Кибернетический сборник

НОВАЯ СЕРИЯ

ВЫПУСК

23

Сборник переводов

под редакцией
О. Б. ЛУПАНОВА

МОСКВА «МИР» 1986

ББК 32.81
К38
УДК 519.95

К 38 **Кибернетический** сборник. Новая серия. Вып. 23. Сб. статей: Пер. с англ. — М.: Мир, 1986, 190 с., ил.

Продолжение серии, начатой издательством «Мир» в 1965 г. В выпуске содержатся обзорные статьи и оригинальные работы известных зарубежных ученых по наиболее актуальным проблемам теоретической кибернетики. Большой интерес представляют статьи Х. Уильямса, Г. Миллера и др. (США) по вопросам тестирования простоты чисел, И. Филотти (США) по алгоритмической теории графов.

Для научных работников, инженеров-исследователей, аспирантов и студентов, занимающихся и интересующихся теоретической кибернетикой и ее приложениями.

К $\frac{1502000000-353}{041(01)-86}$ 04—86, ч. 1 **ББК 32.81**

Редакция литературы по математическим наукам

Один алгоритм укладки кубического графа на торе[1]

И. С. Филотти[2])

1. ВВЕДЕНИЕ

Род графа — это наименьший из родов ориентируемых поверхностей, на которые его можно уложить. Граф называется *планарным*, если он рода 0, и *тороидальным*, если он рода 1.

В этой работе описан эффективный алгоритм для определения того, является ли кубический граф тороидальным. Алгоритм строит укладку графа на торе, если она существует. Слово «эффективный» понимается в смысле Кука, Эдмондса и Собхема. Алгоритм считается «эффективным», если время его работы ограничено полиномом от длины входа (Ахо и др. [1]). В нашем случае время работы алгоритма ограничено полиномом от числа ребер графа.

Алгоритмы установления планарности графов широко обсуждались в литературе. Наиболее важен с нашей точки зрения алгоритм, предложенный Демукроном и др. [5]. Он был улучшен Рубином [23]. В последующие годы большое внимание привлек линейный алгоритм Хопкрофта и Тарьяна [17], который иллюстрирует применение поиска в глубину, в чем эти два автора были первыми. Этот алгоритм обсуждается также в [6] и [20]. Другой линейный алгоритм был описан Бутом [3]. Обширный список работ по алгоритмам установления планарности содержится в работе [17], но в нем отсутствует алгоритм работы [5], который обсуждается Бонди и Мурти [2].

Алгоритмы определения рода графа развивались в связи с попытками решать проблемы комбинаторного представления топологических укладок графов. Комбинаторное представление укладок (называемых еще картами), которое мы используем здесь, было известно еще Хефтеру [16]. Затем оно было переоткрыто Эдмондсом [7, 8] и существенно обосновано в [9]. Среди авторов, обсуждавших различные аспекты этой методики, — Уолш и Леман [27], Рингель [21, 22] и Татт [26]. Этот список не является полным и не охватывает многих других

---

[1]) Filotti I. S. An Algorithm for Imbedding Cubic Graphs in the Torus. — Journal of Computer and System Sciences, **20**, № 2, 255—276 (1980).

[2]) Columbia University, New York, New York 10027.

аспектов топологической теории графов, которая является относительно старым разделом, но этой теории посвящена только монография [29]. Библиографии по топологической теории графов приведены в [25] и [30].

Обсуждаемое здесь комбинаторное представление — это представление, сопоставляющее каждой вершине графа циклическую ориентацию инцидентных ей ребер. Грани укладки являются орбитами некоторой группы, действующей естественным образом на множестве ребер. Род укладки можно вычислить, используя формулу Эйлера. Чтобы найти род графа, достаточно рассмотреть все возможные назначения вершинам циклических ориентаций. Этот алгоритм имеет сложность $O(n^{n^2})$ (где n — число вершин), и только он был известен.

Подходы к улучшению этого алгоритма не являются очевидными. Гросс и Такер [14] отмечают некоторые из имеющихся трудностей. Они пытались уменьшить род укладки с помощью изменения циклической ориентации для некоторой вершины. Существовала надежда, что можно найти такую последовательность вершин, что изменение их циклических ориентаций даст монотонное улучшение рода. К сожалению, этот подход не увенчался успехом.

Основная идея данной работы навеяна алгоритмом установления планарности Демукрона и др. [5] (обозначаемым далее ДМП). Алгоритм ДМП работает следующим образом. В графе G произвольным образом выбирается цикл C и укладывается на плоскости. В результате образуются две планарные области. Теперь задача — установить, можно ли эту укладку продолжить до всего графа G. Это частный случай задачи о продолжении: дана планарная укладка h подграфа H графа G; определить, можно ли распространить h на весь граф G.

Одно особое обстоятельство делает решение задачи о планарном продолжении особенно легким. Будем говорить, что две части P_1 и P_2 из $G—H$ препятствуют друг другу относительно грани F укладки h (обозначим это $P_1 \mid_F P_2$), если они не могут быть одновременно уложены в F без пересечения внутри грани F. Это топологическое понятие имеет следующий комбинаторный эквивалент: $P_1 \mid_F P_2$ тогда и только тогда, когда они имеют либо три общих прикрепления к F, либо две перекрещивающиеся цепи (см. предложение 5.3). Далее, если $P_1 \mid_{F_1} P_2$ и также P_1 и P_2 укладываются в некоторую другую грань, то это одна и та же грань, скажем грань F_2, и $P_1 \mid_{F_2} P_2$. Последнее обстоятельство, названное нами «Теоремой о безразличии», является решающим.

Для решения задачи о продолжении необходимо выполнение двух условий: (1) каждая часть из $G—H$ должна укладываться по крайней мере в одну грань; (2) каждой части можно назначить инцидентную ей грань таким образом, чтобы никаким двум

частям не была одновременно назначена та грань, относительно которой они препятствуют друг другу. Если в задаче о продолжении существует часть, инцидентная только одной грани (назовем это вынужденным выбором), то мы должны назначить ей именно эту грань. Алгоритм ДМП всегда ищет вынужденный выбор и дает ему приоритет. Каждый раз, когда имеется вынужденный выбор, в соответствующей части произвольным образом выбирается цепь, и она укладывается в соответствующей грани.

Рассмотрим случай, когда все части имеют два выбора. Алгоритм ДМП выбирает произвольную цепь в одной из частей, укладывает ее в одной из возможных граней и продолжает работу таким же образом. Покажем, что эта процедура корректна и не требует возврата к предыдущим шагам. Положим $P_1|P_2$, если $P_1|_F P_2$ для некоторой грани F.

Теорема о безразличии показывает, что все части транзитивного замыкания отношения $|$ инцидентны одной и той же паре граней. Это отношение можно задать графом, если назначать вершину каждой части P, и затем соединять две вершины P и Q ребром, если $P|Q$. Тогда совместимое назначение эквивалентно раскраске этого графа в два цвета. Однако очевидно, что любая раскраска в два цвета является подходящей. И если нет форсированных выборов, то это полностью обосновывает алгоритм ДМП. Этот алгоритм характеризуется квадратичным временем работы.

Представленный здесь алгоритм, если описать его кратко, работает следующим образом. В графе G произвольным образом выбираются два пересекающихся по сегменту цикла из базиса группы циклов G. Эти циклы дают укладку на торе, в которой они существенны и негомологичны. Затем предпринимается попытка расширить полученную укладку до квазипланарной, т. е. до укладки, в которой все замкнутые грани являются просто связными (эти грани не содержат ребер или вершин, повторяющихся на границе). Это сведение описывается в разд. 8, 7, 6. Квазипланарный случай обсуждается в разд. 5. Хотя теорема о безразличии больше не имеет места (необходимо дополнительное условие на укладку h), можно свести задачу о назначении к проблеме о 2-выполнимости формул, в конъюнктивной нормальной форме содержащих не более чем два литерала в дизъюнкте. Полиномиальные алгоритмы для решения последней задачи хорошо известны.

Предварительный вариант [12] этой работы содержит ошибку. Именно, теорема о безразличии как она сформулирована там, неверна, и, следовательно, квазипланарный случай в форме, в которой он там изложен, тоже неверен. Остальная часть работы верна. В данном варианте, кроме того, изложение существенно упрощено.

2. ОБОЗНАЧЕНИЯ И ОПРЕДЕЛЕНИЯ

2.1. Понятие граф имеет как комбинаторный, так и топологический смысл. Мы будем использовать оба смысла. Множество ребер (комбинаторного) графа будем обозначать через $E(G)$, а множество его вершин — через $V(G)$. Мы будем рассматривать только *простые* графы, т. е. графы без петель и кратных ребер. Кроме того, все графы будут конечными, т. е. $|E(G)|$ и $|V(G)|$ (через $|X|$ обозначена мощность множества X) конечны.

В топологическом смысле граф есть CW-комплекс размерности 1 (см., например, Массе [19, с. 190]). Тогда ребра топологического графа есть открытые подмножества, гомеоморфные открытому единичному интервалу вещественной прямой. Каждый комбинаторный граф имеет реализацию как топологический граф, и каждый топологический граф есть комбинаторный граф. Таким образом, существенной разницы между этими двумя понятиями нет, и мы не будем явно указывать, которое из них используется в каждом конкретном случае.

Подграф H графа G — это просто его подкомплекс.

Граф называется *регулярным*, если все его вершины имеют одну и ту же степень, называемую степенью графа. Граф называется *кубическим*, если он регулярный степени 3. Вершина степени 1 называется *листом*. Цепь C, концами которой являются вершины x и y, обозначается C: $x \to y$.

Циклы графа G образуют группу, ранг которой есть цикломатическое число и равен $\nu(G)=|E(G)|-|V(G)|+p$, где p — число компонент связности G.

Два графа G и H *гомеоморфны*, если они гомеоморфны как CW-комплексы. Это топологическое понятие, но существует и чисто комбинаторное определение [2, с. 90].

Пусть H — подграф G. *Части*[1]) G относительно H введены Бонди и Мурти в [2, с. 145], где они названы *мостами*.

2.2. *Укладка* g: $G \to M$ графа G на поверхности M есть гомеоморфизм G в M. Компоненты $M - g(G)$ называются *гранями* или *открытыми* гранями данной укладки. Укладка называется 2-*клеточной*, если каждая грань гомеоморфна открытому кругу. 2-клеточную укладку можно описать комбинаторно, указывая для каждой грани последовательность (ориентированных) ребер на ее границе. В 2-клеточной укладке каждое ребро мо-

[1]) Частями графа G относительно H (частями $G-H$) называются подграфы графа $G-E(H)$, порожденные классами эквивалентности отношения $\sim$ на $E(G)-E(H)$, определяемого следующим образом: $e_1 \sim e_2$, если существует маршрут W такой, что

1) первое и последнее ребро W есть e_1 и e_2 соответственно;

2) все внутренние вершины W не лежат в H. — *Прим. перев.*

жет появляться на границе грани не более чем дважды. Оно может появляться дважды на границе одной и той же грани. Важно делать различие между открытой и замкнутой гранями. Замкнутая грань получается из открытой грани присоединением к ней границы и добавлением к их объединению частичной топологии.

В дальнейшем под укладкой всегда будем понимать 2-клеточную укладку.

Мы будем часто идентифицировать грань F с ее границей ∂F. Это полезно для проведения различия между границей открытой грани и границей замкнутой грани.

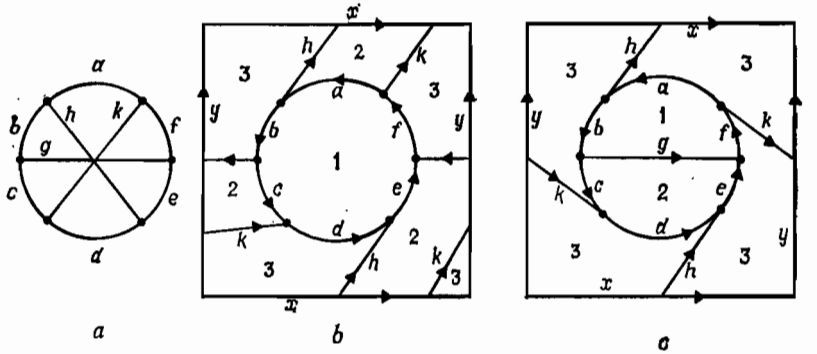


Рис. 1. Две укладки ((*b*) и (*c*)) графа $K_{3,3}$ (*a*) на тор. Ребра, помеченные одной и той же буквой, отождествлены.

Чтобы проиллюстрировать эти понятия, рассмотрим две укладки на рис. 1. Это укладки $K_{3,3}$ (полного двудольного графа на двух множествах, в каждом из которых три элемента) на торе. Комбинаторный граф представлен на рис. 1*a*. Открытые грани укладки на рис. 1*b* не имеют повторяющихся ребер. На рис. 1*c* есть открытая грань, у которой границей является цепь $ahefkdh^{-1}bck^{-1}$, где ориентации ребрам присвоены произвольным образом. Замкнутая грань есть поверхность с краем, полученная отождествлением точек на ребрах h и k, дважды появляющихся в вышеуказанной цепи. В этом случае замкнутая грань есть гомеоморфизм проколотого тора, т. е. то, а, из которого удалена внутренность открытого круга.

Грань, граница которой не имеет повторяющихся ребер или вершин, имеет просто связное замыкание. Повторяющиеся ребра грани, у которой замыкание не является просто связным, будем называть *внутренними*. Все другие ребра назовем *внешними*. 2-клеточную укладку, все грани которой просто связны, будем называть *квазипланарной*.

Если h: $H \to M$ является укладкой подграфа H графа G, и P является частью $G-H$, все прикрепления[1]) которой к H находятся в одной грани i, то мы говорим, что часть P *инцидентна* этой грани.

Следующую систему обозначений будем иногда использовать для 2-клеточной укладки g: $G \to M$. $\alpha_0(g) = V(G)$, $\alpha_1(g) = E(G)$ и $\alpha_2(g)$ обозначает число граней g. Символ g будем опускать, когда ясно, какая укладка имеется в виду. Род укладки $\gamma(g)$ определяется по теореме Эйлера: $\alpha_0 - \alpha_1 + \alpha_2 = 2 - 2\gamma$.

Пусть g: $G \to M$ есть укладка на поверхности M. Образ $g(C)$ цикла C в G называется *существенным*, если $g(C)$ является нестягиваемой.

2.3. Замкнутая грань F, соответствующая открытой грани, есть многообразие с краем. Его род по определению равен роду замкнутой поверхности, полученной наклеиванием кругов на каждый цикл края. Допуская вольность терминологии, назовем это число родом F. Если $\gamma(F) = 0$, то будем говорить, что замкнутая грань *просто связная*.

Грань, не являющаяся просто связной, должна иметь внутренние вершины или ребра. Так как рассматриваются только кубические графы, то легко видеть, что нет изолированных внутренних вершин.

2.4. Хотя понятие укладки является топологическим, существуют и эквивалентные комбинаторные определения (см., например, Уайт [29, с. 61]).

Пусть g: $G \to M$ есть укладка на ориентированную поверхность M. Укладка g порождает циклическую перестановку π_v на окрестности $V(v)$ каждой вершины v (окрестность $V(v)$ вершины v состоит из тех вершин, которые смежны с v). А именно, пусть F — это грань, в которой встречаются ориентированные ребра (u, v) и (v, u'). Тогда $\pi_v(u) = u'$. Например, в укладке на рис. 1*b* π_v задается циклом $(1', 3', 2')$.

Можно показать, что и, наоборот, множество циклических перестановок на окружениях вершин графа определяет укладку, у которой порожденные циклические перестановки совпадают с исходными. Схему доказательства см. в работе Уайта [29, с. 61].

Мы будем широко использовать это утверждение, так как оно позволяет нам решать задачу с комбинаторной точки зрения, в частности представлять укладки структурой данных. Всегда будет подразумеваться, что укладка представлена мно-

[1]) Прикреплениями части P к H называются вершины из $V(P) \cap V(H)$.— *Прим. перев.*

жеством таких циклических перестановок. Циклические перестановки иногда в литературе называются *вращениями*.

Для данной комбинаторной укладки этого типа можно немедленно найти грани укладки. Начнем с вершины v, смежной с u. Пусть $u_0 = u$, $u_1 = v$. Положим $u_2 = \pi_v(u) = \pi_{u_1}(u_0)$. Тогда $u_3 = \pi_{u_2}(u_1)$ и т. д. Этот процесс определяет перестановку на множестве вершин и перестановку на множестве ориентированных ребер. Гранями укладки являются циклы циклической декомпозиции второй перестановки. Тогда род можно вычислить по формуле Эйлера.

2.5. Ниже алгоритмы будут описываться неформально. Мы не будем придерживаться какой-либо конкретной модели вычислений. Получение наиболее эффективной реализации не является главной целью настоящей работы. Поэтому важные вопросы, связанные с выбором наилучшей структуры данных или наилучшего способа реализации различных подпрограмм, здесь не рассматриваются.

Мы находим полезным использовать слово «угадывание» для систематического изучения всех возможностей данного множества. «Угадывание» не означает, что алгоритм не детерминирован, это просто удобный способ представления алгоритмов. Область «угадывания» обозначается помечиванием всех шагов внутри области той же группой символов как утверждение «угадывания». Таким образом, если «угадывание» имеет метку «*abc*», то утверждения в ее области будут помечаться «*abc·a*», «*abc·b*» и т. д.

3. ЗАДАЧИ ОБ ОБОБЩЕННОЙ РАСКРАСКЕ

3.1. Введем теперь один класс задач, который будем называть задачами об обобщенной раскраске. Обычные задачи о раскраске графов являются частным случаем этих более общих задач.

Начнем с определения помеченного графа. Помеченный граф $\mathcal{G}$ есть пара (G, L), состоящая из неориентированного графа G и определенной на $V(G) \cup E(G)$ функции L, удовлетворяющей следующим условиям:

(i) L приписывает конечное множество меток каждому элементу из его области определения;

(ii) для каждого ребра $e = (u, v)$ графа G $L(e) \subset L(u) \times L(v)$.

Определенную на $V(G)$ функцию φ, такую, что $\varphi(v) \in L(v)$ для любого $v \in V(G)$, будем называть *назначением*. Назначение φ называется допустимым, если для любого ребра $e = (u, v)$ из G $(\varphi(u), \varphi(v)) \notin L(e)$.

Задача об обобщенной раскраске состоит в определении того, имеет ли помеченный граф $\mathcal{G}$ допустимое назначение.

Легко видеть, что обычная задача о раскраске вершин графа G в k цветов таким образом, чтобы не было смежных вершин, получивших один и тот же цвет, есть частный случай нашей обобщенной задачи. А именно, достаточно взять $L(v) = \{1, 2, \ldots, k\}$ и $L(e) = \{(i, i) \mid i = 1, 2, \ldots, k\}$. Получаемое допустимое назначение для графа, помеченного таким образом, есть обычная k-раскраска.

3.2. Легко видеть, что общая задача определения того, имеет ли помеченный граф $\mathcal{G}$ допустимое назначение, NP-полна, так как она включает в себя обычную задачу о раскраске.

Однако задача об обобщенной раскраске для помеченных графов $\mathcal{G} = (G, L)$, имеющих не более чем две метки на одной вершине, полиномиально эквивалентна задаче о 2-выполнимости для пропозициональных формул в конъюнктивной нормальной форме.

Теорема 3.1. *Задача об обобщенной раскраске для помеченных графов, имеющих не более чем две метки на одной вершине, и задача о выполнимости пропозиционных формул в конъюнктивной нормальной форме, имеющих не более чем два литерала на дизъюнкт, полиномиально эквивалентны.*

Доказательство. Пусть $\mathcal{G} = (G, L)$ — помеченный граф. Сопоставим $\mathcal{G}$ формулу φ следующим образом. Каждой вершине $v \in V(G)$ сопоставляется переменная, которую будем обозначать так же v. Предположим, что множества $L(v)$ меток, сопоставленных вершинам, упорядочены. В результате $L(v) = \{L_v^0, L_v^1\}$, где эти два элемента не обязательно различны.

Если v — пропозициональная переменная, то $\bar{v}$ обозначает ее отрицание. И пусть $v^0 = \bar{v}$ и $v^1 = v$.

Формула φ будет конъюнкцией дизъюнкций D, которые сопоставлены каждой метке $(L_u^i, L_v^j) \in L(e)$, где $e = (u, v)$ — ребро G. А именно, $D = u^{1-i} V v^{1-j}$ $(i, j = 0, 1)$.

Легко видеть, что φ выполнима тогда и только тогда, когда $\mathcal{G}$ имеет допустимое назначение, и что φ можно получить из $\mathcal{G}$ за полиномиальное время.

И наоборот, пусть φ — формула, имеющая не более чем два литерала в дизъюнкте. Строим помеченный граф $\mathcal{G}$ следующим образом: каждой переменной v из φ сопоставляется вершина G, и пусть $L(v) = \{0, 1\}$. Если $C = u^i V v^j$ $(i, j, = 0, 1)$ есть дизъюнкт φ, то $e = (u, v)$ будет ребром в G, и $(1 - i, 1 - j)$ войдет в $L(e)$. Формула φ, полученная таким образом, выполнима тогда и только тогда, когда G имеет допустимое назначение. Становится

ясным, что φ может быть получено из $\mathcal{G}$ за время, ограниченное полиномом от длины φ.

3.3. Задача о выполнимости КНФ, имеющей не более чем два литерала на дизъюнкт, как легко видеть, разрешима за время, являющееся полиномом от длины формулы (схему соответствующего алгоритма см. в разд. 3.4). Действительно, известны алгоритмы, решающие ее за линейное время (ср. с работой Ивена и др. [10]). Анализируя доказательство теоремы 3.1, легко установить, что при выборе подходящего представления обе редукции линейны относительно длины представлений $\mathcal{G}$ и φ соответственно. Поэтому задача об обобщенной 2-раскрашиваемости, несомненно, разрешима за время, полиномиально зависящее от длины представлений помеченного графа. Более того, если это представление выбрать аккуратно, то задача об обобщенной 2-раскрашиваемости будет разрешимой за линейное время.

3.4. В приложениях задачи об обобщенной раскраске к задачам укладки полезно уметь строить допустимое назначение, когда оно существует.

Так как переход от раскраски к выполнимости довольно прост, то достаточно показать, как строить назначение, выполняющее КНФ, имеющую два литерала на дизъюнкт (если такое назначение существует).

Пусть φ — такая формула. Каждый дизъюнкт из φ имеет вид $x \vee y$ или x, где x и y — это переменные с отрицаниями или без них. Метод для решения задачи о выполнимости состоит в применении *правила сечений* (еще известного как правило резолюций) к двум расходящимся[1]) дизъюнктам до тех пор, пока не перестанут получаться новые дизъюнкты. Первоначальная формула выполнима тогда и только тогда, когда не был получен пустой дизъюнкт. Правило сечений таково:

$$\frac{x \vee C \quad \bar{x} \vee D}{C \vee D},$$

где x — переменная и C и D — две дизъюнкции. Два дизъюнкта называются *расходящимися*, если существует только один литерал, входящий в них как с отрицанием, так и без него.

Если имеется n переменных и дизъюнкты содержат не более двух литералов на дизъюнкт, то существует только $2\left(n + 2\binom{n}{2}\right)$ различных дизъюнктов. Всякий раз, когда правило сечений применяется к двум коротким (т. е. содержащим один или два литерала) дизъюнктам, получается короткий

[1]) clashing. — *Прим. перев.*

дизъюнкт. Таким образом, имеем полиномиально ограниченный алгоритм для 2-выполнимости.

Назначение v, выполняющее φ, можно получить непосредственно из доказательства выполнимости φ. Пусть P есть множество всех дизъюнктов, полученных во время доказательства. Выберем из P однолитеральный дизъюнкт C. Так как P замкнуто относительно правила сечений и пустой дизъюнкт не входит в P, то и отрицание C не входит в P. Положим в назначении $v(x)=0$, если $C=\bar{x}$, и $v(x)=1$, если $C=x$. Теперь вычеркнем из P все дизъюнкты, содержащие C, и удалим $\bar{C}$ из всех оставшихся дизъюнктов. В результате получаем множество дизъюнктов P', не содержащее переменной x, и оно, как легко показать, замкнуто относительно правила сечений. Далее все повторяется.

Подобный анализ может быть сделан для линейных алгоритмов, описанных Ивеном и др. [10].

4. ЗАДАЧА О ПРОДОЛЖЕНИИ И АЛГОРИТМ УКЛАДКИ

4.1. Пусть G — граф, уложенный на торе, и пусть T есть фиксированное остовное дерево G. T определяет базис для группы циклов G. Элемент базиса — это простой цикл, образованный ребром в $G-T$ и единственной цепью в T, соединяющей концы этого ребра.

Предположим теперь, что G имеет 2-клеточную укладку g на торе. Первая группа гомологии этой укладки изоморфна первой группе гомологии тора и, следовательно, свободной абелевой группе ранга 2. Поэтому в базисе циклов должны существовать два цикла C_1 и C_2, укладки которых имеют классы гомологии, порождающие группу гомологии. Очевидно, такие циклы имеют непустое пересечение.

В алгоритме проводится последовательный перебор всех пар таких пересекающихся циклов, и они укладываются на торе так, чтобы была получена 2-клеточная укладка. Так как граф кубический, то подграф, состоящий из двух выбранных пересекающихся циклов, гомеоморфен графу, показанному на рис. 5. Вследствие очевидной симметрии этот граф, по существу, имеет только одну укладку, изображенную на рис. 6; в ней имеется только одна грань, не являющаяся просто связной.

Чтобы определить, имеет ли G укладку на торе, достаточно теперь определить, можно ли эту укладку продолжить на оставшуюся часть графа G.

Это приводит нас к введению следующего определения. Пусть h: $H \to M$ есть укладка подграфа H графа G на поверхности M. Задача о продолжении для тройки (G, H, h) состоит в определении того, существует ли для h продолжение g: $G \to M$.

Как и задача укладки, задача о продолжении допускает и полностью комбинаторную формулировку. А именно, для каждой вершины из G отыскиваются такие циклические перестановки, которые порождают для каждой вершины из H цикли-

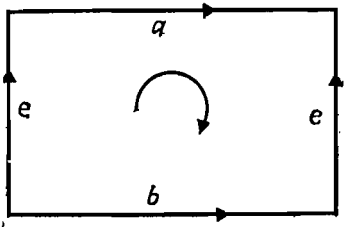


Рис. 2. «Цилиндрическая» грань.

ческие перестановки из h. Кроме того, род g должен быть равен роду h.

Для наших целей не требуется решать общую задачу о продолжении. Вначале необходимо рассмотреть грани типа, указанного на рис. 6 (случай 4 ниже). Но рассмотрения только

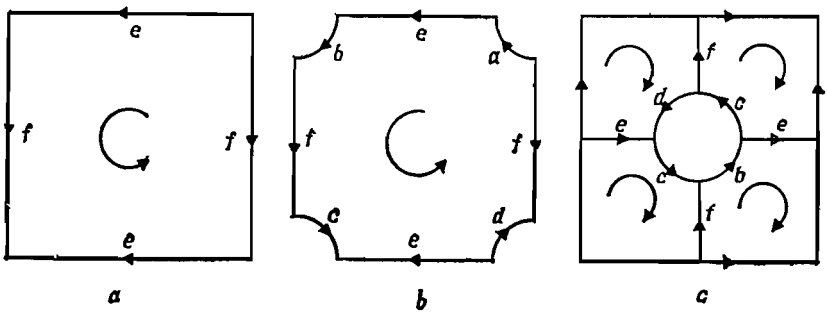


Рис. 3. Проколотые торы.

этого типа граней будет недостаточно. Нам придется разбить типы граней на следующие категории:

(a) просто связные грани;

(b) грани, граница которых является полиномом вида $ae^{-1}b^{-1}e$, где цепи a и b не пересекаются. Такие грани назовем *цилиндрическими* (рис. 2). Они имеют только одну внутреннюю цепь и не являются просто связными, хотя их род равен 0;

(c) грани с двумя внутренними цепями. Они могут быть двух типов: как на рис. 3*a*, т. е. иметь только внутренние цепи, или как на рис. 3*b* и 3*c*. Эти грани будем называть иногда «проколотыми торами»;

(d) грани с тремя внутренними цепями. Как в случае (c), они могут либо иметь (рис. 4*b*), либо не иметь внешних ребер (рис. 4*a*).

Соответственно в задаче о продолжении нужно рассмотреть четыре случая.

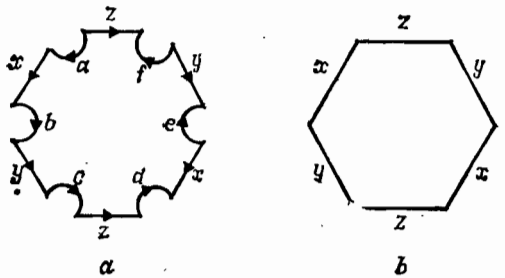


Рис. 4.

Случай 1. Квазипланарный случай, когда укладка h квазипланарна.

Случай 2. Все грани h типов (a) или (b).

Случай 3. Все грани h типов (a), (b) или (c).

Случай 4. Все грани h типов (a), (b), (c) или (d).

В разделах 5, 6, 7 и 8 будет доказана следующая теорема:

Теорема 4.1. *Случаи* 1—4 *задачи о продолжении разрешимы за время, ограниченное полиномом от* $|V(G)|$. *Именно, время работы ограничено полиномом шестой степени от* $|V(G)|$, *коэффициенты которого ограничены числом* $2^{l(s)}$, *где* l — *линейная функция от числа* s *граней* h, *не являющихся просто связными.*

4.2. *Алгоритм укладки кубического графа на торе.*

Ввод: кубический граф G, заданный его матрицей смежности.

Вывод: «нет», если граф нельзя уложить на торе. В противном случае укладка G, заданная указанием вращений в каждой вершине.

(a) Построение остовного дерева T для G.

(b) Угадывание (неупорядоченной) пары пересекающихся циклов, определенных T и ребрами в $G—T$. Если все такие пары исчерпаны, то граф не является тороидальным: выход «нет» и останов.

(b, a) Для выбранной пары пересекающихся циклов выбирается укладка на тор, такая, что оба цикла существенны (т. е. негомологичны нулю). Если укладку нельзя продолжить, переходим к шагу (b). Если укладка может быть продолжена, то выдаем ее в качестве результата.

Ясно, что время работы этого алгоритма определяется временем работы алгоритма продолжения, используемого в шаге (b.a), и числом выборов, которые должны быть рассмотрены в шаге (b). Число выборов ограничено числом $(\alpha_1 - \alpha_0 + 1)$, так как ранг группы циклов G равен $\alpha_1 - \alpha_0 + 1$. Это число в свою очередь ограничено квадратичным полиномом от α_0. Алгоритм продолжения в (b.a) полиномиален, так как только одна грань не является просто связной.

В [12] был использован незначительно отличающийся подход: первой работала модифицированная версия алгоритма установления планарности. Если граф непланарен, алгоритм возвращается к запрещенному подграфу Понтрягина — Куратовского, который может быть гомеоморфен только графу $K_{3,3}$, так как G — кубический граф. Граф $K_{3,3}$ имеет только две укладки на торе. Одна из них квазипланарна, а другая — типа рассмотренного в случае 3 алгоритма продолжения. Таким образом, благодаря использованию более изощренной версии алгоритма установления планарности отпадает необходимость в рассмотрении случая 4 алгоритма продолжения. Мы предпочли метод, описанный здесь, по следующим причинам. Во-первых, отметим, что теорема Понтрягина — Куратовского не является необходимой для алгоритма укладки. Это хотя и не является решающим для укладки на торе, но жизненно важно для укладки на поверхности большого рода, для которых соответствующие обобщения теоремы Понтрягина — Куратовского неизвестны. Во-вторых, алгоритмы установления планарности, возвращающиеся к запрещенным подграфам Понтрягина — Куратовского, представляют и самостоятельный интерес. Мы обсудим их в отдельной работе.

5. КВАЗИПЛАНАРНОСТЬ

5.1. Грани квазипланарной укладки просто связны. Поэтому любая цепь, связывающая две точки на границе грани, может быть уложена на эту грань с точностью до гомологии только одним способом. Комбинаторно это означает, что существует только один способ так продолжить вращения в концевых точках цепи, чтобы в полученном продолжении эта грань расщепилась на две различные грани. Последнее значительно облегчает задачу построения эффективного алгоритма нахождения продолжения. Рассматриваемый алгоритм является обобщением алгоритма нахождения планарного продолжения, предложенного в [5].

Пусть H есть подгомеоморф графа G и $h\colon H \to M$ — квазипланарная укладка H на поверхности M. В этом разделе будем предполагать, что граф G кубический и H не имеет листьев

(т. е. вершин степени 1). Отсюда следует, что каждое прикрепление части G относительно H отождествляется в G с внутренней точкой некоторого ребра из H.

Прикрепления части P к просто связной грани F в h определяют некоторое количество непересекающихся по ребрам цепей, называемых сегментами F, определенными P. Говорят, что две части P_1 и P_2, инцидентные одной и той же грани F, избегают друг друга, если все прикрепления одной части лежат в одном сегменте, определенном на F другой частью. Пусть при этих условиях g: $G \to M$ есть продолжение h. Тогда:

Теорема 5.1 [2, теорема 9.8]. *Пусть P_1 и P_2 — две части G относительно H, такие, что $g(P_1)$, $g(P_2) \subset F$ для некоторой просто связной грани F из h. Тогда P_1 и P_2 избегают друг друга.*

Доказательство. Доказательство Бонди и Мурти [2] приложимо дословно. Оно основано на том факте, что F просто связно.

Предложение 5.2. *Часть P графа G относительно H инцидентна не более чем двум граням h.*

Доказательство. Так как G — кубический граф, то каждое прикрепление P отождествляется в G с внутренней точкой некоторого ребра из H. Такая точка, однако, лежит на границе ровно двух граней h (так как h квазипланарна), и, следовательно, часть P не может быть инцидентна более чем двум граням (напомним, что P называется инцидентной грани F, если все ее прикрепления лежат на границе F).

Части, не избегающие друг друга относительно некоторой грани, называются *препятствующими* друг другу относительно этой грани. Это понятие имеет удобную комбинаторную характеризацию.

Говорят, что две части P_1 и P_2, инцидентные просто связной грани F, *перекрещены*, если P_1 имеет прикрепления a_1, b_1, а P_2 — прикрепления a_2, b_2, появляющиеся на F в следующем установленном порядке $a_1a_2b_1b_2$.

Предложение 5.3 [2, теорема 9.6]. *Если две части P_1 и P_2 препятствуют друг другу относительно квазипланарной грани F, то они или перекрещены, или имеют три общих прикрепления к F.*

Доказательство. Доказательство (см. [2]) воспроизводиться не будет. Оно опирается только на простую связность F.

Предложение 5.4 [2, теорема 9.7]. Если часть P имеет прикрепления v_1, v_2, v_3 к ∂F, то существуют вершина v_0 и цепи p_1: $v_0 \to v_1$, p_2: $v_0 \to v_2$, p_3: $v_0 \to v_3$ такие, что любые две из них пересекаются только в вершине v_0.

5.2. Мы теперь покажем, что квазипланарная задача о продолжении может быть сведена к задаче об обобщенной раскраске из разд. 3.

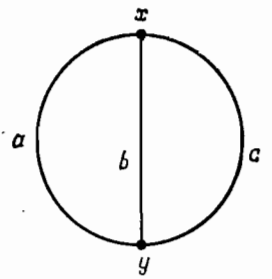


Рис. 5.

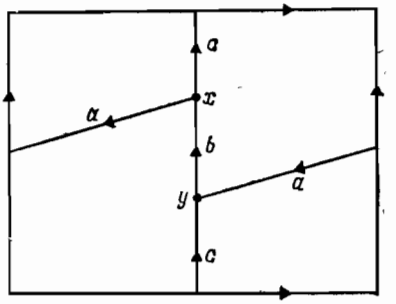


Рис. 6.

Сопоставим тройке (G, H, h) помеченный граф $\mathcal{G}^* = (G^*, L^*)$ следующим образом. Непомеченный граф G^* имеет в качестве вершин части $G—H$. Две части P_1 и P_2 соединены ребром в G^*, если они обе инцидентны одной и той же грани F из h и допускают укладку на F, но препятствуют друг другу относительно F. Пусть для вершины P из G^* $L(P)$ есть множество граней F, на которые P может быть уложена. Пусть для ребра $e = (P_1, P_2)$ из G^* выполняется равенство $L(e) = L(P_1, P_2) = \{(F, F) \mid P_1$ и P_2 препятствуют друг другу относительно $F\}$.

Пример. $H = K_{3,3}$, а h есть укладка $K_{3,3}$ на торе, изображенная на рис. 7. Части G относительно H показаны на рис. 7.

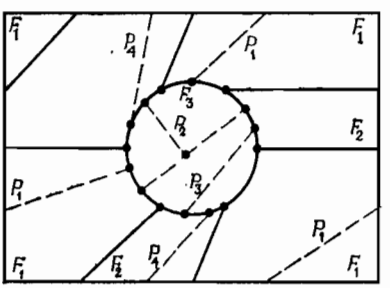


Рис. 7.

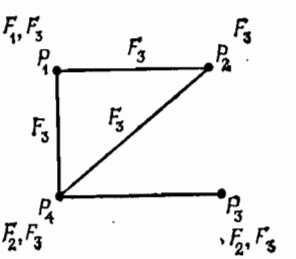


Рис. 8.

штриховыми линиями. Части, изображенные на рис. 7, уложены на грани h, давая тем самым продолжение h на G.

Граф совместимости показан на рис. 8. Отметим, что в нашем случае все метки ребер имеют вид (F, F). Поэтому мы можем обозначать такую метку только номером одной из двух компонент.

Две части могут допускать укладку на одно и то же множество граней, но быть несовместимыми только относительно некоторого подмножества этого множества. В этом примере P_3 и P_4 обе допускают укладки на F_2 и F_3, но препятствуют друг другу только относительно F_3.

Предложение 5.5. Укладка h имеет продолжение на G тогда и только тогда, когда помеченный граф $\mathcal{G}$, сопоставленный тройке (G, H, h), имеет допустимое назначение.

5.3. Алгоритм продолжения для квазипланарного случая.

Ввод: Кубический граф G, подграф H графа G, не содержащий листьев, квазипланарная укладка h: $H \to M$.

Алгоритм определяет, существует ли продолжение h на G, и если существует, то строит его.

(a) Находим все части G относительно H.

(b) Для каждой части P определяем все грани, которым она инцидентна.

(c) Для каждой грани P, используя планарный алгоритм установления планарности, определяем грани, на которые укладывается P. Для каждой такой грани строим укладку P на этой грани. Если ни одной такой грани нет, останов: укладка h не имеет продолжения.

(d) Строим помеченный граф $\mathcal{G}^*$, сопоставленный тройке (G, H, h).

(e) Строим формулу в конъюнктивной нормальной форме, сопоставленную $\mathcal{G}^*$.

(f) Определяем, выполнима или нет формула, и если она выполнима, то определяем назначение, выполняющее ее. Если формула невыполнима, то укладка не имеет продолжения.

(g) Строим продолжение h, используя назначение, построенное на шаге (f), и частичные укладки, построенные на шаге (c).

Мы установили случай 1 теоремы 4.1.

5.4. Квазипланарный алгоритм расширения может быть еще использован для случая произвольных укладок, не имеющих частей H относительно G, прикрепленных к внешнему ребру укладки h.

Рассмотрим укладку h: $H \to M$ с этим свойством и предположим, что h может быть продолжена до укладки g: $G \to M$. Пусть P есть часть G относительно H, и пусть $g(P) \subset F$, где F есть грань, замыкание которой не является просто связным. Так как G — кубический граф, то все прикрепления P есть точки, расположенные во внутренности ребра из подграфа H, лежащего на границе замыкания грани F. Так как P не имеет прикреплений к внутренним ребрам F, то существует замкнутая окрестность $V \subset F$ объединения внутренних ребер F такая, что

$P \subset F - V$. Окрестность V можно на самом деле выбрать так, что $F - V$ будет гомеоморфно кругу. Это показывает, что сужение g на P есть планарная укладка P и что поэтому можно предположить, что все грани в действительности просто связны.

Теорема 5.6. *Задача о продолжении* (G, H, h), *где* h: $H \to M$ *есть укладка, обладающая тем свойством, что не существует частей* G *относительно* H, *прикрепленных к внутреннему ребру грани из* h, *разрешима за время, ограниченное полиномом от числа ребер в* $G - H$. *Если же* h *имеет продолжение на* G, *то такое продолжение алгоритм будет подвергать дальнейшему анализу.*

6. СЛУЧАЙ 2 ЗАДАЧИ ПРОДОЛЖЕНИЯ

6.1. Здесь мы имеем просто связные грани и еще грани, которые мы назвали *цилиндрами*. Цилиндры гомеоморфны грани, показанной на рис. 2, и будут иногда называться *особыми* гранями.

Этот случай не сводится прямо к 2-выполнимости пропозициональной формулы, как было в квазипланарном случае. Трудность в основном заключается в том обстоятельстве, что цепь,

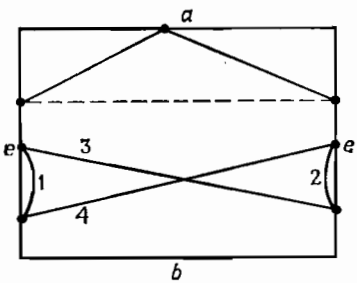


Рис. 9.

инцидентная цилиндрической грани, может позволять четыре различные укладки на грани. Например, это имеет место для цепи, оба конца которой лежат на внутренней цепи e (рис. 9). Если цепь имеет только один конец на e, то имеются только две различные укладки.

Мы можем свести задачу продолжения в этом случае к задаче выполнимости пропозициональных формул. Однако, так как при прямом сведении мы не можем больше гарантировать, что в дизъюнкте не более двух литералов, нам не удается установить полиномиальность алгоритма. Покажем в этом разделе, как такое затруднение можно обойти. Задача будет сведена к ограниченному числу задач о 2-выполнимости.

Введем и будем пользоваться следующими понятиями. Всякий раз, когда рассматривается какая-либо особая грань, мы будем обозначать экстремальные цепи через a и b соответственно. Внутреннюю цепь всегда будем обозначать через e.

Две копии e на внутренней цепи будем называть *левой* и *правой* внутренней цепью соответственно. Таким образом, мы имеем рис. 2 в качестве модельного для введенных обозначений и понятий.

Части $G-H$, прикрепленные к внутреннему ребру, будем называть *особыми*. Остальные — *обыкновенными*. Цепь, имеющая одно прикрепление на цепи x и другое на цепи y, будет называться $(x-y)$-цепью. Укладку, соединяющую левую и правую внутренние цепи цилиндра, назовем укладкой *поперек* грани (например, укладки 3 и 4 на рис. 9). Если все прикрепления части находятся на *правой* (*левой*) внутренней цепи, укладку назовем *правой* (*левой*). Левая и правая укладки будут называться односторонними. Если часть является особой, то либо ее укладка односторонняя, либо часть содержит цепь, укладывающуюся поперек грани. В последнем случае будем говорить, что часть уложена поперек цилиндра.

Изучим теперь отношение «препятствования» для цилиндров. Две укладки частей на грань препятствуют друг другу, если они пересекаются. Как было показано в разд. 5.3, это топологическое понятие может быть охарактеризовано и комбинаторно. Препятствуют ли две укладки друг другу, зависит только от прикреплений частей к внутренним цепям.

Интересен случай обыкновенной части P', прикрепленной только к одной из двух внешних цепей, и особой части P''. Если одна укладка P'' на цилиндр F препятствует укладке P', то и все укладки P'' тоже. Это легко увидеть геометрически, заключив P' в достаточно узкую полосу вдоль того ребра, к которому она прикреплена. Таким образом, в этом случае мы можем расширить понятие препятствования на части.

6.2. В продолжении g: $G \to M$ укладки h цилиндры из h могут быть следующих двух видов:

(i) *односторонние*, т. е. получившие только односторонние укладки;

(ii) *двусторонние*, т. е. получившие по крайней мере одну часть, уложенную поперек цилиндра.

И наоборот, покажем теперь, что если предположить, что задана классификация цилиндров на односторонние и двусторонние, то за полиномиальное время проверяется, возможно ли продолжение h на G относительно этой классификации.

В продолжении h для частей $G-H$ существуют следующие выборы возможной укладки:

(a) Если часть обыкновенная, то она может быть уложена самое большее на одну из двух инцидентных ей граней. Однако такая часть не может быть уложена на двусторонний цилиндр, когда она прикреплена к обеим внешним цепям его.

(b) Если часть особая и если она инцидентна одностороннему цилиндру, то она может получить как левую, так и правую укладку.

(c) Если часть особая и если она инцидентна двустороннему цилиндру, то можно рассмотреть только один выбор, а именно укладку на этот цилиндр.

Пусть P и Q — две части $G-H$, и пусть c — это выбор укладки для P и d — выбор для Q. Мы будем говорить, что c и d препятствуют друг другу, если в заданной ими укладке P и Q пересекаются. Как и раньше, это топологическое определение имеет комбинаторную характеризацию и основано только на прикреплениях частей. Предположим для примера, что P — обыкновенная часть, инцидентная цилиндру F, и что Q — особая часть, инцидентная F. Тогда P имеет выбор F и Q имеет выбор «правая». Чтобы определить, препятствуют ли друг другу выбор F и выбор «правая», достаточно определить, имеет ли P два прикрепления, перекрещивающиеся с двумя прикреплениями Q к правой внешней цепи. Случай обыкновенной части, имеющей выбор типа (a), и особой части, имеющей выбор типа (c), обсуждался в конце предыдущего раздела. Аналогично могут быть рассмотрены и другие случаи.

Сопоставим задаче продолжения (G, H, h) (с заданной классификацией цилиндров h на односторонние и двусторонние) задачу об обобщенной раскраске $\mathcal{G}=(G, L)$, рассмотренную в разд. 3. Положим $V(G)=\{P \mid P$ часть $G-H\}$, $E(G)=\{(P, Q) \mid P, Q \in V(G)\}$. Тогда G — полный граф на частях $G-H$. Для $P \in V(G)$ пусть множество выборов, допустимых для P, есть $L(P)$. Для $(P, Q) \in E(G)$ пусть $L(P, Q)=\{(c, d) \mid c \in L(P), d \in L(Q)$, выборы c для P и d для Q препятствуют друг другу$\}$.

Легко видеть, что если h имеет продолжение на G относительно данной классификации цилиндров h, то задача раскраски $\mathcal{G}$ разрешима.

Строение $\mathcal{G}$ гарантирует, что $L(P)$ имеет не более двух элементов. Поэтому задача об обобщенной раскраске разрешима за полиномиальное время (ср. с разд. 3). Кроме того, граф совместимости $\mathcal{G}$ может быть построен за полиномиальное время относительно длины записи G, H и h.

6.3. Наоборот, предположим, что $\mathcal{G}$ допускает решение. Этого недостаточно для того, чтобы h имела продолжение относительно заданной классификации цилиндров. Необходимо еще проверить,

что все особые части, инцидентные двусторонним цилиндрам, укладываются на них одновременно. Перейдем теперь к рассмотрению этой задачи.

Ясно, что можно предположить, что существует только один такой цилиндр F, все части $G - H$ инцидентны F и обыкновенных частей нет. Граница ∂F грани F тогда гомеоморфна графу, представленному на рис. 10, и F гомеоморфна неограниченной области, определяемой планарной укладкой ∂F.

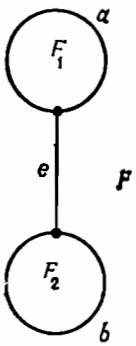


Рис. 10.

Любая укладка особых частей на F определяет планарную укладку $G - H$. Наоборот, любая планарная укладка G определяет укладку h для H и укладку особых частей в неограниченную область h. Таким образом, мы свели нашу задачу к задаче об установлении планарности G.

6.4. *Алгоритм продолжения* (случай 2).

Ввод: Кубический связный граф G, связный подграф H графа G, укладка h: $H \to M$ только с просто связными гранями или гранями, гомеоморфными 2-клетке, граница которой есть полигон вида $ae^{-1}b^{-1}e$ (рис. 2).

Вывод: Продолжение h на G, если оно существует, и отрицательный ответ — в противном случае.

(a) Находим части $G - H$. Определяем h для каждой части грани, которым она инцидентна.

(b) Разбиваем грани h на просто связные и цилиндры.

(c) Угадываем классификацию цилиндров h на односторонние и двусторонние.

(c.a) Строим задачу об обобщенной раскраске для угадывания в (c).

(c.b) Определяем, имеет ли $\mathcal{G}$ допустимое назначение φ. Если да, то строим его. Если нет, то переходим к (c).

(c.c) Определяем для каждой части P, имеет ли она укладку типа $\varphi(P)$. Если нет, то переходим к (c). Иначе, строим для каждой части P укладку типа $\varphi(P)$.

(c.d) Для каждого двустороннего цилиндра F определяем (используя алгоритм установления планарности) укладку на F подграфа, образованного всеми особыми частями, инцидентными F. Если для некоторой грани F это невозможно, то переходим к (c).

(c.e) Строим продолжение укладки h следующим образом:

Для каждой части P, такой, что $\varphi(P)$ не является укладкой на двусторонний цилиндр, продолжаем h, укладывая части как в (c.c). Продолжаем укладку для таких частей P, что $\varphi(P)$

есть укладка на двусторонний цилиндр, укладками, построенными в (c.d).

6.5. *Оценка времени работы алгоритма*

Все шаги этого алгоритма, за исключением шага (c), могут быть выполнены за $O(v^2)$ шагов, где $v = \alpha_0(G)$. Шаг (c) требует $O(2^s)$ шагов, где s — число цилиндров. Таким образом, время работы ограничено $p_1(v) + 2^s p_2(v)$, где p_1 и p_2 — квадратичные полиномы. Случай 2 теоремы 4.1 доказан.

7. СЛУЧАЙ 3 АЛГОРИТМА ПРОДОЛЖЕНИЯ

7.1. Терминология этого раздела аналогична терминологии разд. 6. Грани, гомеоморфные грани, представленной на рис. 3, будем называть *особыми*. *Обыкновенные* грани и части — это грани и части, не являющиеся особыми. Всякий раз, когда рассматривается конкретная особая грань, будем использовать обозначения, приведенные на рис. 3. Не нарушая общности, можно предположить, что все особые грани имеют вид как на рис. 3*a*, поскольку грани на рис. 3*b* — это частный случай.

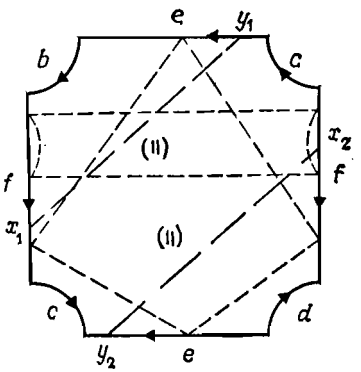


Рис. 11.

Пусть F — особая грань. Две копии внутренней цепи e пометим e и e^{-1}. Аналогично для внутренней цепи f.

Цепь, соединяющая две точки на одном и том же внутреннем ребре, имеет четыре различных укладки на особую грань (рис. 11). Две из них делят грань на два цилиндра. Будем говорить в этом случае, что цепь уложена *поперек* F.

Как и в разд. 6, пусть g: $G \to M$ является продолжением h: $H \to M$. Грань из h назовем *расщепленной*, если все ее подграни в g либо просто связны, либо являются цилиндрами.

Если особая грань F не является расщепленной, то она содержит подгрань g типа (c), определенной в разд. 3. Такая грань может быть расщеплена присоединением к G нового ребра, соединяющего две точки одного и того же внутреннего ребра, и укладкой его поперек F.

Будем говорить, что внутренние цепи e и e^{-1} разделены, если не существует двух таких точек на e и e^{-1}, принадлежащих одной и той же грани g. Легко видеть, что e и e^{-1} могут быть разделены одним из трех способов (рис.11):

(i) одна цепь C: $x \to y$, где x — точка на bfc и y — точка на $df^{-1}a$;

(ii) две цепи C_1: $x_1 \to y_1$, C_2: $x_2 \to y_2$, x_1 — точка на bfc, y_1 — на e, x_2 — на $df^{-1}a$ и y_2 — на e^{-1};

(iii) две цепи C_1: $x_1 \to y_1$, C: $x_2 \to y_2$, где x_1 — точка на $df^{-1}a$, y_1 — на e, x_2 — на bfc и y_2 — на e^{-1}.

В случаях (ii) и (iii) будем говорить, что цепи C_1 и C_2 *перекрываются* (относительно e и e^{-1}). Цепи C_1 и C_2 должны принадлежать особым частям, так как y_1 и y_2 — это точки внутренней цепи.

Эти понятия могут быть распространены с цепей на пары точек. Пара точек (x, y) из ∂F разделяет e и e^{-1}, если цепь C: $x \to y$, уложенная на F, разделяет e и e^{-1}. Аналогично две пары (x_1, y_1) и $(x_2 y_2)$ разделяют e и e^{-1}, если цепи C_1: $x_1 \to y_1$ и C_2: $x_2 \to y_2$, уложенные на F, разделяют e и e^{-1}.

Если особая грань F расщеплена, то либо e и e^{-1} разделены, либо разделены f и f^{-1}. В алгоритме мы должны будем стремиться продолжить укладку h: $H \to M$ при ограничении, что некоторые грани расщеплены. Для этого нужно угадывать цепи, которые расщепляют грань. Число таких угадываний может быть слишком большим, так как число возможных цепей может быть экспоненциальным. И если бы алгоритм перебирал все эти возможности, то нельзя было бы гарантировать полиномиальное время его работы. К счастью, если цепь (или пара цепей) разделяет e и e^{-1}, то их разделяет и любая другая цепь (или пара цепей), соединяющая те же самые прикрепления. Таким образом, вместо угадывания цепи (или пары цепей), разделяющих e и e^{-1}, достаточно будет указать пару (или две пары) прикреплений, обладающих этим свойством и соединенных цепью (или двумя непересекающимися цепями) в $G - H$. Теперь число угадываний ограничено величиной $|V(G)|^4$, т. е. полиномом от $|V(G)|$. Однако необходимо еще проверить, что выбранная пара (или пара пар) может быть соединена цепью (или двумя непересекающимися цепями). Для последней задачи известно много эффективных алгоритмов (см., например, [2]).

7.2. *Алгоритм продолжения*

Ввод: Кубический граф G, подграф H графа G, укладка h: $H \to M$ только с просто связными гранями, гомеоморфными 2-клетке, граница которой есть полигон вида $ae^{-1}b^{-1}e$ (рис. 2), и гранями, гомеоморфными 2-клетке, граница которой есть полигон вида $aebfce^{-1}df^{-1}$ (рис. 3).

Вывод: Алгоритм устанавливает, существует ли продолжение h на G и строит его, если оно существует.

(а) Находим части G относительно H и определяем h для каждой части грани, которым она инцидентна.

(b) Разбиваем грани на обыкновенные и особые.

(c) Угадываем классификацию особых граней на расщепленные и нерасщепленные.

(c.a) Для каждой расщепленной грани F:

(c.aa) Угадываем пару прикреплений (u, v) или две пары прикреплений (u_1, v_1) и (u_2, v_2), разделяющие e и e^{-1} или f и f^{-1}.

(c.aa.a) Определяем, содержит ли $G - H$ цепь C: $u \to v$ или две непересекающиеся цепи C_1: $u_1 \to v_1$ и C_2: $u_2 \to v_2$ (используем для этого стандартный алгоритм (см., например, [2])).

(c.aa.b) Укладываем C или C_1 и C_2 на F.

(c.b) Для каждой нерасщепленной грани:

(c.ba) Угадываем пару точек u и v на e и e^{-1} соответственно.

(c.ba.a) Присоединяем ребро C из u в v к G.

(c.ba.b) Укладываем C на F.

(c.c) Пусть H' получен из H присоединением всех новых цепей из (c.aa.a) и (c.ba.a), и пусть h' — укладка H', определенная укладками в (c.aa.b) и (c.ba.b). Действуем для H', h' и G, как в случае 2 алгоритма продолжения.

7.3. *Время работы алгоритма*

Время работы данного алгоритма определяется числом особых граней.

Шаги (a) и (b) требуют не более $|V(G)|^2$ шагов. В шаге (c) осуществляется 2^s угадываний. В шаге (c.a) — не более $|V(G)|^4$ угадываний. Шаг (c.aa.a) требует не более $|V(G)|^2$ шагов. В (c.b) основное время сосредоточено в шаге (c.ba), требующем $|V(G)|^2$ угадываний. И наконец, шаг (c.c) вызывает использование случая 2 алгоритма продолжения. Это в свою очередь требует не более чем $p(|V(G)|)$ шагов, где степень полинома p определяется числом цилиндров. Особая грань может дать самое большее два цилиндра, и, следовательно, в шаге (c.c) создается не более $2s$ новых цилиндров.

Общее время работы ограничено полиномом степени 6 по v. Коэффициенты этого полинома ограничены числом $2^{l(s)}$, где $l(s)$ — линейная функция. Отметим, что s не встречается в показателе у v. Этим полностью доказывается случай 3 теоремы 4.1.

8. СЛУЧАЙ 4 АЛГОРИТМА ПРОДОЛЖЕНИЯ

Обсуждение этого случая почти дословно повторяет обсуждение случая 3.

Назовем особыми грани типа (d) разд. 4. Другие грани будем называть обыкновенными. Разделение двух внутренних цепей, скажем x и x^{-1}, в точности подобно приведенному выше. Единственный выбор в алгоритме появляется на шаге (c.aa), где нужно вместо прикреплений, разделяющих e и e^{-1} или f и f^{-1},

угадывать прикрепления, разделяющие x и x^{-1}, y и y^{-1} или z и z^{-1}. На шаге (с.с) заменяем случай 2 случаем 3.

9. ЗАКЛЮЧИТЕЛЬНЫЕ ЗАМЕЧАНИЯ

Данная работа наводит на мысль о многочисленных обобщениях. Во-первых, существует публикация, обобщающая предлагаемый алгоритм на произвольные графы и произвольные ориентируемые поверхности. Это было недавно сделано Филотти и Миллером в [13]. Хотя в противоположность некоторым ранним предположениям доказано, что задача укладки графа на поверхность заданного рода легко решаема, для задачи о продолжении это уже не так. Простое прямое рассуждение показывает NP-полноту общей задачи о продолжении (Рейф (не опубликовано))[1]). Тот же самый вывод был еще косвенным путем получен Миллером, показавшим, что задача о продолжении P-эквивалентна задаче о раскраске дуговых ребер графов пересечений. NP-полнота последней задачи показана Гэри и Пападимитру (не опубликовано). С другой стороны, если требовать нахождения продолжения для фиксированного рода, то задача снова легко решаема (Филотти (не опубликовано)). Как и в данном случае, полином зависит от рода. Вопрос о нахождении полиномиального алгоритма определения рода остается еще открытым. Мы подозреваем, что это невозможно. Несомненно, методы этой статьи не могут дать такого результата, так как у нас нет достаточно высокой нижней оценки для рода (ср. Уайт [29], с. 58). Решение неориентированного случая кажется этому автору не представляющим серьезных трудностей.

Важная задача, на которую наводит мысль данная работа, — задача об установлении изоморфизма графов фиксированного рода. Очень эффективный алгоритм для установления изоморфизма планарных графов был дан Вейнбергом [28] и был в значительной степени улучшен Хопкрофтом и Тарьяном [18], использовавшими технику поиска в глубину. Задача об обобщении этих методов на другие поверхности была еще поставлена Фонте [11] и, вероятно, должна была привлечь внимание

[1]) В статье комментируется ряд неопубликованных результатов. Из них к настоящему времени опубликованы только три:

1. Filotti I. S., Mayer J. N. A polinomial-time algorithm for determing the isomorhism of graphs of fixed genus. Proc. 12th. Annu. ACM Symp. Theory Comput., 1980, 236—243.
2. Garey M. R., Johnson D. S., Miller G. L. Papadimitriou C. H. The complexity of coloring circular arcs and chords. SIAM J. Algebraic Discrete Methods, 1980, 1, № 1, 216—227.
3. Miller G. L. Isomorphism testing for graphs of bounded genus. Proc. 12 th, Annu. ASM Symp. Theory Comput., 1980, 225—235.

— *Прим. перев.*

многих исследователей. Недавно такой алгоритм был найден одновременно и независимо Филотти и Мейером (не опубликовано) и Миллером (не опубликовано)[1]).

