Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ МАТЕМАТИКИ И МЕХАНИКИ ИМ. Н.И ЛОБАЧЕВСКОГО

КАФЕДРА АЛГЕБРЫ И МАТЕМАТИЧЕСКОЙ ЛОГИКИ

Специальность 01.01.01

Специализация: Алгебра и математическая логика

ДИПЛОМНАЯ РАБОТА НА ТЕМУ:

ВЕРХНИЕ И НИЖНИЕ ГРАНИЦЫ КОЛМОГОРОВСКОЙ СЛОЖНОСТИ КОНЕЧНЫХ БИНАРНЫХ СТРОК

Студент 5 курса

Группа №05-902

«__» _____ 2014 г. Михайлов И.И.

Научный руководитель:

«__» _____ 2014 г. Файзрахманов М.Х.

Казань - 2014

Содержание

Глава 1. Колмогоровская сложность 2

1.1. Основные понятия теории алгоритмов 2

1.1.1. Конструктивные объекты 2
1.1.2. Алгоритмы 3

1.2. Определение колмогоровской сложности 6

1.3. Несжимаемые последовательности 9

1.4. Сложность пары 12

1.5. Количество информации 13

Глава 2. Верхние и нижние оценки колмогоровской сложности 15

2.1. Неравенства для простой колмогоровской сложности 15

2.2. Неравенства для беспрефиксной колмогоровской сложности 17

2.3. Неравенства для условной колмогоровской сложности 20

Литература 22

Глава 1.

Колмогоровская сложность

1.1. Основные понятия теории алгоритмов

1.1.1. Конструктивные объекты

Определения колмогоровской сложности и алгоритмической случайности основываются на использовании общей теории алгоритмов, которая называется также теорией рекурсивных функций, а также ее основного результата – теоремы о существовании универсальной функции. Алгоритмы применяются к конструктивным объектам и в качестве значений также выдают конструктивные или конечные объекты. Простыми примерами конструктивных объектов являются слова в некотором конечном алфавите

$$A = \{a_1, \ldots, a_k\},$$

где $k \geq 1$. Алфавит состоит из букв $a_1, \ldots, a_k$. Слово в алфавите A – это конечная последовательность букв $x = x_1 \ldots x_n$ этого алфавита, т.е. $x_i \in A$ при $i = 1, \ldots, n$. Множество всех слов в алфавите A обозначается A^*.

Длина слова x равна числу букв в этом слове. Удобно рассматривать пустое слово λ, которое не содержит ни одной буквы.

Определение 1.1. *Конкатенацией* двух слов $x = x_1 \ldots x_n$ и $y = y_1 \ldots y_m$ называется слово $xy = x_1 \ldots x_n y_1 \ldots y_m$, длина которого равна сумме длин слов x и y.

Слово x является префиксом слова y, обозначается это как $x \subseteq y$, если $y = xz$ для некоторого слова z. Если слово z непустое, то пишем $x \subset y$.

В теории информации широко используется двоичный алфавит $I = \{0, 1\}$. Выделение такого алфавита связано со способом хранения информации в памяти компьютера. Слова в алфавите I называются двоичными (бинарными) последовательностями или строками. Множество всех двоичных слов обозначаем $\Omega = \{0, 1\}^*$. Множество всех натуральных чисел $\mathcal{N} = \{1, 2, \ldots\}$ также является множеством конструктивных объектов. Натуральные числа можно представлять в виде слов в унарном алфавите $U = \{1\}$: последовательность $11 \ldots 1$ из n единиц представляет число $n \in \mathcal{N}$.

Мы будем использовать стандартное представление натуральных чисел с помощью двоичных строк, которое определяется следующим образом. Для удобства мы включим число 0 в это соответствие.

Пусть $bin(n+1) = 1\nu_{k-1} \ldots \nu_0$ представляет собой запись числа $n+1$ в двоичной форме:

$$n + 1 = 2^k + \nu_{k-1}2^{k-1} + \ldots + \nu_1 2^1 + \nu_0 2^0$$

Сопоставляем натуральному числу n строку $str(n) = \nu_{k-1} \ldots \nu_0$. Каждая строка соответствует некоторому натуральному числу, пустая строка соответствует числу 0.

Имеет место неравенство $2^k \leq n + 1 < 2^{k+1}$ и поэтому $k = \lfloor \log(n+1) \rfloor$. Отсюда длина строки, сопоставленной числу n, равна $l(\mathrm{str}(n)) = \lfloor \log(n+1) \rfloor \leq \log n + 1$ при $n \geq 1$.

Пример сопоставления указан в таблице.

Число n	Строка $\mathrm{str}(n)$	Число $n+1$	дв. запись $\mathrm{bin}(n+1)$
0	λ	1	1
1	0	2	10
2	1	3	11
3	00	4	100
4	01	5	101
5	10	6	110
6	11	7	111
...	...	...	...

Отождествляем натуральное число n его запись в виде строки $\mathrm{str}(n)$.

Пары строк могут кодироваться строками различным образом. Мы не можем рассматривать конкатенацию двух строк $x = x_1 \dots x_n$ и $y = y_1 \dots y_m$ как код пары (x, y), так как по ней нельзя однозначным образом разделить элементы пары. Поэтому необходимо использовать дополнительную информацию для разделения пары на ее элементы. Для произвольной строки $x = x_1 ... x_n$ обозначим строку x, в которой все биты повторены по два раза $x = x_1 x_1 \dots x_n x_n$. Теперь сопоставим паре строк (x, y) последовательность

$$\bar{x}01y = x_1 x_1 \dots x_n x_n 01 y_1 \dots y_m.$$

Отсюда видно, что существует алгоритм, который восстанавливает элементы пары x и y по последовательности $\bar{x}01y$. При этом длина кода равна $l(\bar{x}01y) = 2l(x) + l(y) + 2$.

Таким образом, можно устроить и более экономное кодирование пар (x, y). Сопоставим паре (x, y) строку

$$\overline{\mathrm{str}(l(x))}01xy,$$

которая состоит из удвоенной двоичной записи $\overline{\mathrm{str}(l(x))}$ длины строки x, разделителя 01 и строк x и y. По этому коду можно однозначно восстановить x и y, при этом длина кодирующей последовательности равна

$$l(x) + l(y) + 2l(\mathrm{str}(l(x)) + 2 \leq l(x) + l(y) + 2\log l(x) + 3,$$

и так далее.

Аналогичным образом можно кодировать тройки строк (x, y, z), если записывать их в виде $(x, (y, z))$.

Вещественные числа не являются конструктивными объектами, так как их множество несчетно. Поэтому алгоритмы не будут работать непосредственно с вещественными числами. Вместо этого, они будут применяться к их рациональным приближениям.

