

Проблемы кибернетики

ПРОБЛЕМЫ КИБЕРНЕТИКИ

ПОД РЕДАКЦИЕЙ
А. А. ЛЯПУНОВА

ВЫПУСК 4



ГОСУДАРСТВЕННОЕ ИЗДАТЕЛЬСТВО
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1960

В СОСТАВЛЕНИИ И РЕДАКТИРОВАНИИ
ПРИНИМАЛИ УЧАСТИЕ

Г. В. ВАКУЛОВСКАЯ, Т. Л. ГАВРИЛОВА, Б. Ю. ПИЛЬЧАК,
Я. И. СТАРОБОГАТОВ, В. С. ШТАРКМАН, С. В. ЯБЛОНСКИЙ

Сборник статей под редакцией *Алексея Андреевича Лапунова*

Редакторы: *Г. В. Вакуловская, Я. И. Старобогатов и Б. И. Фиников.*

Технический редактор *С. Н. Ахламов.*

Корректор *И. С. Цветкова.*

Сдано в набор 7/XII 1959 г. Подписано к печати 21/III 1960 г. Бумага $70 \times 108^{1/16}$
Физ. печ. л. 16,25. Условн. печ. л. 22,25. Уч.-изд. л. 22,23. Тираж 10000 экз. Т-01058.
Цена книги 12 р. 60 к. Заказ № 1410

Государственное издательство физико-математической литературы.
Москва, В-71, Ленинский проспект, 15.

Московская типография № 5 Мосгорсовнархоза.
Москва, Трехпрудный пер., 9.

СОДЕРЖАНИЕ

От редакции	4
I. ОБЩИЕ ВОПРОСЫ	
О. Б. Лупанов. Об асимптотических оценках числа графов и сетей с n ребрами	5
Л. А. Скорняков. Об одном классе автоматов (нервные системы)	23
II. ТЕОРИЯ УПРАВЛЯЮЩИХ СИСТЕМ	
Г. В. Савинов. Электрическое моделирование гомеостатических систем	37
В. И. Мудров. К вопросу об определении вероятности отказа в однолинейных системах массового обслуживания смешанного типа	45
III. ТЕОРИЯ ИНФОРМАЦИИ	
А. А. Харкевич. О ценности информации	53
IV. ПРОГРАММИРОВАНИЕ	
Н. Г. Арсентьева. О некоторых преобразованиях схем программ	59
В. А. Федосеев. Методы автоматизации программирования на вычислительных машинах	69
В. С. Корольюк. О понятии адресного алгоритма	95
V. ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ	
М. А. Карцев. Логические методы ускорения умножения в цифровых вычислительных машинах	111
VI. ПРОЦЕССЫ УПРАВЛЕНИЯ В ЖИВЫХ ОРГАНИЗМАХ	
И. И. Шмальгаузен. Основы эволюционного процесса в свете кибернетики	121
А. А. Малиновский. Типы управляющих биологических систем и их приспособительное значение	151
А. В. Напалков. Некоторые принципы работы головного мозга	183
VII. ВОПРОСЫ МАТЕМАТИЧЕСКОЙ ЛИНГВИСТИКИ	
Р. М. Фрумкина. Некоторые данные о распределении форм многоосновных глаголов в связи с проблемой составления словаря основ для машинного перевода	197
О. С. Кулагина. О машинном переводе с французского языка на русский. II. Алгоритм перевода с французского языка на русский . . .	207

ОТ РЕДАКЦИИ

Четвертый выпуск сборника «Проблемы кибернетики» по своему содержанию достаточно близко примыкает к предыдущим выпускам. Единственный раздел, который появляется в этом выпуске впервые, — это раздел теории информации.

Редакция сожалеет о том, что до сих пор на страницах сборника не развернулся отдел математической экономики, а также о том, что в сборниках не публикуется рецензий и библиографических материалов.

За последнее время в Советском Союзе сильно расширилось издание литературы в области кибернетики. Появился целый ряд книг как переводных, так и оригинальных, посвященных изложению кибернетики в целом или отдельных ее ветвей, ее приложениям, а также развитию областей науки, тесно переплетающихся с кибернетикой. Например:

Эшби У. Р., Введение в кибернетику, ИЛ, М., 1959.

Косса, Кибернетика, ИЛ, М., 1958.

Винер Н., Кибернетика, «Советское радио», М., 1958.

Томсон Д., Предвидимое будущее, ИЛ, М., 1958.

Блекуэлл Д. и Гиршик М. А., Теория игр и статистических решений, ИЛ, М., 1958.

Бартлетт М. С., Введение в теорию случайных процессов, ИЛ, М., 1958.

Андронов А. А., Витт А. А., Хайкин С. Э., Теория колебаний, Физматгиз, М., 1959.

Трахтенброт Б. А., Алгоритмы и машинное решение задач, Гостехиздат, М., 1957.

Машинный перевод, Сб. переводов под ред. Молошной Т. Н. и Пурто В. А., ИЛ, М., 1957.

Логические исследования, Сб. статей, АН СССР, М., 1959.

Ниль Дж., Шэлл У., Наследственность человека, ИЛ, М., 1958.

Современная математика для инженеров, ИЛ, М., 1959.

Теория информации, Сб. переводов, ИЛ, М., 1959.

Редакция считает, что было бы целесообразно организовать на страницах сборника «Проблемы кибернетики» обсуждение этих книг, а также и других изданий, относящихся к области кибернетики.

Редакция приносит глубокую благодарность Ф. Я. Ветухновскому, А. П. Ершову, В. М. Золотареву, В. К. Коробкову, В. И. Левенштейну, О. Б. Лупанову, Б. А. Севастьянову и М. Л. Цетлину, оказавшим большую помощь в подготовке к печати настоящего сборника.

І. ОБЩИЕ ВОПРОСЫ

ОБ АСИМПТОТИЧЕСКИХ ОЦЕНКАХ ЧИСЛА ГРАФОВ И СЕТЕЙ С n РЕБРАМИ*)

О. Б. ЛУПАНОВ

(МОСКВА)

При многих рассмотрениях, связанных с оценками сложности управляющих систем [1], приходится использовать асимптотические оценки числа графов и сетей, обладающих теми или иными свойствами. Кроме того, этот вопрос представляет самостоятельный интерес.

Такого рода оценками занимались многие авторы [2—9]. В работах [4—6, 10] для числа неизоморфных графов (сетей) с n ребрами были получены верхние оценки вида $(Cn)^n$, в работе [5] — нижняя оценка вида $\left(D \frac{n}{\ln^2 n}\right)^n$ (здесь C, D — некоторые константы, причем константы C у разных авторов разные). Ф. Я. Ветухновским [9] построено семейство неразложимых сетей [11], содержащее примерно $\left(\frac{1}{2e} \frac{n}{\ln^2 n}\right)^n$ сетей с n ребрами. В настоящей работе будет показано, что для числа $\tilde{G}(n)$ неизоморфных графов с n ребрами справедливо соотношение **)

$$\ln \tilde{G}(n) = n \ln n - 2n \ln \ln n + (\ln 2 - 1)n + \gamma'(n), \quad (1)$$

где

$$\frac{2n \ln \ln n}{\ln n} \leq \gamma'(n) \leq \frac{4n \ln \ln n}{\ln n}.$$

Работа состоит из двух параграфов. В первом — основном — параграфе вводятся необходимые понятия и формула (1) доказывается для связных графов, не содержащих параллельных ребер и петель (теорема 1), и формулируются более общие результаты (теорема 2 и 3). Их доказательство приводится во втором параграфе.

После того как статья была сдана в печать, автору стало известно, что в [17] приводятся даже более сильные результаты, принадлежащие Пойя. Однако, поскольку, как отмечается в той же работе, эти результаты Пойя не опубликованы, автор считает возможным, не претендуя на приоритет, опубликование предлагаемой статьи.

§ 1

1. Напомним некоторые понятия из теории графов***). Конечная совокупность элементов a_1, \dots, a_m и элементов $\alpha_1, \dots, \alpha_k$ называется (конечным) *графом*, если каждому a_i поставлены в соответствие два

*) Краткое изложение результатов статьи опубликовано в [18].

**) Запись $\alpha(n) \leq \beta(n)$ означает, что $\lim_{n \rightarrow \infty} \frac{\alpha(n)}{\beta(n)} \leq 1$.

***). Ср. [11—14].

элемента α_j и α_h (быть может, совпадающие); элементы α_i называются *ребрами*, а элементы α_j *вершинами* графа. При этом допускается также граф с пустым множеством ребер и пустым множеством вершин. Вершины α_j и α_h , поставленные в соответствие ребру α_i , будем называть *концами* этого ребра. Если вершина α соответствует ребру a , то будем говорить, что α и a *инцидентны*.

Граф, в котором выделено некоторое подмножество вершин, называется *сетью*; выделенные вершины называются *полюсами* сети (в частности, граф является сетью). Две сети называются *изоморфными*, если можно установить взаимно однозначное соответствие между множествами их вершин и взаимно однозначное соответствие между множествами их ребер, такие, что

- а) соответствующие ребра инцидентны соответствующим вершинам,
- б) полюсы одной сети соответствуют полюсам другой сети и обратно.

Подмножество ребер и вершин графа G называется *подграфом* графа G , если оно является графом. Подграф графа G называется *цепью*, соединяющей вершины α и β , если он состоит из попарно различных вершин $\alpha_{j_1}, \dots, \alpha_{j_n}$ ($n \geq 1$) и ребер $\alpha_{i_1}, \dots, \alpha_{i_{n-1}}$, таких, что α_{i_s} инцидентно с вершинами α_{j_s} и $\alpha_{j_{s+1}}$ ($1 \leq s \leq n-1$), причем $\alpha_{j_1} = \alpha$, $\alpha_{j_n} = \beta$. Граф называется *связным*, если для любых двух его вершин существует цепь, их соединяющая. Ребро с совпадающими концами называется *петлей*. Вершина, не являющаяся концом никакого ребра, называется *изолированной*.

Каждому графу сопоставим некоторую геометрическую фигуру (геометрическая реализация графа). Выберем k различных точек в пространстве (на плоскости) и обозначим их через $\alpha_1, \dots, \alpha_k$. Затем точку α_j соединим с точкой α_h дугой столько раз, сколько имеется в графе ребер с концами α_j и α_h (эти дуги обозначим символами соответствующих ребер), и таким образом, чтобы дуги не проходили через другие вершины α_s и чтобы все дуги (для всех пар $\alpha_j \alpha_h$) попарно не пересекались во внутренних точках.

2. Пусть $G(n)$ — число неизоморфных связных графов с n ребрами без параллельных ребер*) и без петель.

В этом параграфе будет доказано следующее утверждение

Теорема 1. 1) $G(n) = \left(\frac{2}{e} \frac{n}{\ln^2 n} \gamma(n) \right)^n$, где

$$\frac{2 \ln \ln n}{\ln n} \leq \gamma(n) - 1 \leq \frac{4 \ln \ln n}{\ln n}.$$

2) Доля связных графов с n ребрами без петель и без параллельных ребер, число k вершин которых удовлетворяет условию**))

$$\left| k - \frac{2n}{\ln n} \right| > \frac{14n (\ln \ln n)^{1/2}}{(\ln n)^{3/2}},$$

стремится к 0 с ростом n .

Из этой теоремы следует, что почти все графы рассматриваемого вида с n ребрами имеют приблизительно $\frac{2n}{\ln n}$ вершин и что средняя степень (т. е. среднее число ребер, исходящих из одной вершины) этого большинства графов асимптотически равна $\ln n$.

