1974 г. январь—февраль *т. XXIX, вып. 1 (175)*
УСПЕХИ МАТЕМАТИЧЕСКИХ НАУК

УДК 517.19

ТЕОРЕМА ГЁДЕЛЯ О НЕПОЛНОТЕ В ЭЛЕМЕНТАРНОМ ИЗЛОЖЕНИИ

В. А. Успенский

Теорема Гёделя о неполноте отрицает возможность такой системы аксиом и правил вывода, при которой совокупность утверждений, выводимых из аксиом, совпадает с совокупностью истинных утверждений арифметики (более того, для всякой непротиворечивой системы может быть эффективно построено истинное, но не выводимое утверждение). Статья посвящена доказательству этой теоремы, опирающемуся на понятия и методы теории алгоритмов; необходимые сведения из теории алгоритмов сообщаются по мере надобности. Статья не требует никаких специальных знаний (в частности, из области математической логики), а предполагает лишь знакомство с элементарной общематической терминологией и символикой.

СОДЕРЖАНИЕ

Предисловие

Есть в математике темы, пользующиеся достаточной известностью и, в то же время, признаваемые традицией слишком сложными (или маловажными) для включения в обязательное обучение; обычай относит их к занятиям факультативным, дополнительным, специальным и т. п. В перечне таких тем есть несколько, остающихся сейчас там исключительно в силу инерции. Таковы комбинаторика в рамках школьного курса и теорема Гёделя о неполноте в рамках курса университетского.

Здесь не место говорить сколько-нибудь подробно о комбинаторике, которую когда-то проходили в выпускных классах, а сейчас в средней школе

не проходят вовсе [1]). Заметим лишь, что простейшие комбинаторные задачи доступны не только школьникам младших классов, но и дошкольникам.

Что же касается теоремы Гёделя о неполноте, то она, безусловно, достойна того, чтобы быть сообщаемой каждому, получающему высшее математическое образование. Вместе с тем причины, вызывающие неустраняемую неполноту (т. е. невозможность добиться того, чтобы каждое осмысленное утверждение было либо доказуемо, либо опровержимо [2])), столь просты, что теорема Гёделя могла бы излагаться на самых младших курсах.

Цель настоящей публикации — предложить способ изложения теоремы Гёделя о неполноте, претендующий одновременно и на достаточную общность и на достаточную простоту. Этот способ отличен от способа, предложенного самим К. Гёделем [3] [3]) и опирается на понятия и методы теории алгоритмов (которые, впрочем, не предполагаются известными читателю). По-видимому, впервые упоминание о теореме Гёделя в рамках алгоритмических рассмотрений было сделано Э. Л. Постом [12]. На общий характер связи теоремы Гёделя и теории алгоритмов указал А. Н. Колмогоров в конце 1952 г.; в начале 1953 г. это указание было развито автором этих строк в докладе [17] и заметке [18]. Предлагаемый ниже подход к вопросу о неполноте неоднократно излагался автором в Московском университете: в 1964 г. в лекциях для студентов отделения структурной и прикладной лингвистики филологического факультета (а впоследствии в лекциях для студентов механико-математического факультета), в 1966 г. в докладе на семинаре А. А. Маркова и С. А. Яновской на механико-математическом факультете и в 1967 г. на семинаре И. М. Гельфанда на том же факультете.

Статья не предполагает каких-либо специальных знаний.

План статьи таков.

В § 1 обсуждается первоначальная формулировка самого Гёделя. Этот параграф, строго говоря, не необходим для понимания дальнейшего: систематическое изложение начинается лишь с § 2, в котором дается первое, еще очень расплывчатое, представление о семантическом варианте теоремы о неполноте. В § 3 начинается уточнение этого представления и, в частности, вводится центральное для данной статьи понятие дедуктики.

В § 4 излагаются, на неформальном уровне, начальные понятия теории алгоритмов, и на их основе формулируются первые критерии полноты и неполноты. В § 5, носящем иллюстративный характер, эти критерии применяются к исследованию языков, связанных с ассоциативными исчислениями. В § 6 происходит дальнейшее развитие критериев неполноты.

---

[1]) В программе по математике для поступающих в вузы по 1971 г. включительно содержался раздел VII «Теория соединений. Бином Ньютона». Однако этот раздел из года в год (начиная с 1967 г.) снабжался подстрочным примечанием «Раздел VII из программы вступительных экзаменов для поступающих в вузы в 19 ** г. исключается». В программах на 1972 г. и 1973 г. какое-либо упоминание о соединениях отсутствует вовсе.

[2]) В другом варианте — чтобы каждое истинное утверждение было доказуемо.

[3]) Популярный очерк идей, лежащих в основе гёделевского доказательства теоремы о неполноте, дан в брошюре Э. Нагеля и Дж. Ньюмена [11], а строгое изложение этого доказательства можно найти в монографиях С. К. Клини [4] и Э. Мендельсона [10].

Хотя язык формальной арифметики хорошо известен, в целях внутренней законченности изложения он описывается в § 7; здесь же делается первый шаг к установлению неполноты этого языка. В § 8 на основе дальнейшего развития тех представлений об алгоритмах, которые были начаты в § 4,— развития, закрепляемого в виде трех аксиом теории алгоритмов,— завершается доказательство теоремы о неполноте формальной арифметики (в семантической форме).

§§ 9 и 10 носят более специальный (хотя по-прежнему элементарный) характер. Здесь выясняется возможность эффективного построения — по каждому возможному «уточнению» понятия доказательства — истинного, но не доказуемого утверждения. Предлагаются соответствующие точные формулировки.

§ 1. Исходная гёделевская формулировка

Знаменитая работа Курта Гёделя «Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I» («О формально неразрешимых предложениях Principia Mathematica [1]) и родственных систем I») была напечатана на стр. 173—198 1-й тетради 38-го тома (за 1931 г.) лейпцигского журнала «Monatshefte für Mathematik und Physik» (поступила 17.XI.1930). Предварительная сводка результатов была опубликована в венском журнале Anzeiger der Akademie der Wissenschaften in Wien, Mathematisch-naturwissenschaftliche Klasse, № 19 за 1930 г.) (заседание от 23 октября 1930 г.).

В этой статье для широкого класса формальных систем устанавливалось неизбежное существование в каждой из таких систем неразрешимого утверждения — неразрешимого в том смысле, что ни оно, ни его отрицание не могло быть выведено из аксиом системы. Именно, в статье Гёделя была сформулирована следующая теорема (теорема VI на стр. 187):

> Для каждого ω-непротиворечивого рекурсивного класса $\varkappa$ *формул* существует такая рекурсивная *классовая формула* r, что ни $v\,\mathrm{Gen}\,r$, ни $\mathrm{Neg}(v\,\mathrm{Gen}\,r)$ не принадлежат к $\mathrm{Flg}(\varkappa)$ (где v есть *свободная переменная формулы* r).

Дадим некоторые пояснения к приведенным формулировкам. Эти пояснения предполагают наличие у читателя простейших сведений из математической логики; зато они не необходимы для понимания дальнейшего.

Речь здесь идет о формулах некоторой формальной системы P, которая строится на страницах 176—178 статьи Гёделя. Мы не будем приводить точных формулировок, а ограничимся следующей цитатой из Гёделя: «В сущности, P есть та система, которая получается, если надстроить пеановские аксиомы логикой Principia Mathematica (числа [2]) в качестве индивидов, отношение „следования за" в качестве неопределяемого понятия)» (стр. 176). Курсив несет на себе определенный смысл. Он означает, что речь идет не непо-

---

[1]) Имеется в виду монография: A. Whitehead, B. Russell, Principia Mathematica, изд. 2, Cambridge, 1925.

[2]) Как в статье Гёделя, так и в данной статье, под «числами» всегда понимаются натуральные числа, причем 0 считается натуральным числом.

средственно о знакосочетаниях рассматриваемой формальной системы (переменных, формулах и т. п.), а о номерах этих знакосочетаний в некоторой фиксированной нумерации (называемой теперь гёделевской). Классовая формула — это формула с одной свободной переменной. Стало быть, *классовая формула* — это натуральное число, являющееся номером классовой формулы. Через v Gen r обозначается номер формулы, полученной навешиванием квантора общности по переменной с номером v на формулу с номером r; через Neg (v Gen r) — номер отрицания предыдущей формулы. Через Flg ($\varkappa$) обозначается класс номеров всех тех и только тех формул, которые выводимы из класса формул, номера которых образуют класс $\varkappa$ [1]). Термины «рекурсивный класс» и «рекурсивная формула» мы оставим без объяснений; эти термины означают некоторую определимость рассматриваемых классов и формул с помощью примитивно-рекурсивных функций (в статье Гёделя такие функции называются просто «рекурсивными»). Свойство ω-непротиворечивости, налагаемое на класс, означает условие более сильное, нежели простая непротиворечивость [2]). Если непротиворечивость класса означает невозможность вывести из него как некоторую формулу, так и ее отрицание, то ω-непротиворечивость означает невозможность вывести как некоторую формулу вида «существует такое x, что $\mathfrak{A}(x)$», так и все формулы вида «не $\mathfrak{A}(0)$», «не $\mathfrak{A}(1)$», «не $\mathfrak{A}(2)$» и т. д. В обозначениях обсуждаемой статьи Гёделя класс $\varkappa$ *формул* (т. е. номеров формул) называется ω-непротиворечивым, если не существует *классовой формулы* a, для которой 1) Neg (v Gen a) $\in$ Flg ($\varkappa$), 2) $Sb(a^v_{Z(n)}) \in$ Flg($\varkappa$) при всех n (здесь $Sb(a^v_{Z(n)})$ означает номер результата подстановки в формулу с номером a формулы с номером $Z(n)$ вместо переменной с номером v, причем $Z(n)$ — номер выражения для числа n). Таким образом, теорема VI гласит, что для любого класса формул, подчиненного некоторым условиям, существует формула сравнительно простого вида, такая что ни она, ни ее отрицание не выводимы из этого класса. Поскольку в основе формальной системы P, подразумеваемой в названной теореме (ведь речь идет о формулах этой системы и о выводимости по правилам этой системы), лежат арифметические аксиомы Пеано, то сама эта теорема часто интерпретируется как теорема о неполноте формальной арифметики. Неполнота понимается здесь как существование неразрешимых утверждений.

Замечание 1. Если под формальной арифметикой понимать систему P, то неполнота формальной арифметики представляет собой весьма частный случай теоремы VI, получающийся при $\varkappa = \phi$ (и справедливый в предположении, что сама P является ω-непротиворечивой, т. е. что ω-непротиворечив класс ее аксиом); в этом случае Flg($\varkappa$) состоит просто из номеров всех формул, доказуемых в P.

Замечание 2. Правда, сама неразрешимая формула, указываемая в теореме VI, а именно формула с номером v Gen r, еще не имеет арифметического характера, т. е. еще не записана на простейшем арифметическом языке. Однако на этот счет в статье Гёделя содержатся важные дальнейшие результаты. Именно, формула называется «арифметической», если она строится с помощью переменных, пробегающих натуральный ряд, отношения равенства и операций сложения и умножения [3]). Далее, на стр. 193 статьи Гёделя формулируется теорема VIII:

В каждой формальной системе, упоминаемой в теореме VI, существуют неразрешимые арифметические утверждения.

---

[1]) Вывод из произвольного класса формул предполагает возможность использовать в процессе вывода также и аксиомы, так что в данном случае происходит присоединение класса $\varkappa$ к аксиомам исходной системы.

[2]) Впоследствии Россер [14] усилил первоначальную формулировку Гёделя, заменив требование ω-непротиворечивости более слабым требованием непротиворечивости.

[3]) Заметим, что знаки $=$, $+$, $\cdot$ не входят в исходный алфавит системы P. Поэтому «арифметическая формула» может существовать лишь в подходящем расширении системы P. В рамках же P эти знаки следует рассматривать как сокращающие. Так, выражение $x_1 = y_1$ понимается, согласно подстрочному примечанию 21 на стр. 177, как сокращение для формулы «$x_2\Pi(x_2(x_1) \supset x_2(y_1))$»; здесь $x_2\Pi$ означает квантор общности по x_2. (Для $x + y$ и $x - y$ таких расшифровок в статье Гёделя не приводится.)

З а м е ч а н и е 3. Как указывает Гёдель (стр. 190), его доказательство теоремы VI проходит не только для конкретной системы P, о которой идет речь в его статье, но для любой системы, обладающей следующими основными свойствами:

1) аксиомы и правила вывода системы рекурсивно определимы;

2) каждое рекурсивное отношение определимо внутри системы.

Как отмечает Гёдель, эти свойства выполняются для аксиоматических систем теории множеств Цермело — Френкеля и фон Неймана, а также для аксиоматической теории чисел, основанной на аксиомах Пеано и рекурсивных определениях. Во всех этих системах существуют, следовательно, неразрешимые предложения: чтобы обнаружить это, достаточно положить $\varkappa = \phi$ (ср. выше замечание 1). Правда, утверждение предыдущей фразы справедливо лишь в предположении ω-непротиворечивости рассматриваемой системы. Это предположение во всех конкретных случаях образует рабочую гипотезу, вытекающую из нашего убеждения в разумности рассматриваемой системы, т. е. в том, что она адекватно отражает некоторую реальность.

§ 2. Синтаксическая и семантическая формы теоремы о неполноте

Как уже отмечалось, сделанные в предыдущем параграфе комментарии не обязательны для понимания дальнейшего. Нам достаточно следующее приблизительное толкование Гёделевой теоремы VI: при определенных условиях, накладываемых на аксиомы и правила вывода, существует такое утверждение (формулируемое на том же языке, что и аксиомы), что ни оно, ни его отрицание не выводимы из указанных аксиом по указанным правилам. Важно подчеркнуть, что, во-первых, и аксиомы, и утверждения, о которых говорится в теореме VI, представляют собой так называемые «формулы», т.е. просто комбинации знаков, или букв, образованные согласно некоторым принятым «правилам образования», что, во-вторых, сами понятия «утверждение» и «отрицание утверждения» определены также совершенно формально как комбинации знаков, имеющие определенное строение (без ссылки на то, что эти комбинации на самом деле что-то утверждают или отрицают), что, далее, в-третьих, правила вывода формулируются чисто комбинаторно в виде разрешенных преобразований одних цепочек знаков в другие и что, наконец, в-четвертых, таким же «внешним», комбинаторным образом формулируются и условия, накладываемые в теореме VI на аксиомы и правила вывода. Вся теория имеет, следовательно, чисто комбинаторный характер, нигде не происходит апелляции к смыслу рассматриваемых знаков и знакосочетаний [1]). Поскольку исследование способов образования и преобразования используемых в математической логике знакосочетаний относится к области так называемого логического синтаксиса, подобную теорему о неполноте естественно назвать *теоремой о неполноте в синтаксической форме*. Предположим теперь, что рассматриваемым знакам и их разрешенным комбинациям придан некоторый смысл, так что каждое утверждение (при описанном выше формальном понимании термина «утверждение») оказывается истинным или ложным, причем отрицание истинного утверждения оказывается ложным и наоборот [2]). Тогда или само неразрешимое утверждение, существующее в силу синтаксической теоремы о неполноте, или его отрицание, оказывается

[1]) Такая апелляция, разумеется, играет решающую роль при выборе тех или иных аксиом, правил преобразования и т. п., но ее нет в окончательных формулировках.

[2]) Мы предполагаем, что отрицание утверждения также является утверждением.

истинным; в обоих случаях мы получаем пример истинного утверждения, не выводимого из аксиом. Итак, при определенных условиях для заданных аксиом и заданных правил вывода существует истинное утверждение, не выводимое из этих аксиом посредством этих правил. Теорему подобного вида естественно назвать *теоремой о неполноте в семантической форме*.

К проведенному рассуждению, выводящему семантическую формулировку теоремы о неполноте из синтаксической, следует подходить с осторожностью.

Во-первых, приписывание смысла цепочкам знаков (и возникающая в этой связи семантическая формулировка) имеет содержательную ценность, лишь если оно происходит не произвольно, а неким естественным образом. Эта «естественность» предполагает, что истинность или ложность некоторых простейших цепочек задана заранее (например, $0=0$ есть истинная цепочка, а $0=1$ — ложная), а истинность или ложность более сложных цепочек определяется через истинность или ложность более простых. (Занимаясь, в рамках данной статьи, семантической формулировкой теоремы о неполноте, мы будем в дальнейшем исходить из того, что некоторое определенное приписывание смысла цепочкам знаков задано заранее.)

Во-вторых, сама возможность естественного приписывания истинного или ложного значения всем формулам, формально опознаваемым как утверждения (так называемым «формулам без свободных переменных»), далеко не всегда представляется бесспорной. Если, например, говорить об аксиоматической теории множеств (которая удовлетворяет условиям синтаксической теоремы о неполноте), то вопрос об истинности и ложности ее «утверждений» носит достаточно сложный характер. Здесь имеется в виду не то, как узнать, истинно или ложно данное «утверждение», а что означает, как понимается его истинность или ложность. Что, в самом деле, означает истинность или ложность аксиомы выбора или континуум-гипотезы? Однако в целом ряде важных случаев, например, в применении к утверждениям элементарной арифметики, истинность или ложность имеет очевидный (с наивной точки зрения) смысл. Так, считается, что великая теорема Ферма является либо истинной либо ложной независимо от нашей способности это установить. (Мы не касаемся здесь точки зрения конструктивного направления в математике, отрицающей априорную убежденность в истинности или ложности утверждений, подобных теореме Ферма.)

Пусть теперь в ситуации, когда истинность и ложность утверждений как-то определены, имеется истинное утверждение A, не выводимое из аксиом. При естественном предположении, что все аксиомы суть истинные утверждения и что правила вывода сохраняют такую истинность, все выводимые из аксиом утверждения оказываются истинными, и потому отрицание утверждения A, будучи ложным, не является выводимым. Следовательно, ни A, ни отрицание A не выводимо, и мы получаем синтаксическую формулировку теоремы о неполноте как следствие семантической формулировки.

Со времени работы Гёделя появилось много формулировок теоремы о неполноте (исчерпывающий обзор таких формулировок приведен в монографии Ладриера [5]).

В настоящей статье мы займемся семантическим вариантом теоремы о неполноте, причем постараемся сформулировать этот вариант при возможно более общих допущениях.