1.1.2. Алгоритмы

В качестве основной модели алгоритма будет использоваться понятие Машины Тьюринга (**MT**). Машина Тьюринга представляет собой, бесконечную в обе стороны ленту, и считывающее устройство. Лента разделена на ячейки, в каждую из которых головка может записывать символ некоторого входного алфавита. На этом же алфавите записывается выходное слово . Машина Тьюринга задается набором $(A, \Gamma, Q, \delta, q_0, q_K)$, где

- A – основной алфавит, на котором задается входное слово **MT** и записывается результат;

- Γ – ленточный или рабочий алфавит, который используется для вычислений **MT**, при этом, $A \subseteq \Gamma$;

- Q – множество внутренних состояний или память головки; в процессе работы головка может запоминать ограниченную по объему информации с помощью своих состояний $q \in Q$; говорят также, что **MT** находится в состоянии q;

- $\delta(q, a)$ – функция переходов, где $q \in Q$ и $a \in \Gamma$; ее значения тройки $\delta(q, a) = (q', a', M)$, где $q' \in Q$ и $a' \in \Gamma$ и $M \in (R, L, S)$, которые называются командами **MT**; команды интерпретируются следующим образом: если **MT** находится в состоянии q, и читает на ленте букву a, то команда $\delta(q, a) = (q', a', M)$ дает указание стереть букву a, записать вместо нее букву a', изменить текущее состояние q головки на q' и переместить головку влево, если $M = L$, сдвинуться вправо, если $M = R$, или оставаться над той же ячейкой, если $M = S$;

- q_0 – начальное состояние головки, при этом головка устанавливается над самым левым символом входного слова;

- q_K – заключительное состояние; как только головка **MT** первый раз перейдет в состояние q_K, машина прекращает работы, при этом, слово, которое записано на ленте, считается результатом ее работы.

Текущее состояние работы **MT** описывается конфигурацией – словом

$$x_1 \dots x_s q a y_1 \dots y_k, \tag{1.1}$$

где q – текущее состояние головки, которая обозревает символ a на ленте; справа и слева от q находятся все символы находящиеся на ленте. Один шаг работы **MT** – это переход от одной конфигурации к непосредственно следующей конфигурации.

MT вычисляет некоторую функцию $\psi : A^* \to A^*$ следующим образом. Перед запуском **MT** на ленте записан аргумент x, над первой буквой которого установлена головка **MT**, находящаяся в начальном состоянии q_0. Если в процессе вычисления **MT** переходит в заключительное состояние q_K, то в качестве значения функции $\psi(x)$ берется часть содержимого ленты – от символа, отмеченного состоянием q_K, до маркера конца выходного слова. Если одно из этих правил нарушено или **MT** никогда не приходит в состояние q_K, то значение функции ψ на аргументе x не определено.

Определение 1.2. Функция $\psi : A^* \to A^*$ называется *вычислимой*, если существует **MT**, которая вычисляет значения этой функции.

Характерной особенностью процесса вычисления **MT** является то, что на некоторых входных словах **MT** может никогда не достичь конечного состояния. На таких словах соответствующая функция не определена. В таком случае, вычислимая функция является частично определенной.

Можно построить "универсальную" **MT**, которая может интерпретировать работу любой программы. Фиксируем входной алфавит A и рассматриваем все **MT**, которые вычисляют функции типа $A^* \to A^*$. В частности, можно рассмотреть $A = \{0, 1\}$ и считать, что рассматриваются все вычислимые функции, аргументы и значения которых – натуральные числа. Пусть символ # не принадлежит алфавиту A. Каждую программу для вычисления функции $\psi : A^* \to A^*$ можно закодировать словом p в алфавите A. Слово содержит закодированную последовательность команд

программы **МТ**, вычисляющей значения функции ψ. Также можно написать программу–интерпретатор, на вход которой подаются слова $p\#x$, где p – код некоторой программы, а $x \in A^*$ – входное слово для этой программы. Интерпретатор может использовать более широкий ленточный алфавит. На каждом шаге работы интерпретатора на ленте записана конфигурация моделируемой **МТ**; интерпретатор декодирует команды из слова p и выполняет необходимые переходы от одной конфигурации к другой, которые предписываются этими командами. Таким образом, интерпретатор вычисляет некоторую функцию $U(p, x)$, где $p, x \in A^*$, обладающую свойством:

- для любой вычислимой функции $\psi : A^* \to A^*$ найдется $p \in A^*$ такое, что $\psi(x) = U(p, x)$ для всех x.

Определение 1.3. Функция $U(p, x)$, обладающая этим свойством, называется *универсальной функцией* для всех вычислимых функций типа $A^* \to A^*$.

Определение 1.4. Множество конструктивных объектов называется *перечислимым*, если либо оно пусто, либо является множеством значений некоторой вычислимой функции.

Предложение 1.1. *Область определения любой вычислимой функции $f(x)$ является перечислимым множеством.*

Доказательство. Пусть, область определения функции $f(x)$ непустое множество. Построим алгоритм для вычисления функции $g(n)$, перечисляющий область определения функции $f(x)$. Определим $g(0) = a$, где a – любой элемент из области определения функции f.

Запускаем процесс одновременного вычисления всех значений $f(x)$ на всех возможных входах x, на каждом шаге процесса делаем один шаг вычисления значения $f(x)$ только для одного из таких x. Если на шаге n значение $f(x)$ впервые определилось, полагаем $g(n) = x$. В противном случае, $g(n) = g(n-1)$.□

Верно и обратное утверждение к предложению 1.1. А именно, произвольное перечислимое множество C является областью определения вычислимой функции

$$\xi(a) = \begin{cases} 1, \text{ если } a \in C, \\ \text{неопределено, в противном случае.} \end{cases}$$

Пусть

$$W_p = \{x : U(p, x) \quad \text{определена}\}.$$

Из определения и предложения 1.1 следует, что

- W_p – перечислимое множество для любого p;
- для любого перечислимого множества C найдется такое p, что $C = W_p$;
- множество $(p, x) : x \in Wp$ перечислимо.

Эти свойства означают, что имеется алгоритм, который "равномерно" перечисляет все перечислимые множества.

Определение 1.5. Множество конструктивных объектов C называется *разрешимым*, если его характеристическая функция

$$\xi(a) = \begin{cases} 1, \text{ если } a \in C, \\ 0, \text{ если } a \notin C. \end{cases}$$

является вычислимой. Область определения функции $U(p,x)$ называется универсальным множеством.

Предложение 1.2. *Универсальное множество*

$$\{(p,x) : U(p,x) \text{ определено}\} \tag{1.2}$$

перечислимо, но не разрешимо.

Доказательство. Множество (1.2) перечислимо как область определения вычислимой функции. От противного. Пусть множество (1.2) разрешимо, т.е. функция

$$\xi(p,x) = \begin{cases} 1, \text{ если } U(p,x) \text{ определено,} \\ 0, \text{ в противном случае.} \end{cases}$$

является вычислимой. Тогда функция

$$\theta(p) = \begin{cases} 1, \text{ если } \xi(p,p) = 0, \\ \text{неопределено, если } \xi(p,p) = 1. \end{cases}$$

также является вычислимой. Существует такое q, что $\theta(p) = U(q,p)$ для всех p. Пусть $p = q$. Если $U(q,q)$ определено, то $\xi(q,q) = 1$. В этом случае значение $\theta(q)$ не определено. Но $\theta(q) = U(q,q)$, значит и $U(q,q)$ не определено. Получаем противоречие. Пусть значение $U(q,q)$ не определено. Тогда $\xi(q,q) = 0$. В этом случае $\theta(q) = 0$, т.е. это значение определено. Но $\theta(q) = U(q,q)$, т.е. значение $U(q,q)$ также определено. Опять получаем противоречие.