*) Параллельными называются ребра, каждое из которых инцидентно одной и той же паре различных вершин.

**) Мультипликативная константа, фигурирующая во втором утверждении теоремы, может быть заменена меньшей, но порядок правой части неравенства рассматриваемыми здесь методами не может быть понижен, так как он связан с порядком оценки функции $\gamma(n) - 1$ (см. первое утверждение теоремы).

3. Ниже будут использоваться следующие обозначения: $G'(n, k)$ (соответственно $G(n, k)$) — число неизоморфных графов (соответственно неизоморфных связных графов) с n ребрами и k вершинами без параллельных ребер и без петель,

$$\Psi_1(n) = n \ln n - 2n \ln \ln n + n(2 \ln 2 - 2),$$

$$\Psi_2(n) = \Psi_1(n) + \frac{4n \ln \ln n}{\ln n},$$

$$\Psi_3(n) = \Psi_1(n) - 5n,$$

$$\Psi_4(n) = n \ln n - 2n \ln \ln n + n(\ln 2 - 1),$$

$\alpha_1(n), \alpha_2(n), \dots$ — функции, возникающие по ходу изложения и удовлетворяющие условию $\alpha(n) = 1 + O\left(\frac{1}{\ln n}\right)$,

A_1, A_2, \dots — положительные константы, возникающие по ходу изложения. $\beta_1(n) < \beta_2(n)$ (или $\beta_2(n) > \beta_1(n)$) означает, что при достаточно больших n выполняется условие $\beta_1(n) \leq \beta_2(n)$.

Пусть \mathfrak{A} — некоторое множество графов. Через $N(\mathfrak{A})$ будем обозначать число (неизоморфных) элементов этого множества.

4.1. Лемма 1.

$$G(m, k) \leq k^2 \left(\frac{e}{2}\right)^m 8^k \left(\frac{k^2}{m}\right)^{m-k}.$$

Доказательство. Как известно [12], всякий связный граф без петель и параллельных ребер с m ребрами и k вершинами может быть получен из некоторого дерева *) с k вершинами (и $k-1$ ребрами) в результате добавления $m-k+1$ ребер, инцидентных соответственно некоторым различным парам вершин (не соединенным ребром в дереве); число таких пар вершин равно

$$C_k^2 - (k-1) = \frac{(k-1)(k-2)}{2}.$$

Так как число неизоморфных деревьев с $k-1$ ребрами не превосходит **) 4^{k-1} , то

$$G(m, k) \leq 4^k C_{\frac{(k-1)(k-2)}{2}}^{m-k+1}. \quad (2)$$

Далее,

$$\begin{aligned} C_{\frac{(k-1)(k-2)}{2}}^{m-k+1} &< \frac{\left(\frac{k^2}{2}\right)^{m-k}}{(m-k)!} \cdot \frac{k^2}{2(m-k+1)} < k^2 \left(\frac{e}{2}\right)^{m-k} \left(\frac{k^2}{m-k}\right)^{m-k} = \\ &= k^2 \left(\frac{e}{2}\right)^{m-k} \left(\frac{k^2}{m}\right)^{m-k} \left(1 + \frac{k}{m-k}\right)^{m-k} < k^2 \frac{e^m}{2^m} 2^k \left(\frac{k^2}{m}\right)^{m-k}. \end{aligned} \quad (3)$$

Из (2) и (3) следует утверждение леммы.

4.2. Лемма 2. Функция

$$R(n, \xi) = \ln(1 + \xi) - \xi + \frac{2 \ln \ln n}{\ln n} (1 + \xi)$$

при одновременном выполнении неравенств

$$-1 < \xi \leq 9 \quad (4)$$

*) То есть связного графа без петель, такого, что для любых двух вершин существует единственная цепь, их соединяющая.

**) См. [2]. Другое доказательство содержится в [15].

и

$$|\xi| > 7 \sqrt{\frac{\ln \ln n}{\ln n}} \quad (5)$$

удовлетворяет условию

$$R(n, \xi) < 0. \quad (6)$$

Доказательство. Положим

$$\varphi(\xi) = \ln(1 + \xi) - \xi + \frac{1}{20} \xi^2.$$

Тогда

$$\varphi'(\xi) = \frac{\xi(\xi - 9)}{10(1 + \xi)}.$$

Кроме того,

$$\varphi(0) = 0, \quad \varphi'(\xi) \geq 0 \quad \text{при } -1 < \xi < 0, \quad \varphi'(\xi) \leq 0 \quad \text{при } 0 < \xi \leq 9.$$

Поэтому при условии (4) $\varphi(\xi) \leq 0$, т. е.

$$\ln(1 + \xi) - \xi \leq -\frac{1}{20} \xi^2. \quad (7)$$

Из (7) имеем

$$R(n, \xi) \leq -\frac{1}{20} \xi^2 + \frac{2 \ln \ln n}{\ln n} \xi + \frac{2 \ln \ln n}{\ln n}.$$

Правая часть последнего неравенства имеет корни

$$\begin{aligned} \xi_1 &= -\sqrt{\frac{40 \ln \ln n}{\ln n}} \left(1 + O\left(\sqrt{\frac{\ln \ln n}{\ln n}}\right) \right), \\ \xi_2 &= \sqrt{\frac{40 \ln \ln n}{\ln n}} \left(1 + O\left(\sqrt{\frac{\ln \ln n}{\ln n}}\right) \right), \end{aligned}$$

откуда, учитывая (5), получаем (6).

Лемма 3. Функция

$$f(n, k) = k\sigma + (n - k)(2 \ln k - \ln n),$$

где $k > 0$, σ — некоторая константа, обладает следующими свойствами:

- 1) $f(n, k) < \Psi_2(n) + O\left(\frac{n}{\ln n}\right)$;
- 2) при $k = \frac{2n}{\ln n}(1 + \xi)$, где $|\xi| > 7 \sqrt{\frac{\ln \ln n}{\ln n}}$,

$$f(n, k) < \Psi_1(n) + O\left(\frac{n}{\ln n}\right)$$
;
- 3) при $k > \frac{20n}{\ln n}$

$$f(n, k) < \Psi_3(n).$$

Доказательство. А. Пусть

$$k = \frac{2n}{\ln n}(1 + \xi) \quad \text{и} \quad -1 < \xi \leq 9. \quad (8)$$

Тогда

$$\begin{aligned} f(n, k) &= \frac{2n}{\ln n}(1 + \xi)\sigma + \left(n - \frac{2n}{\ln n}(1 + \xi)\right)(\ln n - 2 \ln \ln n + \\ &\quad + 2 \ln 2 + 2 \ln(1 + \xi)) = \Psi_1(n) + \frac{4n \ln \ln n}{\ln n} + O\left(\frac{n}{\ln n}\right) + 2n(R_1 + R_2), \end{aligned}$$

где

$$R_1 = \ln(1 + \xi) - \xi, \quad R_2 = \frac{2 \ln \ln n}{\ln n} \xi.$$

Рассмотрим три случая:

а) Если $\xi \leq 0$, то $R_1 \leq 0$, $R_2 \leq 0$.

б) Если $0 < \xi \leq 7 \sqrt{\frac{\ln \ln n}{\ln n}}$, то $R_1 \leq 0$, $R_2 = O\left(\left(\frac{\ln \ln n}{\ln n}\right)^{3/2}\right) = O\left(\frac{1}{\ln n}\right)$.

Таким образом, в случаях а) и б)

$$f(n, k) < \Psi_2(n) + O\left(\frac{n}{\ln n}\right). \quad (9)$$

в) Если (при условии (8)) $|\xi| > 7 \sqrt{\frac{\ln \ln n}{\ln n}}$, то на основании леммы 2

$$R_1 + R_2 + \frac{2 \ln \ln n}{\ln n} \leq 0$$

и

$$f(n, k) < \Psi_1(n) + O\left(\frac{n}{\ln n}\right). \quad (10)$$

Б. Пусть теперь *)

$$k = \frac{n}{\ln n} \zeta, \quad \zeta > 20. \quad (11)$$

Тогда

$$\begin{aligned} f(n, k) &= \sigma \frac{n}{\ln n} \zeta + \left(n - \frac{n}{\ln n} \zeta\right) (\ln n + 2 \ln \zeta - 2 \ln \ln n) = \\ &= \Psi_1(n) + \sigma \frac{n}{\ln n} \zeta + n(2 - 2 \ln 2) - n\zeta + 2n \ln \zeta - \frac{2n}{\ln n} \zeta \ln \zeta + \frac{2n \ln \ln n}{\ln n} \zeta \leq \\ &\leq \Psi_1(n) - n\zeta \left(1 - \frac{2 + 2 \ln \zeta}{\zeta} - \frac{\sigma}{\ln n} - \frac{2 \ln \ln n}{\ln n}\right) \leq \\ &\leq \Psi_1(n) - n\zeta \left(\frac{1}{2} - \frac{\sigma}{\ln n} - \frac{2 \ln \ln n}{\ln n}\right) < \Psi_3(n), \end{aligned} \quad (12)$$

так как при условии (11) $\frac{2 + 2 \ln \zeta}{\zeta} < \frac{1}{2}$.

Поскольку $\Psi_3(n) < \Psi_1(n) < \Psi_2(n)$, первое утверждение леммы следует из (9), (10) и (12), второе — из (10) и (12), третье — из (12).

Лемма доказана.

4.3. Введем обозначения:

$K_n^{(0)}$ — множество натуральных чисел k , удовлетворяющих условию

$$\left|k - \frac{2n}{\ln n}\right| > \frac{14n(\ln \ln n)^{1/2}}{(\ln n)^{3/2}};$$

$K_n^{(1)}$ — множество всех натуральных чисел;

$$G^{(i)}(n) = \sum_{k \in K_n^{(i)}} G(n, k) \quad (i = 0, 1).$$

Очевидно, что $G^{(1)}(n) = G(n)$.

Лемма 4.

$$G^{(i)}(n) \leq \left(\frac{2}{e} \frac{n}{\ln^2 n} e^{c_i \frac{\ln \ln n}{\ln n}} \alpha_1(n)\right)^n \quad (i = 0, 1),$$

*) Очевидно, что если $k > 0$, то для k справедливо одно из соотношений (8) и (11).

где

$$c_0 = 0, \quad c_1 = 4.$$

Доказательство. Так как связный граф с n ребрами имеет не более $n+1$ вершин, то $G(n, k) = 0$ при $k > n+1$. Поэтому на основании леммы 1 имеем

$$\begin{aligned} G^{(i)}(n) &= \sum_{k \in K_n^{(i)}} G(n, k) \leq \sum_{k \in K_n^{(i)}} k^2 \left(\frac{e}{2}\right)^n \cdot 8^k \left(\frac{k^2}{n}\right)^{n-k} \leq \\ &\leq \left(\frac{e}{2}\right)^n A_1 n^3 \max_{k \in K_n^{(i)}} 8^k \left(\frac{k^2}{n}\right)^{n-k}. \end{aligned}$$

Из последнего неравенства на основании леммы 3

$$G^{(i)}(n) \leq A_1 \left(\frac{e}{2}\right)^n n^3 \left(\frac{4}{e^2} \frac{n}{\ln^2 n} e^{c_i \frac{\ln \ln n}{\ln n}} \alpha_2(n)\right)^n = \left(\frac{2}{e} \frac{n}{\ln^2 n} e^{c_i \frac{\ln \ln n}{\ln n}} \alpha_1(n)\right)^n.$$

Лемма доказана.