Я очень признателен ряду коллег за полезные предложения. Джонотан Гросс предложил задачу, введшую меня в эту область, и оказывал мне всяческую помощь. Обсуждение работы с Гэри Миллером в конечном счете привело к обобщению этого алгоритма [13]. Идеи свести квазипланарный случай к 2-выполнимости и начать работу алгоритма с циклов в базисе циклов вместо использования алгоритма распознавания планарности были найдены независимо нами обоими. Давид Джонсон предложил ссылку [10]. Джек Мейер и Ласло Вабан сделали ряд важных замечаний.

ЛИТЕРАТУРА

1. Aho A. V., Hopcroft J. E., Ullman J. E. The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., 1974. [Имеется перевод: Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979.]
2. Bondy J. A., Murty U. S. R. Graph Theory with Applications, American Elsevier, New York, 1976.
3. Booth K. S. P-Q-Tree Algorithms, Ph. D. Thesis, University of California, Berkeley, 1975.
4. Chartrand G., Behzad M. Introduction to the Theory of Graphs, Allyn & Bacon, Boston, 1971.
5. Demoucron G., Malgrange Y., Pertuiset R. Graphes Planaires: Reconnaissance et Construction de Représentations Planaires Topologiques, Rev. Francaise Informat. Recherche Opérationnelle 8 (1964), 33—47.
6. Deo N. Note on Hopcroft and Tarjan planarity algorithm. J. Assoc. Comput. Mach. 23 (1976), 74—75.
7. Edmonds J. A combinatorial representation for polyhedral surfaces, Notices Amer. Math. Soc. 7 (1960), 646.
8. Edmonds J. Symmetric imbeddings of complete graphs, Notices Amer. Math. Soc. 7 (1960), 948.
9. Edmonds J. On the surface duality of linear graphs, J. Res. Nat. Bur. Standards Sect. B 69 No. 1, 2, 121—123.
10. Even S., Ital A., Shamir A. On the complexity of timetable and multicomodity flow problems, SIAM J. Comput. 5 (1976), 691—703.
11. Fontet M. Automorphismes de Graphes et Planarité, Astérisque, 38—39 (1976), Société Mathématique de France, 73—90.
12. Filotti I. S. An efficient algorithm for determining whether a cubic graph is toroidal, in Proceedings, Tenth Annual ACM Symposium on the Theory of Computing, pp. 133—142, Association for Computing Machinery, New York, N. Y., 1978.

[1]) Результат автора и Мейера, а также Миллера о полиномиальном решении проблемы изоморфизма для графов ограниченного рода был позднее обобщен в работах Миллера (Miller G. L. Isomorphism of k-contractible graphs A generolisation of bounded valence and bounded genus. Inf. and Contr., 1983, 56, № 1—2, p. 1—20) и Пономаренко (Пономаренко И. Н. Полиномиальный алгоритм изоморфизма для графов, не стягиваемых на $K_{3,g}$. — Зап. науч. семинаров Ленингр. отд. Мат. ин-т АН СССР, 1984, 137, с. 99—114). — *Прим. ред.*

13. Filotti I. S., Miller G., Reif J. On determining the genus of a graph in $O(vOg)$ steps. Proc. 11th. Annu. ACM Symp. Theory Comput., 1979, 27—37.
14. Gross J., Tucker T. Local Maxima in Graded Graphs, Ann. New York Acad. Sci. 13 (1979), 254—257.
15. Giblin P. J. Graphs, Surfaces and Homology: An Introduction to Algebraic Topology. Chapman and Hall, London, 1977.
16. Hefter L. Uber das Problem der Nachbargebiete, Math. Ann. 38 (1891), 477—508.
17. Hopcroft J. E., Tarjan R. E. Efficient planarity testing, J. Assoc. Comp. Mach. 21 (1974), 549—568.
18. Hopcroft J. E., Tarjan R. E. Isomorphism of planar graphs (working paper), in «Complexity of Computations» (Miller et al. Eds.), Plenum, New York, 1972.
19. Massey W. A. Algebraic Topology: An Introduction, Harcourt, Brace and World, New York, 1967.
20. Reingold E. M., Nievergelt J., Deo N. Combinatorial Algorithms: Theory and Practice Prentice-Hall Englewood Cliffs, N. J., 1977. [Имеется перевод: Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы: теория и практика. — М.: Мир, 1980.]
21. Ringel G. Map Color Theorem, Springer-Verlag, Berlin, 1974. [Имеется перевод: Рингель Г. Теорема о раскраске карт. — М.: Мир, 1977.]
22. Ringel G. The combinatorial map color theorem, J. Graph Theory 1 (1977), 141—155.
23. Rubin F. An improved algorithm for testing the planarity ot a graph, IEEE Trans. Computers C-24, No. 2 (1975), 113—121.
24. Spanier E. Algebraic Topology, McGraw-Hill, New York, 1960.
25. Stahl S. The embeddings of a graph — A survey, J. Graph Theory 2 (1978), 275—298.
26. Tutte W. T. A census of planar maps, Canad. J. Math. 15 (1963), 249—271.
27. Walsh T., Lehman A. Counting rooted maps by genus, (I) J. Combinatorial Theory B 13 (1972), (I) 192—218; (II) 122—141.
28. Weinberg L. A simple and efficient algorithm for determining isomorphism of planar triply connected graphs, IEEE Trans. Circuit Theory CT-13, No. 2 (1966), 142—148
29. White A. T. Graphs, Groups and Surfaces, North-Holland/American Elsevier, New York, 1973.
30. White A. T., Beinecke L. W. Topological graph theory, in Selected Topics in Graph Theory (L. W. Beinecke and R. J. Wilson, Eds.), Academic Press, New York, 1978.

Гипотеза Римана и способы проверки простоты чисел[1)]

Г. Л. Миллер[2)]

В этой статье мы приводим два алгоритма проверки простоты целого числа. Первый алгоритм делает $O(n^{1/7})$ шагов; второй делает $Q(\log^4 n)$ шагов, однако при условии справедливости расширенной гипотезы Римана. Мы также показываем, что функции некоторого класса, включающего функцию Эйлера ϕ, по сложности вычисления эквивалентны разложению целых чисел на множители.

ВВЕДЕНИЕ

Существуют две классические вычислительные проблемы нахождения эффективных алгоритмов: (1) для проверки простоты (определения, является ли целое число простым или составным); (2) для разложения целых чисел на множители. Наилучшие верхние оценки для числа шагов, требуемых алгоритмами для (1) или (2), принадлежат Полларду [14] [3]). Поллард доказывает верхнюю оценку $O(n^{1/8+\varepsilon})$ шагов для проверки простоты и верхнюю оценку $O(n^{1/4+\varepsilon})$ шагов для разложения на множители, где ε — произвольная положительная постоянная. Мы даем алгоритм, который проверяет простоту числа и делает $O(n^{1/7})$ шагов. Слегка модифицируя этот алгоритм и допуская, что справедлива расширенная гипотеза Римана (РГР), получаем алгоритм, который проверяет простоту числа и делает $O(\log^4 n)$ шагов. Таким образом, мы показываем, что простота чисел проверяема за время, полиномиальное от длины двоичной записи числа n. Используя терминологию Кука [6] и Карпа [9], мы говорим, что простота числа проверяема за полиномиальное время при условии справедливости РГР.

---

[1]) Miller G. L. Riemann's Hypothesis and tests for primality, Journal of Computer and System Sciences, **13**, 300—317 (1976).

[2]) Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.

[3]) В настоящее время наилучшей оценкой числа шагов для проверки простоты чисел обладает алгоритм Адлемана (Adleman L. M., Pomerance C., Rumely R. S., On distinguishing prime numbers from composite numbers. — Ann. Math., 117 (1983), 173—206). — *Прим. ред.*

Одна из важных причин для нахождения быстрого алгоритма разложения чисел на множители заключается в том, что тогда многие другие вычислительные задачи можно быстро разрешить. Например, функцию Эйлера ϕ можно, очевидно, быстро вычислить, если задано разложение числа n на простые сомножители.

В качестве побочного результата работы по проверке простоты чисел мы показываем, что на самом деле верно и обратное, если допустить справедливость РГР. Таким образом, вычисление функции Эйлера ϕ по сложности эквивалентно разложению чисел на множители при условии, что справедлива РГР.

В последнем разделе мы обсуждаем соотношение между проблемами распознавания и вычислительными проблемами. Мы показываем, что некоторый класс функций, включающий разложение на простые сомножители и функцию Эйлера ϕ, обладает тем свойством, что график каждой функции этого класса распознаваем за полиномиальное время при условии, что справедлива РГР.

Способы проверки простоты чисел

Нашей основной целью в этом разделе является теорема 2, но сначала мы точно определим понятие проверки простоты числа за $O(f(n))$ шагов.

Определение. Мы говорим, что *алгоритм проверяет простоту за* $O(f(n))$ *шагов*, если существует детерминированная машина Тьюринга, которая выполняет этот алгоритм, и эта машина точно указывает, является ли число n простым или составным за не более чем $K \cdot f(n)$ шагов при некоторой постоянной K.

Используя это определение, мы можем сформулировать теорему 1:

Теорема 1. *Существует алгоритм, который проверяет простоту числа* n *за* $O(n^{0.134})$ *шагов.*

Если мы допустим, что справедлива расширенная гипотеза Римана (см. добавление), то теорема 1 может быть очень сильно улучшена. Так как время работы мало, то более удобно записывать его с помощью длины двоичной записи. Таким образом, пусть $|n|$ обозначает длину двоичной записи n. При этом обозначении основная теорема такова:

Теорема 2 (РГР). *Существует алгоритм, который проверяет простоту числа за* $O(|n|^4 \log \log |n|)$ *шагов.*

Трудная часть доказательства приведенных выше теорем заключается в том, чтобы показать, что существует «маленький»

квадратичный невычет. В теореме 1 мы обращаемся к работе Берджеса [3], который использует доказательство Вейля гипотезы Римана над конечными полями, в то время как в теореме 2 мы используем принадлежащее Энкени сведéние оценки величины первого квадратичного невычета к расширенной гипотезе Римана.

В статье используются следующие соглашения или обозначения:

Обозначения. Мы будем считать, что число n, которое нужно разложить на множители или проверить на простоту, нечетное, так как случай четного n легко сводится к нечетному. Символы p и q будут означать нечетные простые числа, (a, b) обозначает наибольший общий делитель a и b. Показатель, с которым 2 входит в разложение на простые сомножители n будет обозначаться $\#_2(n)$, т. е. $\#_2(n) = \max\{K : 2^K | n\}$

Нам будут нужны также следующие функции:

Определение. Пусть $n = p_1^{v_1} \dots p_m^{v_m}$ — разложение на простые сомножители нечетного числа n. Словами «разложение на простые сомножители» мы будем обозначать функцию, сопоставляющую натуральным числам запись их простых делителей вместе с показателями. Мы также рассмотрим следующие три функции:

(i) $\phi(n) = p_1^{v_1-1}(p_1 - 1) \dots p_m^{v_m-1}(p_m - 1)$ (функция Эйлера ϕ);

(ii) $\lambda(n) = \text{ОНК}\{p_1^{v_1-1}(p_1 - 1), \dots, p_m^{v_m-1}(p_m - 1)\}$ (λ-функция Кармайкла);

(iii) $\lambda'(n) = \text{ОНК}(p_1 - 1, \dots, p_m - 1)$.

Мотивировка доказательств

Ферма доказал, что для простого числа p

$$a^{p-1} \equiv 1 \bmod p, \quad \text{если } (a, p) = 1.$$

Поэтому, если для некоторого a, $1 < a < n$,

$$a^{n-1} \not\equiv 1 \bmod n, \tag{1}$$

то n должно быть составным. Далее, $a^m \bmod n$ может быть вычислено за $O(|m|M(|n|))$ шагов (где $M(|n|)$ обозначает число шагов для умножения двух чисел длины $|n|$) при использовании стандартной техники, описанной в [10]. Возможным способом для распознавания составных чисел может быть систематический поиск a, удовлетворяющих (1). Этот способ может оказаться неудачным для составных n по двум причинам:

(a) Может найтись составное n, которое удовлетворяет тождеству Ферма, т. е.

$$a^{n-1} \equiv 1 \bmod n \quad \text{для всех } (a, n) = 1.$$

(b) Первое a, удовлетворяющее (1), может быть очень большим, что приведет к неэффективному методу.

Остальная часть данного раздела будет посвящена изучению этих двух проблем. Для начала покажем, что в действительности некоторые составные числа удовлетворяют тождеству Ферма.

Теорема (Кармайкл [6]). *n удовлетворяет сравнению Ферма тогда и только тогда, когда $\lambda(n) \mid n - 1$.*

Например, составное число $561 = 3 \cdot 11 \cdot 17$ таково, что $\lambda(n) = OHK(2, 10, 16) = 80$, и 80 делит 560. Поэтому $(a, 561) = 1$ влечет за собой равенство $a^{560} \equiv 1 \bmod 561$ для всех натуральных чисел a. Таким образом, есть составные числа, удовлетворяющие сравнению Ферма. На первый взгляд кажется более трудным доказать, что эти числа составные. Но мы не только распознаем, что они составные, но и быстро найдем делитель. Может показаться, что очевидный подход состоит в использовании теста Ферма для распознавания тех составных n, что $\lambda(n) \nmid n - 1$, и некоторых других тестов для тех n, что $\lambda(n) \mid n - 1$. Вместо этого мы будем разбивать все составные числа на множества в соответствии с тем, что $\lambda'(n) \nmid n - 1$ или $\lambda'(n) \mid n - 1$.

Так как алгоритмы, используемые в теоремах 1 и 2, по сути, одинаковы, мы определим следующий класс алгоритмов:

Определение A_f. Пусть f — вычислимая функция на множестве натуральных чисел. Мы определяем алгоритм A_f с вводом n следующим образом:

(1) Проверить, является ли n степенью целого числа, т. е. $n = m^s$, $s \geqslant 2$. Если n — степень, выдать «составное» и остановиться.

(2) Проверить следующие ниже условия (i) — (iii) для каждого $a \leqslant f(n)$. Если какое-то из них будет выполнено, выдать «составное» и остановиться:

(i) $a \mid n$,

(ii) $a^{n-1} \not\equiv 1 \bmod n$,

(iii) $((a^{(n-1)/2^k} \bmod n) - 1, n) \neq 1, n$ для какого-нибудь k, $1 \leqslant k \leqslant \#_2(n-1)$.

(3) Выдать «простое» и остановиться.

Замечание. Алгоритм A_f, определенный выше, есть упрощенная версия алгоритма, необходимого для доказательства тео-

ремы 2. A_f является алгоритмом проверки простоты за $O(|n|^5 \log^2|n|)$ шагов, при условии что справедлива РГР.

Прежде чем доказывать теоремы 1 и 2, мы должны развить технический аппарат для определения f и показать, что существует $a \leqslant f(n)$, которое «работает».

Начнем с рассмотрения тех составных чисел n, для которых $\lambda'(n) \nmid n-1$. В следующей лемме мы даем характеристику тех a, которые удовлетворяют условию $a^{n-1} \not\equiv 1 \bmod n$.

Лемма 1. *Если* $\lambda'(n) \nmid n-1$, *то существуют простые числа* p *и* q, *такие, что*

(1) $p|n$, $p-1 \nmid n-1$, $q^m|p-1$, $q^m \nmid n-1$ *для некоторого целого* $m \geqslant 1$;

(2) *если* a *— любой невычет степени* q *по модулю* p, *то* $a^{n-1} \not\equiv 1 \bmod n$.

(См. в добавлении определение невычета степени q по модулю p и определение индекса $a \bmod p$, который обозначается $\mathrm{ind}_p a$; невычет степени q по модулю p рассматривается лишь при $q|(p-1)$.)

Доказательство леммы 1. Пусть $q_1, \ldots, q_n$ — различные простые делители n. Тогда из $\lambda'(n) = OHK\{q_1-1, \ldots, q_n-1\} \nmid n-1$ следует, что $q_i - 1 \nmid n-1$ для некоторого i. Полагая $p = q_i$, получим $p|n$ и $p-1 \nmid n-1$. Так как $p-1 \nmid n-1$, то существует простое q и целое $m \geqslant 1$, такие, что $q^m|p-1$ и $q^m \nmid n-1$. Значит, p и q удовлетворяют условию (1). Покажем теперь, что p и q удовлетворяют условию (2).

Предположим, что лемма неверна, т. е. $a^{n-1} \equiv 1 \bmod n$. Так как $p|n$, то

$$a^{n-1} \equiv 1 \bmod p. \tag{2}$$

Пусть b — первообразный корень $\bmod p$; тогда из (2) следует, что $b^{(\mathrm{ind}_p a)(n-1)} \equiv 1 \bmod p$. Так как из $b^m \equiv 1 \bmod p$ следует, что $p-1|m$, то мы имеем

$$p-1 \,|\, (\mathrm{ind}_p a)(n-1). \tag{3}$$

Из того, что a есть невычет степени q, следует $q \nmid \mathrm{ind}_p a$. Таким образом,

$$q \nmid \mathrm{ind}_p a \text{ и } q^m|p-1. \tag{4}$$

Применив (4) к (3), получим $q^m|n-1$ и это противоречит выбору q.

Лемма 1 мотивирует следующее

Определение. Пусть $N(p, q)$ равно наименьшему числу a, такому, что a есть невычет степени q по модулю p; оно определяется лишь при $q|p-1$. Используя рассуждения с индексами, нетрудно показать, что $N(p, q)$ — простое число.

Теорема (Энкени [1]) (РГР)

$$N(p, q) = O(|p|^2).$$

Используя теорему Энкени и лемму 1, получаем, что если $\lambda'(n) \nmid n-1$, то существует $a \leqslant O(|n|^2)$, такое, что $a^{n-1} \not\equiv 1 \bmod n$.

Вернемся теперь к обсуждению составных чисел n, обладающих свойством $\lambda'(n) \mid n-1$. Пусть $q_1, \ldots, q_m$ — различные простые делители n; тогда по определению λ' мы знаем, что $\#_2(\lambda'(n)) = \max(\#_2(q_1-1), \ldots, \#_2(q_m-1))$. Поэтому при некотором i, $1 \leqslant i \leqslant m$, $\#_2(\lambda'(n)) = \#_2(q_i-1)$. Мы теперь будем различать два типа чисел следующим образом.

Определение. Пусть $q_1, \ldots, q_m$ — различные простые делители n. Мы говорим, что n типа A, если при некотором j, $1 \leqslant j \leqslant m$, $\#_2(\lambda'(n)) > \#_2(q_j-1)$. С другой стороны, говорим, что n типа B, если $\#_2(\lambda'(n)) = \#_2(q_1-1) = \ldots = \#_2(q_m-1)$.

Отвлекаясь на время, чтобы мотивировать следующие три леммы, предположим, что у нас есть составное число $n = pq$. Предположим далее, что мы имеем число m, такое, что

$$m \equiv 1 \bmod q \quad \text{и} \quad m \equiv -1 \bmod p. \tag{5}$$

Первое из ограничений (5) влечет за собой $q \mid m-1$, из второго следует $m \not\equiv 1 \bmod n$. Таким образом, $q = (m-1, n)$. Если бы мы могли быстро вычислить некоторое m, удовлетворяющее сравнениям (5), то быстро нашли бы делитель n. В следующих леммах мы развиваем метод нахождения m, удовлетворяющего (5). Скажем, что у b есть нетривиальный НОД с n, если $(b, n) \neq 1, n$.

Лемма 2А. *Пусть n — составное число типа A, где, скажем, p и q делят n, и $\#_2(\lambda'(n)) = \#_2(p-1) > \#_2(q-1)$. Допустим далее, что $0 < a < n$, такое, что $\left(\frac{a}{p}\right) = -1$, где $\left(\frac{a}{p}\right)$ — символ Якоби* (см. добавление); *тогда или a, или $(a^{\lambda'(n)/2} \bmod n) - 1$ имеет нетривиальный НОД с n.*

Доказательство. Допустим, что a имеет тривиальный НОД с n. Так как $1 < a < n$, то должно быть $(a, n) = 1$. Так как $q-1 \mid \lambda'(n)$ и $\#_2(q-1) < \#_2(\lambda'(n))$, то $q-1 \mid (\lambda'(n)/2)$, откуда

$$a^{\lambda'(n)/2} \equiv 1 \bmod q. \tag{1}$$

Поскольку $(a^{\lambda'(n)/2})^2 \equiv 1 \bmod p$, то $a^{\lambda'(n)/2} \equiv \pm 1 \bmod p$, Предположим, что $a^{\lambda'(n)/2} \equiv 1 \bmod p$, тогда $p-1 \mid (\operatorname{ind}_p a)(\lambda'(n)/2)$, откуда следует, что $\operatorname{ind}_p a$ — четное число. С другой стороны, из

$\left(\frac{a}{p}\right) = -1$ следует, что $\mathrm{ind}_p a$ нечетен (см. добавление). Значит,

$$a^{\lambda'(n)/2} \equiv -1 \bmod p. \tag{2}$$

По (1) $q \mid (a^{\lambda'(n)/2} \bmod n) - 1$. По (2) $p \nmid (a^{\lambda'(n)/2} \bmod n) - 1$, так как p нечетное простое. Следовательно,

$$((a^{\lambda'(n)/2} \bmod n) - 1,\ n) \neq 1,\ n.$$

Лемма 2В. *Пусть n — составное число и имеет не менее чем два различных простых делителя, скажем p и q. Далее предположим, что n типа B и $1 < a < n$ таково, что $\left(\frac{a}{pq}\right) = -1$. Тогда или a, или $(a^{\lambda'(n)/2} \bmod n) - 1$ имеет нетривиальный общий делитель с n.*

Доказательство. Как и в доказательстве леммы 2А, мы предполагаем, что a имеет тривиальный НОД с n; тогда $(a, n) = 1$. Без ограничения общности мы предполагаем, что $\left(\frac{a}{p}\right) = -1$ и $\left(\frac{a}{q}\right) = 1$. С помощью техники, подобной использованной выше, показываем, что $a^{\lambda'(n)/2} \equiv -1 \bmod p$ и $a^{\lambda'(n)/2} \equiv 1 \bmod q$. Завершающие доводы следуют из предыдущего доказательства.

Лемма 3. *Если $p \mid n$, $\lambda'(n) \mid m$, и $k = \#_2[m/\lambda'(n)] + 1$, то $a^{(\lambda'(n)/2)} \equiv a^{m/2^k} \bmod p$.*

Доказательство. Так как $a^{\lambda'(n)} \equiv 1 \bmod p$, то $a^{\lambda'(n)/2} \equiv \pm 1 \bmod p$. Рассмотрим по отдельности два возможных значения $a^{\lambda'(n)/2}$:

(1) Если $a^{\lambda'(n)/2} \equiv 1 \bmod p$, то $a^{m/2^k} \equiv 1 \bmod p$, так как в силу нашего выбора k и того, что $\lambda'(n) \mid m$, следует, что $(\lambda'(n)/2) \mid (m/2^k)$.

(2) Если, с другой стороны, $a^{\lambda'(n)/2} \equiv -1 \bmod p$, то мы замечаем, что

$$a^{m/2^k} \equiv a^{(\lambda'(n)/2)\, m/(\lambda'(n)/2^{k-1})} \equiv$$
$$\equiv (-1)^{m/(\lambda'(n)\, 2^{k-1})} \bmod p.$$

Так как $m/\lambda'(n)\, 2^{k-1}$ нечетно, то $a^{m/2^k} \equiv -1 \bmod p$.

Используя леммы 2А и 3, мы видим, что если n — составное число типа A, $\lambda'(n) \mid n - 1$ и $a = N(p, 2)$, то либо $a \mid n$, либо $((a^{(n-1)/2^k} \bmod n) - 1,\ n) \neq 1,\ n$, где $1 \leqslant k \leqslant \#_2(n-1)$. Для чисел типа B нам необходимо следующее

Определение. *Пусть $N(pq)$* равно наименьшему a, такому, что $\left(\frac{a}{pq}\right) \neq 1$, где $\left(\frac{a}{pq}\right)$ — символ Якоби, и $N(pq)$ определено лишь для $p \neq q$. Заметим снова, что $N(pq)$ — простое число.

Теорема (Энкени [1] (РГР).

$$N(pq) = O(|pq|^2).$$

В действительности Энкени не рассматривал случая $N(pq)$, одиако этот случай следует без каких-либо изменений из его рассуждений. Нужно только использовать более сильную форму теоремы 6 Сельберга [16], на которую есть ссылка в [1, лемма 2(c)]. См. также в [12] утверждение и доказательство теоремы Энкени.

Доказательство теоремы 2 (слабая форма). По теоремам Энкени мы можем подобрать такое целое число $c \geqslant 1$, что $N(p, q) \leqslant c|p|^2$ и $N(pq) \leqslant c|pq|^2$. Рассмотрим A_f, где $f(n) = c|n|^2$.[1])

Анализ времени работы

(1) Алгоритм A_f сначала должен проверить, является ли n степенью целого числа, что занимает $O(|n|^4)$ шагов. Мы оставляем читателю проверку этой границы.

(2) A_f должен проверить (i), (ii) и (iii) для $f(n)$ различных a.

Проверка (i) занимает $O(|n|^2)$ шагов.

Проверка (ii) занимает $O(|n|M(|n|))$ шагов.

Проверка (iii) занимает $O((|n|M(|n|)+|n|^2)|n|)$ шагов, так как НОД может быть вычислен за $O(|n|)^2$ шагов, см. [10], и $1 < k < |n|$. На умножение уходит не менее $|n|$ шагов, так что на проверку (iii) уходит не более $O(|n|^2M(|n|))$ шагов.

Таким образом, A_f делает $O(|n|^4M(|n|))$ шагов. Если мы используем алгоритм Шёнхаге — Штрассена [18] для умножения двоичных целых, то $M(|n|) = O(|n|\log|n|\log\log|n|)$, и мы получаем $O(|n|^5 \log|n| \log\log|n|)$ шагов.

Доказательство корректности A_f. Если n простое, A_f укажет, что n простое, так что нам нужно лишь показать, что A_f распознает составное n. Если n составное, n относится к одному из следующих трех случаев:

(1) n — степень простого;

(2) $\lambda'(n) \nmid n - 1$;

(3) $\lambda'(n) \mid n - 1$ и n не есть степень простого.

Случай 1. Если n — степень простого, то в этом случае n есть степень целого числа, и тогда A_f установит, что n составное.

Случай 2. Если $\lambda'(n) \nmid n - 1$, то по лемме 1 найдутся p и q, такие, что если $a = N(p, q)$, то $a^{n-1} \not\equiv 1 \bmod n$. Таким образом,

[1]) Можно взять $c = 70$ (см. Math. Comput., v. 42, № 165, 1984, 297—330).

нужно лишь заметить, что $N(p, q) \leqslant f(n)$; это следует из нашего выбора f.

Случай 3. Если $\lambda'(n) \mid n-1$ и n не есть степень простого:

(A) Предположим, что n типа A; тогда по леммам 2A и 3 можно выбрать p и k $(k \leqslant \#_2(n-1))$, так, что если $a = N(p, 2)$, то либо $a \mid n$, либо $\left((a^{(n-1)/2^k} \bmod n) - 1, n\right) \neq 1, n$. Так как $N(p, 2) \leqslant f(n)$, n будет опознано как составное на шаге (i) либо (iii).

(B) Предположим, что n типа B. Тогда по леммам 2B, 3 и по предположению, что n не есть степень, можно выбрать p, q и $k \leqslant \#_2(n-1)$ так, что если $a = N(pq)$, то либо $a \mid n$, либо $\left((a^{(n-1)/2^k} \bmod n) - 1, n\right) \neq 1, n$. Поскольку $N(pq) \leqslant f(n)$, то A_f обнаружит, что n составное.

Чтобы доказать теорему 1, нам потребуется следующий результат Берджеса.

Теорема [2—4]. *Для любого* $\varepsilon > 0$

$$N(p, q) = O\left(p^{(1/4e^{1/2})+\varepsilon}\right),$$
$$N(pq) = O\left((pq)^{(1/4e^{1/2})+\varepsilon}\right).$$

Доказательство теоремы 1. По теореме Берджеса можно подобрать такое целое число $c \geqslant 1$, что

$$N(p, q) \leqslant cp^{(1/4)/(2.71)^{1/2}} \quad \text{и} \quad N(pq) \leqslant c(pq)^{(1/4)/(2.71)^{1/2}}.$$

Положим $l = 4(2.71)^{1/2}$. Рассмотрим A_f, где $f(n) = \lceil cn^{1/(l+1)} \rceil \leqslant \lceil cn^{0.133} \rceil$. Так как A_f делает $O(n^{0.134})$ шагов, нам нужно только показать, что A_f проверяет простоту числа. Если n простое, то A_f укажет, что n простое.

Предположим, что n составное. Тогда для n предоставляется по крайней мере один из следующих четырех случаев:

Случай 1. n степень простого.

Случай 2. n имеет делитель $\leqslant f(n)$.

Случай 3. $\lambda'(n) \nmid n-1$, у n нет делителя $\leqslant f(n)$.

По лемме 1 существуют такие простые числа p, q, что если $a = N(p, q)$, то $a^{n-1} \not\equiv 1 \bmod n$. Значит, нужно лишь показать, что $a = N(p, q) \leqslant f(n)$. По предыдущему выполняется неравенство

$$a \leqslant cp^{1/l}. \tag{5}$$

Так как n составное и для всех $a \leqslant f(n)$, $a \nmid n$, то

$$p \leqslant n/f(n), \text{ т. е. } p \leqslant (1/c)\, n^{1/(l+1)}. \tag{6}$$

Подставив (6) в (5), получим

$$a \leqslant n^{1/(l+1)} \leqslant f(n), \text{ так как } c \geqslant 1.$$

Случай 4. $\lambda'(n) \mid n-1$, у n нет делителя $\leqslant f(n)$, и n не является степенью простого.

(A) Предположим, что n типа A. Тогда, как и в случае 3A теоремы 1, нужно лишь показать, что $a = N(p, 2) \leqslant f(n)$, где $p \mid n$. Так как в этом случае (5) и (6) выполняются, получаем $a \leqslant f(n)$.

(B) Пусть n типа B. Так как n не является степенью простого, то у n есть не менее двух различных простых делителей, скажем p и q. Нужно лишь показать, что $N(pq) \leqslant f(n)$; это будет выполнено, если мы покажем, что $pq \leqslant n/f(n)$.

Утверждение. $n \neq pq$ (см. [5]).

Предположим, что $n = pq$, где $p < q$. Тогда $q-1 \mid pq-1$, так как $\lambda'(n) \mid n-1$. Но отсюда следует, что $q-1 \mid p-1$. Следовательно, $q \leqslant p$, что противоречит предположению $p < q$.

Согласно утверждению, $n = pqr$, где $r \neq 1$. Так как $r \mid n$, то $r \geqslant f(n)$. Следовательно, $pq \leqslant n/f(n)$.

Модификация алгоритма A_f

Прежде всего заметим, что a на шаге (2) A_f не обязано пробегать все числа $\leqslant f(n)$, а только простые числа $\leqslant f(n)$. Так как число простых $\leqslant f(n)$ равно $O(f(n)/\log f(n))$ по закону распределения простых чисел, то для верхней границы в теореме 2 получаем $O(|n|^5 \log\log|n|)$ шагов.

Мы дополним A_f следующим образом:

(1) Если n есть степень, выдать «составное».

(2) Вычислить $p_1, \ldots, p_m$, где p_i есть i-е простое число и m таково, что $p_m \leqslant f(n) < p_{m+1}$. Вычислить такие Q, S, что $n-1 = Q2^S$, Q нечетное. Положить $i = 1$ и перейти на (ii) (обозначив p_i через a повсюду).

(i) Если $i < m$, заменить i на $i+1$. Если $i = m$, выдать «составное» и остановиться.

(ii) Если $a \mid n$, то выдать «составное» и остановиться. Вычислить $a^Q \bmod n$, $a^{Q2} \bmod n$, ..., $a^{Q2^S} \bmod n$.

(iii) Если $a^{Q2^S} \bmod n \neq 1$, выдать «составное» и остановиться.

(iv) Если $a^Q \bmod n = 1$, вернуться на (i). Положить $J = \max(J: a^{Q2^J} \bmod n \neq 1)$.

(v) Если $a^{Q2^J} \bmod n = n-1$, вернуться на (i).

(vi) Выдать «составное» и остановиться.

Как и в доказательстве теоремы 2 (слабая форма), время работы алгоритма A_f оценивается в основном временем шага (2), где f та же, что и прежде. По существу, A_f должен вычислить следующее:

(1) первые m простых, что составит $O((f(n))^{3/2})$ шагов по методу решета;

(2) $a^{n-1} (\mathrm{mod}\, n)$, где a пробегает первые m простых, что составит $O(m|n|M(|n|))$ шагов. Таким образом, время работы A_f есть $O(|n|^4 \log\log|n|)$.

Чтобы доказать, что A_f проверяет простоту числа, нужно лишь вновь рассмотреть третий случай:

Случай 3: $\lambda'(n) \mid n-1$ и n не есть степень простого.

(A) Предположим, что n типа A, $\#_2(\lambda'(n)) = \#_2(p-1) > \#_2(q-1)$ и $p, q \mid n$. Положим $a = N(p, 2)$ (таким образом, a простое). Тогда нужно лишь показать, что один из шагов (ii), (iii) или (vi) выдает «составное» для этого a. Пусть $a \nmid n$ и $a^{n-1} \equiv 1 \bmod n$. Покажем, что A_f достигнет шага (vi). Если $a^S \equiv 1 \bmod p$, то $2 \mid S$, поскольку $\left(\frac{a}{p}\right) = -1$ и p нечетно. Поскольку $p \mid n$, то $a^Q \not\equiv 1 \bmod n$. Значит, A_f достигнет шага (v). По леммам 2A и 3 мы знаем, что существует такое k, что $a^{Q2^k} \equiv 1 \bmod q$ и $a^{Q2^k} \equiv -1 \bmod p$. Допустим, что $a^{Q2^J} \equiv -1 \bmod n$, тогда $a^{Q2^J} \equiv -1 \bmod p$ и q. Далее $a^{Q2^k} \equiv a^{Q2^J} \equiv -1 \bmod p$, откуда $k = J$. С другой стороны, $a^{Q2^k} \equiv 1 \bmod q$ и $a^{Q2^J} \equiv -1 \bmod q$, откуда $k > J$. Из этого противоречия следует, что $a^{Q2^J} \not\equiv -1 \bmod n$. Значит, A_f достигает шага (vi).

(B) Допустим, что n типа B. Доказательство этого случая повторяет рассуждения в случае A.

Относительная вычислительная сложность

В этом разделе мы обсудим относительную вычислительную сложность некоторых функций теории чисел.

Для начала рассмотрим следующий пример: функция Эйлера $\phi(n)$ определяется равной числу целых чисел между 1 и n, взаимно простых с n. Вычисление $\phi(n)$ по этому определению с проверкой каждого числа, меньшего n, на взаимную простоту с n требует не менее n шагов. Значит, этот метод требует экспоненциального относительно $|n|$ числа шагов. При заданном разложении n на простые сомножители, скажем $p_1^{v_1} \cdots p_m^{v_m}$, мы можем вычислить $\phi(n)$ с помощью произведения.

$$\phi(n) = p_1^{v_1-1}(p_1 - 1) \cdots p_m^{v_m-1}(p_m - 1)$$

не более, чем за $\log_2 n$ умножений, т. е. не более, чем за полиномиальное от $|n|$ время. Мы можем выразить формулу произведения в терминах сложности: если разложение n на простые сомножители может быть вычислено «быстро», то $\phi(n)$ может быть вычислена «быстро». Мы теперь займемся формализацией предыдущего утверждения и докажем его обращение при условии справедливости РГР.

Определения приводимости в кругу проблем распознавания (множеств) были введены многими авторами, см., в частности, работы Кука [6] и Карпа [9]. Так как мы в первую очередь занимаемся функциями, введем понятие функциональной приводимости.

Определение. Если заданы функции f и g, то мы скажем, что f *за полиномиальное время приводима к* g, обозначая это $f \leqslant_p g$, если существует машина Тьюринга, которая при вводе n и $g(n)$ вычисляет $f(n)$ за $O(|n|^k)$ шагов при некоторой постоянной k. Мы скажем, что f *эквивалентна за полиномиальное время* g, если $f \leqslant_p g$ и $g \leqslant_p f$, и обозначим это отношение $f \approx_p g$.

Данное определение приводимости за полиномиальное время очень сильное. В нем говорится, что если две функции эквивалентны за полиномиальное время, то верхние (нижние) границы времени для их вычисления различаются лишь добавлением многочлена. В последующих примерах мы дадим определение приводимости за полиномиальное время, которое требует лишь сохранения асимптотического времени работы.

Формализуем теперь наше утверждение о функции Эйлера.

Лемма 4. *Функции* ϕ, λ, λ' *за полиномиальное время приводимы к «разложению на простые сомножители», т. е.* ϕ, λ, $\lambda' \leqslant_p \phi$ *«разложение на простые сомножители».*

Доказательство. $\phi \leqslant_p$ «разложение на простые сомножители» следует из нашего обсуждения в начале этого раздела. Чтобы показать, что λ, λ' приводимы к разложению на простые сомножители, отметим следующие два факта относительно функции наименьшего общего кратного ОНК:

(1) $\text{ОНК}(a, b) = ab/(a, b)$,

(2) $\text{ОНК}(a, b, c) = \text{ОНК}(\text{ОНК}(a, b), c)$.

По лемме 4 получаем, что если функция Эйлера ϕ не может быть вычислена за полиномиальное время, то и разлагать числа на множители за полиномиальное время нельзя. Но не исключено, что вычисление $\phi(n)$ может быть сделано быстро, в то время как разложение на множители является трудным. Следующая лемма показывает, что все функции из некоторого класса, который включает ϕ, не более легки для вычисления, чем разложение на простые сомножители, при условии что справедлива РГР.

Лемма 5 (РГР). *Пусть* g *— любая функция, удовлетворяющая условиям:*

(1) $\lambda'(n) \mid g(n)$;

(2) $|g(n)| = O(|n|^k)$ *при некоторой постоянной* k. *Тогда «разложение на простые сомножители»* $\leqslant_p g$.

Доказательство. Рассмотрим следующую процедуру с n и m.

(1) Проверить, является ли n степенью целого числа.

(2) Выполнить шаги (i) и (ii) для каждого $a \leqslant f(n)$ (где f та же, что и в доказательстве теоремы 1).

(i) $a \mid n$,

(ii) $((a^{m/2^k} \bmod n) - 1, n) \neq 1$ для некоторого $0 \leqslant k \leqslant \#_2(m)$.

Если $\lambda'(n) \mid m$, то рассуждениями, аналогичными использованным в случае 3 доказательства теоремы 2, получаем, что эта процедура выдаст делитель числа n, если n составное. Если положить $m = g(n)$, то за $O(|g(n)||n|^3 M(|n|))$ шагов мы либо узнаем, что n простое, либо найдем n' некоторый делитель n. Если в приведенной выше процедуре мы заменим n на n', то $\lambda'(n') \mid g(n)$, поскольку $n' \mid n$ влечет $\lambda'(n') \mid \lambda'(n)$. Тогда за $O(|g(n)||n'|^3 M(n'))$ шагов мы либо узнаем, что n' простое, либо что n'' множитель n'. Итерируя эту процедуру не более чем $|n|$ раз, мы найдем все простые делители n. Таким образом, мы найдем разложение n на простые сомножители за $O(|g(n)||n|^4 M(|n|))$ шагов. Поскольку $|g(n)| = O(|n|^k)$, будет сделано $O(|n|^{k+4} M(|n|))$ шагов.

Мы получили следующую теорему.

Теорема 3 (РГР). *Функции ϕ, λ, λ' и «разложение на простые сомножители» эквивалентны за полиномиальное время, т. е. «разложение на простые сомножители»* $\simeq_p \phi \approx_p \lambda \approx_p \lambda'$.

Другая проблема, связанная с разложением чисел на множители, состоит в нахождении периода при разложении рационального числа в бесконечную дробь. Мы знаем, что такое разложение периодично при любой базе. Следовательно, функция «Период» (a, b) = минимальный период $1/a$ в базе b, корректно определена. Не представляется возможным доказать эквивалентность между «периодом» и «разложением на простые сомножители», используя предыдущее определение приводимости. Поэтому мы вводим более слабое определение приводимости, подобное приводимости по Тьюрингу из рекурсивной теории.

Определение. Если заданы функции f и g, мы скажем, что f *приводима по Тьюрингу за полиномиальное время к* g, обозначая это $f \leqslant_p^T g$, если существует машина Тьюринга со следующими свойствами:

(1) машина имеет отдельную ленту, с которой она может извлекать значения g, причем для извлечения $g(m)$ требуется $|m| + |g(m)|$ шагов;

(2) машина вычисляет $f(n)$ за $O(|n|^K)$ шагов, где K — некоторая постоянная.

Мы скажем, что f *и* g *эквивалентны по Тьюрингу за полиномиальное время*, если $f \leqslant_p^T g$ и $g \leqslant_p^T f$; обозначаем это $f \approx_p^T g$.

В лемме 5 были введены некоторые ограничения на рост функции g. Требовалось, чтобы длина g росла не быстрее, чем многочлен от длины ее аргумента. Мы будем говорить, что такая функция имеет *синтаксический полиномиальный рост.*

Лемма 6. *В классе функций с синтаксическим полиномиальным ростом отношения* $\leqslant_p$, $\approx_p$, $\leqslant_p^T$ *и* $\approx_p^T$ *обладают следующими свойствами:*

(1) $\leqslant_p$ *и* $\leqslant_p^T$ *— транзитивные отношения. Таким образом,* $\approx_p$ *и* $\approx_p^T$ *— транзитивные отношения.*

(2) *Из* $f \leqslant_p g$ *следует, что* $f \leqslant_p^T g$.

(3) *Класс функций, вычислимых за полиномиальное время, образует класс эквивалентности по отношению* $\approx_p$ *и* $\approx_p^T$.

Используя наше второе определение приводимости, мы можем теперь доказать эквивалентность между периодом и разложением на простые сомножители.

Теорема 4 (РГР). *«Период» эквивалентен по Тьюрингу за полиномиальное время «разложению на простые сомножители», т. е. «период»* $\approx_p^T$ *«разложение на простые сомножители».*

Доказательство. Стандартная теорема теории чисел (см. Харди и Райт [8]) гласит, что если $a = uv$, где $(v, b) = 1$ и u состоит только из простых чисел, делящих b, то «период» $(a, b) = \min\{m\colon b^m \equiv 1 \bmod v\}$. Другими словами, «период» (a, b) равен порядку $b \bmod v$.

Мы сначала покажем, что «период» $\leqslant_p^T$ «разложение на простые сомножители». Предположим, что на вводе $[a, b]$. Машина сначала вычисляет u и v, такие же, как прежде, посредством последовательного применения НОД. Для полноты мы приводим возможный способ. Рассмотрим последовательности $u_0, u_1, \ldots$ и $v_0, v_1, \ldots$, определенные равенствами $u_0 = 1$, $v_0 = a$, $u_{i+1} = (v_i, b)u_i$ и $v_{i+1} = a/u_{i+1}$. Здесь $a = u_i x_i$ и u_i состоит только из простых, делящих b. Если $u_i \neq u$, то $2u_i \leqslant u_{i+1}$. Поэтому при $i = |a|$, $u_i = u$ и $v_i = v$. Теперь машина извлекает разложение v на простые сомножители, по которому вычисляет $\lambda(v)$. Далее она извлекает разложение $\lambda(v)$ на простые сомножители, скажем $p_1^{\tau_1} \ldots p_m^{\tau_m}$. Мы знаем из теоремы Кармайкла, см. [5], что порядок $b \bmod v$ делит $\lambda(v)$. Значит, нужно лишь определить, какие из p_i отбросить. Это может быть сделано посредством вычисления минимальных h_i, удовлетворяющих сравнениям

$$p_1^{\tau_1} \ldots p_{i-1}^{\tau_{i-1}} p_i^{h_i} p_{i+1}^{\tau_{i+1}} \ldots p_m^{\tau_m} \equiv 1 \bmod v$$

для каждого i между 1 и m.

Очевидно, что порядок $b \bmod v$ равен $p_1^{h_1} \dots p_m^{h_m}$.

Чтобы доказать, что «разложение на простые сомножители» $\leqslant_p^T$ «период», возьмем в качестве f такую же функцию, как в доказательстве теоремы 2, и в качестве разлагаемого числа возьмем n. Предположим, что n не имеет делителей между 2 и $f(n)$, иначе просто разделим на них.

Утверждение (РГР). Пусть $h(n) = \max\{$«Период»$(n, 2), \dots, $«Период»$(n, f(n))\}$. Тогда $h(n) \leqslant \lambda(n)$ и $\lambda'(n) \mid h(n)$.

Так как «Период» $(n, i) \mid \lambda(n)$ при $2 \leqslant i \leqslant f(n)$, то $h(n) \mid \lambda(n)$; следовательно, $h(n) \leqslant \lambda(n)$. Пусть $q^m \mid \lambda'(n)$. Тогда $q^m \mid p - 1$ при некотором $p \mid n$. Пусть a — наименьший q-й невычет по модулю p, т. е. $a = N(p, q)$. Тогда получаем:

(1) По расширенной гипотезе Римана $a \leqslant f(p)$. Значит, $a \leqslant f(n)$.

(2) q^m делит порядок $a \bmod p$, поскольку a есть невычет степени q по модулю p и $q^m \mid p - 1$. Так как $(a, n) = 1$ по (1), то q^m делит порядок $a \bmod n$.

Из этих двух фактов следует, что $q^m \mid h(n)$. Утверждение доказано.

Поскольку h удовлетворяет условиям леммы 5, то «разложение на простые сомножители» $\leqslant_p h$. $h \leqslant_p^T$ «период», так как можно определить машину, которая просто извлекает «Период»$(n, 2), \dots,$ «Период»$(n, f(n))$ и вычисляет их ОНК. Эта машина работает за полиномиальное время, поскольку $f(n) = O(\log^2 n)$. Наконец, по лемме 6 получаем, что «разложение на простые сомножители» $\leqslant_p^T$ «период».

Разложение на множители и P — NP

Вероятно, самый интересный вопрос в теории вычислительной сложности есть вопрос **P**—**NP**. Кук [6] и Карп [9] показали, что удивительное множество проблем распознавания являются **NP**-полными. Одна проблема распознавания, о которой неизвестно, является ли она **NP**-полной, есть множество составных чисел, {составные}. Пратт [15] показал, что множество простых чисел, {простые}, принадлежат **NP**. Поэтому кажется невероятным, что {составные} является **NP**-полным, поскольку отсюда следовало бы, что $\mathbf{NP} = \overline{\mathbf{NP}}$, где $\overline{\mathbf{NP}}$ состоит из тех множеств, чьи дополнения принадлежат **NP**. Далее из теоремы 2 следует, что {составные} $\in \mathbf{P}$ при условии справедливости РГР. Эти два факта в некотором смысле устанавливают отношение {составные} к **P**—**NP** вопросу. Вопрос о сложности разложения на множители кажется более трудным.

В свете работ Пратта кажется естественным рассматривать разложение в терминах недетерминизма, не как проблему распознавания, а скорее как функцию, которая недетерминированно вычислима. В следующем определении мы вводим понятие детерминированной (недетерминированной) полиномиально вычислимой функции.

Определение. Пусть $\mathbf{P}^*$ обозначает множество функций от натуральных чисел, вычислимых за полиномиальное время. Мы скажем, что недетерминированная машина вычисляет f за T шагов, если машина при вводе n проделывает некий путь, который заканчивается, и любой путь, который заканчивается, выдает $f(n)$ за $O(T(n))$ шагов. Используя это определение, мы будем обозначать $\mathbf{NP}^*$ множество функций от натуральных чисел, вычислимых за недетерминированное полиномиальное время.

Как и в начале этого раздела, мы будем обозначать через $\mathbf{P}(\mathbf{NP})$ подмножества натуральных чисел, распознаваемые за детерминированное (недетерминированное) полиномиальное время. Очевидно, что множество составных чисел содержится в $\mathbf{NP}$, т. е. {составные} $\in \mathbf{NP}$. Пратт доказал следующий удивительный результат.

Теорема [15]. {простые} $\in \mathbf{NP}$.

Приводимые ниже следствия получаются при использовании результатов Пратта.

Следствие 1. «*Разложение на простые сомножители*» $\in \mathbf{NP}^*$.

Доказательство. Машина просто угадывает разложение на простые сомножители, распознает простоту каждого из сомножителей и затем выдает «разложение на простые сомножители». Так как имеется не более чем $\log n$ сомножителей, машина работает за полиномиальное время.

Следствие 2. «*Период*», ϕ, λ, $\lambda' \in \mathbf{NP}^*$.

Доказательство. Так как все четыре функции приводимы по Тьюрингу за полиномиальное время к функции «разложение на простые сомножители» и «разложение на простые сомножители» лежит в $\mathbf{NP}^*$, то нужно лишь доказать следующую лемму.

Лемма 7. *Если* $f \leqslant_p^T g$ *и* $g \in \mathbf{NP}^*$, *то* $f \in \mathbf{NP}^*$.

Доказательство. Пусть M — машина, которая вычисляет f с помощью метода, заданного посредством $f \leqslant_p^T g$. Пусть M' — машина, которая вычисляет g недетерминированно за полиномиальное время. Чтобы сделать машину, которая вычисляет f, мы просто заменим извлечение значений g в M вычислением

с помощью M'. Тогда новая машина будет работать за полиномиальное время, так как M извлекает не более чем полиномиальное количество значений g, каждое из которых может быть вычислено за полиномиальное время.

Сейчас мы введем две различные конструкции для получения посредством функций проблем распознавания и исследуем их свойства в свете результатов предыдущего раздела. Пусть $\langle a, b\rangle$ есть некоторое кодирование упорядоченной пары a, b натуральным числом, которое «эффективно», т. е. мы можем кодировать и декодировать за полиномиальное время.

Определение. Если f есть функция на множестве натуральных чисел, то мы полагаем график функции f равным

$$G_f = \{\langle n, f(n)\rangle \mid n \text{ натуральное число}\}$$

и проекцию f равной

$$P_f = \{\langle n, m\rangle \mid f(n) \leqslant m\}.$$

Используя эти два определения, получаем следующие леммы. (Мы обозначаем СПР множество функций с синтаксическим полиномиальным ростом, определенных в предыдущем разделе.)

Лемма 8. *Если $f \in$ СПР, то следующие утверждения эквивалентны:*

(1) $G_f \in \mathbf{NP} \cap \overline{\mathbf{NP}}$;
(2) $G_f \in \mathbf{NP}$:
(3) $f \in \mathbf{NP}^*$;
(4) $P_f \in \mathbf{NP} \cap \overline{\mathbf{NP}}$.

Доказательство. Импликации $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (4)$ и $(3) \Rightarrow (1)$ очевидны. Мы докажем случай $(4) \Rightarrow (3)$. Поскольку $P_f \in \mathbf{NP} \cap \overline{\mathbf{NP}}$, то существует недетерминированная машина Тьюринга, которая вычисляет характеристическую функцию P_f за полиномиальное время. Так как f из СПР, то существуют постоянные k и c, такие, что $|f(n)| \leqslant c|n|^k$. Величина $f(n)$ лежит между 0 и $2^{c|n|^k}$. Таким образом, используя двоичный поиск, нужно лишь вычислить $c|n|^k$ значений характеристической функции P_f.

Лемма 9. *Если $g, f \in$ СПР, то выполнены следующие утверждения:*

(1) *$P_f \in \mathbf{P}$ тогда и только тогда, когда $f \in \mathbf{P}^*$.*
(2) *Если $f \approx_p g$ и $G_g \in \mathbf{P}$, то $G_f \in \mathbf{P}$.*

Доказательство. (1) То, что из $f \in \mathbf{P}^*$ следует $P_f \in \mathbf{P}$, очевидно; из $P_f \in \mathbf{P}$ следует, что $f \in \mathbf{P}^*$ по тем же соображениям,

которые использовались для доказательства, что из $P_f \in \mathbf{NP} \cap \overline{\mathbf{NP}}$ следует $f \in \mathbf{NP}^*$.

(2) Рассмотрим следующую машину, скажем M, с вводом $\langle n, m\rangle$:

(i) M декодирует $\langle n, m\rangle$ в n и m и пытается вычислить $g(n)$ посредством алгоритма, заданного отношением $g \leqslant_p f$ с вводом n и m. Если он останавливается с возможным значением $g(n)$, скажем h, то M переходит к шагу (ii). Если алгоритм использует больше чем $c|n|^k$ шагов, то он отвергает $\langle n, m\rangle$, где c и k заданы приводимостью g к f.

(ii) M вычисляет $\langle n, h\rangle$ и проверяет включение $\langle n, h\rangle \in G_g$ с помощью алгоритма, заданного посредством $G_g \in \mathbf{P}$. Если $\langle n, h\rangle \notin G_g$, то M отвергает $\langle n, m\rangle$, в противном случае она переходит к (iii), и мы знаем $h = g(n)$.

(iii) M вычисляет $f(n)$, используя n, $g(n)$, с помощью алгоритма, заданного $f \leqslant_p g$. Если $f(n) = m$, то M допускает $\langle n, m\rangle$, иначе она отвергает $\langle n, m\rangle$.

Должно быть понятно, что M работает полиномиальное время и допускает в точности G_f.

Используя последние две леммы и приведения предыдущего раздела, получаем:

Теорема 5 (РГР). *Графики ϕ, λ, λ' и «разложения на простые сомножители» распознаваемы за полиномиальное время.*

Доказательство. По лемме 9 нужно лишь показать, что график «разложения на простые сомножители» принадлежит **P**, так как по теореме 3 все четыре функции эквивалентны за полиномиальное время. Но график «разложения на простые сомножители» принадлежит **P** по теореме 2.

Теорема 6. *Проекции «периода» ϕ, λ, λ' и «разложения на простые сомножители» принадлежат* $\mathbf{NP} \cap \overline{\mathbf{NP}}$, *и если какая-либо из этих проекций принадлежит* **P**, *то все эти функции принадлежат* $\mathbf{P}^*$.

Доказательство. Первая часть теоремы 6 вытекает из следствия 2 и леммы 8, а вторая часть следует из леммы 9, часть (1).

Эти результаты позволяют сделать различие между нашими двумя методами конструирования проблем распознавания с помощью функций. Теорема 5 показывает, что график функции может быть легко распознаваем, в то время как функция может быть сложно вычислима. Леммы 8 и 9 показывают, что проекция есть сохраняющее сложность естественное отображение функций в множество отношений. Теорема 6 выявляет возможные кандидаты для проблем распознавания из $(\mathbf{NP} \cap \overline{\mathbf{NP}}) - \mathbf{P}$,

ДОБАВЛЕНИЕ

Пусть Z_n обозначает кольцо классов вычетов целых чисел по модулю n. Пусть Z_n^* обозначает множество классов вычетов, взаимно простых с n, при умножении по модулю n. Z_n^* есть группа, и если p простое, то Z_p^* — циклическая группа порядка $p-1$. Таким образом, единственные решения уравнения $x^2 \equiv 1 \bmod p$ есть ± 1. Мы можем выбрать образующую циклической группы Z_p^*, скажем b; тогда определим $\operatorname{ind}_p a = \min\{m: b^m \equiv a \bmod p\}$. Заметим, что $\operatorname{ind}_p a$ зависит от нашего выбора образующей. Мы скажем, что a есть вычет степени q по модулю p, если существует такое b, что $b^q \equiv a \pmod p$.

Замечание. Если p, q простые и $q \mid p-1$, то a есть q-й вычет по модулю p тогда и только тогда, когда $q \mid \operatorname{ind}_p a$.

Определение. Символ Лежандра $\left(\frac{a}{p}\right)$ определяется так:

$\left(\frac{a}{p}\right) = 1$, если a — квадратичный вычет по модулю p и $(a, p) = 1$;

$= -1$, если a квадратичный невычет по модулю p и $(a, p) = 1$;

$= 0$, если $(a, p) \neq 1$.

Символ Якоби определяется так:

$$\left(\frac{a}{pq}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{a}{q}\right),$$

где $\left(\frac{a}{p}\right)$ и $\left(\frac{a}{q}\right)$ — символы Лежандра.

Эти два символа при фиксированных знаменателях определяют функции, относящиеся к общему классу функций, называемых характерами. Мы определим еще один характер следующим образом:

$$\chi(a) = \begin{cases} e(2\pi i (\operatorname{ind}_p a)/q), & \text{если } (a, p) = 1, \\ 0, & \text{если } (a, p) \neq 1, \end{cases}$$

где $q \mid p-1$ и e — показательная функция.

L-функция Дирихле определяется равенством

$$L(S, \chi) = \sum_{n=1}^{\infty} \chi(n)/n^S$$

где χ — характер.

Расширенная гипотеза Римана. Нули $L(S, \chi)$ в критической полосе $0 \leqslant$ (вещественная часть S) $\leqslant 1$ все лежат на прямой (вещественная часть S) $= 1/2$, где χ — один из трех приведенных выше характеров.

ЛИТЕРАТУРА

1. Ankeny N. C. The least quadratic non-residue, Ann. of Math. 55 (1952) 65—72.
2. Burgess D. A. The distribution of quadratic residues and non-residues Mathematika 4 (1957) 106—112.
3. Burgess D. A. On character sums and primitive roots. Proc. London Math. Soc. 12. № 3 (1962) 193—206.
4. Burgess D. A. On character sums and L-series. Proc. London Math. Soc. 12, № 3 (1962) 193—206.
5. Carmichael R. D. On composite numbers p which satisfy the Fermat congruence $a^{p-1} = p$. Amer. Math. Monthly 19 (1912) 22—27.
6. Cook S. A. The complexity of theorem-proving procedures. In: Conference Record of Third ACM Symposium on Theory of Computing, 1970, 151—158.
7. Davenport H., Erdös P. The distribution of quadratic and higher residues. Publ. Math. Debrecen, 2 (1952) 252—265.
8. Hardy G., Wright E. An introduction to the theory of numbers. p. 111. Oxford Press, New York — London, 1968.
9. Karp R. M. Reducibility among combinatorial problems. In: Complexity Of Computer Computation (R. E. Miller, J. W. Thatcher, Eds.), pp. 85—103, Plenum, New York, 1972.
10. Knuth D. The art of computer programming. Vol. 2: Seminumerical algorithms. Addison-Wesley, Reading, Mass., 1969. [Имеется перевод: Кнут Д. Искусство программирования для ЭВМ. Т. II Получисленные алгоритмы. — М.: Мир, 1977).
11. Miller G. L. Riemann's hypothesis and tests for primality. In: Proceedings of Sevens Annual ACM Symposium on Theory of Computing, 1975, 234—239.
12. Montgomery H. Topics in multiplicative number theory. Springer-Verlag Lecture Notes vol. 227, p. 120. Springer-Verlag, Berlin, 1971. [Имеется перевод: Монтгомери Г. Мультипликативная теория чисел. — М.: Мир, 1974.]
13. Pollard J. An algorithm for testing the primality of any integer. Bull. London Math. Soc. 3 (1971) 337—340.
14. Pollard J. Theorems on factorization and primality testing. Proc. Cambridge Philos. Soc. 76 (1974) 521—528.
15. Pratt V. Every prime has a succint certificate. SIAM J. Computing 4 (1975) 214—220.
16. Selberg A. Contributions to the theory of Dirichlets L-functions. Avh. Nor. Vidensk., Akad. Oslo, 1934.
17. Shanks D Class number, a theory of factorization and genera. Proc. Symp. Pure Math. 20 (1969). Number Theory Institute American Mathematical Society, 1971, 415—440.
18. Schönhage, Strassen V. Schnelle Multiplikation Grosser Zahlen. Computing 7 (1971) 281—292.

Проверка чисел на простоту с помощью вычислительных машин[1)]

Х. Уильямс[2)]

1. ВВЕДЕНИЕ

Простые числа пленяли людей еще в античное время. В самом деле, похоже, что понятие простоты восходит к пифагорейской школе [24]. Несомненно, что во времена Евклида (III век до н. э.) оно было уже хорошо известно. Некоторые идеи древних о простых числах и разложении на простые множители до сих пор продолжают активно разрабатываться. Мы обсудим их в начале настоящей статьи.

В большей части данной работы описан современный подход к проверке чисел на простоту. Хотя он всегда предполагает возможность использования вычислительных машин, не следует забывать, что истоки многих современных идей восходят к докомпьютерной эпохе. Первые исследователи простоты чисел часто располагали лишь электромеханической настольной счетной машиной, целеустремленностью и огромным энтузиазмом. Поэтому для них было жизненно важно найти какой-нибудь способ уменьшить утомительную работу, которой требовало тогда доказательство простоты даже 20—25-разрядного числа. Образ мыслей и энтузиазм этих ранних исследователей (а также многих современных) лучше всего описан в очаровательной работе Д. Лемера «Hunting big game in the theory of numbers» [48].