Итак, наша ближайшая цель — найти возможно более общую формулировку семантической теоремы о неполноте. В поисках такой общности мы прежде всего откажемся от рассмотрения каких-либо аксиом и правил вывода. Вывод из аксиом по заданным правилам мы будем рассматривать как частный случай более общего понятия доказательства. Мы будем говорить, стало быть, об истинных, но не доказуемых утверждениях. Что же касается того, что́ такое «истинное утверждение», то здесь нам будет достаточно общего представления, что некоторые комбинации знаков являются истинными утверждениями. Однако, как показал уже предыдущий параграф, точные формулировки могут потребовать предварительного развития достаточно громоздкого аппарата. Мы поэтому предложим формулировку, которая на первых порах может быть понята лишь приблизительно, а затем вложим в нее точный смысл. В качестве такой формулировки мы выберем следующую:

«при определенных условиях в языке существует недоказуемое истинное утверждение».

Основная часть статьи будет посвящена осмыслению и доказательству этой формулировки.

§ 3. Уточнение основных терминов

Итак, наша семантическая теорема о неполноте гласит, что при определенных условиях в языке существует недоказуемое истинное утверждение. В этой формулировке едва ли не каждое слово нуждается в разъяснениях. Сделаем такие разъяснения.

1. Язык. Мы не будем давать какого бы то ни было определения языка (поскольку не беремся это сделать с достаточной общностью), а ограничимся теми относящимися к языку понятиями, которые единственно и будут нужны нам для дальнейшего. Таких понятий нам потребуется два: «алфавит языка» и «множество истинных утверждений языка».

1.1. Алфавит. Следуя А. А. Маркову [8], под *алфавитом* понимается конечный список элементарных (т. е. считающихся не членимыми далее) знаков, называемых *буквами* этого алфавита. Конечная цепочка следующих друг за другом букв некоторого алфавита называется *словом* в этом алфавите. Так, слова русского языка (включая и собственные имена) суть слова в 66-буквенном алфавите (33 строчные буквы, 31 прописная буква [1]), дефис, апостроф); десятичные записи натуральных чисел — слова в десятибуквенном алфавите {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. Для называния алфавитов используются обычно прописные русские буквы. Множество всех слов в алфавите Б будем обозначать $Б^{\infty}$. Предполагается, что для каждого языка имеется такой алфавит, что все выражения этого языка (т. е. имена тех или иных предметов, утверждения об этих предметах и т. п.) суть слова в этом алфавите; каждую русскую фразу, например, и даже каждый русский текст можно

[1]) Кроме твердого и мягкого знаков.

рассматривать как слово в алфавите, представляющем собой расширение указанного выше 66-буквенного алфавита за счет знаков препинания, знака пробела между словами, знака абзацного отступа и, быть может, еще некоторых знаков). Предполагая, что выражения языка являются словами в некотором алфавите, мы тем самым налагаем запрет на такое «многоэтажное» выражение, как, например, $\int_a^b f(x)dx$, или такие плоские фигуры, как знакосочетания по Бурбаки [1]. Этот запрет, однако, не является слишком ограничительным, поскольку все подобные выражения можно, при подходящей кодировке, «вытянуть в строку». Всякое множество M, такое что $M \subseteq$ $\subseteq$ Б$^\infty$, называется *словарным* в Б. Просто *словарным* называется множество, словарное в каком-либо алфавите. Сделанное только что предположение может быть теперь сформулировано короче: множество выражений всякого языка словарно.

1.2. Множество истинных утверждений. Предполагается, что в множестве Б$^\infty$, где Б — алфавит рассматриваемого языка, задано подмножество T, называемое множеством «истинных утверждений» (или, короче, просто «истин»). Таким образом, мы опускаем все промежуточные этапы, посредством которых, во-первых, среди слов в алфавите Б выделяются правильно построенные *выражения* языка, получающие определенный смысл при интерпретации (такие, как $2 + 3$, $x + 3$, $x = y$, $x = 3$, $2 = 3$, $2 = 2$ — в отличие от таких как $+ = x$); во-вторых, среди выражений выделяются так называемые *формулы*, означающие при интерпретации «утверждения, зависящие, быть может, от параметра» (такие, как $x = 3$, $x = y$, $2 = 3$, $2 = 2$); в-третьих, среди формул выделяются так называемые *замкнутые формулы*, или утверждения, не зависящие от параметра, (такие, как $2 = 3$, $2 = 2$); и, лишь в-четвертых, среди утверждений выделяются *истинные утверждения* (такие, как $2 = 2$).

1.3. Для наших целей будет достаточным считать язык полностью заданным, коль скоро задан алфавит Б и подмножество T множества Б$^\infty$. Всякую такую пару $\langle$Б, $T\rangle$ мы будем называть *фундаментальной парой*.

2. Недоказуемое. «Недоказуемое» значит не являющееся доказуемым, а «доказуемое» значит имеющее доказательство.

3. Доказательство. Хотя термин «доказательство» является едва ли не самым главным в математике [1]), он не имеет точного определения. Понятие доказательства во всей его полноте принадлежит математике не более, чем психологии: ведь доказательство — это просто рассуждение, убеждающее нас настолько, что с его помощью мы готовы убеждать других

3.1. Будучи записанным, доказательство становится словом в некотором алфавите Д (вспомним, что говорилось выше о русских текстах); все доказательства образуют некую (достаточно, впрочем, расплывчатую) совокупность в Д$^\infty$. Не претендуя на то, чтобы дать точное определение для такого «наивного» или «абсолютного» понятия доказательства (или, что то же самое, для соответствующей совокупности в Д$^\infty$), мы займемся его формальным

[1]) Н. Бурбаки начинает свои «Начала математики» словами «Со времен греков говорить „математика“ — значит говорить «доказательство»».

аналогом, для которого, однако, сохраним тот же термин «доказательство». Этот аналог в двух существенных чертах будет отличаться от интуитивного понятия [1]): во-первых, мы будем допускать существование разных понятий «доказательство» (что приведет к различным подмножествам в множестве $Д^{\infty}$, да и сам алфавит Д может варьироваться); во-вторых, для каждого из таких понятий мы будем требовать эффективного способа проверки, является ли данное слово в алфавите Д доказательством или нет. Далее, будем предполагать наличие алгоритма, который каждому доказательству ставит в соответствие утверждение, имеющее это доказательство, или *доказываемое* этим доказательством (в обычных случаях таковым является последнее утверждение в цепочке, образующей доказательство).

3.2. Итак, окончательное определение таково:

1°. Имеются алфавиты Б (*алфавит языка*) и Д (*алфавит доказательств*).

2°. В множестве $Д^{\infty}$ выделено подмножество D, элементы которого называются *доказательствами*; предполагается наличие алгоритма, позволяющего по произвольному слову в алфавите Д узнавать, принадлежит оно D или нет.

3°. Имеется функция δ (*функция выделения доказанного*), у которой область определения Δ удовлетворяет соотношению $D \subseteq \Delta \subseteq Д^{\infty}$ и которая принимает свои значения в $Б^{\infty}$; предполагается наличие алгоритма, вычисляющего [2]) эту функцию; доказательство d из D называется доказательством слова $\delta(d)$.

3.3. Тройку $\langle Д, D, \delta \rangle$, удовлетворяющую условиям 1° — 3°, назовем *дедуктикой* над алфавитом Б.

3.4. Следующий, *регулярный* способ задания дедуктики охватывает обычные приемы задания понятия доказательства посредством «аксиом» и «правил вывода». Пусть Б — алфавит рассматриваемого языка; обозначим $Б^{\infty}$ через X и $\underbrace{X \times \ldots \times X}_{\mu \text{ раз}}$ через X^{μ}. Пусть $m > 0$ и пусть в каждом X^{μ} ($\mu = 1, 2, \ldots, m$) задано некоторое подмножество S_{μ}, причем для каждого μ существует алгоритм, позволяющий по произвольному элементу из X^{μ} распознавать, принадлежит он к S_{μ} или нет. Элементы множества S_{μ} называются *аксиомами* при $\mu = 1$ и *правилами вывода* при $\mu > 1$. *Доказательством* назовем такую цепочку $\langle C_1, \ldots, C_n \rangle$ слов из X, в которой каждый член C_k удовлетворяет по крайней мере одному из следующих m условий:

1) $C_k \in S_1$.

2) Существует такое i, что $i < k$ и $\langle C_i, C_k \rangle \in S_2$.

. . . . . .

μ) Существуют такие $i_1, \ldots, i_{\mu-1}$, что $i_1 < k, \ldots, i_{\mu-1} < k$ и $\langle C_{i_1}, \ldots, C_{i_{\mu-1}}, C_k \rangle \in S_{\mu}$.

. . . . . .

m) Существуют такие $i_1, \ldots, i_{m-1}$, что $i_1 < k, \ldots, i_{m-1} < k$ и $\langle C_{i_1}, \ldots, C_{i_{m-1}}, C_k \rangle \in S_m$.

[1]) Впрочем, и интуитивное понятие не вовсе лишено этих черт.

[2]) Этот термин уточняется в следующем параграфе.

Каждое доказательство будем считать доказательством своего последнего члена, т. е. $\langle C_1, \ldots, C_n \rangle$ есть доказательство слова C_n. Чтобы сделать это определение частным случаем определения из п. 3.2, нужно условиться каждую цепочку слов считать словом. Для этого достаточно ввести в рассмотрение новую, не входящую в исходный алфавит Б букву, используя ее в качестве разделительного знака между словами. Такую новую букву мы будем обозначать звездочкой $*$. Таким образом, цепочку $\langle C_1, \ldots, C_n \rangle$ слов в алфавите Б мы будем отождествлять со словом $C_1 * \ldots * C_n$ в алфавите Б $\cup \{*\}$. Заметим, что произвольное слово в Б $\cup \{*\}$ однозначно представляется в виде $C_1 * \ldots * C_n$, где каждое C_i — слово (быть может, пустое) в Б. Итак, в нашей дедуктике Д = Б $\cup \{*\}$, D — множество слов $C_1 * \ldots * C_n$, у которых каждое C_k подчинено одному из условий 1) — m), и $\delta(C_1 * \ldots * C_n) = C_n$.

4. Попытки уточнения первоначальной формулировки

4.1. Первая попытка. При определенных условиях для фундаментальной пары $\langle$Б, $T\rangle$ и дедуктики $\langle$Д, D, $\delta\rangle$ над Б существует слово из T, не имеющее доказательства. Такая формулировка еще слишком неопределенна. К тому же ясно, что можно придумать много дедуктик, в каждой из которых будет очень мало доказуемых слов. В пустой дедуктике (где $D = \varnothing$) вообще нет ни одного доказуемого слова.

4.2. Вторая попытка. Более естественным является другой подход. Задан некоторый язык, в том точном смысле, что задана фундаментальная пара $\langle$Б, $T\rangle$. Мы теперь ищем дедуктики над Б (содержательно — ищем такие способы доказывания), в которых доказывалось бы как можно больше слов из T, в идеале — все слова из T. Нас интересует ситуация, когда такой дедуктики (в которой каждое слово из T имело бы доказательство), не существует. Итак, нас заинтересовала бы следующая формулировка: при определенных условиях, налагаемых на фундаментальную пару $\langle$Б, $T\rangle$, не существует дедуктики над Б, в которой каждое слово из T имеет доказательство. Однако пары $\langle$Б, $T\rangle$ с этим свойством просто не может быть. В самом деле, достаточно положить Д = Б, D = Д$^\infty$, $\delta(d) = d$ для всякого d из Д$^\infty$; тогда всякое слово из Б$^\infty$ окажется доказуемым (его доказательством будет оно само).

5. Непротиворечивость. Естественно потребовать, чтобы доказуемыми были лишь «истинные утверждения», т. е. слова, принадлежащие множеству T. Назовем дедуктику $\langle$Д, D, $\delta\rangle$ *непротиворечивой относительно* (или *для*) фундаментальной пары $\langle$Б, $T\rangle$, коль скоро $\delta(D) \subseteq T$. В дальнейшем будем интересоваться лишь непротиворечивыми дедуктиками. Если имеется язык, то представляется весьма заманчивым найти такую непротиворечивую дедуктику, в которой каждое истинное утверждение было бы доказуемым. Теорема Гёделя в интересующем нас варианте именно и утверждает, что при определенных условиях, налагаемых на фундаментальную пару, этого сделать нельзя.

6. Полнота. Назовем дедуктику $\langle$Д, D, $\delta\rangle$ *полной относительно* (или *для*) фундаментальной пары $\langle$Б, $T\rangle$, коль скоро $\delta(D) \supseteq T$. Занимающая нас формулировка приобретает такой вид:

при определенных условиях, налагаемых на фундаментальную пару $\langle Б, T\rangle$, *не существует дедуктики над* Б, *полной и непротиворечивой относительно* $\langle Б, T\rangle$.

На этой формулировке мы и остановимся, и в следующих параграфах найдем те условия, о которых в ней идет речь.

§ 4. Начальные понятия теории алгоритмов и их применения

Условия, при которых не существует полная непротиворечивая дедуктика, легко формулируются в терминах теории алгоритмов [1]).

Нам достаточно на первых порах лишь самых общих интуитивных представлений об *алгоритме* как о предписании, позволяющем по каждому *исходному данному*, или *аргументу*, из некоторой совокупности *возможных* (для данного алгоритма) *исходных данных* (аргументов) получить *результат* в случае, если таковой существует, или не получить ничего в случае, если для рассматриваемого исходного данного не существует результата. Если для выбранного исходного данного результат существует, говорят, что алгоритм *применим* к этому исходному данному и *перерабатывает* его в этот результат.

Для наших целей будет достаточным, и это позволит избежать лишних обсуждений, считать, что исходные данные и результаты любого алгоритма суть слова. Более точно: для каждого алгоритма можно указать некоторый *алфавит исходных данных*, так что все возможные исходные данные являются словами в этом алфавите, и некоторый *алфавит результатов*, так что все результаты являются словами в этом алфавите. Поэтому, чтобы иметь дело с алгоритмами, применяемыми, скажем, к парам слов или к цепочкам слов, мы должны предварительно записать эти образования в виде слов в какомнибудь алфавите. Для определенности условимся соотносить с каждым алфавитом Б некоторую не входящую в него букву и обозначать эту букву звездочкой (подчеркнем, что, таким образом, эта звездочка в различных ситуациях обозначает различные буквы). Первоначальный алфавит Б, пополненный этой новой буквой, будем обозначать $Б_*$. В п. 3.4 предыдущего параграфа мы уже договорились записывать цепочку $\langle C_1, \ldots, C_n\rangle$ слов в алфавите Б посредством слова $C_1 * \ldots * C_n$ в алфавите $Б_*$; в частности, в том же $Б_*$ запишется в виде слова $C_1 * C_2$ и пара $\langle C_1, C_2\rangle$. Пусть, далее,— при фиксированном n — $Б_1, Б_2, \ldots, Б_n$ суть произвольные алфавиты; обозначая по-прежнему через $*$ дополнительную букву, соотнесенную с алфавитом $(Б_1 \cup \ldots \cup Б_n)$ — и тем самым заведомо не входящую ни в один из $Б_i$ — мы будем отождествлять цепочку $\langle C_1, \ldots, C_n\rangle$, где каждое C_i является словом в $Б_i$, со словом $C_1 * \ldots * C_n$ в алфавите $(Б_1 \cup \ldots \cup Б_n)_*$; совокупность всех таких цепочек (и отождествленных с ними слов) будем обозначать через $Б_1^\infty \times \ldots \times Б_n^\infty$.

Совокупность всех исходных данных, к которым алгоритм применим, называется *областью применимости* алгоритма; каждый алгоритм задает

[1]) Систематическое изложение этой теории читатель может найти в монографиях А. А. Маркова [9] и А. И. Мальцева [6]; см. также статью «Алгоритм» в Большой Советской Энциклопедии.

функцию, относящую каждому элементу области применимости соответствующий результат; область определения этой функции совпадает, таким образом, с областью применимости алгоритма; говорят, что рассматриваемый алгоритм *вычисляет* функцию, задаваемую указанным способом. Условимся обозначать через $\mathfrak{A}(x)$ результат применения алгоритма $\mathfrak{A}$ к объекту x [при этом $\mathfrak{A}(\langle x_1, x_2, \ldots, x_n \rangle)$ для краткости будем записывать просто как $\mathfrak{A}(x_1, \ldots, x_n)$]. Тогда определение термина «вычисляет» можно переформулировать следующим образом: алгоритм $\mathfrak{A}$ вычисляет функцию f, коль скоро $\mathfrak{A}(x) \simeq f(x)$ для всех x. (Знак $\simeq$ есть знак «условного равенства»: утверждение $A \simeq B$ считается истинным в двух случаях: либо когда выражения A и B оба не определены либо когда A и B оба определены и обозначают одно и то же.)

Функция, которая вычисляется некоторым алгоритмом, называется *вычислимой*. В части 3° определения понятия доказательства (п. 3.2 предыдущего параграфа) говорится, следовательно, о том, что функция выделения доказанного должна быть вычислимой функцией.

В силу сделанных предположений относительно понятия алгоритма для каждой вычислимой функции можно указать такие два алфавита, что все ее аргументы суть слова в первом из этих алфавитов, а все ее значения — слова во втором из этих алфавитов.

Особый интерес представляют функции, аргументы и значения которых суть натуральные числа. Такие функции условимся называть *числовыми*. Чтобы иметь право говорить о вычислимых числовых функциях, мы должны ввести в рассмотрение алгоритмы, имеющие дело с числами, а для этого прежде всего необходимо представить числа в виде слов в каком-либо алфавите, называемом в этом случае *цифровым*. Возможны различные способы такого представления, например: 1) двоичная запись чисел в алфавите {0, 1}; 2) десятичная запись чисел в алфавите {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}; 3) запись чисел в однобуквенном алфавите { | }, причем число n записывается словом $\underbrace{| | \ldots |}_{n \text{ раз}}$; 4) запись чисел в трехбуквенном алфавите {|, (,)}, причем число n записывается словом $(\underbrace{| | \ldots |}_{n \text{ раз}})$, и т. д. Для тех или иных целей выбираются наиболее удобные способы записи. Каждая запись числа (в какой-либо фиксированной системе) называется *цифрой*. Допуская вольность речи, говорят об алгоритмах и вычислимых функциях, оперирующих с числами, имея в виду алгоритмы и вычислимые функции, оперирующие с изображающими эти числа цифрами (в какой-либо выбранной системе записи).

