ЛЕКЦИЯ 12

ЧАСТЬ 2. ТЕОРИЯ АЛГОРИТМОВ

Глава 3. Основы теории алгоритмов

3.1. Понятие алгоритма и неформальная вычислимость

Определение 1:

Алгоритм - способ преобразования представления информации, другими словами, формальное предписание, согласно которому можно получить решение задачи. Алгоритм может решать не только частичную (одну задачу), но и целый класс задач (например с помощью параметров и исходных данных).

Основные особенности алгоритма:

1) Определенность – разбиение на этапы

2) Ввод- алгоритм имеет некоторое число исходных данных.

3) Вывод – один или несколько результатов, сведенных к входным данным.

4) Детерминированность – после каждого шага однозначно определено, что делать дальше.

Замечание 1:

Для некоторых задач алгоритма решения не существует.

Например, в 1970 г. Ю.В. Матиясевич показал, что не существует алгоритма, выясняющего, имеет ли полиномиальное дифференциальное уравнение целочисленное решение.

Замечание 2:

Алгоритмы можно считать связанными с натуральными числами и нулем, т.к. любые объекты можно закодировать целыми числами.

Пусть N={0,1,2,…} .Объекты будут функциями с областью определения$D_f \subseteq N^n$ (n-целое положительное число) и областью значений $R_f \subseteq N$, т.е. имеем n-местные частичные функции.

Эти функции частичные, если $D_f \subset N^n$, или всюду определенные, если $D_f = N^n$.

Определение 2:

Функция $f: N^n \to \mathrm{N}$ будет называться «эффективно вычислимой», если существует алгоритм, вычисляющий эту функцию f, который должен удовлетворять условиям:

1) Вычисление $f(\bar{x}) = \mathrm{f}(x_1, x_2, \dots, x_n,)$ должно закончиться за конечное число шагов для входного вектора $(\bar{x})$.

2) Если $(\bar{x}) \notin D_f$, то алгоритм никогда не заканчивается.

Замечание 3:

Эффективно вычисляемые функции не есть обязательно практически вычисляемые, поэтому существуют различные формализации понятия алгоритма.

3.2. Частично-рекурсивные функции (подход Геделя и Клини к понятию вычислимости функций).

Идея Геделя-Клини: получить все вычисляемые функции из существенно ограниченного множества базисных функций с помощью простейших алгоритмических средств:

А) Множество исходных функций

1) Постоянная функция: $0(x) = 0$

2) Одноместная функция следования: $S(x) = 0$

3) Функция проекции: $P_{ri}(x) = \mathrm{x}_i$, где $1 \leq i \leq n$

Б) Алгоритмические способы построения (алгоритмические операторы)

1) Оператор «суперпозиции»

$f(x_1, x_2, \dots, x_n) = \varphi(\ g_1(x_1, x_2, \dots, x_n), \dots, g_m(x_1, x_2, \dots, x_n)\ ),$

который получается из $\varphi(y_1, y_2, \dots, y_n)$, где $y_i = g_i(x_1, x_2, \dots, x_n)$ где $1 \le i \le m$.

2) оператор «примитивной рекурсии»

Строится (n+1) местная функция f по схеме:

а) $f(x_1, x_2, \dots, x_n, 0) = h(x_1, x_2, \dots, x_n)$;

б) $f(x_1, x_2, \dots, x_n, y+1) = g\big(x_1, x_2, \dots, x_n, y, f(x_1, x_2, \dots, x_n, y)\big)$

При n=0, f(0)=a=const, f(y+1)=g(y,f(y)).

<u>Замечание 1</u>: **Способы 1-2 порождают всюду определенные функции**.

3) Оператор «минимизации»

Частичной функции $f: N^n \to \mathrm{N}$ ставится в соответствие частичная функция $h: N^n \to \mathrm{N}$, которая определяется так:

а) область определения

$$D_h = \{< x_1, x_2, \dots, x_n > | \ \exists_{x_{n+1}} \ge 0 \text{ и } < x_1, x_2, \dots, x_n, y > \in D_f \text{ для } \forall_{\mathrm{y}} \le x_{n+1} \}$$

б) $h(x_1, x_2, \dots, x_n)$ = наименьшее значение *y*, при котором $f(x_1, x_2, \dots, x_{n+1}) = 0$.

Оператор минимизации обозначается как

$$h(x_1, x_2, \dots, x_n) = \mu y\ [f(x_1, x_2, \dots, x_n, y) = 0]$$

Таким образом, эти **три оператора являются механизмом построения частично-рекурсивных функций**, которые решаются за конечное число шагов.

<u>**Определение 1**</u>**:** Если же конкретная **частично-рекурсивная** функция получается **всюду определенной**, тогда она называется «**общерекурсивной**».

<u>**Определение 2:**</u> Если **частично-рекурсивная** функция получается **без оператора минимизации**, то она называется «**примитивно-рекурсивной**».

Получение частично рекурсивных функций с помощью различных методов.

а) Введение фиктивных переменных:

Если $g(x_1, x_2)$ примитивно рекурсивная функция и

$f(x_1, x_2, x_3) = g(x_1, x_2)$, то $f(x_1, x_2, x_3)$ тоже примитивно рекурсивная функция.

б) Перестановка переменных:

Если $g(x_1, x_2)$ примитивно рекурсивная функция и

$f(x_1, x_2,) = g(x_1, x_2)$, то $f(x_2, x_1)$ тоже примитивно рекурсивная функция.

в) Отождествление переменных:

Если $g(x_1, x_2, x_3)$ примитивно рекурсивная функция и

$f(x_1, x_2,) = g(x_1, x_2, x_3)$, то $f(x_1, x_2,)$ тоже примитивно рекурсивная функция.

<u>Замечание 2</u>: Все функции, которые можно получить из базисных функций за конечное число шагов только с помощью 3-х

операторов, являются частично рекурсивными функциями, которые позволяют получить новые частично рекурсивные функции.

Замечание 3: Существуют критерии, которые позволяют получать частичную рекурсивность сразу для широкого класса функций.

Замечание 4: Используя оператор минимизации, можно получать частично определенные функции из всюду определенных функций.

Замечание 5: С помощью оператора минимизации можно построить больше функций, чем только с помощью операторов суперпозиции и рекурсий, т.к. эти два оператора порождают из всюду определенных функций только всюду определенные функции.

Замечание 6: существуют общерекурсивные функции, которые нельзя простроить без 3-го оператора минимизации.

Например, функция Аккермана:

$f(0, y) = y + 1$, $f(x + 1, 0) = f(x, 1)$,

$f(x + 1, y + 1) = f\big(x, (f(x + 1, y)\big)$