До недавнего времени работы по теории и практике проверки чисел на простоту не имели (и многие до сих пор не имеют) какой-либо другой цели, кроме удовлетворения чисто интеллектуальной любознательности. В самом деле, у них и не могло быть других мотивов, поскольку такая работа не имела никаких практических приложений. Однако оказалось, что большие простые числа могут применяться в стратегически важной области — криптографии. Хотя это открытие привлекло значительный интерес к работам по проверке простоты чисел, но оно также, по-видимому, вызвало растущую склонность жертвовать строгостью доказательства в некоторых новых способах проверки простоты.

[1)] Williams H. C. Primality testing on a computer. Ars Combinatoria, 5 (1978) 127—185.

[2)] Department of Combinatorics and Optimization; Faculty of Mathematics, University of Waterloo, Ontario, Canada.

Поскольку задача определения простоты даже 65-разрядного числа посредством математически корректного алгоритма может в некоторых случаях оказаться крайне трудной, мы теперь становимся свидетелями разработки методов Монте-Карло для «решения» этой задачи. Такие методы не ставят вопрос о простоте данного целого числа; вместо этого они пытаются показать теоретико-вероятностными методами, что любое число, которое указанный метод объявил простым, с чрезвычайно малой вероятностью может оказаться на самом деле составным. Некоторые из таких методов мы описываем в настоящей работе.

Мы излагаем также подход, который можно рассматривать как промежуточный между методами Монте-Карло и математически строгими методами. Он зависит от истинности очень правдоподобной (но не доказанной) математической гипотезы. Если гипотеза верна, то число, признанное простым с помощью такого алгоритма, действительно просто.

Как было отмечено в начале этого раздела, проблема простых чисел и способов проверки простоты очень стара; поэтому соответствующая литература весьма обширна. Библиография к данной статье охватывает работы, опубликованные после 1915 г. Ссылки на более ранние источники можно найти в работе [16].

2. РЕШЕТО ЭРАТОСФЕНА

Первым зафиксированным письменно методом отыскания простых чисел является метод Эратосфена (около 250 г. до н. э.). Метод находит все нечетные простые числа, меньшие данного целого N, с помощью следующих правил:

1) Выписать все нечетные числа от 3 до N.

2) Вычеркнуть 3^2, а далее — каждое третье число; затем вычеркнуть 5^2 и далее каждое пятое число.

3) Продолжать этот процесс до тех пор, пока первое оставшееся число, следующее за тем, кратные которому были вычеркнуты последними, не будет иметь квадрата, превосходящего N.

Те числа, которые не были вычеркнуты, суть все простые числа, не превосходящие N. См. ниже пример для $N = 101$.

3	5	7	~~9~~	11	13	~~15~~	17	19	~~21~~
23	~~25~~	~~27~~	29	31	~~33~~	~~35~~	37	~~39~~	41
43	~~45~~	47	~~49~~	~~51~~	53	~~55~~	~~57~~	59	61
~~63~~	~~65~~	67	~~69~~	71	73	~~75~~	~~77~~	79	~~81~~
83	~~85~~	~~87~~	89	91	~~93~~	~~95~~	97	~~99~~	101

Оставшиеся числа — все простые, меньшие или равные 101.

Эту весьма простую процедуру очень легко реализовать на ЭВМ. Фактически из-за его быстроты именно этот алгоритм большинство специалистов по вычислительной теории чисел применяют для порождения всех простых чисел, не превосходящих N. Время работы алгоритма имеет порядок $N \log\log \sqrt{N}$; однако для больших N такая оценка нереалистична, поскольку память вычислительной машины ограничена, и решето приходится строить состоящим из сегментов.

Бэйс и Хадсон [4] указали, как это можно сделать, и с помощью своего очень простого алгоритма смогли на IBM 370-168 найти все простые числа до 10^{12}. Они привели таблицу $\pi_{b,c}(x)$ для $b = 24$ и всех c, таких что $0 < c < 24$ и $(c, 24) = 1$. Значения x в этой таблице суть $x = 10^{11}, 2\cdot 10^{11}, \ldots, 10^{12}$. Здесь $\pi_{b,c}(x)$ означает количество простых чисел в последовательности вида $\{bn + c\}$, $n = 0, 1, 2, 3, \ldots$. Полученные ими значения $\pi(x)$ для $x = 10^{11}$, 10^{12} подтверждают значения $\pi(x)$, которые получил Боман [5]. Результаты Бомана были вычислены по формуле, аналогичной формуле Майселя (1845) (см. также [44] и [52]). В приведенной ниже табл. 1 даны значения $\pi(x)$ для $x = 10^n$ $(n = 1, 2, 3, \ldots, 13)$.

Таблица 1

x	
10	4
10^2	25
10^3	168
10^4	1 229
10^5	9 592
10^6	78 498
10^7	664 579
10^8	5 761 455
10^9	50 847 534
10^{10}	455 052 511
10^{11}	4 118 054 813
10^{12}	37 607 912 018
10^{13}	346 065 535 898

3. ПРОБНОЕ ДЕЛЕНИЕ

Леонардо Пизанский (1202), по-видимому, был первым, кто в опубликованной работе отметил, что для определения простоты числа N достаточно попробовать разделить его на числа, меньшие или равные $\sqrt{N}$, хотя эта идея неявно присутствует в алгоритме решета Эратосфена. Он также привел таблицу

простых чисел от 11 до 97. Это, по-видимому, первая таблица простых чисел, происхождение которой известно.

Поскольку составное N должно иметь хотя бы один простой делитель $p \leqslant \sqrt{N}$, то достаточно попробовать разделить N на все простые числа, не превосходящие $\sqrt{N}$, чтобы определить, простое оно или нет. Ясно, что если у N 30 разрядов или больше, то этот метод доказательства простоты работает слишком медленно; однако в сочетании с другими методами метод пробного деления оказался весьма мощным.

Необходимо сразу заметить, что порождение всех простых чисел, меньших определенного числа, все равно требует времени и памяти, даже если используется решето. Ясно, что метод пробного деления работает быстрее всего, если список простых чисел доступен для машины. Сами эти числа в памяти не хранятся, поскольку это потребовало бы слишком много места; обычно запоминаются разности между последовательными простыми числами. Даже если применяется такой принцип, то требуемый объем памяти и время доступа могут оказаться весьма значительными, и во многих случаях проще организовать пробное деление на некоторые легко определимые целые числа, в которые входят и простые (например, на нечетные и на 2). Такой принцип описан в [99].

Пусть $d_1, d_2, \ldots, d_m$ суть $m = 5760 = \varphi(30030)$ положительных целых чисел, меньших $30030 = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13$ и взаимно простых с ним.

Если у N есть простой делитель, то он либо равен одному из чисел 2, 3, 5, 7, 11, 13, либо должен иметь вид

$$f(k, i) = 30030k + d_i.$$

Если положить $\Delta_i = d_{i+1} - d_i$, то мы видим, что

$$f(0, 1) = 1,$$
$$f(k, i+1) = f(k, i) + \Delta_i,$$
$$f(k+1, 1) = f(k, m) + 2.$$

Поэтому чтобы для числа N найти все его простые делители, меньшие заданного фиксированного B, достаточно осуществить пробное деление N на все $f(k, i) < B$. Такие методы называются методами колеса. Каждый раз, когда k увеличивается на 1, говорят, что колесо с 5760 спицами «повернулось».

Вундерлих и Селфридж [99] приводят следующие данные о времени t работы их метода на IBM 360-67:

для $N < 10^{20}$, $B = 10^7$, $t < 32c$,
для $N < 10^{30}$, $B = 10^7$, $t < 70c$.

Возрастание времени работы связано с тем, что использовались подпрограммы для арифметических действий с большими целыми числами (multi-precise routines). Во всех рассуждениях настоящей статьи будет предполагаться, что такие программы существуют. Они обычно меняются от машины к машине, поскольку они составлены на том или другом машинном языке и поэтому не являются универсальными. Хорошее описание всевозможных известных алгоритмов арифметики больших целых чисел имеется в книге Кнута [29]. Рассматривая такие операции, необходимо помнить, что время, которого требует деление одного числа на другое, на много больше времени, которого требует умножение тех же чисел. Далее, время умножения двух чисел гораздо больше времени их сложения или вычитания. Поэтому самое лучшее — попытаться минимизировать число умножений и особенно делений во всяком алгоритме, в котором применяются программы арифметических действий с большими целыми числами.

4. СОВЕРШЕННЫЕ ЧИСЛА

Понятие совершенного числа также было уже хорошо известно во времена Евклида. Оно на самом деле довольно курьезно для современного математика; тем не менее с ним связано много работ по проверке на простоту, и следовательно, краткое обсуждение здесь этой темы необходимо.

Пусть

$$\sigma(N)=\sum_{d\mid N} d \quad \text{(сумма всех делителей числа } N\text{)}$$

и

$\sigma^*(N)=\sigma(N)-N$ (сумма всех собственных делителей числа N)

Говорят, что N — совершенное число, если $N=\sigma^*(N)$. Например,

$$6=1+2+3, \quad 28=1+2+4+7+14.$$

Евклид показал, что если $N=2^{n-1}(2^n-1)$ и 2^n-1 простое, то N совершенное число. Эйлер показал, что если N совершенно и четно, то оно должно иметь такой вид. До сих пор неизвестно, существуют ли нечетные совершенные числа. Если они существуют, то во всяком случае они должны превышать 10^{50} [21] и, возможно, 10^{200} [10].

Если $M_n=2^n-1$ простое, то n должно быть простым. Эти числа называют числами Мерсенна. Проблема определения таких простых чисел возникла уже в 1644 г. Хорошие обзоры ранних работ на эту тему сделали Арчибальд [3] и Улер [84]. Ниже приведена таблица значений n, для которых M_n — простое число. Первая часть этой таблицы была получена в докомпьютерную эпоху, а вторая — после появления ЭВМ.

Таблица 2. **Список всех значений $n>21\,000$, для которых $M_n = 2^n - 1$ простое число**

n	Число разрядов	Дата	Автор или источник
2	1		
3	1		Известны
5	2		Никомаху
7	3		(около 100 л до н. э.)
13	4	$\leqslant$1461	Codex Lat. Monac 24908
17	6	1603	Катальди
19	6	1603	Катальди
31	10	1772	Эйлер
61	19	1883	Первушин
89	27	1911	Пауэрс
107	33	1914	Пауэрс, Фокемберг
127	39	1876	Люка
		Докомпьютерная эпоха	
521	157	1952	Робинсон (SWAC) [42], [43], [70]
607	183		
1279	386		
2203	664		
2281	687		
3217	969	1957	Ризель (BESK) [65]
4253	1281	1961	Гурвиц [25] (IBM 7090) 25
4423	1332		
9689	2917		
9941	2993	1963	Гиллис [18]
11213	3376		(ILLIACII)
19937	6002	1971	Таккерман [83] (IBM 360/91)
		Компьютерная эпоха	

Легко видеть, что определение простоты столь огромного числа, как $2^{19937} - 1$ (наибольшее известное к моменту написания настоящей статьи простое число) — это грандиозная задача, как интеллектуальная, так и вычислительная. В следующем разделе мы обсудим процедуру Люка — Лемера, с помощью которой проверяется простота чисел Мерсенна.

Следует также упомянуть дружественные числа, т. е. такие числа N и $\sigma(N)$, что $N = \sigma^*(\sigma^*(N))$. Например, 284 и 220 — дружественные числа. Наибольшие известные дружественные числа нашел Риле [63].

5. ЧИСЛА ФЕРМА

Уже в 1640 г. (см. Арчибальд [2]) Ферма предположил, что числа вида $F_n = 2^{2^n} + 1$ всегда простые. С этого времени такие числа постоянно интересуют математиков. Красивый результат Гаусса о том, что правильный многоугольник с числом сторон F_n всегда можно построить с использованием лишь циркуля и линейки, несомненно, в значительной степени объясняет этот интерес. К сожалению, не известно ни одно простое число

Таблица 3. **Делимость чисел F_m, $5 \leqslant m \leqslant 22$**

m	Простые множители	Год	Первооткрыватель
5	641	1729	Эйлер
5	6700417	1729	Эйлер
6	274177	1880	Лэндри
6	67280421310721	1880	Лэндри, Леласье, Жерардин
7	59649589127497217	1970	Моррисон, Бриллхарт [55]
7	5704689200685129054721	1970	Моррисон, Бриллхарт [55]
8	c	1909	Морхэд, Уэстерн
9	2424833	1903	Уэстери
9	c	1967	Бриллхарт [23]
10	45592577	1953	Селфридж [75]
10	6487031809	1962	Бриллхарт [7]
10	c	1967	Бриллхарт [23]
11	319489	1899	Каннингхэм
11	974849	1899	Каннингхэм
12	114689	1877	Люка, Первушин
12	26017793	1903	Уэстерн
12	63766529	1903	Уэстерн
12	190274191361	1974	Холлибёртон, Бриллхарт [23]
13	2710954639361	1974	Холлибёртон, Бриллхарт 23
14	c	1961	Селфридж, Гурвиц [76]
15	1214251009	1925	Крайчик [30]
16	825753601	1953	Селфридж [75]
17	?		
18	13631489	1903	Уэстерн
19	70525124609	1962	Ризель [66]
19	646730219521	1963	Рэтхолл [98]
20	?		
21	4485296422913	1963	Рэтхолл [98]
22	?		

? = характер F_m неизвестен
c = составное число

Ферма, кроме F_0, F_1, F_2, F_3 и F_4. С тех пор, как Эйлер обнаружил, что F_5 делится на 641, большая часть работ по числам Ферма была связана с опровержением их простоты. В самом деле, у некоторых математиков в настоящее время возникает искушение предположить, что числа F_n всегда составные при $n > 4$. Несмотря на это, числа Ферма стимулировали появление множества работ по проверке чисел на простоту. Существует также несложный критерий (тест Пепина) определения простоты F_n. Мы приводим ниже воспроизведенный из работы [23] перечень того, что известно о простоте и делителях чисел Ферма F_n для $5 \leqslant n \leqslant 22$.

Таблица 4

n	k	m	Дата	Открыватель
23	5	25	1878	Первушин
25	48413	29	1964	Рэтхолл [98]
26	143165	29	1964	Рэтхолл [98]
27	141015	30	1964	Рэтхолл [98]
30	127589	30	1964	Рэтхолл [98]
30	149041	32	1964	Рэтхолл [98]
32	1479	34	1964	
36	5	39	1886	Солхофф
38	2653	40	1964	Рэтхолл [98]
38	3	41	1903	Каллен, Кэннингем, Уэстерн
39	21	41	1956	Робинсон [71]
42	43485	45	1964	Рэтхолл 98
52	4119	54	1964	Рэтхолл 98
55	29	57	1956	Робинсон [71]
58	95	61	1957	Робинсон 73
62	697	64	1978	Шиппн 80
63	9	67	1956	Робинсон [71]
66	7591	69	1978	Шиппн [80]
71	683	73	1978	Шиппи [80]
73	5	75	1906	Морхед
77	425	79	1957	Робинсон, Селфридж [73]
81	271	84	1957	Робинсон, Селфридж [73]
91	1421	93	1978	Шиппн [80]
117	7	120	1956	Робинсои [71]
125	5	127	1956	Робинсон [71]
144	17	147	1956	Робинсон [71]
150	1575	157	1956	Робинсон [71]
207	3	209	1956	Робинсон [71]
226	15	229	1956	Робинсон [71]
228	29	231	1956	Робинсон [71]
250	403	252	1957	Робинсон, Селфридж [73]
267	177	271	1957	Робинсон, Селфридж [73]
268	21	276	1956	Робинсон [71]
284	7	290	1956	Робинсои [71]
316	7	320	1956	Робинсои [71]
452	27	455	1956	Робинсон [71]
556	127	558	1977	Мэттью, Уильямс [53]
744	17	747	1977	Мэттью, Уильямс [53]
1945	5	1947	1957	Робинсон [73]

Результат Люка (1878) о том, что если p — простой делитель F_n, то

$$p = k2^{n+2} + 1,$$

позволяет предложить метод отыскания составных чисел Ферма. Мы определяем простые числа вида

$$k2^m + 1$$

и проверяем, не являются ли они делителями чисел F_n для $n \leqslant m - 2$. Эта идея была, по-видимому, впервые использована Первушиным при отыскании делителя $5 \cdot 2^{25} + 1$ числа F_{23}. Первым автоматизировал эту процедуру Робинсон в 1956 г. На сегодня делители F_n известны для 37 значений $n > 22$. Эти результаты приведены в табл. 4.

Простой делитель $5 \cdot 2^{1947} + 1$ числа F_{1945} — это число, содержащее 587 десятичных разрядов. Определение того, что столь большое число является простым, — довольно трудная задача. В нескольких следующих разделах мы покажем, как доказывается простота таких чисел.

В заключение этого раздела заметим, что некоторые теоретические результаты о F_n можно найти в работах Кармайкла [12] и Робинсона [73]. Следует также упомянуть исследования Ризеля [68] чисел вида $6^{2^n} + 1$ и $10^{2^n} + 1$.

6. РАЗЛОЖЕНИЕ НА МНОЖИТЕЛИ

Мы увидим, что проблемы разложения чисел на простые множители и определения простоты тесно связаны. Поскольку разложение чисел на множители не является предметом данной статьи, мы не будем обсуждать многочисленные доступные в настоящее время алгоритмы этой процедуры. Прекрасный обзор по этим вопросам, снабженный обширной библиографией, написал Ги [20]. Рекомендуем также очень легко читаемый отчет Моррисона и Бриллхарта [57] о их мощном алгоритме разложения.

Самые эффективные из известных ныне методов разложения основаны на идеях, изложенных в [57]. Вундерлих (неопубликовано) эмпирически установил, что время, необходимое для разложения числа N с помощью этих методов, растет пропорционально $N^{0,15}$. Возможно, что этот показатель уменьшится для больших значений N. (Вундерлих в основном интересовался числами в диапазоне от 35 до 42 десятичных разрядов.)

Р. Шреппель (не опубликовано) внес некоторые изменения в метод Моррисона — Бриллхарта. При использовании его нового метода получается следующая приближенная оценка количества операций, необходимых для разложения числа N:

$$\exp\left(c\sqrt{\log N \log\log N}\right).$$

Райвест, Шамир и Эдлимен [69] составили на основании этой оценки следующую таблицу, которая позволяет в какой-то степени судить о том, сколько времени потребуется *очень* быстродействующей вычислительной машине (каждая операция занимает 1 мкс), чтобы разложить на множители данное число N[1]).

Таблица 5.

Число цифр в N	Число операций	Время
50	1.4×10^{10}	3.9 ч
75	9.0×10^{12}	104 суток
100	2.3×10^{15}	73 лет
200	1.2×10^{23}	3.8×10^{9} лет
300	1.5×10^{24}	4.8×10^{15} лет
500	1.3×10^{39}	4.2×10^{25} лет

Очевидно, что задача разложения на простые множители даже сегодня очень сложна и требует для своего решения чрезвычайно большого времени (и, вероятно, останется такой всегда). Естественно, что в любой процедуре определения простоты чисел следует стремиться минимизировать количество выполняемых разложений и даже постараться вовсе обойтись без них.

7. СТЕПЕННОЙ АЛГОРИТМ

В значительной части дальнейших рассуждений часто будет необходимо определять значение степенного вычета

$$r \equiv b^n \pmod{m}$$

для очень больших n. В этом кратком разделе мы опишем весьма простой способ делать это. Из книги Кнута [29] явствует, что можно разработать множество других приемов решения указаиной задачи. Изложенное ниже знакомит с общей идеей, лежащей в основе всех этих методов.

Пусть n представлено в двоичной записи вида

$$n = d_0 2^k + d_1 2^{k-1} + d_2 2^{k-2} + \ldots + d_k,$$

где $d_0 = 1$, $d_j = 0$, $1 (j \geqslant 1)$. Заметим, что $k = [\log_2 n]$. Тогда, если $s_0 = d_0 = 1$ и

$$s_{j+1} = 2s_j + d_{j+1},$$

[1]) Здесь предполагается, что N не удается легко разложить каким-либо известным методом.

видно, что $s_k = n$. Если положить $r_j \equiv b^{s_j} \pmod m$, то

$$r_0 = b,\ r_{j+1} \equiv b^{s_{j+1}} = b^{2s_j + d_{j+1}} \equiv r_j^2 b^{d_{j+1}} \pmod m$$

и

$$r \equiv r_k \pmod m.$$

Отсюда ясно, что для нахождения r достаточно на каждом из $[\log_2 n]$ шагов его вычисления выполнить одно возведение в квадрат и не более чем одно умножение на само b. Поскольку $\log_2 n$ намного меньше n, то этот метод очень быстрый.

8. ПСЕВДОПРОСТЫЕ ЧИСЛА

Если N — простое число и N и b взаимно простые, то по теореме Ферма

$$b^{N-1} \equiv 1 \pmod N. \tag{8.1}$$

Если для некоторого N выбрать значение b, вычислить $r \equiv b^{N-1} \pmod N$ с помощью степенного алгоритма и при этом окажется, что $r \not\equiv 1 \pmod N$, то мы будем знать, что N не является простым. Это, к сожалению, вся информация, которую можно таким образом получить. Ничего о возможных делителях N мы не узнаем.

Хотя (8.1) должно выполняться для любого b, взаимно простого с N, если N — простое, отсюда не следует, что если (8.1) выполняется для некоторого b, то N обязательно простое. В самом деле, можно показать, что для любого b существует бесконечно много составных чисел N, для которых выполняется (8.1). Мы будем называть любое целое N, удовлетворяющее (8.1), *псевдопростым по основанию* b, или b-псп [1]). Существует обширная литература по составным псевдопростым числам; см., например, [74].

Теорема (Чиполла (1904)). ***Существует бесконечно много составных*** b-псп.

Доказательство. Пусть

$$y_p = \frac{b^{2p} - 1}{b^2 - 1},$$

[1]) Существует некоторая полемика по поводу термина «составное псевдопростое по основанию b». Приставка «псевдо» сама по себе должна означать, что число составное. Некоторые видные авторы предлагают использовать другой термин для произвольных N, удовлетворяющих (8.1), и сохранить название «b-псевдопростое» лишь для *составных* значений N, удовлетворяющих (8.1). Большинство исследователей в области проверки чисел на простоту согласны, что в принципе это было бы гораздо разумней, но в то же время опасаются, что принятая ныне терминология настолько укоренилась, что изменить ее окажется слишком трудно.

где p — такое нечетное простое число, что [1]) нод$(p, b^2-1)=1$. Тогда

$$y_p = \frac{b^p-1}{b-1}\cdot\frac{b^p+1}{b+1}$$

является составным целым числом. Кроме того,

$$b^{2p} \equiv 1 \pmod{y_p},$$

$$y_p - 1 = \frac{b^{2p}-1}{b^2-1} - 1 = b^2\left(b^{p-1}+1\right)\frac{b^{p-1}-1}{b-1}$$

$$\equiv 0 \pmod{2p}.$$

следовательно,

$$b^{y_p-1} = \left(b^{2p}\right)^{(y_p-1)/2p} \equiv 1 \pmod{y_p}$$

и y_p является b-псп.

Если взять $b=2$ и $p=5$, мы получим $y_p = 341 = 11\cdot 31$ и $2^{340} \equiv 1 \pmod{341}$.

Мы видим, что для любого b существует бесконечно много составных b-псевдопростых чисел. Может показаться, что если N b-псевдопростое и составное, то оно не может быть псевдопростым по основанию, отличному от b. Однако существуют составные числа N, которые являются псевдопростыми для всякого b, такого, что нод$(b, N)=1$. Такие числа называются *числами Кармайкла или абсолютно псевдопростыми.*

Таблица 6.

N	$CP_2(N)$	$C(N)$
10^3	3	1
10^4	22	7
10^5	78	16
10^6	245	43
10^7	750	105
10^8	2057	255
10^9	5597	646
10^{10}	14885	1547

Кармайкл (1912) показал, что число N является числом Кармайкла, если и только если

$$N = p_1p_2p_3 \ldots p_k \quad (k \geqslant 3),$$

где p_1, p_2, p_3, ..., p_k — различные простые числа, такие что

$$p_i - 1 \mid N-1$$
$$(i = 1, 2, 3 \ldots k).$$

Например, числа 561 и 1729 являются числами Кармайкла. Хотя Черник [13] дал метод получения чисел Кармайкла, до сих пор еще неизвестно, существует ли бесконечное множество таких чисел.

Пусть $CP_2(N)$ — число составных 2-псп чисел, меньших или равных N, и пусть $C(N)$ число чисел Кармайкла, меньших или равных N. Селфридж и Вогстофф (неопубликовано) получили следующую таблицу (см. табл. 6).

[1]) Нод $(a, b, c, \ldots, z)$ означает наибольший общий делитель чисел a, b, c, ... и z.

Если мы сравним эту таблицу с таблицей 1, мы увидим (см. Шенкс [79, гл. 4]), что неравенство

$$CP_2(N) < \sqrt{\pi(N)}$$

справедливо для N из всей этой области и отношение $CP_2(N)/\sqrt{\pi(N)}$ очень медленно уменьшается к концу таблицы Селфриджа и Вогстоффа. Однако не ясно, выполняется ли это неравенство для всех N. Эрдёш [17] высказал предположение, что даже $C(N)/N^{1-\varepsilon}$ будет бесконечно возрастающим для всякого $\varepsilon > 0$. Это утверждение кажется очень неожиданным ввиду имеющихся численных данных, но, по-видимому, Эрдёш имеет хорошие эвристические доводы в его пользу.

9. ПРОВЕРКА НА ПРОСТОТУ С ПОМОЩЬЮ ОБРАТНОЙ ТЕОРЕМЫ ФЕРМА

Нельзя определить, является ли целое число N простым, с помощью прямого обращения теоремы Ферма; но можно использовать модифицированную версию обращения. Например, уже в 1876 г. (см. [50]) Люка получил следующий результат.

Теорема. *Если* $a^x \equiv 1 \pmod N$ *для* $x = N-1$, *но не для* x, *являющегося собственным делителем* $N-1$, *то* N *— простое.*

Чтобы можно было применить эту теорему, мы, очевидно, должны иметь полное разложение числа $N-1$ на простые множители. Однако, как мы отмечали в гл. 6, наша цель — свести к минимуму число разложений.

В 1918 г. Поклингтон [58] получил очень важный результат в теории определения простоты. Казалось, что его работа не имеет никакого практического значения, вплоть до 1927 г., когда Лемер [32] улучшил и расширил его идеи. В течение нескольких последующих лет Лемер написал ряд статей [33, 34, 37, 40], в которых содержались следствия или обобщения результатов из [32]. В 1957 г. Робинсон также получил несколько результатов об обращении теоремы Ферма. Многие идеи, которые ранее рассматривались в этой книге, и несколько новых идей опубликованы в очень важной работе Бриллхарта, Лемера и Селфриджа [9]. Ниже в кратком обсуждении мы рассмотрим некоторые из методов, развитых в [9]. Нам понадобятся некоторые предварительные результаты.

Мы говорим, что число b, для которого нод$(b, m) = 1$, принадлежит экспоненте t по модулю m (или, в записи $t = \mathrm{ord}_m(b)$), если t — наименьшее положительное целое, такое что

$$b^t \equiv 1 \pmod m$$

То есть, t — порядок по b в группе приведенных вычетов по модулю m.

Лемма. *Если* $b^n \equiv 1 \pmod{m}$ *и* $t = \mathrm{ord}_m(b)$, *то* $t \mid n$.

Следствие. *Если* $t = \mathrm{ord}(b)$ *и* p *простое, то* $t \mid p - 1$.

Используя эти простые идеи, мы можем теперь доказать следующую теорему Поклингтона.

Теорема 9.1. *Пусть* q *— любое простое число, такое, что* $q^a \parallel m$ [1]). *Если*

$$b^m \equiv 1 \pmod{N}$$

и

$$\text{нод}\,(b^{m/q} - 1,\ N) = 1,$$

тогда всякий простой делитель N *должен иметь вид*

$$1 + kq^a.$$

Доказательство. Пусть p — любой простой делитель N и пусть $t = \mathrm{ord}_p(b)$. Тогда $t \mid m$ и $t \nmid m/q$; следовательно, $q^a \mid t$. Так как $t \mid p - 1$, мы видим, что

$$p = 1 + kq^a.$$

В следующих разделах мы увидим, как этот простой результат может быть использован для получения мощных тестов на простоту.

10. НЕКОТОРЫЕ ТЕСТЫ НА ПРОСТОТУ

Для дальнейшего обсуждения нам понадобятся некоторые замечания.

Пусть N — любое нечетное целое число, которое проверяется на простоту. Запишем $N - 1$ в виде произведения

$$N - 1 = F_1 R_1$$

где F_1 — число, для которого мы знаем все простые делители (F_1 полностью разложено на простые множители), а каждый простой делитель R_1 должен быть больше числа B_1, выбранной границы делителей. Мы можем получить такую информацию о F_1 и R_1 путем деления числа $N - 1$.

Для числа R_1 мы получаем

$$R_1 = S_1 S_2 \ldots S_l,$$

где $\text{нод}\,(S_i, S_j) = 1$ $(i \neq j)$ и каждый простой делитель превосходит некоторую границу делителей B_{1i}. Мы дадим такой пример в разд. 11. Обычно мы имеем $S_1 = R_1$ и $B_1 = B_{1i}$.

Используя эти замечания, мы можем получить теперь следующие *тесты* на простоту.

---

[1]) Мы используем обозначение $q^a \parallel m$, если $q^a \mid m$ и $q^{a+1} \nmid m$.

I. Для каждого простого делителя q_i числа F_1 существует некоторое a_i, такое, что

$$a_i^{N-1} \equiv 1 \pmod{N},$$
$$\text{нод}\left(a_i^{(N-1)/q_i} - 1,\ N\right) = 1.$$

II. Для каждого S_i существует некоторое b_i, такое, что

$$b_i^{N-1} \equiv 1 \pmod{N}$$

и

$$\text{нод}\left(b_i^{(N-1)/S_i} - 1,\ N\right) = 1.$$

Теорема 10.1. *Если выполнено* I *и* p *есть произвольный простой делитель числа* N, *то*

$$p \equiv 1 \pmod{F_1}.$$

Доказательство. Пусть

$$F_1 = \prod_{i=1}^{m} q_i^{a_i},$$

где все q_i различны. Из теоремы 9.1 видно, что каждый $q_i^{a_i}$ является делителем $p-1$, следовательно, $F_1 \mid p-1$.

Заметим, что если (I) справедливо и $F_1 > \sqrt{N}$, тогда, поскольку каждый простой делитель N должен превосходить $F_1 > \sqrt{N}$, N — простое. Приведем некоторые примеры.

1. Рассмотрим число

$$N = 156 \cdot 5^{202} + 1.$$

Мы видим, что

$$13^{N-1} \equiv 1 \pmod{N}$$

и

$$\text{нод}\left(13^{(N-1)/5} - 1,\ N\right) = 1;$$

таким образом, всякий простой делитель p числа N должен иметь вид

$$1 + k \cdot 5^{202} > \sqrt{N}$$

Отсюда следует, что N — простое.

2. Пусть $F_n = 2^{2^n} + 1$ $(n > 1)$ и пусть k — любое целое, для которого символ Якоби $(k/F_n) = -1$. (Например, $k = 3$, 5, 10 и т. д.). Для каждого значения k число F_n является простым, если и только если

$$k^{(F_n-1)/2} \equiv -1 \pmod{F_n}. \qquad (10.1)$$

Мы можем выразить это в виде алгоритма, полагая $T_1 = k$ и определяя $T_{r+1} \equiv T_r^2 \pmod{F_n}$. F_n — простое, если и только если

$$F_n \mid T_{2^n} + 1.$$

Доказательство. Очевидно, что для простого F_n условие (10.1) выполняется в силу критерия Эйлера. С другой стороны, если выполнено (10.1), всякий простой делитель p числа F_n должен превосходить 2^{2^n}; следовательно, F_n — простое.

Этот тест для определения простоты чисел Ферма впервые предложил Пепин в 1877 г.

Легко видеть, что для $N = Aq^n + 1$, где q простое и $q^n > A$, мы можем получить тест на простоту тем же методом, каким был получен тест (I). Эта идея была использована Робинсоном [73] и Мэттью и Уильямсом [53] для нахождения простых делителей чисел Ферма. Таблица простых чисел вида $k2^m + 1$ дана в [73] и [53]. Таблицу простых чисел вида $2A3^n + 1$ можно найти у Уильямса и Цернке [90].

Мы видели, как тест (I) может быть использован для проверки простоты. Теперь покажем ценность теста (II).

Теорема 10.2. *Если выполнено* (II) *и* p *— некоторый простой делитель* N, *то*

$$p \equiv 1 \pmod{r_1\, r_2\, r_3 \ldots r_l},$$

где r_i *некоторые простые делители* S_i.

Доказательство. Пусть $p \mid N$ и $t = \operatorname{ord}_p(b_i)$; тогда

$$t \mid S_i (N-1)/S_i$$

и

$$t \nmid (N-1)/S_i.$$

Отсюда видно, что $\operatorname{нод}(t, S_i) > 1$; поэтому существует некоторый простой делитель r_i числа S_i, который делит t. Из того, что $t \mid p-1$ и $\operatorname{нод}(S_i, S_j) = 1$, мы получаем утверждение теоремы.

Нам неизвестен набор $r_1, r_2, r_3, \ldots, r_l$, но мы знаем, что каждое r_i должно превосходить B_{1i} и, следовательно, каждое простое p, которое делит N, должно превосходить

$$1 + B_{11}B_{12}B_{13} \ldots B_{1l}.$$

Этот результат обычно не очень полезен сам по себе, но используя его совместно с (I) и теоремой 10.1, мы получим очень полезную теорему.

Теорема 10.3. *Если выполняются условия* (I) *и* (II), *то всякий простой делитель p числа N имеет вид*

$$1 + kr_1r_2r_3 \ldots r_lF_1,$$

где r_i *некоторый простой делитель* S_i. *Если*

$$(1 + B_{11}B_{12}B_{13} \ldots B_{1l}F_1)^2 > N,$$

то N — *простое.*

Идея о том, что информацию относительно размера границ делителей, т. е. чисел B_{11}, B_{12}, B_{13}, ..., B_{1l} (или только B_1) можно действительно использовать в тесте на простоту, является одной из наиболее интересных и новых в [9].

11. ДАЛЬНЕЙШИЕ РЕЗУЛЬТАТЫ И ПРИМЕР

В работе [9] есть много других интересных идей, которые мы не излагаем из-за недостатка места. Однако приведем здесь простой результат, в неявном виде присутствующий в одной из теорем указанной работы.

Теорема 11.1. *Если* $R_1 \equiv A \pmod{F_1}$ $(0 < A < F_1)$ *и выполняются* (I) *и* (II), *то* N — *простое, если*

$$N < \frac{1}{2} AB_{11}B_{12} \ldots B_{1l}F_1^2.$$

Доказательство. Пусть p — произвольный простой делитель N. Поскольку $N = pv$ и

$$N \equiv p \equiv 1 \pmod{r_1r_2r_3 \ldots r_lF_1},$$

то должно выполняться

$$v \equiv 1 \pmod{r_1r_2r_3 \ldots r_lF_1}.$$

Поэтому $p = 1 + m_1F_1$, $v = 1 + m_2F_1$, где m_1, $m_2 > B_{11}B_{12}B_{13} \ldots B_{1l}$.

Поскольку $R_1 = (N-1)/F_1 = m_1 + m_2 + F_1m_1m_2 \equiv m_1 + m_2 \pmod{F_1}$, то $m_1 + m_2 = A + tF_1$ для $t \geqslant 0$. Поэтому или m_1, или m_2 превосходит A/2, и

$$N = pv > \frac{1}{2} AB_{11}B_{12} \ldots B_{1l}F_1^2.$$

Покажем, как этот результат используется в следующем примере, взятом из работы Уильямса [95].

Рассмотрим числа вида $R(n) = (10^n - 1)/9$ во всех разрядах которых стоят одни единицы. Пусть $N = R(317)$. Получим

$$N - 1 = 10 \cdot (10^{158} + 1)(10^{79} + 1)(10^{79} - 1)/9.$$

Теперь, взяв $B_{11} = B_{12} = 10^8$ и $B_{13} = 6000$, получим

$$(10^{79} - 1)/9 = 317 \cdot 6163 \cdot 10271 \cdot 307\,627 \cdot S_1,$$
$$10^{79} + 1 = 11 \cdot 1423 \cdot S_2,$$
$$10^{158} + 1 = 101 \cdot 5689 \cdot S_3,$$

где все числа S_1, S_2 и S_3 — составные. Вагштаф и Лемер независимо нашли простой делитель

$$q = 9\,615\,060\,929$$

числа S_2. Фактически, когда Лемер повысил верхнюю границу делителей для каждого из чисел S_1, S_2 и S_3 до 3×10^{11}, это оказался единственный простой делитель, найденный им. Кроме того, 65-разрядное число M

$$M = S_2/q =$$
$$= 66443174541490579097997510158021076958392938976011506949065646573$$

является, как будет показано ниже, простым.

Полностью разложив $10^{79} + 1$ на простые множители, получаем $F_1 > 3{,}54 \times 10^{101}$. Тесты (I) и (II) выполняются, и A оказывается превышающим $3{,}53 \times 10^{101}$. Поскольку

$$0{,}5AB_{11}B_{13}F_1^2 > 0{,}5\,(3{,}53 \times 10^{101})\,(10^8)\,(6000)\,(3{,}54 \times 10^{101})^2 >$$
$$> 1{,}32 \times 10^{316} > N,$$

где N — простое число.

Из дальнейшего станет ясно, что обычно не удается успешно применить какую-либо проверку на простоту к столь большому числу, как $R(317)$. В данном конкретном случае нам чрезвычайно повезло, что $10^{79} + 1$ разложилось на множители столь удачным образом.

Весьма любопытно, что из четырех значений $n \leqslant 1000$, для которых $R(n)$ является простым, два вызвали споры. Дискуссию относительно $R(23)$ можно найти в работах Лемера [32, 34] и Крайчика [31]; см. также работу Бриллхарта и Селфриджа [8], утверждения которой противоречат приведенным выше.

12. ФУНКЦИИ ЛЮКА

В последних трех разделах изучались методы проверки простоты, основанные на разложении числа $N - 1$. В следующих трех разделах мы рассмотрим проблему проверки простоты числа N с использованием делителей числа $N + 1$. Для этого нужно сначала обсудить функции Люка.

Определим *функции Люка* формулами

$$V_n(P, Q) = \alpha^n + \beta^n,$$
$$U_n(P, Q) = (\alpha^n - \beta^n)/(\alpha - \beta),$$

где α, β суть корни вспомогательного квадратного уравнения

$$x^2 - Px + Q = 0,$$

а P и Q — взаимно[1]) простые целые числа. Рассмотрим также дискриминант $\Delta = (\alpha - \beta)^2 = P^2 - 4Q$. Заметим, что если $\Delta = 0$, то

$$U_n(P, Q) = \lim_{\beta \to \alpha} (\alpha^n - \beta^n)/(\alpha - \beta) = n\alpha^{n-1}.$$

Функции Люка всегда целые при $n \geqslant 0$ и удовлетворяют большому числу интересных тождеств. Многие из них можно найти в работе Люка [49]. Приведем ниже лишь некоторые из них[2]).

$$V_n^2 - \Delta U_n^2 = 4Q^n \tag{12.1}$$

$$V_0 = 2, \quad U_0 = 1, \quad V_1 = P, \quad U_1 = 1,$$

$$V_{n+1} = PV_n - QV_{n-1}, \quad U_{n+1} = PU_n - QU_{n-1}, \tag{12.2}$$

$$2V_{n+1} = PV_n + \Delta U_n, \quad 2U_{n+1} = V_n + PU_n, \tag{12.3}$$

$$V_{2n} = V_n^2 - 2Q^n, \quad U_{2n} = U_n V_n, \tag{12.4}$$

$$\frac{V_n + \sqrt{\Delta}\, U_n}{2} = \left(\frac{P + \sqrt{\Delta}}{2}\right)^n. \tag{12.5}$$

Из (12.2) ясно, что если $P' \equiv P$ и $Q' \equiv Q \pmod N$, то

$$V_n(P', Q') \equiv V_n(P, Q)$$
$$U_n(P', Q') \equiv U_n(P, Q) \pmod N.$$

Кроме того, если $\text{нод}(Q, N) = 1$ и

$$QP' = P^2 - 2Q \pmod N,$$

то

$$\begin{cases} Q^n V_n(P', 1) \equiv U_{2n}(P, Q) \\ PQ^{n-1} U_n(P', 1) \equiv U_{2n}(P, Q) \pmod N. \end{cases} \tag{12.6}$$

Теперь можно увидеть, как вычисляются $U_n(P, Q)$ и $V_n(P, Q)$ по модулю N при больших n. Мы применим алгоритм, аналогичный степенному алгоритму из разд. 7, но при этом нужно

[1]) Хотя обычно мы требуем, чтобы $\text{нод}(P, Q) = 1$, но если функции рассматриваются по модулю N, достаточно потребовать $\text{нод}(Q, N) = 1$. Это легко получить из того факта, что если $\text{нод}(Q, N) = 1$, то всегда можно найти такое P', для которого $\text{нод}(P' Q) = 1$ и $P' \equiv P \pmod N$.

[2]) Аргументы P, Q функций $V_n(P, Q)$ и $U_n(P, Q)$ часто опускаются, если это не приводит к путанице.

будет вычислять значения V_j и U_j. Для любой пары значений V_j, U_j можно найти V_{2j}, $U_{2j} (\mathrm{mod}\, N)$ с помощью (12.4). Затем можно вычислить V_{2j+1}, $U_{2j+1} (\mathrm{mod}\, N)$ с помощью (12.3). Заметим, что в (12.3) умножение на $2^{-1} (\mathrm{mod}\, N)$ (N нечетное) никогда не бывает очень трудным при использовании машины, работающей в двоичном коде. Оно требует самое большее сложения с N (когда сомножитель нечетный) и сдвига. Кроме того, применение формул (12.6) позволяет избежать вычисления Q^n в (12.4). Однако, несмотря на эти упрощения, следует заметить, что если время, необходимое для вычисления $r_n \equiv b^n (\mathrm{mod}\, N)$ равно T, то время, затраченное на вычисление $U_n(P, Q)$ $(\mathrm{mod}\, N)$, приблизительно равно $2T$.

Нам также потребуются некоторые арифметические свойства функций Люка. Их можно найти в [11] и в [9]. Заметим, что эти результаты вполне аналогичны приведенным выше. (Вряд ли можно было ожидать иного, поскольку функция $b^n - 1$ равна $(b-1)U_n(b+1, b)$.)

Первый из них формулируется так: *последовательность Люка*

$$U_1(P, Q),\ U_2(P, Q),\ \ldots,\ U_n(P, Q) \tag{12.7}$$

является *последовательностью, сохраняющей свойство делимости*. Это значит, что если $n \mid r$, то $U_n(P, Q) \mid U_r(P, Q)$. Если $U_\omega(P, Q)$ есть первый член (12.7), делящийся на m, мы назовем ω или $\omega(m)$ *рангом появления* m в последовательности Люка $\{U_k(P, Q)\}$. (Эта идея, конечно, аналогична идее использования экспоненты, которой b принадлежит по модулю m.)

Лемма. *Если* $m \mid U_n(P, Q)$ $(n \geqslant 1)$ *и* ω *— ранг появления* m *в последовательности Люка* $\{U_n(P, Q)\}$, *то* $\omega \mid n$.

Теперь приведем результат, аналогичный теореме Ферма!

Теорема 12.1. *Если* p *— простое число, такое, что* $\mathrm{нод}(p, 2\Delta Q) = 1$, *то*

$$p \mid U_{\Phi(p)}(P, Q),$$

где $\Phi(p) = p - (\Delta \mid p)$ *и* $(\Delta \mid p)$ *символ Лежандра.*

Следствие. *Если* p *— простое число, такое что* $\mathrm{нод}(p, 2\Delta Q) = 1$ *и если* ω *— ранг появления* p, *то* $\omega \mid \Phi(p)$.

Следующий результат можно теперь доказать так же, как теорему 9.1.

Теорема. *Пусть* q *— простое число, такое что* $q^a \| m$. *Если* $\mathrm{нод}(N, 2\Delta Q) = 1$,

$$U_m(P, Q) \equiv 0 (\mathrm{mod}\, N)$$

и

$$\mathrm{нод}(U_{m/q}(P, Q), N) = 1,$$

то любой простой делитель p числа N должен иметь вид

$$p = (\Delta \mid p) + kq^a,$$

т. е.

$$\Phi(p) \equiv 0 \pmod{q^a}.$$

13. ПРОВЕРКА ЧИСЕЛ НА ПРОСТОТУ С ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ ЛЮКА

В этом разделе мы приведем некоторые тесты на простоту числа N из работы [9], в которых применяются делители числа $N+1$. Определим

$$N + 1 = F_2 R_2,$$

где F_2 полностью разложено и любой простой делитель R_2 должен превышать B_2. Как и ранее, нод $(F_2, R_2) = 1$.

Имеем следующие тесты:

III. Для каждого делителя q_i числа F_2 существует последовательность Люка $\{U_k^i\} = U_k(P_i, Q_i)\}$ с одним и тем же дискриминантом Δ, для которого $(\Delta \mid N) = -1$, $N \mid U_{N+1}^{(i)}$, и

$$\text{нод}\left(N,\ U_{(N+1)/q_i}^{(i)}\right) = 1.$$

Для того, чтобы использовать (III), нам придется иногда менять значения P_i и Q_i, сохраняя при этом Δ неизменным. Это, однако, обычно совсем нетрудно сделать, поскольку

$$\Delta = P_i^2 - 4Q_i = (P_i + 2)^2 - 4(P_i + Q_i - 1).$$

IV. Для некоторой последовательности Люка $\{U_k\}$ с дискриминантом Δ, для которого $(\Delta \mid N) = -1$, мы имеем

$$N \mid U_{N+1}$$

и

$$\text{нод}\,(U_{(N+1)/R_2},\ N) = 1.$$

Мы могли бы сделать этот тест столь же общим, как и (II), но для целей объяснения этот более простой вариант вполне достаточен.

Теперь можно привести следующую теорему.

Теорема 13.1 (Моррисон [56]). *Если выполнено* (III), *то каждый простой делитель p числа N должен удовлетворять*

$$\Phi(p) \equiv 0 \pmod{F_2},$$

т. е.

$$p = \pm 1 + kF_2.$$

Результат Моррисона действительно очень важен. Прежде никто не обнаружил, как найти результаты, позволяющие воспользоваться всеми известными делителями числа $N+1$. Простая

идея сохранять дискриминант Δ нескольких различных последовательностей Люка неизменным позволяет получить простой тест.

Теорема 13.2. *Если выполнено* (IV), *то всякий простой делитель p числа N должен удовлетворять*

$$\Phi(p) \equiv 0 \pmod{q},$$

где q — некоторый простой делитель R_2.

Теорема 13.3. *Если условия* (III) *и* (IV) *удовлетворяются для некоторого Δ, тогда для всякого простого делителя p числа N*

$$\Phi(p) \equiv 0 \pmod{qF_2},$$

то есть

$$p = \pm 1 + kqF_2,$$

где q некоторый простой делитель R_2. Если $(B_2F_2 - 1)^2 > N$, то N — простое.

Примеры.

1. С помощью результатов, полученных выше, можно легко показать, что число

$$N = 156 \cdot 5^{202} - 1$$

является простым, поскольку

$$N \mid U_{N+1}(9, 1) \text{ и } \text{нод}(U_{(N+1)/5}(9, 1), N) = 1.$$

Пара целых чисел $156 \cdot 5^{202} \pm 1$ является близнецовой парой 144-разрядных простых чисел. Наибольшую известную пару близнецовых чисел представили Кренделл и Пенк [14] Наибольшая пара, которую они дали, имеет 303 разряда. Следует также отметить, что Рабин [62] открыл пару 124-разрядных чисел, которые весьма вероятно являются парой близнецовых простых чисел.

2. **Теорема.** *Пусть $M_m = 2^m - 1$, где m есть нечётное простое. Если мы положим $S_1 = 4$ и определим*

$$S_{k+1} \equiv S_k^2 - 2 \pmod{M_m},$$

где M_m будет простым, если и только если $M_m \mid S_{m-1}$.

Доказательство. Если мы положим $P = 2$, $Q = -2$, то из (12.6) мы получим, что

$$2^n V_n(-4, 1) \equiv V_{2n}(2, -2) \pmod{N},$$

где $N = M_m$. Итак

$$V_{2n}(-4, 1) = V_n^2(-4, 1) - 2;$$

следовательно,

$$S_{m-1} \equiv V_{(N+1)/4}(-4,\ 1) \pmod N$$

и

$$V_{(N+1)/2}(2,\ -2) \equiv 2^{(N+1)/4} S_{m-1} \pmod N.$$

Если N — простое, то в силу того, что $\Delta = 2^2 - 4(-2) = 12$, мы имеем $(\Delta | N) = -1$ и

$$N \mid U_{N+1}(2,\ -2) = V_{(N+1)/2}(2,\ -2)\, U_{(N+1)/2}(2,\ -2)$$

по (12.4) и теореме 12.1. Если $N \mid U_{(N+1)/2}(2, -2)$, то из (12.1) мы получим

$$V^2_{(N+1)/2} \equiv 4(-2)^{(N+1)/2} \equiv 8 \pmod N,$$

а из (12.4)

$$V_{N+1} = V^2_{(N+1)/2} - 2(-2)^{(N+1)/2} \equiv 8 + 4 = 12 \pmod N.$$

Но из (12.5) мы видим, что

$$V_{N+1} \equiv 2(1+\sqrt{3})^{N+1} \equiv 2(1+\sqrt{3})(1+3^{(N-1)/2}\sqrt{3}) \equiv$$
$$\equiv 2(1-3) \equiv -4 \pmod N;$$

Таким образом, $N \mid V_{(N+1)/2}(2, -2)$ и, следовательно, $N \mid S_{m-1}$.

Если $N \mid S_{m-1}$, то $N \mid V_{(N+1)/2}$. Из (12.1) мы видим, что нод$(N, U_{(N+1)/2}) = 1$ и из (12.4) мы имеем, что $N \mid U_{N+1}$; таким образом, всякий простой делитель p числа N должен иметь форму $\pm 1 + k2^n > \sqrt{N}$. Отсюда следует, что N — простое.

Хотя Люка [49] знал, что вышеприведенный тест является достаточным для того, чтобы доказать простоту M_m, когда $m \equiv 1 \pmod 4$, Лемер [35] был первым, кто дал доказательство необходимости и достаточности этого теста для всякого нечетного m. После работы Лемера [35] этот результат был получен и расширен еще в нескольких работах, в том числе в работах Уэстерна [86], Лемера [38], Капланского [28], Уорда [85] и Бревера [6].

В [35] Лемер также дал доказательство необходимости и достаточности условий простоты для чисел вида $k2^n - 1$ $(k < 2^n)$. (Тесты, основанные на (I), (II), (III) и (IV) дают только достаточные условия простоты.) Ризель [64], [67], Инкери [26] и Стечкин [82] также получили некоторые результаты для этих чисел, таблицы простых чисел этого вида можно найти в [64], [67], [87] и [27].

Необходимые и достаточные условия простоты чисел вида $k3^n - 1$ $(k < 3^n)$ и $k2^n3^m - 1$ $(k < 2^n3^m)$ даны Уильямсом [89]

и [96]. Он также дал необходимые и достаточные условия для простоты некоторых чисел вида $kq^n - 1$ при простых $q > 3$ и при $k < q^n$. Таблицы некоторых таких простых чисел можно найти в [100] и [90].

14. ОБЪЕДИНЕННАЯ ТЕОРЕМА И ПРИМЕР

Другая важная идея в [9] состоит в том, чтобы объединить тесты (I), (II), (III) и (IV) для получения доказательства простоты. Зачастую мы не имеем достаточно больших F_1 или F_2, чтобы доказать простоту N только лишь с помощью тестов (I), (II) или (III), (IV); однако мы можем доказать простоту N с помощью теоремы, подобной следующей.

Теорема 14.1. *Если выполняются условия* (I), (II), (III) и (IV) ((III) *и* (IV) *для одного и того же* Δ), *тогда* N *— простое, если* $B = B_1 = B_2$ *превосходит*

$$(2N/F_1^2F_2)^{1/3} \text{ или } (2N/F_1F_2^2)^{1/3}.$$

Заметим, что степень 1/2 из предыдущего результата можно теперь заменить на меньшую степень 1/3.

Иногда после того, как процесс пробного деления на множители, не превосходящие B завершен, оказывается, что R_1 и R_2 (или оба) сами являются псевдопростыми. Поэтому если бы мы могли доказать, что это псевдопростое число в действительности простое, то мы бы полностью разложили на простые множители $N+1$ или $N-1$ и смогли бы затем сделать вывод о простоте N. (Если бы даже мы не смогли доказать, что это псевдопростое число — простое, мы скорее всего смогли бы получить, что любой простой делитель этого числа больше B, тем самым подняв границу простых делителей для R_1 или R_2). Реализацию этой идеи на вычислительной машине обсуждали более или менее подробно Селфридж и Вундерлих [78]. Мы покажем, как использование таких псевдопростых может помочь в определении простоты для следующего довольно яркого примера.

Пример. Рассмотрим 65-разрядное 13-псп число

$$N = \overbrace{1\ldots1}^{32}41\overbrace{1\ldots1}^{32}$$

взятое из [95]. Мы будем сохранять $B = B_1 = B_2 = 5 \times 10^6$ неизменными во время всех вычислений.

Сначала мы найдем, что

$$F_1 = 2 \cdot 5 \cdot 7, \quad R_1 \text{ составное},$$

$$F_2 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 8681, \quad R_2 \text{ 13-псп}$$

У нас нет достаточного количества делителей чисел $N \pm 1$ для проверки простоты N, поэтому приходится работать с числом $N' = R_2 =$

$$= 5333060280646964208765844522093946123291820792108777363932 3$$

Здесь мы получим

$$F_1'' = 2, \quad R_1' \text{ составное},$$
$$F_2' = 2 \cdot 2 \cdot 3 \cdot 479, \quad R_2' \text{ 13-псп}.$$

Теперь рассмотрим число

$$N'' = R_2' =$$

$$= 92781146149042522769064796835315694559704606682476989 63$$

и найдем

$$F_1'' = 2 \cdot 149, \qquad R_1'' \text{ составное},$$
$$F_2'' = 2 \cdot 2 \cdot 3 \cdot 11, \quad R_2'' \text{ 13-псп}.$$

Положим

$$N''' = R_2'' =$$

$$= 702887470826079717947460582085724958785640959715734 77,$$

мы получим

$$F_1''' = 2 \cdot 2 \cdot 3 \cdot 11 \cdot 41 \cdot 129\,289, \quad R_1''' \text{ 13-псп}.$$
$$F_2''' = 2, \quad R_2''' \text{ составное}.$$

Окончательно, совместно с

$$N^{IV} = R_1''' = 1004538156437913141648802578728319087367292 57,$$

мы имеем

$$F_1^{IV} = 2 \cdot 2 \cdot 2 \cdot 89 \cdot 89 \cdot 89 \cdot 173$$
$$F_2^{IV} = 2 \cdot 3 \cdot 7 \cdot 11 \cdot 239 \cdot 9041.$$

Используя объединенную теорему 14.1, мы можем доказать, что N^{IV} — простое. Таким образом, F_1''' полностью разложено и мы можем доказать, что N''' — простое. Продолжая действовать таким же образом, мы получим окончательно, что N — простое.

15. ФУНКЦИИ ЛЕМЕРА И ОБОБЩЕНИЯ

Как мы упоминали ранее, Лемер впервые получил тест, который определяет простоту чисел Мерсенна. Он сделал это, расширив функции Люка заменой целого P в определении этих функций на $\sqrt{R}$, где $(R, Q) = 1$. В этом случае $\Delta = R - 4Q$ и теперь возможно (как хотел Лемер) получить $\Delta = -1 \pmod 4$.

Результирующие функции $V_n(\sqrt{R}, Q)$ и $U_n(\sqrt{R}, Q)$ уж не являются целыми для всех $n \geqslant 0$; они вместо этого теперь являются попеременно целыми числами и целыми, умноженными на $\sqrt{R}$. После деления на $\sqrt{R}$, всякий раз, когда он входит в $V_n(\sqrt{R}, Q)$ или в $U_n(\sqrt{R}, Q)$, мы будем иметь функции, которые обозначим $\overline{V}_n(R, Q)$ и $\overline{U}_n(R, Q)$. Эти функции имеют свойства, аналогичные свойствам функций Люка, но являются более общими.

Уильямс [91] обобщил идеи Лемера следующим образом. Предположим, что нам дан фиксированный многочлен с целыми коэффициентами степени ν:

$$f(x) = x^\nu - P_1 x^{\nu-1} + P_2 x^{\nu-2} - \ldots + (-1)^\nu P_\nu.$$

Также нам дано целое Q, такое что

$$\text{нод}(Q, P_1, P_2, P_3, \ldots, P_\nu) = 1.$$

Наконец, потребуем для простоты, чтобы $f(x)$ имел различные нули $\rho_1, \rho_2, \rho_3, \ldots, \rho_\nu$.

Если мы положим

$$\alpha_i + \beta_i = \rho_i, \quad \alpha_i \beta_i = Q \quad (i = 1, 2, 2, 3, \ldots, \nu),$$

то

$$\alpha_i^n + \beta_i^n = \sum_{j=0}^{\nu-1} V_{j,n} \rho_i^j$$

$$(\alpha_i^n - \beta_i^n)/(\alpha_i - \beta_i) = \sum_{j=0}^{\nu-1} U_{j,n} \rho_i^j,$$

где $V_{j,n}$ и $U_{j,n}$ $(j = 0, 1, 2, \ldots, \nu - 1)$ являются целыми для всякого $n \geqslant 0$. Мы имеем

$$\Delta = \begin{vmatrix} 1 & \alpha_1 & \alpha_1^2 & \ldots & \alpha_1^{2\nu-1} \\ 1 & \alpha_2 & \alpha_2^2 & \ldots & \alpha_2^{2\nu-1} \\ \hline 1 & \alpha_\nu & \alpha_\nu^2 & \ldots & \alpha_\nu^{2\nu-1} \\ 1 & \beta_1 & \beta_1^2 & \ldots & \beta_1^{2\nu-1} \\ \hline 1 & \beta_\nu & \beta_1^2 & \ldots & \beta_\nu^{2\nu-1} \end{vmatrix}^2$$

и $\Delta = (-1)^\nu E D^2 Q^{\nu(\nu-1)}$, где

$$D = \begin{vmatrix} 1 & \rho_1 & \rho_1^2 & \ldots & \rho_1^{\nu-1} \\ 1 & \rho_2 & \rho_2^2 & \ldots & \rho_2^{\nu-1} \\ \hline 1 & \rho_\nu & \rho_\nu^2 & \ldots & \rho_\nu^{\nu-1} \end{vmatrix}^2$$

и $E = F(2\sqrt{Q}) \cdot F(-2\sqrt{Q})$.

Если $\nu = 1$, ясно, что

$$V_{0,\,n} = V_n(P_1, Q)$$
$$U_{0,\,n} = U_n(P_1, H).$$

Если $\nu = 2$, $P_1 = 0$, $P_2 = R$, тогда $f(x) = x^2 - R$ и мы получаем функции Лемера, как показано в таблице ниже.

	$V_{0,\,n}$	$V_{1,\,n}$	$U_{0,\,n}$	$U_{1,\,n}$
n четное	$\bar{V}_n$	0	0	$\bar{U}_n$
n нечетное	0	$\bar{V}_n$	$\bar{U}_n$	0

Как и следовало ожидать, существует большое количество тождеств, удовлетворяющих этим обобщенным функциям Лемера. Несколько таких тождеств можно найти в [91]. Для того, чтобы вычислить $V_{l,\,n}$ по модулю N (нод$(Q, N) = 1$) для больших значений n, мы будем использовать мощную алгоритмическую схему, в которой это может быть сделано при помощи формулы, данной ниже.