Понятие вычислимой числовой функции, таким образом, выглядит зависящим от принятого способа записи чисел. Однако легко обнаружить, что всякая числовая функция, вычислимая при одной системе записи, будет вычислима и при другой, по крайней мере для широкого класса таких систем. Назовем две системы эквивалентными, если существуют как алгоритм, дающий по записи произвольного числа в первой системе запись этого же числа во второй системе, так и алгоритм, дающий по записи произвольного

числа во второй системе запись этого же числа в первой системе. Приведенные выше примеры систем записи очевидным образом эквивалентны. Покажем, что числовая функция f, вычислимая при одной из двух эквивалентных систем записи, вычислима и при другой системе. Пусть $\mathfrak{C}$ и $\mathfrak{D}$ — алгоритмы перехода от первой системы ко второй и обратно и пусть алгоритм $\mathfrak{A}$ вычисляет функцию f при первой системе записи (т. е. $\mathfrak{A}$ вычисляет функцию на цифрах первой системы, индуцированную функцией f). Тогда следующий алгоритм $\mathfrak{B}$ будет вычислять f при второй системе записи (т. е. вычислять индуцированную функцию на цифрах второй системы):

$$\mathfrak{B}(x) \simeq \mathfrak{C}(\mathfrak{A}(\mathfrak{D}(x))).$$

Аналогичным образом не зависит от выбора системы записи чисел — для случая числовых множеств — вводимое ниже понятие перечислимого множества.

Ввиду сказанного мы позволим себе, ко[illegible]коро фиксирована какая-либо система записи чисел, не слишком до[illegible]ти различать числа и цифры; множество и тех и других будем называ[illegible]льным рядом и обозначать буквой N.

Множество называется *перечислимым*, если оно либо пусто либо является множеством элементов какой-нибудь вычислимой последовательности (т. е. множеством значений какой-нибудь вычислимой функции, определенной на натуральном ряду); о такой функции (последовательность) говорят, что она *перечисляет* рассматриваемое множество. Очевидно, каждое перечислимое множество словарно.

Пример 1. Множество N^2 всевозможных пар натуральных чисел перечислимо: одна из перечисляющих функций — функция $\varphi(n) = \langle a, b \rangle$, где $n = 2^a(2b + 1) - 1$.

Пример 2. Множество $Ж^\infty$ всех слов в произвольном алфавите $Ж$ перечислимо. Один из возможных способов построения перечислимой последовательности таков: упорядочиваем произвольным образом элементы $Ж$; затем слова в $Ж$ упорядочиваем следующим образом: из слов разной длины предшествующим считается то, которое короче, а на словах одинаковой длины вводим словарный порядок (т. е. слово $\xi_1 \ldots \xi_n$ считаем предшествующим слову $\eta_1 \ldots \eta_n$, если ξ_i совпадает с η_i при $i = 1, 2, \ldots, m$ и буква ξ_{m+1} предшествует букве η_{m+1}). Выписывая слова в порядке следования друг за другом, получаем требуемую перечислимую последовательность. (Может возникнуть вопрос, почему она вычислима, т. е. почему существует алгоритм, дающий по k член a_k этой последовательности с номером k; искомый алгоритм, например, таков: выписывай члены последовательности, пока их не станет $k + 1$; последний из выписанных членов и будет a_k [1].)

Пример 3. Вычислимая функция f, перечисляющая $Ж^\infty$ и построенная в примере 2, осуществляет взаимно однозначное отображение N на $Ж^\infty$; поэтому можно говорить об обратной функции f^{-1}, осуществляющей взаимно однозначное отображение $Ж^\infty$ на N. Эта f^{-1} тоже вычислима, поскольку

[1] Напомним, что мы считаем нуль натуральным числом, и потому начинаем с $k = 0$.

вычисляется следующим алгоритмом: чтобы вычислить $f^{-1}(a)$, вычисляй последовательно $f(0)$, $f(1)$, $f(2)$, . . . и т. д., пока для некоторого n не получишь $f(n) = a$; это n и есть $f^{-1}(a)$.

Пример 4. Для любых алфавитов $Ж_1$ и $Ж_2$ композиция вычислимых функций, взаимно однозначно отображающих N на $Ж_2^\infty$ (пример 2) и $Ж_1^\infty$ на N (пример 3) дает вычислимую же функцию, взаимно однозначно отображающую $Ж_1^\infty$ на $Ж_2^\infty$.

Подмножество S множества A называется *разрешимым* относительно A, коль скоро существует такой алгоритм (*разрешающий* S относительно A), который распознает принадлежность элементов A к S, т. е. такой алгоритм, который все элементы из S перерабатывает в некоторое одно и то же слово x (например, в слово «да»), а все элементы из $A \setminus S$ в некоторое одно и то же, но отличное от x слово y (например, в слово «нет»; разумеется, выбор слов x и y совершенно несуществен). Очевидно, разрешимость множества S относительно A равносильна разрешимости множества $A \setminus S$ относительно того же A. В части 2° определения понятия доказательства требовалось, чтобы множество всех доказательств было разрешимо относительно множества всех слов в алфавите доказательств, а в п. 3.4 — чтобы каждое S_μ было разрешимо относительно соответствующего X^μ.

Из определения разрешимости вытекает, что область применимости алгоритма, разрешающего S относительно A, объемлет A. Возникает естественный вопрос, что произойдет, если предложить другое, более узкое определение разрешимости, потребовав, чтобы разрешающий алгоритм был применим *только* к элементам множества A? При таком определении разрешимость S относительно A равносильна, очевидно, вычислимости характеристической функции множества S относительно A (т. е. определенной на A функции, равной 1 на S и 0 на $A \setminus S$). Как будет показано в § 8 (следствие 1 аксиомы протокола), область применимости любого алгоритма всегда есть перечислимое множество, и потому лишь перечислимые множества могут обладать разрешимыми — в смысле нового, узкого определения — подмножествами. Если же множество A перечислимо, то для него оба определения разрешимого подмножества приводят к одинаковым результатам. В самом деле, пусть вычислимая функция f перечисляет A, а алгоритм $\mathfrak{B}$ разрешает S относительно A в прежнем, широком смысле. Тогда следующий алгоритм будет также разрешать S относительно A и притом иметь A своею областью применимости: бери произвольное a и вычисляй последовательно $f(0)$, $f(1)$, $f(2)$, . . .; как только получишь $f(n) = a$, применяй к a алгоритм $\mathfrak{B}$.

Замечание 1. Поскольку каждая вычислимая функция, каждое перечислимое множество и каждое разрешимое подмножество задается некоторым алгоритмом, существование функций, множеств и подмножеств, не являющихся соответственно вычислимыми, перечислимыми или разрешимыми, усматривается из количественных соображений. Действительно, каждый алгоритм может быть записан в конечном счете на русском языке (с добавлением, если надо, необходимых математических символов), т. е., согласно п. 1.1 из предыдущего параграфа, в виде слова в некотором достаточно обширном алфавите, а всех слов в произвольном алфавите — счетное множество.

Конечно, от такого рассуждения еще далеко до построения индивидуальных примеров неалгоритмических объектов.

Приложим теперь описанные только что понятия теории алгоритмов к исследованию возможности существования полной непротиворечивой дедуктики.

Лемма 1. *Каково бы ни было словарное множество X, множества $\varnothing$ и X разрешимы относительно X.*

Доказательство. Пусть X словарно в Ж. Достаточно взять алгоритм, который каждому слову из $Ж^\infty$ ставит в соответствие некоторое одно и то же слово x. Этот алгоритм будет разрешать каждое из множеств $\varnothing$ и X относительно X.

Теорема 1. *Если T — перечислимое множество, то для фундаментальной пары $\langle Б, T\rangle$ можно ввести полную непротиворечивую дедуктику.*

Доказательство. Требуется задать тройку $\langle Д, D, \delta\rangle$. Замечаем, что $\varnothing$ и $Д^\infty$ разрешимы (относительно $Д^\infty$) по лемме 1. Если $T = \varnothing$, то берем $\langle Д, \varnothing, \delta\rangle$, где Д и δ — любые. Если $T \neq \varnothing$, то $T = \{\tau(0), \tau(1), \tau(2), \ldots\}$, где τ — вычислимая функция; отождествим число n со словом $|\,|\ldots|$ длины n и положим $Д = \{\,|\,\}$, $D = Д^\infty$, $\delta = \tau$.

Замечание 2. Это доказательство не такое искусственное, как может показаться на первый взгляд. В самом деле, если множество истин некоторого языка перечислимо, т. е. может быть расположено в вычислимую последовательность, то для того, чтобы убедиться в принадлежности какого-либо выражения к этому множеству (т. е. доказать истинность рассматриваемого выражения), достаточно указать номер этого выражения в этой последовательности (каковой номер поэтому и можно считать доказательством).

Лемма 2 (о перечислимости разрешимого подмножества). *Разрешимое подмножество перечислимого множества перечислимо.*

Доказательство. Пусть $S \subseteq A$, причем A перечисляется вычислимой функцией f. Если S пусто, то S перечислимо по определению. Если S непусто, то существует такое s, что $s \in S$. Положим

$$g(n) = \begin{cases} f(n), & \text{если } f(n) \in S, \\ s, & \text{если } f(n) \in S \setminus A. \end{cases}$$

Очевидно, g есть вычислимая функция, перечисляющая множество S.

Теорема 2. *Множество всех доказательств (для данной дедуктики) перечислимо.*

Доказательство. Множество всех слов в алфавите доказательств перечислимо (см. пример 2). Поэтому достаточно применить лемму 2.

Лемма 3 (об образе перечислимого множества). *Пусть R перечислимо и f — вычислимая функция, определенная на всех элементах множества R. Тогда $f(R)$ перечислимо.*

Доказательство. Если R перечисляется вычислимой функцией ρ, то $f(R)$ перечисляется вычислимой функцией $y = f(\rho(x))$.

Пример 5. Пусть $\neg \in Л$, $A \subseteq Л^\infty$. Обозначим через $\neg A$ множество всех слов вида $\neg a$, где $a \in A$. Полагая в лемме 3 $R = A$, $f(a) = \neg a$, получаем, что из перечислимости A вытекает перечислимость $\neg A$; полагая

$R = \neg A$, $f(\neg a) = a$ получаем, что из перечислимости $\neg A$ вытекает перечислимость A.

Пример 6. Множество $И^\infty \times И^\infty$ перечислимо. В самом деле, множества N^2 и $И^\infty$ перечислимы (примеры 1 и 2). Пусть $И^\infty$ перечисляется вычислимой последовательностью g. Определим на N^2 вычислимую функцию f, полагая $f(a, b) = \langle g(a), g(b)\rangle$. Очевидно, $f(N^2) = И^\infty \times И^\infty$ и остается применить лемму 3.

Как обычно, через $K_1 \times K_2 \times \ldots \times K_n$ обозначается прямое произведение множеств $K_1, \ldots, K_n$, т. е. множество всех таких цепочек $\langle k_1, \ldots, k_n\rangle$, что $k_1 \in K_1, \ldots, k_n \in K_n$. В силу соглашений, сделанных в начале параграфа, в случае, если $K_1 \subseteq Б_1^\infty, \ldots, K_n \subseteq Б_n^\infty$, где $Б_1, \ldots, Б_n$ — алфавиты, прямое произведение $K_1 \times \ldots \times K_n$ отождествляется с некоторым множеством слов из $Б_1^\infty \times \ldots \times Б_n^\infty$.

Следствие 1 леммы 3. *Если $K_1, \ldots, K_n$ суть перечислимые множества, то их прямое произведение $K_1 \times \ldots \times K_n$ также перечислимо.*

Доказательство. Для $n = 2$ — как в примере 6. Далее — по индукции, применяя лемму 3 к «естественному» вычислимому отображению множества $(K_1 \times \ldots \times K_s) \times K_{s+1}$ на множество $K_1 \times \ldots \times K_s \times K_{s+1}$.

Цепочка $\langle C_{i_1}, \ldots, C_{i_r}\rangle$, где $i_1 \leqslant n, \ldots, i_r \leqslant n$, называется *проекцией* цепочки $\langle C_1, \ldots, C_n\rangle$ на оси $i_1, \ldots, i_r$ и обозначается $\text{пр}_{i_1, \ldots, i_r}\langle C_1, \ldots, C_n\rangle$. В частности, $\text{пр}_1\langle C_1, \ldots, C_n\rangle = C_1$, $\text{пр}_2\langle C_1, \ldots, C_n\rangle = C_2$. Если $M \subseteq K_1 \times \ldots \times K_n$, то через $\text{пр}_{i_1, \ldots, i_r}M$ обозначается множество всевозможных проекций $\text{пр}_{i_1, \ldots, i_r}m$, где $m \in M$.

Следствие 2 леммы 3. *Если M — перечислимое подмножество множества $Б_1^\infty \times \ldots \times Б_n^\infty$, где $Б_1, \ldots, Б_n$ — некоторые алфавиты, а $i_1, \ldots, i_r$ — числа, не превосходящие n, то $\text{пр}_{i_1, \ldots, i_r}M$ перечислимо.*

Доказательство. Достаточно рассмотреть вычислимую функцию $x \to \text{пр}_{i_1, \ldots, i_r}x$.

Теорема 3. *Множество всех доказуемых слов (для данной дедуктики) перечислимо.*

Доказательство. Пусть P — множество всех доказуемых слов для дедуктики $\langle Д, D, \delta\rangle$. Очевидно, что $P = \delta(D)$. По теореме 2 множество D перечислимо. Остается применить лемму 3.

Таким образом, если T неперечислимо, то для пары $\langle Б, T\rangle$ невозможно ввести полную непротиворечивую дедуктику; для всякой непротиворечивой дедуктики множество доказуемых слов P будет собственным подмножеством множества T и в разности $T \setminus P$ всегда найдется элемент; этот элемент будет истинным, но не доказуемым утверждением!

Теоремы 1 и 3 в совокупности дают условия, налагаемые на фундаментальную пару и необходимые и достаточные для того, чтобы для этой пары можно было ввести полную непротиворечивую дедуктику. Это условие — перечислимость множество всех истин. Можно ожидать (и так и оказывается на самом деле), что в «богатых», «выразительных» языках множества всех истин настолько сложны, что неперечислимы, и потому для этих языков невозможны полные непротиворечивые дедуктики. Найденный критерий,

однако, не слишком удобен, поскольку рассмотрение множества T всех истин может оказаться затруднительным. В § 6 поэтому мы переформулируем этот критерий, сделав его более «применимым». Однако сперва в § 5 мы рассмотрим ситуации столь простые, что применение нашего критерия в его настоящей форме оказывается еще уместным.

§ 5. Языки, связанные с ассоциативными исчислениями

В этом параграфе мы рассмотрим примеры языков с относительно просто устроенными множествами истинных утверждений. Эти примеры будут связаны с так называемыми ассоциативными исчислениями.

Ассоциативным исчислением в алфавите И называется, согласно А. А. Маркову [9], произвольная конечная совокупность правил, разрешающих определенного вида преобразования слов в И. Эти правила называются двусторонними подстановками или (коль скоро мы не рассматриваем здесь односторонних подстановок) просто *подстановками* в алфавите И. Каждая подстановка в алфавите И записывается в виде

$$P \leftrightarrow Q,$$

где P и Q суть слова в И, а буква $\leftrightarrow$ не принадлежит алфавиту И. (Например, ци $\leftrightarrow$ цы есть подстановка в русском алфавите.) Подстановка $P \leftrightarrow Q$ означает разрешение заменять слово P, если оно встретится как часть другого слова, на слово Q, и обратно. Сказанное оформляется более точно в виде следующих определений. Для каждого ассоциативного исчисления (т. е. для каждого списка подстановок) вводится понятие смежных слов и эквивалентных слов. Два слова A и B называются *смежными* (записывается $A \perp B$), коль скоро существуют такие слова P, Q, X, Y, что: 1) $A = XPY$, 2) $B = XQY$ и 3) хотя бы одна из подстановок $P \leftrightarrow Q$ и $Q \leftrightarrow P$ есть подстановка рассматриваемого исчисления. Цепочку $\langle C_1, \ldots, C_n \rangle$ слов из И$^\infty$ назовем цепочкой смежности, если для каждого i имеет место $C_i \perp C_{i+1}$. Два слова, A и B, называются *эквивалентными*, коль скоро существует такая цепочка смежности C_1, C_2, $\ldots$, C_n, что $C_1 = A$, $C_n = B$.

Замечание 1. Если произвести факторизацию множества И$^\infty$ по так введенному отношению эквивалентности, получится алгебраическая система с ассоциативной операцией (возникающей при факторизации из операции приписывания друг к другу слов); отсюда и название — ассоциативное исчисление.

Пусть фиксировано некоторое ассоциативное исчисление в алфавите И. Существует алгоритм, позволяющий для каждых двух слов A и B из И$^\infty$ распознавать, смежны они или нет. Такой алгоритм состоит, например, в переборе всех четверок слов P, Q, X, Y, длина которых не превосходит длин A и B, и проверке условий 1), 2), 3). Таким образом, множество всех пар смежных слов разрешимо относительно И$^\infty \times$ И$^\infty$. Однако существование алгоритма, распознающего эквивалентность слов, очевидно лишь в простейших случаях.

Пример 1. Пусть И $=\{a, b, c\}$ и ассоциативное исчисление задано следующими подстановками:

$$ab \leftrightarrow ba,$$
$$ac \leftrightarrow ca,$$
$$bc \leftrightarrow cb.$$

Очевидно, что A и B эквивалентны тогда и только тогда, когда число букв a в слове A равно числу букв a в слове B, и то же самое выполняется для букв b и c. Такое исчисление естественно назвать *коммутативным*.

В общем случае не ясно, каким алгоритмом можно было бы обнаружить для произвольных слов, эквивалентны они или нет, т. е. имеется ли связывающая их цепочка смежных слов. И действительно, как показали А. А. Марков [7] и Э. Л. Пост [13], возможно ассоциативное исчисление с неразрешимой проблемой распознавания эквивалентности (под проблемой распознавания эквивалентности как раз и понимается проблема отыскания алгоритма, распознающего эквивалентность слов). Доказательство существования таких исчислений приводится, например, в монографии С. К. Клини [4]. Мы здесь приведем, без доказательства, следующий пример, принадлежащий Г. С. Цейтину.

Пример 2. Пусть И $=\{a, b, c, d, e\}$ и ассоциативное исчисление задано подстановками

$$ac \leftrightarrow ca,$$
$$ad \leftrightarrow da,$$
$$bc \leftrightarrow cb,$$
$$bd \leftrightarrow db,$$
$$eca \leftrightarrow ce,$$
$$edb \leftrightarrow de,$$
$$cca \leftrightarrow ccae.$$

Как показал Г. С. Цейтин [19], для этого исчисления не существует алгоритма, распознающего эквивалентность слов.

Ассоциативное исчисление будем называть *разрешимым*, если для него существует алгоритм распознавания эквивалентности; в противном случае будем называть его *неразрешимым*. Очевидно, что разрешимость ассоциативного исчисления равносильна разрешимости множества всех пар эквивалентных слов (и разрешимости множества всех пар неэквивалентных слов) относительно $И^\infty \times И^\infty$.

