1 АБСТРАКТНАЯ СЛОЖНОСТЬ ВЫЧИСЛЕНИЙ

1.1 Геделевские (главные) нумерации

В этом разделе напомним некоторые понятия и результаты из теории алгоритмов. Пусть $\mathcal{F}$ – класс одноместных частично рекурсивных функций (ч.р.ф.). Отображение $f : \mathbf{N} \to \mathcal{F}$ называется *вычислимой нумерацией* этого класса, если существует ч.р.ф. $F(i,x)$, называемая *универсальной*, такая что

$$\forall i \forall x [f(i)(x) = F(i,x)]$$

Функцию $f(i)$ будем обозначать f_i, а нумерацию, задаваемую $F(i,x)$ – через $\{f_i\}$. Число i называется *номером* функции f_i. Аналогично вводится понятие вычислимой нумерации и для классов k-местных ч.р.ф.

Определение 1 *Пусть $\{f_i\}$ и $\{g_j\}$ две вычислимые нумерации одного класса ч.р.ф. $\mathcal{F}$. Скажем, что $\{f_i\}$ сводится к нумерации $\{g_j,\}$ если существует рекурсивная функция h (сводящая функция, транслятор) такая, что*

$$g_{h(i)} \;=\; f_i.$$

Всякую вычислимую нумерацию $\{f_i\}$ класса всех ч.р.ф. можно рассматривать как некоторый универсальный язык программирования, а соответствующую ей универсальную функцию $F(i,x)$ – как интерпретатор этого языка. Сводимость нумераций соответствует транслируемости языков программирования. Среди всех нумераций (языков программирования) выделим те, к котрым сводятся все остальные.

Определение 2 *Нумерация $\{f_i\}$ класса всех ч.р.ф. называется* геделевской или главной, *если к ней сводится всякая вычислимая нумерация этого класса.*

Геделевские нумерации достаточно просто можно охарактеризовать.

Теорема 1 *Вычислимая нумерация $\{f_i\}$ класса всех ч.р.ф. является геделевской тогда и только тогда, когда для нее справедлива следующая теорема об итерации Клини (s-m-n теорема):*
для любых $m,n \geq 1$ существует рекурсивная функция s_n^m от $m+1$ переменных, такая что для всех $x, y_1, ..., y_m,\ z_1, ..., z_n$

$$f_x(y_1, ..., y_m, z_1, ..., z_n) = f_{s_n^m(x,y_1,...,y_m)}(z_1, ..., z_n).$$

В частности, в случае $m=1, n=1$ теорема об итерации утверждает существование функции $s(x,y)$ такой, что $f_x(y,z) = f_{s(x,y)}(z)$. Нетрудно понять, что такая функция легко может быть построена для естественной нумерации машин Тьюринга и для всех "обычных"языков программирования. Поэтому можно высказать тезис о том, что *всякий "разумный"способ задания всех вычислимых функций (язык программирования) задает геделевскую нумерацию.*
Из определения непосредственно следует, что любые две геделевские нумерации взаимно сводимы. Следующая теорема Роджерса об изоморфизме усиливает это утверждение.

Теорема 2 *Для любых двух геделевских нумераций $\{f_i\}$ и $\{g_i\}$ класса всех ч.р.ф. существует рекурсивная перестановка натурального ряда $\sigma(i)$, такая что*

$$f_{\sigma(i)} = g_i \qquad f_i = g_{\sigma^{-1}(i)}.$$

Отметим еще два важных результата, относящихся к геделевским нумерациям.

Теорема 3 *(Теорема Клини о неподвижной точке) Для любой геделевской нумерации {*
рекурсивной функции $r(i)$ можно эффективно найти такое i_0, что $f_{i_0} = f_{r(i_0)}$.

Теорема 4 *(Теорема Райса о неразрешимости нетривиальных свойств) Пусть $\mathcal{G}$ - произво*
нетривиальный класс ч.р.ф. (т.е. непустой класс, не содержащий все ч.р.ф.), $\{f_i\}$ - гедел
нумерация. Тогда множество $A = \{i | f_i \in \mathcal{G}\}$ номеров программ функций из $\mathcal{G}$ неразрешимо.