Мы определим сначала целые S, M, так чтобы

$$QM = (-1)^{\nu+1} P_\nu S \equiv 1 \pmod N$$

а затем определим

$$W_{h,\,m} = \begin{cases} S^2 M^{m/2} V_{h,\,m} & (m \text{ четное}) \\ S M^{(m+1)/2} V_{h,\,m} & (m \text{ нечетное}). \end{cases}$$

Тогда

$$W_{\nu-1,\,2m+1} \equiv W_{0,\,2m+2} + W_{0,\,2m}$$
$$W_{h-1,\,2m+1} \equiv (-1)^{\nu+1} P_\nu (W_{h,\,2m+2} + W_{h,\,2m}) + (-1)^{\nu-h} P_{\nu-h} W_{\nu-1,\,2m+1} \pmod N$$
$$(h = 1, 2, \ldots, \nu - 1).$$

Итак, для $h = 0, 1, 2, \ldots, \nu - 1$

$$W_{h,\,2m} \equiv Q \sum_{i=0}^{\nu-1} \sum_{j=0}^{\nu-1} W_{i,\,m} W_{j,\,m} Z_{h,\,i+j} - 2\delta_{h,\,0} S^2 \pmod N \quad (m \text{ — четное}),$$

$$W_{h,\,2m} \equiv P_\nu^2 \sum_{i=0}^{\nu-1} \sum_{j=0}^{\nu-1} W_{i,\,m} W_{j,\,m} Z_{h,\,i+j} - 2\delta_{h,\,0} S^2 \pmod N \quad (m \text{ — нечетное}).$$

Здесь δ_{ij} — дельта-символ Кронекера, $Z_{ij} = \delta_{ij}$ $(0 \leqslant j < \nu)$, $Z_{i,\nu} = (-1)^{\nu-i+1} P_{\nu-1}$ и

$$Z_{j,\,n+\nu} = \sum_{i=1}^{\nu} P_i (-1)^{i+1} Z_{j,\,n+\nu-i}.$$

Например, если $\nu = 3$, мы получаем

$$W_{2,\,2m+1} \equiv W_{0,\,2m+2} + W_{0,\,2m}$$
$$W_{0,\,2m+1} \equiv P_3 (W_{1,\,2m+2} + W_{1,\,2m}) + P_2 W_{2,\,2m+1}$$
$$W_{1,\,2m+1} \equiv P_3 (W_{2,\,2m+2} + W_{2,\,2m}) - P_1 W_{2,\,2m+1} \pmod N$$

и для m нечетного

$$W_{0,\,2m} \equiv Q\left(W_{0,\,m}^2 + 2P_3 W_{1,\,m} W_{2,\,m} + P_1 P_3 W_{2m}^2\right) - 2S^2$$
$$W_{1,\,2m} \equiv Q\left(2W_{0,\,m} W_{1,\,m} - 2P_2 W_{1,\,m} W_{2,\,m} + (P_3 - P_1 P_2)\, W_{2,\,m}^2\right)$$
$$W_{2,\,2m} \equiv Q\left(W_{1,\,m}^2 + 2W_{0,\,m} W_{2,\,m} + 2P_1 W_{1,\,m} W_{2,\,m} + \right.$$
$$\left. + (P_1^2 - P_2)\, W_{2,\,m}^2\right) \pmod N.$$

Для m четного в этих трех формулах нужно Q заменить на P_3^2.

Если, как и в разд. 12, T — время, необходимое для вычисления $r_n \equiv b^n \pmod N$, то время, необходимое для вычисления $W_{0,\,n}$, $W_{1,\,n}$, $W_{2,\,n}$, ..., $W_{\nu-1,\,n}$, имеет порядок $(\nu^2 + 3\nu)T/2$. Эти функции намного более дорогие для вычислений по сравнению со степенными вычетами или с функциями Люка.

Нам понадобятся некоторые арифметические свойства функций

$$A_n = \text{нод}\,(U_{0,\,n},\ U_{1,\,n},\ U_{2,\,n},\ \ldots,\ U_{\nu-1,\,n})$$

и

$$C_n = \text{нод}\,(V_{1,\,n},\ V_{2,\,n},\ V_{3,\,n},\ \ldots,\ V_{\nu-1,\,n}).$$

Доказательство этих результатов можно найти в [91].

Во-первых, мы имеем, что обе последовательности $\{A_n\}$ и $\{C_n\}$ сохраняют свойство делимости. Также, если A_ω — первый член последовательности

$$A_1,\ A_2,\ A_3,\ \ldots,\ A_k,\ \ldots$$

который делит m, мы говорим, что ω есть ранг появления m в обобщенной последовательности Лемера $\{A_k\}$.

Лемма. *Если* $m \mid A_n$ *и* ω *есть ранг появления* m *в* $\{A_k\}$, *то* $\omega \mid n$.

Эти результаты, конечно, аналогичны результатам разд. 12 относительно фуикций Люка. Однако, когда мы имеем дело с последовательностью $\{C_k\}$, аналогичных результатов не суще-

ствует. Вместо ранга появления мы определим порядок появления.

Пусть m — целое, такое, что нод$(Q, m) = 1$. Пусть C_{τ_0} — первый член последовательности

$$C_1, C_2, C_3, \ldots, C_k, \ldots, \tag{15.1}$$

который делит m. Мы определим порядки появления m в последовательности $\{C_n\}$ как элементы возрастающей последовательности целых чисел

$$\tau_0, \tau_1, \tau_2, \ldots, \tau_j, \ldots,$$

в которой числа определены по правилу, что C_{τ_j} будет первым членом (15.1), таким, что m делит C_{τ_j} и $\tau_i \nmid \tau_j$ для любых $\tau_i < \tau_j$.

Теорема 15.1. *Если τ_j — любой порядок появления m, то $\tau_j \mid 2\omega$, где ω — ранг появления m в обобщенной последовательности Лемера.*

Предположим, что для любого простого p, такого что нод$(p, 2DEQ) = 1$, мы имеем

$$f(x) \equiv \prod_{i=1}^{\lambda} \varphi_i(x) \pmod{p},$$

где $\varphi_i(x)$ — неприводимый и степени μ_i по модулю p. Определим

$$E_i = \varphi_i\left(2\sqrt{Q}\right)\varphi_i\left(-2\sqrt{Q}\right),$$

$$\varepsilon_i = (E_i \mid p) \text{ (символ Лежандра)},$$

и

$$\Phi(p) = \left[p^{\mu_1} - \varepsilon_1, p^{\mu_2} - \varepsilon_2, \ldots, p^{\mu_\lambda} - \varepsilon_\lambda\right].$$

Теперь мы можем сформулировать обобщенную теорему Ферма для последовательности $\{A_n\}$.

Теорема 15.2. *Если p — простое, такое, что* нод$(p, 2DEQ) = 1$, *то $p \mid A_{\Phi(p)}$.*

Следствие. *Если ω — ранг появления p в $\{A_n\}$, то $\omega \mid \Phi(p)$.*

Теорема 15.3. *Пусть q — простое, такое что $q^a \mid m$. Если* нод$(N, 2DEQ) = 1$, *$N \mid A_m$ (или C_m) и* нод$(N, A_{m/q}) = 1$ (*или* нод$(N, C_{m/q}) = 1$), *то всякий простой делитель p числа N должен быть таким, что*

$$\Phi(p) \equiv 0 \pmod{q^a},$$

16. ТЕСТЫ НА ПРОСТОТУ, ИСПОЛЬЗУЮЩИЕ ОБОБЩЕННЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ ЛЕМЕРА

Уильямс и Джадд [92, 93] показали, как можно получить дальнейшие тесты на простоту с использованием обобщенных последовательностей Лемера с $v = 2$ и $v = 3$. По-видимому, соответствующие тесты для бо́льших значений ω, чем 3, настолько сложны и время их выполнения так велико, что нет смысла применять обобщенные последовательности Лемера с большими значениями.

Для $v = 2$ положим

$$N^2 + 1 = F_4 R_4, \quad \text{нод}(F_4, R_4) = 1,$$

а для $v = 3$ положим

$$N^2 + N + 1 = F_3 R_3, \quad \text{нод}(F_3, R_3) = 1,$$
$$N^2 - N + 1 = F_6 R_6, \quad \text{нод}(F_6, R_6) = 1,$$

где, как и прежде, все F_i $(i = 3, 4, 6)$ полностью разложены и каждый простой делитель числа R_i $(i = 3, 4, 6)$ должен быть больше соответствующей фиксированной границы B_i.

Трудность разработки тестов на простоту для $v = 2$ и $v = 3$ заключается в попытке приписать различные значения для P_1, P_2, Q (при $v = 2$) или P_1, P_2, P_3, Q (при $v = 3$) таким образом, чтобы $\Phi(p)$ осталось неизменным для всех специальных последовательностей $\{C_h^i\}$, которые могут быть сгенерированы при использовании этих значений P_1, P_2, Q $(v = 2)$ или P_1, P_2, P_3, Q $(v = 3)$. Здесь p означает возможный простой делитель N — числа, которое мы проверяем. Другими словами, мы стремимся получить теоремы, аналогичные теореме Моррисона 13.1.

При $v = 2$ эти значения для P_1, P_2, Q можно определить через два независимых параметра H_i и K_i (см. [92]). При $v = 3$ искомые значения величин P_1, P_2, P_3 и Q можно выразить через фиксированное предварительно выбранное значение G и три независимых параметра h_i, k_i, l_i (см. [93]). Подробности этих вычислений достаточно сложны, и поэтому рекомендуем читателю обращаться за дополнительной информацией к [92, 93]. Здесь достаточно отметить, что после того, как они проделаны, можно предложить следующие тесты:

$v = 2$

V. Для каждого простого q_i, делящего F_4, существует пара H_i, K_i, такая что для специальной последовательности $\{C_n^{(i)}\}$ выполняется

$$N \mid C_{N^2+1}^{(i)} \text{ и } \text{нод}\left(N, C_{(N^2+1)/q_i}^{(i)}\right) = 1,$$

VI. Существует пара H, K, такая что для специальной последовательности $\{C_n\}$ выполняется

$$N \mid C_{N^2+1} \text{ и нод}(N, C_{(N^2+1)/R_4}) = 1.$$

$v = 3$, $(G \mid N) = +1$

VII. Для каждого простого q_i, делящего F_3, существует тройка h_i, k_i, l_i, такая что для специальной последовательности $\{C_n^{(i)}\}$ выполняется

$$N \mid C^{(i)}_{N^2+N+1} \text{ и нод}\left(N, C^{(i)}_{(N^2+N+1)/q_i}\right) = 1.$$

VIII. Существует тройка h, k, l, такая что для специальной последовательности $\{C_n\}$ выполняется

$$N \mid C_{N^2+N+1} \text{ и нод } (N, C_{(N^2+N+1)/R_5}) = 1.$$

$v = 3$, $(G \mid N) = -1$

IX. Для каждого простого q_i, делящего F_6, существует тройка h_i, k_i, l_i, такая что для специальной последовательности $\{C_n^{(i)}\}$ выполняется

$$N \mid C^{(i)}_{N^2-N+1} \text{ и нод } \left(N, C^{(i)}_{(N^2-N+1)/q_i}\right) = 1.$$

X. Существует тройка h, k, l, такая что для специальной последовательности $\{C_n\}$ выполняется

$$N \mid C_{N^2-N+1} \text{ и нод } (N, C_{(N^2-N+1)/R_6}) = 1.$$

Если выбрать последовательности $\{C_n^{(i)}\}$ этим способом, то можно доказать теоремы, аналогичные теоремам 13.1, 13.2 и 13.3 для каждой из пар тестов: (V) и (VI), (VII) и (VIII), (IX) и (X). К сожалению, тесты (VI), (VIII) и (X) уже не столь полезны, как тесты (II) и (IV); однако иногда их можно с успехом применить (см. [93]). Кроме того, сам по себе любой из этих 6 тестов не очень полезен, но когда их применяют в сочетании друг с другом или с четырьмя описанными выше тестами, часто удается получить очень мощный алгоритм проверки чисел на простоту.

Сочетания этих тестов описаны в [93], а в применении к некоторым очень «трудным» числам — в [94]. На самом деле, взяв

$$K = \frac{1}{12} F_1 F_2 F_4 F_6,$$

обычно можно бывает установить для любого нечетного $K < 10^{100}$, является ли N простым или нет, если $N < 10K^3$.

Приведем теперь несколько примеров. Для каждого из них положим $B = B_1 = B_2 = B_3 = B_4 = B_6$.

1) Для числа M из разд. 11 получаем при $B = 2 \times 10^6$

$$F_1 = 2^2 \cdot 79,$$
$$F_2 = 2 \cdot 3 \cdot 61 \cdot 157 \cdot 199^2$$
$$F_3 = 19 \cdot 2671 \cdot 2719$$
$$F_4 = 2 \cdot 5 \cdot 29 \cdot 149 \cdot 421 \cdot 541 \cdot 2137$$
$$F_6 = 2 \cdot 3 \cdot 13 \cdot 1411783$$

С использованием тестов, описанных выше, этого достаточно, чтобы показать, что M простое. Когда Лемер увеличил B до $4{,}2 \times 10^9$, он обнаружил, что изменилось лишь F_2. Оно стало следующим:

$$F_2 = 2 \cdot 3 \cdot 61 \cdot 157 \cdot 199^2 \cdot 20373173 \cdot 2220165587 \cdot 2746999987.$$

Эта информация позволяет доказать простоту M с использованием только тестов из разд. 13.

В следующих двух примерах фигурируют числа, простоту которых доказать очень трудно.

2) 121-разрядное число

$$N = 2^{400} - 593 =$$
$$= 258224987808690858965591917200301187432970579282922351$$
$$28306593565404762201684119462964535328013783143590317197274749278 3$$

было обнаружено Рабином [62] и является наибольшим простым числом, не превосходящим 2^{400}. Методы Рабина позволяют предполагать, что N простое, но недостаточны, чтобы это доказать. При $B = 1{,}5 \times 10^8$ имеем:

$$F_1 = 2 \cdot 1384711,$$
$$F_2 = 2^4 \cdot 3^2 \cdot 3023 \cdot 23251903,$$
$$F_3 = 7 \cdot 2521 \cdot 2213647 \cdot 70792627,$$
$$F_4 = 2 \cdot 5 \cdot 13 \cdot 298013,$$
$$F_6 = 3 \cdot 19 \cdot 43.$$

Располагая этой информацией, методами работы [94] можно доказать, что N простое.

3) 65-разрядный делитель

$$N = 19809950476703891759635852223863606381827838846342829232189869441$$

числа Люка L_{470} является самым неподдающимся анализу на простоту числом такого размера, известным автору. Положив $B = 1{,}3 \times 10^9$, получаем

$$F_1 = 2^7 \cdot 5 \cdot 19 \cdot 37 \cdot 47 \cdot 139,$$
$$F_2 = 2 \cdot 3^3 \cdot 7,$$
$$F_3 = 13 \cdot 31 \cdot 73 \cdot 79,$$
$$F_4 = 2,$$
$$F_6 = 3.$$

Заметим, что большая часть информации об этом числе сосредоточена в размере числа B. Этой скудной информации достаточно, однако, чтобы доказать, что N простое [94]. Пытаясь увеличить размер всех или хотя бы некоторых делителей F_1, F_2, F_3, F_4, F_6, Лемер провел пробное деление $N \pm 1$, $N^2 + 1$, $N^2 \pm \pm N + 1$ на ЭВМ ILLIAC IV вплоть до границы делителей $3{,}8 \times 10^{10}$ (4,75 ч работы центрального процессора) и не обнаружил ни одного нового делителя.

17. ДАЛЬНЕЙШИЕ РЕЗУЛЬТАТЫ О ПСЕВДОПРОСТЫХ ЧИСЛАХ

В дальнейшем нам потребуются некоторые результаты о числах, псевдопростых в более сильном смысле, чем определенные в разд. 8.

Сначала определим эйлеровы b-псевдопростые числа (или эйлеровы b-псп) как такие нечетные N, для которых

$$b^{(N-1)/2} = (b \mid N) \pmod{N}.$$

Здесь $(b \mid N)$ — это символ Якоби. Если N простое и $(b, N) = 1$, то N заведомо эйлерово b-псп.

Селфридж (не опубликовано) определяет *сильные b-псевдопростые* числа как такие нечетные N, для которых если

$$N - 1 = 2^s t \quad (t \text{ нечетно}),$$

то либо

$$b^t \equiv 1 \pmod{N},$$

либо

$$b^{t2^r} \equiv -1 \pmod{N}$$

для некоторого неотрицательного $r < s$. Снова мы видим, что если N простое и $(b, N) = 1$, то N *сильное* b-псп.

Наконец, мы будем говорить, что нечетное N удовлетворяет *свойству A для данного b*, если

$$b^{N-1} \equiv 1 \pmod{N}$$

и

$$\text{нод}\left(b^{(N-1)/2^l} - 1, N\right) = 1,$$

где l — наименьшее целое число (если оно существует), такое что

$$b^{(N-1)/2^l} \not\equiv 1 \pmod{N}.$$

Последняя идея была предложена в важной работе Миллера [54]. В ней также обсуждается разновидность тестов на простоту, которую с тех пор называют проверкой простоты с помощью гипотез. Мы обсудим эту идею более подробно в следующем разделе. Теперь же приведем следующий результат Селфриджа.

Теорема 17.1. *Нечетное N удовлетворяет свойству A для данного b тогда и только тогда, когда оно является сильным* b-псп.

Доказательство. Пусть N удовлетворяет свойству A для данного b. Если не существует такого значения l, для которого

$$b^{(N-1)/2^l} \not\equiv 1 \pmod{N},$$

то $b^t \equiv 1 \pmod{N}$. Если такое значение l существует, то $0 < l \leqslant s$ и

$$b^{(N-1)/2^{l-1}} \equiv 1 \pmod{N},$$

т. е.

$$\left(b^{(N-1)/2^l} - 1\right)\left(b^{(N-1)/2^l} + 1\right) \equiv 0 \pmod{N}.$$

Поскольку нод $\left(N, b^{(N-1)/2^l} - 1\right) = 1$,

то

$$b^{(N-1)/2^l} = b^{2^r t} \equiv -1 \pmod{N},$$

где $r = s - 1$. Так как $0 \leqslant r < s$, то N является сильным b-псп.

Теперь предположим, что N — это сильное b-псп. Если $b^t \equiv 1 \pmod{N}$, то $b^{N-1} \equiv 1 \pmod{N}$, и не существует такого l, что

$$b^{(N-1)/2^l} \not\equiv 1 \pmod{N}.$$

Если $b^{2^r t} \not\equiv -1 \pmod{N}$ для некоторого r, такого что $0 \leqslant r < s$, то положим $l = s - r$. Поскольку

$$b^{2^{r+1} t} \equiv 1 \pmod{N},$$

то видно, что l — наименьшее целое число, для которого

$$b^{(N-1)/2^l} \not\equiv 1 \pmod{N}.$$

Поскольку N делит $b^{(N-1)/2^l} + 1$, то видно также, что

$$\text{нод}\left(N, b^{(N-1)/2^l} - 1\right) = 1.$$

Следовательно, N удовлетворяет свойству A для b.

Так как определение сильного b-псп не требует вычисления никаких наибольших общих делителей, то мы предпочитаем идею Селфриджа идее Миллера. Селфридж также показал, что сильное b-псп является эйлеровым b-псп.

Теорема 17.2. *Если N является сильным* b-псп, *то оно также является эйлеровым* b-псп.

Доказательство. Оно основано на идее **Робинсона** [72]. Для начала, если $b^t \equiv 1 \pmod{N}$, то поскольку t нечетно,

$$p^{(p-1)/2} \equiv 1 \pmod{p},$$

где p — любой простой делитель числа N. Отсюда следует, что

$$b^{(N-1)/2} \equiv 1 \equiv (b \mid N) \pmod{N}.$$

Если $b^{2^r t} \equiv -1$, то по теореме 9.1 мы получаем, что $d \mid 2^{r+1} t$ и $d \nmid 2^r t$, где $d = \operatorname{ord}_p(b)$ и p — любой простой делитель N. Поэтому d равно 2^{r+1}, умноженному на нечетное число, и $p = 2^{r+1}k + 1$. Теперь

$$b^{d/2} \equiv -1 \pmod{p}$$

и

$$(b \mid p) \equiv b^{(p-1)/2} \equiv (-1)^{(p-1)/d} \pmod{p};$$

следовательно, $(b \mid p) \equiv (-1)^{(p-1)/2^{r+1}} = (-1)^k$.

Пусть $N = \prod_{i=1}^{m} p_i^{\alpha_i}$, где p_i $(i = 1, 2, \ldots, m)$ — различные простые числа. Поскольку $p_i = 2^{r+1}k_i + 1$, то

$$N = 1 + 2^{r+1} \sum_{i=1}^{m} \alpha_i k_i \pmod{2^{2r+2}};$$

тогда $t 2^{s-1} = (N-1)/2 \equiv 2^r \sum_{i=1}^{m} \alpha_i k_i \pmod{2^{r+1}}$,

$$2^{s-1-r} \equiv \sum_{i=1}^{m} \alpha_i k_i \pmod{2}.$$

Так как

$$b^{(N-1)/2} \equiv (-1)^{2^{s-1-r}} = (-1)^{\sum_{i=1}^{m} \alpha_i k_i} \pmod{N}$$

и

$$(b \mid N) = \prod_{i=1}^{m} (b \mid p_i)^{\alpha_i} = (-1)^{\sum_{i=1}^{m} \alpha_i k_i}$$

то теорема доказана.

Из-за растущей популярности малых вычислительных машин следует упомянуть методы проверки на простоту, пригодные для таких устройств. Некоторые идеи, аналогичные описанным в предыдущих разделах, предложены в работе [77]. Один весьма простой метод требует, однако, составления таблицы всех таких нечетных составных чисел n (не превосходящих некоторой границы B), что n делит $2^{n-1} - 1$. Тогда, если проверяемое N меньше B, $N \mid 2^{N-1} - 1$ и N не содержится в указанной таблице, то N является простым. Эту идею, а также соответствующие таблицы для B вплоть до 2×10^8, предложил Лемер [39, 41]. Трудность реализации этой идеи состоит в том, что составление таблицы требует очень большого труда, а сама таблица очень велика.

Селфридж и Вагштаф (не опубликовано) справились с проблемой размера таблицы, использовав сильные псевдопростые числа. Они после долгих вычислений нашли, что

$N_1 = 2047 = 23 \cdot 89$ наименьшее сильное b-псп для $b = 2$,

$N_2 = 1373653 = 829 \cdot 1657$ „ „ „ „ $b = 2, 3$,

$N_3 = 25326001 = 2251 \cdot 11251$ „ „ „ „ $b = 2, 3, 5$,

$N_4 = 3215031751 = 151 \cdot 751 \cdot 28351$ „ „ „ „ $b = 2, 3, 5, 7$.

N_4 — это единственное сильное b-псп для $b = 2, 3, 5, 7$, которое не превосходит $2{,}5 \times 10^{10}$. Это дает нам легкий тест на простоту для любого $N < 2{,}5 \times 10^{10}$. Достаточно определить, является ли N сильным b-псп для $b = 2, 3, 5, 7$. Если это не так, то N составное; если это так и $N \neq N_4$, то N должно быть простым. С использованием этой идеи мы в результате имеем таблицу всех простых чисел вплоть до $2{,}5 \times 10^{10}$, располагая вычислительной машиной не сложнее программируемого микрокалькулятора.

18. КРИПТОГРАФИЯ

В разд. 1 было отмечено, что новые достижения в области криптографии до некоторой степени стимулировали дальнейшую работу по проверке чисел на простоту. В настоящем разделе мы вкратце опишем эти новые достижения.

Начнем с понятия «открытой» системы шифрования, предложенной в работе [15]. В такой системе использующий ее не пытается скрыть процедуру шифрования E, но хранит в тайне соответствующую процедуру дешифровки. Эти процедуры должны обладать следующими свойствами.

1) Если M — сообщение, то дешифровка зашифрованного сообщения $E(M)$ дает M. Это значит, что $D(E(M)) = M$.

2) E и D легко вычисляются.

3) Обнародовав E, пользователь дает мало (или никакой) информации о D.

4) Если сообщение M сначала дешифровать, а затем зашифровать, то получится M. Это значит, что $E(D(M)) = M$.

Функция E, определенная на множестве сообщений и удовлетворяющая условиям (1), (2) и (3), называется *труднообращаемой функцией*. Если она к тому же удовлетворяет условию (4), то она называется *труднообращаемой перестановкой*.

Это условие (4) очень полезно, если нужно «подписать» сообщение в юридическом смысле. Пусть имеется система, состоящая из некоторого количества пользователей, в которой каждый из них (скажем, K) обнародовал свою систему зашифровки

E_K, но хранит в тайне соответствующую D_K. Теперь предположим, что A и B — два таких пользователя, и A желает послать B «подписанное» сообщение M. Сначала, чтобы подписать свое сообщение, A вычисляет $D_A(M)$; затем, используя принадлежащую B процедуру зашифровки E_B, он определяет $M' = E_B(D_A(M))$. Это сообщение M' он посылает B вместе с неформальным извещением, что он является отправителем. Чтобы убедиться, что отправителем действительно является A, и определить M, B вычисляет $D_B(M') = D_A(M)$ и использует принадлежащую A процедуру зашифровки для отыскания $M = E_A(D_A(M))$. Поскольку лишь A может использовать D_A, этот процесс можно использовать в качестве юридического подтверждения того, что M действительно исходит от A.

Определение функций, удовлетворяющих (1), (2), (3) и (4) — нелегкая задача, однако Райвест, Шамир и Эдлимэн [69] нашли одну такую функцию.

Сначала заметим, что любое сообщение M можно рассматривать как число. Например, мы можем просто заменить каждую букву от A до Z на соответствующее число от 01 до 26, а каждый пробел — на 00. Обозначим эту цифровую форму сообщения символом M. Теперь обнародуем два целых числа R и s, где $R = pq$, причем p и q — большие целые числа (хранящиеся в тайне), а s должно удовлетворять условию $\text{нод}(s, p-1) = \text{нод}(s, q-1) = 1$. Определим

$$E(M) \equiv M^s \pmod{R} \quad (0 < D(M) < R).$$

Это, конечно, означает, что $M < R$. Если $M > R$, то сообщение нужно разбить на части, каждая из которых не больше R. Ясно, что

$$D(M) \equiv M^t \pmod{R} \quad (0 < D(M) < R),$$

где $ts \equiv 1 \pmod{(p-1)(q-1)}$.

Итак, функция E удовлетворяет (1) и в силу свойств степенного алгоритма из разд. 7 удовлетворяет (2). В [69] показано, что задача отыскания t (и тем самым процедуры дешифровки) эквивалентна задаче разложения числа R. Как было показано в разд. 6, для большого (скажем, 150-разрядного) числа эта задача очень трудна; поэтому E является труднообращаемой функцией; более того, легко показать, что она также удовлетворяет условию (4).

Основная сложность использования этой процедуры возникает при попытке найти много больших (65—75-разрядных или даже большего размера) p и q, чтобы все соответствующие значения R были надежными. Как мы видели, легко проверить, что p простое, если у нас есть достаточно делителей чисел $p \pm 1$,

но это может оказаться чрезвычайно трудным делом, если делителей $p \pm 1$ у нас мало. Если у $p \pm 1$ есть только весьма малые простые делители, то возможно применение метода Полларда [60] или метода, упомянутого в [20], для разложения R на множители, а именно этого мы стремимся избежать. Поэтому хотелось бы иметь быстрый и эффективный метод проверки простоты p, когда p не имеет вида, допускающего быструю проверку методами, обсужденными выше. Это привело к разработке *стохастических* (или *Монте-Карло*) методов проверки чисел на простоту. Мы опишем их в следующем разделе.

19. МЕТОДЫ МОНТЕ-КАРЛО

Пусть N — любое нечетное число, и пусть R — приведенное множество вычетов (mod N). Множество

$$G = \{b \in R \mid N \text{ есть эйлерово } b\text{-псп}\}$$

является группой. Далее, если N составное, то G — собственная подгруппа группы R (см. [81, 47]). Поэтому

$$|G| \leqslant \tfrac{1}{2} R,$$

когда N составное. Равенство между $|G|$ и $|R|/2$ действительно имеет место для некоторых чисел Кармайкла, например для 1729. Поэтому если b является случайно выбранным числом из набора $\{1, 2, 3, \ldots, N-1\}$ и N — эйлеровым b-псп, то вероятность ошибочно решить, что N простое, меньше $1/2$.

Теперь мы можем описать тест Соловея и Штрассена [81] на простоту. Выберем из набора $\{1, 2, \ldots, N-1\}$ k случайных значений b и проверим, что N является эйлеровым b-псп для каждого из этих оснований. Если это не так, то N заведомо составное; если это так, то мы можем заявить, что N простое, с вероятностью при этом ошибиться $< 2^{-k}$. Это значит, что в 2^k различных реализациях этого алгоритма мы можем ожидать самое большее одного неверного вывода о простоте N. При $k = 30$ вероятность ошибки становится меньше одной биллионной.

Рабин [62] показал, что если нечетное N составное и

$$S = \{1 \leqslant b \leqslant N-1 \mid N \textit{ не} \text{ является сильным } b\text{-псп}^1)\},$$

то $|S| \geqslant (3/4)(N-1)$. (В [6] показано, что $|S| \geqslant (1/2)(N-1)$. Тест Рабина проводится так. Случайным образом выбираем из набора $\{1, 2, 3, \ldots, N-1\}$ k значений b и проверяем, является ли N сильным b-псп для каждого из этих оснований. Если нет,

[1]) Рабин на самом деле пользуется здесь свойством A_1, но мы видели, что понятие сильного b-псп удобнее.

то мы знаем, что N составное; если да, то можно утверждать, что N простое, с вероятностью ошибиться $\leqslant 4^{-k}$. Кроме того, из теоремы 17.2 видим, что тест Соловея и Штрассена — это лишь более слабая версия теста Рабина.

Малм [51] использовал результат Лемера [45], чтобы получить весьма интересный стохастический тест на простоту.

Лемер показал, что если p — нечетное простое, а b и h — такие целые числа, что $(b\,|\,p)=1$, $(h^2-4b\,|\,p)=1$, то

$$c^2 \equiv b \pmod{},$$

где $2c \equiv V_{(p+1)/2}(h, b)$. (Фактически эту идею определения квадратного корня из b по модулю p первоначально предложил Чиполла (1903).)

В тесте Малма нужно выбрать k случайных пар (h, b) из множества R^2, таких что $(b\,|\,N)=1$ и $(h^2-4b\,|\,N)=1$, и определить, выполняется ли

$$V^2_{(N+1)/2}(h, b) \equiv 4b \pmod{N}.$$

Если это не так, то N составное; если же это выполняется, то можно сказать, что N простое, с вероятностью ошибки меньше $2^{-k(m-1)}$, где m — количество различных простых делителей числа N. Недостаток этого теста состоит в том, что он более трудоемок, чем предыдущие, но при этом вероятность ошибки не снижается. Однако численные результаты, приведенные Малмом, по-видимому, указывают на его большую силу.

Возвращаясь к тесту Рабина, мы видим, что если $k \leqslant 30$, то вероятность ошибки меньше 10^{-18}, что безусловно очень мало. К тому же, увеличив значение k, мы можем сделать эту вероятность сколь угодно малой. Рабин замечает, что в среднем одна ошибка на 10^{18} тестов — это пренебрежимо мало по сравнению с частотой, с которой встречаются другие ошибки при практических вычислениях. Может оказаться, как он предполагает, что этот «тест» на простоту окажется достаточным «для всех практических целей». Действительно, при нормальном ходе вычислений большинство работающих в данной области считают число простым, если оно является b-псп для одного предварительно выбранного значения b (обычно 2 или 13). (См. по этому поводу примечания к [8].) Например, работая с аликвотными последовательностями, Гу и Уильямс [19] считали число простым, если оно было 2-псп. Позднее они пытались доказать простоту этих чисел, и самым большим составным 2-псп, которое они обнаружили, было $2350141 = 31 \cdot 47 \cdot 1613$.

Несмотря на все это, математик все же испытывает неловкость, встречаясь с этими нестрогими методами. В 1929 г. Крайчик [31] использовал нестрогий метод для установления «субъективной убежденности» в том, что число N простое. (См. об-

суждение в работе Лемера [36] и последующие результаты Холла [22].) Хотя он не анализировал свой метод столь же тщательно, как упомянутые здесь авторы анализировали свои, но основная мысль о том, что «мы считаем наши доводы в пользу простоты N достаточными» пронизывает всю эту работу. Остается надеяться, что существование этих методов Монте-Карло не заставит работающих в области проверки чисел на простоту отказаться от дальнейшего развития математически строгих алгоритмов.

20. МЕТОД ГИПОТЕЗ

В [54] Миллер предлагает следующий алгоритм проверки на простоту:

1) Определить, что N не является степенью какого-либо числа, т. е. $N \neq M^s$ для $s = 2$.
2) Проверить, что N является сильным b-псп для всех $b \leqslant f(N)$.

Если (1) или (2) не верно, то N составное. Если (1) верно, а (2) верно для достаточно большого $f(N)$, то N простое. Трудность возникает при попытке найти, каким должно быть $f(N)$. Миллер показал, что $f(N) = O(N^{0,134})$. Это лучше предыдущего результата Полларда [59], но хуже, чем $O(N^{1/8+\varepsilon})$; последний результат Полларда привел в [60].

Слегка модифицировав этот алгоритм и используя результат Анкени [1], Миллер изобрел метод проверки простоты N, требующий $O((\log N)^4)$ операций. С этим алгоритмом связаны две трудности. Первая состоит в том, что результат Анкени, необходимый Миллеру, верен лишь в предположении истинности недоказанной обобщенной гипотезы Римана; вторая — вычислительная неэффективность алгоритма. Селфридж и Вайнбергер улучшили идею Миллера, и в разд. 21 мы изложим суть этой работы.

Сначала вкратце опишем обобщенную гипотезу Римана (О. Г. Р.).

Пусть G — произвольная группа. Комплекснозначная функция g, определенная на G, называется *характером* группы G, если

$$g(ab) = g(a)\,g(b)$$

для всех $a, b \in G$, и $g(c) \neq 0$ для некоторого $c \in G$.

Пусть R — группа приведенных вычетов по модулю k. Для каждого характера g группы R определим функцию $\chi = \chi_g$ следующим образом:

$$\chi(n) = g(m) \quad \text{при } n \equiv m \pmod{k}$$
$$\chi(n) = 0 \quad \text{при } (n, k) \neq 1.$$

Назовем такую функцию χ *характером Дирихле* по модулю k.

Хорошо известная и до сих пор недоказанная гипотеза Римана касается функции

$$\zeta(s) = \sum_{n=1}^{\infty} n^{-s}$$

комплексной переменной s. Гипотеза утверждает, что $\zeta(s) \neq 0$ для всех s, для которых $\mathrm{Re}(s) > \frac{1}{2}$.

Если χ — характер Дирихле, определим *L-функцию Дирихле* комплексной переменной s формулой

$$L(s, \chi) = \sum_{n=1}^{\infty} \chi(n) n^{-s}.$$

О. Г. Р. утверждает, что $L(s, \chi) \neq 0$ для любого s, такого что $\mathrm{Re}(s) > \frac{1}{2}$.

21. ПСЕВДОПРОСТЫЕ ЧИСЛА И ТЕКСТ НА ПРОСТОТУ

Для простого p_k определим псевдоквадрат $F(p_k)$ как наименьшее положительное целое число, такое что символ Якоби

$$(p_i \mid F(p_k)) = 1$$

для всех простых $p_i \leqslant p_k$. Так, если $N < F(p_k)$ и N не является полным квадратом, то должно существовать некое $p_i \leqslant p_k$, для которого

$$(p_i \mid N) \neq 1.$$

Гипотеза, о которой мы упоминали выше, вводя этот метод, относится к размеру $F(p_k)$. Приведем таблицу псевдоквадратов из работы [46] (см. с. 92).

Вайнбергер (не опубликовано) показал (предполагая истинность О. Г. Р.), что если существуют константы c_1, c_2, c_3, такие что

$$p_k > (c_1 \log N + c_2 \log\log N + c_3)^2,$$

то

$$N < F(p_k).$$

В дальнейшем ему удалось получить не очень большие верхние границы для этих констант.

Нужно упомянуть здесь, что Човла (см. [1]) показал, что если p — простое, то наименьший квадратичный невычет $n(p)$ от p превышает $c_4 \log p$, где c_4 некоторая константа, бесконечно часто. Это означает, что существует константа c_5, такая, что для бесконечного числа значений N мы имеем $N \geqslant F(p_k)$, когда

Таблица 7

p_k	$F(p_k)$	p_k	$F(p_k)$
2	7	59, 61	6077111
3	23	67	98538359
5	71	71	120292879
7	311	73, 79	131486759
11	479	83	508095719
13	1559	89	2570169839
17	5711	97	2570169839
19	10559	101	2570169839
23	18191	103	2570169839
29	31391	107	2570169839
31	307271	109	2570169839
37, 41	366791	113, 127	196265095009
43, 47, 53	2155919		

$p_k > c_5 \log N$. Это легко увидеть, если предположить, что P — простое, такое что $n(P) > c_4 \log P$. Так как $n(P)$ простое, положим $p_{k+1} = n(P)$. Мы имеем $(p_i | P) = 1$ для всех $p_i \leqslant p_k$, где p_k — наибольшее простое, меньшее, чем p_{k+1}. Так как $2p_k > p_{k+1}$, то $p_k > c_5 \log F(p_k)$, где $c_5 = c_4/2$. Следовательно, для $p_k > c_5 \log P$ мы имеем из определения $F(p_k)$, что $P \geqslant F(p_k)$. Таким образом, возможно, что результат Вайнбергера близок к наилучшему.

Теперь мы представим тест на простоту Селфриджа и Вайнбергера. Пусть N — нечетное число, подлежащее проверке.

1) Проверить, что N — не является степенью простого числа, т. е. $N \neq p^a$ для некоторого простого p и некоторого $a \geqslant 2$.
2) Попробовать разделить N на все простые числа, меньшие чем фиксированная граница делителей B.
3) Найти p_k, такое что $N/B < F(p_k)$.
4) Для каждого простого $p_i \leqslant p_k$ проверить, что

$$p_i^{(N-1)/2} \equiv \pm 1 \pmod{N},$$

а также что по крайней мере для одного из этих p_i, скажем, p_j

$$p_j^{(N-1)/2} \equiv -1 \pmod{N}.$$

(На практике это выполняется приблизительно для половины чисел p_i.) Если все эти условия выполняются, то N — простое.

Доказательство. Предположим, что число N прошло все вышеперечисленные проверки, и что число N — составное. Тогда

$$N = q_1 q_2 q_3 \cdots q_r,$$

где q_i $(i = 1, 2, 3, \ldots, r)$ — простые, которые не обязательно все различны, но не все равны. Пусть

$$N - 1 = 2^s t \quad (t \text{ нечетно})$$

и

$$q_i = 1 + 2^{s_i} t_i \quad (t_i \text{ нечетно}) \ (i = 1, 2, 3, \ldots, r).$$

Так как

$$p_i^{(N-1)/2} \equiv -1 \pmod{q_i},$$

из теоремы 9.1 мы видим, что $s_i \geqslant s$ для $i = 1, 2, 3, \ldots, r$. Теперь мы рассмотрим два случая.

Случай 1. $s_1 > s$. В этом случае мы имеем $q_1 < N/B$ и должно существовать некоторое простое, скажем, $p_1 \leqslant p_k$, такое что $(p_1 | q_1) \neq 1$. Так как нод$(p_1, N) = 1$, мы имеем $(p_1 | q_1) = -1$ и

$$p_1^{(q_1-1)/2} \equiv -1 \pmod{q_1}.$$

Отсюда следует, что если $d = \operatorname{ord}_{q_1}(p_1)$, то $2^{s_1} | d$. Но так как

$$p_1^{(N-1)/2} \equiv \pm 1 \pmod{q_1},$$

то d должно делить $N - 1$. Это невозможно, когда $s_1 > s$. Так как каждое $s_i \geqslant s$, и никакое s_i не может превысить s, то возможен лишь

Случай 2. $s_1 = s_2 = s_3 = \ldots = s_r = s$. В этом случае N должно быть произведением нечетного числа простых чисел; запишем для числа

$$N = \prod_{i=1}^{r} (1 + 2^s t_i),$$

$$1 + 2^s t \equiv 1 + 2^s \prod_{i=1}^{r} t_i \pmod{2^{s+1}}$$

и

$$t \equiv \sum_{i=1}^{r} t_i \pmod{2}.$$

Так как $t, t_1, t_2, \ldots, t_r$ все нечетные, то r тоже должно быть нечетно.

Если N составное, то $r \geqslant 3$ и по крайней мере два простых числа, которые делят N, должны быть различны. Обозначим их

q_1 и q_2. Теперь $q_1q_2 < N/B$; следовательно, существует некоторое простое число $p_2 \leqslant p_k$, такое, что

$$(p_2 | q_1q_2) = -1.$$

Предположим без уменьшения общности, что $(p_2|q_1) = +1$ и $(p_2|q_2) = -1$; кроме того, $d_1 = \mathrm{ord}_{q_1}(p_2)$ и $d_2 = \mathrm{ord}_{q_2}(p_2)$.
Так как

$$p_2^{(q_2-1)/2} \equiv 1 \pmod{q_2},$$

мы имеем $2^s | d_2$; таким образом, $d_2 \nmid (N-1)/2$ и

$$p_2^{(N-1)/2} \equiv 1 \pmod{q_2}.$$

Так как

$$p_2^{(q_1-1)/2} \equiv 1 \pmod{q_1},$$

то $d_2 | 2^{s-1}t_1$ и так как

$$p_2^{(N-1)/2} \equiv \pm 1 \pmod{q_1},$$

то также выполняется $d_1 | N-1$; следовательно, $d_1 | (N-1)/2$ и

$$p_2^{(N-1)/2} \equiv 1 \pmod{q_1}.$$

Так как

$$p_2^{(N-1)/2} \equiv \pm 1 \pmod{q_1q_2}$$

мы не можем иметь одновременно

$$p_2^{(N-1)/2} \equiv 1 \pmod{q_1} \text{ и } p_2^{(N-1)/2} \equiv -1 \pmod{q_2}$$

Следовательно, N — простое. Одна трудность в применении этого очень элегантного теста состоит в том, что для очень больших N, например, 100-разрядных, даже если $B = 10^{10}$, нам все равно пришлось бы вычислять (даже с учетом результата Вайнбергера) больше чем $\pi((\log 10^{80})^2) = 4391$ различных значений $p_i^{(N-1)/2}$ по модулю N. Это большая работа, и может оказаться, как отмечалось выше, что дальнейшие улучшения не смогут существенно сократить объем вычислений. Тем не менее, метод по-прежнему не требует очень трудоемкого процесса разложения на множители, и это существенно говорит в его пользу.

21. ЗАКЛЮЧЕНИЕ

Мы видим, что было разработано три основных метода проверки чисел на простоту. Обсудим теперь вкратце достоинства и недостатки каждого из них.

1. Проверка на простоту с использованием специальных функций (степенных вычетов, функций Люка, функций Лемера, обобщенных функций Лемера).

Достоинства

1) Этими методами (если не учитывать ошибок вычислений) вопрос о простоте решается в строгом смысле.

2) При реализации на вычислительной машине вопрос о простоте числа часто удается решить очень быстро.

Недостатки

1) В конечном итоге эти методы требуют разложения на простые множители, что часто оказывается очень долгой и трудоемкой процедурой.

2) В некоторых случаях эти тесты весьма сложны, и поэтому их трудно реализовать. Даже когда эти более сложные тесты были реализованы, вероятность сбоя машины или ошибки программиста и оператора выше, чем для других тестов.

2. Методы Монте-Карло

Достоинства

1) Эти методы не требуют разложения на простые множители.

2) Их легко реализовать на вычислительной машине.

3) Они работают очень быстро.

Недостаток

1) Эти методы не дают строгого математического доказательства простоты.

3. Метод гипотез

Достоинства

1) Легкость реализации на вычислительной машине.

2) Все числа, объявленные «простыми» таким тестом, действительно будут простыми, если будет доказана соответствующая гипотеза относительно $F(p_k)$ (например, О. Г. Р.).

Недостатки

1) Эти методы не являются доказательствами простоты, пока не доказана соответствующая гипотеза. Если она окажется неверной, то придется начать все с начала.

2) Эти методы приводят к длительным вычислениям.

Как мы видим, в настоящее время исследования простоты больших чисел находятся на перепутье. Есть несколько различных направлений, по которым можно продвигаться, но ни одно из них не кажется вполне удовлетворительным. Остается надеяться, что недавний рост активности в этом уже обширном поле деятельности будет продолжать стимулировать дальней-

шие поиски по-настоящему эффективного и строгого алгоритма, если такой и в самом деле существует.

Автор глубокого признателен Джону Бриллхарту, Д. Г. и Эмме Лемер, Джону Селфриджу и Дэннэлу Шэнксу за советы и предложения, высказанные при подготовке настоящей статьи. Хочется особо поблагодарить Дэннэла Шэнкса за представление препринта гл. 4 из работы [79], на которой основана значительная часть разд. 8 и 17 нашей работы, и Джона Селфриджа за разрешение опубликовать некоторые из его ранее не опубликованных результатов. Наконец, хочется выразить признательность Д. Мальму, М. Рабину, Р. Ривесту и П. Вайнбергеру за предоставленную возможность ознакомиться с их неопубликованными работами.

ЛИТЕРАТУРА

[1] Ankeny N. C. The least quadratic non-residue. Annals of Math. 55 (1952) 65—72.
[2] Archibald R. C. Remarks on Klein's 'Famous Problems of Elementary Geometry. Amer. Math. Monthly, 21 (1914), 247—259.
[3] Archibald R. C. Mersenne's numbers. Scripta Mathematica, 3 (1935), 112—119.
[4] Bays C. and Hudson R. H. The segmented sieve of Eratosthenes and primes in arithmetic progressions to 10^{12}, BIT, 17 (1977), 121—127.
[5] Bohman J. On the number of primes less than a given limit, BIT, 12 (1972), 576—577.
[6] Brewer B. W. Tests for primality, Duke Math. J., 18 (1951), 757—763.
[7] Brillhart J. Some miscellaneous factorizations. Math. Comp., 17 (1963), 447—450.
[8] Brillhart J. and Selfridge J. L. Some factorizations of $2^n \pm 1$ and related results, Math. Comp., 21 (1967), 87—96; corrigendum, ibid. 751.
[9] Brillhart J., Lehmer D. H. and Selfridge J. L. New primality criteria and factorizations of $2^m \pm 1$. Math. Comp., 29 (1975), 620—647.
[10] Buxton M. and Elmore S. An extension of lower bounds for odd perfect numbers, Abstract. Notices, AMS, 23 (1976), A-55.
[11] Carmichael R. D. On the numerical factors of the arithmetic forms $\alpha^n \pm \beta^n$, Annals of Math. (2), 15 (1913—14), 30—70.
[12] Carmichael R. D. Fermat numbers $F_n = 2^{2^n} + 1$, Amer. Math. Montly, 26 (1919), 137—146.
[13] Chernick J. On Fermat's simple theorem, Bull. Amer. Math Soc., 45 (1939), 269—274.
[14] Crandall R. E. and Penk M. E. A search for large twin prime pairs, Math Comp., to appear.
[15] Diffie W. and Hellman M. New directions in cryptography, IEEE Transactions on information theory, IT-22 (1976), 644—654.
[16] Dickson L. E. History of the theory of numbers, Vol. I, Divisibility and primality, Carnegie Institution, Washington, 1918.
[17] Erdős P. On pseudoprimes and Carmichael numbers, Publ. Math. Debrecen, 4 (1956), 201—206.
[18] Gillies B. Donald. Three new Mersenne primes and a statistical theory, Math. Comp., 18 (1964), 93—95.
[19] Guy Richard K. and Williams M. R. Aliquot sequences near 10^{12}, Congressus Numerantium XII, Proc. 4th Conf. Numerical Math., Winnipeg, 1974, 387—406.
[20] Guy Richard K. How to factor a number. Congressus Numerantium XVI, Proc. Fifth Manitoba Conf. on Numerical Math., Winnipeg, 1976, 49—89.
[21] Hagis Peter, Jr. A lower bound for the set of odd perfect numbers, Math. Comp., 27 (1973), 951—953.

[22] Hall Marshall. Quadratic residues in factorization, Bull. Amer. Math. Soc., 39 (1933), 758—759.
[23] Hallyburton John C. and Brillhart John. Two new factors of Fermat numbers. Math. Comp., 29 (1975), 109—112; Corrigendum, ibid. 30 (1976), 198.
[24] Heath Sir Thomas L. A Manual of Greek Mathematics, Dover Publications, New York, 1963.
[25] Hurwitz Alexander. New Mersenne primes. Math. Comp., 16 (1962), 249—251.
[26] Inkeri K. Tests for primality. Annales Academiae Scientiarum Fennicae Series A, No. 279, 1960.
[27] Jönsson Ingemar. On certain primes of Mersenne type, BIT, 12 (1972), 117—118.
[28] Kaplansky Irving. Lucas's tests for Mersenne numbers. Amer. Math. Monthly, 52 (1945), 188—190.
[29] Knuth Donald E. The Art of Computer Programming, Vol. II, Seminumerical Algorithms, Addison — Wesley, 1969, p. 388 ff. [Имеется перевод: Кнут Д. Искусство программирования для ЭВМ. — Т.II. Получисленные алгоритмы. — М.: Мир, 1977.]
[30] Kraitchik M. Théorie des Nombres, Tome 2, Gauthier-Villars, Paris, 1926, p. 220.
[31] Kraitchik M. Recherches sur la théorie des Nombres, Tome 2, Gauthier—Villars, Paris, 1929.
[32] Lehmer D. H. Tests for primality by the converse of Fermat's theorem. Bull. Amer. Math. Soc., 33 (1927), 327—340.
[33] Lehmer D. H. A further note on the converse of Fermat's theorem. Ibid., 34 (1928), 54—56.
[34] Lehmer D. H. On the number $(10^{23} - 1)/9$. Ibid., 35 (1929), 349—350.
[35] Lehmer D. H. An extended tehory of Lucas' functions. Annals of Math., (2) 31 (1930), 419—448.
[36] Lehmer D. H. A falacious principle in the theory of numvers. Bull. Amer. Math. Soc., 36 (1930), 847—850.
[37] Lehmer D. H. Some new factorizations of $2^n \pm 1$. Bull. Amer. Math. Soc., 39 (1933), 105—108.
[38] Lehmer D. H. On Lucas's test for the primality of Mersenne's numbers. J. London Math. Soc., 10 (1935), 162—165.
[39] Lehmer D. H. On the converse of Fermat's theorem. Amer. Math. Monthly, 43 (1936), 347—354.
[40] Lehmer D. H. A factorization theorem applied to a test for primality. Bull. Amer. Math. Soc., 45 (1939), 132—137.
[41] Lehmer D. H. On the converse of Fermat's theorem II. Amer. Math. Monthly, 56 (1949), 300—309.
[42] Lehmer D. H. Recent discoveries of large primes. MTAC, 6 (1952), 61, 205.
[43] Lehmer D. H. Two new Mersenne primes. Ibid., 7 (1953), 72.
[44] Lehmer D. H. On the exact number of primes less than a given limit. Illinois J. Math., 3 (1959), 381—388.
[45] Lehmer D. H. Computer technology applied to the theory of numbers. In: W. J. LeVeque (ed.). Studies in number theory, M. A. A. Studies in Math., 6 (1969), 117—151.
[46] Lehmer D. H., Lehmer Emma and Shanks D. Integer sequences having prescribed quadratic character. Math. Comp., 24 (1970), 433—451.
[47] Lehmer D. H., Lehmer Emma and Shanks D. Strong Crmichael numbers. J. Aust. Math. Soc. A. 21 (1976), 508—510.
[48] Lehmer D. H. Hunting big game in the theory of numbers. Scripta Math., 1 (1932—33), 229—235.

[49] Lucas E. Théorie das fonctions numériques simplement périodiques. Amer. J. Math., 1 (1878), 184—239, 289—321.
[50] Lucas E. Tpéorie des Nombres. Tome 1, Librarie Blanchard, Paris, 1961.
[51] Maim D. E. G. Monte Carlo tests for primality. unpublished.
[52] Mapes D. C. Fast method for computing the number of primes less than a ginev limit. Math. Comp., 17 (1963), 179—185.
[53] Matthew G. and Williams H. C. Some new primes of the form $k \cdot 2^n + 1$. Math. Comp., 31 (1977), 797—798.
[54] Miller Gary L. Riemann's hypothesis and tests for primality. Jour. Computer and System Sci., 13 (1976), 300—317.
[55] Morrison Michael A. and Brillhart John. The factorization of F_7, Bull. Amer. Math. Soc., 77 (1971), 264.
[56] Morrison Michael A. A note on primality testing using Lucas sequences. Math. Comp., 29 (1975), 181—182.
[57] Morrison M. A. and Brillhart John. A method of factoring and the factorization of F, Math. Comp., 29 (1975), 183—205.
[58] Pocklington H. C. The determination of the prime or composite nature of large numbers by Fermat's theorem. Proc. Camb. Philos. Soc., 18 (1914—1916), 29—30.
[59] Pollard J. M. An algorithm for testing the primality of any integer. Bull. London Math. Soc., 3 (1971), 337—340.
[60] Pollard J. M. Theorems on factorization and primality testing. Proc. Cambridge Philos. Soc., 76 (1974), 521—528.
[61] Rabin M. O. Probabilistic Algorithms. in J. F. Traub (ed.) Algorithms and Complexity, Recent Results and New Direction. Academic Press, New York, 1976, 21—40.
[62] Rabin M. O. Probabilistic algorithm for testing primality, unpublished.
[63] Reile H. J. J. te. Four large amicable pairs, Math. Comp., 28 (1974), 309—312.
[64] Riesel H. A note on the prime numbers of the forms $N = (6a + 1)2^{2n-1} - 1$ and $M = (6a - 1)2^{2n} - 1$. Ark. Mat., 3 (1956), 253.
[65] Riesel H. Mersenne Numbers, MTAC, 12 (1958), 207—213.
[66] Riesel H. A factor of the Fermat number F_{19}, Math. Comp., 17 (1963), 458.
[67] Riesel H. Lucasian criteria for the primality of $N = h \cdot 2^n - 1$. Ibid., 23 (1969), 869—875.
[68] Riesel H. Some factors of the numbers $G_n = 6^{2^n} + 1$ and $H_n = 10^{2^n} + 1$. Ibid., 23 (1969), 413—415; Corrigenda, ibid., 24 (1970), 243.
[69] Rivest Ronald, Shamit Adi and Adleman Len. A method for obtaining digital signatures and public-key cryptosystems, MIT Laboratory for Computer Science, Technical Memo LCS/TM82, 1977.
[70] Robinson Raphael M. Mersenne and Fermat numbers, Proc. Amer. Math. Soc., 5 (1954), 842—846.
[71] Robinson Raphael M. Factors of Fermat numbers, M. T. A. C. (Math. Comp.) 11 (1957), 21—22.
[72] Robinson Raphael M. The converse of Fermat's theorem, Amer. Math Mothly, 64 (1957), 707—710.
[73] Robinson Raphael M. A report on primes of the form $k \cdot 2^n + 1$ and on factors of Fermat numbers, Proc. Amer. Math. Soc., 9 (1958), 673—681.
[74] Rotkiewicz A. Pseudoprime Numbers and their Generalizations. University of Novi Sad, Novi Sad, Yougoslavia, 1972.
[75] Selfridge J. Factors of Fermat numbers. M. T. A. C. (Math. Comp.), 7 (1953), 274—275.
[76] Selfridge J. and Hurwitz Alexander. Fermat numbers and Mersenne numbers, Math. Comp., 18 (1964), 146—148.

[77] Selfridge J. and Guy Richard K. Primality testing with applications to small machines. Proc. Washington State Univ. Conf. on Number Theory, Pullman, 1971, 45—51.
[78] Selfridge J. and Wunderlich M. C. An efficient algorithm for testing large numbers for primality. Congressus Numerantium XII, Proc, 4th Manitoba Conf. on Numerical Math., Winnipeg, 1974, 109—120.
[79] Shanks D. Solved and Unsolved Problems in the Theory of Numbers. 2nd edition, Chelsea, New York, 1978.
[80] Shippee D. E. Four new factors of Fermat numbers, Math. Comp., 32 (1978), to appear.
[81] Solovay R. and Strassen V. A fast Monte-Carlo test for primality. SIAM J. on Computing, 6 (1977), 84—85.
[82] Стечкин С. Б. Критерий Люка для простоты чисел вида $N = h2^n - 1$. Мат. Заметки, 10 (1971).
[83] Tuchkerman Bryant. The 24th Mersenne prime. Proc. Nat. Acad. Sct. USA, 68 (1971), 2319—2320.
[84] Uhler H. S. A brief history of the investigations on Mersenne's numbers and the latest immense primes. Scripta Mathematica, 18 (1952), 122—131.
[85] Ward Morgan. Tests for primality based on Sylvester's cyclotomic numbers. Pacific J. Math., 9 (1959), 1269—1272.
[86] Western A. E. On Lucas's and Pepin's tests for the primeness of Mersenne's numbers. Journal London Math. Soc., 7 (1932), 130—137.
[87] Williams H. C. and Zarnke C. R. A report on prime numbers of the forms $M = (6a+1)2^{2m-1} - 1$ and $M' = (6a-1)2^{2m} - 1$, Math. Comp., 22 (1968), 420—422.
[88] Williams H. C. and Zarnke C. R. An algorithm for determining certain large primes, Congressus Numerantium III, Proc. of the second Louisiana Conf. on Combinatorics, Graph Theory and Computing, Utilitas Mathematica, Winnipeg, 1971, 533—556.
[89] Williams H. C. and Zarnke C. R. The primality of $N = 2A3^n - 1$, Can. Math. Bull., 15 (1972), 585—589.
[90] Williams H. C. and Zarnke C. R. Some prime numbers of the form $2A3^n + 1$ and $2A3^n - 1$, Math. Comp., 26 (1972), 995—998.
[91] Williams H. C. A generalization of Lehmer's functions, Acta Arith., 29 (1976), 315—341.
[92] Williams H. C. and Judd J. S. Determination of the primality of N by using factors of $N^2 \pm 1$, Math. Comp., 30 (1976), 157—172.
[93] Williams H. C. and Judd J. S. Some algorithms for prime testing using generalized Lehmer functions, Math. Comp., 30 (1976), 867—886.
[94] Williams H. C. and Holte R. Some observations on primality testing. Ibid., 32 (1978), to appear.
[95] Williams H. C. Some primes with interesting digit patterns. Ibid., 32 (1978), to appear.
[96] Williams H. C. Some properties of a special set of recurrence sequences, Pacific J. Math., to appear.
[97] Williams H. C. The primality of certain integers of the form $2Ar^n - 1$, submitted for publication.
[98] Wrathall Claude P. New factors of Fermat numbers, Math. Comp., 18 (1964), 324—325.
[99] Wunderlich Marvin C. and Selfridge J. L. A design for a number theory package with an optimized trial division routine, Comm. ACM, 17 (1974), 272—276.
[100] Zarnke C. R. and Williams H. C. Computer determination of some large primes. Congressus Numerantium III, Proc. of the second Louisiana Conf. of Combinatorics, Graph. Theory and Computing, Utilitas Mathematica, Winnipeg, 1971, 563—570.

Свойства решетки NP-множеств, зависящие от оракулов[1]

С. Хомер[2], *В. Маас*[3]

В предположении $P \neq NP$ рассматриваются вопросы, касающиеся решетки NP-миожеств и ее подрешетки P. Показывается, что две задачи, немного более сложные, чем известные задачи расщепления в этой решетке, не могут быть решены средствами, поддающимися релятивизации. Эти две задачи состоят в следующем: каждое ли бесконечное NP-множество содержит бесконечное P-подмножество и существуют ли простые P-миожества. Мы строим несколько оракулов, каждый из которых дает $P \neq NP$ (для релятивизированных P, NP) и к тому же делает вышеупомянутые утверждения либо истинными, либо ложными. В частности, дается положительный ответ на вопрос Беннетта и Гилла [3]: существует ли такой оракул B, что $P^B \neq NP^B$ и каждое бесконечное множество в NP^B содержит бесконечное подмножество из P^B? Построения оракулов проводятся с помощью метода приоритета с конечным числом предпочтений.[4]

1. ВВЕДЕНИЕ

Известно очень немного свойств семейства NP-множеств. Центральные проблемы теории сложности связаны с NP, например, верно ли, что $P = NP$ или $NP = co\text{-}NP$, и ожидают еще своего решения. Наш подход, который дал ряд интересных результатов, направлен на изучение структуры класса NP в предположении, что $P \neq NP$. Такие результаты представляют интерес не только в силу широко распространенной уверенности в том, что $P \neq NP$, но также потому, что рассмотрение следствий этой гипотезы может пролить некоторый свет на саму гипотезу.