5.1. Лемма 5.

$$G'(n, k) \geq \frac{\left(\frac{k(k-1)}{2} - n\right)^n}{n! k!}.$$

Доказательство. Каждый граф с n ребрами и k вершинами без параллельных ребер и без петель может быть задан матрицей инцидентий

$$A = \|a_{ij}\|, \quad 1 \leq i \leq n, \quad 1 \leq j \leq k,$$

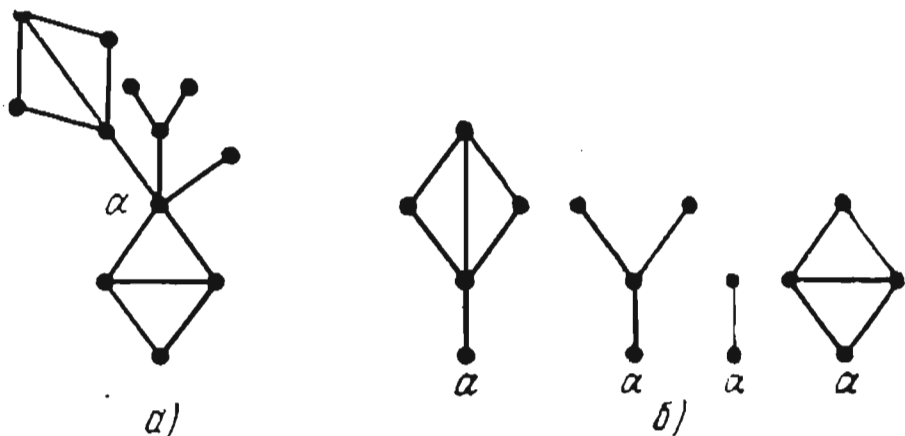
$$a_{ij} = \begin{cases} 1, & \text{если } i\text{-е ребро инцидентно } j\text{-й вершине,} \\ 0 & \text{в противном случае.} \end{cases}$$

Каждая строка матрицы A имеет две единицы; в матрице A все строки попарно различны. Во всех матрицах A (для всех рассматриваемых графов) имеется $K = C_k^2$ различных строк. Поэтому число различных матриц равно $K(K-1) \dots (K-n+1) > (K-n)^n$. Каждый граф может задаваться не более $n! k!$ различными матрицами (получающимися из одной матрицы перестановками строк и столбцов). Поэтому

$$G'(n, k) \geq \frac{(K-n)^n}{n! k!} = \frac{\left(\frac{k(k-1)}{2} - n\right)^n}{n! k!}.$$

Лемма доказана.

5.2. Вершина α связного графа без петель называется *разделяющей*, если все остальные вершины этого графа можно разбить на два (непустых) класса K_1 и K_2 так, что для любых двух вершин β_1 и β_2 , принадлежащих соответственно классам K_1 и K_2 , любой путь, их соединяющий, проходит через α .



Разделяющая вершина α однозначно определяет разбиение графа на минимальные связные

подграфы, α -компоненты, такие, что любые два из них имеют единственную общую вершину — вершину α . Например, α -компоненты графа рисунка а) изображены на рисунке б).

Лемма 6. *Всякий связный граф G без петель содержит вершину, не являющуюся разделяющей *).*

Доказательство. Пусть β_1 — некоторая вершина графа G . Если β_1 не разделяющая, то утверждение доказано. Если же β_1 — разделяющая вершина, то рассмотрим некоторую β_1 -компоненту G_1 графа G и выберем в ней некоторую вершину β_2 , отличную от β_1 . Если β_2 — разделяющая вершина в G_1 , то продолжим процесс дальше. В силу конечности числа вершин в G на некотором n -м шаге мы придем к вершине β_{n+1} , не являющейся разделяющей в подграфе G_n . Эта вершина, очевидно, не будет разделяющей и в G .

Лемма 7. *Если существует множество \mathfrak{A} (не обязательно связных) графов с n ребрами без петель и изолированных вершин, содержащих каждый не более s вершин, то существует множество \mathfrak{B} связных графов с n ребрами без петель такое, что*

$$N(\mathfrak{B}) \geq \frac{1}{s} N(\mathfrak{A}).$$

Доказательство. Пусть $G \in \mathfrak{A}$. Выберем (на основании леммы 6) в каждой связной компоненте одну вершину, не являющуюся разделяющей, и отождествим эти вершины; получится связный граф G' . Рассмотрим множество \mathfrak{B} таких графов G' . Каждый граф G' имеет не более s вершин. Поэтому из него можно получать графы G из \mathfrak{A} , выбирая разделяющую вершину, не более чем s способами. Так как графы из \mathfrak{B} порождают таким образом все графы из \mathfrak{A} , то

$$N(\mathfrak{B}) s \geq N(\mathfrak{A}).$$

Тем самым лемма доказана.

5.3. **Лемма 8.**

$$G(n) \geq \left(\frac{2}{e} \frac{n}{\ln^2 n} e^{\frac{2 \ln \ln n}{\ln n}} \alpha_3(n) \right)^n.$$

Доказательство. Каждый граф с n ребрами имеет не более $2n$ вершин. На основании леммы 7 для каждого k существует не менее $\frac{1}{2n} G'(n, k)$ неизоморфных связных графов без петель и без параллельных ребер и, следовательно,

$$G(n) \geq \frac{1}{2n} G'(n, k). \quad (13)$$

Из (13) и леммы 5, применяя формулу Стирлинга, имеем

$$\ln G(n) \geq n \ln \left(\frac{k^2}{2} - \frac{k}{2} - n \right) - n \ln n + n - k \ln k + k + O(\ln n).$$

Положим

$$k = \left\lfloor \frac{2n}{\ln n} \right\rfloor.$$

Тогда

$$k = \frac{2n}{\ln n} \left(1 + O\left(\frac{\ln n}{n}\right) \right), \quad \ln k = \ln n - \ln \ln n + \ln 2 + O\left(\frac{\ln n}{n}\right),$$

$$k \ln k = 2n - \frac{2n \ln \ln n}{\ln n} + O\left(\frac{n}{\ln n}\right),$$

$$\ln \left(\frac{k^2}{2} - \frac{k}{2} - n \right) = 2 \ln k - \ln 2 + O\left(\frac{\ln^2 n}{n}\right)$$

*) Ср. [16].

и

$$\ln G(n) \geq n \ln n - 2n \ln \ln n + n(\ln 2 - 1) + \frac{2n \ln \ln n}{\ln n} + O\left(\frac{n}{\ln n}\right).$$

Лемма доказана.

6. Доказательство теоремы 1. Первое утверждение непосредственно следует из леммы 4 при $i = 1$ и леммы 8, второе — из леммы 4 при $i = 0$ и леммы 8.

7. Сформулируем теперь два более общих предложения (доказательство их содержится во втором параграфе).

Пусть \mathfrak{G} — множество всех графов без изолированных вершин, $\mathfrak{G}^1, \mathfrak{G}^2, \mathfrak{G}^3$ — подмножества этого множества, состоящие соответственно из связных графов, графов без параллельных ребер, графов без петель. Введем обозначения:

$$\begin{aligned} \mathfrak{G}_0^i &= \mathfrak{G}^i, \quad \mathfrak{G}_1^i = \mathfrak{G} \quad (i = 1, 2, 3), \\ \mathfrak{G}_{\delta_1, \delta_2, \delta_3} &= \mathfrak{G}_{\delta_1}^1 \cap \mathfrak{G}_{\delta_2}^2 \cap \mathfrak{G}_{\delta_3}^3 \quad (\delta_i = 0, 1 \text{ при } i = 1, 2, 3). \end{aligned}$$

Например, $\mathfrak{G}_{1,1,1} = \mathfrak{G}$, а $\mathfrak{G}_{0,1,0}$ есть множество всех связных графов без петель.

Пусть $\mathfrak{G}_{\delta_1, \delta_2, \delta_3}(n)$ — множество графов из $\mathfrak{G}_{\delta_1, \delta_2, \delta_3}$, имеющих n ребер, и $G_{\delta_1, \delta_2, \delta_3}(n)$ — число (неизоморфных) графов в $\mathfrak{G}_{\delta_1, \delta_2, \delta_3}(n)$.

Заметим, что

$$G_{0,0,0}(n) = G(n). \quad (14)$$

Аналогичные обозначения вводятся для сетей (в них вместо \mathfrak{G} и G фигурируют соответственно \mathfrak{S} и S).

Теорема 2. 1)

$$G_{\delta_1, \delta_2, \delta_3}(n) = \left(\frac{2}{e} \frac{n}{\ln^2 n} \gamma_{\delta_1, \delta_2, \delta_3}(n) \right)^n,$$

где

$$\frac{2 \ln \ln n}{\ln n} \leq \gamma_{\delta_1, \delta_2, \delta_3}(n) - 1 \leq \frac{4 \ln \ln n}{\ln n}.$$

2) Доля графов из $\mathfrak{G}_{\delta_1, \delta_2, \delta_3}$, имеющих n ребер и k вершин, где *)

$$\left| k - \frac{2n}{\ln n} \right| > \frac{18n (\ln \ln n)^{1/2}}{(\ln n)^{3/2}},$$

стремится к 0 с ростом n .

Доказательство теоремы 2 опирается на нижнюю оценку для $G_{0,0,0}(n)$ и верхнюю оценку для $G_{1,1,1}(n)$.

Теорема 3. 1)

$$S_{\delta_1, \delta_2, \delta_3}(n) = \left(\frac{2}{e} \frac{n}{\ln^2 n} \sigma_{\delta_1, \delta_2, \delta_3}(n) \right)^n,$$

где

$$\frac{2 \ln \ln n}{\ln n} \leq \sigma_{\delta_1, \delta_2, \delta_3}(n) - 1 \leq \frac{4 \ln \ln n}{\ln n}.$$

2) Доля сетей из $\mathfrak{S}_{\delta_1, \delta_2, \delta_3}$, имеющих n ребер и k вершин, где *)

$$\left| k - \frac{2n}{\ln n} \right| > \frac{18n (\ln \ln n)^{1/2}}{(\ln n)^{3/2}},$$

стремится к 0 с ростом n .

*) См. вторую сноску на стр. 6.

§ 2

8.1. Введем дополнительно обозначения:

$G_{\delta_1, \delta_2, \delta_3}(n, k)$ — число неизоморфных графов из $\mathcal{G}_{\delta_1, \delta_2, \delta_3}$, имеющих n ребер и k вершин,

$G_{\delta_1, \delta_2, \delta_3}(n, k, s)$ — число неизоморфных графов из $\mathcal{G}_{\delta_1, \delta_2, \delta_3}$, имеющих n ребер и k вершин и состоящих из s связных компонент.

Лемма 9.

$$G_{1, 0, 0}(n, k, s) \leq (k - s + 1) G_{0, 0, 0}(n, k - s + 1).$$

Лемма 10.

$$G_{1, 0, 0}(n) \leq (n + 1) G_{0, 0, 0}(n).$$

Доказательство этих утверждений вполне аналогично доказательству леммы 7.

Лемма 11. Если $k \in K_n^{(0)}$ (см. стр. 9),

то

$$G_{0, 0, 0}(n, k) \leq \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_4(n) \right)^n.$$

Эта лемма непосредственно следует из леммы 4.

