Практика №3 по курсу «Теория алгоритмов»
«Рекурсивные функции»

Группы ФТ-301, ФТ-302

На прошлых лекциях и практиках рассматривались машины Тьюринга — математическая модель, построенная для формализации понятия алгоритма. Параллельно с Тьюрингом развивался альтернативный **функциональный** подход для формализации понятния алгоритмов. Алонзо Чёрч вместо «императивного» определения абстрактного вычислителя и его возможностей решил определить класс базовых функций, которые естественно считать алгоритмически вычислимыми, а также определить класс операций, которые можно совершать с такими функциями, чтобы они не теряли данного свойства. В результате данной работы сформулировались понятия примитивно рекурсивных и частично рекурсивных функций, где последние соответствуют языкам, перечислимым с помощью машины Тьюринга (соответствие достигается путем некоторых трансформаций между входом МТ и натуральными числами).

1 Примитивно рекурсивные функции

Определим базовые функции, которые будут лежать в основе нашего класса примитивно рекурсивных функций:

- $\mathbf{0}() = 0$ — константная 0-арная функция, значение которой равно нулю
- $\mathbf{S} : \mathbb{N} \mapsto \mathbb{N}, \mathbf{S}(x) = x + 1$ — функция инкремента
- $\pi_\mathbf{i}^\mathbf{n} : \mathbb{N}^n \mapsto \mathbb{N}, \pi_\mathbf{i}^\mathbf{n}(x_1, x_2, ..., x_n) = x_i$ — семейство функций проекции i-го аргументра из n

Определим также правила, позволяющие комбинировать примитивно рекурсивные функции:

- Подстановка — если функции $\mathbf{f} : \mathbb{N}^k \mapsto \mathbb{N}$ и $\mathbf{g_1}, \mathbf{g_2}, ..., \mathbf{g_k} : \mathbb{N}^\ell \mapsto \mathbb{N}$ примитивно рекурсивны, то функция $\mathbf{h} : \mathbb{N}^\ell \mapsto \mathbb{N}, \mathbf{h}(x_1, x_2, ..., x_\ell) = \mathbf{f}(\mathbf{g_1}(x_1, x_2, ..., x_\ell), ..., \mathbf{g_k}(x_1, x_2, ..., x_\ell))$ также является примитивно рекурсивной
- Примитивная рекурсия — если функции $\mathbf{f} : \mathbb{N}^k \mapsto \mathbb{N}$ и $\mathbf{g} : \mathbb{N}^{k+2} \mapsto \mathbb{N}$ примитивно рекурсивны, то функция $\mathbf{h} : \mathbb{N}^{k+1} \mapsto \mathbb{N}, \mathbf{h}(x_1, x_2, ..., x_k, y) = \begin{cases} \mathbf{f}(x_1, x_2, ..., x_k) & \text{, если } y = 0 \\ \mathbf{g}(x_1, x_2, ..., x_k, y-1, h(x_1, x_2, ..., x_k, y-1)) & \text{, иначе} \end{cases}$

В результате класс примитивно рекурсивных функций PR можно определить как минимальный по включению класс функций натуральных аргументов такой, что $\{\mathbf{0}, \mathbf{S}\} \cup \{\pi_\mathbf{i}^\mathbf{n}\}_{n \in \mathbb{N} \wedge 1 \le i \le n} \in PR$ и PR замкнут относительно операций подстановки и примитивной рекурсии.

1.1 Пример. Факториал

Для примера построим из базовых функций с помощью правил подстановки и примитивной рекурсии функцию, вычисляющую факториал числа n. Для этого вспомним рекурсивное определение факториала:

$$\mathbf{fact}(n) = \begin{cases} n \cdot \mathbf{fact}(n-1) & \text{, если } n > 0 \\ 1 & \text{, иначе} \end{cases}$$

Правило примитивной рекурсии по сути представляет из себя обыкновенную математическую индукцию (база определяется функцией $\mathbf{f}$, а шаг — $\mathbf{g}$).Попробуем описать рекурсивный переход для факториала с помощью примитивной рекурсии. Для этого необходимо определить $\mathbf{f}$ как $\mathbf{f}() = \mathbf{S}(\mathbf{0}) = 1$ (здесь мы воспользовались базовыми функциями $\mathbf{S}$ и $\mathbf{0}$, а также правилом композиции функций), а $\mathbf{g}(\cdot, \cdot)$ определить как $\mathbf{g}(y, z) = prod(\mathbf{S}(\pi_\mathbf{1}^\mathbf{2}(y, z)), \pi_\mathbf{2}^\mathbf{2}(y, z)) = (y + 1) \cdot z$ (здесь мы также воспользовались композицией и неявно построили дополнительную примитивно рекурсивную функцию $\mathbf{t}(y, z) = \mathbf{S}(\pi_\mathbf{1}^\mathbf{2}(y, z))$). В таком случае $\mathbf{h}$, построенная с помощью правила примитивной рекурсии из функций $\mathbf{f}$ и $\mathbf{g}$ будет вычислять значение $\mathbf{fact}(n)$.