Значит исходное предположение о том, что функция $\xi(p,x)$ вычислимая, или то же самое, что множество (1.2) – разрешимое, неверно.□

Заметим, что предложение 1.2 эквивалентно тому, что не существует алгоритма, который решает вопрос о том, остановится ли произвольная программа p на входе x или нет. Задача построения такого алгоритма называется "проблемой остановки".

1.2. Определение колмогоровской сложности

А.Н.Колмогоров предложил измерять сложность конечного объекта x при заданном конечном объекте y длиной самой короткой последовательности p (программы для x), состоящей из 0 и 1, по которой некоторой способ декодирования B может восстановить x, используя в качестве дополнительной информации слово y. Математически это записывается следующим образом:

$$K_B(x|y) = \min\{l(p) : B(p,y) = x\},$$

где $l(p)$ – длина длина последовательности $p \in \{0,1\}^*$, а $B(p,y)$ – некоторая вычислимая функция. Мы считаем, что $\min\emptyset = \infty$. Называем функцию $K_B(x|y)$ мерой сложности относительно способа декодирования $B(p,y)$.

Учитывая то, что мы можем кодировать любые конструктивные объекты двоичными строками, можно считать, что $x, y \in \{0,1\}^*$.

Имеет место основная теорема теории алгоритмической сложности – теорема инвариантности.

Теорема 1.1. *Существует такая вычислимая функция $A(p,y)$, что для любой вычислимой функции $B(p,y)$ имеет место неравенство*

$$K_A(x|y) \leqslant K_B(x|y) + c, \tag{1.3}$$

где c – некоторая константа, не зависящая от x и y.

Доказательство. Определим "универсальный" способ декодирования:

$$A(\bar{q}01p, y) = U(q, p, y)$$

для всех $p, q, y \in 0, 1^*$. Для всех остальных входов, не имеющих вида $(\bar{q}01p, y)$, значение функции A не определено.

Пусть $B(p, y)$ – произвольная вычислимая функция, представляющая некоторый способ декодирования. По определению универсальной функции, для некоторого q будет $B(p, y) = U(q, p, y)$ для всех p и y. Допустим, что p – самый короткий код для строки x при способе описания B и дополнительной информации y. Для него выполнено $B(p, y) = x$. Тогда

$$A(\bar{q}01p, y) = U(q, p, y) = B(p, y) = x$$

Сравниваем длины кратчайших кодов для x при способах декодирования A и B:

$$K_A(x|y) \leqslant K_B(x|y) + 2l(q) + 2.$$

Соответствующая константа с имеет вид $c = 2l(q) + 2$. Теорема доказана. □

Функция $A(p, y)$, называется *оптимальным способом декодирования.*

Для любых двух оптимальных способов декодирования A_1 и A_2 для всех x и y по теореме выполнено:

$$|K_{A_1}(x|y) - K_{A_2}(x|y)| \leqslant c, \tag{1.4}$$

где c – некоторая константа, зависящая от $A_1 A_2$.

Запишем (1.3) и (1.4) в в виде:

$$K_A(x|y) \leqslant K_B(x|y) + O(1),$$

$$K_{A_1}(x|y) = K_{A_2}(x|y) + O(1).$$

Фиксируем одну такую оптимальную функцию $A(p, y)$ и обозначим

$$K(x|y) = K_A(x|y).$$

Назовем функцию $K(x|y)$ *условной колмогоровской сложностью* слова x при известном y.

Определим *безусловную колмогоровскую сложность*

$$K(x) = K(x|\lambda)$$

конечного объекта x. В этом случае, оптимальный способ декодирования для восстановления x не использует никакой дополнительной информации.

Отметим некоторые простейшие свойства колмогоровской сложности. Первое из них – колмогоровская сложность строки не превосходит с точностью до константы ее длины:

$$K(x|y) \leqslant l(x) + O(1).$$

Для этого способа декодирования выполнено $K_B(x|y) = l(x)$ и поэтому по теореме

$$K(x|y) \leqslant K_B(x|y) + O(1) = l(x) + O(1).$$

В частности, сложность натурального числа n ограничена его логарифмом:

$$K(n) \leqslant \log n + O(1).$$

Предложение 1.3. *Безусловная сложность $K(x)$ слова x связана с условной сложностью $K(x|y)$ относительно другого слова y и сложностью $K(y)$ этого слова следующим образом*:

$$K(x|y) - O(1) \leqslant K(x) \leqslant K(x|y) + K(y) + 2\log K(y) + O(1). \tag{1.5}$$

Доказательство. Пусть по самому короткому коду p и условию y можно восстановить x. Кроме того, пусть q – самый короткий код для восстановления y. Можно добавить код условия y к коду p и по сложному коду $\overline{\mathrm{str}(l(q))}01qp$ восстановить x. Длина такого кода равна правой части неравенства (1.5). □

Приведем еще некоторые свойства колмогоровской сложности. Пусть $\psi(x)$ – вычислимая функция типа $\{0,1\}^* \to \{0,1\}^*$. Тогда

$$K(\psi(x)|y) \leqslant K(x|y) + O(1), \tag{1.6}$$

где константа $O(1)$ зависит от ψ.

Для доказательства этого неравенства рассмотрим способ декодирования $B(p,y) = \psi(A(p,y))$, где A(p,y) – оптимальный способ декодирования. Имеем $K_B(\psi(x)|y) = l(p)$, где $A(p,y) = x$. Отсюда получаем (1.6).

Верно также соотношение

$$K(x|y) \leqslant K(x|\psi(y)) + O(1). \tag{1.7}$$

Для доказательства рассмотрим способ декодирования $B(p,y) = A(p,\psi(y))$, где $A(p,y)$ – оптимальный способ декодирования. Имеем $K_B(x|y) = l(p)$, где $A(p,\psi(y)) = x$. Отсюда получаем (1.7).

Легко видеть, что колмогоровская сложность $K(x)$ неограничена. Кроме того, она не является вычислимой. Это свойство следует из более сильного свойства – функция $K(x)$ не только не вычислимая, но и не имеет неограниченной вычислимой нижней оценки.

Предложение 1.4. *Не существует вычислимой функции $\psi(x)$, которая принимает как угодно большие значения и такой, что $\psi(x) \leqslant K(x)$ для всех x, для которых $\psi(x)$ определена.*

Доказательство. Допустим, что функция $\psi(x)$ является всюду определенной и $\psi(x) \leqslant K(x)$ для всех x. Определим другую функцию

$$\mu(x) = \min\{y : \psi(y) > x\}. \tag{1.8}$$

По определению $\psi(\mu(x)) > x$ для всех x.

Так как $\psi(x)$ неограничена, функция $\mu(x)$ является всюду определенной и вычислимой. По свойству (1.6)

$$K(\mu(x)) \leqslant K(x) + (1).$$

С другой стороны, по определению функции ψ будет

$$x < \psi(\mu(x)) \leqslant K(\mu(x)) \leqslant K(x) + O(1) \leqslant l(x) + O(1).$$

Получаем противоречие, так как длина $l(x) = O(\log x)$.

Когда функция $\psi(x)$ не является всюду определенной, для вычисления функции $\mu(x)$ по формуле (1.8) может не существовать алгоритма. Не обязательно искать минимум в (1.8), достаточно найти y такое, что $\psi(y) > x$. Определим процесс одновременного вычисления всех значений $\psi(y)$ до тех пор пока не найдется y такое, что $\psi(y) > x$. Определим значение $\mu(x)$ равным первому такому y. Такое y найдется, так как функция $\psi(y)$ принимает бесконечно много значений. □

Определение 1.6. Функция $f : 0,1* \to 0,1^*$ называется *перечислимой сверху*, если ее надграфик, т.е. множество $\{(x, y) : y > f(x)\}$ перечислимо.