Лемма 12.

$$G_{\delta_1, 1, 0}(n, k) \leq \sum_{m=0}^n G_{\delta_1, 0, 0}(m, k) C_n^m.$$

Доказательство. Утверждение следует из того, что:

1) каждый граф, имеющий k вершин и n ребер, может быть получен из некоторого графа G без параллельных ребер, имеющего k вершин и m ребер, $0 \leq m \leq n$, путем добавления некоторых $n - m$ ребер, параллельных некоторым ребрам из G (при этом допускается введение нескольких ребер, параллельных одному и тому же ребру);

2) число неизоморфных графов, которые могут быть получены таким способом из одного графа G , не превосходит числа сочетаний с повторениями из m элементов по $n - m$, т. е.

$$C_{n-m}^{n-m} = C_{n-m}^{m-1} \leq C_n^m,$$

Следствие.

$$G_{1, 1, 0}(n) = \sum_k G_{1, 1, 0}(n, k) \leq \sum_{m=0}^n C_n^m \sum_k G_{1, 0, 0}(m, k) = \sum_{m=0}^n G_{1, 0, 0}(m) C_n^m.$$

Лемма 13.

$$G_{1, 1, 1}(n, k) \leq \sum_{l=0}^k \sum_{p=0}^{n-k+l} G_{1, 1, 0}(p, l) C_{n-p}^{n-p-l}.$$

Доказательство. Каждый граф G из $\mathcal{G}_{1, 1, 1}$ с n ребрами и k вершинами может быть получен из некоторого (быть может, пустого) графа \tilde{G} из $\mathcal{G}_{1, 1, 0}$ с p ребрами ($0 \leq p \leq n$) и l вершинами ($0 \leq l \leq k$) следующим образом: 1) к \tilde{G} добавляется $k - l$ новых вершин с петлями (однозначно); 2) в полученном графе (с k вершинами) к некоторым вершинам присоединяется $n - p - k + l$ петель; число способов присоединения этих петель равно числу N сочетаний с повторениями из k элементов по $n - p - k + l$, т. е.

$$N \leq C_{n-p-k+l}^{n-p-k+l}.$$

Так как $k \geq l$ и при $c \geq 0$ $C_{a+c}^{b+c} \geq C_a^b$ и $C_{a+c}^b \geq C_a^b$, то

$$N \leq C_{n-p+l}^{n-p-k+l} \leq C_{n-p+k}^{n-p} \leq C_{3n-p}^{n-p},$$

поскольку во всяком графе из \mathcal{G} $k \leq 2n$.

Лемма доказана.

Следствие.

$$\begin{aligned} G_{1,1,1}(n) &= \sum_k G_{1,1,1}(n, k) \leq \sum_k \sum_l \sum_p G_{1,1,0}(p, l) C_{3n-p}^{n-p} \leq \\ &\leq \sum_k \sum_l \sum_p G_{1,1,0}(p) C_{3n-p}^{n-p} \leq A_2 n^3 \max_{0 \leq p \leq n} (G_{1,1,0}(p) C_{3n-p}^{n-p}). \end{aligned}$$

8.2. Введем функцию

$$\tilde{\Psi}_{\gamma, \delta}(m) = m \ln m - 2m \ln \ln m + \gamma m + \delta \frac{m \ln \ln m}{\ln m}.$$

Лемма 14. Если $m \leq n$, то

$$\tilde{\Psi}_{\gamma, \delta}(m) + \ln(C_n^m C_{3n-m}^{n-m}) \leq \tilde{\Psi}_{\gamma, \delta}(n) + O\left(\frac{n}{\ln n}\right). \quad (15)$$

Доказательство. Введем обозначение

$$\Phi(n, m) = \tilde{\Psi}_{\gamma, \delta}(m) + \ln(C_n^m C_{3n-m}^{n-m}).$$

Тогда, используя формулу Стирлинга, получим

$$\begin{aligned} \Phi(n, m) &= (3n-m) \ln(3n-m) + n \ln n - 2(n-m) \ln(n-m) - 2n \ln(2n) - \\ &- 2m \ln \ln m + \gamma m + \delta \frac{m \ln \ln m}{\ln m} + O(\ln n). \end{aligned} \quad (16)$$

Перейдем теперь непосредственно к доказательству неравенства (15).

1°. Пусть

$$m \leq n \left(1 - \frac{1}{\ln \ln n}\right).$$

Тогда

$$n-m \geq \frac{n}{\ln \ln n}, \quad \ln \frac{3n-m}{n-m} \leq \ln \ln \ln n + \ln 3.$$

В этом случае из (16) имеем

$$\begin{aligned} \Phi(n, m) &\leq m \ln(3n-m) + 2(n-m) \ln \frac{3n-m}{n-m} + n \ln \frac{n(3n-m)}{(2n)^2} + \gamma m + \\ &+ \delta \frac{m \ln \ln m}{\ln m} + O(\ln n) \leq n \left(1 - \frac{1}{\ln \ln n}\right) (\ln n + \ln 3) + 2n (\ln \ln \ln n + \ln 3) + \\ &+ n \ln \frac{3}{4} + \gamma n + \delta \frac{n \ln \ln n}{\ln n} + O(\ln n) \leq n \ln n - \frac{n \ln n}{\ln \ln n} + 2n \ln \ln \ln n + \\ &+ n \left(3 \ln 3 + \ln \frac{3}{4}\right) + \gamma n + \delta \frac{n \ln \ln n}{\ln n} + O(\ln n) < \tilde{\Psi}_{\gamma, \delta}(n), \end{aligned}$$

так как

$$3 \ln 3 + \ln \frac{3}{4} + 2 \ln \ln \ln n - \frac{\ln n}{\ln \ln n} + O(\ln n) < 2 \ln \ln n.$$

2°. Пусть

$$m = n(1 - \xi), \quad 0 \leq \xi < \frac{1}{\ln \ln n}.$$

Тогда

$$\ln \left(1 + \frac{\xi}{2} \right) \leq \frac{\xi}{2}, \quad \ln m = \ln n + O\left(\frac{1}{\ln \ln n}\right),$$

$$\ln \ln m = \ln \ln n + O\left(\frac{1}{\ln n \cdot \ln \ln n}\right).$$

Поэтому из (16) имеем

$$\begin{aligned} \Phi(n, m) &< n(2 + \xi) \ln n + n(2 + \xi) \ln 2 + n(2 + \xi) \frac{\xi}{2} + \\ &+ n \ln n - 2n\xi \ln n - 2n\xi \ln \xi - 2n \ln n - 2n \ln 2 - \\ &- 2n(1 - \xi) \left(\ln \ln n + O\left(\frac{1}{\ln n \cdot \ln \ln n}\right) \right) + \gamma n + \delta \frac{n \ln \ln n}{\ln n} + O(\ln n) = \\ &= \tilde{\Psi}_{\gamma, \delta}(n) + O\left(\frac{n}{\ln n \cdot \ln \ln n}\right) + R, \end{aligned}$$

где

$$R = -n\xi \left(\ln n + 2 \ln \xi - \ln 2 - 1 - \frac{\xi}{2} - 2 \ln \ln n \right). \quad (17)$$

$$\text{Если } \xi < \frac{1}{\ln^2 n}, \text{ то } n\xi \ln n = O\left(\frac{n}{\ln n}\right)$$

и

$$R = -2n\xi \ln(n\xi) + n\xi \left(2 \ln \ln n + 1 + \ln 2 + \frac{\xi}{2} \right) + O\left(\frac{n}{\ln n}\right) = O\left(\frac{n}{\ln n}\right),$$

так как первое слагаемое не превосходит $\frac{2}{e}$, а второе есть величина порядка $O\left(\frac{n \ln \ln n}{\ln^2 n}\right)$.

Если $\xi \geq \frac{1}{\ln^2 n}$, то $\ln \xi \geq -2 \ln \ln n$ и выражение в скобках в (17)

при достаточно больших n положительно, поэтому $R < 0$.

Лемма доказана.

8.3. Лемма 15.

$$G_{1, 1, 1}(n) \leq \left(\frac{2}{e} \frac{n}{\ln^2 n} e^{\frac{4 \ln \ln n}{\ln n}} \alpha_5(n) \right)^n.$$

Доказательство. На оснований следствий из лемм 13 и 12 и лемм 10, 4 (учитывая (14) и лемму 14) имеем

$$\begin{aligned} G_{1, 1, 1}(n) &\leq A_2 n^3 \max_{0 \leq p \leq n} (G_{1, 1, 0}(p) C_{3n-p}^{n-p}) \leq \\ &\leq A_2 n^3 \max_{0 \leq p \leq n} \left(C_{3n-p}^{n-p} \sum_{m=0}^p G_{1, 0, 0}(m) C_p^m \right) \leq \\ &\leq A_3 n^4 \max_{0 \leq p \leq n} \max_{0 \leq m \leq p} (C_{3n-p}^{n-p} C_p^m G_{1, 0, 0}(m)) \leq \\ &\leq A_3 n^4 \max_{0 \leq m \leq n} (C_{3n-m}^{n-m} C_n^m G_{1, 0, 0}(m)) \leq \\ &\leq A_4 n^5 \max_{0 \leq m \leq n} (C_{3n-m}^{n-m} C_n^m G_{0, 0, 0}(m)) \leq \\ &\leq A_4 n^5 \max_{0 \leq m \leq n} \left(C_{3n-m}^{n-m} C_n^m \left(\frac{2}{e} \frac{m}{\ln^2 m} e^{\frac{4 \ln \ln m}{\ln m}} \alpha_1(m) \right)^m \right) \leq \\ &\leq A_4 n^5 \left(\frac{2}{e} \frac{n}{\ln^2 n} e^{\frac{4 \ln \ln n}{\ln n}} \alpha_5(n) \right)^n = \left(\frac{2}{e} \frac{n}{\ln^2 n} e^{\frac{4 \ln \ln n}{\ln n}} \alpha_5(n) \right)^n. \end{aligned}$$

Лемма доказана.

8.4. Лемма 16. Существует функция $g(n)$, удовлетворяющая условиям:

1) для всех n

$$\ln G_{0,0,0}(n) \leq g(n),$$

$$2) g(n) = n \ln n - 2n \ln \ln n + (\ln 2 - 1)n + \frac{5n \ln \ln n}{\ln n} + O\left(\frac{n}{\ln n}\right),$$

3) для любых n_1, n_2 таких, что $n_1 \geq n_2 > 1$,

$$g(n_1) + g(n_2) \leq g(n_1 + 1) + g(n_2 - 1).$$

Доказательство. На основании леммы 4

$$\ln G_{0,0,0}(n) \leq n \ln n - 2n \ln \ln n + (\ln 2 - 1)n + \frac{4n \ln \ln n}{\ln n} + O\left(\frac{n}{\ln n}\right).$$

Поэтому

$$\ln G_{0,0,0}(n) \leq n \ln n - 2n \ln(4 + \ln n) + (\ln 2 - 1)n + \frac{4n \ln(4 + \ln n)}{4 + \ln n} + O\left(\frac{n}{\ln n}\right)$$

и

$$\ln G_{0,0,0}(n) < n \ln n - 2n \ln(4 + \ln n) + (\ln 2 - 1)n + \frac{5n \ln(4 + \ln n)}{4 + \ln n},$$

и для некоторой константы A_5 неравенство

$$\ln G_{0,0,0}(n) \leq n \ln n - 2n \ln(4 + \ln n) + (\ln 2 - 1)n + \frac{5n \ln(4 + \ln n)}{4 + \ln n} + A_5 \quad (18)$$

справедливо для всех $n, n \geq 1$. Функцию, выражаемую правой частью неравенства (18), и возьмем в качестве $g(n)$. Эта функция, очевидно, удовлетворяет условиям 1) и 2).