В настоящей статье мы рассмотрим вопросы, относящиеся к решетке NP-множеств (где решеточными операциями являются объединение и пересечение множеств) и ее подрешетке P в предположении $P \neq NP$. Один аспект этой решетки уже изучался — это свойства расщепления (см. Лэднер [6]). Сильнейший результат, вероятно, принадлежит здесь Брейдбарту [4].

---

[1]) Homer S., Maass W. Oracle-dependent properties of the lattice of *NP* sets. Theor. Comp. Sci., 1983, 24, № 3, 279—289.

[2]) Department of Mathematics, Boston University, Boston, MA 02215, USA.

[3]) Department of Mathematics, University of California, Berkeley, CA 94720, USA.

[4]) Часто этот метод называют методом приоритета с конечным числом *нарушений. — Прим. перев.*

Он показал, что для каждого бесконечного и кобесконечного рекурсивного множества A существует распознаваемое в реальное время с логарифмической памятью множество B, которое расщепляет A (т. е. каждое из множеств $A \cap B$, $A \cap \bar{B}$, $\bar{A} \cap B$, $\bar{A} \cap \bar{B}$ бесконечно). Отсюда следует, что каждое бесконечное NP- или co-NP-множество можно расщепить на два бесконечных множества того же класса. Таким образом, в решетке NP-множеств, рассматриваемых по модулю конечных множеств, нет максимальных элементов.

Мы исследуем два других свойства NP-множеств: существуют ли в NP «простые» множества и каждое ли бесконечное NP-множество содержит бесконечное P-подмножество? Как и в случае максимальных множеств, простыми NP-множествами являются те, дополнения которых в известном смысле малы [1]). Следовательно, коль скоро решен вопрос о максимальных множествах, естественно рассмотреть вопрос об NP-простых множествах. Мы покажем, что любой ответ на этот вопрос нельзя релятивизировать. А именно, мы построим оракулы, для которых $P \neq NP$ и для одних оракулов NP-простые множества существуют, а для других не существуют.

P-иммунным называется бесконечное множество, которое не содержит бесконечных P-подмножеств. Вопрос о существовании P-иммунных множеств в NP представляет некоторый практический интерес, поскольку полезно иметь практически вычислимую аппроксимацию множеств из NP. В статье [3] Беннетт и Гилл показали, что с вероятностью 1 для оракула A существует бесконечное множество в NP^A, не содержащее ни одного бесконечного подмножества из P^A. Затем они поставили вопрос: существует ли такой оракул B, что $P^B \neq NP^B$ и каждое бесконечное множество в NP^B содержит бесконечное подмножество в P^B? Мы дадим на него утвердительный ответ. Следовательно, всякое рассуждение, решающее в предположении $P \neq NP$ проблему: каждое ли бесконечное NP-множество содержит бесконечное подмножество из P, не релятивизируется [2]).

---

[1]) Соответствующие понятия заимствованы из теории рекурсивно перечислимых множеств при аналогии последних с NP-множествами, при которой рекурсивным множествам соответствуют P-множества. Максимальным называются такое кобесконечное перечислимое множество $M \subseteq \mathbb{N}$ (натуральный ряд), что для всякого перечислимого миожества $M_1 \supseteq M$ одно из множеств $(M_1 \setminus M)$ и $\mathbb{N} \setminus M_1$ конечно. Очевидно, максимальное множество нельзя «расщепить» даже перечислимым множеством (тем более рекурсивным), и это свойство характеризует максимальные множества среди перечислимых. (См. Роджерс Х. Теория рекурсивных функций и эффективная вычислимость. — М.: Мир, 1972, § 12.4). В то время как максимальные перечислимые множества существуют, их NP-аналоги не существуют в силу теоремы Брейдбарта. — *Прим. перев.*

[2]) См. по этому поводу добавление переводчика, а также [114]. — *Прим. перев.*

Эти рассмотрения до некоторой степени аналогичны исследованию решеток рекурсивно перечислимых (р. п.) и рекурсивных множеств. Это исследование привело к открытию новых важных конструкций в теории рекурсий. Задачи о решетке р. п. множеств, аналогичные тем, которые мы ставим для NP-миожеств, решаются легко. Из определений почти сразу следует, что всякое бесконечное рекурсивно перечислимое множество содержит бесконечное рекурсивиое подмножество. Различные построения простых множеств повсеместно встречаются в теории рекурсий. (См. Соаре [7].)

Способы построения оракулов в нашей статье, вообще говоря, более сложны, чем те, что использовались раньше для этих целей (см. [1] или [2]). Эти построения (за исключением теоремы 3.1) основаны на приоритетных конструкциях с конечным числом предпочтений. Построенные в теоремах 3.1, 4.1 и 4.5 оракулы рекурсивны, а оракул из теоремы 3.2 является р. п. множеством.

Мы предполагаем, что, применяя более сложные построения чем построения оракулов настоящей статьи, можно показать, что даже в предположении $P \neq NP$ на некоторые непосредственные вопросы о P-степенях в NP нельзя дать ответы с помощью релятивизируемых рассуждений (например, доказательств из теорем рекурсий [1]).

В следующем разделе мы представим основные определения и обозначения. Раздел 3 содержит построение такого оракула A, что $NP^A = co\text{-}NP^A$ и некоторое бесконечное множество в NP^A не содержит бесконечных подмножеств из P^A, и такого оракула A, что существует NP^A-простое множество. В разд. 4 строится такой оракул B, что $P^B \neq NP^B$, и каждое бесконечное множество в NP^B содержит бесконечное подмножество в P^B. Наконец, тот же метод применяется для построения такого оракула B с $P^B \neq NP^B$, что не существует NP^B-простых множеств. В разд. 5 мы показываем, что предыдущие аргументы можно применить для установления независимости утверждения: «В NP существует P-универсальное миожество» от утверждения $P \neq NP$ (в том же, что и выше, смысле).

2. ОПРЕДЕЛЕНИЯ

Рассмотрим вычисления на машинах Тьюринга с оракулом. Без уменьшения общности предположим, что ленточный алфавит наших машин есть $\Sigma = \{0, 1\}$. Наши языки будут подмножествами $\Sigma^* = \{$конечные слова в алфавите $\Sigma\}$ [2]).

[1]) См. добавление.

[2]) Термины «слово» и «цепочка» являются у нас синонимами и употребляются на равных правах. — *Прим. перев.*

Фиксируем перечисления $\{P_i\}_{i\in\mathbb{N}}$ и $\{N_i\}_{N\in\mathbb{N}}$ ($\mathbb{N}$ обозначает натуральный ряд) соответственно детерминированных и недетерминированных машин Тьюринга с оракулом и полиномиально ограниченным временем вычисления. Можно предположить, что $p_i(n)=i+n^i$ является строгой верхней границей длины всякого вычисления машины P_i или N_i с любым оракулом X на входах длины n. P_i^X и N_i^X обозначают машины Тьюринга с оракулом X. Через P_i^X обозначим множество $\{\alpha\in\Sigma^*\mid$ машина P_i^X допускает α (т. е. машина P_i^X дает выход 0 на входе $\alpha)\}$. Аналогично обозначаем через N_i^X множество $\{\alpha\in\Sigma^*\mid N_i^X$ допускает $\alpha\}$. P^X обозначает семейство всех множеств P_i^X, $i\in\mathbb{N}$, а NP^X обозначает семейство всех множеств N_i^X, $i\in\mathbb{N}$. Более полный перечень этих определений читатель найдет в [5].

Для любой цепочки s через s^n обозначим слово s, повторенное n раз. Выражение $|\cdot|$ применяется для обозначения как длины слова, так и мощности множества в зависимости от контекста. Наконец, используется рекурсивная спаривающая функция $\langle\cdot,\cdot\rangle$ на множестве натуральных чисел. Потребуем, чтобы спаривающая функция была взаимно однозначной и при фиксированном первом аргументе строго монотонной по второму аргументу.

3. ОРАКУЛЫ, ОТНОСИТЕЛЬНО КОТОРЫХ СУЩЕСТВУЮТ P-ИММУННЫЕ И NP-ПРОСТЫЕ МНОЖЕСТВА

Беннетт и Гилл [3] уже показали, что с вероятностью 1 для случайного оракула A существует бесконечное множество в NP^A, которое не содержит бесконечных подмножеств из P^A, и $\mathrm{NP}^A\neq\mathrm{co\text{-}NP}^A$.

Мы здесь показываем, что можно также прямо построить такой рекурсивный оракул A, что некоторое множество в NP^A не содержит бесконечных подмножеств из P^A. Соответствующие требования R_i не пропускают в A много элементов (вообще говоря, бесконечно много для некоторого R_i). Поэтому наиболее интересной является конструкция, которая комбинирует эти требования с другими, обеспечивающими $\mathrm{NP}^A=\mathrm{co\text{-}NP}^A$ и вынуждающими отнесение большого количества элементов к A. В этом построении нет изменений условий.

Теорема 3.1. *Существует такой рекурсивный оракул* A, *что некоторое бесконечное множество в* NP^A *не содержит бесконечных подмножеств из* P^A, *и* $\mathrm{NP}^A=\mathrm{co\text{-}NP}^A$.

Доказательство. Для построения такого A, что некоторое множество из NP^A не содержит бесконечных подмножеств из P^A,

достаточно сделать множество $M = \{0^k \mid k \in \mathbb{N} \& (\exists \alpha \in \Sigma^*)(|\alpha| = k \wedge \alpha \in A)\}$ бесконечным и обеспечить для каждого i, чтобы имело место

$$R_i\colon\ P_i^A \cap \{0^k \mid k \in \mathbb{N}\} \text{ бесконечно} \rightarrow P_i^A \not\subseteq M$$

(очевидно, что $M \in \mathrm{NP}^A$).

Согласно работе Бэйкера, Гилла и Соловая [1], для $\mathrm{NP}^A = \mathrm{co\text{-}NP}^A$ достаточно, чтобы дополнение множества $K(A) = \{\langle i, \alpha, 0^n\rangle \mid$ некоторое вычисление машины N_i^A допускает слово α меньше чем за n шагов$\}$ лежало в NP^A (множество $K(A)$ полиномиально полно в NP^A).

Построение

Шаг k. Пусть A_k обозначает множество элементов, уже имеющихся в A в начале шага k.

Для каждого $i \leqslant (1/8)\cdot k$, такого, что $p_i(k) < 2^{k/8}$, в A не пропускаются все слова длины $\geqslant k$, которые не принадлежат A_k и которые запрашиваются при вычислении машины $P_i^{A_k}$ на входе 0^k. Затем, если для одного из этих i требование R_i не рассматривалось и $P_i^{A_k}$ допускает слово 0^k, то в A не пропускаются все слова длины k (что обеспечивает $0^k \notin M$). Мы говорим тогда, что требование R_i рассматривается на шаге k.

Наконец, для каждого слова $\alpha \notin K(A_k)$, такого, что существует такая цепочка β длины k, что β продолжает слово α, β не пропущена в A, $2|\alpha| < |\beta| \leqslant 4|\alpha|$ и в β на месте $|\alpha|+1$ стоит 1, а на местах $|\alpha|+2, \ldots, 2|\alpha|$ стоит 0, мы помещаем цепочку β в A. (Тогда мы снова можем получить α из кода β, беря первую половиину слова β и отбрасывая 1 и все буквы 0 после первой половины слова.)

Заметим сначала, что существует такое k_0, что для всех $k \geqslant k_0$ самое большее $2^{k/4}$ слов длины k не пропущены в A по первому условию. Результат этого условия состоит в том, что когда некоторая машина P_i^A допускает бесконечно много слов вида 0^k, то существует бесконечно много таких k, что $P_i^{A_k}$ допускает 0^k. Поэтому условие R_i рассматривается на некотором шаге, и, таким образом, оказывается, что $P_i^A \not\subseteq M$.

Дальше легкое вычисление доказывает, что для всякого слова α имеет место $\alpha \notin K(A)$ тогда и только тогда, когда в A существует некоторое β, находящееся в таком отношении к α, как указано в построении. Здесь используется то обстоятельство, что в не более чем $k/8$ шагов из первых k шагов построения все слова длины k помещаются в A в силу второго условия. Поэтому $\Sigma^* - K(A) \in NP^A$.

Заметим, что в этом построении мы не всегда могли поместить код β длины $2|\alpha|$ слова $\alpha \in K(A)$ в множество A, как в [1], ввиду сильного удерживающего воздействия условий R_i.

Теорема 3.2. *Существует такой оракул A, для которого существует NP^A-простое множество.*

Доказательство. Для всякого оракула A множество $M = \{0^l \mid l \in N \& \exists \alpha \in \Sigma^* (|\alpha| = l \wedge \alpha \in A)\}$, очевидно, принадлежит классу NP^A.

Мы построим A таким образом, что для каждого $i \in \mathbb{N}$ и каждого $n \in \mathbb{N}$ выполняются условия

$$R_i\colon\ N_i^A \cap \{0^l \mid l \in \mathbb{N}\} \text{ бесконечно} \to N_i^A \cap M \neq \varnothing$$

и

$$S_n\colon\ \{l' \mid A \text{ не содержит слов длины } l'\} \geqslant n.$$

Это немедленно повлечет за собой то, что множество $S = M \cup \cup \{\alpha \in \Sigma^* \mid (\forall l \in \mathbb{N})(\alpha \neq 0^l)\}$ является NP^A-простым.

Имеют место очевидные конфликты между требованиями R_i, стремящимися добавить элементы к множеству A, и требованиями S_n, «стремящимися не впускать» элементы в множество A. Эти конфликты разрешаются приписыванием всем требованиям приоритетов. Мы позволяем только, чтобы R_i нарушало S_n (изменением множества первых n длин l', таких, что ни одно слово длины l' не принадлежит A), если $i < n$, т. е. если R_i имеет более высокий приоритет, чем S_n.

Построение. Будем говорить, что требование R_i выполняется в начале k-го шага, если найдется шаг $k' < k$ такой, что R_i рассматривалось на шаге k', и ни одна цепочка, не пропущенная в A по R_i на шаге k', не была до сих пор помещена в A.

Обозначим через A_k множество элементов, взятых в A к началу шага k.

Шаг k. Проверяем, существует ли такое $i \leqslant k$, что условие R_i не выполнено, и существуют $l \leqslant k$, $\alpha \in \Sigma^*$ с $|\alpha| = l$ и допускающее вычисление $N_i^{A_k}$ на входе 0^l, такие, что слово α не запрашивалось в этом вычислении и не было задержано относительно A по некоторому $R_{i'}$ с $i' < i$ и $|\{l' < l \mid A_k$ не содержит цепочек длины $l'\}| \geqslant i$. Если такие i, l, α существуют, выберем среди них минимальное i и для этого i — минимальное $\langle l, \alpha\rangle$. Будем говорить, что R_i рассматривается на шаге k. Помещаем α в A и не пропускаем в A, согласно R_i, все слова из $\Sigma^* - A_k$, которые запрашивались в некотором канонически выбираемом допускающем вычислении машины $N_i^{A_k}$ на входе 0^l, в котором не запрашивалось α.

Тривиальной индукцией по i доказывается, что каждое требование R_i рассматривается только на конечном числе шагов (заметим, что R_i может рассматриваться на шагах k_1 и k_2, где $k_1 < k_2$, если только существует некоторое $i' < i$, такое, что $R_{i'}$ рассматривается на некотором шаге k' с $k_1 < k' < k_2$). Отсюда уже следует, что выполняется каждое требование S_n, поскольку S_n может быть исправлено на шаге k, если только на шаге k рассматривается некоторое условие R_i с $i < n$. Таким образом, множество $\{0^l \mid l \in N\} - M$ бесконечно.

Лемма 3.3. *Для каждого* $i \in \mathbb{N}$

$$N_i^A \cap \{0^l \mid l \in \mathbb{N}\} \text{ бесконечно} \rightarrow N_i^A \cap M \neq \varnothing.$$

Доказательство. Пусть множество $N_i^A \cap \{0^l \mid l \in \mathbb{N}\}$ бесконечно. Выберем такое $0^l \in N_i^A$, что

$$|\{l' < l \mid A \text{ не содержит слов длины } l'\}| \geqslant i,$$

ни одно слово длины l не задерживается относительно A по условию $R_{i'}$ с $i' < i$, и некоторое допускающее вычисление машины N_i^A на входе 0^l не запрашивает никакую цепочку длины l (последнее имеет место, когда $p_i(l) < 2^l$).

Возьмем такое k_0, что ни одно условие $R_{i'}$ с $i' \leqslant i$ не рассматривается на шаге $\geqslant k_0$. Тогда требование R_i постоянно выполняется, начиная с шага k_0, поскольку в противном случае существование l с вышеупомянутыми свойствами гарантирует, что R_i рассматривается после шага k_0. Таким образом, $N_i^A \cap M \neq \varnothing$.

4. ОРАКУЛЫ, ОТНОСИТЕЛЬНО КОТОРЫХ НЕ СУЩЕСТВУЮТ НИ *P*-ИММУННЫЕ, НИ NP-ПРОСТЫЕ МНОЖЕСТВА

Теорема 4.1. *Существует оракул* B, *такой, что* $\mathrm{P}^B \neq \mathrm{NP}^B$ *и такой, что каждое бесконечное множество в* NP^B *имеет бесконечное подмножество из* P^B.

Доказательство. Мы построим оракул B и для каждого $i \in N$ детерминированную машину Тьюринга Q_i с оракулом, такую, что

$$Q_i^B \subseteq N_i^B \& N_i^B \text{ бесконечно} \rightarrow Q_i^B \text{ бесконечно}.$$

Определим $Q_i^B = \{\alpha \in \Sigma^* \mid t_{i,\alpha} \in B\}$, где $t_{i,\alpha} \in \Sigma^*$ является «проверочным словом», связанным с α равенством $t_{i,\alpha} = \alpha 10^i 10^n$, где $n = |\alpha| + i + 2 + p_i(|\alpha|)$ (p_i есть полином, ограничивающий время работы N_i).

Очевидно, что для каждого $i \in \mathbb{N}$ существует детерминированная машина Тьюринга, работающая за полиномиальное вре-

мя и дающая на входе α выход $t_{i,\alpha}$. Поэтому $Q_i^B \in P^B$ для всякого $B \subseteq \Sigma^*$.

Проверочные слова $t_{i,\alpha}$ выбираются так, что недетерминированная машина N_i не может запрашивать оракул о цепочке $t_{i,\alpha}$ на протяжении всего вычисления на входе α (так как $|t_{i,\alpha}| > p_i(|\alpha|)$). Далее, функция $\langle \alpha, i\rangle \to t_{i,\alpha}$ является взаимно однозначной. Заметим наконец, что для каждого $l \in \mathbb{N}$ множество $F_l = \{\beta \in \Sigma^* \mid |\beta| = l \,\&$ последние $[l/2]$ элементов слова β не все суть нули$\}$ не содержит цепочек вида $t_{i,\alpha}$. Далее, F_l содержит не менее чем $(2^{[l/2]} - 1)$ элементов и, таким образом, функция $l \to |F_l|$ мажорирует каждый полином, начиная с некоторого момента.

Зададим множество $M \in \mathrm{NP}^B$ посредством

$$M = \{0^l \mid l \in \mathbb{N} \,\&\, F_l \cap B \neq \varnothing\}.$$

Для каждого $j \in \mathbb{N}$ имеется условие

$$S_j\colon\ M \neq P_j^B.$$

Если выполнены все условия S_j, то $M \notin \mathrm{P}^B$.

Кроме того, для всех $i \geqslant 0$, $n > 0$ должны быть выполнены требования

$$R_{i,n}\colon\ N_i^B \text{ бесконечно} \to |Q_i^B| \geqslant n.$$

Припишем приоритет $\langle j, 0\rangle$ требованию S_j и приоритет $\langle i, n\rangle$ требованию $R_{i,n}$. Следуя обычному соглашению, будем говорить, что требование T' имеет более высокий приоритет, чем требование T, если (приоритет T') $<$ (приоритет T).

В дальнейшем построении оракула B мы время от времени пытаемся удовлетворить требованию S_j на некотором шаге построения, а затем видим, что должны пожертвовать этой попыткой, чтобы выполнить требование с более высоким приоритетом, чем у S_j. Тем не менее мы в состоянии удовлетворить каждое требование S_j, поскольку оно может быть отвергнуто только конечное число раз (не больше чем по одному разу для каждого условия $R_{i,n}$ с более высоким приоритетом, чем у S_j). Таким образом, нам надо быть достаточно настойчивыми в попытках выполнить условие S_j.

Очевидно, что построенный оракул B рекурсивен, так как мы перечисляем слова в B в порядке возрастания их длины.

Построение. Будем говорить, что условие S_j выполнено в начале k-го шага ($k \in \mathbb{N}$), если найдется шаг $k' < k$, на котором рассматривалось условие S_j и такой, что ни одно слово, не пропущенное в B на шаге k' по условию S_j, не было отнесено к B до k-го шага.

Будем говорить, что условие $R_{i,n}$ выполнено в начале k-го шага, если найдется шаг $k' < k$, на котором рассматривалось условие $R_{i,n}$.

Шаг k. Пусть B_k обозначает множество элементов, уже отнесенных к B в начале шага k. Определим $l_k = \max[\{k\} \cup \{\text{длины всех цепочек из } B_k \text{ или не пропущенных в } B \text{ на предыдущих шагах}\}] + 1$. Возьмем такое минимальное j, что условие S_j не выполнено и $p_j(l) < |F_{l_k}|$.

Случай 1. Существуют не выполняющееся требование $R_{i,n}$ с более высоким приоритетом, чем у S_j, и такая цепочка α, что $\alpha \in N_l^{B_k}$, $|t_{i,\alpha}| \leqslant l_k$, слово $t_{i,\alpha}$ не было задержано относительно B по требованию с более высоким приоритетом, чем $R_{i,n}$, и $|t_{i,\alpha}| > \max\{|\beta| \mid \beta \in B_k\}$.

Выберем $R_{i,n}$ и α с такими свойствами, что $R_{i,n}$ имеет наибольший возможный приоритет, и отнесем $t_{i,\alpha}$ к B. Скажем, что условие $R_{i,n}$ рассматривалось на шаге k.

Случай 2. Всё остальное.

В этом случае S_j рассматривается на шаге k. Не пропускаются в B по требованию S_j все слова, о которых запрашивался оракул в течение вычисления $P_j^{B_k}$ на входе 0^{l_k}. Далее, если $P_j^{B_k}$ не допускает 0^{l_k}, отнесем к B первую в алфавитном порядке цепочку $\beta \in F_{l_k}$ такую, которая не задерживается относительно B по требованию S_j. Если $P_j^{B_k}$ допускает 0^{l_k}, все слова из F_{l_k} ие пропускаются в B по требованию S_j.

Лемма 4.2. $M = \{0^l \mid l \in \mathbb{N} \,\&\, F_l \cap B \neq \varnothing\} \notin \mathrm{P}^B$.

Доказательство. Заметим сначала, что каждое требование $R_{i,n}$ рассматривается самое большее один раз за все построение. Дальше условие S_j может рассматриваться только на шагах k_1, k_2 с $k_1 < k_2$, если некоторое условие $R_{i,n}$ с $\langle i, n\rangle < \langle j, 0\rangle$ рассматривается на некотором шаге k' с $k_1 < k' < k_2$. Поэтому каждое требование S_j рассматривается во всем построении лишь конечное число раз.

Фиксируем некоторое $j \in \mathbb{N}$. Для доказательства $M \neq P_j^B$ рассмотрим достаточно далекий шаг k такой, что ни одно требование с приоритетом $\leqslant \langle j, 0\rangle$ не будет рассмотрено на шагах $\geqslant k$ и такой, что $p_j(l_k) < |F_{l_k}|$. Тогда S_j выполнено в начале шага k, потому что иначе некоторое условие с приоритетом $\leqslant \langle j, 0\rangle$ рассматривалось бы на шаге k. Таким образом, найдется шаг $k' < k$, на котором рассматривается S_j и такой, что ни одно слово, не пропущенное в B на шаге k' по условию S_j, не отнесено к B в начале шага k. По выбору k ни одно из этих задержанных слов не помещается в B на любом шаге $\tilde{k} > k$

(так как ни одно условие $R_{i,n}$ с приоритетом, большим чем у S_j, не рассматривается на шагах $\tilde{k} \geqslant k$). Это влечет, что P_j^B допускает 0^{l_k} тогда и только тогда, когда $P_j^{B_{k'}}$ допускает $0^{l_{k'}}$, что равносильно $0^{l_{k'}} \notin M$.

Лемма 4.3. *Для каждого* $i \in \mathbb{N}$

$$Q_i^B \subseteq N_i^B \text{ и } (N_i^B \text{ бесконечно}) \to (Q_i^B \text{ бесконечно}).$$

Доказательство. Цепочка $t_{i,\alpha}$ помещается в B на шаге k, если только некоторое условие $R_{i,n}$ рассматривается на шаге k. В этом случае мы имеем $\alpha \in N_i^{B_k}$, и поэтому $\alpha \in N_i^B$ так как лишь слова длины $\geqslant |t_{i,\alpha}| > p_i(|\alpha|)$ относятся к B на шагах $\geqslant k$, и машина N_i не может запрашивать свой оракул о таких длинных словах за все время вычисления на входе α. Таким образом, $Q_i^B \subseteq N_i^B$.

Предположим, что множество N_i^B бесконечно. Мы покажем, что для каждого $n \in \mathbb{N}$ условие $R_{i,n}$ будет рассматриваться на некотором шаге. Отсюда следует, что множество Q_i^B бесконечно, так как на каждом шаге, на котором рассматривается требование, мы помещаем в Q_i^B новый элемент. Итак, фиксируем некоторый элемент $n \in \mathbb{N}$. Выберем такое $\alpha \in N_i^B$, что лишь слова длины меньшей, чем $|t_{i,\alpha}|$, удерживались от B по требованиям с приоритетом $\leqslant \langle i, \alpha \rangle$ за время всего построения, и такое, что на первом шаге k, на котором цепочка длины $\geqslant |t_{i,\alpha}|$ относится к B, рассматривается требование с приоритетом $\geqslant \langle i, n \rangle$. Поскольку для этого k только элементы длины $\geqslant |t_{i,\alpha}| > p_i(|\alpha|)$ помещаются в B на шагах $\geqslant k$, то $\alpha \in N_i^{B_k}$. Далее, ввиду того что некоторая цепочка длины $\geqslant |t_{i,\alpha}|$ помещается в B на шаге k, мы имеем $|t_{i,\alpha}| \leqslant l_k$. Поэтому $R_{i,n}$ рассматривается на этом шаге k, если оно не рассматривалось ни на одном предыдущем шаге.

Следствие 4.4. *Аргументов, которые допускают релятивизацию* [1]*, недостаточно для доказательства того, что*

$\mathrm{P} \neq \mathrm{NP} \to$ *каждое бесконечное множества в* NP *содержит некоторое бесконечное подмножество из* P *или*

$\mathrm{P} \neq \mathrm{NP} \to$ *не каждое бесконечное множество в* NP *содержит бесконечное подмножество из* P.

Доказательство. Для доказательства первого утверждения рассмотрим оракул теоремы 3.1, а для второго утверждения — оракул теоремы 4.1.

[1]) См. примечание 2 на с. 101. — *Прим. перев.*

Теорема 4.5. *Существует оракул B, такой, что* $\mathrm{NP}^B \neq \mathrm{co\text{-}NP}^B$ *и каждое бесконечное множество в* $\mathrm{NP}^B \cup \mathrm{co\text{-}NP}^B$ *содержит бесконечное подмножество из* P^B.

Доказательство. Построение требуемого оракула является несущественным расширением конструкции в доказательстве теоремы 4.1. Нам надо обеспечить, чтобы множество $M = \{0^l \mid l \in \mathbb{N} \text{ и } F_l \cap B \neq \varnothing\}$ не принадлежало классу $\mathrm{co\text{-}NP}^B$. Таким образом, если условие S_j рассматривается на шаге k, то мы делаем $F_{l_k} \cap B \neq \varnothing$ тогда и только тогда, когда $0^{l_k} \in N_j^{B_k}$. Если $0^{l_k} \in \mathbb{N}_j^{B_k}$, не пропускаем в B по условию S_j все слова, которые запрашивались в наикратчайшем допускающем вычислении $N_j^{B_k}$ на входе 0^{l_k}. С другой стороны, если $0^{l_k} \in \mathbb{N}_j^{B_k}$, не пропустим в B все цепочки, запрашиваемые при каком-либо вычислении машины N_j^B на входе 0^{l_k}.

Помимо множеств $Q_i^B \subseteq N_i^B$ строятся бесконечные подмножества $\tilde{Q}_i^B$ из P^B для каждого бесконечного множества $(\Sigma^* - N_i^B)$ из $\mathrm{co\text{-}NP}^B$. Возьмем «проверочные слова» $\tilde{t}_{i,\alpha}$, отличные от $t_{i',\alpha'}$ и положим $\tilde{Q}_i^B = \{\alpha \mid \tilde{t}_{i,\alpha} \in B\}$.

Если условие $\tilde{R}_{i,n}$: $\Sigma^* - N_i^B$ бесконечно $\rightarrow |\tilde{Q}_i^B| \geqslant n$ рассматривается на шаге k, помещаем некоторое $\tilde{t}_{i,\alpha}$ в B с $\alpha \notin N_i^{B_k}$. Как и прежде, имеем $\alpha \notin N_i^B$, поскольку лишь слова длины $\geqslant |\tilde{t}_{i,\alpha}| > p_i(|\alpha|)$ попадают в B на шагах $\geqslant k$.

Следствие 4.6. *Аргументы, допускающие релятивизацию, недостаточны для того, чтобы доказать, что* $P \neq NP \rightarrow$ *в решетке* NP*-множеств существует* NP*-простое множество или* $P \neq \mathrm{NP} \rightarrow$ *в решетке* NP*-множеств не существует* NP*-простых множеств.

Доказательство. Для первого утверждения рассмотрим оракул теоремы 4.5, а для второго — оракул теоремы 3.2.

5. P-УНИВЕРСАЛЬНЫЕ МНОЖЕСТВА В NP

Технику предыдущих параграфов можно применить к изучению других вопросов. Назовем множество U P^A-универсальным, если

$$\mathrm{P}^A = \{\{\alpha \in \Sigma^* \mid \langle \gamma, \alpha \rangle \in U\} \mid \gamma \in \Sigma^*\}.$$

Вопрос о существовании в NP P-универсального множества U является открытым. Существование такого множества означало бы, что, используя недетерминизм в вычислениях с полиномиальным временем, можно было бы разрешать все множества из P с фиксированной полиномиальной оценкой времени распозна-

вания U в NP. Аналогия с рекурсивной теорией внушает мысль о существовании P-универсального множества в NP. Многие специалисты полагают, что верно противоположное утверждение.

Метод теоремы 3.1 можно использовать для доказательства следующей теоремы.

Теорема 5.1. *Существует такой рекурсивный оракул A, что $P^A \neq NP^A$ и в NP^A существует P^A-универсальное множество.*

С другой стороны, имеет место следующая

Теорема 5.2. *Существует рекурсивный оракул B, такой, что $P^B \neq NP^B$ и в NP^B не существует P^B-универсального множества.*

Доказательство. Построение здесь аналогично построению из доказательства теоремы 4.1. Помимо обычного множества M, которое свидетельствует, что $NP^B \neq P^B$, для каждого множества N_i^B в NP^B строится такое множество Q_i^B из P^B, что $\forall\gamma\exists\alpha$ $(\langle\gamma, \alpha\rangle \in N_i^B \leftrightarrow \alpha \notin Q_i^B)$.

Добавление переводчика „О релятивизации и сложности вычислений"

1. NP-ПРОБЛЕМА

Статья касается наиболее злободневной и интригующей проблемы сложности вычислений $P \stackrel{?}{=} NP$ или проблемы перебора. Впервые вопрос о неизбежности перебора в решении некоторых алгоритмических задач (а именно, минимизации контактных схем) был поставлен в [8*], где была сделана попытка уточнения проблемы и установления неизбежности перебора. Общепринятое в настоящее время определение проблемы перебора как NP-полной проблемы появилось в разных вариантах в начале 70-х гг., и была доказана NP-полнота ряда важных комбинаторных задач логики и кибернетики [9*, 10*, 11*]. Число таких задач быстро росло, что вместе с обманчивой простотой формулировки и реальной трудностью и важностью вопроса привлекло к этой проблеме внимание многих исследователей [12*, 12а*, гл. 10, 11]. Бросалась в глаза аналогия с рекурсивной теорией, при которой NP-множества и предикаты соответствуют рекурсивно перечислимым (р. п.) (а P-множества — рекурсивным) множествам, но квантор существования для

NP-предикатов берется по словам длины, ограниченной некоторым полиномом от длины входного слова [1]).

Универсальному р. п. множеству соответствует NP-полное множество слов U, которое определяется в алфавите $\{0, 1, \#\}$ как множество слов вида $x\#y$, где $x, y \in \{0,1\}^*$, (вычислимо) кодирует программу Π_x недетерминированной машины Тьюринга M_x, которая, работая время $t_x(y) \leqslant p(|x|, |y|)$ (p — фиксированный полином), допускает слово y. Доказывается, что всякое NP-множество многозначно P-сводится (т. е. детерминированно вычислимой за полиномиальное время функцией f) к U. Так же, т. е. P-многозначно, обычно удавалось сводить этот NP-полный предикат к конкретному NP-полному предикату, например, к SAT-предикату выполнимости формул классического исчисления высказываний. Однако диагональную процедуру доказательства того, что универсальное р. п. множество не является рекурсивным (см. [13*, 14*]) не удается перенести на NP-полные множества. Более того, не удается доказать аналог теоремы Поста о том, что р. п. множество является рекурсивным, если его дополнение также р. п. В [1] было доказано, что для вычислений даже с рекурсивными оракулами возможно как $NP^A = P^A$, так и $P^A \subsetneq NP^A$ (в зависимости от оракула A). Это дает основание полагать, что обычный диагональный метод, как известно, поддающийся релятивизации, [14*, §§ 9.1—9.4] не достаточен для решения NP-проблемы [2]). Это напоминает ситуацию с независимыми утверждениями в теории множеств (ZF), арифметике Пеано (PA), евклидовой геометрии без аксиомы о параллельных (абсолютной геометрии) и другнх неполных теориях, допускающих существенно разные модели.

Были сделаны некоторые попытки такого рода (см. 15*, 15a*, 15b*], где приведена библиография. Однако независимость устанавливалась либо в очень слабых подсистемах PA и при дополнительных ограничениях, либо зависела не от сути задач, а от их специфического представления. Впрочем, даже если бы была доказана независимость $P \overset{?}{=} NP$, например в PA, остался бы открытым наиболее важный для приложений вопрос об истинности $P = NP$ (в стандартной модели PA), которую можно пытаться доказывать в более сильных, чем PA системах.

[1]) В дальнейшем эта аналогия была продолжена, когда в [63a*] была определена полиномиальная иерархия предикатов с полиномиально ограниченными чередующимися кванторами $\forall$ и $\exists$, соответствующая арифметической нерархии [14*, гл. 14, 15]. К тому же оказалось, что $NP\left(\Sigma_n^B\right) = \Sigma_{n+1}^p$, что подтверждало эту аналогию. Здесь $P(\Omega)$ ($NP(\Omega)$) обозначают классы (недетерминированно) вычислимых за полиномиальное время предикатов с оракулами из класса Ω. В некоторых работах полиномиальную вычислимость даже рассматривают как рекурсивность в конечных моделях [12c*].

[2]) См., однако, статью [114]. — *Прим. перев.*

Было бы очень интересно дать точное определение того, что определенное доказательство не использует (или использует) тот или иной метод, скажем, диагональную конструкцию, или метод приоритета [17*], применяемый в статье Хомера и Мааса, или форсинг, поскольку тот или иной метод может неявно использоваться в решении задачи, как это имеет место, например, с диагональным методом в теореме о неподвижной точке, иначе называемой второй теоремой о рекурсии (см. [13*, гл. 11]), или с форсингом в доказательствах независимости континуумгипотезы с помощью булевозначных моделей. В этом смысле представляет интерес статья [16*], где рассматривается устранение или упрощение приоритетных конструкций [17*] при исследовании степеней неразрешимости сложностными методами.

Импульс к релятивизации нерешенных сложностных проблем был дан тем обстоятельством, что большинство теорем (но не все!) рекурсивной теории допускали релятивизацию[1]). Что касается теории сложности, то здесь релятивизация проходит для абстрактной теории сложности в духе М. Блюма (см. [13*, гл. 12 и 109*]), т. е. там, где имеется инвариантность относительно любых рекурсивных преобразований информации, но вообще говоря, не для полиномиальной инвариантности. Проблема перебора (в узком смысле) или $(\mathrm{P} \stackrel{?}{=} \mathrm{NP})$-проблема состоит в получении «достаточно близких» нижней и верхней оценок на машине Тьюринга NP-полного предиката $Q(z)$. В настоящее время верхние оценки (полученные из оцененных по времени предложенных алгоритмов) имеют вид $c\exp(|z|^{\alpha})$ с разными для разных задач показателями $\alpha > 0$ и $c > 0$, и приблизительно равны объему перебора в конкретной задаче, а нижние оценки — полиномиальны от $|z|$. Если $\mathrm{NP} = \mathrm{P}$, то существуют и полиномиальные верхние оценки. Важность NP-проблемы со-

[1]) Раньше предполагалось, что для релятивизируемости достаточна инвариантность теоремы относительно любых рекурсивных преобразований информации [14*, гл. 4]. Однако эта гипотеза оказалась неверной. ([14*, § 13.1, заключительные замечания, стр. 335]). В [112] рассмотрены элементарные теории вычислимости с произвольными оракулами и дана игровая интерпретация теорем теории рекурсивных функций, допускающих любую релятивизацию.

Подводные камни релятивизации обнаруживаются и в теории доказательств: например, в [18*] строятся арифметические формулы φ и ψ с одним свободным переменным x такие, что в HA (интуиционистской арифметике) $\vdash \forall x[\varphi(x) \vee \neg\varphi(x)] \to \forall x[\psi(x) \vee \neg\psi(x)]$, однако предикат $\Psi(x)$ перекурсивен относительно $\varphi(x)$. В то же время из реализуемости выводимых в *HA* формул следует, что при $\vdash \forall x[\tau(x) \vee \neg\tau(x)]$, где τ формула HA с одним свободным переменным x, предикат τ рекурсивен [18а*]. Таким образом, закон исключенного третьего выражает разрешимость, но не относительную разрешимость. Аналогично обстоит дело с фуикцнями $y = f(x)$, определяемыми формулами HA вида $\forall x \exists y R(x, y)$, представляющими интерес для логического программирования [18а*].

стоит еще и в том, что полиномиальная вычислимость инвариантна относительно большинства вычислительных моделей, во всяком случае всех моделей с не более чем полиномиальным ростом окрестности каждой ячейки вычисления[1]). Моделирование одного шага вычисления одного такого устройства на другом не превосходит $c \cdot s^{\alpha}$ шагов, где s — величина используемой в этот момент памяти 1-го устройства, α, $c > 0$, — константы, зависящие от обоих устройств и связанные с перекодировкой информации при моделировании. Отсюда следует, что время вычисления $T_2(x)$ для входа x на втором устройстве не превосходит $p(T_1(x))$, где $T_1(x)$ — время вычисления для x на 1-ом устройстве, а p — некоторый полином.

Конкретная («неблюмовская») теория сложности вычислений, не является рекурсивно инвариантной, а часто не является и полиномиально инвариантной (см. обзоры [20*, 21*]), и в осиовном не поддается релятивизации, как это видно, например, из различных решений проблемы $\mathrm{P}^A \stackrel{?}{=} \mathrm{NP}^A$ для разных оракулов A[2]).

2. КЛАССЫ СЛОЖНОСТИ ВЫЧИСЛЕНИЙ

Наряду с классами P и NP исследовались и другие сложностные классы вычисления предикатов и функций, связанные как с ограничениями некоторой меры сложности (времени, памяти, числа поворотов головки, длины или типа задающей формулы и т. д.), так и способа обращения к памяти (магазину, стеку, очереди и т. д.) [23*]. Каждому детерминированному классу сложности обычно соответствует некоторый недетерминированный класс сложности (изредка они совпадают как в случае конечных автоматов или машин Кука [23*], чаще же Det Class $\subsetneq$ Non Det Class, как для р. п. или контекстно свободиых языков, в ряде случаев вроде $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ вопрос открыт), а в последнее время также альтернирующий [24*, 25*, 26*] и вероятностные классы [27*, 28, 60a] сложности, также связанные с

[1]) Физически реализуемыми сегодня выглядят только такие модели, хотя известны абстрактиые модели с экспоиенциальиым ростом окрестности, например, алгоритмы Колмогорова [19*]. Некоторые модели такого типа, ио с параллельиым действием отдельных блоков, (нелокальиые модели), могут давать и сверхполиномиальиое по сравнению с машиами Тьюринга ускорение вычислений.

[2]) Это ие снижает, иа наш взгляд, цениости исследоваиий полиномиальиых вычислеиий с оракулами. Так, например, вычислимые функции веществеиного перемеиного можио трактовать как вычислимые функции натуральиого аргумента с оракулом, роль которого играет вещественный аргумент исходиой фуикции. При этом естественно выделяется класс полиномиально вычислимых вещественых (и комплексных) функций, играющих главиую роль в анализе. Такой подход к вычислимому анализу и вопросам сложности в нем развивается в [22*] (см. еще [22a*]). Прежде сложностиый аспект в исследованиях по вычислимому (и конструктивному) анализу оставался в тени,

различными комбинаторными задачами [28а*]. В указанных работах много внимания уделяется также соотношениям между разными классами сложности. Для этих классов определяются «полные» или «универсальные» или «труднейшие» проблемы относительно той или иной сводимости [29*]. Примеры характеризаций P, NP и других классов сложности средствами логики можно найти в [30*, 31*, 32*, 32а*], а средствами теории рекурсий — в [37*]. В ряде работ исследуются классы (в том числе и релятивизированные), определяемые одновременными ограничениями разных мер сложности [33*, 34*, 40*].

Вернемся, однако, к временны́м классам сложности. При фиксации типа устройств или задач иногда удается дать более тонкие (чем полиномиально инвариантные) оценки времени вычисления. Так для k ленточных 1-мерных детерминированных машин Тьюринга ($k \geqslant 2$) совсем небольшое увеличение времени работы (большее чем $\log^* t$ — функция обратная к «сверхэкспоненте») дает расширения класса вычислимых предикатов [35*][1]). Для недетерминированных машин Тьюринга мне известен результат С. Кука [36*] о том, что при любом степенном увеличении времени от T к T^α, $\alpha > 1$, класс сложности возрастает (впрочем, см. [39*, 39*а]).

NP-проблема является частным случаем проблемы детерминизации для временны́х, а более общо, любых классов сложности вычисления. Другой знаменитой проблемой такого рода является соотношение классов Det SPACE(n) и NDet SPACE(n), известная в теории формальных языков как LBA-проблема Куроды [38*] (NDet SPACE(n) — класс контекстных языков). Лучшим результатом до сих пор остается [38*а]:

$$\text{Det SPACE}\,(s(n)) \subseteq \text{NDet SPACE}\,(s(n)) \subseteq \text{Det SPACE}\,(s(n))^2 \text{ для } s(n) \geqslant \log n.$$

Проблема отделения недетерминированных классов сложности от детерминированных обсуждается в [41*]. До последнего времени эта проблема оставалась открытой. Лишь недавно в [42*] было доказано, что для машин Тьюринга с k одномерными лентами ($k \geqslant 2$) класс DTIME$(n) \subsetneqq$ NTIME(n). Это делается с помощью включения TIME$(f) \subset \Sigma_4^p(f/\log^* f)$, где $f/\log^* f \geqslant n$ (здесь всюду $f = f(n)$), а $\Sigma_4^p(T(n))$ — класс предикатов, выразимых с полиномиально ограниченной кванторной приставкой $\exists\forall\exists\forall$ перед предикатами из класса DTIME$(T(n))$. Метод доказательства — широко известная игра в камешки на графах [43*]. Хотя это и не даёт еще решения «$\mathrm{P} \overset{?}{=} \mathrm{NP}$», но является значительным

[1]) Для устройств этого типа характерна $T \cdot p(\log T)$-инвариантность (где p — полином) для времени при моделировании, как следует из работ Хенни и Стирнза [37*, с. 194—212].

шагом к $P \neq NP$. Что же означает $P^A = NP^A$ для некоторых рекурсивных оракулов A? Обычно A выбирается PSPACE-полным, т. е. из класса, для которого $NPSPACE = PSPACE$, и тогда оказывается даже $P^A = PSPACE^A$ (см. [71*]).

В доказательствах, где A таков, что $P^A \neq NP^A$, в недетерминированном дереве вычисления NP^A-полного предиката делается «много» недетерминированных шагов и обращений к оракулу.

В [71] доказано, что если суммарное число обращений к оракулу A по всем путям недетерминированных вычислений ограничено полиномом от длины входа, то $NP^A = P^A \Leftrightarrow NP = P$. При некотором способе применения оракулов за полиномиальное время распознается в точности $NP \cap$ co-NP [71a]. Поэтому представляло интерес исследование класса $NP^A (\neq P^A)$ с ограничением как числа недетерминированных шагов, так и числа обращений к оракулу. Такие исследования обнаружили существование внутри NP^A (при подходящих оракулах A) тонких иерархий по этим параметрам [44*, 45*]. В [46*] доказано, что $(\forall k \geqslant 1)$ $(\exists$ рекурсивное $A \subset \{0, 1\}^*)(\exists M \in P_k^A)$ $(\neg \exists M_1 \in P_{k-1}^A)$ $(M_1 \subset M$ и M_1 бесконечно), где $P_i^A = \{L \mid L \subseteq \{0, 1\}^*$ распознается недетерминированной машиной Тьюринга за полиномиальное время с не более чем n^i недетерминированными шагами на входах длины $n\}$.

Некоторые результаты о решетке P-степеней NP^A-множеств, связанные со статьей Хомера — Мааса и полученные утонченными диагональными и приоритетными конструкциями, содержатся в [47*, 48*, 49*]. Интересные усиления диагональных построений применяются к исследованию большинства «разумных» классов (не только временно́й) сложности в [50*].

Рассмотрение релятивизированных сложностных проблем сохранит интерес даже после решения NP-проблемы, поскольку некоторые комбинаторные проблемы логики и дискретной математики являются полными в классах, больших чем NP. Кроме того, если, как ожидается, $NP \neq P$, то исследование P-степеней NP-множеств приобретает особый интерес, так как решение NP-полной задачи с помощью оракулов из NP соответствует разбиению NP-полной задачи на ряд NP-задач с отношением (со)подчинения, а это часто играет организующую роль в решении «большой» исходной NP-задачи и, хотя не дает убыстрения на всех аргументах, но облегчает ход решения для ограниченного множества аргументов, а переборные задачи практически решаются лишь для таких множеств.

Хартманис в [66*] доказывает эквивалентность утверждений: 1) $SAT \cap K[\log n, n^2] \in P$, 2) $NEXPTIME = EXPTIME$ и 3) в $NP \setminus P$ нет «разреженных» (т. е. небольшой плотности в натуральном ряде) множеств. Здесь $K[m(n), t(n)]$ есть класс слов, (обобщенная) колмогоровская сложность которых не превосходит $m(n)$ (n — длина слова), а время получения их из

программы $\leqslant t(n)$. Затем эти результаты обобщаются на классы емкостной сложности. Для оракулов имеем следствия:
(1) $\mathrm{P} = \mathrm{NP} \Leftrightarrow (\forall A \subseteq K[\log n, n^2])\,(\mathrm{P}^A = \mathrm{NP}^A)$.
(2) Для «редких» случайных оракулов A из $K[\log n, n^2)$ с вероятностью 1 верно $\mathrm{P}^A \neq \mathrm{NP}^A$.

Ранее Беннет и Гилл доказали, что с вероятностью 1 $\mathrm{NP}^A \neq \mathrm{P}^A$ для случайного оракула A и формально описали «допустимые» утверждения о классах релятивизированной (с параметром A — оракулом), сложности, (причем $\mathrm{P}^A \neq \mathrm{NP}^A$ «допустимое» утверждение) и сформулировали гипотезу: если «допустимое» утверждение S^A истинно для случайного A с вероятностью 1, то истинно нерелятивизированное утверждение $S^{\varnothing}$. Однако в [51*] эта гипотеза опровергается, и предлагается искать другое определение «допустимых» утверждений.

Укажем ряд «полных» проблем классов емкостной сложности.

1. Проблемы выполнимости в H, $S4$, программных и некоторых других неклассических логиках PSPACE-полны [52*, 24*]. (PSPACE = NPSPACE есть класс предикатов, вычислимых с полиномиальной памятью (зоной, емкостью, лентой) при P-сводимости (т. е. сводимости с полиномиальным временем)).

2. Проблема заполнения графов камешками (pebbling) и проблемы выигрыша в игре Hex (обобщение игры Шеннона) PSPACE-полны [53*, 54*, 55*].

3. Ряд проблем (выполнимость пропозициональных формул с двумя литерами в элементарных дизъюнкциях, разрешимость системы линейных неравенств с двумя слагаемыми в каждом, эквивалентность обобщенных автоматов с заключительными состояниями, разрешимость $LL(k)$ и $LR(k)$-условий для контекстно-свободных грамматик) полны в NLOGSPACE (классе множеств слов, недетерминированно распознаваемых с памятью $\sim \log n$) относительно DLOGSPACE-сводимости [56*, 56a*].

4. Некоторые проблемы выполнимости для разрешимых фрагментов УИП (узкого исчисления предикатов) и некоторые проблемы разрешения комбинаторных игр полны в EXPSPACE [57*].

5. Новая характеризация класса PSPACE в терминах теории оптимизации и принятия решений, а также ряд PSPACE-полных задач из этой области приводятся в [55a].

Условия возрастания сложностных классов емкостной иерархии давно даны в [58*]. Дальнейшие результаты см. в [59*].

3. ДРУГИЕ ПЕРЕБОРНЫЕ ЗАДАЧИ

Наряду с NP рассматривались другие типы проблем переборного характера: (#P) подсчет числа вычислений (путей в графе) длины, ограниченной полиномом от входа (такова проблема вычисления перманента [60*], (PP)-проблемы вероятностных вычислений полиномиальной длины, проблема нахож-

дения $\mu y_{|y| \leqslant p(|x|)} Q(y, x)$, где μy — «наименьший y...» или обращение полиномиально ограниченной полиномиально вычислимой функции $y = f(x)$, проблема униформизации $Q(y, x), Q \in \mathrm{P}$, состоящая в нахождении для каждого x такого y, $|y| \leqslant p(|x|)$ (если он существует), где p — фиксированный полином, что $Q(y, x)$[1]) проблемы вычисления предикатов из P с кванторами, ограниченными полиномами от длины входа. Последние образуют уже упоминавшуюся полиномиальную иерархию Мейера — Стокмейгера, аналогичную арифметической иерархии рекурсивной теории [14*, гл. 14, 15]. Мы дадим ее другое (эквивалентное) определение для произвольного оракула X:

$$\Sigma_0^{p,X} = \Pi_0^{p,X} = \Delta_0^{p,X} = \mathrm{P}(X), \quad \Sigma_{i+1}^{p,X} = \mathrm{NP}(\Sigma_i^{p,X}),$$
$$\Pi_{i+1}^{p,X} = \text{co-}\Sigma_{i+1}^{p,X}, \quad \Delta_{i+1}^{p,X} = \mathrm{P}(\Sigma_i^{p,X}),$$

где X — оракул, $\mathrm{P}(\Omega)$ и $\mathrm{NP}(\Omega)$ обозначают классы предикатов, детерминированно (недетерминированно) вычислимых за полиномиальное время с оракулами из класса Ω, а co-Ω — обозначает класс дополнений к предикатам из класса Ω. Очевидны включения:

$$\Sigma_i^{p,X} \cup \Pi_i^{p,X} \subseteq \Delta_{i+1}^{p,X} \subseteq \Sigma_{i+1}^{p,X} \cap \Pi_{i+1}^{p,X} \quad (i \geqslant 1)$$

Обычные нерелятивизированные классы получаются, если $X = \varnothing$, т. е. $\Sigma_i^p = \Sigma_i^{p,\varnothing}$ и т. д., $\mathrm{NP} = \Sigma_1^p$, co-$\mathrm{NP} = \Pi_1^p$. (В [62*] определяется также экспоненциально-временная иерархия).

Некоторые комбинаторные задачи естественно формулируются как полные относительно P-сводимости в классах полиномиальной иерархии. Так, проблема существования единственного оптимального решения задачи коммивояжера Δ_2^p-полна [63*] (в рекурсивной теории [14*] в классах Δ_i не может быть полных множеств!). Неизвестно, конечно, вырождена ли полиномиальная иерархия. Лишь совсем недавно (1985 г.) и только для подходящих оракулов X был получен результат о том, что эта иерархия (относительно X) не вырождена, а значит, строго содержится в PSPACE^X [69*]. С этой теоремой контрастирует работа [67*], где установлено, что существуют оракулы X и Y, такие, что $\mathrm{NP}^X \subsetneq \Delta_2^{p,X} = \Sigma_2^{p,X}$ и $\Delta_2^{p,Y} \subsetneq \Sigma_2^{p,Y} = \Pi_2^{p,Y}$. Там утвер-

[1]) Интересна ситуация, когда надо для каждого x найти такое y, что выполняется предикат $Q(y, x)$, вычисление которого связано с перебором, скажем $Q \in \Delta_i^p$, $i \geqslant 2$, но, вместе с тем, для всякого x предикат $Q(y, x)$ является истинным «для почти всех» y (асимптотически при $|x| \to \infty$) данной длины, зависящей от x. Тогда вероятностный алгоритм без перебора по x вычисляет (случайно выбирая) y, который с вероятностью, стремящейся к 1 при $|x| \to \infty$, удовлетворяет $Q(y, x)$. Именно такая ситуация рассмотрена в [8*].

ждается (§ 4) также, что для «большинства» рекурсивных оракулов $\Sigma_2^{p,X} \neq \Pi_2^{p,X}$.

Представляется правдоподобным, что для любого тривиально неопровержимого включения классов P-иерархии существует оракул, относительно которого это включение имеет место. Все перечисленные проблемы переборного типа характеризуются тем, что выполняются на полиномиальной зоне, т. е. принадлежат PSPACE, причем вычисление в них естественно разбивается на $\exp p(n)$ однородных блоков (где p — полином) полиномиальной длины, а передаваемая от блока к блоку информация «невелика», т. е. умещается на полиномиальной зоне. (Произвольное вычисление длины $\exp p(n)$ могло бы потребовать памяти порядка $\exp p(n)$. Можно ли соответствующий предикат из класса $\mathrm{DTIME}(\exp p(n))$ вычислить на полиномиальной зоне, неизвестно, и кажется маловероятным. О соотношении времени и памяти будет сказано ниже). Между проблемами переборного типа имеются некоторые очевидные и неочевидные отношения включения и сводимостей [28*, 60*, 60a]. Иные из иных установлены лишь для оракулов [68*], и иногда, как мы видели, контрастируют друг с другом при разных оракулах. Некоторые остаются открытыми даже в релятивизированной версии.

До последнего времени для полных (в своем классе) переборных проблем (с перебором порядка $\exp p(n)$) не было известно ни одной верхней оценки сложности, существенно меньшей, чем перебор, вплоть до работы [65*], в которой в качестве побочного результата исследования о полиномиальных псевдослучайных последовательностях была доказано дизъюнкция:

проблема «порогового подсчета» или проблема обращения функции (неизвестно какая именно!) решается быстрее, чем за время $\exp(\sqrt{n})$, тогда как время перебора здесь $\geqslant \exp(n)$.

При сведении проблемы A к B из простоты проблемы B вытекает «простота» проблемы A, т. е. зависимость A от B прямая. В [65*] зависимость между «сложностями» проблем выглядит сегодня обратной!

Среди переборных проблем выше была упомянута проблема униформизации. В ней требовалось по x найти y, если он существует, такой, что условие $Q(x, y)$ «просто» проверяется. Важно, что если такого y не существует, алгоритм не обязан давать ответ. Такой (если «простоту» Q понимать как рекурсивность) является проблема поиска вывода («y») формулы (слова) x в логических исчислениях, формальных грамматиках и т. д. В [9*] для подобных задач был предложен оптимальный (для алгоритмов Колмогорова даже мультипликативно оптимальный) алгоритм. Неизвестно, правда, сколько времени он требует! Если же для Q из субрекурсивного класса, скажем P,

потребовать, чтобы при отсутствии нужного y алгоритм давал ответ «нет», то удается лишь для некоторых Q доказать существование алгоритма, оптимального в существенно более слабом смысле [69*], и строится такой оракул, что для релятивизации этой задачи нельзя снять ограничения оптимальности из-за эффекта «ускорения» [13*, гл. 9] и [37*, с. 401—431] [1]).

4. СООТНОШЕНИЕ ВРЕМЕНИ И ПАМЯТИ. СХЕМНАЯ СЛОЖНОСТЬ И СВОДИМОСТЬ

В работах [71*, 72*, 39а*] исследуются качественные и количественные ограничения на машины с оракулами, которые обеспечивают совпадение исходной и релятивизированной версий для отношений включения разных классов сложности (в том числе временной и емкостной).

Вопрос о соотношении времени и памяти вычисления играет огромную роль в теории сложности. Для достаточно общих вычислительных моделей, например, многоленточных одномерных машин Тьюринга, лучшим сегодня является результат Хопкрофта, Пауля и Вальянта [73*], полученный методом игры в камешки: всякое вычисление с временем $T(n)$ может быть промоделировано с памятью $T(n)/\log T(n)$ (но с большим временем. Затем в [74*] несколько ослабленные утверждения такого рода были доказаны для машин с произвольным доступом к памяти — с логарифмической стоимостью операций (от длины ячейки) и машин Тьюринга с многомерными лентами. Результат Хопкрофта с соавторами (HPV-теорема) был затем получен совершенно другим методом — методом перекрытий слов [75] [2]), который наряду с игрой в камешки и методом перевода (см. [37, стр. 213—222] и [76]) является одним из главных инструмен-

[1]) В свое время теоремы об ускорении поразили специалистов как своей общностью, так и содержанием: оказалось, что для произвольной меры сложности вычисления существуют функции и предикаты, сложность вычисления которых невозможно удовлетворительно оценить снизу [13, гл. 12], [37, с. 401—431] и [109]. Зато при тех же слабых аксиомах сложности вычислений Блюма была доказана теорема «наименования»: объединение вычислимой последовательности классов сложности само является классом сложности, т. е. классом функций, определяемых верхней оценкой сложности вычисления «почти всюду». (Отсюда вытекало довольно странное утверждение: класс примитивно рекурсивных функций является классом сложности для времени, памяти и многих других мер сложности). Двойственный к этой теореме результат [70*, теорема 2] о существовании общерекурсивной нижней оценки, однозначно определяющей класс всех сигнализирующих вычислений (= функций сложности вычисления для выбранной меры сложности) данной общерекурсивной функции, и характеризация классов сигнализирующих [70*, теорема 3], восстановили гармонию в теории сложности вычислений, нарушенную обескураживающими поначалу теоремами Блюма.

[2]) Это говорит в пользу того, что результат Хопкрофта с соавторами близок к оптимальному.

тов исследования в теории сложности. Было бы интересно проанализировать доказательства [73] и [75] на предмет установления связи первыми двумя методами.

Приведенные результаты сразу породили вопрос о компромиссе (trade-off) между временем и памятью, который часто возникает и в решении конкретных сложностных задач (см. [77], где приведен список литературы по этому вопросу). Ряд соотношений «время—память» удается получить сегодня только для вычислений с подходящими рекурсивными оракулами [73] В. Савич [78] решил проблему Лэднера — Линч, доказав существование такого рекурсивного оракула D, что

$$\mathscr{L}^D = P^D \not\subseteq N\mathscr{L}^D = NP^D,$$

где $\mathscr{L}(N\mathscr{L})$ классы предикатов, вычислимые детерминированно (недетерминированно) с логарифмической памятью. В [80] построены оракулы для машин Тьюринга с вспомогательным магазином (машин Кука) и установлены для них соотношения между классами сложности вычислений, которые контрастируют с результатом Кука для машин без оракулов: $\mathrm{DTIME}(2^{c \cdot s(n)}) = \mathrm{NAUXSPACE}(s(n)) = \mathrm{DAUX\ SPACE}(s(n))$, где $\mathrm{AUXSPACE}(s(n))$ — класс предикатов, вычислимых на машинах Кука с основной памятью $s(n)$ [23].