Легко видеть, что колмогоровская сложность $K(x|y)$ перечислима сверху, так как

$$\{(x, y, n) : n > K(x|y)\} = \{(n, x, y) : \exists p(A(p, y) = x \& l(p) < n)\},$$

где $A(p, y)$ – оптимальный способ декодирования.

Утверждение 1.3 связана с одной интерпретацией теоремы Геделя о неполноте. Теория характеризуется бесконечным набором утверждений, представленных в виде формул. Каждая формула представляет собой слово в некотором алфавите. Можно считать, что формулы закодированы двоичными строками.

Задано множество "истинных" формул TRUTH.

Также есть некоторый алгоритм, перечисляющий множество PROOF "доказуемых" формул.

Предполагаем также, что теория непротиворечива, т.е. все доказуемые формулы являются истинными: PROOF $\subseteq$ TRUTH.

Допустим, что все истинные в обычном математическом смысле утверждения вида: $K(x) > n$, где $x \in \{0,1\}^*$ и $n \in \mathcal{N}$, могут быть записаны на языке нашей теории и принадлежит множеству TRUTH.

Предложение 1.5. *Не более чем конечное число утверждений типа* $K(x) > n$ *могут принадлежать множеству* PROOF, *т.е. могут быть доказуемыми.*

Доказательство. От противного. Пусть существует алгоритм, который перечисляет (доказывает) бесконечную последовательности утверждений $K(x_i) > n_i, i = 1, 2, \ldots$

Пусть, все x_i различные (для этого не перечисляем утверждения с повторяющимися x_i).

Определим частичную вычислимую функцию $\psi(x_i) = n_i$ для всех i. Эта функция принимает бесконечно большие значения.

Тогда $K(x) > \psi(x)$ для всех x, для которых $\psi(x)$ определена. Получаем противоречие с утверждением 1.4.

Итак, имеется бесконечно много истинных утверждений вида $K(x) > n$. Не более чем конечное число таких утверждений может быть доказано. поэтому PROOF $\neq$ TRUTH. □

1.3. Несжимаемые последовательности

Имеется всего 2^n двоичных строк длины n. Для любого $k < n$, число всех двоичных строк x длины n, для которых $K(x) < n-k$, не превосходит числа всех двоичных кодов p, для которых $l(p) < n - k$. Число таких p равно $2^0 + 2^1 + \ldots + 2^{n-k-1} < 2^{n-k}$.

Таким образом, доля x таких, что $K(x) < n - k$, оценивается сверху:

$$\frac{|x : l(x) = n \& K(x) < n - k|}{2^n} < 2^{-k}$$

Величину

$$d(x) = n - K(x)$$

называем *дефектом случайности* последовательности x. Она обладает свойством

$$|x : l(x) = n \& d(x) \leqslant k| \geqslant 2^{n-k}$$

Заменим $K(x)$ на $K(x|l(x))$ и эти рассуждения останутся верными. Сформулируем это свойство в виде утверждения. Пусть $|D|$ обозначает число элементов конечного множества D.

Предложение 1.6. [6] *Пусть*

$$B_{n,k} = \{x : l(x) = n \& K(x|l(x)) \geqslant n - k\}$$

множество всех сжимаемых не более чем на k-битов последовательностей длины n. Тогда

$$2^n(1 - 2^{-k}) \leqslant |B_{n,k}| \leqslant 2^n.$$

Таким образом, для всех последовательностей длины n, кроме малой их доли, дефект случайности ограничен. Иными словами, большинство последовательностей несжимаемые. Покажем, что вследствие этой несжимаемости для большинства последовательностей имеет место свойство устойчивости частот единиц и нулей.

Перейдем теперь к более точным оценкам колмогоровской сложности. Допустим, что некоторое подмножество

$$D = \{x_1, x_2, \ldots x_m\}$$

множества всех строк длины n задано в виде списка $(x_1, x_2, \ldots x_m)$ своих элементов. Для простоты обозначаем эту последовательность также как само множество D. Тогда для задания элемента $x \in D$ при известном списке D достаточно задать номер этого элемента в списке. Отсюда получаем оценку

$$K(x|D) \leqslant \log|D| + O(1).$$

Кроме этого, имеет место оценка

$$(1 - 2^{-k})|D| \leqslant |\{x \in D : K(x|D) \geqslant \log|D| - k\}| \leqslant |D|. \tag{1.9}$$

Можно определить дефект случайности элемента $x \in D$ в виде

$$d(x|D) = \log|D| - K(x|D).$$

Для него верно неравенство

$$\frac{|\{x \in D : d(x|D) \geqslant k\}|}{|D|} \leqslant 2^{-k}.$$

Более точная верхняя оценка колмогоровской сложности строки длины n может быть получена, если предварительно представить множество всех двоичных строк длины

n в виде объединения попарно непересекающихся подмножеств строк с заданным числом единиц:

$$\{0,1\}^n = \cup_{k=0}^{n} C_n^k,$$

где

$$C_n^k = \left\{ x : l(x) = n \& \sum_{i=1}^{n} x_i = k \right\}$$

а затем применить оценку (1.9) и формулу

$$K(x) \leqslant K(x|D) + K(D) + 2\log K(D) + O(1).$$

Если $k = 0$ или $k = n$, то оценка сложности тривиальна: $K(x) = O(1)$. Пусть далее $0 < k < n$.

Напомним, что $|C_n^k| = \left\{ {n \atop k} \right\} = \frac{n!}{k!(n-k)!}$. Применяя эти оценки, получим для строки $x \in C_n^k$

$$K(x) \leqslant K(x|C_n^k) + K(C_n^k) + 2\log K(C_n^k) + O(1). \tag{1.10}$$

Далее, применяя формулу Стирлинга

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n (1 + O(1/n)),$$

для каждого факториала из биномиального коэффициента, получим

$$K(x|C_n^k) \leqslant \log \frac{n!}{k!(n-k)!} + O(1) =$$

$$= n \left(-\frac{k}{n} \log \frac{k}{n} - \left(1 - \frac{k}{n}\right) \log \left(1 - \frac{k}{n}\right) \right) +$$

$$+ \frac{1}{2}\log n - \frac{1}{2}\log(n-k) - \frac{1}{2}\log k + O(1). \tag{1.11}$$

Для задания строки, представляющей множество C_n^k, достаточно знать числа n и k. Поэтому

$$K(C_n^k) \leqslant \log n + \log k + 2\log\log k + O(1) \leqslant 3\log n + O(1). \tag{1.12}$$

Напомним определение энтропии Шеннона

$$H(p) = -p\log p - (1-p)\log(1-p),$$

где $0 \leqslant p \leqslant 1$ (полагаем $0\log 0 = 0$).

Учитывая это представление и оценки (1.11) и (1.12) перепишем оценку (1.10) в упрощенном виде

$$K(x) \leqslant nH\left(\frac{k}{n}\right) + O(\log n). \tag{1.13}$$

Верхняя оценка (1.13) позволяет вывести некоторые статистические закономерности для несжимаемых последовательностей.

