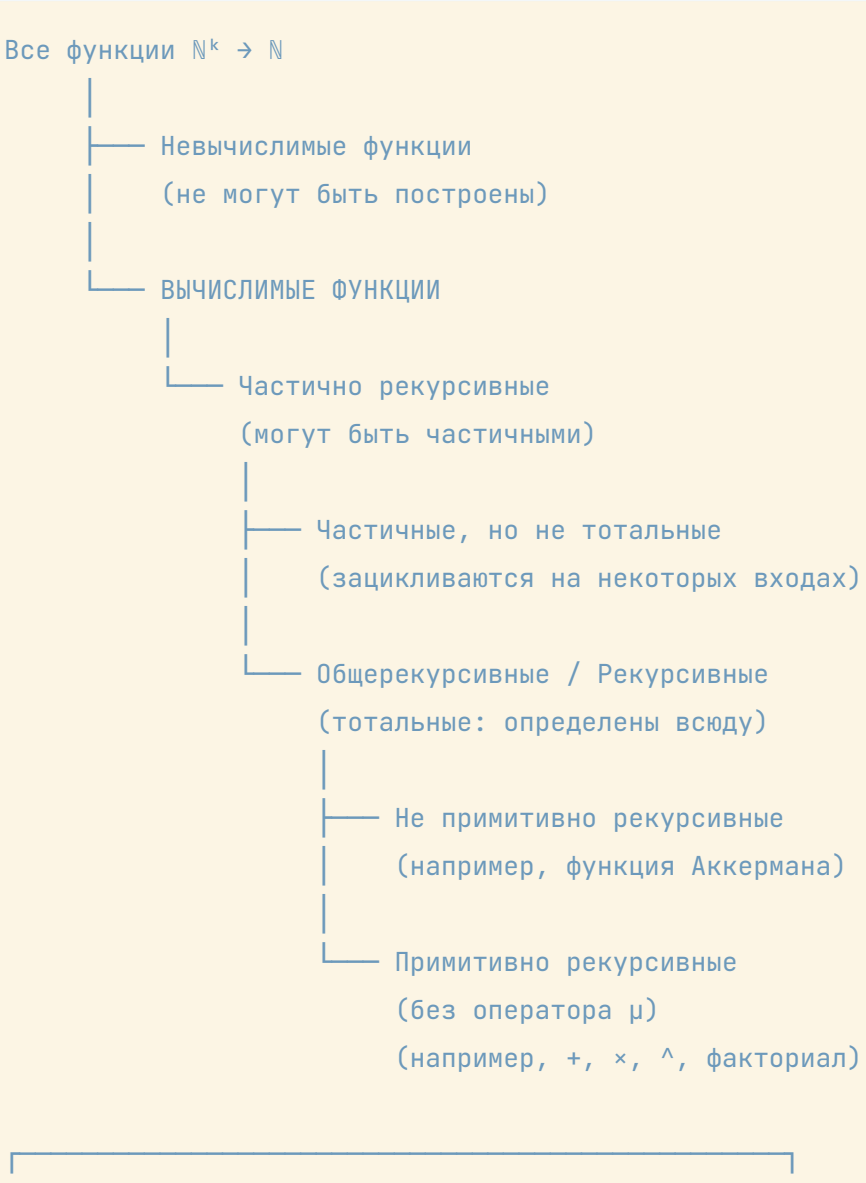
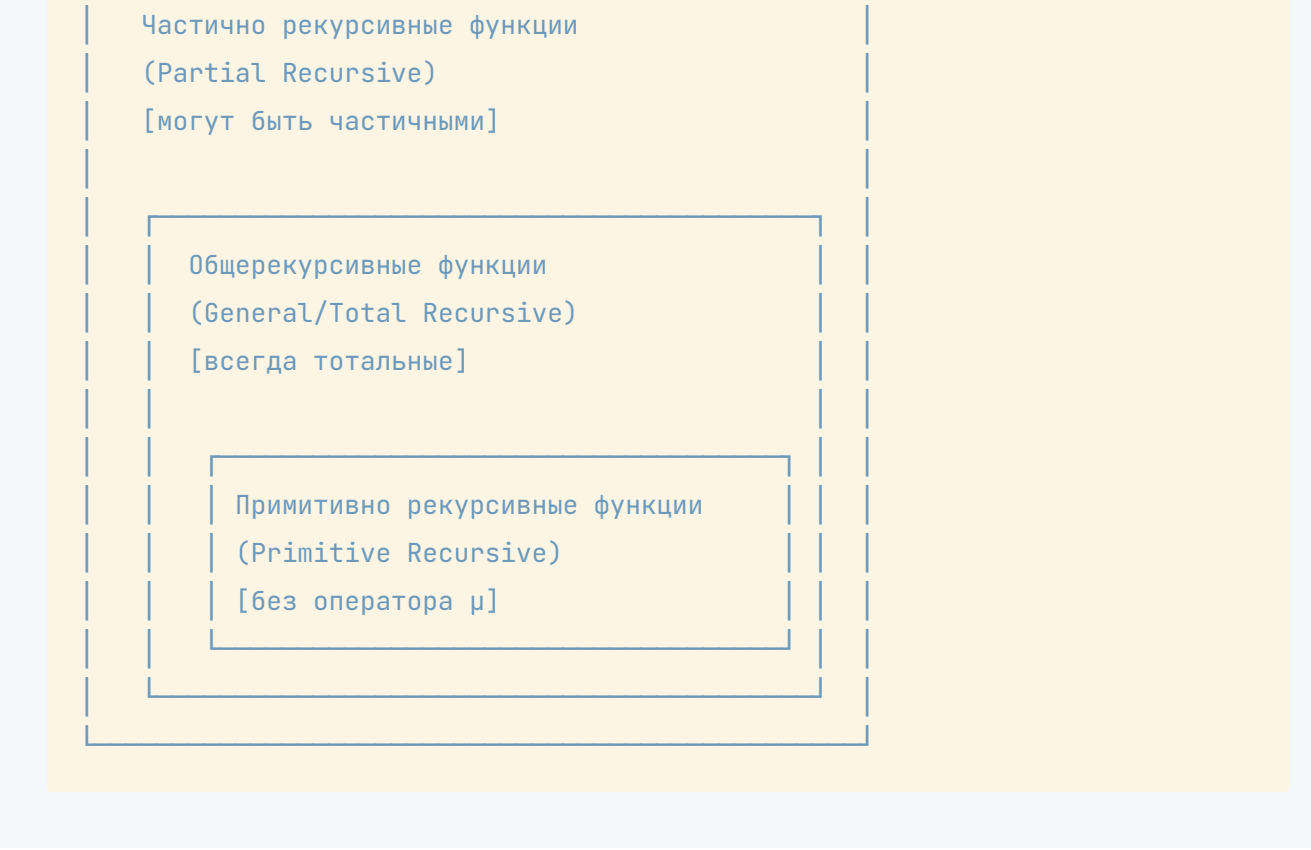
Рекурсивные функции (*recursive functions*) представляют собой одну из фундаментальных математических моделей вычислений, разработанную в 1930-х годах математиками Куртом Гёделем, Жаком Эрбраном и Стивеном Клини. Согласно тезису Чёрча-Тьюринга, множество вычислимых функций совпадает с классом частично рекурсивных функций.

