Кибернетический сборник

НОВАЯ СЕРИЯ

ВЫПУСК

10

Сборник переводов

Под редакцией

А. А. ЛЯПУНОВА и О. Б. ЛУПАНОВА

ИЗДАТЕЛЬСТВО «МИР»

Москва 1973

УДК 519.95

Десятый выпуск серии кибернетических сборников состоит из трех разделов: математические вопросы; вопросы машинного перевода; вычислительные машины и мышление.

В первом разделе представлены работы по теории кодирования, теории графов, синтезу управляющих систем, теории алгоритмов и по математической лингвистике. Особый интерес представляют статьи А. Шёнхаге и В. Штрассена, в которой описывается новый алгоритм умножения n-разрядных чисел, требующий немногим более $n \lg n$ операций, а также работа Н. Дж. А. Слоэна, содержащая обзор последних результатов по конструктивным кодам.

Ко второму разделу относится статья Б. Вокуа, Ж. Вейона, Н. Недобежкина, С. Бургиньона, содержащая изложение работ французских ученых в области машинного перевода.

В третьем разделе помещены статьи Э. Фейгенбаума и Л. Шиклошши, посвященные исследованиям в области искусственного интеллекта. Первая из них содержит подробный обзор работ последних лет в этой области.

Сборник рассчитан на научных работников, инженеров, аспирантов и студентов различных специальностей, занимающихся и интересующихся математической кибернетикой и ее приложениями.

Редакция литературы по математическим наукам

К $\frac{0223-010}{041(01)-78}$

Обзор конструктивной теории кодирования и таблица двоичных кодов с наибольшими известными скоростями

Н. Дж. А. Слоэн [1])

1. ВВЕДЕНИЕ

Эта статья представляет собой обзор современных результатов по построению блоковых кодов для исправления случайных ошибок.

В 1948 г. Шеннон [98] показал, что существуют коды, которые достигают малой вероятности ошибки на скоростях, близких к пропускной способности. В 1952 г. Гилберт [31] получил нижнюю границу для d/n для наилучших кодов с заданной скоростью. С тех пор было потрачено много усилий на то, чтобы построить сколь угодно длинные коды, которые лежат или подходят близко к границе Гилберта, но попытки пока не увенчались успехом (исключение составляют коды со скоростями, стремящимися к 0 или 1). Для умеренных длин блока, однако, были открыты многие хорошие коды. Хорошо известными среди них являются коды Рида — Маллера (РМ) ([79], [93], [B1, гл. 15]), Боуза — Чоудхури — Хоквингема (БЧХ) ([19], [45], [B1, гл. 7 и 10]) и квадратично-вычетные (КВ) коды ([B1, гл. 15], [111, § 4.4]). Систематическое описание этих и других кодов, открытых до 1968 г., можно найти в [B1].

В этой статье будут описаны некоторые достижения теории кодирования, которые имели место после появления работы [B1]. Мы сосредоточим внимание на тех работах, которые посвящены построению новых кодов или представляют новые свойства ранее известных кодов. Важными темами, которые не рассматриваются здесь, являются нумераторы весов кодов [7, 8, 10, 14, 15, 32, 41, 53—57, 73, 81, 86, 94, 100, 106, 114], классификация смежных классов кодов [9, 16, 94, 103], методы декодирования и исправление пакетов ошибок [20, 85, 95, 109, 110], установление синхронизации [105], кодирование для источника и теория передачи с заданной мерой искажения [6, 47]. См.

---

[1]) Sloane N. J. A., A survey of constructive coding theory, and a table of binary codes of highest known rate, *Discrete Mathematics*, 3, № 1, 2, 3 (1972), 265—294.

также недавний обзор Гёталса [35], который шире, чем эта статья, и книгу Ван Линта [111]. Нам не представилась возможность ознакомиться с русскими работами и поэтому отсылаем читателя к обзорам Каутца, Левитта [58] и Добрушина [30].

Статья построена следующим образом. Раздел 2 посвящен циклическим и связанным с ними кодам, в том числе БЧХ, неприводимым, совершенным, групповым абелевым кодам, кодам Гоппы, Сривэставы и циркулянтным кодам. Раздел 3 посвящен нелинейным кодам и кодам, которые строятся на основе объединения других кодов. В разд. 4 приводятся таблицы, содержащие наилучшие из известных автору двоичных кодов. В конце статьи приведена обширная библиография. Краткий предварительный вариант этой статьи был опубликован в [102].

2. ЦИКЛИЧЕСКИЕ КОДЫ

Большинство кодов, рассмотренных к настоящему времени, были линейными и циклическими кодами по той причине, что такие коды более просты при реализации и анализе.

Линейный (n, k)*-код* $\mathscr{C}$ над полем $\mathrm{GF}(q)$ состоит из q^k векторов (которые называются *кодовыми словами*) длины n над $\mathrm{GF}(q)$, таких, что (a) покомпонентная в $\mathrm{GF}(q)$ сумма любых двух кодовых слов снова является кодовым словом и (b) покомпонентное произведение любого кодового слова на любой элемент из $\mathrm{GF}(q)$ также является кодовым словом. *Избыточностью* кода является $r = n - k$ и *скоростью* $R = k/n$. *Минимальное расстояние* обозначается через d.

Дуальный код $\mathscr{C}^\perp$ для кода $\mathscr{C}$ содержит все векторы u длины n над $\mathrm{GF}(q)$, такие, что $u \cdot v = 0$ для всех $v \in \mathscr{C}$, скалярное произведение вычисляется в $\mathrm{GF}(q)$. Таким образом, $\mathscr{C}^\perp$ представляет собой линейный $(n, n-k)$-код. Если $\mathscr{C} = \mathscr{C}^\perp$, то $\mathscr{C}$ *— самодуальный код.*

Связь самодуальных кодов с известной нерешенной проблемой геометрии дана в [74].

Код называется *укороченным*, если он образуется с помощью выбрасывания всех кодовых слов, кроме тех, которые имеют заранее указанные значения некоторых компонент, и дальнейшего выбрасывания этих компонент [B1, стр. 336].

Циклическим кодом является линейный код, обладающий тем свойством, что циклический сдвиг любого кодового слова также является кодовым словом. БЧХ, КВ и укороченные РМ-коды являются циклическими.

2.1. Являются ли длинные циклические коды плохими? Гилберт [31] обнаружил, что существуют линейные коды со сколь угодно большой длиной и фиксированной скоростью $R = k/n$,

для которых d/n больше некоторой положительной константы. Действительно, Кошелев [59] и Козлов [60] показали, что большинство линейных кодов лежат на границе Гилберта.

Вместе с тем для БЧХ-кодов с заданной скоростью R отношение $d/n \to 0$ при $n \to \infty$ ([67], [B1, гл. 12]). Берлекэмп [12] недавно установил, что для БЧХ-кодов действительно

$$d \sim \frac{2n \ln R^{-1}}{\log n}$$

при $n \to 0$. Однако не известно, являются ли также плохими другие длинные циклические коды.

Берман [18] показал, что циклические коды с фиксированной скоростью и с длинами блока n, которые делятся на фиксированное множество простых чисел (и только на эти простые числа), имеют ограниченное минимальное расстояние.

Казами [52] установил, что хорошие линейные коды не могут быть слишком симметричными, доказав, что любой код с заданным отношением d/n, который является инвариантным относительно аффинной группы, должен иметь скорость $R \to 0$ при $n \to \infty$. (Это относится к БЧХ-кодам.)

Совсем недавно Мак-Элис [77] показал, что не только одна симметрия делает код плохим, доказав, что существуют сколь угодно длинные блоковые коды (не обязательно линейные), которые инвариантны относительно больших групп перестановок и которые удовлетворяют границе Гилберта. Велдон [23], [118] обнаружил также, что существуют очень длинные (но не сколь угодно длинные) циркулянтные и квазициклические коды, которые удовлетворяют границе Гилберта. (Квазициклическим кодом называется линейный код, обладающий тем свойством, что циклический сдвиг любого кодового слова на определенное заранее заданное число позиций является также кодовым словом. Циркулянтные коды определяются в § 2.9.) Доказательство Велдона могло бы быть применимо к сколь угодно длинным кодам, если бы была доказана гипотеза о существовании бесконечного числа простых чисел, для которых 2 является примитивным корнем.

Казами [52] и Чен [22] показали, что существуют сколь угодно длинные укороченные циклические коды, которые удовлетворяют границе Гилберта.

Томпсон (см. [75]) обнаружил, что самодуальные линейные двоичные коды, в которых все веса делятся на 4, удовлетворяют границе Гилберта.

Эти результаты, грубо говоря, означают, что хорошее семейство кодов может быть линейным или обладать многими симметриями, но не может обладать этими свойствами одновременно.

2.2. БЧХ-коды. Напомним определение БЧХ-кода. Пусть q — степень простого числа; пусть m — порядок q по модулю n, и пусть α — примитивный корень n-й степени из единицы в $GF(q^m)$. Тогда БЧХ-*код* длины n, имеющий конструктивное расстояние $d = d_{БЧХ}$ и символы из $GF(q)$, описывается проверочной матрицей

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \dots & \alpha^{2(n-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \alpha^{d-1} & \alpha^{2(d-1)} & \dots & \alpha^{(n-1)(d-1)} \end{pmatrix}.$$

Если $n = q^m - 1$, то код называется *примитивным*. Согласно БЧХ-границе, любой такой код имеет истинное минимальное расстояние

$$d_{\min} \geqslant d_{БЧХ}.$$

В работах [55], [B1, стр. 303] было высказано предположение, что для примитивных БЧХ-кодов $d_{\min} = d_{БЧХ}$, но в 1969 г. Казами и Токура [K2] показали, что для $m > 6$, $m \neq 8$, 12, существуют двоичные примитивные БЧХ-коды с длиной $n = 2^m - 1$, для которых

$$d_{\min} > d_{БЧХ}.$$

Вместе с тем Берлекэмп [8] показал, что если расширенный двоичный БЧХ-код длины $n = 2^m$ имеет $d_{БЧХ} = 2^{m-1} - 2^i$ для некоторого $i \geqslant \frac{1}{2}m - 1$, то $d_{\min} = d_{БЧХ}$. Результаты [8] получили дальнейшее обобщение в работе Казами [53]. Но точное определение условий на n и $d_{БЧХ}$ для того, чтобы $d_{\min} = d_{БЧХ}$, остается нерешенной проблемой.

Леонтьев [66] установил, что БЧХ-код длины $n = 2^m - 1$ с конструктивным расстоянием $2t + 1$ не является квазисовершенным при $2 < t < \sqrt{n}/\ln n$ и $m \geqslant 7$.

Сидельников [100] показал, что для $t < \sqrt{n}/10$ число слов веса w в двоичном БЧХ-коде с конструктивным расстоянием $2t + 1$ равно $(n+1)^{-t}\binom{n}{w}(1 + \varepsilon)$, где $|\varepsilon| < Cn^{-0,1}$ для большинства значений w.

Андерсон ([1], [111, стр. 127]) получил следующую границу для дуального БЧХ-кода, используя глубокие теоретико-числовые результаты Вейля, Карлица и Утиямы. Минимальное расстояние дуального двоичного БЧХ-кода длины $n = 2^m - 1$ с расстоянием $d_{БЧХ} = 2t + 1$ по меньшей мере равно

$$2^{m-1} - 1 - (t-1)^{\frac{1}{2}m}.$$

2.3. Расширения БЧХ-кодов. Вулф [117] показал, что к проверочной матрице БЧХ-кода могут быть добавлены два столбца для того, чтобы получить новую проверочную матрицу

$$H' = \begin{pmatrix} 1 & 0 & \\ 0 & 0 & \\ \cdot & \cdot & H \\ 0 & 1 & \end{pmatrix}$$

с сохранением минимального расстояния кода. В некоторых случаях избыточность также не меняется, и при этом получается новый код с

$$n' = n + 2, \quad k' = k + 2, \quad r' = r, \quad d' = d.$$

Это происходит, например, когда исходный код является кодом Рида — Соломона над $\mathrm{GF}(q)$ с $n = q - 1$ и $r = d - 1$. При этом параметры нового кода равны $n' = q + 1$ и $r' = d' - 1 = d - 1$.

Легко показать, что для любого кода $d \leqslant r + 1$. Коды с $d = r + 1$ называются кодами *с достижимым максимальным расстоянием* или кодами с ДМР (см. [101], [B1, стр. 317], [111, стр. 72]). Такие коды также были названы *оптимальными*. Коды Рида — Соломона являются кодами с ДМР, так же как и новое семейство дважды расширенных кодов Рида — Соломона.

Ассмус и Мэттсон [3] установили, что очень часто встречаются коды с ДМР, у которых длина блока n равна простому числу. Они доказали, что любой циклический код с простой длиной n над $\mathrm{GF}(p^i)$ является кодом с ДМР при всех i и для всех (кроме конечного числа) простых p.

Вулф [118] дал дальнейшее расширение БЧХ-кодов, заменив α в H' на $(m \times m)$-матрицу A над $\mathrm{GF}(q)$, где

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 & \dots & 1 \\ a_0 & a_1 & a_2 & \dots & a_{m-1} \end{pmatrix}$$

и где $M(x) = x^m - a_{m-1}x^{m-1} - \dots - a_0$ является минимальным многочленом для α над $\mathrm{GF}(q)$. Назовем новую проверочную матрицу H''. Тогда, если H порождает примитивный БЧХ-код над $\mathrm{GF}(q)$ длины $n = q^m - 1$ с конструктивным расстоянием

$d_{\text{БЧХ}}$ и избыточностью $r = m(d_{\text{БЧХ}} - 1)$, т. е. БЧХ-код с максимальной избыточностью, то H'' является проверочной матрицей для некоторого кода над GF(q) с $n' = m(q^m + 1)$, $r = m(d_{\text{БЧХ}} - 1)$ и $d \geqslant d_{\text{БЧХ}}$. Таким образом, скорость кода была значительно увеличена.

Например, если исходный код над GF(5) есть код с $n = 5^2 - 1 = 24$, $d_{\text{БЧХ}} = 5$, $k = 16$, $R = 0{,}67$, то новый код имеет $n = 52$, $k = 44$, $d \geqslant 5$ и $R = 0{,}87$. Другие расширения БЧХ-кодов можно найти в § 3.3.

2.4. Минимальное расстояние циклических кодов. БЧХ-граница для циклического кода гарантирует, что если порождающий многочлен $g(x)$ имеет $d_{\text{БЧХ}} - 1$ последовательных корней, то минимальное расстояние $d \geqslant d_{\text{БЧХ}}$.

Гёталс [33] и Казами [51] улучшили БЧХ-границу для кодов составной длины. Хартманн [38—43] дал много дальнейших обобщений для БЧХ-границы, в том числе обобщение результатов Казами. Приведем лишь две теоремы Хартманна. Пусть β обозначает примитивный корень n-й степени из единицы. Первой является граница для минимального нечетного веса.

Теорема. *Пусть $k|n$. Если для некоторого $\bar{d} \leqslant n/k$ имеет место $g(\beta^{ki}) = 0$ для всех $i = 1, 2, \ldots, \bar{d}$, то минимальный нечетный вес по меньшей мере равен $\bar{d}$.*

Пример. Согласно БЧХ-границе, БЧХ(33, 13)-код имеет $d_{\text{БЧХ}} = 5$, а также $d_{\text{ЧЕТ}} \geqslant 10$. Но, согласно теореме, $d_{\text{НЕЧЕТ}} \geqslant 11$, и поэтому минимальный вес $\geqslant 10$.

Вторая теорема представляет собой пример обобщения БЧХ-границы, сделанного Хартманном для случая, когда $g(x)$ имеет несколько множеств последовательных корней.

Теорема. *Если $g(\beta^{m \cdot + i + \delta(j-1)}) = 0$ для $i = 0, 1, 2, \ldots, d-1$ и $j = 1, 2, \ldots, r$, где $(\delta, n) = 1$, так что $g(x)$ имеет r множеств по $d - 1$ корней в каждом, то $d_{\min} \geqslant d + r$.*

Казами и Токура [К2] показали, что для любого четного $m \geqslant 6$ существуют двоичные циклические коды длины $2^m - 1$, имеющие больше кодовых слов, чем соответствующие БЧХ-коды. Первым таким примером является циклический (63, 28)-код с $d = 15$, в то время как БЧХ(63, 24)-код также имеет $d = 15$.

Чен [С3] использовал вычислительную машину IBM 360/50 для того, чтобы вычислить минимальное расстояние всех двоичных циклических кодов с длинами $\leqslant 65$. Он нашел, что есть три кода длины 63, имеющие больше кодовых слов, чем соответствующие БЧХ-коды. Это только что упомянутый (63, 28)-код, (63, 46)-код с $d = 7$, ранее указанный Питерсоном [86], и

$(63, 21)$-код с $d = 18$. БЧХ-коды, которые ближе всего к двум последним, — это $(63, 45)$-код с $d = 7$ и $(63, 18)$-код с $d = 21$.

2.5. Неприводимые циклические коды. Циклический код над $\mathrm{GF}(q)$ называется *неприводимым*, если его проверочный многочлен $h(x)$ неприводим над $\mathrm{GF}(q)$ [111, стр. 45]. Простейшими примерами неприводимых кодов являются регистровые коды максимальной длины $(2^m - 1, m)$ (известные также как укороченные РМ-коды первого порядка).

Баумерт, Мак-Элис и Рамсей [5], [78], обобщая более раннюю работу Делзарте и Гётелса [29], дали метод отыскания нумераторов весов всех неприводимых циклических кодов. Так, например, пусть N — фиксированное нечетное число и k — наименьшее положительное число, такое, что $2^k \equiv 1 \pmod N$. Тогда существуют двоичные неприводимые циклические $(n = (2^{km} - 1)/N, km)$-коды $\mathscr{C}_m$ для $m = 2, 3, \ldots$. Каждый $\mathscr{C}_m$ состоит из нулевого вектора и N циклов по n кодовых слов в каждом. Пусть $w_0, w_1, \ldots, w_{N-1}$ являются весами этих циклов. Тогда порождающая функция $w_0 + w_1 y + \ldots + w_{N-1} y^{N-1}$ задается выражением

$$2^{m-1} (E(y))^m \pmod{y^N - 1},$$

где $E(y)$ не зависит от m. Так, например, когда $N = 7$, получаются $(9, 6)$, $(73, 9)$, $(585, 12)$, $(4681, 15)$, ...-коды с минимальными расстояниями, соответственно равными 2, 28, 280, 2320, ... (Полные распределения весов имеются в [78].)

Распределения весов нескольких других классов циклических кодов были даны Оганесяном и Ягджаном [81] и [92]. Упомянем лишь один из них, который состоит из кодов с проверочными многочленами вида $h(x) = \prod_{i=0}^{f} p_i(x)$, где $p_0(x)$ — неприводимый многочлен степени k_0 и периода e_0; $m_0 = (2^{k_0} - 1)/e_0$ — простое число; 2 является примитивным корнем m_0; $p_i(x)$ — примитивный многочлен степени k_i и периода e_i, а числа e_i — взаимно простые.

2.6. Совершенные коды. Код над $\mathrm{GF}(q)$, исправляющий e ошибок, называется *совершенным*, если каждый его вектор находится на расстоянии самое большее e от ближайшего кодового слова.

Примерами совершенных кодов являются различные тривиальные коды, содержащие 1, 2 или q^n кодовых слов; коды Хемминга над любым полем с $d = 3$; два кода Голея: $(11, 6)$-код над $\mathrm{GF}(3)$ с $d = 5$ и $(23, 12)$-код над $\mathrm{GF}(2)$ с $d = 7$.

Давно высказанное предположение о том, что не существуют никаких других совершенных кодов над конечными полями,

было недавно доказано Титавайненом, который использовал ранее вышедшие работы Ллойда и Ван Линта [107], [108], [112].

Расширенные $(12, 6)$- и $(24, 12)$-коды Голея имеют много важных комбинаторных свойств. Их группами симметрии являются группы Матье M_{12} и M_{24}; их векторы малого веса образуют системы Штейнера $S(5, 6, 12)$ и $S(5, 8, 24)$; из них могут быть построены решетки Λ_{12} и Λ_{24} (решетки Лича) [62], [65]. Был доказан ряд теорем единственности. Плесс [87] доказала единственность кодов Голея; Стантон [104] — единственность групп Матье; Витт [116] — единственность ассоциированных систем Штейнера и Конвей [24], [25] — единственность решетки Лича.

Гёталс [34] показал, что код Нордстрома — Робинсона (§ 3.2) содержится в $(24, 12)$-коде. Берлекэмп [11] изучил группы симметрии принципиальных подкодов $(24, 12)$-кода.

Поскольку код Нордстрома — Робинсона является первым кодом множества нелинейных кодов Препараты, исправляющих двойные ошибки (§ 3.2), то естественно поставить вопрос, могут ли другие коды быть обобщены так, чтобы дать коды, аналогичные коду Голея. Препарата [91] показал, однако, что это невозможно сделать по крайней мере одним методом.

Турин [2] показал, что $(24, 12)$-код Голея может быть получен как множество векторов вида $(a + x, b + x, a + b + x)$, $a, b \in \mathscr{C}_1$, $x \in \mathscr{C}_2$, где $\mathscr{C}_1$ и $\mathscr{C}_2$ — два различных РМ-кода первого порядка. Та же самая конструкция была использована в [S4] при получении бесконечного множества линейных кодов с $d/n = \frac{1}{3}$. Первые три кода из этого множества представляют собой $(24, 12)$-код Голея и $(48, 15)$-код с $d = 16$ и $(96, 18)$-код с $d = 32$. С увеличением длины скорость стремится к нулю. Обобщение $(12, 6)$-кода Голея описано в следующем параграфе.

Паркер и Николай [82] описали неудачный поиск простых транзитивных групп, аналогичных группам Матье.

2.7. Абелевы групповые коды. Пусть $\mathscr{C}$ будет двоичным циклическим (n, k)-кодом. Если кодовые слова представлены многочленами $c_0c_1 \ldots c_{n-1} \leftrightarrow c_0 + c_1x + \ldots + c_{n-1}x^{n-1}$, то, как хорошо известно, кодовые слова $\mathscr{C}$ образуют идеал в кольце многочленов по модулю $x^n - 1$ [P1, гл. 8].

Мак-Вильямс [71], [72], Берман [17], [18] и другие [21], [27] изучали следующее обобщение циклических кодов. Пусть $G = \{g_1, \ldots, g_n\}$ — мультипликативная абелева группа, и пусть R обозначает множество всех формальных сумм

$$c_1g_1 + c_2g_2 + \ldots + c_ng_n, \quad c_i = 0 \text{ или } 1,$$

с очевидными сложением и умножением. Ясно, что R — векторное пространство размерности n над GF(2). Идеал $\mathscr{A}$ является

линейным подпространством R, если для любых $A \in \mathscr{A}$, $g \in G$ имеет место $gA \in \mathscr{A}$. При этом $\mathscr{A}$ является естественным обобщением циклического кода и называется абелевым групповым кодом.

Многие свойства циклических кодов переносятся на абелевы групповые коды, как, например, существование порождающего кодового слова, умножением на которое порождается код.

Берман [18] показал, что при фиксированной скорости и для длин блоков n, которые делятся на фиксированное множество простых чисел (и только на эти простые), абелевы групповые коды в пределе при $n \to \infty$ имеют большее минимальное расстояние, чем циклические коды.

2.8. Коды Гоппы и Сривэставы. Недавно Гоппа [G4] описал новый класс линейных нециклических кодов, часть из которых удовлетворяет границе Гилберта.

Пусть целые числа m, t заданы и удовлетворяют неравенствам $3 \leqslant m < mt < 2^m$. Пусть

$$Z = \{z \in \mathrm{GF}(2^{mt}) \mid \text{степень минимального многочлена } z \text{ есть } mt\},$$

и пусть α — примитивный элемент $\mathrm{GF}(2^m)$. Тогда для любого $z \in Z$ двоичный код Гоппы $\mathscr{G}(m, t, z)$ является $(n = 2^m, k \geqslant 2^m - mt)$-кодом, проверочная $(mt \times 2^m)$-матрица которого есть

$$H = \left[\frac{1}{z-0}, \frac{1}{z-1}, \frac{1}{z-\alpha}, \ldots, \frac{1}{z-\alpha^{2^m-2}}\right].$$

Гоппа показал, что минимальное расстояние $\mathscr{G}(m, t, z)$ (1) по меньшей мере равно $2t+1$ для всех $z \in Z$ и (2) достигает границы Гилберта для некоторых $z \in Z$. К сожалению, не известно, как выбрать $z \in Z$ для того, чтобы это имело место.

Коды Сривэставы [B1, § 15.1] похожи на коды Гоппы. Хелгерт [H2], [H3] недавно нашел ряд хороших кодов Сривэставы.

2.9. Циркулянтные коды. В 1964 г. Лич [61] показал, что порождающая матрица для (23, 12)-кода Голея может быть записана в виде

$$\left(\begin{array}{ccc|c} 1 & & & \\ & \ddots & & C \\ & & 1 & \\ \hline & & 1 & 1 \ \ldots \ 1 \end{array}\right),$$

где C — циркулянтная матрица, т. е. такая матрица, каждая строка которой является циклическим сдвигом предыдущей строки на одну позицию. В этом случае первая строка C может

быть взята в виде 1 1 0 1 1 1 0 0 0 1 0 с единицами на позициях 0 и квадратичных вычетах числа 11.

Затем в 1965 г. вышла важная работа Карлина ([К1], см. также [46], [49], [50]), в которой было найдено, что циркулянты порождают большое число двоичных кодов, многие из которых имеют бóльшую скорость, чем наилучшие из ранее известных кодов. Примерами являются (27, 14), (30, 16), (34, 12) и (53, 14)-коды, имеющие минимальные расстояния, соответственно равные 7, 7, 11 и 17.

Этот подход упростил также вычисление минимального расстояния, и Карлин получил возможность определить минимальное расстояние ряда двоичных квадратично-вычетных кодов, так, например, КВ(79, 40)-кода, для которого $d = 15$, и КВ-(89, 45)-кода, для которого $d = 17$. Карлин также показал, что КВ-коды длины 103 и 107 имеют минимальное расстояние, равное 19.

Плесс [88], [89] построила самодуальные $(2q + 2, q + 1)$-коды над GF(3) для всех нечетных простых степеней $q \equiv -1 \pmod 3$. Они являются циркулянтными кодами с порождающей матрицей вида

$$\left(\begin{array}{cccc|cccc} 1 & & & & 1 & & & \\ & 1 & & & 1 & & C & \\ & & 1 & & 1 & & & \\ & & & 1 & 0 & 1 & 1 & 1 \end{array}\right),$$

где C — циркулянтная матрица. Первыми пятью из них являются (12, 6) (троичный код Голея), (24, 12), (36, 18), (48, 24) и (60, 30)-коды с минимальными расстояниями 6, 9, 12, 15 и 18 соответственно. Эти пять кодов имеют скорость 1/2 и $d = \frac{1}{4}n + 3$. Но, к сожалению, дальнейшие коды в этом классе имеют меньшие расстояния. Тем не менее циркулянтные коды представляют собой многообещающую область исследования.

3. НЕЛИНЕЙНЫЕ КОДЫ И КОДЫ, СТРОЯЩИЕСЯ С ПОМОЩЬЮ КОМБИНАЦИИ ДРУГИХ КОДОВ

Так как число кодовых слов в нелинейном коде не обязательно должно быть степенью объема алфавита, то удобно ввести новое обозначение.

(n, M, d)-код $\mathscr{C}$ представляет собой множество M кодовых слов длины n с символами из GF(q) и с минимальным расстоянием d. Размерность этого кода равна $k = \log_q M$; избыточность $r = n - \log_q M$ и скорость $R = k/n$. Здесь k и r не обязательно являются целыми числами.

Смежным классом $\mathscr{C}$ является произвольный сдвиг $a+\mathscr{C}$ кодовых слов $\mathscr{C}$ (где a — произвольный вектор длины n). Если $\mathscr{C}$ — линейный код, то два смежных класса $\mathscr{C}$ либо равны, либо не пересекаются, но это не верно, если $\mathscr{C}$ — нелинейный код.

Говорят, что (n, M, d)-код является *оптимальным*, если он имеет наибольшее возможное число кодовых слов при заданных значениях n и d. Конечно, здесь оптимальность понимается в очень наивном смысле, поскольку при этом не учитываются кодирование и декодирование. Но можно возразить, что если найдены хорошие коды, то методы их реализации будут разработаны позже, как это произошло с БЧХ-кодами [B1, гл. 7].

Разумно ожидать, что оптимальные коды часто будут нелинейными и что даже близкие к оптимальным коды будут иметь сложную структуру. Хорошо известным высказыванием Дж. Л. Мэсси [70] являются слова: «...хорошими кодами могли бы быть только беспорядочные коды» [1]).

Нелинейные коды были успешно использованы для построения плотных сферических упаковок в евклидовом пространстве [63—65].

3.1. Коды, полученные из матриц Адамара и конференционных матриц. *Матрицей Адамара* $\mathscr{H}_n$ размером $n \times n$ является матрица, состоящая из $+1$ и -1, такая, что $\mathscr{H}_n \mathscr{H}_n^t = nI$ (где I — единичная матрица). Замена $+1$ на 0 и -1 на 1 превращает $\mathscr{H}_n$ в *двоичную матрицу Адамара* H_n.

Плоткин [P1, стр. 79], [P3], [B1, стр. 325] показал, что $\left(n, 2n, \frac{1}{2}n\right)$-код, состоящий из строк H_n и их дополнений, является оптимальным. Когда n является степенью 2, код будет (линейным) кодом Рида — Маллера первого порядка, а в других случаях он представляет собой нелинейный код Адамара.

Много других нелинейных кодов можно получить путем комбинирования матриц Адамара. Левенштейн ([L2], см. также [58, стр. 206], [83]) показал, что оптимальные коды для всех d и всех $n \leqslant 2d$ могут быть получены из матриц Адамара (при условии, что требуемые матрицы Адамара существуют), доказав, что такие коды достигают границы Плоткина [P3]. Патель [84] определил оптимальные линейные коды в том же самом диапазоне [2]).

---

[1]) В оригинале игра слов «Massey» и «messy». Английское слово messy означает беспорядочный. — *Прим. перев.*

[2]) Значительно ранее оптимальные линейные коды в диапазоне $n \leqslant 2d$ построил Дж. Вентурини (Проблемы кибернетики, вып. 16, Наука, М., 231—238, 1966). — *Прим. ред.*

Недавно с помощью конференционных матриц были получены нелинейные коды с n чуть большими, чем $2d$ [S2]. *Конференционной* $n \times n$*-матрицей* T_n является матрица, составленная из нулей на диагонали и из ± 1 на всех остальных местах, удовлетворяющая соотношению $T_n T_n' = (n-1)I$. Всегда, когда существует симметричная T_n, может быть построен двоичный нелинейный $\left(n-1,\ 2n,\ \frac{1}{2}(n-2)\right)$- код. Несколькими первыми полученными кодами являются: оптимальный $(9, 20, 4)$-код Юлина [J1]; $(13, 28, 6)$-код, который хуже кода Надлера [80]; $(17, 36, 8)$-код, полученный в [S1], и $(25, 52, 12)$, $(29, 60, 14)$, $(37, 76, 18)$, $(41, 84, 20)$-коды.

3.2. Коды Препараты и Кердока. Нелинейные коды, исправляющие две ошибки, были построены Надлером ([80], $(12, 32, 5)$-код) Грином ([37], $(13, 64, 5)$-код) и Нордстромом и Робинсоном ([N1], $(15, 2^8, 5)$-код). Ван Линт [114] дал простую конструкцию кода Надлера.

Препарата [P4] получил обобщение кода Нордстрома — Робинсона для больших скоростей. Для любого четного $m \geqslant 4$ он построил нелинейный $\left(2^m - 1,\ 2^{2^m - 2m},\ 5\right)$-код. Эти коды являются оптимальными; в них вдвое больше кодовых слов, чем в БЧХ-кодах, исправляющих две ошибки, и для них имеются прямые алгоритмы кодирования и декодирования.

Семаков и Зиновьев [96], [97] и Гёталс и Сновер [36] независимо получили распределение весов для кодов Препараты.

Кердок [K5] дал соответствующее обобщение кода Нордстрома — Робинсона для малых скоростей. Он показал, что для любого четного $m \geqslant 4$ можно взять объединение $(2^m, 2^{m+1}, 2^{m-1})$-кода РМ первого порядка и $2^{m-1} - 1$ его смежных классов и получить нелинейный $\left(2^m,\ 2^{2m},\ 2^{m-1} - 2^{\frac{1}{2}(m-2)}\right)$-код. При $m = 4$ он представляет собой расширенный код Нордстрома — Робинсона; при $m = 6$ он представляет собой $(64, 2^{12}, 28)$-код, содержащий в четыре раза больше кодовых слов, чем наилучший расширенный циклический код с такой длиной и расстоянием.

Коды Препараты и Кердока «дуальны» в том смысле, что их распределения весов удовлетворяют тождеству Мак-Вильямс [B1, стр. 401]. Причина этого еще не ясна.

Следующие четыре параграфа описывают конструкции, в которых для образования новых кодов комбинируются два, три или четыре кода.

3.3. Конструкции *X* и *X4*. Конструкция X комбирирует три кода для получения четвертого. Предположим, что заданы (n_1, M_1, d_1)-код $\mathscr{C}_1$ и $(n_1, M_2 = bM_1,\ d_2)$-код $\mathscr{C}_2$ с условием, что

$\mathscr{C}_2$ является объединением b непересекающихся смежных классов $\mathscr{C}_1$, т. е.

$$\mathscr{C}_2 = (x_1 + \mathscr{C}_1) \cup (x_2 + \mathscr{C}_1) \cup \ldots \cup (x_b + \mathscr{C}_1)$$

для некоторого множества векторов $S = \{x_1, x_2, \ldots, x_b\}$. Пусть

$$\mathscr{C}_3 = \{y_1, y_2, \ldots, y_b\}$$

будет каким-либо (n_3, b, Δ)-кодом.

Обозначим через π произвольную перестановку $\{1, 2, \ldots, b\}$, так что $x_i \to y_{\pi(i)}$ определяет взаимно однозначное отображение S на $\mathscr{C}_3$.

Новый код теперь определяется как

$$\mathscr{C}_4 = (x_1 + \mathscr{C}_1, y_{\pi(1)}) \cup (x_2 + \mathscr{C}_1, y_{\pi(2)}) \cup \ldots \cup (x_b + \mathscr{C}_1, y_{\pi(b)}).$$

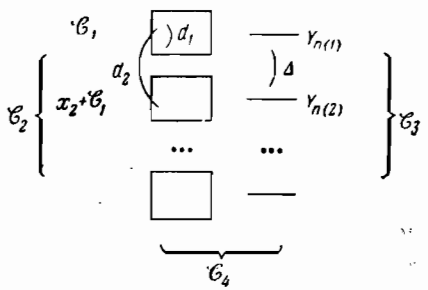


Рис. 1.

Иначе говоря, $\mathscr{C}_2$ разбито на смежные классы $\mathscr{C}_1$, и различные кодовые слова из $\mathscr{C}_3$ связываются с каждым смежным классом. (См. рис. 1.)

Тогда $\mathscr{C}_4$ является $(n_1 + n_3, M_2, d = \min\{d_1, d_2 + \Delta\})$-кодом. Аналогично конструкция $X4$ комбинирует четыре кода для получения пятого. Для более подробного ознакомления см. [S4] и дальнейшие примеры.

Пример 1. Пусть $\mathscr{C}_1$ — код Препараты, $\mathscr{C}_2$ — код Хэмминга, $\mathscr{C}_3$ — код с четными весами. Тогда, убедившись в том, что код Хэмминга является объединением смежных классов кода Препараты, можно получить $\left(2^m + m - 1, 2^{2^m - m - 1}, 5\right)$-коды при $m \geqslant 4$. С помощью конструкции $X4$ можно получить даже большее и, в частности, расширить код Препараты путем добавления $\sqrt{n+1}$ информационных символов за счет добавления одного проверочного символа [S4].

Пример 2. Используя БЧХ-коды, можно получить новые коды, которые имеют по меньшей мере такое же число кодовых слов, что и в конструкции Андрианова — Сасковца [B1, стр. 333]. В некоторых случаях БЧХ-коды, исправляющие e ошибок, могут быть расширены с помощью добавления около $n^{1/e}$ информационных символов за счет добавления одного проверочного символа [S4].

3.4. Конструкции *Y*1, *Y*2, *Y*3. Следующие конструкции были предложены Гёталсом [34].

Конструкция Y1. Пусть $\mathscr{C}_1$ — линейный $(n, 2^{k_1}, d_1)$-код, и пусть $\mathscr{C}_2$ — его дуальный $(n, 2^{r_1}, d_2)$-код с координатами, выбранными так, чтобы имелось слово минимального веса $1\ldots10\ldots0$ в $\mathscr{C}_2$. Пусть S — подгруппа $\mathscr{C}_1$, в которой первые $d_2 - 1$ координат являются нулевыми. Тогда d_2-е координаты S также нули. Если выбросить d_2 начальных нулей из S, то получится линейный

$$(n - d_2,\ 2^{k_1-d_2+1},\ d_1)\text{-код.}$$

Конструкция Y2. Пусть T является объединением S и всех $d_2 - 1$ смежных классов S в $\mathscr{C}_1$ с лидерами смежных классов 110^{n-2}, 1010^{n-3}, ..., $10^{d_2-2}10^{n-d_2}$. Выбрасывая первые d_2 координат из T, получим нелинейный

$$(n - d_2,\ d_2 2^{k_1-d_2+1},\ d_1 - 2)\text{-код.}$$

Конструкция Y3. Выбирая все смежные классы с лидерами веса 2, получим нелинейный

$$\left(n - d_2,\ \left(1 + \binom{d_2}{2}\right)2^{k_1-d_2+1},\ d_1 - 4\right)\text{-код.}$$

Много примеров кодов, полученных согласно этим конструкциям, приведено в [S4].

3.5. Конструкция *Z*. Она комбинирует два кода для получения третьего [P3], [44], [S1], [68].

Пусть $\mathscr{C}_1 = (n_1, M_1, d_1)$ и $\mathscr{C}_2 = (n_2, M_2, d_2)$ — произвольные коды над $\mathrm{GF}(q)$. Пусть σ обозначает нулевой вектор длины $|n_1 - n_2|$. Тогда новый код $\mathscr{C}_3$ определяется следующим образом:

(i) если $n_1 \leqslant n_2$, то $\mathscr{C}_3 = \{(x, (x, \sigma) + y) \mid x \in \mathscr{C}_1, y \in \mathscr{C}_2\}$ является $(n_1 + n_2, M_1M_2, d = \min(2d_1, d_2))$-кодом (в определении $\mathscr{C}_3$ запятая обозначает присоединение, а $+$ обозначает векторное сложение в $\mathrm{GF}(q)$);

(ii) если $n_1 > n_2$, то $\mathscr{C}_3 = \{(x, x + (y, \sigma)) \mid x \in \mathscr{C}_1,\ y \in \mathscr{C}_2\}$ является $(2n_1, M_1M_2, d = \min(2d_1, d_2))$-кодом.

Код $\mathscr{C}_3$ является линейным, если линейны $\mathscr{C}_1$ и $\mathscr{C}_2$. В [S1] дан ряд применений этой конструкции, в том числе для по-

Таблица 1

$\mathscr{C}_1$			$\mathscr{C}_2$			$\mathscr{C}_3$		
n_1	r_1	d_1	n_2	r_2	d_2	n	r	d
23	11	7	11	4,830	4	34	15,830	7
19	12	8	19	18	19	38	30	16
14	13	14	264	101	27	278	114	27

строения бесконечного класса нелинейных кодов, исправляющих одну ошибку и содержащих больше кодовых слов, чем укороченные коды Хэмминга. Другие примеры, типичные из которых приведены в табл. 1, можно найти в табл. 2.

3.6. Циклические коды Ассмуса и Мэттсона со скоростью 1/3. Пусть p будет простым числом вида $8N + 5$, для которого 2 является примитивным корнем (например, $p = 5, 13, 29, 37, \ldots$). Ассмус и Мэттсон [4] показали, как произвести присоединение трех различных вариантов $(p, p-1)$-кода с четными весами для получения линейного двоичного циклического $(3p, p-1)$-кода, обозначаемого через $3E$, с минимальным расстоянием, равным по меньшей мере $2\sqrt{3p}$. Пусть $3E^+$ — циклический $(3p, p)$-код, состоящий из кодовых слов $3E$ вместе с их дополнениями. Первыми несколькими примерами $3E^+$ являются $(15, 5)$, $(39, 13)$, $(87, 29)$ и $(111, 37)$-коды с d, равным 7, 12, 24 и 24 соответственно.

3.7. Другие конструкции. Другие методы построения кодов приведены в [26], [28], [76], [99]. Однако полученные там коды содержат меньше или, в лучшем случае, столько же кодовых слов, сколько и известные циклические коды.

4. ТАБЛИЦА ЛУЧШИХ ИЗ ИЗВЕСТНЫХ В НАСТОЯЩЕЕ ВРЕМЯ КОДОВ

4.1. Для заданных значений длины n и минимального расстояния d через M обозначается максимальное число кодовых слов какого-либо двоичного (n, M, d)-кода (линейного или нелинейного), который в настоящее время известен (автору). Таблица 2 дает избыточность $r = n - \log_2 M$ этого кода как функцию n и d для всех $n \leqslant 512$ и $d \leqslant 30$. Эти коды, представляя сами по себе значительный теоретический интерес, дают основу для обсуждения новых кодов и, как нижние границы к наиболее плотным возможным кодам, дополняют таблицу Джонсона для верхних границ [48]. Ранее опубликованные таблицы кодов могут быть найдены в [B1], [C1], [C3], [G3], [L1], [P1], [P3], [W1] и [86].

4.2. Типы кодов. Коды классифицируются следующим образом:

B = код Боуза — Чоудхури — Хоквингема [B1, гл. 7];
C = циклический линейный код [P1, гл. 8];
D = код Гоппы ([G4] и § 2.8 настоящей статьи);
G = групповой или линейный код [P1, гл. 3];
H = код Адамара [L2];

J = код из конференционной матрицы [S2];
K = циркулянтный код [K1];
N = нелинейный код;
P = код Нордстрома — Робинсона — Препараты [N1], [P4];
Q = квадратично-вычетный код ([B1, § 15.2], [111, § 4.4]);
R = код Рида — Маллера [B1, § 15.3];
S = код Сривэставы [B1, § 15.1], [H2];
XA, XC, XP = коды из конструкции X, примененные к БЧХ-кодам (обобщенная конструкция Андрианова — Сасковца), к циклическим кодам и к кодам Препараты соответственно ([S4] и § 3.3 выше);
X4 = коды из конструкции $X4$ ([S4] и § 3.3);
Y1, Y2, Y3 = коды из конструкций $Y1$, $Y2$, $Y3$ ([S4] и § 3.4 настоящей статьи);
Z = коды из конструкции Z ([S1] и § 3.5 настоящей статьи).

Типы B, C, D, G, K, Q, R, S, XA, XC, Y1 являются линейными, H, J, N, P, XP, Y2, Y3 являются нелинейными, а X4, Z могут быть линейными или нелинейными.

В таблице 2 коды, для которых дана ссылка [S3], являются новыми. За исключением одного случая, все они являются примерами конструкций, упомянутых в тексте. Исключение составляет (85, 18)-код с $d = 25$, который был получен с помощью использования конструкции 39 Хатчера [44].

4.3. Так как (n, M, d)-код при нечетном d эквивалентен $(n+1, M, d+1)$-коду, необходимо лишь дать коды с нечетными d. В (n, M, d)-коде можно выбросить из рассмотрения i компонент, в результате чего получатся $(n-i, M, d-i)$-коды при $0 < i < d$, или укоротить (n, M, d)-код, в результате чего получатся $(n-i, M2^{-i}, d)$-коды при $0 < i < \log_2 M$. В таблице 2 все такие модифицированные коды названы так же, как и исходный код. Можно представлять себе (n, M, d)-код с избыточностью r, как $(n+i, M, d)$-код с избыточностью $r+i$ при $i \geqslant 1$; в этом случае название опущено.

4.4. Некоторые нелинейные коды в табл. 2 имеют избыточность r, которая не равна целому числу. В этих случаях число кодовых слов можно быстро отыскать следующим образом. Если $r = R + a$, где R — целое число и $0 < a < 1$, то число кодовых слов равно $M = i2^{n-R-5}$, где i задается таблицей

a	0,046	0,093	0,142	0,193	0,245	0,300	0,356
i	31	30	29	28	27	26	25
a	0,415	0,541	0,608	0,678	0,752	0,830	0,913
$i22$	24	22	21	20	19	18	17

Таблица 2

Двоичные коды длины n с минимальным расстоянием d и наименьшей известной избыточностью r

Расстояние $d = 3$ (см. § 4.7 этой статьи)

n	r
4—7	3
8	3,678
9	3,752
10—11	3,830
12—15	4
16—17	4,678
18—19	4,752
20—23	4,830
24—31	5
32—35	5,678
36—39	5,752
40—47	5,830
48—63	6
64—71	6,678
72—79	6,752
80—95	6,830
96—127	7
128—143	7,678
144—159	7,752
160—191	7,830
192—255	8
256—287	8,678
288—319	8,752
320—383	8,830
384—511	9
512	9,678

Расстояние $d = 5$

n	r	тип	ссылка
7—8	6	G	[L1]
9—11	6,415	H	[P3]
12—15	7	P	[N1]
16—19	8	XP	[S4]
20	8,678	X4	[S4]
21—23	9	G	[W1]
24—32	10	B	[G3]
33—63	11	P	[P4]
64—70	12	X4	[S4]
71—73	13	S	[H2]
74—128	14	B	[S4]
129—255	15	P	[P4]
256—271	16	X4	[S4]
272—277	17	S	[H2]
278—512	18	B	[S4]

Расстояние $d = 7$

n	r	тип	ссылка
10—11	9	G	[L1]
12—15	10	R	[P1]
16	10,830	J	[S1]
17—23	11	Q	[G1]
24	12		
25—27	13	K	[K1]
28—30	14	K	[K1]
31—32	15	D	[G4]
33	15,752	Z	[S3]
34—35	15,830	Z	[S3]
36—47	16	G *	
48—63	17	C	[C3]
64—67	18	XC	[S4]
68—70	19	XA	[S3]
71—83	20	S	[H2]
84—128	21	D	[G4]
129—135	22	XA	[S3]
136—159	23	Y1	[S4]
160—256	24	D	[G4]
257—264	25	XA	[S3]
265—311	26	S	[H2]
312—512	27	D	[G4]

* Согласно С. Редди

Расстояние $d = 9$

n	r	тип	ссылка
13—14	12	G	[L1]
15	13		
16	13,415	H	[L2]
17—19	13,678	H	[L2]
20	14,678		
21	15,415	H	[L2]
22	16,300	J	[S2]
23—25	16,678	Y2	[S4]
26	17,678		
27—29	18	B	[P1]

Расстояние $d = 11$

n	r	тип	ссылка
16—17	15	G	[L1]
18	16		
19	16,415	H	[L2]
20	17	B	[G3]
21—23	17,415	H	[L2]
24	18,300	J	[S2]
25—26	19	G	[H3]
27—31	20	B	[P1]
32	21		
33—35	22	Y1	[S4]

Продолжение

n	r	тип	ссылка
30—35	18,415	Y2	[S4]
36	19,415		
37—41	20	Q	[B1]
42—45	21	Q	[P2]
46	22		
47—49	22,193	Y2	]S4]
50—52	23	S	[H2]
53—73	24	B	[K3]
74—75	25—26		
76—90	27	S	[H2]
91—128	28	B	[S4]
129—135	29	XA	[S3]
136	30		
137—156	31	S	[H2]
157—256	32	B	[S4]
257—264	33	XA	[S3]
265	34		
266—311	35	S	[H3]
312—512	36	B	[S4]

n	r	тип	ссылка
36—47	23	Q	[P2]
48—50	24—26		
51—63	27	B	[P1]
64—67	28	XA	[S3]
68—70	29—31		
71—74	32	XC	[S4]
75	33		
76—94	34	S	[H3]
95—128	35	D	[G4]
129—135	36	XA	[S3]
136—137	37—38		
138—156	39	S	[H3]
157—256	40	D	[G4]
257—264	41	XA	[S3]
265—266	42—43		
267—311	44	S	[H3]
312—512	45	D	[G4]

Расстояние $d = 13$

n	r	тип	ссылка
19—20	18	G	[C2]
21—22	19—20		
23	20,415	H	[L2]
24	21	G	[C1]
25—27	21,193	H	[L2]
28	22,093	J	[S2]
29	23	R	[P1]
30	24		
31	24,830	H	[L2]
32	25,752	J	[S2]
33—37	26	XA	[S3]
38	27		
39—43	28	C	[B1]
44—45	29—30		
46—55	31	Y2	[S4]
56	32		
57—63	33	B	[P1]
64—70	34	XA	[S3]
71—72	35—36		
73—77	37	Q	[K1]
78—79	38—39		
80—85	40	Q	[K1]
86—96	41	S	[H3]
97—128	42	B	]S4]

Расстояние $d = 15$

n	r	тип	ссылка
22—23	21	G	[C2]
24—25	22—23		
26	23,415	H	[L2]
27	24	G	[L2]
28	24,678	H	[L2]
29—31	25	R	[P1]
32	26		
33	26,830	H	[L2]
34	27,752	J	[S2]
35	28	Z	[S3]
36—37	29	Z	[S3]
38—41	30	XC	[S4]
42	31		
43—47	32	G	[S4]
48—50	33	C	[B1]
51—55	34	C	[B1]
56—63	35	C	[K2]
64—66	36	XC	[S4]
67—68	37—38		
69—71	38,830	Y2	[S4]
72—79	39	Q	]K1]
80—81	40—41		
82—87	42	Q	[K1]
88—91	43—46		

Продолжение

n	r	тип	ссылка
129—135	43	XA	]S3]
136—138	44—46		
139—156	47	S	[H3]
157—256	48	B	[S4]
257—264	49	XA	[S3]
265—267	50—52		
268—311	53	S	[H3]
312—512	54	B	[S4]

Расстояние $d = 17$

n	r	тип	ссылка
25—26	24	G	[C2]
27—28	25—26		
29	26,415	H	[L2]
30	27,415		
31	28	G	[L2]
32	28,415	H	[L2]
33—35	28,830	H	[L2]
36	29,752	J	[S2]
37	30,678	H	[L2]
38	31,608	J	[S2]
39	32,541	H	[L2]
40	33,541		
41	34	N	]H3]
42	35	G	[H3]
43—46	36	G	[H3]
47—49	37	G	[K4]
50	38		
51—53	39	K	[K1]
54—55	39	Y3	[S4]
56	40	Y1	[S4]
57—62	41	C	[C3]
63—66	42	XC	[S4]
67—71	43	Y1	[S4]
72—89	44	Q	[K1]
90—93	45—48		
94—101	49	Q	[K1]
102	50		
103—105	51	K	[K1]
106—107	52—53		
108—125	54	B	[P1]
126	55		
127—128	56	B	[S4]
129—135	57	XA	[S3]
136—141	58—63		
142—256	64	B	[S4]
257—264	65	XA	[S3]
265—270	66—71		
271—512	72	B	[S4]

n	r	тип	ссылка
92—99	47	Q	[K1]
100	48		
101—128	49	D	[G4]
129—135	50	XA	[S3]
136—140	51—55		
141—256	56	D	[G4]
257—264	57	XA	[S3]
265—269	58—62		
270—512	63	D	[G4]

Расстояние $d = 19$

n	r	тип	ссылка
28—29	27	G	[C2]
30—32	28—30		
33	30,415	H	[L2]
34	31	C	[C3]
35	31,678	H	[L2]
36	32,415	H	[L2]
37—39	32,678	H	[L2]
40	33,608	J	[S2]
41	34,541	H	[L2]
42	35,541		
43	36,415	H	[L2]
44	37	G	[H3]
45—48	38	G	[H3]
49—51	39	G	[K4]
52—54	40—42		
55—61	43	B	[P1]
56—57	42	Y3	[S4]
58—61	43	B	[P2]
62—63	44	C	[C3]
64	45		
65—66	46	XC	[S4]
67—68	47	XC	[S4]
69—70	48	XC	[S4]
71—74	49	XC	[S4]
75—83	50	Y1	[S4]
84—103	51	Q	[K1]
104	52		
105—107	53	K	[K1]
108—109	54—55		
110—127	56	B	[P1]
128—131	57—60		
132—139	61	XC	[S4]
140—145	62—67		
146—255	68	B	[P1]
256—260	69	XA	[S3]
261—268	70—77		
269—270	78	Z	[S3]
271—272	79—80		
273—512	81	D	[G4]

Расстояние $d = 21$

n	r	тип	ссылка
31—32	30	G	[C2]
33—35	31—33		
36	33,415	H	[L2]
37	34,415		
38	35	G	[L2]
39	35,678	H	[L2]
40	36,193	H	[L2]
41—43	36,541	H	[L2]
44	37,541		
45	38,415	H	[L2]
46	39,356	J	[S2]
47—48	40	C	[C3]
49—51	41—43		
52—57	43,415	Y2	[S4]
58	44,415		
59—63	45	B	[P1]
64—70	46—52		
71—77	53	XC	[S4]
78	54		
79—85	55	K	[K1]
86—92	56—62		
93—127	63	B	[P1]
128—135	64	XA	[S3]
136—144	65—73		
145—146	74	Z	[S3]
147	75		
148—255	76	B	[P1]
256—264	77	XA	[S3]
265—273	78—86		
274—275	87	Z	[S3]
276—277	88—89		
278—512	90	B	[S4]

Расстояние $d = 23$

n	r	тип	ссылка
34—35	33	G	[C2]
36—38	34—36		
39	36,415	H	[L2]
40	37,415		
41	38	G	[L2]
42	39		
43	39,415	H	[L2]
44	40	C	[C3]
45—47	40,415	H	[L2]
48	41,356	J	[S2]
49—50	42	C	[C3]
51—53	43—45		
54—57	46	Y1	[S4]
58—63	47	B	[P1]
64—66	48	XA	[S3]
67—74	49—56		
75—87	57	K	[K1]
88—99	58—69		
100—127	70	B	[P1]
128—135	71	XA	[S3]
136—145	72—81		
146—147	82	Z	[S3]
148	83		
149—255	84	B	[P1]
256—264	85	XA	[S3]
265—274	86—95		
275—276	96	Z	[S3]
277—278	97—98		
279—512	99	D	[G4]

Расстояние $d = 25$

n	r	тип	ссылка
37—38	36	G	[C2]
39—42	37—40		
43	40,415	H	[L2]
44	41,415		
45	42	G	[L2]
46	42,678	H	[L2]
47	43,415	H	[L2]
48	44	G	[L2]
49—51	44,300	H	[L2]
52	45,245	J	[S2]
53	46,193	H	[L2]
54	47,193		
55	48,093	H	[L2]
56—61	49	N	[K5]
62—63	50—51		

Расстояние $d = 27$

n	r	тип	ссылка
40—41	39	G	[C2]
42—45	40—43		
46	43,415	H	[L2]
47	44,415		
48	45	C	[C3]
49	46		
50	46,678	H	[L2]
51	47,193	H	[L2]
52	47,830	H	[L2]
53—55	48,193	H	[L2]
56	49,193		
57	50,093	H	[L2]
58—63	51	N	[K5]
64—69	52—57		
70—74	58	XC	[S4]

Продолжение

n	r	тип	ссылка
64—66	52	K	[K1]
67	53		
68—70	54	XA	[S3]
71	55		
72—73	56	XC	[S4]
74—83	57—66		
84—85	67	G	[S3]
86—88	68—70		
89	70,678	Z	[S3]
90	71,415	Z	[S3]
91	72	Z	[S3]
92—94	72,300	Z	[S3]
95	73,245	Z	[S3]
96	74,193	Z	[S3]
97—125	75	B	[P1]
126—127	76—77		
128—135	78	XA	[S3]
136—146	79—89		
147—148	90	Z	[S3]
149	91		
150—255	92	B	[P1]
256—264	93	XA	[S3]
265—275	94—104		
276—277	105	Z	[S3]
278—279	106—107		
280—512	108	B	[S4]

n	r	тип	ссылка
75—86	59—70		
87—88	71	Z	[S3]
89	72		
90—91	73	G	[S4]
92—94	74—76		
95—127	77	B	[P1]
128—131	78—81		
132—139	82	XC	[S4]
140—150	83—93		
151—160	94	K	[K4]
161—165	95—99		
166—255	100	B	[P1]
256—264	101	XA	[S3]
265—276	102—113		
277—278	114	Z	[S3]
279—280	115—116		
281—512	117	D	[G4]

Расстояние $d = 29$

n	r	тип	ссылка
43—44	42	G	[C2]
45—48	43—46		
49	46,415	H	[L2]
50	47,415		
51	48,415		
52	49	G	[L2]
53	49,678	H	[L2]
54	50,415	H	[L2]
55	51,193	H	[L2]
56	51,678	H	[L2]
57—59	52,093	H	[L2]
60	53,046	J	[S2]
61	54	R	[P1]
62	55		
63	55,913	H	[L2]
64	56 913		
65—67	57	XA	[S3]
68—74	58—64		
75—78	65	XC	[S4]
79—86	66—73		

n	r	тип	ссылка
87—88	74	Z	[S3]
89—93	75	G	[S4]
94—97	76—79		
98	79,415	Z	[S3]
99—100	80,415	Z	[S3]
101	81,415		
102—125	82	B	[K2]
126—127	83—84		
128—135	85	XA	[S3]
136—148	86—98		
149—150	99	Z	[S3]
151—155	100—104		
156—158	105	Z	[S3]
159—160	106—107		
161—255	108	B	[P1]
256—264	109	XA	[S3]
265—277	110—122		
278—279	123	Z	[S3]
280—281	124—125		
282—512	126	B	[S4]

4.5. Во многих случаях ссылка указывает на работу, в которой можно найти полное распределение весов кода, а не первоначальное определение минимального расстояния.

4.6. Хотя многие из этих кодов, быть может, являются оптимальными в том смысле, что они имеют наименьшую возможную избыточность, лишь немногие из них известны как оптимальные (ср. с [48]). Читателю предлагается проверить и улучшить их. Коды типа Z с расстояниями 25—29 являются особенно слабыми. Было бы интересно узнать о любых улучшениях в этом направлении.

4.7. В первой части табл. 2 приведенные коды для минимального расстояния $d = 3$, когда $n = 3$, 4, 5 и $3 \cdot 2^{m-2} \leqslant n \leqslant 2^m$, $m \geqslant 3$, являются (укороченными) кодами Хэмминга [H1]. Когда $2^m \leqslant n < 3 \cdot 2^{m-1}$, $m \geqslant 3$, приведенные коды являются негрупповыми кодами, открытыми Голеем [G2] и Юлиным [J1] для $n = 8$, 9, 10, 11 и Слоэном и Вайтхедом [S1] для $n \geqslant 16$.

Мне бы хотелось поблагодарить Х. Л. Бергера за помощь по сбору данных для табл. 2; Э. Р. Берлекэмпа — за информацию о недавних русских работах; Ф. Дж. Мак-Вильямс — за полезные замечания по рукописи и Х. Дж. Хелгерта — за ряд хороших кодов.

СПИСОК ЛИТЕРАТУРЫ

В списке употребляются следующие сокращения: *BSTJ* = **Bell System** Technical Journal, *IC* = Information and Control, *JCT* = Journal **of Combinatorial** Theory, *PGIT* = IEEE Transactions on Information Theory.

1. Список литературы к таблице кодов

[B1] Berlekamp E. R., Algebraic coding theory, McGraw-Hill, New York, 1968 (see especially pp. 360, 432—433). (Русский перевод: Берлекэмп Э., Алгебраическая теория кодирования, «Мир», М., 1971 (см. в особенности стр. 368, 438—439).)

[C1] Calabi L., Myrvaagnes E., On the minimal weight of binary group codes, *PGIT*, **10** (1964), 385—387.

[C2] Cordaro J. T., Wagner T. J., Optimum $(n, 2)$ codes for small values of channel error probability, *PGIT*, **13** (1967), 349—350.

[C3] Chen C. L., Computer results on the minimum distance of some binary cyclic codes, *PGIT*, **16** (1970), 359—360.

[F1] Fontaine A. B., Peterson W. W., Group code equivalence and optimum codes, *PGIT*, **5** (1959) (Special Suppl.), 60—70.

[G1] Golay M. J. E., Notes on digital coding, *Proc. IRE*, 37 (1949), 657.

[G2] Golay M. J. E., Binary coding, *PGIT*, 4 (1954), 23—28.

[G3] Goldman H. D., Kliman M., Smola H., The weight structure of some Bose — Chaudhuri codes, *PGIT*, **14** (1968), 167—169.

[G4] Гопа В. Д., Новый класс линейных корректирующих кодов, *Проблемы передачи информации*, **6**, вып. 3 (1970), 24—30.

[H1] Hamming R. W., Error detecting and error correcting codes, *BSTJ*, **29** (1950), 147—160. (Русский перевод: Хэмминг Р. В., Коды с обнаружением и исправлением ошибок, в сб. Коды с обнаружением и исправлением ошибок, ИЛ, М., 1956, стр. 7—23.)
[H2] Helgert H. J., Srivastava codes, *PGIT*, **18** (1972), 292—297.
[H3] Helgert H. J., частное сообщение
[J1] Julin D., Two improved block codes, *PGIT*, **11** (1965), 459.
[K1] Karlin M., New binary coding results by circulants, *PGIT*, **15** (1969), 81—92.
[K2] Kasami T., Tokura N., Some remarks on BCH bounds and minimum weights of binary primitive BCH codes, *PGIT*, **15** (1969), 408—413.
[K3] Kasami T., Lin S., Peterson W. W., Polynomial codes, *PGIT*, **14** (1968), 807—814.
[K5] Kerdock A. M., A class of low-rate nonlinear codes, *IC*, **20** (1972), 182—187.
[K4] Karlin M., personal communication
[L1] Laemmel A. E., Efficiency of noise reducing codes, in: Jackson W., ed., Communication theory, Butterworth, London, 1953, p. 111—118.
[L2] Левенштейн В. И., Применение матриц Адамара к одной задаче кодирования, Проблемы кибернетики, вып. 5, Физматгиз, М., 1961, стр. 123—137.
[L3] Lum V., Chien R. T., On the minimum distance of Bose — Chaudhuri — Hocquenghem codes, *SIAM J. Appl. Math.*, **16** (1968), 1325—1337.
[N1] Nordström A. W., Robinson J. P., An optimum nonlinear code, *IC*, **11** (1967), 613—616.
[P1] Peterson W. W., Error-correcting codes (M. I. T. Press, Cambridge, Mass., 1961) (see especially pp. 71, 166—167). (Русский перевод: Питерсон У. У., Коды, исправляющие ошибки, «Мир», М., 1964.)
[P2] Pless V. S., Power moment identities on weight distributions in error correcting codes, *IC*, **6** (1963), 147—152.
[P3] Plotkin M., Binary codes with specified minimum distances, *PGIT*, **6** (1960), 445—450. (Русский перевод: Плоткин М., Двоичные коды с заданным минимальным расстоянием, Киб. сборник, вып. 7, ИЛ, М., 1963.)
[P4] Preparata F. P., A class of optimum nonlinear double-error correcting codes, *IC*, **13** (1968), 378—400. (Русский перевод: Препарата Ф. П., Класс оптимальных нелинейных кодов с исправлением двойных ошибок, Киб. сборник (нов. серия), вып. 7, 1970, стр. 18—42.)
[S1] Sloane N. J. A., Whitehead D. S., A new family of single-error correcting codes, *PGIT*, **16** (1970), 717—719.
[S2] Sloane N. J. A., Seidel J. J., A new family of nonlinear codes obtained from conference matrices, *Ann. New York Acad. Sci.*, **175** (1970), 363—365.
[S3] A new code.
[S4] Sloane N. J. A., Reddy S. M., Chen C. L., New binary codes, *PGIT*, **18** (1972), 503—510.
[T1] Tokura N., Taniguchi K., Kasami T., A search procedure for finding optimum group codes for the binary symmetric channel, *PGIT*, **13** (1967), 587—594.
[W1] Wagner T. J., A search technique for quasi-perfect codes, *IC*, **9** (1966), 94—99.

2. Список литературы, цитированной в тексте

[1] Anderson D. R., A new class of cyclic codes, *SIAM J. Appl. Math.*, **16** (1968), 181—197.
[2] Assmus E. F., Jr., Mattson H. F., Jr., Turyn R. J., Research to develop the algebraic theory of codes (Sci. Rept. AFCRL-67-0365, Air Force Cambridge Res. Lab., Bedford, Mass., 1967).

[3] Assmus E. F., Jr., Mattson H. F., Jr., New 5-designs, *JCT*, **6** (1969), 122—151.
[4] Assmus E. F., Jr., Mattson H. F., Jr., Some ($3p$, p) codes, in: Information processing 68, North-Holland, Amsterdam, 1969, p. 205—209.
[5] Baumert L. D., McEliece R. J., Weights of irreducible cyclic codes, to appear.
[6] Berger T., Rate distortion theory, Prentice-Hall, Englewood Cliffs, N. J., 1971.
[7] Berlekamp E. R., Weight enumeration theorems, in: Proc. 6th Allerton Conf. on Circuit and Systems Theory, Urbana, Univ. of Illinois Press, Chicago, Ill, 1968, 1968, p. 161—170.
[8] Berlekamp E. R., The weight enumerators for certain subcodes of the second order binary Reed — Muller codes, *IC*, **17** (1970), 485—500.
[9] Berlekamp E. R., Some mathematical properties of a scheme for reducing the bandwidth of motion pictures by Hadamard smearing, *BSTJ*, **49** (1970), 969—986.
[10] Berlekamp E. R., A survey of coding theory for algebraists and combinatorialists, Intern. Centre for Mech. Sci., Udine, Italy, 1970.
[11] Berlekamp E. R., Coding theory and the Mathieu groups, *IC*, **18** (1971), 40—64.
[12] Berlekamp E. R., Long primitive binary BCH codes have distance $d \sim 2n \ln R^{-1}/\log n \ldots$, *PGIT*, 18 (1972), 415—426.
[14] Berlekamp E. R., Sloane N. J. A., Weight enumerator for second order Reed — Muller codes, *PGIT*, **16** (1970), 745—751.
[15] Berlekamp E. R., Welch L. R., Weight distributions of the cosets of the (32, 6) Reed — Muller code, *PGIT*, **18** (1972), 203—207.
[16] Berlekamp E. R., MacWilliams F. J., Sloane N. J. A., Gleason's theorem on self dual codes, **18**, (1972), 409—414.
[17] Берман С. Д., К теории групповых кодов, *Кибернетика*, **3** (1967), 25—31.
[18] Берман С. Д., Полупростые циклические и абелевы коды II, *Кибернетика*, 3 (1967), 17—23.
[19] Bose R. C., Ray-Chaudhuri D. K., On a class of error correcting binary group codes, *IC*, **3** (1960), 68—79, 279—290. (Русский перевод: Боуз Р. К., Рой-Чоудхури Д. К., Об одном классе двоичных групповых кодов с исправлением ошибок, Киб. сборник, вып. 2, 1961, ИЛ, М., стр. 83—94.)
[20] Burton H. O., A survey of error correcting techniques for data on telephone facilities, in: Proc. Intern. Commun. Conf., San Francisco, Calif., 1970.
[21] Camion P., Abelian codes, Math. Res. Center, Univ. of Wisconsin, Rept. 1059, 1970.
[22] Chen C. L., The existence of arbitrarily long pseudo-cyclic codes that meet the Gilbert bound, in: Proc. 5th Ann. Princeton Conf. Inform. Sci., 1971, p. 242.
[23] Chen C. L., Peterson W. W., Weldon E. J., Jr., Some results on quasi-cyclic codes, *IC*, **15** (1969), 407—423.
[24] Conway J. H., A group of order 8, 315, 553, 613, 086, 720, 000, *Bull. London Math. Soc.*, **1** (1969), 79—88.
[25] Conway J. H., A characterization of Leech's lattice, *Invent. Math.*, **7** (1969), 137—142.
[26] Dagnino G., On a new class of binary group codes, *Calcolo*, **5** (1968), 277—294.
[27] Delsarte P., Automorphisms of abelian codes, *Philips Res. Rept.*, **25** (1970), 389—402.
[28] Delsarte P., Majority logic decodable codes derived from finite inversive planes, *IC*, **18** (1971), 319—325.

