое разложение единственно, если простые множители p_i пронумерованы по возрастанию. Более того, такое разложение можно записать в виде бесконечного произведения

$$x = \pi(0)^{c_0} \pi(1)^{c_1} \ldots \pi(n)^{c_n} \ldots \tag{11.1}$$

в котором только конечное число показателей c_i отлично от нуля (ненулевые показатели соответствуют реальным простым множителям в разложении числа x). В результате каждому ненулевому числу поставлена в соответствие последовательность натуральных чисел, в которой только конечное число элементов ненулевое (финитная последовательность). Здесь возникают функции: c_0 как функция x, c_1 как функция x и т.д.

Введем обозначение $\mathrm{pw}_i(x)$ для коэффициента c_i в разложении (11.1) числа x. Полагаем $\mathrm{pw}_i(0) = 0$. Получаем для каждого i функцию pw_i, которая оказывается примитивно рекурсивной. Действительно,

$$\mathrm{pw}_i(x) = \mu(y{<}x)\left[(x \,\vdots\, \pi(i)^y) \wedge \neg(x \,\vdots\, \pi(i)^{y+1})\right].$$

На самом деле последняя формула показывает, что функция двух переменных $\mathrm{pw}(i,x)$ есть примитивно рекурсивная функция (правда, надо доказать, что функция x^y примитивно рекурсивна).

Примитивная рекурсия возникает как реализация метода математической индукции при определении функции. Базовый вариант метода состоит в следующем. Доказываем какое-то утверждение, которое можно рассматривать как истинность некоторого предиката $R(x)$. Если: а) $R(0)$ истинно; б) из истинности $R(n)$ следует истинность $R(n+1)$, то предикат $R(n)$ является тождественно истинным (наше утверждение верно для любого n).

Однако известны различные модификации метода математической индукции, в которых для установления истинности $R(n+1)$ используется истинность нескольких предыдущих значений предиката: истинность $R(k_1)$, $R(k_2)$, …, $R(k_s)$, где $k_1 < k_2 < \ldots < k_s \leqslant n$.

Аналогичным образом значение целочисленной функции $h(x_1, x_2, \ldots, x_n, y+1)$ может определяться на основе не только значения $h(x_1, x_2, \ldots, x_n, y)$, но и некоторых предыдущих значений $h(x_1, x_2, \ldots, x_{n-1}, k_i)$ с $k_i < y$. В этом случае говорят о ***возвратной рекурсии***. Точное определение таково. Говорят, что функция h^{n+1} получается возвратной рекурсией из функций f^n, g^{n+s+1}, t_1^1, …, t_s^1, где $t_i(y) \leqslant y$, $i = \overline{1,\ s}$, если

$$\begin{aligned} &h^{n+1}(x,0) = f^n(x),\\ &h^{n+1}(x,y+1) = g^{n+s+1}(x,y,h^{n+1}(x,t_1^1(y)),\ldots,h^{n+1}(x,t_s^1(y)) \end{aligned} \tag{11.2}$$

(здесь для упрощения через x обозначена группа переменных x_1, x_2, …, x_n).

Теорема 11.6. Функция, полученная возвратной рекурсией из примитивно рекурсивных функций, примитивно рекурсивна.

◀ Идея доказательства состоит в переходе от конструируемой функции h^{n+1} к другой в некотором роде „аккумулирующей“ функции H^{n+1}, для которой* $H(x,y)$ содержит в себе информацию о всех значениях $h(x,k)$, $0 \leqslant k \leqslant y$, причем любое значение $h(x,k)$ извлекается из $H(x,y)$ стандартной процедурой. Тогда из примитивной рекурсивности H мы сможем сделать заключение о примитивной рекурсивности h. Речь идет о кодировании нескольких чисел в одно.

Введем в рассмотрение ***производящую функцию***

$$H(x,y) = \prod_{i=0}^{y} \pi(i)^{h(x,i)}$$

Тогда

$$h(x,k) = \mathrm{pw}(k, H(x,y)), \quad k \leqslant y. \tag{11.3}$$

Докажем, что функция $H(x,y)$ примитивно рекурсивна. Имеем

$$H(x,0) = \pi(0)^{h(x,0)}, \quad H(x,y+1) = \prod_{i=0}^{y-1} \pi(i)^{h(x,i)} \cdot \pi(y+1)^{h(x,y+1)} = H(x,y)\pi(y+1)^{h(x,y+1)}.$$

Поскольку $h(x,y)$ получена возвратной рекурсией, т.е. по формуле (11.2), заключаем, что

$$\begin{aligned} h(x,y+1) &= g(x,y,h(x,t_1(y)),\ldots,h(x,t_s(y)) =\\ &= g(x,y,\mathrm{pw}(t_1(y),H(x,y)),\ldots,\mathrm{pw}(t_s(y),H(x,y)) = G(x,y,H(x,y)), \end{aligned}$$

где

$$G(x,y,z) = g(x,y,\mathrm{pw}(t_1(y),z),\ldots,\mathrm{pw}(t_s(y),z)).$$

В результате

$$H(x,y+1) = H(x,y)\,\pi(y{+}1)^{G(x,y,H(x,y))}$$

и функция $H(x,y)$ получена примитивной рекурсией. Следовательно, $H(x,y)$ примитивно рекурсивна. Согласно формуле (11.3) при $k=y$, функция $h(x,y)$ также примитивно рекурсивна. ▶

*В доказательстве мы опять через x обозначаем группу неизменных аргументов x_1, x_2, …, x_n.