Модель рекурсивных функций основана на **конструктивном подходе** к определению вычислимости: мы начинаем с нескольких тривиально вычислимых функций и постепенно строим всё более сложные функции, применяя определённые операции. Такое построение покрывает все вычислимые функции $f : \mathbb{N}^k \to \mathbb{N}, 0 \in \mathbb{N}$. Идея заключается в том, чтобы определить класс функций $f : \mathbb{N}^k \to \mathbb{N}$ **синтаксически** – через правила построения, а не через абстрактное понятие "алгоритма". Это даёт точное математическое определение вычислимости.

Иерархия функций >





Построение начинается с класса примитивно рекурсивных функций, который порождает множество вычислимых всюду определенных функций. Далее строятся частично рекурсивные функции, которые охватывают все множество вычислимых функций. Это множество содержит также и частичные функции и примитивно рекурсивные функции, включая всюду определенные функции, которые не были покрыты примитивно рекурсивными функциями. Всюду определенные функции из множества частично рекурсивных функций называются **общерекурсивными функциями** (*general recursive functions*).

Примитивно рекурсивные функции

Класс **примитивно рекурсивных** (*primitive recursive* – PR) функций задается индуктивно. Оно состоит из указания класса **базовых функций** и двух операторов (композиции и примитивной рекурсии), позволяющих строить новые примитивно рекурсивные функции на основе уже имеющихся. Примитивно рекурсивные функции заведомо являются всюду определенными вычислимыми, причем число шагов не просто конечно, а примитивно ограничено – существует примитивно-рекурсивная верхняя оценка. Важно понимать, что не все вычислимые функции $f : \mathbb{N}^k \to \mathbb{N}$ сводятся к примитивно рекурсивным функциям (например, функция Аккермана). PR является частным случаем частично рекурсивных функций, которые совпадают с множеством всех вычислимых функций $f : \mathbb{N}^k \to \mathbb{N}$, как частично, так и всюду определенных.

Класс **примитивно рекурсивных функций** $\mathcal{PR}$ – это наименьший класс функций $f : \mathbb{N}^k \to \mathbb{N}$ $(0 \in \mathbb{N})$, который:

1. Содержит базовые функции $z(x)$, $s(x)$ и проективные функции I_i^k.
2. Замкнут относительно композиции и примитивной рекурсии.

Базовые функции

Нулевая функция (*zero function*) $z : \mathbb{N}^n \to \mathbb{N}$ всегда возвращает 0, каким бы ни был аргумент x:

$$z(x_1, \ldots, x_n) = 0$$

Функция следования (*successor function*) или **функция преемника** $s : \mathbb{N} \to \mathbb{N}$ возвращает элемент, следующий за аргументом x:

$$s(x) = x + 1$$

Проекционные функции (*projection functions*) $I_i^n : \mathbb{N}^n \to \mathbb{N}$ возвращает аргумент с индексом i, т.е. для любых $n \in \mathbb{N}^+$ и $1 \leq i \leq n$:

$$I_i^n(x_1, \ldots, x_n) = x_i$$

При $n = 1$ проективная функция является тождественной $I(x) = x$.

В общем случае могли быть определены другие базовые функции на других множествах.

Операции

Композиция

Композиция (*composition*) или **подстановка** (*substitution*). Пусть f – функция от m переменных, а $g_1, \ldots, g_m$ – функции от n переменных. Тогда функция $h : \mathbb{N}^n \to \mathbb{N}$, определяемая равенством

$$h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$$

называется композицией функций $f, g_1, \ldots, g_m$. Коротко можно записать через оператор композиции

$$h = f \circ (g_1, \ldots, g_m) = \mathrm{Cn}[f, g_1, \ldots, g_m]$$

В общем случае, функция g_i не обязана быть функцией от всех аргументов $x_1, \ldots, x_n$. Но такая запись вполне уместна, если предполагать, что внутри функции будут отобраны нужные аргументы при помощи проективных функций.

☰ $x+2$ >

Предположим, мы хотим реализовать функцию $h(x,y)=x+2$. Можно представить ее в виде композиции функций I_1^2 и s:

$$h = s \circ s \circ I_1^2 = \mathrm{Cn}[s, \mathrm{Cn}[s, I_1^2]]$$

Примитивная рекурсия

Примитивная рекурсия (*primitive recursion*). Пусть $f : \mathbb{N}^m \to \mathbb{N}$ и $g : \mathbb{N}^{m+2} \to \mathbb{N}$ – определены как примитивно рекурсивные функции: $f, g \in \mathcal{PR}$. Тогда функция h определяемая операцией примитивной рекурсии из f и g, также примитивно рекурсивна. Примитивная рекурсия задается уравнением:

$$h(x_1, \ldots, x_m, 0) = f(x_1, \ldots, x_m)$$

$$h(x_1, \ldots, x_m, s(y)) = g(x_1, \ldots, x_m, y, h(x_1, \ldots, x_m, y))$$

или в векторной форме, если $\mathbf{x} = (x_1, \ldots, x_m)$:

$$h(\mathbf{x}, 0) = f(\mathbf{x}), \quad h(\mathbf{x}, s(y)) = g(\mathbf{x}, y, h(\mathbf{x}, y))$$