[29] Delsarte P., Goethals J. M., Irreducible binary cyclic codes of even dimension, Univ. North Carolina at Chapel Hill, Inst. Statist., Mimeo Ser. No. 600.27, 1970.
[30] Dobrushin R. L., Survey of Soviet research in information theory, *PGIT*, **18** (1972).
[31] Gilbert E. N., A comparison of signaling alphabets, *BSTJ*, **31** (1952), 504—522.
[32] Gleason A. M., Weight polynomials of self-dual codes and the MacWilliams identities, in: Proc. Intern. Congr. Mathematicians, Nice, 1970, 140—144.
[33] Goethals J. M., Factorization of cyclis codes, *PGIT*, **13** (1967), 242—246.
[34] Goethals J. M., On the Golay perfect binary code, *JCT*, **11** (1971). 178—186.
[35] Goethals J. M., Some combinatorial aspects of coding theory, in: Proc. Combinat. Symp., Fort Collins, 1971, to appear.
[36] Goethals J. M., Snover S. L., Nearly perfect binary codes, *Discrete Math.*, **3** (1972), 65—88.
[37] Green M. V., Two heuristic techniques for block-code construction (Abstract), *PGIT*, **12** (1966), 273.
[38] Hartmann C. R. P., On the minimum distance structure of cyclic codes and decoding beyond the BCH bound, Ph. D. Thesis, Univ. of Illinois, 1970; also Coord. Sci. Lab. Rept. R-458, Univ. of Illinois, 1970.
[39] Hartmann C. R. P., A note on the minimum distance structure of cyclic codes, *PGIT*, **18** (1972), 439—440.
[40] Hartmann C. R. P., A generalization of the BCH bound, submitted to *IC*.
[41] Hartmann C. R. P., On the weight structure of cyclic codes of composite length, in: Proc. Fourth Hawaii Inter. Conf. System Sci., 1971, p. 117—119.
[42] Hartmann C. R. P., Tzeng K. K., A bound for cyclic codes of composite length, *PGIT*, **18** (1972), 307.
[43] Hartmann C. R. P., Tzeng K. K., Chien R. T., Some results on the minimum distance structure of cyclic codes, *PGIT*, **18** (1972), 402—409.
[44] Hatcher T., On minimal distance, shortest length, and greatest number of elements for binary group codes, Parke Mathematical Labs., Carlisle, Mass., Tech. Memo. 6, 1964.
[45] Hocquenghem A., Codes correcteurs d'erreurs, *Chiffres*, **2** (1959), 147—156.
[46] Hoffner C. W. II and Reddy S. M., Circulant bases for cyclic codes, *PGIT*, **16** (1970), 511—512.
[47] Jelinek F., Free encoding of memoryless time-discrete sources with a fidelity criterion, *PGIT*, **15** (1969), 584—590.
[48] Johnson S. M., On upper bounds for unrestricted binary error-correcting codes, *PGIT*, **17** (1971), 466—478.
[49] Karlin M., Decoding of circulant codes, *PGIT*, **16** (1970), 797—802.
[50] Karlin M., Weight/moment relationships in $(Q+E)$ circulants, unpublished.
[51] Kasami T., Some lower bounds on the minimum weight of cyclic codes of composite length, *PGIT*, **14** (1968), 814—818.
[52] Kasami T., An upper bound on k/n for affine-invariant codes with fixed d/n, *PGIT*, **15** (1969), 174—176. (Русский перевод: Касами Т., Верхняя граница для k/n аффинно инвариантных кодов с фиксированным d/n, Киб. сборник, нов. серия, вып. 8, 1971, «Мир», М., стр. 5—11.)
[53] Kasami T., The weight enumerators for several classes of subcodes of the second order binary Reed — Muller codes, *IC*, **18** (1971), 369—394.
[54] Kasami T., Some results on the weight structure of Reed — Muller codes, to appear.

[55] Kasami T., Lin S., Peterson W. W., Some results on weight distributions of BCH codes, *PGIT*, **12** (1966), 274.
[56] Kasami T., Tokura N., On the weight structure of Reed — Muller codes, *PGIT*, **16** (1970), 752—759.
[57] Kasami T., Tokura N., Azumi S., On the weight distribution of Reed — Muller codes, Inst. Electron. Comm. Eng., Japan, PGIT Rept.. 1971 (in Japanese).
[58] Kautz W. H., Levitt K. N., A survey of progress in coding theory in the Soviet Union, *PGIT*, **15** (1969), 197—244.
[59] Кошелев В. Н., О некоторых свойствах случайных групповых кодов большой длины, Проблемы передачи информации I, вып. 4, 1965, стр. 45—48.
[60] Козлов М. В., Корректирующая способность линейных кодов, *ДАН*, **14** (1969).
[61] Leech J., Some sphere packings in higher space, *Can. J. Math.*, **16** (1964), 657—682.
[62] Leech J., Notes on sphere packings, *Can. J. Math.*, **19** (1967), 251—267.
[63] Leech J., Sloane N. J. A., New sphere packings in dimensions 9—15, *Bull. Amer. Math. Soc.*, **76** (1970), 1006—1010.
[64] Leech J., Sloane N. J. A., New sphere packings in more than thirty-two dimensions, in: Proc. second Chapel Hill Conference on Comb. Math., Chapel Hill, N. C., 1970, p. 345—355.
[65] Leech J., Sloane N. J. A., Sphere packing and error-correcting codes, *Can. J. Math.*, **23** (1971), 718—745.
[66] Леонтьев В. К., Одна гипотеза о кодах Боуза — Чоудхури, Проблемы передачи информации, 4, вып. I, 1968, стр. 83—85.
[67] Lin S., Weldon E. J., Jr., Long BCH codes are bad, *IC*, **11** (1967), 445—451.
[68] Liu C. L., Ong B. G., Ruth G. R., A construction scheme for linear and nonlinear codes, in: Proc. 5th Ann. Princeton Conf. Inform. Sci., 1971, p. 245—247.
[69] Lucky R. W., Salz J., Weldon E. J., Jr., Principles of data communication, McGraw-Hill, New York, 1968.
[70] MacWilliams F. J., Error-correcting codes — An historical survey, in: H. B. Mann, ed., Error correcting codes, Wiley, New York, 1968.
[71] MacWilliams F. J., Codes and ideals in group algebras, in: Bose R. C., Dowling T. A., eds., Combinatorial mathematics and its applications, Univ. North Carolina Press. Chapel Hill, 1969.
[72] MacWilliams F. J., Binary codes which are ideals in the group algebra of an abelian group, *BSTJ*, **49**, (1970), 987—1011.
[73] MacWilliams F. J., Mallows C. L., Sloane N. J. A., Generalizations of Gleason's theorem on weight enumerators of self-dual codes, *PGIT*, 18 (1972), to appear.
[74] MacWilliams F. J., Sloane N. J. A., Thompson J. G., On the existence of a projective plane of order 10, *JCT*, to appear.
[75] MacWilliams F. J., Sloane N. J. A., Thompson J. G., Good self-dual codes exist, *Discrete Math.*, 3 (1972), 153—162.
[76] Марчуков А. С., О суммировании произведений кодов, Проблемы передачи информации, 4, вып. 2, 1968, стр. 11—20.
[77] McEliece R. J., On the symmetry of good nonlinear codes, *PGIT*, **16** (1970), 609—611.
[78] McEliece R. J., Rumsey H., Jr., Euler products, cyclotomy, and coding, in: Space programs summary Jet Propulsion Lab., Calif. Inst. Technol., vol. 37-65-III, 1970, p. 22—27; and *J. Number Theory*, **4** (1972), 302—311.
[79] Muller D. E., Application of boolean algebra to switching circuit design and error detection, *IRE Trans. Electronic Computers*, **EC3** (1954), 6—12.
[80] Nadler M., A 32-point $n = 12$, $d = 5$ code, *PGIT*, **8** (1962), **58**.

[81] Оганесян С. Ш., Ягджян В. Г., Весовой спектр для некоторых классов корректирующих циклических кодов, Проблемы передачи информации, **6**, вып. 3, 1969, стр. 28—36.
[82] Parker E. T., Nikolai P. J., A search for analogues of the Mathieu groups, *Math. Comp.*, **12** (1958), 38—43.
[83] Patel A. M., Maximal codes with specified minimum distance, IBM Tech. Rept. TR 44.0085, 1969.
[84] Patel A. M., Maximal group codes with specified minimum distance, *IBM J. Res. Devel.*, **14** (1970), 434—443.
[85] Pehlert W. K., Jr., Analysis of a burst-trapping error correction procedure, *BSTJ*, **49** (1970), 493—519.
[86] Peterson W. W., On the weight structure and symmetry of BCH codes, Air Force Cambridge Res. Lab., Bedford, Mass., Rept. AFCRL-65-515, 1965.
[87] Pless V. S., On the uniqueness of the Golay codes, *JCT*, **5** (1968), 215—228.
[88] Pless V. S., On a new family of symmetry codes and related new five-designs, *Bull. Am. Math. Soc.*, **75** (1969), 1339—1342.
[89] Pless V. S., Symmetry codes over GF (3) and new five-designs, *JCT*, **12**, (1972), 119—142.
[90] Pless V. S., A classification of self-orthogonal codes over GF (2), *Discrete Math.*, **3** (1972), 209—246.
[91] Preparata F. P., A new look at the Golay (23, 12) code, *PGIT*, **16** (1970), 510—511.
[92] Второй международный симпозиум по теории информации, Тезисы докладов, 2—8 сентября 1971 года, Цахкадзор, Армянская ССР, Москва — Ереван, 1971.
[93] Reed I. S., A class of multiple-error-correcting codes and the decoding scheme, *PGIT*, 4 (1954), 38—49.
[94] Sarwate D. V., Berlekamp E. R., On the weight enumeration of Reed — Muller codes and their cosets, to appear.
[95] Savage J. E., The complexity of decoders, II, Computational work and decoding time, *PGIT*, **17** (1971), 77—85.
[96] Семаков Н. В., Зиновьев В. А., Равновесные коды и тактические конфигурации, Проблемы передачи информации, 5, вып. 3 1969, стр. 28—36.
[97] Семаков Н. В., Зиновьев В. А., Зайцев Г. В., Равномерно упакованные коды, Проблемы передачи информации, 7, вып. 1, 1971, стр. 38—50.
[98] Shannon C. E., A mathematical theory of communication, *BSTJ*, 27 (1948), 379—423, 623—656. (Русский перевод: Шеннон К., Математическая теория связи, в книге: Работы по теории информации и кибернетике, ИЛ, М., 1963, 243—332.)
[99] Shiva S. G. S., Certain group codes, *Proc. IEEE*, **55** (1967), 2162—2163.
[100] Сидельников В. М., О спектре весов двоичных кодов Боуза — Чоудхури — Хоквингема, Проблемы передачи информации, 7, вып. 1, 1971, стр. 14—22.
[101] Singleton R., Maximum distance Q-nary codes, *PGIT*, **10** (1964), 116—118.
[102] Sloane N. J. A., A survey of recent results in constructive coding theory, in: National Telemetering Conf. NTC'71 Record IEEE, New York, 1971, p. 218—227.
[103] Sloane N. J. A. and Dick R. J., On the enumeration of costs of first order Reed — Muller codes, IEEE Intern. Conf. Communications, Montreal, 1971.
[104] Stanton R., The Mathieu groups, *Can, J. Math.*, **3** (1951), 164—174.
[105] Stiffler J. J., Theory of synchronous communications, Prentice-Hall, Englewood Cliffs, N. J., 1971.

[106] Sugino M., Ienaga Y., Tokura N., Kasami T., Weight distribution of (128, 64) Reed — Muller code, *PGIT*, 17 (1971), 627—628.
[107] Tietäväinen A., On the nonexictence of perfect codes ower finite fields, *SIAM J*, to appear.
[108] Tietäväinen A., Perko A., There are no unknown perfect binary codes, Ann. Univ. Turku, Ser. A1 148, 1971, p. 3—10.
[109] Tong S. Y., Burst-trapping techniques for a compound channel, *PGIT*, 15 (1969), 710—715.
[110] Tong S. Y., Performance of burst-trapping codes, *BSTJ*, 49 (1970), 477—491.
[111] Van Lint J. H., Coding theory, Lecture Notes in Math. 201, Springer, Berlin, 1971.
[112] Van Lint J. H., A survey of recent work on perfect codes, *Rocky Mountain J. Math.*, to appear.
[113] Van Lint J. H., A new description of the Nadler code, *PGIT*, to appear.
[114] Van Tiborg H. C. A., Weights in the third-order Reed — Muller codes, Jet Propulsion Lab., Calif. Inst. Technol., Tech. Rept. 32—1526, IV, 1971.
[115] Weldon E. J., Jr., Long quasi-cyclic codes are good (abstract), *PGIT*, 16 (1970), 130.
[116] Witt E., Über Steinersche Systeme, *Abh. Math. Sem. Univ. Hamburg*, 12 (1938), 265—275.
[117] Wolf J. K., Adding two information symbols to certain nonbinary BCH codes and some applications, *BSTJ*, 48 (1969), 2405—2424.
[118] Wolf J. K., Nonbinary random error-correcting codes, *PGIT*, 16 (1970), 236—237,

Класс нелинейных двоичных кодов с низкой скоростью передачи[1]

А. М. Кердок

Мы предполагаем, что читатель знаком с полиномами Мэттсона — Соломона, линеаризованными полиномами и аффинными полиномами, описанными в разд. 11.1 и 16.3 работы Берлекэмпа (1968).

Мы начинаем с леммы, которая часто позволяет вычислять вес кодового слова в РМ-коде 2-го порядка исходя из его полинома Мэттсона — Соломона.

Лемма. *Пусть* $f(x)$ *— полином Мэттсона — Соломона кодового слова РМ-кода 2-го порядка* $[f(0), f(1), f(\alpha), \ldots, f(\alpha^{2^m-2})]$, *где* $\alpha \in GF(2^m)$ *— примитивный элемент и* $f(\xi) \in GF(2)$ *для всех* $\xi \in GF(2^m)$. *Тогда производная* $f(x)$ *есть аффинный полином вида*

$$f'(x) = t + L(x),$$

где $t \in GF(2^m)$ *и* $L(x)$ *— линеаризованный полином, а вес кодового слова с полиномом Мэттсона — Соломона* $f(x)$ *определяется следующим образом:*

$$f| = \begin{cases} 2^{m-1}, & \text{если } f(\xi) = f(0) + 1 \text{ и } L(\xi) = 0 \text{ для некоторого } \xi \in GF(2^m), \\ 2^{m-1} \pm 2^{m-1-(m-s)/2} & \text{в остальных случаях,} \end{cases}$$

где s *— размерность пространства корней* $L(x)$ *в* $GF(2^m)$.

Доказательство. В соответствии с теоремой Диксона (Берлекэмп (1968), теорема 16.35) каждое кодовое слово РМ-кода 2-го порядка имеет полином Мэттсона — Соломона вида

$$f(x) \equiv A + T(ux) + \sum_{i=1}^{h} T(\beta_i x)\, T(\gamma_i x) \bmod x^{2^m} + x, \qquad (1)$$

где $A \in GF(2)$, $u \in GF(2^m)$, $T(\xi) = \sum_{i=0}^{m-1} \xi^{2^i}$ и

[1]) Kerdock A. M., A Class of Low-Rate Nonlinear Binary Codes. *Information and Control*, 20, № 2 (1972), 182—187.

$\beta_1, \gamma_1, \beta_2, \gamma_2, \ldots, \beta_h, \gamma_h$ — некоторые элементы $GF(2^m)$, которые линейно независимы над $GF(2)$. Если u линейно независим от $\beta_1, \gamma_1, \ldots, \beta_h, \gamma_h$, то $|f| = 2^{m-1}$, но если u — линейная комбинация $\beta_1, \gamma_1, \ldots, \beta_h, \gamma_h$, то соответствующим аффинным преобразованием $\beta_1, \gamma_1, \ldots, \beta_h, \gamma_h$ можно u перевести в 0 и получить $|f| = 2^{m-1} \pm 2^{m-1-h}$, где знак зависит от двоичной константы A.

Дифференцирование равенства (1) дает

$$f'(x) = u + \sum_{i=1}^{h} (\beta_i T(\gamma_i x) + \gamma_i T(\beta_i x)(\beta_i \gamma_i)^{2^{m-1}}) =$$

$$= u + \sum_{i=1}^{h} ((\beta_i \gamma_i)^{2^{m-1}} + L(x)),$$

где $L(x)$ — линеаризованный полином.

Уравнение $L(\xi) = 0$ эквивалентно

$$\sum_{i=1}^{h} (\beta_i T(\gamma_i \xi) + \gamma_i T(\beta_i \xi)) = 0. \tag{2}$$

Если $\xi \in GF(2^m)$, то $T(\gamma_i \xi)$ и $T(\beta_i \xi)$ принадлежат $GF(2)$. Так как $\beta_1, \gamma_1, \ldots, \beta_h, \gamma_h$ линейно независимы над $GF(2)$, то (2) имеет место тогда и только тогда, когда

$$T(\gamma_i \xi) = T(\beta_i \xi) = 0 \quad \text{для} \quad i = 1, 2, \ldots, h. \tag{3}$$

Пусть $\gamma_1, \beta_1, \ldots, \gamma_h, \beta_h, \delta_1, \delta_2, \ldots, \delta_{m-2h}$ — базис $GF(2^m)$ над $GF(2)$. Тогда ξ единственным образом определяется m двоичными величинами $T(\gamma_i \xi)$, $T(\beta_i \xi)$, $T(\delta_j \xi)$, где $i = 1, 2, \ldots, h$ и $j = 1, 2, \ldots, m - 2h$. Следовательно, уравнения (3) имеют 2^{m-2h} решений, соответствующих 2^{m-2h} выборам $T(\delta_j \xi)$. Таким образом, $s = m - 2h$ и $h = (m - s)/2$. Доказательство завершается тем замечанием, что u линейно независимо от γ_i, β_i $(i = 1, \ldots, h)$ тогда и только тогда, когда существует решение уравнений $T(u\xi) = 1$; $T(\gamma_i \xi) = T(\beta_i \xi) = 0$ для $i = 1, \ldots, h$. Ч. Т. Д.

Следствие. *Если* ξ *— единственный ненулевой корень* $L(x)$ *в* $GF(2^m)$, *то*

$$|f| = \begin{cases} 2^{m-1}, & \text{если } f(\xi) = f(0) + 1, \\ 2^{m-1} \pm 2^{(m-1)/2}, & \text{если } f(\xi) = f(0). \end{cases}$$

КОНСТРУКЦИЯ КОДА

Построим теперь наш код длины 2^{2l}. Каждое кодовое слово определяется в терминах двух полиномов Мэттсона — Соломона, один из которых определяет левую часть кодового слова, а другой — правую. Каждая половина имеет длину 2^m, где $m =$

$= 2l - 1$. Левая половина имеет полином Мэттсона — Соломона вида

$$f_l(x) = T(\eta x) + Q(\varphi x) + A,$$

а правая половина того же кодового слова — полином Мэттсона — Соломона вида

$$f_r(x) = T(\eta x + \varphi x) + Q(\varphi x) + B,$$

где $A,\ B \in GF(2),\ \eta,\ \varphi \in GF(2^m)$ и

$$Q(y) \equiv T\left(y^3 + y^5 + y^9 + \ldots + y^{1+2^{(m-1)/2}}\right) \bmod y^{2^m} + y.$$

Заметим попутно, что

$$Q'(y) = \sum_{i=1}^{m-1} y^{2^i} = y + T(y) = (T_{m-1}(y))^2,$$

где

$$T_{m-1}(y) = \sum_{i=0}^{m-2} y^{2^i}.$$

ДОКАЗАТЕЛЬСТВО ТОГО, ЧТО КОНСТРУКЦИЯ РАБОТАЕТ

Так как существуют две возможности выбора для A, две для B, 2^m для η и 2^m для φ, то очевидно, что конструкция дает код с 2^{4l} кодовыми словами длины 2^{2l}.

Покажем теперь, что разность между любыми двумя кодовыми словами имеет вес по крайней мере $2^m - 2^{(m-1)/2}$. Пусть первое слово имеет параметры η_1, φ_1, A_1, B_1, а второе — η_2, φ_2, A_2, B_2. Определим $\varphi_3 = \varphi_1 + \varphi_2$, $\eta_3 = \eta_1 + \eta_2$, $A_3 = A_1 + A_2$, $B_3 = B_1 + B_2$. Левая половина разности имеет полином Мэттсона — Соломона

$$\Delta_l(x) = Q(\varphi_1 x) + Q(\varphi_2 x) + T(\eta_3 x) + A_3, \tag{4}$$

а правая половина — полином Мэттсона — Соломона

$$\Delta_r(x) = \Delta_l(x) + T(\varphi_3 x) + A_3 + B_3. \tag{5}$$

Случай $\varphi_3 = 0$ тривиален, так как тогда разность двух кодовых слов имеет вес $0, 2^m$ или 2^{m-1}. (Действительно, разность тогда принадлежит РМ-коду 1-го порядка длины 2^{m+1} с кодовым расстоянием 2^m.) Поэтому предположим, что $\varphi_3 \neq 0$. Так как $(\varphi_1 + \varphi_2)/\varphi_3 = 1$ и m нечетно, то, следовательно,

$$T(\varphi_1/\varphi_3) + T(\varphi_2/\varphi_3) = T(1) = 1. \tag{6}$$

Кроме того, так как

$$(\varphi_1^2 + \varphi_1\varphi_2)/\varphi_3^2 = \varphi_1/\varphi_3,$$

то

$$T(\varphi_1^2/\varphi_3^2) + T(\varphi_1\varphi_2/\varphi_3^2) = T(\varphi_1/\varphi_3),$$

откуда

$$T(\varphi_1\varphi_2/\varphi_3^2)=0. \tag{7}$$

Из (4) и (5) получаем

$$\Delta_l'(x)=\varphi_1(\varphi_1 x+T(\varphi_1 x))+\varphi_2(\varphi_2 x+T(\varphi_2 x))+\eta_3$$

и

$$\Delta_r'(x)=\varphi_1(\varphi_1 x+T(\varphi_1 x))+\varphi_2(\varphi_2 x+T(\varphi_2 x))+\eta_3+\varphi_3.$$

Обе производные имеют линеаризованную часть вида

$$L(x)=\varphi_3^2 x+\varphi_1 T(\varphi_1 x)+\varphi_2 T(\varphi_2 x).$$

Имеем $L(\xi)=0$ тогда и только тогда, когда

$$\xi=[\varphi_1 T(\varphi_1\xi)+\varphi_2 T(\varphi_2\xi)]/\varphi_3^2.$$

Если $\xi\in GF(2^m)$, то $T(\varphi_1\xi)$ и $T(\varphi_2\xi)$ принадлежат $GF(2)$. Следовательно,

$$\xi=\begin{cases} 0, & \text{если } T(\varphi_1\xi)=T(\varphi_2\xi)=0,\\ \varphi_1/\varphi_3^2, & \text{если } T(\varphi_1\xi)=1,\ T(\varphi_2\xi)=0,\\ \varphi_2/\varphi_3^2, & \text{если } T(\varphi_1\xi)=0,\ T(\varphi_2\xi)=1,\\ 1/\varphi_3, & \text{если } T(\varphi_1\xi)=0,\ T(\varphi_2\xi)=1. \end{cases} \tag{8}$$

Последняя возможность, $\xi=1/\varphi_3$, не может быть реализована, так как это ведет к немедленному противоречию с (6).

Если $T(\varphi_1\xi)=1$, то $\xi=\varphi_1/\varphi_3^2$ и $T(\varphi_2\xi)=0$ из (7). Аналогично из $T(\varphi_2\xi)=1$ следует, что $T(\varphi_1\xi)=0$. Следовательно, (8) можно переписать в виде

$$\xi=\begin{cases} 0,\\ \varphi_1/\varphi_3^2, & \text{если } T(\varphi_1/\varphi_3)=1,\\ \varphi_2/\varphi_3^2, & \text{если } T(\varphi_2/\varphi_3)=1. \end{cases}$$

Из (6) мы выводим, что всегда существует единственное ненулевое решение для ξ, которое можно записать в виде

$$\xi=T(\varphi_1/\varphi_3)(\varphi_1/\varphi_3^2)+T(\varphi_2/\varphi_3)(\varphi_2/\varphi_3^2).$$

Согласно следствию, веса $|\Delta_l|$ и $|\Delta_r|$ зависят от величин $\Delta_l(\xi)$ и $\Delta_r(\xi)$, которые мы сейчас подсчитаем:

$$\Delta_l(\xi)+\Delta_l(0)=\Delta_r(\xi)+\Delta_r(0)+T(\xi\varphi_3),$$

$$T(\xi\varphi_3)=[T(\varphi_1/\varphi_3)]^2+[T(\varphi_2/\varphi_3)]^2=T(\varphi_1/\varphi_3)+T(\varphi_2/\varphi_3)=1.$$

Следовательно, либо $|\Delta_l| = 2^{m-1}$ и $|\Delta_r| = 2^{m-1} \pm 2^{(m-1)/2}$, либо $|\Delta_l| = 2^{m-1} \pm 2^{(m-1)/2}$ и $|\Delta_r| = 2^{m-1}$. В любом случае $|\Delta_l| + |\Delta_r| = 2^m \pm 2^{(m-1)/2}$. Ч. Т. Д.

Предыдущее доказательство также показывает, что все кодовые слова с весом 2^m или 2^{m+1} лежат в РМ-коде 1-го порядка; все другие кодовые слова имеют вес $2^m \pm 2^{(m-1)/2}$. Следовательно, нумератор весов нашего кода длины 2^{2l} задается полиномом

$$\mathcal{K}(z) = 1 + K_w z^w + (2^{2l+1} - 2) z^{2^{2l-1}} + K_w z^{2^{2l} - w} + z^{2^{2l}},$$

где $w = 2^{2l-1} - 2^{l-1}$, $K_w = 2^{2l}(2^{2l-1} - 1)$.

Гёталс (1971) заметил, что распределения весов наших кодов двойственны распределениям весов расширенных кодов Препараты (1968) в том смысле, что они удовлетворяют тождеству Мак-Вильямс (1963), а именно:

$$\mathcal{P}(z) = 2^{-4l}(1 + z)^{2^{2l}} \mathcal{K}\left(\frac{1 - z}{1 + z}\right),$$

где $\mathcal{P}/z$ — нумератор весов расширенного кода Препараты, который имеет 2^{4l} кодовых слов длины 2^{2l} и минимальное расстояние 6. Нумератор весов кодов Препараты был впервые определен Семаковым и Зиновьевым (1969). Группы симметрий наших кодов еще не известны, за исключением случая $l = 2$, когда и наша конструкция и конструкция Препараты дают расширенный код Нордстрома — Робинсона (1968) длины 16, чья группа симметрий — одна из тех, которые изучались Берлекэмпом (1971).

Вэлч (1971) показал, что линейное пространство, образованное любым из наших кодов, является полным РМ-кодом 2-го порядка такой же длины. Миккельтвейт (1972) установил, что эти коды систематические и что первые $2l$ разряда в правой и левой половинах кода могут быть использованы в качестве $4l$ информационных разрядов.

Я благодарен профессору Дж. К. Вулфу за просмотр тезисов, которые лежат в основе этой статьи, и Э. Р. Берлекэмпу за помощь в написании этой статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Berlekamp E. R. (1968), Algebraic Coding Theory, McCraw-Hill Book Co., New York. (Русский перевод: Берлекэмп Э., Алгебраическая теория кодирования, «Мир», М., 1971.)
2. Berlekamp E. R. (1971), Coding theory and the Mathieu groups, *Information and Control*, **18**, 40—64.

3. Goethals J. M. (1971). Неопубликованное сообщение.
4. MacWilliams F. J. (1963), A theorem of the distribution of weights in a systematic code, *Bell. Syst. Tech. J.*, **42**, 79--94.
5. Mykkeltveit J. (1972), Note in JPL Deep Space Net Progress report. Не опубликовано.
6. Nordstrom A. W., Robinson J. P. (1967), An optimum nonlinear code. *Information and Control*, **11**, 613—616.
7. Preparata F. P. (1968), A class of optimum nonlinear double-error-correcting codes, *Information and Control*, **13**, № 4, 378—400. (Русский перевод: Кибернетический сборник, вып. 7, «Мир», М., 1970.)
8. Семаков Н. В., Зиновьев В. А. (1969), Равновесные коды и тактические конфигурации, *Проблемы передачи информации*, 5 (3), 28—36
9. Welch L. R. (1971). Неопубликованное сообщение.

Класс конструктивных асимптотически хороших алгебраических кодов[1]

Йёрн Юстесен

1. ВВЕДЕНИЕ

Граница Варшамова — Гилберта [1, 2] гарантирует существование двоичных (n, k)-кодов любой размерности k и любой длины блока n $(0 < k < n)$, таких, что их минимальное расстояние d_n удовлетворяет неравенству

$$H(d_n/n) \geqslant 1 - R_n,$$

где $R_n = k/n$ — скорость, а $H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ — двоичная энтропия. Таким образом, по крайней мере в принципе, можно для любой скорости R, $0 < R < 1$, выбрать такую последовательность линейных кодов с возрастающей длиной блока n, минимальным расстоянием d_n и скоростью $R_n \geqslant R$, что

$$\liminf_{n \to \infty} (d_n/n) \geqslant H^{-1}(1-R), \tag{1}$$

где предполагается $0 < H^{-1}(1-R) < 1/2$, так что обратная функция определяется однозначно.

До сих пор, однако, не была найдена никакая конструктивная последовательность двоичных (n, k)-кодов с $R_n \geqslant R$ и

$$\liminf_{n \to \infty} (d_n/n) > 0 \tag{2}$$

для любой R, $0 < R < 1$. Назовем бесконечную счетную последовательность (n, k)-кодов конструктивной, если она может быть описана с помощью понятий, которые обычно считаются известными в том смысле, что не требуют перебора для своего задания[2]. Таким образом, циклические двоичные коды Хэмминга в указанном смысле представляют собой конструктивную последовательность кодов. Будем называть последовательность кодов, которая удовлетворяет (2), *асимптотически хорошей*.

Во втором разделе этой статьи для любой скорости R, $0 < R < 1$, будет точно описана конструктивная бесконечная

---

[1]) Justesen J., A class of constructive, asymptotically-good, algebraic codes, *IEEE Trans. Inform. Theory*, **IT-18**, № 5 (1972), 652—656.

[2]) Естественно, что многие читатели не будут удовлетворены этим определением. Однако нужно учесть, что основное содержание статьи состоит в предложенном классе кодов. — *Прим. ред.*

счетная последовательность (n, k)-кодов со скоростями $R_n \geqslant R$, для которой имеет место (2). Конструкция основывается на введенном Форни [3] понятии каскадных кодов, в которых m информационных символов внутреннего двоичного кода рассматриваются как единые символы внешнего кода Рида — Соломона [4] над $GF(2^m)$. Однако мы обобщаем это понятие для того, чтобы имелась возможность изменения внутреннего кода. Внутренние коды, как мы их определим, задаются с помощью простого алгебраического описания и оказываются эквивалентными $2^m - 1$ различным кодам из ансамбля случайно сдвинутых кодов, описанных Месси [5] и приписываемых Возенкрафту. Вместе с тем кодовые слова двоичных кодов, которые мы построим, могут быть рассмотрены как пары кодовых слов в различных кодах Рида — Соломона над $GF(2^m)$.

В третьем разделе будет дан метод декодирования кодов, построенных в разд. 2, который близко связан с предложенным Форни [3] декодированием по обобщенному минимальному расстоянию, использованным Редди и Робинсоном [6] и Велдоном [7] для декодирования итеративных кодов. Доказывается, что все ошибки с относительным весом (отнесенным к длине блока), гарантирующим их исправление в соответствии с нижней границей для минимального расстояния из разд. 2, будут исправляться этой процедурой и что число двоичных операций, требуемых для декодирования, пропорционально $n^2 \log n$.

2. КОДЫ

Пусть

$$\boldsymbol{i} = [i_0, i_1, \ldots, i_{K-1}], \quad i_j \in GF(2^m)$$

представляет собой информационную последовательность (N/K)-кода Рида — Соломона (РС) над $GF(2^m)$ с длиной блока $N = 2^m - 1$ и скоростью $r_n = K/N$, и определим связанный с ней полином

$$i(x) = i_0 + i_1 x + \ldots + i_{K-1} x^{K-1}.$$

Пусть α — примитивный элемент $GF(2^m)$. Закодированная последовательность может быть записана в виде **N-последовательности** [8]:

$$\boldsymbol{a} = [a_0, a_1, \ldots, a_{N-1}], \quad a_j \in GF(2^m)$$

или, равносильно, в виде полинома

$$a(x) = a_0 + a_1 x + \ldots + a_{N-1} x^{N-1},$$

где

$$a_j = i(\alpha^j), \quad 0 \leqslant j \leqslant 2^m - 2.$$

Так как $GF(2^m)$ — векторное пространство над $GF(2)$, то элементы $GF(2^m)$ могут, и будут в дальнейшем, рассматриваться как двоичные m-последовательности.

Рассмотрим теперь двоичный (n, k)-код $\mathscr{C}_m$ с кодовыми словами

$$\boldsymbol{c} = [c_0, c_1, \ldots, c_{N-1}], \tag{3}$$

где

$$c_j = [a_j, \alpha^j a_j], \quad 0 \leqslant j \leqslant N-1. \tag{4}$$

Этот двоичный код имеет длину $n = 2mN$, размерность $k = mK$ и скорость $R_n = \frac{1}{2} r_N$. Код $\mathscr{C}_m$ является линейным кодом над $GF(2^m)$, когда c_j рассматриваются как пары элементов из $GF(2^m)$, и, следовательно, является также линейным над $GF(2)$, когда, как было сделано, c_j выражаются в виде 2^m-последовательностей над $GF(2)$.

Код b_m может быть интерпретирован как каскадный код с внешним кодом Рида — Соломона и меняющимся внутренним кодом. Для каждого j (4) определяет $(2m, m)$-двоичный код; все эти коды являются различными и фактически составляют коды ансамбля Возенкрафта случайно сдвинутых кодов для $R = 1/2$. Эти коды обладают интересным свойством, состоящим в том, что любая $2m$-последовательность a, b с $a \neq 0$ и $b \neq 0$ появляется ровно в одном коде, а именно в коде, для которого $\alpha^j = b/a$. Это свойство было использовано Месси [5], чтобы показать, что ансамбль содержит коды, которые лежат на границе Варшамова — Гилберта.

Иным образом кодовые слова $\mathscr{C}_m$ могут быть интерпретированы как пары $[\boldsymbol{a}, \boldsymbol{b}]$, где $\boldsymbol{a}$ и $\boldsymbol{b} = [b_0, b_1, \ldots, b_{N-1}]$ — кодовые слова РС-кодов. Так как РС-коды принадлежат подклассу кодов Боуза — Чоудхури — Хокингема (БЧХ) [8], то эти коды являются циклическими, и РС-код, описанный выше, имеет, как легко видеть, порождающий полином

$$g(x) = (x + \alpha)(x + \alpha^2) \ldots (x + \alpha^{N-K}).$$

Вектор $\boldsymbol{b}$ имеет ассоциированный полином

$$b(x) = a_0 + \alpha a_1 x + \ldots + \alpha^{N-1} a_{N-1} x^{N-1} = a(\alpha x),$$

и, следовательно, он является кодовым словом РС-кода с порождающим полиномом

$$g_b(x) = (x + 1)(x + \alpha) \ldots (x + \alpha^{N-K-1}).$$

Принято обычно считать конструктивной последовательность кодов, определяемую с помощью примитивных элементов последовательно расширяющихся полей $GF(2^m)$. Таким образом,

последовательность определенных выше кодов $\mathscr{C}_m$ является конструктивной в том же самом смысле, как и циклические коды Хэмминга или РС-коды. Однако, следуя предложению Мак-Элиса [9] при определении b_m, можно избежать задания примитивного элемента $GF(2^m)$ или, что эквивалентно, примитивного полинома степени m над $GF(2)$, если поступить следующим образом. Пусть $m = 2 \cdot 3^l$-последовательности

$$[f_0, f_1, \ldots, f_{m-1}]$$

соответствует

$$f_0 + f_1\sigma + \ldots + f_{m-1}\sigma^{m-1},$$

где σ — корень неприводимого, но не примитивного полинома [10]

$$F_1(x) = x^{2 \cdot 3^l} + x^{3^l} + 1, \quad \text{т. е.} \quad \sigma^{2 \cdot 3^l} + \sigma^{3^l} + 1 = 0.$$

Если заменить везде α^j на $m = 2 \cdot 3^l$-последовательность, символы которой дают запись в двоичной форме числа j, $0 \leqslant j < 2^{2 \cdot 3^l}$, то определенные таким образом коды для заданных значений $m = 2 \cdot 3^l$ являются в действительности перестановками соответствующей последовательности кодов, определяемых (3) и (4). Эта последовательность кодов является, как мы верим, бесспорно конструктивной.

Коды (без модификации Мак-Элиса) легче всего кодируются РС-кодером, который выдает $a_{N-1}, a_{N-2}, \ldots, a_0$, с последующим кодированием кодером внутренних кодов. Внутренний кодер первоначально хранит множитель α^{N-1}, который затем нормируется умножением на α^{-1} после каждого кодирования и, таким образом, вычисляет произведение $b_j = \alpha^j a_j$. Эти $2N$ умножений в $GF(2^m)$ во внутреннем кодере дают небольшую добавку к приблизительно $N(N-K)$ умножениям, требуемым для РС-кодера.

Теорема 1. *При любой заданной скорости* R, $0 < R < 1/2$, *для последовательности двоичных* $(2mN, mK)$, $m = 1, 2, \ldots$, *кодов* $\mathscr{C}_m$ *со скоростью* $R_n = \frac{1}{2} r_N = \frac{1}{2} \frac{K}{N}$, *выбираемой так, чтобы быть наименьшей скоростью, для которой* $R_n \geqslant R$, *справедливо*

$$\lim_{n \to \infty} \inf \frac{d_n}{n} \geqslant (1 - 2R) H^{-1}\left(\frac{1}{2}\right) \approx 0{,}11\,(1 - 2R), \qquad (5)$$

где d_n — *минимальное расстояние* $\mathscr{C}_m$.

Нижняя граница (5) нанесена на рис. 1 для сравнения вместе с границей Варшамова — Гилберта. Так, например, при $R = 1/4$ коды из теоремы 1 имеют минимальное расстояние, по меньшей мере равное 5,5% от длины блока, в то время как

согласно границе Варшамова — Гилберта существуют коды с минимальным расстоянием, равным 21,5% от длины блока.

При доказательстве теоремы 1 будет использована следующая лемма.

Лемма. *Пусть* $o_2(L) \to 0$ *при* $L \to \infty$. *Тогда для любого* γ, $0 < \gamma < 1$, *и любого* δ, $0 < \delta < 1$, *суммарный хэмминговский*

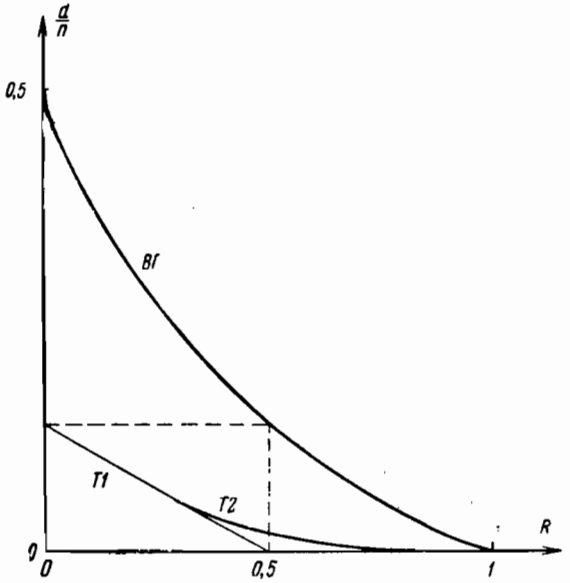


Рис. 1. Сравнение границ теоремы 1 (Т1) и теоремы 2 (Т2) с границей Варшамова — Гилберта (ВГ).

вес W *множества из* $M_L = [\gamma - o_2(L)](2^{L\delta} - 1)$ *различных ненулевых двоичных* L*-последовательностей удовлетворяет неравенству*

$$W \geqslant \gamma L \left[H^{-1}(\delta) - o_1(L)\right](2^{L\delta} - 1), \tag{6}$$

где $o_1(L) \to 0$ *при* $L \to \infty$.

Доказательство. Используем хорошо известное неравенство [8]

$$\sum_{i=0}^{\lambda L} \binom{L}{i} \leqslant 2^{LH(\lambda)}, \quad 0 \leqslant \lambda \leqslant \frac{1}{2}. \tag{7}$$

Отсюда следует, что доля f_L тех указанных ненулевых L-последовательностей из M_L, которые имеют хэмминговский вес, меньший или равный λL, удовлетворяет неравенству

$$f_L \leqslant M_L^{-1} 2^{LH(\lambda)}, \quad 0 \leqslant \lambda \leqslant 1/2,$$

или

$$f_L \leqslant [\gamma - o_2(L)]^{-1} (2^{L\delta} - 1)^{-1} 2^{LH(\lambda)}. \quad (8)$$

Если выбрать

$$\lambda = H^{-1}(\delta - 1/\log L),$$

где L достаточно велико, чтобы $\delta > 1/\log L$, то (8) принимает вид

$$f_L \leqslant [\gamma - o_2(L)]^{-1} 2^{-L/\log L},$$

и, таким образом, $f_L = o_3(L)$, где $o_3(L) \to 0$ при $L \to \infty$. Следовательно, суммарный вес L-последовательностей из M_L удовлетворяет неравенству

$$W \geqslant (1 - f_L) M_L H^{-1}(\delta - 1/\log L) L$$

или

$$W \geqslant [1 - o_3(L)][\gamma - o_2(L)](2^{L\delta} - 1) H^{-1}(\delta)[1 - o_4(L)] L, \quad (9)$$

которое при

$$\gamma - o_1(L) = [1 - o_3(L)][1 - o_4(1)][\gamma - o_2(L)]$$

приводит к (6).

Заметим, что так как никакие два случайно сдвинутых кода не имеют общих ненулевых кодовых слов, то эта лемма может быть интерпретирована так: доля f_L из M_L случайно сдвинутых кодов, которые имеют минимальное расстояние, меньшее или равное $LH^{-1}(\delta - 1/\log L)$, стремится к нулю при $L \to \infty$.

Доказательство теоремы 1. Рассмотрим какое-либо ненулевое кодовое слово $\boldsymbol{c}$ из $\mathscr{C}_m$. Тогда $\boldsymbol{a}$ также является ненулевым, и, поскольку оно является кодовым словом РС-кода, оно содержит по меньшей мере $N - K + 1 > N - K = N(1 - r_N) = N(1 - 2R_n) = (2^m - 1)[1 - 2R - o_2(m)]$ ненулевых символов, где $o_2(m) \to 0$ при $m \to \infty$. Так как никакая из $2m$-последовательностей не появляется более чем в одном внутреннем коде, то можно применить лемму с $L = 2m$, $\delta = 1/2$ и $\gamma = 1 - 2R$. Хэмминговский вес $\boldsymbol{c}$ ограничен снизу неравенством

$$W \geqslant \left[H^{-1}\left(\frac{1}{2}\right) - o_1(2m)\right](1 - 2R)(2^m - 1) 2^m.$$

Следовательно, минимальное расстояние d_n кода $\mathscr{C}_m$ удовлетворяет неравенству

$$d_n/n = d_n/[2m(2^m - 1)] \geqslant (1 - 2R)\left[H^{-1}\left(\frac{1}{2}\right) - o_1(2m)\right]. \quad (10)$$

Так как $m \to \infty$ при $n = 2m(2^m - 1) \to \infty$, то (5) следует непосредственно из (10), и теорема доказана.

Заметим, что этот результат нельзя получить, рассматривая РС-код просто как двоичный код. Пусть d_0 — минимальное расстояние РС-кода над $GF(2^m)$, и пусть i — наименьшее целое число, такое, что $d_0' = 2^i - 1 \geqslant d_0$. Как было доказано Касами, Лином и Питерсоном [11], БЧХ-код длины $2^m - 1$ с конструктивным расстоянием d_0' имеет минимальное расстояние, в точности равное d_0'. Теперь, как показано в [3], БЧХ-коды с конструктивным расстоянием, по крайней мере равным d_0, являются подкодами РС-кода с минимальным расстоянием d_0, и, следовательно, их минимальное расстояние является границей сверху для минимального расстояния $d_{\text{ДРС}}$ РС-кода, рассматриваемого как двоичный код длины $n = mN$. Так как $d_0' \leqslant 2d_0$, то

$$d_{\text{ДРС}}/n = d_{\text{ДРС}}/(mN) < 2(N - K)/(mN)$$

и, следовательно,

$$d_{\text{ДРС}}/n \to 0 \quad \text{при} \quad m \to \infty.$$

Для того чтобы распространить конструкцию на бо́льшие скорости, произведем укорачивание построенных ранее кодов с помощью отбрасывания последних s символов из каждого внутреннего кода. Для образующейся скорости внутреннего кода получается выражение

$$r_n = m/(2m - s), \quad 0 \leqslant s < m.$$

При заданном значении R, $0 < R < 1$, и произвольном выборе r_n выберем скорость РС-внешнего кода как наименьшую скорость, такую, что

$$R_n = r_n r_N \geqslant R.$$

Заметим вновь, что любое ненулевое кодовое слово $\boldsymbol{c}$ в укороченном коде $\mathscr{C}_m$ будет содержать

$$M > (2^m - 1)(1 - r_N) = (2^m - 1)[1 - (R/r_n) - o_2(m)]$$

ненулевых a_j. Ненулевые (a_j, b_j) различны, но после укорачивания могут оказаться 2^s одинаковых ненулевых c_j, так как любое c_j является кодовым словом в точности в 2^s внутренних кодах. Таким образом, по крайней мере

$$2^{-s}M > (2^{m-s} - 1)[1 - (R/r_n) - o_3(m)]$$

ненулевых $(2m - s)$-последовательностей различны, и можно применить лемму с $L = 2m - s$, $\delta = (m - s)/L = 1 - r_n$ и $\gamma = 1 - (R/r_n)$. Получим следующую нижнюю границу для хэмминговского веса $\boldsymbol{c}$:

$$W \geqslant [1 - (R/r_n)](2m - s)[H^{-1}(1 - r_n) - o_1(m)](2^{m-s} - 1)2^s. \quad (11)$$

Но если r_n стремится к r, $1/2 \leqslant r < 1$, то отношение s/m отделено от 1 и (11) означает, что

$$\liminf_{n\to\infty} d_n/n \geqslant [1 - (R/r)]\, H^{-1}(1-r). \tag{12}$$

Минимальное расстояние может быть максимизировано приравниванием нулю производной по r правой части неравенства (12). Максимизирующее значение r удовлетворяет условию

$$R = \frac{r^2}{1 + \log_2 [1 - H^{-1}(1-r)]}, \tag{13}$$

с тем исключением, что следует положить $r = 1/2$, когда решение (13) будет таким, что $r < 1/2$, так как наша конструкция допускает лишь скорости внутреннего кода, бóльшие или равные $1/2$.

Резюмируем сказанное в виде теоремы.

Теорема 2. *При любой скорости R, $0 < R < 1$, для двоичных (n, k)-кодов $\mathscr{C}_m$, укороченных так, что r_n стремится к r, где r — максимум из $1/2$, и решения уравнения* (13) *справедливо*

$$\liminf_{n\to\infty} d_n/n \geqslant [1 - (R/r)]\, H^{-1}(1-r). \tag{14}$$

Граница (14) может быть интерпретирована как огибающая семейства прямых линий, проведенных следующим образом. Точка $[r, H^{-1}(1-r)]$ на границе Варшамова — Гилберта проектируется на оси, давая точки $[r, 0]$ и $[0, H^{-1}(1-r)]$. Эти две точки соединяются затем прямой линией. В силу того что имеется ограничение $r \geqslant 1/2$, граница теоремы 2 совпадает с границей теоремы 1 при $0 < R < 0{,}30$. Для бо́льших скоростей граница (14) совпадает с нижней границей для минимального расстояния в ансамбле каскадных кодов, полученной Зябловым [12]. Для скоростей $r < 0{,}30$ граница расположена ниже границы Зяблова, так как конструкция разд. 2 требует хорошего ансамбля внутренних кодов с самое большее $2^m - 1$ кодами, и мы не можем конструктивно описать такой ансамбль для скоростей, меньших чем $1/2$.

Заметим, что для скоростей, близких к единице, максимизирующее значение r имеет вид

$$r \approx R_n \approx R^{1/2}.$$

3. ПРОЦЕДУРА ДЕКОДИРОВАНИЯ

Все ошибки веса, меньшего чем половина значения нашей асимптотической нижней границы для минимального расстояния (14), могут быть исправлены процедурой декодирования,

которая является одной из модификаций декодирования по обобщенному минимальному расстоянию, использованной Редди и Робинсоном [6] и Велдоном [7] для произведения кодов.

Как было указано ранее, все коды, кроме стремящейся к нулю доли внутренних кодов, имеют минимальное расстояние d_i, удовлетворяющее неравенству

$$\frac{d_i}{2m-s} \geqslant \frac{D}{2m-s} = H^{-1}(1-r_n),$$

где D — нижняя граница для минимального расстояния, задаваемая (1).

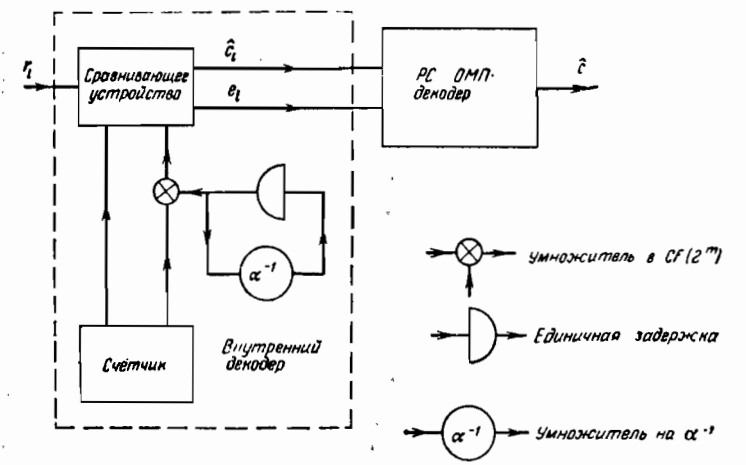


Рис. 2. Функциональная схема декодера (n, k)-кодов теоремы 2.

Для нашей процедуры декодирования мы выберем схему, изображенную на рис. 2, в которой принимаемый блок

$$\boldsymbol{r} = [r_0, r_1, \ldots, r_{N-1}],$$

где r_i — принимаемая $(2m-s)$-последовательность, соответствующая внутреннему кодовому слову c_i и сначала декодируемая внутренним декодером, решения которого затем поступают к декодеру по обобщенному минимальному расстоянию (ОМР) [3] для РС-внешнего кода. Внутренний декодер может быть построен так, как показано на рис. 2, т. е. так, что для каждого принимаемого внутреннего слова r_i декодер (с помощью m-разрядного двоичного кольцевого счетчика) генерирует все 2^m кодовых слов внутреннего кода и подает их на сравнивающее устройство, которое также принимает r_i. Внутренний декодер декодирует r_i в $\hat{c}_i$, как только кодовое слово $\hat{c}_i$ обнаруживается на расстоянии, меньшем чем $D/2$ от r_i. Так как

этот процесс декодирования должен быть выполнен $N=2^m-1$ раз, то внутренний декодер выполняет всего $2^m(2^m-1)$ или приближенно $(n/\log n)^2$ умножений.

Пусть выходом сравнивающего устройства будет число e_i вместе с оценкой $\hat{c}_i$ для c_i, где e_i определяется следующим образом:

$$e_i=\begin{cases}\text{вес } (\hat{c}_i+r_i), & \text{если } r_i \text{ декодируется,}\\ D/2 & \text{в остальных случаях (в этом случае } \hat{c}_i \text{ может быть произвольным).}\end{cases}$$

Определим нормированный вес ошибки в позиции i следующим образом:

$$\beta_i=\begin{cases}e_i/D, & \text{если } \hat{c}_i=c_i,\\ \dfrac{D-e_i}{D}, & \text{если } \hat{c}_i\neq c_i.\end{cases} \tag{15}$$

В соответствии с теорией ОМР [3] РС-внешний **декодер** будет декодировать верно всегда, когда

$$\sum_{i=0}^{N-1}\beta_i<\frac{N-K+1}{2}. \tag{16}$$

Заметим, что в силу леммы доля r_i, дающих ошибки из-за того, что $d_i<D$, стремится при $N\to\infty$ к нулю при $d_i<D$ и что число ошибок, требующихся для того, чтобы вызвать заданное значение β_i для внутреннего кода с минимальным расстоянием не менее D, равно $t_i\geqslant\beta_i D$. Учитывая это, из (16) получаем, что минимальное число ошибок, которое может вызвать ошибку декодирования, асимптотически удовлетворяет неравенству

$$t=\sum_{i=0}^{N-1}t_i\geqslant\sum_{i=0}^{N-1}\beta_i D>(D/2)(N-K+1)>\tfrac{1}{2}DN(1-r_N). \tag{17}$$

Но асимптотически наша нижняя граница d_B для минимального расстояния d имеет вид

$$d_B=ND(1-r_N),$$

так что (17) показывает, что наша процедура декодирования исправляет все ошибки, исправление которых гарантируется асимптотической нижней границей d_B.

Заметим также, что для того, чтобы произвести ОМР-декодирование внешнего РС-кода, требуется число попыток, равное числу различных значений e_i. Так как существуют $D/2$ допустимых значений e_i и так как предложенный Берлекэмпом

[13] итеративный алгоритм декодирования ошибок и стираний в применении к РС-коду требует число умножений, пропорциональное 2^{2m}, то отсюда следует, что общее число умножений в нашей процедуре декодирования пропорционально $m2^{2m}$. Это соответствует числу двоичных операций, пропорциональному $m^3 2^{2m}$ или, приближенно, $n^2 \log n$.

4. ЗАМЕЧАНИЯ

Конструкция кода, задаваемая теоремой 2 для двоичных кодов, легко может быть обобщена для того, чтобы получить конструкции асимптотически хороших кодов над любым конечным полем $GF(q)$. В этом случае следует использовать РС-код над расширенным полем $GF(q^m)$ и выбрать α в (4) как примитивный элемент этого поля. Модификация Мак-Элиса, однако, не может быть использована для устранения последнего возможного возражения к конструктивности при $q > 2$, так как мы не знаем выражения для неприводимых полиномов произвольно большой степени над произвольным полем $GF(q)$.

Наконец, отметим, что для любой заданной длины блока $n = (2m - s)2^m$ поиск наилучшего фиксированного внутреннего кода требует меньше, чем n^2, операций. Несмотря на то что хороший каскадный код с наилучшим фиксированным внутренним кодом может быть найден таким образом с помощью весьма умеренного перебора, мы отвергли этот подход как неконструктивный, так как мы не можем описать внутренний код априори. Открытым остается вопрос, является ли истинное расстояние наших конструктивных кодов с различными внутренними кодами лучше или хуже, чем то, которое получается при использовании наилучшего фиксированного «внутреннего» кода.

Автор пользуется возможностью поблагодарить проф. Дж. Л. Месси за помощь в подготовке окончательного варианта этой статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Gilbert E. N., A Comparison of Signaling Alphabets, BSTJ, **31** (1952), 504—522.
2. Варшамов Р. Р., Оценка числа сигналов в кодах с коррекцией ошибок, ДАН, **117** (1957), № 5, 739—741.
3. Forney D. G., Concatenated Codes. MIT Press, Cambridge, Mass., 1966. (Русский перевод: Форни Д. Г., Каскадные коды, М., «Мир», 1970.)
4. Reed I. S., Solomon G., Polynomial Codes over Certain Finite Fields, *J. Soc. Indst. Appl. Math.*, 8 (1960), 300—304.
5. Massey J. L., Threshold Decoding, MIT Press, Cambridge, Mass., 21 (1963). (Русский перевод: Месси Дж., Пороговое декодирование, М., «Мир», 1966.)
6. Reddy S. M., Robinson J. P., Random Error and Burst Correction by Iterated Codes, *IEEE Trans. Inform. Th.*, **IT-18**, January 1972, 182—185.

7. Weldon E. J., Jr., Decoding Binary Block Codes on Q-ary Output Channels, *IEEE Trans. Inform. Th.*, **IT-17**, November 1971, 717—718 (Appendix).
8. Peterson W. W., Error Correcting Codes, MIT Press, and John Wiley, Cambridge, Mass., 1961. (Русский перевод: Питерсон У., Коды, исправляющие ошибки, М., «Мир», 1964.)
9. McEliece R. J., Частное сообщение, 1972.
10. Colomb S. W., Shift Register Sequences, Holden-Day, San Francisco, 1967, p. 96.
11. Kasami T., Lin S., Peterson W. W., Some Results on Weight Distributions of BCH Codes, *IEEE Trans. Inform. Th.*, **IT-12**, April 1966, p. 274.
12. Зяблов В. В., Оценка сложности построения двоичных линейных каскадных кодов, Проблемы передачи информации, 7 (1971), вып. 1, 5—13.
13. Berlekamp E. R., Algebraic Coding Theory, McGraw-Hill, New York, 1968, p. 184. (Русский перевод: Берлекэмп Э., Алгебраическая теория кодирования, М., «Мир», 1971.)

О связи теории графов с планированием эксперимента и некоторые последние результаты о существовании блок-схем [1])

Д. К. Рой-Чоудхури

1. ВВЕДЕНИЕ

Для того чтобы сравнить действия ряда способов обработки (элементов), экспериментатор выбирает некоторое множество опытов (экспериментальных единиц), распределяет способы обработки по опытам в соответствии с планом эксперимента и через некоторое время наблюдает отклики на поставленные опыты, на основе которых делает необходимые выводы. Наблюдаемые отклики подвергаются также воздействиям внешних факторов, изменяющихся от опыта к опыту. Так как наблюдаемые отклики изменяются даже тогда, когда эксперимент повторяется в одинаковых условиях, то экспериментатор, естественно, выбирает вероятностную модель. Предполагается, что отклики — действительно переменные величины, распределение которых зависит от применяемых способов обработки (элементов), а также от ряда внешних факторов. Само планирование эксперимента состоит в выборе заданного числа опытов и приписывании элементов этим опытам. Количество информации, извлекаемой из эксперимента, может быть увеличено, если более тщательно анализировать влияние внешних факторов. Оптимальное планирование эксперимента часто требует сложного упорядочения опытов в соответствии с различными факторами, и такие упорядочения основаны на комбинаторных конфигурациях или комбинаторных схемах.

Как графы, так и комбинаторные схемы являются частными случаями структур инциденций (incidence structures). В некоторых случаях графы являются более простыми структурами инциденций, чем комбинаторные схемы. Но часто задачи, сформулированные в терминах комбинаторных схем, можно перевести на задачи с графами.

В последнее время некоторые авторы использовали результаты теории графов для доказательства теорем существования или несуществования комбинаторных схем. В этой статье дан

---

[1]) Ray Chaudhuri D. K., On Some Connections between Graph Theory and Experimental Designs and Some Recent Existence Results в сборнике «Graph Theory and its Applications, Proc. of an Advanced Seminar at the University of Wisconsin, Madison, October 13—15, 1969, Academic Press, New York — London, 1970, стр. 149—166.

обзор таких теорем без доказательств. Здесь сделана также попытка формализовать и пояснить некоторые основные понятия планирования эксперимента.

2. ОПИСАНИЕ ПЛАНИРОВАНИЯ ЭКСПЕРИМЕНТА

Эксперимент определяется с помощью множества элементов $\mathfrak{T} = (T_1, T_2, \ldots, T_v)$ и множества опытов $\mathfrak{E} = (E_1, E_2, \ldots, E_v)$. Опыты не все тождественны. Множество внешних переменных $X = (x_1, x_2, \ldots, x_p)$ принимает различные значения на множестве $\mathfrak{E}$. $H \subseteq R_p$ (R_p — p-мерное евклидово пространство), H есть множество допустимых значений X. Для каждых $T \in \mathfrak{T}$ и $a \in H$ пусть $\mathfrak{F}_{T,a}$ — некоторый заданный класс функций распределения вероятностей. Эксперимент определяется с помощью отображения D: $\mathfrak{E} \to \mathfrak{T}$. Отображение D обычно называется планом эксперимента. Элемент $D(E)$ применяется в опыте E и затем наблюдается отклик $y(D, E)$. Предполагается, что $y(D, E)$ есть случайная величина с функцией распределения, принадлежащей классу $\mathfrak{F}_{T,a}$, где $D(E) = T$ и $X(E) = a$. Цель эксперимента — принять решение о размещении множества элементов, т. е. выбрать классы функций распределения $\{\mathfrak{F}_{T,a}: T \in \mathfrak{T}, a \in H\}$.

Сама «функция решения» является функцией случайных переменных $Y(D, E)$, $E \in \mathfrak{E}$, и отсюда «качество» (goodness) функции решения зависит от самого плана эксперимента D.

Ниже приводится пример с выбором наилучшей диеты в эксперименте по определению рациона свиней. Элементами здесь являются четыре различные диеты. Они «применяются» к свиньям, которые обозначены пятью различными буквами. Внешняя переменная X принимает соответственно пять различных значений. Откликом является увеличение веса свиньи, предполагается, что y_{ij} — увеличение веса j-й свиньи с i-й буквой — есть случайная величина со средним значением $\beta_i + \tau_j$, где β_i представляет эффект i-й буквы, а τ_j — эффект j-й диеты ($i = 1, \ldots, 5$; $j = 1, 2, 3, 4$). Предполагается также, что дисперсия y_{ij} равна σ^2 и y_{il} и $y_{i'l'}$ некоррелированы для $(i, j) \neq (i', j')$, $i, i' = 1, \ldots, 5$; $j, j' = 1, 2, 3, 4$.

3. СТРУКТУРЫ ИНЦИДЕНЦИЙ И BIB-СХЕМЫ

Структура инциденций — это тройка (P, L, I), где P — множество объектов, называемых точками, элементами (способами обработки) или вершинами, L — другое множество объектов, называемых линиями, блоками или ребрами, а I — подмножество $P \times L$. Предполагается, что P и L не пересекаются. Точки p и l называются инцидентными, если $p \in P$, $l \in L$, $(p, l) \in I$. Графом является такая структура инциденций, в которой любое

ребро инцидентно одной или двум вершинам. Ребро, инцидентное одной вершине, называется петлей. Если двум вершинам p_1 и p_2 инцидентны два ребра l_1 и l_2 (или более), то говорят, что между p_1 и p_2 имеются кратные ребра. Две вершины p_1 и p_2 (не обязательно различные) смежны тогда и только тогда, когда существует ребро, инцидентное p_1 и p_2. Граф без петель и кратных ребер можно определить как $G=(P, E)$, где $E \subseteq P^{(2)}$, а $P^{(2)}$ — множество неупорядоченных пар элементов P. Неупорядоченная пара $(p_1, p_2) \in E$, если и только если существует ребро l, инцидентное p_1 и p_2.

Пусть v и λ — положительные целые числа, K — множество положительных целых чисел. Структура инциденций (P, L, I) называется (v, K, λ)-попарно сбалансированной схемой (PBD), если: (i) $|P|=v$; (ii) для любого $l \in L$ точка l инцидентна k точкам и $k \in K$; (iii) для $p_1, p_2 \in P$, $p_1 \neq p_2$ существует ровно λ различных линий l_i, $i=1, 2, \ldots, \lambda$, таких, что каждое ребро l_i инцидентно как p_1, так и p_2.

Если K состоит из одного целого числа k, то PBD называется (v, k, λ)-BIB-схемой. Обычно линия (или блок) l отождествляется с множеством точек $A_l=\{p \mid (p, l) \in I\}$. Тогда BIB-схему можно рассматривать как пару $(P, \mathfrak{A})$, где $\mathfrak{A}$ — семейство блоков $(A_l: l \in L)$. Ниже приводится BIB-схема с $v=7$, $k=3$, $\lambda=1$, $P=\{1, 2, 3, 4, 5, 6, 7\}$:

$$A_1=(1, 2, 3), \quad A_2=(1, 4, 5), \quad A_3=(1, 6, 7), \quad A_4=(2, 4, 6),$$
$$A_5=(2, 5, 7), \quad A_6=(3, 4, 7), \quad A_7=(3, 5, 6).$$

В (v, k, λ)-BIB-схеме число блоков должно равняться $b=\frac{\lambda v(v-1)}{k(k-1)}$, а каждый элемент (или точка) должен входить в $r=\frac{\lambda(v-1)}{k-1}$ блоков. В (v, k, λ)-BIB-схеме определяется множество bk опытов, в которых одна внешняя переменная (или фактор) X принимает b различных значений $a_1, a_2, \ldots, a_b$ и v элементов образуют множество $P=\{p_1, p_2, \ldots, p_v\}$. «Группа» $B_i=\{E \mid X(E)=a_i\}$ состоит из k опытов (экспериментальных единиц), $i=1, 2, \ldots, b$; $\mathfrak{S}=\bigcup_{i=1}^{b} B_i$; b «групп» опытов сначала приписываются b блокам (v, k, λ)-BIB-схемы, а затем k единиц каждой группы ставятся в соответствие k элементам выделенного блока. Более формально, пусть $I_b=\{1, 2, \ldots, b\}$. Пусть σ — взаимно однозначное отображение $I_b \to I_b$, а τ_i — взаимно однозначное отображение $B_i \to A_{\sigma(i)}$. Для каждого выбора σ и τ_i $(i=1, 2, \ldots, b)$ получаем BIB-схему D эксперимента, определенную с помощью $D(E)=\tau_i(E)$, где $E \in B_i$. В литературе по планированию эксперимента часто не делается различия между комбинаторной схемой и схемой эксперимента. Чтобы

сделать понятия более ясными, нужно ввести, например, следующие разграничения.

(7, 3, 1)-BIB-схему можно использовать в эксперименте для выбора «наилучшего» вида среди семи различных видов мороженого (семь элементов), когда пробу снимают семь дегустаторов, каждый из которых последовательно пробует три из них и дает им оценки. Оценка, данная каждому мороженому, зависит как от его качества, так и от дегустатора. В этом эксперименте семь дегустаторов представляют семь «уровней» (или значений) внешнего фактора. Однако кажется естественным предположить, что различие оценок, которые получила пара элементов (два типа мороженого), не зависит от дегустатора. При этом предположении эксперимент дает больше информации, если выполняется условие, что каждая пара видов мороженого испытывается одинаковое число раз. Это свойство можно получить с помощью использования BIB-схемы эксперимента.

Кифер [8] показал, что в определенном классе схем эксперимента BIB-схемы оптимальны относительно ряда критериев оптимальности. Пусть $\mathfrak{E} = \{E_1, E_2, \ldots, E_v\}$ — множество v опытов ($v = vr$), пусть x — внешний фактор, принимающий b значений (или уровней) a_i, $i = 1, 2, \ldots, b$ и $P = \{p_1, p_2, \ldots, p_v\}$ — множество, содержащее v элементов. Пусть также $\beta = (\beta_1, \beta_2, \ldots, \beta_b)$ и $\tau = (\tau_1, \tau_2, \ldots, \tau_v)$ — соответственно b-мерные и v-мерные векторы с компонентами, принимающими действительные значения, σ^2 — положительное действительное число. Для любого целого n обозначим через $\mathbb{R}^n$ n-мерное пространство действительных переменных; $\mathbb{R}^+$ — множество положительных действительных чисел. Для $\beta \in \mathbb{R}^b$, $\tau \in \mathbb{R}^v$ и $\sigma^2 \in \mathbb{R}^+$ обозначим через $Y_{\beta, \tau, \sigma^2}$ класс всех v-мерных случайных переменных $y = (y_{ij}/i = 1, 2, \ldots, b;\ j = 1, 2, \ldots, v)$, таких, что:

(i) математическое ожидание y_{ij} равно $\beta_i + \tau_j$;

(ii) дисперсия y_{ij} равна σ^2;

(iii) ковариация y_{ij} и $y_{i'j'}$ равна 0 всегда, когда $(i, j) \neq (i', j')$, $i, i' = 1, 2, \ldots, b$; $j, j' = 1, 2, \ldots, v$.

Схема D является отображением $\mathfrak{E} \to P$. Если применяется элемент $D(E)$, то отклик $y(D, E)$ является случайной переменной с таким же распределением, как и y_{ij}, где $D(E) = p_j$ и $X(E) = a_i$, $i = 1, 2, \ldots, b$; $j = 1, 2, \ldots, v$.

„Параметр" β_i обозначает эффект i-го уровня внешнего фактора x, а τ_j — эффект j-го элемента p_j. Пусть $\pi = \sum_{j=1}^{v} a_j \tau_j$, а $a_j \in \mathbb{R}$. Говорят, что „параметрическая функция" π оценивается относительно данной схемы D, если и только если существуют

действительные числа $C(E)$, $E \in \mathscr{E}$, такие, что для каждых $\beta \in \in \mathbb{R}^b$, $\tau \in \mathbb{R}^v$, $\sigma^2 \in \mathbb{R}^+$ и $y \in Y_{\beta, \tau, \sigma^2}$ математическое ожидание $\sum\limits_{E \in \mathscr{E}} C(E)\, y(D, E)$ равно $\sum\limits_{j=1}^{v} a_j \tau_j$. Параметрическую функцию π называют контрастом элемента, если $\sum\limits_{j=1}^{v} a_j = 0$. Пусть $\mathfrak{D}$ множество всех схем D эксперимента, для которых можно оценить каждый контраст элемента. Если существует (v, k, λ)-BIB-схема, то любая BIB-схема D_0 эксперимента принадлежит $\mathfrak{D}$ и обладает рядом оптимальных свойств. Приведем здесь только два из них [8]. Пусть $\pi_1, \pi_2, \ldots, \pi_{v-1}$ — множество $(v-1)$ линейно независимых контрастов, а $\hat{\pi}_1, \hat{\pi}_2, \ldots, \hat{\pi}_{v-1}$ — соответствующие несмещенные оценки с минимумом дисперсии. Ковариационная матрица этих оценок обозначается через $\sigma^2 V_D$. Величина, обратная определителю $|V_D|$, есть мера качества схемы D. Аналогично величина $\lambda(V_D)$, обратная наибольшему собственному значению V_D, есть другая мера качества схемы. Кифер показал, что для любой BIB-схемы эксперимента D_0 имеет место

$$|V_{D_0}| = \min_{D \in \mathfrak{D}} |V_D|, \qquad \lambda(V_{D_0}) = \min_{D \in \mathfrak{D}} \lambda(V_D).$$

4. КВАДРАТЫ ЮДЕНА И ТЕОРЕМА КЕНИГА — ЭГЕРВАРИ

Исторически связь между теорией графов и планированием эксперимента возникла в результате изучения квадратов Юдена (сокращенное YS от Youden Square).