Допустим, что длина x равна n. Рассмотрим разложение энтропии $H(p)$ по формуле Тэйлора в окрестности точки $p = \frac{1}{2}$. Легко проверить, что $H(1/2) = 1, H'(\frac{1}{2}) = 0$ и $H''(1/2) < 0$, где

$$H''(p) = -\frac{1}{p(1-p)\ln 2}.$$

Имеем

$$H(p) = 1 - 4\log e \left(p - \frac{1}{2}\right)^2 + o\left(\left(p - \frac{1}{2}\right)^2\right). \tag{1.14}$$

Из неравенств (1.13), (1.14) и равенства $K(x) = n - d(x)$ получим

$$\left(\frac{k}{n} - \frac{1}{2}\right)^2 = O\left(\frac{\log n + d(x)}{n}\right)$$

Таким образом, мы доказали следующую теорему – закон больших чисел для несжимаемых последовательностей.

Теорема 1.2. [6] *Существует константа C такая, что для любой конечной двоичной последовательности x длины n, содержащей k единиц, выполнено*

$$\left|\frac{k}{n} - \frac{1}{2}\right| \leqslant C\sqrt{\frac{\log n + d(x)}{n}}.$$

Теорема 1.2 показывает, что для несжимаемых последовательностей частота единиц близка к $\frac{1}{2}$. Оценка отклонения частоты от $\frac{1}{2}$ зависит от степени несжимаемости последовательности x.

Вместо меры, граничные условия на случайность задаются в виде разбиения множества всех конечных последовательностей длины n на конечные попарно непересекающиеся подмножества. Конечная последовательность x является случайной, если $K(x|D) \approx \log|D|$, где D – тот элемент разбиения, которому принадлежит $x, x \in D, |D|$ – число его элементов. Из теории кодирования следует, что $K(x|D) \leqslant \log|D| + c$ для любого $x \in D$, где c – константа. Для характеризации "степени случайности" конечной последовательности Колмогоров вводит дефект случайности конечной последовательности x относительно конечного множества D:

$$d(x|D) = \log|D| - K(x|D),$$

где $K(x|D)$ – условная сложность конечной последовательности x относительно конечного объекта D.

Идея Колмогорова заключалась в том, чтобы найти подходящие разбиения множества всех конечных последовательностей и вывести стохастические свойства конечной последовательности из предположения о том, что ее сложность относительно соответствующего элемента разбиения D (для которого $x \in D$) близка к своему максимальному значению: $K(x|D) \approx -\log|D|$.

1.4. Сложность пары

В этом разделе мы докажем теорему Колмогорова–Левина о декомпозиции сложности пары.

Теорема 1.3.

$$K(x, y) = K(x) + K(y|x) + O(\log K(x, y)). \tag{1.15}$$

Доказательство. Пусть по самому короткому коду p и условию y можно восстановить x.Пусть q – самый короткий код для восстановления y. Можно добавить код условия y к коду p и по сложному коду $\overline{\mathrm{str}(l(q))}01qp$ восстановить пару (x, y). Длина такого кода равна правой части неравенства (1.15).

Фиксируем пару (x, y). Обозначим $a = K(x, y)$. Множество

$$A = \{(x', y') : K(x', y') \leqslant a\}$$

является перечислимым. Кроме этого, $|A| < 2^{a+1}$. Для каждой строки x' рассмотрим сечение

$$A_{x'} = \{y' : (x', y') \in A\}.$$

Обозначим $m = \lfloor \log|A_x| \rfloor$. Тогда $2^m \leqslant |A_x| < 2^{m+1}$.

1) Оценим сверху величину $K(y|x)$. Для этого по числу a перечисляем все пары $(x', y') \in A$ и откладываем те пары, для которых $x' = x$. Таким образом, мы перечисляем множество A_x. Для задания $y \in A_x$ при известном x можно использовать число a и порядковый номер перечисления $y \in A_x$, т.е. число $\leqslant |A_x| < 2^{m+1}$.

Таким образом, учитывая неравенство $m \leqslant a$, получаем

$$K(y|x) \leqslant \log|A_x| + 2l(\mathrm{str}(a)) + O(1) \leqslant$$

$$\leqslant m + 2\log K(x, y) + O(1) \leqslant m + 3\log K(x, y) + O(1). \tag{1.16}$$

2) Оценим сверху величину $K(x)$. Пусть

$$B = \{x' : |A_{x'}| \geqslant 2^m\}.$$

Из неравенства

$$2^m|B| \leqslant \sum_{x' \in B} |A_{x'}| \leqslant |A| < 2^{a+1}.$$

следует $|B| < 2^{a-m+1}$.

Элементы множества B можно перечислять зная a и m.

По определению $x \in B$. Для задания x, по a перечисляем пары из множества A. При этом, как только наберется не менее 2^m пар (x', y') с одинаковой первой координатой x', перечисляем x' в B.

Так как $|A_x| \geqslant 2^m$, среди всех x', перечисленных в B будет и x. Для задания x надо кроме выше перечисленной информации знать порядковый номер его перечисления, т.е. число $\leqslant |B| < 2^{a-m+1}$. Двоичный код этого числа имеет длину не более $a - m + 1$. Мы использовали также $a = K(x, y)$ и $m \leqslant a + 1$. Их двоичные коды имеют длину не более $3\log K(x, y)$. Отсюда

$$K(x) \leqslant a - m + O(\log K(x, y)). \tag{1.17}$$

Сложим неравенства (1.16) и (1.17) и получим необходимое неравенство

$$K(y|x) + K(x) \leqslant K(x, y) + O(\log K(x, y)). \ \square$$

1.5. Количество информации

На основе понятия сложности А.Н.Колмогоров предложил определение количества информации в слове y о слове x :

$$I(y : x) = K(x) - K(x|y).$$

Величину $K(x)$ можно интерпретировать как минимальное количество информации, необходимое для воспроизведения слова x; $K(x|y)$ интерпретируется как минимальное количество информации, которое необходимо добавить к информации, содержащейся в y, чтобы восстановить x. Разность между ними естественно интерпретировать как количество информации, содержащейся в слове y о слове x.

Для любого m найдется x длины m такое, что $K(x|m) \geqslant m$. Действительно, если бы такого x не существовало, для любого y длины m существует p длины $\leqslant m-1$ такое, что $A(p,m) = y$. Здесь A – оптимальный метод декодирования. Все такие p различные. Число таких p, а значит и таких число y, не превосходит числа всех двоичных строк длины $< m$. Это число равно $2^m - 1$. Так как число всех двоичных строк длины m равно 2^m, найдется x такое, что $K(x|m) \geqslant m$.

Аналогичным образом, найдутся сколь угодно большие m, для которых $K(m) \geqslant l(m)$.

Очевидно $K(l(m)|m) = O(1)$. Для каждого такого m и для соответствующего x длины m получаем

$$I(x:m) = K(m) - K(m|x) \geqslant l(m) - O(1)$$

$$I(m:x) = K(x) - K(x|m) \leqslant l(x) - m + O(1) = O(1).$$

Значит $I(x:m) - I(m:x) \geqslant l(m) - O(1)$. Величина $l(m) \geqslant \log m - 1$.