y – номер шага итераций. Функцию f можно рассматривать как исходную функцию в начале итерационного процесса (базовый случай). Функция g принимает m переменных x_i, номер шага итераций y и значение функции h на текущем шаге итерации, и возвращает значение функции h на следующем шаге итерации. При $m = 1$:

$$h(x, 0) = f(x), \quad h(x, s(y)) = g(x, y, h(x, y))$$

Функцию h, определяемую путем примитивной рекурсии из функций f и g, обозначим

$$h = \mathrm{Pr}[f, g]$$

Если рассматривается единственная переменная y, то схема принимает вид:

$$h(0) = f(), \quad h(s(y)) = g(y, h(y))$$

В этом случае x_i не появляется вовсе, а функция f нуля переменных есть некоторая константа $s(0)$ или $s(s(0))$ и т.д.

Другой способ – использовать фиктивный (*dummy*) x.

Примеры

Функции бинарных операторов

Сложение

Функция **сложения** add двух натуральных чисел. Определим $f(x)$ как $I(x)$, а $g(x)$ как композицию функций s и I^3_3, т.е. $\mathrm{Cn}[s, I^3_3]$, чтобы отобрать третий аргумент, переданный в g и прибавить к нему 1:

$$\mathrm{add}(x, 0) = I(x) = x$$

$$\mathrm{add}(x, s(y)) = s \circ I^3_3(x, y, \mathrm{add}(x, y))$$

Таким образом можно записать:

$$\mathrm{add} = \mathrm{Pr}[I, s \circ I^3_3] = \mathrm{Pr}[I, \mathrm{Cn}[s, I^3_3]]$$

Умножение

Функция **умножения** mul двух натуральных чисел. Определим $f(x)$ как $z(x)$, а g определим как композицию функции add и функций I^3_1, I^3_3, чтобы производилось сложение первого и третьего аргументов функции g:

$$\mathrm{mul}(x, 0) = z(x) = 0$$

$$\mathrm{mul}(x, s(y)) = \mathrm{add}(I^3_1, I^3_3) = \mathrm{add}(x, \mathrm{mul}(x, y))$$

Таким образом можно записать:

$$\mathrm{mul} = \mathrm{Pr}[z, \mathrm{add} \circ (I^3_1, I^3_3)] = \mathrm{Pr}[z, \mathrm{Cn}[\mathrm{add}, I^3_1, I^3_3]]$$

> константная функция, возвращающая 0, есть базовая функция z

Возведение в степень

$$\exp(x, 0) = s \circ z(x) = 1$$

$$\exp(x, s(y)) = \mathrm{mul}(I^3_1, I^3_3) = \mathrm{mul}(x, \exp(x, y))$$

> константная функция, возвращающая 1, определяется композицией базовых функций s и z. Для краткости просто пишем 0 и 1.

Гипероператоры

Продолжая подобным образом эту серию мы получаем мы получаем операцию **тетрации** (суперэкспоненту), которая представляет собой

стек из экспонент $x^{x^{\cdot^{\cdot^{\cdot}}}}$ (всего в стеке y "иксов"). Эту операцию удобнее записать используя оператор экспоненты $\uparrow$ и оператор тетрации ($\uparrow\uparrow$):

$$x \uparrow\uparrow y = x \uparrow x \uparrow x \cdots \uparrow x$$

Данную серию можно неограниченно продолжать, и так как мы каждый раз получаем функции, на основе примитивно рекурсивных функций, используя операции композиции и примитивной рекурсии, все функции серии являются примитивно рекурсивными.

Обобщенно, операторы этой серии называются гипероператорами n-го порядка. Сложение – гипероператор первого порядка, умножение – второго, экспонента – третьего. Далее идет тетрация $\uparrow\uparrow$, пентация $\uparrow\uparrow\uparrow$, гексация $\uparrow^4$ и т/д.

Другими примерами примитивно рекурсивных функций могут служить: факториал, предшествующее число (кроме случая 0), ограниченное вычитание. Обратные функции вообще говоря не сводятся к примитивно рекурсивным функциям. Для их реализации требуется операция минимизации.

Функции от одной переменной

Для функций от одной переменной в рекурсивной операции участвует только y.