11.3. Частично рекурсивные функции

Оказывается, что класс примитивно рекурсивных функции не охватывает множество всех вычислимых функций (в частности, вычислимых по Тьюрингу или Маркову). Первое обстоятельство — примитивно рекурсивные функции всюду определены. Но есть и второе, более серьезное обстоятельство, связанное с ростом функций.

Функция Аккермана. Рассмотрим функцию $B(x, y)$, называемую ***функцией Аккермана***, которая определяется индуктивно следующими формулами:

$$B(0, y) = 2 + y, \quad B(x, 0) = \mathrm{sg}(x - 1), \ \ x \geqslant 1, \quad B(x + 1, y + 1) = B(x, B(x + 1, y)),$$

Из формул видно, что для вычисления $B(x + 1, y + 1)$ необходимо знать $B(x + 1, y)$ (это укладывается в операцию примитивной рекурсии) и все значения $B(x, z)$, поскольку мы не знаем, какое конкретно $z = B(x + 1, y)$ будет использовано. В действительности вычисление значений функции идет в общем порядке возрастания пар x, y, но конкретный порядок перебора этих пар устроен весьма сложно. Тем не менее эта функция вычислима.

Непосредственно из формул вытекают соотношения

$$B(0, y) = 2 + y, \quad B(1, y) = 2y, \quad B(2, y) = 2^y, \quad B(3, y) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \quad (y \text{ раз}).$$

Видно, что с ростом x функция $B(x, y)$ по y растет все быстрее. В то же время

$$B(x, 0) = \mathrm{sg}(x), \quad B(x, 1) = 2 + \overline{\mathrm{sg}}(x), \quad B(x, 2) = 4, \tag{11.4}$$

т.е. при $y \leqslant 2$ функция $B(x, y)$ как функция переменного x ограничена. Однако

$$B(0, 3) = 5, \quad B(1, 3) = 6, \quad B(2, 3) = 8, \quad B(3, 3) = 16, \quad \ldots,$$

и видно, что функция $B(x, 3)$ растет.

Теорема 11.7. Функция $B(x, y)$ удовлетворяет условиям

$$B(x, y) < B(x, y + 1), \ \ x \in \mathbb{N}, \quad B(x, y + 1) \leqslant B(x + 1, y), \ \ x \in \mathbb{N}.$$

◄ Сначала докажем неравенство $B(x, y) \geqslant y + 2$, верное при $y \geqslant 2$. При $x = 0$ оно верно согласно определению функции Аккермана. Оно также верно при любом $x > 0$ и при $y = 2$, поскольку $B(x, 2) = 4$. Предположим, что оно верно при данном значении x для любого $y \geqslant 2$ и при значениях $x + 1$ и $y > 2$. Тогда

$$B(x + 1, y + 1) = B(x, B(x + 1, y)) \geqslant B(x + 1, y) + 2 \geqslant y + 4 > y + 2.$$

Теперь докажем неравенство $B(x, y + 1) > B(x, y)$. Оно очевидно при $x = 0$. Оно также верно, согласно (11.4), при любом x и при $y < 2$. Пусть $x > 0$ и $y \geqslant 2$. Тогда $B(x, y) \geqslant y + 2 > 2$ и

$$B(x, y + 1) = B(x - 1, B(x, y)) \geqslant B(x, y) + 2 > B(x, y).$$

Наконец, покажем, что $B(x + 1, y) \geqslant B(x, y + 1)$ при $y \geqslant 3$.

$$B(x + 1, y) = B(x, B(x + 1, y - 1)) \geqslant B(x, y - 1 + 2),$$

поскольку $B(x + 1, y - 1) \geqslant y + 1$ при $y \geqslant 3$ и $B(x, z_1) > B(x, z_2)$ при $z_1 > z_2$. ►

Можно показать, что каждая примитивно рекурсивная функция $f(x_1, x_2, \ldots, x_n)$ при некотором достаточно большом m удовлетворяют неравенству

$$f(x_1, x_2, \ldots, x_n) \leqslant B(m, \|x\|), \quad \|x\| > 1.$$

где $\|x\| = \max\{x_1, x_2, \ldots, x_n\}$. Для этого достаточно проверить простейшие функции (для них $m = 0$) и сохранение свойства при суперпозиции и примитивной рекурсии.

Указанным свойством не обладает ***диагональная функция Аккермана*** $A(x) = B(x, x)$. Действительно, тогда должно было бы выполняться неравенство $B(x, x) \leqslant B(m, x)$. Однако это неверно при $x > m$. Значит, диагональная функция Аккермана не является примитивно рекурсивной.

Функция Аккермана — пример вычислимой функции, не являющейся примитивно рекурсивной. Существование таких функций указывает на то, что класс примитивно рекурсивных функций должен быть расширен. Это требование разрешается введением еще одной операции. Скажем, что n-арная функция h^n получена из $(n+1)$-местного предиката P^{n+1} с помощью ***оператора минимизации*** (***μ-оператора***), если

$$h^n(x_1, x_2, \ldots, x_n) = \mu y P^{n+1}(x_1, x_2, \ldots, x_n, y),$$

где символ μ означает выбор первого номера y, при котором $P(x_1, x_2, \ldots, x_n, y)$ является истинным (т.е. равен единице).