Далее имеем (рассматривая $g(n)$ как функцию непрерывного аргумента)

$$g'(n) = 1 + \ln n - \frac{2}{4 + \ln n} - 2 \ln(4 + \ln n) + (\ln 2 - 1) + 5 \frac{1 + (4 + \ln n) \ln(4 + \ln n) - \ln(4 + \ln n)}{(4 + \ln n)^2},$$

$$g''(n) = \frac{1}{n} \left(1 - \frac{2}{4 + \ln n} - \frac{5 \ln(4 + \ln n)}{(4 + \ln n)^2} + \frac{7}{(4 + \ln n)^2} + \frac{10 \ln(4 + \ln n)}{(4 + \ln n)^3} - \frac{15}{(4 + \ln n)^3} \right).$$

Очевидно, что при $n \geq 1$

$$1 - \frac{2}{4 + \ln n} - \frac{5 \ln(4 + \ln n)}{(4 + \ln n)^2} \geq \frac{1}{2} - \frac{5 \ln 4}{16} > 0, \quad \frac{10 \ln(4 + \ln n)}{(4 + \ln n)^3} > 0,$$

$$7 - \frac{15}{4 + \ln n} \geq 7 - \frac{15}{4} > 0.$$

Поэтому при $n \geq 1$ справедливо неравенство $g''(n) > 0$, откуда легко получить (3).

Лемма доказана.

Лемма 17. Если

$$\left| k - \frac{2n}{\ln n} \right| > 16 \frac{n (\ln \ln n)^{1/2}}{(\ln n)^{3/2}},$$

то

$$G_{1,0,0}(n, k) < \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_7(n) \right)^n.$$

Доказательство. Рассмотрим три случая:

$$I. \quad k \leq \frac{2n}{\ln n} \left(1 - 8 \sqrt{\frac{\ln \ln n}{\ln n}} \right).$$

В этом случае

$$k - s + 1 \leq k < \frac{2n}{\ln n} \left(1 - 7 \sqrt{\frac{\ln \ln n}{\ln n}} \right),$$

где s — число связных компонент. Поэтому в силу леммы 11

$$G_{0,0,0}(n, k - s + 1) \leq \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_4(n) \right)^n$$

и по лемме 9

$$\begin{aligned} G_{1,0,0}(n, k) &= \sum_{s=1}^k G_{1,0,0}(n, k, s) \leq \sum_{s=1}^k (k - s + 1) G_{0,0,0}(n, k - s + 1) \leq \\ &\leq k^2 \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_4(n) \right)^n \leq \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_8(n) \right)^n. \end{aligned}$$

$$II. \quad k > \frac{5n}{\ln n}.$$

Так как при отсутствии петель каждая связная компонента имеет по крайней мере две вершины, то в этом случае

$$k \geq 2s \tag{19}$$

$$k - s + 1 \geq \frac{k}{2} > \frac{5}{2} \frac{n}{\ln n} > \frac{2n}{\ln n} \left(1 + 7 \sqrt{\frac{\ln \ln n}{\ln n}} \right).$$

Поэтому в силу леммы 11

$$G_{0,0,0}(n, k - s + 1) \leq \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_4(n) \right)^n$$

и по лемме 9 (учитывая (19))

$$G_{1,0,0}(n, k) \leq \sum_{s=1}^{\left[\frac{k}{2} \right]} G_{1,0,0}(n, k, s) \leq \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_9(n) \right)^n.$$

$$III. \quad \frac{2n}{\ln n} \left(1 + 8 \sqrt{\frac{\ln \ln n}{\ln n}} \right) \leq k \leq \frac{5n}{\ln n}. \tag{20}$$

Оценим сначала функцию $G_{1,0,0}(n, k, s)$ (при условии (20)) в зависимости от s . Рассмотрим два подслучая:

1) Пусть

$$s < \frac{2n (\ln \ln n)^{1/2}}{(\ln n)^{3/2}}.$$

Тогда

$$k - s + 1 > \frac{2n}{\ln n} \left(1 + 7 \sqrt{\frac{\ln \ln n}{\ln n}} \right)$$

и в силу лемм 9 и 11

$$\begin{aligned} G_{1,0,0}(n, k, s) &\leq (k - s + 1) G_{0,0,0}(n, k - s + 1) \leq \\ &\leq A_6 n \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_4(n) \right)^n = \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_{10}(n) \right)^n. \end{aligned} \tag{21}$$

2) Пусть теперь

$$s \gg \frac{2n (\ln \ln n)^{1/2}}{(\ln n)^{3/2}}. \quad (22)$$

В силу (19) и (20)

$$s \leq \frac{5n}{2 \ln n}. \quad (23)$$

Очевидно, что

$$\begin{aligned} G_{1,0,0}(n, k, s) &\leq \sum_{\substack{n_1+n_2+\dots+n_s=n \\ k_1+k_2+\dots+k_s=k \\ n_i \geq 1}} G_{0,0,0}(n_1, k_1) \cdot G_{0,0,0}(n_2, k_2) \dots G_{0,0,0}(n_s, k_s) \leq \\ &\leq C_n^{s-1} \cdot 2^k \max_{n_1+n_2+\dots+n_s=n} (G_{0,0,0}(n_1) \cdot G_{0,0,0}(n_2) \dots G_{0,0,0}(n_s)), \end{aligned}$$

так как $G_{0,0,0}(n_i, k_i) \leq G_{0,0,0}(n_i)$ и число представлений числа n в виде суммы s штук (неупорядоченных) слагаемых не превосходит C_n^{s-1} , а число представлений числа k не превосходит 2^k . В силу леммы 16 из последнего неравенства вытекает

$$G_{1,0,0}(n, k, s) \leq C_n^{s-1} \cdot 2^k e^{g(n-s+1)} e^{(s-1)g(1)}. \quad (24)$$

Из (22) имеем

$$-(s-1) \ln(s-1) < -\frac{s}{2} \ln \frac{s}{2} \leq -\frac{n (\ln \ln n)^{1/2}}{(\ln n)^{1/2}} \left(1 + O\left(\frac{\ln \ln n}{\ln n}\right) \right), \quad (25)$$

а из (23)

$$-2(n-s+1) \ln \ln(n-s+1) \leq -2n \ln \ln n + O\left(n \frac{\ln \ln n}{\ln n}\right). \quad (26)$$

Поэтому из (24), леммы 16, (20), (23), (25), (26) имеем

$$\begin{aligned} \ln G_{1,0,0}(n, k, s) &\leq \\ &\leq n \ln n - (s-1) \ln(s-1) - (n-s+1) \ln(n-s+1) + O(\ln n) + \\ &\quad + k \ln 2 + g(n-s+1) + sg(1) < \\ &< n \ln n - \frac{n (\ln \ln n)^{1/2}}{(\ln n)^{1/2}} - 2n \ln \ln n + (\ln 2 - 1)n + O\left(n \frac{\ln \ln n}{\ln n}\right) < \Psi_4(n). \end{aligned} \quad (27)$$

Таким образом, при условии (20) из (21) и (27) имеем

$$G_{1,0,0}(n, k) = \sum_{s=1}^n G_{1,0,0}(n, k, s) \leq \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_{11}(n) \right)^n.$$

Лемма полностью доказана.

8.5. Лемма 18. Если

$$\left| k - \frac{2n}{\ln n} \right| \geq 18 \frac{n (\ln \ln n)^{1/2}}{(\ln n)^{3/2}}, \quad (28)$$

то

$$G_{1,1,1}(n, k) < \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_{12}(n) \right)^n. \quad (29)$$

Доказательство. Из лемм 13 и 12 имеем

$$\begin{aligned}
 G_{1,1,1}(n, k) &\leq \sum_{l=0}^k \sum_{p=0}^{n-k+l} G_{1,1,0}(p, l) C_{3n-p}^{n-p} \leq \\
 &\leq \sum_{l=0}^k \sum_{p=0}^{n-k+l} C_{3n-p}^{n-p} \sum_{m=0}^p G_{1,0,0}(m, l) C_p^m \leq \\
 &\leq \sum_{l=0}^k \sum_{p=0}^{n-k+l} \sum_{m=0}^p G_{1,0,0}(m, l) C_{3n-m}^{n-m} C_n^m \leq \\
 &\leq A_7 n^3 \max_{l \leq k} F(n, k, l),
 \end{aligned} \tag{30}$$

где

$$F(n, k, l) = \max_{m \leq n-k+l} C_{3n-m}^{n-m} C_n^m G_{1,0,0}(m, l).$$

Для оценки функции $G_{1,0,0}(m, l)$ рассмотрим несколько случаев.

а) $m \leq n \left(1 - \frac{1}{\sqrt{\ln n}}\right)$; l произвольно.

Так как

$$\ln G_{1,0,0}(m, l) \leq \ln G_{1,0,0}(m), \tag{31}$$

то в силу лемм 4 и 10

$$\begin{aligned}
 \ln G_{1,0,0}(m, l) &\leq n \left(1 - \frac{1}{\sqrt{\ln n}}\right) \ln n + (\ln 2 - 1)n + \\
 &+ \frac{4n \ln \ln n}{\ln n} + O\left(\frac{n}{\ln n}\right) < \Psi_4(n) + O\left(\frac{n}{\ln n}\right).
 \end{aligned} \tag{32}$$

б) $m = n(1 - \varepsilon)$, $\frac{1}{\sqrt{\ln n}} > \varepsilon \geq \varphi(n)$, где $\varphi(n) = \frac{5 \ln \ln n}{\ln^2 n}$; l произвольно.

В этом случае

$$\ln m = \ln n - \varepsilon + O(\varepsilon^2), \quad \ln \ln m = \ln \ln n + O\left(\frac{\varepsilon}{\ln n}\right)$$

и в силу (31) и лемм 4 и 10

$$\begin{aligned}
 \ln G_{1,0,0}(m, l) &\leq n(1 - \varepsilon)(\ln n - \varepsilon + O(\varepsilon^2)) - 2n(1 - \varepsilon) \left(\ln \ln n + O\left(\frac{\varepsilon}{\ln n}\right)\right) + \\
 &+ (\ln 2 - 1)n + \frac{4n \ln \ln n}{\ln n} + O\left(\frac{n}{\ln n}\right) = \\
 &= \Psi_4(n) + n \left(-\varepsilon \ln n + 2\varepsilon \ln \ln n + \frac{4 \ln \ln n}{\ln n}\right) + O\left(\frac{n}{\ln n}\right) \leq \\
 &\leq \Psi_4(n) + n\varepsilon \left(-\frac{1}{5} \ln n + 2 \ln \ln n\right) + \\
 &+ O\left(\frac{n}{\ln n}\right) < \Psi_4(n) + O\left(\frac{n}{\ln n}\right).
 \end{aligned} \tag{33}$$

$$\text{в) } n(1 - \varphi(n)) < m \leq n; \quad k - l < n\varphi(n), \tag{34}$$

где $\varphi(n)$ имеет тот же смысл, что и в б). Тогда

$$n < \frac{m}{1 - \varphi(n)} < \frac{m}{1 - \varphi(m)}$$

и

$$\begin{aligned}
 \frac{2n}{\ln n} \left(1 - 9 \sqrt{\frac{\ln \ln n}{\ln n}}\right) &< \frac{2m}{(1 - \varphi(m)) \ln m} \left(1 - 9 \sqrt{\frac{\ln(\ln m - \ln(1 - \varphi(m)))}{\ln m - \ln(1 - \varphi(m))}}\right) < \\
 &< \frac{2m}{\ln m} \left(1 - 8 \sqrt{\frac{\ln \ln m}{\ln m}}\right).
 \end{aligned} \tag{35}$$

Поэтому, если

$$k \leq \frac{2n}{\ln n} \left(1 - 9 \sqrt{\frac{\ln \ln n}{\ln n}} \right),$$

то в силу (30), (35)

$$l < \frac{2m}{\ln m} \left(1 - 8 \sqrt{\frac{\ln \ln m}{\ln m}} \right)$$

и по лемме 17

$$G_{1,0,0}(m, l) < \left(\frac{2}{e} \frac{m}{\ln^2 m} \alpha_7(m) \right)^m. \quad (36)$$

Если

$$k \geq \frac{2n}{\ln n} \left(1 + 9 \sqrt{\frac{\ln \ln n}{\ln n}} \right),$$

то в силу (34)

$$\begin{aligned} l &> \frac{2n}{\ln n} \left(1 + 9 \sqrt{\frac{\ln \ln n}{\ln n}} \right) - n\varphi(n) > \frac{2n}{\ln n} \left(1 + 8 \sqrt{\frac{\ln \ln n}{\ln n}} \right) > \\ &> \frac{2m}{\ln m} \left(1 + 8 \sqrt{\frac{\ln \ln m}{\ln m}} \right) \end{aligned}$$

и по лемме 17 снова справедливо неравенство (36).