☰ Предшествующий элемент >

Получим схему для функции pred от одной переменной, возвращающей **предшествующий элемент** (*predecessor*). Если значение аргумента 0, то возвращается 0:

$$\mathrm{pred}(0) = z = 0, \quad \mathrm{pred}(s(y)) = I_2^3$$

В данном примере $g(x, y, \mathrm{pred}(y)) = I_2^3$. Поэтому для входного значения $s(y) = y + 1$ функция вернет значение y. Так как мы имеем дело с функцией от одной переменно, переменная x является фиктивной. В другой нотации можно записать так:

$$\mathrm{pred} = \mathrm{Pr}[z, I_2^3]$$

☰ Факториал >

Исходя из того, что $0! = 1$ и $y! = y(y-1)!$, функции f и g определим следующим образом:

$$\text{factorial}(0) = 1, \quad \text{factorial}(s(y)) = \text{mul}(s \circ I_2^3, I_3^3)$$

Знаковая функция

Знаковая функция в данном случае будет возвращать 0 только в том случае, если аргумент равен 0, и 1 в противном случае:

$$\text{sgn}(y) = \begin{cases} 0, & \text{if} \quad y = 0, \\ 1, & \text{if} \quad y > 0. \end{cases}$$

Функции f и g определим следующим образом:

$$\text{sgn}(0) = 0, \quad \text{sgn}(s(y)) = s \circ z = 1$$

Все аргументы в рекурсивной формуле фиктивны.

Другие функции

Усеченная разность

Усеченная разность (*Truncated subtraction*) между x и y возвращает значение разности, если $x \geq y$, и 0 в противном случае:

$$\text{sub}(x, y) = x \dot{-} y$$

Функции f и g определим следующим образом:

$$\text{sub}(x, 0) = x = I_1^2, \quad \text{sub}(x, s(y)) = \text{pred}[\text{sub}(x, y)]$$

Замкнутость данной операции обеспечивается тем обстоятельством, что $\text{pred}(0) = 0$. Поэтому, если $x < y$, операция не будет выводить за пределы множества $\mathbb{N}_0$.

Тождественное равенство

На основе знаковой функции и усеченной разности легко создать функцию $\text{eq}(x, y)$, которая будет возвращать 1, если x и y равны, и 0 – в противном случае. Идея заключается в том, что если x и y равны, то обе разности $x \dot{-} y$ и $y \dot{-} x$ будут обращаться в ноль. Тогда знаковая функция от этих разностей будет давать 0. Если же x и y не равны, то знаковая функция от двух разностей будет различной.

$$\mathrm{eq}(x,0) = \mathrm{sub}(s \circ z(I_1^2), \mathrm{sgn}(I_1^2)) = 1 \dot{-} \mathrm{sgn}(x)$$

$$\mathrm{eq}(x,y) = 1 \dot{-} \mathrm{sgn}(x \dot{-} y) \cdot \mathrm{sgn}(y \dot{-} x)$$

Функция h записана в упрощенной нотации, при котором произведение mul обозначено через бинарный оператор $\cdot$. Если бы мы в функциях g и h не вычитали второе слагаемое от 1, то в результате получили бы функцию neq, которая является инверсией функции eq.

Функции eq и neq могут служить характеристическими функциями для бинарных отношений "равны" и "не равны", определенных на $\mathbb{N}_0 \times \mathbb{N}_0$.

☰ Отношения порядка ›

Еще проще получить характеристические функции для отношений порядка. Получим функцию $\mathrm{geq}(x,y)$ – отношение "больше или равно" (*greater or equal*), которая возвращает 1, если $x \geq y$, и 0 – в противном случае.

$$\mathrm{geq}(x,0) = s \circ z(I_1^2) = 1, \quad \mathrm{geq}(x, s(y)) = \mathrm{sgn} \circ \mathrm{sub}(I_1^2, I_2^2)$$

В упрощенной нотации:

$$\mathrm{geq}(x,y) = \mathrm{sgn}[x \dot{-} (y \dot{-} 1)]$$

Для отношения "больше" $\mathrm{gt}(x,y)$ (*greater than*), т.е. $x > y$:

$$\mathrm{gt}(x,0) = \mathrm{sgn} \circ I_1^2 = \mathrm{sgn}(x)$$

$$\mathrm{gt}(x, s(y)) = \mathrm{sgn} \circ \mathrm{sub}(I_1^2, s \circ I_2^2)$$

В упрощенной нотации:

$$\mathrm{gt}(x,y) = \mathrm{sgn}(x \dot{-} y)$$

Меняя местами аргументы x и y или инвертируя, можно получить характеристические фунции и для других отношений порядка $(<, \leq)$.