(v, k, λ)-BIB-схема является симметричной тогда и только тогда, когда $b = v$, где b обозначает число блоков и равно $\dfrac{\lambda v (v-1)}{k(k-1)}$.

(v, k, λ)-YS-схема представляет собой размещение v элементов в v строках и k столбцах, такое, что любой элемент появляется в каждом столбце точно один раз и каждая пара различных элементов появляется совместно ровно в λ различных строках.

Следующая схема является (7, 3, 1)-YS-схемой:

1	2	3
2	4	6
3	6	5
4	5	1
5	7	2
6	1	7
7	3	4

Ясно, что строки (v, k, λ)-YS-схемы образуют блоки некоторой симметричной (v, k, λ)-BIB-схемы. Обратно, блоки симметричной (v, k, λ)-BIB-схемы могут быть упорядочены и преобразованы в строки (v, k, λ)-YS-схемы. Этот факт впервые был

установлен Смитом и Хартли [12]. Теперь ясно, что доказательство этого можно получить из теоремы Кёнига — Эгервари.

Пусть N — матрица инциденций порядка v BIB-схемы, состоящей из элементов, распределенных в блоки, т. е. $N = \|n_{ij}\|$, где $n_{ij} = 1$ (или 0), если и только если i-й блок содержит (или не содержит) j-й элемент; $i, j = 1, 2, \ldots, v$. Легко показать, что сумма по каждой строке и сумма по каждому столбцу матрицы инциденций равны k. Из теоремы Кёнига — Эгервари может быть установлено, что $N = P_1 + P_2 + \ldots + P_k$, где P_i являются матрицами перестановок (см. [11], стр. 58). Чтобы построить YS-схему на множестве объектов $I_v = \{1, 2, \ldots, v\}$, определим $y_{il} = j$, когда матрица перестановок P_l имеет 1 в i-й строке и j-м столбце ($i = 1, 2, \ldots, v$; $l = 1, 2, \ldots, k$), и зададим YS-схему с помощью $Y = \|y_{il}\|$.

YS-схемы полезны для построения эффективных экспериментальных планов в тех случаях, когда опыты подвержены влиянию двух внешних факторов. Пусть имеется vk опытов $E_1, E_2, \ldots, E_v$, где $v = vk$, и два внешних фактора X_1 и X_2, при этом X_1 принимает v значений (или уровней) $a_1, a_2, \ldots, a_v$, а X_2 принимает k значений $c_1, c_2, \ldots, c_k$. Имеется точно один опыт E_{ij}, в котором $X_1(E) = a_i$ и $X_2(E) = c_j$ ($i = 1, 2, \ldots, v$; $j = 1, 2, \ldots, k$). В YS-схеме эксперимента строки и столбцы соответственно связаны с понятием уровней первого и второго внешних факторов. Более формально, пусть определены отображения $\tau: I_v \to I_v$ и $\sigma: I_k \to I_k$. Пусть также $Y = \|y_{il}\|$ ($i = 1, 2, \ldots, v$; $l = 1, 2, \ldots, k$) является YS-схемой. Тогда YS-схема эксперимента будет задаваться отображением $D: \mathfrak{E} \to I_v$, определенным следующим образом: $D(E) = y_{\tau(i),\,\sigma(l)}$, где $X_1(E) = a_i$ и $X_2(E) = c_l$, $i = 1, 2, \ldots, v$; $l = 1, 2, \ldots, k$. Здесь I_v — множество элементов. (7, 3, 1)-YS-схему можно использовать в эксперименте по выбору вида мороженого. Оценки, данные мороженому, по-видимому, будут зависеть от порядка, в котором отдельные эксперты пробуют мороженое. Поэтому целесообразно предположить, что имеется два внешних фактора: X_1 соответствует экспертам и X_2 соответствует порядку проведения дегустирования. X_1 принимает семь значений, а X_2 — три значения. Кифер доказал, что YS-схема эксперимента обладает оптимальными свойствами, подобными свойствам BIB-схемы.

5. ОПИСАНИЕ СИММЕТРИЧНЫХ BIB-СХЕМ В ТЕРМИНАХ ТЕОРИИ ГРАФОВ И НЕКОТОРЫЕ ТЕОРЕМЫ СУЩЕСТВОВАНИЯ И НЕСУЩЕСТВОВАНИЯ

Одной из важных задач, касающихся BIB-схем, является определение всех троек (v, k, λ), для которых существуют (v, k, λ)-BIB-схемы. Но это предельно трудная задача. Один

из ее частных случаев заключается в нахождении всех целых чисел n, являющихся порядками конечной проективной плоскости. По этому предмету существует обширная литература (см. [4], гл. 12). Однако все еще не известно, существует ли целое n, не равное степени простого числа и являющееся порядком конечной проективной плоскости. Для степеней простых чисел это установлено. Существование $(n^2+n+1,\ n+1,\ 1)$-BIB-схемы приводит к существованию проективной плоскости порядка n.

Можно дать различные описания симметричных (v, k, λ)-BIB-схем на языке теории графов. Одно нетривиальное описание было дано только в [6]. Пусть $\Pi = (P, L, I)$ — симметричная BIB-схема (как определено выше). Обозначим через $G(\Pi)$ граф без петель и кратных ребер с множеством вершин $V = \{(p, l) : (p, l) \in I\}$; две вершины (p, l) и (p', l') смежны тогда и только тогда, когда или $p = p'$, или $l = l'$. Матрица смежностей графа с v вершинами — это симметричная матрица $A = \|a_{ij}\|$, в которой элемент a_{ij} равен 1 или 0 в зависимости от того, смежны или нет i-я и j-я вершины. Собственные значения матрицы A называются также собственными значениями графа. Граф G называется связным, если любые две его вершины p и p' соединены цепью — последовательностью вершин и ребер $p_0 = p_1,\ l_1,\ p_2,\ l_2,\ \ldots,\ l_n,\ p_n = p'$, в которой рядом стоящие вершины и ребра инцидентны. В графе G без петель степень вершины равна числу ребер, инцидентных этой вершине. Граф G называется регулярным, если степени всех его вершин равны. Пусть G и G' — графы без кратных ребер. Графы G и G' называются изоморфными (записывается $G \cong G'$), если и только если существует такое отображение φ: $V \to V'$, что вершины v_1 и v_2 смежны тогда и только тогда, когда $\varphi(v_1)$ и $\varphi(v_2)$ смежны. Здесь $v_1,\ v_2 \in V$; V и V' — множества вершин G и G'.

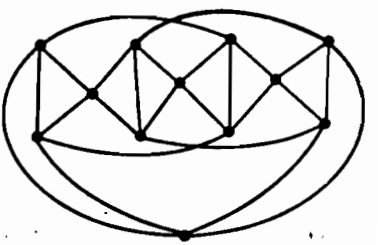


Рис. 1.

Теорема (Гоффман и Рой-Чоудхури [6]). *Пусть v, k и λ — целые числа, Π — симметричная (v, k, λ)-BIB-схема. Тогда*

$$G(\Pi) \text{ — регулярный связный граф с } v, k \text{ вершинами и собственными числами: } -2,\ 2k-2,\ k-2 \pm \sqrt{k-\lambda}. \quad (1)$$

Обратно, если G — граф, обладающий свойством 1, *то $G \cong G(\Pi)$ для некоторой симметричной (v, k, λ)-BIB-схемы или $(v, k, \lambda) = (4, 3, 2)$ и граф G есть граф с* 12 *вершинами* (см. рис. 1).

Следствие. *Симметричная* (v, k, λ)-BIB-*схема существует тогда и только тогда, когда существует регулярный связный граф* G *с* vk *вершинами и собственными числами*: -2, $2k-2$, $k-2 \pm \sqrt{k-\lambda}$.

Легко доказать, что если существует (v, k, λ)-BIB-схема, то ее параметры должны удовлетворять двум условиям:

$$\left.\begin{aligned} \lambda(v-1) &\equiv 0 \pmod{(k-1)}, \\ \lambda v(v-1) &\equiv 0 \pmod{k(k-1)} \end{aligned}\right\} \tag{2}$$

и

$$r \geqslant k. \tag{3}$$

Известно, что эти условия не являются достаточными. Однако существует предположение, что условия (2), (3) «асимптотически достаточны» для существования (v, k, λ)-BIB-схемы, т. е. для фиксированных k и λ существует такое целое $c(k, \lambda)$, что если $v \geqslant c(k, \lambda)$ и v удовлетворяет (2), то (v, k, λ)-BIB-схема существует.

Некоторые результаты для многих частных случаев этого предположения были получены Р. Вильсоном в его докторской диссертации [14]. Пусть K — подмножество положительных целых чисел, $B(K) = \{v$: $(v, K, 1)$-PB-схема существует$\}$ и $\beta(K)$ — единственное положительное целое, которое порождает идеал, порожденный также множеством $\{k(k-1)$: $k \in K\}$, $\alpha(K)$ — единственное положительное целое, которое порождает идеал, порожденный также множеством $\{(k-1)$: $k \in K\}$. Отображение $K \to B(K)$ подмножеств положительных целых чисел в подмножество положительных целых чисел *есть операция замыкания*, т. е. удовлетворяет аксиомам

(i) $$B(K) \supseteq K,$$

(ii) $$K_1 \supseteq K_2 \Rightarrow B(K_1) \supseteq B(K_2),$$

(iii) $$B(B(K)) = B(K).$$

Множество K замкнуто, если и только если $K = B(K)$. Гипотеза существования BIB-схемы связана с проблемой определения замкнутых множеств. Слоем замкнутого множества является класс вычетов по $\bmod \beta(K)$. Слой d называется полным, если и только если существует такая константа M, что

$$\{v\colon \geqslant M,\ v \equiv d \pmod{\beta(K)}\} \subseteq K.$$

Замкнутое множество K называется в конечном счете **периодическим** с периодом $\beta(K)$, если и только если все его **слои** полные.

Теорема (Вильсон [14]). *Все замкнутые множества K являются в конечном счете периодическими с периодом $B(K)$.*

Следствие 1. *Если k — степень простого числа, то необходимое условие* (2) *является асимптотически достаточным.*

Следствие 2. *Если (k, λ) равно* 1 *или степени простого числа, то необходимое условие* (2) *является асимптотически достаточным.*

Если симметричная (v, k, λ) BIB-схема существует, то в дополнение к условиям (2) и (3) тройка (v, k, λ) должна удовлетворять следующему условию:

$$\left.\begin{array}{l}\text{(a) — если } v \text{ четно, то } (k-\lambda) \text{ есть квадрат некоторого целого числа,}\\ \text{(b) — если } v \text{ нечетно, то диофантово уравнение } x^2 = (k-\lambda)y^2 + (-1)^{\frac{v-1}{2}}\lambda z^2 \text{ имеет нетривиальное решение.}\end{array}\right\} \quad (4)$$

Это необходимое условие исключает из рассмотрения многие бесконечные семейства троек (v, k, λ), удовлетворяющих (2) и (3). Целое число c, такое, что p^c делит n, но p^{c+1} не делит n, называется экспонентом p в n. Одним из следствий условия (4) является то, что если экспонент p в n — нечетное простое число и $p \equiv 3 (\mathrm{mod}\, 4)$, то $(n^2+n+1,\ n+1,\ 1)$-BIB-схема не существует. Для $\lambda = 1$ условие (4) было получено сначала Бруком и Райзером, а затем обобщено Шрикхандом и Шюценбергером (см. Райзер [11], гл. 8). Несуществование асимметричных (v, k, λ)-BIB-схем для троек (v, k, λ), удовлетворяющих (2) и (3), в ряде случаев доказывается с помощью теоремы Холла — Коннора о вложении [11, стр. 128].

Пусть $(P, \mathfrak{A})$ — симметричная (v, k, λ)-BIB-схема и $A_{l_0} \in \mathfrak{A}$. Обозначим $P^* = P - A_{l_0}$, $A_l^* = A_l - A_l \cap A_{l_0}$, $L^* = L - \{l_0\}$ и $\mathfrak{A}^* = (A_l^*, l \in L^*)$. Здесь $(P^*, \mathfrak{A}^*)$ есть (v^*, k^*, λ^*)-BIB-схема с параметрами $v^* = v - k$, $k^* = k - \lambda$ и $\lambda^* = \lambda$. Для $(P, \mathfrak{A})$ BIB-схему $(P^*, \mathfrak{A}^*)$ назовем остаточной BIB-схемой.

Теорема (Холл — Коннор). *Пусть* $v = \frac{k(k-1)}{2} + 1$, $v^* = v - k$, $k^* = k - 2$. *Тогда каждая* $(v^*, k^*, 2)$-BIB-*схема есть остаточная схема симметричной* $(v, k, 2)$-BIB-*схемы.*

Следствие. *Если существует* $(v^*, k^*, 2)$-BIB-*схема для* $v = \frac{k(k+1)}{2} + 1$, $v^* = v - k$, $k^* = k - 2$, *то тройка чисел* $(v, k, 2)$ *удовлетворяет условию* (4).

Из этого следствия вытекает, что не существует (15, 5, 2)-BIB-схемы, так как иначе существовала бы симметричная

(22, 7, 2)-BIB-схема и по условию (4) она должна быть совершенным квадратом ($k - \lambda = 7 - 2 = 5$).

Теорему Холла — Коннора о вложении можно получить из одной теоремы теории графов. Пусть G — неориентированный граф без петель и кратных ребер. Для G построим реберный граф $L(G)$, в котором вершинами являются ребра G. Две вершины $L(G)$ смежны, если и только если соответствующие ребра G имеют общую вершину. Граф G называется полным, если в нем любая пара вершин соединена ребром.

Теорема (Чанг [3], Гоффман [5], Шрикханд [13]). *Пусть G_n — полный граф с n вершинами. Тогда*:

(1) *$L(G_n)$ — регулярный связный граф с $\binom{n}{2}$ вершинами и различными собственными числами*: -2, $2n-4$ и $n-4$.

Обратно:

(2) *если $n \neq 8$ и H — регулярный связный граф с $\binom{n}{2}$ вершинами и различными собственными числами*: -2, $2n-4$, $n-4$, *то H изоморфен $L(G_n)$.*

Теорему Холла — Коннора о вложении можно получить именно из теоремы о $L(G_n)$. Для данной $(v^*, k^*, 2)$-BIB-схемы с параметрами $v = \frac{k(k-1)}{2} + 1$, $v^* = v - k$, $k^* = k - 2$ определим граф H, вершинами которого являются блоки схемы и две вершины которого смежны, если и только если два блока имеют точно один общий элемент. Граф H удовлетворяет условиям последней теоремы.

6. РАЗРЕШИМЫЕ СБАЛАНСИРОВАННЫЕ НЕПОЛНЫЕ БЛОК-СХЕМЫ

Пусть $(P, \mathfrak{A})$ является (v, k, λ)-BIB-схемой. Подкласс $\mathfrak{A}_1 \subseteq \mathfrak{A}$ называется параллельным классом блоков, если каждый элемент появляется точно в одном блоке $\mathfrak{A}_1$. (v, k, λ)-BIB-схема считается разрешимой, если и только если семейство $\mathfrak{A}$ блоков может быть разделено на r параллельных классов $\mathfrak{A}_i$, $i = 1, 2, \ldots, r$. Следующая схема является разрешимой (9, 3, 1)-BIB-схемой с $P = \{1, 2, 3, \ldots, 9\}$ и 4 параллельными классами $\mathfrak{A}_i$, $i = 1, 2, 3, 4$, каждый из которых состоит из трех блоков:

$\mathfrak{A}_1$	$\mathfrak{A}_2$	$\mathfrak{A}_3$	$\mathfrak{A}_4$
(1, 2, 3)	(1, 4, 7)	(1, 5, 9)	(3, 5, 7)
(4, 5, 6)	(2, 5, 8)	(2, 6, 7)	(2, 4, 9)
(7, 8, 9)	(3, 6, 9)	(3, 4, 8)	(1, 6, 8)

Разрешимая (v, k, λ)-BIB-схема может быть использована для построения эффективных экспериментальных планов в тех

случаях, когда некоторые внешние факторы, воздействующие на опыты, имеют иерархическую структуру. Например, разрешимую (9, 3, 1)-BIB-схему можно использовать для построения плана эксперимента, предназначенного для сравнения девяти различных видов мороженого. 12 дегустаторов разделяются на 4 возрастные группы, каждая из которых содержит по три дегустатора. Каждый дегустатор «испытывает» последовательно три различных вида мороженого и дает им оценку. Предполагается, что оценка зависит от качества мороженого (эффект мороженого), от возрастной группы дегустатора (эффект возрастной группы) и, наконец, от индивидуальных свойств дегустатора (эффект дегустатора внутри возрастной группы). Здесь внешний фактор «возрастная группа» имеет иерархическую структуру. При построении разрешимой BIB-схемы четыре возрастные группы соответствуют четырем параллельным классам. Внутри каждой возрастной группы три дегустатора соответствуют трем блокам, и каждый дегустатор испытывает три вида мороженого данного блока. Здесь каждому дегустатору ставится в соответствие группа из трех опытов. Если разрешимая (v, k, λ)-BIB-схема существует, то в дополнение к требованиям (2) и (3) должно выполняться третье условие:

$$v \equiv 0 \pmod{k}. \tag{5}$$

Из (2) и (5) следует, что для существования разрешимой $(v, k, 1)$-BIB-схемы требуется выполнение условия

$$v \equiv k \pmod{k(k-1)}. \tag{6}$$

Разрешимая $(v, 3, 1)$-BIB-схема эквивалентна комбинаторной схеме, которая получается при решении задачи о школьниках, предложенной Киркманом в 1847 году. Задача формулировалась следующим образом. В пансионе содержится 15 школьниц, учительница выводит их каждый день недели на прогулку. Школьницы разбиваются на группы по три и прогуливаются по пяти дорожкам. Задача заключается в нахождении различных упорядочений дорожек для семи дней недели так, чтобы гарантировать каждой паре школьниц прогуливаться по одной и той же дорожке точно один день в неделю. В общем случае v девушек каждый день распределяются тройками по $v/3$ дорожкам. Задача заключается в нахождении различных распределений для $\frac{v-1}{2}$ последовательных дней, таких, что любые две девушки прогуливаются по одной и той же дорожке точно один день из $\frac{v-1}{2}$ дней. Распределение Киркмана для v девушек эквивалентно построению разрешимой $(v, 3, 1)$-BIB-схемы: v девушек соответствуют v элементам, дорожки — блокам

и $\frac{v-1}{2}$ дней соответствуют $\frac{v-1}{2}$ параллельным классам. Следовательно, необходимым условием для существования схемы Киркмана для v девушек будет

$$v \equiv 3 \pmod 6. \tag{7}$$

В конце XIX и в начале XX столетия над этой задачей работали многие математики. Было доказано существование распределений Киркмана для многих бесконечных множеств чисел вида $6t+3$. Однако оставалось неизвестным, является ли необходимое условие (7) достаточным. В 1968 г. Рой-Чоудхури и Вильсон [9] доказали достаточность условия (7).

Теорема (Рой-Чоудхури и Вильсон [9]). *Разрешимая* $(v, 3, 1)$-BIB-*схема существует, если и только если* $v \equiv 3 \pmod 6$.

Полную библиографию, относящуюся к задаче Киркмана о школьницах, можно найти в [10]. Необходимое условие (6) для существования разрешимой $(v, k, 1)$-BIB-схемы, как известно, не является достаточным. Можно привести пример (36, 6, 1)-схемы, удовлетворяющей (6), но из теоремы Брука — Райзера следует, что разрешимой (36, 6, 1)-BIB-схемы не существует. Тем не менее Рой-Чоудхури и Вильсон [10] недавно доказали, что условие (6) является асимптотически достаточным.

Теорема (Рой-Чоудхури и Вильсон [10]). *Для каждого целого числа* k *существует константа* $c(k)$, *такая, что если* $v \geqslant c(k)$ *и* $v \equiv k \pmod{k(k-1)}$, *то существует разрешимая* $(v, k, 1)$-BIB-*схема.*

Можно дать интерпретацию на языке теории графов разрешимой $(n^2, n, 1)$-BIB-схемы. Пусть $\Pi = (P, L, I)$ — разрешимая $(n^2, n, 1)$-BIB-схема. Определим $G(\Pi)$ как было сделано выше.

Теорема (Гоффман и Рой-Чоудхури [7]). H *— является регулярным связным графом с* $n^2(n+1)$ *вершинами и с различными собственными значениями:* -2, $2n-1$, $n-2$, $\frac{1}{2}((2n-3) \pm \sqrt{(4n+1)})$, *тогда и только тогда, когда* $H \cong G(\Pi)$, *где* Π *— разрешимая* $(n^2, n, 1)$-BIB-*схема.*

Следствие. *Разрешимая* $(n^2, n, 1)$-BIB-*схема существует, если и только если существует граф* H, *обладающий сформулированными выше свойствами.*

7. МНОЖЕСТВО ОРТОГОНАЛЬНЫХ СХЕМ

(m, n)-ортогональное множество есть отображение A: $S_1 \times S_2 \to S_3$, где S_1 — множество m «строк», S_2 — множество n^2 «столбцов» и S_3 — множество n символов, таких, что для a,

$b \in S_1$, $a \neq b$ и c, $d \in S_2$ существует точно один столбец x, такой, что $A(a, x) = c$ и $A(b, x) = d$. Ниже приведено $(4, 5)$-ортогональное множество.

1 1 1 1 1	2 2 2 2 2	3 3 3 3 3	4 4 4 4 4	5 5 5 5 5
1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5
2 3 4 5 1	3 4 5 1 2	4 5 1 2 3	5 1 2 3 4	1 2 3 4 5
3 4 5 1 2	5 1 2 3 4	2 3 4 5 1	4 5 1 2 3	1 2 3 4 5

(m, n)-ортогональное множество можно использовать для построения эффективного экспериментального плана, если имеется $(m - 1)$ внешних факторов, каждый из которых содержит n уровней и n различных элементов. Имеется n^2 опытов, каждый из которых соответствует одному из n^2 столбцов. Первые $(m - 1)$ строк соответствуют уровням внешних факторов, а последняя строка — элементам, используемым в различных опытах. Такой план назовем (m, n)-ОА-планом эксперимента (сокращение от orthogonal array). $(3, n)$-ОА-план эксперимента есть не что иное, как план на основе латинского квадрата, а $(4, m)$-ОА-план эксперимента — на основе греко-латинского квадрата. Кифер доказал, что в предположении аддитивности эффекта элементов и эффекта внешних факторов $(3, n)$-ОА-план имеет некоторые оптимальные свойства. В общем план эксперимента, построенный на основе (m, n)-ортогонального множества, обладает теми же оптимальными свойствами. В качестве примера рассмотрим эксперимент, поставленный для определения влияния музыки на производительность труда на фабрике. Музыкальные произведения разбивались на пять групп. Эти группы являются элементами. Было установлено, что производительность труда на фабрике зависит также от индивидуальных способностей рабочего, от дня недели и от недели месяца. Рабочие, дни и недели рассматривались как внешние факторы. В данной задаче можно воспользоваться $(4, 5)$-ОА-планом эксперимента. Четыре строки этого плана соответственно представляют рабочих, дни, недели и музыкальные произведения. Например, музыкальное произведение третьей группы исполняется для второго рабочего в первый день первой недели, если пользоваться приведенным выше $(4, 5)$-ОА-планом.

(m, n)-ОА-план представляет собой дробный факторный эксперимент. Полный факторный эксперимент требует n^m опытов, в то время как ОА-план эксперимента — только n^2 опытов. Поэтому желательно знать наибольшее целое число m, для которого существует (m, n)-ОА-план. Необходимо заметить, что (m, n)-ОА-план существует, если и только если существует множество $(m - 2)$ взаимно ортогональных латинских квадратов

порядка n. Легко показать, что если существует (m, n)-ОА-план, то $m \leqslant n+1$. Обратно, Брук доказал следующее.

Теорема (Брук [2]). *Если существует* (m, n)-ОА-*план, где* $n \geqslant \frac{1}{2}(d-1)(d^3-d^2+d+2)$, $d=n+1-m$, *то существует и* $(n+1, n)$-ОА-*план.*

Теорема Брука была обобщена Боузом [1] в его теореме, которую лучше определить как теорему о графах. Пусть G — граф без петель и кратных ребер. Две различные вершины называются первично связанными друг с другом, если и только если они являются смежными.

Граф G называется сильно регулярным с параметрами $(v, n_1, p_{11}^1, p_{11}^2)$, если и только если:

(i) число его вершин равно v;

(ii) каждая вершина имеет степень n_1;

(iii) число общих вершин, первично связанных с двумя смежными вершинами, равно p_{11}^1;

(iv) число общих вершин, первично связанных с двумя несмежными вершинами, равно p_{11}^2.

Говорят, что $\Pi=(P, L, I)$ является (r, k, t)-частичной геометрией, если и только если:

(i) каждая точка инцидентна r линиям;

(ii) каждая линия инцидентна k точкам;

(iii) две различные точки p_1 и p_2 инцидентны самое большее одной общей линии;

(iv) для каждой неинцидентной пары точек (p, l) имеется t линий, инцидентных p и пересекающих l.

Пусть $G(\Pi)$ обозначает граф, у которого множество вершин P и две вершины p_1 и p_2 смежны, если и только если имеется линия l в Π, инцидентная p_1 и p_2. Пусть также $1 \leqslant t \leqslant r, k$. Боуз доказал следующее:

Теорема (Боуз [1]). 1) *Если* Π *является* (r, k, t)-*частичной геометрией, то* $G(\Pi)$ *есть* $(v, n_1, p_{11}^1, p_{11}^2)$ — *сильно регулярный граф, в котором* $n_1=r(k-1)$, $v-1-n_1=\frac{1}{t}(r-1)(k-1)(k-t)$, $p_{11}^1=(r-1)(k-1)+(k-2)$, $p_{11}^2=rt$. 2) *Обратно, если* G — *сильно регулярный граф с параметрами, как в* 1), *и* $k \geqslant \frac{1}{2}(r(r-1)+t(r+1)(r^2-2r+2))$, *то* G *изоморфен* $G(\Pi)$ *для* (r, k, t)-*частичной геометрии* Π.

Теорема Брука непосредственно следует из теоремы Боуза, так как (m, n)-ОА-план существует, если и только если существует $(m, n, m-1)$-частичная геометрия.

СПИСОК ЛИТЕРАТУРЫ

1. Bose R. C., Strongly Regular Graphs, Partial Geometries and Partially Balanced Designs, *Pacific J. Math.*, **13** (1963), 389—419.
2. Bruck R. H., Finite Nets II, Uniqueness and Embedding, *Pacific J. Math.*, **13** (1963), 421—457.
3. Chang L. C., The Uniqueness and Nonuniqueness of the Triangular Association Schemes, *Science Record*, **3** (1959), 604—613.
4. Hall M., Jr., Combinatorial Theory, Blaisdell Publishing Co., Waltham, Mass., 1967. (Русский перевод: Холл М., Комбинаторика, «Мир», М., 1970.)
5. Hoffman A. J., On the Uniqueness of triangular Association Schemes, *Ann. Math. Statist.*, **29** (1958), 262—266.
6. Hoffman A. J., Ray-Chaudhuri D. K., On the Line Graph of Symmetric Balanced Incomplete Block Designs, *Trans. Amer. Math. Soc.*, **116** (1965), 238—252.
7. Hoffman A. J., Ray-Chaudhuri D. K., On the Line Graph of a Finite Affine Plane, *Can. J. Math.*, **17** (1965), 687—694.
8. Kiefer J., On the Non-Randomized Optimality and Randomized Optimality of Symmetrical Designs, *Ann. Math. Statist.*, **29** (1958), 675—699.
9. Ray-Chaudhuri D. K., Wilson Richard M., Solution of Kirkman's School Girl Problem, Proceedings of the Symposia in Pure Mathematics, Combinatorics (19) Amer. Math. Soc.
10. Ray-Chaudhuri D. K., Wilson Richard M., On the Existence of Resolvable Balanced Incomplete Block Designs, Proceedings of the Calgary International Conference on Combinatorial Structures and their Applications, Gordon and Breach, New York, 1970.
11. Ryser H. J., Combinatorial Mathematics, Carus Math., Monograph 14, Math. Assoc. Amer., 1963. (Русский перевод: Райзер Дж., Комбинаторная математика, «Мир», М., 1966.)
12. Smith C. A. B., Hartley H. O., The Construction of Yonden Squares, *J. Roy. Statist. Soc.*, Ser. B, **10** (1948), 262—263.
13. Shrikhande S. S., On a Characterization of the Triangular Association Schemes, *Ann. Math. Statist.*, **30** (1959), 39—47.
14. Wilson Richard M., An Existence Theory for Pairwise Balanced Designs, Ph. D. Dissertation, Dept. of Mathematics., The Ohio State University, 1969.

Теорема о плоских графах[1])

У. Т. Татт

1. ВВЕДЕНИЕ

В этой статье будет обобщена хорошо известная теорема Хасслера Уитни о гамильтоновых циклах триангуляций [2]. Используемые методы по существу те же, что и в [2], но здесь они детально переработаны так, чтобы их можно было применять ко всем плоским графам. В обобщенной теореме (теорема 1, см. ниже) утверждается, что любой плоский граф, имеющий цикл, обладает циклом, удовлетворяющим некоторым особым условиям. Для важного класса плоских графов, включающего и графы, которые изучались Хасслером Уитни, в этих условиях требуется, чтобы цикл был гамильтоновым.

В заключение этой вводной части мы дадим формальные определения некоторых терминов, которые будут использоваться в данной статье, и затем дадим точную формулировку основной теоремы.

Плоский граф — конечное множество простых топологически замкнутых линий, называемых *ребрами,* на 2-сфере, таких, что любая точка пересечения двух различных элементов этого множества является концом каждого из этих элементов. *Вершины* плоского графа есть концы его ребер. Ясно, что всякое подмножество плоского графа есть плоский граф.

В соответствии с этим определением будем говорить, что два плоских графа *пересекаются*, если у них есть общее ребро, и *не пересекаются*, если у них нет общих ребер. Два непересекающихся плоских графа могут иметь одну или более общих вершин.

Пусть G — произвольный плоский граф. Обозначим число его вершин через $\alpha_0(G)$, а число его ребер — через $\alpha_1(G)$. Ребро A и вершина a графа G *инцидентны*, если a является концом A. *Степень* $d(a)$ вершины a — это число ребер G, инцидентных a.

Комплексом $|G|$ графа G называется объединение его ребер. Плоский граф G является *циклом*, если его комплекс — простая замкнутая линия, и *цепью*, если его комплекс — про-

---

[1]) Tutte W. T., A theorem on planar graphs, Transactions of the American Mathematical Society, v. 82, № 1 (May, 1956), 99—116.

стая незамкнутая линия. В последнем случае мы будем называть концы этой простой линии *концами* цепи, а другие вершины цепи — *внутренними* вершинами. Под циклом или цепью графа G понимается его подмножество, которое соответственно есть цикл или цепь. Цикл графа G называется *гамильтоновым*, если он содержит каждую вершину этого графа. Ребро, не входящее ни в один из циклов, называется *перешейком* графа G.

Вершинами сочленения подмножества ребер H графа G называются общие вершины графов H и $G-H$. Обозначим число этих вершин через $w(H)$, назовем его *числом сочленения* H. Ясно, что $w(G-H)=w(H)$.

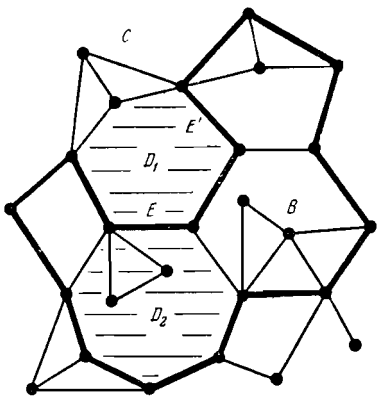


Рис. 1.

Пусть J — цикл графа G. Подмножество H графа $G-J$ *J-ограничено*, если его вершины сочленения есть вершины J. Если $G-J$ не пусто, то оно единственным образом представляется как объединение непересекающихся минимальных непустых J-ограниченных подмножеств графа $G-J$. Эти подмножества называются *мостами* для J' в G.

Комплекс $|J|$ цикла J графа G есть простая замкнутая линия. Области, выделяемые этой кривой на 2-сфере, будем так и называть — *выделенными* областями, а J будем называть циклом, *ограничивающим* каждую из них.

Пусть E — ребро графа G, не являющееся перешейком, и D — область, выделенная циклом J графа G так, что $E\in J$. Тогда D называется *граничной областью* ребра E, если в G нет цикла, содержащего E и выделяющего область, которая является собственным подмножеством D. Если D — граничная область ребра E, то J называется его *граничным циклом*. Ясно, что ребро E имеет точно две граничные области. Соответствующие граничные циклы могут как совпадать, так и не совпадать.

Основной результат настоящей работы следующий:

Теорема 1. *Пусть G — произвольный плоский граф, а E — ребро G, не являющееся перешейком. Пусть E' — ребро, отличное от E и принадлежащее граничному циклу ребра E.*

Тогда существует цикл J графа G, имеющий следующие свойства:

(i) $(E, E')\in J$;

(ii) *если B — мост для цикла J в графе G, то* $w(B)\leqslant 3$;

(iii) *если B — мост для J в графе G, который пересекается с граничным циклом ребра E, то $w(B) = 2$.*

Эта теорема проиллюстрирована на рис. 1. Здесь граничными областями ребра E являются заштрихованные области D_1 и D_2. Жирной линией на рисунке выделен цикл J, удовлетворяющий условиям (i), (ii) и (iii). Буквой B обозначен мост для J, имеющий число сочленения, равное 3, и не пересекающийся с граничным циклом ребра E. Но C является мостом для J, который пересекается с граничным циклом ребра E и имеет соответственно число сочленения, равное 2.

2. МОСТЫ

Пусть J — цикл плоского графа G. Из определения J-ограниченного подмножества графа $G - J$ сразу получаются следующие предложения: а) множество $G - J$ J-ограничено, б) дополнение в $G - J$ к любому J-ограниченному подмножеству графа $G - J$ также J-ограничено и в) любое пересечение J-ограниченных подмножеств графа $G - J$ J-ограничено.

Определим *мост* для цикла J в графе G как минимальное непустое J-ограниченное подмножество графа $G - J$. Если $A \in G - J$, то пересечение всех J-ограниченных подмножеств графа $G - J$, содержащих A, является мостом для J в графе G, содержащим A, в чем легко убедиться при помощи упомянутых выше предложений о J-ограниченных множествах. Поэтому если $G - J$ не пусто, то оно является объединением мостов для J в графе G. Более того, два различных моста для J в G не пересекаются, поскольку их пересечение J-ограничено. Следовательно, мосты для J имеют свойства, определенные во введении.

(2.1) *Пусть x и y — различные вершины моста B для J в графе G. Тогда в B существует цепь, концы которой есть x и y и которая не имеет внутренних вершин, общих с J.*

Доказательство. Пусть H — объединение всех простых цепей в B, у которых одним из концов является x и которые не имеют внутренних вершин, общих с J. Тогда H не пусто, так как x инцидентна по крайней мере с одним ребром из B. Ясно, что x может быть вершиной сочленения для H только тогда, когда она является вершиной из J.

Допустим, что z — произвольная вершина сочленения для H, отличная от x. Пусть C — некоторое ребро из $B - H$, инцидентное с z, и пусть t является другим концом C. Тогда должна существовать цепь L в H с концами x и z, которая не имеет внутренних вершин, общих с J. Далее, t не совпадает с x, так

как цепь $\{C\}$ не содержится в H. Более того, t не является внутренней вершиной для L. Иначе, если M есть множество всех ребер из L между x и t, то $M \cup \{C\}$ будет цепью в H, что противоречит определению C. Следовательно, $L \cup \{C\}$ есть цепь. Из того, что она не является подмножеством в H, следует, что z есть вершина в J.

Мы получили, что H J-ограничено. Так как оно не пусто, то оно совпадает с B. Следовательно, y есть вершина в H. Теорема доказана.

(2.2) *Пусть L — цепь в $G - J$, не имеющая внутренних вершин, общих с J. Тогда L будет подмножеством некоторого моста для J в графе G.*

Доказательство. Занумеруем ребра из L в порядке их следования на линии $|L|$: $A(1), \ldots, A(k)$. Пусть для $1 \leqslant i \leqslant k$ $B(i)$ является мостом для J, содержащим $A(i)$. Если $1 \leqslant i < k$, то $B(i)$ и $B(i+1)$ совпадают, так как они имеют общую вершину, не являющуюся вершиной из J. Теорема доказана.

(2.3) *Если B — мост для J в графе G, то $|B|$ пересекается в точности с одной из областей, выделенной циклом J.*

Доказательство. Пусть D и D' — области, выделенные циклом J. Предположим, что $|B|$ пересекается с каждой из них. Так как $|B|$ пересекается с D, то некоторое ребро A из B пересекается с D. Это ребро должно лежать целиком в D без, быть может, одного из своих концов. Если же оба конца ребра A являются вершинами из J, то $\{A\}$ J-ограничено, $B = \{A\}$ и $|B| \cap D'$ пусто. Мы заключаем, что B имеет вершину x в D. Аналогично оно имеет вершину x' в D'. Любая простая линия в $|B|$ с концами x и x' должна пересекать $|J|$. Но это противоречит (2.1). Теорема доказана.

Если B мост для J и $|B|$ пересекается с областью D, выделенной циклом J, то мы назовем B мостом *через* D.

(2.4) *Пусть J' — цикл в G, выделяющий область D', которая является подмножеством области D, выделенной циклом J. Пусть B — мост для J через D, такой, что некоторое ребро A из B пересекается с D'. Тогда существует мост B' для J' через D', такой, что $A \in B' \subseteq B$.*

Доказательство. Пусть X — множество всех ребер из B, пересекающихся с D'. Так как D' не пересекается с $|J|$, то вершины сочленения для X все будут точками из $|J'|$, поэтому X J'-ограничено. Пусть B' мост для J', содержащий A. Тогда B' — мост через D'. Более того, $B' \cap X$ не пусто. J'-ограничено и, значит, совпадает с B'. Следовательно, $A \in B' \subseteq B$.

Следствие. *Если, кроме того, все вершины, общие для B и J', являются вершинами из J, то само B будет мостом для J' через D'.*

В самом деле, тогда B' J-ограничено и поэтому совпадает с B.

3. МУЛЬТИСВЯЗНОСТЬ

Плоский граф G *n-сепарабелен*, где n — неотрицательное целое число, если его можно разделить на два непересекающихся подмножества H и $G-H$, причем каждое из них имеет вершину, которая не является вершиной другого, так что $w(H)=w(G-H)\leqslant n$. Граф G *n-связен*, если он не является m-сепарабельным для любого неотрицательного целого числа $m<n$. Таким образом, если $j>k\geqslant 0$ и G j-связен, то G также и k-связен. В n-сепарабельном плоском графе найдутся две вершины, разделенные не более чем n другими вершинами, и n-связный граф не имеет никаких других вершин, разделенных менее чем n другими.

(3.1) *Всякий цикл 2-связен.*

Это следует из того, что простая замкнутая линия связна, и ее связность не нарушается при удалении любой одной точки.

(3.2) *Пусть J — цикл в плоском графе G, и пусть B — мост для J в G. Предположим, что существует цикл K в графе G, пересекающийся как с J, так и с B. Тогда $w(B)\geqslant 2$.*

Доказательство. $B\cap K$ и $(G-B)\cap K$ — непустые, дополняющие друг друга подмножества цикла K. Так как K 2-связен, то они имеют по крайней мере две общие вершины. Но они будут вершинами сочленения для B, следовательно, $w(B)\geqslant 2$.

(3.3) *2-связный плоский граф G, для которого $\alpha_1(G)\geqslant 2$, не имеет перешейков.*

Доказательство. Предположим, что A, концы которого a и b, есть перешеек в G. Так как G не 0-сепарабелен, то можно предположить, что $d(a)\geqslant 2$. Пусть H — объединение всех цепей в G, которые имеют концом a и не содержат A. Тогда H не пусто, так как $d(a)\geqslant 2$. Ясно, что H и $G-H$ не имеют общих вершин, кроме a и b. Следовательно, так как G не 1-сепарабелен, обе эти вершины являются общими для них. Отсюда следует, что существует цепь L в H с концами a и b. Тогда $L\cup\{A\}$ будет циклом в G, но это противоречит определению A.

(3.4) *Если G является n-связным плоским графом, для которого $\alpha_0(G)>n$, то $d(a)\geqslant n$ для каждой вершины a из G.*

Доказательство. Предположим, что G имеет вершину a, такую, что $d(a) < n$. Пусть H — множество всех ребер из G, инцидентных с a. Тогда a — вершина из H, но не из $G - H$. С другой стороны, $\alpha_0(H) \leqslant d(a) + 1$ и, таким образом, $G - H$ имеет вершину, не принадлежащую H. Но $w(H) = w(G - H) \leqslant d(a) < n$, что противоречит предположению о том, что G n-связен.

(3.5) *Пусть B — мост для цикла J в n-связном плоском графе G. Тогда если $w(B) < n$, то верно одно из следующих утверждений*:

(i) *B имеет в точности одно ребро, оба конца которого являются вершинами из J*;

(ii) *$\alpha_1(J) = w(B)$ и B является единственным мостом для J в G, имеющим более одного ребра.*

Доказательство. Может случиться, что оба конца некоторого ребра A из B являются вершинами из J. Тогда $\{A\}$ J-ограничено и потому $B = \{A\}$. В этом случае верно утверждение (i).

В ином случае B имеет вершину, не являющуюся вершиной сочленения. Так как $w(B) < n$, то граф G не есть $w(B)$-сепарабельный и, таким образом, вершинами в $G - B$ будут лишь вершины сочленения для B. Отсюда следует, что J имеет ровно $w(B)$ вершин и, стало быть, ровно $w(B)$ ребер. Следовательно, $n > w(B) \geqslant 2$, и в силу (3.4) B имеет по крайней мере два ребра. Никакой мост для J, кроме B, не может иметь более одного ребра, ибо такой мост имел бы вершину, которая не была бы вершиной сочленения для B.

4. ГРАНИЧНЫЕ ОБЛАСТИ

Пусть G — некоторый непустой плоский граф.

Цепь L в G называется *замыкаемой*, если существует цикл J в G, такой, что $L \subset J$. Таким образом, если $E \in G$, то цепь $\{E\}$ будет замыкаемой тогда и только тогда, когда E не является перешейком в G.

Пусть L — замыкаемая цепь в G, и J — такой цикл в G, что $L \subset J$. Пусть D и D' — области, выделенные циклом J. Мы назовем D *граничной областью* цепи L в G, если в G нет цикла K, содержащего L, и такого, что некоторая область, выделенная циклом K, является собственным подмножеством в D. Если или D, или D' является граничной областью цепи L в G, то мы назовем J *граничным циклом* цепи L в G. Таким образом, если E — ребро в G, не являющееся перешейком, то граничные области и циклы цепи $\{E\}$ совпадают соответственно с граничными областями и циклами ребра E, которые были определены в введении.

Если H — подмножество в G и K — подмножество в $G - H$, мы обозначим через $p(H, K)$ число вершин H, которые являются вершинами K, и через $q(H, K)$ — число вершин сочленения H, которые не являются вершинами K. Таким образом, $p(H, K) + q(H, K) = w(H)$.

Если Q — некоторый цикл в G, мы определим *трансверсаль* цикла Q как такую цепь M в $G - Q$, у которой только те вершины, которые являются концами этой цепи, являются вершинами Q. Если M удовлетворяет этому условию, то в силу (2.2) она будет подмножеством некоторого моста B для Q. Если B — мост через область D, выделенную циклом Q, мы назовем M трансверсалью цикла Q *через* D. Тогда из (2.3) следует, что $|M|$ есть простая линия, концы которой лежат в $|Q|$, но которая в остальном целиком лежит в D, так что $|M|$ является *разрезающим путем* в D.

(4.1) *Пусть L — замыкаемая цепь в G, D — граничная область цепи L и J — соответствующий граничный цикл цепи L. Пусть B — произвольный мост для J через D. Тогда $p(B, J - L) \leqslant 1$.*

Доказательство. Допустим, что $p(B, J - L) > 1$. Тогда найдутся две различные вершины x и y, общие для B и J, которые не являются внутренними для L. В силу (2.1) существует трансверсаль M цикла J через D с концами x и y. Но существует цепь N в J с концами x и y, причем $L \subseteq N$. Ясно, что $M \cup N$ есть цикл в G, содержащий L, и одна из выделенных им областей является собственным подмножеством D. Это противоречит определению D.

(4.2) *Замыкаемая цепь L в G имеет ровно две граничные области в G. Более того, если J — произвольный цикл в G, содержащий L и имеющий выделенные области D и D', то одна из граничных областей цепи L будет подмножеством в D, а другая — в D'.*

Доказательство. Существует цикл J в G, содержащий L. Если D и D' — выделенные им области, то, так как G конечен, из определения граничной области следует, что существуют граничные области T и T' цепи L, содержащиеся в D и в D' соответственно. Поскольку T и T' расположены по разные стороны от линии $|L|$, то всякая третья граничная область T'' цепи L должна пересекаться с T или с T'.

Предположим, что T'' пересекается с T. Пусть K и K'' — ограничивающие циклы для T и T'' соответственно. Если K'' пересекается с мостом B для K через T, то непустые плоские графы $K'' \cap B$ и $K'' \cap (G - B)$ имеют по крайней мере две общие вершины ввиду того, что K'' 2-связен в силу (3.1). Так как

$L \subset K''$, то ни одна из них не будет внутренней вершиной для L. Следовательно, $p(B, K - L) \geqslant 2$, что противоречит (4.1).

Мы получили, что $|K''|$ не пересекается с T. Следовательно, T — подмножество области, выделенной циклом K''. Этой выделенной областью должна быть T'', так как T'' пересекает T. Таким образом, $T \subseteq T''$ и, значит, $T'' = T$, так как T'' — граничная область цепи L. Аналогичное рассуждение показывает, что если T'' пересекает T', то $T'' = T'$. Этим завершается доказательство теоремы.

Если G 2-связен, то каждое ребро из G имеет две различные граничные области в силу (3.3) и (4.2) при условии $\alpha_1(G) \geqslant 2$. Граничные области ребер из G не пересекают $|G|$ в силу (4.1). Их можно легко отождествить с компонентами дополнения к $|G|$ на 2-сфере. Эти компоненты часто называют странами или областями на карте, определяемой графом G.

(4.3) *Пусть B — мост через область D, выделенную циклом J в G. Пусть x и y — различные вершины сочленения для B и L — цепь в J с концами x и y. Пусть T — граничная область цепи L, содержащаяся в D. Тогда существует трансверсаль M цикла J через D с концами x и y, такая, что $L \cup M$ — ограничивающий цикл для T.*

Доказательство. В силу (2.1) существует трансверсаль N цикла J через D с концами x и y. Тогда $L \cup N$ есть цикл в G с выделенной областью D', содержащейся в D. В силу (4.2) $T \subseteq D'$. Далее ясно, что ограничивающий цикл для T можно представить как $L \cup M$, где M есть цепь в G с концами x и y. Так как $T \subseteq D'$, то всякая внутренняя вершина для M будет либо внутренней вершиной для N, либо точкой из D'. Она, таким образом, не есть вершина из J. Более того, $|M|$ не пересекается ни с какой областью, выделенной циклом J, кроме D. Следовательно, M есть трансверсаль цикла J через D.

Рассмотрим мост B для цикла Q в G, такой, что $w(B) \geqslant 2$. Пусть вершинами сочленения для B будут $a_1, a_2, \ldots, a_k$. Эти вершины делят Q на $k = w(B)$ непересекающихся цепей $S_1, S_2, \ldots, S_k$. Мы можем так выбрать обозначения, что концами S_i будут a_i и a_{i+1} для каждого i (причем $a_{k+1} = a_1$). Мы назовем цепи S_i *сегментами* цикла Q, определяемыми B.

(4.4) *Пусть B и B' — мосты для цикла J в G через одну и ту же область D, выделенную циклом J. Предположим, что $w(B) \geqslant 2$. Тогда существует сегмент из J, определяемый B, в вершины которого входят все вершины сочленения для B'.*

Доказательство. Можно считать, что $w(B') \geqslant 2$, иначе теорема тривиальна.

а) Допустим сначала, что B' имеет вершину сочленения b, которая есть внутренняя вершина некоторого сегмента S_i цикла J, определяемого B. Предположим, что B' также имеет вершину сочленения c, которая не принадлежит вершинам из S_i. Тогда B' содержит трансверсаль M' цикла J через D с концами b и c в силу (2.1). Аналогично B содержит трансверсаль M цикла J через D, концы которой есть концы a_i и a_{i+1} сегмента S_i. Две линии $|M|$ и $|M'|$ имеют общую вершину в D. Но это невозможно, так как B и B' различны. Значит, в этом случае теорема верна.

б) В оставшемся случае каждая вершина сочленения для B' есть вершина сочленения для B. Тогда можно считать, что $w(B) \geqslant 3$, так как в противном случае теорема тривиальна. Пусть b — вершина сочленения для B', и пусть U и V — сегменты цикла J, определяемые B, для которых b есть один из концов, а a и c — другие концы для U и V соответственно. Тогда a, b и c различны, так как $w(B) \geqslant 3$.

Пусть F — граничная область, содержащаяся в D, для замыкаемой цепи $U \cup V$. В силу (4.3) ее ограничивающий цикл имеет вид $U \cup V \cup M$, где M есть трансверсаль цикла J через D с концами a и c. Разрезающий путь $|M|$ делит D на две простые связные области F и F', ограничивающие циклы которых есть $U \cup V \cup M$ и $(J-(U \cup V)) \cup M$ соответственно. Как $|B|$, так и $|B'|$ пересекаются с F, так как b есть граничная точка для F, но не для F'.

В силу (2.2) M является подмножеством некоторого моста C для J через D. Мы имеем $C = B$. В противном случае, согласно следствию из (2.4), B есть мост через F, что противоречит (4.1). Но далее из (4.1) и следствия из (2.4) следует, что B' — мост через F, не содержащий одновременно a и c в качестве вершин сочленения. Поэтому, по определению B', его вершины сочленения будут либо концами U, либо концами V.

(4.5) *Пусть L есть замыкаемая цепь в G с концами a и b, и пусть E — ребро в L, инцидентное с a. Допустим, что D — граничная область цепи L и D' — граничная область ребра E, содержащаяся в D. Пусть J и J' — ограничивающие циклы для D и D' соответственно. Тогда существует не более чем один мост B через D, который пересекает J' и удовлетворяет условию* $p(B, J-L) > 0$.

Доказательство. Если x и y — различные вершины цикла J, мы обозначим через $N(x, y)$ цепь в J, концы которой есть x и y и которая содержит E.

Обозначим через $\boldsymbol{U}$ класс всех мостов B через D, которые пересекаются с J' и удовлетворяют условию $p(B, J-L) > 0$.

Если $B \in \boldsymbol{U}$, то в силу (3.2) и (4.1) мы имеем также $q(B, J-L) > 0$.

Если $\boldsymbol{U}$ не пусто, мы получаем, используя (2.1), что существуют вершины x и y в J со следующими свойствами:

(i) x есть вершина в $J-L$;

(ii) y есть внутренняя вершина для L;

(iii) существует трансверсаль $M(x, y)$ цикла J через D с концами x и y;

(iv) $\alpha_1(N(x, y))$ имеет наименьшее значение при условии, что выполняются свойства (i), (ii) и (iii).

Пусть $D(x, y)$ — граничная область цепи $N(x, y)$, содержащаяся в D. В силу (4.2) имеем $D' \subseteq D(x, y)$. В силу (2.2) и (4.3) можно предположить, что трансверсаль $M(x, y)$ выбрана так, что ограничивающий цикл для $D(x, y)$ есть $M(x, y) \cup N(x, y)$. Пусть C — мост через D, имеющий подмножеством $M(x, y)$.

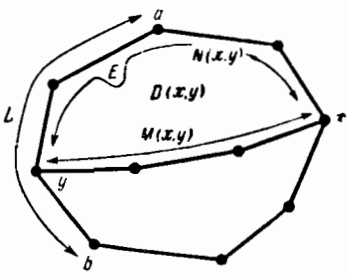


Рис. 2.

Предположим, что $B \in \boldsymbol{U}$ и $B \neq C$. Тогда B включает ребро из G, пересекающееся с $D(x, y)$, так как B пересекается с J' и $D' \subseteq D(x, y)$. Следовательно, B есть мост через $D(x, y)$ согласно следствию из (2.4). Так как $p(B, J-L) > 0$ и $q(B, J-L) > 0$, то из (2.1) и выбора x и y следует, что и x, и y являются вершинами B. Но это противоречит (4.1). Отсюда следует, что C есть единственный элемент в $\boldsymbol{U}$.

5. ДОКАЗАТЕЛЬСТВО ТЕОРЕМЫ 1

Для плоских графов, в которых всякое ребро является перешейком, теорема 1 не имеет смысла. Удобней сказать, что она тривиально выполняется для таких плоских графов.

Обозначим через $\boldsymbol{X}$ класс всех плоских графов, для которых теорема 1 верна. Через $\boldsymbol{Y}$ мы обозначим класс всех плоских графов G, таких, что $H \in \boldsymbol{X}$ всегда, когда H — плоский граф, имеющий меньше ребер, чем G. Заметим, что $G \in \boldsymbol{X}$, если G — плоский граф, удовлетворяющий условию $\alpha_1(G) \leqslant 1$.

Если G — плоский граф, L — замыкаемая цепь графа G и E — ребро в $G-L$, то обозначим через $\boldsymbol{V}(G, L, E)$ класс всех циклов J в G, имеющих следующие свойства:

(i) $L \cup \{E\} \subseteq J$;

(ii) если B есть мост для J, то $p(B, J-L) \leqslant 3$;

(iii) если B мост для J, пересекающий граничный цикл цепи L, то $p(B, J-L) = 2$.

(5.1) *Предположим* $G \in \mathbf{Y}$. *Пусть* L — *замыкаемая цепь в* G, *удовлетворяющая условию* $\alpha_1(L) \geqslant 2$, K — *граничный цикл для* L *и* E — *ребро в* $K - L$. *Тогда* $\mathbf{V}(G, L, E)$ *не пусто.*

Доказательство. Пусть граничными областями цепи L будут D и D' с ограничивающими циклами K и K' соответственно.

Пусть S — множество всех ребер в $G - L$, имеющих одним из концов внутреннюю вершину из L. Мы получим плоский граф H из $G - S$ путем замены элементов из L линией $|L|$, рассматриваемой как отдельное ребро. Та же операция превращает любой цикл Q в графе G, такой, что $L \subseteq Q$, в цикл $O(Q)$ в H, для которого $|O(Q)| = |Q|$ и, значит, $|L| \in O(Q)$. Ясно, что любой цикл в H, содержащий $|L|$ в качестве ребра, имеет вид $O(Q)$, где Q есть цикл в G, для которого $L \subseteq Q$. Получаем, что граничными областями для $|L|$ в H будут D и D', и соответствующими граничными циклами для $|L|$ будут $O(K)$ и $O(K')$ соответственно.

Далее, $\alpha_1(H) < \alpha_1(G)$, так как $\alpha_1(L) \geqslant 2$ и, значит, $H \in \mathbf{X}$. Применив теорему 1 к H, найдем, что существует цикл J в G, содержащий L и такой, что $O(J) \in \mathbf{V}(H, \{|L|\}, E)$.

Пусть B — произвольный мост для J в графе G. Если $B \subseteq S$, то в силу (2.1) B имеет не более одной вершины, которая не была бы внутренней вершиной для L. В оставшемся случае мы обозначим через B' непустое множество $B - (B \cap S)$. Это множество является мостом для $O(J)$ в H. Любое подмножество C в B', которое $O(J)$-ограничено в H, увеличивается до подмножества в B (которое J-ограничено в G), когда мы присоединяем к нему те элементы из S, которые инцидентны с вершинами из C, но не содержатся в J. Так как $O(J) \in \mathbf{V}(H, \{|L|\}, E)$, имеем:

(i) $w(B') \leqslant 3$ в H;

(ii) $w(B') = 2$ в H, если B' пересекается с $O(K)$ или $O(K')$.

Так как ни K, ни K' не пересекаются с S, получаем, что в любом случае $p(B, J - L) \leqslant 3$ в G и, далее, что $p(B, J - L) = 2$ в G, если B пересекается с K или с K'. Но $(|L|, E) \in O(J)$ и, таким образом, $L \cup \{E\} \subseteq J$. Отсюда следует, что $J \in \mathbf{V}(G, L, E)$.

(5.2) $\mathbf{Y} \subsetneqq \mathbf{X}$.

Доказательство. Пусть G — некоторый элемент из $\mathbf{Y}$, E — произвольное ребро в G, не являющееся перешейком, и E' — любое ребро, отличное от E, в некотором граничном цикле для E. Обозначим граничные области ребра E через D_1 и D_2 и отвечающие им граничные циклы через K_1 и K_2 соответственно, согласовывая обозначения так, что $E' \in K_1$.

Рассмотрим два случая. Первый случай. Допустим, что $\{E, E'\}$ есть цикл J в G. Тогда J имеет ровно две вершины, и

поэтому $w(B) \leqslant 2$ для всякого моста B для J. Используя (3.2), получаем, что $J \in V(G, \{E\}, E')$. Таким образом, в этом случае теорема 1 верна.

Второй случай. Здесь K_1 имеет по крайней мере три ребра. Следовательно, можно найти цепь L в K_1 с концами a и a', имеющую следующие свойства:

(i) $(E, E') \in L$;

(ii) a является концом E, но не E', а a' — конец E', но не E (см. рис. 3).

Пусть граничными областями цепи L будут D_3 и D_4. В силу (4.2) мы можем ввести обозначения так, что $D_1 \subseteq D_3$ и $D_2 \subseteq D_4$. Но так как $L \subseteq K_1$, мы имеем опять в силу (4.2), что $D_3 \subseteq D_1$. Следовательно, D_1 и D_3 тождественны. Обозначим ограничивающий цикл для D_4 через K_4. На рис. 3 область D_4 заштрихована.

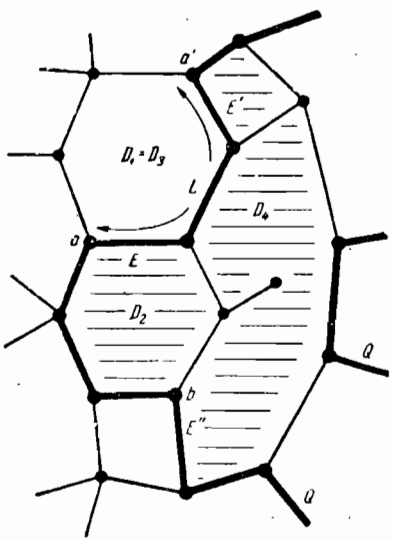


Рис. 3.

Выберем ребро E'' из $K_4 - L$ по следующим правилам. Если существует мост B через D_4, который пересекается с K_2 и имеет общую вершину с $K_4 - L$, то в силу (4.5) такой мост единственный. Более того, в силу (4.1) B имеет только одну вершину, скажем b, общую с $K_4 - L$. В этом случае мы выберем в качестве E'' ребро из $K_4 - L$, инцидентное с b. Если не существует такого моста B, то в качестве E'' выберем любой элемент из $K_4 - L$.

В силу (5.1) существует цикл Q в G, такой, что $Q \in V(G, L, E'')$. Мы обозначим области, выделенные циклом Q через R_1 и R_2, вводя обозначения так, чтобы $D_1 \subseteq R_1$ и $D_2 \subseteq D_4 \subseteq R_2$ (см. (4.2)).

Мост B для Q в G назовем *сингулярным*, если он удовлетворяет одному из следующих условий:

(i) $w(B) > 3$;

(ii) B пересекается с K_1 или с K_2, и $w(B) \neq 2$. В последнем случае имеем $w(B) > 2$ в силу (3.1).

Лемма 1. *Сингулярный мост B для Q есть мост через R_2. Он удовлетворяет условиям $p(B, Q - L) \leqslant 2$ и $q(B, Q - L) \geqslant 2$. Кроме того, если он пересекается с K_2, то выполнено условие $p(B, Q - L) \leqslant 1$.*

Доказательство. Сначала рассмотрим случай, когда $B \cap K_2$ имеет ребро, не принадлежащее K_4. Тогда B — мост через R_2, так как он пересекается с K_2. В силу (2.4) найдется мост B' через D_4, который пересекается с K_2, и для него $B' \subseteq B$. Далее, $p(B', K_4 - L) = 0$ или 1 в силу (4.1). Тогда, согласно выбору E'', всякая вершина, общая для B' и $K_4 - L$, является вершиной из Q. Поэтому в любом случае мост B' Q-ограничен и, таким образом, тождествен с B. Следовательно, $p(B, Q-L) \leqslant 1$. Так как мост B сингулярен, отсюда следует, что $w(B) \geqslant 3$ и $q(B, Q-L) \geqslant 2$.

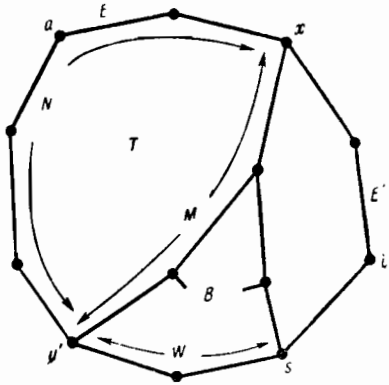


Рис. 4.

Осталось рассмотреть случай, когда B пересекается с K_4, если он пересекается с K_2. Поэтому из определений Q и B соотношения $w(B) > 3$ и $p(B, Q-L) \leqslant 3$ выполняются, если B не пересекается с K_1 или с K_4, и соотношения $w(B) > 2$ и $p(B, Q-L) = 2$ выполняются, если B пересекается с K_1 или с K_4. В любом случае B имеет вершину x, которая является внутренней вершиной для L. Пусть A — ребро из B, инцидентное с x.

Предположим, что B — мост через R_1. Тогда A пересекается с D_1. В силу (2.4) найдется мост B_1 через D_1, такой, что $A \in B_1 \subseteq B$. В силу (4.1) $w(B_1) = 1$. Следовательно, мост B_1 Q-ограниченный и, значит, совпадает с B. Но это невозможно, так как $w(B) \geqslant 3$. Получаем, что B — мост через R_2.

Если B пересекается с K_2, можно полагать, что $B \cap K_2 \subseteq K_4$. Доводы, используемые в этом случае, проиллюстрированы на рис. 4, где внешний цикл представляет собой K_4. Должна существовать цепь W в $B \cap K_2$, концы которой, скажем y и z, являются вершинами сочленения для B. Мы так выбираем обозначения, что либо $y = a$, либо y разделяет a и z в $|K_4 - L|$. Пусть N — цепь в K_4 с концами x и y, включающая E. В силу (2.1) найдется трансверсаль M цикла K_4 через D_4 с концами x и y. Пусть T — область, выделенная циклом $M \cup N$ и содержащаяся в D_4. Тогда в силу (4.2) $D_2 \subseteq T$. Но этого не может быть, так как граничная точка z из D_2 не принадлежит ни T, ни ее границе $M \cup N$.

Наконец, предположим, что B не пересекается с K_2. Поскольку A пересекается с D_4, существует мост B' через D_4, такой, что $A \in B' \subseteq B$. Если каждая вершина сочленения для B' является вершиной из Q, находим, что $B' = B$. Тогда в силу

(4.1) $p(B, Q-L)=p(B', K_4-L)\leqslant 1$. Так как $w(B)\geqslant 3$, то $q(B, Q-L)\geqslant 2$. Если вместо этого B' имеет вершину сочленения z, которая не является вершиной из Q, то B включает два ребра из K_4, инцидентные с z. Поэтому в этом случае $p(B, Q-L)=2$. Но $w(B)>3$, так как B не пересекается с K_2. Следовательно, $q(B, Q-L)\geqslant 2$. Этим завершается доказательство леммы.

Пусть B — любой сингулярный мост для Q. По лемме 1 $q(B, Q-L)\geqslant 2$. Следовательно, существует единственная цепь $L(B)$ в $L-\{E, E'\}$ с концами в вершинах из B, вершины которой включают в себя все вершины сочленения из B, являющиеся внутренними для L. Ясно, что $L(B)$ есть объединение сегментов цикла Q, определяемых B. Будем говорить, что мост B' для Q, отличный от B, *огорожен* B, если одна из его вершин сочленения является внутренней для $L(B)$.

Если B' огорожен B, то все вершины сочленения для B' являются вершинами из $L(B)$. Если B' есть мост через R_2, это следует из (4.4). Если же B' — мост через R_1, то мы докажем это следующим образом. Существует ребро A из B', инцидентное с внутренней для $L(B)$ вершиной x. Это ребро должно пересекаться с D_1. В силу (2.4) найдется мост B'' через D_1, такой, что $A\in B''\subseteq B'$. Согласно (4.1), x — единственная вершина сочленения для B''. Но тогда B'' Q-ограничено и, значит, совпадает с B'.

Если B огораживает другой сингулярный мост B' (т. е. B' огорожен B), то B' не огораживает B. Ибо в противном случае полученные выше результаты в соединении с (4.4) дадут, что $L(B)$ совпадает с $L(B')$ и является сегментом в Q, определяемым одновременно B и B'.

Мы получаем, что существует множество $\boldsymbol{B}=\{B_1, \ldots, B_k\}$ сингулярных мостов для Q, таких, что никакой элемент из $\boldsymbol{B}$ не огораживает другого и всякий сингулярный мост для Q огорожен некоторым элементом из $\boldsymbol{B}$.

Для каждого $B_i\in\boldsymbol{B}$ мы определим соответствующую *сингулярность* Z_i цикла Q как объединение B_i, $L(B_i)$ и всех мостов для Q, огороженных B_i.

Сингулярности Z_i и Z_j, соответствующие двум различным элементам B_i и B_j из $\boldsymbol{B}$, не имеют общих ребер. Действительно, допустим, что у них есть общее ребро A. Если $A\notin L$, то мост для Q, содержащий A, должен быть огорожен как B_i, так и B_j. В таком случае некоторая внутренняя вершина для $L(B_i)$ будет вершиной для $L(B_j)$, и, значит, эти две цепи имеют общее ребро. Мы можем, таким образом, считать, что $A\in L(B_i)\cap L(B_j)$. Так как ни B_i, ни B_j не огораживают друг друга, то цепи $L(B_i)$ и $L(B_j)$ имеют одни и те же концы и, значит, тождественны. Кроме того, $L(B_i)$ является сегментом цикла Q,

определяемым одновременно B_i и B_j. Если $w(B_i) > 2$, то этот сегмент включает в число своих вершин все вершины сочленения для B_j в силу (4.4) и, таким образом, $w(B_j) = 2$. Это противоречит нашему предположению о том, что как B_i, так и B_j — сингулярные мосты.

Для каждого $B_i \in \boldsymbol{B}$ всякое ребро из G, инцидентное с внутренней вершиной из $L(B_i)$, необходимо является ребром из Z_i. Следовательно, вершинами сочленения для Z_i будут вершины, общие для B_i и $Q - L(B_i)$. Определим *норму* Z цикла Q как множество всех ребер из $G - Q$, которые не принадлежат никакой сингулярности цикла Q. Каждая вершина сочленения для Z является вершиной из Q, причем она не внутренняя и для цепи $L(B_i)$.

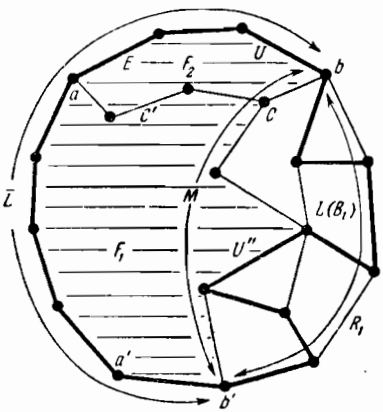


Рис. 5.

Для каждой сингулярности Z_i обозначим $G_i = Q \cup Z_i$. Так как $G_i \subseteq G$, имеем $G_i \in \boldsymbol{Y}$.

Лемма II. *Всякое G_i имеет цикл Q_i со следующими свойствами*:

(i) $Q - L(B_i) \subseteq Q_i$;

(ii) *если C — мост для Q_i в G_i*, то $w(C) \leqslant 3$. *Кроме того, если C пересекается с K_1 или с K_2, то $w(C) = 2$.*

Доказательство. Обозначим концы $L(B_i)$ через b и b' таким образом, что b отделяет b' от конца a в цепи L и через U ребро из $\bar{L}$, инцидентное с b. Обозначим $\bar{L} = Q - L(B_i)$.

Мостами через R_1 в G_i будут лишь те мосты через R_1 в G, которые огорожены B_i. Мы видели, что каждый такой мост имеет число сочленения, равное 1. Следовательно, по (3.2) никакой цикл в G_i, содержащий U, не имеет ребра, пересекающегося с R_1. Отсюда R_1 является граничной областью ребра U и цепи $\bar{L}$ в G_i. Пусть граничными областями ребра U и цепи $\bar{L}$, заключенными в R_2, будут F_2 и F_4 соответственно, и пусть их ограничивающими циклами будут P_2 и P_4 соответственно. Из (4.2) получаем, что $D_2^* \subseteq F_2$, так как P_2 должен включать все ребра из L, лежащие между a и b. На рис. 5 заштрихована область, представляющая собой F_4.