Фиксируем некоторое ассоциативное исчисление $\mathfrak{Z}$ в алфавите И. Множество всех слов (в алфавите $И_*$) вида $A * B$, где $A \in И^\infty$, $B \in И^\infty$ и A эквивалентно (соответственно, неэквивалентно) B, обозначим через T^+ (соответственно, через T^-), так что $T^+ \cup T^- = И^\infty \times И^\infty$. Тогда разрешимость исчисления $\mathfrak{Z}$ означает разрешимость множества T^+ (что равносильно разрешимости множества T^-) относительно $И^\infty \times И^\infty$.

Замечание 2. Поэтому множество T^+ (как и T^-), построенное для исчисления из примера 2, представляет собой индивидуальный пример неразрешимого подмножества множества $И^\infty \times И^\infty$. Характеристическая функция этого подмножества представляет собой в этом случае индивидуальный пример невычислимой функции.

С каждым ассоциативным исчислением в алфавите И мы сопряжем теперь два языка — *позитивный язык*, утверждения которого будут утверждениями об эквивалентности произвольных двух слов в И, и *негативный язык*, утверждениями которого будут утверждения о неэквивалентности произвольных двух слов в И. В обоих случаях в качестве утверждений целесообразно рассматривать элементы множества $И^\infty \times И^\infty$; только в первом случае, для позитивного языка, слово $A * B$ будет интерпретироваться как утверждение об эквивалентности слов A и B, и потому множеством истинных утверждений будет служить T^+, тогда как во втором случае, для негативного языка, слово $A * B$ будет интерпретироваться как утверждение о неэквивалентности слов A и B, и потому множеством истинных утверждений будет служить T^-. Вспомним теперь, что в п. 1.3 § 3 мы договорились считать язык заданным, коль скоро указана соответствующая фундаментальная пара. Итак, пусть фиксирован алфавит И и ассоциативное исчисление $\mathfrak{Z}$ в этом алфавите, Мы объявляем $\langle И_*, T^+\rangle$ фундаментальной парой позитивного языка, а $\langle И_*, T^-\rangle$ — фундаментальной парой негативного языка, сопряженных с исчислением $\mathfrak{Z}$.

Нас будет занимать вопрос о возможности ввести полную непротиворечивую дедуктику для $\langle И_*, T^+\rangle$ и для $\langle И_*, T^-\rangle$. Мы увидим, что в первом случае этот вопрос решается всегда положительно, а во втором — в зависимости от разрешимости исчисления $\mathfrak{Z}$.

Л е м м а 4. *Множество E всех цепочек смежности разрешимо относительно* $И_*^\infty$.

Доказательство вытекает из существования алгоритма, распознающего смежность любых двух слов из $И^\infty$.

Т е о р е м а 4. *Для любого ассоциативного исчисления множество всех пар эквивалентных слов перечислимо.*

Д о к а з а т е л ь с т в о. Введем на $И_*^\infty$ функцию φ, полагая $\varphi(C_1*C_2*\ldots*C_n) = C_1*C_n$ для каждого слова $C_1*C_2*\ldots*C_n$, где все C_i суть слова из $И^\infty$. Очевидно, что A и B эквивалентны тогда и только тогда, когда $A*B = \varphi(C)$ для некоторой цепочки смежности C. Поэтому $T^+ = \varphi(E)$, где E — множество всех цепочек смежности. Множество E разрешимо относительно $И_*^\infty$ (по лемме 4) и, следовательно, перечислимо (по лемме 2). Функция φ очевидным образом вычислима, а потому перечислимым будет и множество $\varphi(E)$. Но $\varphi(E) = T^+$, а перечислимость T^+ и надо было доказать.

З а м е ч а н и е 3. Таким образом, в случае неразрешимости исчисления, T^+ будет служить примером перечислимого, но не разрешимого подмножества перечислимого множества $И^\infty \times И^\infty$. В силу леммы 5 (см. ниже) всякий такой пример является одновременно примером перечислимого подмножества с неперечислимым дополнением. Ср. ниже замечание 5.

С л е д с т в и е т е о р е м ы 4. *Для (фундаментальной пары) позитивного языка, сопряженного с произвольным ассоциативным исчислением, можно ввести полную непротиворечивую дедуктику.*

З а м е ч а н и е 4. Чтобы получить дедуктику, о которой говорится в этом следствии, нет нужды обращаться к теореме 1. Проще предъявить дедуктику $\langle И_*, E, \varphi\rangle$, где E и φ таковы, как в доказательстве теоремы 4;

она и будет полной непротиворечивой дедуктикой для $\langle И_*, T^+\rangle$. Эта дедуктика является совершенно естественной с содержательной точки зрения; в самом деле, лучшим доказательством эквивалентности слов A и B является предъявление связывающей их цепочки смежности.

Перейдем к вопросу о дедуктике для $\langle И_*, T^-\rangle$.

Лемма 5 (об условиях разрешимости перечислимого множества). *Подмножество S перечислимого множества X тогда и только тогда разрешимо относительно X, когда перечислимо как S, так и его дополнение $X \setminus S$.*

Доказательство. Если S разрешимо, то разрешимо и $X \setminus S$ и остается применить лемму 2 о перечислимости разрешимого множества. Пусть теперь X и $X \setminus S$ оба перечислимы. Если хотя бы одно из них пусто, то, по лемме 1, множество S разрешимо. Если оба они непусты, то, стало быть, перечисляются некоторыми вычислимыми функциями f и g. Тогда, чтобы ответить на вопрос «$x \in S$?», поставленный для произвольного x из X, достаточно вычислять последовательно

$$f(0),\ g(0),\ f(1),\ g(1),\ \ldots \tag{1}$$

до тех пор, пока не встретится x (что произойдет непременно, так как образующаяся последовательность исчерпывает все X). Если при этом окажется, что x встретилось среди значений f, то $x \in S$; если же x встретилось среди значений g, то $x \overline{\in} S$.

Теорема 5. *Пусть дано ассоциативное исчисление. Множество всех пар неэквивалентных слов тогда и только тогда перечислимо, когда это исчисление разрешимо.*

Доказательство. Заметим прежде всего, что $И^\infty \times И^\infty$ перечислимо (пример 6 из § 4). Пусть исчисление разрешимо; тогда T^- разрешимо относительно $И^\infty \times И^\infty$, а значит, само перечислимо (по лемме 2). Пусть теперь T^- перечислимо; поскольку его дополнение T^+ до перечислимого множества также перечислимо (по теореме 4), то в силу леммы 5 множество T^- разрешимо (относительно $И^\infty \times И^\infty$), а вместе с ним разрешимо и само рассматриваемое исчисление.

Замечание 5. Множество T^+, таким образом, в случае неразрешимости исчисления, служит примером перечислимого множества с неперечислимым дополнением (до некоторого объемлющего перечислимого множества); в силу леммы 5 всякий такой пример служит одновременно примером перечислимого множества, не являющегося разрешимым (опять-таки относительно некоторого перечислимого надмножества). Существование таких множеств является одним из центральных фактов теории алгоритмов и играет решающую роль при исследовании большинства неразрешимых алгоритмических проблем. В частности, обычные доказательства неразрешимости ассоциативных исчислений (в том числе исчисления из примера 2) как раз и опираются на этот факт, который, следовательно, подлежит сам доказательству, не опирающемуся на существование неразрешимых ассоциативных исчислений. В § 7 мы еще вернемся к этому вопросу.

Следствие теоремы 5. *Для фундаментальной пары негативного языка, сопряженного с некоторым ассоциативным исчислением, тогда*

и только тогда можно ввести полную непротиворечивую дедуктику, когда это исчисление разрешимо.

Введем теперь для произвольного ассоциативного исчисления $\mathfrak{I}$ в алфавите И *универсальный язык*, утверждениями которого будут служить как утверждения об эквивалентности слов, так и утверждения о неэквивалентности. Здесь нам придется отличать первые утверждения от вторых. С этой целью пополним алфавит И еще одной буквой, буквой $\neg$, в предположении, что она, как и $\leftrightarrow$ и $*$, не входит в И. АлфавитИ $\cup \langle *, \neg \rangle$ обозначим через Л. Обозначим через $\neg T^-$ множество всех слов вида $\neg P$, где $P \in T^-$. Положим $T^0 = T^+ \cup \neg T^-$ и образуем фундаментальную пару $\langle$Л, $T^0\rangle$. Элемент t из T^0 естественно интерпретировать как истинное утверждение об эквивалентности слов (если $t \in T^+$) или о неэквивалентности слов (если $t \in \neg T^-$).

Лемма 6. *Теоретико-множественные сумма и пересечение перечислимых множеств перечислимы.*

Доказательство. Пусть R и S — перечислимые множества. Сначала докажем перечислимость $R \cup S$. Если одно из множеств пусто, то это тривиально. Если оба множества непусты, то $R = \{\rho(0), \rho(1), \ldots\}$, $S = \{\sigma(0), \sigma(1), \ldots\}$, где ρ и σ — вычислимые последовательности. Тогда вычислимая последовательность f, заданная соотношениями $f(2n) = \rho(n)$, $f(2n + 1) = \sigma(n)$, будет перечислять $R \cup S$. Докажем теперь перечислимость $R \cap S$. Если $R \cap S$ пусто, то оно перечислимо по определению. В противном случае существует некоторое a, такое, что $a \in R \cap S$, а R и S перечисляются вычислимыми функциями ρ и σ. Поскольку N^2 перечислимо (пример 1 из § 4), оно перечисляется некоторой вычислимой функцией g. Каждое значение $g(n)$ есть некоторая пара натуральных чисел; обозначим через $\xi(n)$ и $\eta(n)$ первый и второй члены этой пары. Функции ξ и η, очевидно, вычислимы. Введем функцию h:

$$h(n) = \begin{cases} \rho(\xi(n)), & \text{если } \rho(\xi(n)) = \sigma(\eta(n)), \\ a & \text{в противном случае.} \end{cases}$$

Функция h вычислима и перечисляет множество $R \cap S$.

Теорема 6. *Для любого ассоциативного исчисления $\mathfrak{I}$ соответствующее ему множество T^0 тогда и только тогда перечислимо, когда исчисление разрешимо.*

Доказательство. Если $\mathfrak{I}$ разрешимо, то T^- перечислимо (теорема 5), а потому перечислимо и $\neg T^-$ (пример 5 из § 4). Тогда T^0 перечислимо по лемме 6. Пусть теперь перечислимо T^0. Образуем множество $\neg$Л$^\infty$ всех слов в алфавите Л, начинающихся с $\neg$; это множество перечислимо (примеры 2 и 5 из § 4). По лемме 6 перечислимо пересечение $T^0 \cap \neg$Л$^\infty$. Но $T^0 \cap \neg$Л$^\infty = \neg T^-$. Поэтому перечислимо $\neg T^-$, а вместе с ним и T^- (пример 5 из § 4); но тогда, по теореме 5, разрешимо исчисление $\mathfrak{I}$.

Следствие. *Для фундаментальной пары универсального языка, сопряженного с некоторым ассоциативным исчислением, тогда и только тогда можно ввести полную непротиворечивую дедуктику, когда это исчисление разрешимо.*

§ 6. Простейшие критерии неполноты

Вернемся к нашей общей схеме. Мы знаем теперь, что неперечислимость множества T необходима и достаточна для того, чтобы для $\langle$Б, $T\rangle$ не существовало полной и непротиворечивой дедуктики. Однако, как отмечалось в конце § 4, само рассмотрение множества T может вызвать известные затруднения.

Речь идет не столько о неудобствах практического характера (заключающихся, например, в том, что T может оказаться слишком сложно устроенным), сколько о трудностях характера принципиального. Не говоря уже о том, что в ряде важных случаев «множество истин» в полном объеме вообще совершенно не определенно — как, например, в случае теории множеств, почти всегда сами истины могут быть как менее, так и более убедительными или наглядными [1]). Множество T, таким образом, можно представлять себе расслаивающимся на подмножества истинных утверждений, обладающих разной степенью достоверности (все эти рассуждения, конечно, носят совершенно неформальный характер). Естественно стремиться к тому, чтобы доказать хотя бы все относительно простые истины или, наоборот, получить недоказуемое утверждение не просто из T, но из заданного его подмножества — подмножества «более простых» истин. Само это подмножество «более простых истин» естественно рассматривать как пересечение T с множеством «более простых утверждений», каковое является некоторым подмножеством множества $Б^{\infty}$. Сказанное оправдывает рассмотрение понятий непротиворечивости и полноты в применении к произвольному подмножеству множества $Б^{\infty}$. Перейдем к формальным определениям.

Пусть $\langle$Б, $T\rangle$ — фундаментальная пара, $\langle$Д, D, $\delta\rangle$ — дедуктика над Б, и P — множество всех доказуемых слов. Пусть $V \subseteq Б^{\infty}$. Скажем, что дедуктика $\langle$Д, D, $\delta\rangle$

а) *непротиворечива применительно к* V, если $V \cap P \subseteq V \cap T$;

б) *полна применительно к* V, если $V \cap T \subseteq V \cap P$.

Т е о р е м а 7. *Если* V *— перечислимое подмножество множества* $Б^{\infty}$, *а множество истинных утверждений, принадлежащих к* V, *неперечислимо, то никакая дедуктика не является одновременно непротиворечивой и полной применительно к* V.

Д о к а з а т е л ь с т в о. По условию, $V \cap T$ неперечислимо. Для непротиворечивой и полной дедуктики $V \cap T = V \cap P$. Но $V \cap P$ обязано быть перечислимым, как это вытекает из теоремы 3 и леммы 6.

З а м е ч а н и е 1. Условие несуществования дедуктики с определенными свойствами, сформулированное в теореме 7, является не только достаточным, но и необходимым (причем даже без предположения о перечислимости V). В самом деле, если $V \cap T$ перечислимо, то полная непротиворечивая дедуктика для $\langle$Б, $V \cap T\rangle$, существующая в силу теоремы 1, будет в то же время полной и непротиворечивой применительно к V.

---

[1]) Мы говорим ведь «верно, как дважды два четыре», а не «верно, как $17^{14} > 31^{11}$» или «верно как $300! > 100^{300}$».

Очевидно, что дедуктика непротиворечива (полна) относительно $\langle Б, T\rangle$ тогда и только тогда, когда она непротиворечива (полна) применительно к любому подмножеству множества $Б^{\infty}$. Поэтому для обнаружения неполноты относительно $\langle Б, T\rangle$ непротиворечивой (относительно той же $\langle Б, T\rangle$) дедуктики, достаточно (и необходимо) найти такое подмножество V множества $Б^{\infty}$, применительно к которому эта дедуктика неполна. Следующее построение помогает найти в ряде важных случаев такое подмножество.

Условимся говорить, что посредством фундаментальной пары $\langle Б, T\rangle$ *выразима принадлежность* к множеству Q натуральных чисел, если существует такая определенная на натуральном ряду вычислимая функция f (*выражающая* эту принадлежность), что:

1) если $n \in Q$, то $f(n) \in T$,

2) если $n \in N \setminus Q$, то $f(n) \in Б^{\infty} \setminus T$.

Для такой функции f множество V всех ее значений перечислимо. Поэтому (в силу теоремы 7) не будет существовать полной и непротиворечивой применительно к V дедуктики, коль скоро множество $V \cap T$ принадлежащих к V истинных утверждений неперечислимо. Неперечислимость же множества $V \cap T$, как мы сейчас увидим, гарантируется неперечислимостью множества Q.

Лемма 7 (о полном прообразе). *Пусть f — вычислимая функция, область определения которой есть перечислимое множество* [1]*, и B — произвольное перечислимое множество. Тогда множество $f^{-1}(B)$ перечислимо.*

Доказательство. Если $f^{-1}(B)$ пусто, оно перечислимо по определению. Пусть теперь $c \in f^{-1}(B)$, множество B перечисляется вычислимой функцией h, а область определения функции f — вычислимой функцией g. Пусть ξ и η определены, как в доказательстве леммы 6. Положим

$$\varphi(n) = \begin{cases} g(\xi(n)), & \text{если } f(g(\xi(\eta))) = h(\eta(n)), \\ c & \text{в противном случае.} \end{cases}$$

Легко видеть, что φ — вычислимая функция, перечисляющая множество $f^{-1}(B)$.

Вернемся теперь к рассмотрениям, предшествующим формулировке леммы 7. Заметим, что $Q = f^{-1}(V \cap T)$. Поэтому если Q неперечислимо, то неперечислимо и $V \cap T$ (в противном случае в силу леммы 7 было бы перечислимо и Q). Таким образом (принимая во внимание теорему 7) нами доказана следующая

Теорема 8. *Если посредством фундаментальной пары $\langle Б, T\rangle$ выразима принадлежность хотя бы к одному неперечислимому множеству натуральных чисел, для $\langle Б, T\rangle$ не может существовать непротиворечивой и полной дедуктики; более того, не существует дедуктики, являющейся одновременно непротиворечивой и полной применительно к множеству значений функции, выражающей указанную принадлежность.*

---

[1] На самом деле область определения всякой вычислимой функции является перечислимым множеством; однако установление этого факта требует дополнительных уточнений наших представлений об алгоритмах, которые будут произведены лишь в § 8.

Замечание 2. Достаточное условие несуществования, сформулированное в теореме 8, является и необходимым. В самом деле, если для $\langle Б, T\rangle$ нельзя ввести полную непротиворечивую дедуктику, то T неперечислимо (теорема 1); $Б^{\infty}$ перечислимо (пример 2 из § 4) и перечисляется некоторой вычислимой функцией f. Поскольку $T = f(f^{-1}(T))$, то множество $f^{-1}(T)$ неперечислимо (в силу теоремы 3 об образе перечислимого множества). В то же время функция f выражает принадлежность к $f^{-1}(T)$ посредством пары $\langle Б, T\rangle$.

§ 7. Язык арифметики

В этом параграфе мы приложим построения предыдущего параграфа к языку арифметики. Содержательно, под языком арифметики понимается язык, утверждения которого формулируются (с помощью обычных логических операций и отношения равенства) в терминах натуральных чисел и операций сложения и умножения (ср. замечание 2 из § 1). На формальном уровне нам надлежит предъявить соответствующую фундаментальную пару. Разумеется, задача построения такой пары не может иметь однозначного решения: ясно, что возможны различные алфавиты для записи одной и той же сути. Здесь будет избран 14-буквенный алфавит А (арифметический алфавит), буквами которого служат следующие знаки:

1°—2° скобки (и);

3° знак для образования цифр |;

4° знак для образования переменных x;

5°—6° знаки сложения $+$ и умножения $\cdot$;

7° знак равенства $=$;

8°—14° логические знаки $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\exists$, $\forall$ (при содержательной интерпретации эти знаки будут иметь следующих смысл: «неверно, что», «и», «или», «если ..., то», «эквивалентно», «существует такой..., что», «для всех».

Чтобы выделить надлежащее множество истинных утверждений, нам предстоят вначале некоторые рассмотрения синтаксического характера: мы должны будем выделить определенные классы слов в А и заняться их строением.

