ТЕОРИЯ АЛГОРИТМОВ: ЕЕ ОСНОВНЫЕ ОТКРЫТИЯ И ПРИЛОЖЕНИЯ

В.А.Успенский, А.Л.Семёнов

Москва, СССР

Поразительно, как много дает теория алгоритмов. С ее помощью проясняются такие фундаментальные понятия, как доказуемость, сложность, случайность. Вообще в теории алгоритмов (как, возможно, и во многих других случаях) открытия состоят не столько в получении новых результатов, сколько в обнаружении новых понятий и уточнении старых. Мальцев пишет в этой связи: "Система понятий и язык не являются чем-то внешним для математических теорий, а составляют одну из определяющих частей их" ([Маль 66, с.72]).

Развитие теории алгоритмов сталкивается с трудностью, вызванной тем, что алгоритмы сами по себе суть объекты весьма специального типа и обладают свойством, нетипичным для математических объектов, а именно, семантическим свойством "иметь смысл". В этом отношении теория алгоритмов подобна математической логике, чьи термы и формулы также имеют смысл. Смысл терма или формулы указателен: терм указывает на, т.е. обозначает, вещь, а формула - факт. Смысл алгоритма повелителен: алгоритм должен быть исполнен. Таким образом, теория, изучающая алгоритмы, может трактоваться как своего рода лингвистика повелительных предложений. Математики еще не привыкли оперировать надлежащим образом с лингвистическими объектами, несущими на себе смысл. Поэтому при создании адекватной теории алгоритмов направляющую роль должна играть семантика, чисто математический подход для этой цели недостаточен (если считать, что чисто математический подход не должен использовать - в качестве технического понятия - понятие смысла). В теорию

Эта статья представляет собой расширенный текст доклада "Что дает теория алгоритмов?", сделанного на симпозиуме. Идея подобной вводной лекции, содержащей обзор основных концепций, связанных с общим понятием алгоритма, принадлежит организатору симпозиума А.П.Ершову.

алгоритмов входит, на равных правах с понятием алгоритма, еще и понятие исчисления. Подобно термам, формулам и алгоритмам, исчисления также являются носителями смысла; однако смысл их не указателен и не повелителен, а разрешителен.

Поэтому самоё теорию алгоритмов - в том виде, как дисциплина с таким названием сложилась к настоящему времени - было бы правильнее именовать теорией алгоритмов и исчислений (сочетание терминов "алгоритмы и исчисления", хотя и понимаемых в ином смысле, впервые появилось в сочинении Шрёдера "Об алгоритмах и исчислениях", см. [Шрё 1880]).

Теория алгоритмов (в широком понимании, включающем и теорию исчислений) может быть разделена на две части. Первая часть есть общая теория, имеющая дело со строением алгоритмов и исчислений самих по себе. Вторая часть представляет собой прикладную теорию, которая имеет дело с проблемами, связанными с понятиями алгоритма и исчисления и возникающими в различных областях математики. В соответствии с этим статья состоит из двух частей: "Основные открытия общей теории алгоритмов" и "Основные математические приложения теории алгоритмов".

Хотя авторы и старались проследить историю возникновения того или иного понятия, термина или результата, им это не всегда удавалось. Поэтому библиографические ссылки могут и не иметь приоритетного характера, а указывать лишь на использованные источники информации.

При ссылках на различные книги и статьи в квадратные скобки заключаются три-четыре начальные буквы фамилии автора (в необходимых случаях также и инициал) и год публикации (для XX века приводятся лишь две последние цифры года); при ссылке с более точным адресом - на определенную страницу или теорему - соответствующая страница или теорема указывается внутри тех же квадратных скобок.

Определяемые термины (а также термины, обозначающие первичные, неопределяемые понятия) подчеркнуты (окружающий текст, следовательно, можно воспринимать как определение); подчеркнуты также имеющие терминологический характер "собственные имена" теорем.

ОБОЗНАЧЕНИЯ И ТЕРМИНОЛОГИЯ

$\mathbb{N}$ – натуральный ряд, т.е. множество всех натуральных чисел $\{0,1,2,\ldots\}$;

$\mathbb{N}^+$ – множество всех положительных натуральных чисел $\{1,2,3,\ldots\}$;

$\mathbb{Q}$ – множество всех рациональных чисел;

$\mathbb{Q}^+$ – множество всех положительных рациональных чисел;

B^* – множество всех слов в алфавите B;

Ξ – множество всех двоичных слов, т.е. $\Xi = \{0,1\}^*$;

Ω – множество всех бесконечных двоичных последовательностей;

$A \simeq B$ означает, что при любых значениях переменных выражения A и B одновременно определены или не определены и, если определены, то имеют равные значения (пример: $x-x \simeq y-y$ истинно, а $\frac{x}{x} \simeq \frac{y}{y}$ ложно);

$A \overset{c}{\leqslant} B$ означает, что существует такое натуральное число c, что неравенство $A \leq B+c$ выполнено для всех значений переменных, для которых A и B определены;

$A \overset{c}{\leqslant} B$ означает, что существует такое натуральное число c, что неравенство $A \leq B \cdot c$ выполнено для всех значений переменных, для которых A и B определены;

$A \overset{c}{\simeq} B$ означает, что $A \overset{c}{\leqslant} B$ и $B \overset{c}{\leqslant} A$;

$A \overset{c}{\geqslant} B$ означает, что $A \overset{c}{\leqslant} B$ и $B \overset{c}{\leqslant} A$;

$f : A \to B$ или f есть отображение (= функция) из A в B } означает, что область определения f есть подмножество множества A, а множество значений f есть подмножество множества B ;

f есть отображение множества X } означает, что область определения f есть X;

f есть отображение на Y } означает, что множество значений f есть Y;

$\mathcal{F}(X,Y)$ – множество всех функций из X в Y ;

$Com(X,Y)$ – множество всех вычислимых функций из X в Y (см. ч.I, § 7);

2^W – множество всех подмножеств множества W ;

$Gen(W)$ – множество всех породимых подмножеств множества W (см. ч.I, § 7);

Числовая функция - функция из $\mathbb{N}^s$ в $\mathbb{N}$;

Числовое множество - подмножество множества $\mathbb{N}^s$;

Словариая функция - функция из множества всех слов в некотором алфавите $Б_1$ в множество всех слов в некотором алфавите $Б_2$;

Словарное множество - некоторое множество слов в некотором алфавите Б;

Решимая проблема - проблема, имеющая решение.

Часть первая

ОСНОВНЫЕ ОТКРЫТИЯ ОБЩЕЙ ТЕОРИИ АЛГОРИТМОВ

В общей теории алгоритмов мы выделяем дескриптивную сторону, занимающуюся лишь вопросами о наличии или отсутствии алгоритмов и исчислений, приводящих к заданной цели (но без оценки затрат на достижение этой цели), и о способах задания этих алгоритмов и исчислений, и метрическую сторону, занимающуюся оцениванием сложности процессов вычисления и порождения. Понятия метрической теории алгоритмов еще не сложились в единую и стройную систему.

Основные открытия, связанные с общими понятиями алгоритма и исчисления, таковы:

1. Общее понятие алгоритма как самостоятельное (отдельное) понятие.
2. Представительные вычислительные модели.
3. Общее понятие исчисления как самостоятельное (отдельное) понятие.
4. Представительные порождающие модели.
5. Выяснение связей между алгоритмами и исчислениями.
6. Время и емкость как сложности вычисления и порождения.
7. Вычислимые функции и породимые множества; разрешимые множества; перечислимые множества.
8. Понятие μ-рекурсивной функции.
9. Возможность арифметического и даже диофантова представления любого перечислимого числового множества.
10. Построение неразрешимого породимого множества.
11. Проблема сводимости Поста.
12. Понятие относительного алгоритма, или алгоритма с оракулом.
13. Понятие вычислимой операции.
14. Понятие программы; программы как объекты вычисления и порождения.
15. Понятие нумерации и теория нумераций.
16. Начало создания инвариантной, или машинно-независимой, теории сложности вычислений.
17. Теория сложности и энтропии конечных объектов.
18. Удобные, или экономные, вычислительные модели.

Часть I содержит параграфы с 1-го по 18-й, каждый из ко-

торых посвящен соответствующему открытию и связанным с ним вопросам, а также § 0, посвященный изложению некоторых предварительных понятий. Для большинства терминов место, где они вводятся, легко усмотреть из названий параграфов; вот наиболее существенные исключения: понятия нормы, нормированного ансамбля и ограниченно-искажающего отображения вводятся в § 6, понятия вычислительной, результатной, универсальной, главной (=гёделевой) и оптимальной функций - в § 14.

Многие достижения теории алгоритмов имеют общематематический и, возможно, общечеловеческий интерес. Авторы стремились поэтому к тому, чтобы данный текст был понятен любому математику, а не только специалисту в области теории алгоритмов.

Ради экономии места, однако, в статье не приводятся (за двумя исключениями, о которых ниже) определения конкретных вычислительных и порождающих моделей - машин Тьюринга, нормальных алгорифмов Маркова, канонических и нормальных систем Поста и т.п.; эти определения, так же как и определение μ-рекурсивных (иначе - частичнорекурсивных) функций, могут быть найдены в учебной литературе или в тех оригинальных работах, на которые мы ссылаемся; специальные комментарии по поводу термина "машина Тьюринга" см. ниже в ч.I, § 6. Упомянутые исключения - машины Колмогорова и машины Шёнхаге, определения которых приводятся в § 2. Используемый нами термин "вычислительная модель" можно понимать интуитивно, однако он может рассматриваться и в несколько более точном значении - как собирательное существительное для обозначения любого известного семейства однотипных вычислительных устройств. Например, все машины Колмогорова составляют одно такое семейство, другое семейство составляют все многоленточные машины Тьюринга, третье - все одноленточные машины Тьюринга; мы имеем здесь, следовательно, три вычислительные модели. Все одноленточные машины Тьюринга с фиксированным ленточным алфавитом также образуют вычислительную модель. Совокупность всех машин Колмогорова над колмогоровскими комплексами, размеченными буквами из фиксированного алфавита, также может рассматриваться как вычислительная модель. Аналогично обстоит дело и с термином "порождающая модель". В § 18 термином "вычислительная мо-

дель" обозначается не любая модель, а лишь выбранная из некоторого ограниченного (впрочем, достаточно широкого), но зато более формально описанного класса. Здесь же оговоримся, что понятие "вычислительная модель", употребляемое нами в том же смысле, что и, например, в [Сли 81], не имеет ничего общего с понятием вычислительной модели в смысле [Бах 82].

§ 0. ПРЕДВАРИТЕЛЬНЫЕ ПОНЯТИЯ ТЕОРИИ АЛГОРИТМОВ: КОНСТРУКТИВНЫЕ ОБЪЕКТЫ И АНСАМБЛИ, ЛОКАЛЬНЫЕ СВОЙСТВА И ЛОКАЛЬНЫЕ ДЕЙСТВИЯ

> Предметом для введения в науку обыкновенно назначают предварительные о ней понятия, т.е. такие понятия, которые не могут войти в состав самой науки, однакож существенно к ней относятся и необходимо ею предполагаются.
>
> Макарий.

"Во всем современном математическом мышлении большое место занимает различие между «конструктивным» и «неконструктивным»" - пишет Колмогоров в [Колм 54]. Далее он отмечает: "Любое натуральное число может быть в принципе задано конструктивно в виде

$$1 + 1 + 1 + \ldots + 1".$$

Здесь наглядно проявляется различие между натуральным числом как количественной сущностью (неконструктивным объектом) и его заданием в виде цепочки единиц и плюсов (конструктивным объектом). Для теории алгоритмов это различие имеет не столько философский, сколько совершенно практический характер: алгоритмы могут иметь дело только с комбинациями знаков, т.е. только с конструктивными объектами. Можно усмотреть некую тонкую разницу между теорией алгоритмов и теорией вычислимых функций: последняя имеет дело не только с объектами, конструктивными в прямом смысле, но и с объектами, имеющими всего лишь конструктивные задания - например, с натуральными или рациональными числами (для рациональных чисел конструктивными заданиями служат дроби, и вычислимость функции от рационального аргумента обеспечивается наличием алгоритма, дающего равные результаты при обработке, скажем, дробей $\frac{3}{7}$ и $\frac{12}{28}$).

А.П.Ершов справедливо включает понятие конструктивного объекта в основы теоретического программирования - см. [ЕршА 77, § 2.1]. Это понятие следует признать не только основным, но и первичным, неопределяемым. Все попытки его определить - и наше изложение не явится исключением - неизбежно уклоняются либо в сторону расплывчатого описания, либо в сторону определения частных видов конструктивных объектов. Как все неопределяемые понятия, оно усваивается на примерах. К таким примерам мы сейчас и перейдем.

Первые примеры конструктивных объектов: слова и деревья

Наиболее изученными конструктивными объектами являются слова, составленные из букв какого-либо конечного алфавита Б, короче - слова в Б, еще короче - Б-слова. Слова часто служат основными объектами при построении теории алгоритмов; наиболее последовательный пример такого построения - теория нормальных алгорифмов Маркова (см. монографию [Марк 54], где словам посвящена глава I). Понятие слова встречается, и притом в алгоритмическом контексте, уже у Туэ в [Туэ 14] - см. ниже добавление к § 3. Однако уже Туэ осознавал существование конструктивных объектов более общего вида, чем слова, а именно - некоторых древовидных образований (см. [Туэ 10]).

Другой традиционный пример - матрицы с целочисленными коэффициентами, записанными в какой-либо системе счисления.

Наконец, третий важный пример образуют (Б,k)-деревья. Пусть Б - алфавит, k - натуральное число. Тогда (Б,k)-деревом называется дерево со следующими дополнительными свойствами. Одна из вершин выделена и названа корнем (такое дерево называется корневым); на всех ребрах задана ориентация, так что в каждую вершину из корня ведет ориентированный путь. Предполагается, что каждая из вершин дерева помечена одной из букв алфавита Б и для каждой вершины все исходящие ребра этой вершины помечены р а з л и ч н ы м и числами из множества {1,...,k} (так что исходящих ребер при данной вершине не более k).

Очевидно, каждое Б-слово можно считать (Б,1)-деревом с корнем в первой букве слова.

Алгебраический пример. Пусть задано конечное множество Б функциональных и предметных имен (сигнатура), причем валентность (число аргументов) функциональных имен не превосходит k. Тогда всякому замкнутому терму сигнатуры Б можно следующим образом поставить в соответствие (Б,k)-дерево. Если a - предметное имя из Б, ему ставится в соответствие дерево из одной вершины, помеченной a. Если f есть n-местное функциональное имя, $t_1,\ldots, t_n$ суть термы, то терму $f(t_1,\ldots,t_n)$ ставим в соответствие дерево с корнем, помеченным f, из которого ведет n ребер в корни деревьев, поставленных в соответствие термам $t_1,\ldots,t_n$; ребра эти занумерованы числами $1,\ldots, n$. На множестве замкнутых термов можно задать структуру алгебры (свободной алгебры, порожденной сигнатурой Б, подробнее об этом будет говориться в добавлении к § 3). Например, свободный группоид с образующими a и b и операцией o вкладывается в множество $(\{a,b,o\}, 3)$-деревьев. Заметим здесь же, что алгебру, свободную в многообразии всех ассоциативных группоидов с единицей - свободную полугруппу - также можно представлять состоящей из конструктивных объектов: ее можно рассматривать как множество слов в алфавите образующих.

Конструктивные объекты: попытка общего описания

В попытках разъяснить наше понимание того, что такое конструктивный объект, мы начнём с более широкого, и также неопределяемого, понятия - понятия конечного объекта. Конечный объект - это объект, о котором можно мыслить, не привлекая абстракции актуальной бесконечности (см. [Наг 77г]), т.е. объект, который может быть как бы предъявлен целиком (разумеется, все сказанное никак не претендует на роль определения, а скорее представляет собой повторение одного и того же). Важно отметить, что конечное множество конечных объектов является конечным объектом. Классический пример конечного объекта - конечный граф.

Некоторые из конечных объектов являются конструктивными объектами (к.о.). Мы считаем, что каждый к.о. состоит из конечного множества элементов, принадлежащих каждый к одному из конечного числа типов (так, слово "слово" состоит из 5 букв,

относящихся к 4 типам) и связанных некоторыми отношениями также из конечного числа типов (так, буквы слова связаны отношением упорядоченного соседства). Таким образом, к.о. имеет расчлененное (дискретное) строение и составлен из отдельных элементов, как молекула из атомов. Мы предполагаем, далее, что для к.о. задана (не м о ж е т б ы т ь задана, а у ж е задана) некоторая "внутренняя система координат", позволяющая однозначно локализовать любой его элемент. Такой внутренней системой координат обладает слово (можно говорить о второй букве слева), матрица (можно говорить о пересечении восьмой строки и третьего столбца), (Б,к)-дерево (можно задать путь от корня к данной вершине). Такой системой координат не обладает ни конечное множество, взятое само по себе (разумеется, на него можно наложить внутреннюю систему координат, упорядочив его), ни конечный граф (и на него можно наложить некую систему, указав порядок обхода вершин). В обоих последних примерах в рассматриваемый объект можно ввести внутреннюю систему координат, но ввести неоднозначно, так что до введения такой системы мы не в состоянии указать "адрес" интересующего нас элемента. Говоря о системе координат, естественно требовать наличия "начала координат"; таким образом, в каждом к.о. некоторый его элемент выделен как начальный - это тот элемент, с которого начинается "чтение" объекта (первая буква слова, первый элемент первой строки матрицы, корень дерева и т.д.). Именно наличие у к.о. внутренней системы координат обеспечивает возможность его ввода в вычислительное устройство.

За исключением [Кри 77а, § 2], где предлагается определение так называемого "конструктивного элемента", в литературе встречаются лишь краткие пояснения к понятию конструктивного объекта - см. [Шан 62], [Манин 80, гл.I, § 6]; в последней из перечисленных работ термином "конструктивный объект" обозначается то, что мы обозначали термином "конечный объект". Мы убеждены, что алгоритмы требуют именно конструктивных объектов и не могут оперировать непосредственно с конечными объектами, не являющимися конструктивными. Тот факт, что конечные объекты могут быть заданы конструктивно, показывает лишь, что для них существуют конструктивные задания (единообразные для

каждого класса "однородных между собой" конечных объектов), и эти конструктивные задания являются конструктивными объектами. Каждый такой к.о. есть исходный конечный объект, дополненный тем или иным способом внутренней системой координат; ввиду множественности этих способов, у одного и того же объекта может быть много различных заданий. Именно с такими заданиями и работают алгоритмы. В случае, когда алгоритм работает согласованным образом (т.е. в применении к заданиям одного и того же выдает в качестве результата снова задания одного и того же), можно принять терминологическое соглашение и говорить о вычислимых функциях над объектами, имеющими задания (см. начало настоящего параграфа). Говорить же - при отсутствии такого соглашения - о вычислимых функциях, а тем паче об алгоритмах над конечными объектами как о чем-то само собой разумеющемся, как это сделано в [Шёнф 71, § 1], нам представляется неправомерным.

Локальные свойства и локальные действия: неформальное изложение

Предположим, что в каждом из рассматриваемых к.о. выделен некоторый ограниченный участок ("активная часть" согласно [Колм 53]), состоящий из начального элемента и всех "достаточно близких к нему элементов". Такое выделение может быть осуществлено, например, если на объекте как-то задана метрика, так что можно говорить о расстоянии от одного элемента до другого; тогда активная часть состоит из всех элементов (и, разумеется, имеющихся связей между ними), которые удалены от начального не более чем на такое-то число, называемое показателем локальности, или радиусом активной части. Так, для слова его активная часть - это начало слова; длина этого начала зависит от выбранного показателя локальности - радиуса активной части. В общем случае активную часть можно интерпретировать как ту область конструктивного объекта, которая доступна наблюдению наблюдателя или устройства, находящегося в начальном элементе. Очевидно, что представляют специальный интерес свойства к.о., зависящие только от активной части объекта. Поскольку - при данном радиусе - возможно лишь конечное число активных частей (на интуитивном уровне это ясно, для конкрет-

ных видов к.о., рассматриваемых ниже, это легко проверяется), каждое локальное свойство можно задать конечным перечнем удовлетворяющих ему активных частей.

Л о к а л ь н о е д е й с т в и е состоит в замене активной части рассматриваемого объекта на некоторый другой "кусок". Как происходит эта замена и, в частности, как новый кусок подклеивается к остающейся неизменной "неактивной" части обрабатываемого объекта - все это, разумеется, требует разъяснений. Для достаточно широких классов к.о. эти разъяснения будут сделаны ниже.

Колмогоровские комплексы

Подход, согласно которому конструктивный объект представляет собою конечное множество элементов, связанных какими-то отношениями, причем и элементы и отношения могут быть одного из заранее указанного конечного числа типов, принадлежит Колмогорову (см. [Колм 53], [Колм Усп 58]); вместо термина "конструктивный объект" Колмогоров употреблял термин "состояние" (алгоритмического процесса).

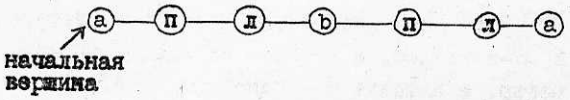
Реализация этого подхода приводит к понятию колмогоровского Б-комплекса. Точное определение этого понятия будет приведено ниже. Сейчас мы поясним, каким образом осуществляется переход от конструктивных объектов как "конечных множеств элементов, связанных какими-то отношениями", к колмогоровским Б-комплексам. Если и элементы, и отношения изображать в виде вершин некоторого графа, соединяя ребрами вершины-отношения с вершинами-элементами и помечая каждую вершину символом соответствующего типа, получится размеченный граф, т.е. граф, вершины которого помечены буквами некоторого конечного алфавита. Мы сознательно упростили картину, на самом деле переход от объекта к изображающему его графу чуть сложнее: ребро не идет непосредственно от отношения к элементу. Между отношением и элементом, участвующим в этом отношении, располагаются еще две промежуточные вершины, помеченные натуральными числами; эти числа указывают, соответственно, порядковый номер элемента среди всех элементов, связанных данным отношением, и порядковый номер отношения среди всех отношений, в которых участвует данный элемент. Оба эти числа считаются ограничен-

ными сверху некоторыми заранее выбранными числами, поэтому добавление соответствующих символов оставляет алфавит разметки конечным. При этом оказывается, что полученный размеченный граф обладает особым "колмогоровским" свойством: для произвольной вершины все соседние с нею вершины несут различные пометки. Вершина, соответствующая начальному элементу, объявляется начальной вершиной и, таким образом, граф оказывается инициальным. В случае связного графа (а только такие графы мы будем рассматривать) инициальность и свойство колмогоровости обеспечивает возможность введения внутренней системы координат: каждую вершину можно однозначно задать, указав цепочку пометок вершин, встречающихся на пути из начальной вершины в задаваемую. Так мы приходим к понятию колмогоровского комплекса:

колмогоровский комплекс над алфавитом Б, или колмогоровский Б-комплекс (короче просто Б-комплекс) есть связный инициальный неориентированный граф, вершины которого помечены буквами конечного алфавита Б и который обладает колмогоровским свойством.

Требование связности комплекса является основным отличием сформулированного только что определения от определения из работы [Колм.Усп.58]. Отметим также, что в отличие от [Колм. Усп.58] мы не требуем, чтобы пометка начальной вершины отличалась от пометок других вершин комплекса, а считаем, что начальная вершина выделена каким-то другим способом.

Как вытекает из самой конструкции Колмогорова, всякий конструктивный объект (если понимать его так, как понимает Колмогоров) естественно изображается в виде некоторого колмогоровского комплекса. Например, вот как изображается в виде комплекса слово aba :



Здесь буквы п и л означают, соответственно, "вправо" и "влево". Таким образом, Б-слово изображается в виде колмогоровского Б'-комплекса, где Б' = БU{л,п}.

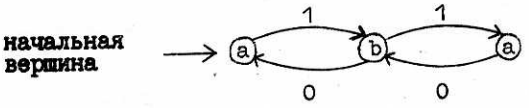
(Б,k)-комплексы

Пусть заданы алфавит Б и натуральное число k . Следующий конструктивный объект называется (Б,k)-комплексом:

инициальный (т.е. с выделенной начальной вершиной) конечный ориентированный граф, являющийся связным (это значит, что всякая вершина достижима из начальной по некоторому ориентированному пути) и несущий следующую разметку на своих вершинах и ребрах: все вершины помечены буквами из Б, и для всякой вершины все исходящие из нее ребра помечены р а з л и ч н ы м и натуральными числами из множества {0,...,k-1} (отсюда вытекает, что исходящая степень любой вершины не превосходит k).

Множество вершин (Б,k)-комплекса U будем обозначать через v(U).

Каждое слово в алфавите Б легко изображается в виде (Б,2)-комплекса, как показывает пример:



Здесь изображено слово aba .

Каждый колмогоровский Б-комплекс следующей процедурой превращается в (Б, k)-комплекс, где k - число букв алфавита Б. Буквы алфавита нумеруются числами от 0 до k - 1, каждое (неориентированное) ребро Б-комплекса заменяется на два ориентированных с противоположными ориентациями, каждому возникшему ориентированному ребру присваивается в качестве его числовой пометы номер буквы, стоящей при той вершине, в которую ведет рассматриваемое ребро.

Частным случаем (Б, k)-комплексов являются структуры Шёнхаге из [Шёнх 80] (см. ниже добавление к § 2): структура Шёнхаге есть (Б, k)-комплекс, в котором из каждой вершины исходит ровно k ребер, а алфавит Б - однобуквенный.

Ансамбли

Как можно было заметить из предыдущего изложения, конструктивные объекты ведут себя как стадные образования. Действительно, они естественным образом группируются в скопления,

состоящие из всех "сходных (или однородных) между собой" объектов. Одно такое скопление образуют, при фиксированном Б, всевозможные Б-слова, другое, при фиксированных Б и k, всевозможные (Б,k)-комплексы. Эти скопления естественно было бы называть "стадами", "роями", "косяками", "стаями" и даже "выводками". Мы, однако, предпочитаем называть их ансамблями (в [Усп Сем 81] - "aggregates"): этот термин более нейтрален и менее зоологичен. Мы не будем пользоваться термином "пространство конструктивных объектов", так как его легко спутать с термином "пространство" из [Шёнф 71, § 1], обозначающим совокупность конечных (не обязательно конструктивных) объектов. Отметим, что далеко не всякое разрешимое (см. ниже, § 7) подмножество ансамбля следует считать ансамблем, поскольку ансамбль должен включать в себя в с е объекты данного (достаточно просто определяемого)типа. Например, множество всех Б-слов, длина которых есть полный квадрат, не образует ансамбля. В этом - еще одно отличие излагаемого здесь подхода от изложенного в [Шёнф 71, § 1], где, по-видимому, всякое разрешимое подмножество пространства само есть пространство.

Похоже, что понятие ансамбля является более первичным, чем понятие конструктивного объекта - объект может восприниматься как конструктивный только в рамках некоторого ансамбля.

Для каждого конечного алфавита Б и каждого натурального числа k можно указать следующие основные ансамбли.

1. Ансамбль Б-слов. Если Б содержит более одной буквы, ансамбль Б-слов называется словарным ансамблем (над Б); если алфавит Б является однобуквенным, мы не называем ансамбль Б-слов словарным - в этом случае ансамбль Б-слов естественно отождествляется с натуральным рядом $\mathbb{N}$.

2. Для произвольных конечного Б и натурального k - ансамбль (Б,k)-деревьев.

3. Ансамбль колмогоровских Б-комплексов, или колмогоровский Б-ансамбль.

4. Ансамбль всех таких колмогоровских Б-комплексов, у которых степень каждой вершины не превосходит k.

5. Ансамбль (Б,k)-комплексов, или, короче, (Б,k)-ансамбль.

6. Расширенный ансамбль Б-слов. Он состоит из всех изображений Б-слов в виде (Б,2)-комплексов (см. пример выше), но

с исключением требования, чтобы начальной вершиной становилась первая буква слова: разрешается, чтобы любая вершина была начальной.

Для дальнейшего можно считать, что понятие ансамбля (а вместе с ним и понятие конструктивного объекта) полностью задается сделанным только что перечислением и, таким образом, является точно определенным. Мы бы не хотели, однако, полностью уничтожать хотя и расплывчатое, но интуитивно ясное и глубокое содержание этих понятий.

Для любых двух ансамблей существует взаимно однозначное соответствие между ними, которое задается (в обоих направлениях) алгоритмами; такое соответствие называется изоморфизмом ансамблей. В этом смысле все ансамбли изоморфны.

Мы уже видели, что между ансамблями имеются некоторые естественные вложения, например, ансамбль Б-слов естественно вкладывается в ансамбль (Б,1)-деревьев. Легко видеть, что любой из рассмотренных ансамблей вкладывается - при подходящих Б и k - в (Б,k)-ансамбль. Поэтому при построении теории алгоритмов можно было бы ограничиться лишь (Б,k)-ансамблями. Так мы и поступим в следующем разделе при строгом определении понятия локального действия (неформально это понятие было уже разъяснено).

Декартово произведение ансамблей легко вкладывается в некоторый новый ансамбль. Множество кортежей (= конечных последовательностей) элементов данного ансамбля также легко вкладывается в подходящий ансамбль. Как мы уже отмечали, конечное подмножество ансамбля не является конструктивным объектом. Чтобы алгоритмы могли работать с такими подмножествами, последние сперва надо как-то конструктивно задать - например, посредством кортежей; тогда надо ограничиться только такими алгоритмами над кортежами, результаты которых не зависят от порядка членов кортежей. В дальнейшем, говоря об алгоритмах и исчислениях, работающих с конечными подмножествами ансамблей, мы будем считать, что все соответствующие соглашения уже приняты.

Локальные свойства и локальные действия: формальное определение

Пусть фиксированы алфавит Б и натуральное число k. Мы

сейчас определим понятия локального свойства и локального действия в ансамбле (Б,k)-комплексов, следуя данным выше неформальным разъяснениям этих понятий. Более точно, мы определим понятия r-локального свойства и r-локального действия, где r есть некоторое натуральное число (показатель локальности; чем он меньше, тем свойство или действие "локальнее").

Локальные свойства. Данное нами выше неформальное разъяснение понятия локального свойства (как свойства, зависящего только от активной части комплекса) становится точным, если дать точное определение активной части. Именно, назовем r-окрестностью данного (Б,k)-комплекса (Б,k)-комплекс, состоящий из всех вершин исходного комплекса, достижимых из начальной по ориентированным путям длины не более r, и всех соединяющих эти вершины ребер. Теперь r-локальным свойством (Б,k)-комплексов назовем любое свойство (Б,k)-комплексов, зависящее только от их r-окрестностей. В дальнейшем упоминания параметров Б,k и r будут опускаться, если значения этих параметров ясны из контекста (так что, например, мы будем говорить просто "локальные свойства", не указывая ни Б, ни k, ни r).

Локальные действия в (Б,k)-ансамбле будут преобразовывать некоторые (Б,k)-комплексы в (Б,k)-комплексы. Дадим точное определение, уточняющее данное выше неформальное описание этого понятия. Любое r-локальное (Б,k)-действие задается указанием числа r и строчки

$$U \to \langle W, \gamma, \delta \rangle$$

где U и W суть некоторые (Б,k)-комплексы, γ есть отображение из v(U) в v(W), δ есть инъективное отображение из v(U) в v(W) (напомним, что v(G) есть множество вершин комплекса G).

Применение r-локального (Б,k)-действия к (Б,k)-комплексу S состоит в следующем переходе от комплекса S к комплексу S* (при этом сначала формируется вспомогательный комплекс S'):

1) r-окрестность комплекса S отождествляется с U (если это возможно; если нет - действие неприменимо). Это отождествление происходит однозначно в силу определения комплекса;

2) вершинами комплекса S' объявляются все вершины комплексов $S \setminus U$ и W;

3) если $b \in U$, $a \in S \setminus U$, в S было ребро $\langle a,b \rangle$ и $\gamma(b)$

определено, то $\langle a, \gamma(b)\rangle$ становится ребром в S' с тем же номером, который имело в S ребро $\langle a,b\rangle$ (как ребро, выходящее из a);

4) если $a \in U$, $b \in S\setminus U$, в S было ребро $\langle a,b\rangle$ и $\delta(a)$ определено, то $\langle \delta(a), b\rangle$ становится ребром с таким же номером, какой имело ребро $\langle a,b\rangle$ (инъективность δ гарантирует различие номеров у всех ребер, выходящих из одной вершины S');

5) начальная вершина W объявляется начальной вершиной S', и из S' выкидываются все вершины, не достижимые из начальной, вместе с ведущими из этих вершин ребрами. То, что получилось, и есть S^*.

Определение результата применения r-локального (Б,k)-действия к (Б,k)-комплексу завершено. Как и в случае свойств, упоминания Б, k и r мы будем опускать, если значения соответствующих параметров ясны из контекста.

Можно рассматривать локальные действия, не выводящие за пределы заданного подансамбля (т.е. подмножества, являющегося ансамблем) (Б,k)-ансамбля. В силу сделанных выше замечаний о вложении одних ансамблей в другие любой из явно перечисленных нами ансамблей есть подансамбль подходящего (Б,k)-ансамбля. Например, можно требовать, чтобы локальное действие не выводило за пределы ансамбля Б-слов, т.е. было действием в этом ансамбле. Тогда это будет действие, состоящее в замене начала слова на другое начало. Можно требовать, чтобы действие не выводило за пределы расширенного ансамбля Б-слов. Тогда это будет действие, состоящее в замене заданного вхождения подслова на некоторое слово. Наконец, можно требовать, чтобы действие не выводило за пределы колмогоровского Б-ансамбля. Тогда мы приходим к понятию колмогоровского Б-действия.

Колмогоровское r-локальное Б-действие задается указанием числа r и строчки $U \to \langle W, \gamma\rangle$, где U, W суть Б-комплексы, а γ есть взаимнооднозначное отображение из $v(U)$ в $v(W)$, сохраняющее пометки и такое, что для всякой вершины a, для которой $\gamma(a)$ определено, множество пометок при вершинах, соединенных в W ребром с $\gamma(a)$, содержится в множестве пометок при вершинах, соединенных в U ребром с a. Применение локального действия к Б-комплексу определяется так же, как в случае произвольных (Б,k)-комплексов; при этом в качестве δ мы берем

отображение, совпадающее с γ.

§ 1. ОБЩЕЕ ПОНЯТИЕ АЛГОРИТМА КАК САМОСТОЯТЕЛЬНОЕ (ОТДЕЛЬНОЕ) ПОНЯТИЕ

Самым главным открытием в науке об алгоритмах, безусловно, было открытие самого понятия алгоритма в качестве новой и отдельной сущности. Мы хотим подчеркнуть, что это открытие следует рассматривать как отдельное, не смешивая его с открытием представительных вычислительных моделей (Тьюринга, Поста, Маркова, Колмогорова), о которых пойдет речь в § 2. Иногда ошибочно полагают, что понятие алгоритма не может быть удовлетворительно воспринято в отрыве от тех или иных формальных конструкций, имея в виду при этом прежде всего вышеназванные модели. Однако эти конструкции были предложены как раз для уточнения, или адекватной формализации, общего интуитивного понятия алгоритма (а вернее сказать, понятия вычислимой функции), само же это понятие, таким образом, признавалось существующим независимо от (а в историческом плане - существующим до) указанных формализаций. Как указывал Гёдель в [Гёд 58], вопрос о том, правильно ли определение вычислимости функции по Тьюрингу, не имеет смысла, если понятие вычислимой функции не является интуитивно понятным априори. Подобная ситуация достаточно типична: общее интуитивное понятие, скажем, поверхности имеет смысл независимо от дефиниций, предполагаемых топологией или дифференциальной геометрией; разница лишь в том, что понятие поверхности известно с античных времен (оно встречается в первых строках евклидовых "Начал"), а понятие алгоритма появилось, по-видимому, лишь в XX веке.

Для историков математики было бы поучительной задачей проследить возникновение и формирование (которое, возможно, еще не завершилось) понятия алгоритма. Евклид и аль-Хорезми сообщили первые примеры алгоритмов, применяемых сегодня. Что же касается общего понятия алгоритма - эффективной вычислительной процедуры, - то едва ли не самые ранние примеры использования такого понятия встречаются в первой четверти XX века, в работах Бореля (1912 г.) и Вейля (1921 г.). Борель [Бор 12, с. 161] выделяет "вычисления, которые могут быть реально осуществлены" и подчеркивает: "Я намеренно оставляю

в стороне большую или меньшую практическую длительность; суть здесь та, что каждая из этих операций осуществима в конечное время при помощи достоверного и недвусмысленного метода" (с. 162; перевод заимствуем из [МедФ 76, с. 100]). Вейль (см. [Вей 21]) при обсуждении понятия "functio discreta" явно выделяет среди произвольных соответствий алгоритмические. Оба они по существу приходят к понятию вычислимой функции (а Борель - даже к термину "fonction calculable" , правда, обозначающему несколько иное алгоритмическое понятие, см.ниже ч.II, § 4). В докладе [Чёрч 36], представленном в 1935 году, Чёрч уверенно пользуется термином "эффективно вычислимая ⟨effectively calculable⟩ функция", считая его общепонятным и предшествующим какой бы то ни было формализации. Отметим, что в ранних англоязычных сочинениях по теории алгоритмов (см., например, [Тью 36],[Тью 37а] и особенно [Тью 39, § 2]) термин "effectively calculable" относится к интуитивному понятию, а термин "computable" - к функциям, вычисление которых осуществляется некоторой вычислительной моделью. В современных публикациях термин "computable" относится к каждому из этих понятий, тогда как термин "calculable" не имеет широкого распространения.

Понятие алгоритма, подобно понятиям множества и натурального числа, принадлежит к числу понятий столь фундаментальных, что не может быть выражено через другие (в частности, теоретико-множественные), а должно рассматриваться как неопределяемое. Лишь как пояснения, а не как определения, следует расценивать формулировки типа "Алгоритм - это точное предписание, которое задает вычислительный процесс, начинающийся с произвольного (но выбранного из фиксированной для данного алгоритма совокупности) исходного данного и направленный на получение полностью определяемого этим исходным данным результата". Однако и подобных пояснений достаточно для установления некоторых содержательных фактов - например, того факта, что не каждая функция с натуральными аргументами и значениями может быть вычислена каким-либо алгоритмом (поскольку невозможно несчетное количество предписаний). Более продвинутое изучение требует дальнейших уточнений (см. такие уточнения в [Колм 53], [Колм Усп 58], [Родж 67, § 1.1], [Кнут 68, § 1.1], [Усп 70],

[Усп 77]). Колмогоров пишет:

"Мы отправляемся от следующих наглядных представлений об алгоритмах:

1) Алгоритм Г, примененный ко всякому "условию" ("начальному состоянию") А из некоторого множества $\mathcal{D}$ ("области применимости" алгоритма Г) дает "решение" ("заключительное состояние") В.

2) Алгоритмический процесс расчленяется на отдельные шаги заранее ограниченной сложности; каждый шаг состоит в "непосредственной переработке" возникшего к этому шагу состояния S в состояние $S^* = \Omega_\Gamma(S)$.

3) Процесс переработки $A^0 = A$ в $A^1 = \Omega_\Gamma(A^0)$, A^1 в $A^2 = \Omega_\Gamma(A^1)$, A^2 в $A^3 = \Omega_\Gamma(A^2)$ и т.д. продолжается до тех пор, пока либо не произойдет безрезультатная остановка (если оператор Ω_Γ не определен для получившегося состояния), либо не появится сигнал о получении "решения". При этом не исключается возможность неограниченного продолжения процесса (если никогда не появится сигнал о решении).

4) Непосредственная переработка S в $S^* = \Omega_\Gamma(S)$ производится лишь на основании информации о виде заранее ограниченной "активной части" состояния S и затрагивает лишь эту активную часть." ([Колм 53]).

Формулировка Колмогорова содержит по крайней мере две существенные идеи. Первая из этих идей - идея итеративности алгоритмического процесса. Формулировка предлагает общую схему детерминированного преобразования одних объектов в другие - схему, согласно которой всякое такое преобразование представляет собою результат многократного применения одной и той же, фиксированной для данного преобразования, сравнительно простой операции (называемой у Колмогорова "оператором") непосредственной переработки. В нормальном алгорифме Маркова, например, операция непосредственной переработки задается списком пар слов $\langle A_1, B_1\rangle, \ldots, \langle A_n, B_n\rangle$; применение операции к слову P заключается в обнаружении того наименьшего i, для которого A_i имеет вхождение в P, и в замене самого левого из таких вхождений на B_i.

Другая существенная идея формулировки Колмогорова выражена в ее последнем пункте; это идея локальности каждого отдель-

него шага. В [Колм 53] (см. также [Колм Усп 58]) предложено, по существу, общее понятие локальной операции (сам термин "локальная операция", впрочем, отсутствует в указанных публикациях: Колмогоров говорит о "непосредственной переработке"). Локальная операция состоит в удалении заранее ограниченного "куска" обрабатываемого объекта (состояния) и замене этого удаляемого "куска" на другой "кусок", определяемый в зависимости от первого (точнее об этом будет сказано ниже, в § 2, при определении оператора непосредственной переработки для машин Колмогорова).

Все вычислительные модели с локальным преобразованием информации легко могут быть описаны в колмогоровских терминах. Мы будем называть их моделями колмогоровского типа. Модели Поста и Тьюринга - примеры таких моделей. С другой стороны, модели с нелокальными шагами, такие как нормальные алгорифмы Маркова (см. [Марк 51], [Марк 54]) или машины с произвольным доступом к памяти (см. [Ахо Хоп Уль 74], [Сли 81]), требуют предварительного расщепления каждого своего шага на локальные шаги (осуществляемые локальными операциями) и, следовательно, не являются моделями колмогоровского типа. Для того, чтобы задать вычислительную модель с локальным преобразованием информации, нужно конкретизировать встречающиеся в формулировке Колмогорова понятия "состояние", "непосредственная переработка", "активная часть", "сигнал о решении". Заметим, между прочим, что состоянием - в колмогоровском смысле - машины Тьюринга мы должны считать не то, что нередко называют ее (внутренним) состоянием (см. [Марк 70, § 5]), или конфигурацией (см. [Кли 52, § 67]), а то, что в [Кли 52, § 67] названо ситуацией ⟨situation⟩, или полной конфигурацией. Эти ситуации, или полные конфигурации, машины Тьюринга естественно было бы называть также "полными состояниями". (Отметим, что в [Маль 65, § 12] термином "состояние" или синонимичным термином "конфигурация" обозначается, в колмогоровском духе, именно полное состояние, или полная конфигурация.) Итак, каждая модель предполагает свою конкретизацию колмогоровских понятий.

В [Колм 53] предложена наиболее общая схема такой конкретизации. Эта схема может рассматриваться как адекватная формализация точного понятия алгоритма (при работе с моделями с

нелокальным преобразованием информации нужно разбить нелокальные шаги на локальные, как было замечено выше).

Вычислительную модель, определяемую этой наиболее общей схемой, мы называем "машины Колмогорова" (точное определение будет дано ниже в § 2).

Мы видим, что общее понятие алгоритма опирается на некоторые предварительные понятия - прежде всего на понятие состояния алгоритмического процесса и понятие непосредственной переработки; оба эти понятия уточняются на основе материала § 0: состояния суть конструктивные объекты, а непосредственная переработка (в случае локального преобразования информации) задается набором локальных действий (см. ниже § 2).

Состояние вычисления (в колмогоровском смысле) для произвольной вычислительной модели может не быть непосредственно конструктивным объектом. Для машины Тьюринга, например, ее полное состояние в каждый отдельный момент времени включает в себя информацию о записи на ленте, о положении головки и о внутреннем состоянии: для машины Поста надо знать запись на ленте, положение головки и адрес, или номер, подлежащей выполнению команды. Путем подходящих соглашений о способе кодирования можно, однако, оформить каждое такое состояние в виде конструктивного объекта и, более того, в виде объекта из некоторого е д и н о г о (для рассматриваемой вычислительной модели) ансамбля. Если мы рассматриваем, например, машины Тьюринга с данным рабочим алфавитом А, то, закодировав каждое внутреннее состояние в виде некоторого слова подходящего алфавита Б, не пересекающегося с А, введя специальный символ для обозначения положения головки и т.д., можно оформить каждое состояние машины в виде слова в некотором фиксированном алфавите (включающем алфавиты А,Б и, если нужно, еще некоторые знаки) или в виде комплекса некоторого фиксированного колмогоровского ансамбля. Если мы рассматриваем в качестве вычислительной модели машины Тьюринга с не фиксированным заранее рабочим алфавитом, то должны договориться о способе кодирования букв произвольного алфавита словами некоторого фиксированного алфавита и т.п. Во всех случаях такое "погружение" множества полных состояний машины в некоторый ансамбль легко осуществить, и в дальнейшем мы будем предполагать его выпол-

ненным.

X - Y -алгоритмы

Полное описание всякого алгоритма начинается с того, что фиксируются два ансамбля - "ансамбль X допустимых исходных данных (или допустимых входов)" и "ансамбль Y допустимых результатов (или допустимых выходов)". Ансамбль X называется ансамблем входов, а ансамбль Y называется ансамблем выходов. Произвольный алгоритм с ансамблем входов X и ансамблем выходов Y кратко называется X - Y-алгоритмом. Имеется в виду, что мы пытаемся применить X - Y -алгоритм к каждому элементу X и что если результат существует, он принадлежит Y . Областью определения, или областью применимости, алгоритма является подмножество ансамбля входов; это подмножество состоит из всех тех входов, для которых алгоритм выдает результат. Каждый алгоритм задает функцию, определенную на его области применимости: значение функции на аргументе x равно результату алгоритма при входе x . Про такую функцию говорят, что она вычисляется рассматриваемым алгоритмом. Два алгоритма называются эквивалентными, если их области определения совпадают и для каждого входа из этой общей области совпадают соответствующие результаты; эквивалентные алгоритмы, таким образом, вычисляют одну и ту же функцию. Пусть A и B суть некоторые множества. Если область определения алгоритма $\mathcal{A}$ является подмножеством множества A и каждый результат алгоритма $\mathcal{A}$ принадлежит B , то $\mathcal{A}$ называется алгоритмом из A в B (пишем: " $\mathcal{A} : A \to B$ ").

Результат применения алгоритма $\mathcal{A}$ к исходному данному x обозначается $\mathcal{A}(x)$. Таким образом, эквивалентность алгоритмов $\mathcal{A}$ и $\mathcal{B}$ может быть записана как $\mathcal{A}(x) \simeq \mathcal{B}(x)$.

§ 2. ПРЕДСТАВИТЕЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МОДЕЛИ

Открытие, обсуждаемое в этом параграфе, заключается в указании точно очерченного и представительного класса алгоритмов. "Представительность" означает существование таких ансамблей X , Y , что рассматриваемый класс содержит алгоритм, эквивалентный (= определяющий ту же функцию) любому заранее заданному X - Y -алгоритму.

Проблема существования таких классов глубоко нетривиальна. Априори не очевидно, что представительный класс алгоритмов может быть описан в точных терминах и трактоваться как предмет изучения традиционной теоретико-множественной математики. Исторически первыми примерами таких классов служат классы алгоритмов, осуществляемых на вычислительных моделях Тьюринга (см. [Тью 36], а также уточняющую критику в [Пост 47]) и Поста (см. [Пост 36]). Более поздние примеры - нормальные алгорифмы Маркова и алгоритмы, осуществляемые на машинах Колмогорова (алгоритмы Колмогорова).

Машины Колмогорова

Обратимся к цитате из [Колм 53], приведенной нами в § 1. Определение машин Колмогорова (как и всякой вообще вычислительной модели, в том числе с нелокальным преобразованием информации) заключается в конкретизации входящих в эту цитату понятий.

Состояниями объявляются комплексы из некоторого ансамбля колмогоровских Б-комплексов. Фиксируется некоторый показатель локальности r; тем самым возникает понятие r-локального действия. Для машины Γ указывается оператор непосредственной переработки Ω_Γ. Оператор непосредственной переработки задается конечным набором колмогоровских r-локальных действий с попарно различными левыми частями в задающих эти действия строчках. Применение оператора к комплексу S состоит в применении того (ровно одного) действия из этого набора, которое оказывается применимым; если ни одно из действий не применимо, то и оператор не применим. Задаваемые таким образом преобразования комплексов мы и будем называть локальными операциями (в интуитивном смысле этот термин уже появлялся в § 1). Теперь можно сказать, что в машинах Колмогорова непосредственная переработка является локальной операцией.

Работа машины заключается в последовательном переходе от начального состояния A^0 к состоянию $A^1 = \Omega_\Gamma(A^0)$, от состояния A^1 к состоянию $A^2 = \Omega_\Gamma(A^1)$ и т.д., пока не возникнет "сигнал окончания". Требуется, чтобы появление сигнала окончания задавалось локальным свойством. Если сигнал окончания возникает на состоянии A^n, оператор Ω_Γ применяется еще ровно один раз, алгоритмический процесс прекращается, и состояние $A^{n+1} = \Omega_\Gamma(A^n)$

объявляется результатом работы машины при исходном данном A^*.

Колмогоровский подход приводит (при несущественных изменениях в технических деталях) к алгоритмам из [Колм Усп 58]; изложению этих алгоритмов был посвящен § 3 главы I монографии [Глу 64].

Определение Колмогорова естественно применимо и к алгоритмам, обрабатывающим (Б,k)-комплексы. Для этого нужно заменить в определении оператора непосредственной переработки колмогоровские действия на действия в ансамбле (Б,k)-комплексов. С неформальной точки зрения, основное отличие получаемых так алгоритмов от алгоритмов, задаваемых описанными выше машинами Колмогорова, состоит в использовании ориентированных графов (с произвольным числом ребер, входящих в вершины графов одного ансамбля). Имея в виду это соображение, мы закрепляем за получаемыми вычислительными устройствами название "ориентированные машины Колмогорова".

В 1970 г. Шёнхаге (см. [Шёнх 70]) предложил вычислительную модель в виде "машин с модифицируемой памятью" (это есть предложенный в [Сли 81, гл. 3, § 1, п. 1] перевод для термина Шёнхаге "storage modification machines", сокращенно SMM). Во введении к статье [Шёнх 80] Шёнхаге пишет: "Как нам было указано многими коллегами, Колмогоров и Успенский ввели в рассмотрение машинную модель, весьма близкую к машинам с модифицируемой памятью, значительно ранее". Машины Шёнхаге можно рассматривать как специальный вид ориентированных машин Колмогорова (несколько подробнее об этом будет сказано ниже), и поэтому для обозначения этих машин в обзоре [Сли 81] использовался термин "алгоритмы Колмогорова - Шёнхаге" (тогда как для обозначения машин Колмогорова в [Сли 81] использовался термин "алгоритмы Колмогорова - Успенского"). Возможно, было бы правильнее называть ориентированные машины Колмогорова (а тем самым и алгоритмы Шёнхаге) алгоритмами с полулокальным преобразованием информации: действительно, при осуществлении одного шага работы алгоритма вновь появившаяся вершина может оказаться концом сколь угодно большого числа ребер.

Определение машин с модифицируемой памятью, или машин Шёнхаге, приводится нами в добавлении к настоящему параграфу. Вкратце, основные отличия этой вычислительной модели от ориен-

тированных машин Колмогорова состоят в следующем: 1) помечены ребра, а не вершины, 2) команды являются специальным видом команд ориентированных машин Колмогорова, 3) на командах задана управляющая структура с помощью линейного упорядочения команд и операторов перехода, 4) имеются входная и выходная ленты (на которых записываются слова в алфавите {0, I}) и соответствующие команды.

Выше уже отмечалось, что определение алгоритмов, предложенное Колмогоровым, было призвано охватывать все другие виды алгоритмов. И действительно, машины Колмогорова непосредственно моделируют работу других известных видов алгоритмов с локальным преобразованием информации. Для алгоритмов с нелокальным преобразованием информации, таких, как нормальные алгорифмы или машины с произвольным доступом к памяти (см. [Ахо Хоп Уль 74]), требуется предварительное разбиение их шагов на локальные шаги. Тезис о возможности прямого моделирования алгоритмов с локальным преобразованием информации машинами Колмогорова может, на первый взгляд, вызвать возражения. Действительно, существование машины Колмогорова, непосредственно моделирующей машину Тьюринга с плоскостной памятью ("плоской лентой") не является самоочевидным, а требует весьма тонких построений. Так обстоит дело, например, при конструировании машины Колмогорова, решающей задачу "о распознавании самопересечения плоской траектории", см. [Куб 72]; существование же машины Тьюринга с плоской лентой, решающей эту задачу, очевидно. Дело здесь, однако, в том, что в действительности на вход рассматриваемой машины Тьюринга с плоской лентой подается не только аргумент, но и необходимая часть рабочей памяти, т.е. плоскости, стандартным образом разбитой на клетки. Это разбиение и возникающее в силу него отношение соседства клеток машина может использовать в своей работе. Если на вход машины Колмогорова подавать вместе с аргументом такую же часть плоскости, разбитой на клетки, то моделирование машины Тьюринга машиной Колмогорова становится очевидным. Тем не менее, даже и без подачи на вход машины Колмогорова участка плоскости моделирование оказывается, в некотором, уточняемом ниже смысле, возможным - машина Колмогорова оказывается в состоянии "достраивать" нужные участки памяти по ходу вычисления; как

это делается, показано в [Шёнх 80].

Формальные задания

Каждая вычислительная модель влечет существование определенного, специфичного для этой модели, класса формальных заданий алгоритмов, которые могут быть реализованы в этой модели. Например, говоря несколько огрубленно, для нормальных алгорифмов Маркова роль формального задания играет схема нормального алгорифма, для машин Тьюринга - система команд, для машин Колмогорова - множество команд вида $U \to \langle W, Y \rangle$; дальнейшее уточнение понятия формального задания предполагает, что указанные схемы, системы и множества изображены в виде конструктивных объектов. Наше понятие формального задания совпадает по существу с понятием изображения алгоритма, объясненном в [Наг 77б] следующим образом: "Алгоритма изображение - конструктивный объект определенного вида..., содержащий в себе закодированную по фиксированным для алгоритмов данного типа правилам полную информацию об этом алгоритме. Обычно определение А.и. формулируется таким образом, чтобы процедуры получения А.и. по исходному алгоритму и восстановления исходного алгоритма по А.и. осуществлялись по возможности более просто". В действительности, для того, чтобы объяснить какую-либо вычислительную модель, достаточно предъявить определенный Универсальный Рецепт, который по каждому формальному заданию и по каждому входу позволяет получить соответствующий выход. (Ср. формулировку из [Кри 77а, § 0], толкующую понятность алгоритма "как знание того, что должно быть сделано для его исполнения, т.е. как наличие алгоритма выполнения алгоритма".) Если формальные описания и исходные данные разумным образом закодированы, пара ⟨ формальное описание, исходное данное ⟩ превращается в элемент подходящего ансамбля, а Универсальный Рецепт превращается в интерпретатор или даже - в случае представительной модели - в универсальный алгоритм (см. § 14 ниже). В соответствии с только что сказанным, вся дескриптивная теория алгоритмов может рассматриваться как теория единственного универсального алгоритма, построенного на основе определённой представительной модели.

Отметим те особенности, которые отличают формальное задание алгоритма вычисления на данной модели от интуитивного по-

нятия алгоритма как предписания. Прежде всего, как уже упоминалось, формальное задание должно быть конструктивным объектом; предписание же может трактоваться как смысл повелительного текста (тем не менее, как показывают приведенные выше примеры, формальные задания, даже для одной и той же модели, не принадлежат, вообще говоря, никакому единому ансамблю). Далее, формальное задание касается только оператора непосредственной переработки, в частности, входная и выходная процедуры (см. ниже) не входят в формальное задание. Наконец, все предписания (даже если их понимать как тексты, а не как смыслы текстов), описывающие вычисления на данной модели, могут содержать некоторую общую информацию. Например, для нормального алгорифма Маркова к такой информации относится указание о том, что формула подстановки применяется к самому левому вхождению. Для машин Тьюринга предписание содержит объяснение понятий "лента", "головка", "сдвиг влево" и т.д. Естественно, эта общая информация может не входить в формальное задание конкретного алгоритма.

Представительные модели

Пусть X, X', Y, Y' - ансамбли. В силу изоморфизма ансамблей, каждый представительный класс алгоритмов из X в Y автоматически порождает представительный класс алгоритмов из X' в Y'. Поэтому для дескриптивной теории алгоритмов (но не для теории сложности и не для конструкций специальных алгоритмов) достаточно изучать только X-Y-алгоритмы для произвольных, но фиксированных X и Y. В частности, можно предполагать, что $X = Y$. В качестве X, далее, можно взять множество всех слов в каком-либо алфавите (в случае однобуквенного алфавита это множество может интерпретироваться как множество $\mathbb{N}$ всех натуральных чисел (ср. [Родж 67, § 1.10])).

Пусть задана вычислительная модель и пусть X и Y суть ансамбли. Договоримся о какой-либо входной процедуре (или правиле начала), с помощью которой любой элемент $x \in X$ вводится в модель в виде начального состояния (например, о том, как число $n \in \mathbb{N}$ преобразуется в начальное состояние машины Тьюринга - о термине "состояние" см. выше § 1), и о какой-либо выходной процедуре (или правиле извлечения результата), с помощью которой из заключительного состояния извлекается элемент

$y \in Y$. Мы предполагаем, что эти процедуры преобразуют объект локальным образом - в колмогоровских терминах они представляют собой однократное применение подходящих локальных операций. Тогда любое допустимое для данной модели формальное задание алгоритма задает следующий алгоритм из X в Y: берется элемент x и вводится в виде начального состояния, к получившемуся начальному состоянию применяется формальное задание, процесс применения продолжается до тех пор, пока не возникнет заключительное состояние, из него извлекается y. Тем самым с вычислительной моделью в целом оказывается сопряженным некоторый класс алгоритмов из X в Y (который в силу предложенного описания является точно очерченным). В дальнейшем, рассматривая вычислительные модели, мы будем, для упрощения изложения, фиксировать соответствующие ансамбли X и Y, при этом мы не будем упоминать о входной и выходной процедурах, считая, что они определены вычислительной моделью и ансамблями входов и выходов.

Итак, пусть дана вычислительная модель с ансамблем входов X и ансамблем выходов Y. Если класс алгоритмов из X в Y, связанный с этой моделью (и, следовательно, точно очерченный), является представительным (в том смысле, что для каждого X - Y-алгоритма найдется эквивалентный ему алгоритм из этого класса), данная модель называется X- Y-<u>представительной</u>. Модель называется <u>представительной</u>, если она является X- Y-представительной для некоторых X, Y. Вычислительные модели Тьюринга и Поста служат исторически первыми примерами представительных моделей. (Машины Поста являются $\mathbb{N}$- $\mathbb{N}$ -представительными, а машины Тьюринга - X- Y-представительными, если X и Y - словарные ансамбли.) Машины Тьюринга с фиксированным ленточным алфавитом и произвольным числом внутренних состояний также образуют представительную модель. Пример непредставительной вычислительной модели - машины Тьюринга с фиксированным ленточным алфавитом и фиксированным же числом внутренних состояний. Разумеется, всевозможные (ориентированные или неориентированные) машины Колмогорова также составляют представительную модель. Более того, все колмогоровские машины, работающие в подходящем ансамбле Б-комплексов, образуют представительную модель, если алфавит Б содержит не менее че-

тырех букв.

Важно понимать, что представительные вычислительные модели не являются формализациями понятия алгоритма: они только обеспечивают подъездные пути для формализации понятия вычислимой (посредством какого-либо алгоритма) функции. В самом деле, если бы мы объявили, что алгоритмами являются только те, которые реализуются машинами Тьюринга, то нормальные алгорифмы Маркова нельзя было бы рассматривать как алгоритмы, и для них мы не смогли бы измерять, скажем, сложность вычисления.

Машины Колмогорова могут рассматриваться, таким образом, в двух аспектах:

1) как одна из представительных вычислительных моделей;

2) как математическое определение общего понятия алгоритма - в том смысле, что любой алгоритм с локальными шагами непосредственно подпадает под это определение (этот второй аспект выделяет машины Колмогорова среди всех прочих представительных моделей).

Тезис Чёрча

Утверждение о представительности точно очерченного класса алгоритмов (т.е. о представительности соответствующей вычислительной модели) составляет содержание тезиса Чёрча для этого класса или для этой модели. Таким образом, мы понимаем этот тезис в широком смысле (как в [Род 67, § 1.7]). Тезис Чёрча в узком смысле утверждает, что всякая вычислимая функция с натуральными аргументами и значениями частичнорекурсивна (см. [Кли 52, § 63]). Эту формулировку, строго говоря, следовало бы назвать тезисом Чёрча - Клини, поскольку первоначальная формулировка Чёрча говорит лишь о всюду определенных вычислимых функциях и утверждает их общерекурсивность (см. [Кли 52, § 60]). Тьюринг в [Тью 36] и Пост в [Пост 36] утверждают, что класс всех всюду определенных функций, вычислимых в определенной модели, совпадает с классом всех всюду определенных вычислимых функций (для фиксированных ансамблей). Поэтому тезис Чёрча может быть назван тезисом Тьюринга, или тезисом Поста, или тезисом Чёрча - Тьюринга - Поста. Вспоминая роль Клини в формулировании этого тезиса - переход к частичнорекурсивным функциям, - можно также называть его тезисом Чёрча - Тьюринга - Поста - Клини.

Языки программирования

Роль адекватных формализаций понятия алгоритма могут играть и так называемые языки программирования: действительно, эти языки могут быть использованы для задания точно очерченного и представительного класса алгоритмов. При этом не каждый осмысленный текст на языке программирования нужно обязательно понимать как алгоритм: существенно лишь, чтобы каждый алгоритм мог быть записан на языке. Для математика, занимающегося теорией алгоритмов, среди языков программирования наибольший интерес представляют лисп Маккарти и алгол-68 ван Вейнгаардена. Можно строить и так называемые абстрактные языки программирования, выступающие в качестве математических моделей реальных языков программирования; эти абстрактные языки, в свою очередь, также можно рассматривать как формальные описания понятия алгоритма. Среди абстрактных языков программирования выделяем язык операторных алгоритмов А.П.Ершова (см. [ЕршА 60], [ЕршА 62]); эти алгоритмы - под названием "вычислительные алгоритмы" - были изложены впервые в феврале - марте 1958 г. в докладе А.П.Ершова на семинаре П.С.Новикова и С.А. Яновской в Московском университете.

ДОБАВЛЕНИЕ К § 2. МАШИНЫ ШЁНХАГЕ

Здесь мы приводим подробное описание этих машин, следуя [Шёнх 80]. Состояния каждой машины являются конструктивными объектами определенного рода. Начнем с описания этих объектов; как мы уже замечали, это (Б,k)-комплексы, у которых алфавит Б состоит из одной буквы (тем самым можно считать, что вершины комплекса не помечены вовсе) и из каждой вершины выходит ровно k ребер. Шёнхаге называет их Δ-структурами. Приводем определение Δ-структуры из [Шёнх 80]. Пусть задан конечный алфавит Δ. Его элементы называются направлениями (имеется в виду аналогия с направлениями перемещения головки машины Тьюринга) и играют роль, аналогичную роли чисел $0,\ldots,k-1$ в определении (Б,k)-комплексов. Δ-структура - это конечный ориентированный граф, из каждой вершины которого выходит одно и то же количество ребер, помеченных взаимно однозначным образом элементами Δ; одна из вершин выделена и называется начальной вершиной, или центром, Δ-структуры, и все вершины

достижимы по ориентированным путям из этого центра. Несколько более формально. Δ-структура - это тройка $\langle X, a, p\rangle$, где X - конечное множество вершин, $a \in X$ - центр, $p=\{p_\alpha \mid \alpha\in\Delta\}$ - семейство отображений X в X, причем $p_\alpha(x)$ - это вершина, в которую из вершины x идет ребро, помеченное направлением α. Обозначим через Δ^* множество всех слов в алфавите Δ, через $\square$ - пустое слово. Для всякой Δ-структуры $S=\langle X,a,p\rangle$ определено отображение $A_S : \Delta^* \to X$ следующим образом:

$$A_S(\square) = a ; \quad A_S(W\alpha) = p_\alpha(A_S(W)), \quad \alpha \in \Delta, \quad W \in \Delta^* .$$

Если $A_S(W)= x$, то W называется адресом вершины x (в Δ - структуре S). Определение Δ-структуры требует, чтобы все вершины имели адреса.

Машина Шёнхаге имеет входную и выходную ленты с головками на них, на лентах записываются слова в алфавите $\{0, 1\}$. Кроме того, имеется рабочая память, которая в каждый момент работы машины является некоторой Δ-структурой, и, наконец, программа - конечная последовательность команд описываемых ниже типов. Каждая команда может быть либо операцией, либо проверкой, либо остановом. Команда останов имеет очевидный смысл. Операции бывают над входной лентой, выходной лентой и рабочей памятью. Операция над входной лентой одна - сдвиг головки на один символ вправо. Операции над выходной лентой две - печать символа 0 или символа 1 в конце уже напечатанного на выходной ленте слова.

Операции над рабочей памятью бывают одного из трех типов. Мы опишем, как команды каждого из типов преобразуют Δ-структуру $S =\langle X,a,p\rangle$ в новую Δ-структуру $\overline{S} =\langle \overline{X}, \overline{a}, \overline{p}\rangle$.

1. Команда добавления новой вершины - new W, где $W\in \Delta^*$. Новое множество вершин состоит из элементов X и еще одной - новой - вершины, обозначим ее y. Для всех $\delta \in \Delta$ полагаем $p_\delta(y) = a$. Если $W = \square$, то $\overline{a} =y$ и при всех $\delta \in \Delta$, $x\in X$ значение $\overline{p}_\delta(x)$ задается как совпадающее с $p_\delta(x)$. В графовых терминах: к имеющемуся графу добавляется новая вершина, она объявляется начальной, и все ребра, которые должны выходить из этой вершины, ведут в старую начальную вершину. Пусть теперь W имеет вид $U\alpha$, где $U\in \Delta^*$, $\alpha \in \Delta$. Тогда $\overline{a} = a$, $\overline{p}_\alpha(A_S(U))= y$; для всех $z \in X$, $\delta \in \Delta$, если $z \neq A_S(U)$ или $\delta \neq \alpha$, полагаем

$\overline{p}_\delta(z) = p_\delta(z)$. В графовых терминах: добавляется новая вершина, все ребра из нее идут в начальную вершину, одну и ту же и в старом и в новом графе, а в новую вершину ведет ровно одно ребро, это ребро помечено направлением α и исходит из вершины с адресом U.

2. Перенос начальной вершины - set □ to V . Полагаем $\overline{a} = A_S(V)$, к $\overline{X}$ относим вершины, достижимые из вершины $\overline{a}$ по путям, определяемым семейством p. После этого определяем $\overline{p}$ как семейство $\{\overline{p}_\alpha | \alpha \in \Delta\}$, где $\overline{p}_\alpha$ получается сужением p_α на $\overline{X}$.

3. Поворот ребра - set Uα to V , где $U, V \in \Delta^*$, $\alpha \in \Delta$. Полагаем $\overline{a} = a$, $\overline{p}_\alpha(A_S(U)) = A_S(V)$, не меняя $p_\delta(z)$ при других δ, z (в графовых терминах: единственное возможное отличие получающейся Δ-структуры от S состоит в том, что ребро, помеченное направлением α, ведет из вершины с адресом U в вершину с адресом V, а не в ту вершину, куда оно шло в S). После этого, как и в команде второго типа, оставляем лишь достижимые вершины и соответственно сужаем отображения p_δ.

После выполнения операции происходит переход к выполнению следующей в программе команды, если такой нет, то машина останавливается.

Всякая <u>проверка</u> имеет вид if P then μ else ν , где μ, ν - натуральные числа, не превосходящие числа команд в программе, а P - предикат одного из следующих двух видов: 1) $A_S(U) = A_S(V)$, что означает совпадение вершин Δ-структуры с адресами U и V, 2) input = β, что означает совпадение считываемого символа с β, где β есть 0 или 1 или символ конца входной ленты. В результате выполнения проверки память не меняется и происходит переход к команде, которая имеет в программе номер μ, если P истинен, и к команде, которая имеет номер ν, если P ложен.