Отметим, что оператор μ уже был использован, но с дополнительным ограничением, когда y не превышал одной из переменных (в силу этого такой вариант называют ***оператором ограниченной минимизации***). Оператор ограниченной минимизации не выводит из класса примитивно рекурсивных функций (это было установлено). Оператор общей минимизации уже нельзя выразить через операции композиции и примитивной рекурсии.

Определение 11.1. Скажем, что целочисленная функция f частично рекурсивна, если она удовлетворяет одному из условий:

1) примитивно рекурсивная функция частично рекурсивна;

2) функция, полученная с помощью оператора минимизации из частично рекурсивной функции, является частично рекурсивной.

Частичная рекурсивная функция при некоторых значениях аргументов может быть не определена. Пусть при данных $x_1, x_2, \ldots, x_n$ значение предиката $P(x_1, x_2, \ldots, x_n, y)$ равно нулю независимо от значения переменной y. Тогда оператор минимизации дает неопределенное значение. Внешне это выглядит как бесконечная проверка $y = 1, 2, 3, \ldots$, которая никогда не заканчивается — ситуация, аналогичная неприменимости нормального алгорифма или машины Тьюринга.

Если частично рекурсивная функция определена при любых значениях аргументов, ее называют ***общерекурсивной***.

Пример 11.4. Рассмотрим предикат $P(x, y, z) = (x = y + z)$. С помощью него и оператора минимизации определим функцию

$$s(x, y) = \mu z\, P(x, y, z).$$

Смысл этой функции очевиден: $s(x, y) = x - y$, если $x \geqslant y$, и $s(x, y)$ не определена, если $x < y$.

Теорема 11.8. Любая частично рекурсивная функция вычислима по Маркову (Тьюрингу).

◄ Понятия вычислимости по Маркову и Тьюрингу эквивалентны. Остановимся на вычислимости по Маркову, для чего будем опираться на сочетания нормальных алгорифмов. Надо доказать, что базисные функции вычислимы по Маркову и что операции образования новых функций сохраняют свойство вычислимости по Маркову.

Очевидно, что простейшие функции вычислимы по Маркову. Также легко установить, что композиция функций, вычислимых по Маркову, является вычислимой по Маркову: достаточно сделать соединение нормальных алгорифмов в соответствии с рис. 11.1, на котором алгоритм Z_0 удаляет лидирующий нуль в результате, представленном как γ-система.

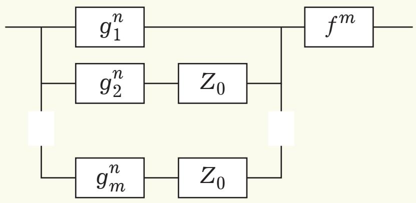
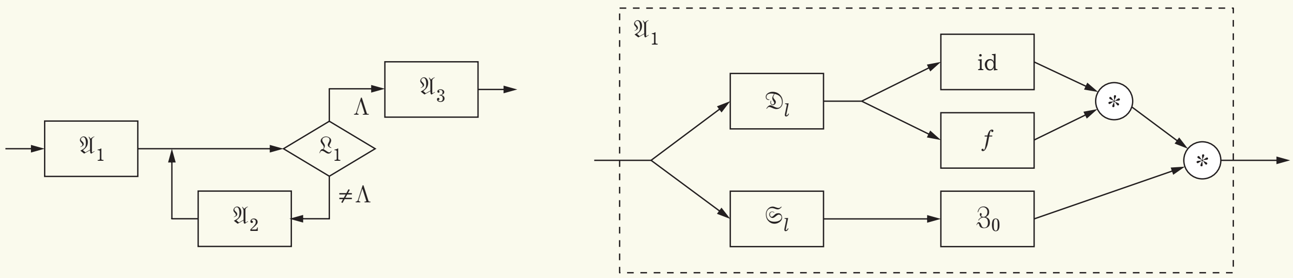


Рис. 11.1

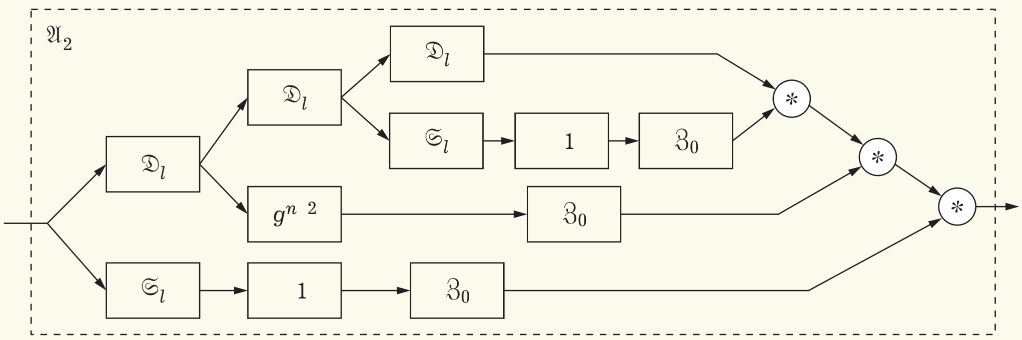
Перейдем к оператору примитивной рекурсии, который определяет функцию h^{n+1} по функциям f^n и g^{n+2}. Чтобы вычислить функцию h^{n+1} для аргументов x_1, x_2, …, x_n, y, необходимо многократно вычислять функцию g^{n+2}, а функцию f^n использовать как начальное значение. В результате наш алгоритм реализуется как повторение с предусловием (рис. 11.2, *а*). Алгоритм $\mathfrak{A}_1$ формирует γ-систему $01^{x_1}0\ldots01^{x_n}01^i01^z01^y0$, где i — текущее значение последнего аргумента функции h^{n+2}, z — текущее значение самой функции h^{n+2}, y — счетчик рекурсии. Начальное значение i нулевое, а начальное значение z есть $f^n(x_1, x_2, \ldots, x_n)$. Далее на каждом шаге мы проверяем условие $y = 0$ (алгоритм $\mathfrak{L}_1$). Если оно выполняется, выделяем из γ-системы слово z и предъявляем как результат (алгоритм $\mathfrak{A}_3$). Если же $y > 0$, то в γ-системе заменяем z на $g^{n+2}(x_1, x_2, \ldots, x_n, y, z)$, а y на $y-1$ (алгоритм $\mathfrak{A}_2$).