☰ Тернарный оператор ›

Тернарный оператор, или функция случая $\mathrm{case}(x,y,z)$, является примитивно рекурсивной:

$$\mathrm{case}(x,y,z) = \begin{cases} x, & \text{if} \quad z = 0, \\ y, & \text{if} \quad z > 0, \end{cases}$$

Мы можем построить эту функцию используя лишь композицию функций рассмотренных выше: умножение, сложение, симметрическая разность, знаковая функция, константная функция:

$$\mathrm{case}(x, y, z) = x \cdot (1 \dot{-} \mathrm{sgn}(z)) + y \cdot \mathrm{sgn}(z)$$

Заметим, что на этот раз через z мы обозначили переменную, а не нулевую функцию.

Минимум

Минимальное значение пары x, y может быть определено следующим образом:

$$\min(x, 0) = z(x) = 0$$

$$\min(x, s(y)) = \mathrm{sub}(s(I_2^3), \mathrm{sub}(s(I_2^3), I_1^3))$$

Функцию h можно было бы упрощенно записать так:

$$\min(x, y) = y \dot{-} (y \dot{-} x)$$

Максимум

$$\max(x, 0) = I_1^2 = x$$

$$\max(x, s(y)) = \mathrm{add}(s(I_2^3), \mathrm{sub}(I_1^3, s(I_2^3)))$$

Функцию h можно было бы упрощенно записать так:

$$\min(x, y) = y + (x \dot{-} y)$$

Расстояние

В качестве функции расстояния будем рассматривать абсолютное значение разности:

$$\mathrm{dist}(x, y) = |x - y|$$

Функции f и g определим следующим образом:

$$\mathrm{dist}(x, 0) = I_1^2 = x$$

$$\mathrm{dist}(x, s(y)) = \mathrm{add}(\mathrm{sub}(I_1^3, s(I_2^3)), \mathrm{sub}(s(I_2^3), I_1^3))$$

Функцию h можно было бы упрощенно записать так:

$$\operatorname{dist}(x, y) = (x \dot{-} y) + (y \dot{-} x)$$

Частично рекурсивные функции

Частично рекурсивные функции (*partial recursive functions*) являются расширением класса примитивно рекурсивных функций путем добавления еще одной операции (минимизации) для продуцирования новых функций. Согласно тезису Чёрча-Тьюринга, класс частично рекурсивных функций (μ-рекурсивных) совпадает с множеством вычислимых по Тьюрингу функций, как всюду определенных, так и частично определенных функций ❐.

Операция минимизации

Минимизация (оператор μ или Mn) позволяет получить частично рекурсивные функции из других частично рекурсивных. Пусть $g : \mathbb{N}^{n+1} \to \mathbb{N}$ – всюду определенная функция. Минимизация определяет частично рекурсивную функцию $f : \mathbb{N}^n \to \mathbb{N}$:

$$f(\mathbf{x}) = \mu\, y[g(\mathbf{x}, y) = 0] = \mathrm{Mn}_y[g](\mathbf{x})$$

Выражение $\mu\, y[g(\mathbf{x}, y) = 0]$ определяет наименьшее y, при котором $g(\mathbf{x}, y) = 0$, если такое значение существует. Если такого y не существует, то f при данном $\mathbf{x}$ будет не определено.

Можно определить минимизацию через частично определенную функцию g. Тогда добавляется еще одно условие: для всех $z < y$ функция $g(\mathbf{x}, z)$ должна быть определена и не равна нулю $g(\mathbf{x}, z) \neq 0$. В противном случае, функция f не будет определена при x.

Таким образом, при фиксированном $\mathbf{x} = (x_1, \ldots, x_n)$ функция $f(x_1, \ldots, x_n)$ возвращает минимальное значение y^* последнего аргумента функции $g(x_1, \ldots, x_n, y)$, при котором g принимает значение 0, при условии, что функция определена при всех значениях $y < y*$. Если нет такого минимального значения y^* или если не для всех $y < y^*$ функция g определена при $\mathbf{x}$, то и значение f при $\mathbf{x}$ не определено.