Работа машины начинается с выполнения первой команды программы, головка на входной ленте ставится на первый символ, память представляет собой Δ-структуру с единственной вершиной - центром.

§ 3. ОБЩЕЕ ПОНЯТИЕ ИСЧИСЛЕНИЯ КАК САМОСТОЯТЕЛЬНОЕ (ОТДЕЛЬНОЕ) ПОНЯТИЕ

Общее понятие <u>исчисления</u>, или дедуктивной системы, столь

же фундаментально, как и понятие алгоритма, и должно рассматриваться отдельно от каких бы то ни было формальных уточнений. Понятие исчисления отражает и обобщает интуитивное представление об индуктивном порождении множества (см. [Мас 67],[Эбб 70], [Мас 79]). Математические истоки понятия исчисления восходят к древности (см. [Яновс 62]). Игры с четкими правилами игры: шахматы, домино, маджонг, различные карточные игры - были, вероятно, первыми реальными примерами исчислений. (Разумеется, рассматривая игру как исчисление, мы полностью отвлекаемся от ее "игровых" аспектов - состязательности и прочего, - а интересуемся исключительно ее, так сказать, "юридической стороной" - возможностью совершать действия, выполняемые по определенным правилам.) Дифференциальное исчисление и интегральное исчисление могут считаться примерами исчислений, если трактовать их как процедуры, позволяющие порождать верные равенства вида $dF(x)=f(x)dx$ и $\int f(x)dx = F(x)$, отправляясь от исходных, или "табличных", равенств и применяя правила типа правила дифференцирования сложной функции или правила интегрирования по частям. Значительную роль в развитии общего понятия исчисления сыграли исчисления математической логики,или логистические системы.Первые логистические системы появились в конце XIX века в работе Фреге (см. [Фре 1879]).

Общее понятие исчисления менее популярно, чем общее понятие алгоритма: в частности, совсем не изучено общее понятие исчисления с локальным преобразованием информации. Возможно, причина такой дискриминации -в давлении вычислительной практики. Терминология, относящаяся к исчислениям (за исключением относящейся к специальным видам исчислений), не устоялась, и авторы должны были лавировать между Сциллой терминологического новаторства и Харибдой терминологической путаницы.

Говоря огрубленно, исчисление есть конечный список разрешительных правил, называемых также порождающими правилами (см. [Шан 55, § 1]), или правилами вывода. Эти правила р а з р е ш а ю т переходить от одних конструктивных объектов к другим (в то время как правила алгоритма п о в е л е в а ю т совершать такие переходы). Типичным примером разрешительных правил служат правила шахматной игры, причем в роли конструктивных объектов, с которыми правила оперируют, высту-

дают шахматные позиции, снабженные указанием об очереди хода и еще некоторой дополнительной информацией. ([Шахм 69, статья 15] именует эту дополнительную информацию "внутренними возможностями позиции", а [Шахм 81, статья 12] - "возможностями игры". Дело в том, что знания одной только позиции, т.е. расположения фигур на доске, и даже очереди хода недостаточно для применения правил игры: необходимо, например, знать еще, ходил ли ранее король и кое-что другое.) Позицию, дополненную всей необходимой для применения правил игры информацией, естественно называть состоянием игры. Именно состояния игры следует рассматривать как те конструктивные объекты, над которыми совершаются операции шахматного исчисления. Другой пример - игра домино. Состояние здесь слагается из расположения костей на столе, информации о наличии костей у игроков и в куче и информации об очереди хода. Как и в шахматах, правила дают возможность переходить от одного состояния к одному из нескольких непосредственно следующих.

Подобно тому, как алгоритм задает алгоритмический, или вычислительный, процесс (т.е. процесс работы алгоритма), каждое исчисление задает исчислительный, или порождающий, процесс, т.е. процесс работы исчисления. Этот процесс разбивается на отдельные шаги (этапы согласно [Шан 55, § 1]). Каждый шаг состоит в получении нового объекта из уже полученных к началу этого шага объектов; получение нового объекта осуществляется путем применения п р о и з в о л ь н о г о разрешительного правила, входящего в данное исчисление. Объекты, к которым применяется правило, называются его посылками. Заметим, что применение одного правила к одним и тем же посылкам может давать разные результаты; правило может применяться различным образом. (Например, имеется правило хода пешки - но оно может применяться к разным пешкам.) Однако, если фиксировать правило и посылки, число различных результатов оказывается всегда конечным. Для каждого правила число посылок фиксировано. Если все такие числа ограничены числом k, исчисление называется k-посылочным. Все вышеприведенные примеры игр являются однопосылочными исчислениями.

Отражением интуитивного представления об объектах, получающихся в процессе работы исчисления, является понятие допус-

тимого (для данного исчисления) объекта. Определение этого понятия индуктивное.

Если b получается из $a_1,\ldots,a_k$ применением одного из разрешительных правил исчисления и если $a_1,\ldots,a_k$ являются допустимыми, то и b объявляется допустимым; это есть шаг индуктивного определения. Начало индукции обеспечивается нольпосылочными правилами: если b удовлетворяет такому правилу (= получается применением этого правила "из ничего"), то b объявляется допустимым. Если нольпосылочных правил нет вовсе, множество допустимых объектов оказывается пустым. В шахматной игре нольпосылочным является правило, задающее начальное состояние, т.е. начальную позицию с соответствующими начальными "внутренними возможностями" (очередь хода за белыми, король не ходил и т.д.). В логических исчислениях в силу нольпосылочных правил объявляются доказуемыми аксиомы.

Всякое исчисление работает с объектами некоторого ансамбля W , называемого рабочей средой исчисления. Все состояния исчислительного процесса лежат в W. Совокупность всех мыслимых состояний игры (шахматной, домино) легко погружается в подходящий ансамбль. Работа исчисления заключается в образовании все новых и новых допустимых элементов рабочей среды, или допустимых состояний. Принципиальная разница между алгоритмическим процессом и исчислительным процессом лежит в следующем. В алгоритмическом процессе каждое возникающее состояние однозначно предопределено предшествующим ходом процесса, в исчислительном же процессе возникающее состояние является всего лишь одним из многих возможных, допускаемых предшествующим ходом процесса. Если понятие времени связывать с чередованием событий (а событие здесь - появление нового состояния), то можно сказать, что в алгоритмическом процессе время течет линейно, а в исчислительном процессе - разветвленно.

Историю появления данного допустимого состояния в исчислительном процессе можно запечатлеть в виде объекта, мыслимого расположенным в пространстве - вывода. Вывод данного допустимого состояния x - это дерево, во всех вершинах которого находятся какие-то допустимые состояния и всем вершинам поставлены в соответствие правила исчисления так, что 1) в корне находится x и 2) для всякой вершины v дерева, если y - сос-

тояние, находящееся в этой вершине, а $y_1,\ldots,y_k$ - все состояния, находящиеся в тех вершинах, куда идут ребра из v, то y получается из $y_1,\ldots,y_k$ по правилу, сопоставленному с вершиной v. (Таким образом, пометки на листьях дерева получены в силу 0-посылочных правил.) Для однопосылочных исчислений получаемые так выводы оказываются просто цепями.

Продолжая уточнять наши представления об исчислениях, мы обнаруживаем, что конечный список разрешительных правил образует хотя и важнейшую, но лишь одну из составных частей исчисления (он составляет, так сказать, "ядро" исчисления - подобно тому, как оператор непосредственной переработки составляет "ядро" алгоритма). Второй составной частью служит инструкция о разделении элементов на основные и вспомогательные. Наличие этих двух составных частей во всяком исчислении явно отмечено в [Цей 64]. Мы будем называть эту инструкцию правилом выделения основных состояний. Необходимость в таком правиле вызвана тем, что нас могут интересовать не все допустимые состояния, а лишь состояния специального вида (они-то и называются основными), тогда как другие состояния рассматриваются лишь как вспомогательное средство для получения основных состояний. Роль правила выделения основных состояний для исчислений аналогична роли сигнала о получении решения для алгоритмов. Наконец, третьей составной частью исчисления является правило извлечения результата, или выходная процедура. Эта процедура аналогична выходной процедуре из § 2. Выходная процедура преобразует каждое основное состояние в некоторый объект. Результат применения выходной процедуры к произвольному допустимому основному объекту называется результатом, или выходом исчисления. Про всякий такой объект будем также говорить, что он порождается исчислением. Про множество всех объектов, порождаемых исчислением, также будем говорить, что оно порождается исчислением. Два исчисления эквивалентны, если они порождают одно и то же множество. Предполагается, что множество, порождаемое исчислением, расположено в некотором ансамбле - ансамбле выходов рассматриваемого исчисления. Поясним все сказанное на двух примерах.

Ш а х м а т н ы й п р и м е р . Цель - построение исчисления, порождающего всевозможные патовые позиции на шахмат-

ной доске, т.е. позиции, возникающие, когда игра пришла к состоянию пата. В качестве разрешительных правил берем правила шахматной игры. Основными объявляем патовые состояния (здесь важно вспомнить, что в состояние включена и очередь хода). Выходная процедура - переход от состояния к позиции (т.е. отбрасывание всей дополнительной информации).

Логистический пример. Цель - построение исчисления, порождающего всевозможные доказуемые формулы какой-либо фиксированной логистической системы (=формальной аксиоматической теории). Пусть Б - алфавит этой системы. Мы предполагаем известными обычные правила образования (= построения) и правила преобразования (= вывода), согласно которым переменные (п), термы (т), формулы (ф) и доказуемые формулы (д) выделяются среди всех слов в алфавите Б (поскольку Б конечен, а количество переменных бесконечно, переменные суть некоторые слова, а не просто буквы). Теперь строим наше исчисление. Состояния в нем имеют вид $\langle a,b\rangle$, где a есть одна из букв "п", "т", "ф" или "д", а b есть слово в Б; все такие состояния вкладываются в подходящую рабочую среду. Основными объявляются состояния вида $\langle д,b\rangle$. Правило извлечения результата - переход от $\langle д,b\rangle$ к b. Разрешительные правила получаются очевидной модификацией упомянутых выше правил образования и преобразования. Так, правилу, гласящему, что всякая переменная есть терм, соответствует однопосылочное правило, разрешающее переход от любого состояния вида $\langle п,b\rangle$ к состоянию $\langle т,b\rangle$ с тем же b, правилу модус поненс соответствует двупосылочное правило, разрешающее переход от $\langle д,b_1\rangle$ и $\langle д,(b_1 \to b_2)\rangle$ к $\langle д, b_2\rangle$. Аксиомам соответствуют нольпосылочные правила: для всякой аксиомы b разрешается "из ничего" сконструировать порожденный объект $\langle д,b\rangle$. Правило суперпозиции термов, дающее по термам t_1 и t_2 новый терм u, приводит к двупосылочному правилу, разрешающему переход от $\langle т,t_1\rangle$ и $\langle т,t_2\rangle$ к $\langle т, u\rangle$. Если среди правил преобразования было, скажем, правило подстановки, которое по доказуемой формуле f, переменной x и терму t дает новую доказуемую формулу g (представляющую собою результат подстановки t в f вместо всех свободных вхождений переменной x), то в наше исчисление должно быть включено трехпосылочное правило, обеспечивающее возмож-

ность перехода от состояний ⟨д,f⟩ , ⟨п,x⟩ , ⟨т, t⟩ к состоянию ⟨д, g⟩. И так далее.

Последний из приведенных примеров показывает, что нет необходимости в принудительном отнесении порождаемых объектов к тому или иному типу (согласно [Шан 55, § 1]) или к той или иной ступени (согласно [Мас 79]): информацию о типе объекта можно упрятать внутрь подходящим образом выбранного состояния. Так, вместо того, чтобы порождать отдельно термы и отдельно (но с помощью термов) формулы, мы совместно порождали объекты вида ⟨т,b⟩ и ⟨ф,b⟩ .

Одни из правил, образующих исчисление, задают те или иные преобразования (или, короче, я в л я ю т с я преобразованиями); таковы правила вывода, правило извлечения результата. Другие правила задают свойства (или, короче, я в л я ю т с я свойствами); таково правило выделения основных состояний.

Что можно сказать о природе всех этих правил? В общем случае, по-видимому, ничего, кроме того, что они должны быть достаточно просты. Аналогично обстоит дело и в случае алгоритмов - в общем случае мы ничего не можем сказать о природе правила непосредственной переработки или правила окончания. Однако для алгоритмов колмогоровского типа, т.е. алгоритмов с локальным преобразованием информации, мы можем сказать, что первое из этих правил представляет собою локальную операцию, а второе - локальное свойство.

Сходным образом мы подойдем и к исчислениям с локальным преобразованием информации. Типичный пример таких исчислений - ассоциативные исчисления и грамматики математической лингвистики (см. добавление к настоящему параграфу, а также [Кли 52, § 71],[Гла 73], [Гла 77], [Сто 80]). Большинство логических исчислений не являются исчислениями с локальным преобразованием информации: правила вывода в них, как правило, нелокальны (см. выше логистический пример). Однако, как и в случае алгоритмов, нелокальные шаги можно расщепить на локальные, т.е., по существу, заменить одно исчисление (нелокальное) на другое (локальное), эквивалентное исходному. Общий вид исчисления с локальным преобразованием информации (или исчисления колмогоровского типа) легко получается конструкцией по-

рождающей модели, аналогичной машинам Колмогорова. Сейчас мы к этому и перейдем.

Итак, даем определение понятия исчисления с локальным преобразованием информации, или, другими словами, исчисления колмогоровского типа. Рабочей средой будем считать некоторый ансамбль (Б,k)-комплексов или колмогоровских Б-комплексов. От выходной процедуры мы требуем, чтобы она представляла собой локальную операцию (см. § 2), от правила выделения основных состояний - чтобы оно было локальным свойством. Каждое из разрешительных правил должно быть локальным действием в смысле § 0.

Отдельного рассмотрения требует случай 0-посылочных разрешительных правил - правил "вывода из ничего". Дело в том, что локальные действия могут применяться к комплексам, а то "ничто", к которому "применяется" 0-посылочное правило, не есть комплекс. Мы будем считать, что каждое 0-посылочное правило объявляет допустимым (="разрешает вывести из ничего") некоторый фиксированный комплекс. Таким образом, допустимыми в силу 0-посылочных правил будут лишь конечное число комплексов (не больше, чем имеется 0-посылочных правил).

В случае k-посылочного правила при $k \geq 1$ разрешительное правило, как уже отмечалось, должно представлять собой локальное действие (в смысле § 1); надо только (при $k > 1$) кортеж комплексов $a_1, \ldots, a_k$, к которому применяется правило, сперва представить в виде одного комплекса (нового, "более богатого" ансамбля - как это указано в § 0).

Простое, но существенное замечание: каждый ансамбль порождается некоторым исчислением (а значит и исчислением колмогоровского типа).

С помощью общего понятия исчисления можно глубже осмыслить многие фундаментальные понятия и результаты математической логики. В частности, знаменитая теорема Гёделя о полноте утверждает, что все истинные формулы логики предикатов первого порядка, т.е. все законы элементарной логики предикатов, могут быть порождены некоторым исчислением; именно в этом - суть теоремы. С чисто математической точки зрения, конкретное указание такого обладающего "свойством полноты" исчисления имеет уже второстепенное значение (удовлетворяющих условию

теоремы полных исчислений - бесконечное количество). (С точки зрения истории науки представляет интерес тот факт, что полное исчисление предикатов было угадано логиками за несколько десятилетий до теоремы Гёделя.) Другая знаменитая теорема Гёделя - теорэма о неполноте - утверждает, что множество всех истинных формул арифметики (а значит, и множество всех общезначимых формул логики предикатов второго порядка) не может быть порождено никаким исчислением.

На базе понятия исчисления по существу можно изложить всю (по крайней мере, дескриптивную) теорию алгоритмов - см. [Цей 64].

Исчисления со входом

В заключение параграфа - об исчислениях со входом. Каждое исчисление можно рассматривать не только как инструмент для образования некоторого множества - а именно, порождаемого множества, но и как инструмент для преобразования одних множеств в другие. Пусть, в самом деле, дано исчисление $\mathcal{L}$ с рабочей средой W, множеством основных объектов $T \subset W$ и порождаемым множеством P, лежащим в некотором ансамбле Y. Фиксируем некоторый ансамбль X - ансамбль входов - и некоторую входную процедуру α, преобразующую элементы X в элементы W (в случае исчислений с локальным преобразованием информации от процедуры α требуется, чтобы она была локальной операцией). Рассмотрим теперь произвольное подмножество $A \subset X$ и соответствующее $\alpha(A) \subset W$. Объявим все элементы из $\alpha(A)$ допустимыми - с тем, чтобы эти новые допустимые элементы включились в индуктивный процесс образования допустимых элементов. Шаг индукции, таким образом, не меняется, но меняется начало индукции: допустимыми объявляются не только результаты применения нольпосылочных правил, но и все элементы из $\alpha(A)$. Новое, расширенное множество допустимых объектов обозначим через Q_A. К элементам пересечения $Q_A \cap T$ применяем правило извлечения результата, и получаем множество $P_A \supset P$. Подобная ситуация имеет место, в частности, когда к формальной аксиоматической теории добавляются аксиомы (A), которые преобразуются этой теорией в расширенное множество теорем P_A; способ ввода α здесь - тождественное преобразование.

Описанная только что процедура задает некоторую операцию,

приводящую от A к P_A ; это P_A естественно обозначить символом $\mathcal{L}(A)$. В этом последнем обозначении в значение символа включена и информация о процедуре ввода α; так понимаемое $\mathcal{L}$ называется исчислением со входом.

Каждое исчисление со входом $\mathcal{L}$ задает, таким образом, свою исчислительную операцию: $A \to \mathcal{L}(A)$. Оказывается, что класс исчислительных операций совпадает с классом вычислимых операций (см. ниже § 13).

Посредством исчислений со входом можно также задавать бинарные отношения. Именно, исчисление $\mathcal{L}$ задает отношение

$$R_{\mathcal{L}} = \{\langle x, y\rangle \mid \mathcal{L}(\{x\}) \ni y\}$$

Всякое заданное таким образом множество пар может быть порождено некоторым исчислением.

ДОБАВЛЕНИЕ К § 3. АЛГЕБРАИЧЕСКИЕ ПРИМЕРЫ

Понятие исчисления имеет глубокие связи с алгеброй. Мы обозначим сейчас три линии таких связей.

Во-первых, аксиомы, задающие тот или иной вид алгебраических систем (полугрупп, колец, упорядоченных групп и т.д.), записаны, как правило, на узком (= элементарном, 1-го порядка) языке логики предикатов. Этим аксиомы алгебры отличаются от аксиом топологии, формулирующихся в языке теории множеств или в предикатной логике высших типов. Поэтому, в силу теоремы Гёделя о полноте, множество всех элементарных (т.е. записанных на том же элементарном языке) следствий тех или иных алгебраических аксиом (скажем, аксиом теории групп) может быть получено как множество, порожденное исчислением предикатов, - если последнее понимать как исчисление со входом и на вход подавать как раз рассматриваемые аксиомы (здесь "следствие аксиом теории групп" понимается как утверждение, верное во всех группах).

Во-вторых, некоторые исчисления, исторически возникшие вне связи с алгеброй, а именно формулируемые ниже ассоциативные исчисления Туэ, имеют ясное алгебраическое содержание. (Как мы увидим, всякое ассоциативное исчисление в алфавите Б задает на множестве слов алфавита Б - рассматриваемом как полугруппа - некоторую конгруэнцию.)

Наконец, в-третьих, для некоторых конечнопорожденных ал-

гебраических систем (а именно, для всех систем, задаваемых квазитождествами, см. ниже) удается непосредственно построить очень простое исчисление, позволяющее получать все следствия из квазитождеств (в частных случаях - из тождеств, из равенств), задающих рассматриваемую систему. (Здесь "следствие" понимается как утверждение, верное во всех алгебраических системах данного вида, имеющих данное множество образующих, и поэтому теорему Гёделя автоматическим образом применить не удается.)

Далее мы не будем касаться намеченной выше первой линии, развитием которой является большая область математической логики - теория моделей, а сосредоточимся на второй и третьей. Начнем мы с исчислений Туэ, приведших их открывателя к постановке алгоритмических проблем алгебры, о которых пойдет речь в ч. II, § 1. Туэ рассматривает конечные ряды знаков ⟨Zeichenreihen⟩, молчаливо предполагая, что они непусты; в дальнейшем это ограничительное предположение было отброшено. Мотивировки рассмотрений Туэ были философские - он интересовался способами формальных записей понятий ⟨Begriffen⟩ и возможностью комбинаторных преобразований одних понятий в другие, эквивалентные (см. [Туэ 10]). Мы изложим построения Туэ, пользуясь вместо термина "ряд знаков" современным термином "слово в алфавите Б". Туэ начинает с того, что выписывает два кортежа непустых слов A и B, состоящие, соответственно, из слов $A_1, \ldots, A_n$ и $B_1, \ldots, B_n$. Два слова U и V в алфавите Б называются эквивалентными (относительно пары A, B), если для некоторого $m \geq 1$ существует такая последовательность слов $C_1, \ldots, C_m$, что $C_1 = U$, $C_m = V$ и для каждого $i = 1, \ldots, m-1$ найдется такое $j \in \{1, \ldots, n\}$, что C_{i+1} получается из C_i заменой некоторого вхождения A_j на B_j или заменой некоторого вхождения B_j на A_j. Проницательный читатель, конечно, уже сообразил, что Туэ изобрел исчисления некоторого специального вида со входом. Правило исчисления:

если U допустимое слово и при некотором $j \in \{1, \ldots, n\}$ слово V получается из U заменой какого-то вхождения A_j на B_j или B_j на A_j, то V допустимое.

Итак, в исчислении одно однопосылочное правило, хотя это правило можно было бы разбить на несколько, введя свое правило

замены для каждого отдельного j. Остается разобраться, что здесь будет рабочей средой, основными состояниями и выходной процедурой. Рабочая среда проста - это ансамбль Б-слов, все состояния основные, выходная процедура, как и во всех алгебраических примерах, которые мы будем рассматривать, тривиальна, она не меняет объекта.

Так задаваемые исчисления называются теперь ассоциативными исчислениями (сам Туэ не употреблял этого термина), а определенная выше эквивалентность называется эквивалентностью в заданном ассоциативном исчислении (см. [Наг 77а]). Это отношение эквивалентности является конгруэнцией на полугруппе Б-слов, т.е. выдерживает умножение справа и слева на элементы этой полугруппы. Факторполугруппа по этому отношению называется полугруппой, заданной системой образующих Б и системой определяющих соотношений $A_j = B_j$, $j=1, \ldots, m$. Замечательный и простой факт состоит в следующем. В этой полугруппе равны только такие произведения образующих, которые равны во всех полугруппах, в которых выполнены соотношения $A_j = B_j$, $j=1, \ldots, m$. Более того, получившаяся конгруэнция на полугруппе Б-слов может быть порождена весьма простым исчислением. Выпишем исчисление, стремясь породить множество, состоящее из всех тех равенств $U=V$, для которых U и V конгруэнтны. Правила:

1. $U = U$ - допустимо для любого $U \in Б^*$
2. если $U=V$ допустимо и $U'=V'$ при некотором $j \in \{1, \ldots, m\}$ получается из $U=V$ заменой некоторого вхождения A_j на B_j или B_j на A_j, то $U' = V'$ - допустимо.

Остальные компоненты исчисления восстанавливаются без труда.

В дальнейшем конструкция Туэ обобщалась в двух направлениях. Первое направление начинается с отказа от симметричности исчисления, т.е. в переходе к исчислению со входом с правилом:

если U допустимое слово и при некотором $j \in \{1, \ldots, m\}$ слово V получается из U заменой какого-то вхождения A_j на B_j, то V допустимое.

Все остальное, как в системах Туэ. Получаются так называемые полусистемы Туэ, введенные в [Пост 47], частным случаем которых являются системы Туэ. Затем было введено разделение алфа-

вита Б на множество основных и вспомогательных символов, и основными считались только состояния, являющиеся словами в алфавите основных символов. Если к тому же перейти к рассмотрению исчислений без входа, но зато, дополнительно, с одним нольпосылочным правилом, то получится определение грамматик (см. [Гла 73], [Сто 80]).

Второе направление - это обобщение ситуации на более широкий класс алгебраических систем, чем полугруппы, и на более широкий класс соотношений, чем равенства произведений образующих. Но тут нам придется сперва позаботиться о терминологии.

Алгебраические системы: алгебры

Естественным источником задач и полем приложений теории алгоритмов оказывается алгебра. Подробнее об этих задачах и приложениях мы будем говорить в ч. П, § 1, сейчас же, толкуя об исчислениях, мы объясним, откуда исчисления берутся в алгебре. Начнем с чисто алгебраических определений. Сигнатурой называется множество символов, разбитое на два подмножества:

1) множество функциональных имен, для каждого из которых указано натуральное число - валентность (число аргументов),
2) множество предикатных имен, для каждого из которых также указана его валентность.

Имена валентности n называются также n-местными (функциональными, предикатными) именами. Функциональные имена валентности нуль называются также предметными именами. Сигнатуры, если не оговорено противное, будем предполагать конечными.

Алгебраической системой сигнатуры σ называется произвольное непустое множество M - носитель алгебраической системы, вместе с отображением (называемым интерпретацией), сопоставляющим с элементами сигнатуры их значения (в определяемой алгебраической системе): значением n-местного функционального имени служит n-местная операция на M (значение предметного имени - элемент M), значением n-местного предикатного имени служит n-местное отношение на M. Если в сигнатуре присутствуют только предикатные и предметные имена, алгебраическая система называется реляционной, если только функциональные (в том числе предметные) имена, алгебраическая система называется операционной. Операционные системы называются также

алгебрами.

Замкнутый терм (з.терм) данной сигнатуры σ определяется так: пусть f есть n-местное функциональное имя из σ, тогда

при n = 0, f - з.терм,

при n > 0, если $t_1, \ldots, t_n$ - з.термы, то и $f(t_1, \ldots, t_n)$ - з.терм.

Таким образом, з.термы - это некоторые конструктивные объекты, а именно специального вида слова в алфавите функциональных имен, двух скобок и запятой. Более естественно представлять з.термы в виде (Б,k)-деревьев, где Б - алфавит функциональных имен, k- максимальная валентность функционального имени (см. § 0).

Множество всех з.термов данной сигнатуры σ легко превратить в алгебраическую систему этой сигнатуры. Для этого нужно значением всякого n-местного функционального имени f из σ объявить:

при n = 0- само f,

при n>0 - отображение, перерабатывающее всякий набор $t_1, \ldots, t_n$ в $f(t_1, \ldots, t_n)$;

значениями всех предикатных имен объявим тождественно ложные отношения. Полученную алгебраическую систему назовем свободной алгебраической системой, порожденной сигнатурой σ. Таким образом, то, что обычно называется свободной алгеброй сигнатуры ρ с образующими $a_1, \ldots, a_l$, мы называем свободной алгеброй, порожденной сигнатурой $\sigma = \rho \cup \{a_1, \ldots, a_l\}$.

Понятным образом (индуктивно) для всякой сигнатуры σ, всякого з.терма t этой сигнатуры и всякой алгебраической системы B этой сигнатуры определено значение t в B. Пусть система B такова, что всякий элемент ее носителя является значением некоторого з.терма сигнатуры σ. Тогда B называется алгебраической системой, порожденной сигнатурой σ. Таким образом, произвольная группа с образующими $a_1, \ldots, a_l$ является - в нашей терминологии - алгебраической системой, порожденной сигнатурой $\{a_1, \ldots, a_l, \circ, {}^{-1}\}$, а также сигнатурой $\{e, a_1, \ldots, a_l, \circ, {}^{-1}\}$. Всякая алгебраическая система, порожденная сигнатурой σ, является гомоморфным образом свободной алгебраической системы, порожденной сигнатурой σ.

Равенства

Фиксируем некоторую сигнатуру σ. Равенствами сигнатуры σ будем называть выражения вида

$$t = s$$

где t, s - з.термы сигнатуры σ. Замкнутыми атомными формулами (з.а.формулами) сигнатуры σ будем называть равенства этой сигнатуры, а также все выражения вида $p(t_1,\ldots,t_n)$, где p есть n-местное предикатное имя, $t_1,\ldots,t_n$ суть з.термы сигнатуры σ. Пусть задана совокупность равенств S и з.а. формула F. Будем говорить, что F является (семантическим) следствием совокупности S, если во всякой алгебраической системе, порожденной сигнатурой σ, в которой выполнены все элементы S, выполнена также и F. Заметим, что здесь, как и дальше, наше понятие следствия расходится со стандартным. При стандартном понимании следствие определяется при помощи класса в с е х алгебраических систем данной сигнатуры. Для нас же, в данном контексте, важны именно системы, п о р о ж д е н н ы е этой сигнатурой, только они и принимаются в расчет при определении следствия. (Впрочем, именно в случае равенств наше понятие следствия разнообъемно стандартному, в аналогичной ситуации с квазитождествами дело будет не так.)

Обозначим через $C(\sigma,S)$ класс всех алгебраических систем, порожденных сигнатурой σ, в которых выполнены все равенства из S. По определению, в любой алгебраической системе, принадлежащей $C(\sigma, S)$, выполнены также и все следствия из S. Оказывается, в $C(\sigma, S)$ существует и единственна такая система $\mathcal{A}(\sigma,S)$, в которой среди з.а.формул выполнены т о л ь к о следствия из S ; все же остальные системы из $C(\sigma, S)$ являются гомоморфными образами $\mathcal{A}(\sigma,S)$. Система $\mathcal{A}(\sigma,S)$ называется алгебраической системой, порожденной сигнатурой σ и заданной совокупностью равенств S. Эта система, в свою очередь, является факторсистемой свободной алгебраической системы, порожденной сигнатурой σ, по следующему отношению конгруэнции: два элемента свободной системы конгруэнтны тогда и только тогда, когда равенство между ними является следствием совокупности S.

То, что обычно называют алгеброй сигнатуры ρ с образующими $a_1,\ldots,a_l$ и совокупностью определяющих соотношений S,

мы, в силу данных определений, называем алгеброй, порожденной сигнатурой $\rho \cup \{a_1,\dots,a_l\}$ и заданной совокупностью S.

Пусть теперь совокупность S конечна. Тогда алгебраическая система, заданная совокупностью S, часто называется конечноопределенной, мы также будем использовать этот термин. В этой ситуации множество всех семантических следствий из S (а значит и соответствующая конгруэнция) может быть порождено исчислением, очень просто получающимся из совокупности S. Правила этого исчисления таковы:

1) допустимо всякое равенство $t=t$,
2) если равенство $t=s$ допустимо, а равенство $v=u$ или $u=v$ принадлежит S, то равенство, получающееся из $t=s$ заменой некоторого вхождения u на v, допустимо.

Для полного задания исчисления нужно указать рабочую среду, правило выделения основных состояний, выходную процедуру. В нашем случае все они очевидны: рабочая среда - подходящий ансамбль, содержащий все равенства данной сигнатуры, все состояния основные, выходная процедура оставляет любой объект неизменным.

В описанной ситуации часто используется и другое исчисление (на этот раз исчисление со входом), связанное с совокупностью S. Рабочей средой его является какой-либо ансамбль, содержащий множество всех термов сигнатуры σ. Разрешительное правило его таково: если равенство $u=v$ или $v=u$ принадлежит S, то в допустимом элементе можно заменить вхождение u на v. Легко проверить, что термы t_1 и t_2 лежат в одном классе упомянутой в предыдущем абзаце конгруэнции тогда и только тогда, когда t_2 принадлежит множеству, получаемому применением рассмотренного исчисления к $\{t_1\}$.

Установленные зависимости между понятиями семантического следствия и выводимости в некотором исчислении являются иллюстрациями как к формулировке, так и к методу доказательства теоремы Гёделя о полноте.

Тождества и квазитождества

Итак, мы описали, что такое задание алгебраической системы конечной совокупностью равенств и в чем исчислительный смысл такого задания. Однако встречаются ситуации, когда алгебраические системы задаются иначе, скажем, совокупностью

тождеств. Например, свободная группа с двумя образующими, т.е. некоторая алгебра, порожденная сигнатурой $\{e,a,b,\circ,^{-1}\}$, задается совокупностью тождеств

$$(x \circ y) \circ z = x \circ (y \circ z)$$
$$x \circ e = x$$
$$e \circ x = x$$
$$x \circ (x^{-1}) = e \,, \quad (x^{-1}) \circ x = e$$

Тождества являются более общим видом соотношений, чем равенства; в них, наряду с предметными именами (в частности, символами образующих, если эти символы включены в сигнатуру) допускаются и предметные переменные, пробегающие носитель алгебраической системы.

Еще более общий чем тождества вид соотношений возникает, когда выполнение некоторого свойства требуется не от всех элементов алгебраической системы, а только от удовлетворяющих каким-то условиям. Например, некоторая упорядоченная группа сигнатуры $\{e,a,\circ,^{-1},\geq\}$ может быть задана так:

1. тождества группы, выписанные выше,
2. $x \geq x$,
3. $x \geq y \;\&\; y \geq z \Rightarrow x \geq z$,
4. $x \geq y \Rightarrow x \circ z \geq y \circ z$,
5. $a \geq e$.

Мальцеву принадлежит описание широкого класса алгебраических систем, для которых конструктивное задание посредством формул некоторого языка тесно связано со структурой алгебраической системы. Более точно, мальцевские соотношения (квазитождества), как мы далее увидим, представляют собой просто иначе записанные правила исчисления, позволяющего порождать конгруэнцию, определяющую нужную алгебру.

Сейчас мы, следуя [Маль 61], [Маль 70], дадим соответствующие определения. Пусть фиксирована конечная сигнатура σ. Выше, говоря о равенствах, мы определили замкнутые (т.е. не содержащие переменных) термы и замкнутые атомные формулы. Теперь мы введем более широкие классы объектов. Термы отличаются от з.термов только тем, что в их состав могут входить, таким же образом, как предметные имена, еще и переменные (специально подобранные комбинации символов) из некоторого бесконечно-

го списка. Атомные формулы отличаются от з.а.формул тем, что в их определении роль з.термов играют термы. Квазитождеством (в заданной сигнатуре σ) назовем всякое выражение вида

$$\bigwedge_{C\in\Phi} C \Rightarrow D ,$$

где Φ-конечное множество атомных формул, D - атомная формула. Конкретизацией квазитождества назовем всякий результат подстановки каких-нибудь з.термов вместо всех переменных в квазитождество. Заметим сразу же, что тождество, т.е. просто атомная формула D (в частности, равенство) во всякой алгебраической системе эквивалентно квазитождеству $x=x \Rightarrow D$. Говорят, что квазитождество выполнено в алгебраической системе, если в этой системе выполнена каждая его конкретизация. З.а.формула F называется (семантическим) следствием совокупности квазитождеств S, если во всякой алгебраической системе, порожденной сигнатурой σ, в которой выполнены все квазитождества из S, формула F также выполнена. Будем говорить, что алгебраическая система B задается совокупностью квазитождеств S, если для всякой з.а.формулы истинность ее в B эквивалентна тому, что эта формула является следствием совокупности S. В случае, когда S состоит из равенств (или эквивалентных им квазитождеств), это определение совпадает с данным выше определением алгебраической системы, заданной совокупностью равенств. Если совокупность S конечна, то задаваемая ею алгебраическая система называется конечнозаданной.

Оказывается, для всякой совокупности квазитождеств S существует и единственна порожденная сигнатурой σ алгебраическая система $\mathfrak{A}(\sigma,S)$, задаваемая этой совокупностью, причем всякая система, порожденная σ, в которой выполнены все квазитождества из S, является гомоморфным образом системы $\mathfrak{A}(\sigma,S)$.

Рассмотрим пример. Пусть совокупность S содержит тождества группы и еще некоторые дополнительные равенства в сигнатуре группы с какими-нибудь образующими. Тогда, в обычной терминологии, $\mathfrak{A}(\sigma,S)$ есть группа, заданная (или определенная) этими образующими и равенствами. Если число равенств конечно, мы, в соответствии с обычной терминологией, будем называть группу конечноопределенной. Аналогичная терминология используется для полугрупп, колец и т.п.

Пусть теперь совокупность S конечна, а сигнатура не содержит предикатных имен, т.е. алгебраические системы являются алгебрами. Мы опишем некоторое исчисление, позволяющее получать все следствия из S, и одновременно получим описание алгебры, задаваемой совокупностью S. Почти та же конструкция годится и в случае любых конечных сигнатур, но мы ее не приводим как более громоздкую.

Вот правила исчисления; в них мы пишем посылки над чертой, заключение под чертой и подразумеваем, что всякое правило начинается словами "для любых термов $p,q,r,s,\ldots$":

Правила, отвечающие определению конгруэнции:

1. $\dfrac{}{t = t}$

2. $\dfrac{t = s}{s = t}$

3. $\dfrac{t = s,\ s = r}{t = r}$

4. $\dfrac{t = s,\ u = v}{\text{результат замены в } t = s \text{ любого вхождения } u \text{ на } v}$

Правило, обеспечивающее выполнение квазитождеств из S: для всякой конкретизации $u_1 = v_1 \,\&\, \ldots\, u_k = v_k \to u_{k+1} = v_{k+1}$ квазитождества из S

$$\frac{u_1 = v_1, \ldots,\ u_k = v_k}{u_{k+1} = v_{k+1}}\ .$$

Обозначим через $\mathcal{A}$ свободную алгебру, порожденную сигнатурой σ. Выписанное выше исчисление задает бинарное отношение на $\mathcal{A}$: термы u и v находятся в этом отношении, если равенство $u=v$ допустимо. Это отношение, как легко видеть, является конгруэнцией, т.е. эквивалентностью, сохраняющейся при операциях из σ. Факторалгебра алгебры $\mathcal{A}$ по этой конгруэнции и оказывается алгеброй, заданной совокупностью S.

§ 4. ПРЕДСТАВИТЕЛЬНЫЕ ПОРОЖДАЮЩИЕ МОДЕЛИ

Открытие здесь состоит в самой возможности предъявить класс исчислений одновременно точно очерченный и представительный, т.е. содержащий исчисление, эквивалентное любому наперед заданному исчислению. (Чтобы быть более точными, мы должны были бы говорить не о представительной, а об Y-представи-

тельных порождающих моделях. Пусть Y- некоторый ансамбль; порождающая модель называется Y-представительной, если для любого исчисления, порождающего подмножество ансамбля Y, существует исчисление данной модели, порождающее то же подмножество.) Понятие порождающей модели возникает таким же образом, как и понятие вычислительной модели. Канонические системы Поста представляют собой самый ранний пример представительной модели. Однако канонические системы (см. [Пост 43], [Мас 64], [Мин 67, § 12.5 и § 13.2]) являются исчислениями с нелокальным преобразованием информации - в противоположность нормальным системам Поста (см. [Пост 43], [Марк 54, гл. VI, § 4], [Мас 64], [Мин 67, гл. 13]), которые также образуют представительную порождающую модель, но с локальным преобразованием информации. Еще один пример представительной модели - грамматики (см. [Гла 73], [Сто 80]). Данное в § 3 описание исчислений колмогоровского типа может также рассматриваться как некоторая представительная порождающая модель (ср. сказанное в § 2 о двух аспектах, в которых могут рассматриваться машины Колмогорова). Утверждение о представительности точно очерченного класса исчислений (т.е. о представительности соответствующей порождающей модели) составляет содержание тезиса Поста для этого класса или для этой модели. Названный тезис был впервые сформулирован - для нормальных систем - в [Пост 43]. Тезис Поста играет ту же роль для исчислений, что и тезис Чёрча для алгоритмов. Он, так же, как и тезис Чёрча, может служить основой для включения дескриптивной теории алгоритмов и исчислений в обычную теоретико-множественную математику (см. [Цей 64]).

Для исчислений, как и для алгоритмов, можно определить понятие формального задания и указать для каждой порождающей модели свой Универсальный Рецепт. Для фиксированной порождающей модели Универсальный Рецепт позволяет породить все пары, образованные формальным заданием некоторого исчисления рассматриваемой модели и произвольным выходом этого исчисления. После соответствующей кодировки Универсальный Рецепт превращается в универсальное исчисление (см. ниже § 14).

§ 5. ВЫЯСНЕНИЕ СВЯЗЕЙ МЕЖДУ АЛГОРИТМАМИ И ИСЧИСЛЕНИЯМИ

1) Для каждого алгоритма существует исчисление, порождающее область определения этого алгоритма. Более того, 2) для каждого алгоритма $\mathcal{A}$ можно указать исчисление, порождающее те и только те пары $\langle x, y \rangle$, для которых $\mathcal{A}(x) = y$. С другой стороны, 3) для каждого исчисления существует алгоритм, область определения которого совпадает с множеством, порождаемым исходным исчислением, и 4) каждое исчисление, порождающее униформное (= функциональное) множество пар $\langle x, y \rangle$, может быть преобразовано в алгоритм, переводящий каждый x в тот y, для которого пара $\langle x,y \rangle$ порождается исчислением. Далее, 5) для каждого алгоритма, распознающего принадлежность к какому-либо множеству, расположенному в некотором ансамбле, существует исчисление, порождающее это множество. Наконец, 6) каждое исчисление может быть заменено алгоритмом, результатами которого служат в точности те объекты, которые порождаются исходным исчислением. Более того, 7) при условии, что такие объекты вообще существуют (т.е. что порождаемое множество непусто), всегда можно добиться, чтобы областью определения получающегося алгоритма служил натуральный ряд.

Нам хотелось бы еще раз подчеркнуть, что понятия алгоритма и исчисления понимаются авторами в самом общем неформальном смысле. Многие теоремы теории алгоритмов, в частности, все теоремы этого параграфа, можно сформулировать и доказать, используя только интуитивное понимание, без ссылок на вычислительные или порождающие модели, хотя, конечно, они могут быть доказаны и для подходящих вычислительных моделей. Эта ситуация в некоторой степени типична: многие теоремы о множествах или натуральных числах можно сформулировать и доказать без обращения к каким-либо формальным (в частности, аксиоматическим) понятиям.

Попытаемся теперь выяснить глубинные причины, приводящие к наличию столь жестких связей между алгоритмами и исчислениями.

Начнем с того, что понятие алгоритма сводится к понятию исчисления. Подчеркнем, что здесь мы имеем в виду не сведение понятия вычислимости к понятиям породимости или перечисли –

мости, а сведение понятия произвольного алгоритмического процесса к понятию порождающего процесса. Сведение это имеет место в силу следующих соображений. В § 3 объяснялось, каким образом исчисление со входом применялось к множеству объектов подходящего ансамбля. Чтобы описать в терминах исчислений со входом алгоритм, нужно: применять это исчисление к одному входному элементу (а точнее, к одноэлементному входному множеству), обеспечить детерминированность такого процесса, т.е. однозначность каждого следующего шага, обеспечить однозначность момента остановки процесса.

Для исчисления со входом, сопоставляемого таким образом с произвольным алгоритмом, оказывается верным следующее: все правила этого исчисления однопосылочные, к основному состоянию никакое правило не применимо, ко вспомогательному состоянию применимо не более одного правила и применение правила дает однозначно определенный результат.

Итак, все алгоритмы могут трактоваться как специального вида исчисления со входом. Заметим здесь же, что так называемые "недетерминированные алгоритмы" также получают естественную трактовку в терминах исчислений со входом. Именно, недетерминированный алгоритм - это исчисление со входом $\mathcal{L}$, для которого выполнены все перечисленные выше условия на исчисление, ставящееся в соответствие алгоритму, со следующим исключением: ко вспомогательному состоянию может быть применимо любое число правил. Фиксируем некоторое основное состояние e и рассмотрим множество $\{x|\, e \in \mathcal{L}(\{x\})\}$. Это множество называется множеством, распознаваемым недетерминированным алгоритмом $\mathcal{L}$.

Таким образом, само понятие алгоритма оказывается возможным свести к понятию исчисления. Обратное сведение имеет место лишь в смысле сведения понятия породимого множества к понятию вычислимой функции. Процесс порождения объектов заданным исчислением разворачивается как бы в ветвящемся времени, проблема состоит в том, чтобы перейти от ветвящегося времени к последовательному, в ходе которого некоторый бесконечный алгоритмический процесс последовательно воспроизведет все объекты, породимые данным исчислением, и только их. Процесс этот можно представлять себе как последовательное добавление

к (первоначально пустому) списку элементов, относительно которых уже установлена их породимость, некоторых новых элементов. Этот процесс оказывается возможным в силу конечности числа правил всякого исчисления и возможности алгоритмически построить по данному правилу и списку посылок список всех (а их обязательно конечное число) результатов применения этого правила к этим посылкам. Располагая описанием этого процесса, нетрудно сконструировать, например, алгоритм, который перерабатывает всякое натуральное число n в n-ый элемент списка элементов, породимых рассматриваемым исчислением, ср. теорему 7) из перечня в начале настоящего параграфа.

В силу вышесказанного, тезисы Чёрча и Поста не независимы: тезис Чёрча для любой вычислительной модели влечет тезис Поста для некоторой порождающей модели и обратно, тезис Поста для любой порождающей модели влечет тезис Чёрча для некоторой вычислительной модели.

§ 6. ВРЕМЯ И ЁМКОСТЬ КАК СЛОЖНОСТИ ВЫЧИСЛЕНИЯ И ПОРОЖДЕНИЯ

Каждое реальное вычисление осуществляется

а) в физическом Времени - имея определенную длительность;

б) в физическом Пространстве - занимая определенное место.

Чтобы формализовать наши интуитивные представления о длительности и месте, мы должны прежде всего фиксировать некоторую вычислительную модель. Затем мы должны указать способ измерения длительности вычисления на этой модели и способ измерения места, им занимаемого. Эти меры длительности и места суть функции от входа (= исходного данного) вычисления. Они называются временем и ёмкостью. Следуя традиции, мы предполагаем, что значения времени и ёмкости являются натуральными числами.

При определении понятий вычислимой функции и породимого множества (см. ниже § 7), как и для всей основывающейся на этих понятиях дескриптивной теории, выбор вычислительной модели не играет существенной роли. Иначе обстоит дело в метрической теории - при определении сложности вычисления и порождения. Различные модели могут отражать существенно различные

аспекты реальных вычислительных и порождающих процессов и потому приводить к различным временным и ёмкостным функциям. В частности, в (абстрактных) машинах той или иной вычислительной модели могут выделяться входное устройство, служащее для считывания входа (= исходного данного) и выходное устройство, служащее для постепенного выписывания выхода (= результата). Если такие устройства выделены, место, занимаемое входом и выходом, обычно не учитывается при определении емкости. Таким образом, вычисление, скажем, тождественной функции $x \mapsto x$ может (при разумной организации вычислений) иметь нулевую или близкую к нулю емкость. Если же входное и выходное устройства не выделены или, даже при их выделении, размеры входа и выхода учитываются при подсчете емкости, вычисление тождественной функции не может иметь емкость меньшую, чем эти размеры.

Машины Тьюринга

Бóльшая часть всех полученных к настоящему времени результатов о времени и емкости относится к многоленточным машинам Тьюринга.

В понимании того, что такое даже обычная одноленточная машина Тьюринга, в литературе наблюдается разнобой. Сам Тьюринг в своей знаменитой статье [Тью 36] не дал достаточно точного описания своей машины. На это обратил внимание Пост в [Пост 47], предложивший свою, уже совершенно строгую, формулировку машины Тьюринга (не смешивать с машиной Поста, см. [Усп 80]). Пост признал, что его формулировка отличалась в деталях от (не сформулированного явно) понимания Тьюринга; в частности, у машины Тьюринга в формулировке Поста лента бесконечна в обе стороны. В [Пост 47, приложение] Пост пишет: "Принимая во внимание частые упоминания Тьюрингом начала ленты, а также то, каким способом его универсальная машина совершает движение влево, мы догадываемся, что лента - в отличие от нашей ленты - представляет собою односторонне бесконечную штуку ⟨affair⟩ , простирающуюся вправо от начальной клетки." Имеются и другие, менее существенные расхождения, присущие различным формулировкам. Так, в указанной формулировке Поста (она же принята и в монографии [Маль 65]) за один шаг головка может либо напечатать один символ, либо сдвинуться на одну клетку, но не может совершить оба действия сразу; в формули-

ровке Клини из [Кли 52, § 67], следующей в этом пункте первоначальной формулировке Тьюринга, оба действия (движение и печать) совмещаются в одном шаге. В дальнейшем, говоря об <u>одноленточной машине Тьюринга</u>, мы будем иметь в виду машину из [Кли 52, § 67], т.е. машину с одной потенциально бесконечной в обе стороны лентой и не имеющую входных и выходных устройств; исходное данное записывается на этой единственной ленте, и на ней же образуется результат.

Во времена Тьюринга и Поста абстрактные машины были только одноленточными. Как уже отмечалось, теперь при исследовании сложности вычислений основным объектом являются многоленточные машины Тьюринга (см. [Усп 60, § 14, п. 3], [Хоп Уль 69, § 6.5 и § 10.2], [Ахо Хоп Уль 74, § 1.6]). В каждой из таких машин имеется одна или несколько рабочих лент и по одной головке для каждой ленты; эти головки могут двигаться в обе стороны, читать и писать. Сами же ленты могут быть либо двусторонне бесконечными, как в [Усп 60] и в [Хоп Уль 69, § 6.5], или же бесконечными только вправо, как в [Хоп Уль 69, § 10.2] и в [Ахо Хоп Уль 74]. Кроме того, может иметься, а может и не иметься входное устройство в виде входной ленты с читающей (только) головкой. Так, в [Хоп Уль 69, § 10.2] при обсуждении ёмкостной сложности рассматривается машина с входной лентой ("машина рисунка 10.1"), а при обсуждении временной сложности – машина без входной ленты ("машина рисунка 10.2"). Головка на входной ленте может иметь разрешенным как двустороннее движение (так в [Хоп Уль 69] и в [Ахо Хоп Уль 74]), так и одностороннее (только вправо) движение (так в [Шёнх 80]); в последнем случае говорят о безвозвратной входной ленте. Наконец, можно требовать наличия выходной ленты, головка на которой – только пишущая и может двигаться только вправо.

Ввиду сказанного, следует договориться, что понимается в дальнейшем под многоленточной машиной Тьюринга. За исключением особо оговоренных случаев, мы принимаем, что <u>многоленточная машина Тьюринга</u> имеет бесконечные только вправо рабочие и выходную ленты и входную ленту с двусторонне движущейся головкой. (Таким образом, многоленточная машина Тьюринга с одной рабочей лентой – это не то же самое, что одноленточная машина Тьюринга!)

Перейдем теперь к изложению некоторых понятий и результатов, относящихся к определенным только что многоленточным машинам Тьюринга. Затем мы обсудим возможность распространения этих определений и результатов на другие вычислительные модели.

Итак, пусть задана некоторая многоленточная машина Тьюринга. Временем ее работы (на данном входе) называется число шагов, выполняемых машиной до получения результата. Ёмкостью работы машины (на данном входе) называется максимальная длина использованного участка на рабочих лентах (максимум берется по всем рабочим лентам и по всем моментам вычисления). Если вычисление не заканчивается, как время, так и емкость считаются бесконечными.

Первый же вопрос, возникающий в связи с этими определениями, таков: существуют ли какие-нибудь связи между временем и емкостью? Эти связи легко найти: ясно, что никакое короткое вычисление не может использовать слишком большую внутреннюю память и что никакое закончившееся вычисление с небольшой емкостью и небольшим входом не может быть продолжительным. Именно, для любой многоленточной машины Тьюринга существует такое число k, что ёмкость S и время T (при работе на любом входе) удовлетворяют неравенствам $S \le k \cdot T$ и (если l есть длина входа и $T < \infty$) $T \le k^{S+l}$. Нетривиальная теорема, связывающая время и ёмкость, такова: для любой функции T каждый предикат (= функция с двумя значениями), вычисляемый на какой-нибудь многоленточной машине Тьюринга за время, не превосходящее $T(x) \log T(x)$ при исходном данном x, может быть вычислен на некоторой (возможно, другой!) машине с ёмкостью, не превосходящей $T(x)$ (см. [Хоп Пауль Вэл 77]). Отметим важную разницу между этой теоремой и двумя предыдущими оценками: оценки связывают время и емкость вычисления на одной и той же машине, а теорема говорит лишь, что если время работы некоторой машины не больше $T(x) \log T(x)$, то существует другая машина, вычисляющая тот же предикат с емкостью $T(x)$ (хотя, быть может, и с большим временем!).

Интересен также вопрос о том, каким образом изменение числа лент влияет на быстродействие машин. Очевидно, что любая функция, вычислимая на k-ленточной машине за время $T(x)$,

вычислима и на m-ленточной машине при $m > k$ за то же время (нужно просто не использовать лишние ленты). То же самое можно сказать, конечно, и про емкость. Таким образом, можно сказать, что вычислительная сила машин не убывает с увеличением числа лент. Оказывается также, что различия во времени вычисления на машинах с разным числом лент не так уж велики: любая функция, вычисляемая на какой-либо многоленточной машине за время, не превосходящее $T(x)$, может быть вычислена на машине с одной рабочей лентой за время, не превосходящее $c \cdot T(x)^2$, где c - некоторая константа. С ёмкостью дело обстоит еще проще: всякая функция, вычисляемая на многоленточной машине с ёмкостью, не превосходящей $S(x)$, может быть вычислена на машине с одной рабочей лентой с ёмкостью, не превосходящей $c \cdot S(x)$, где c - некоторая константа.

Время

Обсудим теперь возможность определения понятий времени и ёмкости вычислений для других вычислительных моделей. Начнем с (более простого) понятия времени. Процесс вычисления во всех известных вычислительных моделях состоит из отдельных шагов. Самый простой ответ на вопрос о том, что такое время, таков: время есть число шагов в процессе вычисления. Такая характеристика вычисления была впервые исследована Цейтиным (см. [Яновс 59, с. 44-45]). Именно, он рассматривал число (нелокальных) шагов работы нормального алгорифма. Исследования Цейтина, доложенные им в ноябре 1956 г. на семинаре П.С.Новикова и С.А.Яновской в Московском университете, были едва ли не самыми первыми исследованиями по метрической теории алгоритмов вообще.

Соответствует ли число шагов интуитивному понятию длительности вычисления? С определенной точки зрения - нет. Действительно, в практических вычислениях различные шаги имеют разную длительность. Кроме того, при теоретических рассмотрениях такой способ измерения времени (как числа шагов) приводит к следующему нежелательному эффекту: мы можем перестроить вычисление, объединяя некоторые последовательные шаги в один макрошаг, и получить "новое" вычисление, которое на самом деле является старым, но имеет меньшее число шагов. Этот эффект используется в так называемой теореме о линейном ускорении,

см. следующий параграф. Более точно было бы определять время как сумму длительностей шагов. Такая точка зрения используется, например, при изучении вычислительных моделей с нелокальным преобразованием информации, в частности, машин с произвольным доступом к памяти, см. [Ахо Хоп Уль 74, гл. 1]. В случае моделей с локальным преобразованием информации разница между обоими определениями времени - как числа шагов и как суммы их длительностей - не так уж велика: для фиксированного алгоритма они, очевидно, совпадают с точностью до ограниченного отделенного от нуля множителя. Именно с этой точностью и формулируется большинство теорем о времени вычисления (среди упоминающихся в этом обзоре теорем лишь теорема о линейном ускорении, см. § 7, представляет собой исключение). Определив время вычисления для различных вычислительных моделей, естественно попытаться сравнить вычисления одной и той же функции на разных вычислительных моделях. (Для машин Тьюринга с разным числом лент такое сравнение проводилось выше в этом параграфе.) Приведем еще несколько результатов подобного рода.

(1) Всякая функция, вычислимая на многоленточной машине Тьюринга за время, не большее, чем $T(x)$, может быть вычислена машиной Колмогорова за время, не большее, чем $c \cdot T(x)$, где c - некоторая константа.

(2) Более того, всякая функция, вычислимая на машине Тьюринга с произвольным числом рабочих "лент" произвольной размерности за время, не превосходящее $T(x)$, может быть вычислена машиной Колмогорова за время, не превосходящее $c \cdot T(x)$, где c - некоторая константа.

(3) Всякая словарная функция, вычислимая на машине Колмогорова за время, не превосходящее $T(x)$, причем $T(x)$ не меньше длины входа x, может быть вычислена на многоленточной машине Тьюринга с одной (одномерной) рабочей лентой за время, не превосходящее $c \cdot T(x)^{2,5}$, где c - некоторая константа.

Результат (3) вытекает, в частности, из более тонкой оценки $T(x)^2 \log^2 T(x)$, имеющейся в [Сли 81, гл. 3, § 1, п. 3].

Результат (2) является усилением (почти тривиального) результата (1). Другой результат, близкий к результату (2), свя-

зывающий машины Тьюринга с алгоритмами колмогоровского типа – теорема Шёнхаге о возможности моделирования в реальное время (определение см. в добавлении к настоящему параграфу) всякой машины Тьюринга с безвозвратной входной лентой и многомерной памятью подходящей машиной Шёнхаге, доказан в [Шёнх 80]. Как отмечено в [Сли 81, гл. 3, § 1, п. 1], аналогичный результат может быть получен и для моделирования машин Тьюринга с безвозвратной входной лентой и многомерной памятью машинами Колмогорова – Успенского. Из результатов работы [Гри 76] вытекает невозможность обратных моделирований.

Ёмкость

Перейдем теперь к обсуждению различных способов определения емкости вычисления. Ситуация здесь, пожалуй, еще более сложная, чем с понятием времени. Но первое решение при определении емкости принимается однозначно: емкость (данного вычисления) – это максимальный размер памяти, используемой при вычислении. Это содержательное представление порождает два вопроса:

(1) что нужно считать памятью, используемой в данный момент вычисления?

(2) как измерить эту память? что есть ее размер?

В работах по сложности вычислений встречаются два различных ответа на первый (более простой) вопрос. Простейший ответ состоит в том, что памятью считается все состояние вычисления. Для таких вычислительных моделей, как одноленточные машины Тьюринга (без входных и выходных лент) или алгоритмы Колмогорова (опять-таки без входных и выходных лент), этот ответ – единственно возможный. Однако для реальных вычислений часто представляет интерес оценка размера только рабочей памяти. Этому соображению соответствует выделение рабочей памяти, входных и выходных лент, как это сделано в приведенном выше определении емкости для многоленточных машин Тьюринга. Если вычислительная модель предусматривает такие ленты (или, гипотетически, какие-либо другие устройства ввода и вывода), то при вычислении емкости естественно учитывать только рабочую память, считая, что только она представляет собою память, используемую при вычислении.

Перейдем теперь ко второму, более сложному вопросу – о

способах измерения памяти. Мы обнаруживаем, что даже в простейших случаях возникает много разных способов определения размера памяти, и трудно решить, какие лучше, какие хуже. Эта множественность возможных решений отражает существо дела. Проиллюстрируем нашу мысль следующим примером. Предположим, что мы хотим определить, что такое размер автоцистерны, и колеблемся, на каком из параметров - длине, ширине, высоте или вместимости - нам остановиться. Скоро мы обнаруживаем, что каждая из этих характеристик существенна в надлежащем контексте: высота определяет возможность проезда под мостами, длина - возможность развернуться в тупике и т.д. Кажется более правильным вообще считать размер не числом, а вектором из нескольких компонент, и признать, что любая подгонка его под число - искусственна.

Аналогично нашему автомобильному примеру обстоит дело и с вычислительными моделями. Например, для машин Тьюринга с одной рабочей лентой неясно, должны ли мы учитывать только длину записи на ленте или еще и увеличение объема за счет фиксации положения рабочей головки. Явное указание положения головки можно осуществить, задав двоичное слово с длиной,равной примерно логарифму длины слова на ленте; нужно ли прибавлять это число к длине записи на ленте? Другая неясность связана с тем, должен ли объем зависеть от алфавита: одинаков ли, например, размер десятибуквенного слова в двух- и трехбуквенных алфавитах? Ведь очевидно, что можно "дешевым способом" сократить длину каждого слова в миллионы раз, объявив новыми буквами длинные комбинации прежних букв.

Если мы теперь перейдем к случаю многих лент, то вопросы продолжают множиться: нужно ли складывать длины записей на лентах или следует умножать максимум длины этих записей на их количество, или, может быть, нужно учитывать еще разницу в направлениях сдвига головок на лентах от начального положения и т.д. При переходе к многомерной памяти возникают еще большие неясности, связанные, в частности, с отсутствием естественной реализации такой памяти в трехмерном пространстве. В случае неориентированных колмогоровских комплексов вопрос об объеме физической реализации комплексов изучался в работе [Колм Бар 65]. За счет несущественного ослабления, результаты из [Колм

Бар 65] могут быть изложены в следующей наглядной форме. Фиксируем некоторый ансамбль колмогоровских комплексов. Предположим теперь, что вершины комплекса реализуются шарами единичного диаметра, а ребра - гибкими трубками тоже фиксированного и притом достаточно малого диаметра. Тогда можно так выбрать диаметр труб, что для некоторых положительных действительных чисел c и d выполнено следующее: 1) всякий комплекс с n вершинами может быть реализован внутри сферы радиуса $c\sqrt{n}$; 2) объем любой реализации почти всякого комплекса с n вершинами не меньше $dn\sqrt{n}$ ("почти всякого" означает, что отношение количества комплексов с этим свойством к количеству всех комплексов с n вершинами стремится к единице при стремлении n к бесконечности).

Из сказанного видно, что разумный выбор способа подсчета размера используемой памяти в большинстве случаев - задача, не имеющая однозначного решения (а порой, быть может, и вовсе неразрешимая). Некоторым утешением может служить тот факт, что различные способы определения понятия размера для одного класса объектов дают близкие по значениям емкостные функции. (Например, для многоленточных машин с фиксированным числом лент все упоминавшиеся способы подсчета размера дают функции, связанные отношением $\asymp$.)

Практически приходится выбирать какую-нибудь одну числовую характеристику - например, длину слова (без учета мощности алфавита) или число вершин комплекса (без учета числа ребер и алфавита разметки) или максимальную длину записей на рабочих лентах машины (без учета записей на других лентах и положения головок) - в качестве суррогата "истинного размера". Первой из таких суррогатных характеристик, встречающихся в литературе, (хотя и не для какой-либо вычислительной модели, а в несколько иной ситуации) была введённая в [Тра 56] длина отрезка вычислимости функции, заданной рекурсивной схемой; под отрезком вычислимости понимался отрезок натурального ряда, содержащий все те натуральные числа, которые могут встретиться в процессе вычисления функции по заданной схеме при фиксированных значениях аргументов.

Существует ли вообще "истинный размер", или же его поиски подобны поискам философского камня? Быть может, имеется боль-

шая совокупность "истинных размеров" – но тогда какова она? Ответы на эти вопросы неясны. Быть может, полезно было бы начать с попыток ввести понятие абстрактного размера на различных ансамблях (а возможно, и на их подмножествах).

Норма

Мы попытаемся сейчас высказать некоторые – далеко не исчерпывающие – соображения по этому поводу. Чтобы подчеркнуть, что мы пытаемся отразить лишь некоторые аспекты интуитивного понятия размера, вместо слова "размер" будем говорить "норма". В проводимом ниже построении предполагается, что норма определена на всем ансамбле. Это предположение вносит довольно существенное ограничение. В самом деле, рассмотрим, например, трехленточные машины Тьюринга с фиксированными алфавитами (рабочим и внутренних состояний). Всевозможные полные состояния рабочей памяти таких машин легко вкладываются в подходящий колмогоровский ансамбль W и образуют некоторое подмножество E этого ансамбля. Если понимать размер (полного) состояния рабочей памяти как максимальную из длин записанных на ленте слов, то, очевидно, такая норма е с т е с т в е н н ы м о б р а з о м определена лишь на E, и всякое ее распространение на все W носит довольно искусственный характер. Однако авторы еще не чувствуют себя готовыми рассматривать нормы, определенные не на всем ансамбле (хотя понимают, что это было бы разумно).

Итак, что же такое ансамбль с нормой, или нормированный ансамбль?

Пусть X – некоторый ансамбль, а n – некоторая всюду определенная функция из X в $\mathbb{N}$. Функция n будет называться нормой на X, если она обладает следующими свойствами:

(N1) Число элементов x, для которых $n(x) \leq m$, равно 2^m с точностью до ограниченного отделенного от нуля множителя.

(N2) Существует алгоритм, дающий по любому числу m список всех элементов ансамбля X, для которых $n(x) \leq m$.

Нормированный ансамбль есть пара $\langle X, n\rangle$, где X – ансамбль, а n – норма на X.

Примеры нормированных ансамблей

Пусть дан словарный ансамбль (напомним, что в алфавите словарного ансамбля должны быть по крайней мере две разные

буквы). В качестве одной из норм на этом ансамбле можно рассматривать функцию, пропорциональную длине слова; при этом коэффициент пропорциональности следует выбрать с таким расчетом, чтобы выполнялось требование (НТ). Более точно, пусть алфавит Б состоит из k букв, $k \geq 2$. Определим функцию n, положив

n(x) = целая часть числа [(длина слова x)$\cdot \log_2 k$].

Легко проверить, что определенная таким образом на ансамбле слов алфавита Б функция, действительно будет нормой.

В § 17 нам понадобится рассматривать $\mathbb{N}$ как нормированный ансамбль. Снабдим его такой нормой: норма числа x равна целой части величины $\log_2(x+1)$. (Нетрудно проверить, что свойства Н1 и Н2 действительно выполнены.)

Ограниченно-искажающие отображения и изоморфизмы

Пусть даны два нормированных ансамбля $\langle X_1, n_1\rangle$ и $\langle X_2, n_2\rangle$. Мы будем говорить, что отображение $f: X_1 \to X_2$ является ограниченно-искажающим, если выполнено условие $n_2(f(x_1)) \overset{+}{<} n_1(x_1)$. Взаимно-однозначное соответствие между ансамблями $\langle X_1, n_1\rangle$ и $\langle X_2, n_2\rangle$, осуществляемое (в обе стороны) вычислимыми ограниченно-искажающими отображениями, мы будем называть изоморфизмом нормированных ансамблей. Нетрудно доказать, что все нормированные ансамбли изоморфны в этом смысле.

Дополнительные требования к нормам

Ясно, что определение нормы оставляет довольно большой произвол и что среди норм имеются функции, мало отражающие наше интуитивное представление о размере. Поэтому может оказаться осмысленным рассматривать не все нормы, а лишь те, которые обладают некоторыми дополнительными свойствами. Мы сейчас приведем несколько примеров дополнительных свойств, которые, быть может, имеет смысл требовать от нормы на ансамбле (Б, k)-комплексов или колмогоровских комплексов.

(Д1) Непрерывность: применение одного локального действия должно увеличивать норму комплекса не более чем на фиксированное число, зависящее от операции, но не от комплекса.

(Д2) Связь с числом вершин: число вершин комплекса не должно превосходить нормы, умноженной на некоторую константу, не зависящую от комплекса.

Существуют нормы, удовлетворяющие требованиям Д1 и Д2. Для построения такой нормы достаточно выбрать подходящее инициальное (однопосылочное) исчисление, расположить все комплексы в порядке возрастания минимального числа шагов, необходимого для их порождения (ср. со сложностью порождения, см. конец настоящего параграфа), и считать нормой комплекса логарифм его порядкового номера.

Нетрудно доказать, что если норма на колмогоровских комплексах удовлетворяет требованиям Д1 и Д2, то норма слов (рассматриваемых как частный случай комплексов) отличается от их длины не более чем на мультипликативную константу, и норма колмогоровского комплекса (с точностью до ограниченного отделенного от нуля множителя) находится между числом вершин и квадратом числа вершин. (Более тонкий анализ, использующий результаты [Колм Бар 65], позволяет заменить квадрат числа вершин на число вершин в степени полтора.)