Алгоритм $\mathfrak{A}_1$ можно представить как соединение нескольких алгоритмов (рис. 11.2, *б*). В этом соединении алгоритм $\mathfrak{D}_l$ удаляет последний элемент γ-системы, алгоритм $\mathfrak{S}_l$, наоборот, извлекает из γ-системы последний элемент, алгоритм id пустой, алгоритм f реализует вычисление функции f, алгоритм $\mathfrak{Z}_0$ удаляет лидирующий нуль в γ-системе. Структура алгоритма $\mathfrak{A}_2$ представлена на рис. 11.2, *в*, алгоритмы, обозначенные „+1“ и „−1“, соответственно прибавляют к числу 1 и вычитают. Алгоритм $\mathfrak{L}_1$ реализуется как композиция двух алгоритмов: первый алгоритм $\mathfrak{S}_l$ извлекает из γ-системы параметр y, второй проверяет условие $y = 0$ и состоит из одной подстановки $00 \to\cdot \Lambda$. Наконец, алгоритм $\mathfrak{A}_3$ есть композиция алгоритмов $\mathfrak{D}_l$ (удаляет счетчик y) и $\mathfrak{S}_l$ (извлекает z).

Оператор минимизации также реализуется как повторение с предусловием. Формируем γ-систему $01^{x_1}0\ldots01^{x_n}01^y0$. Начальное значение y нулевое. На каждом шаге проверяем значение



а *б*



в

Рис. 11.2

предиката P. Если значение 0, увеличиваем значение y на единицу и процедуру проверки повторяем. Если $P = 0$, выделяем из γ-системы последнее слово и завершаем работу алгоритма. ►

Теорема 11.9. Любая функция, вычислимая по Тьюрингу (Маркову), частично рекурсивна.

◄ В данной теореме мы будем ориентироваться на машины Тьюринга, учитывая, что вычислимость по Тьюрингу равносильна вычислимости по Маркову. Конечно, и машины Тьюринга, и нормальные алгорифмы реализуют словарные функции; числовые функции трактуются как особый тип словарных функций. Но для доказательства теоремы такой трактовки недостаточно, поскольку в процессе работы алгоритм часто выходит за рамки двухбуквенного алфавита. Поэтому необходимо любой словарной функции придать смысл функции числовой.

Пусть $A = a_0 a_1 \ldots a_{m-1}$ — внешний алфавит машины Тьюринга, $Q = q_0 q_1 \ldots q_{n-1}$ — ее внутренний алфавит. Каждое слово в алфавите A можно интерпретировать как запись числа в системе исчисления с основанием m и цифрами a_0, a_1, …, a_n, т.е. слову $X = a_{j_1} a_{j_2} \ldots a_{j_s}$ мы ставим в соответствие число

$$n(X) = j_s + j_{s-1} * m + j_1 * m^{s-1}.$$

Каждому состоянию q_i ставим в соответствие число i. Представленная числовая интерпретация элементов машины Тьюринга называется ее ***арифметизацией***. Арифметизация машины Тьюринга позволяет любую конфигурацию $Xq_j a_i Y$ представить как совокупность четырех чисел: $n(X)$, j, i, $n(\overline{Y})$, $\overline{Y}$ — слово Y в обратном порядке. Инверсия слова Y позволяет пустые символы неограниченной ленты интерпретировать как незначащие разряды числа. Каждый шаг машины Тьюринга состоит из трех действий, зависящих от внутреннего состояния q_j и обозреваемого символа a_i: смене внутреннего состояния, замене обозреваемого символа и сдвиге читающей головки. Каждое из этих действий описывается соответствующей функцией: внутреннее состояние $\hat{q}(j, i)$, печатаемый символ $\hat{a}(j, i)$, сдвиг $\hat{s}(j, i)$, который можно описать тремя значениями 0 (сдвига нет), 1 (влево), 2 (вправо). Каждая из этих функций определена на конечном множестве пар натуральных чисел. Доопределив их нулевым значением вне этого множества, получим примитивно рекурсивные функции.