В силу (4.3) $P_4 = \bar{L} \cup M$, где M есть трансверсаль цикла Q через R_2 с концами b и b'. Так как $\alpha_1(\bar{L}) \geqslant 2$, то из (5.1) следует, что для каждого ребра U'' в M можно найти цикл Q_i в G_i, имеющий следующие свойства:

(1) $Q - L(B_i) \subseteq Q_i$;

(2) *Если* C *— мост для* Q_i *в* G_i, *то* $p(C, Q_i - L) \leqslant 3$. *Кроме того, если* C *пересекается с* Q *или с* P_4, *то* $p(C, Q_i - L) = 2$;
(3) $U'' \in Q_i$.

На рис. 5 жирной линией выделен цикл, удовлетворяющий этим условиям.

Мы покажем, что всегда можно выбрать U'' так, чтобы Q_i обязательно удовлетворял условию (ii) в лемме 11. Но сперва тем не менее мы установим некоторые свойства Q_i, имеющие место при любом выборе U''.

Обозначим через S_1 и S_2 области, выделенные циклом Q_i, так что $R_1 \subseteq S_1$ и $S_2 \subseteq R_2$. Имеем $F_4 \subseteq S_2$ в силу (4.2).

Пусть C — произвольный мост для Q_i в G_i. Если он пересекается с K_2, то так как $D_2 \subseteq F_4$, он пересекается либо с P_4, либо с мостом C' через F_4 в G_i. В последнем случае C' имеет не более одной вершины сочленения, не являющейся внутренней вершиной для L в силу (4.1). Но если $w(C') \leqslant 1$ в G_i, то ясно, что $w(C') \leqslant 1$ в G, что противоречит (3.2). Следовательно, C пересекается либо с P_4, либо с мостом C' через F_4 в G_i, так что $w(C') \geqslant 2$ и по меньшей мере одна вершина сочленения для C' является внутренней вершиной для L. В последнем случае должно быть $C' \subseteq C$.

Аналогичное рассуждение, в котором L, F_4 и P_4 заменены на $\{U\}$, F_2 и P_2 соответственно, показывает, что если C пересекается с K_2, то C пересекается также и с P_2. Аналогично если C пересекается с K_1, то он пересекается также и с Q.

Если никакая вершина из C не является внутренней вершиной для L, то $p(C, Q_i - L) = w(C)$. Тогда из (2) и полученных выше результатов следует, что C должен удовлетворять (ii).

Допустим, что C — мост через S_1, содержащий внутреннюю вершину из L. Он не пересекается с K_2, так как $D_2 \subseteq S_2$. Он имеет ребро A, инцидентное с внутренней вершиной для L. Это ребро должно пересекаться с R_1. Поэтому по (2.4) существует мост C' через R_1, такой, что $A \in C' \subseteq C$. Тогда из (4.1) получаем $w(C') = 1$. Следовательно, C' Q_i-ограничено и, таким образом, $C = C'$. Значит, $w(C) = 1$. Далее, C не пересекается с Q и, таким образом, не пересекается с K_1. Следовательно, он удовлетворяет условию (ii).

Осталось лишь показать, что можно выбрать U'' так, чтобы (ii) выполнялось для любого моста C через S_2, удовлетворяющего условию $q(C, Q_i - L) \geqslant 1$.

Предположим, $p(B_i, Q - L) = 0$. Тогда не существует моста C через Q_i в G_i, для которого $q(C, Q_i - L) \geqslant 1$ при любом выборе U''. Следовательно, (ii) выполняется для любого выбора U''.

Теперь допустим, что $p(B_i, Q - L) = 1$. Обозначим через x вершину, общую для B_i и $Q - L$. Это единственная внутренняя вершина из L, являющаяся вершиной для Z_i. Она делит L на

две непересекающиеся цепи: L_1 с концами b и x и L_2 с концами x и b' (см. рис. 6).

Теперь все циклы в G_i, имеющие U своим ребром, должны содержать L_1. Следовательно, граничная область цепи L_1, содержащаяся в R_2, есть F_2. В силу (4.3) имеем $P_2 = L_1 \cup M'$, где M' есть трансверсаль цикла Q через R_2 с концами b и x. Пусть y будет последней вершиной в M', отсчитываемой от b, которая является вершиной в M, и пусть T — цепь в M' с концами x и y. Тогда T является трансверсалью цикла P_4 через F_4. В силу (2.2) она будет подмножеством моста C'' для P_4 через F_4 в G_i. Согласно (4.1), y является единственной вершиной, общей для C'' и M. Таким образом, $w(C'') = 2$. Далее, y — внутренняя вершина для M. В противном случае C'' был бы мостом для Q в G, отличным от B_i и не огороженным B_i. Значит, он не был бы подмножеством в G_i. Соответственно y делит M на две непересекающихся цепи: M_1 с концами b и y и M_2 с концами y и b'.

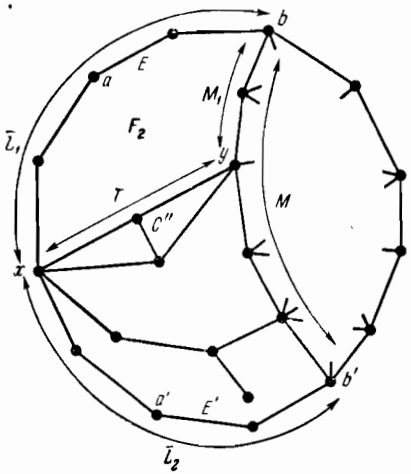


Рис. 6.

Пусть F — область, выделенная циклом $L_1 \cup T \cup M_1$ и содержащаяся в R_2. Тогда $F_2 \subseteq F \subseteq F_4$. Но если ребро A из M' пересекается с F, то оно является ребром трансверсали M'' цикла P_4 через F_4, концами которой будут вершины из M. Это невозможно в силу (2.2) и (4.1). Получаем, что $F = F_2$. Следовательно, $P_2 = L_1 \cup T \cup M_1$.

Выбираем в качестве U'' ребро из M, инцидентное с y. Тогда y — вершина в Q_i. Следовательно, C'' — мост в Q_i. Так как $w(C'') = 2$, то он удовлетворяет (ii).

Пусть C — любой мост через S_2 в G_i, такой, что $q(C, Q_i - L) \geqslant 1$, и содержащий вершину x. Применяя (2.4), получаем, что существует мост B через F_4 в G_i, являющийся подмножеством в C и содержащий вершину x. Тогда $w(B) \leqslant 2$ в силу (4.1). Если всякая вершина сочленения для B является вершиной из Q_i, то $B = C$. Тогда C удовлетворяет (ii), так как B пересекается с $P_2 = L_1 \cup T \cup M_1$, только если $B = C''$. Допустим тем не менее, что B имеет вершину сочленения z, которая не есть вершина из Q_i. Если z — внутренняя вершина для M_1, то $|B|$ пересекается с F_2. Получаем, что B — мост через F_2 согласно следствию из (2.4). Но это невозможно в силу (4.1), так

как $w(B) = 2$. Мы получили, что z — внутренняя вершина для M_2. Тогда C пересекается с M_2 и, согласно (2), $p(C, Q_i - L) = 2$. Следовательно, $w(C) = 3$. Далее, тремя вершинами сочленения для C должны быть x и две вершины из M_2. Не может существовать никакой вершины сочленения C, которая являлась бы вершиной из M_1, и, значит, C не пересекается с M_1. Так как C не совпадает с C'', то C не пересекается с P_2 и, значит, не пересекается с K_2. Условие (ii) опять выполняется.

По лемме 1 остается лишь случай, когда $p(B_i, Q - L) = 2$. В этом случае мы обозначим вершины, общие для B_i и $Q - L$, через x и y, так что x разделяет b и y в L. Тогда лишь x и y будут вершинами из Z_i, являющимися внутренними вершинами для L.

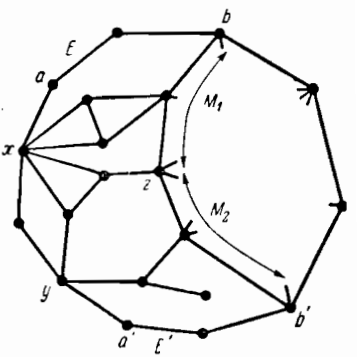


Рис. 7.

Вершина t из M есть *x-связанная* (*y-связанная*), если найдется трансверсаль цикла P_4 через F_4 в G_i с концами x и t (y и t). Теперь если t — x-связанная и u — y-связанная вершины из M, то t и x не разделяют u и y в $|P_4|$ в силу (4.1). Далее, M имеет по меньшей мере одну внутреннюю вершину. Иначе M была бы мостом для Q в G, отличным от B_i и не огороженным B_i, что противоречит определению G_i. Отсюда следует, что существует внутренняя вершина z из M, разделяющая M на две непересекающихся цепи, M_1 с концами b и z и M_2 с концами z и b', таким образом, что все x-связанные вершины из M являются вершинами из M_1 и все y-связанные вершины из M являются вершинами из M_2 (см. рис. 7).

Найдя такую вершину z, мы выбираем U'', инцидентное с z. Тогда z — вершина из Q_i.

Пусть C — произвольный мост через S_2 в G_i, для которого $q(C, Q_i - L) \geqslant 1$. Тогда или x, или y будет вершиной из C. Если x — вершина из C, то, согласно (2.4), найдется мост B через F_4 в G_i, являющийся подмножеством в C и содержащий вершину x. Тогда по (4.1) $w(B) \leqslant 3$. Если всякая вершина сочленения для B есть вершина из Q_i, то $B = C$. Если же B имеет вершину сочленения t, которая не принадлежит Q_i, то t есть внутренняя вершина для M_1, так как она является x-связанной вершиной из M в силу (2.1). Следовательно, либо $w(C) \leqslant 3$, либо M_1 и C имеют общее ребро. Аналогично если y есть вершина из C, то или $w(C) \leqslant 3$, или C пересекается с M_2.

Предположим, что $w(C) \geqslant 4$. Тогда C пересекается либо с M_1, либо с M_2 и, значит, в силу (2), $p(C, Q_i - L) = 2$.

Соответственно как x, так и y, должны быть вершинами из C. Но тогда C пересекается и с M_1, и с M_2. Следовательно, по меньшей мере две вершины из M_1 и по меньшей мере две вершины из M_2 являются вершинами сочленения для C. Это невозможно, так как $p(C, Q_i - L) = 2$ и цепи M_1 и M_2 имеют только одну общую вершину.

Мы получили, что $w(C) \leqslant 3$. Теперь всякое ребро из C, инцидентное с x или y, должно принадлежать B_i в силу определения G_i. Но $B_i \cap C$ Q_i-ограничено, так как C не пересекается с Q (потому что $S_2 \subseteq R_2$). Значит, $B_i \cap C = C$, и поэтому $C \subseteq B_i$. Следовательно, по лемме 1, C не пересекает K_2. Соответственно C удовлетворяет (ii).

Теперь мы проверили, что при удачном выборе U'' цикл Q_i должен удовлетворять условию (ii). Тогда утверждение леммы следует из (I).

Мы получим плоский граф J из Q путем замены цепи $L(B_i)$ на цепь $Q_i - L$ в каждой сингулярности Z_i. Легко проверить, что J есть цикл в G, включающий все вершины сочленения сингулярностей Z_i и нормы Z для Q в качестве своих вершин.

Пусть B — любой мост для J в G. Тогда B пересекается с W, где W есть либо Z, либо одна из сингулярностей Z_i цикла Q. В любом случае $W \cap B$ J-ограничено, откуда следует, что $B \subseteq W$. Если $W = Z$, то это означает, что B — несингулярный мост для Q в G. Тогда $w(B) \leqslant 3$ и $w(B) = 2$, если B пересекается с K_1 или с K_2. Если $W = Z_i$, то B есть мост для Q_i в G_i и мы применяем лемму II для получения того же результата.

Отсюда следует, что теорема 1 верна для плоского графа G. Так как G может быть любым элементом из $\boldsymbol{Y}$, получаем, что $\boldsymbol{Y} \subseteq \boldsymbol{X}$, что и требовалось.

Теорема 1 далее доказывается по индукции. Ибо мы знаем, что она тривиально выполняется при $\alpha_1(G) \leqslant 1$. И если она верна для всех плоских графов с n или менее ребрами, то она также верна и для всех графов с $n + 1$ ребрами, поскольку такие графы принадлежат $\boldsymbol{Y}$ и, согласно (5.2), также и $\boldsymbol{X}$.

6. ГАМИЛЬТОНОВЫ ЦИКЛЫ

Используя (3.2) и (4.1), видим, что если K — граничный цикл ребра E в плоском графе G и если E' — другое ребро из K, то K будет также и граничным циклом для E'. Соответственно мы говорим о граничных циклах для ребер из G как о граничных циклах в G.

Теорема II. *Пусть G — любой 4-связный плоский граф, имеющий по меньшей мере два ребра. Тогда G имеет гамильто-*

нов цикл. Кроме того, если никакие два ребра из G не имеют оба конца общими и если E и E' — различные ребра одного и того же граничного цикла в G, тогда найдется гамильтонов цикл в G, содержащий как E, так и E'.

Доказательство. Предположим сперва, что никакие два ребра из G не имеют два общих конца. Пусть E и E' — различные ребра граничного цикла K в G. Найдется цикл J в G, удовлетворяющий условиям (i), (ii) и (iii) теоремы 1. Если каждый мост цикла J в G имеет ровно одно ребро, обоими концами которого являются вершины из J, то J есть гамильтонов цикл для G, включающий ребра E и E'. Если же нет, то из (3.5) и из ограничения на граф G следует, что $\alpha_1(J) = 3$, что найдется только один мост для J в G и что этот мост имеет число сочленения, равное 3. Но в силу (4.3), мост должен пересекать граничный цикл для E. Это невозможно в силу условия (iii) теоремы 1.

Для доказательства первой части теоремы заметим, что G имеет по меньшей мере один цикл в силу (3.3). Если G имеет только две вершины, то всякий цикл в G является гамильтоновым. Соответственно будем предполагать $\alpha_0(G) \geqslant 3$. Если мы уберем некоторое ребро из G, которое имеет те же концы, что и некоторое другое, то получим из G другой 4-связный плоский граф с теми же вершинами. Повторив эту операцию нужное число раз, получим 4-связный плоский граф H, в котором никакие два ребра не имеют оба своих конца общими и который имеет те же вершины, что и G. Так как $\alpha_0(G) \geqslant 3$, то $\alpha_1(H) \geqslant 2$. Следовательно, по (3.3) H имеет цикл и, значит, граничный цикл. Поэтому в силу уже доказанного H имеет гамильтонов цикл J. Он же будет гамильтоновым циклом и в G.

Теорема Х. Уитни [2] относится к частному случаю теоремы II, когда каждый граничный цикл имеет ровно три ребра.

7. ЗАМЕЧАНИЕ К ГИПОТЕЗЕ О ЧЕТЫРЕХ КРАСКАХ

Банальной истиной в теории раскрашивания карт является тот факт, что всякую карту, определяемую плоским графом, имеющим гамильтонов цикл, можно раскрасить в четыре краски. Районы в одной области, выделяемой циклом, можно раскрасить, чередуя красный и синий цвета, а в другой — чередуя зеленый и желтый. Поэтому теперь мы можем утверждать, что гипотеза о четырех красках верна для всех карт, определяемых 4-связными плоскими графами.

К сожалению, наиболее интересные карты определяются плоскими графами, в которых степень каждой вершины равна 3.

Если такой плоский граф имеет 5 или более вершин, то в силу (3.4) он не является 4-связным. Мало известно об условиях существования гамильтонова цикла в плоском графе G такого вида, кроме того, что недостаточно 3-связности [1].

СПИСОК ЛИТЕРАТУРЫ

1. Tutte W. T., On Hamiltonian circuits, *J. London Math. Soc.*, **21** (1946), 98—101.
2. Whitney Hassler, A theorem on graphs, *Ann. of Math.*, **32** (1931), 378—390.

Быстрое умножение больших чисел[1)]

А. Шёнхаге, В. Штрассен

1. ВВЕДЕНИЕ

«Школьный» метод умножения двух десятичных чисел может быть использован без существенных изменений для умножения N-разрядных двоичных чисел на многоленточной машине Тьюринга или для построения специальной логической сети (из логических элементов с двумя входами). В обоих случаях сложность (под сложностью сети мы понимаем число ее элементов) растет как N^2.

На машине Тьюринга, которая производит умножение чисел любой длины, сложность умножения N-разрядных чисел определяют как максимум числа сдвигов головок по всем входным парам чисел длины N.

Распространенное и казавшееся правдоподобным мнение, что требуемая в школьном методе сложность не может быть уменьшена, было опровергнуто в 1962 г. А. А. Карацубой [4], построившим сеть с

$$O\left(N^{\log_2 3}\right)$$

элементами ($\log_2 3 \approx 1{,}58$). Этот метод без затруднения переносится на машины Тьюринга.

Дальнейшие сюрпризы содержала появившаяся в следующем году заметка А. Л. Тоома [7], в которой была указана сеть для умножения N-разрядных двоичных чисел из

$$O\left(N2^{\text{const}\cdot\sqrt{\lg N}}\right)$$

логических элементов.

Независимо от Тоома и совсем иными методами Шёнхаге [6] в 1966 г. показал, что на машине Тьюринга умножение N-разрядных чисел можно делать со сложностью

$$O\left(N2^{\sqrt{2\log_2 N}}\,(\lg N)^{3/2}\right).$$

Впоследствии алгоритм Тоома был перенесен на машины Тьюринга (Кук [1]) с более точной оценкой сложности:

$$O\left(N2^{\sqrt{2\log_2 N}}\,\lg N\right)$$

[1)] Schönhage A., Strassen V., Schnelle Multiplikation großer Zahlen, *Computing, Archiv für elektronisches Rechnen*, **7**, Fasc. 3—4 (1971), 281—292.

(Кук [1] и Кнут [5, стр. 273]). Существенно различные методы Тоома и Шёнхаге дают практически одинаковые оценки сложности, и это позволяет считать оценки близкими к оптимальной.

Интересующемуся читателю рекомендуем гл. 4 блестящей энциклопедии Кнута «Искусство программирования», в которой доказаны и обстоятельно изложены перечисленные здесь результаты.

В настоящей работе будут указаны два способа умножения N-разрядных двоичных чисел, которые допускают реализацию как в логической сети, так и на машине Тьюринга. Сложность одного способа

$$O(N \lg N (\lg \lg N)^{1+\varepsilon}),$$

другого —

$$O(N \lg N \lg \lg N).$$

Оба способа используют быстрое преобразование Фурье (Кулей и Тьюки [3]; независимо от нас эту же идею — использовать быстрое преобразование Фурье для умножения больших чисел — высказывал Кнут). Роль преобразования Фурье можно понять, заметив, что произведение многоразрядных чисел, отвлекаясь от переносов, является сверткой. Если разбить сомножители на блоки подходящей длины и рассматривать их как элементы такого кольца R, операции которого верно передают исходные вычисления (в кольце $\boldsymbol{Z}$ целых чисел) и которое, кроме того, содержит необходимые нам корни из единицы, то можно требуемое «большое» умножение разложить в такую последовательность действий: преобразование Фурье последовательностей блоков для обоих сомножителей, покомпонентное перемножение преобразованных последовательностей, обратное преобразование и выполнение переноса. С возникающими при этом «малыми» умножениями будем обходиться аналогично. Таким образом получается итеративная процедура.

В нашем первом способе в качестве R берется поле $\mathbf{C}$ комплексных чисел, длина блока $\approx \lg N$, и можно получить сложность $O(N \lg N (\lg \lg N)^{1+\varepsilon})$ уже после третьей итерации.

Во втором нашем способе мы употребляем в качестве R кольцо классов вычетов $\mathbf{Z}_{F_n}$ кольца $\mathbf{Z}$ по модулю чисел Ферма $F_n = 2^{2^n} + 1$, длина блока $\approx \sqrt{N}$ и требуется приблизительно $\lg \lg N$ итераций. Числа Ферма здесь удобны тем, что двойка является примитивным корнем из 1 степени 2^{n+1} по модулю F_n, а вычеты ее степеней по тому же модулю имеют весьма простое двоичное представление, так что умножение на них не составляет труда.

Второй способ можно так реализовать в логической сети, что глубина сети (определяющая время ее срабатывания) бу-

дет $O(\lg N)$. Однако доказательства этого факта мы не приводим. Ясно, что глубина порядка $\lg N$ — наилучшая из возможных.

Мы не считаем, что минимальная сложность умножения есть величина порядка $N \lg N \lg\lg N$, и предполагаем, что ее порядок равен $N \lg N$ (см. глубокий результат Кука и Андераа [2]; к сожалению, on-line — ограничение [1]) для логической сети — неприемлемо, а для вычисления на машине Тьюринга, во всяком случае, слишком сильно и, кроме школьного метода, никакой из указанных способов для вычислений on-line не подходит).

В следующем разделе мы описываем быстрое преобразование Фурье в удобной нам форме. В третьем разделе мы кратко излагаем первый, более простой способ умножения с $R = \mathbf{C}$ и, наконец, в четвертом разделе подробно рассматривается второй способ с $R = \mathbf{Z}_{F_n}$.

2. БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ

Пусть R — коммутативное кольцо с единицей, $w_n \in R$ — примитивный корень из 1 степени 2^n, так что $w_n^{2^{n-1}} = -1$, а $2 = 1 + 1$, 1 — единица кольца R.

Тогда преобразование

$$\hat{a}_k = \sum_{j=0}^{2^n-1} a_j w_n^{jk} \quad (0 \leqslant k < 2^n), \tag{2.1}$$

которое каждому вектору $a \in R^{2^n}$ сопоставляет фурье-образ $\hat{a}$, может быть разложено по следующей схеме в отдельные шаги.

Для k и j используем двоичное представление

$$k = \sum_{\nu=0}^{n-1} k_\nu 2^\nu, \quad j = \sum_{\nu=0}^{n-1} j_\nu 2^{n-1-\nu} \quad (k_\nu, j_\nu = 0 \text{ или } 1). \tag{2.2}$$

Отправляясь от

$$A_0(j_0, \ldots, j_{n-1}) = a_j \quad (0 \leqslant j < 2^n), \tag{2.3}$$

определяем $A_1, \ldots, A_n$ рекуррентно следующим образом:

$$A_{\nu+1}(k_0, \ldots, k_\nu, j_{\nu+1}, \ldots, j_{n-1}) =$$

$$= \sum_{j_\nu=0}^{1} A_\nu(k_0, \ldots, k_{\nu-1}, j_\nu, \ldots, j_{n-1}) w_n^{j_\nu 2^{n-1-\nu}(k_\nu 2^\nu + \ldots + k_0 2^0)} \tag{2.4}$$

[1]) В on-line вычислениях на машине Тьюринга предполагается, что входные данные поступают в машину по специальному входу символ за символом. Каждый входной символ требует выдачи соответствующего выходного символа, и новый символ не может быть подан на вход, пока машина вычисляет выходной символ, соответствующий предыдущему входному. (См. сборник «Проблемы математической логики», изд-во «Мир», 1970, стр. 223.) — *Прим. перев.*

или подробнее, принимая во внимание равенство $w_n^{2^{n-1}} = -1$,

$$\left.\begin{aligned} &A_{\nu+1}(\ldots, k_{\nu-1}, 0, j_{\nu+1}, \ldots) = \\ &= A_\nu(\ldots, k_{\nu-1}, 0, j_{\nu+1}, \ldots) + A_\nu(\ldots, k_{\nu-1}, 1, j_{\nu+1}, \ldots) w_n^\varkappa, \\ &A_{\nu+1}(\ldots, k_{\nu-1}, 1, j_{\nu+1}, \ldots) = \\ &= A_\nu(\ldots, k_{\nu-1}, 0, j_{\nu+1}, \ldots) - A_\nu(\ldots, k_{\nu-1}, 1, j_{\nu+1}, \ldots) w_n^\varkappa, \end{aligned}\right\} \quad (2.5)$$

где $\varkappa = 2^{n-1-\nu}(k_{\nu-1}2^{\nu-1} + \ldots + k_0 2^0)$.

Из (2.3) и (2.4) с учетом того, что

$$w_n^{(j_0 2^{n-1} + \ldots + j_{\nu-1}2^{n-\nu})\, k_\nu 2^\nu} = 1,$$

получаем по индукции окончательное выражение:

$$A_{\nu+1}(k_0, \ldots, k_\nu, j_{\nu+1}, \ldots, j_{n-1}) = \sum_{j_\nu=0}^{1} \ldots \sum_{j_0=0}^{1} a_j w_n^{(j_0 2^{n-1} + \ldots + j_\nu 2^{n-1-\nu})(k_\nu 2^\nu + \ldots + k_0 2^0)}.$$

Из (2.1) и (2.2), в частности, следует, что

$$A_n(k_0, \ldots, k_{n-1}) = \hat{a}_k. \quad (2.6)$$

Для обратного преобразования $\hat{a} \longmapsto a$ из системы (2.5) получаем

$$\left.\begin{aligned} &A_\nu(\ldots, k_{\nu-1}, 0, j_{\nu+1}, \ldots) = \\ &= 2^{-1}(A_{\nu+1}(\ldots, k_{\nu-1}, 0, j_{\nu+1}, \ldots) + A_{\nu+1}(\ldots, k_{\nu-1}, 1, j_{\nu+1}, \ldots)), \\ &A_\nu(\ldots, k_{\nu-1}, 1, j_{\nu+1}, \ldots) = \\ &= 2^{-1} w_n^{-\varkappa}(A_{\nu+1}(\ldots, k_{\nu-1}, 0, j_{\nu+1}, \ldots) - A_{\nu+1}(\ldots, k_{\nu-1}, 1, j_{\nu+1}, \ldots)), \end{aligned}\right\} \quad (2.7)$$

где $\varkappa = 2^{n-1-\nu}(k_{\nu-1}2^{\nu-1} + \ldots + k_0 2^0)$.

Далее будет существенно следующее обстоятельство.

В силу (2.3) и (2.4) имеем для $0 \leqslant j < 2^{n-1}$

$$A_1(1, j_1, \ldots, j_{n-1}) = A_0(0, j_1, \ldots, j_{n-1}) - A_0(1, j_1, \ldots, j_{n-1}) = a_j - a_{j+2^{n-1}}, \quad (2.8)$$

и на последующих шагах нашей рекуррентной процедуры (2.5) эти разности используются лишь для получения A_ν с $k_0 = 1$, а на последнем шаге только для получения $\hat{a}_k$ с нечетным k; наоборот, для нахождения этих разностей при обратной процедуре (2.7) достаточно знать только 2^{n-1} значений $\hat{a}_k$ с нечетным k.

3. УМНОЖЕНИЕ С ПОМОЩЬЮ КОМПЛЕКСНЫХ ЧИСЕЛ

В этом разделе мы описываем конструкцию для получения быстрых способов умножения с использованием быстрого преобразования Фурье в кольце $R = \mathbf{C}$ комплексных чисел. Отправляясь от школьного метода V_0, мы строим с ее помощью последовательность способов $V_1, V_2, \ldots$. Предполагая, что V_m уже построен, мы укажем, как по целым $a \geqslant 0$, $b \geqslant 0$, заданным N-разрядным двоичным представлением, получается $2N$-разрядное произведение $c = ab$ в способе V_{m+1}.

Возьмем натуральные числа l и n, удовлетворяющие условию

$$l \cdot 2^n \geqslant 2N \tag{3.1}$$

(позднее мы их выберем подходящим образом в зависимости от N), и разобьем сомножители a и b соответственно в блоки длины l:

$$a = \sum_{j=0}^{2^n-1} a_j 2^{jl}, \quad b = \sum_{j=0}^{2^n-1} b_j 2^{jl}, \tag{3.2}$$

где $0 \leqslant a_j < 2^l$, $0 \leqslant b_j < 2^l$ и $a_j = b_j = 0$ для $j \geqslant 2^{n-1}$. Тогда произведение запишется в виде

$$c = ab = \sum_{\tau=0}^{2^n-1} c_\tau 2^{\tau l}, \tag{3.3}$$

где

$$c_\tau = \sum_{\rho+\sigma=\tau} a_\rho b_\sigma = \sum_{\rho+\sigma \equiv \tau \pmod{2^n}} a_\rho b_\sigma,$$

т. е. c_τ получается в результате свертки a_ρ с b_σ. Описанное в разд. 2 преобразование Фурье с $w_n = e^{2\pi i \cdot 2^{-n}}$ переводит эту свертку в 2^n умножений, а именно

$$\hat{a}_k \hat{b}_k = \left(\sum_{\rho=0}^{2^n-1} a_\rho w_n^{\rho k}\right)\left(\sum_{\sigma=0}^{2^n-1} b_\sigma w_n^{\sigma k}\right) = \sum_{\tau=0}^{2^n-1} c_\tau w_n^{\tau k} = \hat{c}_k \quad (0 \leqslant k \leqslant 2^n). \tag{3.4}$$

Теперь можно в общих чертах наметить способ V_{m+1}:

переход от a_ρ к $\hat{a}_k$ и соответственно от b_σ к $\hat{b}_k$ согласно (2.5); (3.5)

выполнение 2^n умножений $\hat{a}_k \hat{b}_k = \hat{c}_k$; (3.6)

переход от $\hat{c}_k$ к c_τ в соответствии с (2.7); (3.7)

сложение c_τ согласно (3.3). (3.8)

Чтобы реализовать этот план, необходимо взять достаточно точные приближенные значения для комплексных чисел, возникающих на шагах (3.5) — (3.7). Погрешность при этом должна

быть настолько малой, чтобы целые числа c_τ получались с ошибкой $<1/2$ и тем самым точно определялись из округления. При переходе от b_j к $\hat{b}_k$ и от $\hat{c}_k$ к c_j вычисляются величины $B_\nu(\ldots)$ и $C_\nu(\ldots)$, аналогичные $A_\nu(\ldots)$ (см. 2.2). Так как $a_j < 2^l$, $b_j < 2^l$, то в силу (2.5) и (2.7)

$$|A_\nu(\ldots)| < 2^{l+\nu},\quad |B_\nu(\ldots)| < 2^{l+\nu},\quad |\hat{a}_k| < 2^{l+n},\quad |\hat{b}_k| < 2^{l+n},$$
$$|\hat{c}_k| < 2^{2l+2n},\quad |C_\nu(\ldots)| < 2^{2l+2n}.$$

Нужные нам приближенные вычисления могут быть выполнены в комплексной арифметике с фиксированной запятой, одним местом перед запятой и подходящим числом s мест после запятой с использованием вместо $\omega_n^\varkappa$ его s-разрядного приближения

$$\omega_{n,\varkappa},\quad \left|\omega_{n,\varkappa} - \omega_n^\varkappa\right| < 2^{-s} \quad \text{для } |\varkappa| < 2^{n-1}$$

и нормированных величин

$$\alpha_\nu(\ldots) \approx 2^{-l-\nu} A_\nu(\ldots),\quad \beta_\nu(\ldots) \approx 2^{-l-\nu} B_\nu(\ldots),$$
$$\gamma_\nu(\ldots) \approx 2^{-2l-2n} C_\nu(\ldots).$$

Из (2.5), (3.6) и (2.7) ясно, как, принимая во внимание эту нормировку, следует производить расчет α_ν, β_ν, γ_ν. Необходимые при этом умножения комплексных чисел будут реализованы при помощи четырех умножений действительных s-значных чисел по способу V_m и дополнительного сложения с округлением s-й цифры после запятой. Если выбрать

$$s \geqslant 2l + 2n + \lg n + \text{const},$$

то оценка ошибки округления будет такова:

$$\left|2^{2l+2n}\gamma_0(j_0, \ldots, j_{n-1}) - c_j\right| \leqslant \text{const} \cdot n \cdot 2^{2l+2n-s} < 1/2. \quad (3.9)$$

Оценку $M_{m+1}(N)$ сложности способа V_{m+1} можно теперь получить, используя значение $M_m(s)$ и употребляя оценку $O(s)$ для сложения s-разрядных чисел. В логической сети числам $\omega_{n,\varkappa}$ отвечают специальные блоки. При использовании многоленточной машины Тьюринга они предварительно вычисляются. Как показывают формулы

$$w_1 = -1,\quad w_2 = i,\quad w_{\nu+1} = \frac{1 + w_\nu}{|1 + w_\nu|} \quad \text{для } \nu \geqslant 2,$$
$$w_{\nu+1}^{2\varkappa} = w_\nu^\varkappa,\quad w_{\nu+1}^{2\varkappa+1} = w_{\nu+1} \cdot w_\nu^\varkappa,$$

это можно сделать посредством $O(2^n)$ операций сложения, умножения, деления, извлечения корня, и это потребует при их элементарном вычислении (с точностью до s знаков) самое большее $O(2^n \cdot s^2)$ шагов.

Выполнение шагов (3.5) и (3.7) при описанных реализациях происходит со сложностью $O(2^n \cdot n \cdot (M_m(s) + s))$, сложность шага (3.6) имеет величину $O(2^n (M_m(s) + s))$ и, наконец, (3.8) вносит еще $O(2^n (2l + n))$. Таким образом,

$$M_{m+1}(N) = O(2^n (s^2 + nM_m(s) + ns + 2l + n)).$$

После выбора наименьшего s, удовлетворяющего условию (3.9),

$$M_{m+1}(N) = O(2^n ((l + n)^2 + nM_m(3(l + n)))). \tag{3.10}$$

Для школьного метода V_0 имеем $M_0(s) = O(s^2)$, поэтому

$$M_1(N) = O(2^n \cdot n (l + n)^2).$$

Если выбрать, не нарушая (3.1),

$$l = [\log_2 N], \quad n = [\log_2 (2N/l)],$$

то для способа V_1 получаем сложность

$$M_1(N) = O(N (\lg N)^2).$$

Теперь используем этот результат в (3.10) **для $m = 1$; это** приводит при том же выборе l и n к оценке

$$M_2(N) = O(N \lg N (\lg \lg N)^2).$$

Далее можно найти, что

$$M_3(N) = O(N \lg N \lg \lg N (\lg \lg \lg N)^2),$$

и т. д. Мы не станем искать наилучшую степень итерации m в зависимости от N, так как в следующем разделе описывается еще более быстрый способ.

4. ПРИМЕНЕНИЕ ЧИСЕЛ ФЕРМА

Вместо комплексных чисел мы используем **теперь кольцо** классов вычетов $\mathbf{Z}_{F_n}$ по модулю чисел Ферма

$$F_n = 2^{2^n} + 1.$$

Умножение в кольце $\mathbf{Z}_{F_n}$ точно воспроизводит умножение N-разрядных двоичных чисел a, $b \in \mathbf{Z}$ $(a, b \geqslant 0)$, если

$$2N \leqslant 2^n, \tag{4.1}$$

т. е. $c = ab$ однозначно определяется условием

$$c \equiv ab \pmod{F_n} \quad \text{и} \quad 0 \leqslant c < F_n, \tag{4.2}$$

и мы можем ограничиться рассмотрением **умножения в этом** кольце вычетов.

Элементы $\mathbf{Z}_{F_n}$ удобно представлять двоичными числами длины 2^{n+1}, т. е. целыми числами x вида

$$x = \sum_{j=0}^{2^{n+1}-1} x_j 2^j \quad (x_j = 0 \text{ или } x_j = 1). \tag{4.3}$$

Это представление, правда, не однозначно, но обладает другими преимуществами, проистекающими из сравнения

$$2^{2^{n+1}} \equiv 1 \pmod{F_n}. \tag{4.4}$$

Так как $2^{2^n} \equiv -1 \pmod{F_n}$, то кольцо $R = \mathbf{Z}_{F_n}$ удовлетворяет условиям разд. 2 с $n+1$ вместо n и $w_{n+1} = 2$ или с $w_n = 4$. Необходимые в преобразовании Фурье в $\mathbf{Z}_{F_n}$ умножения на $w_{n+1}^{\varkappa} = 2^{\varkappa}$ легко реализовать в силу (4.4) циклическим сдвигом на $\varkappa$ мест:

$$2^{\varkappa} \cdot x = \sum_{j=0}^{2^{n+1}-1} x_j 2^{j+\varkappa} \equiv \sum_{k=0}^{2^{n+1}-1} y_k 2^k \pmod{F_n},$$

где $x_j = y_k$ для $j + \varkappa \equiv k \pmod{2^{n+1}}$.

Сложение по модулю F_n чисел вида (4.3) происходит с циклическим переносом, т. е. перенос в 2^{n+1}-й разряд отмечается прибавлением 1 в нулевом разряде. В силу соотношения $2^{2^n} \equiv -1$ вычитание может быть сведено к сложению с циклическим сдвигом вычитаемого на 2^n мест. Эти операции требуют сложности, не большей $O(2^n)$. Кроме того, в дальнейшем нам понадобится решение следующей вспомогательной задачи: если

$$x = u + v \cdot 2^{2^n}, \quad 0 \leqslant u < 2^{2^n}, \quad 0 \leqslant v < 2^{2^n}, \tag{4.5}$$

то наименьший неотрицательный остаток

$$\xi \equiv x \pmod{F_n}, \quad 0 \leqslant \xi \leqslant 2^{2^n},$$

можно вычислить со сложностью $O(2^n)$ по формуле

$$\xi = \begin{cases} u - v, & \text{если } u \geqslant v, \\ 2^{2^n} + 1 + u - v, & \text{если } u < v. \end{cases} \tag{4.6}$$

Вычисление по заданным ξ и η числа z, однозначно определяемого соотношением

$$z \equiv \begin{cases} \xi \pmod{F_n}, & 0 \leqslant \xi \leqslant 2^{2^n}, \\ \eta \pmod{2^{n+2}}, & 0 \leqslant \eta < 2^{n+2}, \end{cases} \tag{4.7}$$
$$0 \leqslant z < 2^{n+2} F_n,$$

также реализуется со сложностью $O(2^n)$, если сначала найти $\delta \equiv \eta - \xi \pmod{2^{n+2}}$, где $0 \leqslant \delta < 2^{n+2}$, и затем вычислить $z = \xi + \delta(2^{2^n} + 1)$.

После этих приготовлений мы опишем метод, который сводит умножение в $\mathbf{Z}_{F_m}$ к умножению в $\mathbf{Z}_{F_n}$, различая при этом случаи

$$m = 2n - 1 \quad \text{и} \quad m = 2n - 2. \tag{4.8}$$

Заметим, что $n < m$, если $m \geqslant 3$.

Вначале рассмотрим случай $m = 2n - 1$.

Подлежащие умножению элементы из $\mathbf{Z}_{F_m}$ даны в форме (4.3) как 2^{m+1}-разрядные числа a и b. Если разбить их на 2^{n+1} блоков длины 2^{n-1} каждый:

$$a = \sum_{\rho=0}^{2^{n+1}-1} a_\rho 2^{\rho \cdot 2^{n-1}}, \quad b = \sum_{\sigma=0}^{2^{n+1}-1} b_\sigma 2^{\sigma \cdot 2^{n-1}}, \quad 0 \leqslant a_\rho,\ b_\sigma < 2^{2^{n-1}}, \tag{4.9}$$

то произведение запишется в виде

$$ab \equiv \sum_{\tau=0}^{2^{n+1}-1} c_\tau 2^{\tau \cdot 2^{n-1}} \pmod{F_m}, \tag{4.10}$$

где

$$c_\tau = \sum_{\substack{\rho+\sigma \equiv \tau \pmod{2^{n+1}} \\ 0 \leqslant \rho,\ \sigma < 2^{n+1}}} a_\rho b_\sigma < 2^{n+1+2^n}.$$

Поскольку

$$2^{2^n \cdot 2^{n-1}} = 2^{2^m} \equiv -1 \pmod{F_m},$$

произведение можно записать и в такой форме:

$$ab \equiv \sum_{j=0}^{2^n-1} \left(c_j - c_{j+2^n} + 2^{n+1+2^n}\right) 2^{j \cdot 2^{n-1}} + \sum_{j=2^n}^{2^{n+1}-1} 2^{n+1+2^n} 2^{j \cdot 2^{n-1}} \pmod{F_m}.$$

Если положить

$$z_j = \begin{cases} c_j - c_{j+2^n} + 2^{n+1+2^n}, & 0 \leqslant j < 2^n, \\ 2^{n+1+2^n}, & 2^n \leqslant j < 2^{n+1}, \end{cases} \tag{4.11}$$

то

$$ab \equiv \sum_{j=0}^{2^{n+1}-1} z_j \cdot 2^{j \cdot 2^{n-1}} \pmod{F_m}. \tag{4.12}$$

В (4.11) слагаемое $2^{n+1}+2^n$ добавлено для того, чтобы обеспечить выполнение неравенства $0 \leqslant z_j < 2^{n+2}F_n$ (см. 4.10).

Так как 2^{n+2} и F_n взаимно просты, достаточно вычислить z_j по модулю 2^{n+2} и по модулю F_n. Вычисление z_j по модулю 2^{n+2} проводится простым образом при помощи следующего искусственного приема: по числам α_j, β_j, определяемым из сравнений

$$\alpha_j \equiv a_j, \quad \beta_j \equiv b_j \pmod{2^{n+2}}, \quad 0 \leqslant \alpha_j, \ \beta_j < 2^{n+2},$$

образуем числа

$$u = \sum_{\rho=0}^{2^{n+1}-1} \alpha_\rho 2^{\rho(3n+5)}, \quad v = \sum_{\sigma=0}^{2^{n+1}-1} \beta_\sigma 2^{\sigma(3n+5)}, \quad u, v < 2^{2^{(n+1)}(3n+5)}. \tag{4.13}$$

Их произведение содержит как непересекающиеся части длины $3n+5$ суммы

$$\gamma_\tau = \sum_{\rho+\sigma=\tau} \alpha_\rho \beta_\sigma < 2^{n+1}(2^{n+2})^2, \quad 0 \leqslant \tau < 2^{n+2}.$$

В силу (4.10) для $0 \leqslant \tau < 2^{n+1}$ справедливо сравнение

$$c_\tau \equiv \gamma_\tau + \gamma_{\tau+2^{n+1}} \pmod{2^{n+2}},$$

а в силу (4.11) — сравнение

$$z_j \equiv \eta_j \pmod{2^{n+2}} \quad \text{для} \quad 0 \leqslant j < 2^n, \tag{4.14}$$

где

$$\eta_j \equiv \gamma_j - \gamma_{j+2^n} + \gamma_{j+2\cdot 2^n} - \gamma_{j+3\cdot 2^n} \pmod{2^{n+2}} \quad \text{и} \quad 0 \leqslant \eta_j < 2^{n+2}.$$

Вычисление η_j может быть произведено со сложностью, не большей $O(2^{2n})$, а сложность умножения $u \cdot v$ — чисел длины, не превышающей $2^{n+1}(3n+5)$, может быть оценена при помощи (3.11):

$$M_1(2^{n+1}(3n+5)) = O(2^n \cdot n^3) \leqslant O(2^{2n})$$

(мы используем здесь (3.11) лишь для удобства; можно употреблять оценку Карацубы $O(N^{2-\varepsilon})$; см. [4, 5].

Вычисление $z_j \bmod F_n$ сводится к умножению в $\mathbf{Z}_{F_n}$ с помощью преобразования Фурье в $\mathbf{Z}_{F_n}$ с $w_{n+1} = 2$; для этого нужно выполнить шаги (3.5)—(3.7). Важно то, что в нашем случае, принимая во внимание (4.11), вместо c_j достаточно вычислить только 2^n разностей $c_j - c_{j+2^n}$ по модулю F_n. Согласно замечанию в конце разд. 2 для вычисления величин

$$C_1(1, j_1, \ldots, j_n) \equiv c_j - c_{j+2^n} \pmod{F_n}, \quad j = j_1 \cdot 2^{n-1} + \ldots + j_n \cdot 2^0,$$

аналогичных величинам (2.8), требуется, в отличие от (3.6), только 2^n умножений

$$\hat{a}_k \hat{b}_k \equiv \hat{c}_k \pmod{F_n} \quad \text{для нечетных } k < 2^{n+1}.$$

Преобразование Фурье в $\mathbf{Z}_{F_n}$ состоит из $O(n \cdot 2^n)$ шагов сложностью $O(2^n)$ каждый, т. е. имеет общую сложность $O(n \cdot 2^{2n})$. Прибавление числа 2^{n+1+2^n} и приведение по модулю F_n, согласно (4.6), осуществляется со сложностью $O(2^{2n})$ и дает для $0 \leqslant j < 2^n$ остатки

$$\xi_j \equiv C_1(1, j_1, \ldots, j_n) + 2^{n+1+2^n} \equiv z_j \pmod{F_n}, \quad 0 \leqslant \xi_j \leqslant 2^{2^n}.$$

По этим остаткам и остаткам (4.14), решая вспомогательную задачу (4.7), находим 2^n значений z_j со сложностью $O(2^{2n})$. Наконец, сложение z_j (см. формулу (4.2)) осуществляется тоже со сложностью $O(2^{2n})$.

Тем самым мы показали, что умножение в $\mathbf{Z}_{F_{2n-1}}$ реализуется 2^n умножениями в $\mathbf{Z}_{F_n}$ с дополнительной сложностью $O(n \cdot 2^{2n})$.

Случай четного $m = 2n-2$ рассматривается аналогично: сомножители a и b разбиваются в 2^n блоков длины 2^{n-1}. Их свертка сводится преобразованием Фурье в $\mathbf{Z}_{F_n}$ — теперь с $w_n = 4$ — к умножению в $\mathbf{Z}_{F_n}$. Снова нужно лишь умножение для нечетных индексов k, число которых в этом случае 2^{n-1}. Поэтому умножение в $\mathbf{Z}_{F_{2n-2}}$ реализуется 2^{n-1} умножениями в $\mathbf{Z}_{F_n}$ с дополнительной сложностью $O(n \cdot 2^{2n})$.

Перейдем к окончательной оценке сложности нашего способа. Обозначим $M(n)$ наименьшую сложность логической сети для умножения в $\mathbf{Z}_{F_n}$ (используя представление чисел в виде (4.3)); тогда из вышесказанного при достаточно большом γ_0 для $n \geqslant 3$ справедливы неравенства

$$\left.\begin{aligned} M(2n-2) &\leqslant 2^{n-1} M(n) + \gamma_0 (n-1)\, 2^{2n-1}, \\ M(2n-1) &\leqslant 2^n M(n) + \gamma_0 (n-1)\, 2^{2n}. \end{aligned}\right\} \tag{4.15}$$

С другой стороны эти неравенства можно интерпретировать как временну́ю оценку надлежаще организованной многоленточной машины Тьюринга, которая работает по описанному рекуррентному методу; тогда $M(n)$ будет обозначать максимально необходимое число шагов, требующееся для умножения в $\mathbf{Z}_{F_n}$.

При

$$\gamma = \max\{M(1),\ M(2),\ M(3),\ \gamma_0\}$$

из (4.15) индукцией по k получается оценка

$$M(n) \leqslant \gamma k \cdot 2^{k+n} \quad \text{при} \quad n \leqslant 2^k + 1,$$

следовательно,

$$M(n) = O(2^n n \lg n).$$

Для умножения N-разрядных двоичных чисел выбираем n из условия (4.1):

$$n = [\log 2N].$$

Необходимое для умножения в $\mathbf{Z}_{F_n}$ приведение по модулю F_n осуществляется при помощи решения вспомогательной задачи (4.6) со сложностью $O(2^n) = O(N)$. Итак, умножение N-разрядных чисел может производиться со сложностью $O(N \lg N \lg\lg N)$.

СПИСОК ЛИТЕРАТУРЫ

1. Cook S. A., On the minimum computation time of functions, Diss., Harvard Univ., 1966.
2. Cook S. A., Aanderaa S. O., On the minimum computation time of functions, *Trans. AMS*, **142** (1969), 291—314. (Русский перевод: Кук С. А., Андераа С. О., О минимальном времени вычисления функций, Кибернетический сборник, нов. серия, вып. 8, «Мир», М., 1971.)
3. Cooley J. W., Tukey J. W., An algorithm for the machine calculation of complex Fourier series, *Math. Comp.*, **19** (1965), 297—301.
4. Карацуба А. А., Офман Ю. П., Умножение многозначных чисел на автоматах, *ДАН СССР*, **145** (1962), 293—294.
5. Knuth D. E., The art of computer programming, vol. 2, ch. 4, Addison-Wesley, 1969.
6. Schönhage A., Multiplikation großer Zahlen, *Computing*, **1** (1966), 182—196.
7. Тоом А. Л., О сложности схемы из функциональных элементов, реализующей умножение целых чисел, *ДАН СССР*, **150** (1963), 496—498.

Длины формул и исключение кванторов[1)]

Л. Ходес, Е. Шпекер

ВВЕДЕНИЕ

Вопросы, исследуемые в этой статье, представляют собой частные случаи проблемы следующего типа: дана функция, сколь длинной должна быть формула, представляющая ее.

Здесь рассматривается случай пропозиционального исчисления. Для того чтобы сформулировать результаты, вводятся следующие обозначения:

F_1 есть множество формул пропозиционального исчисления первого порядка, связками в котором являются отрицание и конъюнкция.

Пример: $\neg(x_1 \wedge \neg x_2) \wedge \neg(\neg x_1 \wedge x_2)$.

F_2 есть множество формул пропозиционального исчисления первого порядка, связками в котором являются отрицание, конъюнкция и биимпликация (т. е. эквивалентность).

Пример: $(x_1 \leftrightarrow (x_2 \wedge \neg x_3)) \leftrightarrow (\neg x_1 \wedge x_2)$.

F_3 есть множество формул пропозиционального исчисления второго порядка, связками в котором являются отрицание, конъюнкция и биимпликация.

Пример: $(\forall x_1)((\exists x_2)(x_2 \wedge x_3) \leftrightarrow (\forall x_4)((x_1 \wedge x_2) \leftrightarrow (x_3 \wedge \neg x_4)))$.

Ясно, что $F_1 \subseteq F_2 \subseteq F_3$. Более того, для каждой формулы φ из F_3 существует формула ψ из F_1, такая, что ψ эквивалентна φ. Такая формула ψ обычно должна быть намного длиннее, чем данная формула φ.

И действительно, определяя длину формулы как суммарное количество вхождений в нее всех переменных (так что формулы, указанные в качестве примеров, имеют длины 4, 5, 6 соответственно), для $i = 1, 2$ верна

Теорема (i). *Для каждого целого положительного числа c существует формула φ из F_{i+1}, такая, что для каждой*

[1)] Hodes L., Specker E., Lengths of formulas and elimination of quantifiers, Contributions to mathematical logic, Proceedings of the logic colloquium, Hannover, 1966.

формулы ψ *из* F_i, *эквивалентной* φ, *имеет место следующее неравенство*:

$$\text{длина } \psi \geqslant c \cdot \text{длина } \varphi.$$

Доказательства этих теорем удобнее проводить на языке колец, а не на языке решеток. Введем с этой целью булевскую сумму, произведение (это то же самое, что и конъюнкция) и булевские константы 0, 1. В этом случае можно обойтись без отрицания, так как $1 + x_1$ есть то же самое, что и $\neg x_1$.

О результатах этой статьи и следствиях из нее было сообщено в [1], [2]. Впервые точный пример булевской функции, допускающей только «нелинейные» реализации, был предложен, насколько нам известно, Нечипоруком [3].

1. Определение понятия «формула»: 0, 1, x_0, x_1, ... являются формулами. Если φ, ψ — формулы, то формулами также являются $\varphi + \psi$ и $\varphi \cdot \psi$. (Скобки вводятся обычным образом.)

2. Определение понятия «p-формула» («p» обозначает «произведение»): 0, 1, x_0, x_1, ... являются p-формулами. Если φ есть p-формула, то p-формулами являются $0 + \varphi$ и $1 + \varphi$. Если φ, ψ есть p-формулы, то p-формулой является и $\varphi \cdot \psi$.

Замечание. $\varphi + \psi$ эквивалентна

$$1 + (1 + \varphi \cdot (1 + \psi)) \cdot (1 + (1 + \varphi) \cdot \psi),$$

которая является p-формулой, если φ и ψ — p-формулы. Каждая формула, следовательно, эквивалентна некоторой p-формуле.

3. Сокращенные обозначения. Если φ — формула, то $\varphi \Big| \begin{matrix} x_{i_1} \dots x_{i_m} \\ 0 \dots 0 \end{matrix}$ есть формула, полученная из φ подстановкой 0 вместо переменных $x_{i_1}, \dots, x_{i_m}$.

Если в φ не встречаются никакие другие переменные, кроме $x_{i_1}, \dots, x_{i_m}, x_{j_1}, \dots, x_{j_n}$, и если $i_p \neq j_q$ для всех p, q ($1 \leqslant p \leqslant m$, $1 \leqslant q \leqslant n$), то $\varphi/x_{j_1} \dots x_{j_n}$ — эта формула $\varphi \Big| \begin{matrix} x_{i_1} \dots x_{i_m} \\ 0 \dots 0 \end{matrix}$.

Если $\boldsymbol{x}$ есть последовательность $\langle x_{j_1}, \dots, x_{j_n} \rangle$, то $\varphi/\boldsymbol{x}$ есть формула $\varphi/x_{j_1} \dots x_{j_n}$.

Пример. Если φ — формула $(x_1 + x_2) \cdot (x_2 + x_3)$ и $\boldsymbol{x}$ — последовательность $\langle x_1, x_3 \rangle$, то φ/x — это $(x_1 + 0) \cdot (0 + x_3)$.

Теоремы этой статьи основываются на следующей основной лемме.

Основная лемма. *Для всех положительных целых чисел* m, k *существует положительное целое число* n_0, *такое, что для всех* $n (n \geqslant n_0)$ *имеет место следующее.*

Если φ *— формула, содержащая* n *переменных* $x_1, \ldots, x_n$, *ни одна из которых не входит в* φ *более чем* k *раз, то существует* m *различных целых положительных чисел* $k_1, \ldots, k_m$ $(1 \leqslant k_j \leqslant n,\ 1 \leqslant j \leqslant m)$ *и булевские константы* c_0, c_1, c_2, *такие, что* $\varphi/x_{k_1} \ldots x_{k_m}$ *эквивалентна*

$$c_0 + c_1 \prod_{j=1}^{m} (1 + x_{k_j}) + c_2 \cdot \sum_{j=1}^{m} x_{k_j}.$$

Более того, если φ *есть* p*-формула, то* $c_2 = 0$.

Положив $\pi = \prod_j (1 + x_{k_j})$, $\sigma = \sum_j x_{k_j}$, имеем $\pi \cdot \sigma = 0$. Следовательно, лемма утверждает, что φ/x эквивалентна двоичной формуле в π, σ.

Доказательство леммы проводится с помощью построения все более и более простых формул из данной формулы путем подстановки нулей вместо некоторых переменных. Из построения будет ясно, что заключительная формула является p-формулой (т. е. $c_0 + c_1 \cdot \pi$), если таковой является исходная.

4. Формула φ называется сокращением формулы ψ тогда и только тогда, когда φ эквивалентна ψ и когда для всех $i (0 \leqslant i)$ число вхождений x_i в φ меньше или равно числу вхождений x_i в ψ.

Пусть S — множество переменных и φ — формула, содержащая по крайней мере две различные переменные. Тогда существуют формулы φ_1, φ_2 и булевская константа c, такие, что имеют место следующие условия:

(1) $\varphi_1 + \varphi_2$ или $c + \varphi_1 \cdot \varphi_2$ есть сокращение φ;

(2) и φ_1 и φ_2 содержат по крайней мере по одной переменной;

(3) число различных переменных из S, содержащихся в φ_2, составляет по крайней мере половину числа различных переменных из S, содержащихся в φ.

Замечание. Мы будем обозначать двоичную операцию над φ_1, φ_2 одним символом $*$, $\varphi_1 * \varphi_2$. Надо иметь в виду, что различные вхождения звездочки в одной и той же формуле не обязательно относятся к одной и той же операции.

5. Последовательность формул $\langle \psi_1, \ldots, \psi_n \rangle$ называется нормальной в том и только том случае, если:

(1) каждая формула ψ_i, $1 \leqslant i \leqslant n$, содержит по крайней мере одну переменную;

(2) для каждого i, $1 \leqslant i \leqslant n$, либо ψ_i содержит только одну переменную, либо ψ_i содержит только те переменные, которые содержатся в формуле $\psi_1 \cdot \psi_2 \cdots \psi_{i-1}$.

Замечание. Пусть $\langle \psi_1, \ldots, \psi_n \rangle$ нормальна, и пусть $\langle \chi_1, \ldots, \chi_m \rangle$ — последовательность, полученная из $\langle \psi_1, \ldots, \psi_n \rangle$ подстановкой 0 вместо некоторой переменной во все формулы $\psi_1, \ldots, \psi_n$ и удалением после этого формул, не содержащих переменных. Тогда $\langle \chi_1, \ldots, \chi_m \rangle$ нормальна.

6. Формула ψ называется формулой типа τ_q^1 в том и только том случае, если существуют формулы ψ_1, ψ_2, такие, что для некоторой операции $*$ имеют место следующие условия:

(1) $\psi_1 * \psi_2$ есть сокращение ψ;

(2) существуют по крайней мере q различных переменных, входящих как в ψ_1, так и в ψ_2.

Формула ψ называется формулой типа τ_p^2 в том и только том случае, если существует нормальная последовательность $\langle \psi_1, \ldots, \psi_p \rangle$ формул, такая, что для некоторой последовательности операций $*$ формула

$$\psi_1 * (\psi_2 * (\psi_3 * (\ldots * \psi_p) \ldots))$$

является сокращением ψ.

Примером формулы такого типа является

$$1 + \psi_1 \cdot (\psi_2 + \psi_3 \cdot (1 + \psi_4 \cdot \psi_5)),$$

при этом последовательностью операций $u * v$ будет $1 + u \cdot v$, $u + v$, $u \cdot v$, $1 + u \cdot v$.

Лемма 1. *Если* $1 \leqslant p$, $1 \leqslant q$, $4^p \cdot q \leqslant n$ *и если* φ *— формула, содержащая* n *переменных, то существует последовательность* $\boldsymbol{x}$ *переменных* φ, *такая, что* $\varphi/\boldsymbol{x}$ *является формулой либо типа* τ_q^1, *либо типа* τ_p^2.

Доказательство. Предположим, что не существует последовательности $\boldsymbol{x}$ переменных φ, такой, что $\varphi/\boldsymbol{x}$ является формулой типа τ_q^1 и определим последовательности $\langle \varphi_1, \ldots, \varphi_p \rangle$ и $\langle \psi_1, \ldots, \psi_p \rangle$ формул так, чтобы выполнялись следующие условия:

(1) φ_1 есть φ;

(2) для всех i, $1 \leqslant i \leqslant p$, последовательность $\langle \psi_1, \ldots, \psi_i \rangle$ является нормальной;

(3) для всех i, $1 \leqslant i \leqslant p-1$, формула φ_i содержит по крайней мере $4^{p-i} \cdot q$ переменных, не содержащихся в $\psi_1 \cdot \psi_2 \ldots \psi_{i-1}$;

(4) для всех i, $1 \leqslant i \leqslant p-1$, существует последовательность $\boldsymbol{y}$ переменных φ, содержащая все переменные из $\psi_1 \cdot \psi_2 \ldots$

$\ldots \psi_{i-1}$, такая, что $\psi_i * \varphi_{i+1}$ есть сокращение формулы $\varphi_i/\boldsymbol{y}$ для некоторой операции $*$;

(5) существует последовательность $\hat{\boldsymbol{z}}$ переменных φ, содержащая все переменные формул $\psi_1, \psi_2, \ldots, \psi_{p-1}$, такая, что ψ_p есть $\varphi_p/\boldsymbol{z}$.

Определим φ_1 как φ и предположим, что для некоторого $i (1 \leqslant i \leqslant p-1)$ определены последовательности $\langle \varphi_1, \ldots, \varphi_i \rangle$ и $\langle \psi_1, \ldots, \psi_{i-1} \rangle$, удовлетворяющие условиям, перечисленным выше. Пусть S_i — множество переменных φ_i, не входящих в $\psi_1 \ldots \psi_{i-1}$; S_i содержит по крайней мере $4^{p-i} \cdot q$ $(\geqslant 2)$ элементов. Следовательно, существуют формулы χ_i, ω_i и операция $*$, такие, что:

(a) $\chi_i * \omega_i$ есть сокращение φ_i;

(b) χ_i содержит по крайней мере одну переменную;

(c) ω_i содержит по крайней мере $2 \cdot 4^{p-i-1} \cdot q$ переменных, не содержащихся в $\psi_1 \cdot \psi_2 \ldots \psi_{i-1}$.

Пусть $\boldsymbol{y}$ — последовательность переменных, состоящая из переменных $\psi_1 \ldots \psi_{i-1} \cdot \varphi_i$. Тогда

$$\psi_1 * (\psi_2 * \ldots (\psi_{l-1} * \varphi_i) \ldots)$$

есть сокращение $\varphi/\boldsymbol{y}$. Пусть $\boldsymbol{z}$ — последовательность переменных из $\boldsymbol{y}$, не входящих в $\psi_1 \ldots \psi_{i-1}$. Тогда при некоторых булевских константах c_1, c_2 формула $c_1 + c_2 \cdot \varphi_i/\boldsymbol{z}$ будет сокращением $\varphi/\boldsymbol{z}$.

Если $\chi_i * \omega_i$ — сокращение φ_i, то формула $\chi_i/\boldsymbol{z} * \omega_i/\boldsymbol{z}$ — сокращение $\varphi_i/\boldsymbol{z}$. Следовательно, для некоторой операции $*$ формула $\chi_i/\boldsymbol{z} * \omega_i/\boldsymbol{z}$ является сокращением $\varphi/\boldsymbol{z}$.

По предположению не существует $\boldsymbol{x}$, такого, что $\varphi/\boldsymbol{x}$ является формулой типа τ_q^1, следовательно, формулы $\chi_i/\boldsymbol{z}$ и $\omega_i/\boldsymbol{z}$ имеют меньше, чем q различных общих переменных.

Число различных переменных ω_i, не входящих в $\psi_1 \ldots \psi_{i-1}\chi_i$, не меньше, чем $2 \cdot 4^{p-i-1} \cdot q - q$, и поэтому не меньше, чем $4^{p-i-1} \cdot q$.

Если χ_i содежит переменные, входящие в $\psi_1 \ldots \psi_{i-1}$, то обозначим через $\boldsymbol{u}$ последовательность, содержащую в точности следующие переменные:

все переменные, входящие в $\psi_1, \ldots, \psi_{i-1}$; все переменные ω_i, не входящие в χ_i.

Если определить ψ_i, как $\chi_i/\boldsymbol{u}$, а ψ_{i+1}, как $\omega_i/\boldsymbol{u}$, то видно, что условия (1)—(5) выполняются.

Если же χ_i не содержит переменных $\psi_i \ldots \psi_{l-1}$, тогда пусть $\boldsymbol{u}$ — последовательность, содержащая в точности следующие переменные:

все переменные, входящие в $\psi_1 \ldots \psi_{i-1}$; точно одну переменную из χ_i; все переменные формулы ω_i, не входящие в $\psi_1 \ldots \psi_{i-1} \cdot \chi_i$. Снова определив ψ_i, как $\chi_i/\boldsymbol{u}$, и φ_{i+1}, как $\omega_i/\boldsymbol{u}$, мы видим, что условия (1)—(5) выполняются.

Предположим, что последовательности $\langle \varphi_1, \ldots, \varphi_p \rangle$, $\langle \psi_1, \ldots, \psi_{p-1} \rangle$ определены; φ_p содержит по крайней мере q переменных. Если φ_p содержит переменные, входящие в $\psi_1 \ldots \psi_{p-1}$, то пусть x — последовательность, содержащая переменные из $\psi_1 \ldots \psi_{p-1}$. Если φ_p не содержит переменных из $\psi_1, \ldots, \psi_{p-1}$, то пусть x — последовательность, содержащая все переменные $\psi_1 \ldots \psi_{p-1}$ и в точности одну переменную φ_p. Определим ψ_p, как φ_p/x, в обоих случаях. Условия (1)—(5) выполнены; формула

$$\psi_1 * (\psi_2 * \ldots (\psi_{p-1} * \psi_p) \ldots)$$

является сокращением φ/x, т. е. φ/x является формулой типа τ_p^2.

7. Лемма 2. *Если последовательность $\langle \psi_1, \ldots, \psi_n \rangle$ нормальная, если никакая переменная не входит более чем в k формул из набора $\psi_1, \ldots, \psi_n$ и если $n \geqslant (k+1)^m$, то существует целое положительное число p и различные целые положительные числа $k_1, \ldots, k_p$, такие, что имеет место следующее:*

(1) *x_{k_i} содержится в ψ_i;*

(2) *если $\langle \chi_1, \ldots, \chi_q \rangle$ — последовательность, получающаяся из $\langle \psi_1, \ldots, \psi_n \rangle$ подстановкой 0 вместо всех переменных, кроме $x_{k_1}, \ldots, x_{k_p}$, и вычеркиванием после этого формул, не содержащих переменных, то:*

(a) *$m \leqslant q$;*

(b) *для всех j, $1 \leqslant j \leqslant m$, формула χ_j содержит в точности одну переменную, скажем x_{i_j};*

(c) *последовательность $\langle i_1, \ldots, i_m \rangle$ (определенная в (b)) является последовательностью без чередований.*

(Говорят, что последовательность $\langle i_1, \ldots, i_m \rangle$ является последовательностью без чередований в том и только том случае, если для всех j_1, j_2, j_3 выполняется следующее условие: если $1 \leqslant j_1 < j_2 < j_3 \leqslant m$ и $i_{j_1} = i_{j_3}$, то $i_{j_1} = i_{j_2}$. Последовательность $\langle 9, 9, 7, 8, 8, 8, 1 \rangle$ является последовательностью без чередований, а последовательность $\langle 9, 9, 7, 8, 8, 9, 1 \rangle$ таковой не является.)

Доказательство. Пусть x_{k_1} — переменная, входящая в ψ_1. Если $m = 1$, то $p = 1$ и k_1 удовлетворяет требованиям леммы. Для того чтобы выполнить индукционный шаг, надо рассмотреть два случая.

(a) x_{k_1} не входит ни в одну из формул $\psi_j (2 \leqslant j \leqslant (k+1)^{m-1}+1)$. Если положить $r = (k+1)^{m-1}+1$, то последовательность $\langle \psi_2, \ldots, \psi_r \rangle$ будет нормальной и каждая переменная будет входить не более чем в k формул $\psi_j (2 \leqslant j \leqslant r)$; тогда если $k_2, \ldots, k_p$ — индексы, полученные в соответствии с индукцион-

ной гипотезой, то последовательность $\langle k_1, \ldots, k_p \rangle$ удовлетворяет требованиям леммы.

(b) Предположим, что x_{k_1} входит в некоторые формулы ψ_j $(2 \leqslant j \leqslant (k+1)^{m-1}+1)$, и пусть h — наименьшее из таких чисел j. Кроме того, пусть V — множество переменных, отличных от x_{k_1} и содержащихся в $\psi_2 \ldots \psi_{h-1}$. Подставим 0 вместо всех переменных из V в формулы $\psi_1, \psi_h, \psi_{h+1}, \ldots, \psi_n$ и вычеркнем формулы, не содержащие переменных. Пусть в результате этого получилась последовательность $\langle \omega_1, \ldots, \omega_r \rangle$. Длина последовательности $\langle \psi_1, \psi_h, \psi_{h+1}, \ldots, \psi_n \rangle$ есть по крайней мере $n-(k+1)^{m-1}+1$.

В $\psi_2 \ldots \psi_{h-1}$ содержится меньше, чем $(k+1)^{m-1}$ различных переменных, каждая из которых входит самое большее в $(k-1)$ формулу из $\psi_1, \psi_h, \ldots, \psi_n$. Следовательно,

$$r \geqslant n-(k+1)^{m-1}+1-(k-1)(k+1)^{m-1}, \text{ т. е. } r \geqslant (k+1)^{m-1}+1.$$

Последовательность $\langle \omega_2, \ldots, \omega_r \rangle$ нормальная, и ее длина не меньше $(k+1)^{m-1}$; x_{k_1} входит в ω_2. Если $\langle k_1, \ldots, k_p \rangle$ — последовательность индексов, полученная согласно индукционной гипотезе для $\langle \omega_2, \ldots, \omega_r \rangle$, то та же самая последовательность удовлетворяет требованиям леммы для $\langle \psi_1, \ldots, \psi_n \rangle$.