Слово вида $\underbrace{\alpha\alpha\ldots\alpha}_{n \text{ раз}}$, где α — какая-то буква, будем обозначать через α^n.

Цифрами будем называть слова вида $(|^n)$, где $n \geqslant 0$, а *переменными* — слова вида (x^n), где $n > 0$. При интерпретации языка слово $(|^n)$ будет служить записью числа n, а слово (x^n) будет одной из переменных, пробегающих натуральный ряд (для записи утверждений арифметики может потребоваться сколь угодно много таких переменных). Введем теперь следующее индуктивное определение *терма*:

1° все цифры и все переменные суть термы;

2° если α и β суть термы, то $(\alpha + \beta)$ и $(\alpha\cdot\beta)$ суть термы.

Терм, не содержащий переменных, будем называть *постоянным*.

Пример 1. Терм $((|||)\cdot(||))$ — постоянный, а терм $((|||)\cdot(xx))$ — не постоянный.

Каждому постоянному терму естественно поставить в соответствие некоторое число, его *значение*, по следующему правилу:

1° значением цифры ($|^n$) является число n;

2° значением постоянного терма $(\alpha + \beta)$ служит сумма значений постоянных термов α и β, а значением постоянного терма $(\alpha \cdot \beta)$ служит произведение значений постоянных термов α и β.

Пример 2. Значением постоянного терма $((|||) + ((|) \cdot (||)))$ служит число пять.

Всякое слово вида $(\alpha = \beta)$, где α и β суть термы, будем называть *элементарной формулой*. Наконец, введем следующее индуктивное определение *формулы*:

1° все элементарные формулы суть формулы;

2° если α есть формула, то $\neg\alpha$ есть формула;

3° если α и β суть формулы, то $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \to \beta)$, $(\alpha \leftrightarrow \beta)$ суть формулы;

4° если α есть формула, а ξ есть переменная, то $\exists\xi\alpha$ и $\forall\xi\alpha$ суть формулы.

Пример 3. Слово

$$(\exists(x)\forall(xx)\neg((x)=(xx)) \leftrightarrow \forall(xx)((x) = (xx)))$$

является формулой.

Среди формул и будут выделяться истинные утверждения. Но сначала нам потребуются сравнительно более технические понятия *свободного вхождения* переменной и *подстановки* цифры вместо переменной.

Для этого нам нужно научиться опознавать различные положения (так называемые *вхождения*) одного слова внутри другого слова. Так, слово АРА имеет одно вхождение в слово МАРАТ (со 2-й по 4-ю букву) и два вхождения в слово АРАРАТ (с 1-й по 3-ю букву и с 3-й по 5-ю букву). Вхождения слова P в слово Q полностью определяются своими левыми и правым контекстами, т. е. такими словами X и Y, что $XPY = Q$. Так, у первого из вхождений АРА в АРАРАТ левый и правый контексты суть пустое слово и РАТ, а у второго вхождения соответственно АР и Т. (А. А. Марков [9] определяет вхождение P в Q как такое слово $X*P*Y$, что $XPY = Q$ и называет X и Y левым и правым крыльями вхождения). Рассмотрим некоторое вхождение P в Q и одновременно некоторое вхождение N в P; ясно, что этим *индуцируется* некоторое вхождение N в Q; если X и Y суть левый и правый контексты рассматриваемого вхождения P в Q, а U и V — левый и правый контексты рассматриваемого вхождения N в P, то XU и VY будут левым и правым контекстами индуцированного вхождения N в Q.

Определим теперь индуктивно *связанное* вхождение переменной в формулу следующим образом:

1° каждое вхождение переменной ξ в формулу, начинающуюся со слова $\exists\xi$ или со слова $\forall\xi$, является связанным;

2° пусть α и β суть формулы, ξ и η суть переменные; каждое связанное вхождение ξ в α индуцирует связанное же вхождение ξ в формулы $\neg\alpha$, $\exists\eta\alpha$, $\forall\eta\alpha$, $(\alpha \wedge \beta)$, $(\beta \wedge \alpha)$, $(\alpha \vee \beta)$, $(\beta \vee \alpha)$, $(\alpha \to \beta)$, $(\beta \to \alpha)$, $(\alpha \leftrightarrow \beta)$, $(\beta \leftrightarrow \alpha)$.

Всякое вхождение переменной, не являющееся связанным, называется *свободным* вхождением.

Пример 4. Все четыре вхождения переменной (xx) в формулу из примера 3 — связанные. Первые два вхождения переменной (x) в эту формулу — связанные, третье — свободное.

Формула, не содержащая свободных вхождений переменных, называется *замкнутой*. Формула из примера 3 не является замкнутой. Замкнутые формулы интерпретируются как утверждения; каждой из них приписывается значение «истина» или значение «ложь» согласно описываемому ниже способу (согласующемуся с обычной интерпретацией знаков алфавита А). Те замкнутые формулы, которым окажется приписанным значение «истина», и будут служить «истинными утверждениями».

Пример 5. Предложение «для всякого натурального числа, кроме нуля, найдется меньшее натуральное число» «переводится» следующей формулой нашего языка арифметики:

$$\forall(x)(\neg((x) = ()) \to \exists(xx)\exists(xxx)(\neg((xxx) = ()) \wedge (((xx) + (xxx)) = (x)))).$$

Прежде чем перейти к установлению значений замкнутых формул, нам понадобится еще одно, на этот раз последнее,— техническое понятие. *Подстановкой* цифры c вместо переменной w в формулу α называется замена всех *свободных* вхождений (в α) переменной w на c (эта процедура корректна, так как вхождения переменных в слово не могут перекрываться). Результат такой подстановки будем обозначать через $S_c^w\alpha$; можно показать, что $S_c^w\alpha$ снова является формулой (для любых α, w, c).

Пример 6. Если α — формула из примера 3, то $S_{()}^{(x)}\alpha$ есть

$$(\exists(x)\forall(xx)\neg((x) = (xx)) \leftrightarrow \forall(xx)(() = (xx))),$$

а $S_{(|||)}^{(xx)}\alpha$ есть α.

Теперь мы уже в состоянии перейти к снабжению замкнутых формул *значениями*. Как уже говорилось, таких значений будет два — «истина» ($И$) и «ложь» ($Л$). Значение замкнутой формулы α будем обозначать $|\alpha|$. Понятие значения определяется индуктивно — индукцией по процессу построения формулы; причем мы будем указывать лишь условия, при которых значением рассматриваемой формулы является истина, приняв наперед соглашение, что если значение замкнутой формулы не есть истина, то оно есть ложь.

1° $|(\alpha = \beta)| = И$ тогда [1]) значения термов α и β совпадают;

2° $|\neg\alpha| = И$ тогда $|\alpha| = Л$;

3° $|(\alpha \wedge \beta)| = И$ тогда $|\alpha| = И$ и $|\beta| = И$;
$|(\alpha \vee \beta)| = И$ тогда $|\alpha| = И$ или $|\beta| = И$; $|(\alpha \to \beta)| = И$ тогда $|\alpha| = Л$ или $|\beta| = И$; $|(\alpha \leftrightarrow \beta)| = И$ тогда $|\alpha| = |\beta|$;

4° $|\exists\xi\alpha| = И$ тогда существует такая цифра t, что $|S_t^\xi\alpha| = И$; $|\forall\xi\alpha| = И$ тогда для всякой цифры t справедливо, что $|S_t^\xi\alpha| = И$.

Как легко проверить, описанная процедура приписывает значение каждой замкнутой формуле. Соответственно своему значению, замкнутые формулы подразделяются на *истинные* и *ложные*.

[1]) Сокращение для оборота «тогда и только тогда, когда».

Пример 7. Формула из примера 5 является истинной. Формула из примера 3 не является ни истинной, ни ложной, поскольку не является замкнутой; однако результат подстановки в нее вместо переменной (x) любой цифры является истинной формулой.

Истинные замкнутые формулы мы и объявим истинными утверждениями арифметики. Обозначая их множество буквой T, мы приходим к фундаментальной паре $\langle A, T\rangle$ языка арифметики. Нас будет интересовать возможность ввести для этой пары полную непротиворечивую дедуктику. Мы покажем, что это невозможно, ссылаясь на критерий, установленный в предыдущем параграфе.

Итак, нам надо показать, что существует такое неперечислимое множество натуральных чисел, принадлежность к которому выразима посредством только что введенной фундаментальной пары $\langle A, T\rangle$. С этой целью мы введем в рассмотрение некоторый класс множеств, принадлежность к которым заведома выразима посредством $\langle A, T\rangle$, а затем попытаемся установить наличие в этом классе неперечислимого множества. Класс, о котором идет речь,— класс так называемых арифметических множеств, вводится следующим образом.

Пусть α — формула, не имеющая свободных вхождений никаких переменных, кроме, быть может, переменной (x). Тогда для каждой цифры c формула $S_c^{(x)}\alpha$ является замкнутой и потому истинной либо ложной. Рассмотрим множество всех тех и только тех цифр c, для которых $S_c^{(x)}\alpha$ — истинное утверждение. Будем говорить, что это множество *сопряжено* с формулой α. Каждое множество цифр (а также соответствующее множество чисел), сопряженное с некоторой формулой языка арифметики, будем называть *арифметическим по Гёделю*, или, короче, просто *арифметическим*.

Арифметические множества обладают рядом очевидных свойств:

Свойство 1. Дополнение к арифметическому множеству (до натурального ряда N) есть арифметическое множество. В самом деле, если M сопряжено с α, то $N \setminus M$ сопряжено с $\neg\alpha$.

Свойство 2. Объединение и пересечение арифметических множеств суть арифметические множества. В самом деле, если M_1 и M_2 сопряжены с α_1 и α_2, то $M_1 \cup M_2$ сопряжено с $(\alpha_1 \vee \alpha_2)$, а $M_1 \cap M_2$ — с $(\alpha_1 \wedge \alpha_2)$.

Свойство 3. Принадлежность к произвольному арифметическому множеству выразима посредством $\langle A, T\rangle$. В самом деле, пусть множество M сопряжено с формулой α. Определим функцию f следующим образом: значение f на цифре c есть слово $S_c^{(x)}\alpha$. Тогда f будет вычислимой функцией, выражающей принадлежность к множеству M.

$(*)$ *Существует неперечислимое арифметическое множество.*

Обоснование этого утверждения $(*)$ мы отложим до следующего параграфа. А сейчас заметим, что из него в силу 3-го свойства арифметических множеств и теоремы 8 вытекает, что для пары $\langle A, T\rangle$ нельзя ввести полной непротиворечивой дедуктики. Этот результат может быть назван *теоремой Гёделя о неполноте для формальной арифметики*. Он показывает, что для любого точно сформулированного определенного понятия доказательства найдется

истинное утверждение, формулируемое на языке арифметики, но не являющееся доказуемым.

З а м е ч а н и е. Пусть M — неперечислимое арифметическое множество. Как гласит вторая часть теоремы 8, не существует дедуктики, одновременно непротиворечивой и полной применительно к множеству V значений произвольной функции f, выражающей принадлежность к M. Таким образом, для непротиворечивой дедуктики уже среди членов последовательности $f(0)$, $f(1)$, $f(2)$,[1] . . . . непременно встретятся истинные, но не доказуемые утверждения. В качестве f, как мы только что видели, достаточно взять функцию $n \to \mathbf{S}_n^{(x)}\alpha$, где M сопряжено с α. При таком выборе f слово $f(n)$ естественно интерпретируется как утверждение «$n \in M$». Поэтому, говоря неформально, истинное, но не доказуемое утверждение можно найти (для любой непротиворечивой дедуктики!) среди утверждений вида «$n \in M$». В следующем параграфе мы увидим, что M может быть выбрано так, что его дополнение E до натурального ряда N окажется перечислимым. Итак, существует такое перечислимое множество $E \subseteq N$ что среди истинных утверждений вида «$n \overline{\in} E$» для любой непротиворечивой дедуктики найдется недоказуемое (заменить в этой формулировке «$n \overline{\in} E$» на «$n \in E$» было бы, ввиду теоремы 1, невозможно).

§ 8. Три аксиомы теории алгоритмов

8.0. Наша цель теперь — доказать утверждение (*) из предыдущего параграфа. Однако наших расплывчатых представлений об алгоритмах, которыми мы довольствовались до сих пор, недостаточно для этой цели. Традиционный путь состоит в том, чтобы обратиться к одному из так называемых «уточнений» понятия алгоритма, т. е. заменить несколько неопределенное, но зато совершенно общее, понятие алгоритма, которым мы все время пользовались, достаточно точным, но зато и более узким, понятием «алгоритма специального вида» [2]; в качестве таких «алгоритмов специального вида» наиболее широкое признание получили алгоритмы, связанные с вычислениями на машинах Тьюринга [15], и нормальные алгорифмы А. А. Маркова [8], [9]. Мы здесь, однако, изберем другой путь: не привязывая изложения к тому или иному специальному классу алгоритмов, мы вместо этого попытаемся сформулировать некоторые ограничения, налагаемые на наши первоначальные представления об алгоритмах. Эти ограничения будут сформулированы в виду трех аксиом: аксиомы протокола, аксиомы программы и аксиомы арифметичности.

8.1. Первая аксиома. Рассмотрим процесс применения какого-либо алгоритма $\mathfrak{A}$ к исходному данному x с получением результата y. Мы пред-

---

[1]) Мы отождествляем здесь 0 с (), 1 с (|), 2 с (||) и т. д.

[2]) Это более узкое понятие провозглашается, впрочем, равносильным первоначальному, широкому в том точном смысле, что классы вычислимых функций, возникающие на базе каждого из этих понятий, совпадают (а следовательно, совпадают и классы перечислимых множеств). Указанное совпадение воспринимается не как теорема, подлежащая доказательству, а как естественнонаучная гипотеза, проверяемая на практике.

полагаем, что все промежуточные выкладки, весь процесс вычисления [1]), ведущие от x к y, можно запротоколировать так, чтобы этот протокол содержал исчерпывающую информацию о последовательных этапах процесса.

Пример 1. При работе вычислительной машины, в целях проверки ее работы, часто бывает нужно выдать наружу, «на печать», не только конечный результат, но и все промежуточные результаты. Получаемый таким способом «протокол работы машины» будет словом в выходном алфавите машины — с добавлением, если нужно, знака пробела, знака новой строки и т. п.

Пример 2. Желая проверить, правильно ли усвоен обучающимися алгоритм сложения чисел столбиком, мы можем требовать, чтобы в своих письменных работах они не только указывали конечный результат, но и записывали в определенной системе записи все свои действия. Можно договориться о такой системе записи вычислений, чтобы для сложения, например, чисел 68 и 9967 протокол выглядел так:

		1	11	111	1111	1111	
68,9967	68	68	68	68	68	68	10035
	9967	9967	9967	9967	9967	9967	
		5	35	035	0035	10035	

Каждый из образующих протокол членов есть либо число в десятичной системе (в нашем примере 10035), либо пара чисел (в нашем примере 68, 9967), либо, наконец, четырехэтажное образование вида

$$\begin{array}{r} 11 \\ 68 \\ 9967 \\ 35 \end{array}$$

(«подвальный» и «чердачный» этажи могут быть и пустыми). Не представляет труда оформить протокол в виде слова в некотором алфавите. Для этого достаточно ввести некоторые дополнительные знаки, с тем чтобы только что изображенный четырехэтажный объект записать прежде в виде матрицы

$$\begin{array}{l} **11* \\ ***68 \\ *9967 \\ ***35 \end{array}$$

а затем в виде слова (**11*/***68/*9967/***35). А весь протокол сложения 68 и 9967 запишем так:

(68 + 9967)(*****/***68/*9967/*****)(***1*/***68/*9967/****5)(**11*/***68/*9967/***35) (*111*/***68/*9967/**035) (1111*/***68/*9967/*0035) (1111*/***68/*9967/10035) (10035).

При такой системе записи протокол сложения любых двух чисел является словом в 15-буквенном алфавите {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (,), /, +, *}.

Эти примеры подводят нас к следующим соображениям общего характера. Мы предполагаем, что:

1) для каждого алгоритма $\mathfrak{A}$ имеется некоторый алфавит Π_0 (алфавит протоколов), и всевозможные протоколы, фиксирующие работу $\mathfrak{A}$ при

[1]) Слово «вычисление» понимается здесь в самом широком смысле, отнюдь не исчерпывающимся обычными «цифровыми» подсчетами.

различных исходных данных из его области применимости, образуют подмножество P_0 множества Π_0^∞;

2) существуют такие вычислимые функции α и ω, что для каждого протокола p_0 из P_0 значениями $\alpha(p_0)$ и $\omega(p_0)$ служат соответственно то исходное данное x и тот результат y, для которых составлен данный протокол (т. е. для которых протоколируется переработка x в y);

3) P_0 разрешимо относительно Π_0^∞.

Переформулируем сказанное короче, в виде следующей аксиомы, которую и будем называть *аксиомой протокола*;

для каждого алгоритма $\mathfrak{A}$ *существуют алфавит* Π_0, *разрешимое подмножество* P_0 *множества* Π_0^∞, *вычислимая функция* α *и вычислимая функция* ω, *обладающие следующим свойством*: $\mathfrak{A}(x) = y$ *тогда и только тогда, когда существует такое* p_0 *из* P_0, *что* $\alpha(p_0) = x$ *и* $\omega(p_0) = y$.

Эта аксиома имеет следующее непосредственное

Следствие 1. *Область применимости и множество результатов любого алгоритма перечислимы.*

Доказательство. Первое из этих множеств есть $\alpha(P_0)$, а второе $\omega(P_0)$; оба эти множества перечислимы ввиду лемм 2 и 3 и примера 2 из § 4.

Следствие 2 (из следствия 1). *Область определения и множество значений любой вычислимой функции перечислимы.*

Следствие 3. *График произвольной вычислимой функции* (*т. е. множество всех таких пар* $\langle x, y\rangle$, *что* $f(x) = y$) *есть перечислимое множество.*

Доказательство. Применяем аксиому протокола к алгоритму, вычисляющему f, и берем соответствующие множества P_0, функции α и ω. Строим вычислимую функцию ψ, полагая $\psi(p) = \langle\alpha(p), \omega(p)\rangle$. Замечаем, что график функции f совпадает с множеством $\psi(P_0)$ и применяем лемму 3.

Замечание 1. Следствие 2 можно было бы получить из следствия 3, с учетом следствия 2 леммы 3 и того, что область определения функции и множество значений функции представляются соответственно в виде $\text{пр}_1 M$ и $\text{пр}_2 M$, где M — график функции.