Упомянем еще об одном направлении, устанавливающем связь между схемной сложностью последовательностей булевых функций и сложностью вычисления предикатов, задающих эти последовательности [1]). При моделировании тьюринговых вычис-

[1]) Возможно, первые работы по сложности вычислений и алгоритмов относятся как раз к исследованию схемной сложности булевых функций [110, 111], начатому К. Шэнноном еще в 40-х годах. Схемы и формулы можно рассматривать как программы алгоритмов вычисления функций в конечной области, причем необходимые элементарные операции и в какой-то мере их последовательность непосредственно обозначены в программе. Отсюда вытекает прямая зависимость между длиной записи таких алгоритмов и длиной и памятью их вычислений. Для алгоритмов более широких классов часто имеет место как раз обратная зависимость: ускорение вычисления достигается усложнением алгоритма в смысле увеличения длины его записи, в смысле расширения выразительности языка, на котором он сформулирован (см. результаты об ускорении и объеме машин [37*, предисловие и с. 423—431]), а также [82]). В более широком языке строятся последовательности, сложные или «случайные» относительно класса алгоритмов более узкого языка, т. е. более узкого класса вычислимых функций [83, 83a]. С другой стороны, в более широком языке часто удается дать более компактную запись алгоритмов. (Конечно, желательно, чтобы алгоритмический язык сочетал большую выразительность с простотой и естественностью семантики). Теория сложности алгоритмов при ограниченных ресурсах вычисления, в частности, теория оптимальных алгоритмов с ограниченными ресурсами значительно менее развита, чем обычная теория сложности алгоритмов [83b].

Возникает проблема компромисса (trade-off) «сложность алгоритма — сложность вычисления» (См. [84*, 83b*]).

лений конечных функций схемам времени вычисления соответствует глубина схемы, а памяти — ширина схемы. Это соответствие сохраняется и при моделировании схем тьюринговыми вычислениями с оракулами зависящими от числа n булевых переменных [85, 85a]. Если для последовательности схем существует просто вычислимый алгоритм построения булевой схемы для каждого n, то говорят о равномерной схемной сложности. Такие последовательности схем реализуются в устройствах параллельного действия, для которых хорошей моделью являются альтернирующие машины Тьюринга (AMT). При этом объем схемы соответствует AltSPACE, а глубина — AltTIME [88*]. Схемы полиномиального от n объема и глубины poly log n характеризуются AMT с ограниченными деревьями альтернации.

Рассматриваются и другие способы моделирования схем и машинных вычислений с другими соответствиями сложностных оценок [86*, 87*, 89a, 113].

Особый интерес в этом контексте привлекают схемы ограниченной глубины, но с элементами «$\vee$» и «$\wedge$» с произвольным числом входов. В ряде работ получены нетривиальные (сверхполиномиальные) нижние оценки сложности таких схем, реализующих известные булевы функции (четности, умножения и транзитивного замыкания). Для получения этих результатов в [89, 89a] было введено понятие сводимости схем («схемная релятивизация»), в частности сводимости полиномиального объема и постоянной глубины. Было доказано также, что из нижней оценки схем для функции чётности порядка $\Omega(n^{\text{poly log } n})$ следует невырожденность полиномиальной иерархии для некоторого оракула A. В вышеупомянутой работе [64*] получена именно такая оценка. Об оценках схем постоянной глубины для других задач и связи упомянутой сводимости с рекурсивными сводимостями см. [90*, 91*]. В статье [92*] рассматриваются неравномерные условия на равномерные классы схем, и строится новая иерархия Σ_i/poly, Π_i/poly, Δ_i/poly, связанная с полиномиальной иерархией $\{\Sigma_i^p, \Pi_i^p\}$ импликацией:

$$\Sigma_i/\text{poly} = \Pi_i/\text{poly} \Rightarrow \Sigma_{i+2}^p = \Pi_{i+2}^p \quad (i = 1, 2, 3, \ldots)$$

Ряд результатов изложен в обзоре [20, гл. 3, §§ 1, 2].

Вальянт определил понятие алгебраической сложности вычисления и сводимости предикатов, аналогичное схемной сложности и сводимости [92a*].

Практическое применение оракулов для ускорения и упрощения вычислений состоит в обращении к таблицам и подпрограммам, другим процессорам, словом в использовании ранее построенных алгоритмов и результатов уже проведенных вычислений и измерений. Сложность сведения = сложности релятив-

визированного вычисления (СРВ) $\leqslant$ сложности исходного вычисления (СВ). Часто СРВ $\ll$ СВ. Однако иногда СРВ = СВ. Не всегда алгоритм решения даже «родственной» задачи может сколько-нибудь помочь (послужить эффективным оракулом) в решении другой задачи. Так в [81] доказано, что умножение матриц не ускоряется с помощью (независимо от) вычисления транзитивиого замыкания графа, ранга матрицы, множества линейно независимых строк и столбцов матрицы и т. д.

5. ОБОБЩЕННАЯ ВЫЧИСЛИМОСТЬ

Нельзя не сказать еще об одной области, где применяются вычисления с оракулами — вычислимости на множествах объектов произвольной природы. Эта область привлекала и продолжает привлекать к себе внимание исследователей [93*, 98*, 101*].

Представляется неслучайным то обстоятельство, что определения вычислимости («абсолютной») и относительной вычислимости (т. е. с оракулами) были одновременно даны А. Тьюрингом с помощью идеализированных вычислительных устройств. В самом деле, любое определение универсальных алгоритмов, по-видимому, должно предполагать некоторую структуру их программ, выделение взаимодействующих блоков. Но при этом одни блоки по отношению к другим играют роль оракулов. Кроме того, реализация любых алгоритмов или процедур, встречавшихся в человеческой практике, состоит в выполнении определенных «элементарных» операций, и предполагается, что их можно эффективно провести. Так в школьных геометрических построениях считаются эффективно выполнимыми простые построения циркулем и линейкой, с помощью которых осуществляются более сложные построения. В арифметике считаются элементарными операции прибавления единицы и рекурсивные построения и т. д. При этом кроме осуществимости прибавления единицы принимается еще гипотеза потенциальной осуществимости сколь угодно длинных, но конечных построений и объектов. Таким образом, всякие алгоритмы оказываются релятивизированными к некоторым элементарным операциям над объектами из своей области, и к гипотезе потенциальной бесконечности. Утрируя, некоторые математики склонны считать все алгоритмы (даже в арифметике) относительными. Однако, математическая практика все-таки заставляет выделять «абсолютные» алгоритмы в областях конструктивных объектов, считая, скажем, операцию прибавления единицы «абсолютно» осуществимой. (Я не хочу касаться здесь парадоксов «кучи» и теорий достижимости в натуральном ряде [94, 95]).

Можно двумя способами определять вычислимость на множествах объектов произвольной природы. Один из них, основанный

на нумерациях, даже в принципе применим лишь для счетных множеств. Он является более распространенным [13, гл. 3, § 6], [19, ч. I, §§ 14, 15 и ч. II, §§ 4, 5], но имеет ряд недостатков. Будем различать множества объектов, имеющих внутреннюю структуру (которая может быть связана со структурой всего множества и даже определять ее, т. е. функции и предикаты на этом множестве), скажем слова, графы и т. д., и множества бесструктурных объектов, но имеющие структуру, т. е. определенные на них функции и отношения. Эти множества со своей структурой обычно называются алгебраическими системами. Нумерационный подход в принципе всегда годится для множеств конструктивных объектов первого типа, хотя и здесь удобнее и естественнее оказываются такие вычислительные модели (алгоритмов), которые ориентированы на их структуру. Так алгоритмы в форме Тьюринга, Поста или Маркова приспособлены к словарной природе объектов, алгоритмы Колмогорова работают с комплексами или графами, ЛИСП — со списками и массивами и т. д. Наиболее интересными (выразительными) представляются такие алгоритмические языки, на которые проще транслировать имеющиеся алгоритмические системы и в которых сами программы и их вычисления выглядят наиболее просто и «естественно». Пальма первенства здесь сегодня принадлежит алгоритмическим языкам аппликативного типа (моделям λ-исчисления), построенным в работах Д. Скотта и Ю. Л. Ершова [196*, 98*, 97*]. Конструктивными объектами тут являются функции, определенные на конечных множествах ранее построенных объектов, построение ведется индуктивно вместе с отношением частичного порядка по определенности. В [97*] внутренними средствами определяются понятия перечислимости и вычислимости на множествах таких объектов, и демонстрируется большая выразительность этого языка, связанная с простотой трансляций на него и относительной краткостью программ и вычислений. Этот язык удобен еще и тем, что допускает предельный переход к бесконечным объектам с сохранением структуры, аналогичный построению вещественного континуума. Идея оракулов пронизывает языки такого типа, поскольку значение каждой функции по определению вычисляется путем обращения к оракулам, роль которых играют функции-аргументы. Имеются и типовые (иерархические) варианты таких моделей (см. [98*, приложение А.3.].

Вернемся к нумерационному подходу. Естественное требование к нему состоит в согласованности, т. е. частично рекурсивная функция, на номерах объектов, задающая алгоритм, должна быть одновременно определена на любых двух разных номерах одного и того же объекта и давать (быть может, разные) номера одного и того же объекта — результата работы

алгоритма. Кроме того, при наличии структуры на множестве объектов крайне желательно, чтобы ее базисные функции и отношения выражались частично рекурсивными функциями и предикатами на множестве номеров. Последнему требованию далеко не всегда удается удовлетворить. Так любая нестандартная модель арифметики неконструктивизируема [19, с. 279—280]. Однако, это не мешает определить «внутренним» образом вычислимые функции в каждой такой модели, считая сложение и умножение в ней элементарно вычислимым, хотя бы с помощью гёделевского метода, и не умаляет значение таких моделей.

Это обстоятельство подсказывает другой более универсальный способ определения вычислимости в произвольных областях, основанный на использовании структур, имеющихся на них. В [99*] дается изложение этого способа и хороший обзор работ по этой теме. (См. также [100*, 105*]). При этом подходе не обязательно ограничиваться счетными областями. Так С. Клини определил иерархию вычислимых функционалов на объектах всех конечных типов — т. е. функций, функций от функций и т. д., естественно интерпретируемых с помощью машин с оракулами [14*, § 16.5], впрочем так же хорошо описываемых системами рекурсивных уравнений. (Это другой вариант относительной вычислимости — относительная рекурсивность).

Интересны случаи, когда оба подхода дают одинаковый результат (из примера с нестандартной арифметикой видно, что это имеет место не всегда), т. е. приводят к одному и тому же классу вычислимых функций в рассматриваемой области. Именно так обстоит дело с алгоритмическими («нумерационными») операторами на частично рекурсивных и, как показал Г. С. Цейтин [101*], даже на общерекурсивных функциях (и на конструктивных вещественных числах). В областях своего определения эти операторы совпадают с частично рекурсивными операторами, т. е. являются непрерывными в бэровской топологии (соответственно, в конструктивном континууме) [19, § 14, с. 207]. Первый результат обобщается на частично рекурсивные функционалы высших типов [102].

Замечательными свойствами обладают общерекурсивные (т. е. определенные на всех функциях типа: $N \to N$) функционалы и операторы (типа 2). Ю. Д. Стригин построил иерархии таких функционалов и операторов по конструктивной системе O обозначений для ординалов [14*, §§ 11.7, 11.8 и упражнения 11.46—63], связанные с числом и объемом обращений к оракулу [103*]. В теории множеств и теории рекурсивных функций давно известны иерархии по ординалам, например, борелевская и гиперарифметическая (по обозначениям ординалов) [14*, гл. 15, 16, особенно § 16.8]. Они основаны на сложности построения множеств — числе операций предельного перехода.

Гиперарифметическая иерархия обладает свойствами полноты, невырожденности и однозначности (разным обозначением одного ординала α соответствует один и тот же класс). Делались попытки классифицировать все общерекурсивные функции по ординалам в соответствии с их «сложностью» (в силу диагональных соображений натуральных чисел не хватает). Однако эти иерархии оказывались либо неполными, либо вырожденными (часто на уровне ω^2) (см. [37, Добавление редактора] и [70*, Раздел I]). Иерархии Стригина занимают промежуточное положение: они полны, невырождены, но неоднозначны (и тоже на уровне ω_2). Насчет общерекурсивных функционалов типа выше второго мало что известно. В теории рекурсий высших типов рассматриваются оракулы, дающие ответы, связанные с бесконечным перебором, например ${}^2E$, соответствующий квантору $\exists x$ по N. На этих идеях в [104] строится обобщенно-конструктивный анализ, более гибкий и приспособленный к потребностям практической математики, чем обычный конструктивный анализ, и одновременно сохраняющий подход теории вычислимости.

Построены теории вычислимости на ординалах, и даже с трансфинитными оценками сложности вычислений ([100*, 102*, 105*])[1]. В теории вычислимых функций на ординалах к элементарным вычислимым операциям арифметики добавляется операция предельного перехода. Возможны и трактовки на основе машин Тьюринга с трансфинитными лентами и временем.

Значение обобщенных теорий вычислимости помимо того, что они описывают практические ситуации с операциями над неконструктивными объектами, вроде аналоговых устройств с программным управлением состоит на наш взгляд в том, что они позволяют по-новому взглянуть на обычную теорию алгоритмов и теоретическое программирование, и часто подсказывают интересные идеи (скажем, «смешанные» вычисления в программировании, см. также [106*]) и пути решения старых проблем (как нестандартные модели арифметики и анализа помогают в решении проблем «стандартной» математики).

В данном добавлении рассматривались в основном работы, опубликованные в последние годы. Более ранние результаты см. в сборниках [37*, 70*, 21*, 107*], обзорах [20*, 21*, 108*, 109*].

Надеюсь, что этот краткий и весьма неполный обзор с библиографией все-таки поможет читателю ориентироваться в увлекательной области, переживающей сейчас бурный период, полный загадок, неожиданных открытий и остроумных решений.

[1] В связи с вычислимостью на ординалах остается открытым вопрос, поставленный автором в 1979 г.: Описать класс функций на ординалах $< \omega_1$ или хотя бы $< \omega^\omega$, определимых с помощью «согласованных» частично рекурсивных функций на O, соответственно фрагмента O до ω^ω. Такие «вычислимые» функции как «сложение» $+_O$ и «умножение» $\times_O$ входят в этот класс.

ЛИТЕРАТУРА

1. Baker T., Gill J. and Solovay R. Relativizations of the $P \stackrel{?}{=} NP$ question. SIAM J. Comput. 4 (4) (1975) 431—442.
2. Baker T. and Selman A. A second step toward the polynomial hierarchy. Proc. 17th IEEE Symp. on the Foundations of Computer Science (1976) 71—75. См. также Theoret. Comput. Science 8, № 1 (1979) 177—187.
3. Bennet C. H. and Gill J. Relative to random oracle $A.P^A \neq NP^A \neq co\text{-}NP^A$ with probability one. SIAM J. Comput. 10 (1) (1981) 96—113.
4. Breidbard S. On splitting sets. J. CSS (1978) 56—64.
5. Hopcroft J. and Ulman A. Introduction to Automata Theory. Languages and Computations (Addison — Wesley, Reading, MA, 1979).
6. Ladner R. On the structure of polynomial tive reducibility. J. Assoc. Comput. Mach. 22 (1975) 155—171.
7. Soare R. I. Recursively enumerable sets degrees. Bull. Amer. Math. Soc. 84 (1978) 1149—1181.
8. Яблонский С. В. — В кн. Проблемы кибернетики. Вып. 2. 1959, Физматгиз, М., 75—121.
9. Левин Л. А. Проблемы передачи информации. 1973, т. 9, № 3, 115—116.
10. Cook S. A. Proc. 3d Ann. ACM STOC, 1971, 151—159 (Имеется перевод: Кук С. А. — В кн. Киберн. сб., вып. 12, 1975, Мир, М., 5—15).
11. Karp R. M. — In: Complexity of computer computations, 1972, 85—104 (Имеется перевод: Карп Р. М. — В кн.: Киберн. сб., вып. 12, 1975, Мир, М., 16—38).
12. Garey M. R., Johnson D. S. Computers and Intractability, 1979 (Имеется перевод: Гэри М. Р., Джонсон Д. С. Вычислительные машины и трудноразрешимые задачи. 1982, Мир, М.).

12a. Ахо А., Хопкрофт Д., Ульман Д. Построение и анализ вычислительных алгоритмов. Пер. с англ. — М.: Мир, 1979.

13. Cutland N. Computability. An introduction to recursive function theory. 1980 (Имеется перевод. Катленд Н. Вычислимость. Введение в теорию рекурсивных функций. 1983, Мир, М., гл. 4 и 9).
14. Rogers H. Theory of recursive functions and effective computability. 1967, N. Y. (Имеется перевод: Роджерс Х. Теория рекурсивных функций и эффективная вычислимость. 1972, Мир, М., гл. 5—10).
15. Hartmanis J. Inform. Proc. Letters. 1985, 20, № 5, 241—248.

15a. Joseph D. J. Comp. Syst. Sci., 1983, 26, № 3, 311—338.

15b. Sazonov V. Y. Lect. Not. Comput. Sci., 1980, № 88, 562—575.

15c. Sazonov V. Y. EIK, 1980, 16, № 7, 319—323.

16. Daley R. J. Symb. Logic, 1981, 46, № 3, 460—474.
17. Шэнь А. Х. Доклады АН СССР, 1979, 248, № 6, 1309—1313.
18. Goodman N. D. J. Symb. Logic, 1978, 43, 497—501.

18a. Непейвода Н. Н. Программирование, 1979, № 1, 15—22.

19. Успенский В. А., Семенов А. Л. Теория алгоритмов. — В кн.: Алгоритмы в современной математике и ее приложениях, ч. I. 1982, Новосибирск, 99—342.
20. Слисенко А. О. Сложностные задачи теории вычислений. Усп. мат. наук, 1981, т. 36, в. 6, 21—103.

20a. Адельсон-Вельский Г. М., Слисенко А. О. — В кн.: Алгоритмы в современной математике и ее приложениях, ч. II. 1982, Новосибирск, 93—123.

21. Бельтюков А. П. — В кн.: Теория сложности вычислений. I. Зап. науч. семин. ЛОМИ, т. 118, 1982, Наука, Л., 4—24.
22. Ko K. I. Theor. Comp. Sci., 1984, 31, № 1—2, 101—124.

22a. Hausman, Korte B. Discrete Math., 1978, 24, 261—276.

23. Cook S. A. J. Ass. Comp. Mach., 1971, 18, № 1, 4—18 (Имеется перевод: Кук С. — В кн.: Сложность вычислений и алгоритмов, 1974, Мир, М., 266—288).
24. Мучник А. А. — В кн.: Киберн. сб. вып. 20, 1983, 141—158.

25. Ladner R. E., Lipton R. J., Stockmeyer L. J. SIAM J. Comp. 1984, 13, N 1, 135—155.
26. Ladner R. E., Stockmeyer L. J., Lipton R. J. Inform. Contr., 1984, 62, N 1, 93—108.
27. Фрейвалд Р. В. Известия ВУЗ. Сер. математика, 1981, № 5 (288), 26—34.
27a. Freivalds R. In: Information Processing 83 (Proc. IFIP Congress 83), Amsterdam, 1983, 157—162.
28. Ruzzo W. L., Simon J., Tompa M. J. Comp. Syst. Sci., 1984, 28, N 2, 216—230.
28a. Manber U., Tompa M. J. Ass. Comp. Mach., 1985, 32 № 3, 720—732.
29. Дехтярь М. И. Сводимость с ограниченной сложностью. — В кн.: Актуальные проблемы математической логики и теории множеств. 1975, МГПИ нм. Ленина, М., 88—104.
30. Volger H. Société Mathem. de France. 2° sér., mèm., N° 16, 1984, 41—51.
31. Ливчак А. Б. Программирование 1985, № 1, 59—65.
32. Grandiean Et. Math. Syst. Theory, 1985, 18, № 2, 171—180.
32a. Fagin R. — In: Complexity of Computation. SIAM — AMS, 1974, 43—74.
33. Book R., Wilson Ch. Xu M. Relativizing time and space. «22nd Annu. Symp. FOCS. Oct. 1981», N. Y., 1981, 254—259.
34. Meinhardt D. Mathem. Research, 1984, 20, 338—344.
35. Paul W. J. Comp. Syst. Sci. 1979, 19, 197—202 (имеется перевод: Пауль В. — В кн.: Киберн. сб. вып. 18, 1981, 46—54).
36. Cook S. A. J. Comp. Syst. Sci., 1973, 7, N 4, 343—353.
37. Сб. «Проблемы математической логики» 1970, Мир, М., 9—49, 123—138.
38. Hartmanis J., Hunt H. B. III. — In: Complexity of Computation. SIAM — AMS, 1974, 1—26.
38a. Savitch W. J. Comp. Syst. Sci., 1970, 4, 177—192.
39. Seiferas J. J., Fischer M. J., Meyer A. R. J. Ass. Comp. Mach., 1978, 25, N 1, 146—167.
39a. Rackoff Ch. W, Seiferas J. SIAM J. on Comp., 1981, 10, № 4, 742—745.
40. Pippenger N. Proc. 20th Annu. Symp. FOCS, Oct. 1979, 307—311.
41. Kannan R. Math Syst. Theory, 1984, 17, N 1, 29—45.
42. Paul W., Pippenger N., Szemeredy E., Trotter W. «24th Annu. Symp. FOCS, Nov. 1983», 1983, 429—438.
43. Paul W. J., Reischuk R. On alternation II. Acta Inform., 1980, 14, 391—403 (имеется перевод: Пауль В., Райшук Р. Об альтернировании II. — В кн.: Киберн. сб. вып. 20, 1983, 123—140).
44. Schöning U., Book R. SIAM J. Comp., 1984, 13, N 2, 329—337.
45. Xu M., Doner J., Book R. J Ass. Comp. Mach., 1983, 30, N 3, 677—685.
46. Li Xiang. Intern. J. Comp. Mach., 1985, 17, 151—153.
47. Ambos-Spies K. Lect. Not. Comp. Sci., 1984, 166, 198—208.
48. Ambos-Spies K. Lect. Not. Comp. Sci., 1984, 171; 1—23.
49. Schöning U. Theor. Comp. Sci., 1984, 31, N 1, 41—48.
50. Schmidt D. Lect. Not. Comp. Sci., 1984, 171, 77—87.
51. Stuart A. Inform. Contr., 1983, 57, N 1, 40—47.
52. Statman R. Theor. Comp. Sci., 1979, 9, N 1, 67—72.
53. Gilbert J. R. et al. SIAM J. Comp., 1980, 9, N 3, 513—524.
54. Paul W., Tarjan R. Acta Inform., 1978, 10, 111—115 (имеется перевод: Пауль В., Тарджен Р. — В кн.: Киберн. сб. вып. 21, 1984, 133—138).
55. Even S., Tarjan R. J. Ass. Comp. Mach., 1976, 23, N 4, 710—719.
55a. Papadimitriou Ch. H. Games against nature. J. Comp. Syst. Sci., 1985, 31, 288—301.
56. Jones N. D. et al Math. Syst. Theory. 1976, 10, № 1, 1—17.
56a. Kasai T., Iwata S. Math. Syst. Theory, 1985, 18, 153—170.
57. Robson J. M. Lect. Not. Comp. Sci., 1984, 176, 498—506.
58. Стирнз Р., Хартманис Дж., Льюис П. М. — В сб.: Проблемы математической логики, 1970, Мир, М., 301—338.

59. Seiferas J. J. Comp. Syst. Sci., 1977, 14, № 1, 73—129.
60. РЖ математика, 1983, 4В 1358.
60a. Hinman P. G., Zachos St. Probabilistic machines, oracles, and quantifiers. In Lect. Notes in Math., 1985, 1141, 159—192.
61. Reif J. H. J. Ass. Comp. Mach., 1984, 31, № 2, 401—421.
62. Heller H. SIAM J. Comp., 1984, 13, № 4, 717—725.
63. Papadimitriou Ch. H. J. Ass. Comp. Mach., 1984, 31, № 2, 392—400.
63a. Stockmeyer L. J. Theor. Comp. Sci., 1977, 3 № 1, 1—22.
64. Yao A. C.—C. Separating the polynomial—time hierarchy by oracles: Part 1, 1985, Comp. Sci. Dept. Stanford Univ. (preprint).
65. Blum M., Micali S. SIAM J. Comp., 1984, 13, № 4, 850—864.
66. Hartmanis J. Generalized kolmogorov complexity and the structure of feasible computations. Bull. Europ. Ass. for Theor. Comp. Sci., 1984, № 24, 73—78.
67. Heller H. Math. Syst. Theory, 1984, 17, № 1, 71—84.
68. Rackoff Ch. J. Ass. Comp. Mach., 1982, 29, № 1, 261—268.
69. Айзенштейн М. Х. О существовании оптимальных алгоритмов для комбинаторных задач и языков. УМН, 1980, 35, № 5, 221—222 и Изв. ВУЗов (Математика), 1985, № 5, 7—15.
70. Сб. Сложность вычислений и алгоритмов. 1974, Мир. М., 174—185.
71. Book R., Long T., Selman A. SIAM J. Comp., 1984, 13, № 3, 461—487.
71a. Schöning U. Theor. Comp. Sci., 1985, 40, № 1, 57—66.
72. Book R., Long T., Selman A. J. Comp. Syst. Sci., 1985, 30, 395—413.
73. Hopcroft J., Paul W., Valiant L. J. Ass. Comp. Mach., 1977, 24, № 2, 332—337.
74. Paul W., Reischuk R. J. Comp. Syst. Sci., 1981, 22, 312—327.
74a. Dymond P. W., Tompa M. J. Comp. Syst. Sci., 1985, 30, 149—161.
75. Adleman L., Loui M. Math. Syst. Theory, 1981, 14, № 3, 215—222. Имеется русский перевод: Эдлимэн Л., Луи М. В кн.: Кибернетический сб., вып. 22, 1985, 102—111.
76. Monien B. Acta Inform. 1976, 6 № 1, 95—108.
77. Ja'Ja'Joseph. J. Ass. Comp. Mach., 1983, 30, № 3, 657—667.
78. Savitch W. J. Math. Syst. Theory, 1983, 16, № 4, 229—236.
79. Book R. V., Wilson Ch., Xu M. SIAM J. Comp., 1982, 11. 571—581.
80. Angluin D. Math. Syst. Theory, 1980, 13, 283—299.
81. Ibarra O. H., Moran Sh. Intern. J. Comput. Math., 1985, v. 17, 113—122.
82. Meyer A. R. Inform. Control, 1972, 21, № 4, 382—394.
83. Агафонов В. Н. Сложность алгоритмов и вычислений. Ч. II. Новосибирск, 1975.
83a. Daley R. P. Inform. Contr., 1973, 23, № 4, 301—312.
83b. Daley R. P. J. Ass. Comp. Mach., 1973, 20, № 4, 687—695.
84. Бургин М. С. Докл. АН СССР, 1982, 264, № 1, 19—23.
85. Schnorr C. P. Acta Inform., 1976, 7, N 1, 95—107.
85a. Ko K. I., Schoning. SIAM J. Comput., 1985, 14, № 1, 41—51.
86. Stockmeyer L., Vishkin U. SIAM J. Comp., 1984, 13, № 2, 409—422.
87. Fischer M., Pippenger N. J. Ass. Comp. Mach., 1979, 26 № 2, 361—381.
87a. Skyum S., Valiant L. J. Ass. Comp. Mach., 1985, 32, № 2, 484—501.
88. Ruzzo W. L. J. Comp. Syst. Sci., 1981, 22, 365—383.
89. Furst M., Saxe J., Sipser M. Math. Syst. Theory, 1984, 17, № 1, 13—27.
89a. Wilson Chr. B. J. Comp. Syst. Sci. 1985, 31, № 2, 169—181.
90. Chandra A. K., Stockmeyer L., Vishkin U. SIAM J. Comp., 1984, 13, № 2, 423—439.
91. Gurevich Y., Lewis H. R. Inform. Contr., 1984, 61, № 1, 65—74.
92. Yap Ch. K. Theor. Comp. Sci., 1983, 26, 287—300.
92a. Valiant L. Proc. 11th Annu. Conf.—ACM FOCS, 1979, 259—271.

93. Клини С. К. Алгоритмы в различных смыслах. — В кн.: Алгоритмы в современной математике и ее приложениях. Ч. II. 1982, Новосибирск. 139—148.
94. Рашевский П. К. Усп. мат. наук, 1973, т. 28, № 4, 243—246.
95. Гавриленко Ю. В Докл. АН СССР, 1984, 276. № 1, 18—22.
96. Scott D. — In: Proc. 4th Annu. Princeton Conf. on Inform. Sci. a. Syst., 1970, 169—176 (Имеется перевод: Скотт Д. — В кн.: Кибери. сб., вып. 14, 1977, 105—121).
97. Сазонов В. Ю. Сибир. мат. ж. 1976, XVII, № 3, 648—672.
98. Барендрегт Х. Ламбда-исчисление. 1985, Мир, М., гл. 5, 18, 19.
99. Ершов А. П. — В кн.: Алгоритмы в современной математике и ее приложениях. Ч. II. 1982, Новосибирск, 194—229.
100. Fenstad J. E. General recursion theory: an axiomatic approach. 1980. Springer — Verlag.
101. Цейтин Г. С. — Труды МИАН им. Стеклова, 1962, т. 67, 295—361.
102. Ершов Ю. Л. Алгебра и логика. 1972, т. II, 367—437.
103. Стригин Ю. Д. Докл. АН СССР, 1973, 210, № 3, 537—540.
104. Ганов В. А. Сибир. мат. ж. 1973, 14, № 5, 1235—1259.
105. Справочная книга по математической логике. Т. I, гл. 7, 235—288 и т. 3, гл. 5, 134—165, 1982, Наука, М.
106. Ершов Ю. Л. Докл. АН СССР, 1983, 273, № 5, 1045—1048.
107. Сб.: Языки и автоматы. 1975, Мир, М.
108. Марченков С. С., Матросов В. Л. — В кн.: Теория вероятн., матем. стат., теор. кибернетика. Т. 16 (Итоги науки и техники), 1978, 103—149.
109. Hartmanis J., Hopcroft J. J. Assoc. Computer Mach. 1971, 18, № 3, 444—475.
110. Лупанов О. Б. Асимптотические оценки сложности управляющих систем. Изд. МГУ, 1984.
111. Храпченко В. М. Нижние оценки сложности схем из функциональных элементов. — В кн.: Кибернетический сб., вып. 21, 1984, 3—54.
112. Мучник Ан. А. Докл. АН СССР, 1985, 285, № 2, 280—283.
113. Johnson D. S. Journ. Algorithms, 1986, June, v. 7.
114. Hartmanis J. Búil. Europ. Assoc. Theor. Comp. Sci., 1985, 27, 40—49.

Добавлено при корректуре. В журнале Journal of Algorithms ежеквартально помещаются обзоры Д. Джонсона: The NP-completeness column: An ongoing guide, посвященные последним достижениям в данной области. Последний обзор см. [113].

Группы ранга 3 и сильно регулярные графы[1]

М. Д. Хестенс, Д. Г. Хигман

Эта статья содержит первые разделы общей теории групп подстановок ранга 3, имеющих четный порядок, которая была заложена в [4]. Изложение основано на рассмотрении сильно регулярных графов, связанных с такими группами. Там, где это возможно, мы изучаем сильно регулярные графы без каких-либо предположений об их группах автоморфизмов. Основной причиной такого подхода является то, что возникновение этих графов связано с группами большего ранга. Цель статьи состоит в том, чтобы (а) дать удобное и замкнутое в себе изложение основных фактов, используемых в теоретических и вычислительных работах по группам ранга 3, (б) изложить в § 5 результаты, развивающие предложенную Симсом в [14] идею изучения четырех вершинных подграфов и (в) привести иное доказательство (см. § 6) классификации сильно регулярных графов с минимальным собственным значением, равным —2, которая опубликована Дж. А. Сайделем в [12].

1. ГРАФЫ И ГРУППЫ ПОДСТАНОВОК

С отношением $f \subseteq X \times X$ на множестве $X \neq 0$ связан граф $\mathcal{G}_f = (X, f)$, имеющий X в качестве множества вершин и f в качестве множества ребер. Под *графом* мы всегда будем понимать граф такого типа. Мы будем называть граф *ориентированным* либо *обычным* в зависимости от того, является ли f строго антирефлексивным и строго антисимметричным, либо строго антирефлексивным и симметричным. В случае обычного графа мы отождествляем (x, y) и (y, x). Под *подграфом* мы понимаем некоторое подмножество вершин вместе со всеми ребрами, соединяющими вершины из этого подмножества. *t-Вершинный подграф* — это подграф, содержащий в точности t вершин. Матрица $A_f = (\alpha_{x,y})$, строки и столбцы которой помечены элементами из X, причем $\alpha_{x,y} = 1$, если $(x, y) \in f$ и $\alpha_{x,y} = 0$, если $(x, y) \notin f$, называется *матрицей* отношения f или *матрицей смежности* гра-

[1]) M. D. Hestens, D. G. Higman, Rank 3 groups and strongly regular graphs, SIAM—AMS Proceedings, vol. 4, Providence, 1971, 141—159.

фа $\mathcal{G}_f$. *Дополнительным* графом к графу из $\mathcal{G}_f$ называется граф $\bar{\mathcal{G}}_f = \mathcal{G}_{\bar{f}}$, где $\bar{f} = X \times X - (f \cup I)$, $I = \{(x, x) \mid x \in X\}$.

Граф $\mathcal{G}_f$ является *связным*, если существует *путь* (длины n) из любой вершины x в любую другую вершину y, т. е. такая конечная последовательность $x_0 = x, x_1, \ldots, x_n = y$ вершин, что $(x_i, x_{i+1}) \in f$ для $i = 0, 1, \ldots, n-1$.

Для заданной вершины x число вершин в множестве

$$f(x) = \{y \mid (x, y) \in f\}$$

вершин, *смежных* с x, называется *валентностью* x. Граф называется *регулярным*, если все его вершины имеют одну и ту же валентность; это в точности означает, что матрица A_f имеет постоянные строковые суммы.

m-Клика в обычном графе это m-вершинный подграф, в котором любые две вершины смежны, а *m-лапа* это $(m+1)$-вершинный подграф, в котором одна из вершин смежна со всеми остальными, которые между собой попарно несмежны.

Пусть G — транзитивная группа подстановок на множестве X. Обозначим действие G на X следующим образом: $x \to x^g$, $x \in X$, $g \in G$. Число G-орбит на множестве $X \times X$ (относительно покомпонентного действия), число G_x-орбит [1]) на X для фиксированного элемента $x \in X$ и число (G_x, G_x)-двойных смежных классов в G равны между собой. Это следует из того, что для каждой G-орбиты Δ на множестве $X \times X$ и каждого элемента $x \in X$ отображения $\Delta \to \Delta(x) \to \{g \in G \mid x^g \in \Delta(x)\}$ являются биекциями [2]). Назовем G-орбиты на множестве $X \times X$ *орбиталями*, а их число — *рангом* G.

Если Δ орбиталь, то $\Delta(x)^g = \Delta(x^g)$ для всех $x \in X$ и всех $g \in G$. Более того, обращение $\Delta^\smile = \{(y, x) \mid (x, y) \in \Delta\}$ орбитали Δ также является орбиталью, поэтому $\Delta \to \Delta^\smile$ — спаривание на множестве орбиталей [15, § 16]. Каждая орбиталь является либо симметричной $(\Delta = \Delta^\smile)$, либо строго антисимметричной $(\Delta \cup \Delta^\smile = \varnothing)$. Отметим, что $\Delta^\smile(x) \to \{g \in G \mid x^{g^{-1}} \in \Delta(x)\}$, поэтому (G_x, G_x) — двойной смежный класс, соответствующий $\Delta^\smile$ является обратным к двойному смежному классу, соответствующему Δ.

(1.1) [15, с. 45] *G обладает симметричной орбиталью, отличной от I тогда и только тогда, когда $|G|$ — четное число.*

Доказательство. Выберем $\bar{x} \in X$. G имеет симметричную орбиталь, неравную I тогда и только тогда, когда $G_x g G_x = G_x g^{-1} G_x$ для некоторого $g \notin G_x$, а это возможно в точности тогда, когда $|G|$ четен.

---

[1]) $G_x = \{g \mid g \in G, x^g = x\}$. — *Прим. перев.*

[2]) $\Delta(x) = \{y \mid y \in X, (x, y) \in \Delta\}$. — *Прим. перев.*

Как хорошо известно (см. [15]), справедливо следующее утверждение.

(1.2) *Матрицы A_Δ составляют базис кольца централизаторов подстановочного представления.*

В частности, ранг G равен размерности соответствующего кольца централизаторов, а поэтому и длине (ρ, ρ) подстановочного характера ρ.

Каждая орбиталь $\Delta \neq I$ строго антирефлексивна и, как отмечено выше, либо симметрична, либо строго антисимметрична. Следовательно, граф $\mathcal{G}_\Delta$ является либо обычным, либо ориентированным.

(1.3) [5, с. 26] *G является примитивной тогда и только тогда, когда граф $\mathcal{G}_\Delta$ связен для всех $\Delta \neq I$.* •

Доказательство. Пусть Δ — орбиталь, отличная от I. Для $x \in X$ положим $\Sigma_\Delta(x) = \{y \in X |$ в $\mathcal{G}_\Delta$ существует путь из x в $y\}$. Тогда $\Sigma_\Delta(x)^g = \Sigma_\Delta(x^g)$ для всех $x \in X$ и $g \in G$, а для $y \in \Sigma_\Delta(x)$ имеет место $\Sigma_\Delta(y) \subseteq \Sigma_\Delta(x)$, поэтому $\Sigma_\Delta(y) = \Sigma_\Delta(x)$. Следовательно $\Sigma_\Delta(x)$ — блок импримитивности, а если G примитивна, то $\Sigma_\Delta(x) = X$ и граф $\mathcal{G}_\Delta$ связен. С другой стороны, если G импримитивна и Σ — нетривиальный блок импримитивности, то для $x \in \Sigma$ множество $\Sigma = \{x\} + \Delta(x) + \ldots$ является объединением G_x-орбит, причем встретится хотя бы одна орбиталь Δ, отличная от I. Если $y \in \Delta(x)$, то $y = x^g$ для некоторого $g \in G$ и $\Delta(y) = \Delta(x^g) = \Delta(x)^g \subseteq \Sigma^g = \Sigma$. Следовательно $\Sigma_\Delta(x) \subseteq \Sigma$ и $\mathcal{G}_\Delta$ не является связным.

2. ГРУППЫ РАНГА 3 И СИЛЬНО РЕГУЛЯРНЫЕ ГРАФЫ

Пусть G — группа подстановок ранга 3 на множестве X с орбиталями I, Δ и Γ. Для $x \in X$ G_x-орбиты на X — это $I(x) = \{x\}$, $\Delta(x)$ и $\Gamma(x)$. Числа $1 = |I(x)|$, $k = |\Delta(x)|$ и $l = |\Gamma(x)|$ называются *подстепенями* G, а $n = 1 + k + l$, где $n = |X|$, — *степенью* G. Предположим, что $|G|$ четен, при этом хотя бы одна, а значит и обе орбитали Δ, Γ являются симметричными. Обычный граф $\mathcal{G}_\Delta$ является регулярным и каждая его вершина имеет валентность k. *Числа пересечений* λ, μ *группы* G [4] определяются следующим образом:

$$|\Delta(x) \cap \Delta(y)| = \begin{cases} \lambda, & \text{если } y \in \Delta(x), \\ \mu, & \text{если } y \in \Gamma(x). \end{cases}$$

В $\mathcal{G}_\Delta$ величина λ равна числу 3-клик (т. е. треугольников), содержащих заданное ребро (x, y), а μ — числу 2-лап, содержащих заданное неребро и эти величины не зависят от выбора ребра и

неребра соответственно. Это означает, что $\mathcal{G}_\Delta$ и $\mathcal{G}_\Gamma$ — пара дополнительных *сильно регулярных графов* [1]. Такие графы будут детально рассмотрены в §§ 4—6.

Сильно регулярный граф, возникающий таким образом из группы подстановок ранга 3, т. е. сильно регулярный граф, группа автоморфизмов которого имеет ранг 3 при действии на вершинах будет называться графом ранга 3.

3. ТЕОРЕМЫ ИЗ ТЕОРИИ МАТРИЦ

В этом разделе мы приведем некоторые результаты о матрицах, которые будут играть важную роль в нашем обсуждении сильно регулярных графов.

Пусть A — действительная симметричная матрица порядка n со спектром $\lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n$. Хорошо известно, что

(3.1) *Если B — главная подматрица в A, то*

(а) *спектр B содержится в $[\lambda_1, \lambda_n]$ и*

(б) *если λ_j, $j = 1$ или n является собственным значением B, y — собственный вектор B с собственным значением λ_j, а z — собственный вектор A с собственным значением, отличным от λ_j, то y перпендикулярен проекции z на подпространство, соответствующее B.*

Под *спектром* графа $\mathcal{G}$ мы понимаем спектр его матрицы смежности A. Если $\mathcal{G}$ обычный граф, то A симметрично, поэтому из (3.1) непосредственно следует

(3.2) *Пусть $\mathcal{G}$ — обычный граф. Предположим, что $\mathcal{G}$ регулярен и каждая вершина имеет валентность $k > 0$ и пусть λ_1 — минимальное собственное значение $\mathcal{G}$. Если $\mathcal{G}_0$ — подграф $\mathcal{G}$, то спектр $\mathcal{G}_0$ содержится в $[\lambda_1, k]$. Если λ_1 является собственным значением $\mathcal{G}_0$, то любой собственный вектор $\mathcal{G}_0$ с собственным значением λ_1 имеет нулевую сумму координат.*

3-Лапа $[P; Q_1, Q_2, Q_3]$ — это граф

Используя (3.2), А. Дж. Хоффман [8] доказал

(3.3) *Если обычный граф $\mathcal{G}$, имеющий минимальное собственное значение равное -2, содержит 3-лапу $[P; Q_1, Q_2, Q_3]$, то каждая вершина $R \neq P$, смежная с Q_1 и Q_2, смежна с P, но не смежна с Q_3.*

Последний результат этого раздела в определенном смысле двойствен к (3.1). Пусть снова A — действительная симметрич-

ная матрица порядка n со спектром $\lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n$. Для заданного разбиения $\{1, 2, \ldots, n\} = \Delta_1 + \Delta_2 + \ldots + \Delta_m$, такого что $|\Delta_i| = n_i > 0$, рассмотрим разложение матрицы A на блоки: $A = (A_{ij})$, где $A_{ij} - n_i \times n_j$ блок. Пусть e_{ij} — сумма элементов в блоке A_{ij} и положим $\bar{A} = (e_{ij}/n_i)$. Как указал Симс, рассуждения, аналогичные стандартному доказательству утверждения (3.1), позволяют получить следующее:

(3.4) (а) *Спектр матрицы $\bar{A}$ содержится в $[\lambda_1, \lambda_n]$ и*

(б) *если λ_1 — собственное значение $\bar{A}$, y — собственный вектор $\bar{A}$ с собственным значением λ_1 и z — собственный вектор A с собственным значением, большим, чем λ_1, то* $\sum_{i=1}^{m} \sum_{a \in \Delta_i} z_a y_i = 0$.

Этот результат будет применяться к матрице смежности обычного графа $\mathcal{G}$ в случае, когда $\Delta_1, \ldots, \Delta_m$ — разбиение множества вершин. При этом e_{ij} — число ориентированных ребер из Δ_i в Δ_j.

4. СИЛЬНО РЕГУЛЯРНЫЕ ГРАФЫ

Сильно регулярный граф [1] — это обычный граф $\mathcal{G}$ с n вершинами, такой, что

(1) $\mathcal{G}$ регулярный валентности k, $0 < k < n - 1$,

(2) число λ 3-клик (т. е. треугольников) в $\mathcal{G}$, содержащих заданное ребро, не зависит от выбора ребра, и

(3) число μ 2-лап, содержащих заданное неребро, не зависит от выбора неребра.

Пусть $\mathcal{G}$ — сильно регулярный граф, Δ — множество ребер $\mathcal{G}$, а Γ — множество его неребер. Тогда $\mathcal{G} = \mathcal{G}_\Delta$, $\bar{\mathcal{G}} = \mathcal{G}_\Gamma$, $\Delta(x)$ — множество вершин, смежных с x, т. е. $|\Delta(x)| = k$ и $\Gamma(x)$ — множество вершин, отличных от x и несмежных с x. Положим $l = |\Gamma(x)|$, так что $n = 1 + k + l$. Из определений следует

(4.1) $\qquad 0 \leqslant \lambda \leqslant k - 1$, $0 \leqslant \mu \leqslant k$ *и* $k - \mu \leqslant l - 1$.

Матрицы смежности $A = A_\Delta$ графа $\mathcal{G}$ и $B = B_\Gamma$ дополнения $\bar{\mathcal{G}}$ графа $\mathcal{G}$ удовлетворяют следующим соотношениям:

(4.2) (i) $I + A + B = J$,

(ii) A *имеет строковую сумму* k, B *имеет строковую сумму* l *и*

(iii) $A^2 - (\lambda - \mu) A - (k - \mu) I = \mu J$, *где* J *обозначает матрицу, состоящую целиком из единиц.*

Доказательство. Первые два утверждения следуют непосредственно из определений. Поскольку (x, y)-элемент матрицы A^2

равен числу путей длины 2 в графе $\mathcal{G}$ из x в y, мы имеем $A^2 = \lambda A + \mu B + kI$ и соотношение (iii) следует из (i) и (ii).

Нетрудно заметить, что (0, 1)-матрица со строковой суммой k, $0 \leqslant k \leqslant n-1$, имеющая на диагонали одни нули и удовлетворяющая равенству (iii) для $0 \leqslant \lambda \leqslant k-1$ и $0 \leqslant \mu \leqslant k$ является матрицей смежности сильно регулярного графа.

Из определений следует

(4.3) *$\mathcal{G}$ связен тогда и только тогда, когда $\mu \neq 0$ и*

(4.4) *дополнительный граф $\bar{\mathcal{G}}$ сильно регулярен, причем*

$$\bar{n} = n, \quad \bar{k} = l, \quad \bar{l} = k, \quad \bar{\lambda} = l - k + \mu - 1$$

и

$$\bar{\mu} = l - k + \lambda + 1.$$

Число ребер в $\mathcal{G}$ равно $nk/2$, следовательно

(4.5) *По крайней мере одно из чисел k, l является четным.*

Число треугольников с заданной вершиной равно $k\lambda/2$, а общее число треугольников — $nk\lambda/6$, поэтому

(4.6) $2 \mid k\lambda$ *и* $3 \mid nk\lambda$ *и, двойственно,* $2 \mid l\bar{\lambda}$ *и* $3 \mid nl\bar{\lambda}$.

Число 2-лап, содержащих заданное ребро (x, y), равно $k - \lambda - 1$, а число 2-лап, содержащих заданное неребро (x, z), равно μ. Следовательно, общее число 2-лап, содержащих x, равно

(4.7) $$k(k - \lambda - 1) = \mu l.$$

Из (4.6) и (4.7) следует

(4.8) $$2 \mid \mu l \text{ и } 2 \mid \bar{\mu} k.$$

Из (4.3), (4.4) и (4.7) получаем

(4.9) *$\bar{\mathcal{G}}$ связен тогда и только тогда, когда $\mu \neq k$.*

Мы будем говорить, что $\mathcal{G}$ *примитивный*, если $\mathcal{G}$ и $\bar{\mathcal{G}}$ являются связными.

(4.10) *$\mathcal{G}$ примитивен тогда и только тогда, когда $\mu \neq 0$, k.*

В силу (4.7) имеем

(4.11) *Если $\mathcal{G}$ примитивен, то $(k, l) \neq 1$.*

Из (4.2) следует, что A удовлетворяет многочлену

(4.12) $$(x - k)(x^2 - (\lambda - \mu)x - (k - \mu)) = 0.$$

Таким образом, спектр $\mathcal{G}$ состоит из k и двух собственных значений

$$\begin{Bmatrix} r \\ s \end{Bmatrix} = \frac{\lambda - \mu \pm d^{1/2}}{2}, \quad d = (\lambda - \mu)^2 + 4(k - \mu) \neq 0. \tag{4.13}$$

Собственные значения $\begin{Bmatrix} \bar{r} \\ \bar{s} \end{Bmatrix} = (\lambda - \mu - 2 \pm d^{1/2})/2$ графа $\bar{\mathcal{G}}$ связаны с собственными значениями $\mathcal{G}$ соотношениями

$$r + \bar{s} = s + \bar{r} = -1, \tag{4.14}$$

что можно заметить либо из явных формул, либо путем одновременной диагонализации A и B.

Если $\mathcal{G}$ связен, то A неприводима, и поэтому k имеет кратность 1 по теореме Перрона — Фробениуса. Если же $\mathcal{G}$ несвязен, то $\begin{Bmatrix} r \\ s \end{Bmatrix} = \begin{Bmatrix} k \\ -1 \end{Bmatrix}$. Собственные значения r и s различны и имеют соответственно кратности f и g (при подходящей интерпретации в случае, когда $\mathcal{G}$ несвязен), которые в силу соотношений

$$n = 1 + f + g,$$
$$0 = k + rf + sg \quad (\text{след } A = 0)$$

равны

$$f = \frac{(n-1)s + k}{s - r} \quad \text{и} \quad g = \frac{(n-1)r + k}{r - s}. \tag{4.15}$$

То есть

$$\begin{Bmatrix} f \\ g \end{Bmatrix} = \frac{(k + l)(\lambda - \mu) + 2k \mp d^{1/2}(k + l)}{\mp 2d^{1/2}}. \tag{4.16}$$

Отметим, что $\bar{f} = g$ и $\bar{g} = f$ это, соответственно, кратности $\bar{r}$ и $\bar{s}$ как собственных значений $\bar{\mathcal{G}}$.

Мы будем называть девятку $(n, k, l, \lambda, \mu, r, s, f, g)$ *набором параметров, описывающих* $\mathcal{G}$.

Матрица $C = B - A$ (так называемая $(-1, 1, 0)$-матрица смежности, см. [11]) имеет собственные значения $\rho_0 = l - k$ (строковая и столбцовая сумма), $\rho_1 = -(2r + 1)$ и $\rho_2 = -(2s + 1)$ с кратностями 1, f и g, соответственно. Из равенств

$$A = -\tfrac{1}{2}(B - A) - \tfrac{1}{2}I + \tfrac{1}{2}J, \quad B = \tfrac{1}{2}(B - A) - \tfrac{1}{2}I + \tfrac{1}{2}J,$$

$$A^2 = (\lambda - \mu)A + (k - \mu)I + \mu J, \quad B^2 = (\mu - \lambda - 2)B + (k - \lambda - 1)I + \mu J$$

и

$$AB = -2(k - \lambda - 1)A - 2(k - \mu)B,$$

мы получаем, что

$$C^2 - (\rho_1 + \rho_2)C + \rho_1\rho_2 I = (n - 1 + \rho_1\rho_2)J. \tag{4.17}$$

Поскольку f и g положительные целые, мы получаем из (4.16) следующее:

(4.18) [4, Лемма 7] *Имеет место одно из следующих утверждений:*

(I) $k = l = 2\mu = 2(\lambda + 1)$, *или*

(II) $d = (\lambda - \mu)^2 + 4(k - \mu)$ *является квадратом и*

(i) *если* n *четно, то* $d^{1/2}$ *делит* $2k + (\lambda - \mu)(k + l)$, *в то время как* $2d^{1/2}$ *не делит и*

(ii) *если* n *нечетно, то* $2d^{1/2}$ *делит* $2k + (\lambda - \mu)(k + l)$. *Если* $f = g$, *то имеет место случай* (I). *В случае* (II) *собственные значения* $\mathcal{G}$ *целые. Случаи* (I) *и* (II) *не являются взаимно исключающими.* $\overline{\mathcal{G}}$ *относится к случаю* (I) *или* (II) *в зависимости от того, относится ли* $\overline{\mathcal{G}}$ *к случаю* (I) *или* (II).

В случае (I)

$$n = 1 + 4\mu, \quad d = n^{1/2}, \quad \begin{Bmatrix} r \\ s \end{Bmatrix} = \frac{-1 \pm n^{1/2}}{2} \quad \text{и} \quad f = g = k.$$

Единственными известными нам такими сильно регулярными графами являются графы ранга 3 зингерова типа [7], а также три примера, имеющие $n = 7^2$, 23^2 и 47^2 соответственно, которые связаны с исключительными разрешимыми группами ранга 3. Дж. Я. Сайдел получил следующий полезный результат

(4.19) *Если* $\mathcal{G}$ *— сильно регулярный граф, для которого* $k = l = 2\mu = 2(\lambda + 1)$, *то* n *является суммой двух квадратов.*

Доказательство. Положим

$$D = \begin{bmatrix} 0 & j \\ j^T & C \end{bmatrix},$$

где $j = (1, 1, \ldots, 1)$ и $C = B - A$. В силу (4.17), $D^2 = nI$. Поскольку D — симметричная, рациональная матрица порядка, сравнимого с 2 по модулю 4, отсюда теория рациональных конгруэнтностей дает нам, что n — сумма двух квадратов.

Предположим, что $\mathcal{G}$ связный и импримитивный, так что $\mu = k$ и $\lambda = k - l - 1$. Тогда

$$\begin{Bmatrix} r \\ s \end{Bmatrix} = \begin{Bmatrix} 0 \\ -(l+1) \end{Bmatrix}$$

и

$$\begin{Bmatrix} f \\ g \end{Bmatrix} = \begin{Bmatrix} \frac{nl}{l+1} \\ \frac{k}{l+1} \end{Bmatrix}.$$

В частности,

(4.20) *Если $\mathcal{G}$ импримитивен, то $l+1 \mid k$ или $k+1 \mid l$ в зависимости от того, связан $\mathcal{G}$ или нет.*

По-другому мы можем получить (4.20), заметив, что если $\bar{\mathcal{G}}$ несвязан, то компоненты связности — это множества $\{x\}+\Gamma(x)$. Если G — импримитивная группа ранга 3 (и, следовательно, с необходимостью четного порядка), причем, скажем, $\mathcal{G}_\Gamma$ несвязен, то имеется единственное импримитивное разложение $\mathcal{G}$ и блоками являются множества $\{x\}+\Gamma(x)$. G дважды транзитивна на множестве блоков, а стабилизатор блока дважды транзитивен на этом блоке.

(4.21) *Если $\mathcal{G}$ примитивный и $n \neq 5$, то $r \geqslant 1$ и $s \leqslant -2$.*

Доказательство. В силу (4.18) $r \geqslant 1$ и $\tilde{r} \geqslant 1$, следовательно, $s \leqslant -2$ по равенству (4.14).

Выберем вершину x и пусть $\Delta_1 = \{x\}$, $\Delta_2 = \Delta(x)$ и $\Delta_3 = \Gamma(x)$, и пусть

$$e_{ij} \text{ — число ориентированных ребер из } \Delta_i \text{ в } \Delta_j.$$

Тогда

$$(e_{ij}) = \begin{bmatrix} 0 & k & 0 \\ k & k\lambda & k(k-\lambda-1) \\ 0 & \mu l & (k-\mu)l \end{bmatrix}.$$

Матрица $\hat{A}(x)=(e_{ij}/n_i)$, $n_i = |\Delta_i|$ называется *матрицей пересечений* графа $\mathcal{G}$. Мы видим, что

$$\hat{A}(x) = \begin{bmatrix} 0 & k & 0 \\ 1 & \lambda & k-\lambda-1 \\ 0 & \mu & k-\mu \end{bmatrix}$$

не зависит от выбора x и имеет те же собственные значения, k, r, s, что и A.

Более того,

(4.22) *Для $\theta \in \{k, r, s\}$ вектор $(k, \theta, \rho)^T$, где*

$$\rho = \frac{\theta^2 - \lambda\theta - k}{k-\lambda-1} = \frac{\mu\theta}{\theta - k + \mu}$$

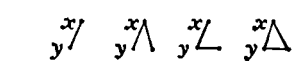
является собственным вектором $\hat{A}(x)$ с собственным значением θ.

Образуя $B(x)$ и B относительно того же разбиения вершин, таким образом, что $B(x)$ — матрица пересечений графа $\bar{\mathcal{G}}$, мы получаем $I + \hat{A}(x) + B(x) = J$, где каждая строка матрицы J

равна $(1, k, l)$. Из общей теории матриц пересечений ([5] и [6]) известно, что $\{I, A, B\}$ и $\{I, A(x), B(x)\}$ — базисы изоморфных матричных алгебр.

5.4-ВЕРШИННЫЕ ПОДГРАФЫ

Пусть $\mathcal{G}$ — обычный граф. Скажем, что два подграфа в $\mathcal{G}$ имеют *один и тот же тип относительно пары* (x, y) *различных вершин*, если оба содержат и x, и y, а кроме того существует изоморфизм одного подграфа на другой, обладающий тем свойством, что x переходит в x, а y — в y. Например, если (x, y) — ребро, то возможные типы 3-вершинных подграфов в $\mathcal{G}$ относительно (x, y) исчерпываются следующими:



Ясно, что графы ранга 3 удовлетворяют следующему условию (которое мы будем называть *t-вершинным условием*) для всех $t \geqslant 2$. Число t-вершинных подграфов заданного типа относительно заданной пары (x, y) различных вершин зависит лишь от того, является (x, y) ребром или нет.

Если $\mathcal{G}$ удовлетворяют t-вершинному условию для некоторого t, то ему удовлетворяет и $\bar{\mathcal{G}}$. t-Вершинное условие бессодержательно при $t = 1$ и автоматически выполнено для $t = 2$ в любом обычном графе. Если $\mathcal{G}$ не является ни полным графом, ни его дополнением, то 3-вершинное условие выполнено для $\mathcal{G}$ тогда и только тогда, когда $\mathcal{G}$ сильно регулярен; в этом случае число 3-вершинных подграфов может быть получено из матриц пересечений для $\mathcal{G}$ и $\bar{\mathcal{G}}$. Следуя методу Симса, мы будем интересоваться 4-вершинными подграфами в сильно регулярном графе. В частности, утверждения (5.21), (5.22) и (5.23), а также метод применения утверждения (3.4) принадлежат Симсу.

Предполагая, что $\mathcal{G}$ сильно регулярен, выберем ребро (x, y) и определим $\Delta_i = \Delta_i(x, y)$ по следующему правилу:

$$\Delta_1 = \{x\}, \quad \Delta_2 = \{y\}, \quad \Delta_3 = \Delta(x) \cap \Delta(y),$$
$$\Delta_4 = \Delta(x) \cap \Gamma(y), \quad \Delta_5 = \Delta(y) \cap \Gamma(x) \text{ и } \Delta_6 = \Gamma(x) \cap \Gamma(y).$$

Количества вершин в этих множествах следующие:

$$n_1 = n_2 = 1, \quad n_3 = \lambda, \quad n_4 = n_5 = k - \lambda - 1 \text{ и } n_6 = \bar{\mu}.$$

Как и в разд. 3 положим $e_{ij} \equiv e_{ij}(x, y)$ равным числу ориентированных ребер из Δ_i в Δ_j. При этом

$$0 \leqslant e_{ij} = e_{ji} \leqslant n_i(n_j - \delta_{ij}) \tag{5.1}$$

и

$$e_{11} = e_{15} = e_{16} = e_{22} = e_{24} = e_{26} = 0,$$
$$e_{12} = 1, \quad e_{13} = e_{23} = \lambda \quad \text{и} \quad e_{14} = e_{25} = k - \lambda - 1. \tag{5.2}$$

Положим $\alpha \equiv \alpha(x, y)$ равным числу 4-клик в $\mathcal{G}$, содержащих x и y. При этом $e_{33} = 2\alpha$, и мы можем вычислить оставшиеся e_{ij} в терминах k, λ, μ, $\bar{\mu}$ и α.