8. Лемма 3. *Если φ — формула типа τ_p^2, если никакая переменная не входит в φ более k раз и если $p \geqslant (k+1)^{k \cdot m}$, то существуют m различных переменных $x_{k_1}, \ldots, x_{k_m}$, входящих в φ, таких, что $\varphi/x_{k_1} \ldots x_{k_m}$ эквивалентна некоторой формуле ω, удовлетворяющей следующим условиям:*

Существует последовательность $\langle \omega_0, \ldots, \omega_m \rangle$ формул, такая, что:

(1) *никакая переменная не входит в ω_0 более чем $k-1$ раз;*

(2) *для всех j, $1 \leqslant j \leqslant m$, существуют булевские константы a_j, b_j, c_j, такие, что ω_j эквивалентна*

$$a_j + (b_j + c_j \cdot x_{k_j}) * \omega_{j-1}$$

для некоторой операции $*$;

(3) *ω — это формула ω_m.*

Доказательство. Так как φ является формулой типа τ_p^2, то существует нормальная последовательность $\langle \psi_1, \ldots, \psi_p \rangle$, такая, что некоторая формула

$$\psi_1 * (\psi_2 * \ldots * \psi_p)$$

является сокращением формулы φ. Каждая переменная входит самое большее в k формул из $\psi_1, \ldots, \psi_p$. По лемме 2

существует последовательность $\boldsymbol{x}$ переменных φ, такая, что последовательность

$$\langle \chi_1, \ldots, \chi_q \rangle,$$

полученная из последовательности

$$\langle \psi_1/\boldsymbol{x}, \ldots, \psi_p/\boldsymbol{x} \rangle$$

вычеркиванием формул, не содержащих переменных, обладает следующими свойствами:

(1) $k \cdot m \leqslant q$;

(2) для всех j, $1 \leqslant j \leqslant k \cdot m$, формула χ_j содержит в точности одну переменную, скажем x_{i_j};

(3) последовательность $\langle i_1, \ldots, i_m \rangle$ является последовательностью без чередований.

Каждая переменная содержится самое большее в k формулах из χ_j, $1 \leqslant j \leqslant q$. Следовательно, последовательность $\langle i_1, \ldots, i_{k \cdot m} \rangle$ содержит по крайней мере m различных чисел; пусть $k_1, \ldots, k_m$ — такие числа, и положим $\boldsymbol{y} = \langle x_{k_1}, \ldots, x_{k_m} \rangle$. Тогда существует последовательность $\langle r_1, \ldots, r_{m+1} \rangle$, удовлетворяющая следующим условиям:

(a) $1 = r_{m+1} < r_m < \ldots < r_1 \leqslant p$;

(b) для всех j, $1 \leqslant j \leqslant m$, формула $\prod\limits_{r_{j+1} \leqslant i < r_j} \psi_i$ содержит ровно одну переменную из $\boldsymbol{y}$;

(c) для всех h, j $(1 \leqslant h < j \leqslant m)$ переменные из $\boldsymbol{y}$, входящие в $\prod\limits_{r_{h+1} \leqslant i < r_h} \psi_i$ и $\prod\limits_{r_{j+1} \leqslant i < r_j} \psi_i$, различны.

Можно предположить, что переменная из $\boldsymbol{y}$, входящая в $\prod\limits_{r_{j+1} \leqslant i < r_j} \psi_i$, есть x_{k_j}, $1 \leqslant j \leqslant m$.

Определяя ω_0 как

$$\psi_{r_1}/\boldsymbol{y} * \ldots * \psi_p/\boldsymbol{y}$$

и для всех j, $1 \leqslant j \leqslant m$, определяя константы a_j, b_j, c_j и формулы ω_j так, что

$$\psi_{r_{j+1}} * \left(\psi_{r_{j+1}+1} * \ldots * \left(\psi_{r_j - 1} * \omega_{j-1}\right) \ldots\right)/\boldsymbol{y}$$

эквивалентна

$$a_j + (b_j + c_j x_{k_j}) * \omega_{j-1}$$

для некоторой операции $*$, мы убеждаемся в том, что условия леммы выполнены.

9. Определение понятия «базисная формула»: формула φ является базисной в $\langle x_{i_0}, x_{i_1}, \ldots, x_{i_n}\rangle$ типа $\langle \alpha_1, \ldots, \alpha_n\rangle$ в том и только том случае, если выполняются следующие условия:

(1) $\langle x_{i_0}, x_{i_1}, \ldots, x_{i_n}\rangle$ есть последовательность отличных друг от друга переменных;

(2) $\langle \alpha_1, \ldots, \alpha_n\rangle$ есть последовательность операций α_i, $1 \leqslant i \leqslant n$, каждая из которых является либо булевским произведением, либо булевской суммой;

(3) существует последовательность формул $\langle \varphi_0, \ldots, \varphi_n\rangle$ и три последовательности констант $\langle a_1, \ldots, a_n\rangle$, $\langle b_0, \ldots, b_n\rangle$ и $\langle c_0, \ldots, c_n\rangle$, такие, что φ_0 есть $b_0 + c_0 x_{i_0}$;

для всех k $(0 \leqslant k \leqslant n-1)$ φ_{k+1} есть $a_{k+1} + (b_{k+1} + c_{k+1} x_{i_{k+1}})\,\varphi_k$, если α_{k+1} — произведение, или $\left(b_{k+1} + c_{k+1} x_{i_{k+1}}\right) + \varphi_k$, если α_{k+1} — сумма;

φ_n есть φ.

Формула φ, базисная в $\langle x_{i_0}, \ldots, x_{i_n}\rangle$, имеет вид

$$\varphi_n * (\varphi_{n-1} * \ldots * (\varphi_1 * \varphi_0) \ldots),$$

где φ_j содержит переменную x_{i_j} и не содержит никаких других переменных.

Если φ является базисной в $\langle x_{i_0}, \ldots, x_{i_n}\rangle$ типа $\langle \alpha_1, \ldots, \alpha_n\rangle$ и $1 \leqslant i \leqslant n$, то $\varphi\Big|_0^{x_{i_j}}$ эквивалентна формуле, базисной в $\langle x_{i_0}, \ldots, x_{i_n}\rangle$ типа $\langle \alpha_1, \ldots, \alpha_n\rangle$.

Лемма 4. *Если φ является базисной в $\langle x_{i_0}, \ldots, x_{i_n}\rangle$ типа $\langle \alpha_1, \ldots, \alpha_n\rangle$ и $n \geqslant 6m$, $m \geqslant 1$, то существуют m различных чисел $k_1, \ldots, k_m$ среди $i_1, \ldots, i_n$ и булевские константы d_0, d_1, d_2, такие, что формула $\varphi/x_{i_0} x_{k_1} \ldots x_{k_m}$ эквивалентна либо*

$$d_0 + (d_1 + d_2 x_{i_0}) \cdot \prod_{j=1}^{m} (1 + x_{k_j}), \text{ либо } d_0 + d_1 x_{i_0} + d_2 \sum_{j=1}^{m} x_{k_j}.$$

Доказательство. Положим $i_j = j$, $0 \leqslant j \leqslant n$, и $m \geqslant 2$. Кроме того, пусть последовательности формул и булевских констант будут такими, как указано в определении базисной формулы. Пусть n_1 — число операций суммирования среди $\alpha_1, \ldots, \alpha_n$, а n_2 — число операций умножения; $n_1 + n_2 = n$. Если $n_1 \geqslant 2m$, то пусть $i_1, \ldots, i_{2m}$ — $2m$ различных чисел, таких, что α_{i_j}, $1 \leqslant j \leqslant 2m$, являются суммами. Тогда формула $\varphi/x_0 x_{i_1} \ldots x_{i_{2m}}$ эквивалентна некоторой формуле

$$d_0 + d_1 x_0 + \sum_{j=1}^{2m} e_j x_{i_j}.$$

Существуют m различных чисел j и булевская константа d_2, такие, что $e_j = d_2$. Следовательно, существуют различные $k_1, \ldots, k_m$, такие, что $\varphi/x_0x_{k_1} \ldots x_{k_m}$ эквивалентна

$$d_0 + d_1x_0 + d_2 \sum_{j=1}^{m} x_{k_j}.$$

Предположим, что $n_1 < 2m$; тогда α_k есть операция умножения по крайней мере для $4m$ различных индексов k. Если существуют m различных чисел $k_1, \ldots, k_m$ среди $1, \ldots, n$, таких, что $c_{k_j} = 0$, $1 \leqslant j \leqslant m$, то формула $\varphi/x_0x_{k_1} \ldots x_{k_m}$ эквивалентна $d_0 + d_1x_0$ для некоторых констант d_0, d_1. Предположим противное, т. е. что существуют $3m$ различных индексов k, таких, что α_k есть операция умножения и $c_k = 1$. Положив x равной последовательности из соответствующих $3m$ переменных, заменив φ/x на φ и изменив обозначения, можно доказать следующее. Если $n \geqslant 3m$, то φ_0 есть $b_0 + c_0x_0$; для всех $k (0 \leqslant k \leqslant n-1)$ φ_{k+1} есть $a_{k+1} + (b_{k+1} + x_{k+1}) \cdot \varphi_k$; φ_n есть φ, то тогда существуют различные числа $k_1, \ldots, k_m$, такие, что $\varphi/x_0x_{k_1} \ldots x_{k_m}$ эквивалентна некоторой формуле

$$d_0 + (d_1 + d_2x_0) \cdot \prod_{j=1}^{m} (1 + x_{k_j}).$$

Если существует индекс k, такой, что $b_k = 0$ и $m + 1 \leqslant k$, то формула $\varphi/x_0x_1 \ldots x_m$ эквивалентна константе. Предположим поэтому, что $b_k \neq 0$ для всех k, $m + 1 \leqslant k$, подставим 0 вместо $x_1, \ldots, x_m$ и опять изменим обозначения: $n \geqslant 2m$; φ_0 есть $b_0 + c_0x_0$; для всех $k (0 \leqslant k \leqslant n-1)$ φ_{k+1} есть $a_{k+1} + (1 + x_{k+1}) \cdot \varphi_k$; φ_n есть φ. Если все константы a_k, $1 \leqslant k \leqslant n-1$, есть 0, то φ эквивалентна

$$a_n + (b_0 + c_0x_0) \prod_{k=1}^{n} (1 + x_k)$$

и требование леммы выполняется.

В противном случае пусть $i_1, \ldots, i_{p-1}$ — индексы i, такие, что $a_i = 1$ и $1 \leqslant i < n$; $i_1 < i_2 < \ldots < i_{p-1}$.

Определим последовательность $\langle \psi_1, \ldots, \psi_p \rangle$ следующим образом:

$$\psi_1 \quad \text{есть} \quad \prod_{1 \leqslant i \leqslant i_1} (1 + x_i)(b_0 + c_0x_0);$$

$$\psi_{h+1} \quad \text{есть} \quad \prod_{i_h < i \leqslant i_{h+1}} (1 + x_i) \quad (1 \leqslant h \leqslant p-2);$$

$$\psi_p \quad \text{есть} \quad \prod_{i_{p-1} < i} (1 + x_i).$$

Тогда φ эквивалентна формуле ψ, равной

$$c_p + (\ldots (1 + \psi_3 (1 + \psi_2 (1 + \psi_1))) \ldots).$$

Для всех h, j, таких, что $1 \leqslant h < j \leqslant p$, формулы ψ_h, ψ_j не имеют общих переменных.

Подставляя 0 в ψ вместо всех переменных, входящих в одну из формул ψ_{2j}, $j = 1, 2, \ldots$, мы получим формулу χ_1, эквивалентную

$$c + \psi_{2s+1} \cdot \psi_{2s-1} \cdots \psi_s \cdot \psi_1 \quad \left(c = c_p;\ s = \left[\tfrac{1}{2}(p-1)\right]\right).$$

Подставляя 0 в ψ вместо всех переменных, входящих в одну из формул ψ_{2j+1}, $j = 1, 2, \ldots$, а также вместо всех переменных x_i, $i = 1, \ldots, i_1$ (входящих в ψ_1), мы получим формулу χ_2, эквивалентную

$$c + \psi_{2t} \ldots \psi_2 (1 + b_0 + c_0 x_0) \quad \left(t = \left[\tfrac{1}{2} p\right]\right).$$

Среди переменных $x_1, \ldots, x_n$, $n \geqslant 2m$, существуют либо m переменных, входящих в одну из формул $\psi_1, \ldots, \psi_{2s+1}$, либо m переменных, входящих в одну из формул $\psi_2, \ldots, \psi_{2t}$. В обоих случаях существуют различные числа $k_1, \ldots, k_m$, такие, что $\varphi / x_0 x_{k_1} \ldots x_{k_m}$ эквивалентна некоторой формуле

$$d_0 + \prod_j (1 + x_{k_j})(d_1 + d_2 x_0).$$

10. Основная лемма. *Пусть F — рекурсивная функция, определенная следующим образом:*

$$F(m, 0) = m;$$

$$F(m, k) = 4^{(k+1)^{6 \cdot k \cdot F(m, k-1)}} F(F(m, k-1), k-1).$$

Если $n \geqslant F(m, k)$ и φ — формула, зависящая от переменных $x_1, \ldots, x_n$, ни одна из которых не встречается в φ более чем k раз, то существуют различные целые положительные числа $k_1, \ldots, k_m$ $(1 \leqslant k_j \leqslant n,\ 1 \leqslant j \leqslant m)$ и булевские константы c_0, c_1, c_2, такие, что $\varphi / x_{k_1} \ldots x_{k_m}$ эквивалентна

$$c_0 + c_1 \prod_{j=1}^{m} (1 + x_{k_j}) + c_2 \sum_{j=1}^{m} x_{k_j}.$$

Более того, если φ является p-формулой, то $c_2 = 0$.

Доказательство. Доказательство ведется индукцией по k, случай $k = 0$ тривиален. Положив $p = (k+1)^{6 \cdot k \cdot F(m, k-1)}$, $q = F(F(m, k-1), k-1)$ и применив лемму 1, получим, что существует последовательность $\boldsymbol{x}$ переменных x_i $(1 \leqslant i \leqslant n)$, такая, что $\varphi / \boldsymbol{x}$ является либо формулой типа τ_q^1, либо формулой типа τ_p^2.

(1) Если $\varphi/\boldsymbol{x}$ является формулой типа τ_q^1, то существуют формулы ψ_1, ψ_2 и операция $*$, такие, что $\psi_1 * \psi_2$ есть сокращение $\varphi/\boldsymbol{x}$, и такие, что существует по крайней мере q различных переменных $x_{i_1}, \ldots, x_{i_q}$, входящих как в ψ_1, так и в ψ_2. Переменные $x_{i_1}, \ldots, x_{i_q}$, следовательно, имеют не более чем $(k-1)$ вхождение в ψ_j $(j=1, 2)$. Подстановка 0 вместо всех остальных переменных в $\psi_1 * \psi_2$ приводит к формуле $\psi_1' * \psi_2'$. Применяя индукционную гипотезу к ψ_1', видим, что существуют $F(m, k-1)$ переменных $x_{j_1}, \ldots, x_{j_r}$ $(r = F(m, k-1))$, таких, что если положить $\boldsymbol{y} = \langle x_{j_1}, \ldots, x_{j_r}\rangle$, то формула $\psi_1'/\boldsymbol{y}$ окажется эквивалентной некоторой формуле

$$c_0 + c_1\pi + c_2\sigma, \quad \text{где} \quad \pi = \prod_i (x_{l_i} + 1), \quad \sigma = \sum_i x_{l_i}.$$

Применяя индукционную гипотезу к $\psi_2/\boldsymbol{y}$, видим, что существуют m различных переменных $x_{k_1}, \ldots, x_{k_m}$ среди переменных $x_{j_1}, \ldots, x_{j_r}$, таких, что если положить $\boldsymbol{z} = \langle x_{k_1}, \ldots, x_{k_m}\rangle$, то формула $\psi_2/\boldsymbol{z}$ будет эквивалентна некоторой формуле

$$c_0' + c_1'\pi' + c_2'\sigma', \quad \text{где} \quad \pi' = \prod_{j=1}^{m} (1 + x_{k_j}), \quad \sigma' = \sum_{j=1}^{m} x_{k_j}.$$

Формула $\psi_1/\boldsymbol{z}$ эквивалентна

$$c_0 + c_1\pi' + c_2\sigma',$$

и, следовательно, формула $(\psi_1 * \psi_2)/\boldsymbol{z}$ эквивалентна некоторой формуле

$$d_0 + d_1\pi' + d_2\sigma'.$$

Формула $(\psi_1 * \psi_2)/\boldsymbol{z}$ эквивалентна формуле $\varphi/\boldsymbol{z}$.

(2) Предположим теперь, что $\varphi/\boldsymbol{x}$ является формулой типа τ_p^2. Положив $s = 6F(m, k-1)$ и применив лемму 3, получим, что существуют s различных индексов $i_1, \ldots, i_s$ и формулы $\omega_0, \ldots, \omega_s$, такие, что если положить $\boldsymbol{y} = (x_{i_1}, \ldots, x_{i_s})$, то формула ω_s станет эквивалентной $\varphi/\boldsymbol{y}$ (это то же самое, что и $\varphi/\boldsymbol{x}/\boldsymbol{y}$), а последовательность $\langle\omega_0, \ldots, \omega_s\rangle$ будет удовлетворять следующим условиям:

(1) ни одна переменная не входит в ω_0 более чем $(k-1)$ раз;

(2) для всех j, $1 \leqslant j \leqslant s$, формула ω_j эквивалентна некоторой формуле

$$a_j + (b_j + c_j x_{i_j}) \cdot \omega_{j-1} \quad \text{или} \quad (b_j + c_j x_{i_j}) + \omega_{j-1}.$$

Определим последовательность формул $\langle \psi_0, \ldots, \psi_s \rangle$ следующим образом:

$$\psi_0 \text{ есть } x_0.$$

Для всех j, $1 \leqslant j \leqslant s$, формула ψ_j эквивалентна

$$a_j + (b_j + c_j x_{l_j}) \cdot \psi_{j-1} \quad \text{или} \quad (b_j + c_j x_{l_j}) + \psi_{j-1}$$

согласно соотношению между ω_j и ω_{j-1}. Формула ψ_s является базисной в $\langle x_0, x_{l_1}, \ldots, x_{l_s} \rangle$; более того, $\psi_s \Big/ \begin{matrix} x_0 \\ \omega_0 \end{matrix}$ эквивалентна ω_s. Положив $t = F(m, k-1)$ мы получим $s \geqslant 6t$. Применяя лемму 4, получим, что существуют различные целые положительные числа $j_1, \ldots, j_t$, такие, что если положить $\boldsymbol{z} = \langle x_{j_1}, \ldots \cdots, x_{j_t} \rangle$, то формула $\psi_s/\boldsymbol{z}$ станет эквивалентной некоторой формуле

$$d_0 + (d_1 + d_2 x_0) \cdot \prod_{i=1}^{t} (1 + x_{l_i}),$$

$$d_0 + d_1 x_0 + d_2 \cdot \sum_{i=1}^{t} x_{l_i}.$$

Формула $\varphi/\boldsymbol{z}$, следовательно, эквивалентна **некоторой формуле**

$$d_0 + (d_1 + d_2 \omega_0/\boldsymbol{z}) \cdot \prod_{i} (1 + x_{l_i}),$$

$$d_0 + d_1 \omega_0/\boldsymbol{z} + d_2 \sum_{i} x_{l_i}.$$

Мы имеем $t = F(m, k-1)$; применяя индукционную гипотезу к $\omega_0/\boldsymbol{z}$, получаем m различных целых положительных чисел $k_1, \ldots, k_m$, таких, что если положить $\boldsymbol{u} = \langle x_{k_1}, \ldots, x_{k_m} \rangle$, то формула $\omega_0/\boldsymbol{u}$ станет эквивалентной некоторой формуле

$$c_0 + c_1 \pi + c_2 \sigma, \quad \text{где} \quad \pi = \prod_{j} (1 + x_{k_j}), \quad \sigma = \sum_{j} x_{k_j}.$$

Так как $\pi \cdot \sigma = 0$, то формула $\varphi/\boldsymbol{z}$ сама будет эквивалентна некоторой формуле

$$e_0 + e_1 \pi + e_2 \sigma.$$

Если φ есть p-формула, то можно принять $e_2 = 0$.

11. Теорема. *Если* $n \geqslant 2 \cdot F(m, 2 \cdot c)$ (F — *функция, определенная в п.* 10) *и* φ — *формула, зависящая от переменных*

$x_1, \ldots, x_n$, *длина которой меньше, чем* $c \cdot n$, *то существуют целые числа* $k_1, \ldots, k_m$ $(1 \leqslant k_1 < \ldots < k_m \leqslant n)$ *и булевские константы* c_0, c_1, c_2 *такие, что* $\varphi / x_{k_1} \ldots x_{k_m}$ *эквивалентна*

$$c_0 + c_1 \prod_{j=1}^{m} (1 + x_{k_j}) + c_2 \sum_{j=1}^{m} x_{k_j}.$$

Более того, если φ *является p-формулой, то* $c_2 = 0$.

Доказательство. Пусть n_1 — число переменных x_i, $1 \leqslant i \leqslant n$, входящих в φ более чем $2c$ раз, и n_2 — число переменных, входящих в φ не более чем $2c$ раз. Ясно, что $n_1 + n_2 = n$ и $c \cdot n \geqslant n_1 \cdot 2c$. Следовательно, $n_2 \geqslant F(m, 2c)$. Пусть $x_{l_1}, \ldots, x_{l_{n_2}}$ — различные переменные, входящие в φ не более чем $2c$ раз. Определим $\boldsymbol{x} = \langle x_{l_1}, \ldots, x_{l_{n_2}} \rangle$, и пусть ψ — формула $\varphi / \boldsymbol{x}$. Утверждение теоремы получается, если к формуле ψ применить основную лемму.

Теорема 1. *Для каждого числа* c *существует формула* φ, *такая, что для каждой p-формулы* ψ, *эквивалентной* φ, *имеет место следующее неравенство*:

$$\textit{длина}\ \psi \geqslant c \cdot \textit{длина}\ \varphi.$$

Доказательство. Предположим, что $n = 2F(2, 2 \cdot c)$, и пусть φ — формула $\sum_{i=1}^{n} x_i$. Если ψ — p-формула от переменных $x_1, \ldots, x_n$ длины, меньшей чем $c \cdot n$, то существуют целые положительные числа h, i и булевские константы d_0, d_1, такие, что $1 \leqslant h < i \leqslant n$ и что $\psi / x_h x_i$ эквивалентна $d_0 + d_1(1 + x_h)(1 + x_i)$. Формула $\varphi / x_h x_i$ эквивалентна $x_h + x_i$, следовательно формулы φ и ψ не эквивалентны.

Теорема 2. *Для каждого числа* c *найдется формула пропозиционального исчисления второго порядка* Φ (в базисе конъюнкция и отрицание), *такая, что для каждой формулы* ψ (*формулы пропозиционального исчисления первого порядка*), *эквивалентной* Φ, *имеет место следующее неравенство*:

$$\textit{длина}\ \psi \geqslant c \cdot \textit{длина}\ \Phi.$$

Доказательство. Предположим, что $n = 2 \cdot F(3, 150 \cdot c)$, и определим формулы φ_k, $1 \leqslant k \leqslant n$, следующим образом:

$$\varphi_1 \text{ есть } (x_1 + u_1) \cdot (1 + x_1 + v_1) \cdot (1 + w_1),$$

$$\varphi_{k+1} \text{ есть } (1 + u_k + x_{k+1} \cdot u_k + x_{k+1} \cdot w_k + u_{k+1}) \times$$
$$\times (1 + v_k + x_{k+1} \cdot v_k + x_{k+1} \cdot u_k + v_{k+1}) \times$$
$$\times (1 + w_k + x_{k+1} \cdot w_k + x_{k+1} \cdot v_k + w_{k+1}).$$

Длина формулы φ', определяемой как $\prod_{i=1}^{n} \varphi_i \cdot u_n$, равняется $18n - 12$. Существует p-формула φ, эквивалентная φ', длина которой не превосходит $4 \cdot (18n - 12)$. Пусть Φ — формула

$$(\exists u_1)(\exists u_2) \ldots (\exists u_n)(\exists v_1) \ldots (\exists v_n)(\exists w_1) \ldots (\exists w_n)\, \varphi.$$

Тогда длина Φ меньше, чем $75 \cdot n$.

Если $\langle c_1, \ldots, c_n \rangle$ — последовательность булевских констант, то $\Phi\Big/{}^{x_1 \ldots x_n}_{c_1 \ldots c_n}$ принимает значение 1 в том и только том случае, если число индексов i, таких, что $1 \leqslant i \leqslant n$ и $c_i = 1$, равно 0 по модулю 3 («u_k», «v_k», «w_k» говорят о том, что число констант среди $c_1, \ldots, c_k$, равных 1, равно соответственно 0, 1, 2 по модулю 3).

Пусть ψ — формула длины, меньшей чем $c \cdot (75 \cdot n)$. Так как $n = 2 \cdot F(3, 150 \cdot c)$, то существуют индексы h, i, j и булевские константы d_0, d_1, d_2, такие, что $1 \leqslant h < i < j \leqslant n$ и что $\psi/x_h x_i x_j$ эквивалентна

$$d_0 + d_1 (1 + x_h) \cdot (1 + x_i) \cdot (1 + x_j) + d_2 (x_h + x_i + x_j).$$

Если $\langle c_1, \ldots, c_n \rangle$ — последовательность, такая, что $c_h = 1$ и $c_k = 0$ для $k \neq h$ $(1 \leqslant k \leqslant n)$, то $\psi\Big/{}^{x_1 \ldots x_n}_{c_1 \ldots c_n}$ принимает значение $d_0 + d_2$. Если $\langle c'_1, \ldots, c'_n \rangle$ — последовательность, такая, что $c'_h = c'_i = c'_j = 1$ и $c_k = 0$ для остальных k $(1 \leqslant k \leqslant n)$, то $\psi\Big/{}^{x_1 \ldots x_n}_{c'_1 \ldots c'_n}$ принимает значение $d_0 + d_2$. С другой стороны, $\Phi\Big/{}^{x_1 \ldots x_n}_{c_1 \ldots c_n}$ равно 0, а $\Phi\Big/{}^{x_1 \ldots x_n}_{c'_1 \ldots c'_n}$ равно 1; следовательно, формулы Φ и ψ не эквивалентны.

СПИСОК ЛИТЕРАТУРЫ

1. Hodes L., Specker E., Elimination of quantifiers and the length of formulae, *Notices Am. Math. Soc.*, **12** (1965), 242.
2. Hodes L., Specker E., Elimination von Quantoren und Länge von Formeln, Abstract J. Symb. Logic.
3. Нечипорук Е. И., Булевские функции, *Доклады Академии наук СССР*, **169** (1966), 765—766.

Характеристика контекстно-свободных языков

Й. Груска

1. ВВЕДЕНИЕ

В последние годы теория языков и автоматов развивалась в направлении изучения новых грамматик, автоматов и языков, абстрактных семейств языков, трансляторов и т. д. Несмотря на это, центральное место в теории языков по-прежнему занимают регулярные и контекстно-свободные языки (КСЯ), и можно ожидать, что их изучение окажется полезным и послужит стимулирующим фактором для дальнейшего развития теории языков.

Важным преимуществом регулярных языков является возможность описать их с помощью регулярных выражений наряду с представлением этих языков посредством грамматик. Аналогичная характеристика КСЯ, возможно, также окажется полезной. Однако едва ли КСЯ можно охарактеризовать так же элегантно и просто, как регулярные языки.

В разд. 2 дается характеристика КСЯ регулярного типа (регулярно-подобная, regular-like) с использованием объединения, произведения (конкатенации) и новой операционно-символьной итерации. При этом мы вынуждены использовать некоторые «вспомогательные» символы. Их роль и количество обсуждаются в разд. 3. Применение результатов второго раздела к полулинейным языкам и их подклассам представлено в разд. 4.

2. РЕГУЛЯРНО-ПОДОБНАЯ ХАРАКТЕРИСТИКА КСЯ

В этом разделе устанавливается регулярно-подобная характеристика КСЯ.

Мы начнем с введения основных понятий и обозначений теории языков. Если L_1 и L_2 — языки, то произведение L_1 на L_2, короче $L_1 \cdot L_2$ или $L_1 L_2$, есть язык $L_1 \cdot L_2 = \{z \mid z = xy;\ x \in L_1;\ y \in L_2\}$. Для языка L и целого числа $i \geqslant 0$ L^i определяется как $L^0 = \{\varepsilon\}$, где ε — пустое слово, $L^{i+1} = L^i \cdot L$ для $i \geqslant 0$. Наконец,

$$L^* = \bigcup_{i=0}^{\infty} L^i, \quad L^+ = \bigcup_{i=1}^{\infty} L^i.$$

---

[1]) Gruska Jozef, A characterization of context-free languages, *J. Comput. Syst. Sci.*, 5, № 4 (1971), 353—364.

Контекстно-свободная грамматика G определяется 4-набором $G = \langle \Sigma_N, \Sigma_T, P, \sigma \rangle$, где Σ_N — алфавит нетерминальных символов, короче говоря, переменных; Σ_T — алфавит терминальных символов; $\Sigma_N \cap \Sigma_T = \varnothing$; начальный символ $\sigma \in \Sigma_N$ и P — конечное множество правил вида $A \to w$, где $A \in \Sigma_N$, $w \in (\Sigma_T \cup \Sigma_N)^*$. Для двух слов x и y мы записываем $x \Rightarrow y$ тогда и только тогда, когда $x = uAv$, $y = uwv$ для некоторых u и v из $(\Sigma_T \cup \Sigma_N)^*$, а в P есть правило $A \to w$. Через $\overset{*}{\Rightarrow}$ обозначим транзитивное и рефлексивное замыкание $\Rightarrow$. Множество $L(G) = \{x \mid \sigma \overset{*}{\Rightarrow} x \in \Sigma_T^*\}$ называется языком, порожденным грамматикой G. Язык L называется контекстно-свободным, если $L = L(G)$ для некоторой контекстно-свободной грамматики G.

Основная операция в нашей регулярно-подобной характеристике контекстно-свободных языков есть операция «символьной итерации», которая аналогично специальному случаю гнездной итеризованной подстановки Крала и Грейбах [5]. Для определения этой операции введем следующее

Определение 2.1. Пусть σ есть буква, а L и L_1 — языки. Тогда σ-*подстановкой* L_1 *в* L, обозначаемой $L \overset{\sigma}{\uparrow} L_1$, называется

$$L \overset{\sigma}{\uparrow} L_1 = \{z \mid z = x_1 y_1 x_2 \ldots x_k y_k x_{k+1};\ x_1 \sigma x_2 \sigma \ldots \sigma x_{k+1} \in L;$$

σ не встречается в словах $x_1, x_2, \ldots, x_{k+1}$, а $y_1 \in L_1$ для $1 \leqslant i \leqslant k\}$.

Следствие 2.2. *Для каждого символа* σ *операция* $\overset{\sigma}{\uparrow}$ *ассоциативна.*

Определение 2.3. Пусть σ — буква, а L — язык. σ-*итерацией* L, обозначаемой через L^σ, называется множество

$$L^\sigma = \{z \mid z \in L \cup L \overset{\sigma}{\uparrow} L \cup L \overset{\sigma}{\uparrow} L \uparrow L \cup \ldots;\ z \text{ не содержит } \sigma\}.$$

Замечание. Если L — язык над алфавитом Σ_L, а σ — буква, то L^σ является языком, порожденным грамматикой $G = \langle \{\sigma\}, \Sigma_L - \{\sigma\}, P, \sigma \rangle$, где $P = \{(\sigma \to x) \mid x \in L\}$ есть, вообще говоря, бесконечное множество продукций. Если язык L конечен, то L^σ является контекстно-свободным языком.

Если σ не принадлежит Σ_L, то $L^* = (L\sigma \cup \{\varepsilon\})^\sigma$, а $L^+ = (L\sigma \cup L)^\sigma$.

Следующая лемма подытоживает некоторые отношения между операциями символьной подстановки и символьной итерации.

Лемма 2.4. *Пусть* L, L_1 *и* L_2 — *языки над алфавитами* Σ_L, Σ_{L_1} *и* Σ_{L_2} *соответственно, а* σ *и* ρ — *буквы. Тогда:*

(i) $$(L^\sigma)^\rho = (L^\rho)^\sigma;$$

(ii) *если* $\sigma \notin \Sigma_L$, *то*

$$(L_1 \overset{\sigma}{\uparrow} L_2) \overset{\rho}{\uparrow} L = (L_1 \overset{\rho}{\uparrow} L) \overset{\sigma}{\uparrow} (L_2 \overset{\rho}{\uparrow} L);$$

(iii) *если* $\sigma \notin \Sigma_L$, *то*

$$L \overset{\rho}{\uparrow} (L_1 \overset{\rho}{\uparrow} L_2) = (L \overset{\rho}{\uparrow} L_1) \overset{\sigma}{\uparrow} L_2;$$

(iv) *если* $\sigma \notin \Sigma_L$, *то*

$$(L_1 \overset{\sigma}{\uparrow} L)^\sigma = (L_1^\sigma) \overset{\rho}{\uparrow} L.$$

Доказательство. Очевидно, для доказательства (i) достаточно показать, что $(L^\sigma)^\rho \subseteq (L^\rho)^\sigma$. Если $z \in (L^\sigma)^\rho$, то существует последовательность слов $z_1, z_2, \ldots, z_r$, такая, что $z_1 \in L^\sigma$, $z_k = z$, а $z_i = u_i \rho v_i$, $z_{i+1} = u_i w_{i+1} v_i$ и $w_{i+1} \in L^\sigma$ для $i = 1, 2, \ldots, r-1$. Если положить $w_1 = z_1$, то $w_1 \in L^\sigma$ для $i = 1, 2, \ldots, r$, и поэтому для каждого такого i существует последовательность $z_1^i, z_2^i, \ldots, z_{k_i}^i$, такая, что $z_1^i \in L$, $z_{k_i}^i = w_i$ и z_{j+1}^i для $j = 1, 2, \ldots, k_i - 1$ получается из z_j^i заменой σ словом из L. Образуем теперь последовательность

$$(*) \qquad z_1^1, \ldots, z_{k_1}^1 = z_1, \quad u_1 z_1^2 v_1, \ldots, u_1 z_{k_2}^2 v_1 = z_2,$$
$$u_2 z_1^3 v_2, \ldots, u_{r-1} z_{k_r}^r v_{r-1} = z,$$

которая является фактически выводом z из z_1^1 в контекстно-свободной грамматике G с двумя нетерминалами σ и ρ и правилами $\sigma \to \alpha$, $\rho \to \alpha$ для всякого $\alpha \in L$. Вывод (*) обладает следующим свойством: буква ρ заменяется на слово из L в том и только том случае, когда ни одна буква σ не может быть заменена. Очевидно, что существует дерево вывода, соответствующее (*), и этому дереву соответствует другой вывод z из z_1^1 в G, при котором σ заменяется тогда и только тогда, когда ни одна буква ρ не может быть заменена. Но это означает, что $z \in (L^\rho)^\sigma$. Следовательно, $(L^\sigma)^\rho \subseteq (L^\rho)^\sigma$, что завершает доказательство (i).

Остальные утверждения (ii) — (iv) теперь очевидны.

Определение 2.5. Пусть Σ — алфавит (возможно бесконечный). Обозначим через $\mathscr{E}_\Sigma$ наименьший класс языков, удовлетворяющий следующим условиям:

(1) $\{\varepsilon\} \in \mathscr{E}_\Sigma$;
(2) если $a \in \Sigma$, то $\{a\} \in \mathscr{E}_\Sigma$;
(3) если L_1, L_2 принадлежат $\mathscr{E}_\Sigma$ и $\sigma \in \Sigma$, то $L_1 \cup L_2$, $L_1 \cdot L_2$ и L_1^σ принадлежат $\mathscr{E}_\Sigma$.

Замечание. Будем говорить, что L есть язык над бесконечным алфавитом Σ, если $L \subset \Sigma_L^*$ для некоторого конечного подмножества $\Sigma_L \subset \Sigma$.

Лемма 2.6. *Если Σ — бесконечный алфавит, $\sigma \in \Sigma$ и L_1, L_2 — языки из $\mathscr{E}_\Sigma$, то $L_1 \overset{\sigma}{\uparrow} L_2$ принадлежит $\mathscr{E}_\Sigma$.*

Доказательство. Фиксируем L_2 в $\mathscr{E}_\Sigma$. Если $a \in \Sigma$, то $\{a\} \overset{\sigma}{\uparrow} L_2$ есть либо $\{a\}$, либо L_2 и в обоих случаях принадлежит $\mathscr{E}_\Sigma$. Семейство Ξ языков L, таких, что язык $L \overset{\sigma}{\uparrow} L_2$ принадлежит $\mathscr{E}_\Sigma$, замкнуто относительно объединения, произведения и σ_0-итерации для $\sigma_0 \in \Sigma$. Так как Ξ содержит $\{\varepsilon\}$ и $\{a\}$ для каждого $a \in \Sigma$, мы видим, что Ξ содержит $\mathscr{E}_\Sigma$. Таким образом, если L_1 принадлежит $\mathscr{E}_\Sigma$, то $L_1 \overset{\sigma}{\uparrow} L_2$ также принадлежит $\mathscr{E}_\Sigma$. Поскольку язык L_2 выбран произвольно, это завершает доказательство.

Теперь мы в состоянии доказать основной результат этого раздела, а именно, что каждый КСЯ может быть представлен с помощью операций $\cup$, $\cdot$ и символьной итерации.

Теорема 2.7. *Если Σ — бесконечный алфавит, то L является (непустым) КСЯ над Σ тогда и только тогда, когда $L \in \mathscr{E}_\Sigma$.*

Доказательство. Согласно определению 2.5, для того чтобы доказать, что каждый язык из $\mathscr{E}_\Sigma$ является контекстно-свободным, достаточно показать, что для всякого контекстно-свободного языка L и всякой буквы σ язык L^σ также является контекстно-свободным. Пусть дан язык L, а G есть контекстно-свободная грамматика (КСГ), порождающая L, причем σ не является переменной G. Тогда L^σ является языком, порожденным грамматикой G' с начальным символом σ, а правила G' — те же, что у G, плюс еще одно правило: $\sigma \to \sigma_0$, где σ_0 есть начальный символ G. Таким образом, L^σ является КСЯ. Остается доказать, что $\mathscr{E}_\Sigma$ содержит все КСЯ над Σ.

Пусть L — непустой КСЯ над алфавитом $\Sigma_L \subset \Sigma$. Поскольку Σ — бесконечный алфавит, существует КСГ $G = \langle \Sigma_N, \Sigma_L, P, \sigma \rangle$, порождающая L и такая, что $\Sigma_N \subset \Sigma$. Пусть $\sigma_1, \sigma_2, \ldots, \sigma_n$ является списком переменных G и $\sigma_i \neq \sigma_j$ при $i \neq j$. Без уменьшения

общности можно предположить, что множества $\theta_i = \{x \mid \sigma_i \overset{*}{\Rightarrow} x \in \Sigma_L^*\}$, $1 \leqslant i \leqslant n$, непусты.

Определим множества (для $1 \leqslant i \leqslant n$)

$$V_i = \{\sigma_j \mid i \leqslant j \leqslant n\}, \quad P_i = \{\sigma' \to \alpha) \mid (\sigma' \to \alpha) \in P,\ \sigma' \in V_i\}$$

и грамматики

$$G_i = \langle V_i,\ \Sigma_L \cup (\Sigma_N - V_i),\ P_i,\ \sigma_i \rangle.$$

Кроме того, определим для $1 \leqslant i \leqslant k \leqslant n$ языки $\Phi_{i,k}$ с помощью двойной спускающейся от n индукции:

$$\Phi_{i,n} = \{\alpha \mid (\sigma_i \to \alpha) \in P\} \quad \text{для} \quad 1 \leqslant i \leqslant n,$$

а для $i < k \leqslant n$

$$(*) \qquad \Phi_{i,k-1} = \Phi_{i,k} \overset{\sigma_k}{\uparrow} \Phi_{k,k}^{\sigma_k}.$$

Согласно лемме 2.6, множества $\Phi_{i,k}$ входят в $\mathscr{E}_\Sigma$. Используя (*), определения 2.1 и 2.3 и спускающуюся от n индукцию по k, мы получим для $1 \leqslant i \leqslant k \leqslant n$:

$\Phi_{i,k} = \{x \mid$ существует $\alpha \in \Phi_{i,n}$, такое, что $\sigma_i \to \alpha \overset{*}{\Rightarrow} x$ в G, x не содержит букв из V_{k+1} и существует вывод x из α в G, в котором применяются лишь правила из $P_{k+1}\}$ [1])

и

$\Phi_{i,i}^{\sigma_i} = \{x \mid \sigma_i \overset{*}{\Rightarrow} x$, где x не содержит букв из V_i и существует вывод x из σ_i, в котором применяются лишь правила из $P_i\}$.

Поэтому $\Phi_{i,i}^{\sigma_i} \subseteq L(G_i)$ для $1 \leqslant i \leqslant n$. Для завершения доказательства достаточно показать, что $L(G_i) \subseteq \Phi_{i,i}^{b_i}$ также для $1 \leqslant i \leqslant n$. В доказательстве применяется индукция по i, спускающаяся от n.

Это, очевидно, имеет место для $i = n$. Предположим, что $L(G_j) \subseteq \Phi_{j,j}^{\sigma_j}$ (а потому и $L(G_j) = \Phi_{j,j}^{\sigma_j}$) для $j = i+1,\ i+2,\ \ldots,\ n$. Покажем, что тогда $L(G_i) \subseteq \Phi_{i,i}^{\sigma_i}$, для чего положим $\pi(w) = \max\{i \mid \sigma_j$ встречается в $w\}$ для каждого $w \in (\Sigma_N \cup \Sigma_L)^* - \Sigma_L^*$. Сначала покажем, что

если в P есть правило $\sigma_i \to \alpha$ и x может быть выведено из α в G_{i+1}, то $x \in \Phi_{i,i}$. (1)

[1]) Ради простоты положим $V_{n+1} = \theta = P_{n+1}$.

В самом деле, пусть x обладает указанным свойством. Тогда существует вывод τ слова x из σ_τ:

$$\tau\colon \sigma_l = w_1,\ w_2,\ \ldots,\ w_l = x,$$

такой, что $\pi(w_j) > i$ для $1 < j \leqslant l$ и на каждом шаге вывода заменяется буква σ_j с максимальным j. Для простоты, пусть $\pi(j)$ обозначает $\pi(w_j)$ для $1 \leqslant j \leqslant l$. Определим последовательность целых чисел $k_1, k_2, \ldots, k_r$ следующим образом: $k_1 = 2$. Пусть уже определены числа $k_1, k_2, \ldots, k_j$. Если $k_j = l$, то $j = r$. Если $k_j < l$ и $\pi(\nu) \geqslant \pi(k_j)$ для $k_j < \nu < l$, то $k_{j+1} = l$; в противном случае $k_{j+1} = \min\{\nu \mid \nu > k_j,\ \pi(\nu) < \pi(k_j)\}$. Очевидно, что w_{k_2} получается из $w_{k_1} = w_2$ заменой всех букв $\sigma_{\pi(2)}$ словами из $L(G_{\pi(2)}) = \Phi^{\sigma_{\pi(2)}}_{\pi(2),\,\pi(2)}$. Таким образом, $w_{k_2} \in \Phi_{l,\,\pi(2)-1}$. Вообще, если $2 < j < l$ и $w_{k_j} \in \Phi_{l,\,\pi(k_{j-1})-1}$, то $w_{k_j} \in \Phi_{l,\,\pi(k_j)}$ также, и $w_{k_{j+1}}$ получается из w_{k_j} заменой всех букв $\sigma_{\pi(k_j)}$ некоторыми словами $L(G_{\pi(k_j)})$ и поэтому $w_{k_{j+1}} \in \Phi_{i,\,\pi(k_j)-1}$. Таким образом, $x \in \Phi_{i,\,i}$, и утверждение (1) доказано.

Теперь предположим, что $x \in L(G_i)$, и пусть опять $\sigma_l = w_1,\ w_2,\ \ldots,\ w_l = x$ является выводом x, таким, что на каждом шаге вывода заменяется буква σ_j с максимальным индексом. Пусть $1 = k_1 < k_2 < k_3 \ldots < k_r = l$ является последовательностью всех целых j, таких, что $\pi(j) = i$ и $j < l$. Согласно (1), это означает, что $w_{k_2} \in \Phi_{i,\,i}$. Кроме того, w_{k_j} (для $j > 2$) получается из $w_{k_{j-1}}$ заменой букв σ_i некоторыми словами, которые, согласно (1), принадлежат $\Phi_{i,\,i}$. Так как σ_i не входит в x, то x должно принадлежать $\Phi^{\sigma_i}_{i,\,i}$. Таким образом, $L(G_i) \subseteq \Phi^{\sigma_i}_{i,\,i}$, что и завершает доказательство теоремы.

Замечание. Регулярные языки часто определяются как события, представимые регулярными выражениями. Ввиду доказанной теоремы, КС-языки над бесконечным алфавитом также можно определить как языки, представленные контекстно-свободными выражениями (КСВ) над Σ, которые определяются следующим образом (здесь мы предполагаем, что для каждой буквы $\sigma \in \Sigma$ есть специальная буква σ, не входящая в Σ):

(1) если $a \in \Sigma \cup \{\varepsilon\}$, то a является КСВ;

(2) если E_1, E_2 являются КСВ и $\sigma \in \Sigma$, то $(E_1 \cup E_2)$, $(E_1 \cdot E_2)$, $(E_1\sigma)$ являются КСВ.

Здесь мы предполагаем, что если КСВ E представляет язык L, то выражение $(E\sigma)$ представляет язык L^σ.

Следствие 2.8. *Если* Σ — *конечный алфавит, то* L *является КСЯ над* Σ *тогда и только тогда, когда* $L \in \mathscr{E}_{\Sigma'}$ *для некоторого конечного алфавита* $\Sigma' \supseteq \Sigma$.

Доказательство немедленно следует из теоремы 2.7.

Пример 1. Если $L = \{a^i b^i \mid i \geqslant 1\}$, то $L = (a\sigma b \cup ab)^\sigma$.

Пример 2. Пусть L является языком, порожденным грамматикой с начальным символом σ_1 и правилами

$$\sigma_1 \to a\sigma_1 b, \quad \sigma_1 \to ba^2\sigma_2 ba,$$
$$\sigma_2 \to a^2\sigma_2 b, \quad \sigma_2 \to ba^3\sigma_3 ba,$$
$$\sigma_3 \to a^3\sigma_3 b, \quad \sigma_3 \to ba\sigma_1 ba, \quad \sigma_3 \to b.$$

Тогда $L = (ba^2 (ba^3 (a^3\sigma b \cup bapba \cup b)^\sigma ba \cup a^2\sigma b)^\sigma ba \cup apb)^\sigma$.

3. ВСПОМОГАТЕЛЬНЫЕ СИМВОЛЫ

Для конечного алфавита Σ и целого числа k пусть $\mathscr{E}_{\Sigma, k}$ обозначает класс всех КСЯ над Σ, входящих в $\mathscr{E}_{\Sigma'}$, где $\Sigma' = \Sigma \cup \{\sigma_1, \ldots, \sigma_k\}$, а $\sigma_1, \ldots, \sigma_k$ — различные буквы, не принадлежащие Σ. (Иными словами k в $\mathscr{E}_{\Sigma, k}$ является числом «вспомогательных» символов, не входящих в Σ, но используемых в регулярно-подобных конструкциях языков из $\mathscr{E}_{\Sigma, k}$.

Возникает естественный вопрос: существует ли для данного конечного алфавита Σ такое число k, что $\mathscr{E}_{\Sigma, k}$ содержит все КСЯ над Σ? Мы уверены в том, что если Σ содержит по крайней мере два символа, то ответ на поставленный вопрос отрицательный, но мы не располагаем доказательством этого [1]). Если Σ содержит только один символ, то все КСЯ над Σ регулярны и содержатся в $\mathscr{E}_{\Sigma, 1}$.

В связи с доказательством теоремы 2.7 кажется, что вспомогательные символы, используемые в регулярно-подобных конструкциях КСЯ, играют ту же роль, что и переменные в контекстно-свободных грамматиках. Известно [7], что для всякого натурального n и любого алфавита Σ с не менее чем двумя символами существует линейный КСЯ L_n, который не может быть порожден КС-грамматикой, содержащей меньше чем n переменных. Например, если $\Sigma = \{a, b\}$, то язык, порожденный КСГ с начальным символом σ и правилами

$$\sigma_i \to a^i\sigma_1 b, \quad \sigma_i \to ba^{i+1}\sigma_{i+1} ba, \quad 1 \leqslant i \leqslant n-1$$
$$\sigma_n \to a^n\sigma_n b, \quad \sigma_n \to ba\sigma_1 ba, \quad \sigma_n \to b^2a^2$$

[1]) Доказательство утверждения, связанного с этой проблемой, приведено в статье Мак-Вертера «Подстановочные выражения», помещенной в наст. сборнике (стр. 127—136). — *Прим. перев.*

обладает таким свойством. Однако, применяя тот же способ, что в примере 2 из разд. 1, можно показать, что этот язык входит в $\mathscr{L}_{\Sigma, 2}$. Мы даже можем доказать следующую теорему.

Теорема 3.1. *Для любого конечного алфавита* Σ *класс* $\mathscr{L}_{\Sigma, 2}$ *содержит все линейные языки над* Σ.

Эта теорема следует из леммы 3.2, доказательство которой проще.

Лемма 3.2. *Пусть* $G = \langle \Sigma_N, \Sigma, P, \sigma\rangle$ — *линейная КСГ и* A, B *различные буквы, не входящие в* $\Sigma_N \cup \Sigma$. *Пусть* $G' = \langle \Sigma_N, \Sigma \cup \{A\}, P \cup P', \sigma\rangle$ — *новая грамматика, образованная из* G *добавлением конечного множества* P' *правил вида* $X \to uAv$, *где* $X \in \Sigma_N$, *а* $uv \in \Sigma^*$. *Тогда* $L(G) \in \mathscr{L}_{\Sigma \cup \{A, B\}}$, $L(G') \in \mathscr{L}_{\Sigma \cup \{A, B\}}$ *и оба языка* $L(G)$ *и* $L(G')$ *могут быть получены из языков* $\{a\}$, $a \in \Sigma \cup \{A, B\} \cup \{\varepsilon\}$, *посредством конечного числа операций* $\cup$, $\cdot$, A-*итераций и* B-*итераций.*

Доказательство. Доказательство проведем индукцией по числу переменных G. Лемма очевидным образом справедлива, если G содержит только одну переменную. Предположим теперь, что лемма выполняется для всех линейных грамматик с не более чем $k-1$ переменными. Пусть $G = \langle \Sigma_N, \Sigma, P, \sigma\rangle$ — линейная грамматика с k переменными $\sigma = \sigma_1, \sigma_2, \ldots, \sigma_k$. Определим грамматики G_i $(2 \leqslant i \leqslant n)$, полагая $G_i = \langle \Sigma_N - \{\sigma\}, \Sigma \cup \{\sigma\}, \bar{P}, \sigma_i\rangle$, $\bar{P} = P - \{(\sigma \to \alpha);\ (\sigma \to \alpha) \in P\}$. В каждой грамматике G_i $(2 \leqslant i \leqslant n)$ имеется $k-1$ переменная, поэтому, согласно второй части предположения индукции, $L(G_i) \in \mathscr{L}_{\Sigma \cup \{\sigma, b\}}$ и язык $L(G_i)$ может быть получен из языков $\{a\}$: $a \in \Sigma \cup \{\sigma, B\} \cup \{\varepsilon\}$ посредством конечного числа операций $\cup$, $\cdot$, σ-итерации и B-итерации.

Пусть $\sigma \to \alpha_j$ $(1 \leqslant j \leqslant r)$ является списком всех правил грамматики G с σ в левой стороне. Определим языки L_j $(1 \leqslant j \leqslant r)$: если $\alpha_j \in (\Sigma \cup \{\sigma\})^*$, то $L_j = \{\alpha_j\}$. Если $\alpha_j = u\alpha_l v$ для некоторого $l > 1$, то $L_j = uL(G_l)v$. Очевидно, что $L(G) = \left(\bigcup_{j=1}^{r} L_j\right)^{\sigma}$ и поэтому $L(G) \in \mathscr{L}_{\Sigma \cup \{\sigma, B\}}$. Букву σ можно заменить на A, и это завершает доказательство.

Пусть теперь $G' = \langle \Sigma_N, \Sigma \cup \{A\}, P \cup P', \sigma\rangle$, где P' — конечное множество правил вида $X \to uAv$, $X \in \Sigma_N$, $uv \in \Sigma^*$. Определим для $2 \leqslant i \leqslant k$ грамматики G_i и G_i', полагая

$G_i = \langle \Sigma_N - \{\sigma\}, \Sigma \cup \{A\}, P_i, \sigma_i\rangle$, где P_i получается из $P \cup P'$ удалением всех правил, содержаших (слева или справа) букву σ;

$G_i' = \langle \Sigma_N - \{\sigma\}, \Sigma \cup \{\sigma\}, P_i', \sigma_i\rangle$, где P_i' получается из P удалением всех правил, содержащих σ слева.

В соответствии с предположением индукции оба языка $L(G_i)$ и $L(G_i')$ входят в $\mathscr{E}_{\Sigma\cup\{\sigma,\ A\}}$ и могут быть получены из языков $\{a\}$, $a\in\Sigma\cup\{\sigma,\ A\}\cup\{\varepsilon\}$ операциями $\cup$, $\cdot$, σ-итерацией и A-интерацией.

Пусть $\sigma\to\alpha_j$ $(1\leqslant j\leqslant r)$ является списком всех правил грамматики G с σ в левой части. Определим языки $L_j(1\leqslant j\leqslant r)$ следующим образом: если $\alpha_j\in(\Sigma\cup\{\sigma,\ A\})^*$, то $L_j=\{\alpha_j\}$; если $\alpha_j=u\sigma_i v$ для некоторого $i>1$, то $L_j=u(L(G_i)\cup L(G_i'))\,v$. Очевидно, что $L(G')=\left(\bigcup_{j=1}^{r}L_j\right)^{\sigma}$ и лемма также справедлива для G'. Тем самым завершен шаг индукции, а также доказательство леммы.

Замечание. Теорема 3.1. верна также для полулинейных языков (см. разд. 4), так как класс полулинейных языков является наименьшим классом, содержащим все линейные языки, и замкнутым относительно подстановки.

Из теоремы 3.1. следует, что из того факта, что для порождения всех КСЯ требуется бесконечно много переменных, нельзя заключить, что необходимо также неограниченно много вспомогательных символов в регулярно-подобных построениях КСЯ, и поэтому вспомогательные символы играют несколько отличную роль.

Вспомогательные символы необходимы, так как класс $\mathscr{E}_\Sigma$, где Σ — конечный алфавит, не содержит всех регулярных языков над Σ. Однако все они содержатся в $\mathscr{E}_{\Sigma,\,1}$.

Существуют также нелинейные КСЯ, которые не могут порождаться грамматиками с менее чем n нетерминальными символами, но которые принадлежат $\mathscr{E}_{\Sigma,\,2}$. Например, язык, порожденный грамматикой с начальным символом σ_1 и правилами

$$\sigma_i\to a^i\sigma_i b,\ \sigma_i\to ca\sigma_1 bca^2\sigma_2 bc\ \ldots\ ca^n\sigma_n bc,\ \sigma_i\to c,\ 1\leqslant i\leqslant n$$

обладает таким свойством.

4. ПРИМЕНЕНИЕ К ПОЛУЛИНЕЙНЫМ ЯЗЫКАМ

Согласно разд. 2, каждый КСЯ можно охарактеризовать, используя только операции объединения, произведения и символьных итераций. Секрет заключается в том, что символьные итерации являются достаточно сложными и не очень прозрачными операциями. В случае линейных и полулинейных языков символьные итерации имеют более простой и прозрачный вид, и эти языки характеризуются более простыми операциями. Ре-

зультаты этого раздела основаны на докладе [6] и имеют много общего с результатами Интемы [8] и Брзозовски [2][1]).

Теорема 4.1. *Для алфавита* Σ *через* $\boldsymbol{S} \subset 2^{\Sigma^*}$ *и* $\boldsymbol{D} \subset 2^{\Sigma^* \times \Sigma^*}$ *обозначим наименьшие классы, удовлетворяющие следующим условиям:*

(1) *если* $a, b \in \Sigma \cup \{\varepsilon\}$, *то* $\{a\} \in \boldsymbol{S}$, $\{\langle a, b\rangle\} \in \boldsymbol{D}$;

(2) *если* $X, Y \in \boldsymbol{S}$, $P, Q \in \boldsymbol{D}$, *то*

(2a) $X \cup Y \in \boldsymbol{S}$, $P \cup Q \in \boldsymbol{D}$;

(2b) $P \otimes X = \{p_1 x p_2 \mid \langle p_1, p_2\rangle \in P,\ x \in X\} \in \boldsymbol{S}$;

(2c) $P \odot Q = \{\langle p_1 q_1, q_2 p_2\rangle \mid \langle p_1, p_2\rangle \in P,\ \langle q_1, q_2\rangle \in Q\} \in \boldsymbol{D}$;

(2d) $P^+ = P \cup P \odot P \cup P \odot P \odot P \cup \ldots \in \boldsymbol{D}$.

Язык L *над* Σ *является линейным тогда и только тогда, когда* $L \in \mathbf{S}$.

Доказательство. Чтобы доказать «тогда», достаточно показать, что для каждых $L \in \mathbf{S}$, $P \in \mathbf{D}$ и σ — буквы не из Σ — языки L и $P \otimes \{\sigma\}$ являются линейными языками. Это, конечно, верно для $L = \{a\}$ и $P = \{\langle a, b\rangle\}$, где a и b принадлежат $\Sigma \cup \{\varepsilon\}$. Согласно (2a)—(2d), если это имеет место для X, Y из $\mathbf{S}$ и P, Q из $\mathbf{D}$, то это верно и для $X \cup Y$, $P \cup Q$, $P \otimes X$, $P \odot Q$, P^+ поэтому и для всех $X \in \mathbf{S}$, $P \in \mathbf{D}$.

Для доказательства «только тогда» достаточно повторить вторую часть доказательства теоремы 2.7 и заметить, что если G является линейной грамматикой, то все множества $\Phi_{i,k}$ представляются в виде $\Phi_{i,k} = \bigcup_{j=1}^{k} P_j^{(i,k)} \otimes \{\sigma_j\} \cup L_{i,k}$, где

$$P_j^{(i,k)} \in \boldsymbol{D},\ L_{i,k} \in \boldsymbol{S} \text{ и } \phi_{i,i}^{\sigma_i} = \bigcup_{j=1}^{i-1} P_j^{(i)} \otimes \{\sigma_j\} \cup L_i,$$

где опять-таки $P_j^{(i)} \in \boldsymbol{D}$ и $L_i \in \boldsymbol{S}$. Таким образом, $L(G) = \Phi_{1,1}^{\sigma_1} \in \boldsymbol{S}$.

Аналогичную характеристику можно дать классу полулинейных языков.

Определение 4.2. Грамматика $G = \langle \Sigma_N, \Sigma, P, \sigma\rangle$ называется *полулинейной*, если для каждого $A \in \Sigma_N$ и $x \in (\Sigma \cup \Sigma_N)^*$ из $A \overset{*}{\Rightarrow} x$ следует, что A входит в x не более одного раза. Язык L называется *полулинейным*, если L порождается некоторой полулинейной грамматикой.

---

[1]) См. также статью Мак-Вертера «Подстановочные выражения», помещенную в наст. сборник (стр. 127—136).) — *Прим. перев.*

Существует много различных названий для полулинейных языков и грамматик: множества стандартно согласованного выбора [8], нерасширяющиеся грамматики, языки с ограниченным выводом [4], суперлинейные грамматики и языки [2]. Мотивом для введения еще одного названия служит лемма 4.4, которая оказывается также полезной для получения дальнейших результатов.

Определение 4.3. Пусть $G = \langle \Sigma_N, \Sigma_T, P, \sigma \rangle$ — контекстно-свободная грамматика, а $\equiv$ есть отношение на P, определяемое так: $(A \to \alpha) \equiv (B \to \beta)$ тогда и только тогда, когда $A = B$ или $A \overset{\cdot}{\Rightarrow} u_1 B v_1$ и $B \overset{\cdot}{\Rightarrow} u_2 A v_2$ для некоторых u_1, u_2, v_1, v_2 из $(\Sigma_N \cup \Sigma_T)^*$. $\equiv$ является *отношением эквивалентности* на P. Классы отношений эквивалентности на P относительно $\equiv$ называются *грамматическими уровнями* G.

Замечание. Грамматический уровень G_0 КСГ G является множеством контекстно-свободных правил грамматик G. Можно рассматривать G_0 как КСГ без начального символа, переменные которой суть символы из левых частей правил G_0, а все другие буквы, встречающиеся в правилах G_0, суть терминальные символы относительно G_0. Таким образом, ни один терминальный символ G не может быть терминальным символом G_0.

Если мы рассматриваем только приведенные КСГ, т. е. грамматики, в которых каждый нетерминальный символ достижим из начального символа и каждый нетерминальный символ порождает непустое множество терминальных слов, то, используя определения 4.3 и 4.2, легко можно доказать следующую лемму.

Лемма 4.4. *Грамматика G является полулинейной тогда и только тогда, когда каждый грамматический уровень G_0 грамматики G является линейной грамматикой (относительно нетерминалов G_0).*

Доказательство. Если G не является полулинейным языком, то в G существует нетерминал A, такой, что $A \overset{\cdot}{\Rightarrow} uAwAv$ для некоторых слов u, w, v, и поэтому грамматический уровень, содержащий правила с A в левой части, не является линейным (т. е. линейной грамматикой). С другой стороны, если некоторый грамматический уровень грамматики G не является линейным, то G содержит правило $A \to uBwCv$, где B и C суть нетерминалы, такие, что правила с B или C в левой части находятся на том же грамматическом уровне, что и $A \to uBwCv$. Следовательно, существуют слова u_1, v_1, u_2, v_2, такие, что

$B \overset{*}{\Rightarrow} u_1 A v_1$, $C \overset{*}{\Rightarrow} u_2 A v_2$ и поэтому G не является полулинейной грамматикой.

Из приведенной леммы вытекает также хорошо известный факт о том, что класс полулинейных языков замкнут относительно суперпозиций. Теорема 4.5 является аналогом теоремы 4.1.

Теорема 4.5. *Пусть* Σ — *алфавит. Определим* $\boldsymbol{S} \subset 2^{\Sigma^*}$, $\boldsymbol{D} \subset 2^{\Sigma^* \times \Sigma^*}$ *как наименьшие классы, для которых выполнены условия* (1)—(2d) *теоремы* 4.1 *и условие*

$$\langle X, Y \rangle = \{\langle x, y \rangle \mid x \in X, \ y \in Y\} \in \boldsymbol{D}. \tag{2e}$$

Язык L *является полулинейным языком над* Σ *тогда и только тогда, когда* $L \in \boldsymbol{S}$.

Доказательство. Заметим, что из (2e) следует, что при $X, Y \in \boldsymbol{S}$ язык XY также принадлежит $\boldsymbol{S}$. Доказательство «тогда» аналогично доказательству теоремы 4.1. Далее, пусть L_0 — линейный язык над $\Sigma' \supset \Sigma$, а L — язык, получающийся из L_0 применением подстановки ϕ, где $\phi(a) = \{a\}$ для $a \in \Sigma$ и $\phi(a) \in \boldsymbol{S}$ для $a \notin \Sigma$. Из теоремы 4.1 и (2e) следует, что L принадлежит $\boldsymbol{S}$. Таким образом, $\boldsymbol{S}$ содержит все языки над Σ, которые получаются из линейных языков подстановкой, а потому $\boldsymbol{S}$ содержит все полулинейные языки над Σ.

Если условие (2e) заменить на

$$XY \in \boldsymbol{S}, \tag{2e}'$$

то $L \in \boldsymbol{S}$ тогда и только тогда, когда L является ультралинейным языком.

Ультралинейным языком [3] называется язык, порожденный нетерминально ограниченной грамматикой [1]. КГС $G = \langle \Sigma_N, \Sigma_T, P, \sigma \rangle$ является нетерминально ограниченной, если существует натуральное число k, такое, что из $\sigma \overset{*}{\Rightarrow} x$ следует, что x содержит не более чем k нетерминальных символов. Легко доказывается, что контекстно-свободная грамматика G нетерминально ограничена тогда и только тогда, когда каждый грамматический уровень грамматики G является линейным. Кроме того, если $A \to xByCz$ является правилом G с переменными B и C, то никакое правило с B или C в левой части не принадлежит тому же грамматическому уровню G, которому принадлежит правило $A \to xByCz$. Используя этот факт и теорему 4.1, мы без труда можем получить наш результат.

Введя новую операцию на множестве пар слов, можно дать более изящную характеристику полулинейных языков.

Теорема 4.6. *Пусть* Σ *есть алфавит. Определим* $\boldsymbol{D}_0 \subset 2^{\Sigma^* \times \Sigma^*}$ *как наименьший класс, удовлетворяющий следующим условиям:*

(i) *если* $a \in \Sigma$, *то* $\{\langle a, \varepsilon\rangle\} \in \boldsymbol{D}_0$, $\{\langle \varepsilon, \varepsilon\rangle\} \in \boldsymbol{D}_0$;

(ii) *если* $P, Q \in \boldsymbol{D}_0$, *то*

(iia) $P \cup Q$, $P \odot Q$, $P^+ \in \boldsymbol{D}_0$;

(iib) $P \circledast Q = \{\langle p_1 p_2, q_1 q_2\rangle \mid \langle p_1, p_2\rangle \in P, \langle q_1, q_2\rangle \in Q\} \in \boldsymbol{D}_0$.

Пусть $\boldsymbol{S}_0 = \{P \otimes \{\varepsilon\}, P \in \boldsymbol{D}_0\}$. *Тогда* $L \in \boldsymbol{S}_0$ *в том и только том случае, когда* L *является полулинейным языком над* Σ.

Доказательство. Достаточно показать, что $\boldsymbol{S} = \boldsymbol{S}_0$, $\boldsymbol{D} = \boldsymbol{D}_0$, где $\boldsymbol{S}$ и $\boldsymbol{D}$ определены, как в теореме 4.5. Для доказательства включений $\boldsymbol{S} \subseteq \boldsymbol{S}_0$, $\boldsymbol{D} \subseteq \boldsymbol{D}_0$ достаточно показать, что условия (1) — (2e) теоремы 4.5 выполняются для $\boldsymbol{S}_0$, $\boldsymbol{D}_0$. Чтобы убедиться в этом, достаточно установить, что если $X, Y \in \boldsymbol{S}_0$ и $a, b \in \Sigma \cup \{\varepsilon\}$, то $\langle a, b\rangle \in \boldsymbol{D}_0$, $X \cup Y \in \boldsymbol{S}_0$, $\langle X, Y\rangle \in \boldsymbol{D}_0$. Но $\langle a, b\rangle = \langle a, \varepsilon\rangle \circledast \langle b, \varepsilon\rangle \in \boldsymbol{D}_0$ и если $X, Y \in \boldsymbol{S}_0$, то $X = P \otimes \{\varepsilon\}$, $Y = Q \otimes \{\varepsilon\}$ с $P, Q \in \boldsymbol{D}_0$, а потому $X \cup Y = (P \cup Q) \otimes \{\varepsilon\} \in \boldsymbol{S}_0$, $\langle X, Y\rangle = P \odot Q \in \boldsymbol{D}_0$. Аналогично доказывается, что $\boldsymbol{S}_0 \subseteq \boldsymbol{S}$, $\boldsymbol{D}_0 \subseteq \boldsymbol{D}$. Теорема доказана.

СПИСОК ЛИТЕРАТУРЫ

1. Banerji R. B., Phrase structure languages, finite machines and channel capacity, *Information and Control*, **6** (1963), 153—162.
2. Brzozowski Y. A., Regular-Like Expressions for Some Irregular Languages, IEEE Conference record of Ninth Annual Symposium on Switching and Automata Theory, 1968, p. 278—280.
3. Ginsburg S., Spanier E. H., Finite-turn pushdown automata, *SIAM J. Control*, **4** (1966), 429—453.
4. Ginsburg S., Spanier E. H., Derivation-Bounded Languages, *J. Comput. System Sci.*, **2** (1968), 228—250.
5. Greibach S. A., Full AFL's and Nested Iterated Substitution, IEEE Conference record of Tenth Annual Symposium on Switching and Automata Theory, 1969, p. 222—230.
6. Gruska J., Generalization of Regular Sets, Abstracts of the Mathematics Congress, Moscow, 1966.
7. Gruska J., On a Classification of context-free grammars, *Kybernetika Prague*, **3** (1967), 22—29.
8. Yntema M. K., Inclusion relations among families of context-free languages, *Information and Control*, **10** (1967), 572—597.

Подстановочные выражения[1])

И. П. Мак-Вертер

1. ВВЕДЕНИЕ

Хорошо известно, что всякий регулярный язык R над Σ может быть определен как первая координата единственного решения системы уравнений вида

$$X_i = C_{i1}X_1 \cup \ldots \cup C_{in}X_n \cup F_i, \qquad i = 1, \ldots, n,$$

где каждое C_{ij} и F_i являются конечными подмножествами Σ^*, а λ (пустое слово) не входит ни в одно из множеств C_{ij} [1]. Эти уравнения, которые непосредственно получаются из автомата, распознающего R, можно использовать для того, чтобы найти регулярное выражение для R, повторно решая уравнение вида $X = A \cdot X \cup B$, которое имеет единственное решение $X = A^*B$, если A не содержит λ.

Хорошо известно также, что каждый бесконтекстный[2]) язык L может быть определен как первая координата наименьшего решения системы уравнений, которые мы будем называть контекстно-свободными уравнениями, вида

$$X_i = f_i(X_1, \ldots, X_n) = \bigcup_l w_{il} \cup F_i, \qquad i = 1, \ldots, n,$$

где $w_{ij} = w_{ij0}X_{ij1}w_{ij1} \ldots X_{ijm}w_{ijm}$, причем каждое $w_{ijk} \in \Sigma^*$ для всех i, j, $X_{ijk} = X_l$ для некоторого l (зависящего от i, j, k), а F_i суть конечные подмножества Σ^* [5].