Случай

$$n(1 - \varphi(n)) < m \leq n; \quad k - l \geq n\varphi(n) \quad (37)$$

невозможен, так как из второго неравенства и $m \leq n - k + l$ (см. (30)) вытекает $m \leq n(1 - \varphi(n))$, что противоречит первому неравенству в (37).

Из (32), (33), (36) в силу леммы 14 имеем

$$C_{3n-m}^{n-m} C_n^m G_{1,0,0}(m, l) \leq \left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_{13}(n) \right)^n,$$

а из последнего неравенства и (30) вытекает (29).

Лемма полностью доказана.

Следствие. Число графов из $\mathfrak{G}_{1,1,1}$ с n ребрами, число k вершин которых удовлетворяет условию (28), не превосходит

$$\left(\frac{2}{e} \frac{n}{\ln^2 n} \alpha_{14}(n) \right)^n.$$

9. Доказательство теоремы 2. Очевидно, что если

$$\delta'_1 \leq \delta''_1, \quad \delta'_2 \leq \delta''_2, \quad \delta'_3 \leq \delta''_3,$$

то

$$G_{\delta'_1, \delta'_2, \delta'_3}(n) \leq G_{\delta''_1, \delta''_2, \delta''_3}(n).$$

Поэтому первое утверждение теоремы вытекает из лемм 8 и 15 (учитывая (14)), а второе утверждение — из леммы 8 и следствия из леммы 18.

10. Наметим идею доказательства теоремы 3.

Легко видеть, что

$$G_{0,0,0}(n, k) \leq S_{\delta_1, \delta_2, \delta_3}(n, k) \leq 2^k G_{1,1,1}(n, k).$$

Далее, используя третье утверждение леммы 3, легко установить, что при $k \geq \frac{A_8 n}{\ln n}$

$$G_{1,1,1}(n, k) < \left(\frac{1}{e} \frac{n}{\ln^2 n} \alpha_{15}(n) \right)^n.$$

Поэтому

$$S_{\delta_1, \delta_2, \delta_3}(n) = \sum_{k < \frac{A_8 n}{\ln n}} S_{\delta_1, \delta_2, \delta_3}(n, k) + \sum_{k \geq \frac{A_8 n}{\ln n}} S_{\delta_1, \delta_2, \delta_3}(n, k) \leq \\ \leq \frac{A_8 n}{\ln n} \cdot 2^{\frac{A_8 n}{\ln n}} G_{1,1,1}(n) + 2^n \left(\frac{1}{e} \frac{n}{\ln^2 n} \alpha_{15}(n) \right)^n = \left(\frac{2}{e} \frac{n}{\ln^2 n} e^{\frac{4 \ln \ln n}{\ln n}} \alpha_{16}(n) \right)^n.$$

Отсюда и из леммы 8 вытекает первое утверждение теоремы 3. Второе утверждение доказывается аналогично.

ЛИТЕРАТУРА

- [1] Яблонский С. В., Основные понятия кибернетики. Сб. Проблемы кибернетики, вып. 2, 1959, 7—38.
- [2] P o l y a G., Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemischen Verbindungen, Acta Math. 68, 1—2, 1937, 145—254.
- [3] R i o r d a n J., S h a n n o n C. E., The number of two-terminal series-parallel networks, J. Math. and Phys. 21, 2, 1942, 83—93.
- [4] S h a n n o n C. E., The synthesis of two-terminal switching circuits, Bell. Syst. Techn. J. 28, 1, 1949, 59—98.
- [5] G i l b e r t E. N., The synthesis of N -terminal switching circuits, Bell. Syst. Techn. J. 30, 3, 1951, 668—688.
- [6] П о в а р о в Г. Н., Математическая теория синтеза контактных $(1, k)$ -полюсников, ДАН 100, 5, 1955, 909—912.
- [7] К р и ч е в с к и й Р. Е., О реализации функций суперпозициями. Сб. Проблемы кибернетики, вып. 2, 1959, 123—138.
- [8] D a v i s R. L., The numbers of structures of finite relations. Proc. Amer. Math. Soc. 4, 3, 1953, 486—495.
- [9] В е т у х н о в с к и й Ф. Я., О числе неразложимых сетей и некоторых их свойствах, ДАН 123, 3, 1958, 391—394.
- [10] Л у п а н о в О. Б., О возможностях синтеза схем из произвольных элементов, Труды МИАН, т. 51, 1958, 158—173.
- [11] Т р а х т е н б р о т Б. А., К теории неповторных контактных схем, Труды МИАН, т. 51, 1958, 226—269.
- [12] К ö n i g D., Theorie der endlichen und unendlichen Graphen, Leipzig, 1936.
- [13] К у д р я в ц е в Л. Д., О некоторых математических вопросах теории электрических цепей, УМН 3, 4, 1948, 80—117.
- [14] Яблонский С. В., Функциональные построения в k -значной логике, Труды МИАН, т. 51, 1958, 5—142.
- [15] Л у п а н о в О. Б., Об одном методе синтеза схем, Известия ВУЗ Радиофизика, 1, 1958, 120—140.
- [16] K e l l y P., M e r r i e l D., On the transpose-connectivity of graphs. Math. Mag. 32, 1, 1958, 1—3.
- [17] F o r d G. W., N o r m a n R. Z., U h l e n b e c k G. E., Combinatorial problems in the theory of graphs. I, II, III, IV. Proc. Nat. Acad. Sci. USA, 1956, 42, 3, 122—128, 4, 203—208, 8, 529—535, 1957, 43, 1, 163—167.
- [18] Л у п а н о в О. Б., Об асимптотических оценках числа графов с k ребрами, ДАН 126, 3, 1959, 498—500.

Поступило в редакцию 28 XI 1958.

ОБ ОДНОМ КЛАССЕ АВТОМАТОВ (НЕРВНЫЕ СИСТЕМЫ)*)

Л. А. СКОРНЯКОВ

(МОСКВА)

В настоящей заметке исследуются некоторые автоматы, являющиеся обобщением нервных сетей, рассмотренных Клини [2].

§ 1. Лабиринты и частично упорядоченные множества

Множество S назовем *лабиринтом*, если между некоторыми его элементами установлено бинарное отношение $x \triangleright y$ (словами: « x покрывает y »). Элемент лабиринта, не покрывающий никакого элемента, называется *входным*, а не покрываемый никаким элементом — *выходным*. Остальные элементы назовем *внутренними*. Множество входных (выходных) элементов лабиринта S называется его *входом* (*выходом*) и обозначается через $U(S)$ [$W(S)$]. Множество элементов, покрываемых элементом s , будем называть *s-основанием* и обозначать через $\Pi(s)$. Последовательность x_1, \dots, x_n, \dots различных элементов лабиринта называется *возрастающей* или *убывающей цепью*, если соответственно имеет место

$$x_1 \triangleleft x_2 \triangleleft \dots \triangleleft x_n \triangleleft \dots$$

или

$$x_1 \triangleright x_2 \triangleright \dots \triangleright x_n \triangleright \dots$$

(иногда слова «возрастающая» или «убывающая» мы будем опускать).

Множество, состоящее из одного элемента, также считается цепью. Число элементов цепи называется ее *длиной*. Если первый элемент конечной возрастающей цепи принадлежит множеству P , а последний — множеству Q , то мы скажем, что эта цепь *соединяет* P с Q . Цепь, соединяющая вход и выход, называется *максимальной*. Конечная последовательность x_1, \dots, x_n различных элементов лабиринта S называется *петлей*, если имеет место

$$x_1 \triangleleft x_2 \triangleleft \dots \triangleleft x_n \triangleleft x_1.$$

Теорема 1. *Если положить $x \leq y$ тогда и только тогда, когда в лабиринте S существует (возрастающая) цепь, соединяющая x и y , то S превращается в квазиупорядоченное множество ([1], стр. 20 — 21). Это множество оказывается частично упорядоченным тогда и только тогда, когда в S нет петель.*

*) Результаты настоящей заметки были доложены на Всесоюзном коллоквиуме по общей алгебре в 1958 г. и сформулированы в соответствующем резюме (см. УМН 13, № 3, 1958, стр. 233 — 234). Название «нервные системы» объясняется тем, что рассматриваемый класс автоматов реализуется, в частности, в нервных системах [4].

Доказательство. Рефлексивность отношения \leq очевидна. Пусть теперь $x \leq y$ и $y \leq z$. Тогда

$$x = x_1 \triangleleft \dots \triangleleft x_m = y \quad (1)$$

и

$$y = y_1 \triangleleft \dots \triangleleft y_n = z. \quad (2)$$

Обозначим через k наименьший номер, для которого при некотором l имеет место $x_k = y_l$. Тогда последовательность $x = x_1, \dots, x_k = y_l, \dots, y_n = z$ оказывается цепью, что доказывает транзитивность отношения \leq . Если в S имеются петли, то полученное квазиупорядоченное множество не является частично упорядоченным. Если же в S петель нет и имеет место $x \neq y$, $x \leq y$ и $y \leq x$, то должны выполняться соотношения, полученные из (1) и (2) заменой z на x . Рассуждая, как выше, построим петлю, а это противоречит условию.

Если два лабиринта в естественном смысле изоморфны, то, очевидно, окажутся изоморфными и соответствующие им квазиупорядоченные множества. Обратное, вообще говоря, неверно. Действительно, лабиринты $S = \{x, y, z\}$, где $x \triangleleft y$, $y \triangleleft z$ и $x \triangleleft z$, и $T = \{x, y, z\}$, где $x \triangleleft y$ и $y \triangleleft z$, очевидно, не изоморфны, хотя им обоим соответствует одно и то же квазиупорядоченное множество (цепь из трех элементов).

§ 2. Нервные системы

Пусть имеется лабиринт S , не содержащий бесконечных убывающих цепей, и произвольное множество A , в котором выделен элемент 0 . Допустим, что каждому элементу $s \in S$ поставлена в соответствие k -местная операция f_s над A , где k — число элементов в множестве $\Pi(s)^*$ (см. [1], стр. 7—9), аргументы которой поставлены во взаимно однозначное соответствие элементам множества $\Pi(s)$, а 0 является идемпотентом этой операции**). Получившуюся систему операций обозначим через F . Объединение лабиринта S , множества A и системы операций F , связанных указанным выше образом, назовем *нервной системой* (S, A, F) ***). Для удобства обозначений аргументы операций f_s часто будут писаться в виде последовательности: $f_s(x_1, \dots, x_a, \dots)$.