Каждую конфигурацию $Xq_j a_i Y$, как сказано, можно представить как совокупность четырех чисел $u_1 = n(X)$, $u_2 = j$, $u_3 = i$, $u_4 = n(\overline{Y})$. Один шаг машины Тьюринга можно представить как преобразование этих чисел в новую четверку $u_1' = n(X')$, $u_2' = j'$, $u_3' = i'$, $u_4' = n(\overline{Y}')$. Это преобразование описывается функциями

$$u_1' = \begin{cases} u_1, & s(u_2, u_3) = 0; \\ [u_1/m], & s(u_2, u_3) = 1; \\ u_1 * m + \hat{a}(u_2, u_3), & s(u_2, u_3) = 2; \end{cases} \qquad u_2' = \hat{q}(u_2, u_3);$$

$$u_3' = \begin{cases} \hat{a}(u_2, u_3), & s(u_2, u_3) = 0; \\ u_1 \bmod m, & s(u_2, u_3) = 1; \\ u_4 \bmod m, & s(u_2, u_3) = 2; \end{cases} \qquad u_4' = \begin{cases} u_4, & s(u_2, u_3) = 0; \\ u_4 * m + \hat{a}(u_2, u_3), & s(u_2, u_3) = 1; \\ [u_4/m], & s(u_2, u_3) = 2. \end{cases}$$

Таким образом, один шаг машины Тьюринга описывается четырьмя функциями, которые в силу их представления являются примитивно рекурсивными. Для описания многошаговой работы нужно ввести еще один аргумент, играющий роль времени: $t = 0$ соответствует начальной конфигурации машины Тьюринга, $t = k$ — конфигурации, полученной после k-го шага. В результате получим функции $u_i(t)$, которые определяются следующим образом:

$$u_i(u^0, 0) = u_i^0, u_i(u^0, t+1) = u_i'(u_1(u^0, t), u_2(u^0, t), u_3(u^0, t), u_4(u^0, t)), \quad i = \overline{1, 4},$$

где $u^0 = (u_1^0, u_2^0, u_3^0, u_0^4)$. Это похоже на примитивную рекурсию, но все четыре функции перевязаны. Выход: упаковать все четыре функции в одну так, что каждая извлекается из упакованного варианта с помощью примитивно рекурсивной функции (например, используя характеристическую функцию как в возвратной рекурсии).

Наконец, условие останова реализуется с помощью оператора минимизации. Сперва определяется t как функция начальной конфигурации: $t = \mu\tau(u_2(u^0, t) = 0)$. Затем эта функция начальных условий подставляется в функции u_i, и мы получаем u_i как функции только начальных условий.

Можно считать, что машина Тьюринга начинает и завершает работу в стандартном положении, т.е. с положением читающей головки перед первым символом. Тогда в начале работы $u_1^0 = 0$, $u_1^0 = 1$, $u_2^0 = \overline{X} \bmod m$, $u_3^0 = [\overline{X}/m]$, в конце работы $Y = \overline{u}_4^e * m + u_3^e$. Здесь X — исходное слово, Y — результирующее слово, в которое машина Тьюринга преобразовала слово X. В результате мы получили интерпретацию работы машины как вычисление частично рекурсивной функции.

Пусть задана функция $f(x_1, x_2, \ldots, x_n)$, вычислимая по Тьюрингу. Соответствующая машина Тьюринга преобразует слово, представляющее собой запись аргументов функции в „палочной“ системе, в слово, представляющее собой запись результата в той же „палочной“ системе. Эту машину представляет некоторая функция частично рекурсивная $g(x)$ одного переменного, в которой исходное слово X и результирующее слово Y интерпретируются как запись чисел $n(X)$ и $n(Y)$ в системе исчисления с основанием m. Исходную функцию $f(x_1, x_2, \ldots, x_n)$ можно представить как композицию трех функций, первая преобразует набор чисел x_1, x_2, …, x_n в число $n(X)$, вторая — это $g(x)$, третья преобразует число $n(Y)$ в результат $f(x_1, x_2, \ldots, x_n)$. В свете этого, чтобы доказать частичную рекурсивность функции f, достаточно доказать частичную рекурсивность двух преобразований. Первое преобразование описывается формулой

$$n(X) = m + \ldots + m^{x_n} + m^{x_n+2} + m^{x_n+3} + \ldots + m^{x_n+x_{n-1}+2} + \ldots,$$

из которых видно, что это преобразование есть примитивно рекурсивная функция. Второе преобразование обратное. Его можно интерпретировать как пересчет числа в сумму его цифр. Можно показать, что она примитивно рекурсивна. С учетом этого факта заключаем, что функция $f(x_1, x_2, \ldots, x_n)$ частично рекурсивна. ▶

11.4. Универсальные рекурсивные функции

Пусть дана $(n+1)$-арная частично рекурсивная функция $f(x_0, x_1, \ldots, x_n)$. Ясно, что для любого конкретного числа l функция $g_l(x_1, x_2, \ldots, x_n) = f(l, x_1, \ldots, x_n)$ является частично рекурсивной. Другими словами, каждая частично рекурсивная функция от $n+1$ аргументов порождает счетное семейство частично рекурсивных функций от n аргументов. Может ли быть такое, что это семейство накрывает множество всех частично рекурсивных функций от n аргументов? Существование универсального алгорифма наталкивает на мысль, что такое вполне возможно.

Ключевым пунктом здесь является следующая теорема.

Теорема 11.10 (*теорема Клини о нормальной форме*). Для каждого натурального $n \geqslant 1$ существуют такая примитивно рекурсивная функция F^1 и такой примитивно рекурсивный предикат D^{n+2}, что для любой n-арной частично рекурсивной функции f^n при некотором значении l имеет место представление

$$f^n(x_1, x_2, \ldots, x_n) \simeq F^1(\mu y D^{n+2}(l, x_1, x_2, \ldots, x_n, y)). \quad \#$$

Как видим, любая частично рекурсивная функция может порождаться только одним применением оператора примитивной рекурсии.