Цель этой статьи состоит в изучении формальных выражений, которые обозначают решения таких уравнений. Будут использоваться два конечных непустых алфавита: $\Sigma = \{a_1, \ldots, a_m\}$ — «основной» алфавит и $\Delta = \{\delta_1, \ldots, \delta_n\}$ — «вспомогательный» алфавит.

Определение 1. (а) *Подстановка.* Для подмножеств L и L' множества $(\Sigma \cup \Delta)^*$ определим $L[\delta \leftarrow L'] = \{w_0u_1w_1 \ldots u_nw_n \mid$ каждое $u_i \in L'$, $w_0\delta w_1 \ldots \delta w_n \in L$ и δ не входит ни в какое $w_i\}$.

---

[1]) McWhirter I. P., Substitution expressions, *J. Comp. Syst. Sci.*, 5 (1971), 629—637.

[2]) Такие языки называются также контекстно-свободными. — *Прим. перев.*

(b) Звездная (или *итеративная*) *подстановка*. Для $L \subseteq (\Sigma \cup \Delta)^*$ определим $L^{*\delta} = \bigcup_{n \geqslant 0} (L_n)$, где

$$(L)_0 = \{\delta\}, \text{ а } (L)_{n+1} = (L)_n \cup L\,[\delta \leftarrow (L)_n].$$

Заметим, что если L и L' суть подмножества Σ^*, можно проверить, что $(L\delta)\,[\delta \leftarrow L'\delta] = L \cdot L'\delta$ и $(L\delta)^{*\delta} = L^*\delta$. Таким образом, при этих ограничениях подстановка и звездная подстановка аналогичны конкатенации и итерации Клини (или, как иногда говорят, операция «звезда»), соответственно. Эта аналогия может быть усилена следующим образом. Перепишем систему контекстно-свободных уравнений с помощью вспомогательного алфавита в виде

$$X_i = \bigcup_j v_{ij}\,[\delta_1 \leftarrow X_1] \ldots [\delta_n \leftarrow X_n] \cup F_i,$$

где $v_{ij} = w_{ij0}\delta_{ij1}w_{ij1} \ldots \delta_{ijm}w_{ijm}$ для всех i.

Теперь X_i характеризуется двумя конечными множествами $W_i = \bigcup_j v_{ij}$ и F_i. В этом случае, когда W_i является подмножеством $\Sigma^*\Delta$, эти уравнения приводят к регулярному случаю, когда подстановка языка вместо вспомогательного символа приводит к конкатенации двух языков. Далее, аналогия между операциями конкатенации и подстановки распространяется на решение уравнений. Будет показано, что при подходящих ограничениях на A и B уравнение

$$X = A\,[\delta \leftarrow X] \cup B$$

имеет единственное решение $A^{*\delta}\,[\delta \leftarrow B]$. Повторно решая это уравнение, можно получать решения системы контекстно-свободных уравнений, использующие операторы подстановки. Такие выражения аналогичны регулярным.

Примеры. (a) $\{a^n b^n \mid n \geqslant 0\}$. Уравнение $X = aXb \cup \lambda$ приводится, в наших обозначениях, к виду

$$X = (a\delta b)\,[\delta \leftarrow X] \cup \lambda$$

и имеет единственное решение $(a\delta b)^{*\delta}\,[\delta \leftarrow \lambda]$.

(b) Язык Дикка над a и b. Уравнение $A = (\delta a \delta b \delta)\,[\delta \leftarrow A] \cup \lambda$ имеет единственное решение $(\delta a \delta b \delta)^{*\delta}\,[\delta \leftarrow \lambda]$.

Предполагается, что читатель знаком с основными результатами теории формальных языков [9].

2. СВОЙСТВА ОПЕРАЦИЙ ПОДСТАНОВКИ

Ниже приводится несколько алгебраических свойств простой и звездной подстановок. Доказательства их вполне просты, подробности можно найти в [11].

P1. $A[\delta \leftarrow \{\delta\}] = \{\delta\}[\delta \leftarrow A] = A$.

(Вспоминая аналогию между конкатенацией и подстановкой, видим, что здесь δ соответствует λ).

P2. Ассоциативность:

$$A[\delta \leftarrow B[\delta \leftarrow C]] = (A[\delta \leftarrow B])[\delta \leftarrow C].$$

P3. Правая дистрибутивность:

$$(A \cup B)[\delta \leftarrow C] = A[\delta \leftarrow C] \cup B[\delta \leftarrow C].$$

P4. $A[\delta \leftarrow A^{*\delta}] \cup \{\delta\} = A^{*\delta}$

(P4 снова показывает аналогию между δ и λ.)

P5. Коммутативность: если α и β не входят ни в одно слово из B и A соответственно, то

$$C[\alpha \leftarrow A][\beta \leftarrow B] = C[\beta \leftarrow B][\alpha \leftarrow A].$$

P6. Если β не входит в слова B, то

$$A[\alpha \leftarrow B][\beta \leftarrow C[\alpha \leftarrow B]] = A[\beta \leftarrow C][\alpha \leftarrow B].$$

P7. Если α не входит в слова из B, то

$$(A[\beta \leftarrow B])^{*\alpha} = A^{*\alpha}[\beta \leftarrow B].$$

Эти свойства подстановочных операторов подразумеваются при многих доказательствах. Теперь мы докажем, что при подходящих ограничениях $A^{*\delta}[\delta \leftarrow B]$ есть единственное решение уравнения $X = A[\delta \leftarrow X] \cup B$.

Теорема 1. *Если* $X = A[\delta \leftarrow X] \cup B$, *то* $A^{*\delta}[\delta \leftarrow B]$ *есть наименьшее решение для* X. *Если, кроме того,* $A \cap \delta^* = \varnothing$, *то это решение единственно.*

Доказательство. Алгебраические преобразования по правилам P1 — P4 покажут, что $A^{*\delta}[\delta \leftarrow B]$ есть наименьшее решение уравнения $X = A[\delta \leftarrow X] \cup B$.

Заметим, что если $A \cap \delta^* = \varnothing$, то каждое слово из A, содержащее δ, содержит также и другие буквы. Отсюда следует, что любое слово из $A[\delta \leftarrow X]$ — это или слово из A, или оно длиннее кратчайшего слова из X. Пользуясь этим, можно проверить, что $A^{*\delta}[\delta \leftarrow B]$ есть единственное решение.

3. ХАРАКТЕРИСТИКА БЕСКОНТЕКСТНЫХ ЯЗЫКОВ

Результат теоремы 1 может быть использован для решения некоторых систем контекстно-свободных уравнений вида

$$X_i = W_i[\delta_1 \leftarrow X_1] \ldots [\delta_n \leftarrow X_n] \cup F_i, \qquad i = 1 \ldots n.$$

Каждый бесконтекстный язык может быть определен в нормальной форме Грейбах [6]. Контекстно-свободные уравнения, полученные из этой формы, обладают свойством $W_i \cap \Delta^* = \varnothing$, и это позволяет нам искать единственное решение системы.

Подробнее, по теореме 1, $(W_n[\delta_1 \leftarrow X_1] \ldots [\delta_{n-1} \leftarrow X_{n-1}]^{*\delta_n}[\delta_n \leftarrow F_n]$ есть единственное выражение для X_n через $X_1, X_2 \ldots X_{n-1}$. Используя свойства P1 и P7, приведем его к виду

$$W_n^{*\delta_n}[\delta_n \leftarrow F_n][\delta_1 \leftarrow X_1] \ldots [\delta_{n-1} \leftarrow X_{n-1}].$$

Подставляя это выражение в уравнение для X_i и используя те же свойства, получим

$$X_i = \left(W_i\left[\delta_n \leftarrow W_n^{*\delta_n}\right][\delta_n \leftarrow F_n]\right)[\delta_1 \leftarrow X_1] \ldots [\delta_{n-1} \leftarrow X_{n-1}] \cup F_i.$$

Повторяя этот процесс, получим описание бесконтекстного языка в терминах объединения, подстановки и звездной подстановки на конечных множествах.

Для бесконтекстных языков, как и для регулярных, удобно пользоваться задающими их выражениями, поэтому введем следующие определения.

Определение 2. Множество $\mathscr{C}_{\Sigma,\Delta}$ подстановочных выражений над конечными алфавитами Σ и Δ есть наименьшее множество выражений, построенных по следующим правилам:

1. Базис: $a_1, \ldots, a_n$ принадлежат $\mathscr{C}_{\Sigma,\Delta}$ для всех положительных n, и всех $a_i \in (\Sigma \cup \Delta)$, λ и $\varnothing$ принадлежат $\mathscr{C}_{\Sigma,\Delta}$.
2. Если E и E' принадлежат $\mathscr{C}_{\Sigma,\Delta}$ и $\delta \in \Delta$, то $(E \cup E')$, $E[\delta \leftarrow E']$ и $E^{*\delta}$ принадлежат $\mathscr{C}_{\Sigma,\Delta}$.

Множество $\mathscr{C}$ подстановочных выражений определяется как $\mathscr{C} = U_{\Sigma,\Delta}\mathscr{C}_{\Sigma,\Delta}$. Отображение $|\,|: \mathscr{C} \rightarrow$ (множество языков) определяется рекурсивно:

1) $|\,a_1, \ldots, a_n\,| = \{a_1, \ldots, a_n\}$, $|\,\lambda\,| = \{\lambda\}$, $|\,\varnothing\,| = \varnothing$;

2) $|\,(E \cup E')\,| = |\,E\,| \cup |\,E'\,|$;

3) $|\,E[\delta \leftarrow E']\,| = |\,E\,|[\delta \leftarrow |\,E'\,|]$;

4) $|\,E^{*\delta}\,| = |\,E\,|^{*\delta}$.

$|E|$ обозначает язык, выражаемый E, $|\mathscr{C}|$ обозначает область значений $\mathscr{C}$.

Заметим, что определение 2 для множества языков $|\mathscr{C}_{\Sigma,\Delta}|$ можно изменить так, что базис станет конечным, а именно $(\Sigma \cup \Delta \cup \Delta^2 \cup \{\lambda, \varnothing\})$, при условии, что Δ содержит не меньше двух элементов, потому, что любое слово вида aw может быть выражено как $\beta\alpha[\beta \leftarrow a][\alpha \leftarrow w]$, где α выбрано отличным от a.

Отсюда следует, что каждый бесконтекстный язык может быть задан при помощи подстановочного выражения. С другой стороны, можно доказать, что $|\mathscr{C}|$ замкнуто относительно пересечения с регулярными языками, откуда $|\mathscr{C}|$ есть супер-AFL [1]) [7]. Из этого немедленно следует, что каждый бесконтекстный язык содержится в $|\mathscr{C}|$. Возможно, существует более простой способ решения контекстно-свободных уравнений.

Чтобы получить обратный результат, т. е. что каждое подстановочное выражение обозначает бесконтекстный язык, надо только заметить, что множество бесконтекстных языков замкнуто относительно объединения, подстановки [2] и итеративной подстановки [10]. Отсюда вытекает

Теорема 2. *Некоторый язык L бесконтекстен тогда и только тогда, когда существует выражение E, такое, что $|E| = L$.*

4. РЕГУЛЯРНАЯ И ЛИНЕЙНАЯ МОДЕЛИ

Как было установлено выше, если δ не входит в L или L', то $(L\delta)[\delta \leftarrow L'\delta] = L \cdot L'\delta$ и $(L\delta)^{*\delta} = L^*\delta$. Поэтому чтобы получить модель регулярных языков, мы определим $\mathscr{R}_{\Sigma,\delta}$ как наименьшее множество выражений, построенных по следующим правилам:

1. $a\delta$ принадлежит $\mathscr{R}_{\Sigma,\delta}$ для всех $a \in \Sigma$, λ и $\varnothing$ принадлежат $\mathscr{R}_{\Sigma,\delta}$;

2. Если E и E' принадлежат $\mathscr{R}_{\Sigma,\delta}$, то $(E \cup E')$, $E[\delta \leftarrow E']$ и $E^{*\delta}$ принадлежат $\mathscr{R}_{\Sigma,\delta}$.

---

[1]) Супер-AFL есть семейство языков, содержащее хотя бы одно не унитарное множество (множество, в котором есть слова длиной больше 1) и замкнутое относительно пересечения с регулярными языками, объединения с унитарными множествами и гнёздной итеративной подстановки. Определение гнездно-итеративной подстановки: τ — подстановка, $\tau : \Sigma^* \to 2^{(\bigcup \Sigma_a)^*}$; доопределим τ на $\bigcup_{a \in \Sigma} \Sigma_a$ так: $\tau(b) = b$, если $b \in \left(\bigcup_{a \in \Sigma} \Sigma_a\right) - \Sigma$; пусть $\tau^1(L) = \tau(L)$, а $\tau^{n+1}(L) = \tau(\tau^n(L))$. Тогда если $\bigcup a \in \left(\Sigma \bigcup_{a \in \Sigma_a} \bigcup \Sigma_a\right)$, $a \in \tau(a)$, то τ^∞ — есть гнездно-итеративная подстановка, где $\tau^\infty(L) = \bigcup_n \tau^n(L)$. Свойство супер-AFL: каждое супер-AFL содержит все бесконтекстные языки (теорема 2.2, [7]). — *Прим. перев.*

Пусть $\|E\| = |E|[\delta \leftarrow \lambda]$ для $E \in \mathscr{R}_{\Sigma,\delta}$ и $\|\mathscr{R}_{\Sigma,\delta}\| = \{\|E\| \mid E \in \mathscr{R}_{\Sigma,\delta}\}$. Используя упоминавшуюся аналогию между операторами подстановки и конкатенации, можно легко доказать, что $\|\mathscr{R}_{\Sigma,\delta}\|$ есть множество всех регулярных языков.

Подобным же образом для получения подстановочной модели линейных языков мы определим $\mathscr{L}_{\Sigma,\delta}$ как и $\mathscr{R}_{\Sigma,\delta}$, но правило 1 заменим на 1′: $a\delta$ и δa принадлежат $\mathscr{L}_{\Sigma,\delta}$ для всех $a \in \Sigma$, λ и $\varnothing$ принадлежат $\mathscr{L}_{\Sigma,\delta}$. Так как каждое слово в любом языке из $|\mathscr{L}_{\Sigma,\delta}|$ имеет только одно вхождение δ, то ленту, на которой оно записано, можно «сложить» в месте этого вхождения, как показано в [3]. Так как линейные языки выражаются регулярными операциями над сложенными лентами, доказательство того, что $\|\mathscr{L}_{\Sigma,\delta}\|$ есть множество всех линейных языков над Σ сводится к доказательству, что $\|\mathscr{R}_{\Sigma\times\Sigma,\delta}\|$ есть множество всех регулярных языков над $(\Sigma \times \Sigma)$.

Левая дистрибутивность подстановки над объединением свойственна языкам из $|\mathscr{L}_{\Sigma,\delta}|$ и не свойственна всем бесконтекстным языкам. Формально, $L1$ (левая дистрибутивность): если A принадлежит $|\mathscr{L}_{\Sigma,\delta}|$, то

$$A[\delta \leftarrow (B \cup C)] = A[\delta \leftarrow B] \cup A[\delta \leftarrow C].$$

5. МОДЕЛЬ ГРУСКИ

Другую модель выражений для бесконтекстных языков дал Груска в [8]. Он использовал операции объединения, конкатенации и другого типа итеративной подстановки, записываемой так: L^{δ}, где $L^{\delta} = L^{*\delta}[\delta \leftarrow \varnothing]$. В то время как в его системе конкатенация вводится как простая операция, в нашей модели ее можно заменить подстановкой. Введением некоторых ограничений на базис определения $\mathscr{C}_{\Sigma,\Delta}$ (как было показано в нашей модели) мы можем получить полную характеристику линейных и регулярных языков. В обоих случаях необходим только один вспомогательный символ. Хотя подход Груски дает простую модель для регулярных языков, в своей статье Груска не дал полной характеристики множества линейных языков.

6. ИЕРАРХИЯ ПОДСТАНОВОЧНОЙ ЗВЕЗДНОЙ ВЫСОТЫ

Понятие звездной высоты $\mathrm{sh}(R)$ используется в теории регулярных языков как мера сложности языка. Аналогично можно определить подстановочную звездную высоту для контекстно-свободных языков.

Определение 3. Подстановочная звездная высота, обозначаемая sh_s, для некоторого выражения из $\mathscr{C}$ определяется рекурсивно:

1) $\mathrm{sh}_s(w) = 0$ для любых $w \in (\Sigma \cup \Delta)^*$, $\mathrm{sh}_s(\varnothing) = 0$;

2) $\mathrm{sh}_s((E \cup E')) = \max\{\mathrm{sh}_s(E), \mathrm{sh}_s(E')\}$;

3) $\mathrm{sh}_s(E[\delta \leftarrow E']) = \max\{\mathrm{sh}_s(E), \mathrm{sh}_s(E')\}$;

4) $\mathrm{sh}_s(E^{*\delta}) = \mathrm{sh}_s(E) + 1$.

Мы распространим sh_s на бесконтекстные языки, положив $\mathrm{sh}_s(L) = \min\{\mathrm{sh}_s(E) \mid |E| = L\}$.

Из-за упоминавшейся связи между конкатенацией и операторами подстановки может показаться, что для регулярных языков $\mathrm{sh}_s(R)$ совпадает с $\mathrm{sh}(R)$. Это неверно, так как язык $(0 \cup 10^*1)^*$, имеющий регулярную звездную высоту 2, записывается как

$$(\delta\delta)^{*\delta}[\delta \leftarrow (0 \cup 1\,\delta 1)][\delta \leftarrow (0\delta)^{*\delta}][\delta \leftarrow \lambda] \cup \lambda,$$

а это выражение имеет подстановочную звездную высоту 1. На самом деле, подстановочная звездная высота всех регулярных языков равна 0 или 1. Однако при доказательстве того, что для всех n существуют языки L_n с подстановочной звездной высотой n, можно пользоваться некоторыми результатами, установленными для регулярной звездной высоты.

Пусть $W = \{w\delta w^R \mid w \in \Sigma^*\}$. Если L — подмножество W и $L_R = \{w \mid w\delta w^R \in L\}$, то L_R — регулярный язык тогда и только тогда, когда L бесконтекстен по теоремам 2.1 и 7.2 из [2]. Мы покажем, что $\mathrm{sh}_s[L_R]$ совпадает с $\mathrm{sh}(L)$.

Лемма 3. *Пусть E — некоторое подстановочное выражение для подмножества W. Тогда существует выражение E' из $\mathscr{L}_{\Sigma,\delta}$, такое, что $|E'| = |E|$ и $\mathrm{sh}_s(E') = \mathrm{sh}_s(E)$.*

Прежде чем доказывать лемму, установим несколько предложений, доказательства которых вполне просты. Подробности их можно найти в [11].

Определение 4. Выражение E называется приведенным, если для всех E' подвыражений E имеет место:

1) Если $E' = E_0[\delta \leftarrow E_1]$, то δ входит в некоторые слова $|E_0|$ и

2) если $E \neq \varnothing$, то $|E'| \neq \varnothing$.

Предложение 1. *Если E принадлежит $\mathscr{C}_{\Sigma,\Delta}$, то существует некоторое приведенное выражение E' из $\mathscr{C}_{\Sigma,\Delta}$, выражающее $|E|$, и такое, что $\mathrm{sh}_s(E') \leqslant \mathrm{sh}_s(E)$.*

Предложение 2. *Если E — некоторое приведенное выражение из $\mathscr{C}_{\Sigma,\Delta}$, $|E| \neq \varnothing$ и w — принадлежит $|E'|$, где $|E'|$ — некоторое подвыражение E, то существуют вспомогательные символы $\alpha_0, \ldots, \alpha_m$, не обязательно из Δ, слово u_0, в которое*

входит α_0, *и слова* $u_1, \ldots, u_m$ *из* $(\Sigma \cup \Delta)^*$, *такие, что* $u_0[\alpha_0 \leftarrow w] \times [\alpha_1 \leftarrow u_1] \ldots [\alpha_m \leftarrow u_m]$ *принадлежат* $|E|$. *Если же* $|E|$ — *подмножество* Σ^*, *можно даже предположить, что* u_0 *из* $\Sigma^*\alpha_0\Sigma^*$, *что* $u_1, \ldots, u_m$ *из* Σ^*, *и* $\alpha_0 \ldots \alpha_m$ *из* Δ.

Доказательство леммы 3. Пусть E выражает некоторое подмножество из W и принадлежит $\mathscr{C}$. Можно предположить, что E — выражение из $\mathscr{C}_{\Sigma \cup \delta, \Delta}$, где $(\Sigma \cup \delta) \cap \Delta = \varnothing$. Сначала докажем, что можно считать каждое слово, являющееся подвыражением E, принадлежащим $\Sigma^*(\Delta \cup \delta)\Sigma^*$.

Пусть слово w — подвыражение E. Выберем u из $(\Sigma \cup \delta \cup \Delta)^*$, $u_1, \ldots, u_m$ из $(\Sigma \cup \delta)^*$, $\alpha_1 \ldots, \alpha_m$ и β из Δ так, что $u[\beta \leftarrow w] \times [\alpha_1 \leftarrow u_1] \ldots [\alpha_m \leftarrow u_m]$ принадлежит $|E|$. Предположим, что некоторый символ α_i входит в w, но δ не входит в u_i. По P5 можно считать, что $i = m$. Теперь рассмотрим слово $u[\beta \leftarrow w] \times [\alpha_1 \leftarrow u_1] \ldots [\alpha_{m-1} \leftarrow u_{m-1}]$. Это слово должно иметь вид $w_0\alpha_m w_1 \ldots \alpha_m w_k\, \delta v_0 \alpha_m v_1 \ldots \alpha_m v_j$, где каждое w_i и v_i принадлежат Σ^*. Мы утверждаем, что только одно слово можно подставлять вместо α_m. Чтобы доказать это, предположим, что α_m можно заменять и на u'. Тогда оба слова $w_0 u_m w_1 \ldots u_m w_k\, \delta v_0 u_m v_1 \ldots \ldots u_m v_{j-1} u_m v_j$ и $w_0 u_m w_1 \ldots u_m w_k\, \delta v_0 u_m v_1 \ldots u_m v_{j-1} u' v_j$ принадлежат $|E|$ и, следовательно, W. Отсюда следует, что $v_0 u_m v_1 \ldots \ldots u_m v_{j-1} u_m v_j = (w_0 u_m w_1 \ldots u_m w_k)^R = v_0 u_m v_1 \ldots u_m v_{j-1} u' v_j$ и $u' = u_m$.

Поскольку вместо α_m можно подставлять только слово u_m, то, подставив это слово прямо в w, получим w', и язык, выражаемый E, совпадает с $|E|$, где w заменено на w'. Поэтому считаем, что в каждом слове w, входящем как подвыражение в E, каждый символ из Δ можно заменить некоторым словом, таким, в котором есть символ δ. Отсюда следует, что в w не может быть больше одного вхождения буквы из $\{\delta\} \cup \Delta$, так как δ входит в каждое слово из W не больше одного раза.

Теперь покажем, как по индукции построить E' из $\mathscr{L}_{\Sigma, \delta}$, выражающее язык $|E|$ и имеющее такую же подстановочную звездную высоту. Чтобы сделать это, найдем, что каждое выражение E, порожденное словами с одним вхождением вспомогательного символа или δ, может быть записано как объединение выражений таких подмножеств $|E|$, которые содержат специальный нетерминал. Если $w\alpha w'$ входит в E, где $\alpha \in \Delta \cup \{\delta\}$, то w и w' входят в Σ^*; отсюда $|w\alpha w'| = |w\, \delta w'|[\delta \leftarrow \alpha]$. Предположим, что E_i, подвыражение E, обозначает тот же язык, что и $E_i' = \bigcup_{\alpha} E_{i\alpha}[\delta \leftarrow \alpha]$ для α из $\Delta \cup \{\delta, \gamma\}$, где каждое $E_{i\alpha} \in \in \mathscr{L}_{\Sigma, \delta}$, $|E_{i\gamma}| \in \Sigma^*$ и $\mathrm{sh}_s(E_i') = \mathrm{sh}_s(E_i)$; $i = 0, 1$.

Очевидно, такое же выражение можно найти для $(E_0 \cup E_1)$. Пользуясь тем, что $|E_{0\beta}[\delta \leftarrow \beta]|[\alpha \leftarrow L] = |E_{0\beta}[\delta \leftarrow \beta]|$, если

$\alpha \neq \beta$, для любого языка L (так как α не входит в слова из $|E_{0\beta}[\delta \leftarrow \beta]|$) можно получить такое же выражение для $E_0[\alpha \leftarrow E_1]$. Сократив это выражение, используя правую и левую дистрибутивность и заменив выражение вида $[\delta \leftarrow \alpha][\alpha \leftarrow E']$ на $[\delta \leftarrow E']$, получаем

$$|E_0[\alpha \leftarrow E_1]| = \left| \bigcup_{\beta \neq \alpha} (E_{0\beta} \cup E_{0\alpha}[\delta \leftarrow E_{1\beta}])[\delta \leftarrow \beta] \cup E_{0\alpha}[\delta \leftarrow E_{1\alpha}][\delta \leftarrow \alpha] \right|.$$

И таким же образом: $|E_0^{*\alpha}| = \left| \bigcup_{\beta \neq \alpha} E_{0\alpha}^{*\delta}[\delta \leftarrow E_{0\beta}][\delta \leftarrow \beta] \cup E_{0\alpha}^{*\delta}[\delta \leftarrow \alpha] \right|$.

Пользуясь этим способом, можно найти выражение E_δ из $\mathscr{L}_{\Sigma,\Delta}$, такое, что $|E| = |E_\delta|$ и $\mathrm{sh}_s(E) = \mathrm{sh}_s(E_\delta)$. Доказательство леммы закончено.

Теорема 4. *Для каждого n существует такой бесконтекстный язык L_n, что* $\mathrm{sh}_s(L_n) = n$.

Доказательство. Пусть f — отображение регулярных выражений над Σ в $\mathscr{L}_{\Sigma,\delta}$, определенное так: $f(w) = w\,\delta w^R$ для каждого $w \in \Sigma$; $f(E \cup E') = f(E) \cup f(E')$; $f(E \cdot E') = f(E)[\delta \leftarrow f(E')]$; $f(E^*) = f(E)^{*\delta}$. Легко доказать, что $f(E)$ выражает $\{w\,\delta w^R \mid w$ из $|E|\}$ и $\mathrm{sh}_s(f(E)) = \mathrm{sh}(E)$. Таким же образом определим отображение g выражений из $\mathscr{L}_{\Sigma,\delta}$ подмножеств W в регулярные отображения: $g(w\,\delta w') = w$ для каждых w, $w' \in \Sigma^*$; $g((E \cup E')) = g(E) \cup g(E')$; $g(E[\delta \leftarrow E']) = g(E) \cdot g(E')$; $g(E^{*\delta}) = g(E)^*$. Тогда $g(E)$ выражает множество $\{w \mid w\,\delta w^R \in |E|\}$ и $\mathrm{sh}(g(E)) = \mathrm{sh}_s(E)$.

Теперь выберем регулярный язык P_n, такой, что $\mathrm{sh}(P_n) = n$. Такой язык должен быть, если Σ содержит не меньше двух элементов [4]. Положим $L_n = \{w\,\delta w^R \mid w \in P_n\}$, и пусть E из $\mathscr{L}_{\Sigma,\delta}$ — такое выражение для L_n, что $\mathrm{sh}_s(E) = \mathrm{sh}_s(L_n)$. Тогда $\mathrm{sh}_s(L_n) = \mathrm{sh}_s(E) = \mathrm{sh}(g(E)) \geqslant \mathrm{sh}(P_n) = n$. Таким же способом, найдя регулярное выражение E' для P_n, такое, что $\mathrm{sh}(E') = \mathrm{sh}(P_n)$, покажем, что $n = \mathrm{sh}(E') = \mathrm{sh}_s(f(E')) \geqslant \mathrm{sh}_s(L_n)$. Отсюда $\mathrm{sh}_s(L_n) = n$, и теорема доказана.

Интересно заметить, что семейство языков с подстановочной звездной высотой не больше единицы связано с семейством последовательных языков. Мы называем язык последовательным, если он определим системой контекстно-свободных уравнений вида $X_i = f_i(X_i, \ldots, X_n)$, $i = 1, \ldots, n$. Легко проверить, что такие уравнения можно решить, не вводя выражения с подстановочной звездной высотой больше единицы.

ДОБАВЛЕНИЕ РЕДАКТОРА

М. К. Интема предложила язык, родственный языку подстановочных выражений, также описывающий класс всех контекстно-

свободных языков (см., например, Jntema M. K., Cap Expressions for Context-Free Languages, *Information and Control*, **18** (1971), № 4, 311—318). В своих ранних работах (Jntema M. K., Inclusion Relations among Families of Context-Free Languages, Inf. Contr., **10** (1967), 572—597) Интема рассматривала расширения языка регулярных выражений с помощью matching choise — операции, описывающей подкласс контекстно-свободных языков. В первой из упомянутых статей Интема вводит операцию — аналог итеративной подстановки, которая вместе с объединением и конкатенацией позволяет получить любой контекстно-свободный язык над алфавитом Σ, исходя из «букв» Σ — однобуквенных слов в Σ^*.

Операции, предложенные в статьях Груски (см. этот сборник), Мак-Вертера и Интемы для порождения контекстно-свободных языков, несколько отличаются, хотя и тесно связаны между собой. Эта связь — фактически, сведение некоторых операций к другим — рассматривается в статье Груски. Поставленная в этой статье задача о минимальном числе вспомогательных символов не сводится к задаче о звездной глубине подстановочного выражения, решаемой в статье Мак-Вертера, хотя эти задачи и связаны одна с другой.

А. А. Мучник

СПИСОК ЛИТЕРАТУРЫ

1. Arden D. N., Delayed logic and finite state machines, Proc. Second Annual Symposium on Switching Circuit Theory and Logical Design, Detroit, Michigan (1961), pp. 131—151.
2. Bar-Hillel Y., Language and Information, Addision-Wesley, Reading, Mass., 1964.
3. Brzozowski J. A., Regular-like expressions for some irregular languages, Proceedings of the Ninth Annual Symposium on Switching and Automata Theory, General Electric Research and Development Center, Schenectady, N. Y., 1968.
4. Eggan L. C., Transition graphs and the star height of regular events, *Michigan Math. J.*, **10** (1963), 385—397.
5. Ginsburg S., Rice H. C., Two families of languages related to ALGOL, *J. Assoc. Comput. Mach.*, **9** (1962), 3.
6. Greibach S., A new normal form theorem for context-free phase structure grammars, *J. Assoc. Comput. Mach.*, **12** (1965), 42—52.
7. Greibach S., Full AFLs and nested iterated substitution, *Information and Control*, **16** (1970), 7—35.
8. Gruska J., A characterization of context-free languages, *J. Comput Systems Sci.*, **5** (1971), 353—364. (Русский перевод см. наст. сборник, стр. 114—126.)
9. Hopcroft J. E., Ullman J. D., Formal Languages and Their Relation to Automata, Addison-Wesley, Reading, Mass., 1969.
10. Kral J., A modification of a substitution theorem and some necessary and sufficient conditions for sets to be context-free, *Math. Systems Theory*, **4** (1970), 129—139.
11. McWhlrter I. P., Research Report CSRR-2016, University of Waterloo, Waterloo, Ontario. Canada, June 1970.

Об эффективности алгоритмов [1])

Давид Пейджер

Обычно уделяется мало внимания тому, чтобы дать формальное определение эффективности алгоритма, хотя в таких работах, как (1—3, 8, 9, 12), выявлено большое отличие между различными алгоритмами в тех случаях, когда для всех аргументов (или для всех достаточно больших аргументов) вычисления по некоторому алгоритму постоянно требуют большего времени или пространства, чем соответствующие вычисления по алгоритмам другого рода. В этой статье мы даем естественную формулировку понятия эффективности алгоритмов в терминах *пространственно-временной меры.* Исследуемые вопросы носят чисто теоретический характер, но наш метод оценивания алгоритмов с некоторыми изменениями может быть применен к реальным вычислительным программам. Это было сделано в Гавайском университете и будет служить предметом последующей статьи.

Мы пользуемся обозначениями Дэвиса (4) и его описанием машин Тьюринга, но в своих доказательствах свободно пользуемся тезисом Чёрча. Машины Тьюринга обозначаются прописными буквами, в то время как их Гёделевские номера изображаются соответствующими строчными буквами. Гёделевские номера машин Тьюринга, вычисляющих примитивно-рекурсивные функции, называются *примитивными индексами. S*-символ S_1 рассматривается в качестве символа единицы. Мы нумеруем квадраты ленты ... —2, —1, 0, +1, +2, ..., причем квадрат с 0 является начальным обозреваемым квадратом. *Актом* машины Тьюринга считается сдвиг влево или вправо или запись определенного символа в обозреваемом квадрате. Говорят, что две машины Тьюринга имеют *тождественное поведение*, если для любого аргумента они совершают одинаковую последовательность актов. Символ равенства (=) используется нами

[1]) Pager D., On the Efficiency of Algorithms, *Journal of the Association for Computing Machinery*, 17, № 4 (October 1970), 708—714.

в сильном смысле, а формула, в которой два выражения приравнены, имеет еще тот дополнительный смысл, что левая и правая ее части или обе определены, или обе не определены. Утверждая, что $f^{(n)}$ — частично рекурсивная функция на множестве аргументов S, мы имеем в виду, что существует машина Тьюринга Z, такая, что

$$f^{(n)}(x_1, \ldots, x_n) = U(\min_y T(z, x_1, \ldots, x_n, y))$$

для всех $(x_1, \ldots, x_n) \in S$,

и говорим в этом случае, что Z *вычисляет* f на множестве S [1]). Под равенством $f^{(n)} = g^{(n)}$ мы понимаем, что $f(x_1, \ldots, x_n) = g(x_1, \ldots, x_n)$ для всех $(x_1, \ldots, x_n) \in S$. Если $\zeta(x)$ есть некоторая функция, значениями которой являются машины Тьюринга, а P есть свойство (например, рекурсивность) функций с натуральными значениями, то говоря, что $\zeta(x)$ имеет свойство P, мы подразумеваем, что функция, равная Гёделевскому номеру $\zeta(x)$, имеет свойство P. Мы говорим, что $f(x, y)$ — *возрастающая функция от* x и y, если для любых x_0 и y_0 функции $f(x_0, y)$ и $f(x, y_0)$ обе являются строго возрастающими.

ОПРЕДЕЛЕНИЕ ПРОСТРАНСТВЕННО-ВРЕМЕННОЙ МЕРЫ

Время, затраченное при вычислении на машине Тьюринга Z на аргументах $(x_1, \ldots, x_n)$, отождествляется с числом имеющихся в этом вычислении мгновенных описаний и обозначается через $E(z, x_1, \ldots, x_n)$. *Пространство* $M(z, x_1, \ldots, x_n)$, использованное в таком вычислении, есть некоторая возрастающая функция от (a) *рабочего пространства*, в качестве которого берется (пользуясь понятием объема ленточного выражения, данным Рабином и Ваном (10)) мера максимального объема ленты при этом вычислении, и (b) от пространства, необходимого для хранения программы (*программного пространства*); мы отождествляем его с общим числом символов, использованных в таблице четверок, предполагая, что индексы S- и q-символов записаны. *Пространственно-временной мерой* $\mu(z, x_1, \ldots, x_n)$ *вычисления* машины Тьюринга Z на аргументах $(x_1, \ldots, x_n)$ мы считаем либо $E(z, x_1, \ldots, x_n)$ (в этом случае мы называем ее *чисто-временной мерой*), либо $M(z, x_1, \ldots, x_n)$, либо некоторую возрастающую рекурсивную функцию от $E(z, x_1, \ldots, x_n)$ и $M(z, x_1, \ldots, x_n)$ (в последних двух случаях

[1]) Нет необходимости требовать, чтобы S содержалось внутри области определения $f^{(n)}$. Для n-местной функции общепринятое понятие частичной рекурсивности (такое, как, например, у Дэвиса (4)) может быть теперь выражено как частичная рекурсивность на множестве всех n-ок.

мы говорим, что мера содержит *пространственную компоненту* [1]). Для любой функции $f(x_1, \ldots, x_n)$, которую мы хотим оценить, постулируем существование вероятностной функции $p(x_1, \ldots, x_n)$, так что $p(a_1, \ldots, a_n)$ представляет вероятность, которую мы бы назвали предварительной оценкой $f(a_1, \ldots, a_n)$. Предполагается, что $p(x_1, \ldots, x_n) > 0$ только для $(x_1, \ldots, x_n)$ из области определения функции f. Множество $\{(x_1, \ldots, x_n) \mid p(x_1 \ldots x_n) > 0\}$ называется множеством значимости (функций). Буква p во всех случаях обозначает n-местную вероятностную функцию вышеуказанного вида. Следующие определения сделаны с учетом такой вероятностной функции.

Пространственно-временная мера $\gamma_p(z)$ *машины Тьюринга* Z есть среднее значение пространственно-временной меры ее вычислений, т. е.

$$\sum_{x_1, \ldots, x_n=0}^{\infty} p(x_1, \ldots, x_n)\, \mu(z, x_1, \ldots, x_n).$$

Эффективность вычисления (*или машины Тьюринга*) есть величина, находящаяся в обратной зависимости от их пространственно-временной меры. *Пространственно-временная мера функции* f есть точная нижняя грань пространственно-временных мер машин Тьюринга Z, вычисляющих f на множестве значимости. *Оптимальная машина Тьюринга*, вычисляющая $f^{(n)}$ на множестве значимости, есть такая машина, которая имеет максимально возможную эффективность (т. е. имеет наименьшую возможную пространственную меру). Оптимальная машина Тьюринга, вычисляющая функцию $[z]_n$ на множестве значимости, обозначается через $\varphi_p(z)$ [2]).

Она называется *нетривиальной*, если она существует по крайней мере для некоторого z. В тех случаях, когда не может возникнуть недоразумений, $\varphi_p(z)$ и $\gamma_p(z)$ записываются просто как $\varphi(z)$ и $\gamma(z)$ соответственно.

Пример. Оценим $\gamma_p(z_0)$ для машины Тьюринга Z_0, вычисляющей $2x$, если $M(z, x)$ есть сумма программного и рабочего

---

[1]) Дополнительный интерес представляет тот случай, к которому также применимы результаты статьи, когда в качестве $\mu(z, x_1 \ldots x_n)$ берется $E(z, x_1, \ldots, x_n)$

$$\sum_{i=1} m(z, x_1, \ldots, x_n, i),$$

где $m(z, x_1, \ldots, x_n, i)$ есть некоторая возрастающая функция от программного пространства и размера i-го ленточного выражения (т. е. длины ленты на i-м шаге вычисления) в вычислении Z для аргумента $(x_1, \ldots, x_n)$.

[2]) Мы допускаем, что $\varphi_p(z)$ неоднозначна в том случае, когда существует более чем одна оптимальная машина Тьюринга.

пространств, $\mu(z)$ есть $M(z, x) + E(z, x)$, а $p(x) = 0$ при $x = 0$ и $p(x) = 90/\pi^4 x^4$ при $x > 0$ [1]).

Соответствующая машина Z_0 определяется четверками $q_1 1 B q_1$; $q_1 B R q_2$; $q_2 1 S_3 q_3$; $q_3 S_3 L q_3$; $q_3 1 L q_3$; $q_3 B 1 q_4$; $q_4 1 R q_4$; $q_4 S_3 R q_4$; $q_4 1 1 q_2$; $q_4 B L q_5$; $q_5 S_3 1 q_6$; $q_6 1 L q_5$. Здесь программное пространство равно 108, рабочее пространство получаем из $2x + 1$ и $E(z, x) = 2x^2 + 7x + 3$. Следовательно, $\gamma(z_0) = 127,\ 03\ldots$.

Применяя эти понятия к программам вычислительных машин, часто бывает легче определить их эффективность статистическими методами с помощью выборочных измерений использованного пространства и времени. Определение пространственно-временной меры вычисления как некоторой возрастающей функции от затраченного времени и пространства уместно, в частности, для систем с разделением времени, где обе эти величины являются ценными.

Теорема 1. *Пусть $\varphi(z)$ не тривиальна. Тогда $\varphi(z)$ является частично рекурсивной в том и только том случае, когда множество значимости конечно. Кроме того, если множество значимости бесконечно, то $\varphi(z)$ не частично рекурсивна на примитивных индексах.*

Доказательство. *Пусть множество значимости бесконечно.* Покажем, прежде всего, что существует такая примитивно рекурсивная функция $G(z, r)$, что $[G(z, r)]_n$ примитивно рекурсивна относительно $[z]_n$ и что если S есть произвольное бесконечное множество n-ок, то

$$[G(z, r)]_n \underset{S}{=} [z]_n \leftrightarrow\sim \bigvee_y T(r, r, y).$$

Пусть $\sigma(x_1, \ldots, x_n)$ — некоторая рекурсивная функция, упорядочивающая n-ки, и пусть $f(z, r, x_1 \ldots x_n)$ определяется следующим образом:

$$f(z, r, x_1, \ldots, x_n) = \begin{cases} [z]_n(x_1, \ldots, x_n) + 1, \text{ если } \bigvee_{y=0}^{\sigma(x_1, \ldots, x_n)} T(r, r, y); \\ [z]_n(x_1, \ldots, x_n) \text{ в противном случае.} \end{cases}$$

Применяя теорему об итерации, получаем

$$f(z, r, x_1, \ldots, x_n) = [G(z, r)]_n(x_1, \ldots, x_n),$$

где $G(z, r)$ есть функция указанного выше типа.

Если $\varphi(z)$ существует для некоторого z, то существует и $\varphi(c)$, где C есть машина Тьюринга $\{q_1 B B q_1\}$, вычисляющая при-

[1]) Таким образом, $\sum_{x=0}^{\infty} p(x) = 1$.

митивно рекурсивную функцию $(x_1 + 1,\ x_2 + 1,\ \ldots,\ x_n + 1)$ [1]). Это имеет место потому, что пространство и время, необходимые для вычислений на C, не превосходят пространства и времени, необходимых для вычислений на любой другой машине Тьюринга. Пусть H есть множество оптимальных машин Тьюринга Z, таких, что $[z]_n \overline{\overline{S}} [c]_n$. В случае чисто-временной меры H есть множество тех машин Тьюринга, которые не содержат четверок, начинающихся с $q_1 1$, и поэтому является рекурсивным множеством. С другой стороны, если рассматриваемая мера имеет пространственную компоненту, то существует верхняя граница для наибольшего индекса у S- или q-символа, который может появиться в произвольном элементе H. Таким образом, H — множество конечное, а значит и рекурсивное. Следовательно, если бы $\varphi(z)$ была частично рекурсивной на примитивных индексах, то множество $\{(c, r) \mid \varphi(G(c, r)) \in H\}$ было бы полувычислимым. Но это невозможно, так как $\varphi(G(c, r)) \in H$ имеет место тогда и только тогда, когда $\sim \bigvee\limits_y T(r, r, y)$.

Пусть множество значимости конечно. Чтобы найти $\varphi(z_0)$, начнем с вычисления значения $\gamma(z_0)$. Заметим, что оно определено здесь тогда и только тогда, когда определено $\varphi(z_0)$. Если мера имеет пространственную компоненту, то $\gamma(z_0)$ задает верхнюю границу обеим ее составляющим: пространству, занятому для хранения программы, и рабочему пространству. А это ограничивает число машин Тьюринга и число мгновенных описаний, которые нужно просмотреть, чтобы подобрать $\varphi(z_0)$. Если мера чисто-временная, то мера времени оптимальной машины Тьюринга Z_0^* должна быть ограничена следующим образом:

$$E(z_0^*,\ x_1,\ \ldots,\ x_n) \leqslant \frac{\gamma(z_0)}{\min\limits_{(x_1,\ \ldots,\ x_n) \in S} p(x,\ \ldots,\ x_n)}; \qquad \text{(A)}$$

однако существует бесконечное множество машин Тьюринга, удовлетворяющих этому неравенству. Отыскивая оптимальную машину Тьюринга, мы можем ограничиться теми машинами Z, которые имеют следующее свойство: если q^i или S_i принадлежит алфавиту Z, то и q_j или S_j для всех $1 \leqslant j \leqslant i$ также принадлежит ему. На каждом шаге вычисления для любого элемента из конечного множества аргументов в мгновенное описание может быть добавлено самое большее одно новое состояние и один новый символ. Следовательно, с помощью (A) мы проверяем некоторое конечное число машин Тьюринга, и, значит, используя (A), мы сможем эффективно подобрать $\varphi(z_0)$.

[1]) Вспомним, что $\varphi(z)$ есть сокращение для $\varphi_p(z)$, где p есть n-местная функция.

Замечание. В противоположность последней части предыдущей теоремы, если воспользоваться одним из следующих критериев для образования понятия «оптимального алгоритма» $\varphi_p(z)$, то даже для конечного множества значимости $\varphi_p(z)$ не будет частично рекурсивной:

(a) $\varphi_p(z)$ есть наиболее эффективная машина Тьюринга, вычисляющая $[z]_n$;

(b) $\varphi_p(z)$ есть машина Тьюринга с кратчайшей программой, вычисляющая $[z]_n$ на множестве значимости [1]).

Обобщение теоремы 1. *Пусть $\varphi(z)$ есть некоторая нетривиальная оптимизирующая функция, ассоциированная с бесконечным множеством значимости, мера вычисления относительно которого есть многочлен от рабочего пространства, программного пространства и времени. Тогда существует такое рекурсивное множество примитивных индексов, что $\varphi(z)$ на нем определено, но не рекурсивно.*

Доказательство. Функцию $G(c, r)$ можно выбрать так, чтобы она была возрастающей относительно r; в этом случае множество примитивных индексов $\{G(c, r)$ по всем $r\}$ — рекурсивное. Из доказательства теоремы 1 следует, что $\varphi(z)$ не является частично рекурсивной на этом множестве. Нам осталось только доказать, что $\varphi(G(c, r))$ действительно определено для каждого r. Это очевидно, если $\sim \bigvee_y T(r, r, y)$. Предположим $T(r, r, y_0)$ для некоторого y_0, и положим

$$h = \max_{\sigma(x_1, \ldots, x_n) \leqslant y_0} \sum_{i=1}^{n} x_1 + 2n - 1.$$

Тогда можно непосредственно построить машину Тьюринга, которая вычисляет $[G(c, r)]_n$ для любого аргумента не более чем за $h + 3$ шага, не заходя правее квадрата h, и которая использует в качестве S-символов только пустой символ и символ единицы. Возможно лишь конечное множество ζ таких машин Тьюринга с различным поведением, и по предположению, сделанному относительно меры вычисления, так как $\gamma(c)$ определено, то и $\gamma(z)$ должно быть определено для любого $z \in \zeta$ [2]). Легко видеть, что если мера чисто-временная, то некоторый элемент множества будет оптимальной машиной для $[G(c, r)]_n$,

[1]) Этот результат доказан Пейджером в работе (7), где в качестве объема программы Z берется любая функция $L(z)$ (не обязательно частично рекурсивная), такая, что выполняется неравенство $L(u) < k$ для произвольного k, которое истинно только для конечного числа программ U.

[2]) Конечно, это справедливо для более широкого класса мер вычисления, чем рассмотренный здесь класс мер.

если же мера имеет пространственную компоненту и z_1 есть произвольный элемент из ζ, то оптимальная машина Тьюринга должна находиться среди конечного множества тех машин, пространственно-временная мера которых не превосходит $\gamma(z_1)$.

Как мы увидим из теорем 2 и 3, условия существования $\varphi(z)$ зависят от того, имеет или не имеет выбранная пространственно-временная мера пространственную компоненту.

Теорема 2. *Для бесконечных множеств значимости и чисто-временных мер справедливо следующее утверждение: не существует оптимальной машины Тьюринга для такой функции* $f(x)$, *что* $f(x) - x$ *монотонно возрастает* [1]).

Доказательство. Расположим аргументы из множества значимости S в возрастающую последовательность $x_0, x_1, x_2, \ldots$. Предположим, что оптимальная машина Тьюринга Z^*, вычисляющая $f(x)$ на S, существует, и обозначим через m число содержащихся в ней q-символов. Так как S бесконечно, то для каждого аргумента x_i машина Z^* должна обязательно зайти в первый пустой квадрат справа (квадрат $x_i + 1$), ибо в противном случае $[z^*]_t(x_t) - (x_t)$ было бы постоянным для всех $t \geqslant i$. В любом случае наиболее коротким вычислением является такое вычисление, при котором Z^* движется к $x_i + 1$, а затем, не изменяя направления, добавляет $f(x_i) - x_i - 1$ единиц справа. Но Z^* не может так поступать более чем на m аргументах, так как после посещения (x_i+1)-го квадрата она должна добавлять различное число единиц, не изменяя направления движения, для чего потребуется новое внутреннее состояние для каждого i.

Для сравнения рассмотрим машину Тьюринга Z_h, $h > m$, которая работает оптимальным образом на аргументах $x_0, x_1, \ldots, x_h$, тогда как на x_i, $i > h$, она после посещения $(x_h + 1)$-го квадрата меняет направление и преобразует ленту к такому виду, в какой привела бы ее машина Z^* за время работы до первого посещения ею квадрата $x_h + 1$. По меньшей мере для одного из элементов (назовем его x_t) множества $\{x_0, \ldots, x_m\}$ вычисления на Z_h должны содержать меньшее число шагов, чем на Z^*; с другой стороны, они не превосходят вычислений на Z^* для всех аргументов из $\{x_0, \ldots, x_m, \ldots, x_h\}$. Следовательно, имеет место следующее неравенство:

$$\gamma(z^*) - \gamma(z_h) \geqslant p(x_t)(E(z^*, x_t) - E(z_h, x_t)) - 3\left(\sum_{i=h+1}^{\infty} p(x_i)\right)(x_h + 1).$$

[1]) Для машин, использующих кодирование чисел с основанием системы счисления, большим 1, вместо унарного кодирования Дэвиса (4), соответствующий результат имеет место для тех функций, для которых $f(x)/x$ монотонно возрастает.

Так как каждое вычисление требует посещения квадрата $x_i + 1$, то $x_i \leqslant E(z^*, x_i)$ для каждого i, и поэтому

$$\sum_{i=0}^{\infty} p(x_i)\, x_i \leqslant \sum_{i=0}^{\infty} p(x_i)\, E(z^*, x_i) = \gamma(z^*),$$

которое, по предположению для Z^*, определено. Это влечет за собой

$$\lim_{h\to\infty} \sum_{i=h+1}^{\infty} p(x_i)\, x_i = 0,$$

что в свою очередь приводит к тому, что

$$\lim_{h\to\infty} \left(\sum_{i=h+1}^{\infty} p(x_i) \right) x_h = 0.$$

Отсюда следует, что для достаточно больших h имеет место $\gamma(z^*) > \gamma(z_h)$, а это противоречит сделанному предположению.

Теорема 3. *Пусть рассматриваемое множество значимости бесконечно. Тогда в том и только том случае, когда употребляемая мера содержит пространственную компоненту, имеет место следующая импликация: всякий раз, когда* $\gamma(z)$ *определено, существует и* $\varphi(z)$ [1]).

Доказательство. Если мера имеет пространственную компоненту, то ясно, что это условие является достаточным. Это же условие является и необходимым, так как, если пространственно-временная мера есть чисто-временная, то можно привести пример, противоречащий рассматриваемой импликации. Если Z_0 вычисляет x^2, то, согласно теореме 2, $\varphi_p(z_0)$ не определена при любом выборе p. Но $\gamma_p(z_0)$ определена, например, для функции $p(x)$, равной

$$\frac{1}{E(z_0, x)(x+1)^2 \sum_{x=0}^{\infty} \frac{1}{E(z_0, x)(x+1)^2}}.$$

Теорема 4. (Теорема оптимизации.) *Если используемая мера имеет пространственную компоненту, то существует такая примитивно-рекурсивная функция* χ_p, *что если* $\varphi_p(z)$ *существует, то* $\chi_p(z, t) = \varphi_p(z)$ *для почти всех* t [2]).

Доказательство. Пусть Z_0 есть некоторая машина Тьюринга, для которой $\varphi_p(z) = Z_0^*$ (предположим) существует.

[1]) Хотя в этом случае в соответствии с теоремой 1 функция $\varphi(z)$ не эффективно вычислима.

[2]) Согласно работе Голда (5), $\varphi(z)$ тогда «ограниченно-рекурсивная» и, следовательно, 2-рекурсивная.

Занумеруем n-ки аргументов из множества значимости S эффективным образом в последовательность $a_1, a_2, \ldots$. Пусть $\gamma_t(z)$ обозначает соответствующую величину $\sum_{i=1}^{t} p(a_i)\,\mu(z, a_i)$, и Z_t^* есть такая машина Тьюринга, вычисляющая $[z_0]_n$ на $\{a_1, \ldots, a_t\}$, для которой $\gamma_t(z)$ имеет наименьшее значение. Машина Z_0^* также вычисляет $[z_0]_n$ на $\{a_1, \ldots, a_t\}$ и, следовательно, ее пространственно-временная мера дает верхнюю границу для программного пространства машины Z_t^*. Пусть P_1 есть конечное множество машин Тьюринга, программное пространство которых не превосходит этой границы. Для любой машины Тьюринга Z из P_1, вычисляющей на S функцию, отличную от той, которую вычисляет Z_0, должен существовать некоторый аргумент $a_u \in S$, такой, что $[z]_n(a_u) \neq [z_0]_n(a_u)$. Пусть t_z есть наименьший индекс у тех a, для которых это имеет место, P_2 — подмножество множества P_1 таких машин Тьюринга, а $t_{\max}$ есть максимальное значение t_z для $Z \in P_2$. Ясно, что для $t > t_{\max}$ машины Z_t^* и Z_0^* вычисляют одну и ту же функцию на S, и, следовательно, обе являются элементами $P_1 - P_2$.

Можно подумать, однако, что в некоторых случаях Z_t^* с возрастанием t может неограниченно «колебаться» среди элементов множества $P_1 - P_2$, отличных от Z_0^*. Покажем, что такой случай не имеет места. Если $[z]_n = [z_0]_n$, то, поскольку $\gamma_t(z)$ монотонно возрастает относительно t, либо $\lim_{t\to\infty} \gamma_t(z) = \infty$, либо $\lim_{t\to\infty} \gamma_t(z)$ существует и величина его не меньше, чем $\gamma(z_0)$. Так как $\gamma_t(z_t^*) \leqslant \gamma_t(z_0^*) \leqslant \gamma(z_0)$ для всех t, то отсюда следует, что почти для всех t выполняется неравенство $\gamma_t(z_t^*) < \gamma_t(z)$ для каждой такой машины Тьюринга из конечного множества $P_1 - P_2$, для которой $\lim_{t\to\infty} \gamma_t(z) \neq \gamma(z_0)$.

Согласно теореме 1, Z_t^* может быть выражена как $\psi(z_0, t)$, где ψ — частично рекурсивная функция. Пусть ζ_z есть некоторая процедура, последовательно вычисляющая $\psi(z, 0)$, $\psi(z, 1)$, $\ldots$. Если ζ_z делает более чем k шагов при вычислении $\psi(z, 0)$, то $\chi(z, k)$ может быть выбрана произвольным образом (например, $\{q_1 B B q_1\}$), в противном случае положим $\chi(z, k)$ равной последнему значению ψ, полученному процедурой ζ_z на ее первых k шагах.

Рабин (9) и Блюм (2) доказали существование «произвольно сложных» функций. Их понятие сложности совершенно отлично от нашей концепции пространственно-временной меры, но все же может возникнуть вопрос, имеют ли место сходные результаты. Вопрос такого рода: «Существуют ли функции произвольно большой пространственно-временной меры?» — здесь тривиален,

так как мы можем специально определить функции со столь большими значениями $f(x)-x$, что их пространственно-временная мера неизбежно превзойдет любую наперед заданную границу. Вопрос приобретает смысл, если мы исключим функции такого типа и рассмотрим вместо них те, для которых функция $f(x)-x$ ограничена. Тогда оказывается, как показывает следующая теорема, что ответ зависит от того, конечно или нет рассматриваемое множество значимости.

Теорема 5. *Если используемое множество значимости конечно и h — произвольное фиксированное число, то существует верхняя грань пространственно-временной меры тех функций, для которых $f(x)-x<h$ при всех $x\in S$. С другой стороны, если множество значимости S бесконечно, то существует рекурсивная функция $f(x)$, такая, что $f(x)=x$ или $x+1$ для всех $x\in S$, пространственно-временная мера которой бесконечна.*

Доказательство. Теорема очевидна, если множество значимости конечное. Рассмотрим тот случай, когда оно бесконечно. Определим рекурсивно $(f(x), R_x)$, где R_x есть некоторое множество номеров машин Тьюринга, следующим образом: $f(0)=0$, $R_0=\Lambda$. Если $p(y+1)\neq 0$, то для определения $f(y+1)$ рассмотрим наименьшее из чисел (скажем, z_0) множества $\{1, 2, \ldots, y+1\}-R_y$, которое является Гёделевским номером некоторой машины Тьюринга и такое, что

$$\mu(z_0, y+1)\leqslant \frac{1}{p(y+1)}, \tag{A}$$

и положим

$$f(y+1)=\begin{cases} y+2, \text{ если } [z_0]_1(y+1)=y+1, \\ y+1 \text{ в противном случае} \end{cases} \tag{B}$$

и $R_{y+1}=R_y\cup\{z_0\}$; если не существует чисел, удовлетворяющих (A), или если $p(y+1)=0$, то пусть $f(y+1)=y+1$ и $R_{y+1}=R_y$. Рассмотрим теперь неравенство $\mu(z,x)\leqslant 1/p(x)$ для любой машины Тьюринга Z, вычисляющей $f(x)$. Оно может иметь место самое большее для $2z$ значений x, таких, что $p(x)\neq 0$, так как в противном случае появляется противоречие, заключающееся в том, что z встречается в качестве z_0 в (B) при вычислении некоторого значения функции $f(x)$. Поэтому почти для всех y имеет место $p(y)\mu(z,y)>1$ и, следовательно, $\gamma(z)=\sum_{x\in S} p(x)\mu(z,x)=\infty$. Так как это справедливо для любой машины Тьюринга, вычисляющей $f(x)$, то ее пространственно-временная мера бесконечна.

СПИСОК ЛИТЕРАТУРЫ

1. Arbib M. A., Blum M., Machine dependence of degrees of difficulty, *Proc. Amer. Math. Soc.*, **16** (1965), 442—447.
2. Blum M., A machine independent theory of the complexity of recursive functions, *J. ACM*, **14**, 2 (April 1967), 322—336. (Русский перевод: Блюм М., Машинно-независимая теория сложности рекурсивных функций, Сб. Проблемы математической логики, «Мир», 1970, 401—422.)
3. Cleave J. P., A hierarchy of primitive recursive functions, *Z. Math. Logik Grundlagen Math.*, **9** (1963), 331—346. (Русский перевод: Клив Дж. П., Иерархия примитивно рекурсивных функций, Сб. Проблемы математической логики, «Мир», М., 1970, 94—113.)
4. Davis M., Computability and Unsolvability, McGraw-Hill, New York, 1958.
5. Gold E. M., Limiting recursion, *J. Symbolic Logic*, **30** (1965), 28—57
6. Kleene S. C., Introduction to Metamathematics Van Nostrand, Princeton, N. Y., 1952. (Русский перевод: Клини С. К., Введение в метаматематику, ИЛ, М., 1957.)
7. Pager D., On the problem of finding minimal programs for tables, *Inform Contr.*, **14** (1969), 550—554. (Русский перевод: Пэйджер Д., О проблеме нахождения по таблицам минимальных программ.)
8. Ritchie R. W., Classes of predictably computable functions, *Trans. Amer Math. Soc.*, **106** (1963), 139—173. (Русский перевод: Риги Р. В., Классы предсказуемо вычислимых функций, Сб. Проблемы математической логики, «Мир», М., 1970, 50—93.)
9. Rabin M. O., Degree of difficulty of computing a function and a partial ordering of recursive sets, Tech. Rep. 2, Hebrew U., Jerusalem, 1960.
10. Rabin M. O., Wang H., Words in the history of a Turing machine with a fixed input, *J ACM*, **10**, 4 (Oct. 1963), 526—527.
11. Shepherdson J. S., Sturgis H. E., Computability of recursive functions, *J. ACM*, **10**, 2 (April 1963), 217—255.
12. Stearns R. E., Hartmanis J., Lewis P. M., Hierarchies of memory limited computations, Proc. Sixth Annual Symposium on Switching Circuit Theory and Logical Design, IEEE, New York, 1965, p. 179—190. (Русский перевод: Стириз Р. Е., Хартманис Дж., Льюис П. М., Иерархии вычислений с ограниченной памятью, Сб. Проблемы математической логики, «Мир», М., 1970, 301—319.)

Запись содержания текстов, не зависящая от их морфологических и синтаксических особенностей[1])

Б. Вокуа, Ж. Вейон, Н. Недобежкин, С. Бургиньон

1. ЦЕЛЬ И ГРАНИЦЫ ИСПОЛЬЗОВАНИЯ СЕМАНТИЧЕСКОЙ ЗАПИСИ

Назовем «выражением» письменный (или устный) текст на естественном языке. Как известно, тексты состоят из фраз, фразы состоят из простых предложений и т. д. вплоть до слов, которые состоят из морфем. На всех этих уровнях тексты подчиняются правилам морфологии и синтаксиса соответствующего языка. Вообще говоря, при построении текста морфологические и синтаксические конструкции, с одной стороны, и лексику — с другой, можно выбирать по-разному. Такое разнообразие выбора позволяет получать для одного и того же содержания много различных выражений. Поэтому при разработке моделей типа «Текст $\Rightarrow$ Смысл» и «Смысл $\Rightarrow$ Текст» необходимо определить такую запись, посредством которой содержание текстов можно было бы представлять независимо от грамматических и лексических свойств языка этих текстов. Такая запись и была бы идеальной семантической записью. Можно предположить, что семантическая запись тем эффективнее и тем ближе к идеальной, чем больше выражений, имеющих эквивалентное содержание, она позволяет записать в одинаковой форме. Другими словами, достаточно эффективная семантическая запись обеспечивает как распознавание равнозначности большого числа выражений эквивалентного содержания, так и порождение большого числа выражений одного и того же содержания.

Проблема построения семантической записи рассматривалась А. К. Жолковским и И. А. Мельчуком [1], [2] в рамках модели «Смысл $\Leftrightarrow$ Текст», а также С. Лэмбом [3] в его известной стратификационной модели. Она была одной из основных задач исследований С. Е. Т. А. по автоматическому переводу и остается в центре внимания в связи с другими приложениями.

Семантическая запись подчинена некоторым ограничениям. Эти ограничения естественным образом определяют границы применимости записи.

---

[1]) Vauquois B., Veillon G., Nedobejkine N., Bourguignon C., Une notation des textes hors des contraintes morphologiques et syntaxiques de l'expression, Stockholm, 1969 (COLING, 1969, Preprint № 17).

Ограничения первого типа связаны с тем, что модель «Смысл ⇒ Текст» является «порождающей», следовательно запись должна гарантировать возможность выводить из нее (т. е. строить по ней) синтаксические и лексические структуры данного языка.

Ограничения второго типа более жесткие, чем предыдущие, связаны с необходимостью уметь автоматически распознавать семантическую эквивалентность двух выражений. В самом деле, неоднозначность содержания двух выражений разрешать труднее, чем строить множество эквивалентных выражений.

Кроме того, ограничения на семантическую запись сами зависят от конкретной задачи. Действительно очень различны, по-видимому, условия использования этой записи в АП (автоматическом переводе) для внутриязыкового перефразирования, в системе «человек — машина» и т. д. Так, при АП разрешение неоднозначностей иногда оказывается излишним, а именно в тех случаях, когда в выходном языке имеется точно такая же неоднозначность, что и во входном. Напротив, система «человек — машина» предполагает, что машина может задавать человеку вопросы, позволяющие ей разрешить эту неоднозначность. Подобная обратная связь не может быть использована в АП.

Запись, о которой здесь идет речь («ЯП1» — «язык-посредник 1»), разработана специально для АП. Это означает, в частности, что для входного текста нужно уметь построить его семантическую запись. Соответствующий метод подробно изложен в разд. 3 данной статьи.

На первых порах были приняты следующие дополнительные ограничения:

а) в качестве единицы выражения (единицы перевода) рассматривается «синтаксическая фраза»;

б) в качестве семантически эквивалентных выражений в данной работе рассматриваются выражения, различающиеся синтаксическими конструкциями, тогда как различие в лексике допускается лишь в немногих случаях.

2. СТРУКТУРА ЯП

Элементы языка

Семантическая запись строится из элементов трех типов: лексические единицы, переменные, отношения.

Лексические единицы. Запись, не зависящая от форм языка, должна иметь свой собственный набор лексических единиц. Однако на первом этапе, который рассматривается в настоящей работе, ЯП не имеет собственного словаря. Поэтому в применении к АП переход от слова входного языка к слову

выходного языка осуществляется как простая операция "перехода". Причем эта операция является операцией, которая не относится ни к анализу, ни к синтезу.

Лексическая единица ЯП — это пара «русская лексическая единица с некоторым значением 's' — множество французских лексических единиц, имеющих то же значение 's'» (см. примеры в разд. 3).

Какова бы ни была лексика ЯП (собственная или, так сказать, заимствованная), его лексические единицы разбиваются на два класса: предикатные единицы (глаголы, отглагольные существительные, прилагательные, предлоги, союзы и т. д.) и непредикатные единицы (в основном слова-дескрипторы).

Переменные. К переменным относятся элементы, информация которых служит для построения правильного текста (с их помощью осуществляется переход от лексии к элементарному высказыванию, переход от соединения лексий к актуализации этого соединения и т. д.).

В применении к АП эти элементы называются «устойчивыми переменными», так как они извлекаются из текста на входном языке и для сохранения смысла должны быть выражены в тексте на выходном языке.

В качестве примера переменной можно привести переменную «тип высказывания»; ее значения — «утвердительность» и «отрицательность». Другой пример — время глагола (реальное «независимое» время, а не синтаксическое, т. е. обусловленное согласованием времен); вид и т. д.

Отношения. Отношения представляют собой метапредикаты ЯП. Одни из этих метапредикатов определяют место аргументов предикатов, другие указывают отношения между лексиями или их аргументами.

Все используемые отношения являются двуместными метапредикатами.

Лексии и элементарные высказывания

Принимаются следующие допущения: самая простая конструкция в речи — это элементарное высказывание, представленное предикатом с его аргументами. Для предиката и аргументов указаны соответствующие переменные.

В действительности прежде всего представляет интерес более абстрактная конструкция — «лексия». Определение лексии было дано в [6], [7]. Напомним, однако, что предикатная лексическая единица представляет некоторое «понятие». Например, единица *LIRE* 'ЧИТАТЬ' представляет понятие *lire,* которое впоследствии может быть реализовано предикатом *lire* или *ne pas lire* 'не читать' в разных временах, наклонениях и т. д.

LIRE имеет два аргумента, так что мы будем писать *LIRE* (x, y), где x и y суть формальные переменные, которые пробегают множество лексических единиц. Если принять за x лексическую единицу *secrétaire* 'секретарь/секретарша', а за y — лексическую единицу *journal* 'газета', то получим лексию:

LIRE (secrétaire, journal)

'ЧИТАТЬ (секретарь/секретарша, газета)',

которую можно реализовать по-французски следующими элементарными высказываниями в соответствии со значениями переменных:

Le secrétaire n'a pas lu les journaux

'Секретарь не прочел газет'

или

La secrétaire est en train de lire ce journal

'Секретарша занята чтением этой газеты

или

La lecture des journaux par les secrétaires

'Чтение газет секретарями'

и т. п.

Для построения лексии в ЯП имеются отношения, которые позволяют связать предикат с его аргументами.

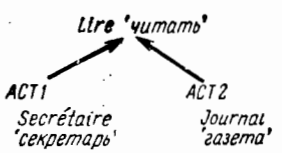


Рис. 1.

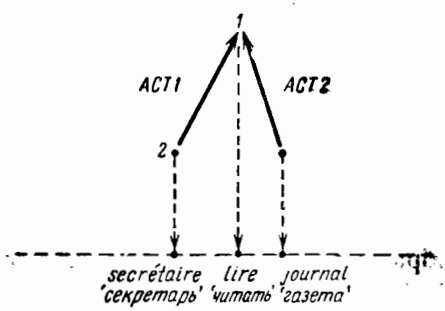


Рис. 2.

Пусть имеются отношения ACT n(a, P), где n = 1, 2 или 3, a — лексическая единица, P — лексическая единица с предикатным значением.

И пусть

$$\text{ACT1}(a, P(x, y)) = P(a, y)$$

или

$$\text{ACT3}(c, P(x, y, z)) = P(x, y, c).$$

Лексию *LIRE* (*secrétaire, journal*) мы представим с помощью двух отношений:

ACT1 (*secrétaire, Lire* (x, y))

и

ACT2 (*journal, Lire* (x, y)).

Эти отношения удобно изображать в виде графа, где каждая дуга (стрелка) соответствует отношению; причем стрелка направлена от первого члена отношения к его второму члену.