Сложности порождения

Подобно соответствующим понятиям для алгоритмов, понятия времени и емкости (порождения) можно пытаться определять и для исчислений - считая (для однопосылочных исчислений), что время порождения (при фиксированном выводе) есть число шагов вывода, а емкость - максимальный размер встречающихся в этом выводе объектов (см. [Гла 73, гл. 2 и гл. 7]). Можно считать, что числовые значения времени и емкости сами порождаются в ходе вывода. Дополнительные эффекты возникают из-за того, что может быть несколько способов порождения одного и того же объекта. За сложность естественно принять минимум по всем способам порождения.

Эффективные алгоритмы

Наконец, важную область теории сложности образует построение конкретных эффективных алгоритмов для решения частных задач и доказательства того, что эти алгоритмы имеют заданную оценку сложности (т.е. действительно эффективны). Эта область относится к прикладной теории алгоритмов (см. ч. П, § 8).

ДОБАВЛЕНИЕ К § 6. МОДЕЛИРОВАНИЕ В РЕАЛЬНОЕ ВРЕМЯ

Приведем определение моделирования в реальное время - в

той форме, как это определение приведено в [Шёнх 80]. Наряду с машинами Шёнхаге, описанными в § 2, будем рассматривать машины Тьюринга с многомерной памятью. Предполагается, что каждая такая машина имеет входную и выходную (одномерные) ленты, движение головок по этим лентам - о д н о с т о р о н н е е , со входной ленты возможно только чтение, на выходную - только запись. Рабочая память может содержать произвольное фиксированное число массивов фиксированной размерности (лент, плоскостей и т.д.), головки по этим массивам способны передвигаться за один шаг на одну клетку в любом направлении. Во всяком вычислении такой машины Тьюринга, так же как и во всяком вычислении машины Шёнхаге, выделяются моменты сдвига входной ленты и моменты печати (очередного символа на выходную ленту). Пусть теперь $\mathcal{A}$ и $\mathcal{B}$ - две машины со входным и выходным алфавитом $\{0,1\}$, каждая из которых может быть машиной Тьюринга с многомерной памятью и безвозвратной входной лентой или машиной Шёнхаге (напомним, что и ее входная лента безвозвратна). Будем, в соответствии с [Шёнх 80], говорить, что машина $\mathcal{A}$ в реальное время моделирует машину $\mathcal{B}$, если выполнено следующее. Для всякого исходного данного x, на котором $\mathcal{B}$ кончает работу, пусть $t_0 < t_1 < \ldots < t_p$ такая последовательность моментов времени, что $t_0 = 0$, t_p - момент окончания работы машины, а $t_1, \ldots, t_{p-1}$ - все моменты времени, в которые машина $\mathcal{B}$ производила сдвиг входной ленты или запись на выходную. Тогда должны существовать такие моменты времени $\tau_0 < \tau_1 < \ldots < \tau_p$, что $\tau_0 = 0$, τ_p - момент окончания работы машины $\mathcal{A}$ с аргументом x и при всяком $i = 1,2,\ldots,p$:

1) при вычислении с исходным данным x часть этого x, прочитанная машиной $\mathcal{A}$ к моменту τ_i, совпадает с частью, прочитанной машиной $\mathcal{B}$ к моменту t_i ;

2) при вычислении с исходным данным x часть результата, напечатанная машиной $\mathcal{A}$ к моменту τ_i, совпадает с частью результата, напечатанного машиной $\mathcal{B}$ к моменту t_i;

3) для некоторого c, зависящего от $\mathcal{A}$ и $\mathcal{B}$, но не от x, имеет место

$$\tau_i - \tau_{i-1} \leq c(t_i - t_{i-1}).$$

Из этих условий очевидно следует, что машина $\mathcal{A}$ вычисляет ту же функцию, что и машина $\mathcal{B}$, или некоторое продолжение

этой функции; при этом время вычисления для машины $\mathcal{A}$ не превосходит времени вычисления $\mathcal{L}$, умноженного на константу c.

§ 7. ВЫЧИСЛИМЫЕ ФУНКЦИИ И ПОРОДИМЫЕ МНОЖЕСТВА; ПЕРЕЧИСЛИМЫЕ МНОЖЕСТВА; РАЗРЕШИМЫЕ МНОЖЕСТВА

Вычислимая функция - это функция, которая вычисляется каким-либо алгоритмом. Слово "вычисляется" понимается, согласно § 1, следующим образом: при применении к какому-нибудь входу вычисляющий алгоритм должен не только давать результат, совпадающий со значением функции на этом входе, коль скоро такое значение существует, но и не давать никакого результата вообще, коль скоро функция не определена на данном входе. Пусть A и B - подмножества соответствующих ансамблей; посредством $\mathrm{Com}(A, B)$ будет обозначаться класс всех вычислимых функций из A в B. Таким образом, $\mathrm{Com}(A, B) \subset \mathcal{F}(A,B)$. Для произвольных ансамблей X, Y и X - Y-представительной модели можно, конечно, формально определить $\mathrm{Com}(X, Y)$ как класс всех функций из $\mathcal{F}(X, Y)$, вычислимых на этой модели.

Породимое множество - это множество, которое порождается каким-либо исчислением (опять-таки в том смысле, что порождаются все элементы множества и не порождается ничего лишнего), Каждый ансамбль породим. Понятие породимого множества ⟨generable set⟩ ввел и изучал - в качестве фундаментального понятия логики и математики - Пост (см. [Пост 44]); он, впрочем, пользовался термином "порожденное множество" ⟨generated set⟩.

Пусть A - подмножество некоторого ансамбля. Класс всех породимых подмножеств множества A обозначается $\mathrm{Gen}(A)$. Таким образом, $\mathrm{Gen}(A) \subset 2^A$. Для каждой Y-представительной порождающие модели класс $\mathrm{Gen}(Y)$ можно определить формально как класс всех множеств из 2^Y, которые могут быть порождены на этой модели.

Множество называется разрешимым, или распознаваемым, если оно содержится в некотором ансамбле и для него существует разрешающий алгоритм. Алгоритм $\mathcal{A}$ называется разрешающим алгоритмом для подмножества A ансамбля X, если множество допустимых входов для $\mathcal{A}$ совпадает с X и $\mathcal{A}$ отвечает на все вопросы вида "принадлежит ли $x \in X$ множеству A?" Проблема отыскания такого

алгоритма называется проблемой разрешения для A (см. ч. II, § 1). Таким образом, множество разрешимо тогда и только тогда, когда его проблема разрешения решима (т.е. может быть решена, имеет решение).

Рассмотрение понятий породимого и разрешимого множества способствует значительному прояснению центральных понятий и результатов математической логики. Так, важнейшее требование, предъявляемое к любой разумной формализации понятия доказательства, состоит в том, чтобы множество всех доказательств данной логистической системы было разрешимым множеством. Теоремы Гёделя о полноте и неполноте утверждают, по существу, породимость одного множества формул и непородимость другого.

Понятия вычислимой функции, породимого и разрешимого множества тесно связаны друг с другом. Одну из главных таких связей выражает критерий разрешимости множества: множество A, расположенное в ансамбле W, разрешимо тогда и только тогда, когда $A \in Gen(W)$ и $W \setminus A \in Gen(W)$. Очевидно, что разрешимость множества эквивалентна также вычислимости его характеристической функции. Другие связи являются простыми следствиями указанных в § 5 соотношений между алгоритмами и исчислениями. Так, функция вычислима тогда и только тогда, когда она, рассматриваемая как множество пар, породима. Поэтому породимость может быть использована для определения понятия вычислимой функции. Замечательный исторический факт состоит в том, что так и обстояло дело с первоначальными определениями этого понятия: первые (хронологически) два варианта формального определения понятия вычислимой функции как раз и заключались в отождествлении (предложенном Чёрчем, см. [Чёрч 36]) вычислимых функций с функциями, порождаемыми (как множество пар) исчислением специального вида, а именно, исчислением λ-конверсии Чёрча - Клини в одном варианте и исчислением Эрбрана - Гёделя в другом варианте. В свою очередь, породимые множества могут быть определены:

(1) как области определения вычислимых функций;

(2) как области значений вычислимых функций;

(3) как множества, являющиеся пустыми или областями значений всюду определенных вычислимых функций.

Это позволяет при изложении теории алгоритмов вообще обходить-

ся без понятия исчисления (так делает, например, Роджерс в [Родж 67]).

Любые два бесконечные породимые множества W и W' изоморфны - существует одно-однозначное вычислимое отображение из W на W'; вычислимость этого отображения влечет вычислимость обратного отображения. При таком изоморфизме каждое породимое подмножество W соответствует породимому подмножеству W'. Это соответствие индуцирует очевидное взаимнооднозначное соответствие между Gen(W) и Gen(W') . Пусть теперь X изоморфно X', а Y изоморфно Y'; согласно этим изоморфизмам, каждая вычислимая функция из X в Y соответствует вычислимой функции из X' в Y', так что между Com(X, Y) и Com(X', Y') также возникает естественное взаимно однозначное соответствие. Следовательно, при изучении породимых множеств и вычислимых функций можно фиксировать специальные породимые множества W, X и Y и рассматривать только подмножества W и функции из X в Y . Часто $\mathbb{N}^s$ берется в качестве X и W , а $\mathbb{N}$ берется в качестве Y . При таком подходе предметом теории вычислимых функций оказывается семейство вычислимых функций типа $\mathbb{N}^s \to \mathbb{N}$ (для всех s). Каждая функция, принадлежащая $\bigcup_{s=0}^{\infty} \mathcal{F}(\mathbb{N}^s, \mathbb{N})$, называется числовой. Изучение вычислимости числовых функций играет центральную роль в общей теории алгоритмов. Сходным образом, при рассмотрении породимых множеств можно ограничиться множествами из $\bigcup_{s=0}^{\infty} 2^{\mathbb{N}^s}$; эти множества также называются числовыми.

Поразительно, что класс всех вычислимых (в интуитивном смысле) числовых функций допускает точное математическое определение, и даже много таких определений. А.П.Ершов в [ЕршА 82а] классифицирует эти определения на алгоритмические, логические, функциональные и арифметические и пишет: "Сталкиваясь с разнообразными определениями вычислимости, мы обнаруживаем, что эти определения не объясняют нам, почему они оказываются эквивалентными."

В § 1 мы, по существу, занимались алгоритмическими определениями, в § 8 - займемся функциональным . Каждое, вообще, формальное определение выделяет среди всех функций данного вида некоторый подкласс - например, среди всех числовых функций выделяются частичнорекурсивные. Утверждение, что выделен-

ный посредством рассматриваемого определения подкласс совпадает с подклассом всех вычислимых функций данного вида, образует тезис Чёрча (или тезис Чёрча - Тьюринга - Поста - Клини, см. § 2) для данного определения. В [Пост 36] Пост назвал утверждение о совпадении двух классов функций (рекурсивных и вычислимых) "рабочей гипотезой" и добавил: "В действительности работа, проделанная Чёрчем и другими, привела к тому, что это совпадение уже переросло стадию рабочей гипотезы. Однако замаскировать это совпадение, включив его в определение, значило бы скрыть факт фундаментального открытия, касающегося ограниченности способности Homo Sapiens к математизации, и заслонить от нас необходимость постоянной проверки этого совпадения."

Перечислимое множество - это либо множество значений всюду определенной вычислимой функции натурального аргумента, либо пустое множество. В силу утверждения 7) из § 5, каждое породимое множество перечислимо. Таким образом, обе теоремы Гёделя, отмеченные выше, можно сформулировать в терминах перечислимости и неперечислимости соответствующих множеств. По причинам, отмеченным выше, изучение перечислимых ч и с л о в ы х множеств представляет специальный интерес.

Рассмотрим теперь метрические аспекты введенных в данном параграфе понятий. Все вычислимые функции можно классифицировать по "трудности" их вычисления (а все породимые множества - по "трудности" их порождения). Стремление к такой классификации было одним из основных мотивов введения понятий временной и емкостной сложности. Для заданной вычислительной модели и заданной "верхней оценки сложности" один класс такой классификации образуют все функции или предикаты, для которых существует алгоритм их вычисления со сложностью, не превосходящей данной оценки. В качестве оценок, как правило, рассматриваются функции, зависящие только от длины слова-аргумента; в частности, цитируемые далее теоремы были доказены именно для таких функций (тем не менее, многие из них справедливы и без этого предположения). Мы не видим смысла в том, чтобы заранее суживать таким образом класс оценок, см. также § 17. Длину слова X в настоящем параграфе мы обозначаем l(x).

Все определения и теоремы этого параграфа относятся к времени и емкости вычислений на многоленточных машинах Тьюринга, определение которых давалось в § 6. Лишь в самом конце, обсуждая классы $\mathcal{P}$ и $\mathcal{NP}$, мы упомянем некоторые другие вычислительные и порождающие модели.

Прежде чем сформулировать теоремы о времени и ёмкости вычислений, дадим некоторые определения. В качестве верхних оценок мы будем по большей части рассматривать конструируемые (по ёмкости или времени) функции. Всюду определенная функция f, аргументы которой - слова данного алфавита, а значения - натуральные числа, называется конструируемой по времени, если существует многоленточная машина Тьюринга, время вычисления которой на входе x совпадает с $f(x)$. Разумеется, аналогичное определение можно дать и для любой другой вычислительной модели, заменив многоленточные машины Тьюринга на машины этой модели. Мы определили конструируемость по времени так, как это сделано в [Ахо Хоп Уль 74, упр. 11.1]. Другое определение конструируемости (приводящее к другому, хотя и близкому понятию) сформулировано в [Сли 81, гл. 1, § 1, п. 1]: функция называется конструируемой по времени, если время ее вычисления на любом аргументе не превосходит ее значения на этом аргументе. В [Ахо Хоп Уль 74, § 10.1] определяется также конструируемость по ёмкости: всюду определенная функция, аргументы которой - слова данного алфавита, а значения - натуральные числа, называется конструируемой по ёмкости, если существует машина, ёмкость вычисления которой на входе x совпадает с $f(x)$. Заметим, что понятие конструируемости по ёмкости использует в своем определении понятие ёмкости вычисления, а это последнее имело у нас точное определение только для многоленточных машин Тьюринга. В формулируемых далее теоремах будет идти речь о конструируемости функций с натуральными аргументами. Мы предполагаем при этом, что натуральный аргумент n подается на вход машины в виде n-буквенного слова в однобуквенном алфавите. Таким образом, например, конструируемость функции $T:\mathbb{N}\to\mathbb{N}$ по емкости для вычислительной модели "многоленточные машины Тьюринга" означает, что существует (детерминированная) машина Тьюринга, входной алфавит которой однобуквенный, и работа этой машины на входном слове любой длины n

заканчивается, причем $T(n)$ равно максимальной из длин использованных участков на рабочих лентах.

Как и следовало ожидать, диагональная конструкция позволяет строить сколь угодно сложные вычислимые функции (и даже предикаты). Более точно, для любой всюду определенной вычислимой функции φ из словарного ансамбля X в $\mathbb{N}$ найдется такое разрешимое множество слов A, что всякая многоленточная машина Тьюринга, распознающая множество A, для всех $x \in X$, кроме конечного числа, будет при работе над исходным данным x затрачивать время и емкость, превосходящие $\varphi(x)$ ([Тра 67, гл. II, § 5, теорема 14]).

Принципиальная возможность классификации обеспечивается так называемыми теоремами об иерархии. Теорема об иерархии для заданной сложности (времени или емкости) указывает, какое уменьшение верхней оценки сложности приводит к уменьшению класса функций (или предикатов), вычислимых с такой сложностью. Первая теорема об иерархии принадлежит Цейтину и касается времени работы (числа шагов) нормальных алгоритмов (см. [Яновс 59, с. 45]). Все приводимые ниже теоремы об иерархии относятся к сложности вычислений на многоленточных машинах Тьюринга (в частности, под конструируемостью по времени или емкости понимается конструируемость на этой модели), хотя, разумеется, аналогичные вопросы могут быть сформулированы и для любой другой вычислительной модели, как только определены понятия времени и емкости вычислений на этой модели.

Теорема об иерархии по емкости состоит в следующем (см. [Сей 77]). Пусть S - некоторая конструируемая по емкости функция $\mathbb{N} \to \mathbb{N}$ и пусть S_1 - произвольная функция, удовлетворяющая условию $\lim\limits_{n \to \infty} \frac{S_1(n)}{S(n)} = 0$; тогда существует множество слов, распознаваемое с емкостью $S(l(x))$ и не распознаваемое с емкостью, ограниченной функцией $S_1(l(x))$.

Отметим, что любое множество, распознаваемое с емкостью $o(\log \log l(x))$, распознаваемо с нулевой емкостью (см. [Хоп Уль 69, теорема 10.8]). Это обстоятельство не противоречит сформулированной только что теореме об иерархии, так как не существует неограниченных

конструируемых по емкости функций S, для которых $S(x) = o(\log \log l(x))$.

Теорема о временной иерархии выглядит следующим образом. Пусть T - некоторая конструируемая по времени функция $\mathbb{N} \to \mathbb{N}$ и $T(n) \geq n$. Пусть, далее T_1 - некоторая функция $\mathbb{N} \to \mathbb{N}$, удовлетворяющая условию

$$\lim_{n \to \infty} \frac{T_1(n) \log T_1(n)}{T(n)} = 0.$$

Тогда существует множество слов, распознаваемое с временем $T(l(x))$ и не распознаваемое с временем, ограниченным $T_1(l(x))$ (см. [Сей Фиш Мей 78]). Для многих функций T и T_1, представляющих практический интерес, например, полиномов и экспонент, в предыдущей теореме можно заменить $\log$ на $\log^{\alpha}$, где α - произвольное положительное число (см. там же). Неизвестно, можно ли это сделать в общем случае.

"Предел точности" классификации вычислимых функций по времени вычислений дает теорема о линейном ускорении (ср. [Хоп Уль 69, § 10.3]), которую для рассматриваемых нами многоленточных машин Тьюринга с входным и выходным устройствами можно изложить следующим образом. Пусть c - произвольное положительное число. Если функция f может быть вычислена за время, ограниченное функцией T, то она может быть вычислена и за время, ограниченное функцией $\max\{cT, (1+c)(l(x) + l(f(x)))\}$, здесь l - длина слова. Первая теорема о линейном ускорении (для нормальных алгорифмов) была получена Цейтиным (см. [Яновс 59, с. 44], [Цей 71, теорема 2]).

Потребности практики стимулировали изучение классов функций, имеющих относительно небольшую сложность вычисления. Можно, скажем, рассматривать все функции, вычислимые на многоленточных машинах Тьюринга за линейно зависящее от длины входа время. (Обычно их называют просто "функциями, вычислимыми за линейное время"; в аналогичных ситуациях в дальнейшем мы также будем опускать слова "от длины входа".) С другой стороны, время вычисления большинства практически интересных функций с помощью известных алгоритмов ограничено не линейной функцией, а полиномом. Время вычисления суперпозиции двух таких функций, конечно, также ограничено некоторым полиномом; степени таких полиномов могут быть произвольными. Таким образом, мы приходим

к определению одного из самых важных классов вычислимых функций, а именно, класса $\mathcal{P}$. Класс $\mathcal{P}$ содержит те и только те (словарные) функции, которые вычислимы за полиномиально ограниченное (от длины входа) время на многоленточных машинах Тьюринга. То же обозначение используется для соответствующего класса предикатов (и множеств), см. [Ахо Хоп Уль 74, § 10.2]. Как мы видели в § 6, любая словарная функция из класса $\mathcal{P}$ может быть вычислена на многоленточной машине Тьюринга с одной рабочей лентой за время, не превосходящее некоторого полинома; поэтому при определении класса $\mathcal{P}$ можно было бы ограничиться только такими машинами.

Мы определили, как это обычно и делается, класс $\mathcal{P}$ для словарных функций. Однако столь же естественно можно ввести класс $\mathcal{P}$ для функций из X в Y, если X и Y - ансамбли колмогоровских комплексов (оба - ориентированных или оба - неориентированных). Именно, класс $X-Y-\mathcal{P}$ (для фиксированных X и Y) образован всеми функциями, вычислимыми $X-Y$-алгоритмами Колмогорова за время, не превосходящее полинома от числа вершин входного комплекса. Возникающий класс не изменится, если заменить в приведенном определении число вершин комплекса на норму комплекса - при том условии, что рассматриваемая норма обладает свойствами Д1 и Д2 из § 6. Дело в том, что все такие нормы полиномиально связаны с числом вершин (не меньше его и не больше его квадрата - с точностью до мультипликативной константы).

Таким образом, мы приходим к определению класса $\mathcal{P}$ для функций, аргументами и значениями которых служат колмогоровские комплексы. Так как словарные ансамбли естественно вкладываются в подходящие ансамбли колмогоровских комплексов, мы получаем для словарных функций два определения - исходное, в терминах многоленточных машин Тьюринга, и новое, возникающее при отождествлении словарных функций с соответствующими функциями на комплексах. Как и следовало ожидать, они оказываются эквивалентными. В итоге обозначение $\mathcal{P}$ мы закрепляем за объединением классов $X-Y-\mathcal{P}$, возникающих при всевозможных ансамблях X и Y.

Важную роль играет также класс множеств, порождаемых однопосылочными исчислениями колмогоровского типа за полиномиально

(от числа вершин порождаемого элемента) ограниченное число шагов. (Без изменения возникающего класса множеств можно заменить число вершин на какую-либо норму на ансамбле колмогоровских комплексов, удовлетворяющую условиям Д1 и Д2 из § 6, ср. сказанное выше.) Этот класс обозначается $\mathcal{NP}$. Традиционно класс $\mathcal{NP}$ определяется для словарных множеств - как класс множеств, распознаваемых за полиномиальное время на так называемых недетерминированных машинах Тьюринга (отсюда первая буква обозначения $\mathcal{NP}$); см. [Ахо Хоп Уль 74, § 10.2].

Для словарных множеств может быть дано и другое определение класса $\mathcal{NP}$, равнообъемное традиционному. Именно, фиксируем стандартное вложение данного словарного ансамбля в ансамбль неориентированных колмогоровских комплексов и отнесем к классу $\mathcal{NP}$ те словарные множества, которые при этом вложении переходят в подмножества ансамбля колмогоровских комплексов, принадлежащие классу $\mathcal{NP}$. Наконец, можно определить класс словарных множеств из $\mathcal{NP}$ как состоящий из всех множеств, порождаемых грамматиками Хомского типа 0 за полиномиальное число шагов.

Описывая ситуацию более детально, можно, разумеется, как и в случае класса $\mathcal{P}$, определить сначала класс $\mathcal{NP}$ для каждого отдельного ансамбля, а потом взять в качестве $\mathcal{NP}$ объединение полученных классов.

С точки зрения "практической", "полиномиальной" теории алгоритмов и исчислений, классы $\mathcal{P}$ и $\mathcal{NP}$ подобны классам разрешимых и породимых множеств. Примеры множеств из класса $\mathcal{P}$:

1) произвольный контекстно-свободный язык (см. [Янг 67], [Вэл 75]);

2) (для всякого d) множество пар неизоморфных графов со степенями вершин, не превосходящими d (см. [Зем Кор Тыш 82]);

3) множество всех систем линейных неравенств с целыми коэффициентами, разрешимых в действительных (а следовательно, и в рациональных) числах (см. [Хач 79]).

Примеры множеств из класса $\mathcal{NP}$:

1) множество всех выполнимых формул логики высказываний; (в [Усп Сем 81] вместо этого примера был указан другой, ошибочный);

2) множество всех пар изоморфных графов;

3) множество всех линейных неравенств с целыми коэффициентами, разрешимых в целых числах.

Мы хотели бы заметить, что многие практически важные проблемы принадлежат $\mathcal{NP}$ (в том смысле, что множество объектов, удовлетворяющих условиям проблэмы, является множеством из класса $\mathcal{NP}$). Ясно, что $\mathcal{P} \subset \mathcal{NP}$. Проблема, верно ли, что $\mathcal{P} = \mathcal{NP}$, является важнейшей нерешенной проблемой. Вопрос о совпадении этих классов весьма важен, он имеет следующее содержательное истолкование: "относятся ли практически возникающие задачи (задачи класса $\mathcal{NP}$) к классу реально решаемых (к классу $\mathcal{P}$)?"

Вопрос о совпадении классов $\mathcal{P}$ и $\mathcal{NP}$ может ставиться отдельно для каждого словарного ансамбля и для каждого ансамбля колмогоровских комплексов. Но, как нетрудно видеть, утвердительное или отрицательное решение этого вопроса для любого из словарных ансамблей (алфавит каждого из них, напомним, содержит не менее двух букв) или для любого из таких колмогоровских ансамблей, алфавит которых содержит не менее трех букв, а число, ограничивающее исходящую степень вершины, также не меньше трех, влечет за собой его решение (в ту же сторону) и для всех упомянутых ансамблей.

§ 8. ПОНЯТИЕ μ-РЕКУРСИВНОЙ ФУНКЦИИ

Как сказано в § 7, изучение класса $\bigcup_s \mathrm{Com}(\mathbb{N}^s, \mathbb{N})$ является важной задачей. Этот класс определяется как класс всех вычислимых числовых функций. Вычислимость здесь понимается в интуитивном смысле на основе неформального математического понятия алгоритма. Можно считать (см. опять-таки § 7), что к изучению только ч и с л о в ы х вычислимых функций сводится вся дескриптивная теория вычислимых функций.

Замечательным и неожиданным фактом является то, что класс $\bigcup_s \mathrm{Com}(\mathbb{N}^s, \mathbb{N})$ может быть описан в чисто функциональных терминах без использования вычислительных или порождающих моделей. Это открытие принадлежит Клини. Именно, Клини обнаружил, что понятие вычислимой теоретико-числовой функции совпадает по объему с введенным им понятием μ-рекурсивной функции (см. [Кли 43]). Употребляя для обозначения клиниева понятия термин "μ-рекурсивный", мы следуем, например, монографиям [Хер 65,

гл. 3], [Март 70, § 6]. μ-рекурсивная функция определяется как числовая функция, получаемая из некоторого фиксированного набора простейших исходных функций с помощью применения (в произвольном числе) простейших операций, также выбранных из некоторого фиксированного набора (см. [Хер 65, гл. 3], [Маль 65, § 2], [Усп 60, п. 2.3, п. 2.7 и п. 3.5]). Этими исходными функциями служат константа и функция следования $x \mapsto x+1$; этими операциями являются подстановка (в широком смысле, как в [Усп 60, п. 2.3]), примитивная рекурсия и минимизация (называемая также μ-оператором). Тот факт, что возникающее понятие (μ-рекурсивность функции) оказывается эквивалентным понятию вычислимости, делает возможным изучение принципиально новой - с логической точки зрения - концепции вычислимости стандартными теоретико-множественными и алгебраическими методами.

Конечно, система функций и операций, входящая в определение μ-рекурсивной функции, не является единственно возможной. Например, с точки зрения практического программирования естественно в качестве набора операций взять все операции, задаваемые операторными схемами. Наборы простейших функций, позволяющие в этой ситуации получить все вычислимые функции, построены в [ЕршА 60]. Естественно возникает задача нахождения условий, при которых заданная система функций и операций позволяет породить весь класс вычислимых функций (и не породить ничего лишнего). В случае, когда система операций состоит из всех операторных схем (другими словами, из всех стандартных схем программ), эта задача поставлена в [ЕршА Ляп 67]. О решении ее частного случая см. в [Неп 72], [Неп 72а].

В дальнейшем метод, используемый при определении μ-рекурсивных функций, послужил основой для построения ряда иерархий вычислимых функций. Эти иерархии не используют понятия сложности вычисления; в них бóльшие классы определяются через меньшие классы, фиксированные функции и какие-либо операции, так что каждый следующий класс получается путем добавления к предыдущему некоторой функции и применения - в том или ином числе - некоторых операций. Замечательно, однако, что естественно получаемые при этом классы оказываются тесно связанными с классами, возникающими в теории сложности вычислений (см.

[Муч 70]).

Если в определении перечислимого множества из § 7 слово "вычислимая" заменить на " μ-рекурсивная", получится определение рекурсивно-перечислимого подмножества натурального ряда $\mathbb{N}$; с помощью простейшего изоморфизма между $\mathbb{N}$ и $\mathbb{N}^s$ определение рекурсивной перечислимости распространяется и на подмножества $\mathbb{N}^s$ при $s \geq 2$.

§ 9. ВОЗМОЖНОСТЬ АРИФМЕТИЧЕСКОГО И ДАЖЕ ДИОФАНТОВА ПРЕДСТАВЛЕНИЯ ЛЮБОГО ПЕРЕЧИСЛИМОГО ЧИСЛОВОГО МНОЖЕСТВА

Арифметический, или полиномиальный, терм - это выражение, полученное при помощи операций сложения и умножения из натуральных чисел и натуральных переменных, т.е. полином с натуральными коэффициентами. Полиномиальное равенство - это равенство двух полиномиальных термов. Полиномиальное равенство с n переменными определяет некоторое n-местное отношение и, тем самым, некоторое множество точек в $\mathbb{N}^n$. Это отношение и это множество называются полиномиальными. Отношения, получающиеся из полиномиальных отношений применением любого числа логических связок и кванторов (соответственно, множества, получающиеся из полиномиальных множеств операциями объединения, пересечения, дополнения и проектирования), называются арифметическими.

Всякое рекурсивно-перечислимое, а следовательно, и всякое перечислимое множество натуральных чисел или кортежей натуральных чисел фиксированной длины является арифметическим. Этот факт является следствием одного предложения Гёделя из его статьи [Гёд 31], а именно, предложения V, утверждающего арифметичность примитивно-рекурсивных отношений (см. также [Кли 52, § 49]). Сейчас ясно, что арифметичность перечислимых множеств сразу ведет к неполноте арифметики (ввиду наличия перечислимого множества натуральных чисел с неперечислимым дополнением, см. ниже § 10).

Если при получении вышеописанным способом отношения или множества из полиномиального отношения или множества используются только кванторы существования или, соответственно, только операции проектирования, такое арифметическое отношение

или множество называется диофантовым (см. [Мат 79], [Мат 79а]). В 1953 г. Дейвис в [Дей 53] высказал гипотезу, что всякое перечислимое числовое отношение (соответственно, числовое множество) является диофантовым. Эта гипотеза подтвердилась. Об истории ее доказательства Матиясевич в [Мат 79в] пишет: "В 1961г. было доказано (см. [Дей Пут Роб 61]) более слабое утверждение: каждое перечислимое множество является показательно-диофантовым множеством, т.е. для каждого перечислимого множества $\mathfrak{M}$ существуют такие выражения K и L, построенные из натуральных чисел и переменных $a, z_1, \ldots, z_n$ с помощью сложения, умножения и возведения в степень, что $a \in \mathfrak{M}$ тогда и только тогда, когда показательно-диофантово уравнение K=L разрешимо относительно $z_1, \ldots, z_n$. После этого для доказательства гипотезы Дейвиса осталось указать способ, позволяющий преобразовать произвольное показательно-диофантово уравнение в некоторое диофантово уравнение, имеющее или не имеющее решения одновременно с ним. Было доказано ([Роб 52]), что такое преобразование возможно, если существует диофантово уравнение

$$G(u, v, z_1, \ldots, z_k) = 0$$

обладающее следующими двумя свойствами: 1) в любом решении этого уравнения $v \leq u^u$; 2) для любого c существует решение, в котором $v > u^c$ (про такое уравнение говорят, что оно имеет экспоненциальный рост). Пример диофантова уравнения, имеющего экспоненциальный рост, который впервые был построен в [Мат 70], завершил доказательство гипотезы о диофантовости перечислимых множеств (полностью доказательство гипотезы Дейвиса изложено в [Мат 72], [Манин 73]). Обратное утверждение о перечислимости диофантовых множеств доказывается легко. Таким образом, класс перечислимых множеств совпадает с классом диофантовых множеств."

Теорема о диофантовости перечислимых множеств служит не только усилением результата об их арифметичности, но одновременно и следующего результата: любое перечислимое множество, расположенное в $\mathbb{N}^s$, можно представить как проекцию разрешимого (это разрешимое множество можно брать размерностью на единицу больше, т.е. из числа подмножеств $\mathbb{N}^{s+1}$; в случае полиномиального множества, подлежащего проектированию, доста-

точно повышения размерности на 9, см. [Мат 77], [Мат 77а]).

Из теоремы о диофантовости перечислимых множеств легко получается еще одно замечательное представление для множеств этого класса (см. [Дей Мат Роб 76]): всякое перечислимое множество натуральных чисел можно представить как множество всех неотрицательных значений подходящего многочлена с целыми коэффициентами, переменные которого пробегают натуральный ряд.

Арифметичность, а тем более диофантовость любого породимого множества натуральных чисел демонстрирует особую роль операций сложения и умножения в математике (ср. с теоремой Тенненбаума в части II, § 5).

§ 10. ПОСТРОЕНИЕ НЕРАЗРЕШИМОГО ПОРОДИМОГО МНОЖЕСТВА

Такое множество может быть построено в произвольном ансамбле X . Философская значимость результата состоит в выяснении соотношения между породимостью и разрешимостью, а именно, в установлении существования исчисления, для которого никаким алгоритмом нельзя определить, будет ли произвольный элемент из X когда-либо порожден. Практическая значимость вытекает из того феномена, что все естественно возникающие в математической практике проблемы разрешения (т.е. проблемы построения разрешающих алгоритмов) суть проблемы разрешения для породимых множеств (разумеется, в самой теории алгоритмов и исчислений, а также в математической логике, встречаются проблемы разрешения и иного, высшего, рода). Указанный феномен частично объясняется, если обратить внимание на следующее свойство тех множеств, для которых математическая практика ставит вопрос об их разрешимости: x тогда и только тогда принадлежит множеству, когда существует w , связанное с этим x заранее заданным вычислимым отношением; ясно, что всякое такое множество породимо. Пусть, например, x - диофантово уравнение, а w - его решение в натуральных числах; очевидно, w связано с x вычислимым отношением (по паре $\langle x,w \rangle$ можно вычислить, является w решением для x или нет); поэтому множество всех диофантовых уравнений, разрешимых в натуральных числах, породимо (но - см. § 9 - неразрешимо). Другой пример: x - ра-

венство двух термов в сигнатуре группы, а w - вывод этого равенства в соответствующем исчислении из добавления к § 3; поскольку по паре $\langle x, w \rangle$ мы можем вычислить, является ли w выводом для x или нет, множество всех равенств, верных в рассматриваемой конечноопределенной группе, породимо (разрешимо оно или нет - зависит от группы, см. ч. П, § I).

Существование неразрешимого породимого множества, или, что то же самое, породимого (= перечислимого) множества с непородимым (= неперечислимым) дополнением, равносильно существованию вычислимой функции, не продолжаемой до вычислимой же всюду (т.е. на X) определенной функции. И такое множество, и такая функция чрезвычайно просто строятся диагональным методом (см., например, [Колм 54]).

Итак, процедура порождения множества $P \subset X$ может не сопровождаться процедурой разрешения. Если же существует определенная на X вычислимая функция, для всякого элемента x из P ограничивающая сложность его порождения, то процедуре порождения можно сопоставить процедуру разрешения.

Естественно возникает вопрос о соотношении сложности порождения и сложности разрешения одного и того же множества. Переходя к обсуждению этого вопроса, фиксируем некоторый словарный ансамбль, в котором и будем рассматривать породимые и разрешимые множества. Фиксируем также вычислительную модель - одноленточные машины Тьюринга без выходной и входной лент (так что исходные данные и результаты записываются и читаются на единственной ленте) и исчислительную модель - порождающие грамматики типа 0 (= грамматики по терминологии [Гла 73]). Определим время вычисления (в данном случае - распознавания) и время порождения как число шагов (вычисления и порождения соответственно). Определим емкость вычисления как максимальную длину ленты в вычислении. Определим емкость вывода в грамматике как максимальную длину слова, встречающегося в этом выводе. Определим емкость порождения элемента x в грамматике как наименьшую емкость вывода x в этой грамматике. Тогда будут верны следующие утверждения, связывающие введенные сложности разрешения и порождения:

(а) Всякое множество, распознаваемое на нашей вычислительной модели со временем не большим T и емкостью не большей S,

может быть порождено некоторым исчислением нашей порождающей модели с временем не большим cT и емкостью не большей cS, (где c – константа, не зависящая от распознаваемого элемента).

(б) Пусть задано некоторое исчисление Γ нашей порождающей модели. Тогда существует алгоритм нашей вычислительной модели, который решает задачу: "По словам a, b и числу k узнать, существует ли в Γ вывод b из a, в котором длины всех промежуточных результатов не превосходят k" с емкостью, не превосходящей $c(l(a)+l(b)+k^2)$, где c не зависит от a, b, k.

Доказательство утверждения (а) почти тривиально. Утверждение (б), в несколько иной форме, было независимо доказано Сэвичем (см. [Сэв 70]) и (в 1970 году) Цейтиным (см. [Неп 74]). Формулировка и доказательство теоремы Сэвича – Цейтина имеются в [Ахо Хоп Уль 74, теорема 10.1]; приблизительная формулировка этой теоремы такова: множество, порождаемое с емкостью S, можно распознавать с емкостью S^2.

Имеется класс алгоритмов и соответствующий ему класс исчислений, для которых естественно определяемые классы "породимых" и "разрешимых" множеств совпадают. Этот класс – конечноавтоматные алгоритмы и исчисления. Однако для него оказывается возможным доказать несовпадение понятия конечноавтоматной разрешимости с понятием конечноавтоматной породимости в другом, количественном смысле. Построить для всякого конечного автомата, порождающего (на выходе) некоторое множество слов, конечный же автомат, разрешающий (распознающий на входе) то же самое множество, оказывается, так сказать, "практически невозможно": бывают порождающие автоматы, для которых при переходе к разрешающему автомату обязательно происходит экспоненциальный рост объема (т.е. числа состояний) вычислительного устройства (см. [ЕршО 62], [Кор 63], [Луп 64]).

§ 11. ПРОБЛЕМА СВОДИМОСТИ ПОСТА

Открытие состоит здесь прежде всего в самой постановке проблемы.

Изучая неразрешимые породимые множества, возникающие в математической практике, можно было обнаружить, что их проблемы разрешения (заведомо нерешимые) в некотором смысле сводятся друг к другу. Именно, неразрешимость любого такого мно-

жества, как показало наблюдение, можно свести к неразрешимости любого другого. Подчеркнем, что речь идет не о всех мыслимых неразрешимых породимых множествах, а лишь об исторически возникших (включая всевозможные конкретные примеры "диагональных" множеств). В силу сказанного, неразрешимость проблемы разрешения для любого такого множества можно свести к неразрешимости "эталонной" проблемы разрешения для какого-либо из диагональных множеств; именно такое сведение - в прямой или косвенной форме - осуществлялось и продолжает осуществляться в математической литературе. Встает естественная проблема, носит ли данное явление универсальный характер, т.е. действительно ли все проблемы разрешения для всех мыслимых перечислимых неразрешимых множеств сводимы друг к другу; разумеется, слово "сводимы" нуждается при этом в должном уточнении. Эта проблема, известная теперь под названием проблемы сводимости, была - вместе с соответствующим уточнением - предложена Постом в 1944 г. в докладе [Пост 44]. В том же докладе Пост указал неразрешимые породимые множества, для которых естественный способ доказательства их неразрешимости не требует обращения к эталонной проблеме (это были первые примеры подобных необычных доказательств неразрешимости множеств). Разумеется, это еще не решало проблему сводимости - тем более, что для многих из указанных Постом множеств самому Посту в [Пост 44] и его последователям (см. [Родж 67, § 9.6]) удалось найти традиционные доказательства их неразрешимости, опирающиеся на отсутствие решения у эталонной проблемы.

Проблема сводимости Поста получила полное и притом отрицательное решение в работах Мучника (см. [Муч 56], [Муч 58]) и Фридберга (см. [Фри 57]): были построены неразрешимые породимые множества с неэквивалентными проблемами разрешения (и тем самым - породимое множество, неразрешимость которого может быть установлена лишь методами, отличными от диагонального; о диагональном методе см. [Шень 79]).

Уточнение понятия сводимости для проблем разрешения представляет собой самостоятельную задачу (которая является предпосылкой для формальной постановки проблемы Поста), решенную, как отмечалось, Постом в [Пост 44]. Для произвольных проблем

A и B сводимость B к A означает нечто большее, чем просто импликацию "A имеет решение" ⇒ "B имеет решение" (эта импликация тривиально истинна, когда обе проблемы одновременно решимы или нерешимы). Если A и B - проблемы, то сведение B к A образует новую, самостоятельную проблему; на это обстоятельство впервые было указано Колмогоровым в [Колм 32] (ср. часть II, § 2). В случае, когда A и B суть проблемы разрешения, представляется естественным следующее понимание - оно было предложено Постом в [Пост 44]. Пусть X и Y - ансамбли, $P \subset X$, $Q \subset Y$, а A и B - проблемы разрешения для, соответственно, P и Q. Сводимость проблемы B к проблеме A или, другими словами, сводимость по разрешимости множества Q к множеству P, означает, согласно Посту, наличие некоторого единого способа преобразования информации о принадлежности или непринадлежности к P всевозможных элементов X в информацию о принадлежности или непринадлежности к Q произвольно заданного элемента Y; наличие такого способа позволяет эффективно давать ответ на любой вопрос вида "$y \in Q$?", пользуясь готовыми ответами на все вопросы вида "$x \in P$?". Сводимость по разрешимости Пост конкретизировал в виде так называемой тьюринговой сводимости (см. [Родж 67, § 9.4]).

Точное определение сводимости по Тьюрингу (и, таким образом, сводимости по разрешимости) будет дано в следующем параграфе. Здесь же мы ограничиваемся неформальным понятием сводимости по разрешимости.

Проблема Поста стимулировала два больших направления исследований. Первое из них пытается ответить на следующий вопрос: "может ли проблема Поста быть решена методами Поста (его доклада [Пост 44])?" или, более технично, "можно ли методами Поста построить неполное перечислимое неразрешимое множество?" (Множество называется полным, если оно перечислимо и к нему тьюрингово сводится любое перечислимое множество; так, все "диагональные" множества полны.) Поскольку "методы Поста" могут пониматься как в более узком, так и в более широком смысле, в результате уточнения возникают два различных вопроса:

1) существует ли такое непустое свойство перечислимых неразрешимых множеств типа малости ("почти-конечности", см. [Родж 67, § 12.6]) дополнения, что любое удовлетворяющее это-

му свойству множество неполно?

2) можно ли сформулировать и притом без использования понятия "тьюрингова сводимость" такое непустое свойство перечислимых неразрешимых множеств, что любое удовлетворяющее этому свойству множество неполно?

Сам Пост рассматривал различные понятия типа почти-конечности дополнения (простоту, гиперпростоту, гипергиперпростоту), но ни одно из них (и даже более сильное понятие максимальности, введенное впоследствии Фридбергом) не оказалось пригодным для утвердительного ответа на первый вопрос. (Существование полных максимальных множеств доказано в [Ейтс 65].)

Положительный ответ на второй вопрос дал Марченков в [Марч 76], доказав, что перечислимые неразрешимые множества с некоторым свойством (а именно, являющиеся одновременно полурекурсивными и α-гипергиперпростыми для некоторой позитивной эквивалентности α) не могут быть полны; существование таких множеств было ранее установлено Дёгтевым в [Дёг 73].

Это первое направление тесно связано с изучением того, как вообще может быть устроено породимое (=перечислимое) множество, расположенное в каком-либо фиксированном ансамбле (например, сколь "плотно" оно может заполнять объемлющее пространство и каков запас перечислимых множеств, содержащихся в его дополнении). В соответствии со своим устройством перечислимое множество может быть отнесено к тому или иному классу (см. [Родж 67, § 8.7]). Перечислимое множество может быть разрешимым (если его дополнение перечислимо) или простым (если его дополнение хотя и бесконечно, но не содержит бесконечных перечислимых подмножеств) или креативным (если любая программа, см. § 14, любого перечислимого подмножества его дополнения может быть эффективно преобразована в элемент, принадлежащий дополнению, но не его подмножеству) и т.д. Одним из центральных результатов является теорема Майхилла (см. [Май 55]), утверждающая, что любые два креативных множества могут быть получены одно из другого вычислимой перестановкой объемлющего ансамбля.

Второе направление занимается так называемыми степенями неразрешимости. На совокупности всех множеств, расположенных в данном ансамбле (или даже в различных таких ансамблях), за-

дается предпорядок: а именно, $P \geq Q$, если проблема разрешения для множества Q сводится к проблеме разрешения для множества P (т.е. Q тьюрингово сводится к P). Классы эквивалентности этого предпорядка называются тьюринговыми степенями неразрешимости (короче – степенями неразрешимости, или тьюринговыми степенями, или Т-степенями). Все разрешимые множества образуют одну тьюрингову степень, она обозначается $\mathbf{0}$ и называется нулевой. Тьюрингова степень называется перечислимой, если она содержит хотя бы одно перечислимое множество. Существование неразрешимого перечислимого множества означает, что существует по крайней мере одна ненулевая перечислимая степень; вопрос о том, существуют ли по крайней мере две такие степени, образует проблему сводимости Поста. На самом деле множество всех перечислимых степеней – бесконечно, хотя и счетно ([Муч 56, теорема 2]; см. также [Родж 67, § 10.2]).

На тьюринговых степенях возникает естественный частичный порядок, относительно которого множество этих степеней образует верхнюю полурешетку (континуальной мощности). Устройству этой полурешетки посвящено большое количество исследований. Начало было положено совместной статьей Клини и Поста [Кли Пост 54]. Вторым важным этапом были упоминавшиеся выше работы Мучника и Фридберга. В качестве третьего этапа можно указать книгу [Сакс 63]. Сейчас проблематика, связанная с верхней полурешеткой Т-степеней, прочно вошла в монографическую и обзорную литературу: см. [Родж 67, гл. 13], [Шёнф 71], обзор [Шор 81], посвященный в основном элементарным теориям этой решетки и ее подструктур, а также обзор [Соа 78], специально посвященный перечислимым Т-степеням. Вот два результата из обсуждаемой области, представляющиеся авторам принципиальными: 1) среди ненулевых Т-степеней существуют минимальные (см. [Родж 67, § 13.5] и [Шенф 71, § 11]; 2) в частично упорядоченном подмножестве ненулевых перечислимых Т-степеней нет минимальных элементов (см. [Муч 56, теорема 3] и [Родж 67, упражнение 10–11]). Отметим в заключение, что наряду с Т-сводимостью Пост ввел и другие виды сводимости; некоторые связанные с ними результаты можно найти в [Дег 79].

§ 12. ПОНЯТИЕ ОТНОСИТЕЛЬНОГО АЛГОРИТМА, ИЛИ АЛГОРИТМА С ОРАКУЛОМ

Чтобы получить определение алгоритма с оракулом A, надо

следующим образом изменить формулировку Колмогорова из § 1. Оракул A – это некоторое множество в каком-то ансамбле. Алгоритм с оракулом имеет вопросное устройство – некий вспомогательный алгоритм $\mathcal{J}$, определенный (т.е. дающий результат) на множестве всех возможных состояний S. Каждый шаг процесса, задаваемого алгоритмом с оракулом, определяется не только возникшим к этому шагу состоянием S, но и истинностью утверждения $\mathcal{J}(S) \in A$. Таким образом, оператор непосредственной переработки Ω_T, дающий следующее состояние S^*, оказывается теперь функцией от двух аргументов – от S и от числа b, принимающего значение 0 или 1 в зависимости от того, верно ли соотношение $\mathcal{J}(S) \in A$. Алгоритм с оракулом A называется также алгоритмом относительно A (см. [Родж 67, § 9.2]).

Понятие алгоритма с оракулом оказалось важным с методологической точки зрения. Дело в том, что теория алгоритмов и исчислений, как и математическая логика (понимаемая, по Чёрчу, как теория формализованных языков), формализует некоторые стороны деятельности человека (в отличие от других математических дисциплин, которые формализуют нечто, не предполагающее непременного присутствия человека). В частности, теория алгоритмов использует понятие "элементарной операции". Понятие элементарности – существенно человеческое понятие. То, что элементарно для человека, может оказаться неэлементарным для других существ, и наоборот. Можно считать, что человек, осуществляя вычисление, непрерывно обращается к некоторому оракулу, только оракул этот отвечает на столь "элементарные" вопросы (типа "тождественны или нет эти два символа?"), что даже не замечается. Можно представить себе более мощный, чем у человека, запас вычислительных средств, подразумевающий, в частности, обращение к некоторому нетривиальному (с человеческой точки зрения) оракулу (который в рамках этих средств не осознается, скорее всего, как внешний оракул, а признается частью самих этих средств).

Высказанные соображения подтверждаются следующими попытками аксиоматически определить понятие вычислимой функции.

Анализируя доказательства, встречающиеся в теории вычислимых функций, можно заметить, что возможен – и даже иногда используется (например, в [Родж 67]) – следующий способ рас-

суждений. Сначала устанавливаются некоторые основные и интуитивно очевидные свойства класса вычислимых функций, а затем требуемые утверждения выводятся из них. (Подробнее об этом см в [Усп 74, § 8], [Усп 82, § 5].) Сформулируем упомянутые выше основные свойства класса K всех вычислимых числовых функций:

(1) Аксиома функциональных констант:

Класс K содержит все вычислимые числовые функции.

Это свойство применяется тогда, когда в ходе доказательства нужно установить принадлежность к K какой-либо конкретной функции. Его можно заменить на

(1′) Класс K содержит константу 0 и функцию следования.

(2) Аксиома операторных констант:

Класс K замкнут относительно операторов подстановки, рекурсии и минимизации.

Это свойство применяется, если из утверждения о принадлежности к K каких-то функций нужно вывести утверждение о принадлежности к K некоторой другой функции, выражающейся через первые. В связи с аксиомами (1′) и (2) естественно вспомнить наш § 8.

(3) Аксиома протокола:

Для всякой функции f из класса K существуют:

I. Множество натуральных чисел E, характеристическая функция которого лежит в K, и

II. Функции a и b, определенные на всех элементах E, принадлежащие K и удовлетворяющие условию: значение функции f на числе x равно y тогда и только тогда, когда существует такое q из E, что $a(q) = x$ и $b(q) = y$.

Содержательная интерпретация этой аксиомы такова. Мы предполагаем, что для каждого вычисления существует его протокол (= запись), представляющий собой последовательность сменяющих друг друга состояний алгоритмического процесса (см. § 1). Множество E есть множество кодов всех таких протоколов. В случае, когда K есть просто класс всех вычислимых числовых функций, множество всех протоколов разрешимо, и, следовательно, характеристическая функция E в этом случае принадлежит K. Функции

a и b выделяют из кода протокола исходное данное и результат вычисления.

(4) Аксиома универсальной функции:

В классе K существует двуместная функция, универсальная для всех одноместных функций из K ("универсальность" $U(x, y)$ означает, что $\forall f \in K \exists x \forall y\, f(y) \simeq U(x, y)$).

В качестве основания теории вычислимых функций эти аксиомы намного более очевидны, чем тезис Чёрча. В самом деле, они не позволяют утверждать, что некоторые функции невычислимы. В противоположность этому, наиболее неочевидная часть тезиса Чёрча утверждает, что функции, невычислимые на модели, невычислимы и каким бы то ни было образом.

Кроме других преимуществ аксиоматического подхода - например, замены сложных прямых конструкций короткими аксиомами - укажем два следующих преимущества. Первое состоит в том, что данные аксиомы не только более очевидны, но также и менее техничны, чем тезис Чёрча. Второе преимущество (и недостаток) состоит в том, что аксиоматическая система может иметь (и имеет) различные модели. Действительно, четыре перечисленные выше аксиомы выполнены не только для класса всех вычислимых функций, но и для любого класса всех функций, вычислимых с данным оракулом. Таким образом, все теоремы, выводимые из (1) -(4), выполнены для любого такого класса. Это объясняет возможность "релятивизации" многих теорем (см. [Кли 52,§ 58, теорема X, § 65, теорема XXIУ и т.д.]). И наоборот, только те теоремы, которые следуют из аксиом(1) -(4),можно релятивизировать. Действительно, как доказано в [Шень 80], любой класс функций, удовлетворяющий этим аксиомам, в действительности есть класс всех функций, вычислимых с некоторым фиксированным оракулом.

Таким образом, мы видим, что с чисто теоретической стороны понятие алгоритма с оракулом позволяет релятивизировать теорию алгоритмов (см. [Родж 67, § 9.3]). С более практической стороны, оно позволяет дать точное определение общего понятия сводимости по разрешимости и, следовательно, дать точную формулировку фундаментальной проблемы сводимости (см. § 11). Действительно, теперь можно ввести следующее определение.

Множество Q сводится по Тьюрингу (= сводится по разрешимости) к множеству P тогда и только тогда, когда существует относительный алгоритм, вычисляющий характеристическую функцию множества Q относительно множества P, или, в оракульных терминах, существует алгоритм с оракулом P, вычисляющий эту характеристическую функцию. Понятие алгоритма с оракулом и сам термин "оракул" впервые появились в статье Тьюринга [Тью 39], по этой причине Пост ввел в [Пост 44] термин "сводимость по Тьюрингу" для обозначения сводимости проблемы разрешения самого общего вида.

Важным и естественным частным случаем сводимости по Тьюрингу является сводимость за полиномиальное время. (Она определяется заданием полиномиального - от длины входа - ограничения на время работы алгоритма с оракулом.) Естественно поставить проблему полиномиальной по времени сводимости: все ли множества из класса $\mathcal{NP}\setminus\mathcal{P}$ сводятся друг к другу за полиномиальное время? (Конечно, если $\mathcal{NP} = \mathcal{P}$, то проблема тривиальна.) Заведомо сводятся друг к другу за полиномиальное время многие представители класса $\mathcal{NP}$, возникшие из математической практики; к каждому из таких представителей все множества из $\mathcal{NP}$ сводятся за полиномиальное время (см. [Ахо Хоп Уль 74, гл. 10]). Неизвестно, сводятся ли за полиномиальное время все множества из $\mathcal{NP}$ к множеству всех пар изоморфных графов (ср. § 7, второй пример множества из $\mathcal{NP}$).

§ 13. ПОНЯТИЕ ВЫЧИСЛИМОЙ ОПЕРАЦИИ

Под операцией понимается функция, аргументы и значения которой суть множества. В то время как обычная вычислимая функция эффективно дает по одному конструктивному объекту другой, вычислимая операция должна эффективно давать по одному множеству конструктивных объектов другое; говоря неформально, эффективность естественно усматривать в том, что операция обеспечивает процесс порождения результирующего множества, коль скоро задан процесс порождения исходного множества; как следствие этого обстоятельства вычислимая операция переводит породимые множества в породимые. Сказанное естественно воплощается в нижеследующее формальное определение вычислимой опера-

ции (см. [Усп 55], а также [Родж 67, § 9.7], где вычислимые операции называются операторами перечисления, в оригинале - "enumeration operator").

Пусть X и Y - два произвольных множества, X_F - множество всех конечных подмножеств множества X. Пусть далее R-произвольное отношение между X_F и Y (т.е. произвольное подмножество произведения $X_F \times Y$). Определим отображение Φ множества 2^X в множество 2^Y формулой:

$$\Phi(A) = \{y \mid \exists D(D \subset A \;\&\; \langle D, y\rangle \in R)\} .$$

Отображение Φ, получаемое таким образом, будем называть R-отображением 2^X в 2^Y. Всякое R-отображение Φ монотонно: $(A' \supset A) \Rightarrow \Phi(A') \supset \Phi(A)$. Если X состоит из элементов некоторого ансамбля, то элементы множества X_F суть конечные объекты; в силу сделанных в § 0 соглашений, осмысленно говорить о породимости R. Так вот, если X,Y и R породимы, R-отображение 2^X в 2^Y называется вычислимой операцией (из 2^X в 2^Y). Ясно, что если $A \in Gen(X)$ и Φ- вычислимая операция, то $\Phi(A) \in Gen(Y)$. В соответствии с нашим неформальным представлением о вычислимой операции, процесс преобразования одного множества в другое можно описать следующим образом. Мы одновременно порождаем конечные подмножества исходного множества A и элементы отношения R . Всякий раз, как первая компонента порожденного элемента отношения R совпадает с порожденным конечным подмножеством множества A, мы засылаем соответствующую вторую компоненту в множество $\Phi(A)$.

В действительности, выше мы определили только одноместные вычислимые операции; понятие многоместной вычислимой операции может быть определено совершенно аналогично (или сведено к) понятию одноместной вычислимой операции.

В терминах вычислимых операций легко могут быть определены вычислимые операторы. Под оператором мы понимаем функцию, аргументами и значениями которой служат функции. Определение вычислимого оператора таково - см. [Родж 67, § 9.8], где эти операторы названы частичнорекурсивными ("partial recursive"); придумавший эти операторы Клини называл их частичнорекурсивными функционалами ("functionals"), см. [Кли 52, § 63 и § 64]. Для произвольной операции Φ, отображающей $2^{U \times V}$ в $2^{X \times Y}$, оп-

ределим оператор Ψ, осуществляющий отображение из $\mathcal{F}(X, Y)$ в $\mathcal{F}(U,V)$: оператор Ψ определен на функции $f \in \mathcal{F}(X,Y)$ тогда и только тогда, когда $\Phi(f) \in \mathcal{F}(U,V)$, и в этом случае $\Psi(f) = \Phi(f)$. Оператор Ψ, полученный таким образом для породимых X , Y, U, V и вычислимого Φ, называется вычислимым, или частичнорекурсивным. Ясно, что если $f \in Com(X,Y)$, то $\Psi(f) \in Com(U, V)$. Частичнорекурсивный оператор называется рекурсивным, если он определен для каждой функции из $\mathcal{F}(X, Y)$.

Замечательно, что интуитивное понятие вычислимой операции не потребовало для своей формализации никаких новых понятий, кроме понятия алгоритма; этот факт еще раз подтверждает "емкость" и "универсальность" понятия алгоритма. Замечательно также, что понятие вычислимой операции совпадает по объему с понятием исчислительной операции (см. выше § 3): этот факт подтверждает "емкость" и "универсальность" понятия исчисления. Замечательно, наконец, что вычислимые операции являются непрерывными отображениями - в предположении, что произвольная система множеств $\mathcal{T}$ рассматривается как топологическое пространство с некоторой естественной топологией. Эта топология задается как в [Усп 55] и [Родж 67, упр. 11-35], а именно: для всякого конечного множества D положим $\mathcal{O}_D = \{A | D \subset A \in \mathcal{T}\}$ и совокупность всех таких $\mathcal{O}_D$ объявим базой топологии. Естественность такой топологии заключается в том, что "близкими" (т.е. принадлежащими одной и той же базовой окрестности) объявляются те множества из $\mathcal{T}$, которые характеризуются одной и той же конечной информацией (т.е. информацией о принадлежности к множеству фиксированного - для данной окрестности - конечного перечня элементов).

С помощью вычислимых операций легко определяются вычислимость одной функции относительно другой и тьюрингова сводимость множеств: g вычислима относительно f, если существует вычислимая операция, переводящая f (как множество пар) в g (как множество пар); множество Q сводится по Тьюрингу к множеству P, если характеристическая функция множества Q вычислима относительно характеристической функции множества P (или, непосредственно в терминах операций над множествами, если существуют такие вычислимые операции Φ_1 и Φ_2, что $Q = \Phi_1(P, \bar{P})$,

$\overline{Q} = \Phi_2(P, \overline{P})$, где $\overline{P}$ и $\overline{Q}$ суть дополнения к P и Q в соответствующих ансамблях).

Один из наиболее принципиальных фактов, относящихся к вычислимым операциям (в частности, к частичнорекурсивным операторам), устанавливается в так называемой теореме о неподвижной точке, или первой теореме о рекурсии Клини (см. [Родж 67, § 11.6]). Эта теорема утверждает, что уравнение $\Psi(A) = A$, где Ψ - вычислимая операция, имеет минимальное решение (что верно для всякой монотонной Ψ), и это решение перечислимо (а если Ψ к тому же есть рекурсивный оператор, то это решение еще и униформно, т.е. является вычислимой функцией). Первая теорема о рекурсии позволяет естественным образом задавать породимые множества (в частности, вычислимые функции) с помощью вычислимых операций. При этом описание вычислимой операции рассматривается как финитное задание множества или функции, являющейся неподвижной точкой операции. Последнее обстоятельство послужило одной из отправных точек для работ Маккарти и Скотта по математической теории вычислений (см. [Манна 74, гл. 5]).

Весьма плодотворным оказалось изучение специальных способов задания вычислимых операторов - так называемых схем программ (см. [Лак Парк Пат 70]). Важнейшие классы схем программ таковы: рекурсивные схемы Де Беккера - Скотта (см. [Кот 78]), стандартные схемы А.П.Ершова (см. [ЕршА 73]), структурированные схемы Глушкова (см. [Глу 65], [Сем 78]). Вся теория схем программ началась со схем Янова (см. [Янов 58], [ЕршА 68], [ЕршА 77, § 8.5]) - стандартных схем с одной переменной. Работы Глушкова по структуированным схемам программ составляют раздел созданной им теории систем алгоритмических алгебр (см. [Глу Цей Юще 78]). Эта теория позволила использовать алгебраические и логические методы для изучения схем программ и послужила основой для интенсивно развивающейся дисциплины - программной логики (см. [Вал 79]).

Вычислимые операции приводят не только к определению сводимости множеств конструктивных объектов, но также и к некоторым вариантам представления о сводимости совокупностей таких множеств. Пусть $\mathcal{A}$, $\mathcal{B}$ - совокупности множеств, причем элементами этих множеств являются конструктивные объекты.

Тогда, по определению, $\mathcal{B}$ слабо сводится к $\mathcal{A}$, если для каждого $A \in \mathcal{A}$ существует вычислимая операция Φ такая, что $\Phi(A) \in \mathcal{B}$; совокупность $\mathcal{B}$ сильно сводится к $\mathcal{A}$, если существует вычислимая операция Φ такая, что $\Phi(A) \in \mathcal{B}$ для каждого $A \in \mathcal{A}$. В том случае, когда совокупности $\mathcal{A}$ и $\mathcal{B}$ состоят из всюдуопределенных числовых функций, понятие сильной сводимости и слабой сводимости были предложены, соответственно, Медведевым (в [МедД 55], [МедД 56]) и Мучником (в [Муч 63]), поэтому сильную сводимость можно называть сводимостью по Медведеву, а слабую сводимость - сводимостью по Мучнику; мы вернемся к этим сводимостям в конце § 1 части II. Существуют такие две совокупности всюдуопределенных функций, что одна из них сводится к другой слабо, но не сильно (см. [Муч 63]). Класс всех совокупностей всюдуопределенных числовых функций, слабо (соответственно, сильно) сводящихся друг к другу, называется слабой (соответственно, сильной) степенью трудности (см. [МедД 55], [МедД 56], [Муч 63]).

Пусть $\mathbb{A}$, $\mathbb{B}$ - (сильные или слабые) степени трудности; по определению $\mathbb{A} \leq \mathbb{B}$, если некоторый элемент степени $\mathbb{A}$ (сильно или, соответственно, слабо) сводится к некоторому элементу степени $\mathbb{B}$. (Заметим, что в [Муч 63, с. 1331, строка 25-я сверху] при введении знака "$\leq$" допущена опечатка: вместо "$\mathbb{B}$ слабо сводится к проблеме A" следует читать "A слабо сводится к проблеме $\mathbb{B}$".) Сильные степени трудности, частично упорядоченные отношением сильной сводимости, образуют решетку, которая называется решеткой Медведева (см. [Родж 67, § 13.7]). Верхняя полурешетка Т-степеней вложена в решетку Медведева: для этого каждую Т-степень A следует отождествить с сильной степенью трудности одноэлементной совокупности функций $\{\chi\}$, где χ есть характеристическая функция числового множества A, а A есть произвольное множество, имеющее A своей Т-степенью. Слабые степени трудности, частично упорядоченные отношением слабой сводимости, также образуют решетку, которую можно назвать решеткой Мучника. Имеется очевидное отображение решетки Медведева в решетку Мучника, и это отображение оказывается гомоморфизмом, см. [Муч 63]. Для каждой из решеток наименьшим элементом $\mathbb{0}$ является степень трудности, содержащая вычислимую функцию, а наибольшим элементом $\mathbb{1}$ является степень труд-

мости пустой совокупности. Эти решетки будут использованы в части II, § 2.

§ I4. ПОНЯТИЕ ПРОГРАММЫ: ПРОГРАММЫ КАК ОБЪЕКТЫ ВЫЧИСЛЕНИЯ И ПОРОЖДЕНИЯ

Существенным этапом в развитии теории алгоритмов было осознание того, что алгоритмы и исчисления имеют формальные задания (см. выше § 2 и § 4) и что эти формальные задания (точнее, некоторые их записи) сами могут служить объектами алгоритмических и исчислительных преобразований. Это открытие было сделано Тьюрингом в [Тью 36]. В этом параграфе мы обсудим возникающие в связи с этим понятия "способ программирования" и "программа".

Совокупность всевозможных формальных заданий, естественно возникающая при рассмотрении фиксированной вычислительной (или порождающей) модели, не обязательно лежит в каком-нибудь одном ансамбле. Поэтому элементы этой совокупности не могут, вообще говоря, ни подаваться на вход какого-либо одного алгоритма, ни порождаться каким-либо одним исчислением. Чтобы формальные задания во всей их совокупности могли быть входами или выходами алгоритма или порождаться исчислением, эти задания надо предварительно записать (закодировать) так, чтобы полученные записи, или коды, принадлежали какому-нибудь одному ансамблю.

Именно так осуществляется переход от изображений нормальных алгорифмов к записям этих алгорифмов (см. [Наг 77б]): изображения суть слова, но не принадлежащие никакому единому словарному ансамблю, тогда как записи суть слова в двубуквенном алфавите. Такая предварительная обработка не нужна для реальных языков программирования, где описание алгоритма представляет собой слово в некотором общем для данной вычислительной модели алфавите языка программирования. Она не нужна и для машин Колмогорова над ориентированными или неориентированными колмогоровскими комплексами с фиксированным алфавитом разметки: формальные задания всех таких машин можно представить себе лежащими в одном и том же ансамбле. С другой стороны, переход к записям необходим, например, для машин Тьюринга. Формаль-

ное задание алгоритма вычисления на такой машине может содержать символы (в частности, обозначения внутренних состояний), число которых заранее не ограничено. Наиболее естественный выход - считать эти символы комбинациями более простых, принадлежащих уже к ограниченному алфавиту. Так это и делается в математических текстах, где для обозначения состояния машины Тьюринга может использоваться, скажем, символ q_{236}.

Итак, пусть фиксирована некоторая вычислительная модель. Пусть также фиксированы входной ансамбль X и выходной ансамбль Y . Тогда возникает некоторое семейство X-Y-алгоритмов - семейство всех тех алгоритмов, которые реализуются конкретными представителями данной вычислительной модели. Например, если наша вычислительная модель - трехленточные машины Тьюринга, а X и Y- словарные ансамбли в некоторых алфавитах, то каждой конкретной трехленточной машине Тьюринга (характеризуемой своими ленточными алфавитами, совокупностью внутренних состояний и системой команд) соответствует некоторый X-Y-алгоритм. Но ни сами эти представители (в примере - трехленточные машины Тьюринга), ни формальные задания соответствующих алгоритмов еще не представляют собой конструктивных объектов какого-либо единого ансамбля и, тем самым, не могут служить объектами алгоритмических преобразований. Поэтому нам следует договориться о том, каким образом формальные задания следует погрузить в некоторый ансамбль. Такое погружение мы будем называть <u>способом программирования</u> (для данной вычислительной модели). Подчеркнем, что для одной и той же вычислительной модели и при одних и тех же входных и выходных ансамблях возможны различные способы программирования. Например, для рассмотренного случая машин Тьюринга можно по-разному договориться о способе кодирования внутренних состояний (число которых заранее не ограничено) с помощью конечного набора символов; традиционная запись вроде "q_{236}" (см. выше) отнюдь не является единственной возможной. Введенное понятие "способ программирования" можно обозначать более полным термином "способ программирования алгоритмов" - для отличения от рассматриваемого ниже способа программирования вычислимых функций.