Замечание. 2. Перечислимость графика есть не только необходимое (как это устанавливается следствием 3), но и достаточное условие вычислимости функции. В самом деле, если график пуст, функция нигде не определена и потому вычислима. Если же график функции f не пуст и перечисляется вычислимой функцией ψ, то предлагается такой алгоритм, вычисляющий функцию f: для того, чтобы вычислить значение $f(a)$, перебирай пары $\psi(0)$, $\psi(1)$, $\psi(2)$, . . . до тех пор, пока не получишь пары с первым членом a; второй член этой пары и есть $f(a)$.

8.2. Вторая аксиома. Функции, аргументы которых лежат в X, а значения — в Y, принято называть функциями *из* X *в* Y. Аналогично, алгоритмы, у которых возможные исходные данные лежат в X, а результаты — в Y, будем называть алгоритмами *из* X *в* Y; в этом случае мы принимаем, что $X = К^\infty$, $Y = Л^\infty$, где К и Л — некоторые алфавиты. Каждый алгоритм из К в Л есть предписание, т. е. текст на русском или каком-либо другом

(в частности, искусственном, специально созданном для записи алгоритмов) языке. Хотя в конкретных случаях обычно не возникает сомнений, является или нет данный текст алгоритмом, само понятие предписания слишком неопределенно для того, чтобы мы могли недвусмысленно отличать предписания от непредписаний. Кроме того, у нас нет единого и достаточно точного способа понимать предписания — ведь они могут быть написаны на разных языках, да и в пределах одного языка проблема смысла достаточно сложна. Тем не менее мы предполагаем (и это предположение и составит аксиому программы), что можно выделить четко очерченное множество единообразно понимаемых предписаний (называемых *программами*), причем такое множество, которое было бы представительным в уточняемом ниже смысле. Два алгоритма назовем *равносильными*, коль скоро у них совпадают области применимости и для любого объекта из этой области, взятого в качестве исходного данного, совпадают результаты обоих алгоритмов. Множество алгоритмов из $К^\infty$ в $Л^\infty$ назовем *представительным* (для алфавитов К и Л), коль скоро любой алгоритм из $К^\infty$ в $Л^\infty$ равносилен некоторому алгоритму из рассматриваемого множества. Под «четко очерченным» множеством будем понимать здесь разрешимое подмножество множества всех слов в некотором алфавите. Под «единообразным пониманием» разумеем наличие алгоритма $\mathfrak{U}$, применяемого к парам ⟨программа p, исходное данное a⟩ и дающего в качестве своего результата результат применения программы p к исходному данному a.

Замечание 3. К такой схеме легко сводятся упоминавшиеся уже «уточнения» понятия алгоритма. Каждое такое уточнение состоит, по существу, в том, что указывается некоторое множество P_1 программ, некоторый неформальный алгоритм $\mathfrak{U}$, объясняющий, как применяется программа к заданному исходному объекту; затем провозглашается (в качестве недоказываемой догмы) представительность множества P_1.

Итак, мы предполагаем, что:

1) для каждых двух алфавитов К и Л имеется некоторый алфавит $П_1$ (алфавит программ) и некоторое множество алгоритмов P_1, называемых программами и записанных в алфавите $П_1$ (так что $P_1 \subseteq П_1^\infty$);

2) существует алгоритм $\mathfrak{U}$ из $П_1^\infty \times К^\infty$ в $Л^\infty$ (алгоритм применения программы), такой, что $\mathfrak{U}(p, a)$ есть результат применения p к a;

3) множество P_1 является представительным;

4) множество P_1 разрешимо относительно $П_1^\infty$.

При этом вовсе не предполагается, что алфавит $П_1$, множество P_1 и алгоритм $\mathfrak{U}$ могут быть выбраны лишь единственным образом. Всякую тройку ⟨$П_1$, P_1, $\mathfrak{U}$⟩, где $П_1$ — алфавит, P_1 — множество всех программ, записанных в этом алфавите, и $\mathfrak{U}$ — алгоритм применения программы к аргументу, будем называть *способом программирования из* $К^\infty$ *в* $Л^\infty$.

Таким образом, при заданных К и Л возможны различные способы программирования.

Замечание 4. Предположения 1) — 4) вовсе не определяют понятия «способ программирования» (это понятие остается понимаемым интуитивно), а лишь указывают некоторые (причем, как показывает дальнейший

анализ, еще не все [1])) его свойства и постулируют, что тройка с такими свойствами существует.

Переходим теперь к формулировке второй аксиомы. Но сначала одно обозначение. Пусть $\mathfrak{E}$ — произвольный алгоритм из $\Pi_1^\infty \times K^\infty$ в $Л^\infty$. Через $\mathfrak{E}_p$, где $p \in \Pi_1^\infty$, обозначим следующий алгоритм из K^∞ в $Л^\infty$: для любого a из K^∞ в качестве результата применения $\mathfrak{E}_p$ к a берем результат применения $\mathfrak{E}$ к паре $\langle p, a\rangle$ [так что $\mathfrak{E}_p(a) \simeq \mathfrak{E}(p, a)$]. С помощью этого обозначения мы можем переформулировать предположения 1)—4) в виде следующей *аксиомы программы*:

для любых двух алфавитов К *и* Л *существуют алфавит* Π_1, *разрешимое подмножество* P_1 *множества* Π_1^∞ *и алгоритм* $\mathfrak{U}$ *из* $\Pi_1^\infty \times K^\infty$ *в* $Л^\infty$, *обладающие следующими свойствами: для всякого алгоритма* $\mathfrak{A}$ *из* K^∞ *в* $Л^\infty$ *найдется такое* p *из* P_1, *что алгоритмы* $\mathfrak{A}$ *и* $\mathfrak{U}_p$ *равносильны.*

Эта аксиома также имеет важные следствия. Но прежде ряд определений.

Всякую пару $\langle I, \mu\rangle$, где I — словарное множество, а μ — отображение с областью определения I, условимся называть *семейством*; множество I будем называть *множеством индексов* семейства. Значение отображения μ на ι будем обозначать μ_ι и называть *членом* семейства, а ι — *индексом* этого члена.

Пример 1. Любая дедуктика $\langle Д, D, \delta\rangle$, у которой область определения функции δ совпадает с D, приводит к семейству $\langle D, \delta\rangle$, членами которого служат доказуемые слова, а индексами — доказательства.

Особый интерес представляют два частных вида семейств: *семейство множеств*, все члены которого суть множества, и *семейство функций*, все члены которого суть функции. В первом случае условимся называть *универсальным множеством* семейства совокупность всех пар $\langle \iota, x\rangle$, у которых $\iota \in I$, $x \in \mu_\iota$; во втором случае условимся называть *универсальной функцией* семейства функцию Ψ, определяемую равенством: $\Psi(\iota, x) \simeq \mu_\iota(x)$. Семейство множеств назовем *перечислимым*, если его универсальное множество перечислимо; семейство функций назовем *вычислимым*, если его универсальная функция вычислима [2]).

Пример 2. Пусть $\langle \Pi_1, P_1, \mathfrak{U}\rangle$ — некоторый способ программирования из K^∞ в $Л^\infty$. Каждой программе p поставим в соответствие вычисляемую ею функцию μ_p из K^∞ в $Л^\infty$, т. е. функцию μ_p, определяемую равенством:

---

[1]) А именно, недостает еще следующего предположения «5)»: к семейству, задаваемому способом программирования так, как это указано ниже в примере 2, должно сводиться (в смысле, определяемом в § 10) любое вычислимое семейство, всякий член которого представляет собой вычислимую функцию из K^∞ в $Л^\infty$. Существование тройки, удовлетворяющей предположениям 1)—4), постулируется аксиомой программы; существование же тройки, удовлетворяющей предположениям 1)—5), уже может быть доказано (разумеется, с привлечением аксиомы программы) тем же методом, что и лемма 12 в § 10 (либо получено как следствие этой леммы).

[2]) Если отождествить каждую функцию с ее графиком, понятие семейства функций станет частным случаем понятия семейства множеств и можно будет говорить, в частности, о перечислимых семействах функций. Следствие 3 аксиомы программы и замечание 2 показывают, что для семейства функций его вычислимость и перечислимость оказываются равносильными.

$\mu_p \simeq \mathfrak{U}_p(x)$. Тогда $\langle P_1, \mu \rangle$ есть вычислимое семейство, в котором программа функции служит ее индексом, а универсальная функция вычисляется алгоритмом $\mathfrak{U}$. При этом множество индексов разрешимо относительно Π_1^∞ и потому (пример 2 из § 4 и лемма 2) перечислимо.

Пример 2 расплывчат постольку, поскольку расплывчато понятие «способ программирования». Однако использованное в этом примере построение можно без каких бы то ни было изменений применить к тройке $\langle \Pi_1, P_1, \mathfrak{U} \rangle$, существующей в силу аксиомы программы, и получить тем самым такое

Следствие 1. *Для любых двух алфавитов* К *и* Л *существует вычислимое семейство функций, множеством членов которого является совокупность всех вычислимых функций из* $К^\infty$ *в* $Л^\infty$, *а множество индексов перечислимо.*

Далее, последовательно получаем:

Следствие 2 (из следствия 1). *Существует такая вычислимая числовая функция одного аргумента, которая не продолжается до вычислимой числовой функции, определенной на всем натуральном ряду.*

Доказательство. В следствии 1 в качестве К и Л берем один и тот же алфавит, а именно один из цифровых алфавитов для записи чисел. Образуем вычислимое семейство функций $\langle I, \mu \rangle$, существующее согласно следствию 1. Поскольку I перечислимо, оно перечисляется некоторой вычислимой функцией ρ, определенной на натуральном ряду. Полагаем $\psi(x) \simeq \simeq \Psi(\rho(x), x) + 1$, где Ψ — универсальная функция семейства. Покажем, что ψ — искомая. Вычислимость ψ очевидна. Предположим, что ψ продолжается до некоторой вычислимой φ, определенной на всем натуральном ряду. Тогда при некоторых ι из I и q из N будет $\varphi(x) \simeq \Psi(\iota, x) \simeq \Psi(\rho(q), x)$. Так как φ всюду определена, то определено и значение $\varphi(q)$, а вместе с ним и значения $\Psi(\rho(q), q)$, $\Psi(\rho(q), q) + 1$, $\psi(q)$. Так как φ служит продолжением для ψ, то $\varphi(q) = \psi(q)$, что невозможно.

Следствие 3 (из следствия 2). *Существует перечислимое подмножество натурального ряда, не являющееся разрешимым (относительно натурального ряда).*

Доказательство. Берем вычислимую числовую функцию ψ, не продолжаемую до вычислимой же всюду определенной. Ее область определения E перечислима в силу следствия 2 из аксиомы протокола. Если бы E было разрешимым подмножеством натурального ряда, то мы могли бы следующим образом получить для ψ вычислимое всюду определенное продолжение:

$$\varphi(x) = \begin{cases} \psi(x), & \text{если } x \in E, \\ 0, & \text{если } x \bar{\in} E. \end{cases}$$

Следствие 4 (из следствия 3). *Существует перечислимое подмножество натурального ряда с неперечислимым дополнением.*

Доказательство — по лемме 5.

8.3. Третья аксиома. Если отвлечься от того (впрочем, весьма существенного) обстоятельства, что на вычислительных машинах могут вычисляться лишь функции, определенные на конечных множествах натуральных чисел (поскольку слишком большие аргументы просто не смогут поместиться в машине), то можно считать, что на этих машинах вычисляются вычислимые

числовые функции. Как известно, основными операциями, совершаемыми машиной, являются сложение, умножение и логические операции. Опыт работы на машинах приводит к убеждению, что с помощью этих операций можно запрограммировать любую вычислимую функцию. Следовательно, и всякое перечислимое множество натуральных чисел (как множество значений вычислимой функции) может быть записано в терминах сложения, умножения и логических операций. Сказанное делает естественным формулировку следующей аксиомы (оправдывающейся при любом конкретном «уточнении» понятия алгоритма [1]), которую будем называть *аксиомой арифметичности*:

всякое перечислимое множество натуральных чисел является арифметическим.

Непосредственным следствием этой аксиомы и служит интересующее нас утверждение предшествующего параграфа:

существует арифметическое множество, не являющееся перечислимым.

Таковым является существующее в силу следствия 4 из п. 8.2 неперечислимое множество с перечислимым и, следовательно, арифметическим дополнением. Само это неперечислимое множество будет арифметическим, как дополнительное к арифметическому (1-е свойство арифметических множеств).

§ 9. Эффективно гёделевы языки

Язык, для которого не существует полной непротиворечивой дедуктики, назовем *гёделевым*. Если $\langle$Б, $T\rangle$ — фундаментальная пара гёделева языка, то для любой дедуктики над Б непусто множество $T+P$, где через P обозначено множество доказуемых слов дедуктики, а через $T+P$ — симметрическая разность $(T \setminus P) \cup (P \setminus T)$. В настоящем параграфе мы обсудим вопрос о существовании процедуры, которая для каждой дедуктики над Б эффективно указывала бы элемент в $T+P$: в случае заведомо непротиворечивой дедуктики этот элемент будет принадлежать $T \setminus P$, т. е. будет служить конкретным примером истинного, но не доказуемого утверждения. Языки, для которых такая процедура существует, естественно называть *эффективно гёделевыми*. Наша ближайшая цель — дать точное определение для понятия эффективной гёделевости. Затем мы сформулируем критерии эффективной гёделевости (аналогичные ранее найденным критериям гёделевости), а в следующем параграфе с их помощью установим, что эффективно гёделевым является язык формальной арифметики.

Итак, зададимся фундаментальной парой $\langle$Б, $T\rangle$ и попытаемся уточнить наш пока еще расплывчатый замысел «эффективной гёделевости».

Прежде всего мы фиксируем алфавит доказательств Д и будем рассматривать лишь такие дедуктики над Б, у которых первым членом служит этот Д. Речь должна теперь идти, стало быть, об эффективной процедуре, которая по множеству доказательств D и функции δ выделения доказанного давала бы некоторый элемент в $T+P$.

[1]) При любом из известных уточнений понятия алгоритма как эта, так и предшествующие две аксиомы могут быть доказаны в качестве теорем возникающей теории.

Понятие эффективной процедуры уточняется, как известно, с помощью понятия алгоритма. Однако исходными данными и результатами алгоритмов могут служить лишь слова в фиксированном для каждого данного алгоритма алфавите. Мы перейдем поэтому от пары $\langle D, \delta \rangle$ к так называемому программному заданию этой пары. *Заданием* же пары $\langle D, \delta \rangle$ условимся называть пару $\langle \mathfrak{D}', \mathfrak{D}'' \rangle$, где $\mathfrak{D}'$ — алгоритм, вычисляющий характеристическую функцию множества D относительно $Д^\infty$, а $\mathfrak{D}''$ — алгоритм, вычисляющий функцию δ. В целях дальнейшего уточнения в качестве алгоритмов $\mathfrak{D}'$ и $\mathfrak{D}''$ будем брать лишь программы при произвольных, но фиксированных способах программирования.

Итак, пусть $а$ — какой-то способ программирования из $Д^\infty$ в $Ц^\infty$ (где Ц — какой-то цифровой алфавит [1])), а $б$ — какой-то способ программирования из $Д^\infty$ в $Б^\infty$. Пару $\langle p', p'' \rangle$ назовем *программным заданием* дедуктики $\langle Д, D, \delta \rangle$ (относительно выбранных $а$ и $б$), если p' есть программа, вычисляющая характеристическую функцию множества D (относительно $Д^\infty$), а p'' есть программа, вычисляющая функцию δ. Язык с фундаментальной парой $\langle Б, T \rangle$ назовем *эффективно гёделевым* относительно Д, $а$, $б$, если существует алгоритм, перерабатывающий всякое программное задание дедуктики над Б в слово, принадлежащее симметрической разности $T + P$, где P — множество доказуемых в дедуктике слов. Эта формулировка, однако, лишь по видимости выглядит определением: ведь у нас нет определения для понятия «способ программирования» (см. замечание 4 в § 8). Тем не менее достаточно лишь незначительно подправить эту формулировку, чтобы получить то, что нам надо. В самом деле, мы знаем (пример 2 из § 8), что каждый способ программирования естественным образом приводит к вычислимому семейству, в котором программы служат индексами. Поэтому если мы вместо «способ программирования» и «программа» будем говорить «вычислимое семейство» и «индекс», то таким более общим и одновременно более строгим рассмотрением мы заведомо охватим интересующий нас более частный и одновременно более неопределенный случай.

Итак, окончательные определения таковы.

1) Пусть $а$ и $б$ — семейства функций, а $\langle Д, D, \delta \rangle$ — дедуктика. *Индексным заданием* этой дедуктики относительно $а$ и $б$ назовем всякую пару $\langle \iota', \iota'' \rangle$, где ι' есть индекс относительно $а$ характеристической функции множества D, а ι'' есть индекс относительно $б$ функции δ.

2) Пусть даны язык с фундаментальной парой $\langle Б, T \rangle$, алгоритм $\mathfrak{C}$, алфавит Д и семейства функций $а$ и $б$. Условимся говорить, что $\mathfrak{C}$ устанавливает *гёделевость* рассматриваемого языка относительно Д, $а$, $б$, если всякое индексное задание (относительно $а$ и $б$) всякой дедуктики над Б, имеющей Д своим алфавитом доказательств, перерабатывается алгоритмом $\mathfrak{C}$ в элемент множества $T + P$, где P — множество всех доказуемых (в рассматриваемой дедуктике) слов.

---

[1]) Хотя формально последующие определения зависят от выбора Ц и системы записи чисел, на самом деле это не так (в пределах эквивалентных систем), в чем легко убедиться рассуждениями, сходными с теми, которые проводились в § 4 на стр. 14—15.

3) Язык называется *эффективно гёделевым* относительно Д, $\mathfrak{a}$, $\mathfrak{b}$, если существует алгоритм, устанавливающий его гёделевость относительно Д, $\mathfrak{a}$, $\mathfrak{b}$.

4) Язык называется *эффективно гёделевым*, если он эффективно гёделев относительно любого алфавита Д и любых вычислимых семейств функций $\mathfrak{a}$ и $\mathfrak{b}$.

Простейший критерий гёделевости языка с фундаментальной парой $\langle$Б, $T\rangle$ состоит в неперечислимости T. Естественно ожидать, что критерий эффективной гёделевости должен состоять в эффективной неперечислимости при некотором разумном понимании термина «эффективная неперечислимость». Так оно и оказывается. Что касается эффективной неперечислимости какого-либо множества, то она определяется через наличие эффективной процедуры, устанавливающей отличие этого множества от любого перечислимого множества. Эта идея уточняется следующим образом.