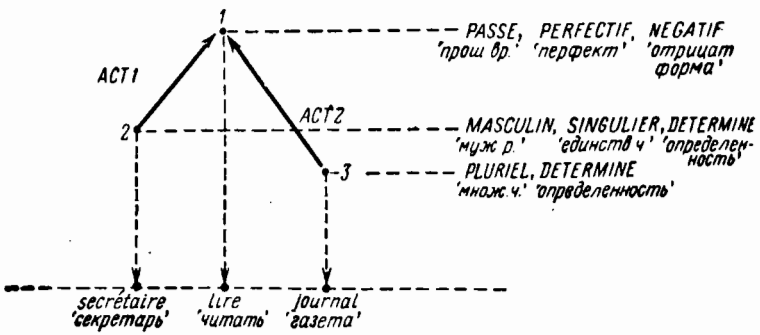


Рис. 3.

Итак, мы имеем граф, изображенный на рис. 1.

Для того чтобы показать, с одной стороны, граф отношений некоторого текста, а с другой — его лексику, определим отображение Δ вершин графа в лексику. Тогда мы получим следующее представление лексии (см. рис. 2).

Элементарное высказывание

Le secrétaire n'a pas lu les journaux

'Секретарь не прочел газет'

записывается посредством прибавления к предыдущему графу отображения Γ вершин графа в множество последовательностей переменных (см. рис. 3).

Композиция лексий и элемеитарных высказываний

В одной и той же фразе возможно наличие нескольких элементарных высказываний, поэтому необходимо уметь записывать композиции элементарных выражений. Это обеспечивается отношениями ЯП. Можно поставить вопрос о минимальном числе таких отношений; в настоящее время набор отношений ЯП явно не является минимальным (тем самым имеет место некоторая избыточность).

Для упрощения изложения допустим, что существует только одно такое отношение «EPITHETE» (эпитет). Это двуместный метапредикат

EPITHETE (x, y).

Рассмотрим фразу: *Le petit garçon porte un livre.* 'Маленький мальчик несет книгу'. Эта фраза содержит два элементарных высказывания:

E_1: *Le garçon porte un livre*
'Маленький мальчик несет книгу',
полученное из лексии: *Porter* (*garçon, livre*) 'Носить (мальчик, книга)'

E_2: *Le garçon est petit*
'Мальчик маленький',
полученное из лексии: *Petit* (*garçon*) 'Маленький (мальчик)'.

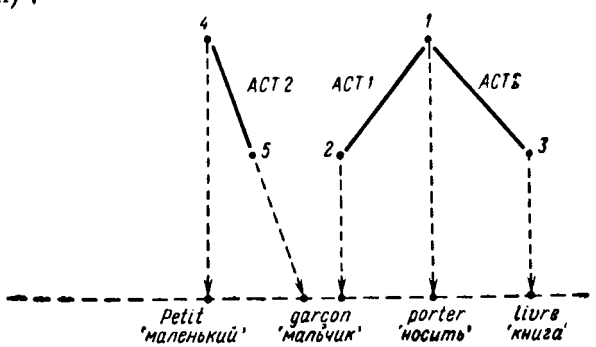


Рис. 4.

Указанные лексии представляются графами, изображенными на рис. 4. В данной фразе имеется отношение между точками 4 и 2 рассматриваемых графов:

EPITHETE [*Petit* [ACT1 (*garçon, porter* (x, *livre*))],
ACT1 (*garçon, porter* (x, *livre*))].

EPITHETE ['Маленький' [ACT1 ('мальчик, носить (x, книга'))],
ACT1 ('мальчик, носить (x, книга'))].

Это приводит к записи в лексиях, изображенной на рис. 5.

Рассмотрим теперь структуру лексий на более сложном примере:

On voit la maison sur le toit de laquelle les oiseaux font leur nid

'Виден дом, на крыше которого птицы вьют свои гнезда'

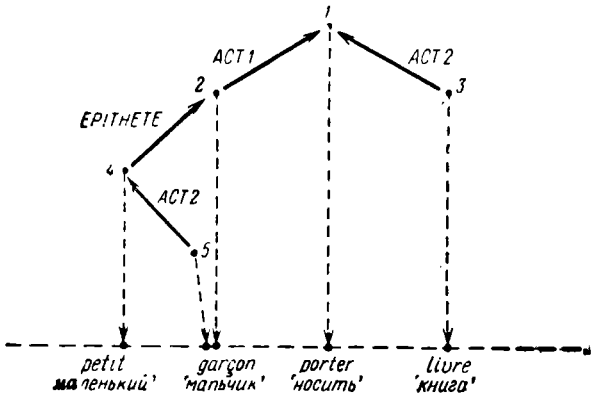


Рис. 5.

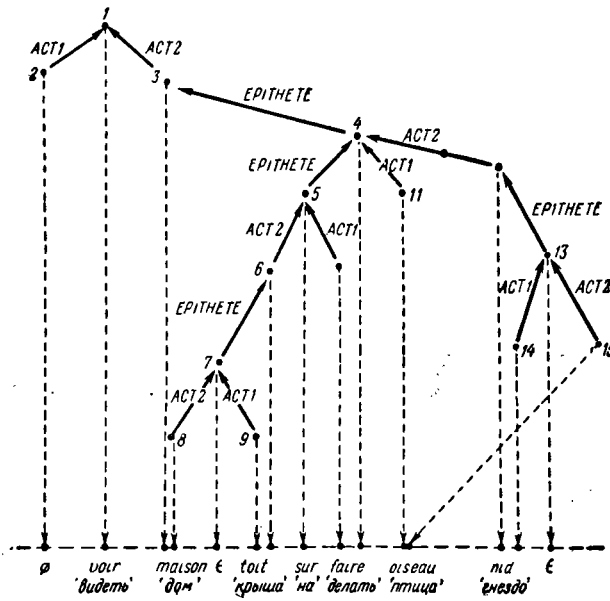


Рис. 6.

В этой фразе имеется пять элементарных высказываний, полученных из следующих лексий:

VOIR (∅, maison)	'видеть (∅ дом)'
FAIRE (oiseau, nid)	'делать (птица, гнездо)'
SUR (faire, toit)	'на (делать, крыша)'
∈ [1]) *(toit, maison)*	'∈ (крыша, дом)'
∈ *(nid, oiseau)*	'∈ (гнездо, птица)

[1]) ∈ — знак принадлежности. — *Прим. перев.*

Всему сложному высказыванию, т. е. фразе, соответствует граф отношений, изображенный на рис. 6.

3. АВТОМАТИЧЕСКИЙ ПЕРЕХОД К ЗАПИСИ НА ЯП

Исходная синтаксическая структура

После применения морфологической модели М1 и синтаксической модели М2 получается исходная синтаксическая структура. Эта структура представляется в виде графа зависимостей, где каждой вершине приписываются: номер словоформы (порядковый номер ее вхождения в текст), элементарная синтагма — терминальная категория и словообразовательный номер, неэлементарная синтагма — значения грамматических переменных и нетерминальная категория и, наконец, адрес в словаре (т. е. номер соответствующей лексической единицы). Кроме того, стрелка графа, связывающая вершину с ее управляющим, снабжена номером примененного синтаксического правила (см. стр. 167).

Интерпретация синтаксической структуры

Будучи чисто формальным объектом, синтаксическая структура чаще всего допускает несколько возможных логико-семантических интерпретаций для синтаксических отношений между вершинами ее графа. Так, отношению V201 (см. пример на стр. 157) могут отвечать различные смысловые отношения предиката к аргументу в зависимости от значений переменных предиката, а также от «семантического» класса, к которому он принадлежит. Если предикат — это переходный глагол в несовершенном виде, в пассиве или в возвратной форме, то вершина, зависящая от него по отношению V201, будет его вторым аргументом; в противном случае, она чаще всего будет его первым аргументом.

С другой стороны, одной фразе может быть в действительности сопоставлено несколько различных синтаксических структур, причем часто оказывается невозможным выбрать лучшую (=наиболее правильную) из них. Так обстоит дело, в частности, в случаях нахождения управляющих для предложных, союзных или наречных дополнений и обстоятельств. Для подобных вершин на уровне синтаксического анализа трудно, а иногда и невозможно ввести правила, позволяющие находить истинное управляющее. Поэтому мы выбрали следующее решение: все вершины указанного типа всегда подчиняются самому «высокому» управляющему в структуре (см. пример на стр. 167).

В силу сказанного перед ЯП ставятся следующие задачи:

позволить определить смысловые отношения между вершинами синтаксического графа;

среди множества синтаксических структур, представленных одной данной структурой, найти правильную структуру.

Для получения правильной структуры исходную структуру иногда приходится модифицировать. Это происходит, в частности, при опущении служебных слов (см. рис. 7). В этом случае

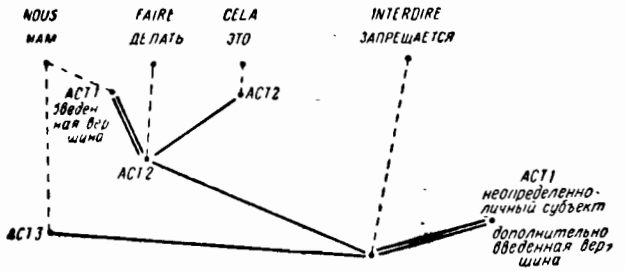


Рис. 7.

в структуру вводятся дополнительные вершины, так чтобы были заполнены все места предикатов (обычно нужное число раз повторяется вхождение уже имеющейся в структуре вершины). Дополнительные вершины вводятся для подразумеваемых аргументов (безличные, неопределенно-личные и т. д.). Для иллюстрации этого рассмотрим пример, изображенный на рис. 7.

Описание грамматики этикетажа (М3)

Чтобы осуществить все названные выше трансформации, мы воспользуемся так называемой грамматикой «этикетажа», метаязык которой описан в [8].

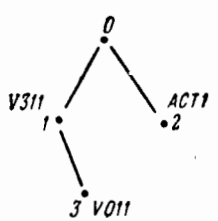


Рис. 8.

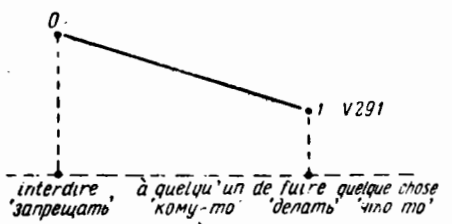


Рис. 9.

Эта грамматика состоит приблизительно из 80 правил. Каждое правило содержит номер, левую часть и правую часть.

Левая часть правила этикетажа (GAUCHE) описывает некий подграф графа зависимости в терминах символов отношений, введенный в [8]. Вершины графа перенумерованы от 0 до n в порядке их появления.

Пример:

GAUCHE (FS, V311 (FR, ACT 1) (FS V011)) [1])

Пример правила грамматики этикетажа

	REGLE 'правило'	R220, SUITE	GIQ25910	2590
	GAUCHE 'левая ч. п.'	((FS, V291)) ('сын, V291'))	GIQ25920	2591
	TABLO		GIQ25930	2592
A	LIGNE	(CE, TRIVA)	GIQ25940	2593
B	LIGNE	(R(FS(BI.. L)))	GIQ25950	2594
BI	LIGNE	((E, V132)U(F, V151))	GIQ25960	2595
C	LIGNE	(R(FS, ACT2))	GIQ25970	2596
D	LIGNE	(R(FS, ACT3))	GIQ25980	2597
	FTABLO		GIQ25990	2598
	CONDIT 'правая ч. п.'		GIQ26000	2599
	STRUCT	((1, BENJ, 2))	GIQ26010	2600
	SYMBO	2((E, V201))	GIQ26020	2601
	SINON	(B, O)AA	GIQ26030	2602
	STRUCT	((O, BENJ, 3))	GIQ26040	2603
AA	SYMBO	((2SBT,., 3))	GIQ26050	2604
	SINON	((C, O)U(D, O))ELIMIN	GIQ26060	2605
	SINON	(A. O)AB	GIQ26070	2606
	SYMBO	1((E, ACT3))	GIQ26080	2607
	SYMBO	3((E, ACT2)) (A, SUITE)	GIQ26090	2608
AB	SYMBO	1((E, ACT2))	GIQ26100	2609
	SYMBO	3((E, ACT3))	GIQ26110	2610
	FREGLE	R220	GIQ26120	2611

Согласно приведенной здесь левой части правила, в графе зависимостей (т. е. в синтаксической структуре) ищется подграф, изображенный на рис. 8.

Правая часть правила этикетажа (DROITE) присваивает новые символы определенным вершинам графа (указанным в левой части) и производит нужное преобразование графа.

В описанном примере левая часть задает подграф, изображенный на рис. 9.

Правая часть (CONDIT) непосредственно преобразует этот подграф, вводит новую вершину 2 и осуществляет поиск вершины 3, если она существует, или же вводит ее в граф, если

[1]) FS — сокращенное *от fils* 'сын'.
FR — сокращенное от *frère* 'брат'. — *Прим. перев.*

она отсутствует. Вершине 2 приписывается этикетка V201 и тот же самый адрес (т. е. то же самое лексическое значение), что и вершине 3. После этой операции получается подграф, изображенный на рис. 10. Заметим, что приписывание этикеток называется «этикетажем», этим и объясняется название. Затем вершинам 1 и 3 в соответствии с кодами вершины 0 приписываются этикетки. В случае, если вершина 0 — это глагол типа *interdire*, 'запрещать', то вершина 3 будет ее ACT3, а вершина 1 — ее ACT2; если же вершина 0 — глагол, типа *prier* 'просить', то этикетки вершин 1 и 3 поменяются местами.

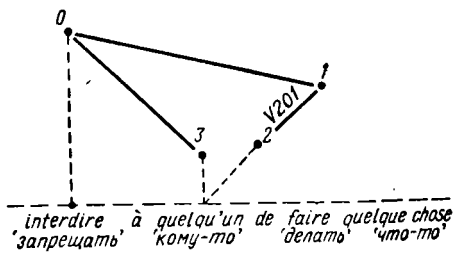


Рис. 10.

Этикетажные коды, упомянутые в приведенном выше примере, представляют собой символы семантических классов, к которым принадлежит соответствующее слово. Эти классы образованы в соответствии с теми признаками, которые отличают один язык от другого. Для русского языка мы выделяем три основных признака: 1) переходность; 2) валентность; 3) предложное управление.

В зависимости от этих признаков мы ввели три словарные классификации: первые две классификации задаются в форме бинарных деревьев (см. ниже), а третья — в форме списка.

Дерево № 1

00	пустой	нейтральный			
01	непустой			непереходный	
02	внешний	активный			процесс
03	внутренний				
04	постоянный	явный			
05	переменный			переходный	
06	возвратный	скрытый			
07	поглощенный				
08	дескриптер	выражение			
09	идея			абстрактный	
10	оператор	понятие			имя
11	сорт				
12	аппелатив	исчислимый			
13	единица			конкретный	
14	твердый	неисчислимый			
15	жидкий				

Дерево № 2

00	отсутствие меры	количество	модельность	обстоятельство
01	мера			
02	манера	конечная цель		
03	мотив			
04	продолжительность	время	протяженность	
05	кратковременность			
06	плоскость	пространство		
07	объем			
08	двухвалентный	прямой	многовалентный	валентность
09	трехвалентный			
10	простой	непрямой		
11	составной			
12	личный	одновалентный	моновалентный	
13	безличный			
14	объект выражен	авалентный		
15	метео			

Заметим, что признаком «обстоятельство» обладают предлоги, наречия, прилагательные и существительные, не являющиеся отглагольными: признаком «валентность» обладают глаголы и отглагольные существительные.

Поясним обозначения:

00 — служебные слова, предлоги, местоимения, простые наречия

01 — определение с подчиненным ACT2

02 — переходное действие с подчиненным ACT2

03 — переходное действие с подчиненным ACT1

04 — переходное действие с подчиненным ACT1, если нет возвратности, и с подчиненным ACT2, если есть возвратность

05 — переходное действие с подчиненным ACTIE2

06 — переходное действие с подчиненным ACT1, если нет возвратности, и ACTIE2, если есть возвратность

07 — переходное действие с подчиненным ACT1 для возвратности и невозвратности

08 — глагол с двумя актантами: ACT1, ACT2.

09 — глагол с тремя актантами: ACT1, ACT2 и ACT3 (выражен в DATIF)

10 — глагол с тремя актантами: ACT1, ACT2 (в DATIF), ACT3 (не в DATIF)

11 — глагол с тремя актантами: ACT1, ACT2 (в ACC), ACT3 (не в DATIF)

12 — глагол с одним актантом: ACT1

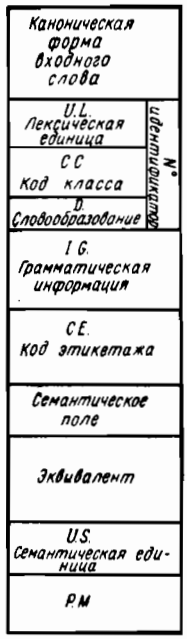
13 — глагол с двумя актантами: ACT1, ACT2 (может иметь определение объекта)

14 — безличные глаголы типа: *il faut* 'надо', *possible* 'возможно'

15 — «метеорологические» глаголы типа: *il pleut* 'идет дождь'.

Словарь

Словарь представлен в следующей форме:



Каноническая форма слова входного языка и эквивалент записываются удобным образом. Единицы UL, CC, D составляют идентификатор слова.

I. G. представляют собой значения грамматических характеристик, с помощью которых строится та или иная словоформа.

C. E. — последовательность символов (цифр или букв), отделяемых друг от друга запятой, которые являются признаками, записанными с помощью деревьев грамматики «этикетажа».

Семантическое поле и код микроглоссария определяют ту область языка, в которой возможно употребление слова с данным эквивалентом.

U. S. — порядковый номер, который определяет номер семантического семейства этого слова (см. стр. 161). Это семейство может состоять из слов, которые имеют следующие функции:

01: Процесс
02: Вещь
03: Активное свойство
04: Способ
05: Характер
06: Пассивное свойство

Пример семантических семейств

(пустые клетки, в случае необходимости, могут быть заполнены при перефразировании)

Вариант \ Лексема		*distinguer* 'различать'	*comprendre* 'понимать'	*reel* 'реальный'	*pour* 'для'	*et* 'и'	*absence* 'отсутствие'
Процесс	01 11	*distinguer*	*comprendre*				
Вещь	02 12	*distinction* 'различие'	*compréhension* 'понимание' *incompréhension* 'непонимание'				*absence*
Активное свойство	03 13	*distinct* 'различный'	*compréhensif* 'понятливый' *incompréhensif* 'непонятливый'				*absent* 'отсутствующий'
Способ	04 14	*distinctement* 'внятно' *indistinctement* 'невнятно'		*réelement* 'реально'	*pour*	*et*	
Характер	05 15			*réalité* 'реальность' *irréalité* 'нереальность'			
Пассивное свойство	06 16	*distingué* 'различаемый'	*compris* 'понимаемый' *incompris* 'непонимаемый'	*reel* 'реальный' *irréel* 'нереальный'	*pour que* 'для того, чтобы'		

4. ВОЗМОЖНЫЕ ПРИЛОЖЕНИЯ ЯП1

Данный ЯП был задуман специально для АП, и поэтому мы подробно остановимся на этом его применении.

Использование ЯП1 в процессе АП

Предложенная запись на ЯП сохраняет содержание текста, устраняя его морфологические и синтаксические особенности. В действительности такая запись вполне соответствует той независимости анализа и синтеза, которая предполагается в модели типа «Смысл ⇔ Текст».

Создавая этот ЯП, мы не стремились к получению точной и глубокой записи содержания. Для нас было достаточно построить такой ЯП, в терминах которого можно описывать структуру текстов, не зависящую от входного и выходного языков,

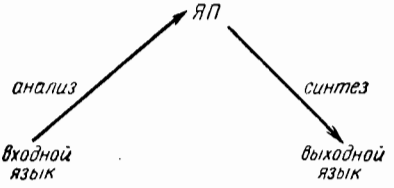


Рис. 11.

устраняя те особенности выражения, которыми характеризуются тексты на этих языках.

При практическом использовании наш ЯП был упрощен. Это объясняется, в частности, известным сходством между языками, что позволяет избежать слишком глубокого анализа и не заниматься устранением некоторых неоднородностей. В системе перевода, отправляясь от записи на ЯП, мы можем синтезировать выражение на выходном языке. Таким образом, анализ и синтез оказываются вполне независимыми, что иллюстрирует схема, изображенная на рис. 11.

Анализ был нами уже рассмотрен; остановимся теперь подробнее на синтезе.

Лексика ЯП

Запись на ЯП должна обеспечить нам построение на выходном языке поверхностной синтаксической структуры (затем по этой структуре строится цепочка реальных слов). Будем предполагать, что выходным языком является французский. Каждой

вершине структуры на ЯП сопоставлена лексическая единица, определенная при анализе входного текста. Каждой лексической единице соответствует так называемая семантическая парадигма из 12 членов (выражений). Каждое из этих выражений характеризуется своей грамматической информацией, которая определяет свойства слова: синтаксическая категория (часть речи), род, число, предложное управление, корень и морфологические свойства.

Одна из целей синтеза — выбор для каждой лексической единицы той синтаксической категории, которая навязывается ограничениями, возникающими в синтезируемой структуре.

Представление выходной синтаксической структуры

В результате последовательных преобразований графа ЯП получается выходной синтаксический граф в виде дерева зависимостей. Однако для получения этого графа мы не пользуемся правилами типа НС — правил, которые имеются в порождающих грамматиках Хомского. Поэтому порядок слов в структуре

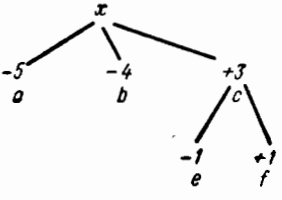


Рис. 12.

не определен, за исключением того, что должны соблюдаться требования проективности. Следовательно, для каждой вершины требуется указать ту позицию, которую она занимает в цепочке по отношению к своему управляющему. Эта позиция кодируется положительным числом, если вершина должна стоять справа от управляющего, и отрицательным числом, если вершина должна стоять слева. Рассмотрим пример на рис. 12. Вершины *a*, *b* и *c* упорядочены по отношению к x: abxc и e и f: ecf. Соблюдение проективности определяет цепочку (abx(ecf)).

Таким образом, выходная поверхностно-синтаксическая структура представляется деревом зависимостей, в котором каждой вершине приписаны следующие характеристики: «вес» (число, характеризующее ее линейную позицию в цепочке), символ ее синтаксической категории и множество грамматических переменных, которые позволяют построить нужную форму соответствующего слова (см. пример на стр. 169).

Грамматика синтеза

Для каждой вершины дерева должна быть определена информация двух типов: ее синтаксическая функция и ее синтаксическая категория. Категорию мы можем определить, только если знаем синтаксическую функцию, что видно из примера на рис. 13.

Предположим, что корень дерева мы сделаем существительным: *explication* 'объяснение'. Тогда глагол *apparaitre* 'появляться' окажется приименным дополнением, что потребует в

Рис. 13.

свою очередь превратить и его в существительное *explication de l'apparition de nucléons* 'объяснение появления нуклонов'.

Если же мы поставили бы в корне глагол *expliquer* 'объяснять', тогда слово *apparaitre* 'появляться' было бы прямым дополнением, которое может быть либо существительным, либо глагольным выражением (*expliquer que...* 'объяснять, что...').

Грамматика синтеза, начиная с корня дерева, применяется к каждой вершине. Для каждой вершины применяются последовательно правила трех типов:

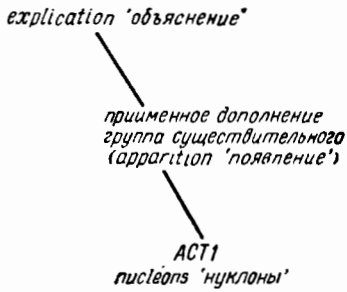
1) Выбор синтаксической категории. Так как синтаксическая функция вершины известна, то определение ее синтаксической категории зависит от словообразовательных возможностей данной лексической единицы и, в известных случаях, от ее окружения.

2) Определение синтаксических функций для различных вершин, зависящих от данной в графе ЯП.

3) Терминальное правило. Это правило, с помощью которого находятся морфологические характеристики. Оно позволяет осуществить в некоторых случаях следующие преобразования: преобразования, необходимые для образования сложных времен, преобразования, связанные со вставкой предлогов и артиклей, преобразования, связанные с введением местоимений-заменителей.

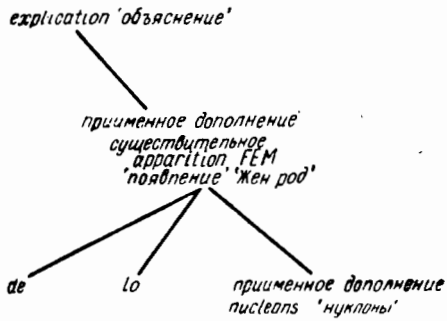
Рассмотрим предыдущий пример. Возьмем вершину *apparaitre* 'появляться'. Перед применением грамматики к этой вершине мы имеем следующую структуру:

После применения правила выбора синтаксической **категории**



После применения правила определения синтаксической функции:

После применения терминального правила



В действительности мы смешали понятие синтаксической функции и понятие веса; это последнее является более точным.

Другие приложения

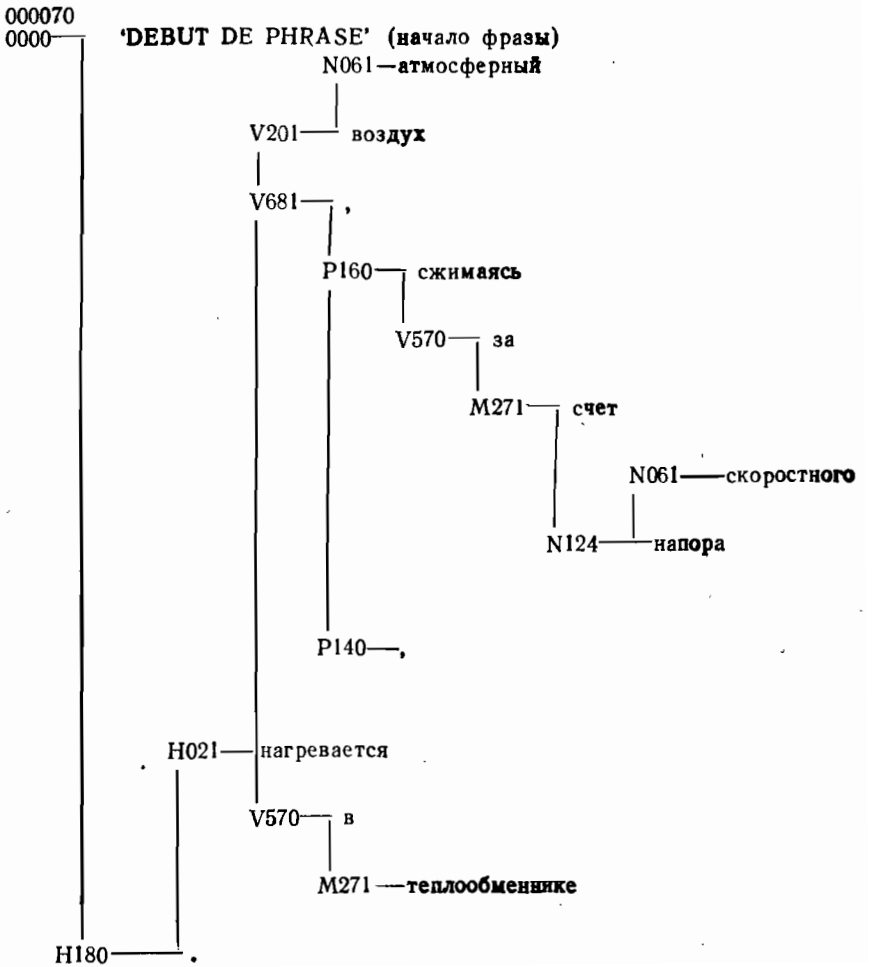
Можно попытаться использовать предложенную запись и связанные с ней системы анализа и синтеза в системах общения с машиной на естественном языке. Так как данная запись дает запись смысла многих эквивалентных фраз, то ее можно рассматривать как каноническую форму всех фраз одного и того же содержания. Если уметь интерпретировать эту запись, то в языке общения человек — машина можно допустить существование всех семантически эквивалентных фраз. Более того, возможность перефразирования путем порождения одной или нескольких фраз позволит осуществить диалог между машиной и пользователем. В частности, в случае когда предложенная фраза неоднозначна и имеет несколько записей, машина может прибегнуть к дополнительному вопросу, чтобы устранить неоднозначность.

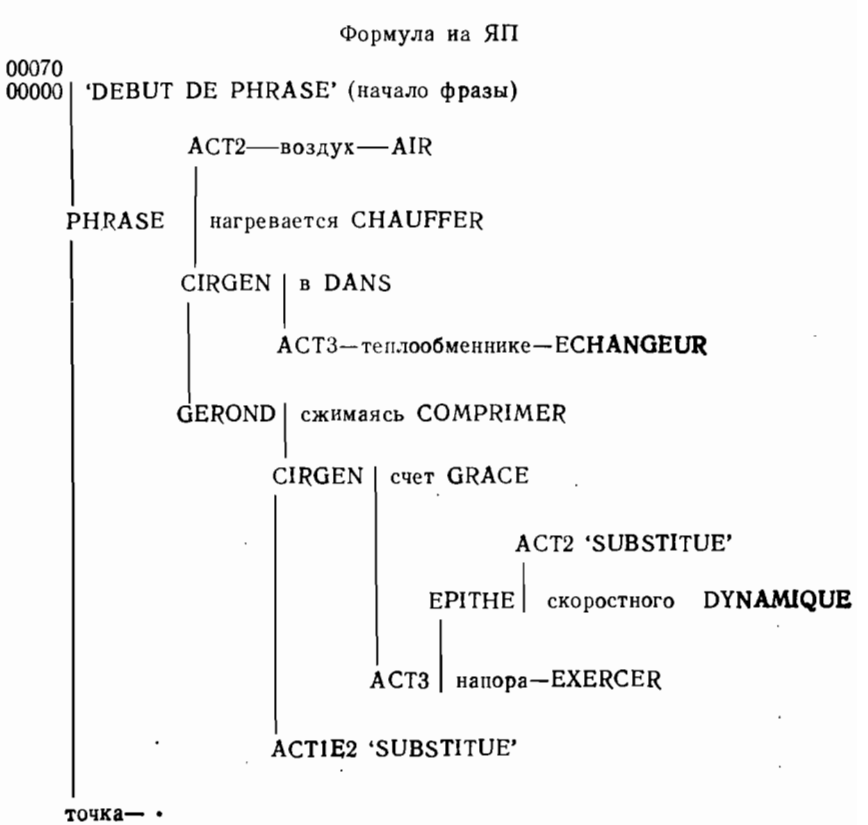
Среди приложений этого типа мы отметим автоматическое наведение справок в картотеке и программированное обучение.

Разумеется, необходимо специально исследовать проблемы интерпретации предложенной записи в рамках новых приложений. В частности, при наведении справок в картотеке может оказаться необходимым, исходя из записи на ЯП, построить некоторую последовательность команд или даже целую программу.

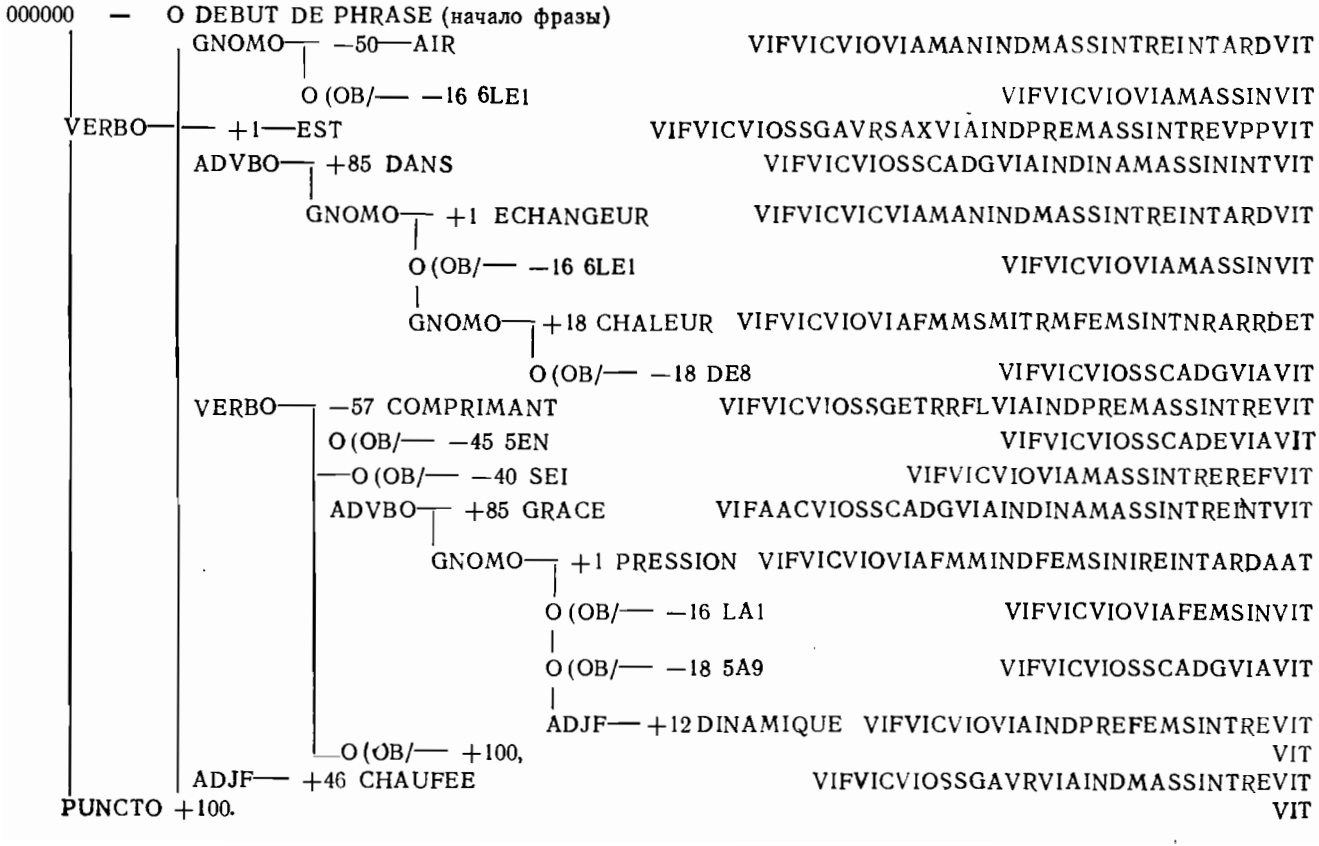
ПРИЛОЖЕНИЕ

Русская синтаксическая структура




Формула на ЯП
00070
00000
'DEBUT DE PHRASE' (начало фразы)
ACT2—воздух—AIR
PHRASE
нагревается CHAUFFER
CIRGEN
в DANS
ACT3—теплообменнике—ECHANGEUR
GEROND
сжимаясь COMPRIMER
CIRGEN
счет GRACE
ACT2 'SUBSTITUE'
EPITHE
скоростного DYNAMIQUE
ACT3
напора—EXERCER
ACT1E2 'SUBSTITUE'
точка— •

Французская синтаксическая структура



СПИСОК ЛИТЕРАТУРЫ

1. Jolkovsky A., Meltchouk Y., Essai d'une théorie sémantique applicable au traitement de langage, Annales de la Conférence Internationale sur le traitement automatique des langues, Grenoble, 1967.
2. Жолковский А., Мельчук И., О семантическом синтезе, *Проблемы кибернетики*, **19** (1967), 177—238.
3. Lamb S., Outline of stratificational grammar, Georgetown University Press, 1966.
4. Vauquois B., Le systéme de traduction automatique du C. E. T. A., Congrès d'EREVAN, Avril 1967.
5. Veillon G., Description du langage pivot du système de Traduction automatique du C. E. T. A.
6. Fuchs C., Pecheux M., Lexis et Métalexis, Linguistique mathématique, Collection de D. Herault (Dunod) (à paraitre).
7. Dupraz M., Rouault J., Lexis-Affirmation-Négation, Etude fondée sur les Classes, Colloque de Balatonszabadi, Septembre 1968.
8. Vauquois B., Veillon G., Un métalangage de grammaires transformationnelles, Proceading of International Conference on Computational Linguistics, Grenoble, 1967.
9*. Вокуа Б., О деятельности центра исследований по автоматическому переводу при Национальном центре научных исследований, Франция, Сб. «Автоматический перевод», «Прогресс», М., 1971.
10*. Мельчук И. А., Об одном эксперименте АП с русского языка на французский в рамках Гренобльской системы, НТИ-2, № 12 (1969), 23—29.

* Литература, рекомендованная переводчиком.

Искусственный интеллект; темы исследований во втором десятилетии развития[1])

Э. А. Фейгенбаум

Цель настоящего сообщения состоит в обзоре последних публикаций, посвященных исследованиям в области искусственного интеллекта, и установлении и оценке основных тенденций, проявляющихся в этих исследованиях. Имея дело с областью исследований, проходящей первые этапы развития и развивающейся столь же быстро, как и та, о которой пойдет речь, но которая не обладает к тому же прочным теоретическим фундаментом, не так просто разрешить обе проблемы — и выделение тенденции, и попытки их оценивать.

Наиболее примечательным изо всех научных докладов, на которых я присутствовал, мне кажется выступление-экспромт моего коллеги профессора Дж. Лидерберга перед неформальной Станфордской исследовательской группой. Его доклад был посвящен проблеме, которую можно было бы определить как «обработка информации посредством РНК и ДНК». Хотя его интересы весьма разносторонни, тема того дня явно была его коронной — мало кто мог сравниться с ним в этой области.

Доклад этот, подобно двойной спирали, развивал две переплетающиеся темы. В рамках одной из них сообщались основные сведения о характере проведенных экспериментов и полученных в их ходе результатах. Другая представляла личную оценку Лидербергом содержательности отдельных экспериментов и научной ценности их результатов — обсуждалась потенциальная плодотворность продолжения определенных исследований и вероятная бесперспективность других, сравнение ключевых проблем, требующих разрешения, с самими по себе интересными, но по существу побочными проблемами и много других рассуждений оценочного характера. В общем доклад

---

[1]) Feigenbaum E. A., Artificial intelligence: themes in the second decade, Stanford Artificial Intelligence Project, Stanford University, Memo № 67, August 1968.

содержал широкий спектр комментариев — от покоившихся на строго научной основе («практически доказано») до отражающих субъективные и интуитивные ощущения, которые может дать длительный опыт работы в данной области («Я чувствую, что...»). В результате в голове слушателя возникала «карта» лабиринта «задача — эксперимент — теория», отражавшая текущее состояние исследований в этой области молекулярной биологии и содержавшая оценки настоящего и будущего для различных путей через этот лабиринт.

Этот двойственный подход я выбрал в качестве эталона для нашего обзора. Он не должен сводиться ко всеобъемлющему набору библиографических указателей. Существенным является тщательный отбор материала, основанный в какой-то степени на личной оценке его уместности и важности; конструктивную роль играют и субъективные критерии достоверности и плодотворности отдельных аспектов исследований.

Так как я делаю доклад, а не пишу книгу, то у меня нет возможности рассмотреть все направления исследований, которые с полным основанием можно отнести к «исследованиям в области искусственного интеллекта». Выбранные мной названия разделов не следует рассматривать как оглавление дешифратора к понятию «искусственный интеллект». Примечательные подразделы, обладающие собственными научными традициями и авторитетными публикациями, были оставлены «на произвол судьбы». Так, например, важное направление, называющее себя «исследования в области распознавания образов», отсутствует в нашем обзоре, так же как и лингвистика, машинный перевод, бионика, нейрофизиологические модели процесса обработки информации и некоторые другие.

Предметом настоящего сообщения являются эвристическое программирование, решение проблем и примыкающие к этим темам модели процесса обучения. В рамках этой тематики мы концентрируем внимание на исследованиях, проводившихся в 1963—1968 годах, так как мне кажется, что сборник «Вычислительные машины и мышление» [21] содержит соответствующий материал, характеризующий период 1956—1962 годов. Отметим, что ниже для обозначения понятия «искусственный интеллект» будет использоваться аббревиатура ИИ.

ОБЩАЯ ХАРАКТЕРИСТИКА ИССЛЕДОВАНИЯ В ОБЛАСТИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

«Взрыв», характеризующийся увеличением количества ставящихся задач, запускаемых исследовательских проектов и публикуемых сообщений, в первую очередь вызывает интерес. Не-

смотря на этот бурный рост уровень качества исследований оставался, с моей точки зрения, достаточно высоким [1]).

С самого начала исследований в области ИИ, к которым Ньюэлл и Саймон приступили в Технологическом институте Карнеги в 1955—1956 годах (их я при любых обстоятельствах рассматриваю как «НАЧАЛО»), они называли их изучением «сложных процессов обработки информации». И поныне они пользуются этим названием, хотя не одна исследовательская программа родилась с тех пор под названием «проект ИИ». Следует отдать должное предусмотрительности, проявленной в в этом отношении Ньюэллом и Саймоном. Пограничные области исследований ИИ настолько переплетены с исследованиями, выполняемыми в рамках теории и приложений вычислительных машин, что их вполне можно характеризовать понятием «сложные процессы обработки информации». Например, следует ли работы с информационными программами, обеспечивающими получение ответов на конкретные вопросы, продолжать считать исследованиями в области искусственного интеллекта или их нужно рассматривать как естественное развитие направления, определяемого понятием «информационный поиск»? Являются ли попытки совершенствовать программы решения проблем, придавая им способность создавать операционные системы [24], изучением ИИ или они представляют собой исследования в области системного программирования? Относится ли программа, обеспечивающая выдвижение гипотез при анализе масс-спектров органических молекул, к ИИ или химии [22]?

Эти вопросы не столь тривиальны, как подразумевает недвусмысленный ответ: «А какая, собственно, разница!». Мы сталкиваемся здесь практически с общей для всех направлений науки тенденцией дробления тем исследований по мере лучшего уяснения возникающих задач и включения в работы практиков, стремящихся применить полученные результаты, причем

[1]) Ряд наблюдателей отмечает падение продуктивности исследований в 1960—1963 годах, и оценка эта, по всей вероятности, вполне обоснованна. Я считаю, что это вызвано следующими факторами: сдвиг интересов основных исследовательских центров в сторону технических аспектов создания вычислительных устройств (например, работы по созданию вычислительной машины с множественным доступом в Массачусетском технологическом институте, проект вычислительной системы с разделением времени, осуществлявшийся в Станфорде под руководством Маккарти); необходимость переоценки смысла и значения результатов, полученных в конце пятидесятых годов; переключение внимания на такие задачи, разрешение которых требует длительного времени (например, анализ естественных языков, «роботы», проблема представления); организация в высших учебных заведениях факультетов вычислительной техники, составление программ, осуществление учебных курсов и т. п. поглотили значительную долю сил людей, творчески работавших в этой области. Каждому из этих факторов принадлежит свой «вклад» в снижение продуктивности исследований периода 1963—1968 годов.

«успех» в этих случаях приписывается отпочковавшимся дисциплинам, а вклад «делящейся» области знания игнорируется.

Исследования в области ИИ демонстрируют многочисленные примеры развития подобных процессов. Рассмотрим распознавание символов (т. е. то, что делают те самые «оптические считывающие устройства»). Большинство ранних работ по распознаванию образов были посвящены распознаванию символов, которое рассматривалось как интересная исходная задача. Это исследование, относящееся приблизительно к 1955 году, не так пыталось установить: «Каким образом можно добиться того, чтобы вычислительная машина надежно считывала алфавитные символы?», как было посвящено выяснению: «Какие представляющие интерес паттерны поведения в состоянии воспроизводить вычислительная машина, кроме решения рутинных повседневных задач (типа вычисления таблиц значений функции и составления платежных ведомостей)?» [62]. Изучение более общих проблем вдохновляло первые работы по программам решения проблем. В конце концов эта ветвь исследований ушла в прикладную область создания устройств, предназначенных для распознавания символов, и тем самым выпала практически из области ИИ. Обработка списков представляет еще один пример. Насколько я понимаю, она была введена как частная методика Ньюэллом, Шоу и Саймоном, с тем чтобы снять затруднения, возникавшие при организации памяти и иерархического (и рекурсивного) управления процессом обработки информации в программе «Логик-теоретик» и первом варианте «Универсального решателя задач». Дальнейшее развитие «списковая» методика получила в работах другого исследователя, работавшего в области ИИ, — Маккарти (LISP)[1]). Он предложил дальнейшие изменения («сквозные» списки, «связанные» списки, симметричные списки и т. д.), что позволило осуществить переход от «чего-то, чем занимаются эти исследователи ИИ» к «методам создания систем математического обеспечения». В настоящее время использование списочных структур является стандартным приемом для многих частных разделов, например, при реализации компиляторов и операционных систем.

Процветание любой дисциплины определяется ее успехами, в частности от этого зависит привлечение талантов и субсидирующих исследования фондов. Существует опасность, что область ИИ, монополизируя те проблемы, которые еще не решены, те, которые еще недостаточно поняты, а также проблемы, в которых провал является возмездием за их предварительную

[1]) Аббревиатура, употребляемая для обозначения названия List processing, — языка для обработки списков. — *Прим. перев.*

хищническую разработку, обретет репутацию «неудачливого» раздела теории вычислительных систем и прибежища прожекторов, «счастье которым суждено лишь на том свете». Имеется множество свидетельств развития этой тенденции; этот процесс мне кажется крайне неблагоприятным и несоответствующим истинному положению.

И наконец, бурный рост исследований ИИ сделал менее определенной «незримую корпорацию» работающих в этой области; теперь в рамках Ассоциации по вычислительной технике действует специальная группа «Искусственный интеллект», подготавливается к выпуску новый журнал [1]) и организовывается международная конференция.

В ПОИСКАХ ОБЩИХ МЕТОДОВ

Так называется сделанный Ньюэллом и Эрнстом на конгрессе IFIP 1965 года доклад, который стоит того, чтобы его перечитали [48]. С тех пор многие присоединились к этим поискам. Существуют две дороги — верхняя и нижняя.

Идущие верхней дорогой стремятся определить разрешающую систему самого общего типа, включающую набор методов решения проблем, не ориентированных на задачи конкретного типа, и позволяющую решать задачи, относящиеся к широкому диапазону классов. В этом случае решающее значение приобретает проблема внутреннего представления в том языке, в котором будут «работать» общие методы решения проблем. Если способ представления является достаточно общим для того, чтобы любой новый объект, предлагаемый системе, мог быть избавлен от мучительной переделки, то могут ли сами методы решения проблем и связанные с их реализацией процедуры быть настолько общими, чтобы при этом не понижалась их разрешающая способность? Мы не достигли еще удовлетворительной степени понимания проблемы общности метода решения и представления задачи. Анализ существующих программ решения проблем, как, впрочем, и обычный здравый смысл, показывают, что здесь действует некоторая разновидность «закона природы», связывающего обратнопропорциональной зависимостью степень общности метода решения (размер области применения) и его мощность (возможность получения решения, эффективность и т. п.) и прямопропорциональной зависимостью мощность метода с его специализацией (информация, выражающая специфику задачи). Нам не известно, каким образом можно строить системы решения проблем, способные воспринимать

[1]) Упоминаемый журнал выпускается с 1971 г. под названием Artificial Intelligence. — *Прим. перев.*

задачи в самом общем виде и преобразовывать их последовательно, по мере получения в процессе решения информации, воспроизводящей особенности конкретной задачи, к виду более специфическому и обеспечивающему большую разрешающую способность. Имеется один подробно рассмотренный пример системы такого рода [4], однако о реализации подобных систем ничего не известно.

Система GPS в одиночестве шествовала верхней дорогой в течение почти десяти лет и доказала жизнеспособность направления исследований, целью которого является отыскание общего метода. Новая монография Эрнста и Ньюэлла [17], содержащая анализ успехов и затруднений, которыми сопровождалось применение системы GPS при решении самых разнообразных задач, свидетельствует завершение первого этапа приключений этой системы. Совсем недавно система GPS начала, уже независимо от своих «родителей», «вторую» жизнь. Так, например, структура системы GPS в чуть более общем виде была воспроизведена в программе «Дедуктивная система на языке ФОРТРАН» [54]; она же в существенно измененном виде была реализована в программе «Разбиение графов» [15, 16]. Еще один вариант системы GPS появился только что в Швеции [61].

Идущие нижней дорогой ищут не универсальные системы решения проблем, а теоремы и обобщения метода, связанные с основными принципами, общими для некоторого класса программ решения проблем. Наилучшей иллюстрацией подобной тенденции является, пожалуй, принцип эвристического поиска.

Как и десять лет назад, краеугольным камнем исследований ИИ остается проблема эвристического поиска. Некоторый генератор используется (или может быть использован) для построения дерева «попыток» (иначе — подзадач, преобразований, потенциальных возможностей, решений, альтернатив и следствий и т. д.). Решения (различным образом определенные) находятся на некоторых (неизвестных) траекториях пути по дереву на некоторых (неизвестных) расстояниях от исходной точки. «Проблема» состоит в отыскании решения. Для любой нетривиальной задачи пространство поиска очень велико. Эвристические правила и процедуры используются для непосредственного поиска, для ограничения его продолжительности, уменьшения ветвления дерева и т. д.

В то время как часть этого аппарата поиска по дереву полностью ориентирована на конкретную задачу, его остальные элементы могут оставаться достаточно общими для того, чтобы покрывать некоторую агломерацию систем, в которых используется эвристическая процедура поиска. Классическим в этом отношении примером служит так называемая «альфа-бета» про-

цедура [70, 60]. Использование ее является «очевидным», если организация процедуры поиска тщательно продумана. Ньюэлл, Шоу и Саймон использовали этот принцип еще в своей шахматной программе 1958 года, причем он настолько органично входил в основную процедуру, что они не сочли необходимым специально выносить его «на суд общественности».

Однако очевидное для одних не всегда столь же очевидно другим. Любой исследователь, создающий новую программу, не должен, «вступив в связь» с эвристической процедурой поиска, пренебрегать (или, что еще хуже, пропускать по неведению) стандартными процедурами, облегчающими процесс поиска, особенно наиболее тонкими и квалифицированными из них.

Все вышеперечисленное уже сформировалось в небольшую, но растущую область знания. Заслуживают внимания исследования Слэгла, особенно его «Многоцелевая система» [70], представляющая по существу «конгломерат» подобных методов. Кроме того, интерес представляют статьи Нилсона «Траектории минимальной стоимости» [50], Флойда «Недетерминированные алгоритмы» [23] и Голомба и Бомерта «Программирование с коррекцией по принципу обратной связи» [25].

Можно сказать, что идущие нижней дорогой заняты, в известном смысле, созданием «инструментов», но процесс этот часто носит весьма абстрактный характер и выглядит очень элегантно.

РОБОТЫ

История зафиксирует, что в году 1968 роботы, создававшиеся в трех ведущих в области исследований ИИ лабораториях, состояли из следующих частей:

a) сложного рецептора (обычно телевизионной камеры), посылающего афферентные сигналы в ...;

b) достаточно «мощной» вычислительной машины, снабженной памятью большого объема на магнитных сердечниках, набором программ, обеспечивающих анализ афферентных видеосигналов и принятие решений, связанных с исполнительными действиями ...;

c) механического манипулятора типа «плечо — рука» или снабженной электроприводом тележки.

Интенсивные усилия, направленные на разработку управляемых вычислительной машиной систем типа «глаз — рука» и «глаз — тележка», представляют, с моей точки зрения, самое неожиданное событие в мире исследований ИИ в период 1963—1968 годов.

Начало доскональным исследованиям этой проблемы положила диссертация Эрнста, посвященная управляемой

вычислительной машиной механической руке (МН—1) [18]. Он создал интересную эвристическую программу, обеспечивающую решение проблем, связанных с работой манипулятора в реальной среде. Система МН—1 отличалась почти полной «слепотой», но могла запомнить символическое внутреннее представление внешней ситуации («модель») в языке, использовавшемся для реализации процесса решений проблем. Плодотворным вкладом в исследования оптической («глазной») обработки информации является часто цитируемая работа Робертса [58], посвященная трехмерному восприятию твердых тел при подаче на вход информационной системы двумерного изображения.

Три осуществляемые в настоящее время исследовательских проекта, посвященные созданию робота, близки друг другу. Это Станфордский проект «Глаз — рука» (Маккарти и др.), проект «Глаз — рука» Массачусетского технологического института (Минский и Пейперт) и проект «Робот» Станфордского научно-исследовательского института (Нильсон, Рейфл, Розен и др.).

Было опубликовано довольно мало сведений, относящихся к этим проектам, и поэтому то, что о них известно, носит противоречивый и порой неожиданный характер.

Как можно предположить, проектирование, реализация и применение робота ставят целый ряд сложных и часто связанных с большими затратами технических и эксплуатационных проблем. Для исследователей, работающих в этой области, преодоление подобных затруднений представляет неизбежную, но чаще невознаграждаемую, чем вознаграждаемую, прелюдию, поскольку эта деятельность не способствует разрешению тех проблем ИИ, которые составляют смысл существования этих исследований. Зачем же тогда строить эти устройства? Почему бы не моделировать робот и ту среду, в которой он должен действовать? Действительно, исследовательская группа Станфордского исследовательского института провела полезную работу, смоделировав поведение одного из вариантов своего робота в упрощенной среде. (Имеется посвященный этой работе фильм.) Следовательно, подобный подход может быть использован, и поднятые выше вопросы правомерны.

Ответ на эти вопросы выглядит следующим образом. Группа Станфордского исследовательского института считает, что наиболее неудовлетворительным элементом их модели является воспроизведение среды. Кроме того, они указывают, что девяносто процентов усилий, затраченных их специалистами на моделирование, было посвящено именно этой части модели. Оказалось крайне сложным «вложить в вычислительную машину все то разнообразие свойств среды, которое необходимо роботу с хорошими адаптивными характеристиками в качестве стимула

для демонстрации интересных паттернов поведения». Легче и дешевле построить робота «во плоти» для того, чтобы выяснить в какой информации из реального мира он нуждается, чем синтезировать и воспроизвести на вычислительной машине представительную модель. Грубо говоря, точка зрения группы Станфордского исследовательского института сводится к тому, что наиболее экономичным и эффективным хранилищем информации о реальном мире является сам реальный мир.

Задача создания робота предоставляет, по моему убеждению, возможность изучить целый ряд проблем первостепенной важности для исследований ИИ, в том числе: формирование и планирование стратегий; способы представления ситуаций при решении проблем и последующее видоизменение этого представления по мере поступления новой информации; обработка информации при зрительном восприятии образов. Только группа Станфордского исследовательского института, единственная из трех разрабатывающих эту тему, опубликовала статьи, трактующие особенности и цели этого исследования с более общих позиций [57, 59].

Как в Массачусетском технологическом институте, так и в Станфордском университете проводилась разработка программ, обеспечивающих управление манипуляторами «плечо — рука» самых различных типов, от простейших до самых сложных, от антропоморфических до сугубо неантропоморфических. Ни один из более сложных манипуляторов не кажется достаточно удачным, хотя публикации, посвященные анализу соответствующих достижений и неудач, отсутствуют.

Во всех этих проектах важную роль играют программы, обеспечивающие восприятие зрительных образов. Большая часть усилий, связанных с составлением программ, была затрачена на поиск средств и методов, расширяющих возможности разрешения этой части задачи. Задача восприятия зрительного образа заключается в следующем: имеется телевизионное изображение объекта (цифровое), которое может быть введено в запоминающее устройство вычислительной машины; необходимо осуществить сканирование этого изображения и обработку его результатов таким образом, чтобы получить символическое описание элементов, входящих в изображение, и их взаимосвязей. Газман из Массачусетского технологического института [29] рассмотрел задачу анализа изображения в довольно общем виде (телевизионная камера исключена из рассмотрения, исследуется символическое представление подаваемого на вход изображения в условиях отсутствия помех), добившись очевидного успеха. Кстати сказать, работы Газмана должны заинтересовать психологов, занимающихся моделями процессов визуального восприятия у человека.

Смысл «игры» все-таки заключается в целостном поведении, реализуемом при решении задачи. Установки «рука — глаз — вычислительная машина», созданные как в Массачусетском технологическом институте, так и в Станфордском университете, были способны продемонстрировать несколько паттернов поведения, связанного с выбором отдельных кубиков и постройкой из них различных сооружений. Работа, выполненная в Станфорде, излагается в докладе, приведенном в трудах конгресса IFIP 68 [53]; мне не удалось обнаружить публикации, в которой бы описывалась система МТИ, способная составлять «что-то» из кубиков.

Итак, вы хотите построить робот? Потерпите! Три могучие группы, с одаренностью и финансовыми возможностями которых едва ли можно тягаться, не разобрались еще со всеми первоочередными проблемами. Определить и оценить их, а также подготовить соответствующие публикации удастся, очевидно, в течение следующих двух лет. В настоящее время деятельность, развертывающаяся в рамках упоминавшихся нами проектов, почти полностью посвящена созданию аппаратуры и ее отладке. По сей день осталось недоказанным, что задача создания робота представляет собой рациональное и адекватное направление для разрешения общих проблем исследований ИИ.

ДОКАЗАТЕЛЬСТВО ТЕОРЕМ

Так как Робинсон специально выступает на нашем конгрессе с обзором, посвященным «автоматическому» доказательству теорем, мне не следует касаться этого материала. Однако несколько вопросов заслуживают упоминания, а ряд замечаний представляется уместным.

Многие, работающие в этой области, упорно называют свои программы, предназначенные для доказательства теорем, программами дедукции (например, упоминавшаяся выше «Дедуктивная система на языке ФОРТРАН — DEDUCOM» [69]; термин «дедуктивная информационно-поисковая программа»; в монографии Ханта [32] подобная подмена понятия встречается неоднократно). Все это — терминологические ошибки. Если внимательно проследить, каким образом эти программы отыскивают доказательства, становится очевидным, что в них реализуется значительно больше, чем чистая дедукция. Когда программа доказательства теорем использует некоторое правило вывода, например modus ponens [1]), реализуя пробный шаг, то

[1]) Так называемое «правило вывода» — аксиома исчисления высказываний $\frac{A, A \supset B}{B}$, устанавливающая, что формула B истинна, если истинность формул A и $A \supset B$ была установлена. — *Прим. перев.*

совершенно очевидно, что в ней воспроизводится примитивная дедукция. Поиск же доказательства не есть дедукция. По существу в этой терминологической ошибке выражается тенденция подменять непосредственное исследование основных задач, например встречающихся при попытке воссоединить отдельные части некоторой области (как в случае эвристического поиска), доскональным изучением основных приемов, заложенных в продемонстрировавшую свою эффективность программу. Терминологическая практика, против которой я возражаю, далеко не безобидна, поскольку вводит неправильную классификацию направлений исследования.

Я думаю, что предпочтение следует отдать определению, введенному Амарелом, — «задачи, заключающиеся в выводе», поскольку оно является точным, понятным и содержательным. Определение «процессы поиска» мне кажется приемлемым, и его употребление может быть целесообразным при описании реализаций, приводящих к установлению способа доказательства теоремы (или определению очередного хода в некоторой шахматной позиции, или выдвижению химической гипотезы, объясняющей результаты масс-спектрографического анализа, и т. д.).

В последние годы много внимания было уделено предложенному Робинсон методу резолюции, предназначенному для доказательства теорем в исчислении предикатов. К сожалению, одновременно распространялось представление о том, что метод резолюции кладет конец «миру» эвристического поиска вместе со всеми его хаотическими предположениями и неупорядоченностями. И снова неверное определение, возникшее в результате неправильно понятых метода резолюции, либо известных эвристических программ поиска доказательства, а может быть и того, и другого, приводит к неправильной классификации всей области исследования. Метод резолюции устанавливает систематический формальный аппарат, обеспечивающий полноту набора объектов доказательства, но он сам по себе никак не связан с хорошо известной и неизбежно возникающей проблемой множества форм выражения задачи [58а]. Итак, наложение различных стратегий поиска дало эффективный способ решения проблем, предусматривающий использование резолюции (например, «блочный приоритет», «опорное множество»). В результате оказывается, что процессы, воспроизводимые этими программами, во многом аналогичны (в некоторых случаях идентичны) реализуемым в программах нахождения доказательств, базирующихся на эвристическом поиске; в свое время считалось, что их отличия чрезвычайно велики [17]. Я уверен, что в грядущем десятилетии мы еще многое услышим по этому поводу.

В 1959 году Маккарти предложил новый класс программ («советчики»), которые должны оценивать окружающий мир на уровне здравого смысла. «Окружающий мир» при этом задается однородным представлением в исчислении предикатов. Решение предъявленных задач должно осуществляться в виде поиска доказательств в пространстве представлений. Недавно Грин и Рейфл [26] использовали процедуру метода резолюции в качестве отыскивающей доказательства «машины» в информационно-поисковой программе (программа дает «советы», отвечает на вопросы и сообщает необходимые сведения). И сама идея, и ее реализация представляют интерес; однако, как только что указывалось, эффективность и практичность подобной системы неизбежно связаны с необходимостью предусмотреть некоторую эвристическую процедуру, направляющую процесс поиска.

ИГРОВЫЕ ПРОГРАММЫ

Некоторые исследователи занимались в первом десятилетии программами, способными играть в шахматы, потому что шахматы обеспечивают интересную и сложную среду, позволяя изучать процессы решения проблем (краеугольный камень этой линии исследования — блестящая статья Ньюэлла и Саймона [49], в которой пример из обычной шахматной партии тщательно исследуется с точки зрения того, что можно выяснить о процессах решений проблем в шахматах, создавая играющие в шахматы программы). Были и другие, которые занимались такими программами, отвечая на «вызов», брошенный шахматами, — ведь уже много веков эта игра является основным интеллектуальным развлечением.

На такой основе работает группа Гринблатта. Им удалось составить программу, играющую в шахматы очень хорошо (однако еще не профессионально). Я знаком только с одной статьей, посвященной программе Гринблатта [27]. В ней, помимо краткого общего описания, приводятся примеры партий, «сыгранных» программой. В статье сообщается, что, играя с шахматистами по обычным турнирным правилам, программа победила в турнире класса D, состоявшемся в середине 1967 года в Бостоне; в ходе турнира программа обыграла шахматиста класса C [1]). Кроме того, отмечается существенное улучшение результатов программы к настоящему времени. Очевидно, наиболее славной победой программы является изящный выигрыш у Хьюберта Дрейфуса.

[1]) Хотя принципы американской шахматной классификации существенно отличаются от принятых в СССР (точнее, в США нет единой системы классификации шахматистов), можно считать, что группа C приблизительно соответствует нашему третьему разряду, а группа D — четвертой категории — *Прим. перев.*

Почему же эта программа играет настолько лучше предыдущих? Я не знаю ни одного человека, способного убедительно объяснить этот феномен (частично из-за скудности информации о программе). С моей точки зрения, в программу не заложены принципиально новые идеи относительно организации шахматных программ. В этой программе содержится значительно больше, чем в любой из предыдущих, конкретных сведений о шахматах. Машинное время, талантливейшие программисты, фантастические механизмы второго десятилетия (вывод данных на экран электронно-лучевой трубки, работа в режиме диалога с машиной, большой объем памяти на магнитных сердечниках) — все это было в изобилии: пациент получил любви и заботы больше, чем все предыдущие (как вы должны помнить, все остальные пациенты были выписаны в весьма печальном состоянии; большинство из них умерло). Наконец, имелась возможность, использовав сидящего за пультом управления человека, организовать прекрасный «обучающий контур». Ошибки анализировались и быстро устранялись с помощью подпрограмм изменения программы и/или введения дополнительной шахматной информации. Влияние новых подпрограмм и дополнительной информации на работу системы, если оно имело место, можно было быстро обнаружить; при возникновении каких-либо осложнений это влияние устранялось. Таким способом очень удобно улучшать (или обучать) программу, и им целесообразно пользоваться в тех случаях, когда вы больше заинтересованы в высоком качестве работы системы при разрешении конкретной задачи, чем в получении общих моделей, обеспечивающих реализацию процессов решения проблем. Я считаю, что этот метод, хотя и в более сложных формах, будет широко использоваться во втором десятилетии.

В первом международном соревновании между шахматными программами победила программа, созданная в СССР, в московском Институте теоретической и экспериментальной физики (Адельсон-Вельский и др.; подробные публикации по этому поводу отсутствуют). Проигравшей стороной в этом матче была старая программа МТИ, составленная Котоком [37] и несколько модифицированная Станфордской группой Маккарти. Запись партии опубликована в «SICART Bulletin» [56]. Игру и той, и другой программы на фоне средних результатов программы Гринблатта нельзя признать удовлетворительной.

Хорошо известная шашечная программа Сэмуэля была подвергнута существенной переделке [60] и теперь весьма близка к совершенству в своей области. Главные изменения были внесены в процедуры обучения, и мы несколько позже обсудим их.

Уильямс работает над проблемой моделирования общих принципов, используемых людьми в распространенных настольных и карточных играх [74]. В его программе («General Game Playing Program» — «Игровая программа общего типа») в качестве входной информации используется описание объектов, применяемых в игре; правила игры, взятые из руководства Морхеда и Мотт-Смита (Morehead A. H., Mott-Smith G., Hoyle's Rules of Games, New American Library of World Literature, New York, 1963), непосредственно преобразуются в соответствующий входной язык. Программа по меньшей мере способна к действиям «по правилам» в большинстве игр, описанных в руководстве.

МАШИННОЕ ОБУЧЕНИЕ (В ЧАСТНОСТИ, МЕХАНИЗМЫ ОБУЧЕНИЯ) [1])

До сих пор в области ИИ значение машинного обучения для решения проблем осознавалось весьма слабо. Единственную, по существу за много лет заслуживающую упоминания работу представляет известная шахматная программа Сэмюэля и использованная в ней процедура обучения. (Большой интерес в свое время вызвала предложенная Ньюэллом, Шоу и Саймоном система обучения, предназначенная для программы GPS, однако она осталась нереализованной.) Как это ни удивительно, и в наши дни ситуация остается прежней.

В одной из последних работ Сэмюэля [60] описываются основные изменения, внесенные в систему оценки позиции, и соответствующие процедуры обучения шашечной программы. Вместо оценки, основанной на использовании функции, представляемой линейным многочленом, вводится сложная система нелинейных оценок. Свойства позиции, которые раньше фигурировали в виде членов многочлена, в новой системе объединяются на основе их (выявленной) взаимозависимости. Свойства эти характеризуются с помощью «грубой» системы оценок, например трех-, пяти- или семизначных функций. (Довольно старая идея; Саймон обосновал ее еще в 1951 году применительно к шахматам. Она используется в шахматной программе, разработанной Ньюэллом, Шоу и Саймоном.) Векторные величины, характеризующие свойства позиции, используются как входы в таблицах показателей первого уровня. Затем осуществляется просмотр таблицы; результат относится к одному из пяти «грубых» классов и засылается в таблицы показателей второго уровня (просмотр осуществляется по всем таблицам первого уровня). Эта операция повторяется еще раз на третьем уровне и результат, полученный при обработке этой «верхней»

[1]) Не рассматриваются процессы с участием человека-учителя.

таблицы показателей, используется в качестве значения оценки. Сэмюэль показал, что подобная иерархическая схема обладает существенными преимуществами по сравнению с (уже достаточно удачной) оценкой с помощью линейного многочлена.

Назвав один из разделов своей работы «Эвристический поиск эвристик», Сэмюэль указывает, что «принятие решения о выборе эвристики также представляет собой такую задачу, решение которой возможно лишь посредством использования эвристической процедуры, так как она в действительности еще сложнее, чем проблемы, возникающие в связи с собственно игрой». Интересный случай выявления эвристики с помощью эвристической программы рассмотрен в диссертационном исследовании Уотермана (Станфорд) [72]. Среда задачи представлена игрой в покер. Эвристики при этом не являются программами в обычном смысле, а «выносятся на поверхность» в виде правил поведения в конкретных ситуациях, записываемых в «Номенклатуру готовой продукции». Первоначально этот перечень содержит лишь «исходное» правило: «какова бы ни была ситуация — играй случайным образом». В принципе таблица правил может изменяться четырьмя способами. Могут быть добавлены ситуационные правила. Может быть изменен порядок применения правил (так как таблица просматривается сверху вниз в жесткой фиксированной последовательности, то это обстоятельство может привести к существенно разным видам поведения). Правило может быть обобщено с помощью изменения его ситуационной части таким образом, что не будут приниматься во внимание некоторые аспекты игровых ситуаций и, следовательно, правило будет «покрывать» большее их количество. С другой стороны, ситуационная часть правила может быть специализирована, с тем чтобы ввести большую степень дифференциации игровых ситуаций; в результате область применения правила уменьшается. Подобная процедура обучения хорошо работает в целом ряде обучающихся систем.

ОДНО ЗАМЕЧАНИЕ ПО ХОДУ: НЕКОТОРЫЕ ОСОБЕННОСТИ, СВЯЗАННЫЕ С СОБСТВЕННО ПРОГРАММИРОВАНИЕМ

В своей шашечной программе Сэмюэль применил схему обучения с «механическим» запоминанием, которая предусматривала накопление в памяти множества позиций и соответствующих им оценок, полученных в процессе работы программы. Если «заученные» позиции возникали снова, то можно было воспользоваться готовой оценкой, не вычисляя ее каждый раз заново. В программах доказательства теорем «механическое» запоминание использовалось с аналогичными целями при фиксации системы доказательств. Подобные процедуры

предусмотрены и во многих других программах, например в рассматриваемой ниже программе «Heuristic DENDRAL».