Дальнейшее расхождение между величинами $I(x:y)$ и $I(y:x)$ в общем случае нельзя увеличить. Величина $I(x:y)$ коммутативна с точностью до $O(\log K(x,y))$. По теореме 1.3

$$K(x,y) = K(x) + K(y|x) + O(\log K(x,y)),$$

$$K(x,y) = K(y) + K(x|y) + O(\log K(x,y)).$$

Отсюда получаем

$$|I(y:x) - I(x:y)| = O(\log K(x,y)),$$

$$|I(y:x) - (K(x) + K(y) - K(x,y))| = O(\log K(x,y)).$$

Глава 2.

Верхние и нижние оценки колмогоровской сложности

2.1. Неравенства для простой колмогоровской сложности

Пусть $\{\varphi_e\}_{e\in\mathbb{N}}$ - эффективная нумерация всех частично рекурсивных функций, причем φ_1 - тождественная функция. Определим оптимальную машину Тьюринга $\mathbb{V}$, полагая

$$\mathbb{V}(0^{e-1}1\rho) = \varphi_e(\rho),$$

Для обозначения колмогоровской сложности относительно машины $\mathbb{V}$ будем использовать C без нижнего индекса. Отметим, что при таком определении $\mathbb{V}$, справедливо неравенство $C(x) \leqslant |x| + 1$.

Согласно следующей теореме, кратчайшие описания не могут быть сжаты более чем на константу.

Теорема 2.1. *Существует такая константа b, что для каждой пары строк x, σ, удовлетворяющей условиям $\mathbb{V}(\sigma) = x$ и $C(x) = |\sigma|$, справедливо неравенство*

$$C(\sigma) > |\sigma| - b$$

.

Доказательство. Рассмотрим машину M, $M(\tau) \simeq \mathbb{V}(\mathbb{V}(\tau))$. Каждое $\mathbb{V}$ - описание σ является M - описанием, следовательно $C_M(x) \leqslant C(\sigma)$. Если $M = \Phi_b, b > 0$, то $C_(x) \leqslant C_M(x) + b$, следовательно

$$|\sigma| = C(x) \leqslant C(\sigma) + b. \ \square$$

Теорема 2.2. *Если $x(i) = 0$ для каждого четного $i < |x|$, то $C(x) \leqslant^+ |x|/2$.*

Доказательство. Определим машину M, на аргументах вида $a\sigma$, $a \in \{0,1\}$, $\sigma \in \{0,1\}^*$. Если $\mathbb{V}(\sigma) = y$ и $|y| = n > 0$, то полагаем

$$M(a\sigma) = \begin{cases} 0y_0 0y_1 \dots 0y_{n-1}, \text{ если } a = 0, \\ 0y_0 0y_1 \dots 0y_{n-1}0, \text{ если } a = 1. \end{cases}$$

Имеем, $C(x) \leqslant^+ C_M(x) \leqslant^+ |x|/2$. $\square$

Помимо основного определения, существуют характеризации колмогоровской сложности $C(x)$, не зависящие от оптимальной машины Тьюринга.

Определение 2.1. Функция $D : \{0,1\} \to \mathbb{N} \cup \{\infty\}$ называется *вычислимо аппроксимируемой сверху*, если существует не возрастающая по первому аргументу вычислимая функция

$$G : \mathbb{N} \times \{0,1\}^* \to \mathbb{N} \cup \{\infty\},$$

поточечный предел которой по первому аргументу есть $D(x)$:

$$\lim_S G(s,x) = D(x).$$

Так для каждой машины M колмогоровская сложность C_M является вычислимо аппроксимируемой сверху функцией. Функцию $G(s,x)$ можно положить равной

$$C_{M_s}(x) = \min\{|\sigma| : M_s(\sigma) = x\},$$

где через M_s обозначена машина Тьюринга, ограниченная s шагами работы машины M. Обозначим через $\mathfrak{F}$ класс всех функций $D : \{0,1\} \to \mathbb{N} \cup \{\infty\}$, удовлетворяющих для каждого условию

$$|\{x : D(x) < k\}| < 2^k.$$

Для фиксированного $d \in \mathbb{N}$ строку x назовем *d_C-сжимаемой*, если

$$C(x) \leqslant |x| - d.$$

Множество всех d_c-сжимаемых строк будем обозначать через Cpr_d. [4]

Теорема 2.3. 1) *Для каждой машины Тьюринга M функция C_M принадлежит классу $\mathfrak{F}$.*

2) *Для каждой машины Тьюринга M и каждого $n \in \mathbb{N}$ существует такая строка x длины n, что $C_M(x) > |x|$.*

3) *Пусть $W \subseteq \mathbb{N} \times \{0,1\}^*$ – вычислимо перечислимое множество, удовлетворяющее для каждого n условию*

$$|\{x : (n,x) \in W\}| \leqslant 2^n.$$

Тогда существует такая машина Тьюринга M, что

$$(n,x) \in W \Leftrightarrow \exists\sigma\,(|\sigma| = n \;\&\; M(\sigma) = x)$$

для всех $n \in \mathbb{N}$, $x \in \{0,1\}^*$.

4) *Пусть $D : \{0,1\} \to \mathbb{N} \cup \{\infty\}$ – вычислимо аппроксимируемая сверху функция, принадлежащая классу $\mathfrak{F}$. Тогда существует такая машина Тьюринга M, что*

$$C_M(x) = D(x) + 1, (\infty = \infty + 1).$$ [4]

Теорема 2.4. [4] *Колмогоровская сложность C является наименьшей относительно порядка $\leqslant^+$ среди всех вычислимо аппроксимируемых сверху функций, принадлежащих классу $\mathfrak{F}$.*

Теорема 2.5. *Эта теорема показывает, что оценка $C(x) \leqslant |x| + 1$ не может быть улучшена.*

(i) *Пусть дана функция $E \in \mathfrak{F}$. Если $n \in \mathbb{N}$ удовлетворяет условию*

$$\forall x\,(|x| \leqslant n \Rightarrow E(x) \leqslant n),$$

то

$$\forall x\,(|x| \leqslant n \Rightarrow E(x) = n).$$

(ii) *Пусть дана машина Тьюринга M.* Если $C_M(x) < |x|$ *для некоторой строки x длины n_0, то для всех $n \geqslant n_0$ существует такая строка y длины n, что $C_M(y) > |y|$.*

Доказательство. (i) Если $E(y) < |y|$ для некоторого y длины $k \leqslant n$, тогда $\#\{x : E(x) < k\} \geqslant 2^k$. Это неравенство противоречит определению класса $\mathfrak{F}$.

(ii) Применим утверждение класса (i) для $E = C_M$. Для любого $n \geqslant n_0$,

$$\forall x[|x| \leqslant n \to E(x) \geqslant |x|]$$

неверно. Следовательно существует x длины n, такой, что $C_M(x) > |x|$. □

2.2. Неравенства для беспрефиксной колмогоровской сложности

Множество $S \subseteq \{0,1\}^*$ называется *беспрефиксным*, если для любых σ, $\rho \in S$ строка σ не является началом строки ρ. Машина Тьюринга называется *беспрефиксной*, если ее область определения есть беспрефиксное множество. Будем обозначать через Ω_M значение равномерно распределенной вероятностной меры открытого множества, порожденного областью определения беспрефиксной машины M, а именно

$$\Omega_M = \sum_{\sigma, M(\sigma)\downarrow} 2^{-|\sigma|}.$$

Для обозначения колмогоровской сложности относительно беспрефиксных машин будем использовать K_M вместо C_M.