Для проведения вычислений положим $e(S, T)$ равным числу ориентированных ребер из S в T для произвольных двух множеств вершин S и T. Тогда $e_{ij} = e(\Delta_i, \Delta_j)$ и можно заметить, что

(5.3) *Для* $z \in \Delta_i$

$$\begin{aligned}
e(z, \Delta(x)) &= k \quad и \quad e(z, \Gamma(x)) = 0, && i = 1,\\
&= \lambda, && = k - \lambda - 1, && i = 2,\ 3 \text{ или } 4,\\
&= \mu, && = k - \mu, && i = 5 \text{ или } 6;\\
и\ e(z, \Delta(y)) &= k \quad и \quad e(z, \Gamma(y)) = 0, && i = 2,\\
&= \lambda, && = k - \lambda - 1, && i = 1;\ 3 \text{ или } 5,\\
&= \mu, && = k - \mu, && i = 4 \text{ или } 6.
\end{aligned}$$

Далее,

(5.4) *Для* $z \in \Delta_i$

$$n_i e(z, \Delta(x)) = e_{i2} + e_{i3} + e_{i4} \quad и \quad n_i e(z, \Gamma(x)) = e_{i5} + e_{i6},$$
$$n_i e(z, \Delta(y)) = e_{i1} + e_{i3} + e_{i5} \quad и \quad n_i e(z, \Gamma(y)) = e_{i4} + e_{i6}.$$

Комбинируя (5.1) с учетом (5.4), получаем

$$\begin{aligned}
&e_{33} = 2\alpha,\\
&e_{34} = e_{35} = \lambda(\lambda - 1) - 2\alpha,\\
&e_{36} = e_{44} = e_{55} = \lambda(k - 2\lambda) + 2\alpha,\\
&e_{45} = (\mu - 1)(k - \lambda - 1) - \lambda(\lambda - 1) + 2\alpha,\\
&e_{46} = e_{56} = (k - \lambda - \mu)(k - \lambda - 1) + \lambda(\lambda - 1) - 2\alpha,\\
&e_{66} = \bar{\mu}(k - \mu) - (k - \lambda - \mu)(k - \lambda - 1) - \lambda(\lambda - 1) + 2\alpha.
\end{aligned} \tag{5.5}$$

Мы видим, что любая из величин e_{ij} при $i \geqslant 3$, $j \geqslant 3$ определяет все остальные. Более того, число 4-вершинных подграфов любого заданного типа относительно (x, y) определяет все такие e_{ij} и определяется по любому из них. Следовательно,

(5.6) *Если число 4-вершинных подграфов в* $\mathcal{G}$ *некоторого одного типа относительно* (x, y) *не зависит от выбора ребра* (x, y), *то также не зависит от этого выбора число 4-вершинных подграфов каждого типа относительно* (x, y).

Заметим, что

(5.7) *Число 3-лап*

в $\mathcal{G}$ *равно*

$$\tfrac{1}{2}\{(k-\lambda-1)(k-2\lambda-2)+\lambda(\lambda-1)-2\alpha\},$$

поскольку это $\tfrac{1}{2}\{n_4(n_4-1)-e_{44}\}$.

Предположим теперь, что $\mathcal{G}$ примитивен и что $\lambda \neq 0$, т. е. что все n_i отличны от нуля. В виду (4.22), (5.3) и (5.4) мы заключаем, что для $\theta \in \{r, s\}$ матрица

$$\hat{A}(x, y) = (e_{ij}/n_i)$$

имеет $(k, \theta, \theta, \theta, \varphi, \rho)^T$ и $(\theta, k, \theta, \rho, \theta, \rho)^T$ в качестве собственных векторов с собственным значением θ, где ρ то же, что и в (4.22).

Поскольку таким образом мы получаем четыре линейно независимых вектора, то и r, и s имеют как собственные значения $\hat{A}(x, y)$ кратность по меньшей мере 2. Так как k является собственным значением с кратностью 1 как строковая сумма, мы получаем, что

$$t = \operatorname{trace} \hat{A}(x, y) - (k+2)(r+s))$$

является собственным значением $\hat{A}(x, y)$. В силу (5.2) и (5.5)

$$\operatorname{trace} \hat{A}(x, y) = \frac{2\alpha}{\lambda} + 2\lambda - 2\frac{\lambda(\lambda-1)-2\alpha}{k-\lambda-1} + k - \mu - \frac{(k-\mu-\lambda)(k-\lambda-1)+(\lambda(\lambda-1)-2\alpha)}{\bar{\mu}}.$$

Воспользовавшись равенствами $\bar{\mu}k = l(k-\mu)$ и $r+s = \lambda-\mu$, мы получаем

$$t = 2\left\{\frac{1}{\lambda} + \frac{2k-\mu}{(k-\lambda-1)(k-\mu)}\right\}\alpha + \frac{k\lambda(\mu-2\lambda+2)-2\mu\lambda}{(k-\lambda-1)(k-\mu)}. \tag{5.8}$$

Теперь мы повторим изложенную выше процедуру, начиная в этот раз с неребра $(a, b) \in \Gamma$ и рассматривая множества $\Gamma_i = \Gamma_i(a, b)$, определяемые следующим образом:

$$\Gamma_1 = \{a\}, \quad \Gamma_2 = \{b\}, \quad \Gamma_3 = \Delta(a) \cap \Delta(b),$$
$$\Gamma_4 = \Delta(a) \cap \Gamma(b), \quad \Gamma_5 = \Gamma(a) \cap \Delta(b) \quad \text{и} \quad \Gamma_6 = \Gamma(a) \cap \Gamma(b),$$

которые содержат $m_1 = m_2 = 1$, $m_3 = \mu$, $m_4 = m_5 = k-\mu$ и $m_6 = \bar{\lambda}$ вершин соответственно. Положим

$$f_{ij} = f_{ij}(a, b) = e(\Gamma_i, \Gamma_j).$$

Тогда

(5.9) $$0 \leqslant f_{ij} = f_{ji} \leqslant m_i (m_j - \delta_{ij})$$

и

(5.10) $$f_{11} = f_{12} = f_{15} = f_{16} = f_{22} = f_{24} = f_{26} = 0,$$
$$f_{13} = f_{23} = \mu \quad и \quad f_{14} = f_{25} = k - \mu.$$

Далее положим $\beta = \beta(a, b)$ — число подграфов в $\mathcal{G}$ вида

тогда $f_{33} = 2\beta$. Вычислим теперь все оставшиеся f_{ij} в терминах k, λ, $\bar{\lambda}$, μ и β. Метод тот же, что и для e_{ij}, поэтому мы опускаем детали. Результат следующий

$$f_{33} = 2\beta,$$
$$f_{34} = f_{35} = \mu\lambda - 2\beta,$$
$$f_{36} = \mu(k - 2\lambda - 2) + 2\beta,$$
(5.11) $$f_{44} = f_{55} = (k - \mu)\lambda - \mu\lambda + 2\beta,$$
$$f_{45} = (k - \mu)\mu - \mu\lambda + 2\beta,$$
$$f_{46} = f_{56} = (k - \mu)(k - \lambda - \mu - 1) + \mu\lambda - 2\beta,$$
$$f_{66} = (k - 2\mu)\bar{\lambda} + \mu(k - 2\lambda - 2) + 2\beta.$$

Здесь мы снова замечаем, что любая из f_{ij} при $i \geqslant 3$, $j \geqslant 3$ определяет все остальные. Поскольку число 4-вершинных подграфов каждого типа относительно (a, b) определяет одно из этих f_{ij} и наоборот, мы получаем

(5.12) *Если число 4-вершинных подграфов в $\mathcal{G}$ некоторого одного типа относительно (a, b) не зависит от выбора неребра (a, b), тогда от этого выбора не зависит число 4-вершинных подграфов каждого типа относительно (a, b).*

Комбинируя (5.6) и (5.12), мы видим, что

(5.13) *Если $e_{ij}(x, y)$ не зависит от выбора $(x, y) \in \Delta$ для некоторых $i \geqslant 3$, $j \geqslant 3$ и $f_{pq}(a, b)$ не зависит от выбора $(a, b) \in \Gamma$ для некоторых $p \geqslant 3$, $q \geqslant 3$, то $\mathcal{G}$ удовлетворяет 4-вершинному условию.*

(5.14) *Отметим, что число* 3-*лап*

в $\mathcal{G}$ — это $f_{36} = \mu(k - 2\lambda - 2) + 2\beta$.

Из (5.7), (5.13) и (5.14) получаем

(5.15) *Следующие утверждения эквивалентны:*

(а) $\mathcal{G}$ *не содержит* 3-*лап*;

(б) $\alpha(x, y) = \frac{1}{2}\{(k-\lambda-1)(k-2\lambda-2)+\lambda(\lambda-1)\}$ *для всех* $(x, y) \in \Delta$;

(в) $\beta(a, b) = -\frac{1}{2}\{\mu(k-2\lambda-2)\}$ *для всех* $(a, b) \in \Gamma$;

(г) $\mathcal{G}$ *удовлетворяет* 4-*вершинному условию, причем* α *и* β *принимают значения, заданные в* (б) *и* (в).

Предположим снова, что $\mathcal{G}$ примитивный и пусть теперь $\bar{\lambda} \neq 0$. Тогда все m_i отличны от нуля. Используя те же рассуждения, что и выше, мы замечаем, что

$$u = \operatorname{trace} \hat{A}(a, b) - (k + 2(r+s))$$

является собственным значением

$$\hat{A}(a, b) = (f_{ij}/m_i).$$

В силу (5.10) и (5.11) имеем

$$\operatorname{trace} \hat{A}(a, b) = \frac{2\beta}{\mu} + 2\lambda - 2\frac{\mu\lambda - 2\beta}{k-\mu} + k - 2\mu + \frac{\mu(k-2\lambda-2)+2\beta}{\bar{\lambda}}.$$

Следовательно,

$$u = 2\left\{\frac{k+\mu}{\mu(k-\mu)} + \frac{1}{\bar{\lambda}}\right\}\beta - \frac{2\mu\lambda}{k-\mu} + \frac{\mu(k-2\lambda-2)}{\bar{\lambda}}. \tag{5.16}$$

Предполагая, что $\mathcal{G}$ примитивный и $\lambda \neq 0$, мы выбираем ребро $(x, y) \in \Delta$, т. е. неребров $\bar{\mathcal{G}}$, и рассматриваем соответствующее разбиение $\Delta_1, \ldots, \Delta_6$. Тогда

$$\operatorname{trace} \hat{A}(x, y) = k + 2(r+s) + t.$$

Но это в точности то разбиение, из которого мы вычислили $\bar{u}$ для нереброа (x, y) в $\bar{\mathcal{G}}$. Это означает, что если $\hat{B}(x, y)$ — матрица, получаемая из B с использованием этого разбиения, то

$$\operatorname{trace} \hat{B}(x, y) = l + 2(\bar{r}+\bar{s}) + \bar{u},$$

где $\bar{u}$ соответствует u и задается равенством (5.16), где над соответствующими параметрами следует поставить черту и учесть, что $\bar{\bar{\lambda}} = \lambda$. Поскольку $I + \hat{A}(x, y) + \hat{B}(x, y) = J$, где каждая строка в J равна $(n_1, n_2, \ldots, n_s)$, то, взяв след, мы видим, что

$$t + \bar{u} = -1. \tag{5.17}$$

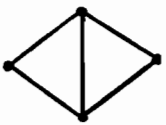
Из § 3 мы знаем, что $s \leqslant t$ и $\bar{s} \leqslant \bar{u}$. Следовательно,

(5.18) $s \leqslant t \leqslant r$ *в случае, когда* $\mathcal{G}$ *примитивный и* $\lambda \neq 0$.

Аналогично мы видим, что

(5.19) $u+\bar{l}=-1$ *и* $s\leqslant u\leqslant r$ *в случае, когда* $\mathcal{G}$ *примитивный и* $\bar{\lambda}\neq 0$.

Подсчитывая двумя способами общее число подграфов вида



в графе $\mathcal{G}$, мы получаем уравнение

$$\sum_{(x,\ y)\in\Delta}[n_4(n_4-1)-e_{44}(x,\ y)]=\sum_{(a,\ b)\in\Gamma}f_{44}(a,\ b).$$

которое сводится к следующему

$$nk\binom{\lambda}{2}-\sum_{(x,\ y)\in\Delta}\alpha(x,\ y)=\sum_{(x,\ y)\in\Gamma}\beta(x,\ y). \tag{5.20}$$

В частности,

(5.21) *Если* $\mathcal{G}$ *удовлетворяет 4-вершинному условию, то*

$$k\left\{\binom{\lambda}{2}-\alpha\right\}=l\beta.$$

Аналогичный подсчет общего числа 4-вершинных подграфов других типов дает тот же результат (5.21).

Общее число 4-клик в $\mathcal{G}$, содержащих вершину x, равно $\frac{1}{3}\sum_{y\in\Delta(x)}\alpha(x,\ y)$, а общее число 4-клик равно $\frac{1}{12}\sum_{(x,\ y)\in\Delta}\alpha(x,\ y)$. В частности,

(5.22) *Если* $\mathcal{G}$ *удовлетворяет 4-вершинному условию, то* $3\mid k\alpha$ *и* $4\mid nk\alpha$ *и, двойственно,* $3\mid l\bar{\alpha}$ *и* $4\mid nl\bar{\alpha}$.

Поскольку в этом случае относительные числа 4-клик равны $k\alpha/3$ и $nk\alpha/12$. Пусть, как обычно, черта означает соответствующий параметр для $\bar{\mathcal{G}}$, т. е. $\bar{\alpha}(a,\ b)=\binom{\bar{\lambda}}{2}-f_{66}(a,\ b)/2$ для $(a,\ b)\in\Gamma$ и $\bar{\beta}(x,\ y)=\binom{\bar{u}}{2}-e_{66}(x,\ y)/2$.

Из (5.1) нам известно, что $0\leqslant\alpha\leqslant\binom{\lambda}{2}$. Предположим, что $\alpha=\binom{\lambda}{2}$, т. е. что $\beta=0$. Тогда для заданной вершины x множества $\{y\}+(\Delta(x)\cap\Delta(y))$ для $y\in\Delta(x)$ являются $(\lambda+1)$-кликами, осуществляющими разбиение $\Delta(x)$. Более того, множества

$$\{x,\ y\}+(\Delta(x)\cap\Delta(y)),\quad (x,\ y)\in\Delta$$

являются $(\lambda+2)$-кликами и каждое ребро содержится ровно в одной из таких клик. Поэтому их общее число равно

$$\frac{nk}{2}\binom{\lambda+2}{2}^{-1}=\frac{nk}{(\lambda+1)(\lambda+2)}.$$

Отсюда

(5.23) *Если $\mathcal{G}$ удовлетворяет 4-вершинному условию и* $\alpha=\binom{\lambda}{2}$, *то* $\lambda+1\mid k$ *и* $\lambda+2\mid nk$.

В заключение мы отметим, что с использованием (5.21) равенство (5.8) сводится к следующему.

(5.24) *Если $\mathcal{G}$ примитивный,* $\lambda\neq 0$ *и $\mathcal{G}$ удовлетворяет 4-вершинному условию, то*

$$t=\frac{2\alpha}{\lambda}-\left\{\frac{4}{\mu}+\frac{2}{k-\mu}\right\}\beta+\frac{\lambda\mu}{k-\mu}.$$

6. СИЛЬНО РЕГУЛЯРНЫЕ ГРАФЫ С МИНИМАЛЬНЫМ СОБСТВЕННЫМ ЗНАЧЕНИЕМ -2

С целью иллюстрации методов разд. 4 и 5 мы дадим иное доказательство следующего результата (Сайдел [12]).

(6.1) *Если $\mathcal{G}$ — сильно регулярный граф с минимальным собственным значением* -2, *то имеет место один из следующих случаев*:

(I) $n=2m$, $k=\mu=2(m-1)$, $(m\geqslant 2)$.

(II) $n=m^2$, $k=2(m-1)$, $\mu=2$, $(m\geqslant 3)$.

(III) $n=\binom{m}{2}$, $k=2(m-2)$, $\mu=4$, $(m\geqslant 5)$.

(IV) $n=10$, $k=3$, $\mu=1$,

(V) $n=16$, $k=10$, $\mu=6$,

(VI) $n=27$, $k=16$, $\mu=8$.

Все графы с такими параметрами известны.

Случай (I). $\mathcal{G}$ изоморфен дополнению к *лестничному графу* (см. [11]).

Случай (II). *Решетчатый граф* $\mathcal{L}_2(m)$ порядка m является единственным при $m\neq 4$, а при $m=4$ существует ровно один граф, неизоморфный $\mathcal{L}_2(4)$, [13]. Этот граф содержит 3-лапу, не удовлетворяет 4-вершинному условию и не обладает транзитивной группой автоморфизмов.

Случай (III). *Треугольный граф* $\mathcal{T}(m)$ порядка m является единственным для $m\neq 8$. При $m=8$ существуют в точности

три исключительных графа ([2], [3] и [8]), все они обладают интранзитивными группами автоморфизмов.

Случай (IV). Единственный граф, так называемый *граф Петерсена* [9], который является дополнительным к $\mathcal{T}(5)$.

Случай (V). Единственный граф, *граф Клебша*, соответствующий 16 прямым на поверхности 4-го порядка Клебша [12, с. 295]. Этот граф допускает группу автоморфизма ранга 3.

Случай (VI). Единственный граф, *граф Шлефли*, соответствующий 27 прямым на кубической поверхности [12, с. 296]. Этот граф допускает $U_4(2) \cong O_6(-1, 2)$ в качестве группы автоморфизмов ранга 3.

Наше доказательство существенно отличается от доказательства Сайдела [12], в котором рассматривается более широкий класс *сильных графов*. Доказательство утверждения (6.1) с точностью до конечного числа исключений и опирающееся на характеризацию реберных графов, которую получил Рей-Чоудхури [10], было дано Симсом в работе [14].

Доказательство (6.1). Пусть $\mathcal{G}$ — сильно регулярный граф, у которого $s = -2$. В силу (4.13) имеем

(1) $$k = 2\lambda - \mu + 4 \quad и \quad r = \lambda - \mu + 2.$$

Если $\mu = 0$, то ввиду (1) и (4.7), $k = 2\lambda + 4 = \lambda + 1$, что невозможно. Следовательно,

(2) $$\mu > 0 \quad и \quad l = k(\lambda - \mu + 3)/\mu,$$

а по (4.15)

(3) $$f = k(2\lambda - \mu + 6)/\mu(\lambda - \mu + 4).$$

Формулами (1), (2) и (3) мы ниже будем широко пользоваться без явных ссылок.

Поскольку $l - k = k(\lambda - 2\mu + 3)/\mu$, то

(4) $k \leqslant l$, *если и только если* $2\mu \leqslant \lambda + 3$.

Если $\mathcal{G}$ imprimitивный, то $\mu = k$ в силу (4.10), поэтому $\mu = \lambda + 2$ и $l = 1$. Следовательно, n четно в силу (4.20), поэтому

(5) *Если* $\mathcal{G}$ *imprimitивный, то выполнено* (I).

Если $\mu = 1$, то $f = (2\lambda + 3)(2\lambda + 5)/(\lambda + 3)$, откуда следует, что $\lambda = 0$, поэтому имеем

(6) *Если* $\mu = 1$, *то имеет место* (IV).

С этого места мы предполагаем, что $\mathcal{G}$ *примитивный и что* $\mu \geqslant 2$. Полагая $m = \lambda + 2$, мы замечаем, что

(7) *Если* $\mu = 2$ *или* 4, *то имеет место* (II) *или* (III) *соответственно.*

Следующий наш шаг состоит в доказательстве такого утверждения:

(8) *Если* $\mathcal{G}$ *содержит* 3-лапу, *то выполнено либо* (II) *при* $m=4$, *либо* (III) *при* $m=8$.

Доказательство. Пусть $[P; Q_1, Q_2, Q_3]$ — некоторая 3-лапа. Из (3.3) мы знаем, что каждая из $\mu-1$ вершин, отличных от P и смежных с двумя из вершин Q_i, смежна с P, но несмежна с третьей из Q_i. В сумме получаем $3+3(\mu-1)$ вершин, смежных с P. Из них $2(\mu-1)$ смежны также с Q_1, поэтому $2(\mu-1)\leqslant\lambda$. Оставшиеся $\lambda-2(\mu-1)$ вершины смежны с Q_1 и с P, но не смежны ни с Q_2, ни с Q_3. Аналогичное утверждение справедливо для любой подстановки индексов $\{1, 2, 3\}$. Следовательно мы пишем по меньшей мере

$$3+3(\mu-1)+3\{\lambda-2(\mu-1)\}=3(\lambda-\mu)+6$$

вершин, смежных с P, т. е. $k\geqslant 3(\lambda-\mu)+6$. Следовательно, $0\leqslant k-3(\lambda-\mu)-6=2(\mu-1)-\lambda$, тем самым $\lambda\leqslant 2(\mu-1)$ и поэтому $\lambda=2(\mu-1)$. Отсюда $k=3\mu$ и из $f=3(3\mu+2)/(\mu+2)$ следует $\mu=2$, 4 или 10.

Первые два случая как раз и соответствуют списку (8). В третьем случае мы получаем $n=64$, $k=30$, $\lambda=18$, $\mu=10$, $r=10$ и $s=-2$. В силу (4.17) матрица $C=B-A$ удовлетворяет уравнению $C^2-18C+63=0$ и при подходящем упорядочении строк и столбцов, матрица ECE, где

$$E=\begin{bmatrix} I_{31} & 0 \\ 0 & -I_{33} \end{bmatrix},$$

имеет вид

$$\begin{bmatrix} 0 & -j \\ j^T & D \end{bmatrix}, \quad j=(1, \ldots, 1)$$

и удовлетворяет тому же уравнению. Следовательно, $D^2-18D+63=-J$, а так как D имеет строковую сумму -18, то это влечет существование сильно регулярного графа с параметрами $n=63$, $k=40$, $\lambda=28$, $\mu=20$, $r=10$ и $s=-2$. Такой граф не содержит 3-лап и поэтому удовлетворяет 4-вершинному условию вместе со своим дополнением. Для дополнения мы имеем $\bar{k}=22$, $\bar{\lambda}=1$ и $\bar{\mu}=11$, поэтому в силу (5.21), $\alpha=\beta=0$, а в силу (5.16) $\bar{u}=-2+99/14>1$, противоречие с тем, что $\bar{r}=1$ и с неравенством (5.19).

С этого места мы предполагаем, что $\mathcal{G}$ *не содержит* 3-*лап.* При этом в силу (5.16)

(9) $\mathcal{G}$ *удовлетворяет* 4-*вершинному условию, причем*

$$\alpha=\tfrac{1}{2}\{(k-\lambda-1)(k-2\lambda-2)+\lambda(\lambda-1) \text{ и } \beta=-\tfrac{1}{2}\mu(k-2\lambda-2).$$

Пусть P и Q — несмежные вершины. Тогда $\Delta(P)=\Delta+\Lambda$, где $\Delta=\Delta(P)\cap\Delta(Q)$ и $\Lambda=\Delta(P)\cap\Gamma(Q)$. Пусть m — число вершин, смежных с P, Q и X для некоторой вершины $X\in\Delta$. Так как в графе отсутствуют 3-лапы, каждая из $k-2$ вершин, отличных от P, Q и смежных с X, является смежной либо с P, либо с Q. Следовательно, $k-2=m+2(\lambda-m)=2\lambda-m$ и, тем самым, $m=\mu-2$, т. е.

(10) *Для каждой вершины $X\in\Delta$ существует единственная $\bar{X}\in\Delta$, такая что X несмежна с $\bar{X}$. В частности μ четно.*

Теперь выберем $X\in\Delta$ и $A\in\Lambda$. Так как нет 3-лап, A должна быть смежна с одной из вершин X, $\bar{X}$. С другой стороны P, Q и $\mu-2$ вершин, отличных от X, $\bar{X}$ из множества Δ, которые смежны с X и $\bar{X}$, дают в сумме μ вершин, смежных одновременно и с X и с $\bar{X}$. Следовательно,

(11) *Каждая вершина $A\in\Lambda$ смежна ровно с одной из вершин X, $\bar{X}$ для $X\in\Delta$.*

(12) *Если $k>l$, то либо $k=2\mu$, либо $\bar{\lambda}=0$.*

Доказательство. Предположим, что $k>l$ и $\bar{\lambda}\neq 0$. Подграф с множеством вершин $\Gamma(P)\cap\Gamma(Q)$, где P и Q — несмежные вершины, содержит $\bar{\lambda}$ вершин и является регулярным с валентностью $k-2\mu$. Если в $\Gamma(P)\cap\Gamma(Q)$ найдется пара несмежных вершин, то из отсутствия 3-лап следует, что все μ вершин, смежных с каждой из них, находятся в $\Gamma(P)\cap\Gamma(Q)$. Следовательно, в этом случае $\mu<k-2\mu$ или $2\mu<\lambda+2$, что противоречит (4). Это означает, что наш подграф является кликой, т. е. $k-2\mu=\lambda-1$. Так что, в силу (4.7), $\mu(4\lambda-5\mu+10)=k(\lambda-\mu+3)$, отсюда $(\lambda+3-2\mu)(2\lambda+4-3\mu)=0$, т. е. $(\lambda+3-2\mu)(k-2\mu)=0$. Следовательно, $k=2\mu$, в силу (4).

(13) *Если $k=2\mu$, то имеет место случай* (II), (III) *или* (VI).

Доказательство. Если $k=2\mu$, мы имеем $3\mu=2(\lambda+2)$ и из $f=8(\mu+1)/(\mu+4)$ получаем, что $\mu=2$, 4, 8 или 20. В первых двух случаях имеет место соответственно (II) и (III), а в третьем — (VI). В последнем случае параметры следующие $n=63$, $k=40$, $\lambda=28$, $\mu=20$. Невозможность этого случая была показана в доказательстве утверждения (8).

(14) *Если $\bar{\lambda}=0$, то выполнено* (II), (III) *или* (V).

Доказательство. Если $\bar{\lambda}=0$, то подграф на $\Gamma(P)$ является кликой и, следовательно, $\Lambda=\Gamma(Q)-\{P\}$, где Q не смежна с P, и мы можем пользоваться обозначениями из (10) и (11). Для $A\in\Lambda$ множество $\Delta(A)\cap\Delta(P)$ состоит из $\mu/2$ вершин, лежащих в Δ по (10) и $l-2$ вершин, отличных от A в множестве Λ.

Следовательно, $\lambda = l - 2 + \mu/2 = 2\lambda - 3\mu/2 + 3$, откуда $3\mu = 2\lambda + 6$. Тем самым $k = 2\mu - 2$ и

$$f = 8(\mu - 1)/(\mu + 2).$$

Поэтому $\mu = 2$, 4, 6, 10 или 20. В первых трех случаях выполнено (II), (III) и (V) соответственно. Используя (9) и (5.8) для последних двух случаев, мы имеем

n	k	λ	μ	α	t
28	18	12	10	46	$-10/3$
64	42	30	22	325	$-22/3$

что противоречит (5.18).

Для завершения доказательства (6.1) мы покажем, что

(15) *Если* $\bar{\lambda} \neq 0$ *и* $k \leqslant l$, *то выполнено* (II) *или* (III).

Доказательство. В силу (7) и (11), достаточно показать, что $\mu \leqslant 4$. Для заданной вершины Q подграф $\mathcal{G}_0$ на множестве $\Gamma(Q)$ регулярен валентности $k - \mu$. Поскольку $\bar{\lambda} \neq 0$, подграф $\mathcal{G}_0$ не является кликой, а из отсутствия в $\mathcal{G}$ 3-лап следует, что все μ путей длины 2, соединяющих несмежные вершины в $\Gamma(Q)$, содержатся в $\Gamma(Q)$. Пусть P и A — смежные вершины в $\Gamma(Q)$, так что $A \in \Lambda = \Delta(P) \cap \Gamma(Q)$. В силу (11) в точности $\mu/2$ из λ вершин, смежных и с P и с A, лежат в множестве $\Delta = \Delta(P) \cap \Delta(Q)$, поэтому ровно $\lambda - \mu/2$ таких вершин лежат в $\Gamma(Q)$. Это означает, что $\mathcal{G}_0$ является сильно регулярным графом с параметрами $k_0 = k - \mu$, $l_0 = \bar{\lambda}$, $\lambda_0 = \lambda - \mu/2$ и $\mu_0 = \mu$. Так как $k_0 = 2\lambda_0 - \mu_0 + 4$, -2 — это собственное значение $\mathcal{G}_0$, и, в силу (3), кратность другого собственного значения, отличного от k_0, равна

$$f_0 = 8(\lambda - \mu + 2)(\lambda - \mu + 3)/(2\lambda - 3\mu + 8).$$

Предположим, что $\mu > 4$ и пусть $\mu = 2t + 4$, $t > 0$. Тогда

$$f_0 = 2(\lambda - 2t - 2)(\lambda - 2t - 1)/(t + 2)(\lambda - 3t - 2),$$

откуда мы получаем, что $(\lambda - 3t - 2 \mid t(t+1))$ либо $2t(t+1)$, в зависимости от того, четно или нечетно t. С другой стороны, $f = 2(\lambda - t)(\lambda - t + 1)/(t + 2)(\lambda - 2t)$, поэтому $\lambda - 2t \mid t(t+1)$ или $2t(t+1)$ в зависимости от того, четно или нечетно t. Так как $(\lambda - 2t,\ \lambda - 3t - 2) \mid t + 2$ и $(t + 2,\ t(t+1)) \mid 2$, получаем, что в любом случае $(\lambda - 2t)(\lambda - 3t - 2) \mid 2t(t+1)$.

С другой стороны, поскольку $k \leqslant l$, мы имеем $2\mu \leqslant \lambda + 3$, в силу (4), т. е. $\lambda \geqslant 4t + 5$, — противоречие. Этим завершается доказательство (15), а следовательно и (6.1).

(6.2) *Пусть* $\mathcal{G}$ *— сильно регулярный граф с параметрами, такими же как в случаях* (II) *или* (III) *утверждения* (6.1), *где* $m \geqslant 2$ *или* $m \geqslant 3$, *соответственно. Тогда следующие условия эквивалентны*

(а) $\mathcal{G}$ *удовлетворяет* 4*-вершинному условию.*

(б) $\mathcal{G}$ *не содержит* 3*-лап.*

(в) $\mathcal{G}$ *изоморфен решетчатому графу* $\mathcal{L}_2(m)$ *порядка* m *для некоторого* $m \geqslant 2$, *либо треугольному графу* $\mathcal{T}(m)$ *для* $m \geqslant 3$.

Доказательство. (а)$\Rightarrow$(б). В силу утверждения (8) возможность для существования 3-лап — это случай (II) при $m = 4$ и (III) при $m = 8$. В первом из этих случаев с учетом (5.21) мы находим $\alpha = 1$, поэтому из (5.15) следует отсутствие 3-клик. Во втором случае мы находим, что $\alpha = 0$, 5, 10 или 15. В силу (5.23), $\alpha \neq 15$, а из (5.8) мы получаем $t = -12$ или $-16/3$ в случаях $\alpha = 0$ и 5, соответственно. Следовательно, $\alpha = 10$, в силу (5.18), и из (5.15) вытекает, что 3-лап нет.

(в)$\Rightarrow$(а) вытекает из того, что графы $\mathcal{L}_2(m)$ и $\mathcal{T}(m)$ допускают группы автоморфизмов ранга 3.

(б)$\Rightarrow$(в). Предполагая несуществование 3-лап, мы покажем, что каждая клика из множества Σ клик, состоящих из $k/2+1$ вершин, имеет те же свойства, что (I*) каждая вершина лежит ровно в двух членах множества Σ, и (II*) каждое ребро лежит ровно в одном члене из Σ. Для заданной вершины P пусть Q — вершина, несмежная с P. Положим $\Delta = \Delta(P) \cap \Delta(Q)$, $\Lambda = \Delta(P) \cap \Gamma(Q)$ и для $X \in \Delta$ обозначим через $\bar{X}$ единственную вершину в Δ, несмежную с X. Это отвечает обозначениям утверждения (10), которое применимо в силу отсутствия 3-лап. Из отсутствия 3-лап следует, что $\Delta(X) \cap \Lambda$ является кликой, а Λ — объединение двух непересекающихся клик $\Delta(X) \cap \Lambda$ и $\Delta(\bar{X}) \cap \Lambda$, каждая из которых содержит $(k-\mu)/2$ вершин. Для $Y \in \Delta - \{X, \bar{X}\}$ имеем, что $\Delta(Y) \cap \Lambda = \Delta(X) \cap \Lambda$ или $\Delta(\bar{X}) \cap \Lambda$. В противном случае существуют $A \in \Delta(Y) \cap \Delta(X) \cap \Lambda$ и $B \in \Delta(Y) \cap \Delta(\bar{X}) \cap \Lambda$, откуда следует существование трех попарно смежных вершин P, A и Y в множестве $\Delta(X) \cap \Delta(B)$, что противоречит (10), поскольку $\mu \leqslant 4$. Теперь зафиксируем $X \in \Delta$ и положим $K_1 = \{P\} \cup (\Delta(X) \cap \Lambda) \cup \{Y \in \Delta \mid \Delta(Y) \cap \Lambda = \Delta(X) \cap \Lambda\}$, а K_2 — то же, что и K_1, но с $\bar{X}$ вместо X. Тогда K_1 и K_2 — клики, состоящие из $k/2+1$ вершин каждая, причем $K_1 \cup K_2 = \{P\} + \Delta(P)$ и $K_1 \cap K_2 = \{P\}$. Если K — третья клика, отличная от K_1, K_2, включающая P и содержащая, по меньшей мере, $k/2+1$ вершин, то можно предполагать, что $|K \cap K_1| \geqslant k/4+1$ и $K \cap K_2$ содержит вершину A, отличную от P. Тогда $\Delta(P) \cap \Delta(A)$ содержит по меньшей мере $k/4$ вершин, отличных от P, которые лежат в $K \cap K_1$ и все $k/2-1$ вершин, отличных от P, A в K_2, так что $\lambda \geqslant 3k/4$, что невозможно в случаях (II) или (III).

Поэтому в множестве Σ всех клик, содержащих $k/2+1$ вершин, каждая, любая клика обладает свойствами (I*) и (II*). Следовательно, по теореме (6.2) из статьи Боуза [1] $\mathcal{G}$ является графом частичной геометрии с параметрами $(2, m, 1)$ или $(2, m-1, 2)$ в соответствии с тем, имеет место случай (II) или (III). Известно (и легко проверяется), что частичные геометрии с такими параметрами определяются однозначно.

ЛИТЕРАТУРА

1. Bose R. C. Strongly regular graphs, partial geometries and partially balanced designs. Pacific J. Math. 13 (1963), 389—419. MR 28 # 1137.
2. Chang L. C. The uniqueness and nonuniqueness of the triangular association scheme. Sci. Record 3 (1959), 604—613. MR 22 # 7950.
3. Chang L. C. Association schemes of partially balanced block designs with parameters $v = 28$, $n_1 = 12$, $n_2 = 15$ and $p_{11}^2 = 4$. Sci. Record 4 (1960), 12—18. MR 22 # 7951.
4. Higman D. G. Finite permutation groups of rank 3. Math. Z. 86 (1964), 145—156. MR 32 # 4182.
5. Higman D. G. Intersection matrices for finite permutation groups. J. Algebra 6 (1967), 22—42. MR 35 # 244.
6. Higman D. G. Coherent configurations. I. Rend. Sem. Mat. Univ. Padova 44 (1970), 1—26.
7. Higman D. G. Solvability of a class of rank 3 permutation groups. Nagoya Math. J. 41 (1970).
8. Hoffman A. J. On the uniqueness of the triangular association scheme. Ann. Math. Statist. 31 (1960), 492—497. MR 22 # 7949.
9. Hoffman A. J. and Singleton R. R. On Moore graphs with diameters 2 and 3. IBM J. Res. Develop. 4 (1960), 497—504. MR 25 # 3857.
10. Ray-Chaudhuri D. K. Characterization of line graphs. J. Combinatorial Theory 3 (1967), 201—214. MR 35 # 4119.
11. Seidel J. J. Strongly regular graphs of L_2-type and of triangular type. Nederl. Akad. Wetensch. Proc. Ser. A 70 = Indag. Math. 29 (1967), 188—196. MR 35 # 88.
12. Seidel J. J. Strongly regular graphs with (—1, 1, 0) adjacency matrix having eigenvalue 3. Linear Algebra Appl. 1 (1968), 281—298. MR 38 # 3175.
13. Shrikhande S. S. The uniqueness of the L_2 association scheme. Ann. Math. Statist. 30 (1959), 781—798.
14. Sims C. C. On graphs with rank 3 automorphism groups. J. Combinatorial Theory (to appear).
15. Wielandt H. Finite permutation groups. Lectures. University of Tübingen, 1954/55; English transl., Academic Press. New York. 1964. MR 32 # 1252.

Двоичные дизъюнктивные коды[1])

В. Кауц, Р. Сингелтон

Двоичный дизъюнктивный код состоит из множества кодовых слов, обладающих тем свойством, что их поразрядные булевы суммы ($1+1=1$) удовлетворяют предписанному уровню различимости. Такие коды находят применение главным образом при представлении атрибутов документов в информационно-поисковых системах, а также при распределении каналов для облегчения разгрузки сильно загруженных линий связи. В работе устанавливаются некоторые основные свойства кодов такого типа и выводятся формулы и границы, связывающие их основные параметры. Наконец, здесь описано несколько семейств кодов, основанных на (1) обыкновенных q-ичных кодах, исправляющих ошибки, (2) комбинаторных конфигурациях вроде блок-схем и латинских квадратов, (3) графических конструкциях, и (4) проверочных матрицах стандартных двоичных кодов, исправляющих ошибки.

I. ВВЕДЕНИЕ

Опишем две задачи из области теории кодирования, связанные с представлением и обработкой данных в информационно-поисковых системах некоторых типов. Будем называть *дизъюнкцией* двух n-разрядных двоичных слов их поразрядную булеву сумму; например, $011001 \vee 010010 = 011011$. Требуется найти код, состоящий из достаточно большого числа N кодовых слов, такой чтобы для заданного натурального m всевозможные дизъюнкции, составленные из не более, чем m различных кодовых слов, были различны (задача 1), или не покрывали кодовых слов, отличных от тех, из которых составлена данная дизъюнкция (задача 2). Вскоре будет установлено, что эти две задачи внутренне взаимосвязаны. По этой причине они рассматриваются в данной статье совместно.

Коды, удовлетворяющие условию задачи 1, будем называть *кодами с однозначным декодированием порядка m* (uniquely

[1]) Kautz W. H., Singleton R. C.; Nonrandom binary superimposed codes, IEEE Trans. on Info. Theory, v. IT-10, No 4 (1964), 363—377.

decipherable) сокращенно UD_m-кодами. Название этих кодов непосредственно связано с их определением, которое гарантирует, что любое слово, представляющее собой дизъюнкцию не более, чем m кодовых слов UD_m-кода, может быть разложено в дизъюнкцию слов этого кода лишь единственным образом. Например, рассмотрим следующий список из восьми 7-разрядных кодовых слов

1	1	0	0	0	0	0
1	0	1	0	0	0	0
0	1	0	0	1	0	0
0	0	1	1	0	0	0
0	0	0	1	1	0	0
0	0	1	0	0	1	0
0	0	0	0	1	0	1
0	0	0	0	0	1	1

Если этот список, не содержащий повторяющихся слов, пополнить всеми $\binom{8}{2}=28$ попарными дизъюнкциями входящих в него слов, то полученный список также не будет содержать повторений. (Этот факт легко проверить, выписывая всевозможные попарные дизъюнкции; или, что проще, проверяя по отдельности, каким образом могут возникать дизъюнкции, содержащие три или четыре единичных разряда.) Таким образом, указанное множество из восьми кодовых слов составляет UD_2-код.

Код, слова которого удовлетворяют условию задачи 2, будем называть *кодом с отсутствием ложных соответствий* (zero-false-drop) порядка m, сокращенно ZFD_m-кодом. Это название связано с применениями таких кодов в задачах поиска информации, о которых будет рассказано в следующем разделе.

Следующий код из трех 3-разрядных слов, содержащих по одному единичному разряду, а именно 100, 010, 001, является, очевидно, ZFD_2-кодом, ибо ни одна из попарных дизъюнкций его слов (скажем, 110), не покрывает оставшегося слова (001). Более того, в некотором роде тривиальным образом этот код является ZFD_3-кодом. Отметим также, что он одновременно является UD_2- и UD_3-кодом.

В разд. II описаны причины, в силу которых возникла потребность в дизъюнктивных кодах и некоторые их применения, которые могут быть допущены читателем, интересующимся лишь самими кодами как таковыми. Основные свойства этих кодов и границы их мощности N в терминах порядка m и длины кодовых слов n излагаются в разд. III и IV. Некоторые семейства кодов произвольно большой мощности и порядка описываются в разд. III—VII.

II. ПРИМЕНЕНИЯ

A. Поисковые массивы

Дизъюнктивный код, такой, как *ZFD*-код, можно использовать для поиска в информационном массиве [1]—[3]. До стадии кодирования, поисковый массив состоит из длинного списка заголовков, по одному на каждый документ, входящий в информационный массив. Каждый заголовок содержит идентифицирующий номер (адрес) документа (для последующего физического поиска), и короткий список атрибутов, называемых *дескрипторами*, которые выбираются из словаря дескрипторов с целью описания содержания рассматриваемого документа. Типичный словарь может содержать число N дескрипторов в пределах от 10^3 до 10^4, а наибольшее число дескрипторов, приходящееся на один документ данного массива, колеблется от 5 до 15. Размеры самого массива по существу неограниченны.

Запрос к такому массиву имеет вид предписания, содержащего список запрашиваемых дескрипторов и указания, требующего установить, либо (а) *наличие*, либо (б) *список* тех документов информационного массива, которые содержат в своих заголовках все дескрипторы из запрашиваемого списка. Таким образом, для автоматизации процесса поиска требуется, чтобы все документированные данные были закодированы таким образом, чтобы описанная проверка осуществлялась быстро и с использованием минимума оборудования.

Известны методы, которые позволяют эффективно кодировать адреса документов и определять соответствие документов имеющемуся запросу (шаг (б)) [4], если имеются средства для того, чтобы определять лишь наличие или отсутствие запрашиваемых документов (шаг (а)). Для этой цели предлагается использовать ZFD_m-коды, при помощи которых можно кодировать дескрипторные части заголовков документов, составляющих поисковый массив.

Пусть каждому из N дескрипторов, входящих в словарь, сопоставлено одно из n-разрядных кодовых слов некоторого ZFD_m-кода. Списку дескрипторов, связанному с каждым документом, сопоставляется новое n-разрядное двоичное слово, представляющее собой поразрядную булеву сумму кодовых слов, соответствующих всем дескрипторам, входящим в рассматриваемый список. Из кодовых слов, соответствующих запрашиваемым дескрипторам, аналогичным образом формулируется «запросное» слово. Непосредственно из определения ZFD_m-кодов следует, что коль скоро с каждым документом связано не более m дескрипторов, *запросное слово покрывается каким-либо документным словом тогда и только тогда, когда*

все запрашиваемые дескрипторы содержатся среди дескрипторов, связанных с этим документом. Если при запросе описанная проверка окажется успешной хотя бы для одного из документных слов массива, то можно извлечь из массива нужную информацию; в противном случае нужная информация в массиве отсутствует.

Известны различные электрические и механические устройства, в которых реализуется описанный тип поискового массива [5—7]; некоторые из подобных устройств имеются в продаже. Например, при использовании карт с краевой перфорацией каждый документ представлен картой, несущей на себе двоичное слово, нанесенное в виде набора вырезов, сделанных на местах n возможных позиций, располагающихся на одном или нескольких краях карты — скажем, на нижнем крае. Чтобы осуществить запрос, колоду таких карт помещают на множество небольших стерженьков, размещенных под колодой; часть стерженьков приподнимают в соответствии с позициями вырезов, соответствующих единицам запрашиваемого кодового слова. Все те карты, у которых имеются вырезы во всех указанных позициях, будут неподвижны, в то время как ненужные карты окажутся приподнятыми, и могут быть отделены от искомых.

Коды, используемые в настоящее время в подобных поисковых массивах порождаются путем случайного отбора [1—3]. Каждое дескрипторное кодовое слово формируется путем случайного размещения небольшого числа единиц (обычно трех-четырех) в n-разрядном двоичном блоке. Подходящее значение разрядности n случайного дизъюнктивного кода можно определить путем статистического анализа, сводящего к предусмотренному минимуму вероятность ошибочного извлечения не относящегося к делу документа [8—12]. Подобное «ложное соответствие» может возникнуть, если рассматриваемая дизъюнкция кодовых слов покрывает какие-либо кодовые слова, отличные от тех, из которых она составлена.

Хотя пользователь массива легко может справиться с небольшим числом ложных соответствий, они, тем не менее, составляют предмет беспокойства и вряд ли могут считаться переносимыми, когда их число становится чересчур большим. Поскольку даже самый лучший статистический анализ случайных дизъюнктивных кодов опирается на ряд упрощающих предположений (равное использование дескрипторов, неограниченный размер словаря, независимый выбор дескрипторов), невозможно гарантировать желаемый минимальный уровень вероятности ложных соответствий, без использования весьма консервативных конструкций. Однако даже тогда случайный код всегда будет давать отклонения от средних значений. Так, всегда можно подозревать, что в конкретном новом коде имеется

небольшое число плохих комбинаций кодовых слов, и всегда есть шанс, что новый код будет иметь плохие рабочие характеристики. Наконец, еще один недостаток случайных кодов состоит в том, что поиск, пренебрегающий одним или несколькими дескрипторами, не может быть осуществлен без риска «ложных промахов» — т. е., когда желаемые вхождения отвергаются. *ZFD*-коды свободны от этих недостатков.

Новое семейство дизъюнктивных кодов было введено и изучено в первую очередь с целью преодоления указанных недостатков. Точно так, как в случае обычных кодов, исправляющих ошибки, они позволяют полностью гарантировать отсутствие ошибок вплоть до некоторого уровня активности. Аналогичным образом, случайные дизъюнктивные коды соответствуют обычным случайным кодам, вроде тех, которые были изучены Шенноном [13] и Элайесом [14].

В отношении случайно порожденных дизъюнктивных кодов верно также то, что как только дизъюнкция кодовых слов сформирована — в соответствии с кодируемым документом, по этой дизъюнкции, вообще говоря, уже невозможно непосредственно определить, какие именно дескрипторы в нее вошли. Иначе говоря, декодирование слов вообще говоря, не является однозначным. С другой стороны, как будет показано в следующем разделе, любой ZFD_m-код автоматически является UD_m-кодом, так что дизъюнкции кодовых слов кодов нового семейства автоматически обладают свойством однозначности декодирования.

Б. Передача данных

Некоторые загруженные линии связи, такие, как любительский диапазон, общие телефонные линии и некоторые военные радиодиапазоны, характеризуются ограниченным числом n каналов при большем числе N пользователей низкого уровня. В таких случаях невозможно отдать каждому пользователю по каналу на все времена, и следует ввести в действие какую-либо процедуру, позволяющую варьировать распределение каналов по мере требования. Обычная практика состоит в привлечении главного блока управления, переключающего центра, или «оператора», для хранения информации о доступных каналах и распределения их по мере необходимости. В любительских диапазонах централизованным управлением пренебрегают в пользу менее надежной практики, позволяя каждому пользователю самому искать себе свободный канал наилучшим доступным ему способом.

Если можно быть уверенным, что не более m пользователей будут одновременно нуждаться в доступе к линии, то каждому пользователю можно будет навсегда приписать *подмножество*

каналов, которыми ему разрешено пользоваться (передавать и/или слушать одновременно).

Если такое приписывание осуществить в соответствии с некоторым ZFD_m-кодом, то пользователь может быть уверен, что указанное ему множество каналов ни в какой момент не будет *полностью* занято никаким другим пользователем, или группой пользователей. Таким образом, он сможет осуществлять связь в любой момент, не запрашивая главный блок управления и, подчиняясь лишь существующему ограничению на наибольшее число m одновременно работающих пользователей. Если для какой-либо конкретной системы такой предел не установлен статистикой ее использования, разумно позаботиться о наличии простейшего блока управления, который только сообщает всем пользователям о достижении предела заполнения линии.

Использование широкополосных методов для облегчения тесноты в загруженных линиях связи обосновывалось Костасом [15]. Предложенное применение ZFD_m-кодов может дать нужные средства, в то время как практичность широкополосной философии еще нуждается в проверке [1])[2]).

III. ТЕОРЕТИЧЕСКИЕ РЕЗУЛЬТАТЫ

В этом разделе дается математическое определение ZFD_m- и UD_m-кодов и устанавливается взаимосвязь между этими двумя понятиями.

Дизъюнкция $z = x \vee y$ двух двоичных векторов

$$x = (x^1, x^2, \ldots, x^n) \quad \text{и} \quad y = (y^1, y^2, \ldots, y^n)$$

размерности n определяется следующим образом:

$$z^i = \begin{cases} 0, & \text{если } x^i = y^i = 0, \\ 1 & \text{в противном случае.} \end{cases}$$

$i = 1, \ldots, n$ (до сих пор мы называли ее поразрядной дизъюнкцией). Будем говорить, что вектор x *покрывается* вектором y, если $x \vee y = y$.

Пусть дан код C_1, представляющий собой совокупность N двоичных векторов размерности n; эти векторы носят название кодовых слов. Для каждого $k = 2, 3, \ldots, N$ построим k-е *множество дизъюнкций* C_k, образуя всевозможные дизъюнкции ко-

[1]) Другое применение двоичных дизъюнктивных кодов к проблемам связи было предложено Коном и Гормэном [16], и касается использования предложенного семейства кодов, обладающих ограниченными дизъюнктивными свойствами для селективного обращения к станциям в сети связи.

[2]) Последний подраздел В разд. II под названием «магнитные запоминающие устройства», почти не содержащий сведений о дизъюнктивных кодах, при переводе опущен. — *Прим. перев.*

довых слов из C_1, каждая из которых составляется в точности из k различных кодовых слов. Таким образом, множество C_k содержит $\binom{N}{k}$ векторов, причем при $k>1$ эти векторы не обязательно различны. При рассмотрении последовательности множеств $C_1, C_2, \ldots, C_k, \ldots$ нас интересует, в частности, то значение k, при котором впервые появляются повторения векторов, встречающиеся или в самом множестве C^k, или же в C_k и каком-либо из предшествующих множеств. Для этой цели устанавливается следующая теорема и следствие из нее.

Теорема 1. Если множества $C_1, C_2, \ldots, C_{m+1}$ *попарно не пересекаются, то множество* C_m *содержит в точности* $\binom{N}{m}$ *различных векторов.*

Доказательство. Допустим, что два из $\binom{N}{m}$ векторов, входящих в C_m, одинаковы:

$$x_1 \vee x_2 \vee \ldots \vee x_m = y_1 \vee y_2 \vee \ldots \vee y_m,$$

где $x_1, x_2, \ldots, x_m$ и $y_1, y_2, \ldots, y_m$ — кодовые слова из C_1. Тогда

$$y_j \vee x_1 \vee x_2 \vee \ldots \vee x_m = x_1 \vee x_2 \vee \ldots \vee x_m$$

для каждого $j=1, 2, \ldots, m$. Но поскольку пересечение множеств C_{m+1} и C_m пусто, каждое из кодовых слов $y_1, \ldots, y_m$ должно принадлежать множеству кодовых слов $\{x_1, \ldots, x_m\}$. Таким образом, множество C_m не содержит повторений, и все $\binom{N}{m}$ входящих в него векторов различны.

Следствие. *Если множества* $C_1, C_2, \ldots, C_{m+1}$ *попарно не пересекаются, то каждое из множеств* C_k *содержит в точности* $\binom{N}{k}$ *различных векторов,* $k=1, \ldots, m$.

Эта теорема и следствие из нее используются далее для установления связи между *ZFD*- и *UD*-кодами.

Если попарно не пересекаются лишь множества $C_1, C_2, \ldots, C_m$, то C_m не обязательно содержит $\binom{N}{m}$ элементов. Например, для кода C_1, состоящего из семи циклических сдвигов вектора (1101000), множества C_1, C_2 и C_3 попарно не пересекаются, однако C_3 содержит лишь восемь элементов, в то время как $\binom{7}{3}=35$.

Далее, если $C_1, C_2, \ldots, C_m$ попарно не пересекаются, и при этом C_m содержит $\binom{N}{m}$ элементов, то множества $C_1, C_2, \ldots, C_{m+1}$ не обязательно являются попарно непересекающимися. Например, для кода C_1, состоящего из векторов $a=(1100)$, $b=(0011)$ и $c=(0110)$, имеем для C_2 дизъюнкции $a \vee b =$

$=(1111)$, $a \vee c = (1110)$, и $b \vee c = (0111)$, а для C_3 единственный элемент $a \vee b \vee c = (1111)$; множества C_1 и C_2 не пересекаются, C_2 содержит $\binom{3}{2} = 3$ элемента, а пересечение C_2 и C_3 непусто.

ZFD_m-код определяется как такое множество C_1 кодовых слов, для которого никакая дизъюнкция $y_1 \vee y_2 \vee \ldots \vee y_j$, составленная из $j \leqslant m$ кодовых слов, не покрывается никакой другой дизъюнкцией $x_1 \vee x_2 \ldots \vee x_k$, составленной из $k \leqslant m$ кодовых слов, если только все $y_1, y_2, \ldots, y_j$ не принадлежат множеству кодовых слов $x_1, \ldots, x_k$. Очевидно, что всякий ZFD_m-код является одновременно ZFD_k-кодом для любого k из промежутка $1 \leqslant k < m$. Эквивалентное, но несколько менее интуитивно ясное определение извлекается из следующей теоремы.

Теорема 2. *Множество двоичных векторов является ZFD_m-кодом тогда и только тогда, когда никакая дизъюнкция $x_1 \vee x_2 \vee \ldots \vee x_k$, составленная из $k \leqslant m$ кодовых слов, не покрывает иных кодовых слов, кроме тех, из которых она составлена.*

Доказательство. Достаточность следует прямо из определения. С другой стороны, если дизъюнкция $x_1 \vee x_2 \vee \ldots \vee x_k$, составленная из $k \leqslant m$ кодовых слов, не покрывает иных кодовых слов, кроме $x_1, \ldots, x_k$, то она не может покрывать и дизъюнкций, в состав которых входят кодовые слова, отличные от $x_1, \ldots, x_k$.

В терминах последовательности множеств $C_1, C_2, \ldots, C_k, \ldots$ имеем следующую теорему.

Теорема 3. *Множество C_1 является ZFD_m-кодом тогда и только тогда, когда множества $C_1, C_2, \ldots, C_{m+1}$ попарно не пересекаются.*

Доказательство. 1) Пусть множества $C_1, C_2, \ldots, C_{m+1}$ попарно не пересекаются, тогда кодовое слово y_1 может покрываться дизъюнкцией $x_1 \vee \ldots \vee x_k$, $k \leqslant m$, лишь в том случае, когда y_1 совпадает с одним из слов $x_1, \ldots, x_k$; стало быть, C_1 есть ZFD_m-код.

2) Пусть C_1 есть ZFD_m-код, и пусть при этом C_j и C_k для $1 \leqslant j < k \leqslant m+1$ имеют общий элемент $x_1 \vee x_2 \vee \ldots \vee x_j = y_1 \vee y_2 \vee \ldots \vee y_k$. Тогда каждый вектор y_i совпадает с одним из векторов $x_1, \ldots, x_j$, а поскольку и те и другие попарно различны, неравенство $j < k$ невозможно; стало быть, $C_1, C_2, \ldots, C_{m+1}$ попарно не пересекаются.

UD_m-код определяется как множество C_1 кодовых слов, такое, что равенство любых двух дизъюнкций, каждая из которых

составлена из не более, чем m кодовых слов, влечет за собой совпадение множеств кодовых слов, из которых эти дизъюнкции составлены. Отсюда следует теорема 4.

Теорема 4. *Множество C_1 является UD_m-кодом тогда и только тогда, когда множества $C_1, C_2, \ldots, C_m$ попарно не пересекаются, и при этом C_m содержит $\binom{N}{m}$ различных векторов.*

Доказательство. 1) Предположим, что множества $C_1, \ldots, C_m$ попарно не пересекаются, и что C_m содержит $\binom{N}{m}$ различных элементов. Тогда каждое из множеств C_k, $1 \leqslant k \leqslant m$, содержит $\binom{N}{k}$ различных элементов.

Стало быть, две дизъюнкции, каждая из которых составлена из не более, чем m кодовых слов, не могут оказаться равными, если соответствующие множества слов не совпадают, без того, чтобы возникло противоречие или с тем, что множества $C_1, \ldots, C_m$ попарно не пересекаются, или с тем, что для любого $1 \leqslant k \leqslant m$ множество C_k содержит $\binom{N}{k}$ различных элементов.

2) Предположим, что C_1 есть UD_m-код. Поскольку равенство двух дизъюнкций, составленных из не более, чем m кодовых слов, влечет совпадение двух соответствующих множеств кодовых слов, множества $C_1, C_2, \ldots, C_m$ попарно не пересекаются и C_m содержит $\binom{N}{m}$ различных элементов. Существование взаимной связи между ZFD- и UD-кодами вытекает непосредственно из теорем 3 и 4, и может быть изображено следующим образом:

$$ZFD_m \Rightarrow UD_m \Rightarrow ZFD_{m-1} \Rightarrow UD_{m-1} \Rightarrow \ldots \Rightarrow ZFD_1 \Rightarrow UD_1.$$

Кроме того, как показывают контрпримеры изложенные выше в терминах множеств C_k, обратные импликации, вообще говоря, места не имеют:

$$ZFD_m \not\Rightarrow UD_m \not\Rightarrow ZFD_m \text{ и т. д.}$$

Рассмотрим матрицу A размеров $N \times n$, в строках которой расположены кодовые слова, составляющие множество C_1.

Справедлива следующая теорема.

Теорема 5. *Множество C_1 является ZFD_m-кодом тогда и только тогда, когда для любых $m+1$ строк матрицы A можно указать $m+1$ столбцов так, чтобы на их пересечении стояла единичная подматрица.*

Доказательство. Свойство множества C_1 быть ZFD_m-кодом равносильно требованию, чтобы для любых $m+1$ строк матрицы A ни одна из этих строк не покрывалась дизъюнкцией m остальных. Так оно и будет, если, и только если, для каждой

из строк этой $(m+1)$-строчной подматрицы найдется столбец, на пересечении с которым эта строка содержит единицу, а остальные строки — нули. Обратно, если любые $m+1$ строк содержат единичную подматрицу порядка $m+1$, то ни одна из этих строк не может быть покрыта дизъюнкцией остальных; стало быть, C_1 является ZFD_m-кодом.

IV. ГРАНИЦЫ

Слабая верхняя граница для мощности N n-разрядного UD_m-кода может быть получена путем простого подсчета общего числа различных векторов, входящих в множества $C_1, \ldots, C_m$, и указания на то, что это число не может превосходить числа всех ненулевых n-разрядных двоичных наборов:

$$\sum_{k=1}^{m} \binom{N}{k} \leqslant 2^n - 1. \tag{1}$$

Лучшие границы получаются при использовании некоторых промежуточных параметров. Число единиц, входящих в кодовое слово x_i называется *весом* w_i этого кодового слова; *перекрытием* λ_{ij} двух кодовых слов x_i и x_j называется их скалярное произведение, т. е. число общих единичных разрядов. Удобно обозначать через $w_{\min} = \min\limits_{i} w_i$ наименьший вес, а через $\lambda_{\max} = \max\limits_{i,\ j:\ i \neq j} \lambda_{ij}$ — наибольшее перекрытие кодовых слов, где минимум и максимум берутся по всем N словам рассматриваемого кода.

Пусть рассматриваемый код имеет наибольшее перекрытие $\lambda_{\max}$. Если фиксировать любые $\lambda_{\max}+1$ разрядов, то самое большее одно кодовое слово может содержать единицы во всех указанных позициях. Общее число возможностей, при выборе указанных разрядов из общего числа n есть в точности $\binom{n}{\lambda_{\max}+1}$, а i-му кодовому слову соответствует в точности $\binom{w_i}{\lambda_{\max}+1}$ возможностей. Таким образом, суммируя по всем N кодовым словам, имеем

$$\sum_{i=1}^{N} \binom{w_i}{\lambda_{\max}+1} \leqslant \binom{n}{\lambda_{\max}+1}. \tag{2}$$

Если веса всех кодовых слов одинаковы и равны w, эта граница принимает вид

$$N \leqslant \binom{n}{\lambda_{\max}+1} \Big/ \binom{w}{\lambda_{\max}+1}. \tag{3}$$

Далее, если $w_i \geqslant m\lambda_{\max} + 1$, то i-е кодовое слово не может быть покрыто дизъюнкцией каких-либо m из остальных кодовых слов, поскольку оно перекрывается с каждым из остальных кодовых слов не более, чем по $\lambda_{\max}$ позициям. Поэтому любой код, имеющий наименьший вес $w_{\min}$ и наибольшее перекрытие $\lambda_{\max}$, заведомо является ZFD_m-кодом для всех значений m вплоть до некоторого, удовлетворяющего границе

$$m \geqslant [(w_{\min} - 1)/\lambda_{\max}], \tag{4}$$

где квадратные скобки обозначают целую часть числа.