Пусть $\mathcal{F}^n$ — некоторое множество n-арных частично рекурсивных функций. Скажем, что функция U^{n+1} является универсальной для множества $\mathcal{F}^n$, если оно совпадает с множеством всех функций $f_l(x_1, \ldots, x_n) = U^{n+1}(l, x_1, \ldots, x_n)$, $l \in \mathbb{N}$. В качестве такого множества будем рассматривать множество $\mathcal{F}^n_{\text{чр}}$ всех частично рекурсивных функций, множество $\mathcal{F}^n_{\text{ор}}$ всех общерекурсивных функций и множество $\mathcal{F}^n_{\text{пр}}$ всех примитивно рекурсивных функций.

Теорема 11.11. 1. Существует универсальная частично рекурсивная функция для множества $\mathcal{F}^n_{\text{чр}}$.

2. Существует общерекурсивная функция, универсальная для множества $\mathcal{F}^n_{\text{пр}}$, причем ни одна такая функция не является примитивно рекурсивной.

3. Множество $\mathcal{F}^n_{\text{ор}}$ не имеет универсальной общерекурсивной функции.

Первое утверждение теоремы — очевидное следствие из теоремы Клини. Два других утверждения менее очевидны. Отметим, что из второго утверждения вытекает, что существуют общерекурсивные функции, не являющиеся примитивно рекурсивными.

Универсальные частично рекурсивные функции в определенном смысле имеют максимальную область определения. Скажем, что функция g^n является доопределением функции f^n, если ее область определения включает область определения f^n и на области определения f^n значения двух функций совпадают.

Теорема 11.12. Никакая универсальная частично рекурсивная функция для множества $\mathcal{F}^n_{\text{ор}}$ не может быть доопределена до общерекурсивной.

◀ Пусть U^{n+1} — универсальная частично рекурсивная функция и G^{n+1} — ее общерекурсивное доопределение. Рассмотрим функцию $g(x_1, x_2, \ldots, x_n) = G^{n+1}(x_1, x_1, x_2, \ldots, x_n) + 1$. Это общерекурсивная функция n переменных и должна выражаться через универсальную функцию. Следовательно, при некотором l

$$g(x_1, x_2, \ldots, x_n) = U^{n+1}(l, x_1, x_2, \ldots, x_2) = G^{n+1}(l, x_1, x_2, \ldots, x_2).$$

Но, учитывая определение функции g, заключаем, что

$$G^{n+1}(x_1, x_1, x_2, \ldots, x_n) + 1 = G^{n+1}(l, x_1, x_2, \ldots, x_2).$$

В частности,

$$G^{n+1}(l, l, \ldots, l) + 1 = G^{n+1}(l, l, \ldots, l),$$

а это невозможно. ▶

Конечно, множество $\mathcal{F}^n_{\text{ор}}$ имеет универсальную частично рекурсивную функцию (таковой является любая универсальная функция для множества $\mathcal{F}^n_{\text{чр}}$). Согласно теореме 11.12 среди универсальных функций для множества $\mathcal{F}^n_{\text{ор}}$ нет общерекурсивных.

Следствие 11.1. Существует одноместный частично рекурсивный предикат, не имеющий общерекурсивных доопределений.

◀ Выберем какую-нибудь универсальную общерекурсивную функцию U^{n+1} и положим $f(x) = \overline{\text{sg}}(U^{n+1}(x, x, \ldots, x))$. Пусть эта функция имеет общерекурсивное доопределение $v(x)$. Образуем функцию n переменных, введя фиктивные переменные: $g(x_1, x_2, \ldots, x_n) = v(x_1)$. Эта функция, как частично рекурсивная, выразима через универсальную. Поэтому при некотором l

$$\overline{\text{sg}}(U^{n+1}(x, x, \ldots, x)) = v(x) = U^{n+1}(l, x, x_2, \ldots, x_n).$$

В частности,

$$\overline{\text{sg}}(U^{n+1}(l, l, \ldots, l)) = U^{n+1}(l, l, l, \ldots, l),$$

а это равенство неверно. ▶

11.5. Разрешимые и перечислимые множества

Под разрешимым понимают множество, для которого можно построить разрешающий алгоритм, т.е. алгоритм, который для любого объекта a определяет, принадлежит a множеству или нет. Мы рассматриваем подмножества множества натуральных чисел. Ясно, что среди них есть неразрешимые (на всех алгоритмов не хватает).

Каждое подмножество A в $\mathbb{N}$ можно описать ***характеристической функцией***

$$\chi_A(x) = \begin{cases} 1, & x \in A; \\ 0, & x \notin A. \end{cases}$$

Разрешимость множества равносильна вычислимости его характеристической функции. Понятие вычислимой функции мы отождествляем с понятием частично рекурсивной функции. Для целей описания множеств нужны всюду определенные, т.е. общерекурсивные функции. Это объясняет следующее определение.

Определение 11.2. ***Разрешимое множество*** (также ***рекурсивное множество***) — любое подмножество множества натуральных чисел, характеристическая функция которого является общерекурсивной. Если характеристическая функция примитивно рекурсивна, то множество называют ***примитивно рекурсивным***.