Пусть задана нервная система (S, A, F) . Функция $\Phi(s)$, отображающая множество S в A , называется *состоянием нервной системы*. Аналогично определяется состояние подмножества P множества S . Если всем элементам из P поставлен в соответствие нуль, то будем говорить, что подмножество P находится в покое. Если натуральному чи-

*) Вообще говоря, нет нужды считать это число конечным. Однако во всех известных приложениях множество $\Pi(s)$ оказывается конечным; так что при желании читатель может считать, что все операции конечно-местны. Это не вносит никаких изменений в изложение. См. также § 6.

**) То есть результат операции f_s равен нулю, если равны нулю все ее аргументы.

***) Если предположить, что A состоит из двух элементов, соответствующих возбужденному и невозбужденному состоянию нейрона, все операции конечно-местны и определяются так, как это описано на стр. 18—19 в [2], то (S, A, F) можно интерпретировать как нервную сеть. Однако нервными сетями не исчерпываются нервные системы с конечно-местными операциями над множеством из двух элементов. Например, операция

$$f(x_1, x_2, x_3) = \begin{cases} 1, & \text{если } x_1 + x_2 - x_3 \geq 1, \\ 0 & \text{в противном случае} \end{cases}$$

не может быть получена способом, указанным Клини.

слу t соответствует состояние $\Phi_t(s)$ (мы будем говорить: нервная система в момент t находится в состоянии $\Phi_t(s)$), то состояние $\Phi_{t+1}(s)$ в момент $t+1$ определяется следующим образом:

$$\Phi_{t+1}(s) = \begin{cases} f_s(\mathfrak{B}), & \text{где } \mathfrak{B} \text{ — состояние множества } \Pi(s) \\ & \text{в момент } t, \text{ если } s \notin U(S)^*; \\ & \text{специально задается, если } s \in U(S)^{**}. \end{cases}$$

Условимся, что в момент 0 нервная система всегда находится в покое. Состояние \mathfrak{B} подмножества P называется *допустимым*, если существуют такие состояния входа $\mathfrak{U}_1, \dots, \mathfrak{U}_t$ в моменты 1, 2, ..., t , что в момент t множество P находится в состоянии \mathfrak{B} .

Нервные системы (S, A, F) и (T, A, G) называются *изоморфными* (слабо изоморфными), если:

И1. Существует изоморфное отображение σ лабиринта S на лабиринт T .

Очевидно, что при этом $\sigma(\Pi(s)) = \Pi(\sigma(s))$.

И2. Соответствие между местами операций $f_s \in F$ и $g_{\sigma(s)} \in G$ индуцируется отображением σ^{***} .

И3 (И3'). Как для всякого (допустимого) состояния \mathfrak{B} множества $\Pi(s)$, так и для всякого (допустимого) состояния \mathfrak{B} множества $\Pi(\sigma(s))$ имеет место $f_s(\mathfrak{B}) = g_{\sigma(s)}(\mathfrak{B})$.

З а м е ч а н и е. Для того чтобы состояние \mathfrak{B} было допустимым для подмножества P , необходимо и достаточно, чтобы это состояние было допустимым для подмножества $\sigma(P)$. Для доказательства достаточно провести индукцию по величине момента t , в который P находится в состоянии \mathfrak{B} .

Если для нервной системы (S, A, F) заданы состояния $\mathfrak{U}_1, \dots, \mathfrak{U}_t, \dots$ ее входа в моменты 1, ..., t, \dots , то этим однозначно определяются состояния $\mathfrak{W}_1, \dots, \mathfrak{W}_t, \dots$ выхода этой нервной системы в моменты 1, ..., t, \dots . Совокупность $\mathfrak{W}_1, \dots, \mathfrak{W}_t, \dots$ называется *финальной реакцией* нервной системы на систему раздражений $\mathfrak{U}_1, \dots, \mathfrak{U}_t, \dots$.

Нервные системы (S, A, F) и (T, A, G) называются *эквивалентными*, если:

Э1. Как входы, так и выходы лабиринтов S и T совпадают.

Э2. Финальные реакции нервных систем (S, A, F) и (T, A, G) на любую систему раздражений совпадают.

Будем говорить, что в нервной системе (S, A, F) элемент x *покрывает y фиктивно*, если $x \triangleright y$, но для всякого допустимого состояния \mathfrak{B} множества $\Pi(x)$, для которого состояние \mathfrak{B}' множества $\Pi(y)$, полученное из \mathfrak{B} заменой нулем состояния элемента y , также является допустимым, имеет место $f_x(\mathfrak{B}) = f_y(\mathfrak{B}')$.

Пусть u — входной элемент нервной системы (S, A, F) , а $\mathfrak{U}_1, \dots, \mathfrak{U}_t, \dots$ — состояния входа этой системы в моменты 1, 2, ..., t, \dots . Обозначим через \mathfrak{U}'_i состояние входа, отличающееся от \mathfrak{U}_i только тем, что элемент u находится в покое. Если при всяком выборе системы $\mathfrak{U}_1, \dots, \mathfrak{U}_t, \dots$ состояний входа финальные реакции на системы раздражений $\mathfrak{U}_1, \dots, \mathfrak{U}_t, \dots$ и $\mathfrak{U}'_1, \dots, \mathfrak{U}'_t, \dots$ совпадают, то элемент u назовем *фиктивным входным элементом*. Выходной элемент,

*) Если $\Pi(s) = \{x_1, \dots, x_k, \dots\}$, то $\Phi_{t+1}(s) = f_s(\Phi_t(x_1), \Phi_t(x_2), \dots, \Phi_t(x_k), \dots)$.

**) Если задание отсутствует, то считается, что соответствующий элемент находится в покое.

***) Если $\Pi(s) = \{x_1, \dots, x_k, \dots\}$ и $f_s = f_s(x_1, \dots, x_k, \dots)$, то

$$g_{\sigma(s)} = g_{\sigma(s)}(\sigma(x_1), \dots, \sigma(x_k), \dots).$$

остающийся в покое при любой системе состояний входа в моменты $1, 2, \dots, t, \dots$, назовем *фиктивным выходным элементом*.

Ясно, что фиктивные входные элементы одной из эквивалентных нервных систем остаются таковыми и в другой. То же самое, очевидно, справедливо и для фиктивных выходных элементов.

§ 3. Представление событий в нервных системах

Пусть (x_{ij}) — $(m \times n)$ -матрица, элементами которой служат символы произвольной природы, а A — некоторое множество.

Событием длины 0 назовем замещение некоторого элемента x_{ij} этой матрицы каким-нибудь элементом из множества A . Такое событие будем записывать в виде формулы $x_{ij} = a_{ij}$, где $a_{ij} \in A$. События $x_{ij} = a_{ij}$ и $x_{ij} = b_{ij}$ при $a_{ij} \neq b_{ij}$, равно как и события $x_{i_1 j_1} = a_{i_1 j_1}$ и $x_{i_2 j_2} = a_{i_2 j_2}$ при $x_{i_1 j_1} \neq x_{i_2 j_2}$, считаются различными.

Формулы $E_1 \& E_2 \& \dots \& E_k$ или $E_1 \vee E_2 \vee \dots \vee E_k$, где $k \geq 1$, а E_i — события длины, меньшей чем l , назовем событиями длины l , если они не являются событиями меньшей длины. Определенные таким образом события назовем $m \times n$ -событиями над A . Событие E называется *выполнимым*, если оно не является тождественно ложным.

Скажем, что $(m \times n)$ -событие E над A представлено в нервной системе (S, A, F) равенством $\Phi_{m+t}(s) = a$, если:

П1. Вход лабиринта S состоит из n элементов (обозначим их через u_1, \dots, u_n).

П2. Существует такой момент $m+t$, что s в момент $m+t$ находится в состоянии a тогда и только тогда, когда истинна формула, получаемая из события E заменой равенства $x_{ij} = a_{ij}$ высказыванием: «в момент i элемент u_i находится в состоянии a_{ij} ».

Число t , указанное в П2, назовем *определяющим моментом*.

$(m \times n)$ -событие E представимо в (S, A, F) с определяющим моментом t , если существуют такие $s \in S$ и $a \in A$, что E представимо равенством $\Phi_{m+t}(s) = a$. Представимость, например, (3×5) -события $(x_{11} = \alpha) \& (x_{12} = \beta) \& (x_{23} = \gamma) \& (x_{34} = \delta)$ равенством $\Phi_{3+t}(s) = a$ означает, что элемент s находится в момент $3+t$ в состоянии a тогда и только тогда, когда в первый момент вход находился в состоянии $(\alpha, \beta, 0, 0, 0)$, во второй — в состоянии $(0, 0, \gamma, 0, 0)$, в третий — в состоянии $(0, 0, 0, \delta, 0)$.

Конечно-местная операция $f(x_1, \dots, x_m)$ над A называется *конъюнктивной относительно* a_1, \dots, a_m , если $f(x_1, \dots, x_m) = f(a_1, \dots, a_m)$ тогда и только тогда, когда $x_i = a_i$ для всех $i = 1, \dots, m$.

Конечно-местная операция $f(x_1, \dots, x_m)$ над A называется *дизъюнктивной относительно* a_1, \dots, a_m , если существует такое $a \in A$, что $f(x_1, \dots, x_m) = a$ тогда и только тогда, когда $x_i = a_i$ хотя бы для одного $i = 1, \dots, m$.

Теорема 2. Если в универсальной алгебре (A, F) зафиксирован идемпотентный элемент 0 и для каждого натурального h и для каждой последовательности a_1, \dots, a_h элементов из A в F существует как конъюнктивная, так и дизъюнктивная относительно a_1, \dots, a_h h -местная операция, то всякое выполнимое $(m \times n)$ -событие над A представимо в подходящей нервной системе (S, A, G) с n входными элементами, где $G \subseteq F$. При этом определяющий момент может быть сделан равным любому числу, равному длине события E или превышающему ее, лабиринт S не содержит петель, а все его s -основания конечны*).

*) Так как из рассуждений, приведенных Кливи ([2], стр. 26), вытекает, что построенная им функциональная система удовлетворяет условиям теоремы 2, то эта теорема является обобщением теоремы 1 на стр. 25 работы [2].

Сначала докажем вспомогательное утверждение:

а) Если $(m \times n)$ -событие E представимо в нервной системе (S, A, G) без петель, все s -основания которой конечны, а $G \subseteq F$, с определяющим моментом t , то оно представимо в нервной системе (S', A, G') , обладающей теми же самыми свойствами, с определяющим моментом $t+k$, где k — произвольное натуральное число.

Действительно, пусть E представимо в (S, A, G) равенством $\Phi_{m+t}(s) = a$. Построим лабиринт S' , присоединяя к лабиринту S цепь $s = s_0 \triangleleft s_1 \triangleleft \dots \triangleleft s_k$. Далее найдем в F одноместные операции f_i , $i = 1, \dots, k$, конъюнктивные относительно a_{i-1} , где $a_0 = a$, $a_r = f_r(a_{r-1})$, $r = 1, \dots, k$. Положим $f_{s_i} = f_i$ и $G' = G \cup \{f_{s_1}, \dots, f_{s_k}\}$. Легко видеть, что событие E представляется в (S', A, G') равенством $\Phi_{m+t+k}(s_k) = a_k$.