С каждым способом программирования связан некоторый ансамбль, <u>программный ансамбль</u> данного способа программирования,

некоторые элементы которого рассматриваются как коды формальных заданий соответствующих алгоритмов и называются программами. Вообще говоря, не все элементы программного ансамбля являются программами, однако при всех разумных способах программирования множество P_0 всех программ является разрешимым подмножеством программного ансамбля P. Можно условиться трактовать всякий элемент из P как программу, рассматривая любое p из $P \setminus P_0$ в качестве программы алгоритма с пустой областью применимости. Так мы и будем поступать в дальнейшем.

Всякий алгоритм вычисления на данной модели имеет программу в P. Эта программа содержит минимально необходимую информацию, которая отличает данный алгоритм от всех других алгоритмов вычисления на рассматриваемой модели (но не включает информации о способе программирования, общей для всех алгоритмов модели) и получается некоторым "простым и естественным" образом из формального задания алгоритма. Этот простой и естественный образ состоит в выявлении в формальных заданиях более "мелкой", "внутриклеточной" структуры, погружающем все формальные задания в один общий ансамбль.

Понятие способа программирования не является точным математическим понятием (как и понятие вычислительной модели). Мы попытаемся сейчас указать некоторые свойства, которыми должен обладать каждый разумный способ программирования. С этой целью попробуем ответить на следующий вопрос: какие математические понятия связаны в нашем представлении с термином "способ программирования"? Прежде всего - и это уже отмечалось - способ программирования предполагает фиксацию, помимо входного ансамбля X и выходного ансамбля Y, некоторого третьего ансамбля - ансамбля программ. Далее, следует как-то отразить наши представления о способе применения данной программы к данному аргументу. Нам представляется, что достаточно (и, может быть, исчерпывающе) полное представление об этом дает трехместная функция $A(p, x, t)$, аргументами которой являются программа p, исходное данное x и натуральное число (момент времени) t, а значение равно (полному) состоянию вычислительного процесса в момент времени t при работе машины с программой p на исходном данном x. Рассмотрение этой функции предполагает, разумеется, что все состояния вычислительного процесса сделаны, как это

описано в § 1, элементами некоторого ансамбля (такое погружение может, конечно, происходить разными способами). Так определенную функцию A естественно назвать вычислительной функцией - для заданной вычислительной модели (с фиксированным способом погружения ее состояний в некоторый ансамбль) и заданного способа программирования.

В терминах вычислительной функции A могут быть выражены и некоторые другие понятия, связанные с вычислительной моделью. Например, входная процедура есть по существу отображение $x \mapsto A(p,x,0)$. Напротив, выходная процедура, так же как и правило, определяющее момент окончания работы, должны быть заданы помимо вычислительной функции. Именно, правило окончания представляет собой предикат Π, заданный на множестве всех возможных состояний вычислительного процесса (множестве, элементами которого являются значения вычислительной функции A), а выходная процедура - отображение R этого множества в выходной ансамбль Y. Введя эти понятия, мы можем определить теперь, например, время работы данной программы p на данном аргументе x как наименьшее число t, при котором $A(p, x, t)$ обладает свойством Π. Значение $R(A(p,x,t))$ при этом t естественно считать результатом работы программы p на исходном данном x.

Вычислительные функции, естественным образом построенные по известным вычислительным моделям, обладают некоторыми общими свойствами. Отметим свойство "непрерывности по t", утверждающее, что объект $A(p,x,t+1)$ может быть получен из объекта $A(p,x,t)$ некоторой локальной операцией (и, стало быть, - если принять свойство Д1 из § 6 - норма объекта $A(p, x, t+1)$ близка к норме объекта $A(p,x,t)$). Другое подобное свойство - непрерывность входной процедуры - утверждает, что близким парам $\langle x,p\rangle$ и $\langle x',p'\rangle$ должны соответствовать близкие значения $A(p,x,0)$ и $A(p',x',0)$. Предикат Π, как правило, является локальным свойством, а функция R - локальной операцией.

Было бы интересно построить "аксиоматическую" теорию способов программирования, то есть объявить аксиомами какие-то свойства вычислительной функции, правила окончания и выходной процедуры (в том числе, возможно, уже указанные) и выводить из них различные следствия. Но какие именно свойства следует

считать аксиомами, какие следствия выводить, да и вообще можно ли эту программу осуществить, - об этом говорить еще рано.

Поэтому мы ограничимся лишь частью информации о способах программирования - именно, не будем интересоваться всем процессом вычисления, а будем интересоваться только результатом. Говоря более точно, свяжем с каждым способом программирования двуместную функцию $U(p,x)$, аргументами которой являются программа p и исходное данное x, а значением - результат применения программы p к исходному данному x. (Выше было сказано, каким образом эта функция может быть выражена через A, Π и R:

$$U(p, x) \simeq R(A(p, x, \mu t \Pi(A(p, x, t)))),$$

где символ "μt" имеет смысл "наименьшее t, такое, что".)
Эта функция не определяет однозначно способ программирования (в ее терминах, например, невозможно сформулировать утверждения, касающиеся промежуточных состояний, возникающих при работе машины с данной программой на данном объекте, или времени работы машины с данной программой на данном объекте).
Функцию U естественно назвать <u>результатной</u> функцией данного способа программирования.

Что можно сказать о свойствах результатной функции U?

Первое свойство функции U состоит, конечно, в ее вычислимости. Следуя программистской традиции, вычисляющий ее алгоритм следует назвать <u>интерпретатором</u> (интерпретаторов - при одном и том же способе программирования, т.е. для одной и той же U - может быть много, но все они, конечно, эквивалентны).

До сих пор мы не предполагали представительности рассматриваемой модели. Сделаем теперь такое предположение: ясно, что наиболее интересны и важны именно представительные вычислительные модели. Итак, пусть модель $X-Y$-представительна. В этом случае интерпретатор называется <u>универсальным алгоритмом</u>, а второе свойство результатной функции состоит в том, что всякий $X-Y$-алгоритм может быть задан некоторой программой. Сформулируем это свойство более точно:

(УМ) для любого $X-Y$-алгоритма $\mathcal{A}$ существует такая программа $p \in P$, что для всех $x \in X$ имеет место условное равенство

$$\mathcal{A}(x) \simeq U(p, x).$$

Свойство (УМ) и выражает, конечно, то обстоятельство, что рассматриваемая модель представительна. Это свойство не предполагает, однако, "эффективного" нахождения p по $\mathcal{U}$. Слово "эффективного" взято здесь в кавычки, так как нуждается в уточнении - ведь алгоритм $\mathcal{U}$ не представляет собой конструктивного объекта. Это слово можно уточнить следующим образом. Рассмотрим семейство алгоритмов $\mathcal{U}_i$, заданное параметрически: $\mathcal{U}_i(x)=\mathcal{U}(\langle i,x\rangle)$, $i \in I$, где $\mathcal{U}$ - некоторый алгоритм, а I - породимое множество. Оказывается, что (при всех разумных способах программирования) программу, соответствующую алгоритму $\mathcal{U}_i$, можно эффективно найти по i. Сформулируем теперь это свойство более точно:

(ГМ) для каждого породимого множества I и каждого алгоритма $\mathcal{U}: I \times X \to Y$ существует алгоритм $\vartheta: I \to P$ с областью определения I, для которого $\mathcal{U}(\langle i,x\rangle) \simeq U(\vartheta(i),x)$ при всех $x \in X$, $i \in I$.

Свойство (ГМ) представляется совершенно естественным. В самом деле, достаточно взять в качестве ϑ алгоритм, который в применении к i просто выдает программу $X-Y$-алгоритма: "образуй пару $\langle i,x\rangle$ и примени к ней алгоритм $\mathcal{U}$".

Поясним смысл утверждения (ГМ). Говоря неформально, оно утверждает, что имеется способ трансляции программ любого другого способа программирования в программы нашего способа программирования. В данной выше формулировке $\mathcal{U}$ и задает этот "другой" способ программирования, I - его программный ансамбль, ϑ - транслятор, существование которого утверждается.

Все естественно возникающие способы программирования обладают свойством (ГМ). Это свойство можно рассматривать как уточняющее наши интуитивные представления о способах программирования подобно тому, как тезис Чёрча уточняет наши интуитивные представления о вычислимости. При неформальном понимании способа программирования тезис (ГМ), так же как и тезис Чёрча, нельзя доказать в обычном математическом смысле; но его можно подтверждать, рассматривая различные представительные вычислительные модели и различные точно описанные способы программирования. (Важная разница: тезис Чёрча дает точное описание класса вычислимых функций, в то время как здесь мы имеем лишь "верхнюю оценку" для класса всех способов програм-

мирования!) Утверждение о том, что свойство (ГМ) выполнено для некоторого конкретного способа программирования (связанного с рекурсивными функциями), составляет содержание клиниевой s-m-n-теоремы (точнее, s-1-1-теоремы), см.[Родж 67, § 1.8]. О связи s-m-n-теоремы с программистскими понятиями см. [ЕршА 81], [ЕршА 82, п. 3.2].

Все указанные свойства результатных функций, соответствующих известным способам программирования, можно обратить в требования, накладываемые на произвольную функцию V. Мы приходим тогда к следующим определениям.

Пусть X, Y, E - некоторые множества конструктивных объектов, V - вычислимая функция из $E \times X$ в Y. Функцию V назовем универсальной функцией для X, Y с индексным множеством E, если для нее выполнено такое условие:

(У) для каждого X-Y -алгоритма $\mathcal{A}$ существует такое $e \in E$, что для всех $x \in X$ выполнено условное равенство

$$\mathcal{A}(x) \simeq V(e, x).$$

Пусть X и Y - ансамбли. Существование универсальной вычислимой функции для X, Y с индексным множеством X немедленно приводит - диагональным методом - к такому важному результату, как существование породимых неразрешимых множеств, ср. § 10.

Вычислимую функцию V из $E \times X$ в Y назовем гёделевой, или главной функцией для X, Y с индексным множеством E, если для нее выполнено такое условие:

(Г) для каждого породимого множества I и для каждого алгоритма $\mathcal{A}: I \times X \to Y$ существует алгоритм $\vartheta: I \to E$ с областью определения I, для которого $\mathcal{A}(\langle i, x \rangle) \simeq V(\vartheta(i), x)$ при всех $x \in X$, $i \in I$.

[Очевидно, условие (У) вытекает из условия (Г).] С помощью введенных определений можно переформулировать сказанное ранее - и выраженное в свойствах (УМ), (ГМ) - так: для любого способа программирования результатная функция U является универсальной и даже гёделевой функцией для X, Y с индексным множеством P (здесь X, Y и P - входной, выходной и программный ансамбли). Для произвольных ансамблей X, Y, E существуют функции, обладающие свойством (У), но не свойством (Г). Эти функции, однако, не соответствуют никаким способам программирования и конструируются с помощью искусственного приема.

Пусть по-прежнему фиксирована вычислительная модель с соответствующими ансамблями X, Y, P и какой-либо способ программирования для этой модели. Рассмотрим какую-либо вычислимую функцию f из X в Y. Любая программа любого X-Y-алгоритма, вычисляющего f, называется программой функции f. Мы получаем тем самым некоторое соответствие между элементами P и вычислимыми функциями, при котором каждому $p \in P$ соответствует функция f с программой p. Говоря формально, это соответствие есть некоторое подмножество произведения $P \times Com(X,Y)$. Очевидно, оно полностью задается результатной функцией U. По этой причине саму эту результатную функцию часто называют способом программирования или методом программирования - см., например, [Колм 65, § 3]; при этом, подчеркнем еще раз, имеют в виду соответствие не между программами и алгоритмами, а между программами и вычисляемыми этими алгоритмами функциями (так что более точным было бы говорить о "способе программирования вычислимых функций"). Само указанное соответствие $p \mapsto f$ также называют способом программирования. Итак, мы различаем способы (= методы, системы) программирования алгоритмов и возникающие на их основе способы (= методы, системы) программирования вычислимых функций. Как уже отмечалось, способы программирования функций выделяются среди всевозможных функций из P_X X в Y тем, что они (среди прочих, не полностью, быть может, нам известных свойств) обладают свойством гёделевости. Поэтому указанное свойство гёделевости - для подходящего ансамбля E в качестве индексного множества - само может быть предложено (и, действительно, было предложено в докладах [Усп 56], [Усп 56а]) в качестве формального определения понятия способа программирования вычислимых функций (и, тем самым, косвенно в качестве определения понятия программы функции). Согласно этому определению, способ программирования отождествляется с главной, или гёделевой универсальной функцией. Гёделевы универсальные функции (точнее, сопряженные с ними гёделевы нумерации, см. § I5) постепенно начали трактоваться в современных журналах по вычислительной математике в качестве "простых моделей для языков программирования", как в [Хар Бей 75], или даже просто отождествляться с "системами программирования" ("programming systems"), , как в [Мач Вин Янг 78].

Развитие теории алгоритмов показало, что ни конкретный выбор вычислительной модели, ни конкретный выбор способа программирования (при фиксированной модели) почти не влияет на то, какие теоремы о вычислимых функциях и их программах окажутся справедливыми; мы говорим "почти", имея в виду некоторые исключения, связанные с понятием объема (или нормы) программы, см. далее. Таким образом, можно утверждать, что - в широких пределах - возможна е д и н а я теория способов программирования вычислимых функций. Объяснение этому явлению дает теорема Роджерса об изоморфизме гёделевых (= главных) нумераций, которая будет сформулирована в § 15. Из нее вытекает, что если X, Y, E_1 и E_2 - произвольные ансамбли, а V_1 и V_2 - произвольные гёделевы функции для X, Y с индексными множествами, соответственно, E_1 и E_2, то не только существуют алгоритмы "трансляции" в обе стороны - что очевидно вытекает из определения гёделевости - но можно выбрать эти алгоритмы так, чтобы они вычисляли взаимно обратные функции: существует такой изоморфизм ε ансамблей E_1 и E_2, что для любых $e \in E_1$, $x \in X$ выполнено условное равенство

$$V_2(\varepsilon(e), x) \simeq V_1(e, x).$$

Требование гёделевости (Г) не накладывает никаких ограничений на алгоритм ϑ. В частности, норма объекта $\vartheta(i)$ может значительно превышать норму объекта i. Мы можем стремиться исключить такую возможность и потребовать, чтобы выполнялось неравенство $n(\vartheta(i)) \leq_{(+)} n(i)$. (Разумеется, все сказанное имеет смысл только в том случае, если I и E являются нормированными ансамблями.) Неравенство $n(\vartheta(i)) \leq n(i)$ означает, что $\vartheta(i)$ не содержит "ничего или почти ничего лишнего" по сравнению с i.

Дадим соответствующие определения. Пусть E - нормированный ансамбль. Будем называть вычислимую функцию $V: E \times X \to Y$ <u>оптимальной</u>, или <u>экономной по норме</u>, если выполнено такое свойство:

(О) для всякого нормированного ансамбля I и всякого алгоритма $\mathcal{A}: I \times X \to Y$ существует ограниченно-искажающий алгоритм $\vartheta: I \to E$ с областью определения I, для которого $\mathcal{A}(\langle i, x\rangle) \simeq V(\vartheta(i), x)$ при всех $x \in X$, $i \in I$.

Заметим, что поскольку между любыми двумя нормированными ансамблями существует (ограниченно-искажающий) изоморфизм, то

вместо слов " для всякого нормированного ансамбля" в этом определении можно было бы (без изменения класса оптимальных функций) говорить о каком-то фиксированном нормированном ансамбле, например, об ансамбле двоичных слов. Заметим также, что свойства (У) и (Г), как нетрудно проверить, являются следствиями свойства (0). (То обстоятельство, что в свойстве (0) множество является не произвольным породимым множеством, а ансамблем, как легко проверить, несущественно.)

Первый же вопрос, возникающий в связи с этим определением, таков: а существуют ли вообще оптимальные функции? Оказывается, что ответ на этот вопрос положительный: оптимальную функцию нетрудно построить, используя идеи Колмогорова из [Колм 65]. Это построение можно найти в [Ага 75, с. 44] или в [Шно 75] (в работе Шнорра рассматриваются не оптимальные универсальные функции, а так называемые оптимальные нумерации, но это различие несущественно, см. § 15).

Другой интересный вопрос, возникающий в связи с понятием оптимальности, таков: приводят ли известные способы программирования к оптимальным функциям? Как правило, возникающие результатные функции оказываются неоптимальными, см. [Ага 75, с. 45].

Мы уже упоминали выше теорему Роджерса об изоморфизме гёделевых нумераций. Оказывается, что аналогичная теорема - об изоморфизме любых двух оптимальных нумераций - также имеет место; она доказана Шнорром (см. следующий параграф). Из нее вытекает, что для любых двух оптимальных гёделевых универсальных функций V_1 и V_2 для X и Y с индексными множествами E_1 и E_2 существует такой (ограниченно-искажающий) изоморфизм u нормированных ансамблей E_1 и E_2, что для любых $e \in E_1$ и $x \in X$ выполнено условное равенство

$$V_2(u(e), x) \simeq V_1(e, x).$$

Представляют интерес и другие требования, которые можно пытаться наложить на гёделевы универсальные функции. Можно хотеть, к примеру, чтобы транслятор Θ был бы достаточно быстро работающим алгоритмом, например, требовал бы не более чем полиномиального времени. См. об этом в § 15 в связи с понятием полиномиально главной нумерации.

Все сказанное выше об алгоритмах и их программах можно

повторить и для исчислений. С каждым способом программирования (для данной порождающей модели и данного ансамбля Y) будет связан тогда программный ансамбль P и множество $U \subset P \times Y$, состоящее из тех пар $\langle p, y \rangle$, для которых y принадлежит множеству, порождаемому исчислением с программой p . Это множество (которое естественно назвать результатным множеством данного способа программирования исчислений) оказывается породимым; исчисления, его порождающие, естественно называть универсальными исчислениями для данного способа программирования. Нетрудно сформулировать аналоги (У'), (Г'), (О') требований (У), (Г) и (О) для произвольного перечислимого подмножества $V \subset E \times Y$; как и в случае алгоритмов, множества, соответствующие известным способам программирования, удовлетворяют требованиям (У') и (Г').

Понятие универсального исчисления можно понимать и в более широком смысле, введя некоторую кодировку пар. Именно, пусть задан некоторый способ кодирования пар φ , сопоставляющий с каждой парой $\langle e, y \rangle$ некоторый элемент ансамбля кодов Z. Тогда с каждым исчислением, порождающим объекты ансамбля Z , будет связано множество $V \subset E \times Y$, состоящее из тех пар $\langle e, y \rangle$, код $\varphi(\langle e, y \rangle)$ которых порождается этим исчислением. Если это множество обладает свойством (У') (то есть для всякого породимого множества S объектов ансамбля Y существует такой объект $e \in E$, что свойства $y \in S$ и $\langle e, y \rangle \in V$ равносильны), то исходное исчисление можно назвать универсальным в широком смысле. Многие логистические системы (исчисление предикатов, формальная арифметика, аксиоматическая теория множеств) оказываются универсальными исчислениями в таком широком понимании. Эти универсальные исчисления появились в математике до формирования общего понятия исчисления. Поскольку всю (дескриптивную) теорию исчислений допустимо трактовать как теорию какого-нибудь одного универсального исчисления (сравни с трактовкой в § 4 теории алгоритмов как теории одного универсального алгоритма), можно считать, что уточнения понятия исчисления и общая теория исчислений возникли раньше, чем общее понятие исчисления. В этом – своеобразие развития понятия исчисления, отличающее это развитие от развития понятия алгоритма (ведь все алгоритмы, возникшие в математике до создания общей теории алгорит-

мов, ни в каком смысле не могут рассматриваться как универсальные!).

Отметим теперь связь между способами программирования алгоритмов и способами программирования исчислений. Пусть задан способ программирования алгоритмов с входным ансамблем X и выходным ансамблем Y, а также способ программирования исчислений с ансамблем $X \times Y$. Тогда каждая вычислимая функция из X в Y имеет как вычислительные программы, так и (рассматриваемая как породимое подмножество множества $X \times Y$) порождающие программы. Возникает вопрос: можно ли эффективно переходить от программ одного типа к программам другого типа (для той же функции)? Ответ, как и следовало ожидать, оказывается положительным - такая возможность обеспечивается свойством (Г), справедливым для способа программирования алгоритмов, и его аналогом (Г'), справедливым для способа программирования исчислений.

Введение понятия способа программирования (а на абстрактном уровне - понятия гёделевой универсальной функции и соответствующего понятия для множеств) позволяет сформулировать ряд важных результатов теории алгоритмов. Приведем сейчас некоторые из них.

Вторая теорема о рекурсии, принадлежащая Клини (см. [Родж 67, гл. 11]), утверждает, что невозможен алгоритм, преобразующий всякую программу вычислимой функции в программу совершенно другой (т.е. отличной от исходной) вычислимой функции; аналогичное утверждение справедливо для программ перечислимых множеств. (На самом деле справедливо даже несколько более сильное утверждение, позволяющее "эффективно" находить для всякого преобразующего программы алгоритма ту программу, которая переводится им в программу той же самой функции.)

Естественно задаться вопросом, какие свойства вычислимых функций (или породимых множеств) можно алгоритмически распознать по их программам; оказалось, что никакие, кроме тривиальных (см. [Райс 53], [Усп 55а], [Усп 60, § 11.2]). Замечательно, что этот факт оказался следствием топологической связности системы всех породимых множеств и системы всех вычислимых функций при той естественной топологии, которая упоминалась в § 13. В самом деле, множество всех программ множеств (или

функций), удовлетворяющих (а также не удовлетворяющих) какому-либо свойству, алгоритмически распознаваемому по программам, - перечислимо; с другой стороны, любая совокупность множеств или функций, для которой все программы членов этой совокупности образуют перечислимое множество (такая совокупность называется вполне перечислимой), оказывается открытой в указанной топологии (см. [Усп 55а]).

Пусть фиксирован некоторый способ программирования исчислений. Тогда каждая вычислимая операция Φ очевидным образом приводит к алгоритму, переводящему каждую программу порождения какого-либо множества A в некоторую программу порождения результирующего множества $\Phi(A)$. Теорема о том, что любая вполне перечислимая совокупность множеств открыта в топологии из § 13, служит также для установления следующего принципиального результата (обратного к упомянутому в начале этого абзаца очевидному факту): любой алгоритм, переводящий всякую программу в программу же, причем так, что программы одного и того же переводятся в программы одного и того же, соответствует некоторой вычислимой операции (см. [Усп 55а], [Май Шеп 55]). Более точно, пусть X и Y - перечислимые множества, F - (частичное) отображение из $Gen(X)$ в $Gen(Y)$, задаваемое (в разъясняемом ниже смысле) вычислимой функцией φ на программах: если P - программа для $A \in Gen(X)$ и $F(A)$ не определено, то $\Phi(P)$ не определено; если $F(A)$ определено, то $\Phi(P)$ определено и представляет собой программу для $F(A)$. В этом случае отображение F может быть продолжено до вычислимой операции, отображающей 2^X в 2^Y.

То же самое верно и для частичных отображений из $Com(X,Y)$ в $Com(U,V)$, программ вычислимых функций и вычислимых операторов. Сходная теорема верна и для алгоритмов, которые применимы к любой программе всюду определенной функции и дают в качестве результата программу другой всюду определенной функции, зависящей только от первой (но не от того, какая из ее программ взята): любому такому алгоритму соответствует вычислимый оператор (см. [Цей 62, теорема 2]).

§ 15. ПОНЯТИЕ НУМЕРАЦИИ И ТЕОРИЯ НУМЕРАЦИЙ

Нумерацией (или более точно, числовой нумерацией) множест-

ва M называется произвольное отображение α произвольного множества $E \subset \mathbb{N}$ на M ; если при этом $\alpha(e)=m$, то e называется (α-) номером элемента m (см. [Усп 55а], [Усп 60, § II], [Маль 6I, п. 2.I], [Маль 65, гл. IУ]).

Множество E называется основанием нумерации α (как в [Усп 60]) или номерным множеством (как в [Маль 6I], [Маль 65]) нумерации α . Если $E=\mathbb{N}$, нумерация называется натуральной, как в[Усп 60] или простой, как в [Маль 6I], [Маль 65, п. 9.I]. Иногда (например, в [Лавр 82]) термин "нумерация" употребляется как синоним термина "натуральная нумерация". Если каждый элемент имеет только один номер (т.е. α является взаимно однозначным соответствием), нумерация называется нумерацией без повторений или однозначной нумерацией (см. [Маль 6I], [Маль 65, п. 9.I]). Нумерация называется разрешимой, если существует алгоритм, который применим к любой паре элементов из E и дает ответ на вопрос, являются они или нет номерами одного и того же элемента из M (см. [Маль 6I]).

При естественном более широком понимании понятия нумерации основанием нумерации может служить любое подмножество любого ансамбля. Определения однозначности и разрешимости нумераций переносятся на общий случай без изменений. В роли натуральных нумераций выступают в этом случае нумерации, у которых основанием служит весь ансамбль. Такие нумерации называются тотальными.

Пусть нумерации α и β одного и того же множества имеют соответственные основания E и F ; они называются изоморфными (см.[Маль 6I, п. 2.I], [ЕршО 77, гл. 2, § I]), если для них существует изоморфизм, т.е. такое вычислимое взаимно-однозначное соответствиеf между E и F , что для всякого $e \in E$ выполнено $\alpha(e)=\beta(f(e))$. Говоря о вычислимости соответствия f , мы имеем в виду, что отображения f и f^{-1} являются сужениями некоторых вычислимых функций на множества E иF соответственно. Очевидно, всякая нумерация изоморфна некоторой числовой, а каждая тотальная нумерация изоморфна некоторой натуральной; поэтому, если пренебречь изоморфиями, можно ограничиться числовыми нумерациями; в частности, изучение тотальных нумераций можно заменить изучением натуральных нумераций.

Примеры нумераций:

1) Отображение, относящее каждому имени из некоторой совокупности имен его денотат (т.е. предмет, носящий это имя, см. [Чёрч 56, § 01]); это главная философская мотивировка теории нумераций.

2) Для фиксированного способа программирования отображение, относящее каждой программе задаваемую ею вычислимую функцию или задаваемое ею породимое множество (см. [Усп 56], [Усп 56а]); это главная математическая мотивировка теории нумераций.

3) Система обозначений для ординалов (см. [Родж 67, § 11.7]); это главная историческая мотивировка теории нумераций.

Все перечисленные примеры представляют собою, вообще говоря, не тотальные нумерации. Нумерация из второго примера может быть превращена в тотальную с помощью приема, указанного в начале § 14 и позволяющего каждый элемент программного ансамбля рассматривать как программу.

А л г е б р а и ч е с к и й п р и м е р . Каждый из вышеприведенных трех примеров дает, конечно, не какую-то одну нумерацию, а целую серию нумераций. Среди "философских" нумераций первой серии имеются нумерации, имеющие ясный алгебраический смысл. Речь идет о нумерациях конечнопорожденных алгебр. Пусть некоторая алгебраическая система конечно порождена, т.е. имеет конечное число образующих и конечную сигнатуру. Пусть алфавит Б содержит все имена $a_1, a_2, \ldots$ образующих, все имена $f, g, \ldots$ сигнатурных операций, левую и правую скобки и запятую. В ансамбле Б-слов возникает подмножество всевозможных замкнутых (т.е. не содержащих переменных) термов вида $g(a_5, f(a_2, a_1), a_5)$ и т.п. (ср. добавление к § 3). Каждый из таких термов обозначает некоторый (и притом только один) элемент нашей алгебры, причем разные термы могут обозначать один и тот же элемент. Тем не менее обычно говорят "элемент $f(a_2, a_1)$", а не "элемент, обозначенный через $f(a_2, a_1)$"; такой способ речи показывает, что мы используем выражение "$f(a_2, a_1)$" так, как используют имя объекта.

Итак, терм "$f(a_2, a_1)$" следует рассматривать как имя некоторого элемента алгебры (являющегося, в свою очередь, денотатом

этого терма); один и тот же элемент может иметь много имен. Поскольку $a_1, a_2, \ldots$ суть образующие, отображение, сопоставляющее с каждым термом его денотат, представляет собой сюръекцию, т.е. отображение на весь носитель рассматриваемой алгебраической системы. Таким образом, указанное отображение есть нумерация носителя; допуская вольность речи, об этой нумерации говорят как о нумерации самой алгебраической системы. Следуя Мальцеву (см. [Маль 61, п. 4.1]), эту нумерацию будем называть стандартной. (Строго говоря, в [Маль 61] стандартной называется не сама только что построенная нумерация, а некоторая изоморфная ей числовая нумерация.) Очевидно, что основание стандартной нумерации породимо (= перечислимо) и даже разрешимо.

Нумерация α называется позитивной, если породимо (= перечислимо) как ее основание E, так и подмножество $R \subset E^2$ всех таких пар $\langle e_1, e_2 \rangle$, для которых $\alpha(e_1) = \alpha(e_2)$ (см. [Маль 61, п. 2.1]).

А л г е б р а и ч е с к и й п р и м е р (п р о д о л ж е н и е). Пусть алгебраическая система не только конечно порождена, но и конечно задана, т.е. задана конечным числом квазитождеств. Как мы видели в добавлении к § 3, породимо множество всех таких пар $\langle t_1, t_2 \rangle$, что термы t_1 и t_2 равны в рассматриваемой системе, т.е. обозначают один и тот же элемент. Поэтому стандартная нумерация всякой конечно заданной алгебраической системы позитивна.

Пусть α и β – две нумерации множества M. Про функцию, которая по любому α-номеру любого элемента из M дает какой-то β-номер того же элемента, говорят, что она сводит α к β (см. [Усп 60, § 11], [Маль 61, п. 2.2]). Говорят, что α сводится по Колмогорову к β, если существует вычислимая функция, сводящая α к β. Наконец, две нумерации называются эквивалентными относительно колмогоровской сводимости, или, короче, эквивалентными по Колмогорову, если они сводятся друг к другу по Колмогорову.

В ситуации, когда не рассматривается иных видов сводимости и эквивалентности, слова "по Колмогорову" опускаются. Именно такая ситуация имеет место в настоящем параграфе, и потому мы будем говорить просто о сводимости и эквивалентности нумераций.

(В ч. II, § 5, нам предстоит рассмотреть и другие виды сводимости и эквивалентности.)

Отношение сводимости задает на совокупности всех (числовых) нумераций фиксированного множества M предпорядок. Тем самым на совокупности всех классов эквивалентности возникает частичный порядок. Частично упорядоченное множество классов эквивалентности оказывается при этом верхней полурешеткой. То же построение можно проделать только для натуральных нумераций. В этом случае мы также получим верхнюю полурешетку, которая, очевидно, изоморфно вложена в первую.

Идея абстрактного изучения нумераций была впервые высказана (в связи с изучением систем обозначений для ординалов) Колмогоровым в феврале 1954 г. на руководимом им семинаре по рекурсивной арифметике в Московском университете (именно, Колмогоров сформулировал общее понятие числовой нумерации и понятие сводимости нумераций). Эта идея получила развитие в исследованиях Мальцева (собранных впоследствии в [Маль 76]) и его ученика Ю.Л.Ершова. Монографии [Маль 65] и [ЕршО 77] подытоживают эти исследования.

Теорию нумераций можно считать новой самостоятельной областью математики, рожденной теорией алгоритмов. Самостоятельность этой новой области оправдывается наличием в ней глубоких математических результатов, как почти очевидных, так и совершенно неожиданных и нетривиальных. Вот пример результата первого типа: тотальная нумерация бесконечного множества является разрешимой тогда и только тогда, когда она эквивалентна некоторой однозначной тотальной нумерации того же множества (см. [ЕршО 77, гл. I, § 3]; мы сошлемся на этот результат ниже, в ч. II, § 5). А вот пример результата второго типа: для любых двух неодноэлементных конечных множеств верхняя полурешетка классов эквивалентных натуральных нумераций одного множества изоморфна верхней полурешетке классов эквивалентных натуральных нумераций другого множества (см. [ЕршО 77, приложение II]).

Наиболее разработана теория тотальных нумераций (ср. [ЕршО 77, с. 12]). При переходе к нетотальным нумерациям возникают новые эффекты: существует, например, разрешимая нумерация бесконечного множества, не эквивалентная никакой однознач-

ной нумерации (см. [Шэнь 81]). Как указано выше, для тотальных нумераций этого не бывает, поэтому нумерация с таким свойством не эквивалентна никакой тотальной. Другим примером нумераций, не эквивалентных тотальным, служат нумерации конструктивного континуума, рассматриваемые в ч. II, § 4. Основания этих нумераций неперечислимы: действительно, как легко видеть, всякая нумерация с перечислимым основанием эквивалентна тотальной.

Вычислимые нумерации

Большое количество понятий и результатов теории нумераций возникло из изучения нумераций, связанных со способами программирования (см. второй из приведенных примеров). Некоторые из этих понятий по существу уже обсуждались в предыдущем параграфе. Дело в том, что имеется естественное взаимно-однозначное соответствие между тотальными нумерациями семейств функций из X в Y с основанием E и функциями из $E \times X$ в Y. Именно, каждой функции F из $E \times X$ в Y соответствует нумерация, при которой объект $e \in E$ является номером функции $F_e : x \mapsto F(\langle e,x \rangle)$. В обратную сторону: каждой тотальной нумерации соответствует функция, относящая паре $\langle e,x \rangle$ значение функции с номером e на элементе x. Если данная функция и данная нумерация соответствуют друг другу при описанном только что взаимно-однозначном соответствии, то мы будем называть их сопряженными. Теперь мы можем перевести некоторые введенные в § 14 понятия на язык теории нумераций. Пусть X, Y и E - некоторые ансамбли.

(1) Тотальная нумерация α семейства функций из X в Y с основанием нумерации E называется вычислимой (см. [Усп 55а], [Лавр 77]), если сопряженная с ней функция, то есть функция, сопоставляющая паре $\langle e,x \rangle$ значение функции с номером e на элементе x, является вычислимой.

(2) Тотальная вычислимая нумерация семейства Com(X, Y) с основанием нумерации E называется главной (см.[Усп 55а], [Усп 60, § 11]), или гёделевой, если сопряженная с ней функция является гёделевой универсальной функцией; как легко видеть, это эквивалентно тому, что всякая вычислимая нумерация семейства Com(X,Y) сводится к ней.

(3) Пусть ансамбль E нормирован. Тогда имеет смысл говорить об оптимальных нумерациях. Именно, тотальная вычислимая

нумерация семейства $\mathrm{Com}(X,Y)$ с основанием нумерации E называется оптимальной, или шнорровой, или экономной по объему номеров, если сопряженная с ней функция является оптимальной гёделевой универсальной функцией; как легко видеть, это эквивалентно тому, что всякая вычислимая нумерация семейства $\mathrm{Com}(X, Y)$, основание которой является нормированным ансамблем, сводится к ней с помощью ограниченно-искажающего алгоритма.

Как видно из этих определений, всякая главная нумерация вычислима, а всякая шноррова нумерация - главная. Как отмечалось в § 14, оптимальные гёделевы универсальные функции существуют; тем самым существуют и шнорровы (и тем более главные и вычислимые) нумерации семейства $\mathrm{Com}(X, Y)$.

Распространим теперь введенные понятия на несколько более общий случай. Именно, дадим определение вычислимой нумерации для случая нумераций, не являющихся тотальными, а также определение главной нумерации для случая не обязательно тотальных нумераций не обязательно всего семейства $\mathrm{Com}(X,Y)$ (а лишь некоторого подмножества). Эти определения будут таковы:

(1') Нумерация α семейства функций из X в Y, основанием которой является подмножество E' некоторого ансамбля E, называется вычислимой, если E' породимо и функция из $E\times X$ в Y, ставящая в соответствие паре $\langle e,x\rangle$ значение функции с номером e на аргументе x в том случае, если $e\in E'$, и не ставящая в соответствие ничего, если $e\overline{\in}E'$, является вычислимой.

Для случая тотальных нумераций это определение, очевидно, совпадает с данным выше определением (1). Разница между определениями (1) и (1') не очень существенна: как легко видеть, всякая вычислимая нумерация эквивалентна некоторой тотальной вычислимой нумерации (ср. сказанное выше о нумерациях с перечислимым основанием).

(2') Вычислимая нумерация α некоторого семейства $\mathrm{ScCom}(X,Y)$ называется главной, или гёделевой, если всякая вычислимая нумерация этого семейства сводится к ней.

В силу сказанного выше об эквивалентности всякой вычислимой нумерации некоторой тотальной вычислимой нумерации, неважно, будем ли мы требовать сводимости к α всех вычислимых нумераций или только тотальных. Поэтому определение (2) являет-

ся частным случаем определения (2'). Семейство $S \subset Com(X,Y)$ может обладать вычислимой нумерацией, но не обладать главной: таково, например, семейство всех примитивно-рекурсивных функций из $\mathbb{N}$ в $\mathbb{N}$ (см. [ЕршО 77, гл. 1, § 2]).

Все сказанное выше может быть перенесено на случай нумераций семейств породимых множеств (вместо вычислимых функций). Приведем соответствующие формулировки. Пусть W и E - некоторые ансамбли. Нумерация α некоторого подмножества S множества Gen(W), основанием которой является подмножество $E' \subset E$, называется вычислимой, если множество E' и множество $\{\langle e,w\rangle \mid e \in E', w \in \alpha(e)\}$ породимы. Вычислимая нумерация α некоторого семейства $S \subset Gen(W)$ называется главной, или гёделевой, если всякая вычислимая нумерация этого семейства сводится к ней. Пусть теперь ансамбль E нормирован. Тотальная вычислимая нумерация семейства Gen(W), основанием которой служит E, называется оптимальной, или шнорровой, или экономной по объему номеров, если всякая вычислимая нумерация этого семейства, основание которой является нормированным ансамблем, сводится к ней с помощью ограниченно-искажающего вычислимого отображения.

Каждая вычислимая функция из X в Y может рассматриваться как породимое подмножество $X \times Y$. Поэтому нумерации семейств вычислимых функций могут рассматриваться также и как частный вид нумераций семейств породимых множеств и утверждение о вычислимости некоторой нумерации некоторого семейства вычислимых функций может пониматься двояко: либо в соответствии с данным выше определением (1), либо как утверждение о вычислимости нумерации соответствующего семейства породимых множеств. Легко проверить, что эти два понимания равносильны. Это замечание можно отнести к понятию главной нумерации семейства вычислимых функций: здесь также возможны два понимания, и они также равносильны. (К шнорровым нумерациям сказанное не относится, так как, говоря о шнорровости нумерации, мы предполагаем, что она является нумерацией всего Com(X,Y) или всего Gen(W).) Сформулируем теперь упоминавшиеся в § 14 теоремы Роджерса и Шнорра.

Теорема Роджерса. Для любых ансамблей X и Y гёделева нумерация множества Com(X,Y) единственна с точностью до изомор-

физма, осуществляемого вычислимым взаимно-однозначным соответствием между основаниями нумераций (см. [Родж 58], [Маль 63, теорема 7.I], [Маль 65, § 9, теорема 5]).

Аналогичное утверждение верно для гёделевых нумераций множества Gen(W) при любом ансамбле W.

Теорема Шнорра. Для любых ансамблей X и Y шнорrова нумерация множества Com(X,Y) единственна с точностью до изоморфизма, осуществляемого вычислимым взаимно-однозначным ограниченно-искажающим (в обе стороны) соответствием между основаниями нумераций (см. [Шно 72], [Шно 75]).

Аналогичное утверждение верно и для шнорровых нумераций множества Gen(W) при любом ансамбле W.

Заметим, что и в определении гёделевой нумерации, и в определении шнорровой нумерации, так же как и в теоремах Роджерса и Шнорра, на сложность вычисления функции, осуществляющей сведение нумераций, не накладывается никаких ограничений. Одно из самых естественных требований подобного рода состоит в принадлежности сводящей функции классу $\mathcal{P}$. Приняв его, мы приходим к такому определению.

(4) Пусть α есть тотальная вычислимая нумерация семейства Com(X, Y), основанием которой служит словарный ансамбль. Нумерация α называется полиномиально главной (или полиномиально гёделевой - "polynomial time Gödel " согласно [Мач Вин Янг 78]), если всякая нумерация семейства Com(X,Y) основанием которой служит словарный ансамбль, сводится к ней с помощью функции из класса $\mathcal{P}$.

Это понятие впервые приведено в работе [Хар Бей 75]. В этой же работе поставлена следующая проблема: имеет ли место аналог теорем Роджерса и Шнорра для полиномиально главных нумераций? Более точно, верно ли, что полиномиально главная нумерация единственна с точностью до изоморфизма, осуществляемого с помощью вычислимого взаимно однозначного отображения, лежащего вместе с обратным к нему в классе $\mathcal{P}$? (Существование полиномиально главной нумерации легко доказать.) Эта проблема остается открытой. В [Мач Вин Янг 78, теорема 2.6 (а)] показано, что если $\mathcal{P}=\mathcal{NP}$, то ответ на поставленный вопрос - утвердительный. Другой частичный результат в этом направлении получен в [Харт 82], где доказано, что все полиномиально главные

нумерации, обладающие некоторыми дополнительными (на наш взгляд, мало естественными) свойствами, могут быть получены одна из другой с помощью изоморфизма описанного выше типа.

Сформулируем теперь две серии результатов, типичных для теории вычислимых нумераций.

Результаты первой серии касаются существования и количества (рассматриваемых с точностью до эквивалентности) однозначных вычислимых нумераций того или иного семейства породимых множеств (в частности, вычислимых функций). Если семейство конечно, проблематика становится тривиальной. Для бесконечного семейства всякая его однозначная вычислимая нумерация эквивалентна (и даже изоморфна) однозначной вычислимой натуральной нумерации; поэтому при изучении нумераций с точностью до эквивалентности безразлично, рассматривать ли все нумерации или только натуральные. Исследования в этой области начались со статьи Фридберга [Фри 58], в которой были построены однозначные вычислимые нумерации семейств $Gen(W)$ и $Com(X,Y)$. (Такие нумерации, как нетрудно видеть, не могут быть главными.) Возник вопрос о числе однозначных вычислимых нумераций этих семейств (с точностью до эквивалентности). Сперва Пур-Эл установила, что для $Com(X,Y)$ это количество не менее двух (см. [Пур 64]). Затем Хутерецкий (см. [Хут 69]) обнаружил, что оно бесконечно - и для $Gen(W)$, и для $Com(X,Y)$. Оставался вопрос, каким это количество может быть вообще, т.е. для произвольных семейств породимых множеств и вычислимых функций. Известны были лишь примеры семейств, для которых это количество равно бесконечности, единице (такой пример строится без труда) и нулю (пример семейства, для которого вычислимые нумерации существуют, но среди них нет однозначных, можно найти, например, в [ЕршЮ 77, гл. I, § 6]). В 1972 г. Марченков (см. [Марч 72]) показал, что для семейства всюду определенных (определенных на всем X) функций число попарно неэквивалентных однозначных вычислимых нумераций может быть равно либо единице, либо бесконечности. Как установил Гончаров (см. [Гон 80], [Гон 80а]), существуют семейства породимых множеств и вычислимых функций, для которых число неэквивалентных однозначных вычислимых нумераций равно любому наперед заданному натуральному числу.

Вторая серия результатов связана с изучением алгебраического строения верхних полурешеток, которые образованы отношением сводимости на классах эквивалентных вычислимых нумераций различных семейств породимых множеств. Приведем наиболее общие результаты о таких полурешетках. Всякая непустая полурешетка вычислимых нумераций либо одноэлементна, либо бесконечна, см. [Хут 71, следствие 1]; если полурешетка вычислимых нумераций содержит более одного элемента, то она не является решеткой ([Сел 76, теорема 1]). Главные нумерации и только они представляют собой наибольшие элементы полурешеток вычислимых нумераций, поэтому наибольших элементов может и не быть: как уже отмечалось, семейство всех примитивнорекурсивных функций не имеет главной нумерации (более того, как доказано в [Марч 72, теорема 3], полурешетка всех (вычислимых) нумераций любого семейства общерекурсивных функций, содержащая более одного элемента, не имеет наибольшего элемента). Исследовались и минимальные элементы полурешеток. Исторически первыми примерами вычислимых нумераций, являющихся минимальными элементами соответствующих полурешеток, были однозначные вычислимые нумерации ряда семейств. Конструкции таких нумераций для основных семейств, например, для Gen(W) , оказались весьма нетривиальными (см. [Маль 65, п. 7.4]). В [Вью 73, следствие теоремы 1] построен пример семейства породимых множеств, полурешетка вычислимых нумераций которого непуста, но не имеет минимальных элементов. В [ЕршО Лав 73, замечание 1] найдены алгебраические инварианты, позволившие доказать неизоморфность полурешеток вычислимых нумераций для ряда естественных семейств: так, этим методом доказано, что попарно неизоморфны полурешетки вычислимых нумераций семейств $\{\emptyset, \{0\}\}$, $\{\emptyset,\{0\}, \{0,1\}\},\ldots$, $\{\emptyset, \{0\},\ldots,\{0,1,\ldots,n\}\},\ldots$ С другой стороны, в [Дени 78, следствие 3] доказано, что полурешетки вычислимых нумераций семейств $\{\emptyset, \{0\},\ldots,\{n\}\}$ и $\{\emptyset, \{0\},\ldots,\{m\}\}$ изоморфны для любых $m,n \in \mathbb{N}$.

Нумерованные множества

Множество, рассматриваемое вместе с какой-либо своей нумерацией, называется занумерованным, как в [Усп 55а], [Усп 60], или нумерованным, как в [Маль 61], [Маль 65], множеством; если нумерация натуральная, множество называется нату-

рально (за)нумерованным.

Наличие у множества M нумерации позволяет говорить о вычислимости функций из M^k в M и о разрешимости подмножеств M^s (относительно данной нумерации). Именно, функцию $f: M^k \to M$ естественно называть вычислимой относительно нумерации ν, если существует такая вычислимая функция φ, что

$$\nu(\varphi(e_1,\ldots,e_k)) \simeq f(\nu(e_1),\ldots,\nu(e_k))$$

для любых $e_1,\ldots,e_k \in E$, где E – основание нумерации ν. Разрешимость $P \subset M^s$ означает существование такой вычислимой функции χ, что для всех $e_1,\ldots,e_s \in E$:

$$\langle\nu(e_1),\ldots,\nu(e_s)\rangle \in P \Rightarrow \chi(e_1,\ldots,e_s) = 0$$
$$\langle\nu(e_1),\ldots,\nu(e_s)\rangle \notin P \Rightarrow \chi(e_1,\ldots,e_s) = 1$$

Натурально нумерованные множества образуют естественную категорию, в которой они выступают в роли объектов. Морфизмом из натурально нумерованного множества M_1 (с нумерацией α_1) в натурально нумерованное множество M_2 (с нумерацией α_2) называется отображение $\mu : M_1 \to M_2$, для которого существует всюду определенная на $\mathbb{N}$ вычислимая функция f, такая, что $\mu\alpha_1 = \alpha_2 f$. Ряд известных свойств нумерованных множеств удается сформулировать в теоретико-категорных терминах (см. [ЕршО 77, глава 2]).

Можно рассматривать и категорию, образованную всеми нумерованными, а не только натурально нумерованными множествами. При таком подходе морфизмами нужно, по-видимому, считать те отображения μ нумерованного множества M_1 с нумерацией α_1 в нумерованное множество M_2 с нумерацией α_2, для которых существует такая вычислимая функция f, для которой при всех x, принадлежащих основанию нумерации α_1, во-первых, определены значения $f(x)$ и $\alpha_2(f(x))$ и, во-вторых, выполнено равенство $\alpha_2(f(x)) = \mu(\alpha_1(x))$. Возникающая категория в литературе не изучалась, хотя, по мнению авторов, ее изучение не менее естественно, чем изучение категории натурально нумерованных множеств.

Существуют некоторые естественные операции, позволяющие получать из одних нумерованных множеств другие. К их числу относятся операции прямого произведения, сужения, кортежного распространения и факторизации, описанные в начале § 5 из ч. II.

§ 16. НАЧАЛО СОЗДАНИЯ ИНВАРИАНТНОЙ, ИЛИ МАШИННО-НЕЗАВИСИМОЙ ТЕОРИИ СЛОЖНОСТИ ВЫЧИСЛЕНИЙ

Под инвариантной теорией сложности понимается такая, результаты которой формулируются независимо от выбора той или иной вычислительной (или порождающей) модели. Здесь возможны три пути.

Первый - поиск таких широко понимаемых оценок сложности, которые не зависят от выбора вычислительной модели. В § 7 уже шла речь о том, что при определении класса $\mathcal{P}$ можно (без изменения класса) рассматривать разные вычислительные модели. По существу при определении класса $\mathcal{P}$ мы используем не одну верхнюю оценку, а класс верхних оценок, отличающихся друг от друга преобразованиями некоторого семейства. Более точно, рассмотрение оценок сложности с точностью до преобразования из данного семейства преобразований означает следующее. Сложностной класс задается некоторой системой оценок, причем для любых двух оценок α, β из рассматриваемой системы существуют такие преобразования U , V из фиксированного множества, что $\alpha \leq U \circ \beta$ и $\beta \leq V \circ \alpha$ (так, все невырожденные полиномы получаются один из другого возведением в ограниченную и отделенную от нуля степень). Функция считается принадлежащей к данному сложностному классу, если сложность ее вычисления может быть ограничена какой-нибудь оценкой из рассматриваемой системы. В соответствии со сказанным выше, $\mathcal{P}$ можно определить как класс функций, время вычисления которых линейно ограничено с точностью до возведения в ограниченную и отделенную от нуля степень.

Второй путь - попытаться какие-то параметры данного представителя вычислительной модели включить в качестве аргументов в сложностную функцию. Четкая и содержательная формализация этих параметров представляет значительные трудности, до сих пор в качестве таких параметров рассматривались число лент и мощность алфавита многоленточных машин Тьюринга (см., например, [Сей 77]).

Третий путь - аксиоматический. Блюм в [Блюм 67] предложил две аксиомы, которым удовлетворяет любая разумная сложность вычисления; в этих аксиомах понятие сложности вычисления формализовано в виде понятия меры сложности.

Для фиксированной главной (для X,Y с индексным множеством E) функции $V:E\times X\to Y$ мерой сложности называется вычислимая функция $C:E\times X\to\mathbb{N}$, удовлетворяющая следующим двум аксиомам (см. [Блюм 67]):

1) $V(i,x)$ определено $\Leftrightarrow$ $C(i,x)$ определено;

2) множество $\{\langle i,x,y\rangle \mid C(i,x)=y\}$ разрешимо.

Примерами мер сложности являются время и емкость вычислений на машинах Тьюринга, описанные в § 6, так же как и многие другие варианты понятий времени и емкости. В [Блюм 67] доказаны две замечательные теоремы о мерах сложности. Приведем их формулировки.

Теорема о рекурсивной связи различных мер сложности.

Пусть C_1, C_2 - две меры сложности (для одной и той же главной функции V). Тогда существует такая вычислимая функция $D: X\times\mathbb{N}\to\mathbb{N}$ с областью определения $X\times\mathbb{N}$, что для всех $i\in I$ неравенство $C_2(i,x)\le D(x,C_1(i,x))$ выполнено для всех x, для которых $V(i,x)$ определено, кроме конечного их числа.

Теорема об ускорении. Пусть C - мера сложности и пусть R - всюду определенная вычислимая функция из $\mathbb{N}$ в $\mathbb{N}$. Тогда существует такое разрешимое подмножество A множества X (характеристическая функция которого обозначается далее через χ_A), что для любого $i\in E$, для которого $V(i,x)=\chi_A(x)$ при всех x, существует такое j, что $V(j,x)=\chi_A(x)$ при всех x и для всех x, кроме конечного числа, имеет место неравенство $C(i,x)\ge R(C(j,x))$.

Теорию Блюма можно рассматривать как "дескриптивную часть" метрической теории алгоритмов. Действительно, понятия и методы подхода Блюма очень близки к классической дескриптивной теории алгоритмов.

Конечно, чрезвычайная общность аксиом Блюма влечет определенные неудобства: чем шире класс мер сложности, удовлетворяющих этим аксиомам, тем меньше можно о них доказать. Приведем пример одного из возможных дополнительных требований, предъявляемых к мерам сложности. Кажется естественным рассматривать в качестве сложностей только такие функции, сложность вычисления которых невелика, например, не превосходит значения самой сложностной функции. Можно (при $Y=\mathbb{N}$) добавить третью аксиому к аксиоматике сложности:

$$\forall i\, \exists j\, \forall x (C(i,x) \simeq V(j,x) \;\&\; V(j,x) \geq C(j,x)).$$

Такие сложности, как время и емкость для многоленточных машин Тьюринга, удовлетворяют этой аксиоме (и даже ее эффективному варианту, в котором j алгоритмически находится по i). В [Харт Хоп 71, § 4] мера сложности называется удобной ⟨proper⟩, если она удовлетворяет "эффективному" варианту третьей аксиомы.

§ 17. ТЕОРИЯ СЛОЖНОСТИ И ЭНТРОПИИ КОНСТРУКТИВНЫХ ОБЪЕКТОВ

Общий подход к понятию сложности конструктивного объекта как минимального объема описывающей этот объект программы принадлежит Колмогорову (см. [Колм 65]). (Независимо, хотя и в менее ясном виде, аналогичные идеи были высказаны Соломоновым в [Сол 64]). В ходе развития этого подхода выяснилось, что различным интуитивным представлениям о сложности соответствуют различные точные определения.

На интуитивном уровне указанные различия возникают по следующим причинам. Любой конструктивный объект (например, слово) можно рассматривать как сообщение о нем самом, а можно рассматривать как сообщение одновременно обо всех объектах, в каком-то смысле его содержащих (например, обо всех продолжениях слова). При втором подходе естественно считать, что программа, описывающая объект, может задавать не обязательно в точности его, а лишь какое-то его продолжение. Аналогичные соображения относительно соотношения "часть - целое" имеют место и для самих описаний объектов. В частности, если описание и некоторая его часть задают какие-то объекты, то, при некоторой точке зрения, эти объекты не могут "противоречить друг другу", они должны быть "согласованы". Таким образом, на рассматриваемом ансамбле должно быть задано отношение "согласованности". Пример отношения согласованности: "один из двух объектов - часть другого". Естественно предполагать, что отношение согласованности разрешимо.

В свете сказанного, в данном пункте термином "ансамбль" обозначается произвольный ансамбль, рассматриваемый вместе с заданным на нем произвольным разрешимым бинарным отношением, называемым отношением согласованности. Пусть X и Y - ансамбли.

Рассмотрим следующее условие, налагаемое на отношение R между элементами ансамблей X и Y :

(x и x'согласованы) & R(x,y) & R(x',y') ⇒ (y и y' согласованы).

Произвольное перечислимое отношение R между ансамблями X и Y, удовлетворяющее этому условию, называется способом описания (элементов из Y посредством элементов из X). Объект x называется описанием объекта y при способе R, если имеет место R(x,y).

В дальнейшем центральную роль будут играть ансамбли $\mathbb{N}$ и Ξ - ансамбли описаний. Здесь через $\mathbb{N}$ обозначен ансамбль натуральных чисел с отношением равенства в качестве отношения согласованности, а через Ξ - ансамбль всех слов в алфавите {0, 1} со следующим отношением согласованности: слова согласованы, если одно из них - начало другого. На этих ансамблях заданы нормы: в Ξ нормой слова считаем его длину, в $\mathbb{N}$ нормой числа x считаем целую часть числа $\log_2(x+1)$ (эти нормы мы рассматривали в § 6). Чтобы подчеркнуть, что мы рассматриваем не произвольные нормы на Ξ и $\mathbb{N}$, а некоторые конкретные, мы будем называть эти нормы объемами и обозначать буквой l.

Итак, пусть X - это один из ансамблей $\mathbb{N}$ или Ξ, пусть Y - произвольный ансамбль и пусть R - какой-либо способ описания. Сложностью $K_R(y)$ объекта y при способе описания R называется наименьший объем описания этого объекта (если описания не существует, то сложность равна ∞). Пусть, например, $X = \mathbb{N}$, $Y = \Xi$, а R состоит из всех пар вида $\langle x, y\rangle$, где x - программа какого-то нормального алгорифма $\mathcal{A}: \mathbb{N} \to \{0, 1\}$, а y - начальный отрезок последовательности $\mathcal{A}(0)$, $\mathcal{A}(1)$, $\mathcal{A}(2)$, ...; тогда K_R - это введенная Марковым (см. [Марк 64], [Марк 67]) сложность разрешения.

Как установил Колмогоров в [Колм 65], среди всех способов описания R (при фиксированных X и Y, у Колмогорова $X = Y = \mathbb{N}$) имеется оптимальный R_0, то есть такой, что для всякого способа R выполнено $K_{R_0} \leq K_R$. Для данных ансамблей X,Y сложность объекта y при произвольном фиксированном оптимальном способе описания называется энтропией K(y) этого объекта; для того, чтобы явно указать ансамбль X и Y, говорят об X-Y-энтропии. Таким образом, X-Y-энтропия есть отображение множества Y в

$\mathbb{N} \cup \{\infty\}$. Конечно, для данных X,Y существует много X-Y-энтропий. Однако все эти функции асимптотически эквивалентны: это значит, что для любых двух X - Y-энтропий K' и K" выполнено (асимптотическое) неравенство $|K'(y)-K''(y)| \lesssim O$. С точностью до этой эквивалентности X- Y-энтропия единственна. Фактически, само понятие энтропии определено с точностью до этой эквивалентности.

Основная лемма (почти очевидная). Для любой вычислимой функции f из $\mathbb{N}$ в $\mathbb{N}$ верно, что ($\mathbb{N}$-$\mathbb{N}$-энтропия f(n)) $\lesssim$ ($\mathbb{N}$ -$\mathbb{N}$-энтропия n). В общем случае, пусть U, V - произвольные ансамбли, E - способ описания элементов V посредством элементов U ; тогда неравенство (X -V -энтропия v) $\lesssim$ (X -U -энтропия u) имеет место для любых u , v , для которых выполнено E(u, v).

Наиболее изучен случай, когда не только X , но и Y выбирается из числа ансамблей $\mathbb{N}$, Ξ . При комбинации двух возможностей для X с двумя возможностями для Y получаются четыре энтропии: $\mathbb{N}$-$\mathbb{N}$ -энтропия, или простая колмогоровская энтропия (см. [Колм 65]), $\mathbb{N}$-Ξ -энтропия, или энтропия разрешения (см. [Зво Лев 70]), Ξ -Ξ -энтропия, или монотонная энтропия (см. [Лев 73]), Ξ -$\mathbb{N}$-энтропия, или префиксная энтропия (см. [Лев 76]).

Между этими энтропиями имеет место ряд соотношений. Для удобства их формулирования укажем в явном виде один ограниченно-искажающий изоморфизм между нормированными ансамблями $\langle \mathbb{N}, 1\rangle$ и $\langle \Xi, 1\rangle$. Изоморфизм этот таков: ноль $\leftrightarrow$ пустое слово, один $\leftrightarrow$ 0, два $\leftrightarrow$ 1, три $\leftrightarrow$ 00,... ; другими словами, числу соответствует слово, получаемое из двоичной записи x + 1 отбрасыванием начальной единицы. Соотношения между энтропиями мы предоставим в виде таблицы. Все рассматриваемые энтропии суть отображения в $\mathbb{N}$, их областями определения являются $\mathbb{N}$ или Ξ, однако, пользуясь построенным только что изоморфизмом, мы будем рассматривать все энтропии как отображения $\Xi \to \mathbb{N}$. Названия энтропии пишем перед K. Для любых двух функций f и g функция f стоит в таблице левее g тогда и только тогда, когда выполнено $f \lesssim g$.

NΞK	NNK ΞΞK	ΞNK 1	$1+1{,}5\log_2 l$

Как нетрудно проверить, все четыре указанные в таблице энтропии отличаются друг от друга не более чем на $C \log_2 l(x)$, где C - некоторая константа; эта оценка не может быть существенно улучшена: существует бесконечно много слов x, для которых $NNK(x) - \Xi\Xi K(x) \geq C_1 \log_2 l(x)$, а также бесконечно много слов y, для которых $\Xi\Xi K(y) - NNK(y) \geq C_1 \log_2 l(y)$, где C_1 - некоторая положительная константа. В качестве примера других свойств энтропий укажем на монотонность монотонной энтропии: если слово x является началом слова y, то $\Xi\Xi K(x) \leq \Xi\Xi K(y)$.

Одним из применений понятия энтропии является возможность "энтропийной" характеризации таких понятий-антиподов, как вычислимость и случайность. Интуитивно, вычислимая последовательность задается некоторым законом, т.е. сложность ее начальных отрезков ограничена. Если теперь эту интуитивную сложность понимать как монотонную энтропию, приведенные соображения переходят в следующую теорему:

Последовательность вычислима ⇔ монотонная энтропия ее начальных отрезков ограничена ⇔ энтропия разрешения ее начальных отрезков ограничена (см. [Зво Лэв 70, теорема 2.2]).

Интуитивно, случайная последовательность не обладает никакими закономерностями - сложность ее начальных отрезков максимальна. Понимая по-прежнему сложность как монотонную энтропию, можно рассматривать предыдущую фразу как определение случайности (см. об этом ч. II, § 6).

Было бы очень интересно ввести понятие энтропии с ограничениями на сложность вычислений - понятие, намеченное Колмогоровым в [Колм 65]; некоторые попытки такого рода имеются в [Шно 77].

Наряду с энтропией конструктивного объекта рассматривается условная энтропия одного объекта относительно другого. Для ее определения к ансамблям X и Y добавляется еще один ансамбль - ансамбль условий A. Фактически условная энтропия изучалась

только при $X=Y=A=\mathbb{N}$ (см. [Колм 65]). В этом случае способ условного описания R определяется как произвольное перечислимое отношение на $X \times Y \times A$, удовлетворяющее условию: $R(x,y,a) \,\&\, R(x,y',a) \Rightarrow y=y'$. Если R есть способ условного описания, можно ввести следующее понятие и обозначение - <u>условную сложность</u> $K_R(y|a)$ объекта y относительно объекта a при способе описания R. По определению $K_R(y|a)$ есть минимальный из объемов тех x, для которых выполнено $R(x,y,a)$. Среди всех способов условного описания имеется оптимальный, т.е. такой способ R_0, что для всякого способа R выполнено $K_{R_0} \preccurlyeq K_R$. В частности, оптимальным является любой способ R_0, полученный соотношением $R_0(x,y,a) \Leftrightarrow V(x,a)=y$, где V - некоторая оптимальная гёделева функция для ансамблей A, Y с индексным множеством X (см. § 14). Условная сложность при оптимальном способе описания называется <u>условной энтропией</u> и обозначается $K(y|a)$. Если в условной энтропии $K(y|a)$ произвольным образом фиксировать a, то получится простая колмогоровская энтропия, причем все простые колмогоровские энтропии могут быть получены из условных энтропий таким способом. Именно так и была введена Колмогоровым его простая энтропия в [Колм 65].

Уже имея в распоряжении понятие энтропии, можно с новой точки зрения посмотреть на свойство экономности (по норме, см. § 14). Естественно теперь понимать требование "p не содержит ничего лишнего по сравнению с i" так: (простая колмогоровская) энтропия p может превосходить энтропию i не более чем на константу, не зависящую от i. Оказывается, что при таком понимании всякая главная универсальная функция является экономной (теперь уже не по норме, а по энтропии). Получаем следующее свойство, которым обладает любая главная универсальная функция $V: E \times X \to Y$:

(ЭЭ) для каждого ансамбля I и каждого алгоритма $\mathcal{A}$: $I \times X \to Y$ существует такой алгоритм $\mathcal{B}: I \to E$ с областью определения I, что $\mathcal{A}(\langle i,x\rangle) \simeq V(\mathcal{B}(i),x)$ и $\mathcal{NEK}(\mathcal{B}(i)) \lesssim \mathcal{NIK}(i)$; здесь при определении энтропии мы считаем, что отношение согласованности на E и I совпадает с отношением равенства.

В частности, для любой известной вычислительной модели и любого известного способа программирования результатная функция является экономной по энтропии программ, т.е. обладает

свойством (ЭЭ) с программным ансамблем в качестве E .

Введение понятия энтропии слова позволяет по-новому взглянуть и на вопрос о сложности распознавания множеств. Сложность, в частности, время вычисления есть функция аргумента алгоритма. Можно считать, что один алгоритм работает дольше другого, если для временных функций выполняется соответствующее неравенство в каждой точке. При таком подходе, однако, может оказаться, что алгоритмы, предназначенные для распознавания двух таких множеств слов, которые получаются друг из друга переименованием букв, имеют несравнимые сложности. В то же время ясно, что эти алгоритмы, по существу, представляют собой один и тот же алгоритм. Чтобы расширить отношение равной сложности вычисления и на такие алгоритмы, слова группируют в классы по их длинам и строят новую сложностную функцию, переходя к максимуму в каждом классе. Однако и это не решает проблему. Скажем, если ко всем словам некоторого множества двоичных слов приписать в конце 100 нулей, или каждую букву в слове повторить дважды, может получиться множество, распознаваемое проще исходного. Можно, однако, поступить иначе: отнести к одному классу не все слова данной длины, а все слова данной энтропии с каким-либо фиксированным ограничением на сложность вычислений (скажем, рассматривать вычисления в реальное время). Это приведет к тому, что сложностные функции для множеств интуитивно одинаковой сложности окажутся близкими.

§ 18. УДОБНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МОДЕЛИ

Сейчас мы дадим единообразное описание некоторого класса вычислительных моделей. Для этого начнем с примера - многоленточных машин Тьюринга с входной и выходной лентами.

Непосредственное наблюдение показывает, что такая машина имеет управляющее устройство (процессор), которое может находиться в одном из конечного числа состояний, оно взаимодействует с информационными устройствами - входной, выходной и рабочей лентами. Взаимодействие происходит посредством отдачи приказаний и получения ответных сигналов. Набор приказаний для входной ленты таков: "Налево!", "Направо!", для рабочей ленты - вдобавок к этим двум - "Печатай a!" (для всех a из алфавита ленты); для выходной ленты - "Печатай a и направо!".

Сигналы от входной и рабочей лент таковы: "Вижу a " (для всех букв соответствующего алфавита); от выходной ленты сигналов не поступает.

Эта картина может быть обобщена и на другие типы вычислительных моделей. В частности, видно, что характер различных информационных устройств может быть совершенно различным. Скажем, входное устройство может быть лентой, а рабочая память (состояние рабочего устройства) - колмогоровским комплексом. Описанная схема наглядно отражает и выделение различных устройств в реальных ЭВМ. В эту схему можно внести - мы так и сделаем - и еще одну особенность реальных ЭВМ - наличие программы, подаваемой извне. Для этого нужно лишь добавить еще одно информационное устройство, аналогичное входному - программное.

Более формально, аналогично тому, как это сделано в [Пау Сей Вин 80] информационное устройство можно определить как набор, состоящий из множества состояний - подмножества некоторого ансамбля, алфавита приказаний, алфавита сигналов и функции преобразования информации, относящей всякой паре < приказание, состояние > пару <новое состояние, сигнал >. Теперь, определив управляющее устройство как конечный автомат с подходящим входным и выходным алфавитами, можно определить вычислительную модель, все представители которой различаются лишь управляющими устройствами, а входное, выходное, рабочее и программное устройства у них одинаковы. Для полного описания вычислительной модели нужно задать еще начальные состояния выходного и рабочего устройства, входную и выходную процедуры. Один алгоритм вычисления на данной модели определяется фиксацией управляющего устройства и (начального) состояния программного устройства.

Таким образом, одну вычислительную модель образуют,например, все машины Тьюринга, у которых имеется входная лента с алфавитом {0, 1}, программная лента с алфавитом {a,b,c,d,e, f,#} , две рабочих с алфавитом {a,b,1} , еще одна рабочая "плоская" лента с алфавитом {1,e,m} и выходная лента с алфавитом {0, 1}. В то же время если не фиксировать, скажем алфавит рабочей ленты для машин Тьюринга с одной рабочей лентой, то единой вычислительной модели - из рассматриваемого класса -

не получится. (Ранее мы понимали термин "вычислительная модель" иначе, разрешая и такие модели).

