Рекурсивные функции

Данная алгоритмическая система сложилась исторически первой (1931-1934 гг.), ввиду чего получила достаточно полное и всестороннее развитие. В ее основании лежит использование конструктивно определяемых арифметических целочисленных функций, которые получили специальное название — рекурсивные функции.

Определение. *Рекурсией называют способ задания функции, при котором значение определяемой функции для произвольных значений аргументов выражается через значения определяемой функции для меньших значений аргументов.*

Уточняя интуитивное определение алгоритма, А.А. Марков ввел понятие нормального алгоритма, что оказалось весьма удобным для специалистов в области математической логики. Однако для специалистов, имеющих дело преимущественно с ЭВМ, естественно иметь стандартный аппарат, приближенный к форме численных алгоритмов. Исследования показали, что любой алгоритм может быть всегда сведен к численному алгоритму.

Применение рекурсивных функций в теории алгоритмов основано на идее нумерации слов в произвольном алфавите последовательными числами. Например, логично нумеровать целыми числами слова, располагая их в порядке возрастания длины, а слова одинаковой длины — в произвольном порядке или, например, по алфавиту. Тогда после завершения нумерации входных и выходных слов алфавитный оператор превращается в некоторую функцию вида $y = f(x)$, где как аргумент x, так и сама функция y, принимают целые неотрицательные значения. Эта функция может быть определена лишь для некоторых значений аргумента x, составляющих область определения функции y. Подобные, частично определенные, целочисленные функции получили название «арифметических».

Определение. *Функция $y = f(x_1, x_2, ..., x_n)$ называется алгоритмически вычислимой, если существует алгоритм, позволяющий определить значение функции при любых значениях переменных $x_1, x_2, ..., x_n$, входящих в область определения f.*

Это определение носит интуитивный характер, поскольку не ясно что понимается под вычисляющим алгоритмом. Для уточнения попытаемся построить класс всех вычислимых функций, начиная с простейших.

Назовем «элементарными» арифметические функции, получаемые из целых неотрицательных чисел и переменных с помощью операций сложения, арифметического вычитания, умножения и арифметического деления, а также построений сумм и произведений.

Примечание. *Под арифметическим вычитанием будем понимать получение абсолютной величины* $\mid x - y \mid$.

Примечание. *Под арифметическим делением будем понимать целое от частного* a/b *при* $b \neq 0$.

Вычислимость элементарных функций не вызывает никаких сомнений, поскольку операции четырех арифметических действий хорошо известны. В качестве исходного числа для построения элементарных функций можно взять число 1, т.к. $0 =\mid 1 - 1 \mid$, 1, $2 = 1 + 1$, $3 = (1 + 1) + 1$, и т.д. Очевидно, число элементарных функций бесконечно. Некоторые примеры:

- $f(x) = x + 1$;
- $f(y) = 50 \cdot x$;
- $f(a, b, c) = a \cdot b + c$;
- $f(x, y) =\mid x + y \mid /2$.

Попробуем ответить на вопрос: все ли вычислительные функции являются элементарными? Другими словами — шире ли класс вычислимых функций класса элементарных? Чтобы получить ответ на этот вопрос, будем усложнять элементарные функции, взяв за критерий сложности скорость роста значений функций.

Из всех элементарных функций быстрее всего растет произведение, так как $a \cdot n$ — это n раз повторенное сложение $a + a + a + a + ... + a$. Очевидно, что умножение есть итерация сложения.

Операция возведения в степень, в свою очередь, есть итерация умножения: $a^n = a \cdot a \cdot a \cdot ... \cdot a$. И та и другая функции являются элементарными, так как выражаются через сложение и умножение.

Рассмотрим еще более быстро растущую функцию, которая является итерацией возведения в степень:

$$\psi(0, a) = a, \psi(1, a) = a^a, ..., \psi(n + 1, a) = a^{\psi(n,a)}$$

Эта функция растет чрезвычайно быстро. Оказывается, что она не может быть реализована с помощью элементарных функций; начиная с некоторого числа $a = a*$ эта функция мажорирует над всеми элементарными функциями. Это означает, что для любой элементарной функции $\phi(a)$ найдется такое число $m*$, что будет выполняться неравенство $\phi(a) < \psi(m*, a)$ для всех $a \geq a*$, следовательно функция не является элементарной. Действительно, если бы $\psi(x, n)$ была элементарной функцией, то нашлось бы число $m*$ такое, что $\psi(x, n) < \psi(m*, n)$, при $n \geq 2$. Это справедливо при $n = m*$, $m* \geq 2$, получим $\psi(m*, m*) < \psi(m*, m*)$, что невозможно.