Будем говорить, что некоторый алгоритм *отличает* множество M от членов семейства множеств $\langle$I, $\mu\rangle$, если он перерабатывает всякий индекс ι из I в элемент симметрической разности $\mu_\iota + M$. Будем говорить, что некоторое множество *эффективно отличается* от членов некоторого семейства, если существует алгоритм, отличающий это множество от членов этого семейства.

З а м е ч а н и е 1. Для любого семейства множеств существует словарное множество, эффективно отличающееся от членов этого семейства. Построение такого множества осуществляется стандартным методом канторовой диагонали. Именно, если множество индексов семейства словарно в Г, а универсальное множество семейства есть U, то множество M всех и только тех ι из Γ^∞, для которых $\langle\iota, \iota\rangle \overline{\in} U$, будет эффективно отличаться от членов семейства: в качестве отличающего алгоритма достаточно взять алгоритм, перерабатывающий всякое слово из Γ^∞ в самого себя. Если, в частности, рассматриваемое семейство перечислимо, то M будет дополнительным (до Γ^∞) к перечислимому. В самом деле, обозначим через E множество всех таких пар $\langle x, y\rangle$ из $\Gamma^\infty \times \Gamma^\infty$, у которых $x = y$; оно перечислимо в силу леммы 2. Но $\Gamma^\infty \setminus M = \mathrm{пр}_1(E \cap U)$ и потому перечислимо в силу леммы 6 и следствия 2 леммы 3.

Множество называется *эффективно неперечислимым*, если оно эффективно отличается от членов любого перечислимого семейства множеств.

З а м е ч а н и е 2. Каждое множество, не являющееся словарным, тривиальным образом эффективно неперечислимо. В самом деле, пусть M не словарно, и пусть $\langle$I, $\mu\rangle$— перечислимое семейство. Для некоторого алфавита Б универсальное множество семейства словарно в Б, а тогда и все члены семейства словарны в Б. Берем произвольное c, такое что $c \in M \setminus \text{Б}^\infty$, и в качестве отличающего алгоритма — алгоритм, перерабатывающий всякий индекс в это c. Поэтому понятие эффективной неперечислимости является содержательным лишь в применении к словарным множествам. Пример эффективно неперечислимого словарного множества будет построен в следующем параграфе.

Естественность последнего определения обусловлена, в частности, тем, что (I) для любого перечислимого семейства все его члены суть перечислимые

множества и (II) для каждого алфавита Л существует перечислимое семейство, членами которого являются все перечислимые словарные в Л множества. Поэтому эффективно неперечислимое множество можно трактовать как множество, которое в известном смысле эффективно отличается от всех перечислимых множеств. Чтобы доказать (I), возьмем алгоритм $\mathfrak{A}$, вычисляющий какую-нибудь функцию, перечисляющую универсальное множество семейства; для произвольного индекса ι рассмотрим следующий алгоритм $\mathfrak{B}$ с областью возможных исходных данных N: если $\text{пр}_1\,\mathfrak{A}\,(n) = \iota$, то в качестве $\mathfrak{B}\,(n)$ берем $\text{пр}_2\mathfrak{A}(n)$; в противном случае объявляем $\mathfrak{B}$ неприменимым к n. Очевидно, что множеством результатов алгоритма $\mathfrak{B}$ служит как раз член семейства с индексом ι, каковой член, следовательно (в силу следствия 1 из аксиомы протокола), является перечислимым множеством. Что касается (II), то это утверждение будет доказано в следующем параграфе в качестве леммы 11.

Замечание 3. Естественно было бы попытаться следующим образом усилить определение эффективно неперечислимого множества. Будем говорить, что пара алгоритмов $\mathfrak{A}$, $\mathfrak{B}$ отличает множество M от членов семейства множеств $\langle \text{I}, \mu \rangle$, если $\mathfrak{A}$ перерабатывает всякий индекс ι из I в элемент симметрической разности $\mu_\iota + M$, а $\mathfrak{B}$ перерабатывает ι в 0 или 1 в зависимости от того, принадлежит ли $\mathfrak{A}\,(\iota)$ разности $M \setminus \mu_\iota$ или разности $\mu_\iota \setminus M$. Заменяя в определении понятия «эффективно отличается» слово «алгоритм» на «пара алгоритмов», мы приходим к понятию «усиленно эффективно отличается» и, далее, к понятию усиленно эффективно неперечислимого множества (таковым является в силу замечания 2 каждое не словарное множество). Аналогично возникает понятие усиленно эффективно гёделева языка — для этого в определении эффективно гёделевости нужно потребовать дополнительно наличие алгоритма, указывающего, к какой из разностей, $T \setminus P$ или $P \setminus T$, принадлежит конструируемый элемент симметрической разности $T + P$. Однако оказывается, что ни усиленно эффективно неперечислимых словарных множеств, ни усиленно эффективно гёделевых языков не существует. Покажем отсутствие первых. Предположим, что M — усиленно эффективно неперечислимое словарное в Б множество. Выберем такие перечислимые X и Y, чтобы было $X \subset M \subset Y$ (например, $X = \varnothing$, $Y = \text{Б}^\infty$). Возьмем перечислимое неразрешимое подмножество R натурального ряда; существование такого подмножества установлено в следствии 3 из аксиомы программы. Положим $\mu_n = Y$ при $n \in R$ и $\mu_n = X$ при $n \overline{\in} R$. Образуем семейство $\langle N, \mu \rangle$; универсальное множество этого семейства представляется в виде $(R \times Y) \cup (N \times X)$ и потому, по следствию 1 леммы 3 и лемме 6, перечислимо. Следовательно, перечислимо и само семейство. Следовательно, некоторая пара алгоритмов $\mathfrak{A}$, $\mathfrak{B}$ отличает M от членов этого семейства. Если $n \in R$, то $\mu_n + M = \mu_n \setminus M$ и $\mathfrak{B}(n) = 1$. Если $n \overline{\in} R$, то $\mu_n + M = M \setminus \mu_n$ и $\mathfrak{B}(n) = 0$. Поэтому $\mathfrak{B}$ разрешает множество R относительно N, что невозможно, так как R неразрешимо. Сходным образом обнаруживается отсутствие усиленно эффективно гёделевых языков. Предположим, что $\langle \text{Б}, T \rangle$ — фундаментальная пара такого языка; пусть $X \subset T \subset Y \subseteq \text{Б}^\infty$, где X и Y перечислимы. Взяв такое же R, как несколькими строками выше, образуем перечислимое множество $(R \times Y) \cup (N \times X)$. Пусть Д —

произвольный алфавит. Пример 3 из § 4 позволяет построить вычислимую функцию φ, отображающую $Д^\infty$ на $(R \times Y) \cup (N \times X)$. Положим для всякого n из N и всякого d из $Д^\infty$:

$$A(n, d) = \begin{cases} 1, & \text{если } \mathrm{пр}_1 \varphi(d) = n, \\ 0 & \text{в противном случае,} \end{cases}$$
$$B(n, d) = \mathrm{пр}_2 \varphi(d).$$

Образуем вычислимые семейства функций $\mathfrak{a}$ и $\mathfrak{b}$, взяв N в качестве множества индексов и A и B в качестве универсальных функций. При любом n пара $\langle n, n \rangle$ есть индексное задание (относительно $\mathfrak{a}$, $\mathfrak{b}$) некоторой дедуктики над Б с алфавитом доказательств $Д$; множество доказуемых слов этой дедуктики совпадает с X при $n \overline{\in} R$ и с Y при $n \in R$. Поэтому алгоритм, устанавливающий, к какой из разностей $T \setminus P$ или $P \setminus T$ принадлежит элемент симметрической разности $T + P$, одновременно разрешает R относительно N, что невозможно.

Лемма 8. *Для того чтобы подмножество M перечислимого множества Z было эффективно неперечислимым, достаточно, чтобы оно эффективно отличалось от членов всякого перечислимого семейства подмножеств множества Z.*

Доказательство. Нужно показать, что M эффективно отличается от членов *произвольного* перечислимого семейства $\langle \mathrm{I}, \mu \rangle$. Полагая $\mu'_\iota = \mu_\iota \cap Z$, мы приходим к новому семейству $\langle \mathrm{I}, \mu' \rangle$. Все члены этого нового семейства суть подмножества множества Z, а само оно перечислимо, так как его универсальное множество представимо в виде $U \cap (И^\infty \times Z)$, где U — универсальное множество исходного семейства и $И^\infty \supseteq \mathrm{I}$. Алгоритм, отличающий M от членов семейства $\langle \mathrm{I}, \mu' \rangle$, будет одновременно отличать M и от членов семейства $\langle \mathrm{I}, \mu \rangle$.

Следствие. *Для того чтобы множество M, словарное в Г, было эффективно неперечислимо, достаточно, чтобы оно эффективно отличалось от членов всякого перечислимого семейства множеств, словарных в Г.*

Теорема 9. *Язык с фундаментальной парой $\langle$Б, $T\rangle$ тогда и только тогда является эффективно гёделевым, когда T эффективно неперечислимо.*

Доказательство теоремы 9. 1) Пусть язык с фундаментальной парой $\langle$Б, $T\rangle$ эффективно гёделев. Покажем, что T эффективно неперечислимо. Берем произвольное перечислимое семейство $\langle \mathrm{I}, \mu \rangle$, все члены которого словарны в Б. Ввиду следствия леммы 8 достаточно найти алгоритм, отличающий T от членов этого семейства. Берем универсальное множество U рассматриваемого семейства; оно перечисляется некоторой вычислимой функцией φ. Пусть I словарно в Ж. Строим вычислимые функции A и B, определенные на $Ж^\infty \times N$:

$$A(\iota, n) = \begin{cases} 1, & \text{если } \mathrm{пр}_1 \varphi(n) = \iota, \\ 0 & \text{в противном случае,} \end{cases}$$
$$B(\iota, n) = \mathrm{пр}_2 \varphi(n).$$

Образуем семейство функций $\mathfrak{a} = \langle \mathrm{I}, \alpha \rangle$ с универсальной функцией A и семейство функций $\mathfrak{b} = \langle \mathrm{I}, \beta \rangle$ с универсальной функцией B. Полагаем

Д $=\{\ |\ \}$. Поскольку $\mathfrak{a}$ и $\mathfrak{b}$ вычислимы, наш язык эффективно гёделев относительно Д, $\mathfrak{a}$, $\mathfrak{b}$; рассмотрим алгоритм $\mathfrak{E}$, устанавливающий гёделевость. Положим $\mathfrak{E}_1(\iota) \simeq \mathfrak{E}(\iota, \iota)$. Алгоритм $\mathfrak{E}_1$ и будет искомым алгоритмом, отличающим T от членов семейства $\langle \mathrm{I}, \mu \rangle$. В самом деле, достаточно обнаружить, что $\mathfrak{E}(\iota, \iota) \in \mu_\iota + T$ для всех ι из I. Для этого фиксируем ι; рассмотрим в $\{\ |\ \}^\infty$ подмножество D с характеристической функцией α_ι. Тройка $\langle$Д, D, $\beta_\iota\rangle$ представляет собой дедуктику над Б, а пара $\langle \iota, \iota \rangle$ — ее индексное задание относительно $\mathfrak{a}$ и $\mathfrak{b}$. Поэтому $\mathfrak{E}(\iota, \iota) \in P + T$, где P — множество всех доказуемых слов. Но, как нетрудно проверить, $P = \mu_\iota$. Действительно, если $x \in \mu_\iota$, то $\langle \iota, x \rangle \in U$ и при некотором n имеем $\varphi(n) = \langle \iota, x \rangle$. Тогда $\alpha_\iota(n) = A(\iota, n) = 1$, $n \in D$, $\beta_\iota(n) = B(\iota, n) = x$ и потому $x \in P$. Если теперь $x \in P$, то существует такое n, что $\alpha_\iota(n) = 1$, $\beta_\iota(n) = x$. Тогда $A(\iota, n) = 1$, $B(\iota, n) = x$, $\varphi(n) = \langle \iota, x \rangle$, $\langle \iota, x \rangle \in U$, $x \in \mu_\iota$.

2) Пусть T эффективно неперечислимо. Покажем, что язык с фундаментальной парой $\langle$Б, $T\rangle$ эффективно гёделев. Рассмотрим произвольный алфавит Д и вычислимые семейства функций $\mathfrak{a}$ и $\mathfrak{b}$; надо найти алгоритм, устанавливающий гёделевость. Пусть $\mathfrak{a} = \langle \Xi, \mu \rangle$ с универсальной функцией A, а $\mathfrak{b} = \langle \Sigma, \nu \rangle$ с универсальной функцией B и пусть $\Xi \subseteq$ Ж$^\infty$, $\Sigma \subseteq$ И$^\infty$, где Ж и И — некоторые алфавиты. Положим $R = \{\langle\langle \xi, \sigma \rangle, x\rangle \mid \xi \in \in$ Ж$^\infty$, $\sigma \in$ И$^\infty$, $x \in$ Б$^\infty$ и существует такое d из Д$^\infty$, что $A(\xi, d) = 1$ и $B(\sigma, d) = x\}$. Рассмотрим R как универсальное множество семейства $\langle \Xi \times \Sigma, \mu \rangle$ с некоторым μ. Пусть $\langle \xi, \sigma \rangle$ — индексное задание относительно $\mathfrak{a}$ и $\mathfrak{b}$ некоторой дедуктики $\langle$Д, D, $\delta\rangle$ над Б, а P — множество доказуемых слов этой дедуктики. Покажем, что $P = \mu_{\langle \xi, \sigma \rangle}$. В самом деле, пусть $x \in \mu_{\langle \xi, \sigma \rangle}$; это значит, что $\langle\langle \xi, \sigma \rangle, x\rangle \in R$ и, следовательно, для некоторого d из Д$^\infty$ справедливо $A(\xi, d) = 1$, $B(\sigma, d) = x$. Поскольку $\langle \xi, \sigma \rangle$ — индексное задание дедуктики, то $d \in D$, $B(\sigma, d) = \delta(d)$, $x \in P$. В другую сторону: пусть $x \in P$; тогда для некоторого d из D выполняется $\delta(d) = x$. Поскольку $\langle \xi, \sigma \rangle$— индексное задание, то $A(\xi, d) = 1$, $B(\sigma, d) = x$ и $\langle\langle \xi, \sigma \rangle, x\rangle \in R$, $x \in \in \mu_{\langle \xi, \sigma \rangle}$. Если мы обнаружим теперь, что R перечислимо, то построенное семейство множеств будет перечислимым и для него будет существовать алгоритм $\mathfrak{E}$, отличающий T от членов этого семейства; тогда $\mathfrak{E}(\xi, \sigma) \in \mu_{\langle \xi, \sigma \rangle} + \\ + T$. Ввиду равенства $\mu_{\langle \xi, \sigma \rangle} = P$ алгоритм $\mathfrak{E}$ и будет устанавливать гёделевость относительно $\langle$Д, $\mathfrak{a}$, $\mathfrak{b}\rangle$. Итак, все свелось к установлению перечислимости множества R. Этой перечислимостью мы сейчас и займемся. Положим

$$S = \{\langle\langle \xi, \sigma \rangle, x, d\rangle \mid \xi \in \text{Ж}^\infty, \sigma \in \text{И}^\infty, x \in \text{Б}^\infty, d \in \text{Д}^\infty, A(\xi, d) = 1, B(\sigma, d) = x\}.$$

Очевидно, что $R = \text{пр}_{1,\ 2}S$. В силу следствия 2 леммы 3 достаточно доказать перечислимость S. Положим

$$S_A = \{\langle\langle \xi, \sigma \rangle, x, d\rangle \mid \xi \in \text{Ж}^\infty, \sigma \in \text{И}^\infty, x \in \text{Б}^\infty, d \in \text{Д}^\infty, A(\xi, d) = 1\},$$

$$S_B = \{\langle\langle \xi, \sigma \rangle, x, d\rangle \mid \xi \in \text{Ж}^\infty, \sigma \in \text{И}^\infty, x \in \text{Б}^\infty, d \in \text{Д}^\infty, B(\sigma, d) = x\}.$$

Тогда $S = S_A \cap S_B$ и достаточно (в силу леммы 6) доказать перечислимость каждого из множеств S_A и S_B. Определим вычислимые функции A_1 и B_1,

положив для каждого элемента $\langle\langle \xi, \sigma\rangle, x, d\rangle$ из $(Ж^\infty \times И^\infty) \times Б^\infty \times Д^\infty$

$$A_1(\langle\langle \xi, \sigma\rangle, x, d\rangle) \simeq A(\xi, d),$$
$$B_1(\langle\langle \xi, \sigma\rangle, x, d\rangle) \simeq B(\sigma, d).$$

Тогда S_A перечислимо по лемме 7 как полный прообраз относительно A_1 перечислимого одноэлементного множества $\{\ |\ \}$. Чтобы обнаружить перечислимость S_B, рассмотрим такой алфавит Л, что все значения функции B_1 суть слова в Л, так что перечислимый (в силу следствия 3 из аксиомы протокола) график G функции B есть подмножество произведения $((Ж^\infty \times И^\infty) \times Б^\infty \times Д^\infty) \times Л^\infty$. Это последнее произведение обозначим через W, а через E — подмножество всех таких его элементов $\langle\langle\langle \xi, \sigma\rangle, x, d\rangle, y\rangle$, у которых $x = y$. Это E, будучи, очевидно, разрешимо относительно W, будет (в силу следствия 1 леммы 3 и леммы 5) перечислимо. Замечая, что $\text{пр}_1(G \cap E) = S_B$, получаем в силу следствия 2 леммы 3 нужную нам перечислимость множества S_B.

Замечание 4. Для дальнейшего важно обратить внимание на следующее. Во второй части доказательства теоремы 9 для каждой тройки Д, $\mathfrak{a}$, $\mathfrak{b}$ строится такое перечислимое семейство, что любой алгоритм, отличающий T от членов этого семейства, одновременно устанавливает и гёделевость языка относительно Д, $\mathfrak{a}$, $\mathfrak{b}$. Поэтому если для любого перечислимого семейства мы сможем найти такой устанавливающий отличие T от членов этого семейства алгоритм, который обладает некоторыми специальными свойствами (например, такой, что все его результаты принадлежат заданному множеству), то и для любых Д, $\mathfrak{a}$, $\mathfrak{b}$ мы сможем найти устанавливающий гёделевость алгоритм с теми же свойствами.