Время вычисления для всякого устройства описанного типа можно определить как число шагов вычисления. Емкость вычисления - как максимальный объем рабочей памяти, при условии, что на состояниях этой памяти введен объем. В аналогичной ситуации можно определить и объем программы.

Конечно, получающаяся модель не обязана быть представительной. Легко, например, построить модель в списанном сейчас смысле, состоящую из всех автоматов с магазинной памятью.

Всякой вычислительной модели рассматриваемого нами класса естественно соответствует порождающая модель, получающаяся переходом к недетерминированным управляющим устройствам.

Назовем вычислительную модель описываемого в этом параграфе типа (со входным, выходным, программным и рабочим устройствами) удобной, если на ансамбле, из которого берутся программы, задана норма и существует такое управляющее устройство (этой модели) A, что по всякому другому управляющему устройству B той же модели можно указать такую ограниченно-искажающую функцию h (транслятор), что:

а) результат применения A к $h(p)$ и x равен результату применения B к p и x или оба этих результата не определены;

б) время вычисления A на аргументах $h(p)$ и x не превосходит времени вычисления B на p и x, умноженного на константу, не зависящую от p, x;

в) емкость вычисления A на аргументах $h(p)$ и x не превосходит емкости вычисления B на p и x плюс константа, не зависящая от p, x.

Свойство (в), разумеется, предполагает, что определено понятие объема для состояний рабочего устройства.

Удобной вычислительной моделью оказываются многоленточные машины Тьюринга с фиксированным числом рабочих лент с фиксированными алфавитами (где нормой на словах служит норма, описанная в § 6, - длина, умноженная на логарифм числа букв алфавита), а также модель, получающаяся, если выбрать в качестве рабочего устройство, состояниями которого являются $(Б,k)$-комплексы или колмогоровские $Б$-комплексы над фиксированным алфавитом - при условии, что выбранная для измерения объема рабочей

памяти норма на комплексах удовлетворяет требованиям Д1 и Д2 из § 6. В качестве входного и выходного устройств при этом можно брать ленты или устройства, работающие с комплексами (над фиксированными алфавитами).

Часть вторая

ОСНОВНЫЕ МАТЕМАТИЧЕСКИЕ ПРИЛОЖЕНИЯ ТЕОРИИ АЛГОРИТМОВ

В части I, § 1, упоминались алгоритмы, изобретенные задолго до появления общего понятия алгоритма. Однако именно с появлением этого общего понятия связано возникновение теории алгоритмов. Действительно, чтобы постичь эту теорию, недостаточно уметь обращаться с конкретными примерами алгоритмов: необходимо воспринять общее понятие алгоритма.

Огромное число теорем, предполагающих построение тех или иных алгоритмов и встречающихся в различных областях математики, не требует для своего понимания общего понятия алгоритма. Поэтому эти теоремы не будут рассматриваться как принадлежащие приложениям теории алгоритмов. Построение конкретного алгоритма мы относим к той области математики или вычислительной практики, к которой принадлежит решаемая этим алгоритмом задача и методы которой используются при построении алгоритма (впрочем, в отдельных случаях этой областью может оказаться и сама теория алгоритмов). В противоположность этому, теорема о несуществовании алгоритма обращается к идее всего класса алгоритмов в целом, и, следовательно, является теоремой теории алгоритмов.

Подобным же образом мы не рассматриваем в качестве приложений теории алгоритмов оценки сложности конкретных алгоритмов, если только они не были получены для представительной вычислительной модели. Так, в этой части не будут рассматриваться теоремы об алгебраической сложности (заметим, что некоторые их следствия - например, некоторые теоремы о сложности вычисления на машинах Тьюринга - могут и принадлежать к приложениям теории алгоритмов).

Мы рассматриваем здесь только математические приложения теории алгоритмов, оставляя в стороне ее приложения, скажем к биологии (такие, как описание рефлексов в терминах относительных алгоритмов, трактовку генетического кода как программы, понимание макроэволюции как порождающего процесса - относительно этой последней темы см. [Мас 78]), к психологии (см. [Мас 79а]), к теории управления (при том, что ведущие специалисты этой теории проявляют все возрастающий интерес к весьма отвлеченным, на первый взгляд, концепциям теории алгоритмов - см. [Пет Ула Уль 79]), к языковедению (см. [Гла 77а], [Гла 82] [Манин 81]).

Основные математические приложения теории алгоритмов, которых мы здесь касаемся, таковы:

1. Исследование массовых проблем.

2. Приложения к основаниям математики: конструктивная семантика.

3. Приложения к математической логике: анализ формализованных языков логики и арифметики.

4. Вычислимый анализ.

5. Нумерованные структуры.

6. Приложения к теории вероятностей: определения случайной последовательности.

7. Приложения к теории информации: алгоритмический подход к понятию количества информации.

8. Оценки сложностей решения отдельных задач.

9. Влияние теории алгоритмов на алгоритмическую практику.

Соответственно этому перечню, часть II состоит из 9 параграфов.

Имеются, впрочем, приложения, не подпадающие под указанную рубрикацию. С одного такого примера мы и начнем. В части I, § 9, был отмечен следующий результат общей теории алгоритмов: произвольное перечислимое множество натуральных чисел представимо в виде множества всех натуральных значений подходящего многочлена. Частные случаи этого утверждения (для конкретных перечислимых множеств) могут рассматриваться как факты теории чисел и некоторые из них оказались неожиданными для специалистов в этой области математики. Среди таких частных случаев - существование многочлена с целыми коэффициентами,

множество натуральных значений которого (при натуральных значениях переменных) совпадает со множеством всех простых чисел. Один такой многочлен приведен в [Дей Мат Роб 76] и его запись занимает лишь несколько строк. В той же работе отмечается, что многие классические проблемы теории чисел, такие как проблема Ферма, проблема Гольдбаха и гипотеза Римана, могут быть переформулированы как проблемы существования решения у подходящего диофантова уравнения.

§ 1. ИССЛЕДОВАНИЕ МАССОВЫХ ПРОБЛЕМ

Кричит философ: "Дай мне
изобрести неразрешимые
проблемы".

Эразм Роттердамский,
"Домашние беседы", гл. XVII

Основные понятия

Алгоритмическая проблема - это проблема построения алгоритма с заданными свойствами (например, алгоритма, перечисляющего данное множество или алгоритма с данной оценкой сложности, решающего данную задачу). Частным случаем алгоритмических проблем являются алгоритмические массовые проблемы (таковыми не являются оба только что приведенных примера). (Алгоритмические массовые проблемы иногда называются просто "алгоритмическими проблемами", как, например, в [Адян 77], или просто "массовыми проблемами", как, например, в [Адян 82]). Понятие алгоритмической массовой проблемы возникло на основе рассмотрения массовых проблем. Массовые проблемы образуют основное поле приложения теории алгоритмов; более того, именно они и вызвали к жизни само понятие алгоритма.

Единичная проблема состоит в требовании предъявить объект, удовлетворяющий определенным условиям и называемый решением проблемы; решить проблему - значит указать такой объект; если решение существует, т.е. если проблему можно решить, проблема называется решимой (читатель заметит, что мы употребляем словосочетание "решимая проблема" вместо более традиционного, но менее точного словосочетания "разрешимая проблема"). Массовая проблема представляет собой серию (как правило, бесконечную)

единичных проблем и состоит в требовании решить все эти проблемы; понимание того, что же именно считается решением массовой проблемы, нуждается, разумеется, в уточнении. Пример единичной проблемы: для уравнения $x^2-x-1=0$ найти рациональное приближение к его отрицательному корню с точностью 10^{-6}. Пример массовой проблемы: в той же ситуации для любого n найти рациональное приближение с точностью 10^{-n}.

Другой пример единичной проблемы - проблема разрешения для множества A, расположенного в некотором ансамбле W; проблема состоит в требовании указать разрешающий алгоритм для A; существование решения ("решаемость", "решимость") этой проблемы, конечно, эквивалентно разрешимости A (ср.часть I, § 7). Дальнейший пример (единичной проблемы) - проблема разрешимости для множества A; это есть требование дать ответ ("да", "нет") на вопрос: "разрешимо ли A?" Другой пример массовой проблемы - массовая проблема разрешимости для Gen (W) и для фиксированного способа программирования породимых подмножеств ансамбля W; проблема состоит в том, чтобы обеспечить ответ на каждый вопрос: "Является ли подмножество с программой порождения p разрешимым?" К сожалению, термины (а потому и понятия) "проблема разрешения", "проблема разрешимости", "массовая проблема разрешимости" часто смешиваются друг с другом. То же самое верно для следующей тройки терминов (и понятий): "проблема отделения", "проблема отделимости", "массовая проблема отделимости". Такое смешение тем более прискорбно, что в ряде случаев четкое различение существенно проясняет ситуацию, как это видно на примере обсуждаемых в конце этого параграфа результатов из [Муч 65].

Приведем теперь необходимые определения, связанные с задачей отделения. Пусть A,B суть подмножества ансамбля W. Отделяющая функция для пары $\langle A,B\rangle$ - это всюду определенное отображение W в {"да", "нет"}, принимающее значение "да" на всех элементах A и значение "нет" на всех элементах B.

Проблема отделения для пары $\langle A,B\rangle$ состоит в нахождении алгоритма, вычисляющего отделяющую функцию для этой пары. Множества A,B называются отделимыми, если такой алгоритм существует. Единичная проблема отделимости для пары множеств $\langle A,B\rangle$ - это проблема получить ответ ("да", "нет") на вопрос:

"Отделимы ли A,B?" Массовая проблема отделимости для Gen(W) и для фиксированного способа программирования состоит в том, чтобы обеспечить ответ на любой вопрос (для любой пары порождающих программ): "являются ли породимые множества с порождающими программами p_1 и p_2 отделимыми?"

Понятие массовой проблемы несколько расплывчато, и естественно попытаться найти его формальный эквивалент. Обычный способ нахождения такого эквивалента заключается во введении понятия алгоритмической массовой проблемы. Чтобы задать алгоритмическую массовую проблему, следует указать

1) породимое множество X (множество вопросов, или единичных проблем),

2) породимое множество Y (множество ответов, или единичных решений),

3) подмножество $E \subset X$ (ограничение на вопросы),

4) подмножество $R \subset X \times Y$ (отношение "вопрос - ответ", или вопросно-ответное отношение). Тогда проблема состоит в требовании найти алгоритм из X в Y , преобразующий каждый вопрос $\alpha \in E$ в ответ $\beta \in Y$ со свойством $\langle\alpha,\beta\rangle \in R$. В приведенном выше примере с квадратным уравнением:

$X = \mathbb{N}$, $Y = \mathbb{Q}$, $E = X$, $R = \{\langle n, r\rangle \mid |r - x_0| < 10^{-n}$, где x_0 - требуемый корень $\}$. Другой пример: $X = \mathbb{N}^+ \times \mathbb{N}^3$, $Y = \mathbb{Q}$, $E = \{\langle a,b,c,n\rangle \mid b^2 - 4ac \geq 0\}$, $R = \{\langle\langle a,b,c,n\rangle, r\rangle \mid |r - x_0| < 10^{-n}$, где x_0 - наименьший корень уравнения $ax^2 + bx + c = 0\}$.

Замена массовой проблемы на соответствующую алгоритмическую массовую проблему позволяет представить неясное понятие в виде точного определения. В частности, возникают точные понятия алгоритмических массовых проблем разрешимости и отделимости. Кроме того, указанная замена превращает массовую проблему в единичную. Действительно, алгоритмическая массовая проблема заключается в требовании представить единичный объект (= решение), а именно, алгоритм (ср. проблемы разрешения и отделения выше).

Разумеется, наличие решения у каждой единичной проблемы из составляющих данную массовую проблему, т.е. наличие для каждого α из E такого β, что $\langle\alpha,\beta\rangle \in R$, еще не означает наличия решения у соответствующей алгоритмической массовой проб-

лемы. Например, если фиксирован способ программирования, то для каждой программы вычислимой функции существует программа той же функции, имеющая минимальный объем; однако не существует алгоритма, дающего по каждой программе эквивалентную программу минимального объема.

Вот три характерных примера из [Мат 74]:

Пример 1 (см. также [Мат 74а]). Еще в 1908 г. Туэ доказал, что для каждой неприводимой бинарной формы F не менее чем третьей степени с целыми коэффициентами справедливо утверждение (все переменные - целочисленные):

$$\forall\alpha\ \exists\beta\forall x\forall y[F(x, y) = \alpha \Rightarrow |x| + |y| < \beta].$$

Понадобилось, однако, 60 лет, чтобы доказать наличие алгоритма, дающего β по F и α (см. [Бей 68]).

Пример 2 (см. также [Дей Мат Роб 76]). Имеет место теорема Рота (см. [Рот 55]):

$$\forall\,\theta\,\forall r\ \exists s\forall p\forall q[q > s \Rightarrow |\theta - \tfrac{p}{q}| > q^{-2-r}],$$

где θ - алгебраическое, r - положительное рациональное, p,q,s - целые числа; в то же время не известно никакого способа для нахождения s по θ и r.

Пример 3. Матиясевич в [Мат 74а] построил полином A с целочисленными коэффициентами, для которого

$$\forall\alpha\ \exists\beta\forall y\forall z_1\ldots\forall z_n[A(\alpha, z_1,\ldots,z_n) = y + 4^y \Rightarrow y + z_1 + \ldots + z_n \leq \beta]$$

(здесь $\alpha,\beta,y,z_1,\ldots,z_n$ - натуральные числа), но невозможен алгоритм получения β по α.

Каждая проблема разрешения является алгоритмической массовой проблемой. Действительно, пусть W - ансамбль и $A \subset W$. Тогда проблема разрешения для A имеет (как алгоритмическая массовая проблема) W в качестве множества вопросов и то же W в качестве ограничения на вопросы; множество ответов - это {"да","нет"}, и вопросно-ответное отношение - это множество $\{<w, \text{"да"}>| w\in A\} \cup \{<w, \text{"нет"}>| w \in W\setminus A\}$.

Многие алгоритмические массовые проблемы могут быть переформулированы как проблемы разрешения или сведены к таким проблемам (а именно, к проблеме разрешения для области определения подходящей функции или к проблеме разрешения для самой функции, рассматриваемой как множество).

Фиксируем, к примеру, ансамбль W , некоторую W - представительную порождающую модель и ее ансамбль программ, скажем, P .

Тогда алгоритмическая массовая проблема разрешимости для Gen (W) есть проблема разрешения для некоторого подмножества P ,а именно, для множества {p|p является программой разрешимого множества }. Алгоритмическая массовая проблема отделимости для Gen(W)есть проблема разрешения для некоторого подмножества множества P×P , а именно, для множества {$<p_1, p_2>$| множества, породимые посредством программ p_1 и p_2 , отделимы}.

Однако подлинная значимость проблем разрешения определяется их гносеологическим аспектом: это суть проблемы распознавания свойств. Центральной, хотя заведомо нерешимой проблемой разрешения является такая проблема: по произвольному математическому утверждению установить, верно оно или нет.

Как уже было отмечено в преамбуле к этой части, построение алгоритма, служащего решением той или иной алгоритмической массовой проблемы, не является приложением общей теории алгоритмов, а принадлежит той области математики, к которой относится рассматриваемая массовая проблема. Математика полна таких алгоритмов. С другой стороны, если такого алгоритма не существует, доказательство его несуществования принадлежит приложениям общей теории алгоритмов. Многие такие теоремы несуществования были доказаны для проблем разрешения, поставленных, как правило, для породимых множеств (этот эффект разъясняется в ч. I, § 10). Вот простой пример проблемы разрешения (он принадлежит Чёрчу, см. [Чёрч 36]): по произвольному натуральному n определить, существуют ли такие положительные натуральные x,y,z , что $z^n=x^n+y^n$; относительно разрешимости этой проблемы ничего не известно; очевидно лишь, что множество всех тех n , для которых ответ - "да , существуют", породимо.

Первые доказательства нерешимости алгоритмических массовых проблем относятся к 1936 г.; они были опубликованы Чёрчем и Тьюрингом в [Чёрч 36], [Чёрч 36а], [Чёрч 36б] и в [Тью 36], [Тью 37]. Эти нерешаемые проблемы являются проблемами разрешения и связаны с предложенными этими авторами представительными порождающими (в виде λ-конверсии Чёрча) и вычислительными (в виде машин Тьюринга) моделями; однако уже Чёрч заме-

тил (и об этом говорит само название его публикации [Чёрч 36]), что тем самым возникает нерешимая массовая проблема внутри элементарной теории чисел.

Семь нерешимых проблем

Из числа наиболее замечательных нерешимых проблем разрешения мы выделяем следующие семь по принципу философской значимости или наглядности формулировки.

1. Проблема распознавания истинности формул элементарной арифметики. Эти формулы строятся с помощью арифметических действий (сложения и умножения), логических операций (логических связок и кванторов) и знака равенства из константы 0 и натуральнозначных переменных. Проблема состоит в требовании найти алгоритм, который по всякой такой формуле определял бы, истинна она на натуральном ряду или нет. Невозможность такого алгоритма немедленно следует из существования неразрешимого перечислимого множества натуральных чисел (ч. I, § 10) и того, что такое множество является арифметическим (ч. I, § 9).

2. Проблема разрешения для логики предикатов первого порядка <das Entscheidungs-problem> - см. ниже, § 3.

3. Проблема сочетаемости Поста. Пусть дано конечное множество V пар слов в некотором алфавите. Назовем это множество "сочетаемым", если для некоторых пар $\langle A_1, B_1\rangle$, $\langle A_2, B_2\rangle$, ...$\langle A_s, B_s\rangle$ из V выполняется равенство $A_1 \ldots A_s = B_1 \ldots B_s$. Требуется по всякому конечному множеству V пар слов в данном алфавите алгоритмически установить, сочетаемо оно или нет. Эту проблему поставил Пост в [Пост 46] и в той же статье доказал, что соответствующего алгоритма не существует, если только алфавит содержит более одной буквы (в случае однобуквенного алфавита проблема решима). Усиление этого результата, фиксирующее число элементов V, см. в [Марк 54, гл. VI, § 9], где доказано, что алгоритма нет уже при числе элементов V, равном 90 (и любом неоднобуквенном алфавите).

4. Проблема эквивалентности слов для ассоциативного исчисления. Принадлежащее Туэ понятие ассоциативного исчисления (введенное в [Туэ 14]) было определено в добавлении к § 3. Пусть, в обозначениях этого добавления, фиксированы два кортежа A и B. В [Туэ 14, § II] формулируется следующая массовая проблема, названная там "задачей (I)": "найти метод, позволяю-

щий при помощи исчислимого <berechenbare> числа операций всегда решать, будут ли эквивалентны два произвольных ряда знаков". Другими словами, проблема - для данного ассоциативного исчисления - состоит в построении алгоритма, распознающего по всякой паре Б-слов, эквивалентны ли эти слова в данном исчислении или нет. В силу сказанного в добавлении к § 3, эта проблема совпадает с проблемой распознавания равенства слов в полугруппе с множеством образующих Б, заданной совокупностью равенств $A_j=B_j$, $j=1,\ldots,n$ (см. [Наг 77а]). Сам Туэ в своей задаче (I) молчаливо предполагает все элементы кортежей А и В непустыми словами. Однако отдельно (в [Туэ 14, § Ш)] Туэ ставит аналогичную проблему, - "задачу (П)" - для случая, когда А состоит из единственного слова R , а В - в современной терминологии - из пустого слова Λ. Эта последняя проблема оказалась решимой, см. [Адян 66, гл. Ш, § 3, теорема 3]. Существуют ассоциативные исчисления с нерешимой проблемой эквивалентности. Первые такие примеры построили в 1947 г. независимо Марков (см. [Марк 47], [Марк 47а], [Марк 54, гл. УI]) и Пост (см. [Пост 47]). Первоначальные примеры, однако, были довольно громоздкими. В [Цей 58] приведен следующий пример ассоциативного исчисления в пятибуквенном алфавите, проблема эквивалентности для которого не имеет решения (соответствующие друг другу элементы кортежей А и В соединены знаком равенства, как в задании полугруппы): 1) ac=ca , 2) ad=da , 3) bc=cb , 4) bd=db , 5) eca=ce , 6) edb=de , 7) cca=ccae . В [Мат 67] указано ассоциативное исчисление в двубуквенном алфавите с тремя определяющими соотношениями и нерешимой проблемой эквивалентности; в самом длинном из этих соотношений в левой части стоит слово из 304 букв, а в правой - слово из 608 букв. Для ассоциативных исчислений с двумя соотношениями мало что известно. Для ассоциативных исчислений (соответственно, полугрупп) с одним определяющим соотношением разрешающий алгоритм существует в широком классе случаев (см. [Адян 66], [Адян 0га 78]) и есть надежда, что он существует всегда. Можно ограничивать рассматриваемый класс ассоциативных исчислений и иным путем, нежели сокращая число определяющих соотношений. Особый интерес представляет случай, когда соответствующая данному ассоциативному исчислению полугруппа является группой. Частным случаем таких исчислений яв-

ляются инверсивные исчисления (см. [Наг 77в]; алфавит каждого такого исчисления имеет вид $a_1,a_2,\ldots,a_n,a_1^{-1},a_2^{-1},\ldots,a_n^{-1}$, и в число определяющих состояний обязательно входят, для каждого i, "тривиальные" соотношения $a_ia_i^{-1}=\Lambda$ и $\Lambda=a_i^{-1}a_i$). В 1952 г. Новиков (см. [Нов 52], [Нов 55]) построил первый пример инверсивного исчисления с нерешимой проблемой эквивалентности; этот пример дал ответ на первую из поставленных в 1912 г. М.Деном проблем - на проблему тождества для конечно-определенных групп (а именно, продемонстрировал невозможность решить эту проблему).

5. Проблема представимости матриц. По определению матрица U представима через матрицы $U_1,\ldots,U_q$, если для некоторых $r_1,\ldots,r_t$ выполняется равенство $U=\prod_{i=1}^{t}U_{r_i}$. Далее рассматриваются квадратные матрицы порядка n с целыми коэффициентами. Общая проблема представимости состоит в требовании указать алгоритм, посредством которого можно было бы узнавать для любой системы матриц $U,U_1,\ldots,U_q$ порядка n, представима ли U через $U_1,\ldots,U_q$. Как установил Марков (см. [Марк 58]), общая проблема представимости для матриц порядка n не имеет решения при $n\geq 4$ (в силу [Пат 70], см. ниже, число 4 можно заменить на 3). Частная проблема представимости для фиксированных матриц $U_1,\ldots,U_q$ состоит в требовании указать алгоритм, посредством которого можно было бы узнавать для любой матрицы U, представима ли она через $U_1,\ldots,U_q$. В [Марк 58] выписываются 27 матриц шестого порядка, для которых частная проблема представимости оказывается нерешимой. Другая частная проблема представимости матриц состоит в том, чтобы для фиксированной матрицы указать алгоритм, посредством которого можно было узнавать для любых матриц $U_1,\ldots,U_q$, представима ли U через $U_1,\ldots,U_q$. В [Пат 70] доказано, что эта частная проблема нерешима, если U есть нулевая матрица третьего порядка.

6. Десятая проблема Гильберта. Десятая из 23 проблем Гильберта (1900 г.) формулируется так: "10. Задача о разрешимости диофантова уравнения. Пусть задано диофантово уравнение с произвольными неизвестными и целыми рациональными числовыми коэффициентами. Указать способ, при помощи которого возможно после конечного числа операций установить, разрешимо ли это уравнение в целых рациональных числах. И..." ([Гиль 35, с.

310]). Невозможность решения десятой проблемы следует из сочетания двух факторов: а) существования неразрешимого перечислимого множества натуральных чисел (см. ч. I, § 10) и б)диофантовости всякого перечислимого числового множества (см. ч. I, § 9). Остается открытым вопрос, как узнать по произвольному диофантову уравнению, разрешимо ли оно в рациональных (не обязательно целых) числах - и можно ли это узнать вообще (т.е. возможен ли соответствующий алгоритм).

7. Проблема тождества элементарных функций вещественного переменного. Определим класс термов T индуктивно: x (переменная икс), π (число пи) - термы; если u,v - термы, то (u + v), (u·v),(u : v), sin u, |u| - термы (в [Усп Сем 81] в результате опечатки терм (u·v) пропущен). Проблема построения алгоритма, узнающего по двум термам из T, задают ли они одну и ту же функцию одного вещественного переменного x, нерешима (см. [Мат 73]). Можно сформулировать ряд других нерешимых проблем, касающихся функций вещественного переменного. Таковой является проблема существования вещественного корня у функции, задаваемой термом, аналогичным термам из T, но с использованием дополнительно произвольных рациональных констант и без использования деления и абсолютной величины, см. [Уанг 74]. Другим примером является проблема существования у функции, заданной термом из некоторого фиксированного класса, первообразной, задаваемой термом того же класса. Эта проблема нерешима для различных классов термов (см. [Рич 68]). Таким образом, уже обычное интегральное исчисление дает возможность получения нерешимых алгоритмических проблем.

Массовые проблемы в математике

Массовые проблемы возникают во всех областях математики. Наиболее фундаментальные из них - это проблемы выяснения истинности утверждений, формулируемых в некотором математическом языке. Реальные математические языки всегда допускают рассмотрение натуральных чисел с обычными операциями над ними (а как правило, еще и переменные по множествам и функциям), поэтому (см. первую из Семи нерешимых проблем) возникающие при этом множества истинных формул оказываются неразрешимыми. Таким образом, вопрос о разрешимости множества истинных формул неко-

торой логической теории оказывается нетривиальным, только если выразительные средства и объекты рассмотрения этой теории весьма ограничены. В частности, отказ от автоматически приводящего к неразрешимости использования предикатных и функциональных переменных фактически приводит к тому, что рассматриваемые массовые проблемы касаются алгебраических объектов. Для этого есть и другая причина: дело в том, что во многих важных случаях алгебраические объекты допускают естественное конструктивное задание, поэтому вопросы о них легко формулируются в алгоритмической форме.

Итак, нетривиальные результаты о разрешимости или неразрешимости логических теорий относятся, в основном, к элементарным теориям алгебраических систем. Наиболее тонкие из них касаются алгебраических систем, расположенных вблизи границы между разрешимостью и неразрешимостью, важнейшие факты относятся к ходу этой границы среди полей: неразрешима элементарная теория поля рациональных чисел (см. [Роб 49]), разрешима элементарная теория поля действительных чисел (см. [ЕршО 80, гл. 5, § 5, предложение 1]) и элементарная теория поля p-адических чисел (при любом p), см. [ЕршО 80, гл. 5, § 5, предложение 4]. Для других алгебраических систем возникающие элементарные теории, как правило, неразрешимы (так обстоит дело, в частности, для групп, колец и многих классов этих алгебраических систем, см. [Тар Мост Роб 53]); например, неразрешима элементарная теория класса всех простых конечных групп (см. [ЕршО 64а]). К немногим исключениям относятся абелевы группы и некоторые классы упорядоченных множеств (см. [ЕршО 64], [Раб 69]), а также произвольные свободные алгебры (это следует из [Маль 62, теорема 5]). Подробные таблицы, суммирующие результаты о неразрешимости и разрешимости элементарных теорий, имеются в [ЕршО Лав Тайм Тайц 65] и [Сан 78] (в последней работе приводятся также сведения о некоторых неэлементарных теориях).

Если мы хотим сформулировать алгоритмическую массовую проблему, касающуюся элементов какой-либо алгебраической системы, то необходимо сопоставить с этими элементами некоторые конструктивные объекты. Элементы фиксированной свободной конечно-

порожденной алгебраической системы можно считать уже лежащим в подходяще выбранном словарном ансамбле (или ансамбле деревьев, см. ч. I, § 0 и добавление к § 3). В общем случае, если какая-то (не обязательно свободная) алгебраическая система конечно порождена, т.е. имеет конечную сигнатуру и конечное число образующих, всякий ее элемент может быть опять-таки задан некоторым конструктивным объектом - выражением этого элемента через образующие (термом). При этом, однако, различные конструктивные объекты могут задавать один и тот же элемент. Возникает алгоритмическая проблема распознавания равенства, т.е., для заданной алгебраической системы, выяснения того, задают ли два произвольно выбранных конструктивных задания один и тот же элемент системы. Как уже отмечалось, для некоторых полугрупп и для некоторых групп эта проблема оказывается нерешимой - даже если ограничиться только конечноопределенными группами и полугруппами.

Ясно, что сама постановка алгоритмической проблемы распознавания равенства существенно зависит от того, в каком классе конструктивных объектов, задающих элементы фиксированной алгебраической системы, эта проблема ставится, т.е. для пар каких именно объектов ищется соответствующий алгоритм. Можно считать, что конструктивными заданиями элементов являются в с е в о з м о ж н ы е термы, составленные из образующих и сигнатурных операций. Однако не менее естественны и такие подходы, при которых конструктивными заданиями признаются только н е к о т о р ы е из термов.

Проиллюстрируем сказанное на примере группы с образующими a,b. Проблему распознавания равенства можно ставить при следующих вариантах понимания того, что есть конструктивное задание элемента группы.

1) Конструктивное задание есть произвольный терм, составленный из образующих, операции умножения, операции обращения и скобок, например, терм

$$(b(a^{-1}b))^{-1}(ba).$$

2) Конструктивное задание есть "бесскобочный терм", а именно, такой терм, в котором операция обращения применяется только к образующим, а все левые скобки собраны в начале терма и потому вместе с правыми опущены в записи. Пример:

$b^{-1}ab^{-1}ba$.

3) Конструктивное задание есть "несократимый терм" (=элемент свободной группы с образующими a,b), т.е. такой бесскобочный терм, в котором никакая образующая не стоит рядом со своим обращением. Пример: $b^{-1}aa$.

4) (Только для абелевых групп!) Конструктивное задание есть "упорядоченный терм" (= элемент свободной абелевой группы с образующими a,b), т.е. такой несократимый терм, в котором все буквы a идут впереди всех букв b. Пример: aab^{-1}.

Каждый из этих вариантов понимания конструктивного задания приводит к своему варианту алгоритмической проблемы распознавания равенства. В нашем теоретико-групповом примере существование решения у любого из вариантов проблемы приводит к решению для любого другого варианта.

Попытаемся дать математическую картину того, как обстоит дело с постановкой (только с постановкой!) алгоритмической проблемы равенства в общем виде, т.е. для произвольной конечнопорожденной алгебраической системы, т.е. системы, порожденной конечной сигнатурой σ. Можно считать, что элементами всякой такой системы $\mathfrak{L}$ являются классы подходящим образом выбранной конгруэнции $\mathfrak{k}$ на $\mathfrak{A}(\sigma)$, где $\mathfrak{A}(\sigma)$ - свободная алгебраическая система, порожденная сигнатурой σ (см. добавление к § 3). Два терма из $\mathfrak{A}(\sigma)$, принадлежащие одному и тому же классу конгруэнции, называются "равными в $\mathfrak{L}$". Абсолютная проблема распознавания равенства в $\mathfrak{L}$ состоит в отыскании алгоритма, распознающего по любым двум термам из $\mathfrak{A}(\sigma)$, равны ли они в $\mathfrak{L}$. Однако возможны и относительные проблемы распознавания. Чтобы поставить относительную проблему распознавания равенства в $\mathfrak{L}$, надо выбрать 1) некоторое подмножество $E \subset \mathfrak{A}(\sigma)$, элементы которого называются приведенными термами, и 2) некоторое вычислимое отображение h множества $\mathfrak{A}(\sigma)$ в E, называемое процедурой приведения, причем такое, что t и h(t) равны в $\mathfrak{L}$ для любого $t \in \mathfrak{A}(\sigma)$. (Например, в третьем варианте нашего теоретико-группового примера приведенные термы - это несократимые термы, а процедура приведения приводит каждый терм к несократимому виду.) Проблема распознавания равенства в $\mathfrak{L}$ относительно E и h состоит в отыскании алгоритма, распознающего равенство в $\mathfrak{L}$ по любым двум приведенным термам. При наложенных

на E и h требованиях существование решения у относительной проблемы равносильно существованию решения у абсолютной проблемы. На все сказанное можно посмотреть с точки зрения теории нумераций (см. ч. I, § 15). Задание элементов $\mathfrak{A}$ термами из $\mathfrak{T}(\sigma)$ есть не что иное, как стандартная нумерация системы $\mathfrak{A}$. Поэтому наличие решения у абсолютной проблемы распознавания равенства означает просто разрешимость этой нумерации. Далее, каждая относительная проблема, точнее, каждое указание E и h также приводит к некоторой нумерации ν системы $\mathfrak{A}$, а именно к нумерации с основанием E. Нумерация ν задается так: каждому терму $t \in E$ ставим в соответствие тот класс конгруэнции θ, в котором этот терм находится. В силу наложенных на E и h ограничений нумерация ν эквивалентна по Колмогорову стандартной нумерации (причем h сводит стандартную нумерацию к ν); поэтому они обе одновременно разрешимы или не разрешимы.

Проблема распознавания равенства (для данной алгебраической системы) является представителем большого класса проблем, состоящих в распознавании различных свойств элементов системы, пар таких элементов (как в случае распознавания равенства), троек и т.д. Разумно ограничиться свойствами, имеющими алгебраический смысл (= формулируемыми в алгебраических терминах). Что это такое, будет уточнено в § 5 при определении понятия "алгебраически корректной алгоритмической массовой проблемы".

Разумеется, можно интересоваться распознаванием свойств не только элементов какой-нибудь алгебраической системы, но и самих систем (первые свойства естественно называть "внутренними", а вторые - "внешними"). Для того, чтобы рассмотреть соответствующие массовые проблемы, нужно сопоставить конструктивные объекты уже не с элементами систем, а с самими системами. Это естественно делается в случае, когда алгебраическая система задается породимым (в частности, конечным) множеством соотношений (равенств, тождеств, квазитождеств, см. добавление к § 3). Именно задание конечным множеством определяющих соотношений (а конкретнее - равенств) мы и будем сейчас иметь в виду при обсуждении внешних свойств алгебраических систем.

Важнейшим примером внешнего свойства является изоморфия двух алгебраических систем. Для конечноопределенных групп проблема распознавания изоморфии (так называемая "третья проб-

лема Дена") была поставлена Деном наряду с проблемой распознавания внутреннего свойства - равенства двух элементов заданной группы. Нерешимость проблемы распознавания изоморфии конечноопределенных групп вытекает из конструкции, использованной при построении группы с нерешимой проблемой распознавания равенства. Набросок доказательства нерешимости проблемы изоморфии для групп содержится в [Нов 58], более подробно это доказательство изложено в [Адян 73]. Более того, для конечноопределенных групп нерешима и частная проблема изоморфии: для произвольной конечноопределенной группы F_0 невозможен алгоритм, распознающий по заданию произвольной конечноопределенной группы F, изоморфны F и F_0 или нет (см. [Адян 55], [Адян 56], [Адян 57]). И общая, и частная проблемы изоморфии принадлежат к числу проблем распознавания инвариантных (т.е. сохраняющихся при изоморфизмах) свойств алгебраических систем и их кортежей. Вот еще инвариантные свойства групп: "быть абелевой", "быть конечной", "быть циклической", "быть простой". Для каждого из этих свойств проблема его распознавания по заданию конечноопределенной группы нерешима (см. [Адян 55], [Адян 57]). Все перечисленные свойства групп являются марковскими; определение сейчас будет дано. Инвариантное свойство полугрупп (соответственно, групп) α называется марковским, если, во-первых, имеется конечноопределенная полугруппа (группа), обладающая свойством α, и, во-вторых, имеется конечноопределенная полугруппа (группа), не вложимая ни в какую конечноопределенную полугруппу (группу) со свойством α. Каково бы ни было марковское свойство, алгоритмическая проблема его распознавания нерешима. Для полугрупп это установил Марков (см. [Марк 52, п. II.1], [Марк 54, гл. VI, § 11]), для групп - Адян (см. [Адян 57а], [Адян 57б], [Адян 58]). Одно из следствий этого результата Адяна устанавливает следующую дополнительность между внутренними и внешними свойствами конечноопределенных групп: пусть K - непустой замкнутый относительно изоморфизмов класс конечноопределенных групп, тогда в классе K имеется группа с нерешимой проблемой распознавания равенства или проблема принадлежности конечноопределенной группы классу K нерешима (см. [Адян 73]). Несмотря на обилие нерешимых свойств конечноопределенных групп, для некоторых естест-

венных свойств существует разрешающий алгоритм, примером такого (внешнего) свойства является "совпадать со своим коммутантом" (см. [Адян 57а]).

Отметим, наконец, еще один класс проблем, лежащий в "произведении" внешних и внутренних проблем - это так называемые общие (другими словами, равномерные) проблемы. Общая проблема ставится для некоторого класса алгебраических систем и состоит в отыскании общего метода решения некоторой внутренней проблемы для всех систем этого класса, т.е. алгоритма, который по заданию системы и исходному данному внутренней проблемы, относящейся к этой системе, давал бы ответ на внутреннюю проблему. Пример: общая проблема распознавания равенства для групп состоит в том, чтобы по заданию всякой группы и паре слов в алфавите образующих этой группы, определить, равны ли эти слова в этой группе или нет. Разумеется, если некоторая внутренняя проблема нерешима для одной алгебраической системы заданного класса, то соответствующая общая проблема также нерешима; для большинства общих проблем дело обстоит именно так. Однако существуют естественные проблемы, для которых ситуация иная. Именно, рассмотрим следующую внутреннюю проблему: "По всякому элементу x полугруппы с единицей узнать, существует ли такой элемент y, что xy равно в полугруппе единице". Эта проблема решима для каждой конечноопределенной полугруппы с сокращением, однако общая проблема в классе всех конечноопределенных полугрупп с сокращением нерешима (см. [Адян 55а]). О других алгоритмических проблемах алгебры см. [Адян 73], [Адян 77].

Есть еще одна область, в которой с математическими объектами изучения естественно сопоставляются их конструктивные описания. Это - комбинаторная топология, где с полиэдрами (в частности, триангулируемыми многообразиями) сопоставляются записи их триангуляций. Коль скоро такое сопоставление осуществлено, делается осмысленной проблема гомеоморфии, аналогичная проблеме распознавания равенства слов в группе или проблеме изоморфии групп. Согласно [Марк 58а], общая проблема гомеоморфии для полиэдров (соответственно, для триангулируемых многообразий) есть проблема разыскания алгоритма, распознающего для любых двух полиэдров (соответственно многообразий), задан-

ных своими триангуляциями, гомеоморфны ли они; частные проблемы гомеоморфии возникают, если наложить на рассматриваемую пару полиэдров какое-то ограничение, например, указать их размерность или фиксировать первый член пары. Некоторые из этих проблем были давно решены, например, проблема гомеоморфии двумерных многообразий. Как установил Марков, общая проблема гомеоморфии не имеет решения: более того, для всякого $n > 3$ можно указать такое n-мерное многообразие, что проблема гомеоморфии многообразий этому многообразию нерешима (см. [Марк 58а], [Марк 58в]). В [Марк 57б], [Марк 62], [Бун Хак Поэ 68], [Хак 73] эти результаты усиливаются, уточняются и обобщаются. Проблема гомеоморфии трехмерных многообразий остается открытой.

О нетривиальных алгоритмических проблемах в других областях математики мало что известно. (Иногда у специалистов имеется явное ощущение эффективности некоторого построения; в то же время попытка сформулировать соответствующую алгоритмическую проблему либо вообще ни к чему не приводит, либо же приводит к нерешимой проблеме - и, тем самым, к неадекватной формулировке.) Все же, касаясь других областей математики, выделим два примера из теории дифференциальных уравнений: нерешаемость проблемы существования решения, определенного на отрезке $[0,1]$, у системы дифференциальных уравнений (см. [Адл 69]) и формулировку алгоритмической проблемы об устойчивости (см. [Арн 76]).

Приведем еще два примера алгоритмических массовых проблем о целых числах. Несмотря на простоту формулировок, для этих примеров неизвестно существование или отсутствие решения. (Напомним, что два подобных примера - один, связанный с великой теоремой Ферма, другой - с десятой проблемой Гильберта - были сформулированы выше.) Первая проблема поставлена в [Сал Сои 78]: по целочисленной матрице узнать, существует ли ее степень, имеющая ноль в правом верхнем углу. Заметим, что для близкой по формулировке проблемы: "по целочисленной матрице A и целочисленным векторам x и y узнать, существует ли такое i, что $A^i x = y$", недавно была доказана решаемость, см. [Канна Лип 80].

Вторая проблема касается коммутативных исчислений - так называемых систем векторного сложения (vector addition sys-

tems). Она была поставлена Карпом и Миллером в работе по параллельным схемам программ в 1969 г. (см. [Карп Милл 69]). Для кратности приведем мультипликативную формулировку проблемы, в которой используется только умножение и отсутствует сложение, хотя аддитивная формулировка и более естественна. Пусть задано натуральное число a и конечное множество V рациональных чисел. Натуральное число называется достижимым (для данных a и v),если оно совпадает с a или получается из достижимого с помощью умножения на элемент из v .Требуется узнать, достижимо ли данное число b. В 1977 г. Сэсердоут и Тенни объявили о решении этой алгоритмической проблемы при любых фиксированных a и V (см. [Сэс Тен 77]). Однако данное ими описание алгоритма является противоречивым; если пытаться устранить противоречия, то оказывается, что получаемые алгоритмы правильно работают только в тривиальных случаях, и не видно, как можно было бы исправить дело. Наибольший прогресс в направлении частичного решения этой задачи с тех пор достигнут в [Хоп Пан 79].

Наш перечень мы закончим наиболее наглядным примером открытой алгоритмической проблемы. Пример связан с игрой "Жизнь", см. [Гар 70 - 71]. Позиция этой игры - бесконечный лист бумаги в клетку. Клетка может быть пустой или полной (в другой терминологии - мертвой или живой, соответственно). Полных клеток (ячеек) в позиции может быть только конечное число. У каждой клетки восемь (очевидных) соседей - четыре соприкасаются с ней по стороне и четыре по углу:

Ход игры состоит в одновременном изменении содержимого всех тех клеток, к которым применимо одно из следующих правил:

1) Рождение новой ячейки: пустая клетка, имеющая ровно трех полных соседей, становится полной;

2) Смерть ячейки: полная клетка, имеющая более трех полных соседей, становится пустой (умирает от тесноты); полная клетка, имеющая менее двух полных соседей, становится пустой (умирает от скуки).

Таким образом, скажем, пустые клетки с одним полным соседом остаются пустыми, а полные клетки с двумя полными соседями остаются полными. Как мы видим, в игре один партнер (можно считать, что и ни одного), никакого конца у нее нет. Тем не менее, естественно считать, что игра заканчивается, если все клетки становятся пустыми, и в этом случае называть позицию, с которой игра началась, "вымирающей". Неизвестно, существует ли алгоритм, распознающий по произвольной позиции "вымрет" ли она.

Массовые проблемы в смысле Медведева

Наше понимание термина "массовая проблема" близко к [Марк 54, гл. У, преамбула]. Более общее и абстрактное понимание было предложено Медведевым в [МедД 55], [МедД 56]. По Медведеву массовая проблема - это произвольное семейство всюду определенных функций из $\mathbb{N}$ в $\mathbb{N}$, и массовая проблема называется (алгоритмически) разрешимой, если она содержит вычислимую функцию. Таким образом, массовую проблему $\mathcal{A}$ (в смысле Медведева) можно рассматривать как проблему нахождения вычислимой функции в семействе $\mathcal{A}$. Конечно, нет необходимости ограничиваться функциями из $\mathbb{N}$ в $\mathbb{N}$, можно рассматривать всюду определенные (это существенно) функции из X в Y для произвольных ансамблей X,Y. Всякая алгоритмическая массовая проблема (в старом смысле), у которой множество вопросов X есть некоторый ансамбль и E (ограничение на вопросы) равно X, может рассматриваться как массовая проблема в смысле Медведева, состоящая из тех функций f, определенных на всем X, для которых $\langle x, f(x)\rangle \in R$ при всех $x \in X$ (здесь R - вопросно-ответное отношение). Такое толкование допускают, в частности, проблемы разрешения и отделения. Именно:

1. Проблеме разрешения для множества $A \subset X$ соответствует семейство, состоящее из единственной функции φ, такой, что $\varphi(x)=1$ для всех $x \in A$ и $\varphi(x)=0$ для $x \in X \setminus A$ (так что φ - характеристическая функция для A).

2. Проблеме отделения для пары множеств $\langle A, B\rangle$, где $A \subset X$, $B \subset X$, соответствует семейство всех отделяющих функций для $\langle A, B\rangle$ (функция φ называется отделяющей для $\langle A,B\rangle$, если она определена на всем X и $\varphi(x)=1$ для $x \in A$, $\varphi(x)=0$ для $x \in B$). Однако не всякая естественно возникающая массовая проблема в

смысле Медведева есть результат подобного истолкования некоторой алгоритмической массовой проблемы. Например, для каждого непустого множества $A \subset \mathbb{N}$ можно рассмотреть проблему перечисления этого множества, состоящую из всех всюду определенных на $\mathbb{N}$ функций, область значений которых есть A (очевидно, эта проблема разрешима по Медведеву в том и только в том случае, когда A перечислимо). Проблема перечисления не может быть получена описанным выше способом из алгоритмической массовой проблемы. (Это не удивительно, т.к. задача "перечислить данное множество" не представима в виде серии отдельных задач.)

Пусть $\mathcal{A}, \mathcal{B}$ - массовые проблемы в смысле Медведева. Из слабой (а, следовательно, и из сильной) сводимости (см. часть I, § 13) $\mathcal{B}$ к $\mathcal{A}$ вытекает, что если $\mathcal{A}$ разрешима, то и $\mathcal{B}$ разрешима. По этой причине слабую и сильную сводимость из части I, § 13, можно интерпретировать как сводимость проблем.

О пользе правильной терминологии

Закончим параграф одним примером четкого различения понятий и терминов. В первом абзаце из [Муч 65, § 2] говорится: "Теорема 1 показывает, что нельзя свести проблему разрешения нерекурсивного множества к проблеме отделимости перечислимых множеств в смысле алгоритмической сводимости... Однако, как мы докажем далее, можно свести проблему разрешения любого перечислимого множества E к проблеме отделимости двух перечислимых множеств E', E" в следующем смысле..." (этот смысл далее разъясняется). В терминологии данного обзора основные результаты [Муч 65] можно изложить так: теорема 1 утверждает, что никакая нерешимая проблема разрешения не может сильно сводиться к проблеме отделения для пары непересекающихся породимых множеств; из доказательства теоремы 2 вытекает, что алгоритмическая массовая проблема разрешимости для Gen(ℕ) сводится по разрешимости (см. ч. I, § 11) к алгоритмической массовой проблеме отделимости для Gen(ℕ).

§ 2. ПРИЛОЖЕНИЯ К ОСНОВАНИЯМ МАТЕМАТИКИ: КОНСТРУКТИВНАЯ СЕМАНТИКА

Возникновение понятия алгоритма и развитие теории стимулировались, наряду с практикой, заключающейся в решении мас-

совых проблем, также и умозрениями, а именно попытками осмыслить сочетание кванторов $\forall x \exists y$. Оба названных аспекта тесно связаны: с одной стороны, если массовая проблема имеет решение, то это означает, что ($\forall$ единичная проблема) ($\exists$ решение); с другой стороны, обоснование утверждения, начинающегося с $\forall x \exists y$, состоит в предъявлении для каждого x соответствующего y, т.е. в решении некоторой массовой проблемы. Понимание этой последней массовой проблемы как алгоритмической является определяющим при разработке так называемого конструктивного направления в математике (см. [Марк 62а], [Шан 62, введение и приложение], [Шан 70], [Куш 73, введение]), пользующегося особой, "конструктивной" логикой рассуждений. В этой логике, в частности, обоснование суждения $\forall x(A(x) \vee \neg A(x))$ связывается с построением алгоритма, указывающего для каждого x верный член дизъюнкции (и поэтому при неразрешимом A это суждение ложно). При том, что конструктивная логика опирается на понятие алгоритма, оперирование с алгоритмами в рамках этой логики предполагает, в свою очередь, ограничение допускаемых логических средств. В число таких средств Марков включил (см. [Марк 62, с. 11]) следующие два принципа: 1) если предположение о неограниченной продолжаемости процесса применения алгоритма $\mathcal{A}$ к слову P опровергнуто, то $\mathcal{A}$ применим к P; 2) если для свойства $\mathcal{P}$ имеется алгоритм, выясняющий для всякого натурального n, обладает ли оно этим свойством, и если опровергнуто предположение о том, что ни одно n не обладает свойством $\mathcal{P}$, то имеется n со свойством $\mathcal{P}$. Последний принцип впервые сформулирован в [Марк 54а]; в [Марк 56] он назван "ленинградским принципом", а в [Марк 62а] - "методом конструктивного подбора" (см. также [Куш 79]).

Можно, однако, заниматься изучением конструктивной логики, пользуясь обычной, "классической" логикой (ср. [Нов 77]). Такой подход был намечен Колмогоровым в [Колм 32] (Колмогоров писал об интуиционистской логике, но различие между интуиционистской и конструктивной логиками в данном случае несущественно). Подход Колмогорова состоит в истолковании конструктивной логики как логики задач, или проблем; существенно, что для любых двух задач A и B в качестве особой задачи рассматривается задача сведения B к A (ср. ч. I, § 11). Этот общий

подход позволяет, в частности, интерпретировать пропозициональные формулы как выражения для задач (а не для утверждений, как в традиционной логике): при такой интерпретации значениями пропозициональных переменных являются задачи и соответственно понимаются пропозициональные связки: $A \wedge B$ означает "решить и A и B", $A \vee B$ – "решить хотя бы одну из задач A и B ", $A \to B$ – "свести решение B к решению A ", $\neg A$ – "предположив, что решение A дано, получить противоречие". Аналогично интерпретируются и предикатные формулы.

Наиболее полно колмогоровский подход воплощен в понятии реализуемости по Клини (см. [Кли 52, § 82], [Нов 77, гл.У, § 7]).

Реализуемостная семантика Клини может быть следующим образом изложена в виде семантики задач. Каждая арифметическая (т.е. принадлежащая элементарной арифметике) формула интерпретируется как задача нахождения некоторого конструктивного объекта, называемого реализацией формулы и кодирующего информацию о ее конструктивной истинности; формула называется реализуемой, если она имеет реализацию. При этом, например, для формулы без свободных переменных, имеющей вид $\forall x\, A$, ее реализацией считается программа алгоритма, дающего по любому значению переменной x реализацию подстановки этого значения в A вместо x; реализацией формулы вида $A \to B$ (где A и B – формулы без свободных переменных) считается программа алгоритма, перерабатывающего всякую реализацию A в некоторую реализацию B . Далее определяются реализуемость и неопровержимость предикатных (и, в частности, пропозициональных) формул. Именно, предикатная формула называется (согласно [Пли 78]): 1) неопровержимой (в [Нов 77] – "реализуемой"), если для каждой подстановки арифметических формул вместо ее предикатных (в частности, пропозициональных) букв существует реализация результирующей арифметической формулы; 2) реализуемой (в [Нов 77] – "эффективно реализуемой"), если существует алгоритм, указывающий для каждой такой подстановки реализацию результата. Существуют предикатные формулы, являющиеся неопровержимыми, но не являющиеся реализуемыми (см. [Пли 76], [Пли 78]); можно ли найти такую формулу среди пропозициональных – неизвестно. Шанин в [Шан 58], [Шан 58а] подверг критике клиниеву концепцию реализуемости (Клини обсуждает эту критику в [Кли 60]) и пред-

ложил свой собственный вариант конструктивной семантики; этот вариант связывает конструктивные задачи не со всеми суждениями, а лишь с некоторыми, и опирается на так называемый "алгоритм выявления конструктивной задачи" (а также на "алгоритм мажорирования арифметических суждений", см. [Шан 73]).

Другим уточнением замысла Колмогорова служит введенная в [МедД 55], [МедД 56] семантика пропозициональных формул. Подобная же семантика была введена в [Муч 63]. Первая семантика основана на сильных, а вторая на слабых степенях трудности. Для того, чтобы представить каждую из семантик как семантику проблем, необходимо вспомнить (см. часть I, § 13, и часть II, § 1), что сильная (соответственно, слабая) степень трудности представляет собой класс эквивалентных массовых проблем в смысле Медведева, если эквивалентностью проблем считать их сильную (соответственно, слабую) сводимость друг к другу. Итак, возьмем решетку Медведева или решетку Мучника. Для произвольного $\mathbb{E}$ из выбранной решетки пропозициональные связки следующим образом интерпретируются как операции на начальном сегменте $\mathcal{S}_{\mathbb{E}}=\{\mathbb{X}\mid \mathbb{O}\leq\mathbb{X}\leq\mathbb{E}\}$ этой решетки: конъюнкция $\wedge$ как $\cup$, дизъюнкция $\vee$ как $\cap$, импликация $\mathbb{A}\to\mathbb{B}$ есть наименьший элемент (доказывается, что такой существует) множества $\{\mathbb{C}\mid \mathbb{B}\leq\mathbb{A}\cup\mathbb{C}\}$; наконец, $\neg\mathbb{A}$ есть $\mathbb{A}\to\mathbb{E}$. Всякая доказуемая формула интуиционистского пропозиционального исчисления тождественно принимает значение $\mathbb{O}$ на любом $\mathcal{S}_{\mathbb{E}}$. Далее, только для решетки Медведева: при $\mathbb{E}=\mathbb{1}$ для формул, не содержащих отрицания, имеет место теорема полноты (см. [МедД 62]): всякая формула такого вида, тождественно равная $\mathbb{O}$ на $\mathcal{S}_{\mathbb{1}}$, доказуема в интуиционистском пропозициональном исчислении; для формул, содержащих отрицание, эта теорема полноты очевидным образом нарушается (в противоречии со сказанным в [Родж 67, § 13.7]: формула $\neg\mathbb{A}\vee\neg\mathbb{A}$ тождественно равна нулю, но недоказуема), и задача о таком выборе $\mathbb{E}$, чтобы теорема полноты имела место для всех формул интуиционистского пропозиционального исчисления, остается открытой. Имеет ли место какая-либо теорема о полноте для решетки Мучника, неизвестно.

Говоря о конструктивной логике, мы ограничивались до сих пор лишь суждениями - их конструктивным пониманием и умозаключениями, ведущими к конструктивному обоснованию их истинности

(следует заметить, впрочем, что в конструктивной логике понять суждение - значит понять, что является его обоснованием). Однако логика занимается еще и понятиями, и конструктивное осмысление понятий также должно составлять предмет конструктивной логики. Поскольку истоки современного конструктивного направления в математике лежат в интуиционизме, был предпринят ряд попыток интерпретации некоторых интуиционистских понятий на основе алгоритмов: в частности, такая попытка осуществлена в [Кли 52а] для брауэровского понятия множества и в [Усп 57, § 7] для вейлевского понятия функции.

Специфическую область приложений теории алгоритмов к конструктивизации понятий составляет исследование определений с точки зрения их конструктивности. "Конструктивным" условно называется такое определение, в котором определяемое свойство связывается с наличием некоторой конструкции: таково определение перечислимого множества. С другой стороны, определение неперечислимого множества "неконструктивно" в том смысле, что состоит в простом отрицании наличия соответствующей конструкции. Оказывается, что в ряде случаев среди всех объектов, не обладающих свойством A,- "объектов не—A" - удается выделить объекты, не обладающие свойством A в некотором конструктивном смысле - "объекты конструктивно (или эффективно) не— A". Именно, объект "конструктивно не— A",- это такой объект, для которого существует алгоритм, отличающий его от любого объекта, обладающего свойством A. Например, в [Усп 60, § 13] разбираются конструктивизации определений неконечного множества, неперечислимого множества и неотделимой пары множеств, а в [Усп 74, § 9] вводится общее понятие множества, эффективно отличного от множеств заданного семейства. Наконец, проблему естественно называть эффективно (= конструктивно) нерешимой, коль скоро существует алгоритм, отыскивающий для каждого кандидата в решения причину, почему этот кандидат не является в действительности решением; для так называемых параметрических проблем в [МедД 69] этот общий и расплывчатый замысел конкретизируется в точном понятии эффективно опровержимой параметрической проблемы.

§ 3. ПРИЛОЖЕНИЯ К МАТЕМАТИЧЕСКОЙ ЛОГИКЕ: АНАЛИЗ ФОРМАЛИЗОВАННЫХ ЯЗЫКОВ ЛОГИКИ И АРИФМЕТИКИ

В [Чёрч 56, § 07] математическая логика, или символическая логика, или логистика определяется как "предмет формальной логики, изучаемый методом построения формализованных языков". Среди формализованных языков выделяются чисто логические языки (пропозициональные и предикатные), языки арифметики и языки теории множеств. Предикатные языки, как элементарный (он же узкий, или 1-го порядка), так и неэлементарные (расширенные или высших порядков), служат для формальной записи свойств математических (прежде всего, алгебраических) структур, в частности, для аксиоматического описания различных классов таких структур. Языки арифметики служат для описания натурального ряда (который вряд ли может быть задан аксиоматически и, во всяком случае, должен очевидным образом рассматриваться как предшествующий каким-либо аксиоматическим рассмотрениям). Языки теории множеств не имеют отчетливой семантики и предназначены для записи различных аксиоматических теорий.

В тех случаях, когда это осмысленно, для формул языка определяется понятие истинности (на том или ином классе обслуживаемых этим языком структур) и ставится алгоритмическая массовая проблема распознавания истинности, или семантическая проблема разрешения: построить алгоритм, распознающий по формуле языка, истинна она или нет. Для языков, обладающих достаточно богатыми выразительными средствами (а именно, достаточно богатыми для того, чтобы выразить - в разумном смысле - какой-либо неразрешимый предикат), семантическая проблема разрешения оказывается нерешимой. Нерешаемость семантической проблемы разрешения для языка арифметики, содержащего равенство, сложение и умножение, отмечалась выше в § 1. Здесь мы заметим только, что нерешаемость этой проблемы, т.е. неразрешимость множества истинных формул арифметики, является тривиальным следствием неарифметичности указанного множества, а эта неарифметичность, в свою очередь, представляет собой простое применение к данному случаю известной теоремы Тарского о невыразимости понятия истинности в языке средствами того же

языка (см., например, [Усп 82, приложение Б]). Если для языка арифметики естественно рассматривать истинность на одной структуре (на натуральном ряду), то для предикатных языков естественно рассматривать истинность на всех мыслимых структурах, или истинность при всевозможных интерпретациях; так понимаемая истинность называется "общезначимостью". Для расширенных предикатных языков само понимание общезначимости наталкивается на серьезные трудности теоретико-множественного характера. Для узкого предикатного языка (= языка элементарной логики предикатов) проблема распознавания общезначимости привлекала пристальное внимание математических логиков, начиная с 1915 г. (см. [Чёрч 56, § 49]); эта проблема, получившая название "das Entscheidungsproblem ", названа в [Гиль Акк 38, гл. Ш, § 12] главной проблемой математической логики. В силу теоремы Гёделя о полноте, das Entscheidungsproblem равносильна проблеме распознавания доказуемости элементарных предикатных формул (при подходящем понятии доказуемости). В 1936 г. нерешаемость das Entscheidungsproblem была независимо установлена Чёрчем и Тьюрингом (см. [Чёрч 36а], [Чёрч 36б], [Тью 36], [Тью 37]). Аналогичные результаты имеют место и для реализуемостной семантики предикатных формул (см.§2):невозможен ни алгоритм, распознающий реализуемость таких формул, ни алгоритм, распознающий их неопровержимость. Эти результаты следуют из теорем Плиско: 1) множество всех реализуемых предикатных формул не является арифметическим (см. [Пли 73], [Пли 77]); 2) множество всех неопровержимых предикатных формул не является арифметическим (см. [Пли 76], [Пли 78]). Относительно разрешимости и арифметичности множества реализуемых и множества неопровержимых пропозициональных формул ничего не известно.

Наряду с проблемой разрешения для множества истинных формул, представляющей собою проблему построения алгоритма, естественно ставить проблему порождения этого множества, представляющую собою проблему построения исчисления: требуется построить исчисление, порождающее все истинные и только истинные формулы (в эквивалентных терминах: построить логистическую систему, в которой доказываются в точности такие формулы). Теорема Гёделя о полноте дает положительное решение этой проб-

лемы для языка элементарной логики предикатов, а теорема Гёделя о неполноте - отрицательное решение этой проблемы для языка элементарной арифметики. Включение теоремы о неполноте в контекст понятий теоретического программирования осуществлено в [Глу 79].

Теорему о неполноте можно воспринимать как чистую теорему несуществования. Однако уже доказательство (а в косвенной форме - даже формулировка) самого Геделя в [Гёд 31] содержит алгоритм, позволяющий по любому исчислению (логистической системе) указать отличие порождаемого этим исчислением множества (= множества всех формул, доказуемых в этой логистической системе) от множества всех истинных формул арифметики, т.е. указать элемент одного из этих множеств, не принадлежащий другому. Ясно, что этим свойством "эффективной гёделевости" (сравни [Усп 74, § 10]) обладают в точности те языки, у которых множество всех истинных формул эффективно отлично (см. § 2) от всех перечислимых (= породимых) подмножеств множества всех формул.

Невозможность ввести для какого-либо языка адекватное, т.е. равнообъемное истинности, понятие доказуемости (а эта невозможность и составляет предмет теоремы о неполноте для рассматриваемого языка) тесно связана с понятием неотделимости (два множества неотделимы, если они не являются отделимыми, см. § 1). Как подметили Клини (см. [Кли 52, § 61]) и Колмогоров (см. [Усп 53] или [Усп 53а]), если имеются два неотделимых множества формул какого-либо языка, причем все формулы из первого множества истинны, а все формулы из второго множества ложны (в том смысле, что истинны их отрицания), то для рассматриваемого языка нельзя ввести непротиворечивую логистическую систему, являющуюся полной (т.е. такой, в которой все истинные формулы оказывались бы доказуемыми). Это обстоятельство позволяет получать теоремы неполноты, избегая рассмотрения сложно устроенного множества всех истинных формул (содержащего и такие формулы, установление истинности которых встречает трудности), а ограничиваясь построением какого-либо множества "заведомо истинных" формул и какого-либо множества "заведомо ложных" формул - так, чтобы эти два множества были сравнительно просто устроены, но все же неотделимы. (Различные

варианты конструктивизации понятия неотделимости - см. [Усп 53а], [Шму 58], [Шму 60], [Шму 61, гл. У, § 1 и § 12], [Усп 60, § 13] - естественно приводят к соответствующим вариантам эффективной неделимости.) Аппарат неотделимости может быть применен и для установления нерешаемости семантических проблем разрешения (см. [Тра 53]).

Всю теорию доказательств можно рассматривать как ветвь прикладной теории алгоритмов и исчислений. И дело тут не только и не столько в результатах, сколько в исходных идейных предпосылках. Сами понятия формального доказательства и доказуемой формулы, рассматриваемые во всей их общности, опираются на фундаментальные представления алгоритмического или исчислительного характера. Здесь возможны два подхода (разумеется, эквивалентных), отдающие, соответственно, примат понятию исчисления или понятию алгоритма.

Первый подход состоит в том, что понятие доказуемой формулы вводится непосредственно, без использования понятия доказательства: формула называется доказуемой, если она порождается рассматриваемой логистической системой. Доказательства вводятся потом как протоколы (= записи) порождений. Что же касается самого понятия логистической системы, как оно понимается, например, в [Чёрч 56, § 07] или в [Мин 67, § 12.2], то трудно отличить это понятие от общего понятия исчисления - можно просто сказать, что логистические системы - это такие исчисления, которые ориентированы на получение формул в формализованных языках (и эта направленность отражается в сопутствующей терминологии).

Второй подход состоит в том, что сперва вводится понятие доказательства, а затем, через него, определяется понятие доказуемой формулы. Основное, что требуется от доказательства при таком подходе, это чтобы существовал алгоритм, отличающий доказательства от недоказательств, т.е. чтобы множество всех доказательств было разрешимым; это требование обосновано в [Чёрч 56, § 07]. Рассматриваемый подход естественно приводит к понятию дедуктики (см. [Усп 74, § 3], [Усп 82, § 1, п.3.3]). Дедуктикой над алфавитом Б рассматриваемого языка называется произвольная тройка $\langle Д, D, \delta \rangle$, где Д - произвольный алфавит (алфавит доказательств), D - произвольное разрешимое множество

слов в алфавите Д (множество всех доказательств) и δ - такая вычислимая функция (функции выделения доказанного), которая определена на всяком элементе из D и значениями которой служат слова в алфавите Б. Те слова в Б, которые принадлежат множеству $\delta(D)$, называются доказуемыми относительно данной дедуктики. Понятие дедуктики может трактоваться как уточнение наиболее общего интуитивного представления о формальном доказательстве.

Заметим в заключение, что, как подчеркнуто в [Чёрч 56, § 07], использование алгоритмических представлений требуется не только при введении понятия доказуемости, но и на более ранних стадиях изучения формализованных языков, в частности, при определении понятия (правильно построенной) формулы.

§ 4. ВЫЧИСЛИМЫЙ АНАЛИЗ

Понятие вычислимого действительного числа и вычислимой функции действительного переменного восходят к статье Бореля [Бор 12]; в этой же статье намечены и некоторые основные факты вычислимого анализа. Раздел II указанной статьи называется "Вычислимые числа" ⟨ "Nombres calculables"⟩ и начинается со следующего определения: "Мы скажем, что число α вычислимо, коль скоро для произвольного целого n можно получить рациональное число, которое отличается от α менее, чем на $1/n$". Сделанное к этой формулировке примечание о "достоверном и недвусмысленном методе" получения (оно процитировано нами в ч. I, § 1) не оставляет сомнений, что Борель имел ввиду самую общую концепцию алгоритмической вычислимости. Сейчас мы бы сказали: "Действительное число α вычислимо, коль скоро существует алгоритм, дающий по всякому целому положительному n рациональное приближение к α с точностью до $1/n$". Далее Борель указывает, что если два вычислимых числа не равны, то их неравенство может быть рано или поздно обнаружено путем подбора достаточно близких рациональных приближений (хотя точность, с которой необходимо брать такие приближения, и не может быть предвидена заранее); если же два вычислимых числа равны, то попытка обнаружить их равенство может натолкнуться на неразрешимые трудности ⟨ difficultés insolubles ⟩. Совре-

менная формулировка: "каково бы ни было конструктивное действительное число y, невозможен алгоритм, указывающий для любого конструктивного действительного числа x верный член дизъюнкции $(x=y) \vee (x \neq y)$" ([Куш 73, гл. 4, § 1, теорема 3]). Раздел III статьи Бореля называется "Вычислимые функции и функции с асимптотическим определением" ("Les fonctions calculables et les fonctions à définition asymtotique "). Буквальная формулировка гласит: "функция вычислима, коль скоро ее значение вычислимо для каждого значения аргумента". Однако в последующих комментариях Борель по существу требует наличия алгоритма, позволяющего по α и n вычислить $f(\alpha)$ с точностью $1/n$, поясняя при этом, что "задать вычислимое число α - это просто задать метод получения α с произвольной точностью". Современное определение вычислимой функции вычислимого действительного переменного (см. ниже) может поэтому трактоваться как уточнение определения Бореля. (Правильнее было бы сказать, впрочем, что одновременно с уточнением происходит и ограничение области определения: "современные" вычислимые функции рассматриваются лишь на вычислимых действительных числах.) Борель формулирует и утверждение о непрерывности вычислимой функции (доказательство этого утверждения было найдено лишь в 1956 г. Цейтиным, см. ниже). Он пишет: "функция не может быть вычислимой, если она не является непрерывной, по крайней мере для вычислимых значений аргумента". "Более того, - указывает Борель, - нужно предполагать известной меру непрерывности функции, то есть инфинитезимальный порядок (в обобщенном смысле) изменения функции в сравнении с изменением аргумента". Если понимать эту "меру непрерывности" как вычислимый регулятор непрерывности (см. ниже), можно заключить, что Борель имел в виду не просто непрерывность, но вычислимую непрерывность.