1.2 Меры сложности вычислений

Теория сложности вычислений рассматривает измерения трудности вычислений. Мы буде
сматривать меры сложности вычислений, которые определены для всевозможных вычислени
для всех частично рекурсивных функций (ч.р.ф.). Поэтому для определения некоторой меры
ности нам нужен эффективный способ задания всевозможных вычислений или алгоритмов, а
сложности будет показывать, сколько "ресурсов"требуется любому из этих алгоритмов, что
данному значению аргумента вычислить соответствующее значение функции. В качестве осно
способа перечисления алгоритмов мы будем рассматривать перечень (нумерацию программ)
ленточных машин Тьюринга. Мерой сложности для данной машины M , работающей на аргу
x, могло бы быть число шагов, которые она выполнит, прежде чем остановится. Другая мера
ности получается, когда мы рассматриваем перечень всех программ, написанных на Паск
определяем сложность i-ой программы на аргументе x числом команд, которые выполняют
того, как программа заканчивает работу. Отметим, что указанные меры сложности ассоциируі
алгоритмами, а не прямо с теми функциями, которые этими алгоритмами вычисляются. Дело
что для всякой вычислимой функции существует бесконечно много вычисляюших ее алгори
Конечно, хотелось бы связать с каждой функцией самый простой, "оптимальный"относительн
ной меры сложности алгоритм. Однако далее будет показано, что существуют вычислимые
ции, у которых нет "наилучшего"алгоритма. Из примеров выше видно, что мера сложности со
из эффективного списка алгоритмов (программ), вычисляющих все ч.р.ф., каждому из котор
поставлена сигнализирующая функция, задающая количество "ресурсов", использованных да
алгоритмом на данном аргументе. Естественные сигнализирующие функции удовлетворяют
некоторым дополнительным условиям.

Следующие аксиомы для (абстрактных) мер сложности вычислений предложены М.Блю
1967 г.

Определение 3 *Мерой сложности вычисления Φ называется геделевская нумерация ч.р.*
$f_0, f_1, f_2, ...,$ которым сопоставлены ч.р. сигнализирующие функции $\Phi_0, \Phi_1, \Phi_2, ...,$ такие чт
(1) для каждого i $f_i(n)$ определена $\Leftrightarrow \Phi_i(n)$ определена;
(2) предикат

$$M(i,n,m) = \begin{cases} 1, & \text{если } \Phi_i(n) = m, \\ 0 & \text{в противном случае} \end{cases}$$

является обшерекурсивным.

Задача1. Покажите, что число тактов работы машины Тьюринга удовлетворяет аксиомам
ма.

Задача2. Являются ли мерами сложности следуюшие функции ?

а) Число рекурсий, используемых для определения функции в схеме примитивной рекурсии.

б) Функции, $\{\Phi_i\}$, определенные равенством $\Phi_i(n) = 0$ для любых i и n.

в) Функции Φ_i, определенные равенством $\Phi_i(n) = 1$, если $f_i(n)$ определена, и $\Phi_i(n)$ не определена, если $f_i(n)$ не определена.

г)* Для машин Тьюринга функция $C_i(x)$, равная максимальному числу изменений символа (стираний) в некоторой ячейке ленты при вычислении M_i на входе x.

Задача3. Пусть Φ и Ψ - две меры сложности, определенные для одной геделевской нумерации ч.р.ф. $\{f_i\}$. Показать, что Φ^1 и Φ^2, задаваемые равенствами: $\Phi_i^1(n) = \Phi_i(n) + \Psi_i(n)$ и $\Phi_i^2(n) = \Phi_i(n) \times \Psi_i(n)$ также являются мерами сложности вычислений.

Результаты, которые можно получить, исходя из определения 2 относятся к "абстрактной"теории сложности. Они справедливы для всех мер сложности и не зависят от выбора языка описания алгоритмов и конкретного способа измерения сложности вычислений. Установим вначале существование сколь угодно сложных общерекурсивных функций (о.р.ф.). Будем говорить, что i есть номер функции f, если $f_i(n) = f(n)$ для всех n.

Теорема 5 *(Диагональ Цейтина, 1959) Пусть Φ - мера сложности вычислений и h - любая ("большая") о.р.ф. Тогда существует о.р.ф. f такая, что для любого номера i функции f справедливо $\Phi_i(n) > h(n)$ для бесконечно многих n.*

Д о к а з а т е л ь с т в о. Зафиксируем некоторую о.р.ф. $r(n)$ такую, что для каждого i существует бесконечно много n, таких что $r(n) = i$. (Покажите, что функция $r_1(n) = n^2 - \lfloor\sqrt{n}\rfloor^2$ удовлетворяет этому условию!) Определим

$$f(n) = \begin{cases} f_{r(n)}(n) + 1, & \text{если } \Phi_{r(n)} \leq h(n) \\ 0 & \text{в противном случае.} \end{cases}$$

Из рекурсивности f и h и из условия (2) определения 2 следует, что можно эффективно проверить, выполняется ли неравенство $\Phi_{r(n)} \leq h(n)$. Поэтому $f(n)$ - общерекурсивна. Если j - номер f, то для бесконечного числа n, таких что $r(n) = j$, имеем, что $\Phi_j(n) > h(n)$, что и требовалось доказать.