§ 10. Эффективная гёделевость языка арифметики

В силу соображений, высказанных в начале § 6, естественно стремиться к тому, чтобы элемент симметрической разности $T + P$, получаемый как результат устанавливающего гёделевость алгоритма, принадлежал к некоторому заранее выделенному множеству, содержательно интерпретируемому как множество «сравнительно простых утверждений». Это стремление оформляется в виде следующего определения. Пусть $V \subseteq Б^\infty$; скажем, что язык с фундаментальной парой $\langle Б, T\rangle$ эффективно гёделев *применительно* к V, если для любых алфавита Д и вычислимых семейств $\mathfrak{a}$ и $\mathfrak{b}$ существует такой (устанавливающий гёделевость) алгоритм, который перерабатывает всякое индексное задание относительно $\mathfrak{a}$ и $\mathfrak{b}$ всякой дедуктики над Б с алфавитом доказательств Д в элемент пересечения $(T + P) \cap V$. Для установления эффективной гёделевости языка достаточно (и необходимо) найти такое V, применительно к которому язык оказался бы эффективно гёделевым. Справедливо следующее утверждение (служащее аналогом для совокупности, образуемой теоремой 7 и идущим вслед за нею замечанием 1): если V — перечислимое подмножество множества $Б^\infty$, то эффективная гёделевость языка с фундаментальной парой $\langle Б, T\rangle$ применительно к V равносильна эффективной неперечислимости пересечения $V \cap T$. Мы, однако, не будем доказывать

это утверждение, а перейдем сразу к еще более простому критерию, который дает следующий аналог теоремы 8 и замечания 2 из § 6.

Т е о р е м а 10. *Язык тогда и только тогда эффективно гёделев, когда в нем выразима принадлежность к какому-нибудь эффективно неперечислимому множеству натуральных чисел; в таком случае язык эффективно гёделев применительно к множеству значений функции, выражающей эту принадлежность.*

Д о к а з а т е л ь с т в о т е о р е м ы 10. Рассмотрим язык с фундаментальной парой $\langle Б, T\rangle$.

1) Пусть этот язык эффективно гёделев. Функция f, перечисляющая $Б^\infty$, выражает принадлежность к $f^{-1}(T)$ (ср. замечание 2 из § 6). Остается доказать, что $f^{-1}(T)$ эффективно неперечислимо. Для простоты возьмем в качестве f взаимно однозначное вычислимое отображение N на $Б^\infty$ (см. пример 2 из § 4). Пусть $\langle I, \mu\rangle$ — перечислимое семейство, причем I словарно в Ж. Лемма 8 позволяет предполагать, что $\mu_\iota \subseteq N$ для всех ι. Найдем алгоритм, отличающий $f^{-1}(T)$ от членов этого семейства. С этой целью образуем новое семейство $\langle I, \bar{\mu}\rangle$, полагая $\bar{\mu}_\iota = f(\mu_\iota)$. Обозначим через U, $\bar{U}$ универсальные множества этих семейств. Очевидно, $\bar{U}$ получается из U посредством вычислимого отображения $\langle \iota, n\rangle \to \langle \iota, f(n)\rangle$. Поэтому $\bar{U}$ перечислимо (лемма 3). Значит, $\langle I, \bar{\mu}\rangle$ — перечислимое семейство и некоторый алгоритм $\mathfrak{C}$ устанавливает отличие от членов этого семейства множества T, эффективно неперечислимого в силу теоремы 9. Тогда

$$\mathfrak{C}(\iota) \in \bar{\mu}_\iota + T, \quad f^{-1}(\mathfrak{C}(\iota)) \in \mu_\iota + f^{-1}(T)$$

и алгоритм, вычисляющий функцию $y = f^{-1}(\mathfrak{C}(x))$, будет искомым.

2) Пусть g выражает принадлежность к эффективно неперечислимому множеству R натуральных чисел. Сейчас мы для произвольного перечислимого семейства $\langle I, \mu\rangle$ найдем алгоритм, отличающий T от членов этого семейства; тем самым мы установим эффективную неперечислимость T, достаточную, по теореме 9, для эффективной гёделевости языка; некоторое специальное свойство этого алгоритма позволит сделать содержащееся в теореме заключение о множестве значений функции g. Для каждого ι из I положим $\nu_\iota = g^{-1}(\mu_\iota)$, так что $\nu_\iota \subseteq N$, и рассмотрим семейство $\langle I, \nu\rangle$. Его универсальное множество V получается из универсального множества первоначального семейства как полный прообраз при вычислимом отображении $\langle \iota, n\rangle \to \langle \iota, g(n)\rangle$, определенном на $И^\infty \times N$, где $И^\infty \supseteq I$; по лемме 7 V перечислимо. Поэтому семейство $\langle I, \nu\rangle$ перечислимо; пусть алгоритм $\mathfrak{C}$ отличает R от членов этого семейства. Тогда

$$\mathfrak{C}(\iota) \in \nu_\iota + R, \quad g(\mathfrak{C}(\iota)) \in \mu_\iota + T$$

и алгоритм, вычисляющий функцию $y = g(\mathfrak{C}(x))$ будет отличать T от членов семейства $\langle I, \mu\rangle$. Специальным свойством, о котором выше шла речь, служит принадлежность всякого результата этого алгоритма к множеству значений g; в силу замечания 4 из § 9 можно считать, что таковы же и результаты алгоритма, устанавливающего гёделевость (относительно произвольных Д,

$\mathfrak{a}$, $\mathfrak{b}$); это и означает эффективную гёделевость применительно к множеству значений g.

Теорема 10 доставляет нам средство для установления эффективной гёделевости языка формальной арифметики. Эта эффективная гёделевость вытекает в силу 3-го свойства арифметических множеств (§ 8) из утверждения

(**) *существует эффективно неперечислимое арифметическое множество.*

Что же касается самого утверждения (**), то оно получается с учетом аксиомы арифметичности и 1-го свойства арифметических множеств как следствие теоремы 11.

Т е о р е м а 11. *Существует перечислимое множество натуральных чисел с эффективно неперечислимым дополнением.*

Доказательством этой теоремы мы и завершим наше изложение. Но сначала несколько замечаний, определений и лемм, которые помогут нам при доказательстве; некоторые из них могут представить и самостоятельный интерес.

Будем говорить, что функция h *сводит* семейство $\langle \Xi, \alpha \rangle$ к семейству $\langle \Sigma, \beta \rangle$, если h определена для каждого ξ из Ξ и $\beta_{h(\xi)} = \alpha_\xi$. Будем говорить, что семейство $\mathfrak{a}$ *сводится* к семейству $\mathfrak{b}$, если существует вычислимая функция, сводящая $\mathfrak{a}$ к $\mathfrak{b}$. Очевидно, что отношение сводимости транзитивно: если вычислимая функция h сводит $\mathfrak{a}$ к $\mathfrak{b}$, а вычислимая функция g сводит $\mathfrak{b}$ к $\mathfrak{c}$, то вычислимая функция $y = g(h(x))$ сводит $\mathfrak{a}$ к $\mathfrak{c}$.

Л е м м а 9. *Если семейство множеств $\mathfrak{a}$ сводится к семейству множеств $\mathfrak{b}$, а множество M эффективно отличается от членов семейства $\mathfrak{b}$, то оно эффективно отличается и от членов семейства $\mathfrak{a}$.*

Д о к а з а т е л ь с т в о. Пусть функция h, сводящая $\mathfrak{a}$ к $\mathfrak{b}$, вычисляется алгоритмом $\mathfrak{H}$, а алгоритм $\mathfrak{E}$ отличает M от членов $\mathfrak{b}$. Полагая $\mathfrak{E}'(x) \simeq \mathfrak{E}(\mathfrak{H}(x))$, получаем алгоритм $\mathfrak{E}'$, отличающий M от членов $\mathfrak{a}$.

Два семейства назовем *эквивалентными*, если каждое из них сводится к другому. Очевидно, что если два семейства эквивалентны, то множество всех членов одного из них совпадает с множеством всех членов другого.

З а м е ч а н и е 1. Для каждого перечислимого семейства найдется эквивалентное ему перечислимое же семейство, множество индексов которого словарно в произвольном наперед заданном алфавите И. Действительно, пусть $\langle \mathrm{I}, \mu \rangle$ — какое-то перечислимое семейство множеств; пусть I словарно в Ш. Возьмем вычислимую функцию f взаимно однозначно отображающую Ш$^\infty$ на И$^\infty$ (пример 3 из § 4). Построим новое семейство $\langle \mathrm{K}, \nu \rangle$ следующим образом: $\mathrm{K} = f(\mathrm{I})$ и для каждого элемента $f(\iota)$ из K полагаем $\nu_{f(\iota)} = \mu_\iota$. Построенное семейство и будет искомым: его множество индексов словарно в И; оно эквивалентно исходному семейству (f сводит $\langle \mathrm{I}, \mu \rangle$ к $\langle \mathrm{K}, \nu \rangle$, а f^{-1} сводит $\langle \mathrm{K}, \nu \rangle$ к $\langle \mathrm{I}, \mu \rangle$); наконец, оно перечислимо, поскольку его универсальное множество получается из универсального множества семейства $\langle \mathrm{I}, \mu \rangle$ как образ при вычислимом отображении $\langle \iota, z \rangle \to \langle f(\iota), z \rangle$.

Семейство множеств назовем *главным* для алфавита Г, если все его члены словарны в Г и к нему сводится любое перечислимое семейство множеств, все члены которого словарны в Г.

Лемма 10. *Если словарное в* Г *множество эффективно отличается от членов главного для* Г *семейства, то оно эффективно неперечислимо.*

Доказательство. Пусть $M \subseteq \Gamma^\infty$ и M эффективно отличается от членов главного для Г семейства. В силу леммы 9 M эффективно отличается от членов всякого перечислимого семейства словарных в Г множеств. В силу следствия леммы 8 M эффективно неперечислимо.

Замечание 2. В силу транзитивности отношения сводимости всякое семейство словарных в Г множеств, к которому сводится главное для Г семейство, само является главным для Г.

Замечание 3. Пусть Г и И — произвольные алфавиты. Для всякого перечислимого семейства, главного для Г, можно построить, согласно замечанию 1, эквивалентное перечислимое семейство с множеством индексов, словарным в И. Это последнее семейство в силу замечания 2 также будет главным для Г.

Предположим теперь, что для каждого алфавита Г мы умеем строить главное перечислимое семейство. Тогда доказательство теоремы 11 завершается в несколько строк. Действительно, на основании замечания 3 можно считать, что множество индексов рассматриваемого семейства словарно в том же Г. Применяя диагональную конструкцию замечания 1 из § 9, строим множество М, словарное в Г, имеющее перечислимое дополнение (до Γ^∞) и эффективно отличающееся от членов рассматриваемого главного семейства. В силу леммы 10 это M будет эффективно неперечислимым. Если теперь взять в качестве Г однобуквенный алфавит { | } и обычным способом записывать в нем натуральные числа так, что $\Gamma^\infty = N$, то M образует пример эффективно неперечислимого подмножества натурального ряда с перечислимым дополнением, а $\Gamma^\infty \setminus M$ — пример требуемого в теореме 11 множества.

Лежащее в основе изложенного доказательства теоремы 11 предположение составляет предмет леммы 12. Лемма 11 играет вспомогательную роль для леммы 12.

Лемма 11. *Для каждого алфавита* Л *существует такое перечислимое семейство, что всякое перечислимое словарное в* Л *множество является членом этого семейства.*

Доказательство. Пусть К — какой-либо цифровой алфавит. Применяя к алфавитам К и Л следствие 1 из аксиомы программы, получаем вычислимое семейство функций $\langle \mathrm{I}, \mu \rangle$. Для каждого ι обозначим через ν_ι множество значений функций μ_ι. Семейство $\langle \mathrm{I}, \nu \rangle$ и будет искомым. Во-первых, $\langle \mathrm{I}, \nu \rangle$ перечислимо; в самом деле, универсальная функция семейства $\langle \mathrm{I}, \mu \rangle$ вычислима, ее график G, состоящий из пар вида $\langle \langle \iota, x \rangle, y \rangle$, где $\iota \in \mathrm{I}$, $x \in \mathrm{K}^\infty$, $y \in \mathrm{Л}^\infty$, перечислим в силу следствия 3 из аксиомы протокола, а универсальное множество семейства $\langle \mathrm{I}, \nu \rangle$ получается из G вычислимым отображением $\langle \langle \iota, x \rangle, y \rangle \leftrightarrow \langle \iota, y \rangle$. Во-вторых, каждое перечислимое словарное в Л множество есть, по определению, множество значений некоторой вычислимой функции из K^∞ в $\mathrm{Л}^\infty$; эта функция есть μ_ι при некотором ι; тогда ν_ι и есть исходное множество.

Лемма 12. *Для каждого алфавита существует главное перечислимое семейство множеств.*

Доказательство. Пусть Г — алфавит, для которого мы хотим найти главное семейство. Фиксируем произвольный алфавит И. Предположим, что нам удалось построить семейство множеств $\mathfrak{c}$, все члены которого словарны в Г и которое обладает следующим свойством: если семейство $\mathfrak{b}$ перечислимо, все его члены словарны в Г, а множество индексов словарно в И, то $\mathfrak{b}$ сводится к $\mathfrak{c}$. Тогда $\mathfrak{c}$ будет главным для Г. Действительно, каковобы ни было перечислимое семейство $\mathfrak{a}$ словарных в Г множеств, в силу замечания 2 оно эквивалентно перечислимому же семейству $\mathfrak{b}$ с множеством индексов, словарным в И; все члены семейства $\mathfrak{b}$ словарны в Г и потому в силу сформулированного свойства семейства $\mathfrak{c}$ семейство $\mathfrak{b}$ сводится к $\mathfrak{c}$; а тогда $\mathfrak{a}$ сводится к $\mathfrak{c}$ по транзитивности. Итак, достаточно построить *перечислимое* $\mathfrak{c}$, обладающее указанным свойством. С этой целью прежде всего найдем такой алфавит Л, что $\text{И}^\infty \times \Gamma^\infty \subseteq \text{Л}^\infty$. Для этого Л образуем перечислимое семейство $\langle \Delta, \nu \rangle$, удовлетворяющее требованиям леммы 11; множество индексов этого семейства словарно в некотором Ж, а универсальным множеством служит некоторое перечислимое V. Построим семейство $\langle \text{Ж}^\infty \times \text{И}^\infty, \gamma \rangle$ с универсальным множеством W, где $W \subseteq (\text{Ж}^\infty \times \text{И}^\infty) \times \Gamma^\infty$ и $\langle \langle \delta, \iota \rangle, x \rangle \in W$ тогда и только тогда, когда $\langle \delta, \langle \iota, x \rangle \rangle \in V$. Покажем, что так построенное семейство — искомое. Во-первых, все его члены, очевидно, словарны в Г. Во-вторых, оно перечислимо; действительно, W есть полный прообраз перечислимого V относительно следующей вычислимой функции из $(\text{Ж}^\infty \times \text{И}^\infty) \times \Gamma^\infty$ в $\text{Ж}^\infty \times \text{Л}^\infty$: $\langle \langle \delta, \iota \rangle, x \rangle \to \langle \delta, \langle \iota, x \rangle \rangle$. Наконец, в-третьих, если все члены перечислимого семейства $\langle \text{I}, \mu \rangle$ словарны в Г, а I словарно в И, то $\langle \text{I}, \mu \rangle$ сводится к $\langle \text{Ж}^\infty \times \text{И}^\infty, \gamma \rangle$. Действительно, пусть универсальное множество семейства $\langle \text{I}, \mu \rangle$ есть U; оно перечислимо и словарно в Л. Поэтому для некоторого δ из Δ имеем $\nu_\delta = U$. Фиксируем это δ и определим на И^∞ вычислимую функцию h, полагая для всех ι из И^∞ $h(\iota) = \langle \delta, \iota \rangle$. Так определенная h и будет сводить $\langle \text{I}, \mu \rangle$ к $\langle \text{Ж}^\infty \times \text{И}^\infty, \gamma \rangle$. Для этого надо проверить, что для любого ι из I выполняется равенство $\mu_\iota = \gamma_{\langle \delta, \iota \rangle}$. Но это равенство вытекает из цепочки равносильностей

$$x \in \mu_\iota \Leftrightarrow \langle \iota, x \rangle \in U \Leftrightarrow \langle \iota, x \rangle \in \nu_\delta \Leftrightarrow$$
$$\Leftrightarrow \langle \delta, \langle \iota, x \rangle \rangle \in V \Leftrightarrow \langle \langle \delta, \iota \rangle, x \rangle \in W \Leftrightarrow x \in \gamma_{\langle \delta, \iota \rangle}.$$

ЛИТЕРАТУРА

[1] Н. Бурбаки, Теория множеств, М., 1965.

[2] K. Gödel, Einige metamathematische Resultate über Entscheidungsdefinitheit und Widerspruchsfreiheit, Anzeiger der Akademie der Wissenschaften in Wien, Mathematisch-naturwissenschaftliche Klasse, № 19 (1930), 214—215.

[3] K. Gödel, Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I, Monatshefte für Math. und Physik 38:1 (1931), 173—198.

[4] С. К. Клини, Введение в метаматематику, М., 1957.

[5] J. Ladrière, Les limitations internes des formalismes, Louvain—Paris, 1957.

[6] А. И. Мальцев, Алгоритмы и рекурсивные функции, М., 1965.

[7] А. А. Марков, Невозможность некоторых алгорифмов в теории ассоциативных систем, ДАН 55 (1947), 587—590.

[8] А. А. Марков, Теория алгорифмов, Труды Матем. ин-та им. В. А. Стеклова 38 (1951), 176—189.

[9] А. А. Марков, Теория алгорифмов, Труды Матем. ин-та им. В. А. Стеклова **42** (1954).
[10] Э. Мендельсон, Введение в математическую логику, М., 1971.
[11] Э. Нагель, Дж. Р. Ньюмен, Теорема Гёделя, М., 1970.
[12] E. L. Post, Recursively enumerable sets of positive integers and their decision problems, Bull. Amer. Math. Soc. **50**:5 (1944), 284—316.
[13] E. L. Post, Recursive unsolvability of a problem of Thue, J. symbolic logic **12**: 1 (1947), 1—11.
[14] J. B. Rosser, Extensions of some theorems of Gödel and Church. J. symbolic logic **1**:3 (1936), 87—91.
[15] A. M. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proc. London Math. Soc., ser. 2, **42**:3, 4 (1936), 230—265; **43**:7 (1937), 544—546.
[16] A. Whitehead, B. Russell, Principia mathematica, vol. 1—3, Cambridge, 1910—1913 (2-е изд. 1925—1927).
[17] В. А. Успенский, Теорема Гёделя и теория алгоритмов, УМН 8:4 (56) (1953), 176—178.
[18] В. А. Успенский, Теорема Гёделя и теория алгоритмов, ДАН **91**:4 (1953), 737—740.
[19] Г. С. Цейтин, Ассоциативное исчисление с неразрешимой проблемой эквивалентности, Труды Матем. ин-та им. В. А. Стеклова **52** (1958), 172—189.

Поступило в редакцию 8 октября 1973 г.