Обоснование вычислимости >

Если g является вычислимой всюду определенной или частичной функцией, то таковой будет и f. Рассмотрим подробнее процесс вычисления f при заданном $\mathbf{x}$. Начиная с $y = 0$ мы последовательно

вычисляем $g(\mathbf{x}, 0)$, $g(\mathbf{x}, 1)$, $g(\mathbf{x}, 2)$ и т.д. до тех пор, пока не достигнем такого значения y^*, при котором $g(\mathbf{x}, y^*) = 0$. Если $\mathbf{x}$ принадлежит области определения f, то такое y^* существует, и число шагов, необходимых для вычисления $f(\mathbf{x})$, будет будет ограничено сверху суммой числа шагов, необходимых для вычисления $g(\mathbf{x}, 0)$, числа шагов для вычисления $g(\mathbf{x}, 1)$ и т.д. до тех пор, пока не будет вычислено $g(\mathbf{x}, y^*)$, т.е. число шагов для вычисления $f(\mathbf{x})$ будет конечным. Если же $\mathbf{x}$ не входит в область определения f, то это может быть по двум причинам. С одной стороны, может оказаться, что вся последовательность $g(\mathbf{x}, 0)$, $g(\mathbf{x}, 1)$, $g(\mathbf{x}, 2), \ldots$ является определенной, но среди них нет нулей. С другой стороны, может оказаться, что для некоторого i последовательность $g(\mathbf{x}, 0)$, $g(\mathbf{x}, 1)$, $\ldots$, $g(\mathbf{x}, i-1)$ определенна, но не содержит нулей, а $g(\mathbf{x}, i)$ – не определено. В обеих случаях вычисление $f(\mathbf{x})$ будет включать процесс, который продолжается вечно, не выдавая результат.

Регулярные функции >

Всюду определенная функция $g : \mathbb{N}^{n+1} \to \mathbb{N}$ называется **регулярной** (*regular*), если для каждого $\mathbf{x} = (x_1, \ldots, x_n)$ существует такой y^*, при котором $g(\mathbf{x}, y^*) = 0$. Если g – регулярная функция, то $f = \mathrm{Mn}[g]$ будет всюду определенной функцией (при любых $\mathbf{x}$). Более того, если g является всюду определенной функцией, то $f = \mathrm{Mn}[g]$ будет всюду определена в том и только том случае, если g – регулярная.

Example

Умножение $g = \mathrm{mul}(x, y)$ является регулярной функцией: для любого x при $y^* = 0$ имеем $g(x, y^*) = 0$.

Сложение $g = \mathrm{sum}(x, y)$ не является регулярной функцией: $g = 0$ только при $x = y = 0$; для всех прочих значений x нет такого y, при котором $g(x, y) = 0$.

Операция минимизации позволяет получить обратные функции.

Целая часть от деления >

Определим функцию $\mathrm{Div}(x, y)$ – целая часть от деления x на y. Исходная функция может быть выбрана так:

$$g(x, y, z) = [y(z+1) \leq x]$$

Функция $g : \mathbb{N}^3 \to \{0, 1\}$ всюду определена на $\mathbb{N}^3$, $(0 \in \mathbb{N})$. При фиксированных x, y значение $g(x, y, z)$ будет равно 1 для всех z, при которых значение выражения $y(z+1)$ не превышает x. Если увеличивать z, то, как только значение выражения $y(z+1)$ превысит x, функция g вернет 0. Это наименьшее z, при котором g обращается в 0, и будет целой частью от деления x на y:

$$z = \left\lfloor \frac{x}{y} \right\rfloor$$

Таким образом,

$$\mathrm{Div}(x, y) = \mathrm{Mn}_z[g] = \mathrm{Mn}_z\left[y(z+1) \leq x\right]$$

Следует отметить, что функции получения целой части от деления и остатка, вообще говоря, можно реализовать через примитивно рекурсивные функции. Правда, для этого необходимо тотализировать эту функцию, положив $\mathrm{Div}(x, 0) = 0$, в противном случае Div при $y = 0$ не определен, и потому никак не может быть примитивно рекурсивной функцией.