Более сложная диагональная конструкция позволяет установить существование о.р.ф., сложность которых превышает заданную границу почти во всех точках (за исключением, быть может, конечного их числа).

Теорема 6 *(Диагональ Рабина, 1960) Пусть Φ - мера сложности вычислений и $h-$ любая ("большая") о.р.ф. Тогда существует о.р.ф. (предикат) Γ такой, что для любого номера i функции Γ справедливо $\Phi_i(n) > h(n)$ для почти всех n.*

Д о к а з т е л ь с т в о. Предикат Γ будем определять по индукции. Одновременно будем определять список опровергнутых номеров L. Номера, попавшие в L, не смогут быть номерами Γ.
Б а з и с: пусть $\Gamma(0) = 0$ и $L = \emptyset$.
Предположим, что Γ уже определен в точках $0, 1, ..., n-1$ и определен список L_{n-1}.
Ш а г n : ищем такой наименьший еще не опровергнутый номер $j \leq n$, для которого $j \notin L_{n-1}$ и $\Phi_j(n) \leq h(n)$. Если такой номер j найдется, то положим $L_n = L_{n-1} \cup \{j\}$ и Если такого j нет, то полагаем $L_n = L_{n-1}$ и $\Gamma(n) = 0$. Пусть $L = \cup_{n=0}^{\infty} L_n$. Рекурсивность Γ следует непосредственно из его определения и аксиом для Φ. Пусть $i-$ некоторый номер Γ. Предположим, что $\Phi_i(n) < h(n)$ для бесконечного числа n. Покажем, что в этом случае i попало бы в список L и не смогло бы быть номером Γ. Действительно, существует такое m, что $[0, i-1] \cap L = [0, i-1] \cap L_m$. Найдем такое

первое число $N > m$, что $\Phi_i(N) < h(N)$ и рассмотрим шаг N. Если i не содержится в L ,
удовлетворяет условиям на j и является наименьшим из таких чисел (все меньшие опроверг
номера уже содержатся в L). Поэтому оно пападает в L и $\Gamma(N) \neq f_i(N)$, что противоречит в
i. Следовательно, для каждого номера i предиката Γ имеем $\Phi_i(n) > h(n)$ для почти всех n.

Задача 4. Показать, что в утверждении теоремы 2 квантор "для почти всех n"нельзя зам
на квантор "для всех n".

Следующая теорема устанавливает инвариантность различных мер сложности для одного
же класса алгоритмов.

Теорема 7 *Пусть для некоторой геделевской нумерации $\{f_i\}$ имеются две меры сложност
Ψ. Тогда существует о.р.ф. $H(x, y)$, такая что для любого номера i*

$$H(x, \Psi_i(x)) \geq \Phi_i(x) \quad и \quad H(x, \Phi_i(x) \geq \Psi_i(x)$$

для почти всех x.

Д о к а з а т е л ь с т в о. Положим

$$H(x, y) = max\{\Phi_i(z), \Psi_i(z) \mid i \leq x,\ z \leq x \quad \text{и } [(\Phi_i(z) = y) \text{ или } (\Psi_i(z) = y)]\}.$$

Задача 5. Доказать, что так определенная функция H(x,y) удовлетворяет требованиям те
3.

Определенная в теореме 3 функция H может быть весьма грубой оценкой соотношения м
конкретными мерами сложности. Но она позволяет получить в качестве следствия утвержд
том, что всякий класс функций, сложность которых ограничена при одной мере сложности
мер, полиномами, экспонентой, примитивно рекурсивными функциями и т.п.), имеет рекур
ограниченную сложность и при любой другой мере сложности (почему?).

Следующий результат, который мы приводим без доказательства, показывает, что сущес
функции, не имеющие наилучших вычислений.

Теорема 8 *Об ускорении.(М.Блюм, 1967) Пусть Φ - мера сложности для нумерации $\{f_i\}$ и
произвольная (сколь угодно быстро растущая) о.р.ф. Существует такой предикат Γ, чт
каждого его номера i существует такой его номер j, что $\Phi_i(n) > r(\Phi_j(n))$ для почти всех*

Например, возьмем в качестве $r(n)$ функцию 2^n , а в качестве Φ - время работы для машин Тью
t_i. Тогда предикат Γ из теоремы 4 таков, что для всякой машины Тьюринга M_i, его вычисля
существует существенно более быстрая машина M_j, вычисляющая этот предикат: для почт
n $t_j(n) \leq log_2(t_i(n))$. Разумеется, для M_j также существует более быстрая машина M_{j_1} и т

Литература

Хартманис, Хопкрофт *"Абстрактная теория сложности вычислений"*, Кибернетический сб
(новая серия), 1974, стр. 131-176.