Определение 2.2. Беспрефиксная машина R называется *оптимальной*, если для каждой беспрефиксной машины M выполняется неравенство

$$K_R \leqslant^+ K_M.$$

Предложение 2.1. *Существуют оптимальные беспрефиксные машины.*

Доказательство. Пусть $\{M_d\}_{d\in\mathbb{N}}$ - эффективная нумерация беспрефиксных машин.

Определим беспрефиксную машину R, полагая

$$R(0^{d-1}1\sigma) = M_d(\sigma)$$

Отметим, что $K_R(x) \leqslant K_{M_d}(x) + d$.

Таким образом, R оптимальна. □

Теорема 2.6. 1) *Для каждой оптимальной беспрефиксной машины R справедливо неравенство*

$$K_R(x) \leqslant^+ 2|x|.$$

2) *Для каждой оптимальной беспрефиксной машины R справедливы неравенства*

$$K_R(x) \leqslant^+ K_R(|x|) + |x| \leqslant^+ 2\log|x| + |x|.$$

Доказательство. 1) Определим беспрефиксную машину M:

$$M(0^{|x|}1x) = x$$

Имеем, $K_R(x) \leqslant^+ K_M(x) \leqslant 2x + 1 \leqslant^+ 2x$

2) Второе неравенство следует из соотношения $K_R(x) \leqslant^+ 2|x|$, примененного к $|x|$. Докажем первое неравенство. С этой целью определим беспрефиксную машину N следующим образом:

На входе τ ищем такое разложение $\tau = \sigma x$, что $n = R(\sigma) \downarrow$ и $|x| = n$. Если такое разложение найдено, полагаем $N(\tau) = x$.

Очевидно, что N беспрефиксная. Для данной строки x пусть σ кратчайшее R-описание $|x|$. Тогда $N(\sigma x) = x$. Следовательно $K_R(x) \leqslant^+ K_R(|x|) + |x|$. $\square$

Теорема 2.7. *Существует такая оптимальная беспрефиксная машина $\mathbb{U}$, что*

$$K_{\mathbb{U}}(x) \leqslant K_{\mathbb{U}}(|x|) + |x| + 1.$$

Далее вместо записи $K_{\mathbb{U}}(x)$ будем использовать просто $K(x)$.

Доказательство. Определим последовательность машин Тьюринга $\{N_e\}_{e\in\mathbb{N}}$, полагая

$$N_e(\sigma x) = x \Leftrightarrow M_e(\sigma) = |x|.$$

Рассмотрим вычислимую функцию q, определенную следующим образом:

$$\varphi_{q(e)}(0^{d-1}1x) = \begin{cases} N_e(\rho), \text{ если } d = 1, \\ M_d(\rho), \text{ если } d > 1. \end{cases}$$

Выберем неподвижную точку m функции q. Очевидно, что в качестве $\mathbb{U}$ можно взять машину Тьюринга, вычисляющую φ_m. $\square$

Теорема 2.8. *Справедливо неравенство*

$$K(x) \leqslant^+ 2\log\log|x| + \log|x| + |x|.$$

Доказательство. Пусть $n = |x|$. Тогда $K(n) \leqslant^+ |n| + 2\log|n|$. Следовательно

$$K(x) \leqslant K(n) + |x| + 1 \leqslant^+ 2\log\log n + \log n + |x|. \ \square$$

Пусть W_i^s обозначает конечное подмножество пар, перечисленных в W_i за s шагов вычисления универсальной функции. Определим

$$P_i^s(x) = \max(\{r : (r, x) \in W_i^s\} \cup 0).$$

Функция $P_i^s(x) > 0$ для не более чем конечного числа x. Определим

$$P_i(x) = \sup_S\{P_i^s(x) : \sum_y P_i^s(y) \leqslant 1\}.$$

Легко видеть, что P_i является перечислимой снизу полумерой для каждого i и, более того, множество $\{(i, r, x) : r < P_i(x)\}$ перечислимо.

Для любой перечислимой снизу полумеры Q имеем

$$W_i = (r, x) : r < Q(x)$$

для некоторого i. Легко видеть, что $Q(x) = P_i(x)$ для всех x.

Определим

$$P(x) = \sum_{i=1}^{\infty} \frac{1}{i(i+1)} P_i(x)$$

Теорема 2.9. [6] $K(x) = -\log P(x) + O(1)$.

Теорема 2.10. $K(x,y) = K(x) + K(y|x, K(x)) + O(1)$.

Предварительно рассмотрим две простые леммы.

Лемма 2.1. $K(x,y) \leqslant K(x) + K(y|x) + O(1)$.

Лемма 2.2. $K(x, K(x)) = K(x) + O(1)$.

Доказательство. Неравенство $K(x) \leqslant K(x, K(x)) + O(1)$ очевидно.

Пусть p – самый короткий код для x. Тогда некоторый алгоритм может по p вычислить x и $K(x) = l(p)$. Отсюда

$$K(x, K(x)) \leqslant K(x) + O(1).$$

Лемма доказана. □

Пользуясь леммами 2.1 и 2.2 получим неравенство $\leqslant$:

$$K(x,y) \leqslant K(y,x,K(x)) + O(1) \leqslant$$

$$\leqslant K(x, K(x)) + K(y|x, K(x)) + O(1) =$$

$$= K(x) + K(y|x, K(x)) + O(1).$$

Учитывая теорему 2.9 нам необходимо доказать неравенство

$$cP(y|x, K(x)) \geqslant \frac{P(x,y)}{P(x)} = 2^{K(x)} P(x,y),$$

где c – положительная константа. Функция $Q(x) = \sum_y P(x,y)$ является перечислимой снизу полумерой, так как

$$\sum_x Q(x) = \sum_{x,y} P(x,y) \leqslant 1.$$

Поэтому

$$c_1 2^{K(x)} \geqslant P(x) \geqslant \sum_y P(x,y),$$

для некоторой константы c_1, или

$$\sum_y c_1^{-1} 2^{K(x)} P(x,y) \leqslant 1. \tag{2.1}$$

Если бы функция, стоящая под знаком суммы (2.1), была перечислимой снизу полумерой, мы сразу получили бы необходимое неравенство

$$cP(y|x, K(x)) \geqslant 2^{K(x)} P(x,y)$$

для некоторой константы c.

Однако функция, стоящая под знаком суммы (2.1), не является перечислимой снизу. Мы преодолеем этот недостаток с помощью дополнительных построений.

Множество $W = \{(r,x,z) : r < P(x,z)\}$ перечислимо снизу. Пусть W^t обозначает конечное подмножество его элементов, перечисленных за t шагов. Определим

$$P^t(x,z) = \max(\{r : (r,x,y) \in W^t\} \cup 0).$$

Величина $K(x)$ не вычислима по x, но эта трудность преодолевается путем перехода к общему случаю. Определим полумеру $Q(y|x,m)$ следующим образом:

по x, m и s найдем максимальное $t \leqslant s$ такое, что

$$c_1^{-1} 2^m \sum_{z \leqslant t} P^t(x,z) \leqslant 1,$$

и определим

$$Q^s(y|x,m) = \begin{cases} c_1^{-1} 2^m P^t(x,y), \text{ если } y \leqslant t, \\ 0, \text{ если } y > t. \end{cases}$$

По определению $Q^s(y|x,m) \leqslant Q^{s+1}(y|x,m)$ для всех x, m и s.