Запоминать или вычислять каждый раз заново — эта классическая проблема является достаточно общей. Мы предвидим, что наступит день, когда программы станут столь совершенны, что смогут решать, в каких случаях для получения значений оценочной функции следует обращаться к таблицам, сформированным посредством «механического» запоминания, а в каких — вычислять их (это решение будет приниматься исходя из анализа случаев применения оценочной функции и ее параметров, проводимого по мере поступления этой информации в процессе работы). Недавно в этом направлении был сделан первый шаг — в разработанный в Эдинбурге язык POP-2 были включены «запоминаемые функции» [40]. Однако и при наличии «запоминаемых функций» программист сам должен принимать решения, и в будущем мы ждем еще большей «сэмюэлизации» систем программирования, которая облегчит выбор «запоминать или рассчитывать».

Саймон как-то составил программу решения проблем («Эвристический компилятор»), которая была способна составлять простенькие программы на языке ИПЛ-V на основе описаний тех изменений, которые должны быть введены в механизм уже существующей программы на языке ИПЛ-V [63]. Организация этой программы представляла упрощенный вариант системы GPS и оказалась весьма удачной, однако более существенные усилия в этом направлении не последовали. Вы слышите меня, соискатели звания «доктор философии»?

СЕМАНТИЧЕСКАЯ ОБРАБОТКА ИНФОРМАЦИИ И ЕСТЕСТВЕННЫЕ ЯЗЫКИ

Это направление имеет очень большое значение для всех исследований в области ИИ. Важность его определяется не столько предполагаемыми практическими преимуществами возможности общаться с «понимающей» программой на естественном языке, но скорее постановкой и изучением некоторых весьма общих проблем. В последние годы было выполнено несколько очень ценных исследований. В книге [42] излагается большая часть из них. Подробное обсуждение содержания и особенностей направления, невозможное здесь, можно найти в предисловии Минского к этой книге. Хороший обзор имеется и в работе Колса [13]. Тем не менее стоит сделать здесь несколько дополнительных замечаний.

Все эти исследования тем или иным способом пытаются разрешить проблему, связанную со значением обычных выражений естественного языка и степенью понимания вычислительной ма-

шиной этих значений, оцениваемой на основе соответствующих результатов лингвистического анализа, решения проблем и поиска запрашиваемой информации. Значение в этом случае не рассматривается как нечто «вводимое» в программу (например, обычные словарные определения помогут здесь очень мало); оно выясняется в результате взаимодействия синтаксических анализаторов, моделей, воспроизводящих соответствие реальным объектам и соотношениям, информационных структур, устанавливающих связь символов внутренней модели, логических процедур и связанных с ними процессов, реализуемых при решении задач логического вывода. (Этот перечень отнюдь не исчерпывает всех участников взаимодействия.)

Синтаксический анализ, которому уделялось большое внимание в связи с использованием вычислительных машин для лингвистического анализа (что привело к получению изящных результатов), оказывается недостаточен при решении проблем выяснения значения выражения и понимания этих значений [1]). Рассмотрим следующий пример.

Программа Боброу «STUDENT» предназначена для решения проблем и воспринимает входную информацию, записанную на естественном языке (английские фразы) [8]. Предметом исследования служат алгебраические задачи, взятые из вузовских учебников; эти задачи предъявляются программе в словесном описании. Фразы упрощаются и подвергаются грамматическому разбору, идиомы расшифровываются. Соответствие реальному миру устанавливается с помощью специальной таблицы «общих соотношений». Так как обычно программе «STUDENT» сообщается весьма незначительное количество общей информации, то ее «понимание» алгебраических задач, поставленных содержательно, является в основном «синтаксическим». Для получения ответа составляется и решается соответствующая система алгебраических уравнений.

Программу «STUDENT» можно использовать для решения следующей задачи: «Доска распилена на две части. Длина одной части равна двум третям длины всей доски и в то же время короче другой части на четыре фута. Чему была равна длина всей доски до распиливания?» Программа продемонстрирует, что в определенном смысле она «понимает» эту задачу, определив правильное решение, согласно которому длина доски равна минус двенадцати футам. В ориентированных на программу «STUDENT» психологических экспериментах, поставленных Пейджем и Саймоном [51], некоторые испытуемые решали

[1]) Я полагаю, что чрезмерное внимание, уделяемое синтаксическому анализу в ущерб исследованиям семантических процессов, повредило развитию машинного перевода.

задачи так же, как и программа (концентрируя внимание на синтаксическом анализе, приводящем к получению «правильной» системы уравнений), другие, однако, сразу осознав нереальность задачи, не стали составлять уравнения. Эти испытуемые относятся к разряду людей, привыкших пользоваться моделями конкретных ситуаций; получив задание, они попытались найти соответствующую модель и немедленно обнаружили противоречие [1]). Они продемонстрировали уровень «понимания задачи», который программе «STUDENT» недоступен.

Интерпретация естественного языка в терминах внутренних (содержащихся в памяти) моделей — центральная проблема большинства исследований в области семантической обработки информации. В разработанной Рейфлом семантической информационно-поисковой системе [2]) [55] для упорядочивания генеральной совокупности данных используется модель связи в виде сети (с ограниченным набором связей). Эта модель расширяется, усваивая новую информацию об объектах и существующих между ними связях, которая выделяется в результате несложного анализа повествовательных предложений, вводимых в систему исследователем. В других программах в роли модели выступают двумерные изображения. Колс [13], развивая работы Кирша и его сотрудников [33] на тему: «Машина, использующая язык изображений» [3]), составил программу, которая использует изображение (задаваемое исследователем на электронно-лучевой трубке с помощью светового пера) для снятия синтаксической неопределенности описывающих это изображение английских фраз, а также для ответа на вопросы относительно изображения.

Рассмотрим несколько подробнее тему моделей и организации имеющихся данных и остановимся на следующем способе трансформации обычного словаря в «семантический граф». Представим объекты (входы словаря) символами, соответствующими вершинам, а общие и частные соотношения между объектами — ассоциативными связями между вершинами. Вершина, соответствующая каждому объекту, является вершиной подграфа, который содержит различные «значения» объекта, выраженные через другие объекты и связи семантического графа. Куильян [54] сформировал такой граф для небольшого

[1]) Моя интуитивная реакция на Станфордский эксперимент заключается в том, что его результаты не зависят ни от качества самого эксперимента, ни от испытуемого, но определяются тем, какой тип мышления присущ последнему, — «образный» или «формальный». Совершенно очевидно, что программа «STUDENT» относится к группе «формалистов».

[2]) Semantic Information Retrieval (обычно употребляется сокращенное название программы — «SIR»). — *Прим. перев.*

[3]) Picture Language Machine (обычно употребляется сокращенное название программы — «PLM»). — *Прим. перев.*

(однако определенно нетривиального) набора словарных дефиниций, а также создал вычислительное устройство, предназначенное для ассоциативного и семантического поиска ссылок. Я полагаю, что программа Куильянома послужит хорошей основой для дальнейших исследований моделей ассоциативной памяти человека. Она вполне заслуживает тщательного рассмотрения.

МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ПОЗНАНИЯ

Недостаток места лишает нас возможности дать обзор работ в этой интересной области, находящейся на стыке теории вычислительных систем и психологии; при выдвижении и обосновании теорий, объясняющих характер информационных процессов, используемых человеком при обучении и решении проблем, как правило, но не всегда, прибегают к приемам эвристического программирования и методологии моделирования на вычислительных машинах. К счастью, недавно появились два подробных обзора [1, 31], к которым мы и рекомендуем обратиться заинтересованным читателям. Тем не менее я не могу преодолеть искушение поместить собственный вариант путеводителя по литературе.

Решение проблем: Ньюэлл [43, 46, 47].

Анализ поведения человека при решении криптарифметических задач; основные методологические преимущества анализа процесса решения проблем человеком, предусматривающего использование протоколов, фиксирующих рассказ испытуемого о процессе мышления; изучение движений глаза человека, зарегистрированных в процессе решения проблемы.

Вербальное обучение и память: Саймон и Фейгенбаум [66]; Грег и Саймон [28]; Фейгенбаум [20]; Хинцман [30].

Результаты дальнейших исследований «Элементарной программы восприятия и запоминания» (EPAM)[1]; новая интерпретация в рамках теории многоуровневой памяти человека; обобщение Хинтцмана, расширяющее набор «объяснимых» феноменов.

Формирование понятий и вывод закономерностей: Хант, Марин и Стоун описывают большое количество экспериментов с Системой формирования понятий (CLS)[2] [32].

---

[1]) Программа «EPAM» — Elementary Perceiver and Memoriser» — описана в статье Э. Фейгенбаума «Моделирование вербального обучения», помещенной в сб. [21]. — *Прим. перев.*

[2]) CLS — сокращенное название системы программ, предназначенной для формирования понятий при различных ограничениях на память (Concept Learning System). — *Прим. перев.*

Саймон и Котовски [67]; Саймон и Самнер [68]. Простая и остроумно организованная программа, предназначенная для решения задач на продолжение некоторой последовательности по предъявленной части, взятых из стандартных тестов оценки умственных способностей; с помощью этой программы можно также выявлять гармонические закономерности мелодий и создавать новые.

Аффекты, эмоции, представления: Теслер, Иниа и Колби [71]; Эйбелсон и Кэрролл [2]. Модели формирования системы взглядов у человека и их использование при изучении невропатического и нормального поведений.

Саймон [64]. Эмоциональные и мотивационные аспекты процесса познания.

Вынесение суждения: Клайнманц [34, 35].

Модель процесса клинического мышления, реализуемого при воспроизведении структуры личности на основе анализа ответов испытуемых (Миннесотский проект комплексного исследования структуры личности); приводятся также и результаты изучения иных задач из числа тех, с которыми сталкивается практикующий врач.

ОЦЕНКА СОСТОЯНИЯ НАШЕЙ ОБЛАСТИ ИССЛЕДОВАНИЯ В 1968 ГОДУ НА ОСНОВЕ АНАЛИЗА ОДНОЙ ИЗ ПРОГРАММ

Для того чтобы охарактеризовать положение дел в некоторой области исследований, часто оказывается полезным подробно рассмотреть одну из работ, выполняемых в ее рамках. Если вы хотите выяснить авторское кредо, приемы и технологию, использованные при создании, скажем, гравюры, то только крупный план, тщательный осмотр позволят получить достаточно определенное представление.

В качестве объекта подобного тщательного осмотра я выбрал ту исследовательскую работу, в которой сам участвую непосредственно. Выбор такого пути вместо анализа чужой работы определяется не только его органичностью, но и тем обстоятельством, что анализ состояния исследования предполагает знакомство с его деталями. Еще важнее то, что тема нашего исследования относится к тому направлению работ в области ИИ, которое я считаю основополагающим: решение проблемы, предусматривающее использование метода эвристического поиска.

Нашей основной задачей являлось изучение процессов формирования гипотезы при решении сложной проблемы, имеющей

научный характер и связанной с анализом экспериментальных данных. В качестве объекта данного исследования была выбрана задача, связанная с анализом масс-спектров органических молекул, — выдвижение гипотезы, наилучшим образом объясняющей полученные для спектров масс данные. Эта сравнительно новая область органической химии представляет весьма значительный интерес для физико-химиков. В связи с этим данная проблема не принадлежит к числу «условных» задач, и программа, обеспечивающая возможность решения задач подобного типа, представляет конструктивное приложение исследований ИИ к важной научной проблеме.

Мы составили программу, предназначенную для формирования гипотез о структуре молекулы на основе данных об их масс-спектрах. Программа называется «Эвристический дендрал» (Heuristic DENDRAL). Она разработана в рамках проекта ИИ Станфордского университета небольшой группой исследователей, в которую входили профессор Дж. Лидерберг с кафедры генетики, д-р Б. Бьюкэнен, миссис Дж. Сатерланд и я, с помощью химиков и масс-спектрометристов станфордского химического факультета.

Программа состоит из 80 000 слов, записана на языке для обработки списков LISP и предназначена для воспроизведения на вычислительной машине типа PDP-6; программа ориентирована на работу в режиме диалога при наличии системы с разделением времени и именно в таком режиме она и реализовывалась [9, 22, 39].

Программа «Эвристический дендрал» позволяет решать задачи следующих двух классов:

1. Дается масс-спектр и химическая формула исследуемой органической молекулы; программа предлагает небольшой список представляющих молекулы графов в качестве гипотез, интерпретирующих исходные данные с точки зрения заложенных в программу моделей масс-спектрометрических процессов и устойчивости органических молекул. Список ранжирован по степени приемлемости гипотезы от наибольшей до наименьшей.
2. Если масс-спектр не дан и имеется только формула, то программа предлагает список всех химически осуществимых изомеров молекулы исходя из заложенной в нее модели химической устойчивости органических молекул.

Блок-схема программы представляет собой замкнутый контур, в который включены блоки, представляющие этапы рассмотрения исходных данных, формирования гипотезы, прогнозирования и проверки гипотезы, т. е. она весьма точно

соответствует контуру, характеризующему простейший «научный подход».

Сердцевину программы образует детерминированный генератор гипотез. Этот генератор работает на основе предложенного Лидербергом алгоритма «Дендрал», обеспечивающего воспроизведение всех топологических возможных изомеров для данной химической формулы. По сути дела этот генератор представляет собой тополога, не располагающего какими-либо химическими сведениями, за исключением валентности атомов; сам алгоритм, однако, гарантирует полноту пространства гипотез подобно тому, как это делает генератор допустимых ходов в шахматной программе. Так как в основе процесса генерации лежит комбинаторная процедура, для всех молекул, кроме самых простейших, предлагается огромное количество структур, возможных топологически, но маловероятных с точки зрения химических реальностей. Процесс генерации предполагает существование некоторого дерева «Потенциальных гипотез». Присутствие всех атомов, не объединенных в какую-либо структуру, соответствует вершине дерева, концевым же точкам соответствуют только полные структуры и отсутствие нелокализованных атомов. Каждая из промежуточных вершин дерева определяет некоторую часть структуры и группу атомов, которая осталась нелокализованной.

Дерево это в неявном виде представляет пространство задач для программы «Эвристический дендрал». Выбор траекторий решения в этом пространстве определяется рядом эвристических правил и химических моделей типа следующих:

1. Модель химической устойчивости органических молекул, основанная на выделении некоторых запрещенных и предпочтительных подграфов для искомого графа структуры. Эта модель называется *априорной*, так как результаты масс-спектрометрии на нее не влияют.
2. Для предварительного исключения, исходя из имеющихся данных целых классов структур в связи с невозможностью использовать их даже в качестве грубой аппроксимации, применяется весьма приближенная, но эффективная теория поведения молекул в масс-спектрометре. «Теория масс-спектрометрии порядка 0».
3. Набор эвристических правил распознавания, позволяющих проводить предварительную интерпретацию имеющихся данных исходя из наличия основных функциональных групп, отсутствия иных функциональных групп, весов радикалов, связанных с основными функциональными группами и т. п. Эта процедура названа «Системой предварительного принятия решений». Ее существование

позволяет генератору гипотез оперировать лишь с наиболее перспективными поддеревьями пространства гипотез.

Система предварительного принятия решений и генератор гипотез формируют список, содержащий те молекулярные структуры, которые могут рассматриваться в качестве потенциальных гипотез для объяснения результатов масс-спектрометрического анализа. Все эти структуры являются приемлемыми с точки зрения априорной теории, заложенной в модель, и обеспечивают разумное истолкование исходных данных на основе упоминавшейся выше теории масс-спектрометрии порядка 0. Как правило, список этот содержит несколько кандидатур (а не дюжины или сотни).

Затем эти «наиболее вероятные» гипотезы из списка сопоставляются с имеющимися экспериментальными данными. Для каждой структуры прогнозируется ее масс-спектр. Эта процедура реализуется с помощью подпрограммы «Прогнозист», являющейся машинной моделью более совершенной теории масс-спектрометрии. Программа «Прогнозист» не является эвристической. В ней реализуется сложная, но вполне очевидная процедура получения выводов, следующих из теории масс-спектрометрии, почерпнутых нами у специалистов-химиков и в соответствующей литературе. Далее прогноз масс-спектра, полученный для каждой «кандидатуры», сравнивается с исходными экспериментальными данными; реализация этой процедуры приводит к получению функции оценки. Процесс получения оценки отличается иерархичностью, эвристичностью и нелинейностью. Часть кандидатур исключается сразу из-за того, что их прогнозированные масс-спектры не соответствуют граничным эталонам. Остальным присваивается оценка, в соответствии с ней они ранжируются и выводятся на печать в последовательности убывающей приемлемости.

Результаты применения программы для анализа некольцевых структур органических молекул, с которыми мы работали вплоть до настоящего времени, при масс-спектрометрии некоторых классов таких молекул оказываются не хуже результатов работы научного персонала лаборатории или даже превосходят их. Это относится к аминокислотам, с которыми по ряду причин связана большая часть наших работ, а также и к большому количеству простых органических групп, оказавшихся, однако, много сложнее первых с точки зрения масс-спектрометрии.

Эвристическое программирование обеспечило лишь остов для организации процессов решения проблем в программе «Эвристический дендрал» и ее воспроизведения на вычислительной

машине. Эвристические приемы для оценки «химической» приемлемости структур, осуществления предварительного принятия решений, оценки результатов прогноза, а также принципы составления для масс-спектрометрии точной теоретической модели и модели порядка 0 — все это было почерпнуто у наших коллег-химиков при работе в режиме взаимодействия человек — машина (один из наших исследователей посвятил себя этой работе). Успешное развитие этой темы — выуживание из «голов» практикующих специалистов используемых ими эвристик решения проблем — значительно превзошло то, на что мы имели основания рассчитывать, и в настоящее время рассматриваются возможности развития этого процесса «механизации».

ПРОБЛЕМА СПОСОБА ПРЕДСТАВЛЕНИЯ ДЛЯ СИСТЕМ РЕШЕНИЯ ПРОБЛЕМ

Несколько ключевых проблем общего характера будут доминирующими в те годы, которые остались до завершения второго десятилетия исследований ИИ. Проблема способа представления в системах решения проблем — одна из них, причем, с моей точки зрения, важнейшая, хотя и не поддающаяся самому быстрому разрешению [1]).

В эвристических программах решения проблем предусматривается, что поиск решения в пространстве задачи направляется и контролируется эвристическими правилами. Представление, задающее пространство задачи, определяется отношением исследователя к данной проблеме, его точкой зрения, и оно же предопределяет вид решения. Выбрав для задачи удачный способ представления, можно существенно повысить эффективность процессов поиска решения. Выбор способа представления задачи является уделом разрабатывающего программу исследователя и есть акт творческий. Амарел считает, что процесс выбора и формирования соответствующих представлений при решении проблем представляет квинтэссенцию связанного с творческой деятельностью поведения человека [5]. Я с ним согласен.

В литературе можно найти обсуждение ряда случаев, иллюстрирующее влияние выбора способа представления задачи на качество систем решения проблем. Классическим примером

[1]) Термин "проблема представления в системах решения проблем" употребляется нами для того, чтобы отделить эту задачу от проблемы представления данных (и структур данных), которая гораздо чаще является предметом обсуждения. Я думаю, в конце концов мы убедимся в том, что эти две группы задач обладают весьма существенными взаимосвязями, однако в настоящее время лучше обойтись без терминологических двусмысленностей.

в этом отношении служит так называемый «крепкий орешек», предложенный Маккарти и исследованный Ньюэллом [45]. Исключим из шахматной доски два угловых поля, расположенные на одной диагонали, и посмотрим, можно ли такую «усеченную» доску полностью покрыть костяшками домино. Если воспользоваться стандартным способом представления игры с помощью перебора ходов по отдельным полям, то для того, чтобы убедиться в невозможности подобного покрытия, придется обратиться к чудовищной по объему и практически нереализуемой процедуре поиска. С другой стороны, выбор способа представления, основанного на условии одновременного покрытия одной костяшкой белых и черных полей и подсчете количества белых и черных полей, позволяет немедленно прийти к выводу об отсутствии решения в связи с исключением при усечении доски двух полей одного цвета.

Значительно более сложный случай был рассмотрен Амарелом на примере классической задачи о переправе через реку в одной лодке миссионера и людоедов при задании ряда ограничений [4]. Амарел предлагает целую последовательность представлений этой задачи от используемого обычно до очень простого, но элегантного матричного, просмотр которого дает немедленное решение. Амарелу принадлежит и еще один пример, относящийся к доказательству теорем в исчислении высказываний [4а]. Кстати, Гелернтер еще в 1958 году, описывая свое устройство для доказательства геометрических теорем («геометрическая машина»), использовал в качестве вспомогательного способа представления диаграмму, с тем чтобы обеспечить повышение эффективности поиска по дереву задач-подзадач [1]).

До самого последнего времени в литературе обсуждение проблемы выбора способа представления было ограничено несколькими подробно рассмотренными примерами. К счастью, в своей новой работе Амарел [6] дает общее изложение своих взглядов на вопросы решения проблем и представления задачи, предлагая точное определение и доскональный анализ этой сложной области исследований.

Почему же изменение способа представления задачи может вести к столь существенным вариациям эффективности системы решения проблем? Это объясняется многими причинами — вот некоторые из них. Каждому способу представления задачи соответствует ряд специфических методов, определяющих

[1]) На русском языке описание «геометрической машины» можно найти в статьях Г. Гелернтера «Реализация машины, доказывающей геометрические теоремы» и Г. Гелернтера, Дж. Ханзена и Д. Ловленда «Экспериментальное исследование машины для доказательства геометрических теорем», помещенных в сборнике [21]. — *Прим. перев.*

возможности оперирования элементами представления. Отказываясь от представления, для которого соответствующие теория и операционные методы не развиты, в пользу обладающего ими, мы можем употребить весь этот аппарат для решения предложенной задачи. Кроме того, специфические закономерности, присущие определенному способу представления, могут быть включены в предлагаемую формулировку задачи (часто она оказывается неполной), восполняя отсутствующие, но весьма существенные элементы ее описания. Подобная ситуация была рассмотрена Пейджем и Саймоном [51] при решении задач, связанных с водно-спиртовыми смесями (выбранный способ представления позволил получить необходимые уравнения сохранения). И наконец, каждый способ представления может принести с собой некоторые фактические сведения, дополняющие условия задачи или используемые при организации процесса поиска решения.

Амарел рассмотрел возможности автоматизации процесса изменения способа представления с помощью введения шаговой последовательной процедуры, предусматривающий эволюцию каждого представления к более эффективному варианту [4, 4а]. Эволюция эта направляется исходя из той информации о задаче (или классе задач), которая появляется при реализации процесса решения проблемы для текущего представления задачи.

Альтернативой такому последовательному шаговому процессу служит некоторая система (генератор), способная предлагать различные способы представления в виде кандидатур для просмотра при эвристическом поиске адекватного представления. На той первоначальной стадии понимания проблемы представления, которую мы проходим, создание подобного генератора — задача труднопреодолимая. Однако простейший вариант такой системы позволяет воспроизводить содержащиеся в памяти элементы представлений, уже встречавшиеся или оцененные как потенциально полезные. Этот принцип был использован в программе Персона [52] при выборе адекватного способа представления эталона для набора различных задач на продолжение последовательности.

Использование принципа аналогии между различными задачами — решающий, с моей точки зрения, момент при создании генератора способов представления. Процесс поиска, используемый для поиска, обнаружения и испытания кандидатур на роль адекватного способа представления, предусматривает применение аналогичных процедур принятия решений ко всем способам представления (и присущим им методам решения, базису, используемому для представления данных, и т. д.), введенным в память генератора. Поиску решения, в котором

используется метод рассуждений по аналогии, уделяется в исследованиях ИИ поразительно мало внимания, если учесть важность этой проблемы. Работа Эванса [19], посвященная программе, предназначенной для решения входящих в тесты оценки умственных способностей задач, содержащих геометрические аналогии, является единственной, на которую я могу сослаться.

Мы вынуждены ограничить наши соображения по проблеме представления весьма поверхностными замечаниями. Я считал, однако, необходимым привлечь к этой теме внимание из-за ее первостепенной важности. Я считаю, что длительный период вызревания для этой проблемы закончился; что наступило время решительного броска; что в следующие пять лет мы будем свидетелями существенных достижений в этой области и что всему этому будет придаваться такое же первостепенное значение, которое сейчас приписывается собственно эвристическому поиску.

ОСНОВНЫЕ ИССЛЕДОВАТЕЛЬСКИЕ ЦЕНТРЫ

Общепринято составлять обзоры исследований, ориентируясь на их темы, и весьма необычно обращаться при этом к людям, которые их проводят, и исследовательским центрам, в которых они это делают. Тем не менее в частных беседах мне неоднократно в разных формах задавали один и тот же вопрос: «Искусственный интеллект (в частности, эвристическое программирование) — где этим занимаются?» Нет оснований не попытаться ответить на эти вопросы перед лицом большой аудитории.

Сведения сообщаются по состоянию на середину 1968 года. Основной оценкой является количество выполненных в данном центре высококачественных исследований (оценка моя).

Основными исследовательскими центрами, действующими в Соединенных Штатах, являются проекты ИИ, выполняемые в Массачусетском технологическом институте, Университете Карнеги — Меллона (бывший Технологический институт Карнеги) и Станфордском университете. Все три центра финансируются в основном управлением перспективного планирования научно-исследовательских работ (Министерство обороны). В каждом из них значительное количество специалистов готовится к получению докторской степени, исследуя проблемы ИИ. В распоряжении групп, работающих в МТИ и Станфорде, находятся вычислительные машины типа PDP-6, обладающие обширной памятью на магнитных сердечниках; в проекте, осуществляемом Университетом Карнеги, используется вычислительная машина типа IBM 360/67, снабженная очень обширной дополнительной системой памяти на магнитных сердечниках.

В МТИ наиболее видное положение на факультете и в исследованиях занимают Минский, Пейперт и в несколько меньшей степени Вейценбаум [73], в Университете Карнеги — Ньюэлл и Саймон, в Станфорде — Маккарти, Сэмюэль, Колби и Фейгенбаум. Исследователи, работающие в Станфордском университете, поддерживают тесные контакты со своими «соседями» — группой из Станфордского научно-исследовательского института (Нильсон и Рейфл), так же как группа МТИ тесно связана с группой Боброу, работающей в компании «Болт Бирейнек энд Нью-ман инк».

Остановимся на некоторых статистических данных, касающихся исследовательской группы Станфордского университета и потому известных мне лучше. Всего в эту группу входят 75 человек (профессорско-преподавательский состав, студенты и сотрудники), непосредственно связанных с выполнением проекта ИИ; около 25 различных тем находятся в стадии разработки и выпускается серия рабочих отчетов, насчитывающая уже 67 названий [1]). Эти цифры приводятся для того, чтобы продемонстрировать масштабы усилий, предпринимаемых в одном из основных центров исследования ИИ.

Еще пять центров также заслуживают внимания: Западный университет Кейза (Бенерджи, Эрнст); университет штата Висконсин (Травис, Юр и Лондон; лаборатории Рэдио корпорейшн оф Америка в Принстоне (Амарел); Эвристическая лаборатория Национального института здравоохранения (Слэгл); университет штата Вашингтон (Хант).

В первом десятилетии исследований ИИ Европа не имела исследовательских центров, которые могли бы составить в этой области конкуренцию основным американским. Однако в последние несколько лет первоклассный центр был создан в Эдинбургском университете. Исследовательские центры, предназначенные для разработки проблем ИИ, создаются также в Швеции и других странах.

В Эдинбурге исследования ИИ развиваются на кафедре «Машинного интеллекта и восприятия» (например, насколько четко человек осознает (формулирует) свою проблему). Ведущую роль в исследованиях играют Миши, Грегори, Берстолл, Доран и Поплстон. Эти исследования щедро субсидируются правительством Великобритании. Диапазон исследований очень широк и включает как темы, посвященные информационным моделям процессов познания и восприятия и их приложениям [10, 11], так и работы, связанные с созданием языков программирования [12]. В последнем отчете, выпущенном кафедрой, приводятся библиографические данные о 59 оригинальных иссле-

[1]) Данный обзор также входит в эту серию. — *Прим. перев.*

дованиях, проведенных начиная с 1965 года! Кроме того, эта группа обеспечивает выпуск сборников статей, образующих серию «Машинный интеллект».

В Швеции исследования в области ИИ проводятся на кафедре вычислительной техники университета в Упсала (системы GPS, планирование, моделирование роботов, язык обработки списков LISP). Руководит этими исследованиями Сандевалль, который прежде работал в Станфорде. В работе принимают участие психологи, интересующиеся моделированием процессов познания. Исследования субсидируются Шведским советом исследований в области естественных наук.

Этот обзор был подготовлен в качестве доклада на конгрессе Международной федерации по обработке информации [1]) IFIP 68, состоявшемся в Эдинбурге в августе 1968 года. Доклад был подготовлен при поддержке следующих организаций: Управление перспективного планирования научно-исследовательских работ, Министерство обороны (контракт SD — 183), Станфордский проект «Искусственный интеллект» и Вычислительный центр Станфордского университета. Мне бы хотелось также поблагодарить А. Ньюэлла и Дж. Сатерланд за их замечания, сделанные на заключительных стадиях подготовки выступления и рукописи.

СПИСОК ЛИТЕРАТУРЫ

Приводимый список содержит лишь те работы, которые были использованы при составлении настоящего обзора. К нему не следует относиться как к библиографии по данной области исследований. Полный список литературы, вышедшей до 1962 г. включительно, приведен в [21]. Для периода 1963—1968 гг. подобная работа еще не проделана. Группа искусственного интеллекта Ассоциации вычислительных машин занимается в настоящее время подготовкой такого указателя.

1. Abelson R., Computer Simulation of Social Behavior, в сб. G. Lindsley (ed.), Handbook of Social Psychology, 2nd edition, New York, Addison-Wesley, 1968.
2. Abelson R., Carroll J., Computer Simulation of Individual Belief Systems, American Behavioral Scientist, Vol. 8, 1965.
3. Amarel S., On Representations and Modeling in Problem Solving and on Future Directions for Intelligent Systems, Научный отчет (лаборатории Рэдио корпорейшн оф Америка, Принстон, штат Нью-Джерси, июнь 1968).
4. Amarel S., On Representations of Problems of Reasoning about Actions, в сб. Michie D. (ed.), Machine Intelligence 3, Edinburgh University Press, 1968.

4a. Amarel S., An Approach to Heuristic Problem Solving and Theorem Proving in the Propositional Calculus, в сб. J. F. Hart and S. Takasu (eds.), Systems and Computer Science, University of Toronto Press, 1967.

[1]) IFIP — International Federation of Information Processing. — Международная федерация по обработке информации. — *Прим. перев.*

5. Amarel S., On the Mechanization of Creative Processes, IEEE Spectrum, April 1966.
6. Amarel S., On the Representation of Problems and Goal-Directed Procedures for Computers, Рабочий отчет (группа исследований в области теории вычислительных систем лаборатории Рэдио корпорейшн оф Америка, Принстон, штат Нью-Джерси). Будет напечатано в «Трудах первого ежегодного симпозиума Американского общества кибернетики».
7. Amarel S., Problem Solving Procedures for Efficient Syntactic Analysis. Научный отчет (лаборатория Рэдио корпорейшн оф Америка, Принстон, штат Нью-Джерси, май 1968).
8. Borow D. G., A Question Answering System for High School Algebra Word Problems, *Proceedings of the FJCC*, 25 (1964).
9. Buchanan B., Sutherland G., Heuristic Dendral: A Program for Generating Explanatory Hypotheses in Organic Chemistry. Рабочий отчет «ИИ» № 62 (кафедра вычислительной техники, Станфордский университет), в сб. D. Michie et. al. (eds.), Machine Intelligence 4.
10. Burstall R. M., A Heuristic Method for a Job Scheduling Problem, *Operations Research Quarterly*, 17 (1966).
11. Burstall R. M., Computer Design for Electricity Supply Networks by Heuristic Method, *Computer Journal*, 9 (1966), 3.
12. Burstall R. M., Popplestone R. J., POP-2 Reference Manual, в сб. E. Dale and D. Michie (eds.), Machine Intelligence 2, Edinburgh, Oliver and Boyd, 1968, и в POP-2 Papers, Edinburgh, Oliver and Boyd, 1968.
13. Coles S. L., Syntax Directed Interpretation of Natural Language. Диссертация на соискание степени доктора философии (кафедра вычислительной техники, Университет Карнеги — Меллона, 1967).
14. Darlington J. D., Machine Methods for Proving Logical Arguments Expressed in English, *Mechanical Translation*, 8 (1965), 3 and 4.
15. Doran J., New Developments of the Graph Traverser, в сб. E. Dale and D. Michie (eds.), Machine Intelligence 2, Oliver and Boyd. Edinburgh, 1968.
16. Doran J. F., Michie D., Experiments with the Graph Traverser Program, *Proceedings of The Royal Society*, A 294 (1966).
17. Ernst G., Newell A., GPS: A Case Study in Generality and Problem Solving, ACM Monograph Series, Academic Press, New York — London, 1969.
18. Ernst H. A., MH-I. A Computer-operated Mechanical Hand, Диссертация на соискание степени доктора философии (Массачусетский технологический институт, 1961 г.), работа доложена на Западной объединенной конференции по вычислительным машинам в мае 1962 г.
19. Evans T. G., A Heuristic Program to Solve Geometric Analogy Problems, *Proceedings of the SJCC*, 25 (1964).
20. Feigenbaum E. A., Information Processing and Memory, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 4, Biology and Problems of Health, University of California Press, 1967.
21. Feigenbaum E., Feldman J., (eds.), Computers and Thought, McGraw-Hill Book Company, 1963. (Русский перевод: сб. Вычислительные машины и мышление под редакцией Э. Фейгенбаума и Дж. Фельдмана, «Мир», М., 1967.)
22. Feigenbaum E. A., Lederberg J., Buchanan B., Heuristic Dendral, Proceedings of the Hawaii International Conference on System Sciences, University of Hawaii and IEEE, University of Hawaii Press, 1968.
23. Floyd R. W., Non-Deterministic Algorithms, *Journal of the Association for Computing Machinery*, 14 (Oct. 1967), 4.
24. Freeman P. Диссертация на соискание степени доктора философии (Университет Карнеги — Меллона, 1968 г.).
25. Golomb S. W., Baumert L. D., Backtrack Programming, *Journal of the Association for Computing Machinery*, 12 (Oct. 1965), 4.

26. Green C. C., Raphael B., Research on Intelligent Question Answering System, Научный отчет AFCRL-67-0370 (проект «Искусственный интеллект», Станфордский научно-исследовательский институт, Менло-Парк, штат Калифорния, май 1967 г.).
27. Greenblatt G., Eastlake D., Crocker S., The Greenblatt Chess Program, Proceeding of the Fall Joint Computer Conference, Anaheim, Calif., 1967.
28. Gregg L. W., Simon H. A., An Information-Processing Explanation of One-Trial and Incremental Learning, *Journal of Verbal Learning and Verbal Behavior*, **6** (Oct. 1967), 5.
29. Guzman A., Some Aspects of Pattern Recognition by Computer, MIT Masters thesis, отчет № MAC-TR-37 (проект MAC, МТИ, февраль 1967 г.)
30. Hintzman D. L., Exploration for a Discrimination Net Model for Paired Associate Learning, *Journal of Mathematical Psychology*, **5** (Feb. 1968), 1.
31. Hunt E., Computer Simulation: Artificial Intelligence Studies and Their Relevance to Psychology, Annual Review of Psychology, 1968.
32. Hunt E., Marin J., Stone F., Experiments in Induction, Academic Press, 1966 (Русский перевод: Хант Э., Марин Дж., Стоун Ф., Моделирование процесса формирования понятий на вычислительной машине, «Мир», М., 1970.)
33. Kirsch R. A., Computer Interpretation of English Text and Picture Patterns, *IEEE Transaction on Electronic Computers*, **EC-13**, (Aug. 1964).
34. Kleinmuntz B., Profile Analysis Revisited: A Heuristic Approach, *Journal of Counseling Psychology*, **10** (1963).
35. Kleinmuntz B., The Processing of Clinical Information by Man and Machine, в сб. B. Kleinmuntz (ed.), Formal Representation of Human Judgment, Wiley, 1968.
36. Kochen M., On the Representation of Limited Information by Means of Pictures, Trees, and English-like Sentences, Рабочий отчет (группа исследований в области теории вычислительных систем, лаборатории Рэдио корпорейшн оф Америка, Принстон, штат Нью-Джерси, 1965).
37. Kotok A., A Chess Playing Program for the IBM 7090, Неопубликованная диссертация на соискание степени бакалавра наук (МТИ, 1962).
38. Кронрод А., Машина становится умнее, Правда, 15 марта 1967, № 74 (17756).
39. Lederberg J., Feigenbaum E., Mechanization of Inductive Inference in Organic Chemistry, в сб. B. Kleinmuntz (ed.), Formal Representation of Human Judgment, New York, John Wiley, 1968.
40. Michie D., Memo Functions and Machine Learning, *Nature*, **218** (1968).
41. Minsky M. L., Artificial Intelligence, *Information* (Freeman, 1966). (Русский перевод: Минский М., Искусственный разум, в сб. Информация, «Мир», М., 1968.)
42. Minsky M. L., (ed.), Semantic Information Processing, MIT Press, 1968.
43. Newell A., Eye Movements and Problem Solving, Computer Science Research Review, Carnegie — Mellon University, Pittsburgh, Pennsylvania, Dec. 1967.
44. Newell A., Heuristic Programming: III Structured Problems, Рабочий отчет (Университет Карнеги — Меллона, кафедра вычислительной техники, ноябрь 1967); работа будет опубликована в сб. Progress in Operations Research.
45. Newell A., Limitations of the Current Stock of Ideas about Problem Solving, в сб. A. Kent and O. E. Taulbee (eds.), Electronic Information Handling, Spartan, 1965.
46. Newell A., On the Analysis of Human Problem Solving Protocols, Рабочий отчет (Университет Карнеги — Меллона, кафедра вычислительной техники, июль 1967).
47. Newell A., Studies in Problem Solving: Subject 3 on the Crypt-Arithmetic Task Donald + Gerald = Robert, Рабочий отчет (Университет Карнеги — Меллона, кафедра вычислительной техники, июль 1967).

48. Newell A., Ernst G., The Search for Generality, Proceedings of the IFIP65 Congress, New York, Spartan, 1965.
49. Newell A., Simon H. A., An Example of Human Chess Play in the Light of Chess Playing Programs, Рабочий отчет (Университет Карнеги — Меллона, кафедра вычислительной техники, август 1964); работа опубликована в сб. J. Schade and N. Wiener (eds.), Progress in Neurocybernetics, Elsevier, Amsterdam, 1965.
50. Nilsson N. J., A New Method for Searching Problem-Solving and Game Playing Trees, Рабочий отчет (группа «Искусственный интеллект», Стэнфордский научно-исследовательский институт, Менло-Парк, штат Калифорния, ноябрь 1967). Работа доложена на конгрессе IFIP, состоявшемся в Эдинбурге в 1968 г.
51. Paige J., Simon H. A., Cognitive Processes in Solving Algebra Word Problems, в сб. B. Kleinmuntz (ed.), Problem Solving: Research, Method and Theory, Proceedings of the First Carnegie Symposium on Cognition, Wiley, New York, 1966.
52. Persson S., Some Sequence Extrapolating Programs. A Study of Representation and Modeling in Inquiring Systems, Отчет CS50 (Стэнфордский университет, кафедра вычислительной техники, сентябрь 1966).
53. Pingle K. K., Singer J. A., Wichman W. M., Computer Control of a Mechanical Arm through Visual Input. Работа доложена на конгрессе IFIP, состоявшемся в Эдинбурге в 1968 году.
54. Quillian J. R., Hunt E. B., The FORTRAN Deductive System, Технический отчет 68-1-01 (кафедра психологии, Университет штата Вашингтон, Сиэтл, штат Вашингтон, январь 1968).
55. Raphael B., A Computer Program which «Understands», *Proceedings of the FJCC*, **25** (1964).
56. Raphael B. (ed.), ACM SIGART Bulletin (June 1967), 11.
57. Raphael B., Programming a Robot, Рабочий отчет (проект «Искусственный интеллект», Стэнфордский научно-исследовательский институт, Менло-Парк, штат Калифорния). Работа доложена на конгрессе IFIP, состоявшемся в 1968 г. в Эдинбурге.
58. Roberts L. G., Machine Perception of Three Dimensional Solids, в сб. Optical and Electro-optical Processing of Information, MIT Press, Cambridge, Mass., 1965.

58a. Robinson J. A., Heuristic and Complete Processes in the Mechanization of Theorem Proving, в сб. J. F. Hart and S. Takasu (eds.), Systems and Computer Science, University of Toronto Press, 1967.

59. Rosen C. A., Nilsson N. J., An Intelligent Automaton, Рабочий отчет (группа «Искусственный интеллект», Стэнфордский научно-исследовательский институт, Менло-Парк, штат Калифорния); работа доложена на Международной конференции института инженеров-электриков и электроников, состоявшейся в Нью-Йорке в 1967 году.

60. Samuel A., Studies in Machine Learning Using the Game of Checkers, 2, — Recent Progress, *IBM Journal* (Nov. 1967). Работа будет воспроизведена в Annual Review of Automatic Programming, M. Halpern (ed.), Pergamon Press.
61. Sandewall E., The GPS Expressed in LISP-A, Рабочий отчет (Университет в Упсала, кафедра вычислительной техники, октябрь 1967).
62. Selfridge O. G., Pattern Recognition and Modern Computers, Proceedings of the 1955 Western Joint Computer Conference, 1955.
63. Simon H. A., Experiments with a Heuristic Compiler, *Journal of the Association for Computing Machinery*, **10** (Oct. 1963), 4.
64. Simon H. A., Motivational and Emotional Controls of Cognition, Psychological Review, 1967.
65. Simon H. A., Scientific Discovery and the Psychology of Problem Solving в сб. R. G. Colodny (ed.), Mind and Cosmos, University of Pittsburgh Press, 1966.

66. Simon H. A., Feigenbaum E. A., An Information-Processing Theory of Some Effects of Similarity, Familiarization, and Meaningfulness in Verbal Learning, *Journal of Verbal Learning and Verbal Behavior*, **3** (Oct. 1964).
67. Simon H. A., Kotovsky K., Human Acquisition of Concepts for Sequential Patterns, *Psychological review*, **70** (1963).
68. Simon H. A., Sumner R. K., Pattern in Music, в сб. B. Kleinmuntz (ed.), Formal Representation of Human Judgment, Wiley 1968.
69. Slagle J., Experiments with a Deductive Question Answering Program, *Communications of the ACM*, **8** (1965), **12**. (Русский перевод. Слэгл Дж., Эксперименты с дедуктивной программой вопросов — ответов, Современное программирование, «Мир», М., 1970.)
70. Slagle J. R., Bursky P., Experiments with a Multi — Purpose, Theorem Proving Heuristic Program, *Journal of the Association for Computing Machinery*, **15** (Jan 1968), 1
71. Tessler L., Enea H., Colby K. M., A Directed Graph Representation for Computer Simulation of Belief Systems, *Mathematical Biosciences*, **2** (1968).
72. Waterman D., Machine Learning of Heuristics, Диссертация на соискание ученой степени доктора философии (проект «Искусственный интеллект», Станфордский университет).
73. Weizenbaum J., Contextual Understanding by Computers, в сб. Kolers P. A., Eden M., (eds.), Recognizing Patterns: Studies in Living and Automatic Systems, MIT Press 1968. (Русский перевод: Вейценбаум И., Понимание связного текста вычислительной машиной, в сб. под редакцией П. Колерса и М. Идена, «Распознавание образов. Исследование живых и автоматических распознающих систем», «Мир», М., 1970.)
74. Williams P. G., Some Studies in Game Playing with a Digital Computer, Диссертация на соискание ученой степени доктора философии (Университет Карнеги — Меллона, кафедра вычислительной техники, июль 1965).

Об эволюции систем искусственного интеллекта[1])

Л. Шиклоши

1. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Искусственным интеллектом, кратко ИИ, называется изучение устройств, способных выполнять задачи, разрешение которых, как считается, предполагает применение интеллекта[2]). Определение истинной природы интеллекта у животных и человека представляется делом затруднительным, а иногда и бесполезным[3]). Кроме того, порой ощущается недостаток единодушия в том, что на самом деле есть "машинная программа, обладающая свойствами искусственного интеллекта". К настоящему времени, однако, уже существует значительное количество программ, которые заслуживают титула "интеллектуальные".

Сталкиваясь с различными программами ИИ, исследователь пытается разобраться в них, устанавливая их место в некоторой структуре, а часто прибегая к какой-нибудь классификационной схеме. В исследованиях в области ИИ были выделены такие направления, как решение задач, доказательство теорем, распознавание образов, игры и получение ответов на вопросы[4]). Ньюэлл (1967) предложил другой принцип классификации программ ИИ: поскольку при анализе программа рассматривается как объединение небольшого числа образующих элементарных блоков, ее можно охарактеризовать, указав типы использованных блоков и структуру их связи. В настоящей статье мы вводим иной подход: давайте попытаемся рассмотреть эволюцию программ ИИ ретроспективно, начиная с первого существенного достижения в этой области — программы "Логик-теоретик" (Logic Theorist), созданной Нью-

---

[1]) Laurent Siklóssy, On The Evolution of Artificial Intelligence, *Information Sciences*, 2 (1970), 369—377.

[2]) Поскольку существуют иные, с нашей точки зрения не худшие определения искусственного интеллекта, приведенное здесь, очевидно, следует считать частной точкой зрения автора. — *Прим. перев.*

[3]) Наличие интеллекта у животных нельзя считать достоверным научным фактом, и эта проблема уже достаточно давно является предметом острой дискуссии в биологической и биокибернетической среде. — *Прим. перев.*

[4]) Работы, выполненные в рамках создания "программ, отвечающих на вопросы" (question answering), можно отнести к более общему направлению "информационного поиска". — *Прим. перев.*

эллом, Шоу и Саймоном (1956 и 1957). Мы сопоставим хронологию развития программ ИИ со стандартной возрастной хронологией, характеризующей способности человеческого существа к решению задач, аналогичных тем, для решения которых предназначены программы.

Мы не будем рассматривать все известные программы ИИ; в частности, не будут затрагиваться программы, которые ничем не выделяются, за исключением того обстоятельства, что в них моделируются познавательные процессы. Иначе говоря, мы остановимся на программах, обладающих определенным уровнем "умения". Предпочтение будет отдаваться программам, способным "разумно" выполнять такие задачи, с которыми человек начинает справляться, достигнув вполне определенного возраста; следовательно, такие проблемы, как доказательство теорем методом резолюции и специальные случаи распознавания образов, остаются вне нашего внимания.

2. ЭВОЛЮЦИЯ ТЕХНИЧЕСКОГО АРСЕНАЛА ИИ

Для подхода к пониманию ИИ с эволюционной точки зрения оснований вполне достаточно. Применение эволюционных моделей в других областях часто давало как более глубокое понимание проблем данной науки, так и стимул для ее дальнейшего развития. Хотя лишь немногие из биологов считают, что онтогенез рекапитулирует филогенез, тем не менее изучение даже мало проявляющихся соответствий в развитии индивидуума и вида, к которому он принадлежит, оказывается весьма плодотворным.

Технические средства, находящиеся в распоряжении исследователей ИИ, также существенно изменились с тех пор, как в 1956 году появились первые сообщения о машине "Логик-теоретик". Вычислительные машины стали больше, а их быстродействие увеличилось. Таблица 1 позволяет сравнить характеристики вычислительной машины JOHNNIAC, на которую

Таблица 1

Параметры вычислительных машин типов JOHNNIAC и CDC7600

Тип вычислительной машины	Размер памяти (число слов)	Длина слова (в битах)	Время обращения к главной памяти (в наносекундах)
JOHNNIAC	4K	40	28 000
CDC7600	524K	60	275 (благодаря использованию 10-кратного перемежения эта величина соответствует 27,5 наносекундам)

была ориентирована программа "Логик-теоретик", и вычислительной машины типа CDC7600, появившейся в 1969 году. Если умножить преимущество, которым располагает машина CDC7600 в отношении быстродействия, на ее преимущество в скорости обращения к быстрой памяти, становится очевидно, что она более чем в 200 000 раз [1]) превосходит машину JOHNNIAC по мощности, не говоря уже о том, что магнитный барабан на 12 К слов [2]), приданный JOHNNIAC, мало что значит по сравнению с дополнительной памятью на сердечниках и магнитных дисках машины CDC7600.

Параллельно с развитием собственно вычислительных устройств средства программного обеспечения также претерпели эволюцию и в свою очередь расширили возможности специалистов в области вычислительных систем. Появление языков высших уровней и введение режима разделения времени облегчили создание программ и дали возможность разработать системы, обеспечивающие работу человека с вычислительной машиной в режиме диалога.

Усовершенствование доступных исследователям ИИ средств вычислительной техники и программного обеспечения в сочетании с ростом числа этих исследователей могло бы привести нас к выводу, что с течением времени задачи, выполняемые программами ИИ, будут подобны тем, которые оказываются по силам лишь более способным либо более взрослым людям. Эволюция ИИ не подтвердила, однако, этого "обоснованного" прогноза. На самом деле произошло как раз обратное.

Сейчас мы рассмотрим задачи, возможность справляться с которыми человек приобретает на различных этапах своего развития, а затем сопоставим этой хронологии программы, предназначенные для решения подобных задач.

3. ЗАДАЧИ, КОТОРЫЕ РЕШАЕТ ЧЕЛОВЕК

В этом разделе мы представим хронологию возрастов, показывающую, когда человек начинает справляться с решением определенных задач. В следующем разделе этой хронологии будет сопоставлена хронология эволюции программ, решающих подобные задачи.

В первом и втором столбцах табл. 2 указаны стандартный возраст, в котором обучающийся в Соединенных Штатах студент решает определенную задачу, и вид этой задачи. Для некоторых стран возраст, соответствующий решению задач определенных типов, может несколько отличаться от приведенного здесь.

[1]) (524/4) (60/40) (28 000/27,5) $\approx$ 200 000.
[2]) К = 1024. — *Прим. перев.*

Таблица 2

Сравнительная эволюция способностей человека и систем искусственного интеллекта к решению задач

Человек		Программы			
		Первый опыт		Потомки по „побочной линии“	
возраст	задача	год	название	год	автор
22	Аксиоматическая логика (формализм Уайтхеда и Рассела)	1956	“Логик-теоретик”: Ньюэлл, Шоу и Саймон		
20	Шашки	1959	“Шашист”: Сэмюэль	1967	Сэмюэль
19	Исчисление высказываний	1959	“Универсальный решатель задач”: Ньюэлл, Шоу и Саймон	1967 1968 1969	Эрнст Куинлан Файкес; Поцл
18	Анализ	1961	Символический интегратор: Слэйджл	1967	Мозес
17	Программирование	1963	“Эвристический компилятор”: Саймон	1969	Уолдингер и Ли; Уолдингер
16	Геометрическое подобие	1963	“Подобие»: Эванс		
15	Алгебраические задачи в содержательной постановке	1964	Система для решения алгебраических задач в содержательной постановке: Боброу		
15	Планиметрия	1959	“Машина для доказательства геометрических теорем”: Гелернтер и его сотрудники		
5	Школа				
3	**Язык и речь**	1966 1968 1968	Анализ речи: Редди Синтез речи: Каллер Программа, которая учится говорить: Шиклошши		
2	Двигательная активность и способность к координированным действиям	1968 1969 1968	Станфордский проект “рука—глаз”: Фельдман и его сотрудники Робот Станфордского исследовательского института: Нильсон Проект “рука—глаз” МТИ: МАС		
1	Восприятие окружающей среды	1968	“Визуальная программа”: Газман		

К одному году ребенок обладает вполне развитым восприятием материальной среды, в которой он существует. К двум годам его двигательная активность и ее координация становятся довольно совершенными. Около трех лет ребенок овладевает языком и речью, а еще год спустя может связно пересказывать какие-либо события и выслушивать рассказ о них. Около пяти лет от роду ребенок приходит в школу.

Пропустим около десяти лет и вернемся к нашему молодому человеку на втором году обучения в старших классах средней школы — теперь ему около пятнадцати лет. Он учится решать планиметрические и содержательно поставленные алгебраические задачи. Тесты на определение показателя умственных способностей — его старые знакомые. На 17 лет приходится его последний год в средней школе, и вполне возможно, что в отдельных лучших школах его знакомят с началами программирования.

Анализ обычно изучают в колледже. На этом этапе отмечается прогресс будущего логика или математика. В девятнадцать лет он знакомится с исчислением высказываний, а став аспирантом, встретится, по всей вероятности, где-то около двадцати двух лет с аксиоматическим построением логики, развитым Уайтхедом и Расселом. К этому времени, но, как правило, не раньше двадцати лет, он может научиться прекрасно играть в какую-нибудь из "настольных" игр (например, в шашки).

4. ЗАДАЧИ, КОТОРЫЕ РЕШАЮТ ПРОГРАММЫ ИИ

Как мы отметили выше, первой программой ИИ была, очевидно, программа "Логик-теоретик". Она доказывала теоремы из логики, построенной Уайтхедом и Расселом на аксиоматической основе. Задача, поставленная перед этой программой, очень близка той последней задаче, на которой мы покинули нашего молодого человека. В третьем столбце табл. 2 отмечается год первого сообщения о программе, а четвертый столбец содержит название программы, которая предназначена для решения задачи, подобной той задаче, решаемой человеком, которая помещена во втором столбце в этой же строке.

В 1959 году Сэмюэль сообщил об очень хорошей программе, предназначенной для игры в шашки. В этом же году Ньюэлл, Шоу и Саймон представили программу "Универсальный решатель задач" (General Problem Solver — GPS); на самом деле, однако, главным достоинством программы GPS является возможность доказывать теоремы из исчисления высказываний с помощью процедуры, существенно более "близкой" человеку, чем использованная в программе "Логик-теоретик".

Слэйджл (1961) разработал программу для решения задач символического интегрирования. Характеристики программы соответствуют уровню первокурсника колледжа, прослушавшего семестровый курс дифференциального исчисления. Программа Слэйджла может брать интегралы, например следующий:

$$\int \frac{x^4}{(1-x^2)^{4/2}}\,dx.$$

Значение этого интеграла равно $\arcsin x + \frac{1}{3}\operatorname{tg}^3(\arcsin x) - \operatorname{tg}(\arcsin x + C)$.

В 1963 году Саймон сообщил о своей программе "Эвристический компилятор" (Heuristic Compiler), способной составлять

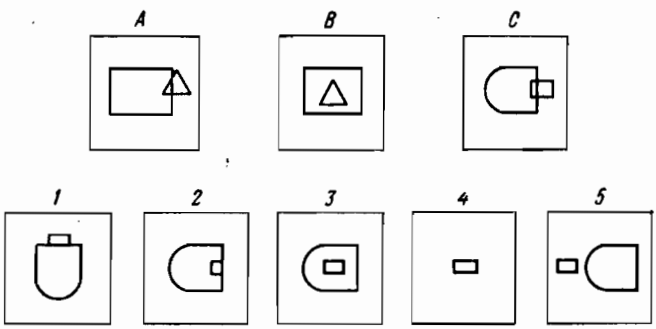


Рис. 1. Пример задачи на установление геометрического подобия. Фигура *A* находится в таком же отношении к фигуре *B*, как фигура *C* к фигурам 1, 2, 3, 4 или 5.

элементарные программы исходя из изложенного на английском языке описания. В этом же году появилось описание программы Эванса, решающей геометрические задачи на подобие. Пример задачи такого типа приведен на рис. 1 (правильный ответ — фигура 3). Программа соответствует уровню ученика последнего класса средней школы. Программа Боброу (1964) опускается на год-два ниже и занимается решением алгебраических задач, поставленных содержательно, например, следующего типа: "Мэри в два раза старше, чем была Энн, когда Мэри было столько же лет, сколько Энн сейчас. Если Мэри сейчас 24 года, сколько лет Энн?"
(Ответ, выданный программой: "Энн восемнадцать лет".)

Тому же пятнадцатилетнему уровню соответствует программа Гелернтера (Gelernter), предназначенная для решения задач из области планиметрии. Эта программа может решить символический эквивалент следующей задачи: "Пусть дан прямоугольник *ABCD* и точки *E*, *F*, *G* и *H* делят отрезки *AB*, *AC*,

CD и *BD* пополам. Требуется доказать, что *EFGH* — параллелограмм".

Эта программа, к сожалению, выпадает из нашей хронологии, так как она работала уже в 1959 году. Существование этого исключения демонстрирует несовершенство нашей эволюционной модели. Действительно, можно отметить, что принципы, заложенные в программу Гелернтера, крайне аксиоматичны и, следовательно, сама программа относится к тому же классу, что и "Логик-теоретик".

Мы покрыли тот длинный путь, который разделяет способности к решению задач трехлетнего ребенка и пятнадцатилетнего юноши. Упоминавшиеся программы в ряде случаев достигают уровня, вполне сравнимого с человеческими возможностями. Программы, к рассмотрению которых мы переходим, отличаются от предыдущих тем, что их возможности существенно слабее навыков даже очень маленького ребенка, с поведением которого сопоставляется их функционирование.

Очень интенсивно ведутся исследования, посвященные проблемам восприятия среды и действий ("манипуляции") в ней. Системы, разрабатываемые в Станфордском университете и Массачусетском технологическом институте в рамках проектов "рука — глаз", располагают глазом — модифицированной телевизионной камерой и рукой, управляет которыми вычислительная машина. Глаз воспринимает окружающую обстановку, а рука может вносить в нее изменения. Последними сообщениями об этих исследованиях, проведение которых продолжается, являются доклад Фельдмана и его сотрудников (1969) и отчеты МАС (1967 и 1968).

Робот, построенный в Станфордском исследовательском институте, снабжен глазом, который управляется вычислительной машиной и установлен на тележке. Робот перемещается в среде, которую он "осматривает" своим глазом. Последние сведения о работе Станфордского исследовательского института содержатся в докладе Нильсона (1969).

Что касается восприятия, то программа Газмана (1968) распознает элементы изображений, составленных из прямоугольных компонент.

5. ЕЩЕ НЕМНОГО НА ТУ ЖЕ ТЕМУ

Побочную линию в эволюции искусственного интеллекта образует развитие программ, отпочковавшихся от некоторых из рассмотренных выше программ. Ряд таких модификаций приведен в пятом и шестом столбцах табл. 2.

В 1967 году Сэмюэль сообщил об усовершенствованной программе для игры в шашки. "Универсальный решатель задач"

обзавелся целым рядом потомков. Эрнсту (1967) удалось продемонстрировать универсальность программы "GPS", применив ее для решения задач, относящихся к дюжине различных областей. В 1968 году Попл объединил метод "GPS", основанный на "анализе средств и целей", с представлением с помощью аппарата узкого исчисления предикатов; Файкес в свою очередь создал язык для представления задач на анализ средств и целей при наличии определенных ограничений и еще один язык, ориентированный на решение задач, представленных таким образом.

В 1967 году Мозес представил очень эффективную программу, предназначенную для символического интегрирования. Программы Уолдингера и Ли (1969) — дальние родственники "Эвристического компилятора", но все же связанные с программированием, — по описанию в языке исчисления предикатов составляют программу, не требующую отладки.

6. ПОИСК

Ребенок не может долго оставаться сосредоточенным и быстро разочаровывается, если не достигает поставленной цели. Став старше, человек может предпринимать многократные попытки разрешить проблему. Объем поиска, реализуемого программой вычислительной машины, можно сравнить с терпением человека: чем длиннее поиск, который "согласна" предпринимать программа, тем более "терпеливой" и "настойчивой" можно ее считать.

Одна из эволюционных моделей искусственного интеллекта основана на оценке поисковых процедур рассмотренных нами программ. Первые программы, такие, как "Логик-теоретик", GPS, символический интегратор Слэйджла, система для доказательства геометрических теорем, часто при получении решения прибегали к длительному поиску. Более поздние программы (например, программа для решения содержательно заданных алгебраических задач Боброу, программа установления геометрического подобия Эванса, "визуальная" программа Газмана) значительно менее "терпеливы" и получают решение, обходясь без чересчур длительного поиска. Некоторые программы "побочной линии" (например, программа символического интегрирования Мозеса) также отличаются небольшим объемом поиска.

7. ПРОБЕЛЫ И ПЕРСПЕКТИВЫ

Способность прогнозировать развитие — обязательное преимущество эволюционной модели. В этом отношении наша модель эволюции искусственного интеллекта разочаровывает: существует очень мало задач, с которыми может справиться

ребенок, не достигший года. Наша модель тем не менее представляет целый ряд сфер, в которых человеческий интеллект торжествует, а оппоненты со стороны искусственного интеллекта отсутствуют.

Разрыв между тремя и пятнадцатью годами медленно заполняется. Ряд программ в небольшой степени обладает свойством вербального понимания. В настоящее время продолжается работа над программой, способной понимать рассказы детей (1969). Проблема обучения, столь существенная для процесса развития человека, оказалась едва затронутой.

Другой существенный пробел относится к задачам, разрешаемым квалифицированными людьми старше двадцати двух лет. Одна программа такого рода уже существует — DENDRAL (1969). Совместные усилия химиков и специалистов в области вычислительной техники позволили создать эту программу; она воспроизводит возможные молекулярные структуры органических соединений по их масс-спектрам и эмпирическим формулам.

В будущем следует ожидать появления и других программ, связанных с научными исследованиями. Разработка "высококвалифицированных" обучающих систем (действующих на базе вычислительных машин) имеет отношение и к пробелу во "взрослой" деятельности; в настоящее время соответствующим проблемам уделяется много внимания.

8. ЗАКЛЮЧЕНИЕ

Мы сопоставили эволюцию задач, которые могут быть решены людьми и программами искусственного интеллекта. Эти эволюции развивались в противоположных направлениях как хронологически, так и в отношении "настойчивости" поискового поведения при решении задач. Предметом исследований в области искусственного интеллекта в будущем являются те "пробелы", которые образованы задачами, выполняемыми людьми и пока не выполняемыми удовлетворительно системами искусственного интеллекта.

СПИСОК ЛИТЕРАТУРЫ

1. Bobrow D. G., Natural Language Input for a Computer Problem-Solving System, Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, Mass., 1964 (перепечатано в сб. [15]).
2. Buchanan B., Sutherland G., Feigenbaum E. A., HEURISTIC DENDRAL: a program for generating explanatory hypotheses in organic chemistry, в сб. Machine Intelligence 4 (D. Michie and B. Meltzer, eds.), American Elsevier, New York, 1969.
3. Charniak E., Understanding of Children's Stories, Project MAC, Massachusetts Institute of Technology, Cambridge, Mass., реферат: SIGART Newsletter, № 17, Aug. 1969, p. 15.

4. Culler G. J., An attack on the problems of speech analysis and synthesis with the power of an on-line system, Proc. Int. Jt. Conf. Art. Intel. (IJCAI), Washington, D. C., May, 1969.
5. Ernst G. W., Newell A., Generality and GPS, Report of the Center for the Study of Information Processing, Carnegie Institute of Technology, Pittsburgh, Pa., Jan. 1967.
6. Evans T. G., A Heuristic Program to Solve Geometric-Analogy Problems, Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, Mass., 1963 (перепечатано в сб. [15]).
7. Feigenbaum E. A., Feldman J. (eds.), Computers and Thought, McGraw-Hill, New York, 1963. (Русский перевод: Вычислительные машины и мышление, под редакцией Э. Фейгенбаума и Дж. Фельдмана, «Мир», М., 1967.)
8. Feldman J. A., Feldman G. M., Falk G., Grape G., Pearlman J., Sobel I., Tenenbaum J. M., The Stanford hand-eye project, Proc. Int. Jt. Conf. Art. Intel. (IJCAI), Washington, D. C., May, 1969.
9. Fikes R. E., A Heuristic Program for Solving Problems Stated as Nondeterministic Procedures, Doctoral Dissertation, Carnegie-Mellon University, Pittsburgh, Pa., 1968.
10. Gelernter H., Realization of a geometry-theorem proving machine, Proc. Intern. Conf. Information Processing (ICIP), Paris, France, June, 1959 (перепечатано в сб. [7]). (Русский перевод: Гелернтер Г., Реализация машины, доказывающей геометрические теоремы, в сб. Вычислительные машины и мышление, под редакцией Э. Фейгенбаума и Дж. Фельдмана, «Мир», М., 1967, стр. 145—165.)
11. Gelernter H., Hansen J. R., Loveland D. W., Empirical explorations of the geometry-theorem proving machine, Proc. Western Joint Computer Conf. (WJCC), 1960 (перепечатано в сб. [7]). (Русский перевод: Гелернтер Г., Ханзен Дж., Ловленд Д., Экспериментальное исследование машины для доказательства геометрических теорем, в сб. Вычислительные машины и мышление, под редакцией Э. Фейгенбаума и Дж. Фельдмана, «Мир», М., 1967, стр. 165—175.)
12. Guzman A., Computer Recognition of Three-Dimensional Objects in a Visual Scene. Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, Mass., 1968.
13. MAC. Project MAC Progress Report IV, Massachusetts Institute of Technology, Cambridge, Mass., 1967.
14. MAC. Project MAC Progress Report V, Massachusetts Institute of Technology, Cambridge, Mass., 1968.
15. Minsky M. (ed.), Semantic Information Processing, MIT Press, Cambridge, Mass., 1968.
16. Moses J., Symbolic Integration, Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, Mass., 1967.
17. Newell A., Heuristic Programming: Ill-Structured Problems, Carnegie—Mellon University, Doctoral Dissertation, Nov. 1967.
18. Newell A., Shaw J. C., Simon G. A., Empirical explorations with the logic theory machine, Proc. Western Joint Computer Conf. (WJCC), Los Angeles, Calif., Feb., 1957 (перепечатано в сб. [7]). (Русский перевод: Ньюэлл А., Шоу Дж., Саймон Г., Эмпирические исследования машины «Логик-теоретик»; пример изучения эвристики, в сб. Вычислительные машины и мышление, под редакцией Э. Фейгенбаума и Дж. Фельдмана, «Мир», М., 1967, стр. 113—144.)
19. Newell A., Shaw J. C., Simon H. A., Report on a general problem-solving program, Proc. Intern. Conf. Information Processing (ICIP), Paris, France, June 1959.
20. Newell A., Simon H. A., The logic theory machine, *IRE Trans.*, **IT2, №** 3, Sept. 1956.

21. Nilsson N. J., A mobile automaton: an application of artificial intelligence techniques, Proc. Int. Jt. Conf., Art. Intel., (IJCAI), Washington, D. C., May 1969.
22. Pople H., A Goal-Oriented Language for the Computer, Doctoral Dissertation, Carnegie — Mellon University, Pittsburgh, Pa.. 1969.
23. Quinlan J. R., Hunt E. B., A formal deductive problem-solving system, *J. Assoc. Comput, Mach.*, **15**, № 4 (Oct. 1968), 625—646.
24. Reddy D. R., An Approach to Computer Speech Recognition by Direct Analysis of the Speech Wave, Doctoral Dissertation, Stanford University, Stanford, Calif., 1966.
25. Samuel A. L., Some studies in machine learning using the game of checkers, *IBM J. Res. Devel.*, **3**, № 3 (1959) (перепечатано в сб. [7]). (Русский перевод: Сэмюэль А., Некоторые исследования возможности обучения машин на примере игры в шашки, в сб. Вычислительные машины и мышление, под редакцией Э. Фейгенбаума и Дж. Фельдмана, «Мир», М., 1967, стр. 71—111.)
26. Samuel A. L., Some studies in machine learning using the game of checkers. II — Recent Progress, *IBM J. Res. Devel.*, **11**, № 6 (Nov. 1967), 601—617.
27. Siklóssy L., Natural Language Learning by Computer, Doctoral Dissertation, Carnegie — Mellon University, Pittsburgh, Pa., 1968.
28. Simon H. A., Experiments with a heuristic compiler, *J Assoc. Comput. Mach.*, **10**, № 4 (Oct. 1963), 482—506.
29. Slagle J., A Computer Program for Solving Problems in Freshman Calculus (SAINT), Doctoral Dissertation, Massachusetts Institute of Technology, Cambridge, Mass., 1961 (в сокращенном виде приведена в сб. [7]). (Русский перевод: Слэйджл Д., Эвристическая программа, решающая задачи символического интегрирования в объеме первого курса университета, в сб. Вычислительные машины и мышление, под редакцией Э. Фейгенбаума и Дж. Фельдмана, «Мир», М., 1967, стр. 204—219.)
30. Waldinger R. J., Constructing Programs Automatically Using Theorem Proving, Doctoral Dissertation, Carnegie — Mellon University, Pittsburgh, Pa., 1969.
31. Waldinger R. J., Lee R. C. T., PROW: a step toward automatic program writing, Proc. Int. Jt. Conf. Art. Intel. (IJCAI), Washington, D. C., May 1969.

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., д. 2, издательство «Мир».

КИБЕРНЕТИЧЕСКИЙ СБОРНИК

Новая серия

Выпуск 10

Редактор Л. Н. Бабынина
Художник Н. К. Сапожников
Художественный редактор В. И. Шаповалов
Технический редактор З. И. Резник

Сдано в набор 31/I 1973 г.
Подписано к печати 31/VII 1973 г.
Бумага тип. № 1 $60\times90^{1}/_{16}$=6,75 бум. л. 13,5 печ. л.
Уч.-изд. л. 13,34. Изд. № 1/7152
Цена 1 р 54 к. Зак. 521
Темплан изд-ва «Мир» 1973 г., пор. № 10

ИЗДАТЕЛЬСТВО «МИР»

Москва, 1-й Рижский пер., 2

Ордена Трудового Красного Знамени
Ленинградская типография № 2
имени Евгении Соколовой Союзполиграфпрома
при Государственном комитете Совета Министров
СССР по делам издательств, полиграфии
и книжной торговли
г. Ленинград, Л-52, Измайловский проспект, 29