В этом примере мы пользовались тем, что $prod(x) \in PR$, однако можно легко доказать примитивную рекурсивность произведения аналогичным образом (для этого потребуется также доказать примитивную рекурсивность сложения).

1.2 Упражнение. Целочисленное деление

Будем считать доказанными факты, что остаток отделения на число, условный оператор, операторы сравнения являются примитивно рекурсивными функциями.

Теперь докажем примитивную рекурсивность функции $\mathbf{div_d}(n) = \lfloor \frac{n}{d} \rfloor$. Для этого рассмотрим два подхода:

1. Заметим, что $\mathbf{div_d}(n) = \mathbf{div_d}(n-1) + [n \equiv_d 0]$. Данное соотношение хорошо ложиться на паттерн примитивной рекурсии и можно легко построить функцию $\mathbf{h}(n) = \mathbf{div_d}(n)$, используя $\mathbf{f}() = \mathbf{0}$ и $\mathbf{g}(y, z) = z + [\mathbf{rem}(y+1, d) = 0]$

2. Для второго подхода воспользуемся следующим определением: $\mathbf{div_d}(n) = \max\{k \mid k \cdot d \le n \land k \in \mathbb{Z}^+\}$. Определим следующую функцию: $\mathbf{m}(n, d, b) = \max\{k \mid k \cdot d \le n \land 0 \le k \le b\}$. Легко заметить, что $\mathbf{m}(n,d,b) = \begin{cases} \mathbf{0} & \text{, если } b = 0 \\ \mathbf{m}(n,d,b-1) & \text{, если } b \cdot d > n \\ \mathbf{b} & \text{, иначе} \end{cases}$ Несложно доказать примитивно рекурсивность функции $\mathbf{m}$, а дальше остается воспользоваться соотношением $\mathbf{div_d}(n) = \mathbf{m}(n, d, n)$.

   Данный пример интересен тем, что в нём неявно используется оператор ограниченной минимизации. Определим оператор $\mu_{\le b(x)}$, после применения которого к предикату $\mathbf{p} : \mathbb{N}^{k+1} \mapsto \{0, 1\}$ получается функция $\mathbf{f} : \mathbb{N}^k \mapsto \mathbb{N}$ следующим образом: $\mathbf{f}(x) = \min\{y \le b(x) \mid \mathbf{p}(x, y)\}$, где $b(x)$ — некоторая примитивно рекурсивная функция (будем считать, что $\mathbf{f}(x) = b(x) + 1$, если $\{y \le b(x) \mid \mathbf{p}(x, y)\} = \varnothing$).

   Таким образом нетрудно представить функцию $\mathbf{div_d}(n)$ как результат применения оператора минимизации к предикату $\mathbf{p_d}(n, k) = [k \cdot d > n]$, а именно $\mathbf{div_d}(n) = (\mu_{\le n} \mathbf{p_d})(n) - 1$. Ход доказательства примитивной рекурсивности ограниченного оператора минимизации в точности повторяет процесс построения функции $\mathbf{m}(n, d, b)$.

2 Примитивно рекурсивная нумерация пар

Рекурсивные функции оперируют только числами, поэтому необходимо придумать способ работы с кортежами чисел в рамках заданных правил. Для начала поймем, как можно построить примитивно рекурсивные функции $\mathbf{enc}(x_1, x_2, ..., x_n)$ и $\mathbf{dec}(code, i)$ для шифрования и расшифровки конкретного элемента кортежа. Данные функции должны удовлетворять следующему соотношение: $\mathbf{dec}(\mathbf{enc}(x_1, x_2, ..., x_i, ..., x_n), i) = x_i$.