Систематическое развитие вычислимого анализа на основе точных алгоритмических представлений началось со статьей Тьюринга [Тью 36], [Тью 37]. История этого развития прослежена в [Куш 73, введение, п. 2]. Публикации в обсуждаемой области можно отнести к одному из двух направлений. За первым из них мы сохраняем термин "вычислимый анализ", второе обычно называется "конструктивным анализом". Объекты, изучаемые в первом

направлении, носят названия "вычислимые числа", "вычислимые функции" и т.п., объекты второго направления называются "конструктивными числами", "конструктивными функциями" и т.п.; к сожалению, это разделение терминологии не всегда соблюдается. Различие между направлениями состоит в следующем. Вычислимый анализ выделяет среди традиционных объектов анализа - чисел и функций - вычислимые, т.е. связанные определенным образом с алгоритмами. Конструктивный анализ рассматривает вычислимые числа и функции не как часть более обширной совокупности всех чисел и функций, а сами по себе. Более того, понятие программы числа или функции является для него исходным: конструктивным числом называется то, что в вычислимом анализе называется программой вычислимого числа, а конструктивной функцией - то, что в вычислимом анализе называется программой вычислимой функции; на конструктивных числах и конструктивных функциях затем вводится отношение равенства, означающее, разумеется, не совпадение соответствующих конструктивных объектов, а совпадение задаваемых ими чисел и функций. Такой способ изложения позволяет говорить непосредственно об алгоритмах над конструктивными числами и конструктивными функциями. Ясно, что понятия и результаты вычислимого анализа и конструктивного анализа без труда переводятся друг в друга. Следует, однако, отметить, что в содержание понятия "конструктивный анализ", как правило, вкладывается еще и требование использовать только конструктивную логику (см. [Куш 79а]).

Начальные понятия вычислимого анализа таковы:

1. Вычислимая последовательность рациональных чисел; это понятие не нуждается в комментариях.

2. Вычислимо сходящаяся (или вычислимо фундаментальная) последовательность. Это последовательность, обладающая вычислимым регулятором фундаментальности. Регулятором фундаментальности (или регулятором сходимости в себе) для последовательности $\{a_n\}$ называется такое отображение h множества $\mathbb{Q}^+$ в $\mathbb{N}$, что $|a_p - a_q| < \varepsilon$ при любых p и q, больших $h(\varepsilon)$. Согласно [Марк 54а], [Марк 58п], последовательность называется регулярно сходящейся, если для любых m и n, таких что $m \leq n$, соблюдается условие $|a_m - a_n| \leq 2^{-m}$. Очевидно, каждая вычислимая вычислимо сходящаяся последовательность обладает вычислимой

же регулярно сходящейся подпоследовательностью.

3. Вычислимое действительное число. Существует несколько эквивалентных определений этого понятия:

1) определение Бореля, уточненное с помощью понятия алгоритма (см. выше);

2) определения Тьюринга из [Тью 36], [Тью 37] - первые определения, использующие какую бы то ни было вычислительную модель (да и вообще первые строгие, в математическом смысле, определения);

3) определение Шпекера из [Шпе 49]: действительное число вычислимо, если оно есть предел вычислимой и вычислимо сходящейся последовательности рациональных чисел;

4) определение Маркова из [Марк 54а], [Марк 58г], переложенное в терминах вычислимого анализа: действительное число вычислимо, если оно есть предел вычислимой регулярно сходящейся последовательности рациональных чисел;

5) определение "по Дедекинду": α вычислимо, если каждое из множеств $\{r \in \mathbb{Q} \mid r<\alpha\}$ и $\{r \in \mathbb{Q} \mid r>\alpha\}$ перечислимо.

Множество всех вычислимых действительных чисел называется вычислимым континуумом.

4. Программа вычислимого действительного числа. Определение этого понятия может быть легко получено на основе любого из перечисленных определений вычислимого числа. Например, определение Бореля приводит к такому определению программы: программой числа α называется программа алгоритма, который по всякому $\varepsilon \in \mathbb{Q}^+$ дает рациональное ε-приближение к α. Нужно иметь в виду, что одно из определений Тьюринга, связанное с разложением числа в бесконечную десятичную дробь (это определение требует, чтобы последовательность десятичных знаков была вычислима), приводит к "дурному" способу программирования, не эквивалентному способам, возникающим из других определений, и потому этот способ мы не будем рассматривать. Все остальные способы, возникающие на основе приведенных выше определений, одинаково хороши, и переход от программы числа при одном способе к программе того же числа при другом способе осуществляется алгоритмически (иными словами, возникающие нумерации вычислимых действительных чисел их программами эквивалентны в смысле ч. I, § 15). Сейчас мы изложим подробно определение

программы, отправляясь от определения вычислимого числа по Шпекеру. Фиксируем какие-либо две представительные вычислительные модели, осуществляющие, соответственно, вычисления функций из $\mathbb{N}$ в $\mathbb{Q}$ и из $\mathbb{Q}$ в $\mathbb{N}$, и соответствующие способы программирования (см. ч. I, § 15). Следуя Шанину (см. [Шан 56], [Шан 62]), назовем вещественным дуплексом, или просто дуплексом, всякую пару $\langle p_1, p_2 \rangle$, в которой p_1 есть программа некоторой последовательности рациональных чисел, а p_2 есть программа некоторого регулятора сходимости в себе этой последовательности. Всякий дуплекс, таким образом, задает некоторую вычислимую и вычислимо сходящуюся последовательность рациональных чисел и, следовательно, вычислимое действительное число, являющееся пределом этой последовательности; рассматриваемый дуплекс называется программой этого числа. Было бы неправильно называть программой числа только первый член дуплекса, поскольку этот первый член еще не несет информации, позволяющей вычислять число с произвольной точностью. Действительно, невозможен алгоритм, дающий по произвольной программе произвольной вычислимой последовательности, которая вычислимо сходится, программу какого-либо регулятора сходимости в себе этой последовательности (см. [Цей 62а, § 3, следствие 3], [Куш 73, гл. 4, § 2, теорема 2]). Отображение, относящее всякому дуплексу задаваемое им вычислимое действительное число, служит примером нумерации (при широком понимании, см. ч.I, § 15) множества всех вычислимых действительных чисел. Основание этой нумерации - множество всех вещественных дуплексов - называется конструктивным континуумом; оно неперечислимо; более того, для любого перечислимого множества дуплексов можно указать вычислимое действительное число, не имеющее программы в этом множестве (см. [Куш 73, гл.3, § 4], [Усп 60, § 12, теорема 11]). При замене одной пары представительных моделей на другую переход от программы вычислимого действительного числа относительно исходной пары к программе того же числа относительно новой пары осуществляется алгоритмически: это вытекает из возможности трансляции, о которой говорилось в ч. I, § 14.

5. Вычислимая функция вычислимого действительного переменного. Для простоты ограничимся функциями одного переменного. Излагаемое определение есть несущественное изменение определе-

ния Маркова из [Марк 58г] (у Маркова - "конструктивная функция вещественной переменной"). Прежде всего фиксируем некоторое понятие программы вычислимого числа, например, в виде дуплекса. Функция из вычислимого континуума в вычислимый континуум называется вычислимой, коль скоро существует алгоритм, который 1) дает по всякой программе аргумента программу соответствующего значения функции и 2) не дает никакого результата для любой программы вычислимого действительного числа, не принадлежащего области определения функции.

Дальнейшие понятия связаны с дифференцированием и интегрированием вычислимых функций вычислимого действительного переменного, см. [Куш 73, главы 6 и 7].

Среди результатов вычислимого анализа наиболее замечательными являются следующие два:

1. Пример Шпекера (см. [Шпе 49]) монотонной ограниченной вычислимой последовательности рациональных чисел, не сходящейся ни к какому вычислимому числу. Построение Шпекера было впоследствии значительно упрощено Райсом, см. [Райс 54], [Усп 60, § 12, п. 3], [Март 70, § 16], [Куш 73, гл. 3, § 3].

2. Теорема Бореля - Цейтина (см. [Бор 12, разд.III], [Цей 59], [Цей 62, гл. V, теорема 3]) о непрерывности и даже вычислимой непрерывности вычислимой функции вычислимого действительного переменного. Пусть f - функция действительного переменного с областью определения E и $x_0 \in E$. Функция h, отображающая $\mathbb{Q}^+$ в $\mathbb{Q}^+$, называется регулятором непрерывности функции f в точке x_0, если $(\forall \varepsilon \in \mathbb{Q}^+)(\forall x \in E)[|x-x_0| < h(\varepsilon) \Rightarrow |f(x)-f(x_0)| < \varepsilon]$. Непрерывность f в точке x_0, очевидно, равносильна наличию регулятора непрерывности в x_0; вычислимая непрерывность f в x_0 по определению означает наличие вычислимого регулятора. Теорема Бореля - Цейтина утверждает, что всякая вычислимая функция вычислимого действительного переменного вычислимо непрерывна в каждой точке x_0 своей области определения; более того, для заданной f программа регулятора может быть алгоритмически найдена по программе числа x_0 (см. [Куш 73, гл. 5, § 2, теорема 2]).

Известно, что многие понятия и результаты традиционного анализа переносятся с действительной прямой на метрические пространства. Аналогичное развитие имеет место и в вычислимом

анализе. Так, теорема Бореля - Цейтина является частным случаем более общей теоремы Цейтина - Московакиса (см. [Куш 73, гл. 9, § 2, теорема II]) о непрерывности всякого вычислимого частичного отображения из одного эффективно метрического пространства в другое (при соблюдении некоторых условий, налагаемых на первое пространство, см. ниже). К понятию эффективно метрического пространства мы сейчас и перейдем.

Эффективно метрическое пространство (см. [Ног 66], [Ног 78, гл. II]) - это метрическое пространство, рассматриваемое с некоторой своей нумерацией, причем требуется, чтобы расстояние между любыми двумя точками этого пространства было вычислимым действительным числом и чтобы существовал алгоритм, дающий программу этого числа по номерам точек. Это понятие по существу эквивалентно понятию конструктивного метрического пространства, введенному Шаниным (см. [Шан 62, § 9], а также [Цей 59], [Цей 62]), и понятию рекурсивного метрического пространства, введенному Московакисом, см. [Моск 64]; последние два понятия различаются между собой лишь техническими деталями. Упоминавшаяся выше теорема Цейтина - Московакиса была доказана Цейтиным и, соответственно, Московакисом именно для конструктивных и, соответственно, рекурсивных пространств. Рекурсивное пространство Московакиса (как и конструктивное пространство Шанина) состоит из конструктивных объектов с заданным на них отношением эквивалентности. Оно превращается в эффективно метрическое пространство, если склеить эквивалентные объекты, так что точками нового пространства объявляются классы эквивалентности, а каждый из исходных конструктивных объектов объявляется номером содержащего этот объект класса. Вычислимый континуум с нумерацией чисел, задаваемой их программами, служит примером эффективно метрического пространства, а конструктивный континуум - примером конструктивного метрического пространства. Другой пример: эффективно метрическое пространство всех вычислимых последовательностей натуральных чисел с бэровской метрикой (соответственно, конструктивное метрическое пространство программ таких последовательностей).

Проиллюстрируем на примере теоремы Цейтина - Московакиса некоторые характеристики эффективно метрических пространств. Условия, требуемые этой теоремой и обеспечивающие непрерывность,

состоят в том, чтобы пространство, на котором задано частичное отображение, было 1) эффективно сепарабельным и 2) эффективно почти полным. Эффективная сепарабельность (согласно [Ног 78, гл. II] , у Московакиса - рекурсивная сепарабельность, у Шанина, Цейтина, Кушнера - просто сепарабельность) означает наличие перечислимого плотного подмножества. Эффективная почти полнота (у Кушнера - слабая полнота, у Московакиса - условие (A)) означает наличие алгоритма, который по программе вычислимой последовательности точек пространства и по программе вычислимого регулятора фундаментальности этой последовательности дает программу предела последовательности в случае, если таковой существует. Оба приведенных выше примера пространств являются и эффективно сепарабельными, и эффективно почти полными.

Дальнейшим обобщением является понятие эффективно топологического пространства, введенное и изученное Ногиной (см. [Ног 66], [Ног 69], [Ног 78, гл. III]). Эффективно топологическое пространство - это топологическое пространство, рассматриваемое вместе с двумя нумерациями, первая из которых нумерует само пространство, а вторая - его (топологическую) базу, так что и пространство и база предполагаются счетными; требуется, чтобы существовал алгоритм, который по номерам двух элементов базы A и B и номеру точки $x \in A \cap B$ дает номер такого элемента базы C , что $x \in C \subset A \cap B$. Для эффективно топологических пространств вводятся естественные эффективные аналоги аксиом отделимости в смысле Хаусдорфа (см. [Ног 78, гл. IV]); в каждом таком аналоге требуется наличие алгоритма, дающего номера отделяющих окрестностей. Одним из основных результатов является получение условий, необходимых и достаточных для того, чтобы эффективно топологическое пространство было эффективно метризуемым, т.е. эффективно гомеоморфным эффективно метрическому пространству (см. [Ног 66], [Ног 78, гл. V]). Другой результат состоит в перенесении на эффективно топологические пространства теоремы Цейтина - Московакиса (см. [Вай Ног 76]).

Наряду с вычислимым анализом в строгом смысле возможен и, так сказать, "отчасти вычислимый" анализ. Этот термин мы прилагаем к алгоритмическим конструкциям, характеризующим множества обычных действительных чисел. Так, среди открытых множеств

действительной прямой можно выделить эффективно открытые множества: множество называется эффективно (конструктивно, рекурсивно) открытым, если оно есть объединение перечислимой системы интервалов с рациональными концами. Подобная "эффективизация" может быть произведена для самых различных типов множеств (в частности, для борелевских множеств - см. [Март 70, § 30]).

Займемся теперь эффективизацией понятия множества действительных чисел, имеющего меру нуль. Такие множества называются также пренебрежимыми. Среди всевозможных пренебрежимых множеств выделяются эффективно (конструктивно, рекурсивно) пренебрежимые множества, или множества эффективно нулевой меры: множество эффективно пренебрежимо, если можно эффективно указать содержащее его эффективно открытое множество сколь угодно малой меры. Более точно, множество называется эффективно пренебрежимым, если существует алгоритм, который по каждому $\varepsilon \in \mathbb{Q}^+$ дает программу покрывающей множество перечислимой системы интервалов с рациональными концами, сумма длин которых меньше ε; заметим, что эта сумма, в силу примера Шпекера, может и не быть вычислимым числом. Как объявлено в [Зас Цей 56] и доказано в [Зас Цей 62], вычислимый континуум является эффективно пренебрежимым множеством. Основной результат в этой области - теорема Мартин-Лёфа (см. [Мартб6], [Март 66а], [Март 70, § 35]): существует эффективно пренебрежимое множество, являющееся наибольшим, т.е. содержащее в качестве подмножества любое эффективно пренебрежимое множество. (Отсюда немедленно вытекает упомянутая эффективная пренебрежимость вычислимого континуума: в самом деле, каждое одноэлементное множество, состоящее из вычислимого действительного числа, эффективно пренебрежимо, а значит и объединение всех таких множеств содержится в наибольшем эффективно пренебрежимом множестве.) Дополнение к наибольшему множеству эффективно нулевой меры - множество эффективно полной меры - называется, согласно [Март 70, § 35], конструктивным носителем меры. К сфере отчасти вычислимого анализа относится и исследование обычных функций действительного переменного, обладающих теми или иными алгоритмическими свойствами типа вычислимой аппроксимируемости (см. [Шпе 49], [Кла 61, § 7]) или вычислимой непрерывности (см.

[Кла 61, § 8]).

Разумеется, "отчасти вычислимый анализ" не ограничивается действительной прямой. Понятия эффективно открытого, эффективно - G_δ и т.п. множеств осмыслены для любого топологического пространства с нумерованной базой. В [Куз Тра 55] эти понятия, в применении к бэровскому пространству, используются для исследования вычислимых (частичнорекурсивных) операторов (см. также [Усп 57, § 11]). Теорема Мартин-Лёфа и предложенное ее автором доказательство сохраняют силу в общей ситуации, которую мы сейчас опишем.

Сперва два определения.

1) Пусть M - занумерованное множество с нумерацией α и основанием нумерации E, пусть μ - функция из M в вычислимый континуум; μ называется вычислимой, коль скоро существует алгоритм, дающий по всякому $n \in E$ программу числа $\mu(\alpha(n))$ (сравни с определением вычислимой функции вычислимого действительного переменного и с определением расстояния в эффективно метрическом пространстве).

2) Пусть α - натуральная нумерация системы множеств M и пусть μ - определенная на M вещественнозначная функция; множество A называется эффективно пренебрежимым, если существует алгоритм, который для каждого $\varepsilon \in \mathbb{Q}^+$ дает программу такого породимого множества $K \subset \mathbb{N}$, что семейство множеств $\{\alpha(k) \mid k \in K\}$ образует покрытие для A и $\sum_{k \in K} \mu(\alpha(k)) < \varepsilon$.

Общая формулировка теоремы Мартин-Лёфа такова. Пусть M - счетная система множеств с натуральной нумерацией α и пусть μ - вычислимая функция из M в вычислимый континуум, всюду определенная на M. Тогда существует наибольшее (по включению) эффективно пренебрежимое множество. (В обычных применениях теоремы M - счетное полукольцо множеств в смысле [Колм Фом 76, гл. I, § 5], а μ является мерой на этом полукольце.)

§ 5. НУМЕРОВАННЫЕ СТРУКТУРЫ

Нумерованная структура - это математическая структура (в широком смысле, как у Бурбаки), рассматриваемая вместе с нумерацией (см. ч. I, § 15) одного из составляющих множеств или с нумерациями нескольких таких множеств. Пример: эффективно

топологическое пространство (см. § 4) является нумерованным топологическим пространством, поскольку имеет нумерацию множества своих точек и нумерацию топологической базы.

Интерес к нумерованным структурам вызван желанием дать (конструктивные) имена рассматриваемым (не конструктивным) объектам. Для примера рассмотрим ординалы и системы обозначений для них. В качестве таковых систем мы, следуя Клини (см. [Родж 67, § 11.7]), будем рассматривать нумерации множеств ординалов, обладающие некоторыми естественными свойствами. Именно, системой обозначений называется нумерация некоторого начального отрезка ординалов, обладающая следующими свойствами:

1) существует алгоритм, определяющий по всякому номеру ординала, какой из трех следующих случаев имеет место: ординал равен нулю, имеет предшественника, является предельным;

2) существует алгоритм, который по номеру всякого ординала, имеющего предшественника, указывает один из номеров этого предшественника;

3) существует алгоритм, который по номеру всякого предельного ординала дает программу такой последовательности номеров ординалов, что последовательность ординалов с этими номерами возрастает и сходится к нашему предельному ординалу.

Самая простая система обозначений - тождественная нумерация натуральных чисел (рассматриваемых как ординалы). Чуть более сложный пример системы обозначений - нумерация всех ординалов, представимых как значение полинома с натуральными коэффициентами от ω, этими самыми полиномами.

Введя понятие системы обозначений, мы получаем возможность ввести понятие конструктивного ординала - такого ординала, который имеет номер хотя бы в одной системе обозначений. Мы видим, как теория нумераций позволяет дать определение важного и естественного класса ординалов.

Как доказал Клини (см. [Родж 67, § 11.7]), существует максимальная система обозначений, то есть такая система обозначений, в которой каждый конструктивный ординал имеет номер. (Имеется и другое описание конструктивных ординалов: ординал конструктивен тогда и только тогда, когда он является порядковым числом некоторого разрешимого порядка - см. [Родж 67, § 11.8].)

Особенностью класса конструктивных ординалов как нумерованного множества является то, что исторически первое определение этого класса фактически использовало общее понятие нумерации (в то время как, например, алгебраические числа, о которых будет идти речь дальше, появились, разумеется, безо всякой связи с нумерациями и вообще теорией алгоритмов). Именно рассмотрение этого класса побудило Колмогорова сформулировать общие определения числовой нумерации и сводимости нумераций (см. ч. I, § 15).

Рассмотрим теперь некоторые другие естественно возникающие нумерованные структуры. Например, у множества $Com(X, Y)$ (см. ч. I, § 14) имеются нумерации, возникающие из способов программирования. Другой пример: стандартная нумерация элементов какой-либо конечнозаданной алгебры (например, группы) посредством термов (см. ч. I, § 15). В этих двух примерах мы встречаемся с нумерациями программного типа и структурами с позитивными операционно-вычислимыми нумерациями.

Нумерация программного типа - это нумерация, которую можно получить из способов программирования с помощью операций: (1) прямого произведения, (2) кортежного распространения, (3) сужения и (4) факторизации. Напомним (см. ч. I, § 15), что каждый способ программирования вычислимых функций из X в Y можно рассматривать как нумерацию семейства $Com(X, Y)$, а каждый способ программирования породимых подмножеств W - как нумерацию семейства $Gen(W)$; здесь X, Y, W - некоторые ансамбли. Только что перечисленные четыре операции достаточно очевидны, вот их формальные определения. Пусть α и β суть нумерации множеств A и B с основаниями E и F. Тогда: 1) прямым произведением нумераций α и β называется нумерация γ множества $A \times B$, имеющая основание $E \times F$ и задаваемая формулой:

$$\gamma(a,b) = \langle \alpha(a), \beta(b) \rangle ;$$

2) нумерация β называется кортежным распространением нумерации α, если $B = A^{\infty}$, $F = E^{\infty}$ и $\beta(\langle e_1, \ldots, e_n \rangle) = \langle \alpha(e_1), \ldots \ldots \alpha(e_n) \rangle$; здесь A^{∞}, E^{∞} суть, соответственно, множества всевозможных кортежей над A и над E ; 3) нумерация β называется сужением нумерации α, если $B \subset A$, $F = \alpha^{-1}(B)$ и β есть сужение α на F ; 4) нумерация β называется фактор-нумерацией нумерации α, если существует отображение $f: A \to B$ с областью

определения A и областью значений B, для которого $\beta = f \circ \alpha$. Типичный пример нумерации программного типа - нумерация вычислимого континуума посредством дуплексов (см. § 4).

Известно, что невозможен алгоритм, который по двум программам устанавливал бы, являются ли они программами одной и той же вычислимой функции. Поэтому, если α - нумерация программного типа, то (за исключением тривиальных вырожденных случаев) не существует алгоритма, который по произвольным элементам m и n основания нумерации распознает, имеет ли место равенство $\alpha(m) = \alpha(n)$: нумерация α оказывается неразрешимой в смысле ч. I, § 15. Более того, если в построении нумерации участвует сужение, основание нумерации оказывается (в невырожденных случаях) неразрешимым, а скорее всего - и неперечислимым: неперечислимо, например, множество всех программ всюду определенных функций из X в Y.

Пример нумерации другого, нежели программный, типа - это стандартная нумерация элементов произвольной конечнопорожденной алгебры (см. ч. I, § 15). Все участвующие в сигнатуре операции, очевидно, оказываются вычислимыми (см. ч. I, § 15) относительно этой нумерации. Если алгебраическая система не только конечно порождена, но и конечно задана, то, как мы знаем (см. ч. I, § 15), стандартная нумерация ее элементов термами позитивна (но разрешима только при наличии решения у проблемы распознавания равенства!). Этот важный алгебраический пример показывает роль позитивных нумераций с разрешимыми основаниями и вычислимыми сигнатурными операциями. По-видимому, именно на такие нумерации должно быть обращено главное внимание при изучении нумерованных алгебр. (Заметим, что такая нумерация может и не быть конструктивизацией, см. ниже). Поэтому они заслуживают того, чтобы дать им какое-нибудь специальное название; авторы, однако, не смогли придумать хорошего названия, удовлетворяющего требованиям [Маль 66, с. 72] и [ЕршА 77, с. 76]. Может быть, называть эти нумерации квазистандартными?

Вот еще пример алгебраической структуры, имеющей некоторую естественную нумерацию, оказывающуюся "квазистандартной". Это - множество алгебраических вещественных чисел, занумерованное с помощью полиномов с целыми коэффициентами. Отношения

равенства и порядка оказываются разрешимыми, а операции сложения и умножения - вычислимыми относительно этой нумерации. Мы не входим в технические детали построения нумерации алгебраических чисел с такими свойствами; ограничимся указанием на то, что по существу именно такая нумерация описывается в обычных доказательствах счетности множества алгебраических чисел.

Изложенный пример подводит нас к понятию нумерованной и, далее, конструктивной алгебраической системы. Теория нумерованных и, прежде всего, конструктивных алгебраических систем была основана Мальцевым и развита Ю.Л.Ершовым (см. [Маль 61], [ЕршЮ 73], [ЕршЮ 74], [ЕршЮ 80], [Гон 79]).

Пусть задана сигнатура $\{=, P_1, P_2, \ldots, f_1, f_2, \ldots\}$, где $=, P_1, P_2, \ldots$ суть предикатные символы, $f_1, f_2, \ldots$ - функциональные символы. Естественно рассматривать такие сигнатуры, для которых валентность символов P_i и f_i эффективно находится по i. Для таких сигнатур множество всех формул логического языка (скажем, первого порядка), естественно расположенное в подходящем словарном ансамбле, разрешимо.

Итак, пусть σ - сигнатура описанного типа. Пусть, далее, $\mathfrak{M} = \langle M, \sigma \rangle$ - алгебраическая система (а.система). Пара $\langle \mathfrak{M}, \nu \rangle$, где ν - натуральная нумерация множества M, называется нумерованной а.системой. Обозначим через σ' сигнатуру, получаемую добавлением к σ символов констант $c_0, c_1, \ldots$. Будем считать, что значением (интерпретацией) константы c_i является $\nu(i)$ для всех $i \in \mathbb{N}$. Назовем нумерованную а.систему $\langle \mathfrak{M}, \nu \rangle$ конструктивной а.системой (или рекурсивной а.системой), если множество бескванторных предложений сигнатуры σ', истинных в а.системе $\langle M, \sigma' \rangle$, разрешимо; нумерация ν называется в этом случае конструктивной нумерацией (см. [Маль 62а]), или конструктивизацией а.системы $\mathfrak{M}$. Другими словами, нумерация называется конструктивизацией, если все сигнатурные функции вычислимы, а все сигнатурные предикаты, включая предикат равенства, разрешимы относительно этой нумерации (в смысле, разъясненном в конце § 15 из ч. I). Алгебраическая система называется конструктивизируемой, если у нее существует конструктивизация. Для случая алгебр понятие (но не термин) конструктивизируемой а.системы впервые появилось в работе Кузнецова (у Кузнецова - "общерекурсивная алгебра", см. [Яновс 59, с. 79]). Если теперь

говорить не о бескванторных предложениях, а о любых предложениях (языка первого порядка) сигнатуры σ', то получится определение сильно конструктивной а.системы, сильной конструктивизации и сильной конструктивизируемости.

Ясно, что сильная конструктивизируемость а.системы влечет разрешимость элементарной теории этой а.системы. Обратное, вообще говоря, неверно: элементарная теория сложения на нестандартном натуральном ряду разрешима, но соответствующая а.система не только не является сильно конструктивизируемой, но даже не конструктивизируема в силу полученного в [Тве 82] усиления теоремы Тенненбаума (см. конец данного параграфа).

Очевидно, каждая конструктивизация является разрешимой нумерацией. Для алгебр конструктивность натуральной нумерации равносильна выполнению двух требований: требования вычислимости сигнатурных операций (относительно нумераций) и требования разрешимости нумерации. Известно (см. ч. I, § 15), что каждая разрешимая натуральная нумерация бесконечного множества эквивалентна некоторой однозначной натуральной нумерации. Поэтому если бесконечная а.система вообще допускает конструктивизацию, то она допускает и конструктивизацию, являющуюся однозначной нумерацией.

В случае конечнопорожденной а.системы естественно рассмотреть ее стандартную нумерацию (см. ч. I, § 15). Конструктивна ли стандартная нумерация? Не обязательно: уже отношение равенства может оказаться неразрешимым. Именно ввиду такой возможности обсуждавшаяся выше в § 1 алгоритмическая проблема распознавания равенства элементов структуры может оказаться нерешимой. Однако, если структура конструктивизируема, то стандартная нумерация конструктивна. Для конечнопорожденных алгебр разрешимость стандартной нумерации равносильна ее конструктивности и, следовательно, равносильна конструктивизируемости рассматриваемой алгебры. (Впервые это обстоятельство отметил Кузнецов, см. [Яновс 59, с. 79]).

Всевозможные конструктивизации данной а.системы образуют верхнюю полурешетку относительно колмогоровской сводимости; в случае, если а.система конечно порождена, а указанная полурешетка непуста, в этой полурешетке есть наименьший элемент – это стандартная нумерация. (Разумеется, педантичнее было бы

говорить не о самих конструктивизациях, а о классах эквивалентных конструктивизаций.)

Алгебраически корректные массовые проблемы

Разумеется, если нумерация конструктивна, алгоритмическая проблема распознавания равенств элементов (заданных номерами в этой нумерации) решима по определению. Однако и в этом случае другие алгоритмические массовые проблемы, формулируемые для данной структуры в алгебраических терминах, могут оставаться нерешаемыми. Возникает вопрос, что такое вообще алгоритмическая массовая проблема, формулируемая в алгебраических терминах, или, как мы будем говорить, алгебраически корректная алгоритмическая массовая проблема. В следующем абзаце мы надеемся предложить адекватное определение (причем не только для конструктивных, но для произвольных нумерованных алгебраических систем).

Фиксируем какую-либо нумерованную а.систему. Пусть M – ее носитель. Обозначим через M^{ω} множество всех кортежей над M. Для любого подмножества $P \subset M^{\omega}$ можно поставить алгоритмическую массовую проблему A_P : найти алгоритм, распознающий по номерам $n_1,\ldots,n_k$ элементов $a_1,\ldots,a_k$, принадлежит ли кортеж $\langle a_1,\ldots,a_k\rangle$ подмножеству P. Назовем P устойчивым, если оно сохраняется при автоморфизмах, т.е. если для любого автоморфизма α нашей а.системы имеет место $\alpha(P) = P$. Проблема A_P называется алгебраически корректной, если она поставлена для устойчивого P. Итак, алгебраически корректная алгоритмическая массовая проблема (сокращенно – АКАМП) есть проблема построения алгоритма, распознающего по номерам каких угодно элементов а. системы, принадлежит ли кортеж этих элементов произвольному, но фиксированному устойчивому множеству. Мы рассматриваем понятие АКАМП как отражение наших интуитивных представлений о (внутренних, см. § 1) массовых проблемах, сформулированных в алгебраических терминах.

На совокупности всевозможных АКАМП для фиксированной а. системы вводится естественный предпорядок: АКАМП A' сложнее, чем АКАМП A'' (A'' проще, чем A') коль скоро при всякой конструктивизации, при которой решима A', непременно решима и A''. Бывают ли проблемы различной сложности, т.е. неэквива-

лентные относительно только что введенного предпорядка? Этот вопрос был задан одним из авторов Гончарову. В ответном письме последнего от 1 июля 1981 г. был указан пример реляционной системы и двух ее АКАМП различной сложности. Указанная система была специально создана с целью ответа на указанный вопрос, и авторам неизвестно, возможно ли обнаружить этот эффект в "реально существующих" а.системах.

Алгоритмические размерности

Алгебраически корректная алгоритмическая массовая проблема, решимая при одной нумерации данной а.системы, может оказаться нерешимой при другой нумерации той же а.системы. Мальцев в [Маль 62а] сделал тонкое наблюдение: он обнаружил, что указанная возможность реализуется даже для конструктивизаций. Именно, в [Маль 62а] исследована абелева группа R_∞ - счетная прямая сумма слагаемых, каждое из которых изоморфно аддитивной группе рациональных чисел - и установлено, что для этой группы алгоритмическая проблема линейной зависимости имеет решение при одной конструктивизации и не имеет решения при другой. Таким образом, для этой группы существуют конструктивизации столь различные, что возможна АКАМП, решимая при одной конструктивизации и нерешимая при другой. Возникает определение существенного различия конструктивизаций (и, вообще, нумераций): две нумерации данной а.системы существенно различны, если существует АКАМП, решимая при одной нумерации и не решимая при другой. Таким образом, Мальцев нашел первый пример существенно различных конструктивизаций; ценность этого примера в том, что и соответствующая а.система и существенно различающая конструктивизации АКАМП не были построены искусственно, а обе были обнаружены в математической действительности.

Отношение "не быть существенно различными" есть некоторое отношение эквивалентности на нумерациях и, в частности, на конструктивизациях данной а.системы. Другое отношение эквивалентности на нумерациях - это отношение эквивалентности по Колмогорову, известное нам из ч. I, § 15. Первое из этих отношений крупнее, или грубее, второго: если две нумерации эквивалентны по Колмогорову, они не могут быть существенно различ-

ными. Между этими двумя отношениями можно расположить еще ряд эквивалентностей, так и иначе отражающих алгоритмическую природу рассматриваемой а.системы. К их изложению мы сейчас и перейдем.

Итак, фиксируем какую-либо а.систему с носителем M и рассмотрим различные ее нумерации (в частности, конструктивизации). Для каждой из таких нумераций ν построим ее кортежное распространение $\overline{\nu}$. Если $P \subset M^\infty$, то $\overline{\nu}^{-1}(P) \subset \mathbb{N}^\infty$; говоря в дальнейшем о характеристической функции множества $\overline{\nu}^{-1}(P)$, мы имеем в виду либо функцию, определенную на $\mathbb{N}^\infty$, либо - это безразлично - функцию, определенную на некотором ансамбле, включающем $\mathbb{N}^\infty$. Пусть теперь α и β - две нумерации нашей а. системы. Фиксируем некоторый способ программирования и будем говорить, что α программно сводится к β, коль скоро существует алгоритм, обладающий следующим свойством: всякий раз, как p оказывается программой характеристической функции множества $\overline{\alpha}^{-1}(P)$, где P есть некоторое устойчивое подмножество множества M^∞, алгоритм дает результат в применении к p, и этим результатом служит какая-то программа характеристической функции множества $\overline{\beta}^{-1}(P)$. В силу взаимной транслируемости способов программирования (см. ч. I, § 14), отношение программной сводимости не зависит от выбора способа программирования.

Будем говорить, что α равномерно сводится к β, коль скоро существует вычислимая (в смысле § 13 из ч. I) операция, преобразующая характеристическую функцию множества $\overline{\alpha}^{-1}(P)$ в характеристическую функцию множества $\overline{\beta}^{-1}(P)$ для любого устойчивого множества P. Очевидно, что из равномерной сводимости вытекает сводимость программная (см. ч. I, § 14). Каждое из этих понятий сводимости приводит к своему понятию эквивалентности: две нумерации программно (соответственно, равномерно) эквивалентны, если они программно (соответственно, равномерно) сводятся друг к другу. Из равномерной эквивалентности вытекает программная эквивалентность, а из последней - отношение "не быть существенно различными". Каждая из этих эквивалентностей в той или иной степени отражает интуитивные представления об эквивалентности нумераций с точки зрения АКАМП. Быть может, все три указанные эквивалентности в действительности совпадают (на конструктивизациях или даже на произвольных ну-

мерациях) - но мы этого не знаем. Они заведомо совпадают на множестве всех конструктивизаций (даже на множестве всех разрешимых нумераций) любой а.системы, не имеющей нетривиальных автоморфизмов.

Алгебраические системы, не имеющие нетривиальных автоморфизмов, называются жесткими. Для нежестких а.систем важную роль играет введенное Мальцевым в [Маль 62а] понятие автоэквивалентности нумераций: натуральные нумерации α и β некоторой а.системы автоэквивалентны, если существует такой автоморфизм σ этой а.системы, что нумерации $\sigma \circ \alpha$ и β эквивалентны по Колмогорову (см. ч. I, § 15). В частности, любые эквивалентные по Колмогорову нумерации автоэквивалентны, но, как замечено в [Маль 62а], обратное, вообще говоря, неверно. Если две нумерации автоэквивалентны, то они, как легко видеть, равномерно эквивалентны, а тем самым программно эквивалентны и не являются существенно различными. Верны ли обратные импликации, неизвестно; известно лишь, что неавтоэквивалентные конструктивизации могут не быть существенно различными (это показал Гончаров в личном сообщении одному из авторов от 6 июня 1981 г.). Можно показать, что если а.система жесткая, то все пять эквивалентностей: по Колмогорову, автоэквивалентность, равномерная, программная и "не быть существенно различными" - совпадают. Число классов эквивалентности, на которые разбивается множество всех конструктивизаций, естественно в этом случае называть алгоритмической размерностью рассматриваемой а.системы. В общем случае, не предполагающем жесткости, понятие алгоритмической размерности распадается на пять понятий: "существенная размерность", "программная размерность", "равномерная размерность", "авторазмерность", "колмогоровская размерность". Понятие авторазмерности ввел в науку и изучил Гончаров (см. [Гон 80], [Гон 80б]). В [Гон 81] авторазмерность названа "алгоритмической размерностью"; в нашем понимании это одна из алгоритмических размерностей.

Известно (см., например, [Гон 76]), что авторазмерность любой а.системы не более, чем счетна; колмогоровская размерность может быть и континуальной. Как установлено в основной теореме из [Гон 80], для любого кардинала $\varkappa \leq \aleph_0$ существует жесткое частично упорядоченное множество авторазмерности $\varkappa$

(в виду жесткости здесь можно говорить об алгоритмической размерности). В то же время для любого кардинала $\varkappa \leq 2^{\aleph_0}$ существует реляционная структура колмогоровской размерности $\varkappa$ ([Гон 80, следствие]).

Для произвольной фиксированной а.системы очевидны нестрогие неравенства: (существенная размерность) $\leq$ (программная размерность) $\leq$ (равномерная размерность) $\leq$ (авторазмерность) $\leq$ (колмогоровская размерность). Естественно возникают следующие два вопроса.

Первый вопрос. Бывают ли а.системы, для которых один из этих знаков "$\leq$" можно заменить на "$<$"? Здесь известно лишь следующее: 1) существуют а.системы, для которых последний из знаков "$\leq$" можно заменить на "$<$" (такова, например, изучавшаяся Мальцевым группа R_∞); 2) существует реляционная а.система, имеющая существенную размерность 2 и авторазмерность $\aleph_0$ (пример такой структуры приведен в вышеуказанном сообщении Гончарова от 6 июня).

Второй вопрос. Когда знаки "$\leq$" можно заменить на знаки "$=$"? Как уже отмечалось, все знаки нестрогого неравенства можно заменить на знаки равенства в случае жесткой а.системы. Очевидно также, что первые три знака можно заменить на знак равенства в случае, когда авторазмерность равна единице. Алгебраические системы авторазмерности единица Мальцев в [Маль 62а] назвал автоустойчивыми; в той же работе он отметил, что для $p \in \mathbb{N}$ всякая полная p-примитивная абелева группа конечного ранга является автоустойчивой. Критерий автоустойчивости для конструктивизируемых булевых алгебр найден в [Гон 75, теорема 5]: этот критерий состоит в конечности алгебры. А вот любопытные условия, достаточные для того, чтобы конструктивизируемая реляционная а.система имела бесконечную авторазмерность: структура должна иметь как сильную конструктивизацию, так и конструктивизацию, не являющуюся сильной (см. [Гон 75а, следствие 3.6]).

Конструктивные и конструктивизируемые модели

В классической теории моделей центральной проблемой является выяснение связей между свойствами какого-либо множества формул S и устройством класса всевозможных моделей для S. Аналогично, в теории конструктивных моделей интересуются уст-

ройством класса всех конструктивных или класса всех сильно конструктивных моделей множества S. (Конструктивная или сильно конструктивная модель для множества формул S - это конструктивная или, соответственно, сильно конструктивная нумерованная а.система $\langle \mathfrak{M}, \nu \rangle$, такая, что $\mathfrak{M}$ есть модель для S в обычном смысле - т.е. все формулы из S истинны в $\mathfrak{M}$). Если S есть теория, т.е. непротиворечивая совокупность формул, замкнутая относительно следствий, то необходимым и достаточным условием наличия у S сильно конструктивных моделей служит разрешимость S (необходимость очевидна и отмечена выше, о достаточности см. [ЕршО 74, гл. 2, § 3, предложение 1], [ЕршО 80, гл. 6, § 2, предложение 3]). Требование, чтобы S образовывало теорию, т.е. было замкнутым относительно следствий, является существенным: можно предъявить непротиворечивое разрешимое, даже одноэлементное S (т.е. попросту формулу), не допускающее конструктивных, тем паче сильно конструктивных, моделей. Первые примеры непротиворечивых формул, не имеющих конструктивных моделей, были предложены в [Кра 53] и [Мост 53]; более того, такой пример можно найти среди формул, содержащих всего только один, и притом бинарный, предикат (см. [Раб 58]). Каждый из трех перечисленных примеров получался конъюнкцией подходящей системы аксиом теории множеств, однако такие примеры возможны и на чисто арифметической основе; см. [Мост 55], [Баур 74], [ЕршО 74, гл. 6, § 2].

Существуют теории, всякая счетная модель которых сильно конструктивизируема: как показывает упомянутое выше предложение из [ЕршО 74, гл. 2, § 3] и [ЕршО 80, гл. 6, § 2], такой является всякая разрешимая теория, категоричная в счетной мощности. Оказывается, что тем же свойством обладает и всякая разрешимая теория, категоричная в некоторой несчетной мощности, см. [ЕршО 74, гл. 3, § 1, стр. 74]. Бывает и иначе, одни счетные модели данной теории допускают конструктивизацию, а другие - нет. В частности, для любого $n \geq 3$ существует полная разрешимая теория, имеющая точно n попарно неизоморфных счетных моделей, из которых конструктивизируема только одна (причем эта модель сильно конструктивизируема), см. [Пер 73]. Существует полная разрешимая теория, число сильно конструктивизируемых моделей которой (разумеется, с точностью до изомор-

физма) равно двум, см. [Мил 79].

Как отмечает Мостовский в своем обзоре [Мост 66, лекция 6], "не вполне ясно, что вызывает такое своеобразное поведение различных аксиоматических теорий, препятствуя одним из них иметь рекурсивные модели вообще, а другим - иметь более одной такой модели".

Пожалуй, одним из наиболее принципиальных вопросов математики является вопрос о том, сколько существует различных (т.е. неизоморфных) конструктивных моделей аксиоматической арифметики. Под аксиоматической арифметикой мы понимаем обычную систему аксиом для сложения и умножения, включая аксиомную схему индукции. По крайней мере одна конструктивизируемая модель у такой аксиоматической системы существует - это обычный натуральный ряд. Как известно, у аксиоматической арифметики существуют нестандартные (т.е. не изоморфные натуральному ряду), в том числе счетные нестандартные модели. Нестандартные модели существуют у любой (даже неразрешимой) системы аксиом арифметики - лишь бы аксиомы были записаны на элементарном (узком, I-го порядка) языке и выражали утверждения, истинные в обычном натуральном ряду (например, в качестве аксиом можно взять все формулы, истинные в натуральном ряду). Это обстоятельство естественно интерпретировать как невозможность описать натуральный ряд никакой системой аксиом. Однако, если ограничиться лишь конструктивизируемыми моделями, ситуация кардинально меняется: для аксиоматической арифметики возможна только одна, с точностью до изоморфизма, конструктивизируемая модель: обычный натуральный ряд. Это утверждает теорема Тенненбаума, объявленная в качестве теоремы 4.3 в [Ско 61] (доказательство имеется в [Коэ 66, гл. I, § 11]). Вот короткая формулировка теоремы Тенненбаума: никакая нестандартная модель арифметики в сигнатуре сложения и умножения не является конструктивизируемой. Ввиду фундаментальной значимости этой теоремы сформулируем ее более явно.

Теорема Тенненбаума. Рассмотрим какую-либо счетную модель аксиоматической арифметики, в которой двухместные функции s и p служат соответственно интерпретацией знаков "+" и "·". (т.е. для s и p выполняются обычные аксиомы, выражающие свой-

ства сложения и умножения, включая аксиомную схему индукции). Пусть существует такая однозначная натуральная нумерация ν носителя модели, что функции f и g из $\mathbb{N}$ в $\mathbb{N}$, заданные равенствами

$$\nu(f(m, n)) = s(\nu(m), \nu(n))$$
$$\nu(g(m, n)) = p(\nu(m), \nu(n)),$$

вычислимы. Тогда рассматриваемая модель изоморфна натуральному ряду. (Как доказано в [Тве 82], для существования указанного изоморфизма достаточно, чтобы хотя бы одна из функций f и g оказалась вычислимой.) Другими словами, не существует однозначной нумерации нестандартной модели арифметики, относительно которой обе функции s, p (Тенненбаум) или даже хотя бы одна из них (Тверской) были бы вычислимыми (определение см. в ч. I, § 15). Поскольку всякая разрешимая нумерация бесконечного множества эквивалентна однозначной, не существует и разрешимой нумерации с такими свойствами.

Таким образом, весь эффект нестандартных моделей объясняется тем, что мы допускаем невычислимые s и p в качестве интерпретаций для " + " и "$\cdot$" . Если же ограничиться рассмотрением только вычислимых "сложения" и "умножения" (а разве бывают другие?), нестандартные модели исчезают, и, ко всеобщему удовлетворению, натуральный ряд оказывается полностью описанным аксиоматической арифметикой.

ДОБАВЛЕНИЕ К § 5. РАСШИРЕНИЯ КОНСТРУКТИВНЫХ СТРУКТУР

Истоки общей теории конструктивных структур лежат в теории конструктивных полей, см. [Раб 60], где, в частности, было введено понятие допустимой ⟨"admissible"⟩ одно-однозначной нумерации какой-либо алгебры: допустимость означает, что операциям алгебры соответствуют вычислимые функции на номерах. Для классической теории полей весьма типичным является рассмотрение различных расширений. В случае конструктивных структур возникает естественный вопрос о возможности продолжения конструктивизации на расширение. Во многих важных случаях любая конструктивизация исходной структуры допускает такое продолжение. Например, всякая конструктивизация поля продолжается на его алгебраическое замыкание, см. [Раб 60], [ЕршЮ 80, гл. 6,

§ 1., предложение 6; следствие 1 теоремы 2], то же верно для вещественного замыкания упорядоченного поля - [ЕршЮ 74, гл.3, § 1, предложение 10; § 4, теорема 3]. В случае произвольного алгебраического расширения F' поля F необходимое и достаточное условие продолжаемости конструктивизации $\nu:\mathbb{N}\to F$ до конструктивизации поля F' состоит в перечислимости множества тех многочленов $f \in F[x]$, которые имеют корень в F', см. [ЕршЮ, гл. 6, § 3, теорема 4]. Однако, конечно, вопрос о продолжаемости конструктивизации интересен не только для полей. Например, для всякой локально нильпотентной группы без кручения ее конструктивизация продолжается до конструктивизации ее пополнения, см. [ЕршЮ 74, гл. 3, § 3], [ЕршЮ 80, гл. 6, § 3, теорема 2].

Вообще, пусть задана структура $\mathfrak{M}$, ее конструктивизация ν и расширение этой структуры $\mathfrak{M}'$. Существует ли продолжение конструктивизации ν на $\mathfrak{M}'$? В широком классе ситуаций, в частности, для всех приведенных выше примеров, положительный ответ на этот вопрос позволяет дать теорема Ершова о ядре, см. [ЕршЮ 72], [ЕршЮ 74, гл. 3, § 9], [ЕршЮ 80, гл. 6, § 3]. Приведем одну из возможных схем применения этой теоремы, формулируя соответствующие понятия не в наибольшей общности. Начнем с понятия ядра. Грубо говоря, $\mathfrak{M}'$ является ядром, если $\mathfrak{M}'$ - наименьшее в некотором классе расширение $\mathfrak{M}$. Более подробно, класс расширений состоит из всех расширений структуры $\mathfrak{M}$, удовлетворяющих подходящей системе аксиом T. Например, если $\mathfrak{M}$ - группа без кручения, в качестве расширений можно рассмотреть всевозможные полные группы без кручения, содержащие $\mathfrak{M}$; если $\mathfrak{M}$ - упорядоченное поле, в качестве расширений можно рассмотреть все вещественно замкнутые расширения поля $\mathfrak{M}$ и т.д. Для того, чтобы структура $\mathfrak{M}'$ была ядром, нужно, помимо системы T, так подобрать совокупность Φ формул с одной свободной переменной в языке структуры $\mathfrak{M}$, чтобы выполнилось следующее. Если $\mathfrak{M}''$ - произвольное расширение $\mathfrak{M}$ из рассматриваемого класса, то $\mathfrak{M}'$ можно так вложить в $\mathfrak{M}''$, что образ при этом вложении будет совпадать с объединением множеств истинности всех тех формул из Φ, для которых эти множества оказываются конечными. Теорема о ядре утверждает: если $\mathfrak{M}'$ - ядро для некоторых породимых T и Φ, то конструктивизация ν продолжается

на $\mathfrak{M}'$.

§ 6. ПРИЛОЖЕНИЯ К ТЕОРИИ ВЕРОЯТНОСТЕЙ: ОПРЕДЕЛЕНИЯ СЛУЧАЙНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

Рассмотрим бесконечные последовательности, составленные из букв какого-либо конечного алфавита - например, двоичные последовательности. Наша интуиция выделяет среди таких последовательностей случайные. Традиционная теория вероятностей оказывается бессильной перед подобной задачей: она не в состоянии определить, что такое индивидуальная случайная последовательность. Теория вероятностей вообще ничего не утверждает ни про какую отдельную последовательность, а только про совокупности таких последовательностей. Если в теории вероятностей и говорят "возьмем случайную последовательность", то это всего лишь вольность речи, "abus de langage" по Бурбаки: когда в дальнейшем про эту "взятую" последовательность нечто утверждается, точный смысл делаемого утверждения состоит в том, что некое свойство выполняется для "подавляющего большинства" последовательностей. В то же время ясно, что задача дать математическое определение понятию "случайная последовательность" важна и с методологической, и с практической точки зрения: говоря о практической точке зрения, мы имеем в виду прежде всего использование метода Монте-Карло. Впервые эта задача была рассмотрена фон Мизесом в [Миз 19]; о его подходе к определению понятия случайной последовательности (в терминологии Мизеса - "Kollektiv") см. далее.

Одно из наиболее выдающихся применений теории алгоритмов состоит как раз в том, что эта теория предлагает определение индивидуальной случайной последовательности - определение, которое, по-видимому, можно рассматривать как окончательное. Мы имеем в виду определение случайности по Колмогорову и равносильное ему определение случайности по Мартин-Лёфу. Эти опредения приводятся ниже.

К понятию случайной последовательности можно подойти с трех сторон. Мы назовем эти подходы частотным, сложностным и количественным.

Частотный подход основан на том, что в случайной последо-

вательности должна соблюдаться устойчивость частот; более того, эта устойчивость должна иметь место и для любой "законной" подпоследовательности рассматриваемой последовательности. Так, в случайной двоичной последовательности, в которой знаки 0 и 1 появляются независимо и с равными вероятностями, эти знаки должны быть распределены равномерно не только в самой последовательности, но и в любой ее подпоследовательности, выделенной каким-нибудь правилом. Таким образом, частотный подход отражает требование, чтобы в случайной последовательности отсутствовали какие-либо выраженные закономерности.

Сложностный подход основан на том, что случайная последовательность должна иметь сложное строение, а именно, энтропии ее начальных отрезков должны быть достаточно велики.

Наконец, количественный подход основан на том, что случайных последовательностей очень много, а неслучайных - очень мало. Более точно - в терминах традиционной теории вероятностей - последовательность случайна с вероятностью единица; поэтому этот подход можно было бы назвать также "теоретико-вероятностным" или "теоретико-мерным", имея в виду, что распределение вероятностей - это просто мера на пространстве всех последовательностей.

Перечисленные три содержательные подхода конкретизируются в излагаемых ниже точных алгоритмических определениях. Но сперва некоторые комментарии. Прежде чем давать обещанные алгоритмические определения, отметим, что понятие случайности очевидным образом зависит от заданного распределения вероятностей на множестве Ω всех двоичных последовательностей (для простоты мы рассматриваем только двоичные последовательности). Последовательность со значительным преобладанием нулей заведомо неслучайна при одинаковых вероятностях появления 0 и 1, но может оказаться случайной в других ситуациях. В качестве допустимых распределений вероятностей в этом параграфе рассматриваются только вычислимые распределения. Вычислимое распределение вероятностей на Ω - это вычислимая мера на Ω, подчиненная условию $\mu(\Omega) = 1$; мера μ на Ω называется вычислимой, если существует алгоритм, дающий по всякому двоичному слову $x \in \Xi$ программу числа $\mu(\Gamma_x)$, где $\Gamma_x = \{\omega \in \Omega \mid \omega$ является продолжением $x\}$.

Простейший класс вычислимых распределений образуют вычислимые бернуллиевые меры. Бернуллиева мера возникает, если 0 и 1 появляются независимо с вероятностями p и q: тогда, если слово $x \in \Xi$ содержит m нулей и n единиц, $\mu(\Gamma_x) = p^m q^n$. Если вероятности p и q суть вычислимые действительные числа, бернуллиева мера является вычислимой. Важнейший частный случай вычислимой бернуллиевой меры - равномерная бернуллиева мера, для которой $p = q = 1/2$, $\mu(\Gamma_x) = 2^{-l(x)}$.

Если ω - последовательность, то через $\omega(i)$ обозначается ее i-ый член, а через $(\omega)_n$ - ее начальный отрезок длины n, т.е. слово $\omega(0)\omega(1)\ldots\omega(n-1)$.

Мы переходим теперь к вариантам определения случайности, воплощающим на основе теории алгоритмов изложенные выше три подхода.

Частотный подход

Этот подход был предложен фон Мизесом в [Миз 19], [Миз 28]. Он применим только в случае бернуллиевой меры и не ясно, как его перенести на произвольные распределения вероятностей (даже на вычислимые распределения).

Пусть фиксированы числа p и q, лежащие в интервале $(0,1)$, и $p+q=1$. Бесконечная двоичная последовательность ω называется случайной по бернуллиевой мере, соответствующей данным p и q, если для любой бесконечной последовательности χ, полученной из ω с помощью некоторого допустимого правила выбора, средняя частота единиц в χ, т.е. предел

$$\lim_{n\to\infty} (\chi(0)+\ldots+\chi(n-1))/n \ ,$$

существует и равна q.

В приведенной формулировке многое требует уточнения. Прежде всего нужно объяснить, что мы называем правилом выбора и как с помощью правил выбора из одной последовательности получить другую. Дадим формальные определения соответствующих понятий. Условимся называть правилом выбора любое отображение, сопоставляющее с каждой бесконечной последовательностью нулей и единиц некоторую (возможно, конечную) последовательность натуральных чисел. Эту последовательность мы будем рассматривать как последовательность номеров членов, выбираемых из ис-

ходной последовательности. Более точно, пусть дано правило выбора F и последовательность нулей и единиц α. Пусть n(0), n(1),... - последовательность натуральных чисел, являющаяся значением F на α. Тогда последовательность $\alpha(n(0))\alpha(n(1))\ldots$ мы будем называть последовательностью, полученной с помощью F из α. Таким образом, с каждым правилом выбора естественно связывается отображение множества Ω бесконечных последовательностей 0 и 1 в множество $\Xi \cup \Omega$ конечных и бесконечных последовательностей 0 и 1. Назвав всевозможные всюду определенные отображения Ω в $\Xi \cup \Omega$ трансформациями, можно сказать, что каждому правилу выбора соответствует трансформация, переводящая последовательность α в последовательность, полученную из α с помощью правила выбора F.

Итак, мы объяснили, что такое правило выбора и как с его помощью из одной последовательности получить другую. Осталось теперь выделить среди правил выбора допустимые - и данному выше определению случайности будет придан точный смысл. Существуют различные варианты такого выделения - и, следовательно, различные определения случайности в рамках частотного подхода. Чем шире класс допустимых правил выбора, тем больше требований предъявляется к случайной последовательности и тем уже получается класс случайных последовательностей.

Осталось дать точные определения различных классов допустимых правил выбора - и формальное определение различных классов случайных последовательностей в рамках частотного подхода будет завершено. Но мы отложим их, а сейчас дадим некоторые неформальные комментарии (в частности, исторического характера). Точные определения будут даны позже, и каждое из них будет начинаться словами: "определение случайности по Мизесу - ..."

Наши комментарии начнем с того, что рассмотрим несколько примеров правил, которые окажутся допустимыми при любом из приводимых далее определений. Первое правило ставит в соответствие любой последовательности нулей и единиц последовательность 0, 1, 2, 3,... С его помощью из любой последовательности получается она сама, так что задаваемая им трансформация - тождественная. Чуть более сложно устроено второе правило, ставящее в соответствие любой последовательности нулей и единиц последова-

тельность 0, 2, 4, ... Соответствующая трансформация состоит в выбрасывании всех членов $\omega(i)$ с нечетными i. Приведем теперь пример правила выбора, значение которого на последовательности ω зависит от этой последовательности: каждой последовательности $\omega = \omega(0)\omega(1)\ldots$ ставится в соответствие возрастающая последовательность, состоящая из всех тех n, для которых $\omega(n-1)=0$. Соответствующая трансформация такова: взяв исходную последовательность нулей и единиц, нужно оставить в ней члены, идущие после нулей, а остальные вычеркнуть. А вот пример правила, не являющегося допустимым ни при каком из приводимых ниже определений допустимости: заменим в описании предыдущего правила выбора условие "$\omega(n-1)=0$" на условие "$\omega(n) = 0$". Тогда соответствующая трансформация будет отображать любую последовательность в последовательность из одних нулей (конечную, если в исходной последовательности было конечное число нулей).

Исторически первое определение допустимого правила выбора было предложено А.Чёрчем в [Чёрч 40] (его формулировку можно найти также в [Март 68] и в [Кнут 69, п. 3, 5С, определение R5]). Допустимые по Чёрчу правила выбора ставят в соответствие элементам Ω некоторые возрастающие последовательности натуральных чисел. Поэтому соответствующие трансформации сопоставляют с каждой последовательностью ее подпоследовательность, получающуюся из исходной отбрасыванием некоторых членов. При этом оставленные члены идут в том же порядке, в котором они шли в исходной последовательности. Вопрос об оставлении или выбрасывании данного члена последовательности решается алгоритмически в зависимости от значений предыдущих членов последовательности. Возникающее понятие случайности (точное определение которого будет дано ниже) мы будем называть случайностью по Мизесу - Чёрчу.

Недостатки этого определения демонстрируются двумя примерами. Первый пример содержится в построениях Вилля (см. [Март 68], [Кнут 69, п. 3.5, упр. 31], [Яко 70]). Именно, из них вытекает существование случайной по Мизесу - Чёрчу последовательности (при $p=q=\frac{1}{2}$), любой начальный отрезок которой содержит больше нулей, чем единиц. Это, по-видимому, противоречит нашей интуиции; кроме того, мера множества последователь-

ностей с таким свойством равна нулю. Второй пример был построен Лавлэндом [Лавл 66, § 3], указавшим случайную (по Мизесу - Чёрчу) последовательность, становящуюся неслучайной после вычислимой перестановки ее членов. Это также противоречит нашей интуиции.

Колмогоров в [Колм 63, замечание 2] предложил модифицировать определение Чёрча, расширив класс допустимых правил выбора. Допустимое в смысле Колмогорова правило относит каждой последовательности некоторую конечную или бесконечную последовательность $k(0)k(1)k(2)\ldots$, причем функция k инъективна ($k(i)\neq k(j)$ при $i\neq j$), но не обязана быть монотонной (в случае определения Чёрча k монотонна). Точное определение допустимости по Колмогорову см. ниже. Благодаря этому расширению пример, аналогичный примеру Лавлэнда, становится невозможным (случайная по Колмогорову последовательность, как нетрудно доказать, остается случайной после вычислимой перестановки ее членов). Можно ли построить аналог примера Вилля для такого определения допустимости, неизвестно. Определение Колмогорова было впоследствии независимо найдено Лавлэндом (см. [Лавл 66а, с. 499]). Поэтому соответствующий класс случайных последовательностей мы будем называть классом случайных по Мизесу - Колмогорову - Лавлэнду последовательностей.

Сходное определение случайности приводит Д.Кнут [Кнут 69, п. 3.5С, определение R6]. В [Усп Сем 81] разница между определением R6 Кнута и определением Мизеса - Колмогорова - Лавлэнда была не замечена авторами, в результате чего случайными по Мизесу - Колмогорову - Лавлэнду ошибочно были названы последовательности, удовлетворяющие требованиям определения R6 Кнута. Последовательности, удовлетворяющие определению Кнута, мы будем называть случайными по Мизесу - Кнуту.

Всякая случайная по Мизесу - Колмогорову - Лавлэнду последовательность является случайной по Мизесу - Кнуту; всякая последовательность, случайная по Мизесу - Кнуту, случайна по Мизесу - Чёрчу (см. [Кнут 69, п. 3.5С]). Не всякая случайная по Мизесу - Чёрчу последовательность случайна по Мизесу - Кнуту: это следует из существования примера Лавлэнда и из того, что случайность по Мизесу - Кнуту сохраняется при вычислимой перестановке. Существуют ли последовательности, случайные по

Мизесу - Кнуту, но не случайные по Мизесу - Колмогорову - Лавлэнду, авторам неизвестно.

В качестве примеров "эмпирических случайных последовательностей ⟨empirische Kollektive⟩ " Мизес рассматривал последовательности, возникающие в азартных играх (при бросании костей и т.п.). Отсутствие правила выбора, позволяющего получить последовательность с "анормальным" распределением нулей и единиц, Мизес интерпретировал как "невозможность системы игры". Формулировка определений случайности по Мизесу - Чёрчу и Мизесу - Колмогорову - Лавлэнду в терминах игр имеется в [Шень 82, § 3 и § 6].

До сих пор мы обсуждали различные частотные определения случайности, не приводя точных формулировок. Теперь настало время их привести. Напомним, что для определения того или иного варианта случайности необходимо указать, какие правила выбора считаются допустимыми (при этом варианте).

Мы предполагаем, что на пространстве Ω задана вычислимая бернуллиева мера и вероятности появления 0 и 1 равны p и q соответственно.

Определение случайности по Мизесу - Чёрчу. Допустимое правило выбора c_S задается разрешимым подмножеством $S \subset \Xi$. Значением этого правила на последовательности ω будет возрастающая последовательность k, которая включает те и только те натуральные числа n, для которых $(\omega)_n \in S$. Таким образом, последовательность ω называется случайной по Мизесу - Чёрчу, если для всякого разрешимого подмножества $S \subset \Xi$, для которого подпоследовательность ω^S, полученная из ω с помощью c_S, бесконечна, средняя частота единиц в этой подпоследовательности существует и равна q.

Определение случайности по Мизесу - Колмогорову - Лавлэнду. Приводя это определение, мы сознательно отказываемся комментировать содержательный смысл допустимых по Колмогорову - Лавлэнду правил. Такие комментарии можно найти в [Шень 82, § 6]; следует иметь в виду, что функции, обозначаемые нами далее через f и g, там обозначены через G и H. Итак, вот формальное определение. Допустимое правило выбора $K_{f,g}$ задается двумя вычислимыми функциями f и g из Ξ в $\mathbb{N}$. Чтобы применить его

к последовательности ω, надо вначале образовать последовательность k с помощью рекуррентной формулы

$$k(n) = f(\omega(k(0))\ \omega(k(1)) \ldots \omega(k(n-1))),$$

применяемой до тех пор, пока $k(0), \ldots, k(n)$ определены и различны. Как только появится первое такое n, что $k(n)$ не определено или совпадает с $k(s)$ при некотором $s<n$ (если такие n вообще существуют), процесс образования последовательности k прекращается; в этом случае k оказывается конечной (а именно, n-членной). Затем нужно исключить некоторые члены из последовательности k (не меняя порядка оставшихся). Именно, нужно оставить те и только те члены $k(m)$, для которых $g(\omega(k(0))\ldots \ldots \omega(k(l-1)))$ определено при всех $l \le m$ и $g(\omega(k(0)) \ldots \ldots \omega(k(m-1)))=0$. Полученная последовательность k' будет значением допустимого по Колмогорову - Лавлэнду правила $K_{f,g}$ на последовательности ω. Таким образом последовательность ω называется случайной по Мизесу - Колмогорову - Лавлэнду, если для всяких вычислимых функций f и g, для которых последовательность, полученная из ω с помощью правила $K_{f,g}$, бесконечна, средняя частота единиц в этой последовательности существует и равна q.

Определение случайности по Мизесу - Кнуту (определение R6 в [Кнут 69, п. 3.5С]) получается из определения Мизеса - Колмогорова - Лавлэнда, если рассматривать только всюду определенные функции g и соответствующие им правила. (Функция f попрежнему может быть не всюду определенной.)

Каким должен быть класс всех допустимых правил выбора?

Мы видели, каким образом определяется понятие случайной последовательности, если выбран некоторый класс правил выбора, объявленных допустимыми. Естественно желать (на это указывал еще фон Мизес), чтобы применение допустимого правила выбора к случайной последовательности (случайной относительно д а н н о г о класса допустимых правил) давало бы либо конечную последовательность, либо последовательность, которая имеет не только заданный предел частот (что гарантируется определением случайности), но и сама является случайной (относительно т о г о ж е класса правил). Это требование очевидно выполнено, если класс трансформаций, соответствующих допустимым правилам выбора, замкнут относительно композиции. [Это озна-

чает, что для всяких трансформаций T_1 и T_2, соответствующих допустимым правилам выбора, существует трансформация T_3, соответствующая некоторому допустимому правилу выбора и такая, что T_3 есть композиция T_2 и T_1, т.е. такая, что результат применения трансформации T_3 к любой последовательности ω, для которой $T_1(\omega)$ - бесконечная последовательность, совпадает с $T_2(T_1(\omega))$.] В самом деле, если класс трансформаций замкнут и если с помощью допустимого правила выбора из случайной последовательности ω получается последовательность ω_1, то ω_1 также случайна: действительно, любая последовательность ω_2, которая может быть получена из ω_1 с помощью допустимого правила выбора, имеет надлежащую среднюю частоту, так как может быть (в силу замкнутости) получена с помощью допустимого правила и непосредственно из ω.

Таким образом, разумно предъявлять к классу допустимых правил выбора такое требование:

(ТЗам) класс трансформаций, соответствующих всем допустимым правилам выбора, должен быть замкнут относительно композиции.

Мы сейчас сформулируем некоторое достаточное условие для этого.

Определим понятие композиции для правила выбора. Пусть даны три правила выбора F_1, F_2 и F_3. Мы хотим определить, что означает, что F_3 есть композиция правил F_2 и F_1. Пусть ω - произвольная последовательность, из которой с помощью правила выбора F_1 получается бесконечная последовательность ω'. Пусть $F_1(\omega)=k$, $F_2(\omega')=l$, $F_3(\omega) = m$ где k,l,m - последовательности натуральных чисел. Проверим, выполнено ли равенство $m=k \circ l$. Если окажется, что это равенство выполнено для всех ω, из которых с помощью F_1 получаются бесконечные последовательности, то мы будем говорить, что F_3 является композицией правил F_2 и F_1.

Легко проверить, что если F_3 является композицией F_2 и F_1, то трансформация, соответствующая F_3, является композицией трансформаций, соответствующих F_1 и F_2. Поэтому, если выполнено условие

(ВЗам) для любых допустимых правил выбора F_1 и F_2 существует допустимое правило выбора F_3, являющееся их композицией,

то выполнено и требование (ТЗам), и, следовательно, из случайной последовательности с помощью допустимого правила может получиться только случайная.

Требование (ВЗам) выполнено для класса допустимых по Чёрчу правил. Для класса правил, допустимых по Колмогорову - Лавлэнду, это не так: как доказано в [Шень 82], не выполнено (ВЗам), и даже (ТЗам). Это еще не означает, что применение допустимого по Колмогорову - Лавлэнду правила у случайной по Мизесу - Колмогорову-Лавлэнду последовательности может дать не случайную (по Мизесу-Колмогорову-Лавлэнду) последовательность. (Так ли это на самом деле, авторам неизвестно.) Однако случайность результата применения допустимого правила выбора к случайной последовательности перестает быть очевидной.