Пример 11.5. Пустое множество, множество всех натуральных чисел примитивно рекурсивны, поскольку их характеристические функции, будучи постоянными, примитивно рекурсивны. Множество четных чисел примитивно рекурсивно: его характеристическая функция описывается формулой $f(x) = 1 \dot{-} (x \bmod 2)$.

Теорема 11.13. Если множества A и B разрешимы (примитивно рекурсивны), то и множества $A \cap B$, $A \cup B$, $A \setminus B$, $\overline{A} = \mathbb{N} \setminus A$ разрешимы (примитивно рекурсивны).

◄ Утверждение теоремы вытекает из следующих представлений теоретико-множественных операций через характеристические функции:

$$\chi_{A\cap B} = \chi_A\,\chi_B, \quad \chi_{A\cup B} = \chi_A + \chi_B \dot{-} \chi_A\,\chi_B, \quad \chi_{A\setminus B} = \chi_A(1 \dot{-} \chi_B), \quad \chi_{\overline{A}} = 1 \dot{-} \chi_A. \qquad \blacktriangleright$$

Пример 11.6. Одноточечное множество $A = \{a\}$ примитивно рекурсивно, поскольку его характеристическая функция $\chi_A(x) = \overline{\mathrm{sg}}(|x - a|)$ примитивно рекурсивна. Конечное множество примитивно рекурсивно как конечное объединение примитивно рекурсивных множеств.

Теорема 11.14. Если общерекурсивная (примитивно рекурсивная) функция f удовлетворяет условию $f(x) \geqslant x$, $x \in \mathbb{N}$, то множество ее значений разрешимо (примитивно рекурсивно).

◄ Суть в механизме проверки, является ли данное число x значением функции f. При $y > x$ имеем $f(y) \geqslant y > x$, так что равенство $x = f(y)$ надо проверять только для значений y, не превышающих x. Характеристическую функцию множества A значений функции f можно записать с помощью ограниченного квантора существования:

$$\chi_A(x) = \exists (y \leqslant x)(f(y) = x).$$

Из этой формулы вытекает общерекурсивность (примитивная рекурсивность) χ_A в случае, когда $f(y)$ общерекурсивна (примитивно рекурсивна). ►

Пример 11.7. Как было показано, функция $\mathrm{Pr}(x)$, значением которой является простое число с номером x, примитивно рекурсивна. Множество ее значений — множество всех простых чисел — является примитивно рекурсивным, поскольку $\mathrm{Pr}(x) \geqslant x$, $x \in \mathbb{N}$.

Определение 11.3. Множество $A \subset \mathbb{N}$ называется ***рекурсивно перечислимым***, если оно совпадает с областью определения некоторой частично рекурсивной функции.

Определение можно переформулировать и так: множество A рекурсивно перечислимо, если существует частично рекурсивная функция f, удовлетворяющая условию: $f(x) = 1$, если $x \in A$, и $f(x)$ не определена, если $x \notin A$. Действительно, если A есть область определения частично рекурсивной функции g, то указанному требованию удовлетворяет функция $f(x) = 1(g(x))$.

Нетрудно понять, что понятие рекурсивно перечислимого множества более широкое по сравнению с понятием разрешимого множества. Действительно, если для каждого n мы можем проверить, принадлежит оно множеству или нет, то ясен алгоритм и перечисления таких чисел: последовательно просматривая натуральный ряд, мы присваиваем функции значение 1, если число принадлежит множеству, или зацикливаем алгоритм в противном случае. Докажем этот факт строго.

Теорема 11.15. Разрешимое множество рекурсивно перечислимо.

◀ Согласно условию характеристическая функция χ_A множества A общерекурсивна. Составим функцию $f_A(x) = \mu y(\chi_A(x)) + 1$. Если $\chi_A(x) = 0$, то значение оператора минимизации не определено и функция $f_A(x)$ не определена. Если $\chi_A(x) = 1$, то $\mu y(\chi_A(x)) = 0$ и значение $f_A(x)$ есть 1. Следовательно, область определения функции f_A совпадает с множеством A и это множество рекурсивно перечислимо. ▶

Теорема 11.16. Непустое множество рекурсивно перечислимо тогда и только тогда, когда оно является областью значений некоторой одноместной примитивно рекурсивной функции.

◀ Доказательство в одну сторону основано на той же идее, что и предыдущая теорема, только квантор существования уже не будет ограниченным и вместо него лучше использовать оператор минимизации. Итак, пусть A — множество значений примитивно рекурсивной функции f. Сформируем двуместный предикат $f(y) = x$ и используем оператор минимизации:

$$g(x) = \mu(y)(f(y) = x).$$

Функция $g(x)$ частично рекурсивна, так как получена с помощью оператора минимизации из примитивно рекурсивного предиката. Она не определена, если x не является значением функции f, и имеет значение наименьшего числа y, для которого $f(y) = x$, если x является значением функции f.