Таким образом, итерация операции возведения в степень позволяет получить неэлементарную функцию. В то же время $\psi(x, a)$ заведомо вычислима. Пусть нужно вычислить значение $\psi(x, a)$ при любых $x = n*$, $a = a*$. Используя обобщенную формулу, имеем при $a = a*$: $\psi(x + 1, a*) = (a*)^{\psi(n,a*)}$. Обозначим $(a*)^m = \chi(m)$, где $\chi(m)$ — элементарная всюду вычислимая функция. Алгоритм ее вычисления сводится к повторенному m раз умножению на $a*$. Запись $\psi(n + 1, a*) = \chi(\psi(n, a*))$ определяет связь значений функции в текущей и предыдущей точках. Задав начальное значение $\psi(0, a*) = a*$, получим последовательность вычисляемых значений:

$$\psi(1, a*) = \chi(\psi(0, a*)) = \chi(a*),$$

$$\psi(2, a*) = \chi(\psi(1, a*)) = \chi(\chi(a*)),$$
$$\psi(3, a*) = \chi(\psi(2, a*)) = \chi(\chi(\chi(a*))),$$
$$\ldots$$

Продолжая вычисления до значения $\psi(n*, a*)$, можно показать, что функция ψ всюду однозначно определена. Это позволяет сделать вывод о том, что класс вычислительных функций шире класса всех элементарных функций.

Обратим внимание на тот факт, что $\psi(x, a)$ была задана по индукции, то есть вначале было определено начальное значение $\psi(0, a)$ — базис, а также указан способ вычисления всех ее значений по предыдущим значениям с помощью вполне доступных операций. Отметим, что определение по индукции может быть введено на любом упорядоченном множестве, где введены понятия «предыдущий» и «следующий».

Обозначим через x' функцию «следование за». Она определяет переход от одного элемента к следующему за ним. Для натурального ряда чисел

$N = 0, 1, 2, 3, ...$ имеем: $0' = 1, 1' = 2, 2' = 3$ и т.д. Очевидно, в этом случае функция следования совпадает с функцией $x + 1$. Однако, в зависимости от вида множества приращения может и отличаться от 1.

Уточним общую схему задания функции $\phi(x)$ по индукции.

1. Задаем базис $\phi(0)$;

2. Для любого x укажем, каким образом значение $f(x')$ формально описано
$$\begin{cases} \phi(0) = q_0; \\ \phi(x') = \psi(x, \phi(x)). \end{cases}$$

В общем случае функция может быть зависимой от n переменных. Тогда схема будет иметь вид

$$\begin{cases} \phi(0, x_1, x_2, ..., x_n) = g(x_1, x_2, ..., x_n); \\ \phi(y', x_1, x_2, ..., x_n) = h(y, \phi(y, x_1, x_2, ..., x_n), x_2, x_3, ..., x_n). \end{cases}$$

Если g и h известны и вычислимы, тогда с помощью указанной схемы можно последовательно вычислять $\phi(1, x_1, x_2, ..., x_n)$, $\phi(2, x_1, x_2, ..., x_n)$ и так далее. Таким образом эта схема определяет вычислимую функцию.

Какие же арифметические функции могут быть определены по индукции? Чтобы дать конкретный ответ необходимо указать какие функции считаются известными первоначально и какие операции кроме, кроме схемы индукции, допустимы при определении функций.

Среди арифметических функций выделим следующие, особо простые, которые получили название «элементарные арифметические».

1. $\phi(x) = x'$ — функция следования, применимая для натурального ряда (сокращенное обозначение $S(x)$);

2. $\phi(x_1, x_2, ..., x_n) = q, q = const$ — функция константы (сокращенное обозначение C_q^n);

3. $\phi(x_1, x_2, ..., x_n) = x_i$ — функция тождества (сокращенное обозначение X_i^n);

4. $\phi(x_1, x_2, ..., x_n) = \psi(\chi_1(x_1, x_2, ..., x_n), ..., \chi_m(x_1, x_2, ..., x_n))$ — функция подстановки.

Используя в качестве исходных функций перечисленные элементарные функции, можно с помощью небольшого числа конструктивных приемов строить все более и более сложные арифметические функции.

Полная система формального описания схем для получения произвольных функций может быть представлена следующим образом.

1. $\phi(x) = S(x) = x'$;

2. $\phi(x_1, x_2, ..., x_n) = C_q^n = q$;

3. $\phi(x_1, x_2, ..., x_n) = X_i^n = x_i$;

4. $\phi(x_1, x_2, ..., x_n) = \psi(\chi_1(x_1, x_2, ..., x_n), ..., \chi_m(x_1, x_2, ..., x_n))$;

5. $$\begin{cases} \phi(0) = q; \\ \phi(x') = \psi(x, \phi(x)). \end{cases}$$

6. $$\begin{cases} \phi(0, x_1, x_2, ..., x_n) = g(x_1, x_2, ..., x_n); \\ \phi(y', x_1, x_2, ..., x_n) = h(y, \phi(y, x_1, x_2, ..., x_n), x_2, x_3, ..., x_n). \end{cases}$$

Схемы под номерами 1-4 задают первоначальные функции, играя роль аксиом, а схемы 5 и 6 можно рассматривать как правила вывода.