Определим

$$Q(y|x,m) = \sup_S Q^s(y|x,m).$$

Тогда нетрудно видеть, что функция Q перечислима снизу и

$$\sum_y Q(y|x,m) \leqslant 1$$

для любых x и m. Имеем для априорной полумеры

$$cP(y|x,m) \geqslant Q(y|x,m)$$

для некоторой константы c.

Полагаем в этом неравенстве $m = K(x)$. Так как для каждого t

$$c_1^{-1} 2^{K(x)} \sum_{z \leqslant t} P^t(x,z) \leqslant$$

$$c_1^{-1} 2^{K(x)} \sum_{z \leqslant t} P(x,z) \leqslant 1,$$

имеем

$$Q(y|x, K(x)) = c_1^{-1} 2^{K(x)} P(x,z)$$

откуда получаем

$$P(y|x, K(x)) O(1) \geqslant 2^{K(x)} P(x,y).$$

Теорема доказана. □

2.3. Неравенства для условной колмогоровской сложности

Пусть $\{\Phi_d^2\}_{d \in \mathbb{N}}$ и $\{M_d^2\}_{d \in \mathbb{N}}$ – геделевские нумерации всех машин Тьюринга и беспрефексных машин соответственно. Определим оптимальные машины $\mathbb{V}^2$ и $\mathbb{U}^2$, полагая

$$\mathbb{V}^2(0^{e-1}1\sigma, y) = \Phi_e^2(y), \ \mathbb{U}^2(0^{d-1}1\sigma, y) = M_d^2(y).$$

Условная колмогоровская сложность $C(x|y)$ определяется следующим образом:

$$C(x|y) = \min\{|\sigma| : \mathbb{V}^2(\sigma, y) = x\}.$$

Аналогично определяется *беспрефиксная* сложность

$$K(x|y) = \min\{|\sigma| : \mathbb{U}^2(\sigma, y) = x\}.$$

Для каждой строки x обозначим через x^* самую левую строку σ, такую что $|\sigma| = K(x)$ и $\mathbb{U}_t(\sigma) = x$, где t – наименьший шаг, удовлетворяющий условию $K_{\mathbb{U}_t}(x) = K(x)$.

Теорема 2.11. [4] *Пусть дана машина Тьюринга M. Тогда*:

(i) $\forall z \forall y\, N(y) \downarrow \Rightarrow C(N(y)|z) \leqslant^+ C(y|z)$;

(ii) $\forall y \forall z\, N(z) \downarrow \Rightarrow C(y|z) \leqslant^+ C(y|N(z))$.

Теорема 2.12. *Для всех строк x, y выполнено соотношение $K(y|x^*) =^+ K(y, \langle x, K(x)\rangle)$.*

Доказательство. Зная x^*, машина N может вычислить пару $\langle x, K(x)\rangle = \langle U(x^*), |x^*|\rangle$. Наоборот, от пары $\langle x, K(x)\rangle$ машина N' может сначала вычислить $t = \mu s.\ K_s(x) = K(x)$ а затем x^*. Теперь применяя теорему 2.11. для K получаем нужное соотношение. □

Теорема 2.13. *Для всех строк x, y, z справедливо неравенство*

$$K(x|z) \leqslant^+ K(x|y) + K(y|z)$$

.

Доказательство. Определим двуместную машину M. Для строк τ, z перечисляем все пары строк τ, ρ, удовлетворяющие условиям $\tau = \sigma\rho$, $\mathbb{U}^2(\sigma, \mathbb{U}^2(\rho, z)) = x$. Если такая пара найдена, то полагаем

$$M(\tau, z) = \mathbb{U}^2(\sigma, \mathbb{U}^2(\rho, z)) = x$$

. Покажем, что при фиксированном z одноместная машина $M(\tau, z)$ является беспрефиксной. Пусть

$$M(\tau_0, z) = \mathbb{U}^2(\sigma_0, \mathbb{U}^2(\rho_0, z)) = x$$
$$M(\tau_1, z) = \mathbb{U}^2(\sigma_1, \mathbb{U}^2(\rho_1, z)) = x$$

Тогда ρ_0 и ρ_1 несравнимы и σ_0, σ_1, несравнимы. Следовательно, τ_0 и τ_1 так же несравнимы. Кроме того, справедливо неравенство

$$K(x|z) \leqslant^+ K_M(x|z) \leqslant K(x|y) + K(y|z). \square$$

Теорема 2.14. *Безусловная сложность $K(x)$ слова x связана с условной сложностью $K(x|y)$ относительно другого слова y и сложностью $K(y)$ этого слова следующим образом*:

$$K(x|y) - O(1) \leqslant K(x) \leqslant K(x|y) + K(y) + 2\log K(y) + O(1). \tag{2.2}$$

Доказательство. Пусть по самому короткому коду p и условию y можно восстановить x. Кроме того, пусть q – самый короткий код для восстановления y. Можно добавить код условия y к коду p и по сложному коду $\overline{\mathrm{str}(l(q))}01qp$ восстановить x. Длина такого кода равна правой части неравенства (2.2). □

Список литературы

[1] Колмогоров А. Н. Три подхода к определению понятия “количество информации” // Проблемы передачи информации. – 1965. – Т. 1., N. 1. – С. 3–11.

[2] Chaitin G. Information-theoretical limitations of formal systems // Journal of the ACM. – 1974. – V. 21. – P. 403–424.

[3] Chaitin G. A theory of program size formally identical to information theory // J. Assoc. Comput. Mach. – 1975. – V. 22. – P. 329–340.

[4] Nies A. Computability and Randomness . – Oxford University Press. – 2009. – P. 456.

[5] Downey R., Hirschfeldt D. Algorithmic Randomness and Complexity. – Springer-Verlag New York, Inc. Secaucus. – 2010. – P. 855.

[6] Вьюгин В. В. Колмогоровская сложность и алгоритмическая случайность: учеб. пособие. – М.: МФТИ, ИППИ РАН. – 2012. – с. 140.

[7] Левин Л.А. О понятии случайной последовательности // Доклады АН СССР. – 1973. – Т. 212., N. 3. – С. 548–550.

[8] Левин Л.А. Законы сохранения (невозрастания) информации и вопросы обоснования теории информации // Проблемы передачи информации. – 1974. – Т. 10., N. 3. – С. 30–35.

[9] Левин Л.А. О различных мерах сложности конечных объектов (аксиоматическое описание) // Доклады АН СССР. – 1976. – Т. 227., N. 4. – С. 804–807.

[10] Левин Л.А. Об одном конкретном способе задания сложностных мер // Доклады АН СССР. – 1977. – Т. 234., N. 3. – С. 536–539.

[11] Schnorr C.P. Process complexity and effective random tests // Journal of Computer and System Sciences. – 1973. – V. 3., N. 4. – P. 376–378.

[12] Роджерс Х. Теория рекурсивных функций и эффективная вычислимость. – М.: Мир. – 1972. – с. 624.

[13] Соар Р. И. Вычислимо перечислимые множества и степени. – Казань, Казанск. матем. о-во. – 2000. – с. 576.