В другую сторону доказательство сложнее. Речь идет о перестройке процедуры, порождающей частично рекурсивную функцию. Согласно теореме Клини о нормальной форме, частично рекурсивная функция одного переменного может быть получена из примитивно рекурсивного предиката одним оператором минимизации (функция F^1 в нормальной форме не играет роли). Итак, пусть частично рекурсивная функция $f(x)$ получена с помощью примитивно рекурсивного предиката $P(x, y)$ согласно формуле $f(x) = \mu y(P(x, y))$. Согласно этой формуле, число x принадлежит области определения функции f, если для этого x при некотором y предикат $P(x, y)$ имеет значение 1. Надо пересчитать все такие x, может быть, с повторением. Для этого достаточно перенумеровать пары $(x,\ y)$ в одну последовательность, т.е. сформировать функцию $c(x, y)$. Этой функции соответствуют функции $l(n)$ и $r(n)$, вычисляющие по номеру пары левую и правую компоненты. Все три функции при подходящем выборе нумерации примитивно рекурсивны (это можно показать). Далее создаем функцию

$$g(n) = \begin{cases} l(n), & P(l(n), r(n)); \\ b, & \neg P(l(n), r(n)), \end{cases}$$

где b — один из фиксированных элементов множества, которое по условию непусто. Очевидно, что множество значений функции g совпадает с рассматриваемым множеством. ▶

Теорема 11.17. Если A и B рекурсивно перечислимы, то $A \cap B$ и $A \cup B$ рекурсивно перечислимы.

◄ Пусть A и B — области определения функций f_A и f_B. Тогда область определения, например, функции $f_A(x)f_B(x)$ есть $A \cap B$.

Отметим, что при $A = \varnothing$ или $B = \varnothing$ множество $A \cup B$ совпадает с B или A, а потому рекурсивно перечислимо. Предположим, что A И B не пусты. Тогда существуют примитивно рекурсивные функции g_A и g_B, множества значений которых совпадают с A и B. Рассмотрим функцию

$$h(x) = \begin{cases} g_A([x/2]), & x \bmod 2 = 0; \\ g_B((x/2]), & x \bmod 2 \neq 0. \end{cases}$$

Множеством значений этой функции является $A \cup B$. ►

Как показано выше класс рекурсивно перечислимых множеств включает в себя класс разрешимых множеств. Уточним связь между двумя классами множеств.

Теорема 11.18. Существуют рекурсивно перечислимые множества, не являющиеся разрешимыми. Существуют рекурсивно перечислимые множества, дополнения к которым не являются рекурсивно перечислимыми.

◄ Область определения A предиката f из следствия 11.1 — рекурсивно перечислимое множество. Рассмотрим функцию $h(x) = 1(f((x))$, равную 1 на множестве A и не определенную вне A. Если бы множество A было разрешимо, существовал бы предикат χ_A, равный 1 на A и 0 вне A, т.е. функция h имела бы общерекурсивное доопределение. Но тогда и функция f имеет общерекурсивное доопределение:

$$\tilde{f}(x) = \begin{cases} f(x), & \chi_A(x); \\ 0, & \neg\chi_A(x). \end{cases}$$

Поскольку в силу следствия 11.1 функция f не имеет общерекурсивного доопределения, то и предикат χ_A не существует, а множество не является разрешимым.

Пусть D^3 — предикат из нормальной формы Клини для $n = 1$. Рассмотрим функцию $\varphi(x) = \mu y D^3(x, x, y)$. Поскольку φ образована из примитивно рекурсивного предиката оператором минимизации, она частично рекурсивна. Ее область определения $B = \{x \in \mathbb{N}\colon \exists y D(x, x, y)\}$. Пусть X — произвольное рекурсивно перечислимое множество. Согласно теореме о нормальной форме его можно определить как область определения функции $\mu y D^3(l, x, y)$ при некотором значении l. Выясняется, что если $l \in B$, т.е. $\exists y D(l, l, y)$, то $l \in X$, а если $l \notin B$, то $l \notin X$. Значит, произвольно взятое рекурсивно перечислимое множество не может быть дополнением к B и дополнение к B не является рекурсивно перечислимым. ►

Теорема 11.19 (*теорема Поста*). Если множество A и его дополнение рекурсивно перечислимы, то множество A разрешимо.

◄ Можно выделить случай, когда либо A, либо его дополнение пусто. В этом случае утверждение тривиально. Будем считать, что $A \neq \varnothing$, $\overline{A} \neq \varnothing$. Существуют примитивно рекурсивная функция f_1, множество значений которой есть A, и примитивно рекурсивная функция f_2, множество значений которой есть $\overline{A}$. Рассмотрим функцию $f(x) = \mu y((f_1(y) = x) \vee (f_2(y) = x))$. Покажем, что она общерекурсивна. Это значит, что для любого x существует такое y, что либо $f_1(y) = x$, либо $f_2(y) = x$. Пусть x произвольно. Если $x \in A$, то $\exists y(f_1(y) = x)$ (т.е. x есть значение f_1), а если $x \in \overline{A}$, то y — значение f_2, т.е. $\exists y(f_2(y) = x)$.

Таким образом, функция f является общерекурсивной. Рассмотрим общерекурсивный предикат $v(x) = (f_1(f(x)) = x)$. Если $x \in A$, то $\exists y(f_1(y) = x)$ и $\nexists y(f_2(y) = x)$. Значит, значением $f(x)$ является первое же y, для которого $f_1(y) = x$. Но тогда $f_1(f(x)) = x$ и $v(x) = 1$. Пусть $x \notin A$. Тогда $\nexists y(f_1(y) = x)$ и $\exists y(f_2(y) = x)$. Следовательно, значение $f_1(f(x) \neq x$, каково бы ни было значение y. Поэтому $v(x) = 0$. Мы доказали, что общерекурсивный предикат v является характеристической функцией множества A. Следовательно, A разрешимо. ►

ОГЛАВЛЕНИЕ