Определение. *Функция $\phi(x_1, x_2, ..., x_n)$ называется примитивно рекурсивной, если она может быть определена с помощью конечного числа применений схем 1-6.*

Рассмотрим пример. Определим функцию $\phi(x, y)$ следующим образом.

$$\begin{cases} \phi(0, x) = x; \\ \phi(y', x) = [\phi(y, x)]'. \end{cases}$$

В соответствии с вышеприведенным определением можно представить следующую последовательность:

$$\phi(1, x) = x' = x + 1,$$

$$\phi(2, x) = [\phi(1, x)'] = (x + 1)' = x + 2,$$

$$\phi(3,x) = [\phi(2,x)'] = (x+2)' = x+3,$$

$$...$$

$$\phi(y,x) = x+y.$$

Таким образом, заданная функция соответствует операции сложения двух аргументов x и y.

Примитивно-рекурсивное описание этой функции составляется с использованием схемы 5.

$$\begin{cases} \phi(0,x) = g(x); \\ \phi(y',x) = h(y,\phi(y,x),x). \end{cases}$$

Здесь функция $g(x)$ имеет вид $g(x) \equiv x$, т.е. это функция тождества $X_1^1(x)$.

В свою очередь функция $h(y,z,x) \equiv z'$ может быть получена из первоначальной функции $X_2^3(y,z,x) \equiv z$ с помощью использования схемы 4, в качестве ϕ берется функция следования $S(z) \equiv z'$. Тогда имеем $h(y,z,x) = S[X_2^3(y,z,x)]$. Примитивно-рекурсивным описанием функции h будет последовательность S, X_2^3, x. Добавляя к ней функцию $g(x) \equiv X_1^1(x)$, от которой зависит функция $\phi(y,x)$ получаем: S, X_2^3, x, X_1^1, ϕ.

Рассмотрим еще один пример. Зададим следующую функцию.

$$\begin{cases} \phi(0,x) = 0; \\ \phi(y',x) = \phi(y,x) + x. \end{cases}$$

Тогда получим последовательность:

$$\phi(1,x) = \phi(0,x) + x = x,$$

$$\phi(2,x) = \phi(1,x) + x = x + x = 2 \cdot x,$$

$$\phi(3,x) = \phi(2,x) + x = 2 \cdot x + x = 3 \cdot x,$$

$$...$$

$$\phi(y,x) = y \cdot x.$$

Очевидно, что произведение аргументов x и y — также есть примитивно-рекурсивная функция. К этому же классу относятся возведение в степень и многие другие операции.

Итак, мы установили, что класс примитивно-рекурсивных функций содержит вычислимые функции. Можно поставить вопрос — все ли вычислимые функции входят в класс примитивно-рекурсивных?

Оказывается, класс вычислимых функций шире, чем класс примитивно-рекурсивных функций. Это показали независимо друг от друга Р. Петер и В. Аккерман, что означает, что средства построения вычислимых функций нуждаются в расширении. Операторы рекурсии не дают желаемого замыкания класса всех вычислимых функций. Подходящим для этой цели оказался μ-оператор.

Определение. *Функция называется частично-рекурсивной, если она построена из простейших функций константы C_q^n, тождества X_i^n, следования $S(x)$ и с помощью применения конечного числа операций суперпозиции, примитивной рекурсии и μ-оператора.*

Определим операцию с применением μ-оператора как «операцию наименьшего корня». Эта операция позволяет определить новую арифметическую функцию $f(x_1, x_2, ..., x_n)$ от n переменных с помощью ранее построенной арифметической функции $g(x_1, x_2, ..., x_n, y)$ от $n + 1$ переменной. Для $x_1 = a_1, x_2 = a_2, ..., x_n = a_n$ в качестве соответствующего значения $f(a_1, a_2, ..., a_n)$ определяемой функции принимается наименьший целый неотрицательный корень $y = \beta$ из уравнения $g(a_1, a_2, ..., a_n, y) = 0$. Если такой целый корень не существует, функция считается неопределенной при соответствующем наборе значений переменных.

Итак, среди рекурсивных функций появляются полностью определенные (частичные) функции, а операторы над частичными функциями порождают новые частичные функции. Характер неопределенности может быть довольно сложным. Действительно, для набора значений $x_1, x_2, ..., x_n$ не всегда найдется способ установить, определена ли функция f на этом наборе, что вызывает продолжение процесса вычислений в течении неопределенного времени, не зная заранее, остановится он или нет.

В случае, когда функция g является частичной, естественным образом функция $f(x_1, x_2, ..., x_n) = \mu y(g(x_1, x_2, ..., x_n, y) = 0)$ вычисляется с учетом следующего условия: если для $y = 0, 1, ..., i-1$ функция $g(x_1, x_2, ..., x_n, y) \neq 0$,а $g(x_1, x_2, ..., x_n, i)$ не определена, то и $f(x_1, x_2, ..., x_n)$ также не определена.

Определение. *Частично-рекурсивная функция называется общерекурсивной, если она всюду определена.*

Понятие частично-рекурсивной функции оказалось достаточной формализацией понятия вычислимой функции, что содержится в тезисе А. Черча

Тезис. *Всякая функция, вычислимая некоторым алгоритмом, частично-рекурсивна.*