Для построения этих функций воспользуемся основной теоремой арифметики: каждое натуральное число имеет единственное разложение на простые множители $n = a_1^{d_1} \cdot a_2^{d_2} \cdots a_k^{d_k}$, где $a_1 < a_2 < ... < a_k$ — простые числа. Таким образом, если занумеровать простые числа по порядку ($p_1 = 2, p_2 = 3, p_3 = 5, ...$), то числу $n = p_{i_1}^{d_1} \cdot p_{i_2}^{d_2} \cdots p_{i_k}^{d_k}$ соответствует кортеж длины i_k, где на позициях $i_1, i_2, ..., i_k$ стоят числа $d_1, d_2, ..., d_k$, а на все остальных — нули.

Таким образом функция $\mathbf{enc}(x_1, x_2, ..., x_n) = p_1^{x_1} \cdot p_2^{x_2} \cdots p_n^{x_n}$ и для доказательства её примитивной рекурсивности достаточно понять, что нахождение i-го простого числа - примитивно рекурсивная функция (ясно, что функция проверки на простоту — примитивно-рекурсивная, а значит $\mathbf{nth\_prime}(x) = (\mu_{\le nth\_prime(x-1)} \mathbf{is\_prime}(\mathbf{nth\_prime}(x-1) + y + 1))(x)$, что является корректным определением из-за постулата Бертрана).

Для функции $\mathbf{dec}(code, i)$ также необходимо воспользоваться функцией **nth_prime**, после чего найти степень вхождения этого простого числа в *code*.

2.1 Упражнение. Совместная рекурсия

Пусть функции $\mathbf{f}$ и $\mathbf{g}$ определены следующим образом:

$$\begin{cases} \mathbf{f}(0) = A \\ \mathbf{g}(0) = B \\ \mathbf{f}(y) = \mathbf{F}(\mathbf{f}(y-1), \mathbf{g}(y-1), y-1) \\ \mathbf{g}(y) = \mathbf{G}(\mathbf{f}(y-1), \mathbf{g}(y-1), y-1) \end{cases}$$

где $\mathbf{F}, \mathbf{G}$ — примитивно рекурсивные функции.

Докажем, что $\mathbf{f}, \mathbf{g}$ также являются примитивно рекурсивными функциями. Для этого рассмотрим функцию $\mathbf{h}(y) = \mathbf{enc}(\mathbf{f}(y), \mathbf{g}(y))$. Ясно, что $\mathbf{f}(y) = \mathbf{dec_1}(\mathbf{h}(y))$ и $\mathbf{g}(y) = \mathbf{dec_2}(\mathbf{h}(y))$. Докажем тогда примитивную рекурсивность функции $\mathbf{h}$. Заметим, что

$$\mathbf{h}(y) = \mathbf{enc}(\mathbf{F}(\mathbf{dec_1}(\mathbf{h}(y-1)), \mathbf{dec_2}(\mathbf{h}(y-1)), y-1), \mathbf{G}(\mathbf{dec_1}(\mathbf{h}(y-1)), \mathbf{dec_2}(\mathbf{h}(y-1)), y-1))$$

откуда следует примитивная рекурсивность функции $\mathbf{h}$.

3 Рекурсивные функции

Определение **частично рекурсивных функций** отличается от определения примитивно рекурсивных функций только тем, что к двум правилам композиции и примитивной рекурсии добавляется третее правило **неограниченной** минимизации:

- Оператор **неограниченной** минимизации μ применяется к $k+1$-арному предикату $\mathbf{p} : \mathbb{N}^{k+1} \mapsto \{0, 1\}$, в результате чего получается частичная k-арная функция $\mathbf{g}(x_1, x_2, ..., x_k) = \min\{y \mid \mathbf{p}(x_1, x_2, ..., x_n, y) = 0\}$. Важно понимать, что в отличие от правил композиции и примитивной рекурсии, применение оператора неограниченной минимизации может превратить полную функцию в частичную (то есть функцию, определённую на некотором множестве $D \subset \mathbb{N}^k$).

Всюду определённые частично рекурсивные функции называются **общерекурсивными**.

3.1 Функция Аккермана

Ясно, что так как множество частично рекурсивных функций R содержит частичные функции, то любая частичная функция $\mathbf{f} \in R$ не содержится во множестве PR. Остается понять, существуют ли общерекурсивные функции, не принадлежащие множеству PR. Одним из примеров таких функций является функция Аккермана:

$$\mathbf{A}(m, n) = \begin{cases} n + 1 & , m = 0 \\ \mathbf{A}(m - 1, 1) & , m > 0, n = 0 \\ \mathbf{A}(m - 1, \mathbf{A}(m, n - 1)) & , m > 0, n > 0 \end{cases}$$

Таким образом $\mathbf{A} \in R \setminus PR \neq \varnothing$.