Этот же недостаток присущ и упомянутому выше определению Кнута, так что, по нашему мнению, есть основания не согласиться с замечанием Кнута о том, что его определение "удовлетворяет всем разумным философским требованиям, предъявляемым к понятию случайности".

Попытка сформулировать определение допустимости, замкнутое относительно композиции, предпринята в [Шень 82]. Прежде чем изложить определение, отметим две его особенности. Первая из них состоит в том, что рассматриваются не правила выбора, а сразу трансформации: определяется понятие допустимой трансформации. Это не мешает дать определение случайной последовательности (ведь в общей схеме Мизеса фактически используются не сами правила выбора, а лишь соответствующие им трансформации) и ставить вопрос о выполнении условия (ТЗам), однако отдаляет от первоначального замысла Мизеса и не позволяет поставить вопрос о выполнении условия (ВЗам). Вторая особенность состоит в том, что разумное определение случайности получается лишь для равномерной бернуллиевой меры (при неравномерной мере случайных последовательностей не оказывается вовсе). Зато в этой ситуации возникает класс случайных последовательностей, совпадающий с классом случайных по Мартин-Лёфу последовательностей (об этом классе см. ниже). Можно ли дать такое определение допустимой трансформации, чтобы оно приводило к классу случайных по Мартин-Лёфу последовательностей и для неравномерной меры - неизвестно. Приведем теперь определение

допустимой трансформации из [Шень 82] (там трансформации называются "правилами выбора").

Введем на $\Xi \cup \Omega$ 1) отношение порядка, считая, что $x \leq y$, если x есть начало y; 2) топологию, считая базовыми открытыми множествами множества $I_x = \{y \in \Xi \cup \Omega \mid x \leq y\}$ для всех конечных x. Всюду определенное непрерывное отображение $F: \Xi\cup\Omega \to \Xi \cup \Omega$ назовем вычислимым, если множество тех пар $\langle x,y\rangle$ конечных последовательностей, для которых $y \leq F(x)$, перечислимо, и регулярным, если для любой конечной последовательности y равномерная бернуллиева мера множества $\{\omega \in \Omega \mid F(\omega) \geq y\}$ не превосходит меры множества Γ_y, т.е. числа $2^{-l(y)}$, где $l(y)$ - длина y. Допустимой трансформацией назовем ограничение на Ω произвольного непрерывного вычислимого регулярного отображения. Утверждение о том, что при таком определении допустимости класс случайных последовательностей совпадает с классом случайных по Мартин-Лёфу последовательностей, можно легко вывести из результатов [Шно 71]; см. также [Шень 82, § 9].

Сложностной подход

Этот подход, предложенный Колмогоровым в [Колм 63], [Колм 65], [Колм 69], связан с материалом ч. I, § 17. Колмогоров исходил из представления о таблице случайных чисел как о длинной, но конечной последовательности знаков (для наглядности двоичных), столь беспорядочно устроенной, что она не допускает простого описания - сложность такого описания должна быть достаточно велика, а именно, близка к длине последовательности. Случайность бесконечной последовательности означает достаточно быстрый рост энтропии начальных отрезков (см. [Колм 69, п. 2]). Оказалось, однако, что для определения случайности бесконечных последовательностей нужно использовать не простую колмогоровскую энтропию, а монотонную энтропию.

Определение случайности по Колмогорову. Рассмотрим сперва случай равномерной бернуллиевой меры. Напомним (см. ч. I, § 17, таблица), что для любой $\omega \in \Omega$ имеет место $\Xi\Xi K((\omega)_n) \underset{+}{\leq} n$. Последовательность ω называется случайной по Колмогорову, если $\Xi\Xi K((\omega)_n) \underset{+}{\geq} n$. В случае произвольного вычислимого распределения вероятностей μ можно доказать, что для любой $\omega \in \Omega$

$$\Xi\Xi K((\omega)_n) \underset{+}{\leq} -\log_2 \mu(\Gamma((\omega))_n)$$

(см. [Левин 73]). В этом общем случае последовательность ω называется случайной по Колмогорову, если

$$\Xi\Xi K((\omega)_n) \gtrsim_{+} -\log_2 \mu(\Gamma(\omega))_n).$$

В этом определении без изменения возникающего класса случайных последовательностей можно заменить $\Xi-\Xi$-энтропию на $\Xi-\mathbb{N}$-энтропию, назвав последовательность случайной, если

$$\Xi\mathbb{N}K((\omega)_n) \gtrsim_{+} -\log_2 \mu(\Gamma(\omega))_n).$$

В частности, если μ – равномерная бернуллиева мера (вероятности появления 0 и 1 равны), то последовательность будет случайной, если выполнено любое из эквивалентных свойств: $\Xi\Xi K((\omega)_n) \gtrsim_{+} n$ или $\Xi\mathbb{N}K((\omega)_n) \gtrsim_{+} n$ (см. [Вью 80, следствие 3.2]). Однако даже в этом простом случае нельзя заменить $\Xi-\Xi$-энтропию на $\mathbb{N}-\mathbb{N}$-энтропию: невозможна последовательность ω, для которой $\mathbb{N}\mathbb{N}K((\omega)_n) \gtrsim n$, см. [Март 66], [Зво Лев 70, теорема 2.6], [Яко 70, п. 2.2].

Количественный, или теоретико-мерный подход

Этот подход был разработан Мартин-Лёфом (см. [Март 66а], [Зво Лев 70, § 4], [Яко 70, § 4]). В [Зво Лев 70, § 4, п. I] после указания некоторых трудностей, связанных с частотным подходом, отмечается: "В 1965 г. П.Мартин-Лёфу удалось, основываясь на идеях А.Н.Колмогорова, дать свободное от подобных трудностей определение случайной последовательности. Идея А.Н. Колмогорова состояла в том, чтобы "не случайными" считать те последовательности, в которых наблюдается достаточно много закономерностей, где под закономерностью подразумевается любое проверяемое свойство последовательности, присущее лишь узкому их классу (достаточно малому по мере)". Случайными, таким образом, предполагается считать последовательности, принадлежащие широкому классу последовательностей, имеющих достаточно мало закономерностей.

Иначе говоря, при количественном подходе случайной объявляется всякая последовательность, которая выдерживает некоторые испытания на случайность, называемые тестами. Тест – это просто разбиение Ω на множество E единичной меры и множество F нулевой меры:

$$E \cup F = \Omega,\ E \cap F = \emptyset,\ \mu(E) = 1,\ \mu(F) = 0.$$

Элементы E называются выдержавшими тест. Если потребовать,

чтобы случайная последовательность выдерживала любой тест, то окажется, что случайных последовательностей не бывает. Мартин-Лёф предложил поэтому ограничиться эффективными тестами, т.е. такими, в которых F имеет эффективно нулевую меру. Теорема Мартин-Лёфа о существовании, в случае вычислимой меры μ, наибольшего множества эффективно нулевой меры (см. выше § 4) показывает, что существуют последовательности, выдерживающие любой эффективный тест: именно из таких последовательностей и состоит конструктивный носитель меры.

Определение случайности по Мартин-Лёфу. Пусть на Ω задано вычислимое распределение вероятностей. Последовательность называется случайной по Мартин-Лёфу, если она выдерживает любой эффективный тест или, что то же самое, если она принадлежит конструктивному носителю меры.

Если последовательность случайна по Мартин-Лёфу относительно равномерной бернуллиевой меры, частота нулей в ее начальных отрезках стремится к $\frac{1}{2}$; действительно, из доказательства теорем теории вероятностей можно извлечь эффективный тест, отвергающий все такие последовательности, у которых эта частота не стремится к $\frac{1}{2}$. Более того, в бернуллиевом случае всякая последовательность, случайная по Мартин-Лёфу, является случайной по Мизесу - Чёрчу (см. [Ага 75, п. 5.I]) и даже случайной по Мизесу - Колмогорову - Лавлэнду (см. [Шень 82, § 9, с. 36, замечание после формулировки теор. 1]).

Соотношения между различными определениями

Замечательным образом оказалось, что сложностной и теоретико-мерный подходы приводят к одному и тому же конечному результату. Именно, имеет место следующая основная теорема (см. ее формулировку в [Левин 73, теорема 2], [Шно 73, теорема 3], [Шно 77, теорема 4.2] и доказательство в [Шно 73], [Вью 80, теорема 3.2]:

> Для любого вычислимого распределения вероятностей последовательность тогда и только тогда случайна по Мартин-Лёфу, когда она случайна по Колмогорову.

Таким образом, определения Мартин-Лёфа и Колмогорова задают один и тот же класс последовательностей. Более широкий класс образуют последовательности, случайные по Мизесу - Кол-

могорову - Лавлэнду (как уже отмечалось), а обратное, как вытекает из сформулированных в [Колм 69] результатов, неверно. Именно, как утверждается в [Колм 69, п. 2, с. 6], существует случайная по Мизесу - Колмогорову - Лавлэнду последовательность $\omega = \omega(0)\omega(1)\ldots$, для которой

$$\mathbb{N}\mathbb{N}K((\omega)_n) \lesssim \log_2 n$$

(авторы должны со всей откровенностью признаться, что не умеют строить такой пример). Такая последовательность, как легко видеть, не может быть случайной по Колмогорову (а тем самым и по Мартин-Лёфу), так как $\mathbb{N}-\mathbb{N}$ и $\Xi-\Xi$ - энтропии слова x отличаются не более чем на $C\log_2 l(x)$, где $l(x)$ - длина слова x. (Напомним, что говоря об $\mathbb{N}-\mathbb{N}$-энтропии слов, мы имеем в виду $\mathbb{N}-\mathbb{N}$-энтропию натуральных чисел, соответствующих словам при взаимнооднозначном соответствии, описанном в ч. I, § 17).

Мы приходим, таким образом, к следующей таблице, отражающей соотношения между различными видами случайностей:

последовательности, случайные по Мартин-Лёфу

$\parallel$

последовательности, случайные по Колмогорову

$\nparallel \cap$

последовательности, случайные по Мизесу - Колмогорову - Лавлэнду

$\cap$

последовательности, случайные по Мизесу - Кнуту

$\nparallel \cap$

последовательности, случайные по Мизесу - Чёрчу

В этой таблице отмечены все известные авторам равенства и строгие включения; совпадает ли класс случайных по Мизесу - Колмогорову - Лавлэнду последовательностей с классом последовательностей, случайных по Мизесу - Кнуту, как уже отмечалось выше, авторам неизвестно.

Для любого вычислимого распределения вероятностей на Ω класс последовательностей, случайных по Колмогорову или по Мартин-Лёфу, объявляется истинным классом случайных последовательностей, а элементы этого класса - подлинно случайными

последовательностями.

Для равномерного бернуллиевого распределения предпринятая в [Шень 82] попытка уточнения определения Мизеса также приводит к этим подлинно случайным последовательностям. Однако уточнение из [Шень 82] основано на понятии трансформации, а не на понятии правила выбора (см. выше). Существует ли класс правил выбора, приводящий к истинному классу случайных последовательностей хотя бы для равномерного бернуллиевого распределения - неизвестно. Несуществование такого класса правил выбора означало бы, что частотный подход (по крайней мере в рамках замысла Мизеса) не в состоянии дать адекватное определение случайности.

Конечные последовательности с точки зрения случайности

Полезно ясно сознавать, что - при любом из перечисленных определений - если приписать к случайной последовательности спереди миллион нулей, снова получится случайная последовательность. Поэтому к практической интерпретации понятия случайной последовательности применительно к методу Монте-Карло следует относиться с осторожностью: ведь может статься, что используемая в этом методе последовательность (если полагаться на ее случайность, и ни на что более) как раз и начинается с миллиона нулей. Поэтому в прикладном аспекте наиболее существенно понятие случайной конечной последовательности (для которого понятие случайной бесконечной последовательности служит, так сказать, "аппроксимацией сверху"). Это понятие намечено Колмогоровым в [Колм 63] (см. также [Кнут 69, п. 35Е]). В [Колм 65, § 4] по этому поводу говорится (под сложностью понимается сложность относительно оптимального способа описания - то, что в нашем тексте названо энтропией): "Грубо говоря, здесь дело идет о следующем. Если конечное множество M из очень большого числа элементов N допускает определение при помощи программы длины пренебрежимо малой по сравнению с $\log_2 N$, то почти все элементы M имеют сложность K(x), близкую к $\log_2 N$. Элементы $x \in M$ этой сложности и рассматриваются как "случайные" элементы множества M". В качестве примера рассмотрим множество M, состоящее из всех последовательностей нулей и единиц длины k. Это множество содержит $N=2^k$ элементов; чтобы задать его,

достаточно указать k, для чего необходимо около $\log_2 k$ двоичных знаков (что мало по сравнению с $\log_2 N = k$). Почти все элементы M имеют сложность, близкую к $\log_2 N$, т.е. к k. Таким образом, в соответствии с приведенной только что цитатой из [Колм 65], последовательность из k нулей и единиц естественно рассматривать как тем более "случайную", чем ближе ее энтропия к k.

Разумеется, понятие энтропии определено по существу с точностью до ограниченного слагаемого, и от выбора того или иного оптимального способа описания зависит, насколько случайной окажется данная конечная последовательность: последовательность, "практически случайная" при одном оптимальном способе описания, может оказаться "совершенно неслучайной" при другом. Однако разница между энтропиями, соответствующими двум способам описания, оставаясь ограниченной, становится с увеличением числа k все менее и менее заметной по сравнению с k.

Зависимость "степени случайности" от выбора способа описания может рассматриваться как аргумент в пользу желательности фиксации какого-нибудь одного "единственно правильного", способа описания и рассмотрения соответствующей ему энтропии. Однако, как пишет Колмогоров в [Колм 65, с. 10] "сомнительно, чтобы это можно было сделать без явного произвола. Следует, однако, думать, что различные представляющиеся здесь разумные варианты будут приводить к оценкам "сложностей", расходящимся на сотни, а не десятки тысяч бит". Требования, которые следует предъявлять к "разумному" способу описания F, должны, по-видимому, включать справедливость утверждения

$$\forall x (K_F(x) \leq K_G(x) + C)$$

со сравнительно небольшими C для возможно большего числа "естественных" способов описания G. Некоторые предложения по выбору такого "разумного" способа имеются в [Левин 77].

Таким образом, выбрав способ описания конечных последовательностей, представляющийся нам более или менее разумным, мы получаем возможность с той же степенью разумности говорить о том, насколько та или иная конкретная конечная последовательность случайна.

Реально в методе Монте-Карло используются именно конечные последовательности. Можно считать, что практическим критерием

"случайности" конечной последовательности как раз служит успешность ее использования при расчетах по методу Монте-Карло; более точно, успешность использования л ю б о й случайной конечной последовательности в методе Монте-Карло естественно включать в число требований, предъявляемых к практически годному понятию случайности. А тогда отождествление случайных конечных последовательностей со сложно устроенными, т.е. последовательностями, сложность которых относительно некоторого "разумного" способа описания близка к их длине, оправдано. Действительно, при "разумном" способе описания F высокая сложность конечной последовательности может служить аргументом в пользу того, что ее использование в конкретном вычислении по методу Монте-Карло даст хороший результат. Ведь последовательности, дающие при использовании их в вычислении по методу Монте-Карло плохой результат, составляют ничтожно малую часть всех последовательностей (это - основное требование, предъявляемое к вычислениям по этому методу). Поэтому всякая "плохая" последовательность допускает такое простое задание: сначала указывается, что она плохая, а затем указывается ее порядковый номер среди всех плохих последовательностей. Для указания номера понадобится сравнительно мало двоичных знаков, так как число "плохих" последовательностей сравнительно мало. Таким образом, для "плохой" последовательности z ее сложность $K_G(z)$ невелика (здесь G - только что рассмотренный способ описания с помощью "простых заданий"). Поскольку способ описания F "разумен", можно надеяться, что при небольшом C и любом x выполнено неравенство $K_F(x) \leq K_G(x) + C$. В этом случае окажется, что сложность (относительно F) любой последовательности, дающей плохой результат при использовании ее в вычислениях по методу Монте-Карло, невелика.

В заключение приведем следующее замечание, высказанное Колмогоровым в январе 1965 г. в его публичной лекции в Московском университете: попытка обнаружить высокоразвитую внеземную цивилизацию, основанная на перехвате сообщения, предназначенного для той же самой или подобной цивилизации, скорее всего обречена на провал. В самом деле, если цивилизация высоко развита, то она умеет экономно кодировать, ее сообщение имеет большую удельную сложность (= сложность, деленную на

длину сообщения) и потому практически не отличимо от случайной последовательности сигналов.

§ 7. ПРИЛОЖЕНИЯ К ТЕОРИИ ИНФОРМАЦИИ: АЛГОРИТМИЧЕСКИЙ ПОДХОД К ПОНЯТИЮ КОЛИЧЕСТВА ИНФОРМАЦИИ

Очень хочется уметь отвечать на вопрос, сколько информации несет то или иное сообщение, т.е. как-то измерять количество информации. Естественно измерять это количество длиной наиболее экономного описания рассматриваемого сообщения: с этой точки зрения космическое сообщение, обсуждавшееся в конце предыдущего параграфа, несет большую информацию - приближающуюся к максимально возможной при данной длине сообщения. Разумеется, и сам выбранный способ описания должен быть достаточно экономным, т.е. давать для всех сообщений как можно более короткие описания. Кроме того, этот способ должен позволять однозначное и эффективное восстановление сообщения по его описанию. Тем самым мы приходим к ситуации, описанной в § 17: количество информации в сообщении - это его энтропия. Напомним, что в ч. I, § 17 мы ввели разные виды энтропий, различавшихся, в частности, ансамблями описаний. Здесь мы будем использовать в качестве ансамбля описаний ансамбль $\mathbb{N}$ или (что эквивалентно в силу наличия указанного в ч. I, § 17 изоморфизма между $\mathbb{N}$ и Ξ) ансамбль двоичных слов Ξ, при том, что отношение согласованности на Ξ будет теперь не тем, которое рассматривалось в. I, § 17, а отношением равенства, т.е. таким же, как на $\mathbb{N}$. Напомним теперь определение простой колмогоровской энтропии, данное в ч. I, § 17.

Пусть Y - ансамбль сообщений, т.е. просто некоторый ансамбль, элементы которого мы называем "сообщениями". Способ описания - это перечислимое отношение $R \subset \mathbb{N} \times Y$, обладающее тем свойством, что если $\langle n, y_1 \rangle \in R$ и $\langle n, y_2 \rangle \in R$, то $y_1 = y_2$. Иначе говоря, способ описания понимается в смысле ч. I, § 17, причем отношение согласованности, заданное на каждом из ансамблей $\mathbb{N}$ и Y , совпадает с равенством. Поэтому по теореме Колмогорова среди всех способов описания существует оптимальный. Объем (напомним, что объемом числа x здесь называется целая часть числа $\log_2(x+1)$) самого короткого описания объекта $y \in Y$ при каком-либо фиксированном оптимальном способе называется простой

колмогоровской энтропией объекта $y \in Y$ (в ч. I, § 17 этот термин применялся лишь к случаю $Y = \mathbb{N}$) и обозначается $\mathbb{N}YK(y)$, а короче просто $K(y)$. Итак, количество информации в сообщении y оказывается естественным измерять простой колмогоровской энтропией этого сообщения. Так возникает алгоритмическая теория информации - см. [Бар 77].

Заметим, что количество информации в y определено лишь с точностью до аддитивной величины порядка $O(1)$. В самом деле, возможны различные оптимальные способы R_1 и R_2, приводящие, соответственно, к энтропиям K_1 и K_2. Однако в силу сказанного в ч. I, § 17, $|K_1(y) - K_2(y)| \lesssim 0$.

Далее, рассмотрим какую-либо однозначную вычислимую нумерацию ансамбля Y, т.е. вычислимое (1-1)-отображение $\mathbb{N}$ на Y. Вспомним основную лемму из ч. I, § 17 о соотношении между X-U-энтропией и X-V-энтропией; в силу этой леммы, если m есть номер сообщения y, то $|\mathbb{N}\mathbb{N}K(m) - \mathbb{N}YK(y)| \lesssim 0$. Таким образом, количество информации в сообщении совпадает с количеством информации в его номере (совпадает с точностью до величины порядка $O(1)$, но ведь и само количество информации определено с точностью до величины этого порядка). Это же рассуждение применимо и к определению количества информации в паре $\langle y_1, y_2 \rangle \in Y \times Y$. Пусть W - такой ансамбль, что $Y \times Y \subseteq W$. Количество информации в $\langle y_1, y_2 \rangle$ можно определять либо как $\mathbb{N}$-W-энтропию самой этой пары, либо же - равносильным образом - как $\mathbb{N}$-$\mathbb{N}$-энтропию номера этой пары (эти определения приводят к величинам, различающимся на ограниченное слагаемое).

Алгоритмическая теория информации была основана Колмогоровым (см. [Колм 65]) с целью придать таким интуитивным понятиям, как "количество информации" и "энтропия", точный смысл в применении к индивидуальным объектам. В традиционной (основанной на вероятности) теории информации, основанной Шенноном, эти понятия, как известно, применяются к случайным объектам, т.е., говоря более строго, к случайным величинам. Исторически сложившуюся вероятностную теорию информации правильнее было бы называть "теорией передачи информации", см. [Доб Пре 79] или "математической теорией связи" по названию основополагающей статьи Шеннона [Шенн 48]: ведь эта теория не охватывает всех (прежде всего семантических) аспектов понятия информации. Не-

ясно, насколько эти различные аспекты могут быть охвачены алгоритмической теорией информации. Во всяком случае, проблемы связи этой теории с семантикой пока даже не поставлены. (Обсуждение некоторых близлежащих проблем начато в [Манин 81]). Реальные достижения алгоритмической теории информации относятся к двум направлениям. Первое состоит в выяснении того, насколько формулы, полученные для случайных величин, оказываются справедливыми применительно к индивидуальным объектам. Второе заключается в установлении соотношений между колмогоровской и шенноновской энтропиями.

Изложим основные из этих достижений.

Согласно [Колм 69], исходным понятием теории информации (как вероятностной, так и алгоритмической) является условная энтропия объекта y при заданном объекте x. Она обозначается $H(y \mid x)$ и интерпретируется как количество информации, необходимое для задания объекта y в обстановке, когда объект x уже задан. Далее определяются:

(1) (безусловная) энтропия объекта y; она обозначается $H(y)$ и определяется равенством $H(y) = H(y \mid e)$, где e-какой-либо "заведомо заданный объект";

(2) количество информации, содержащейся в объекте x об объекте y; оно обозначается $I(x : y)$ и определяется равенством $I(x:y) = H(y) - H(y \mid x)$.

В алгоритмической теории информации x и y суть конструктивные объекты, в вероятностной теории - случайные величины; чтобы подчеркнуть это обстоятельство, в последнем случае вместо x и y будем писать ξ и η. Для простоты будем предполагать, что ξ и η принимают лишь конечное число значений. Пусть ξ принимает значения $x_1,\ldots,x_m$ с вероятностями $p_1,\ldots,p_m$, а η - значения $y_1,\ldots,y_n$ с вероятностями $q_1,\ldots,q_n$; пусть, далее, r_{ij} есть вероятность того, что одновременно $\xi=x_i$, $\eta = y_j$. Тогда, по определению

$$H(\eta \mid \xi) = - \sum_{i,j} r_{ij} \log_2 \frac{r_{ij}}{p_i} .$$

Далее, (шенноновская безусловная) энтропия $H(\eta)=- \sum_j q_j \log_2 q_j$ может быть определена, согласно (1), как $H(\eta \mid \varepsilon)$, где ε принимает одно единственное значение; оказывается, что $H(\eta)$ есть

среднее число двоичных знаков, необходимое для задания одного значения η. Наконец, $I(\xi : \eta)$ определяется в соответствии с (2). Из этих определений сразу следует, что

$$H(\eta|\eta) = 0 \qquad (P_1)$$
$$I(\xi:\eta) \geq 0 \qquad (P_2)$$
$$I(\xi:\eta) = I(\eta:\xi) \qquad (P_3)$$
$$H(\langle \xi,\eta\rangle) = H(\xi) + H(\eta|\xi) \qquad (P_4)$$

(значениями случайной величины $\langle \xi,\eta\rangle$ служат пары $\langle x_i, y_j\rangle$, принимаемые с вероятностями r_{ij}).

В алгоритмической теории информации в качестве $H(y|x)$ берется $K(y|x)$. Что такое $K(y|x)$, было разъяснено в ч. I, § 17 (переход от $\mathbb{N}$ к произвольному X с равенством в качестве отношения согласованности не составляет труда). В том же пункте отмечалось, что при фиксации x в $K(y|x)$ мы приходим к простой колмогоровской энтропии:

$$K(y) \underset{+}{=} K(y|e).$$

Итак, и в алгоритмической теории определение $H(y)(=K(y))$ через $H(y|x)(=K(y|x))$ происходит согласно (1). Далее, согласно (2), вводится количество информации $I(x:y) = K(y)-K(y|x)$.

При переходе к алгоритмической теории информации соотношения вероятностной теории претерпевают некоторые изменения, как очевидные, так и неочевидные. Очевидные изменения вызваны тем, что "все предложения алгоритмической теории информации в их общей формулировке верны лишь с точностью до членов вида $O(1)$" ([Колм 69]). Поэтому (P_1) и (P_2) заменяются на (A_1) и (A_2).

$$0 \leq K(y|y) \underset{+}{\leq} 0 \qquad (A_1)$$
$$I(x:y) \underset{+}{\geq} 0 \qquad (A_2)$$

Неочевидные изменения состоят в том, что в алгоритмических аналогах равенств (P_3) и (P_4) появляется логарифмическая поправка:

$$I(x:y) = I(y:x) + O(\log_2 K(\langle x, y\rangle)) \qquad (A_3)$$
$$K(x:y) = H(x)+H(y|x) + O(\log_2 K(\langle x, y\rangle)) \qquad (A_4)$$

Шенноновский подход можно следующим образом применить к индивидуальному слову A : можно рассмотреть это A как одно из значений некоторой случайной величины ζ и вычислить энтропию ζ, причем ζ естественно задать так, чтобы A было "типичной"

ее реализацией. Например, можно считать, что буквы слова A независимы и имеют вероятности, равные частотам их появления в A. Пусть рассматривается слово длины k в n-буквенном алфавите. Тогда ζ имеет n^k значений и $H(\zeta)$ в k раз больше, чем энтропия $H(\eta)$ случайной величины η, у которой значениями служат буквы алфавита, а вероятностями этих значений - частоты букв в слове A. Энтропию $H=H(\eta)$ естественно называть шенноновской удельной энтропией слова A. Ее содержательный смысл проявляется, например, в следующем. Рассмотрим какой-либо способ побуквенного кодирования слов в n-буквенном алфавите посредством двоичных слов, позволяющий однозначно восстанавливать слова по их кодам. Если теперь заменить в A каждую букву на ее код из Ξ, то всё A в целом также заменится на некоторое слово из Ξ длины k'. Отношение $L=\frac{k'}{k}$ есть "коэффициент удлинения". Известно (см., например, [Про 73], [Левен 74, теорема 8]), что всегда $L \geq H$ и что возможен такой способ кодирования, при котором $L \leq H+1$. Поэтому число $k \cdot H$ можно считать "истинной двоичной длиной" слова A, или его шенноновской сложностью (более точно - шенноновской 1-сложностью; если кодировать не отдельные буквы, а сразу пары соседних букв, учитывая частоты двубуквенных сочетаний, мы приходим к шенноновской 2-сложности и т.д.). Удельная шенноновская энтропия слова A показывает, таким образом, долю шенноновской сложности, приходящейся на одну букву слова, и ее естественно сравнивать с удельной колмогоровской энтропией того же слова. Имеет место следующее основное неравенство (см. [Зво Лев 70, теорема 5.1], [Бар 77]):

$$K(A)/k \leq H+(c_n \log_2 k)/k ,$$

где H - удельная шенноновская энтропия слова A, знаменатель k есть длина слова, а константа c_n зависит лишь от числа букв в алфавите и от выбранного при определении энтропии оптимального способа описания.

Другой результат о связи колмогоровской и шенноновской энтропий относится к начальным отрезкам случайных последовательностей. Пусть дана случайная величина ξ, значениями которой являются буквы данного конечного алфавита, причем для

каждой буквы вероятность принять ее в качестве значения есть вычислимое (например, рациональное) число. Рассмотрим пространство всех бесконечных последовательностей, составленных из букв рассматриваемого алфавита; в предположении, что буквы появляются независимо, ξ задает вычислимую меру на этом пространстве (бернуллиеву в случае двубуквенного алфавита). Среди последовательностей выделяются случайные (по Мартин-Лёфу или, что эквивалентно, по Колмогорову, см. предыдущий параграф). Шенноновскую энтропию $H(\xi)$ можно трактовать как "удельную шенноновскую энтропию случайной последовательности". Оказывается, что для любой случайной последовательности ω удельная колмогоровская энтропия $\frac{K((\omega)m)}{m}$ начальных отрезков этой последовательности стремится к $H(\xi)$, см. [Зво Лев 70, формула (5.18)], [Ага 75, § 5.5, стр. 134], [Бар 77].

§ 8. ОЦЕНКИ СЛОЖНОСТИ РЕШЕНИЯ ОТДЕЛЬНЫХ ЗАДАЧ

В этой области теории алгоритмов естественно выделяются задачи получения верхних и задачи получения нижних оценок. Методы решения задач этих двух категорий совершенно различны.

Верхние оценки

Верхняя оценка строится следующим образом. Указывается неформальный алгоритм вычисления требуемой функции f. Затем этот алгоритм формализуется в виде алгоритма вычисления на подходящей модели и доказывается, что сложность (время или емкость) вычисления для этого алгоритма не превосходит значения подходящей функции φ при всех значениях аргумента. Эта функция и объявляется верхней оценкой сложности вычисления функции f.

Естественно желать получать такие оценки сложности вычисления, которые соответствуют вычислительной практике. Обращаясь к обсуждавшимся в ч. I, § 16, оценкам "с точностью до", мы обнаруживаем, что столь широко понимаемые оценки не вполне удовлетворительны с практической точки зрения и могут рассматриваться только как грубые приближения к реальному положению вещей. При этом оценки "с точностью до мультипликативной константы" хуже оценок "с точностью до аддитивной константы" и т.д. На первый взгляд может показаться,что получить "абсолют-

ные оценки" и невозможно, ведь имеет место теорема о линейном ускорении, см. ч. I, § 7. Дело, однако, в том, что при "ускорении" вычисления, даваемом этой теоремой, растет сложность каждого шага вычисления. Если же ограничить сложность отдельного шага вычислительного процесса, то теорема о линейном ускорении уже не применима. И действительно, в принципе возможны абсолютные оценки времени вычисления, когда последнее понимается как сумма длительностей отдельных шагов (см. часть I, § 6), причем эти длительности оцениваются сложностью соответствующих шагов; конечно, именно такие оценки имеют наибольшее практическое значение.

С точки зрения времени вычисления самыми простыми являются функции, время (т.е. число шагов) вычисления которых совпадает с точностью до мультипликативной константы с размером аргумента и, более того, число шагов работы машины между двумя последовательными сдвигами входной головки ограничено некоторой константой (мы считаем, что вычисление происходит на многоленточных машинах Тьюринга с безвозвратной входной лентой, см. [Раб 63], [Роз 67]). Такое вычисление называется вычислением в реальное время. Как выяснилось в последние годы, среди задач, решаемых в реальное время, имеются многие интересные задачи, связанные с идентификацией слов (в частности, задача распознавания симметрии слова, см. [Сли 77]); правда, для наиболее важных из этих задач построенные алгоритмы не являются алгоритмами колмогоровского типа, см. [Сли 77а], [Сли 78]. Представляет интерес переход от таких алгоритмов к алгоритмам Колмогорова.

Следующий класс образуют полиномиальные верхние оценки времени вычисления, т.е. оценки, в которых время ограничено каким-либо полиномом от длины или нормы входа. Три примера множеств, распознаваемых за полиномиально ограниченное время, были уже приведены в ч. I, § 7 при обсуждении класса $\mathcal{P}$. Большинство получаемых для функций из класса $\mathcal{P}$ оценок - это оценки "с точностью до", как правило, с точностью до мультипликативной константы. В этом отношении характерны названия публикаций, относящихся к упомянутым примерам: в этих названиях встречаются выражения "алгоритм с линейным временем", "менее чем кубическое время", "полиномиальный алгоритм".

Оценки "с точностью до" иногда порождают парадоксальную ситуацию. Для какой-нибудь задачи дальнейший прогресс в направлении нахождения более "эффективного" алгоритма может состоять в переходе от алгоритма с верхней оценкой времени работы $c_1 n^{\alpha_1}$ к алгоритму с верхней оценкой времени работы $c_2 n^{\alpha_2}$, где $\alpha_2 < \alpha_1$. Однако при этом c_2 может быть настолько больше c_1, что для всех практически мыслимых аргументов старый алгоритм эффективней нового. Поучительна в этом отношении ситуация с одной из важных верхних оценок - оценкой сложности умножения матриц. (Чтобы не загромождать изложение, мы будем говорить о числе арифметических операций, но картина для вычислений на представительной модели колмогоровского типа аналогична.) Классический алгоритм дает оценку $c_1 n^3$ для матриц n-го порядка. Алгоритм Штрассена - оценку $c_2 n^{2,81}$, см. [Ахо Хоп Уль 74, п. 6.2]. Анализ работы этого алгоритма показывает, что при некоторой организации вычислений его использование дает выгоду по сравнению с классическим алгоритмом, начиная с матриц 14-го порядка, см. [Фиш 74]. В последние годы были предложены алгоритмы, дающие возможность понизить показатель от 2,81 до (приблизительно) 2.5, однако одновременно мультипликативная константа в полученных оценках растет таким образом, что "эффективный" метод оказывается лучше классического только для матриц астрономического порядка (заведомо большего 10^{10}). Конечно, можно надеяться, что новые методы содержат продуктивные математические идеи, которые в дальнейшем смогут привести к получению действительно осмысленных с прикладной точки зрения алгоритмов.

Что касается емкости вычисления, то с прикладной точки зрения наибольший интерес представляют полиномиальные (от длины аргумента) оценки. Однако, если такая оценка не сопровождается полиномиальной же оценкой времени вычисления, то с практической точки зрения и она является сомнительной, ведь полиномиальная оценка емкости автоматически дает (см. ч. I, § 6) лишь экспоненциальную оценку времени. Для многих важных функций их вычисление может быть осуществлено так, что емкость ограничена какой-либо степенью логарифма длины аргумента, а время (при том же вычислении) ограничено полиномом.

По-видимому, верхние оценки сложности, не мажорируемые

полиномами (т.е. экспоненциальные и т.п. оценки) следует рассматривать не как практически значимые, а как (вместе с соответствующими нижними оценками) позволяющие с чисто теоретической точки зрения расклассифицировать решимые алгоритмические проблемы по "степени сложности решения". О такой классификации для алгоритмических проблем теории групп см. [Канно Гат 73].

Нижние оценки

Как мы уже отмечали, ситуация с нижними оценками принципиально отличается от ситуации с верхними. Установить нижнюю оценку - это значит доказать, что никакой алгоритм вычисления на данной модели не имеет сложности вычисления меньше заданной функции φ. Основным методом получения нижних оценок является диагонализация (о других подходах см. [Сли 81, гл. 3, § 2]). Примерами использования метода диагонализации для построения множеств с заданными нижними оценками сложности являются теоремы об иерархии (см. ч. I, § 7). Диагональные конструкции были использованы также при получении нижних оценок для сложности разрешения логических теорий, см. [Фер Рак 79]. В качестве возможных оценок рассматривались функции от длины формулы. Оказалось, что для многих (можно сказать, для большинства) логических теорий имеет место экспоненциальная нижняя оценка сложности (все равно какой, временной или емкостной) разрешения. А для такой теории, как слабая монадическая теория следования второго порядка, дело обстоит "еще хуже". Названная теория есть теория структуры с носителем $\mathbb{N}$, с единственной сигнатурной функцией - прибавлением единицы, но зато с допущением кванторов не только по натуральным числам, но и по конечным множествам натуральных чисел. Хотя для этой теории и существует разрешающий (= распознающий истинность формул) алгоритм, в практическом смысле она оказывается неразрешимой: действительно, как показано в [Мей 75], для любого разрешающего алгоритма его (все равно какая) сложность не мажорируется никакой сверхэкспонентой $2^{\cdot^{\cdot^{\cdot^{2^{n}}}}}$ с фиксированным числом этажей. Аналогичный результат имеет место для элементарной теории свободной группы: хотя вопрос о разрешимости этой теории остается открытым, она "практически неразрешима":

для нее не существует разрешающего алгоритма со сверхэкспоненциальной (с фиксированным числом этажей) верхней оценкой времени или емкости, см. [Сем 80].

В теории сложности все же есть нижние оценки, полученные не диагональным методом. Это, например, квадратичные нижние оценки для задач типа индентификации слов при условии, что в качестве вычислительной модели берется одноленточная машина Тьюринга. Для задачи распознавания симметрии слов такая оценка найдена в [Бар 65]. Другой пример - это умножение чисел при условии, что очередной разряд результата должен выдаваться "достаточно рано", см. [Пат Фиш Мей 74]. Относительно емкости отметим (упоминавшуюся в ч. I, § 7) нижнюю оценку для емкости распознавания множеств, не являющихся распознаваемыми с нулевой емкостью. Интересно было бы получить соответствующую оценку для вычисления функций (а не только для предикатов) и с другой стороны, для емкости порождения (а не только для разрешения) множеств. При этом, конечно, надо соответствующим образом определить эту емкость порождения. С некоторой точки зрения все перечисленные "недиагональные" оценки имеют "негативный" смысл - они показывают, что рассматриваемая вычислительная модель недостаточно универсальна с точки зрения сложности вычисления.

Заметим, наконец, что соображения об оценках "с точностью до", высказанные в связи с верхними оценками, имеют не меньший смысл и в приложении к нижним оценкам.

§ 9. ВЛИЯНИЕ ТЕОРИИ АЛГОРИТМОВ НА АЛГОРИТМИЧЕСКУЮ ПРАКТИКУ

В настоящее время абсолютное большинство явно сформулированных и используемых человеком в его деятельности алгоритмов - это программы для ЭВМ (см. [Кнут 74], [Кнут 74а]). О масштабах "алгоритмической" деятельности человечества можно судить по возникающим в ней организационным проблемам, см. [Бру 75]. Таким образом, программирование - это алгоритмическая практика, теоретическое программирование - это (при широком понимании термина) - вся теория алгоритмов (например, теорему Гёделя о неполноте можно рассматривать как теорему теоре-

тического программирования, ср. [Глу 79]). Общепринято, однако, другое понимание термина "теоретическое программирование" - как области теории алгоритмов, концентрирующейся вокруг взаимоотношения программы как чисто синтаксического, неинтерпретированного объекта и содержания (смысла, значения) программы. Конечно, для теоретического программирования характерен интерес к порожденным практикой темам, которых не касалась классическая теория алгоритмов, таким как параллельное программирование, см. [Кот 74], или структуры данных, см. [Скотт 71]. Тем не менее, "общая часть" теоретического программирования и теории алгоритмов велика, и классическая теория алгоритмов оказала бесспорное влияние на программирование. Это влияние, однако, состояло не в использовании каких-либо теорем, оно носило скорее идейный характер. Попытаемся проследить, как оно происходило, перечислив соответствующие результаты и понятия общей теории алгоритмов.

Общее понятие алгоритма и возможность его формализации. В вычислительной практике важную роль сыграло осознание того, что любая вычислительная машина (если игнорировать физические ресурсы) может вычислять любую вычислимую функцию и никакая машина не может вычислять невычислимую. Также важную, хотя и не всегда однозначно полезную роль играло утверждение о том, что все задачи, решаемые человеком, могут быть решены подходящими алгоритмами, в частности на ЭВМ.

Существование нерешимых алгоритмических проблем в математике и нерешаемость многих естественно возникших проблем. О некоторых задачах стало заранее известно, что искать их полное и точное решение безнадежно и нужно вырабатывать реалистический подход, основанный на отказе от полноты, абсолютной достоверности или чего-то еще. Конечно, разделение задач на решимые и нерешимые имело и отрицательные последствия, возникло искушение рассматривать всякую решимую задачу как практически решаемую, почти решенную, если не сегодня, то при дальнейшем прогрессе вычислительной техники.

Появление различных понятий сложности вычисления и порождения. Возможность строгого абстрактного определения того, что такое сложность вычисления, стимулировало разработку эффективных алгоритмов и дало возможность их объективного сравнения

(см. [Сли 81]). Большое практическое значение имело определение класса NP и доказательство полиномиальной эквивалентности многих "переборных" задач. Это, наряду с экспоненциальными нижними оценками, разрушило иллюзию, о которой говорилось в предыдущем абзаце; выяснилось, что одного только существования алгоритма, решающего ту или иную массовую проблему, не достаточно для практики. Тем самым еще раз подтвердилась важность нестандартных (эвристических, приближенных и т.д.) подходов (см. [Раб 74]). С другой стороны, теоретические алгоритмы, для которых были доказаны "хорошие" полиномиальные верхние оценки сложности, нашли практические приложения.

Неалгоритмическое описание вычислимых функций (μ-рекурсивные функции, исчисление Эрбрана - Гёделя, исчисление λ-конверсии, неподвижные точки вычислимых операторов и т.д.; обзор и классификацию таких описаний см. в [ЕршА 82 а]). Неалгоритмическое (непроцедурное) описание вычислимых функций оказалось важным средством программирования. Начиная с языка лисп, соответствующие конструкции вошли во многие языки программирования, кроме того, неалгоритмическое описание является основой многих формальных определений семантики программ.

Вычислительные и порождающие модели. Основную роль в программировании играют не сами представительные модели, а их (уже непредставительные) модификации и ограничения. Типичные примеры таких ограничений - магазинные автоматы и контекстно-свободные грамматики. Контекстно-свободные грамматики широко использовались для задания синтаксиса языков программирования, начиная с алгола-60; при описании алгола-68 понадобился более общий вид исчислений - грамматики ван Вейнгаардена.

Именно непредставительность, ограниченность моделей и позволяет в какой-то степени использовать их при создании эффективных алгоритмов обработки программ, прежде всего алгоритмов трансляции. Разнообразие различных представительных вычислительных моделей, которое с точки зрения общей теории алгоритмов может показаться излишним, оказывается весьма осмысленным с практической точки зрения: ведь для практики вопрос о степени удобства, с которой тот или иной алгоритм записывается на языке программирования, оказывается жизненно важным. Рост числа языков программирования в конце 60-х - начале 70-х

годов (речь идет о сотнях и даже тысячах языков, см. [Сем Сем 74]) позволял говорить даже о "вавилонской башне" в программировании.

Трактовка программ как объектов вычисления. Фон Нейман был первым, кто ввел это фундаментальное открытие теории алгоритмов в алгоритмическую практику (см. [Ней 63]). Принцип хранимой и модифицируемой программы стал одной из основ системного программирования. Неотъемлемыми частями каждой ЭВМ являются компилятор и другие компоненты операционной системы, ориентированные на модификацию и выполнение программ потребителей. Машина с действующим интерпретатором есть в точности универсальный алгоритм в смысле теории алгоритмов. Дальнейшим развитием этих идей явилась концепция смешанного вычисления (см. [ЕршА 82]), позволяющая с единой точки зрения взглянуть на многие кажущиеся различными способы обработки программ и данных.

Рассмотрение программ как объектов порождения (как и результаты о логических исчислениях) стимулировало развитие формальных систем, предназначенных для доказательства утверждений о программах, см. [Хоор 69], [Нел 79]. Важнейший класс таких утверждений образуют утверждения о так называемой правильности программ, т.е. о том, что рассматриваемая программа "делает, чего надо".

Методы программирования, т.е. методы построения алгоритмов и доказательства их правильности, появившиеся внутри теории алгоритмов. Наиболее показателен в этом отношении пример структурированного программирования. Основные операторы образования структурированных программ (последовательное выполнение, разветвление, повторение) были введены в начале 50-х годов при описании нормальных алгорифмов Маркова (см. [Марк 54, гл. Ш], [Наг 77]). Одновременно были даны нетривиальные примеры индуктивного доказательства правильности программ, построенных с помощью этих операторов (в частности, программы универсального алгоритма, т.е. интерпретатора). В абстрактной, алгебраической форме операторы структурированного программирования были введены в системах алгоритмических алгебр Глушкова; одновременно были рассмотрены содержательные, с точки зре-

ния практического программирования, примеры преобразования программ и доказательства их правильности, см. [Глу 65]. В 70-е годы структурированное программирование стало одним из инструментов практического программиста.

Программирование как вторая грамотность. "... Мы сами живем в мире программ, подчас не сознавая этого" ([ЕршА Зве 79, с. 47]). Убыстряющееся развитие вычислительной техники и программирования будет включать в алгоритмическую практику (которая, в свою очередь, будет составлять все большую часть разумной деятельности человека) все более широкий круг идей теории алгоритмов (и требовать от теории алгоритмов новых открытий).

А.П.Ершову принадлежит открытие той фундаментальной роли, которую играют алгоритмические концепции в процессе обучения и воспитания современного человека, - роли, сравнимой лишь с ролью письменности (отсюда предложенный А.П.Ершовым термин "вторая грамотность", см. [ЕршА Зве 79], [ЕршА 81а]). Возможность обучения учащихся начальной школы программированию на базе абстрактных вычислительных моделей (не только для подготовки их к будущей профессии, но и для развития способности к формальному мышлению) обсуждается в [Усп 79, гл. 1, § 5].

Согласно [ЕршА Зве 79, с. 48], "программирование ... есть способность выразить любой «правильный» процесс средствами, доступными для передачи машине", т.е., прежде всего алгоритмическими средствами. Поэтому "мы естественно приходим к проблеме фундаментализации программирования, выделения в нем некоторых «натуральных» сущностей" ([ЕршА 81а, с. 81]). Система "натуральных сущностей" программирования формируется под сильным влиянием системы основных понятий теории алгоритмов и исчислений (впрочем, и эта последняя система испытывает влияние первой). "Сумма знаний по этим вопросам должна подвергнуться тщательному концептуальному анализу и в объединении с математическими и лингвистическими концепциями стать фундаментальной компонентой общего образования"([ЕршА 81а, с. 18]).

ЛИТЕРАТУРА

Список сокращений

АиЛ – Алгебра и логика / Институт математики Сибирского отделения АН СССР. Новосибирск.

БСЭ – Большая советская энциклопедия. М.: Большая Советская Энциклопедия (изд. 2-е) и Советская Энциклопедия (изд. 3-е).

ВИНИТИ – Всесоюзный институт научной и технической информации.

ВКМЛ-3 – Третья Всесоюзная конференция по математической логике (23 – 27 июня 1974 г.): Тезисы докладов / Институт математики Сибирского отделения АН СССР. Новосибирск, 1974. 237 с.

ДАН – Доклады Академии наук СССР.

ИАН – Известия Академии наук СССР. Серия математическая.

ИКММЛ – Исследования по конструктивной математике и математической логике.

ИЛ – Издательство иностранной литературы.

КС – Кибернетический сборник. Новая серия. М.: Мир.

МИАН – Математический институт им. В.А.Стеклова Академии наук СССР.

ММО – Московское математическое общество.

МЭ – Математическая энциклопедия. М.: Советская Энциклопедия.

НГУ – Новосибирский государственный университет.

НС ЛОМИ – Научные семинары Ленинградского отделения МИАН.

ПК – Проблемы кибернетики. М.: Физматгиз (до вып. 10) и Наука (с вып. 11).

ПКНМ – Проблемы конструктивного направления в математике.

ТПММЛ – Теоретическое применение методов математической логики.

ТТВМС – Труды Третьего Всесоюзного математического съезда.

УМН – Успехи математических наук.

ACM – the Association for Computing Machinery

AMS – the American Mathematical Society

AW – Reading, Massachusetts, etc.: Addison-Wesley Publishing Company

JCSS – Journal of computer and system sciences

JSL – the Journal of symbolic logic

LMS - the London Mathematical Society

LN - Lecture notes

NH - Amsterdam: North Holland Publishing Company

SIAM - the Society for Industrial and Applied Mathematics

Springer - Berlin, New York, etc.: Springer-Verlag

ZmLGM - Zeitschrift für mathematische Logik und Grundlagen der Mathematik

Агафонов В.Н.

[Ага 75] Агафонов В.Н. Сложность алгоритмов и вычислений: Спецкурс для студентов НГУ, часть 2. Новосибирск: Изд-во НГУ, 1975. 146 с.

Адлер А. (Adler A.)

[Адл 69] Adler A. Some recursively unsolvable problems in analysis. - Proceedings of AMS, 1969, v.22, N 2, p. 523-526.

Адян С.И.

[Адян 55] Адян С.И. Алгоритмическая неразрешимость проблем распознавания некоторых свойств групп. - ДАН, 1955, т. 103, № 4, с. 533 - 535.

[Адян 55а] Адян С.И. О проблеме делимости в полугруппах. - ДАН, 1955, т. 103, № 5, с. 747 - 750.

[Адян 56] Адян С.И. Неразрешимость некоторых алгоритмических проблем теории групп. - В кн.: ТТВМС. Т.1. М.: Издательство АН СССР, 1956, с. 179 - 180.

[Адян 57] Адян С.И. Неразрешимость некоторых алгоритмических проблем теории групп. - Труды ММО, М.: Физматгиз, 1957, т.6, с. 231 - 238.

[Адян 57а] Адян С.И.Конечноопределенные группы и алгоритмы. - УМН, 1957, т. 12, вып. 3 (75), с. 248 - 249. (Резюме доклада на заседании ММО 12 февраля 1957 г.)

[Адян 57б] Адян С.И.Конечно-определенные группы и алгоритмы.- ДАН, 1957, т. 117, № 1, с. 9 - 12.

[Адян 57в] Адян С.И. Проблема алгоритма. - Наука и жизнь, 1957, № 8, с. 13 - 14.

[Адян 58] Адян С.И. Об алгоритмических проблемах в эффективно-полных классах групп. - ДАН, 1958, т. 123, № 1, с. 13 - 16.

[Адян 66] Адян С.И. Определяющие соотношения и алгоритмические проблемы для групп и полугрупп. М.: Наука, 1966. 123 с. (Труды МИАН, т. 85).

[Адян 73] Адян С.И. О работах П.С.Новикова и его учеников по алгоритмическим вопросам алгебры. - В кн.: Математическая логика, теория алгоритмов и теория множеств. М.: Наука, 1973 (Труды МИАН, т. 133), с. 23 - 32.

[Адян 77] Адян С.И. Алгоритмическая проблема. - В кн.: МЭ.

Т.1. 1977, с. 214 - 218.

[Адян 82] Адян С.И. Массовая проблема. - В кн.: МЭ. Т. 3. 1982, с. 538.

Адян С.И., Оганесян Г.У

[Адян Ога 78] Адян С.И., Оганесян Г.У. К проблемам равенства и делимости в полугруппах с одним определяющим соотношением.- ИАН, 1978, т. 42, № 2, с. 219 - 225.

Арнольд В.И. (Arnold V.I.)

[Арн 76] Arnold V.I. Dynamic systems and differential equations (B). -In:[Бра 76], p. 59.

Ахо А.В., Хопкрофт Дж.Е., Ульман Дж.Д. (Aho A.V., Hopcroft J.E., Ullman J.D.)

[Ахо Хоп Уль 74] Aho A.V., Hopcroft J.E., Ullman J.D. The design and analysis of computer algorithms. AW, 1974. X + 470 p. (Русский перевод: Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979. 536 с.)

Барздинь Я.М.

[Бар 65] Барздинь Я.М. Сложность распознавания симметрии на машинах Тьюринга. - ПК, 1965, вып. 15, с. 245 - 248.

[Бар 77] Барздинь Я.М. Алгоритмическая теория информации. - В кн.: МЭ. Т. 1, 1977, с. 219 - 222.

Батс Р.Е., Хинтикка Я. (Butts R.E., Hintikka J.)

[Батс Хин 77] Butts R.E., Hintikka J., eds. Logic, foundations of mathematics, and computability theory. (Proceedings of the Fifth international congress of logic, methodology and philosophy of science. Part 1.) Dordrecht, Holland: D.Reidel, 1977. X + 406 p.

Баур В. (Baur W.)

[Бау 74] Baur W. Uber rekursive Strukturen. - Inventiones mathematicae, 1974, v.23, N 2, p. 89-95.

Бахвалов Н.С.

[Бах 82] Бахвалов Н.С. Модель вычислительная. - В кн.: МЭ. Т. 3, 1982, с. 770.

Бейкер А. (Baker A.)

[Бей 68] Baker A. Contributions to the theory of Diophantine equations. I: On the representation of integers by binary forms. - Philosophical transactions of the Royal Society of London, ser. A, 1968, v.263, N 1139, p. 173-191.

Блюм М. (Blum M.)

[Блюм 67] Blum M. A machine-independent theory of the complexity of recursive functions. - Journal of ACM, 1967, v.14, N 2, p. 322-336. (Русский перевод: Блюм М. Машинно-независимая теория сложности рекурсивных функций. - В кн.: [Коз Муч 70], с. 401 - 422.)

[Блюм 67а] Blum M. On the size of machines. - Information and control, 1967, v.11, N 3, p. 257-265. (Русский перевод: Блюм М. Об объеме машин. - В кн.: [Коз Муч 70], с. 423-431.)

Борель Э. (Borel E.)
[Бор I2] Borel É. Le calcul des intégrales définies. - Journal de Mathematiques pures et appliquées. Sér. 6, 1912, v.8, N 2, p. 159-210. (Перепечатано с некоторыми изменениями и под названием "La théorie de la mésure et la théorie de l'intégration" в [Бор I4], p. 217-256.)

[Бор I4] Borel E. Leçons sur la théorie des fonctions. 2-e éd., augmentéc. Paris: Gauthier-Villars, 1914. XI+259p.

Браудер Ф.Е. (Browder F.E.)
[Бра 76] Browder F.E., ed. Mathematical developments arising from the Hilbert problems. Providence: AMS, 1976. (Proceedings of symposia in pure mathematics, v.28.) 628 p.

Брукс Ф.П., младший (Brooks F.P., Jr.)
[Бру 75] Brooks F.P. The mythical man-month. AW, 1975. (Русский перевод: Брукс Ф.П. Как проектируются и создаются программные комплексы. М.: Наука, I979. I5I с.)

Бун В.В., Каннонито Ф.Б., Линдон Р.К. (Boone W.W., Cannonito F.B., Lyndon R.C.)
[Бун Канно Лин 73] Boone W.W., Cannonito F.B., Lyndon R.C., eds. Word problems: Decision problems and the Burnside problem in group theory. NH, 1973, XII+646 p.

Бун В.В., Хакен В., Поэнару В. (Boone W.W., Haken W., Poénaru V.)
[Бун Хак По 68] Boone W.W., Haken W., Poénaru V. On recursively unsolvable problems in topology and their classification. -In: Contributions to mathematical logic. / Schmidt H.A. et al., eds. NH, 1968, p. 37-74.

Вайнберг Ю.Р., Ногина Е.Ю.
[Вай Ног 76] Вайнберг Ю.Р., Ногина Е.Ю. О двух типах непрерывности вычислимых отображений нумерованных топологических пространств. - Исследования по теории алгорифмов и математической логике. / Марков А.А., Кушнер Б.А., ред.: М.: Вычислительный центр АН СССР, I976, т. 2, с. 84 - 99.

Валиев М.К. (Valiev M.K.)
[Вал 79] Valiev M.K. On axiomatization of deterministic propositional dynamic logic. -In: Mathematical foundations of computer science 1979. / Becvar J., ed., Springer, 1979 (LN in computer science, v.74), p. 482-491.

Вейль Г. (Weyl H.)
Вей 2I Weyl H. Über die neue Grundlagenkrise der Mathematik. - Mathematische Zeitschrift, 1921, Bd. 10, H. 1 - 2, S. 39-79. (Русский перевод: О новом кризисе основ математики. - В кн.: Вейль Г. О философии математики. М.; Л.: Гостехтеориздат, I934, с. 92 - I28.)

Вьюгин В.В.
[Вью 73] Вьюгин В.В. О некоторых примерах верхних полурешеток вычислимых нумераций. - АиЛ, 1973, т. 12, вып. 5, с. 512-529.

[Вью 8I] Вьюгин В.В. Алгоритмическая энтропия (сложность) конечных объектов и ее применение к определению случайности и количества информации. - Семиотика и информатика. М.: ВИНИТИ, 1981, вып. I6 (второй выпуск за I980 год), с. I4 - 43.

Вэлиант Л. (Valiant L.G.)
[Вэл 75] Valiant L.G. General context-free recognition in less than cubic time. - JCSS, 1975, v.10, N 2, p. 308-315.

Гарднер М. (Gardner M.)
[Гар 70 - 71] Gardner M. Mathematical games. - Scientific American, 1970, v.223, N 4, p. 120-123; 1971, v.224, N 2, p. 112-117. (Русский перевод: Гарднер М. Математические досуги. М.: Мир, 1972. Гл. 38. Игра "Жизнь", с. 458-488.)

Гёдель К. (Gödel K.)
[Гёд 31] Gödel K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. - Monatshefte für Mathematik und Physik, 1931, Bd. 38, H. 1, S. 173-198. (Английский перевод: On formally undecidable propositions of Principia Mathematica and related systems I. - В кн.: Хей 67 , с. 596 - 616. Другой английский перевод - в кн.: Дей 65 , с. 5 - 38.)

[Гёд 58] Gödel K. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. - Dialectica, 1958, v.12, N 3/4, p. 280-287. (Русский перевод: Гёдель К. Об одном еще не использованном расширении финитной точки зрения. - В кн.: Математическая теория логического вывода: сборник переводов / Идельсон А.В., Минц Г.Е., редакторы. М.: "Наука", 1967, с. 299 - 305.

Гильберт Д. (Hilbert D.)
[Гиль 35] ,Hilbert D. Mathematische Probleme. -In: Hilbert D. Gesammelte Abhandlungen. Bd. 3. Berlin: Springer Verlag, 1935, S. 290-329. (Русский перевод: Гильберт Д. Математические проблемы. - В кн.: Проблемы Гильберта /Александров П.С., ред. М.: Наука, 1969, с. 11 - 64.

Гильберт Д., Аккерман В. (Hilbert D., Ackermann W.)
[Гиль Акк 38] Hilbert D., Ackermann W. Grundzuge der theoretischen Logik. 2-te, verbesserte Aufl. Berlin: Springer, 1938. VIII+133 S.(Перепечатано: New York: Dover Publications, 1946, VIII+155 S. Русский перевод: Гильберт Д., Аккерман В. Основы теоретической логики. М.: ИЛ, 1947. 304 с.

Гладкий А.В.
[Гла 73] Гладкий А.В. Формальные грамматики и языки. М.: Наука, 1973. 368 с.

[Гла 77] Гладкий А.В. Грамматика порождающая. - В кн.: МЭ. Т. 1. 1977, с. 1092 - 1093.

[Гла 77а] Гладкий А.В. Грамматика составляющих. - В кн.: МЭ. Т. 1. 1977, с. 1093 - 1095.

[Гла 82] Гладкий А.В. Математическая лингвистика. - В кн.: МЭ. Т. 3. 1982, с. 565 - 568.

Глушков В.М.
[Глу 64] Глушков В.М. Введение в кибернетику. - Киев: Издательство АН УССР, 1964. 324 с.

[Глу 65] Глушков В.М. Теория автоматов и формальные преобразования микропрограмм. - Кибернетика, 1965, № 5, с. 1 - 9.

[Глу 79] Глушков В.М. Теорема о неполноте формальных теорий

с позиций программиста. - Кибернетика, 1979, № 2, с. 1 - 5.

Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л.
[Глу Цей Ющ 78] Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра, языки, программирование. 2-е изд. Киев: Наукова думка, 1978. 318 с.

Гончаров С.С.
[Гон 75] Гончаров С.С. Некоторые свойства конструктивизаций булевых алгебр. - Сибирский математический журнал, 1975, т. 16, № 2, с. 264 - 278.

[Гон 75а] Гончаров С.С. Автоустойчивость и вычислимые семейства конструктивизаций. - АиЛ, 1975, т. 14, № 6, с. 647-680.

[Гон 76] Гончаров С.С. Неавтоэквивалентные конструктивизации атомных булевых алгебр. - Математические заметки, 1976, т. 19, № 6, с. 853 - 858.

[Гон 79] Гончаров С.С. Конструктивных моделей теория. - В кн.: МЭ. Т. 2. 1979, с. 1058 - 1060.

[Гон 80] Гончаров С.С. Проблема числа неавтоэквивалентных конструктивизаций. - ДАН, 1980, т. 251, № 2, с. 271 - 274.

[Гон 80а] Гончаров С.С. Вычислимые однозначные нумерации. - АиЛ, 1980, т. 19, № 5, с. 507 - 551.

[Гон 80б] Гончаров С.С. Проблема числа неавтоэквивалентных конструктивизаций. - АиЛ, 1980, т. 19, № 6, с. 621 - 639.

[Гон 81] Гончаров С.С. Группы с конечным числом конструктивизаций. - ДАН, 1981, т. 256, № 2, с. 269 - 272.

Григорьев Д.Ю.
[Гри 76] Григорьев Д.Ю. Алгоритмы Колмогорова сильнее машин Тьюринга. - В кн.: ИКММЛ. УП. (Записки НС ЛОМИ, т. 60). Л.: Наука, 1976, с. 29 - 37.

Дейвис М. (Davis M.)
[Дей 53] Davis M. Arithmetical problems and recursively enumerable predicates. - JSL, 1953, v.18, N 1, p. 33-41. (Русский перевод: Дейвис М. Арифметические проблемы и рекурсивно перечислимые предикаты. - Сборник переводов "Математика", 1964, т. 8, № 5, с. 15 - 22.)

[Дей 58] Davis M. Computability and unsolvability. New York et al.: McGraw-Hill Book Company, inc., 1958, 210 p.

[Дей 65] Davis M. (Ed.) The undecidable. Basic papers on undecidable propositions, unsolvable problems and computable functions. Hewlett (New York): Raven Press, 1965, 440 p.

Дейвис М., Матиясевич Ю.В., Робинсон Дж. (Davis M., Matijasevic Yu., Robinson J.)
[Дей Мат Роб 76] Davis M., Matijasevic Yu., Robinson J. Hilbert's tenth problem. Diophantine equations: positive aspects of a negative solution. -In: [Бра 76] p. 323-378.

Дейвис М., Путнам Х., Робинсон Дж. (Davis M., Putnam H., Robinson J.)
[Дей Пут Роб 61] Davis M., Putnam H., Robinson J. The decision problem for exponential Diophantine equations. - Annals of mathematics, 1961, v.74, N 3, p. 425-436.

(Русский перевод: Дэвис М., Путнам Х., Робинсон Дж. Проблема разрешимости для показательно-диофантовых уравнений. - Сборник переводов "Математика", 1964, т. 8, № 5, с. 69 - 79.)

Ден М. (Dehn M.)

[Ден 12] Dehn M. Über unendliche diskontinuierliche Gruppen. - Mathematische Annalen, 1912, Bd. 71, S. 116-144.

Денисов С.Д.

[Дени 78] Денисов С.Д. Строение верхней полурешетки рекурсивно перечислимых -степеней и смежные вопросы. I. - АиЛ, 1978, т. 17, № 6, с. 643-683.

Дегтев А.Н.

[Дег 73] Дегтев А.Н. О tt - и m-степенях. - АиЛ, 1973, т. 12, № 2, с. 143-161.

[Дег 79] Дегтев А.Н. О сводимостях табличного типа в теории алгоритмов. - УМН, 1979, т. 34, вып. 3 (207), с. 137-168.

Дегтев А.Н., Захаров Д.А.

[Дег Зах 79] Дегтев А.Н., Захаров Д.А. Перечислимые множества. Учебное пособие. Новосибирск: Новосибирский государственный университет, 1979.-92 с.

Добрушин Р.Л., Прелов В.В.

[Доб Пре 79] Добрушин Р.Л., Прелов В.В. Информации теория. - В кн.: МЭ. Т. 2. 1979, с. 653 - 655.

Ейтс К.Е.М. (Yates C.E.M.)

[Ейтс 65] Yates C.E.M. Three theorems on the degrees of recursively enumerable sets. - Duke mathematical journal, 1965, v.32, N 3, p. 461-468. (Русский перевод: К.Е.Ейтис. Три теоремы о степенях рекурсивно перечислимых множеств.-В русском переводе книги [Шенф 71], с.97-108.)

Ершов А.П.

[ЕршА 60] Ершов А.П. Операторные алгорифмы. I. (Основные понятия). - ПК, 1960, вып. 3, с. 5 - 48.

[ЕршА 62] Ершов А.П.Операторные алгорифмы. II. (Описание основных конструкций программирования). - ПК,1962,вып.8,с.211-233.

[ЕршА 68] Ершов А.П. Операторные алгоритмы. III. (Об операторных схемах Янова). - ПК, 1968, вып. 20, с. 181 - 200.

[ЕршА 72] Ershov A.P. Theory of program schemata. - В кн.: [Фрей Гриф Розенф 72], т. 1, с. 28 - 45.

[ЕршА 73] Ершов А.П. Современное состояние теории схем программ. - ПК, 1973, вып. 27, с. 87 - 110.

[ЕршА 77] Ершов А.П. Введение в теоретическое программирование (беседы о методе). М.: Наука, 1977. 288 с.

[ЕршА 79] Ершов А.П. Предисловие редактора. - В кн.: [Бру 75], русский перевод, с. 4 - 6.

[ЕршА 80] Ершов А.П. Международный симпозиум "Алгоритм в современной математике и ее приложениях". - Кибернетика, 1980, № 2, с. 145 - 147.

[ЕршА 81] Ершов А.П. Интервью .-Кибернетика,1981,№4(100),с.9-12.

[ЕршА 81а] Ершов А.П. Программирование - вторая грамотность. Новосибирск: Вычислительный центр Сибирского отделения АН СССР, 1981. (Препринт 293) 18 с.

[ЕршА 82] Ershov A.P. Mixed computation: potential applications and problems for study. - Theoretical computer science, 1982, v.18, N 1, p. 41-67.

[ЕршА 82а] Ершов А.П. Вычислимость в произвольных областях и базисах. - Семиотика и информатика. М.: ВИНИТИ, 1982, вып.19, с. 3 - 58.

Ершов А.П., Звенигородский Г.А.
[ЕршА Зве 79] Ершов А., Звенигородский Г. Зачем нужно уметь программировать? - Квант, 1979, № 9, с. 47 - 51.

Ершов А.П., Кнут Д.Е. (Knuth D.E.)
[ЕршА Кнут 81] Ershov A.P., Knuth D.E., eds. Algorithms in modern mathematics and computer science. Springer, 1981 (LN in computer science, v.122). xi+487 p.

Ершов А.П., Ляпунов А.А.
[ЕршА Ляп 67] Ершов А.П., Ляпунов А.А. О формализации понятия программы. - Кибернетика, 1967, № 5, с. 40 - 57.

Ершов А.П., Успенский В.А.
[ЕршА Усп 80] Ершов А.П., Успенский В.А. Алгоритмы на родине аль-Хорезми. - Научно-техническая информация. Серия 2. Информационные процессы и системы, 1980, № 1, с. 28 - 30.

Ершов Ю.Л.
[ЕршЮ 62] Ершов Ю.Л. О гипотезе В.А.Успенского. - АиЛ, 1982, т. 1, вып. 4, с. 45 - 48.

[ЕршЮ 64] Ершов Ю.Л. Разрешимость элементарной теории дистрибутивных структур с относительными дополнениями и теории фильтров. - АиЛ, 1964, т. 3, вып. 3, с. 17 - 38.