Рассматривая матрицу A можно легко заметить, что величины λ_{ij} являются внедиагональными элементами $N \times N$-матрицы

$$\Lambda = AA^t,$$

в то время как величины w_i стоят на главной диагонали этой матрицы: $w_i = \lambda_{ii}$. Поэтому отыскание n-разрядного ZFD_m-кода C_1 мощности N, для которого нижняя граница (4) для наибольшего порядка m оказывается максимальной, равносильно отысканию матрицы A размеров $N \times n$, для которой в матрице Λ оказывается максимальным отношение наименьшего диагонального элемента (уменьшенного на единицу) к наибольшему внедиагональному элементу.

Следующая теорема дает условие, при выполнении которого порядок m ZFD_m-кода достигает границы (4).

Теорема 6. *Если при любом выборе $\lambda_{\max}$ разрядов в коде имеется не менее двух кодовых слов, содержащих во всех этих разрядах единицы, то этот код является ZFD_m-кодом, и не является ZFD_{m+1}-кодом, где*

$$m = \left[\frac{w_{\min} - 1}{\lambda_{\max}}\right].$$

Доказательство. В силу границы (4) рассматриваемый код является по меньшей мере ZFD_m-кодом. С другой стороны, так как каждый набор из $\lambda_{\max}$ единиц встречается не менее, чем в двух кодовых словах, то любое кодовое слово имеющее вес $w_i \leqslant (m+1)\lambda_{\max}$ можно покрыть дизъюнкцией некоторых $m+1$ кодовых слов из числа оставшихся. Поэтому рассматриваемый код не может быть ZFD_{m+1}-кодом.

Если некоторый код является ZFD_m-кодом при значениях m, превосходящих тот минимум, который обеспечивает граница (4), многочисленные возможности перекрытия все же исключаются при наличии кодовых слов, имеющих вес, меньший $m\lambda_{\max} + 1$. Следующая теорема показывает это для случая слов, имеющих вес, не превосходящий m,

Теорема 7. *Если вес какого-либо слова ZFD_m-кода не превосходит m, то в этом слове есть такая позиция, занятая единицей, в которой все остальные кодовые слова содержат нуль.*

Доказательство. В противном случае указанное кодовое слово оказалось бы покрытым дизъюнкцией некоторых m кодовых слов из числа остальных, и код не был бы ZFD_m-кодом.

Отсюда видно, что если все кодовые слова некоторого ZFD_m-кода имеют веса $w_i \leqslant m$, то $N \leqslant n$, т. е. в этом случае мощность кода не превосходит длины кодового слова. Равенство $N = n$ достигается лишь в том случае, когда все кодовые слова имеют вес 1.[1])

Если лишь *часть* кодовых слов ZFD_m-кода имеет веса, не превосходящие m — скажем, $w_i \leqslant m$ для $i = 1, 2, \ldots, N_1$, то число N кодовых слов удовлетворяет границе

$$\sum_{i=1}^{N_1} \binom{w_i - 1}{\lambda_{\max} + 1} + \sum_{i=N_1+1}^{N} \binom{w_i}{\lambda_{\max} + 1} \leqslant \binom{n - N_1}{\lambda_{\max} + 1},$$

представляющей собой улучшенный вариант границы (2).

Здесь принят во внимание тот факт, что по меньшей мере N_1 из n разрядов используются в рассматриваемом коде лишь по одному разу. На самом деле, если вес какого-либо кодового слова не превосходит m, то можно уменьшить этот вес до единицы без уменьшения порядка m самого кода, ибо в таком кодовом слове есть позиция, занятая единицей, в которой все остальные слова содержат нуль. Аналогично, если вес какого-либо кодового слова превышает $m\lambda_{\max} + 1$, его можно уменьшить до указанной величины с сохранением порядка m рассматриваемого кода, вычеркнув произвольным образом нужное число единиц. Если в результате такого вычеркивания значение $\lambda_{\max}$ уменьшится, описанный процесс можно повторить. Поэтому, если дан какой-либо ZFD_m-код, то может быть построен другой, возможно отличный от исходного, ZFD_m-код с теми же значениями параметров n, N и m, и у которого вес w_i любого кодового слова или равен 1, или удовлетворяет неравенствам $m + 1 \leqslant w_i \leqslant m\lambda_{\max} + 1$.

Ясно, что при одновременном исключении из ZFD_m-кода любого кодового слова веса 1 вместе с соответствующим ему разрядом значения параметров n и N, отвечающие рассматриваемому коду, уменьшается на 1 при сохранении его порядка m. Аналогичным образом любой ZFD_m-код может быть пополнен любым числом кодовых слов веса 1 при одновременном увели-

[1]) Вместе с тем, как будет показано в разд. VI, UD_m-коды веса $w = m$ и мощности $N > n$ существуют.

чении n и N на одну и ту же величину и сохранении порядка m рассматриваемого кода. Описанный процесс «линейного» уменьшения или увеличения может быть полезен при построении кодов заданной мощности с использованием других известных кодов; в то же время его неэффективность указывает на то, что при поиске более совершенных кодов следует исключать кодовые слова веса 1, допуская лишь веса из промежутка

$$m+1 \leqslant w_i \leqslant m\lambda_{\max}+1.$$

Если все кодовые слова имеют одинаковый вес w, то полученные выше границы (2) и (4) принимают вид

$$N \leqslant \binom{n}{\lambda_{\max}+1} \Big/ \binom{w}{\lambda_{\max}+1} \tag{5}$$

и

$$m \geqslant [(w-1)/\lambda_{\max}]. \tag{6}$$

Благодаря Джонсону граница (5) допускает некоторые уточнения. В нашей записи она имеет вид

$$N \leqslant \left[\frac{n}{w}\left[\frac{n-1}{w-1}\left[\frac{n-2}{w-2}\left[\cdots\left[\frac{n-\lambda_{\max}}{w-\lambda_{\max}}\right]\cdots\right]\right]\right]\right]; \tag{7}$$

$$N \leqslant \left[\frac{n(w-\lambda_{\max})}{w^2-n\lambda_{\max}}\right], \quad \text{когда} \quad w^2 > n\lambda_{\max}. \tag{8}$$

Далее, меняя ролями нули и единицы, имеем

$$N(n,\ w,\ \lambda_{\max}) = N(n,\ n-w,\ n-2w+\lambda_{\max}).$$

В том случае, когда $\lambda_{\max}=1$, описанный выше процесс редукции весов приводит к получению весов, равных 1 и $\lambda_{\max}+1=m+1$, при отсутствии каких-либо других. «Линейное» исключение слов веса 1 приводит к получению равновесного кода, достигающего нижней границы (6): $m=w-1$. Эти коды детально обсуждаются в разд. V.

Наконец, для равновесного UD_m-кода граница (1) может быть уточнена:

$$\sum_{k=1}^{m}\binom{N}{k} \leqslant \sum_{i=w}^{mw}\binom{n}{i};$$

сумма, стоящая в правой части выражает число всевозможных n-разрядных двоичных векторов, веса которых лежат между w и mw. В общем случае для произвольного UD_m-кода имеем границу

$$\sum_{k=1}^{m}\binom{N}{k} \leqslant \sum_{i=m}^{n}\binom{n}{i};$$

чтобы убедиться в справедливости этого неравенства достаточно сравнить его правую часть с (1), и показать, что оно скорее выполняется при наличии кодовых слов веса $w_i < m$, чем при их отсутствии.

V. ПОСТРОЕНИЕ *ZFD*-КОДОВ

A. Коды, основанные на обыкновенных двоичных кодах, исправляющих ошибки

Наш подход к проблеме построения *ZFD*-кодов состоит в том, чтобы искать среди известных семейств обыкновенных кодов с исправлением ошибок такие, которые или обладают желаемыми свойствами, или могут быть модифицированы для придания им этих свойств. Этот поиск привел к нахождению ряда потенциально полезных семейств кодов произвольного порядка и произвольно большой мощности и длины. Несомненно, дальнейшие исследования приведут к построению еще лучших кодов, поскольку описываемые здесь коды по большей части допускают пополнение посредством дополнительных кодовых слов (величина N возрастает) без уменьшения величин n и m.

При заданных n и m, описанный в разд. IV «линейный» процесс пополнения показывает, что максимальная мощность *ZFD*-кода, $N_{\max}(n, m)$, является строго возрастающей функцией от n, ибо

$$N_{\max}(n, m) \geqslant N_{\max}(n-1, m) + 1.$$

Поэтому можно строить коды любой наперед заданной мощности или длины, отправляясь от ближайших к ним кодов, входящих в состав семейств, описываемых в данном разделе, и имеющих меньшие значения параметров. Аналогичным образом, нужные коды могут быть получены путем вычеркивания разрядов и кодовых слов из больших кодов. Кроме того,

$$N_{\max}(n, m) \geqslant N_{\max}(n, m'),$$

если $m' \geqslant m$.

Список n-разрядных двоичных векторов веса 1 (т. е. код, отвечающий единичной матрице $A = I$ размеров $n \times n$) доставляет тривиальный пример ZFD_m-кода с параметрами $N = n = m$, не допускающего пополнения до кода большей мощности с сохранением длины и порядка. Для этих кодов достигается граница (1); они будут использоваться позднее для построения больших кодов методами каскадной композиции.

Один из обширных классов известных двоичных кодов — двоичные групповые коды [20], не могут быть использованы непосредственно в качестве дизъюнктивных кодов. Действи-

тельно, поскольку такие коды содержат нулевой вектор, они не являются даже ZFD_1-кодами; а лишь UD_1-кодами. Более того, и после выкалывания нулевого вектора они в большинстве случаев не становятся ZFD_1-кодами, — они как правило, одновременно содержат векторы большого веса (наподобие 11 ... 1), и покрываемые ими векторы малых весов. Вместе с тем, если веса всех кодовых слов ZFD_m-кода одинаковы, то величина перекрытия λ_{max} связана с величиной d, обозначающей наименьшее число разрядов, в которых могут различаться два кодовых слова, соотношением

$$d = 2(w - \lambda_{max}).$$

Это позволяет переписать границу (6) в виде

$$m \geqslant [(w - 1)/(w - d/2)].$$

Величина d — это *минимальное расстояние*, характеризующее корректирующие свойства групповых (и вообще любых) двоичных кодов, исправляющих ошибки. Поэтому для нахождения равновесиых ZFD_m-кодов с весом слов w нужно искать обыкновенные равновесные коды с исправлением ошибок, имеющие минимальное расстояние

$$d = \frac{2w(m-1)+2}{m}.$$

Простой способ построения равновесных кодов, исправляющих ошибки, состоит в извлечеиии всех слов требуемого веса w из произвольного кода, исправляющего ошибки. Такая операция заведомо не уменьшает расстояния. В действительности, если минимальное расстояние исходного кода нечетно, то у полученного кода оно возрастет до ближайшего четного числа, поскольку два кодовых слова одинакового веса могут различаться лишь в четном числе разрядов. Рассмотрим пример. Известно, что число слов веса w в коде Хэмминга, исправляющем единичную ошибку ($d = 3$) при длине кодовых слов $n = 2^{\nu} - 1$, где $\nu = 2, 3, 4, \ldots$, совпадает с коэффициентом при x^w в многочлене [21]

$$P(x) = \frac{1}{n+1}\left((1+x)^n + n(1-x)(1-x^2)^{n-1/2}\right) =$$

$$= 1 + \frac{n(n-1)}{6}x^3 + \frac{n(n-1)(n-3)}{24}x^4 + \ldots.$$

Стало быть, имеется $N = n(n-1)/6$ кодовых слов веса $w = 3$, и все они могут быть использованы для построения ZFD_m-кода длины n. На деле минимальное расстояние d рассматриваемой равновесной части указанного кода равно 4, и в силу (10) построенный код имеет по меньшей мере порядок $m = 2$.

К сожалению, групповые коды по большей части не подходят для построения ZFD_m-кодов, так как в силу известного свойства величина минимального расстояния группового кода совпадает с наименьшим весом ненулевых кодовых слов и, стало быть, $d \leqslant w$. Если указанный вес — четный, то (10) дает (при $w > 1$)

$$m \geqslant [(w-1)/(w-w/2)] = [2 - 2/w] = 1,$$

а если он нечетный, то имеем

$$m \geqslant [(w-1)/(w-(w+1)/2)] = 2.$$

Поскольку это единственные известные границы для величины m, можно ожидать, что даваемая ими величина порядка близка к действительной, кроме тех случаев, когда число N слов веса w существенно меньше величины, которую дает граница (3).

Таким образом, для построения равновесных ZFD-кодов большого порядка следует или обратиться к небольшому числу известных несистематических кодов, или, избрав иной способ построения, отказаться от выбора подмножеств классических двоичных кодов, исправляющих ошибки. ZFD-коды, о которых пойдет речь ниже, строятся на основе q-ичных кодов, исправляющих ошибки и комбинаторных блок-схем.

Б. Коды, основанные на q-ичных кодах

Разряды кодовых слов q-ичного кода, исправляющего ошибки, содержат символы, принадлежащие основному алфавиту из q символов [20]. При $q = 2$ имеем двоичный код — обычно используются символы 0 и 1. В данном разделе нас главным образом будут интересовать значения q, большие 2. Известно множество q-ичных кодов с различными значениями длин n_q и расстояний d_q различной величины (d_q — наименьшее число разрядов, в которых могут различаться два q-ичных кодовых слова) [20, 22].

Мы будем строить двоичные дизъюнктивные коды, отправляясь от q-ичных кодов, путем замены каждого q-ичного символа определенным двоичным набором. Чтобы упростить обсуждение, примем для начала, что каждый из этих q двоичных наборов имеет единичный вес при длине q. Таким образом, q-ичные символы 0, 1, ..., $q-1$ будут заменяться q-разрядными двоичными векторами 100 ... 0, 010 ... 0, 000 ... 1, соответственно. (Обобщение на случай других двоичных наборов будет описано в следующем подразделе.) При этом q-ичный код длины n_q преобразуется в двоичный код длины

$$n = qn_q, \tag{11}$$

имеющий минимальное расстояние $d = 2d_q$, вдвое превосходящее расстояние исходного q-ичного кода. Мощность кода $N = N_q$ остается той же самой. Поскольку полученный двоичный код будет равновесным веса $w = n_q$ (на каждый q-ичный разряд приходится по единице), его порядок, как ZFD-кода дается границей (10), и равен

$$m \geqslant [(n_q - 1)/(n_q - d_q)].$$

В целях максимизации величины m при фиксированной длине n_q и мощности N_q, будем искать q-ичные коды с возможно большим расстоянием. Исследование q-ичных кодов с достижимым максимальным расстоянием (ДМР) привело к открытию нескольких семейств таких кодов. В числе некоторых интересных свойств кодов ДМР было установлено, что такие коды являются *систематическими* — т. е. все n_q разрядов могут быть разделены на k_q (независимых) информационных разрядов и $r_q = n_q - k_q$ (зависимых) проверочных разрядов.

Эти результаты изложены в работе [22].

В частности, установленная там граница для минимального расстояния

$$d_q \leqslant r_q + 1$$

показывает, что для q-ичных кодов ДМР, удовлетворяющих границе $d_q = r_q + 1$, наибольший порядок равен

$$m = [(n_q - 1)/(k_q - 1)]. \tag{12}$$

Равенство в этом выражении есть непосредственное следствие теоремы 6 и того факта, что любой набор из $\lambda_{\max} = k_q - 1$ разрядов повторяется в кодовых словах в точности q раз. Вместе с тем, наличие k_q независимых разрядов дает

$$N_q = N = q^{k_q} \tag{13}$$

кодовых слов. Указанные соотношения (11), (12) и (13) связывают параметры q, k_q и n_q q-ичных кодов ДМР с параметрами n, N и m соответствующих им двоичных дизъюнктивных кодов.

Как известно, существование q-ичных кодов МДР установлено для нескольких областей изменения параметров [22]; для нашей цели лучше всего подходит семейство кодов, для которых q есть степень простого числа, не меньшая 3, а значения k_q и n_q удовлетворяют неравенствам

$$q + 1 \geqslant n_q \geqslant k_q + 1 \geqslant 3. \tag{14}$$

Преобразуя такие коды в ZFD-коды, можно заметить, что в силу (12) для предписанного значения m и любых конкретных значений q и k_q, использование длины n_q большей, чем $1 + m(k_q - 1)$ приводит лишь к увеличению величины n, в то

время как N и m остаются постоянными. С учетом указанного наименьшего значения n_q, параметры описываемого семейства ZFD-кодов принимают вид (прн $k_q \geqslant 2$):

$$\begin{aligned} n &= q(1 + m(k_q + 1)), \\ N &= q^{k_q}, \end{aligned} \tag{15}$$

где q — степень простого чнсла, причем $q \geqslant m(k_q - 1) \geqslant 3$. (При этом неравенство (14) выполняется автоматически.) Таким образом, установлено существование ZFD-кодов произвольно большой мощности и порядка, причем их мощность N экспоненциально растет с ростом длины n при фиксированном порядке m.

Имеющееся здесь ограничение на величину q оказывает влияние на минимальные размеры построенных ZFD-кодов; например, если $k_q = 5$, то $q \geqslant 4m$, и при этом

$$\begin{aligned} n &\geqslant 4m(1 + 4m), \\ N &\geqslant (4m)^5, \end{aligned}$$

так что соответствующие двоичные коды при $k_q \geqslant 5$ оказываются чрезвычайно длинными — заведомо, чересчур длинными, чтобы представлять значительный интерес для применеиий того типа, которые обсуждались в разделе II. Даже при $k_q = 4$ коды разумных размеров существуют лишь при малых значениях наибольшого порядка m (2 или 3). В остальных случаях имеем:

$k_q = 2$	$k_q = 3$
$n = q(1 + m)$	$n = q(1 + 2m)$
$N = q^2$	$N = q^3$
$q \geqslant m$	$q \geqslant 2m$

Для случая $k_q = 2$ известны коды, для которых q уже не обязано быть степенью простого числа, однако величина n_q должна лежать в более узком по сравнению с (14), промежутке

$$L(q) + 2 \geqslant n_q \geqslant 3,$$

где $L(q)$ обозначает число попарно ортогональных латинских квадратов порядка q. Снова, используя наименьшее значение n_q, получаем те же выражения для n и N, а величина q должна теперь выбираться настолько большой, чтобы выполнялось неравенство

$$L(q) \geqslant m - 1.$$

Известно, что величина $L(q)$ не превосходит $q - 1$, и не меньше, чем уменьшенный на единицу наименьший делитель q, имеющий вид степени простого числа [23]; например, $L(12) = L(2^2 \cdot 3) \geqslant 2$ $L(12) \leqslant 11$. Если само q есть степень простого,

эти единицы совпадают и получается оценка, упомянутая выше: $q \geqslant m$.

Если $k_q = 2$ и $m = 2$, то $n_q = 3$ и нужен лишь один латинский квадрат; любое значение $q \geqslant 3$ подходит для построения соответствующего ZFD_2-кода, имеющего вес 3, длину $n = 3q$ и мощность $N = q^2$.

Построение q-ичных кодов МДР описано в работе Синглтона [22]. Здесь достаточно отметить, что указанное семейство кодов включает в себя в качестве частных случаев q-ичные коды Рида — Соломона [24] ($n_q = q - 1$), q-ичные коды «с проверкой на четность» ($r_q = 1$), простые коды с повторением ($k_q = 1$), часть семейства q-ичных кодов Голея [25], исправляющих единичную ошибку ($r_q = 2$, $n_q = q^2 - 1$), и несколько кодов, связанных с ортогональными латинскими квадратами ($k_q = 2$) [26].

Все описанные ZFD-коды, основанные на q-ичных кодах, определенно являются малоэффективными в том отношении, что все они представляют собой коды с разбиением на подблоки, т. е. каждое двоичное кодовое слово, или блок, оказывается разбитым на секции (длины всех секций одинаковы), так что каждая секция кодируется сама по себе. Каждая из w секций имеет длину n_q и содержит одну-единственную единицу.

В общем случае такой код допускает пополнение без уменьшения его минимального расстояния (а стало быть, и порядка), путем добавления к нему слов, содержащих в каждой из секций более одной единицы. Например, ZFD_2-код, основанный на тернарном коде с параметрами $k_q = 2$, $n_q = 3$ содержит $N = q^2 = 9$ кодовых слов

0	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**
0	0	1	0	1	**0**	1	**0**	**0**
0	0	1	1	**0**	**0**	**0**	1	0
0	1	**0**	**0**	**0**	1	1	**0**	**0**
0	1	0	0	1	**0**	0	1	0
0	1	0	1	**0**	**0**	**0**	**0**	1
1	**0**	**0**	**0**	**0**	1	**0**	1	0
1	0	0	0	1	**0**	**0**	**0**	1
1	0	0	1	**0**	**0**	1	**0**	**0**

Без увеличения длины $n = 3q = 9$, или уменьшения m, можно добавить еще три кодовых слова

1	1	1	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1

так, что получится ZFD_2-код мощности $N = 12$.

В. Коды, основанные на каскадной композиции с q-ичными кодами

В предыдущем разд. VБ предполагалось, что каждый из символов q-ичного кода представляется двоичным набором длины q единичного веса. Однако нет причин, которые могли бы помешать использованию более общих представлений q символов исходного алфавита — достаточно потребовать, чтобы множество используемых двоичных наборов само являлось ZFD_m-кодом. Таким образом, можно допустить использование любых q слов произвольного ZFD_m-кода, мощность которого превосходит q. Поскольку длина такого кода может оказаться меньшей q разрядность n слов полученного таким образом двоичного каскадного кода может оказаться существенно меньшей $q^n q$, т. е. длины кодов, рассмотренных ранее.

Описанный способ представления q-ичных символов может быть с успехом использован для каскадной композиции, при которой небольшой ZFD-код с параметрами n_0, N_0 и m_0, преобразуется в больший ZFD-код с параметрами n_1, N_1, m_1 с использованием n_q-разрядного q-ичного кода, имеющего k_q информационных разрядов. Эти параметры связаны между собой следующими соотношениями, обобщающими (11), (12) и (13):

$$n_1 = n_0 n_q,$$
$$N_1 = q^{k_q},$$
$$m_1 = \min(m_0,\ m_q),$$

где q — степень простого числа, удовлетворяющая неравенствам $n_q - 1 \leqslant q \leqslant N_0$, а

$$m_q = [(n_q - 1)/(k_q - 1)].$$

При выборе значения n_q достаточно потребовать выполнения условия $m_0 \geqslant m_q = m_1 = m$, откуда получаем

$$n_1 = n_0 (1 + m (k_q - 1)), \qquad (16)$$
$$N_1 = q^{k_q},$$

причем

$$m (k_q - 1) \leqslant q \leqslant N_0.$$

Если в качестве меньшего ZFD-кода выбирается равновесный код веса 1, то $n_0 = N_0 = m_0 = q$, и мы приходим к семейству кодов (15), рассмотренному в разд. V.Б.

Отправляясь от простого равновесного кода веса 1, можно итерировать каскадные композиции, сохраняя порядок m, для построения произвольно больших *ZFD*-кодов.[1])

На различных стадиях итерации могут использоваться различные q-ичные коды.

При использовании однотипных q-ичных кодов (различающихся лишь величиной q, которая заменяется на $q' = N_1$, вторая итерация кода (16) дает код с параметрами

$$n_2 = n_0(1 + m(k_q - 1))^2,$$

$$N_2 = q^{k_q^2},$$

причем

$$m(k_q - 1) \leqslant q' = N_1 = q^{k_q}.$$

(Очевидно, если q — степень простого числа, то $q' = q^{k_q}$ — также степень простого.) После осуществления c итераций каскадной композиции над исходным равновесным кодом веса 1, имеем

$$\begin{aligned} n_c &= q(1 + m(k_q - 1))^c, \\ N_c &= q^{k_q^c}, \end{aligned} \tag{17}$$

причем, как и ранее, q — степень простого, удовлетворяющая неравенству $q > m(k_q - 1)$.

Число итераций c можно оптимизировать по отношению к q и k_q, заметив, что замена c на $c - 1$ может быть компенсирована заменой q на q^{k_q} при сохранении величины N_c постоянной. Эта замена изменяет длину кода до величины $q^{k_q}(1 + m \times \times (k_q - 1))^{c-1}$, позволяя увеличивать ее (если c слишком мало) или уменьшать (если c слишком велико), в соответствии с тем, оказывается ли величина $q^{k_q - 1}/(1 + m(k_q - 1))$ большей или меньшей 1. Таким образом, для заданных значений m и N, при выборе q можно опираться не только на нижнюю, но и на верхнюю оценку

$$1 + m(k_q - 1) \leqslant q^{k_q - 1} \leqslant 1 + m(k_q - 1)^{k_q}.$$

При $k_q = 2$ эти оценки имеют вид

$$1 + m \leqslant q \leqslant (1 + m)^2,$$

а при $k_q = 3$

$$(1 + 2m)^{1/2} \leqslant q \leqslant (1 + 2m)^{3/2}.$$

---

[1]) Разумеется, исходный код, используемый для итераций не обязан быть равновесным, иметь вес 1 или даже быть q-ичным кодом, достаточно, лишь, чтобы он был *ZFD*-кодом с подходящими параметрами. Очень хорошим примером исходных кодов могут служить коды разд. V. Г, основанные на блок-схемах.

(В последнем случае нижняя граница выполняется автоматически, поскольку $q \geqslant 2m$.)

При $k_q = 2$ описанный метод сохраняет свою силу даже в том случае, когда q не является степенью простого числа, если только как и ранее $L(q) \geqslant m - 1$. Справедливость этого утверждения вытекает непосредственно из того факта, что $L(q^2) \geqslant L(q)$. Последнее неравенство без труда обосновывается следующей конструкцией [1]). Пусть $S_1, \ldots, S_k, \ldots, S_L$ — множество L попарно ортогональных латинских квадратов порядка q, записанных в виде матриц с элементами 0, 1, 2, ..., $q-1$ и $s_{ij}^{(k)}$ — общий элемент матриц. Тогда можно построить множество $T_1, \ldots, T_k, \ldots, T_L$, состоящее из L попарно ортогональных латинских квадратов порядка q^2, каждый из которых имеет вид

$$T_k = \begin{bmatrix} T_{11}^{(k)} & T_{12}^{(k)} & \cdots \\ T_{21}^{(k)} & T_{22}^{(k)} & \\ \cdots & & \end{bmatrix},$$

где $T_{ij}^{(k)}$ — матрица размеров $q \times q$, имеющая вид

$$T_{ij}^{(k)} = S_k + qs_{ij}^{(k)} J,$$

где, наконец, J — матрица размеров $q \times q$, сплошь состоящая из единиц.

Г. Коды, основанные на блок-схемах

Комбинаторные блок-схемы, используемые в статистике, представляют собой многопараметрическое семейство комбинаторных конфигураций. Блок-схемы удобно изображать при помощи матриц из нулей и единиц.

Матрица инцидентности S, изображающая *сбалансированную неполную блок-схему* (*BIB*) с параметрами (v, k, b, r, λ), имеет b строк и v столбцов; в каждой строке содержится ровно k единиц, а в каждом столбце — ровно r единиц, причем скалярное произведение любой пары различных столбцов в точности равно λ.

Как известно, при этом должны выполняться соотношения

$$vr = bk \quad \text{и} \quad k(r-1) = \lambda(v-1).$$

При построении равновесного кода роль кодовых слов играют либо строки, либо столбцы матрицы S. Если роль кодовых слов играют *столбцы* матрицы S, то матрицей кода будет

$$A = S^t$$

[1]) Это доказательство принадлежит Б. Элспасу.

— матрица, транспонированная к S, так что

$$n=b, \quad N=v, \quad w=r, \quad \lambda_{ij}=\lambda=\lambda_{\max}.$$

Поскольку для BIB-схемы $v \leqslant b$, то, стало быть, $N \leqslant n$ — получается неинтересное семейство дизъюнктивных кодов.

Если роль кодовых слов выполняют строки матрицы S, то

$$A=S,$$

так что

$$n=v, \quad N=b, \quad w=k, \quad \lambda_{ij} \leqslant \mu_{\max},$$

где $\mu_{\max}$ — наибольшее скалярное произведение пар *строк* матрицы S. Стало быть,

$$m \geqslant [(w-1)/\mu_{\max}] = [(k-1)/\mu_{\max}].$$

К сожалению, в общем случае для BIB-схем не известны простые соотношения, связывающие величины $\lambda_{\max}$ и $\mu_{\max}$. Если $\lambda=1$, то, как нетрудно убедиться [1]) $\mu_{\max}=1$, так что

$$m=w-1=k-1;$$

в то же время, если $\lambda>1$, то в общем случае можно лишь утверждать, что $2 \leqslant \mu_{\max} \leqslant k$.

Теория блок-схем далеко не закончена, хотя известны конструкции ряда семейств и отдельных схем [27], [28]. К сожалению, значения параметров, представляющие практический интерес для построения дизъюнктивных кодов по большей части выходят за рамки значений, найденных в табулированных для статистических применений. Исключение, в основном, составляет случай $\lambda=1$, для которого известны схемы с параметрами

$$(v,\ k,\ r,\ b,\ \lambda)=\left(n,\ w,\ \frac{n-1}{w-1},\ N=\frac{n(n-1)}{w(w-1)},\ 1\right);$$

а также случаи

$k=3$: $v=1$ или $3 \pmod 6$, $b=r(2r+1)/3$, $r=(v-1)/2$,
так что $w=3$, $n=1$ или $3 \pmod 6$,
$N=n(n-1)/6$, $m=2$;

и

$k=4$: $v=3r-1$, $b=rv/4$, $r=$ степени простого числа, так что
$w=4$, $(n-1)/3=$ степени простого,
$N=n(n-1)/12$, $m=3$.

[1]) Равенство и максимальность этого значения m вытекают непосредственно из теоремы 6.

Отметим, что указанные коды достигают границы (5), и поэтому не могут быть пополнены с сохранением длины и веса. (На самом деле можно показать, что любой ZFD_m-код, достигающий указанной границы эквивалентен некоторой BIB-схеме.) Семейство блок-схем случая $k = 3$ носит название систем троек Штейнера [27], и имеет многочисленные приложения в теории кодирования [26].

VI. ПОСТРОЕНИЕ *UD*-КОДОВ

A. *UD*-коды, основанные на проверочных матрицах

Разумеется, UD-коды могут быть получены из ZFD-кодов того же порядка (теоремы 3 и 4), но в то же время имеется преимущество, дающее возможность воспользоваться менее ограничительными условиями, сформулированными в теореме 4 для построения UD-кодов, превосходящих по мощности ZFD-коды того же порядка и длины. Ниже описаны три различных перехода к построению UD-кодов небольших порядков. Некоторые из них оказываются довольно эффективными.

Как известно, матрица H^t, транспонированная к проверочной матрице H обыкновенного двоичного кода, исправляющего e ошибок, обладает тем свойством, что любое подмножество ее строк и всевозможные их суммы, содержащие не более e слагаемых, не содержат повторений [29]. Это — почти в точности то самое свойство, которому должна удовлетворять матрица A UD_e-кода, с той только разницей, что здесь фигурирует сложение по модулю 2, в то время как для определения дизъюнктивных кодов фигурирует булево сложение, т. е. дизъюнкция. По этой причине матрица H^t не может быть прямо использована в качестве матрицы A дизъюнктивного кода порядка $m = e$, но может оказаться небесполезным поискать такой способ модификации матрицы H^t, чтобы свойство разнозначности сумм сохранялось и в случае булева суммирования.

Ниже мы указываем такие модификации для случаев $e = 2$ и $e = 3$, получая соответственно семейства UD_2- и UD_3-кодов.

Для случая $e = 2$ каждый разряд матрицы H^t дополняется в той же строке его дополнением по правилу

$$0 \to 01, \quad 1 \to 10.$$

Эта подстановка может быть осуществлена, например, посредством введения матрицы

$$A = [H^t \mid \overline{H}^t],$$

где $\bar{H}^t$ представляет собой инверсию матрицы H^t. Сравним таблицы суммирования элементов матриц H^t и A:

$\oplus$	0	1
0	0	1
1	1	0

$\vee$	01	10
01	01	11
10	11	10

Очевидно, булева сумма двух любых строк матрицы A однозначно преобразуется в соответствующую сумму строк матрицы H^t по правилу

$$00 \to 0, \quad 10 \to 0, \quad 11 \to 1.$$

Аналогично, любая строка матрицы A однозначно преобразуется в строку матрицы H^t по правилу $01 \to 0$, $10 \to 1$.

Двойственная интерпретация сочетания 10 не приводит к каким-либо коллизиям, поскольку всегда можно отличить *строку* матрицы A от *дизъюнкции строк* — действительно, строки A не содержат сочетаний 11, в то время как дизъюнкции строк непременно содержат сочетания 11, возникающие по той причине, что различные строки всегда различаются хотя бы в одном разряде.

Таким образом, если матрица H^t отвечает проверочной матрице кода, исправляющего 2 ошибки, то строки соответствующей матрицы A образуют UD_2-код.

Коды, входящие в семейство кодов Боуза — Чоудхури [30] при $e=2$ и любом натуральном $\mu \geqslant 2$ имеют не более 2μ проверочных разрядов при длине $2^\mu - 1$. Соответствующая матрица H^t имеет $2^\mu - 1$ строк и не более 2μ столбцов. Поскольку A содержит вдвое больше столбцов, имеем $n \leqslant 4\mu$ и $N = 2^\mu - 1$. Таким образом, для любого значения n, кратного 4, существует UD_2-код мощности $N = 2^{n/4} - 1$. Экспоненциальный рост мощности этих кодов показывает, что при достаточно больших n они оказываются мощнее всех найденных ранее ZFD_2-кодов (а стало быть и UD_2-кодов).

Для случая $e=3$ нужно каким-то образом отразить в матрице A соотношения, связывающие столбцы матрицы H^t, так как любая простая подстановка вроде $0 \to \alpha$, $1 \to \beta$ уже не может адекватно описать различие между двойными и тройными суммами, поскольку

$0 \oplus 0 \oplus 1 = 1$, а $0 \oplus 1 \oplus 1 = 0$, в то время как

$$\alpha \vee \alpha \vee \beta = \alpha \vee \beta \vee \beta,$$

$1 \oplus 1 = 0$, а $1 \oplus 1 \oplus 1 = 1$, в то время как $\beta \vee \beta = \beta \vee \beta \vee \beta$.

Можно показать, что в том случае, когда *каждая пара столбцов* матрицы H^t представляется в матрице A в соответствии

с правилом

$$00 \to 1000$$
$$01 \to 0100$$
$$10 \to 0010$$
$$11 \to 0001$$

и матрица H^t соответствует коду, исправляющему три ошибки, то строки полученной описанным образом матрицы A образуют UD_3-код.

Кодам Боуза — Чоудхури [30] при $e = 3$ и любом натуральном $\mu \geqslant 3$ соответствует матрица H^t, имеющая $2^\mu - 1$ строк и не более 3μ столбцов. Стало быть, пары столбцов матрицы H^t могут быть выбраны не более, чем $\binom{3\mu}{2}$ способами, и число столбцов матрицы A равняется

$$n \leqslant 4\binom{3\mu}{2} = 6\mu(3\mu - 1),$$

а число строк есть

$$N = 2^\mu - 1.$$

Этот код оказывается малоэффективным, если значения n и N относительно невелики, поскольку $N > n$ лишь при $\mu \geqslant 13$ (когда $n \geqslant 2964$), но является привлекательным с асимптотической точки зрения:

$$N > 2^{\sqrt{n/18}}.$$

Примерно такая же скорость роста наблюдается для q-ичных ZFD_3-кодов, получаемых при помощи итерированной каскадной композиции при $k_q = 2$ и $q = 1 + m = 4$, для которых

$$N = 2^{\sqrt{n}}.$$

Б. Коды веса 2, основанные на графических конструкциях

В силу теоремы 6 наилучшие равновесные коды веса $w = 2$ не могут быть ZFD_2-кодами, но могут быть UD_2-кодами. Ниже мы описываем два семейства таких кодов, и показываем, что их мощность N растет асимптотически как $n^{3/2}$.

Рассмотрим сначала случай кодов с разбиением на подблоки, у которых каждая единица располагается в своей секции n-разрядного кодового слова. Пусть данное кодовое слово содержит свои две единицы соответственно в i-м разряде левой секции, и в j-м разряде правой секции. Тогда весь код удобно изображать при помощи двоичной матрицы G с общим элемен-

том g_{ij}, который равен 1 в том и только том случае, когда в рассматриваемый код входит кодовое слово, содержащее единицы в i-м разряде левой и j-м разряде правой секций соответственно. Очевидно, мощность N рассматриваемого кода равна общему числу единиц в матрице G.

Чтобы удовлетворялось условие UD_2, никакие две единицы в матрице G не должны занимать той же пары строк и столбцов, какие занимают другие пары единиц. Иначе говоря, никакие две строки матрицы G не могут содержать пар единиц в одних и тех же столбцах. Таким образом, мы будем искать такие двоичные матрицы G, которые при фиксированном полупериметре n содержат наибольшее число N единиц, при условии, что скалярные произведения любых двух различных строк не превосходит единицы.

Этому условию удовлетворяет матрица BIB-схемы [27] с параметрами (v, k, b, r, λ), для которой $n = v + b$, $N = kr$, $\lambda = 1$. В пределах существования этих схем, величина N должна быть максимизирована при постоянном значении n и значении $v/b = k/r$ по возможности близком к 1. Поэтому для достижения полной регулярности G должна быть матрицей *симметричной* BIB-схемы:

$$v = b \text{ и } k = r, \text{ откуда } n = 2v \text{ и } N = k^2.$$

Такие симметричные блок-схемы, как известно, существуют при значениях k, для которых $k - 1$ является степенью простого числа [27]. Соотношения, связывающие параметры блок-схем дают

$$n = 2(k^2 - k + 1),$$
$$N = k(k^2 - k + 1).$$

Таким образом для указанных значений k существуют равновесные с UD_2-коды с разбиением на подблоки, имеющие вес 2 и мощность

$$N = \frac{n}{4}\left(1 + \sqrt{2n - 3}\right).$$

Асимптотически,

$$N \sim \frac{n^{3/2}}{2\sqrt{2}}.$$

Рассмотрим теперь случай, когда обе единицы не обязательно должны располагаться в предписанных секциях кодового слова. Пусть каждому из n разрядов кодового слова отвечает вершина некоторого n-вершинного графа. Тогда каждому кодовому слову соответствует (неориентированное) ребро, соединяющее пару вершин, отвечающих положению единиц в этом

кодовом слове. В этих терминах требуется расположить в n-вершинном графе наибольшее число N ребер так, чтобы выполнялось следующее условие, соответствующее условию UD_2: никакая пара ребер не должна быть инцидентна тому же множеству вершин, которому инцидентна какая-либо другая пара ребер.

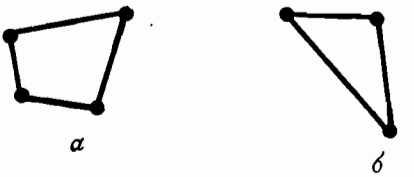


Рис. 1

Это означает, что не должно быть подграфов, изображенных на рис. 1. Стало быть, исключаются циклы длины 4 и 3, а циклы длины 2 (повторяющиеся кодовые слова) и циклы длины 1 (кодовые слова веса 1) исключены из рассмотрения с самого начала. Таким образом, мы ищем максимальные n-вершинные графы, не содержащие *циклов длины меньшей* 5. Достаточность

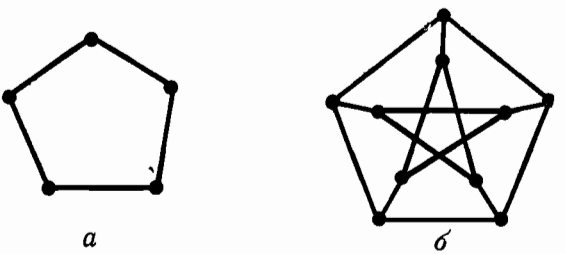


Рис. 2

этого условия очевидна: любой n-вершинный граф, у которого длина кратчайшего цикла не меньше 5, порождает UD_2-код веса 2.

Вполне регулярные графы рассматриваемого типа ранее изучались А. Хоффманом и Р. Синглтоном [31], и были названы «графами Мура диаметра 2». Для *степени* t таких графов — т. е. числа ребер, инцидентных каждой вершине, должны выполняться некоторые соотношения, которые исключают все возможности, кроме четырех следующих:

$$t=2, \quad n=5, \quad N=5,$$
$$t=3, \quad n=10, \quad N=15,$$
$$t=7, \quad n=50, \quad N=175,$$
$$t=57, \quad n=3250, \quad N=92\,625.$$

Первые два из указанных графов изображены на рис. 2, третий приводится в работе Хоффмана, а четвертый случай остается неопределенным. Параметры кода связаны со степенью графа соотношением $n = 1 + t^2$, $N = nt/2 = t(1 + t^2)/2$, так что $N = n\sqrt{n-1}/2$. Для промежуточных значений n, лежащих между указанными, ближайший вполне регулярный граф может быть уменьшен путем отбрасывания вершин вместе с инцидентными им ребрами, так, чтобы на каждом шаге отбрасывалось

Таблица 1

n	N	n	N
5	5	30	70
10	15	35	95
15	25	40	120
20	35	45	145
25	50	50	175

как можно меньшее число ребер. Параметры некоторых из полученных таким образом промежуточных кодов приведены в табл. 1. В любом случае, имеем асимптотический рост

$$N \sim \frac{n^{3/2}}{2}$$

— это немного лучше, чем для соответствующих UD_2-кодов с разделением на подблоки, но намного меньше, чем для UD_2-кодов, основанных на проверочных матрицах обыкновенных кодов, исправляющих 2 ошибки, описанных в предыдущем разделе. Коды табл. 1 также немного слабее простейших q-ичных и блок-схемных ZFD_2-кодов разд. V.

В. Попарная каскадная композиция UD_2-кодов

В разд. V было показано, как из известного ZFD_2-кода может быть получен трехсекционный ZFD_2-код с разделением на подблоки, имеющий утроенную длину. Кодовые слова трехсекционного кода имеют вид

$$(a_1)(b_1)(c_{11}), \quad (a_2)(b_2)(c_{22}) \quad \text{и т. д.},$$

причем две первых секции — подслова a_1, a_2, ... и b_1, b_2, ... выбираются независимым образом из числа слов меньшего кода. Третьи секции — подслова c_{11}, c_{22}, ... выбираются из того же кода в соответствии с некоторым латинским квадратом, причем строки квадрата соответствуют первым секциям кодовых слов, а столбцы — вторым. Таким образом, из данного n-разрядного ZFD_2-кода, содержащего N слов можно получить новый ZFD_2-код, имеющий $3n$ разрядов и N^2 слов.

Покажем теперь, что большие UD_2-коды можно аналогичным образом строить из меньших UD_2-кодов, с той только разницей, что длина третьего подблока может быть сделана существенно меньшей, чем требуется в случае ZFD_2-кодов. Таким образом могут быть построены UD_2-коды, чья мощность N намного превосходит мощность ZFD_2-кодов той же длины.

Если подслова, стоящие в двух первых секциях, a_1, a_2, ... и b_1, b_2, ... взяты из UD_2-кода, то первый и второй подблоки дизъюнкции вида

$$(a_1 \vee a_2)(b_1 \vee b_2)(c_{11} \vee c_{22})$$

можно по отдельности декодировать и определить их составляющие.

При этом однако возникают две интерпретации

$$\begin{matrix}(a_1)(b_1)(c_{11}) \\ (a_2)(b_2)(c_{22})\end{matrix} \quad \text{и} \quad \begin{matrix}(a_1)(b_2)(c_{12}) \\ (a_2)(b_1)(c_{21}),\end{matrix}$$

которые невозможно отличить друг от друга без участия третьего подблока, устроенного подходящим образом. Поэтому потребуем выполнения условия $c_{11} \vee c_{22} \neq c_{12} \vee c_{21}$. Это условие можно выразить естественным образом, располагая все множество подслов, отвечающих третьим секциям, в матрицу C размеров $N \times N$, точно так, как это делалось для латинского квадрата. Индексы строк и столбцов отвечают выбору соответствующих подслов a_i и b_j. Таким образом, каждый из элементов матрицы C представляет собой третью секцию одного из N^2 кодовых слов, полученных описанным способом.

Указанное выше условие можно записать в виде $c_{ij} \vee c_{kl} \neq c_{il} \vee c_{kj}$, $i \neq k$, $j \neq l$; оно должно выполняться для любой четверки элементов, образующих в рассматриваемой матрице прямоугольную подматрицу. Иными словами, для любой 2×2-подматрицы матрицы C, диагональные дизъюнкции должны быть различны.

Если c_{ij} сводится к одному двоичному разряду, то наибольшая матрица C, удовлетворяющая этому условию, как легко видеть, есть единичная матрица размеров 3×3:

$$C_1 = \begin{pmatrix} 100 \\ 010 \\ 100 \end{pmatrix}.$$

Отправляясь от 3-разрядного равновесного кода веса 1 (это, разумеется, UD_2-код), содержащего три кодовых слова

$$\begin{matrix} 001 \\ 010 \\ 100 \end{matrix}$$

и матрицы C_1, получаем 7-разрядный UD_2-код, содержащий $3^2 = 9$ слов

0 0 1	0 0 1	1
0 0 1	0 1 0	0
0 0 1	1 0 0	0
0 1 0	0 0 1	0
0 1 0	0 1 0	1
0 1 0	1 0 0	0
1 0 0	0 0 1	0
1 0 0	0 1 0	0
1 0 0	1 0 0	1

Теперь требуется найти 9×9-матрицу C_2, чьи элементы удовлетворяли бы указанному выше диагональному ограничению на подматрицы. В общем случае, требуется преобразовывать $3^p \times 3^p$-матрицу C_p в $3^{p+1} \times 3^{p+1}$-матрицу C_{p+1}, $p = 1, 2, \ldots$, таким образом, чтобы C_{p+1} удовлетворяла нужному ограничению на подматрицы, если ему удовлетворяет матрица C_p.

С этой целью предположим, что дана матрица C_p, удовлетворяющая требуемому ограничению и такая, что для нее существует разбиение на 9 частей вида

$$C_p = \begin{pmatrix} X & Y & Z \\ Z & X & Y \\ Y & Z & X \end{pmatrix},$$

где X, Y и Z представляют собой $3^{p-1} \times 3^{p-1}$-подматрицы с векторными элементами. Матрица C_1, очевидно, обладает таким строением. Пусть $1X$ обозначает матрицу, которая получается из матрицы X путем приписывания ко всем ее векторным элементам (скажем, к их левым концам) дополнительного разряда, содержащего двоичную 1; аналогично определяются $0X$, $1Y$, $0Y$, $1Z$ и $0Z$. Покажем теперь, что матрица

$$C_{p+1} = \left[\begin{array}{ccc|ccc|ccc} 11X & 10Y & 10Z & 00X & 01Y & 00Z & 00X & 00Y & 01Z \\ 10Z & 11X & 10Y & 00Z & 00X & 01Y & 01Z & 00X & 00Y \\ 10Y & 10Z & 11X & 01Y & 00Z & 00X & 00Y & 01Z & 00X \\ \hline 00X & 00Y & 01Z & 11X & 10Y & 10Z & 00X & 01Y & 00Z \\ 01Z & 00X & 00Y & 10Z & 11X & 10Y & 00Z & 00X & 01Y \\ 00Y & 01Z & 00X & 10Y & 10Z & 11X & 01Y & 00Z & 00X \\ \hline 00X & 01Y & 00Z & 00X & 00Y & 01Z & 11X & 10Y & 10Z \\ 00Z & 00X & 01Y & 01Z & 00X & 00Y & 10Z & 11X & 10Y \\ 01Y & 00Z & 00X & 00Y & 01Z & 00X & 10Y & 10Z & 11X \end{array}\right],$$

имеющая такое же строение, как матрица C_p, также удовлетворяет диагональному ограничению на подматрицы.

Прежде всего отметим, что внутри каждой из девяти частей, на которые разбивается матрица C_{p+1} все двоичные векторы c_{ij} имеют одинаковые части, отвечающие подматрицам X, Y и Z. Поэтому любая 2×2-подматрица, которая целиком попадает в какую-либо из указанных девяти частей, заведомо удовлетворяет требуемому ограничению. Больше того, любая 2×2-подматрица, чьи углы попадают в различные части, будет также удовлетворять этому ограничению — и по той же причине, кроме, разве что тех случаев, когда какая-нибудь горизонтальная или вертикальная пара ее углов попадает в *соответственные* строки или столбцы различных частей. Однако в таком случае добавленные разряды обеспечивают выполнение требуемого ограничения, создавая в соответствующих позициях точно такое же расположение разрядов, какое имелось в матрице C_1. Первый из добавленных разрядов обслуживает тот случай, когда все четыре угла подматрицы попадают в соответственные позиции четырех разных частей. Второй из добавленных разрядов обслуживает тот случай, когда подматрица целиком лежит внутри трех смежных частей, располагающихся вдоль одной линии, а левая и правая пары ее углов (или верхняя и нижняя пары углов) попадают в соответственные столбцы (или строки, соответственно) этих трех частей.

Таким образом, все подматрицы удовлетворяют диагональному ограничению, и, стало быть, C_{p+1} — подходящая матрица для UD_2-кода.

Всякий раз, когда p увеличивается на единицу, к c_{ij} добавляется два двоичных разряда, стало быть, элементы C_p содержат $2p-1$ двоичных разрядов. Таким образом, для каждого натурального p описанная итерация каскадных композиций порождает код UD_2-код мощности N и длины $n(p)$, дающихся выражениями

$$N=3^{2^p},\quad n(p)=2n(p-1)+(2p-1),$$

или $n(p)=6\cdot 2^p-(2p+3)$. При этом асимптотически

$$N\sim 3^{n/6}.$$

Аналогичное применение итеративной каскадной композиции дает ZFD_2-коды, мощность которых также равна $N=3^{2^p}$, но при длине $n(p)=3n(p-1)$, так что $n(p)=3^{p+1}$. Таким образом, для соответствующих значений p имеем

$$N=3^{1/2\cdot n^{\log_3 2}},$$

что намного меньше соответствующего значения для UD_2-кодов.

VII. Обсуждение

В разд. I—III мы показали, как новый класс кодов — двоичные дизъюнктивные коды могут быть использованы в системах хранения информации и связи, исследовали некоторые свойства этих кодов и описали ряд способов их построения, охватывающих широкий круг значений параметров. При этом первоначальном исследовании *ZFD*- и *UD*-кодов остались незатронутыми проблемы, связанные с их реализацией посредством кодирующих и декодирующих логических схем, и с построением по настоящему оптимальных кодов. Было бы также полезно уметь использовать часть кодового расстояния этих кодов для защиты от возможных ошибок, даже если бы ради этого пришлось уменьшить порядок кода, и изучить соотношения между степенью обнаружения ошибок и порядком кода.

В области теории было бы желательно иметь лучшие верхние границы для величины N как функции n и m, как, впрочем, и лучшее понимание внутренней взаимосвязи *ZFD*- и *UD*-кодов.

Сравнение известных *ZFD*-кодов показывает, что среди коротких кодов наибольшими являются те, которые основаны на блок-схемах, а среди более длинных наибольшие — те, которые основаны на q-ичных кодах, исправляющих ошибки. Поскольку все коды, основанные на блок-схемах, равновесны, эти результаты позволяют предположить, что коды большого веса, основанные на блок-схемах, если такие существуют и могут быть найдены, должны оказаться лучше прочих. Разумеется, тот факт, что каскадные коды, основанные на q-ичных кодах, являются кодами с разделением на подблоки и почти во всех случаях допускают пополнение, указывает на их малую эффективность, которая может быть преодолена путем более равномерного распределения единиц внутри кодового слова, так, как это имеет место для кодов, связанных с блок-схемами.

Сравнение *ZFD*- и *UD*-кодов с вероятностными дизъюнктивными кодами наталкивается на те же трудности, с которыми приходится иметь дело при сравнении обычных детерминированных и вероятностных кодов, исправляющих ошибки. Следовало бы, приняв некоторый тип статистики канала (здесь — статистики использования дескрипторов), приписать многочисленным сочетаниям ошибок различных типов (здесь, возможным несовпадениям запрашиваемых дескрипторов с дескрипторами документов) соответствующие вероятности ошибки (ложного соответствия). При рассмотрении дизъюнктивных кодов ситуация еще больше осложняется тем, что для действительного осуществления подобного сравнения — путем аналитических вычислений или моделирования, необходимо учесть влияние других параметров — размеров информационного массива и соот-

ношения между числом запрашиваемых и документальных дескрипторов. Кроме того, в применениях, связанных с информационным поиском, на осмысленность результата весьма критическим образом влияют некоторые предположения, которые вовсе не соответствуют практике (одинаковое использование дескрипторов и отсутствие внутренних зависимостей между ними).

Авторы глубоко признательны д-ру Бернарду Элспасу, чей давний интерес и усилия, направленные на изучение класса кодов, предположенного в данной работе, составили существенный вклад в осуществленные исследования. Он также оказал прямую помощь при получении некоторых доказательств и результатов, касающихся равновесных и q-ичных кодов.

ЛИТЕРАТУРА

[1] Schultz C. K. An application of random codes for literature searching, in «Punched Cards, Their Applications to Science and Industry», Casey R. S., *et al.*, Eds., Reinhold Publishing Corp., New York, N. Y., ch. 10, see also chs. 18 and 23; 1958.

[2] Mooers C. N. The Application of Simple Pattern Inclusion Selection to Large-Scale Information Retrieval Systems. Rome Air Development Center, Rome, N. Y., Rept. No. RADC-TN-59-157, Zator Technical Bulletin No. 131; April, 1959. See Bulletin No. 120 for additional references to Zatocoding.

[3] Taube M. Superimposed Coding for Data Storage. Documentation, Inc., Washington, D. C., Tech. Rept. No. 15; September, 1956.

[4] Frei E. H. and Goldberg J. A method for resolving multiple responses in a parallel search file. IRE Trans, on Electronic Computers, vol. ES-10, pp. 178—722; December, 1961.

[5] Brenner C. W. and Mooers C. N. A case history of a zatocoding information retrieval system, in «Punched Cards, Their Application to Science and Industry», Casey R. S., *et al.*, Eds., Reinhold Publishing Corp., New York, N. Y., ch. 15; 1958.

[6] Goldberg J., *et al.*, Multiple Instantaneous Response File. Stanford Research Institute, Menlo Park, Calif., Final Rept., SRI Project 3101, Rept. No. RADC-TR-61-233; August, 1961.

[7] Rosen S. A. An approach to a distributed memory. *Proc. 1961 Symp. on the Principle of Self-Organizing Mathines*, Pergamon Press, Inc., New York, N. Y., pp. 425—444; 1962.

[8] Singleton R. C. Random Selection Rates for Single-Field Superimposed Coding. Stanford Research Institute, Menlo Park, Calif., Suppl. A to Quarterly Rept. 4, Contract AF 30 (602) — 2142; November, 1960.

[9] Wise C. S. Mathematical analysis of coding systems, in «Punched Cards, Their Application to Science and Industry», Casey G. S., *et al.*, Eds., Reinhold Publishing Corp., New York, N. Y., ch. 21; 1958.

[10] Mooers C. N. The Exact Distribution of the Number of Positions Marked in a Zatocoding Field, Zator Co., Boston, Mass., Zator Technical Bulletin No. 73; 1952.

[11] Orosz C. and Takacs L. Some probability problems concerning the marking of codes into the superposition field. *J. Documentation*, vol. 12, pp. 231—234; December, 1956.

[12] Stiassny S. Mathematical Analysis of Various Superimposed Coding Methods. IBM Research Center, Yorktown Heights, N. Y., IBM Res. Rept. No. RC-103; April, 1959.

[13] Shannon C. E. and Weaver W. Mathematical Theory of Communications. University of Illinois Press, Urbana; 1949.
[14] Elias P. Error-free coding. IRE Trans. on Information Theory, vol. IT-4, pp. 29—35; September, 1954.
[15] Costas J. Poisson, Shannon, and the radio amateur. PROC. IRE, vol. 47, pp. 2058—2068; December, 1959.
[16] Cohn D. L. and Gorman J. M. A code separation property. IRE Trans. on Information theory (Correspondence), vol. IT-8, pp. 382—383; October, 1962.
[17] Minnick R. C. and Ashenfurst R. L. Multiple-coincidence magnetic storage systems. *J. Appl. Phys.*, vol. 26, pp. 575—579; May, 1955.
[18] Minnick R. C. Simultaneous matrix storage systems. *Proc. International Symp. on the Theory of Smitching*, Harvard University Press, Cambridge, Mass., pt. II, pp. 144—148; April, 1957.
[19] Singleton R. C. Load sharing core switches based on block designs. IRE Trans. on Electronic Computers, vol. EC-11, pp. 346—352; June, 1962. Minnick R. C. Magnetic core access switches. IRE Trans. on Electronic Computers, vol. ES-11, pp. 352—368; June, 1962. Neumann P. G. On the logical design of noiseless load-sharing matrix switches. IRE Trans. on Electronic Computers, vol. EC-11, pp. 369—374; June, 1962. See also the references and bibliographies of Singleton, Minnick, and Neumann.
[20] Peterson W. W. Error-Correcting Codes. Mass. Inst. Tech. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, N. Y., p. 30 ff.; 1961.
[21] *Ibid.*, pp. 67—70.
[22] Singleton R. C. Maximum distance q-nary codes. IEEE Trans. on Information Theory, vol. IT-10, pp. 116—118; April, 1964.
[23] Mann H. B. Analysis and Design of Experiments. Dover Publications, Inc., New York, N. Y., chs. 7, 8; 1949.
[24] Reed I. S. and Solomon G. Polynomial codes over certain finite fields. *J. Soc. Indus. Appl. Math.*, vol. 8, pp. 300—304; June, 1960.
[25] *Golay M. J. E.* Notes on digital coding. PROC. IRE (Correspondence), vol. 37, p. 657; June, 1949.
[26] Kautz W. H. and Elspas B. Single-error-correcting codes for constant-weight data words, submitted to IEEE Trans. on Information Theory.
[27] Hall M., Jr., A survey of combinatorial analysis, in «Some Aspects of Analysis of Probability», Kaplansky I., *et. al.*, John Wiley and Sons, Inc., New York, N. Y., 1958.
[28] Cochran W. G. and Cox G. M. Experimental Design. John Wiley and Sons, New York, N. Y., 1957.
[29] Peterson, *op. cit.*, p. 33.
[30] Peterson, *op. cit.*, p. 162 ff.
[31] Hoffman A. J. and Singleton R. R. On Moore graphs with diameters 2 and 3. *IBM J. Res. and Develop.*, vol. 4, pp. 497—504; November, 1960.
[32*] Малютов М. Б. Математические модели и результаты в теории отсеивающих экспериментов, в сб. Вопросы кибернетики, вып. 35, М., «Советское радио», 1977, 5—69.
[33*] Дьячков А. Г., Рыков В. В. Об одной модели для суммирующего канала с интенсивным доступом. Проблемы передачи информации, т. 17, вып. 2 (1982), 26—38.
[34*] Дьячков А. Г., Рыков В. В. Границы длины дизъюнктивных кодов. Проблемы передачи информации, т. 18, вып. 3 (1983), 7—13.

---

* Звездочкой отмечены работы, добавленные переводчиком, содержащие сведения о полученных в последнее время результатах теории дизъюнктивных кодов.

СОДЕРЖАНИЕ

УВАЖАЕМЫЙ ЧИТАТЕЛЬ!

Ваши замечания о содержании книги, ее оформлении, качестве перевода и другие просим присылать по адресу: 129820, Москва, И-110, ГСП, 1-й Рижский пер., 2, издательство «Мир».

Научное издание

КИБЕРНЕТИЧЕСКИЙ СБОРНИК

Ст. научный редактор А. А. Бряндинская
Мл. научный редактор Т. А. Денисова
Художник Н. К. Сапожников
Художественный редактор В. И. Шаповалов
Технический редактор В. П. Снзова
Корректоры Л. Д. Панова, Т. П. Подгорная, А. Ф. Рыбальченко

ИБ № 5941

Сдано в набор 03.01.86. Подписано к печати 01.07.86. Формат 60×90¹/₁₆. Бумага кн.-журнальная имп. Печать высокая. Гарнитура литературная. Объем 6 бум. л. Усл. печ. л. 12. Усл. кр.-отт. 12. Уч.-изд. л. 11,23. Изд. № 1/4928. Тираж 2700 экз. Зак. 30. Цена 2 р. 10 к.

ИЗДАТЕЛЬСТВО «МИР» 129820, ГСП, Москва, И-110, 1-й Рижский пер., 2.

Ленинградская типография № 2 головное предприятие ордена Трудового Красного Знамени Ленинградского объединения «Техническая книга» им. Евгении Соколовой Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли. 198052, г. Ленинград, Л-52, Измайловский проспект, 29.