[ЕршЮ 64а] Ершов Ю.Л. Неразрешимость теорий симметрических и простых конечных групп. - ДАН, 1964, т. 158, № 4, с. 777-779.

[ЕршЮ 65] Ершов Ю.Л. Неразрешимость некоторых полей. - ДАН, 1965, т. 161, № 1, с. 27 - 29.

[ЕршЮ 66] Ершов Ю.Л. Новые примеры неразрешимых теорий. АиЛ, 1966, т. 5, вып. 5, с. 37 - 47.

[ЕршЮ 72] Ершов Ю.Л. Существование конструктивизаций. - ДАН, 1972, т. 204, № 5, с. 1041 - 1044.

[ЕршЮ 73] Ершов Ю.Л. Конструктивные модели. - В кн.: Избранные вопросы алгебры и логики. Новосибирск: Наука, 1973, с. 111-130.

[ЕршЮ 74] Ершов Ю.Л. Теория нумераций. Часть 3. Конструктивные модели. Новосибирск, 1974. 139.с. (Библиотека кафедры алгебры и математической логики Новосибирского университета, вып. 13.)

[ЕршЮ 74а] Ершов Ю.Л. Нумераций теория. - В кн.: Энциклопедия кибернетики. Т. 2. Киев: Главная редакция Украинской Советской Энциклопедии, 1974, с. 93 - 94.

[ЕршЮ 77] Ершов Ю.Л. Теория нумераций. М.: Наука, 1977. 416 с.

[ЕршЮ 80] Ершов Ю.Л. Проблемы разрешимости и конструктивные модели. М.: Наука, 1980. 415 с.

Ершов Ю.Л., Лавров И.А.
[ЕршЮ Лав 73] Ершов Ю.Л., Лавров И.А. Верхняя полурешетка $L(\delta)$. -

АиЛ, 1973, т. 12, № 2, с. 167 - 189.

Ершов Ю.Л., Лавров И.А., Тайманов А.Д., Тайцлин М.А.
[ЕршЮ Лав Тайм Тайц 65] Ершов Ю.Л., Лавров И.А., Тайманов А.Д., Тайцлин М.А. Элементарные теории. - УМН, 1965, т. 20, вып. 4 (124), с. 37 - 108.

Заславский И.Д., Цейтин Г.С.
[Зас Цей 56] Заславский И.Д., Цейтин Г.С. О соотношении между основными свойствами конструктивных функций. - В кн.: ТТВМС. Т. 1. М.: Изд-во АН СССР, 1956, с. 180 - 181.

[Зас Цей 62] Заславский И.Д., Цейтин Г.С. О сингулярных покрытиях и связанных с ними свойствах конструктивных функций. - В кн.: ПКНМ. 2. М.; Л.: Изд-во АН СССР, 1962 (Труды МИАН, т. 67), с. 458 - 502.

Звонкин А.К., Левин Л.А.
[Зво Лев 70] Звонкин А.К., Левин Л.А. Сложность конечных объектов и обоснование понятий информации и случайности с помощью теории алгоритмов. - УМН, 1970, т. 25, вып. 6 (156), с. 85 - 127.

Земляченко В.Н., Корнеенко Н.М., Тышкевич Р.И.
[Зем Кор Тыш 82] Земляченко В.Н., Корнеенко Н.М., Тышкевич Р.И. Проблема изоморфизма графов. - В кн.: Теория сложности вычислений. 1. (Записки НС ЛОМИ, т. 118.) Л.: Наука, 1982, с. 83 - 158.

Каннан Р., Липтон Р.Дж. (Kannan R., Lipton R.J.)
[Канна Лип 80] Kannan R., Lipton R.J. Orbit problem is decidable. -In: Twelfth annual ACM symposium on the theory of computing (Los Angeles, California, April 28-30, 1980). N.Y.: ACM, 1980, p. 252-268.

Каннонито Ф.Б., Гаттердам Р.В. (Cannonito F.B., Gatterdam R.W.)
[Канно Гат 73] Cannonito F.B., Gatterdam R.W. The computability of group constructions. I. - In: [Бун Канно Лин 73] p. 365-400.

Карп Р.М., Миллер Р.Е. (Karp R.M., Miller R.E.)
[Карп Милл 69] Karp R.M., Miller R.E. Parallel program schemata. - JCSS, 1969, v.3, N 2, p. 147-195. (Русский перевод: Карп Р.М., Миллер Р.Е. Параллельные схемы программ.- КС. 1976, вып. 13, с. 5 - 61.)

Клауа Д. (Klaua D.)
[Кла 61] Klaua D. Konstruktive Analysis. Berlin: VEB Deutscher Verlag der Wissenschaften, 1961. VIII+160 S. (Mathematische Forschungsberichte, Bd. 11.)

Клини С.К. (Kleene S.C.)
[Кли 36] Kleene S.C. General recursive functions of natural numbers. - Mathematische Annalen, 1936, Bd. 112, H. 5, S. 727-742. Перепечатано в [Дей 65], с. 237 - 252. В [Кли 52, библиография] автор пишет: "По поводу ошибки и упрощения см. JSL , т. 3, с. 152; т. 2, с. 38 и т. 4, верх с. IV в конце.")

[Кли 43] Kleene S.C. Recursive predicates and quantifiers. - Transactions of AMS, 1943, v.53, N 1, p. 41-73. (Перепе-

чатано в [Дей 65], с. 255 - 287. В [Кли 52, библиография] автор рекомендует опустить § I5 из [Кли 43].)

[Кли 50] Kleene S.C. A symmetric form of Gödel's theorem. - Koninklijke Nederlandsche Akademie van Wetenschappen, Proceedings of the section of sciences, ser. A, 1950, v.53, N 5-6, p. 800-802. (Продублировано в: Indagationes mathematicae, 1950, v.12, N 3, p. 244-246.

[Кли 52] Kleene S.C. Introduction to metamathematics. N.Y.; Toronto: D. Van Nostrand Company, 1952. 516 p. (Русский перевод: Клини С.К. Введение в метаматематику. М.: ИЛ, I957. 526 с.)

[Кли 52а] Kleene S.C. Recursive functions and intuitionistic mathematics. -In: Proceedings of the International congress of mathematicians. (Cambridge, Massachusetts, USA, August 30 - September 6 1950.) V.1. Providence: AMS, 1952, p. 679-685.

[Кли 60] Kleene S.C. Realizability and Shanin's algorithm for the constructive deciphering of mathematical sentences. - Logique et analyse. Nouvelle série, 1960, t.3, N 11-12, p. 154-165.

[Кли 60а] Kleene S. Mathematical logic: Constructive and non-constructive operations. -In: Proceedings of the International congress of mathematicians, 14-21 August 1958. Cambridge: Cambridge University Press, 1960, p. 137-153. (Русский перевод: Клини С. Математическая логика. Конструктивные и неконструктивные операции. - В кн.: Международный математический конгресс в Эдинбурге I958 г. (обзорные доклады) М.: Физматгиз, I962, с. I58 - I80.)

Клини С.К., Пост Э.Л. (Kleene S.C., Post E.L.)

[Кли Пост 54] Kleene S.C., Post E.L. The upper semi-lattice of degrees of recursive unsolvability. - Annals of mathematics, ser. 2, v.59, N 3, p. 379-407.

Кнут Д.Э. (Knuth D.E.)

[Кнут 68] Knuth D.E. The art of computer programming. V.1. Fundamental algorithms. - AW, 1968. XXI+634 p. (Русский перевод: Кнут Д. Искусство программирования для ЭВМ. Т.I. Основные алгоритмы. М.: Мир, I976, 735 с.)

[Кнут 69] Knuth D.E. The art of computer programming. V.2. Seminumerical algorithms. - AW, 1969. (Русский перевод: Кнут Д. Искусство программирования для ЭВМ. Т. 2.Получисленные алгоритмы. М.: Мир, I977. 724 с.)

[Кнут 74] Knuth D.E. Computer science and its relation to mathematics. - American mathematical monthly, 1974, v.81, N 4, p. 323-343. (Русский перевод: Кнут Д. Информатика и ее связь с математикой. В кн.: Современные проблем математики. М.: Знание, I977 (Новое в жизни, науке, технике. Серия "Математика, кибернетика", I977, № I2), с. 4 - 32.)

[Кнут 74а] Knuth D.E. Computer programming as an art. - Communications of ACM, 1974, v.17, N 12, p. 667-673.

Козмидиади В.А., Мучник А.А.

[Коз Муч 70] Козмидиади В.А., Мучник А.А., ред. Проблемы ма-

тематической логики. Сложность алгоритмов и классы вычислимых функций. Сборник переводов. М.: Мир, 1970. 432 с. (Библиотека "Кибернетического сборника".)

Колмогоров А.Н.

[Колм 32] Kolmogoroff A. Zur Deutung der intuitionistischen Logik. - Mathematische Zeitschrift, 1932, Bd. 35, H. 1, S. 58-65.

[Колм 50] Колмогоров А.Н. Алгоритм. - БСЭ, 2-е изд. Т. 2. 1950, с. 65.

[Колм 53] Колмогоров А.Н. О понятии алгоритма. - УМН, 1953, т. 8, вып. 4 (56), с. 175 - 176.

[Колм 54] Колмогоров А.Н. Предисловие редактора перевода. - В кн.: Петер Р. Рекурсивные функции. Пер. с нем. М.: ИЛ, 1954, с. 3 - 10.

[Колм 63] Kolmogorov A.N. On tables of random numbers. - Sankhya. The Indian journal of statistics. Ser. A, 1963, v.25, part 4, p. 369-376. (Русский перевод: Колмогоров А.Н. О таблицах случайных чисел. - Семиотика и информатика. М.: ВИНИТИ, 1982, вып. 18 (второй выпуск за 1981 г.), с. 3 - 13.)

[Колм 65] Колмогоров А.Н. Три подхода к определению понятия "количество информации". - Проблемы передачи информации, 1965, т. 1, вып. 1, с. 3 - 11.

[Колм 69] Колмогоров А.Н. К логическим основам теории информации и теории вероятностей. - Проблемы передачи информации, 1969, т. 5, вып. 3, с. 3 - 7.

Колмогоров А.Н., Барздинь Я.М.

[Колм Бар 65] Колмогоров А.Н., Барздинь Я.М. О реализации сетей в 3-мерном пространстве. - ПК, 1967, вып. 19, с. 261 - 268.

Колмогоров А.Н., Успенский В.А.

[Колм Усп 58] Колмогоров А.Н., Успенский В.А. К определению алгоритма. - УМН, 1958, т. 13, вып. 4 (82), с. 3 - 28.

Колмогоров А.Н., Фомин С.В.

[Колм Фом 76] Колмогоров А.Н., Фомин С.В. Элементы теории функций и функционального анализа. 4-е изд. М.: Наука, 1976. 543 с.

Корпелевич Г.М.

[Кор 63] Корпелевич Г.М. О соотношении понятий разрешимости и перечислимости для конечных автоматов. - ДАН, 1963, т. 149, № 5, с. 1023 - 1025.

Косовский Н.К.

[Кос 81] Косовский Н.К. Элементы математической логики и ее приложения к теории субрекурсивных алгоритмов. Учебное пособие. Л.: Издательство Ленинградского университета, 1981. 192 с.

Котов В.Е.

[Кот 74] Котов В.Е. Теория параллельного программирования: прикладные аспекты. - Кибернетика, 1974, № 1, с. 1 - 16; № 2, с. 1 - 18.

[Кот 78] Котов В.Е. Введение в теорию схем программ. Новосибирск: Наука, 1978. 257 с.

Коэн П.Дж. (Cohen P.J.)
[Коэ 66] Cohen P.J. Set theory and the continuum hypothesis. New York; Amsterdam: W.A.Benjamin, 1966. 144 p. (Русский перевод: Коэн П.Дж. Теория множеств и континуум-гипотеза. М.: Мир, 1969. 347 с.)

Крайзель Г. (Kreisel G.)
[Кра 53] Kreisel G. Note on arithmetic models for consistent formulae of the predicate calculus. II. - In: Proceedings of the XIth International congress of philosophy (Bruxelles, August 20-26, 1953), v.14, Amsterdam; Louvain, 1953. p. 39-49.

Криницкий Н.А.
[Кри 77] Криницкий Н.А. Алгоритмы вокруг нас. М.: Наука, 1977. 224.

[Кри 77а] Криницкий Н.А. Широкое формальное определение алгоритма. - ПК, 1977, вып. 32, с. 161 - 186.

Кубинец М.В.
[Куб 72] Кубинец М.В. Распознавание самопересечения плоской траектории алгорифмом Колмогорова. - В кн.: ИКММЛ. У. Л.: Наука, 1972. (Записки НС ЛОМИ, т. 32), с. 35 - 44.

Кузнецов А.В., Трахтенброт Б.А.
[Куз Тра 55] Кузнецов А.В., Трахтенброт Б.А. Исследование частично-рекурсивных операторов средствами теории бэровского пространства. - ДАН, 1955, т. 105, № 5, с. 897 - 900.

Кушнер Б.А.
[Куш 73] Кушнер Б.А. Лекции по конструктивному математическому анализу. М.: Наука, 1973. 448 с.

[Куш 79] Кушнер Б.А. Конструктивного подбора принцип. - В кн.: МЭ. Т. 2, 1979, с. 1049 - 1050.

[Куш 79а] Кушнер Б.А. Конструктивный анализ. - В кн.: МЭ. Т. 2. 1979, с. 1054 - 1057.

Лавлэнд Д. (Loveland D.)
[Лавл 66] Loveland D. A new interpretation of the von Mises' concept of random sequence. ZmLGM, 1966, Bd. 12, H. 4, S. 279-294.

[Лавл 66а] Loveland D. The Kleene hierarchy classification of recursively random sequences. - Transactions of AMS, 1966, v.125, N 3, p. 497-510.

Лавров И.А.
[Лавр 77] Lavrov I.A. Computable numberings. -In: [Бат Хин 77], p. 195-206.

[Лавр 82] Лавров И.А. Нумерация. - В кн.: МЭ. Т. 3. 1982, с. 1085 - 1088.

Лакхэм Д., Парк Д.М., Патерсон М.С. (Luckham D.C., Park D.M.R., Paterson M.S.)
[Лак Парк Пат 70] Luckham D.C., Park D.M.R., Paterson M.S. On formalized computer programs. - JCSS, 1970, v.4,

№ 3, p. 220-249. (Русский перевод: Лакхэм Д., Парк Д.М., Патерсон М.С. О формализованных машинных программах. - КС, 1975, вып. 12, с. 78 - 114.)

Левенштейн В.И.

[Левен 74] Левенштейн В.И. Элементы теории кодирования. - В кн.: [Ябл Луп 74], с. 207 - 305.

Левин Л.А.

[Левин 73] Левин Л.А. О понятии случайной последовательности.- ДАН, 1973, т. 212, № 3, с. 548 - 550.

[Левин 76] Левин Л.А. О различных мерах сложности конечных объектов. - ДАН, 1976, т. 227, № 4, с. 804 - 807.

[Левин 77] Левин Л.А. Об одном конкретном способе задания сложностных мер. - ДАН, 1977, т. 234, № 3, с. 536 - 539.

Лупанов О.Б.

[Луп 63] Лупанов О.Б. О сравнении двух типов конечных источников. - ПК, 1963, вып. 9, с. 321 - 326.

Майхилл Дж. (Myhill J.)

[Май 55] Myhill J. Creative sets. - ZmLGM, 1955, Bd. 1, H.2, S. 97-108.

Майхилл Дж., Шепердсон Дж. (Myhill J., Shepherdson J.C.)

[Май Шеп 55] Myhill J., Shepherdson J.C. Effective operations on partial recursive functions. - ZmLGM, 1955, Bd. 1, H. 4, S. 310-317.

Мальцев А.И.

[Маль 60] Мальцев А.И. О неразрешимости элементарных теорий некоторых полей. - Сибирский математический журнал, 1960, т. I, № I, с. 71 - 77. (Перепечатано в [Маль 76], с. 113-119.)

[Маль 61] Мальцев А.И. Конструктивные алгебры, I. - УМН, 1961, т. 16, вып. 3 (99), с. 3 - 60. (Перепечатано в [Маль 76], с. 132 - 185.)

[Маль 62] Мальцев А.И. Аксиоматизируемые классы локально свободных алгебр некоторых типов. - Сибирский математический журнал, 1962, т. 3, № 5, с. 729 - 743. (Перепечатано в [Маль 76], с. 216 - 229.)

[Маль 62а] Мальцев А.И. О рекурсивных абелевых группах. - ДАН, 1962, т. 146, № 5, с. 1009 - 1012. (Перепечатано в [Маль 76], с. 235 - 238.)

[Маль 63] Мальцев А.И. Полно нумерованные множества. - АиЛ, 1963, т. 2, вып. 2, с. 4 - 30. (Перепечатано в [Маль 76], с. 275 - 293.)

[Маль 65] Мальцев А.И. Алгоритмы и рекурсивные функции. М.: Наука, 1965. 392 с.

[Маль 66] Мальцев А.И. О стандартных обозначениях и терминологии в теории алгебраических систем. - АиЛ, 1966, т. 5, вып. 1, с. 71 - 77.

[Маль 70] Мальцев А.И. Алгебраические системы. М.: Наука, 1970. 392 с.

[Маль 76] Мальцев А.И. Избранные труды. Т. 2. Математическая

логика и общая теория алгебраических систем. М.: Наука, 1976. 388 с.

Манин Ю.И.

[Манин 73] Манин Ю.И. Десятая проблема Гильберта. - Современные проблемы математики. М.: ВИНИТИ, 1973, т. 1 (Итоги науки и техники), с. 5 - 37.

[Манин 80] Манин Ю.И. Вычислимое и невычислимое. М.: Советское радио, 1980. 128 с.

[Манин 81] Manin Yu.I. Expanding constructive universes. -In: [ЕршА Кнут 81] p. 255-260.

Манна З. (Manna Z.)

[Манна 74] Manna Z. Mathematical theory of computation. N.Y.: McGraw-Hill, 1974, 448 p. (Русский перевод гл. 5 из этой книги: Манна З. Теория неподвижной точки программ. - КС, 1978, вып. 15, с. 38 - 100.)

Марков А.А.

[Марк 47] Марков А.А. Невозможность некоторых алгорифмов в теории ассоциативных систем. ДАН, 1947, т. 55, № 7, с. 587 - 590.

[Марк 47а] Марков А.А. Невозможность некоторых алгорифмов в теории ассоциативных систем. II. - ДАН, 1947, т. 58, № 3, с. 353 - 356.

[Марк 51] Марков А.А. Теория алгорифмов. - Труды МИАН, 1951, т. 38, с. 176 - 189.

[Марк 52] Марков А.А. О неразрешимых алгорифмических проблемах. - Математический сборник, 1952, т. 31 (73), № 1, с. 34 - 42.

[Марк 54] Марков А.А. Теория алгорифмов. М.; Л.: Изд-во АН СССР, 1954. 375 с. (Труды МИАН, т. 42).

[Марк 54а] Марков А.А. О непрерывности конструктивных функций. - УМН, 1954, т. 9, вып. 3 (61), с. 226 - 230.

[Марк 56] Марков А.А. Об одном принципе конструктивной математической логики. - В кн.: ТТВМС. Т. 2. М.: Изд-во АН СССР, 1956, с. 146 - 147.

[Марк 58] Марков А.А. К проблеме представимости матриц. - ZmLGM, 1958, Bd. 4, H. 2, S. 157-168.

[Марк 58а] Марков А.А. Неразрешимость проблемы гомеоморфии.- ДАН, 1958, т. 121, № 2, с. 218 - 220.

[Марк 58б] Марков А.А. О неразрешимости некоторых проблем топологии. - ДАН, 1958, т. 123, № 6, с. 978 - 980.

[Марк 58в] Марков А.А. Неразрешимость проблемы гомеоморфии. - УМН, 1958, т. 13, вып. 4 (82), с. 213 - 216.

[Марк 58г] Марков А.А. О конструктивных функциях. - В кн.: ПКНМ. 1. М.; Л.: Изд-во АН СССР, 1958 (Труды МИАН, т. 52), с. 315 - 348.

[Марк 62] Марков А.А. О вычислимых инвариантах. - ДАН, 1962, т. 146, № 5, с. 1017 - 1020.

[Марк 62а] Марков А.А. О конструктивной математике. - В кн.: ПКНМ. 2. М.; Л.: Изд-во АН СССР, 1962 (Труды МИАН, т. 67), с. 8 - 14.

[Марк 64] Марков А.А. О нормальных алгорифмах, вычисляющих булевы функции. - ДАН, 1964, т. 157, № 2, с. 262 - 264.

[Марк 67] Марков А.А. О нормальных алгорифмах, связанных с вычислением булевых функций. - ИАН, 1967, т. 31, № 1, с. 161 - 208.

Мартин-Лёф П. (Martin-Löf P.)

[Март 66] Мартин-Лёф П. О понятии случайной последовательности. - Теория вероятностей и ее применения, 1966, т. 11, № 1, с. 198 - 200.

[Март 66а] Martin-Löf P. The definition of random sequences. - Information and control, 1966, v.9, N 6, p. 602-619.

[Март 68] Martin-Löf P. On the notion of randomness. -In: Intuitionism and proof theory / Kino A. et al., eds. N.Y., 1968, p. 73-78. (Русский перевод: Мартин-Лёф П. О понятии случайности. - В кн.: Сложность вычислений и алгоритмов / Козмидиади В.А., Маслов А.Н., Петри Н.В., ред. М.: Мир, 1974, с. 364 - 369.)

[Март 70] Martin-Löf P. Notes on constructive mathematics. Stockholm: Almqvist, Wiksell, 1970, 109 p. (Русский перевод: Мартин-Лёф П. Очерки по конструктивной математике. М.: Мир, 1975. 136 с.)

Марченков С.С.

[Марч 72] Марченков С.С. О вычислимых нумерациях семейств общерекурсивных функций. - АиЛ, 1972, т. 11, № 5, с. 588-607.

[Марч 76] Марченков С.С. Об одном классе неполных множеств. - Математические заметки, 1976, т. 20, № 4, с. 473 - 478.

Марченков С.С., Матросов В.Л.

[Марч Матр 79] Марченков С.С., Матросов В.Л. Сложность алгоритмов и вычислений. - Теория вероятностей. Математическая статистика. Теоретическая кибернетика. М.: ВИНИТИ, 1979, т. 16 (Итоги науки и техники), с. 103 - 149.

Маслов С.Ю.

[Мас 64] Маслов С.Ю. Некоторые свойства аппарата канонических исчислений Э.Л.Поста. - В кн.: ПКНМ. 3. М.; Л.: Наука, 1964 (Труды МИАН, т. 72), с. 5 - 56.

[Мас 67] Маслов С.Ю. Понятие строгой представимости в общей теории исчислений. - В кн.: ПКНМ. 4. Л.: Наука, 1967 (Труды МИАН, т. 93), с. 3 - 42.

[Мас 78] Maslov S.Yu. Macroevolution as deduction process. - Synthese, 1978, v.39, p. 417-434.

[Мас 79] Маслов С.Ю. Исчисление. - В кн.: МЭ. Т. 2, 1979, с. 685 - 686.

[Мас 79а] Маслов С.Ю. Теория поиска вывода и вопросы психологии творчества. - Семиотика и информатика. М.: ВИНИТИ, 1979, вып. 13, с. 17 - 46.

Матиясевич Ю.В.
[Мат 67] Матиясевич Ю.В. Простые примеры неразрешимых ассоциативных исчислений. - ДАН, 1967, т. 173, № 6, с. 1264-1266.

[Мат 70] Матиясевич Ю.В. Диофантовость перечислимых множеств.- ДАН, 1970, т. 191, № 2, с. 279 - 282.

[Мат 71] Матиясевич Ю.В. Диофантово представление перечислимых предикатов. - ИАН, 1971, т. 35, № 1, с. 3 - 30.

[Мат 72] Матиясевич Ю.В. Диофантовы множества. - УМН, 1972, т. 27, вып. 5 (167), с. 185 - 222.

[Мат 73] Matijasevič Yu.V. On recursive unsolvability of Hilbert's tenth problem. - В кн.: [Сапп 73] p. 89-110.

[Мат 74] Матиясевич Ю.В. Эффективные и неэффективные методы в теории чисел. - В кн.: ВКМЛ-3, с. 141 - 142.

[Мат 74а] Матиясевич Ю.В. Существование неэффективизируемых оценок в теории экспоненциально диофантовых уравнений. - В кн.: ИКММЛ. VI. Л.: Наука, 1974 (Записки НС ЛОМИ, т. 40), с. 77 - 93.

[Мат 77] Matijasevič Yu.V. Some purely mathematical results inspired by mathematical logic. -In: [Бат Хин 77] p. 121-127.

[Мат 77а] Матиясевич Ю.В. Простые числа перечисляются полиномом от 10 переменных. - В кн.: ТПММЛ. II. Л.: Наука, 1977 (Записки НС ЛОМИ, т. 68), с. 62 - 82.

[Мат 79] Матиясевич Ю.В. Диофантов предикат. - В кн.: МЭ. Т. 2. 1979, с. 157.

[Мат 79а] Матиясевич Ю.В. Диофантово множество. - В кн.: МЭ. Т. 2, 1979, с. 161 - 162.

[Мат 79б] Матиясевич Ю.В. Диофантовых уравнений проблема разрешимости. - В кн.: МЭ. Т.2. 1979, с. 174 - 175.

Мачти М., Винкльман К., Янг П. (Machtey M., Winklmann K., Young P.)
[Мач Вин Янг 78] Machtey M., Winklmann K., Young P. Simple Gödel numberings, isomorphisms and programming properties. - SIAM Journal on computing, 1978, N 1, p. 39-60.

Медведев Ф.А.
[МедФ 76] Медведев Ф.А. Французская школа теории функций и множеств на рубеже XIX - XX вв. М.: Наука, 1976. 231 с.

Медведев Ю.Т.
[МедЮ 55] Медведев Ю.Т. Степени трудности массовых проблем. - ДАН, 1955, т. 104, № 4, с. 501 - 504.

[МедЮ 56] Медведев Ю.Т. О понятии массовой проблемы. - УМН, 1956, т. 11, вып. 5 (71), с. 231 - 232.

[МедЮ 62] Медведев Ю.Т. Финитные задачи. - ДАН, 1962, т. 142, № 5, с. 1015 - 1018.

[МедЮ 69] Медведев Ю.Т. Об одном способе доказательства неразрешимости алгоритмических проблем. - ДАН, 1969, т. 185, № 6, с. 1232 - 1235.

Мейер А.Р. (Meyer A.R.)
[Мей 75] Meyer A.R. Weak monadic second order theory of successor is not elementary-recursive. -In: Logic colloquium (Boston, 1972-1973) / Parikh R., ed. Springer, 1975 (LN in mathematics, v.453), p. 132-153.

Мизес Р., фон (von Mises R.)
[Миз 19] von Mises R. Grundlagen der Wahrscheinlichkeitsrechnung. - Mathematische Zeitschrift, 1919, Bd. 5, S. 52-99.

[Миз 28] von Mises R. Wahrscheinlichkeitsrechnung, Statistik und Wahrheit. Wien: J.Springer, 1928. (Русский перевод: Мизес Р. Вероятность и статистика. М.;Л; Гос. из-во, 1930.-250с.

Миллар Т.С. (Millar T.S.)
[Мил 79] Millar T.S. A complete, decidable theory with two decidable models. - JSL, 1979, v.44, N 3, p. 307-312.

Минский М.Л. (Minsky M.L.)
[Мин 67] Minsky M.L. Computation: finite and infinite machines. Englewood Cliffs, N.J.: Prentice-Hall, 1967, 317 p. (Русский перевод: Минский М. Вычисления и автоматы. М.: Мир, 1971. 364 с.)

Московакис Я.Н. (Moschovakis Y.N.)
[Моск 64] Moschovakis Y.N. Recursive metric spaces. - Fundamenta mathematicae, 1964, v.55, N 3, p. 215-238.

Мостовский А. (Mostowski A.)
[Мост 53] Mostowski A. On a system of axioms which has no recursively enumerable model. - Fundamenta mathematicae, 1953, v.40, N 1, p. 56-61.

[Мост 55] Mostowski A. A formula with no recursively enumerable model. - Fundamenta mathematicae, 1955, v.42, N 1, p. 125-140.

[Мост 66] Mostowski A. Thirty years of foundational studies. Oxford: Basil Blackwell, 1966, 180 p. (Acta philosophica fennica, fasc. 17).

Мучник А.А.
[Муч 56] Мучник А.А. Неразрешимость проблемы сводимости теории алгоритмов. - ДАН, 1956, т. 108, № 2, с. 194 - 197. (Исправление формулировки теоремы 4 указано в [Муч 65], с. 717.)

[Муч 58] Мучник А.А. Решение проблемы сводимости Поста и некоторых других проблем теории алгоритмов. I. - Труды ММО. М.: Физматгиз, 1958, т. 7, с. 391 - 405.

[Муч 63] Мучник А.А. О сильной и слабой сводимости алгоритмических проблем. - Сибирский математический журнал, 1963, т. 4, № 6, с. 1328 - 1341.

[Муч 65] Мучник А.А. О сводимости проблем разрешения перечислимых множеств к проблемам отделимости. - ИАН, 1965, т. 29, вып. 3, с. 717 - 724.

[Муч 70] Мучник А.А. О двух подходах к классификации рекурсивных функций. - В кн.: [Коз Муч 70], с. 123 - 138.

Нагорный Н.М.
[Наг 77] Нагорный Н.М. Алгоритмов сочетание. - В кн.: МЭ. Т. I. I977, с. 225 - 226.

[Наг 77а] Нагорный Н.М. Ассоциативное исчисление. - В кн.: МЭ. Т. 1. I977, с. 338 - 340.

[Наг 77б] Нагорный Н.М. Алгоритма изображение. - В кн.: МЭ. Т. 1. I977, с. 2I0.

[Наг 77в] Нагорный Н.М. Групповое исчисление. - В кн.: МЭ. Т. 1. I977, с. II47 - II49.

[Наг 77г] Нагорный Н.М. Абстракция актуальной бесконечности.- В кн.: МЭ. Т. I. I977, с. 43.

[Наг 79] Нагорный Н.М..Конструктивный объект. - В кн.: МЭ. Т. 2. I979, с. I057 - I058.

Нейман Дж., фон (von Neumann J.)
[Ней 63] von Neumann J. The computer and the brain. New Haven: Yale University Press, 1963, XIII+82 p. (Русский перевод: Нейман Дж. Вычислительная машина и мозг. - В кн.: Кибернетический сборник: сборник переводов / Ляпунов А.А., Лупанов О.Б., редакторы. № I. М.: ИЛ, I960, с. II - 60.)

Непомнящий В.А.
[Неп 72] Nepomnjascii V.A. Conditions for the algorithmic completeness of system of operations. - В кн.: [Фрей Гриф Розенф 72]. Т. I, с. 52 - 55.

[Неп 72а] Непомнящий В.А. Критерий алгоритмической полноты систем операций. - В кн.: Теория программирования: Труды симпозиума (Новосибирск, 7 - II августа I972 г.) Ч.I / Непомнящий В.А., ред. 279 с. Новосибирск: Вычислительный центр Сибирского отделения АН СССР, I972, с. 267 - 279.

[Неп 74] Непомнящий В.А. О емкостной сложности распознавания рудиментарных предикатов и формальных языков. - В кн.: ВКМЛ-3, с. I53 - I55.

[Неп 79] Непомнящий В.А. Практические методы проверки правильности программ. - Семиотика и информатика. М.: ВИНИТИ, I979, вып. I2, с. 86 - 87.

Новиков П.С.
[Нов 52] Новиков П.С. Об алгоритмической неразрешимости проблемы тождества. - ДАН, I952, т. 85, № 4, с. 709 - 7I2. (Перепечатано в [Нов 79], с. 205 - 209.)

[Нов 55] Новиков П.С. Об алгоритмической неразрешимости проблемы тождества слов в теории групп. М.: Изд-во АН СССР, I955. I43 с. (Труды МИАН, т. 44). (Перепечатано в [Нов 79], с. 2I0 - 323.)

[Нов 58] Novikow P.S. Über einige algorithmische Probleme der Gruppentheorie. - Jahresbericht der Deutschen Mathematiker Vereinigung, 1958, Bd. 61, H. 2, S. 88-92. (Русский перевод: Новиков П.С. О некоторых алгоритмических проблемах теории групп. - В кн.: [Нов 79], с. 390 - 392.)

[Нов 77] Новиков П.С. Конструктивная математическая логика с точки зрения классической. М.: Наука, I977. 328 с.

[Нов 79] Новиков П.С. Избранные труды. Теория множеств и функций. Математическая логика и алгебра. М.: Наука, 1979. 396 с.

Ногина Е.Ю.

[Ног 66] Ногина Е.Ю. Об эффективно топологических пространствах. - ДАН, 1966, т. 169, № 1, с. 28 - 31.

[Ног 69] Ногина Е.Ю. Соотношения между некоторыми классами эффективно топологических пространств. - Математические заметки, 1969, т. 5, № 4, с. 483 - 495.

[Ног 78] Ногина Е.Ю. Нумерованные топологические пространства. - ZmLGM, 1978, Bd. 24, H. 2, S. 141-176.

Патерсон М.С. (Paterson M.S.)

[Пат 70] Paterson M.S. Unsolvability in 3 x 3 matrices. - Studies in applied mathematics, 1970, v.49, N 1, p. 105-107.

Патерсон М.С., Фишер М.Дж., Мейер А.Р. (Paterson M.S., Fisher M.J., Meyer A.R.)

[Пат Фиш Мей 74] Paterson M.S., Fisher M.J., Meyer A.R. An improved overlap argument for online multiplication. - In: Complexity of computation. Providence: AMS, 1974 (SIAM - AMS proceedings, v.7), p. 97-111. (Русский перевод: Патерсон М.С., Фишер М.Дж., Мейер А.Р. Улучшенный метод частичного перекрытия для умножения, выполняемого в темпе поступления информации. - КС, 1977, вып. 14, с. 77 - 94.)

Пауль В.Дж., Сейферас Дж.И., Симон Дж. (Paul W.J., Seiferas J.I., Simon J.)

[Пау Сей Сим 80] Paul W.J., Seiferas J.I., Simon J. An information-theoretic approach to time bounds for on-line computation (preliminary version). - In: Conference proceedings of the Twelfth annual ACM symposium on theory of computing. Papers presented at the symposium held in Los Angeles, Calif., April 18-30, 1980. New York: ACM, 1980, p. 357-367.

Перетяткин М.Г.

[Пер 73] Перетяткин М.Г. О полных теориях с конечным числом счетных моделей. - АиЛ, 1973, т. 12, № 5, с. 550 - 576.

Петров Б.Н., Уланов Г.М., Ульянов С.В.

[Пет Ула Уль 79] Петров Б.Н., Уланов Г.М., Ульянов С.В. Сложность конечных объектов и информационная теория управления.- Техническая кибернетика. М.: ВИНИТИ, 1979, т. 11 (Итоги науки и техники), с. 77 - 147.

Плиско В.Е.

[Пли 73] Плиско В.Е. О реализуемых предикатных формулах. - ДАН, 1973, т. 212, № 3, с. 553 - 556.

[Пли 76] Плиско В.Е. Некоторые варианты понятия реализуемости для предикатных формул. - ДАН, 1976, т. 226, № 1, с. 61 - 64.

[Пли 77] Плиско В.Е. Неарифметичность класса реализуемых предикатных формул. - ИАН, 1977, т. 41, № 3, с. 483 - 502.

[Пли 78] Плиско В.Е. Некоторые варианты понятия реализуемости для предикатных формул. - ИАН, 1978, т. 42, № 3, с. 636-653.

Поляков Е.А., Розинас М.Г.
[Пол Роз 76] Поляков Е.А., Розинас М.Г. Теория алгоритмов. Учебное пособие по спецкурсу для студентов-математиков. Иваново: Ивановский государственный университет, 1976, 88 с.

Пост Э.Л. (Post E.L.)
[Пост 36] Post E.L. Finite combinatory processes - formulation 1. - JSL, 1936, v.1, N 3, p. 103-105. (Перепечатано в [Дей 65], с. 289 - 291.) (Русский перевод: Пост Э.Л. Финитные комбинаторные процессы, формулировка I. - В кн.: [Усп 79], с. 89 - 95.)

[Пост 43] Post E.L. Formal reductions of the general combinatorial decision problem. - American journal of mathematics, 1943, v.65, N 2, p. 197-215.

[Пост 44] Post E.L. Recursively enumerable sets of positive integers and their decision problems. - Bulletin of AMS, 1944, v.50, N 5, p. 284-316. (Перепечатано в [Дей 65], с. 305 - 337.)

[Пост 46] Post E.L. A variant of a recursively unsolvable problem. - Bulletin of AMS, 1946, v.52, N 4, p. 264-268.

[Пост 47] Post E.L. Recursive unsolvability of a problem of Thue. - JSL, 1947, v.12, N 1, p. 1-11. (Перепечатано в [Дей 65], с. 297 - 303.)

Прохоров Ю.В.
[Про 73] Прохоров Ю.В. Кодирование. - В кн.: БСЭ, 3-е изд. Т. 12. 1973, с. 373 - 374.

Пур-Эл М. (Pour-El M.)
[Пур 64] Pour-El M. Gödel numberings versus Friedberg numberings. - Proceedings of AMS, 1964, v.15, N 2, p. 252-256.

Рабин М.О. (Rabin M.O.)
[Раб 58] Rabin M.O. On recursively enumerable ans arithmetic models of set theory. - JSL, 1958, v.23, N 4, p. 408-416.

[Раб 60] Rabin M.O. Computable algebra, general theory and theory of computable fields. - Transactions of AMS, 1960, v.95, N 2, p. 341-360.

[Раб 63] Rabin M. Real time computation. - Israel journal of mathematics, 1963, v.1, N 4, p. 203-211. (Русский перевод; Рабин М. Вычисления в реальное время. - В кн.: [Коз Муч 70], с. 156 - 167.)

[Раб 69] Rabin M.O. Decidability of second-order theories and automata on infinite trees. - Transactions of AMS, 1969, v.141, N 7, p. 1-35. (Русский перевод: Рабин М.О. Разрешимость второго порядка и автоматы над бесконечными деревьями. КС, 1969, вып. 8, с. 72 - 116.)

[Раб 74] Rabin M.O. Theoretical impediments to arithmetical intelligence. -In: Information processing 74. Proceedings of IFIP congress 1974 (Stockholm, August 3-10, 1974)./ Rosenfeld J.L., ed. NH, 1974, p. 615-619.

Райс Х. (Rice H.G.)
[Райс 53] Rice H.G. Classes of recursively enumerable sets

and their decision problems. - Transactions of AMS, 1953, v.74, N 2, p. 358-366.

[Райс 54] Rice H.G. Recursive real numbers. - Proceedings of AMS, 1954, v.5, N 5, p. 784-791.

Ричардсон Д. (Richardson D.)
[Рич 68] Richardson D. Some undecidable problems involving elementary functions of a real variable. - JSL, 1968, v.33, N 4, p. 514-520.

Робинсон Дж. (Robinson J.)
[Роб 49] Robinson J. Definability and decision problems in arithmetic. - JSL, 1949, v.14, N 2, p. 98-114.

[Роб 52] Robinson J. Existential definability in arithmetic. - Transactions of AMS, 1952, v.72, N 3, p. 437-449. (Русский перевод: Робинсон Дж. Экзистенциальная выразимость в арифметике. - Сборник переходов "Математика", 1964, т. 8, № 5, с. 3 - 14.)

Роджерс Х., младший (Rogers H., Jr.)
[Родж 58] Rogers H., Jr. Gödel numberings of partial recursive functions. - JSL, 1958, v.23, N 3, p. 331-341.

[Родж 67] Rogers H., Jr. Theory of recursive functions and effective computability. New York et al.: McGraw-Hill Book Company, 1967, XIX+482 p. (Русский перевод: Роджерс Х. Теория рекурсивных функций и эффективная вычислимость. М.: Мир, 1972. 624 с.)

Розенберг А.Л. (Rosenberg A.L.)
[Роз 67] Rosenberg A.L. Real-time definable languages. - Journal of ACM, 1967, v.14, N 4, p. 645-662. (Русский перевод: Розенберг А. Языки, определимые в реальное время. - В кн.: [Коз Муч 70], с. 168 - 193.)

Рот К.Ф. (Roth K.F.)
[Рот 55] Roth K.F. Rational approximations to algebraic numbers. - Mathematica, 1955, v.2, N 1, p. 1-20 (corrigendum p. 168) (Русский перевод: Рот К.Ф. Рациональные приближения алгебраических чисел. - Сборник переводов "Математика", 1957, т. 1, № 1, с. 3 - 18.)

Сакс Дж.Е. (Sacks G.E.)
[Сакс 63] Sacks G.E. Degrees of unsolvability. Princeton, New Jersey: Princeton University Press, 1963. 174 p. (Annals of mathematics studies, number 55.)

Саломаа А., Соиттола М. (Salomaa A., Soittola M.)
[Сал Сои 78] Salomaa A., Soittola M. Automata-theoretic aspects of formal power series. Springer, 1978, X+171 p.

Санкаппанавар Х.П. (Sankappanavar H.P.)
[Сан 78] Sankappanavar H.P. Decision problems: History and methods. -In: Mathematical logic. Proceedings of the first Brasilian conference. / Arruda A.I., Newton da Costa C.A., Chuaqui R., eds. New York and Basel: Marcel Dekker, 1978 (Lecture notes in pure and applied mathematics, v. 39), p. 241-291.

Саппес П. (Suppes P.)
[Сапп 73] Suppes P. et al., eds. Logic, Methodology and Philosophy of Science. IV. NH, 1973.

Сейферас Дж.И. (Seiferas J.I.)
[Сей 77] Seiferas J.I. Relating refined space complexity classes. - JCSS, 1977, v.14, N 1, p. 100-129.

Сейферас Дж.И., Фишер М.Дж., Мейер А.Р. (Seiferas J.I., Fisher M.J., Meyer A.R.)
[Сей Фиш Мей 78] Seiferas J.I., Fisher M.J., Meyer A.R. Separating nondeterministic time complexity classes. - Journal of ACM, 1978, v.25, N 1, p. 146-167.

Селиванов В.Л.
[Сел 76] Селиванов В.Л. Две теоремы о вычислимых нумерациях.- АиЛ, 1976, т.15, № 4, с. 470-484.

Семёнов А.Л.
[Сем 78] Семёнов А.Л. Некоторые алгоритмические проблемы для систем алгоритмических алгебр. - ДАН, 1978, т. 239, № 5, с. 1063-1066.

[Сем 80] Семёнов А.Л. Интерпретация свободных алгебр в свободных группах. - ДАН, 1980, т.252, № 6, с. 1326-1332.

Семёнов А.Л., Семёнова Е.Т.
[Сем Сем 74] Семёнов А.Л., Семёнова Е.Т. Программирование и математическое обеспечение. - В кн.: Радиоэлектроника в 1973 г. Обзор по материалам иностранной печати. Вып. 6. Вычислительная техника. Программирование. М.: Научно-исследовательский институт экономики и информации по радиоэлектронике, 1974, с. 76-91.

Семёнов А.Л., Успенский В.А. (Semenov A.L., Uspensky V.A.)
[Сем Усп 80] Семенов А.Л., Успенский В.А. Международная встреча ученых в Хорезме. - Международный форум по информации и документации, 1980, т.5, № 1, с. 36-37. (Английский перевод: Semenov A.L., Uspensky V.A. International meeting of scientists at Khoresm. - International forum on information and documentation, 1980, v.5, N 1, p. 37-38.)

Скотт Д. (Scott D.)
[Ско 61] Scott D. On constructing models for arithmetic. -In: Infinitistic methods. Proceedings of the Symposium on foundations of mathematics (Warsaw, 2-9 Sept. 1959). Warszawa et al., 1961, p. 235-255.

[Ско 70] Scott D. Outline of a mathematical theory of a computation. - In: Proceedings of the Fourth annual Princeton conference on information sciences and systems, 1970. Princeton (New Jersey): Princeton University Press, 1970, p. 169-176. (Русский перевод: Скотт Д. Набросок математической теории вычислений. - КС, 1977, вып. 14, с. 105-121.)

Слисенко А.О.
[Сли 77] Слисенко А.О. Упрощенное доказательство распознаваемости симметричности слов в реальное время на машинах Тьюринга. - В кн.: ТПММЛ. П. Л.: Наука, 1977 (Записки НС ЛОМИ, т.68), с. 123-139.

[Сли 77а] Слисенко А.О. Распознавание предиката вхождения в реальное время. Л. 1977. 24 с. (Препринт / Ленинградское отделение МИАН: Р 7 - 77)

[Сли 78] Slisenko A.O. String-matching in real time: some properties of the data structure. -In: Mathematical foundations of computer science 1978. / Winkowski J., ed. Springer, 1978 (LN in computer science, v.64), p. 493-496.

[Сли 81] Слисенко А.О. Сложностные задачи теории вычислений.- УМН, 1981, т. 36, вып. 6, с. 21 - 103.

Соар Р. (Soare R.I.)

[Соа 78] Soare R.I. Recursively enumerable sets and degrees. - Bulletin of AMS, 1978, v.84, N 6, p. 1149-1181.

Соломонов Р.Дж. (Solomonoff R.J.)

[Сол 64] Solomonoff R.J. A formal theory of inductive inference I. - Information and control, 1964, v.7, N 1, p. 1-22.

Стоцкий Э.Д.

[Сто 80] Стоцкий Э.Д. Элементы теории формальных грамматик. Препринт. М.: ВИНИТИ, 1980. 67 с.

Сэвич В.Дж. (Savitch W.J.)

[Сэв 70] Savitch W.J. Relationships between nondeterministic and deterministic tape complexities. - JCSS, 1970, v.4, N 2, p. 177-192.

Сэсердоут С., Тенни Р.Л. (Sacerdote G.S., Tenney R.L.)

[Сэс Тен 77] Sacerdote G.S., Tenney R.L. The decidability of the reachability problem for vector addition systems. -In: Conference record of the Ninth annual ACM symposium on theory of computing. Papers presented at the symposium held in Boulder, Colo., May 2-4, 1977. N.Y.: ACM, 1977, p. 61-76.

Тарский А., Мостовский А., Робинсон Р.М. (Tarski A., Mostowski A., Robinson R.M.)

[Тар Мост Роб 53] Tarski A., Mostowski A., Robinson R.M. Undecidable theories. NH, 1953, XI+98 p.

Тверской А.А.

[Тве 82] Тверской А.А. Исследование рекурсивности и арифметичности сигнатурных функций в нестандартных моделях арифметики. - ДАН, 1982, т. 262, № 6, с. 1325 - 1328.

Трахтенброт Б.А.

[Тра 53] Трахтенброт Б.А. О рекурсивной отделимости. - ДАН, 1953, т. 88, № 6, с. 953 - 956.

[Тра 56] Трахтенброт Б.А. Сигнализирующие функции и табличные операторы. - Ученые записки Пензенского государственного педагогического института им. В.Г.Белинского. Пенза, 1956, т. 4, с. 75 - 87.

[Тра 67] Трахтенброт Б.А. Сложность алгоритмов и вычислений: Спецкурс для студентов НГУ. Новосибирск: Изд-во НГУ, 1967. 258 с.

Туэ А. (Thue A.)
[Туэ 10] Thue A. Die Lösung eines Spezialfalles eines generallen logischen Problems. - Skrifter utgit av Videnskapsselskapet i Kristiania, I. Matematisk-naturvidenskabelig klasse, 1910, N 8, 40 p. (Перепечатано в [Туэ 77], с. 273 - 310.)

[Туэ 14] Thue A. Probleme über Veränderungen von Zeichenreihen nach gegebenen Regeln. - Skrifter utgit av Videnskapsselskapet i Kristiania, I. Matematisk-naturvidenskabelig klasse, 1914, N 10, 34 p. (Перепечатано в [Туэ 77], с. 493 - 524.)

[Туэ 77] Thue A. Selected mathematical papers / Nagell T. et al., eds. With an introduction by Siegel K.L. Oslo et al.: Universitetsforlaget, 1977, 592 p.

Тьюринг А.М. (Turing A.M.)
[Тью 36] Turing A.M. On computable numbers, with an application to the Entscheidungsproblem. - Proceedings of LMS. Ser. 2, 1936, v.42, N 3, 4, p. 230-265. (Перепечатано в [Дей 65], с. 116 - 151.)

[Тью 37] Turing A.M. On computable numbers, with an application to the Entscheidungsproblem. A correction. Proceedings of LMS. Ser. 2, 1937, v.43, N 7, p. 544-546. (Перепечатано в [Дей 65], с. 152 - 154.)

[Тью 37а] Turing A.M. Computability and λ-definability. - JSL, 1937, v.2, N 4, p. 153-163.

[Тью 39] Turing A.M. Systems of logic based on ordinals. - Proceedings of LMS. Ser. 2, 1939, v.45, N 3, p. 161-228. (Перепечатано в [Дей 65], с. 155 - 222.)

Уанг П. (Wang P.)
[Уанг 74] Wang P. The undecidability of the existence of zeros of real elementary functions. - Journal of ACM, 1974, v.21, N 4, p. 586-589.

Успенский В.А.
[Усп 53] Успенский В.А. Теорема Гёделя и теория алгоритмов. - УМН, 1953, т. 8, № 4 (56), с. 176 - 178.

[Усп 53 а] Успенский В.А. Теорема Гёделя и теория алгоритмов.- ДАН, 1953, т. 91, № 4, с. 737 - 740.

[Усп 55] Успенский В.А. О вычислимых операциях. - ДАН, 1955, т. 103, № 5, с. 773 - 776.

[Усп 55а] Успенский В.А. Системы перечислимых множеств и их нумерации. - ДАН, 1955, т. 105, № 6, с. 1155 - 1158.

[Усп 56] Успенский В.А. Вычислимые операции и понятие программы. - УМН, 1956, т. 11, вып. 4 (70), с. 172 - 176.

[Усп 56а] Понятие программы и вычислимые операторы. - В кн.: ТТВМС. Т. 1, с. 186.

[Усп 57] Успенский В.А. К теореме о равномерной непрерывности. - УМН, 1957, т. 12, вып. 1 (73), с. 99 - 142.

[Усп 60] Успенский В.А. Лекции о вычислимых функциях. М.: Физматгиз, 1960. 492 с. (Французский перевод: Ouspenski V.A.

Leçons sur les fonctions calculables. Paris: Hermann, 1966, 412 p.)

[Усп 70] Успенский В.А. Алгоритм. - В кн.: БСЭ. 3-е изд. Т. I. 1970, с. 400 - 401.

[Усп 74] Успенский В.А. Теорема Гёделя о неполноте в элементарном изложении. - УМН, 1974, т. 29, вып. 1 (175), с. 3-47.

[Усп 77] Успенский В.А. Алгоритм. - В кн.: МЭ. Т. 1. 1977, с. 202 - 206.

[Усп 79] Успенский В.А. Машина Поста. М.: Наука, 1979. 95 с.

[Усп 82] Успенский В.А. Теорема Гёделя о неполноте. М.: Наука, 1982. 111 с.

Успенский В.А., Семёнов А.Л.

[Усп Сем 81] Uspensky V.A., Semenov A.L. What are the gains of the theory of algorithms: basic developments connected with the concept of algorithm and with its application in mathematics. - В кн.: [ЕршА Кнут 81], с. 100 - 234.

Ферранте Дж., Раков Ч. (Ferrante J., Rackoff C.W.)

[Фер Рак 79] Ferrante J., Rackoff C.W. The computational complexity of logical theories. Springer, 1979, 243 p. (LN in mathematics, v.718).

Фишер П. (Fisher P.C.)

[Фиш 74] Fisher P.C. Further schemes for combining matrix algorithms. -In: Automata, languages and programming. 2nd colloquium (Saarbrücken, July 29 - August 2, 1974). Springer, 1974 (LN in computer science, v.14), p. 428-436.

Фреге Г. (Frege G.)

[Фре 1879] Frege G. Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle, 1879, X+88 S. (Английский перевод: Frege G. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought. - В кн.: [Хей 67], с. 1 - 82.)

Фрейман К.В., Гриффит Дж.Е., Розенфельд Дж.Л. (Freiman C.V., Griffith J.E., Rosenfeld J.L.)

[Фрей Гриф Розенф 72] Freiman C.V., Griffith J.E., Rosenfeld J.L., eds. Information processing 71. Proceedings of IFIP congress 71 (Lubljana, August 23-28, 1971), 1622 p. (В двух томах).

Фридберг Р.М. (Friedberg R.M.)

[Фри 57] Friedberg R.M. Two recursively enumerable sets of incomparable degrees of unsolvability (solution of Post's problem 1944). - Proceedings of the National Academy of Sciences, v.43, N 2, p. 236-238.

[Фри 58] Friedberg R.M. Three theorems on recursive enumeration. I. Decomposition. II. Maximal set III. Enumeration without repetition. - JSL, 1958, v.23, N 3, p. 309-316.

Хакен В. (Haken W.)

[Хак 73] Haken W. Connections between topological and group

theoretical decision problems. -In: [Бук Канно Лин 73], p. 427-441.

Хартманис Дж. (Hartmanis J.)
[Хар 82] Hartmanis J. A note on natural complete sets and Gödel numberings. - Theoretical computer science, 1982, v.17, N 1, p. 75-89.

Хартманис Дж., Бейкер Т.П. (Hartmanis J., Baker T.P.)
[Хар Бей 75] Hartmanis J., Baker T.P. On simple Goedel numberings and translations. - SIAM Journal on computing, 1975, v.4, N 1, p. 1-11.

Хартманис Дж., Хопкрофт Дж. (Hartmanis J., Hopcroft J.E.)
[Хар Хоп 71] Hartmanis J., Hopcroft J.E. An overview of the theory of computational complexity. - Journal of ACM, 1971, v.18, N 3, p. 444-475. (Русский перевод: Хартманис Дж., Хопкрофт Дж.Э. Обзор теории сложности вычислений. - КС, 1974, вып. 11, с. 134 - 176.)

Хачиян Л.Г.
[Хач 79] Хачиян Л.Г. Полиномиальный алгоритм в линейном программировании. - ДАН, 1979, т. 244, № 5, с. 1093 - 1096.

Хейенорт Я.,ван (van Heijenoort J.)
[Хей 67] van Heijnoort J. From Frege to Gödel. A source book in mathematical logic, 1879-1931. Cambridge, Mass.: Harvard University Press, 1967, VII+660 p.

Хермес Х. (Hermes H.)
[Хер 65] Hermer H. Enumerability. Decidability. Computability. An introduction to the theory of recursive fuctions. Springer, 1965, IX+245 p.

Хоор К. (Hoare C.A.R.)
[Хоор 69] Hoare C.A.R. An axiomatic basis for computer programming. - Communications of ACM, 1969, v.12, N 10, p. 576-580, 583.

Хопкрофт Дж., Пансьо Ж.-Ж. (Hopcroft J., Pansiot J.-J.)
[Хоп Пан 79] Hopcroft J., Pansiot J.-J. On the reachability problem for 5-dimensional vector addition systems. - Theoretical computer science, 1979, v.8, N 2, p. 135-159.

Хопкрофт Дж.Е., Пауль В.Й., Вэлиэнт Л. (Hopcroft J.E., Paul W.J., Valiant L.G.)
[Хоп Пау Вэл 77] Hopcroft J.E., Paul W.J., Valiant L.G. On time versus space. - Journal of ACM, v.24, N 2, p. 332-337.

Хопкрофт Дж.Е., Тэрджен Р.Е. (Hopcroft J.E., Tarjan R.E.)
[Хоп Тэр 72] Hopcroft J.E., Tarjan R.E. Isomorphism of planar graphs. -In: Complexity of computer computations (Proceedings of the symposium, IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., 1972), New York: Plenum, 1972, p. 131-152. (Русский перевод: Хопкрофт Дж.Е., Тарьян Р.Е. Изоморфизм планарных графов. - КС, 1975, вып. 12, с. 39 - 61.)

Хопкрофт Дж.Е., Ульман Дж. (Hopcroft J.E., Ullman J.D.)

[Хоп Уль 69] Hopcroft J.E., Ullman J.D. Formal languages and their relation to automata. AW, 1969, X+242 p.

Хопкрофт Дж.Е., Уонг Дж.К. (Hopcroft J.E., Wong J.K.)

[Хоп Уонг 74] Hopcroft J.E., Wong J.K. A linear time algorithm for isomorphism of planar graphs: Preliminary report. -In: Sixth annual ACM symposium on theory of computing (Seattle, Wash., April 30 - May 2, 1974). N.Y.: ACM, 1974, p. 172-184.

Хуторецкий А.Б.

[Хут 69] Хуторецкий А.Б. О сводимости вычислимых нумераций.- АиЛ, 1969, т. 8, № 2, с. 251 - 264.

[Хут 71] Хуторецкий А.Б. О мощности верхней полурешетки вычислимых нумераций. - АиЛ, 1971, т. 10, № 5, с. 561 - 569.

Цейтин Г.С.

[Цей 58] Цейтин Г.С. Ассоциативное исчисление с неразрешимой проблемой эквивалентности. - В кн.: ПКНМ. 1. М.; Л.: Изд-во АН СССР, 1958 (Труды МИАН, т. 52), с. 172 - 189.

[Цей 59] Цейтин Г.С. Алгоритмические операторы в конструктивных полных сепарабельных матрических пространствах. - ДАН, 1959, т. 128, № 1, с. 49 - 52.

[Цей 62] Цейтин Г.С. Алгоритмические операторы в конструктивных метрических пространствах. - В кн.: ПКНМ. 2. М.; Л.: Изд-во АН СССР, 1962 (Труды МИАН, т. 67), с. 295 - 361.

[Цей 62а] Цейтин Г.С. Теоремы о среднем значении в конструктивном анализе. - В кн.: ПКНМ. 2. М.; Л.: Изд-во АН СССР, 1962 (Труды МИАН, т. 67), с. 362 - 384.

[Цей 64] Цейтин Г.С. Один способ изложения теории алгорифмов и перечислимых множеств. - В кн.: ПКНМ. 3. М.; Л.: Наука, 1964 (Труды МИАН, т. 72), с. 69 - 98.

[Цей 71] Цейтин Г.С. Приведенная форма нормальных алгоритмов и теорема о линейном ускорении. - В кн.: ИКММЛ. ЦУ. Л.: Наука, 1971 (Записки НС ЛОМИ, т. 20), с. 234 - 242.

Чёрч А. (Church A.)

[Чёрч 36] (Church A.) An unsolvable problem of elementary number theory. - American journal of mathematics, 1936, v.58, N 2, p. 345-363. (Перепечатано в [Дей 65], с. 89 - 107.)

[Чёрч 36а] Church A. A note on the Entscheidungsproblem. - JSL, 1936, v.1, N 1, p. 40-41. (Перепечатано с учетом исправлений, указанных в [Чёрч 36б], в [Дей 65], с. 110 - 115.)

[Чёрч 36б] Church A. Correction to a note on the Entscheidungsproblem. JSL, 1936, v.1, N 3, p. 101-102.

[Чёрч 40] Church A. On the concept of a random sequence. - Bulletin of AMS, 1940, v.46, N 2, p. 130-135.

[Чёрч 41] Church A. The calculi of lambda-conversion. Princeton, N.J.: Princeton University Press, 1941. 77 p. (Annals of mathematics studies, number 6).

[Чёрч 56] Church A. Introduction to mathematical logic. V. 1. Princeton, N.J.: Princeton University Press, 1956, IX+376 p. (Русский перевод: Чёрч А. Введение в математическую логику. Т. I. М.: ИЛ, I960. 485 с.)

Шанин Н.А.

[Шан 55] Шанин Н.А. О некоторых логических проблемах арифметики. М.: Изд-во АН СССР, I955. II2 с. (Труды МИАН, т. 43.)

[Шан 56] Шанин Н.А. Некоторые вопросы математического анализа в свете конструктивной логики. - ZmLGM, 1956, Bd. 2, H. 1, S. 27-36.

[Шан 58] Шанин Н.А. О конструктивном понимании математических суждений. - В кн.: ПКНМ. Т. М.; Л.: Изд-во АН СССР, I958 (Труды МИАН, т. 52), с. 226 - 3II.

[Шан 58а] Шанин Н.А. Об алгоритме конструктивной расшифровки математических суждений. - ZmLGM, 1958, Bd. 4, H. 4, S. 293-303.

[Шан 62] Шанин Н.А. Конструктивные вещественные числа и конструктивные функциональные пространства. - В кн.: ПКНМ. 2. М.; Л.: Изд-во АН СССР (Труды МИАН, т. 67), с. I5 - 294.

[Шан 70] Шанин Н.А. О рекурсивном математическом анализе и исчислении арифметических равенств Р.Л.Гудстейна. - В кн.: Гудстейн Р.Л. Рекурсивный математический анализ. Пер. с англ. М.: Наука, I970, с. 7 - 76.

[Шан 73] Шанин Н.А. Об иерархии способов понимания суждений в конструктивной математике. - В кн.: ПКНМ. 6. Л.: Наука, I973 (Труды МИАН, т. I29), с. 203 - 266.

Шахматный кодекс СССР

[Шахм 69] Шахматный кодекс СССР. 9-е издание. М.: Центральный шахматный клуб, I969. 48 с.

[Шахм 8I] Шахматный кодекс СССР. II-ое издание, испр. и доп. М.: Физкультура и спорт, I98I. 64 с.

Шеннон К. (Shannon C.)

[Шенн 48] Shannon C. A mathematical theory of communication. - Bell system technical journal, 1948, v.27, N 3, p. 379-423; N 4, p. 623-656. (Русский перевод: Математическая теория связи. - В кн.: Шеннон К. Работы по теории информации и кибернетике. Пер. с англ. / Добрушин Р.Л., Лупанов О.Б., редакторы. М.: Изд-во иностранной литературы, I963, с. 243 - 332.)

Шень А.Х.

[Шень 79] Шень А.Х. Метод приоритета и проблемы отделения. - ДАН, I979, т. 248, № 6, с. I309 - I3I3.

[Шень 80] Шень А.Х. Аксиоматический подход к теории алгоритмов и относительная вычислимость. - Вестник Московского университета. Сер. 1. Математика, механика, I980, № 2, с. 27-29.

[Шень 8I] Шень А.Х. Несколько замечаний о нумерациях, не являющихся натуральными. - В кн.: Математическая логика и математическая лингвистика. Калинин: Калининский государственный университет, I98I, с. I62 - I65.

[Шень 82] Шень А.Х. Частотный подход к определению понятия случайной последовательности. - Семиотика и информатика. М.: ВИНИТИ, 1982, вып. 18 (второй выпуск за 1981 г.), с. 14-42.

Шёнфилд Дж.Р. (Shoenfield J.R.)

[Шёнф 71] Shoenfield J.R. Degrees of unsolvability. NH, 1971, VIII+111 p. (Русский перевод: Шёнфилд Дж. Степени неразрешимости. М.: Наука, 1977. 192 с.)

Шёнхаге А. (Schönhage A.)

[Шёнх 70] Schönhage A. Universelle Turing Speicherung. -In: Automatentheorie und formale Sprachen. / Dorr J., Hotz G., eds. Mannheim, 1970, S. 369-383.

[Шёнх 80] Schönhage A. Storage modification machines. - SIAM Journal on computing, 1980, v.9, N 3, p. 490-508.

Шмульян Р.М. (Smullyan R.M.)

[Шму 58] Smullyan R. Theories with effectively inseparable nuclei. - JSL, 1958, v.23, N 4, p. 458.

[Шму 60] Smullyan R.M. Theories with effectively inseparable nuclei. - ZmLGM, 1960, Bd. 6, H. 3-4, S. 219-222.

[Шму 61] Smullian R.M. Theory of formal systems. Princeton (New Jersey): Princeton University Press, 1961. (Annals of mathematics studies, number 47.) XIV+142 p. (Русский перевод с исправленного издания 1962 г.: Смальян Р. Теория формальных систем. М.: Наука, 1981. 207 с.)

Шнорр К.П. (Schnorr C.P.)

[Шно 71] Schnorr C.P. Zufälligkeit und Wahrscheinlichkeit. Eine algorithmische Begründung der Wahrscheinlichkeitstheorie. Springer, 1971, iv+212 S. (LN in mathematics, v.218).

[Шно 72] Schnorr C.P. Optimal Gödel numberings. -In: Information processing 71. Proceedings of IFIP congress 71 (Lubljana, August 23-28, 1971). V. 1. Foundations and systems. / Freiman C.V. et al., eds. NH, 1972, p. 56-58.

[Шно 73] Schnorr C.P. Process complexity and effective random tests. - JCSS, 1973, v.7, N 4, p. 376-388.

[Шно 75] Schnorr C.P. Optimal enumerations and optimal Gödel numberings.- Mathematical systems theory, 1975, v.8, N 2, p. 182-191.

[Шно 77] Schnorr C.P. A survey of the theory of random sequences. - In: [Батс Хин 77] p. 193-211.

Шор Р.А. (Shore R.A.)

[Шор 81] Shore R.A. The degrees of unsolvability: global results. -In: Logic year 1979-80. / Lerman M., Schmerl J., Soare R., eds. Springer, 1981, VIII+326 p. (LN in mathematics, v.859), p. 283-301.

Шпекер Э. (Specker E.)

[Шпе 49] Specker E. Nicht Konstruktiv beweisbare Sätze der Analysis. - JSL, 1949, v.14, N 3, p. 145-158.

Шрёдер Э. (Schröder E.)
[Шрё] 1887 Schröder E. Uber Algorithmen und Kalkuln. - Archiv für Mathematik und Physik, 1887, 2. Reihe, Teil 5. (Русский перевод: Шрёдер Э. Об алгорифмах и исчислениях. - Физико-математические науки в их настоящем и прошедшем. Журнал чистой и прикладной математики, астрономии и физики, издаваемый В.В.Бобыниным, 1888, т. 7, № 1, с. 76 - 85; № 2, с. 163 - 168; № 3, с. 229 - 242; № 4, с. 345 - 374.)

Эббинхауз Г.Д. (Ebbinghaus H.-D.)
[Эбб 70] Ebbinghaus H.-D. Aufzählbarkeit. -In: [Яко 70a], p. 64-113. (Русский перевод: Эббинхауз Г.Д. Перечислимость. - В кн.: [Яко 70a], русский перевод, с. 86 - 149.)

Яблонский С.В., Лупанов О.Б.
[Ябл Луп 74] Яблонский С.В., Лупанов О.Б., ред. Дискретная математика и математические вопросы кибернетики. Т. 1. М.: Наука, 1974. 312 с.

Якобс. К. (Jacobs K.)
[Яко 70] Jacobs K. Turing-Maschinen und zufällige 0-1-Folgen. - In: [Яко 70a], p. 141-167. (Русский перевод: Якобс К. Машины Тьюринга и случайные 0-1-последовательности. - В кн.: [Яко 70a], русский перевод, с. 183 - 215.)

[Яко 70a] Jacobs K., ed. Selecta mathematica, V. 2. Springer, 1970. (Русский перевод: Якобс К., ред. Машины Тьюринга и рекурсивные функции. М.: Мир, 1972. 264 с.)

Янгер Д.Х. (Younger D.H.)
[Янг 67] Younger D.H. Recognition and parsing of context-free languages in time n^3. - Information and control, 1967, v.10, N 2, p. 189-208. (Русский перевод: Янгер Д.Х. Распознавание и анализ контекстно-свободных языков. - В кн.: [Коз Муч 70], с. 344 - 362.)

Янов Ю.И.
[Янов 58] Янов Ю.И. О логических схемах алгоритмов. - ПК, 1958, вып. 1, с. 75 - 127.

Яновская С.А.
[Яновс 59] Яновская С.А. Математическая логика и основания математики. - В кн.: Математика в СССР за сорок лет. / Курош А.Г. и др., редакторы. Т. 1. М.: Физматгиз, 1959, с. 13-120.)

[Яновс 62] Яновская С.А. Исчисление. - В кн.: Философская энциклопедия. Т. 2, 1962, с. 387 - 390.