Доказательство теоремы 2 будем вести индукцией по длине события E . Если эта длина равна нулю, т. е. $E \equiv (x_{ij} = a_{ij})$, то в качестве S возьмем множество $\{u_1, \dots, u_n, s_1, \dots, s_{m-i+k}\}$, где $u_j = s_0 \triangleleft s_1 \triangleleft \dots \triangleleft s_{m-i+k}$, а k — произвольное целое неотрицательное число (рис. 1). Если в качестве f_{s_i} взять одноместные операции из F , конъюнктивные относительно a_i , где $a_0 = a_{ij}$, а $a_r = f_{s_r}(a_{r-1})$, $r = 1, \dots$

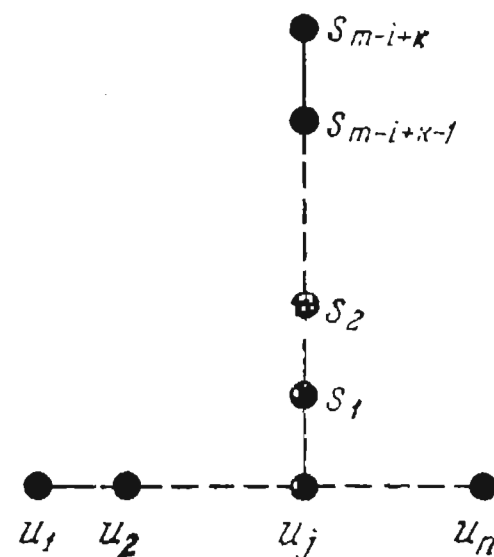


Рис. 1.

$\dots, m-i+k$, нетрудно убедиться, что событие E представимо в нервной системе $(S, A, \bigcup_{r=1}^{m-i+k} f_{s_r})$

равенством $\Phi_{m+k}(s_{m-i+k}) = a_{m-i+k}$ (напомним, что входные элементы, о состоянии которых в данный момент ничего не сказано, считаются находящимися в покое).

Переходя к рассмотрению общего случая, опишем следующую конструкцию. Пусть нервные системы (S_α, A, G_α) имеют один и тот же вход. Превратим $S = \bigcup S_\alpha$ в лабиринт, полагая, что $x \triangleleft y$ имеет место в S тогда и только тогда, когда $x, y \in S_\alpha$ для некоторого α , и $x \triangleleft y$ в S_α . Если положить $G = \bigcup G_\alpha$, то естественным образом определяется нервная система (S, A, G) , называемая объединением нервных систем (S_α, A, G_α) над общим входом. Легко понять, что объединение над общим входом нервных систем, в каждой из которых нет петель, а все s -основания конечны, приводит к нервной системе с теми же свойствами.

Допустим теперь, что длина события E равна l и что теорема доказана для всех событий длины $l-1$. Тогда $E = E_1 \& \dots \& E_k$ или $E = E_1 \vee \dots \vee E_k$ и каждое E_i представляется в некоторой нервной системе (S_i, A, G_i) равенством $\Phi_{m+l-1}(s_i) = a_i$. Можно считать, что все эти нервные системы имеют один и тот же вход. Пусть (S', A, G') — объединение нервных систем (S_i, A, G_i) над общим входом. Присоединив к лабиринту S' элемент s и полагая $s \triangleright s_i$ для всех i , получим лабиринт S . В качестве f_s возьмем k -местную операцию из F , конъюнктивную или дизъюнктивную относительно a_1, \dots, a_k в зависимости от того, знаком $\&$ или \vee соединены E_i . Присоединив к G' эту операцию, получим систему операций G , лежащую в F . Очевидно, что E представляется в нервной системе (S, A, G) равенством $\Phi_{m+l}(s) = a$, где $a = f_s(a_1, \dots, a_k)$. Для завершения доказательства остается принять во внимание предложение а).

Покажем, что существует универсальная алгебра, удовлетворяющая условиям теоремы 2 и содержащая бесконечное множество элементов.

Пусть $\mathfrak{M} = \{a_1, \dots, a_m, \dots\}$ — счетное множество элементов, упорядоченных по их индексам. Обозначим через A' множество ассоциативных

слов над \mathfrak{M} и упорядочим это множество лексикографически. Пусть, далее, \mathfrak{A} — ассоциативная свободная алгебра над каким-либо полем P с системой свободных образующих $\{a_i\}$ (см., например, [3], стр. 21). Если $x \in \mathfrak{A}$, то обозначим через $H(x)$ наибольшее среди самых длинных слов, входящих в запись элемента x с ненулевым коэффициентом.

Для превращения множества $A = A' \cup 0$ в универсальную алгебру зададим на A два типа операций: $f(x_1, \dots, x_n)$ и $\varphi(x_1, \dots, x_n)$, $n = 1, 2, \dots$. Пусть I — некоторая фиксированная система подмножеств множества $\{1, 2, \dots, n\}$, $b_i \in \mathfrak{M}$, $i = 1, 2, \dots, n-1$, $a_j \in A$, $j = 1, 2, \dots, n$. Тогда положим

$$\varphi(x_1, \dots, x_n) = H((x_1 - a_1)b_1(x_2 - a_2)b_2 \dots b_{n-1}(x_n - b_n) + (-1)^{n+1}a_1b_1a_2 \dots b_{n-1}a_n), \quad (3)$$

$$f(x_1, \dots, x_n) = H\left(\sum_{v \in I} v(x_1b_1x_2b_2 \dots b_{n-1}x_n)\right), \quad (4)$$

где запись $v(x_1b_1x_2b_2 \dots b_{n-1}x_n)$ означает, что из произведения $x_1b_1x_2b_2 \dots b_{n-1}x_n$ следует исключить все x_i , для которых $i \in v$.

Выбирая различным образом n , I , $\{a_j\}$ и $\{b_i\}$, получим всю совокупность операций.

Пусть $a_1, \dots, a_n \in A$. Выберем b_1, \dots, b_{n-1} так, чтобы ни один из этих элементов не входил ни в какое из слов a_i . При $a = a_1b_1a_2 \dots b_{n-1}a_n$ формула (3) задает операцию, дизъюнктивную относительно a_1, \dots, a_n . Чтобы получить операцию, конъюнктивную относительно a_1, \dots, a_n , обозначим через Ξ подмножество, составленное из таких индексов i , что $a_i = 0$. Пусть I — система всех подмножеств множества Ξ . Тогда искомая операция определяется формулой (4). Заметим, что $f(a_1, \dots, a_n) = \Xi(a_1b_1a_2b_2 \dots b_{n-1}a_n)$.

§ 4. Канонические нервные системы

Нервная система (S, A, F) называется *канонической*, если:

К1. Для всякого внутреннего элемента x существует в точности один элемент y , покрываемый элементом x и $f_x(a) = a$ для всех $a \in A$.

К2. Для каждого $s \in S$ существует не более одного элемента x , который покрывает s и не принадлежит $W(S)$.

К3. Если $w \in W(S)$ и $w \triangleright v$, то w не может покрывать v фиктивно.

Лабиринт канонической нервной системы изображен на рис. 2. Вход этого лабиринта состоит из элементов u_1, u_2, u_3 , а выход — из элементов w_1, w_2, w_3, w_4 .

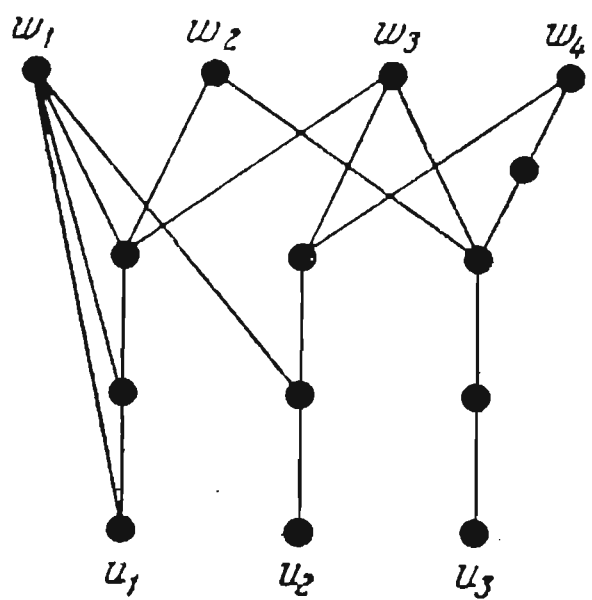


Рис. 2.

Теорема 3. Если Z — петля канонической нервной системы и $z \in Z$, то не существует ни цепи, соединяющей вход с элементом z , ни цепи, соединяющей элемент z с выходом.

Доказательство. Из **К1** вытекает справедливость первого утверждения теоремы. Следовательно, все элементы из Z находятся в покое при любой системе состояний входа. Отсюда ввиду **К2** и **К3** следует справедливость второго утверждения теоремы.

Теорема 4. Если канонические нервные системы без петель и без фиктивных входных элементов (S, A, F) и (T, A, G) эквивалентны, то они слабо изоморфны. Если, кроме того, для каждого $w \in W(S)$ длины цепей, соединяющие вход с w , ограничены в совокупности, то (S, A, F) и (T, A, G) изоморфны.

Для доказательства условимся обозначать через $C_n^S(u, w)$ цепь длины n , соединяющую $u \in U(S)$ с $w \in W(S)$ в лабиринте S . Аналогичный смысл имеет обозначение $C_n^T(u, w)$.

а) Если $m \geq n > 1$, то первые $n-1$ элементов цепей $C_m^S(u, w)$ и $C_n^S(u, w')$ совпадают.

Пусть $C_m^S(u, w) = \{u = a_1 \triangleleft a_2 \triangleleft \dots \triangleleft a_{m-1} \triangleleft a_m = w\}$ и $C_n^S(u, w) = \{u = a'_1 \triangleleft a'_2 \triangleleft \dots \triangleleft a'_{n-1} \triangleleft a'_n = w\}$. Если наименьший номер k , для которого $a_k \neq a'_k$, отличен от n , то мы сразу вступаем в противоречие с К2.

Из К1 вытекает:

б) Если $x \notin W(S)$, то существует в точности один элемент $u(x, S) \in U(S)$, такой, что существует цепь, соединяющая $u(x, S)$ и x . Эта цепь $C(x)$ определяется однозначно. Если длина $L(x)$ цепи $C(x)$ равна l , то элемент x находится в момент $t+l$ в состоянии a в том и только в том случае, когда элемент $u(x, S)$ находится в момент $t+1$ в состоянии a вне зависимости от состояния входа в другие моменты и от состояния множества $U(S) \setminus u(x, S)$ в момент $t+1$.

Из предложения б) и свойства К2 следует:

в) Пусть $P = \{x_\alpha\} \subset S \setminus W(S)$, \mathfrak{B} — такое состояние множества P , при котором x_α находится в состоянии a_α , причем $L(x_\alpha) > t$ влечет $a_\alpha = 0$. Если P' — множество таких элементов x_α , для которых $L(x_\alpha) \leq t$, то множество P находится в состоянии \mathfrak{B} в момент t тогда и только тогда, когда для всех $x_\alpha \in P'$ в момент $t - L(x_\alpha) + 1$ элемент $u(x_\alpha, S)$ находится в состоянии a_α вне зависимости от состояния других элементов входа и от состояния указанных элементов входа в другие моменты.

г) Если существует цепь $C_n^S(u, w)$, то существует и цепь $C_n^T(u, w)$.

Действительно, при $n=1$ это очевидно. Если же $n > 1$, то обозначим через v элемент цепи $C_n^S(u, w)$, покрываемый элементом w . Ввиду К3 существует такое состояние \mathfrak{B} множества $\Pi(w)$, что $a = f_w(\mathfrak{B}) \neq f_w(\mathfrak{B}') = a'$, где состояние \mathfrak{B}' получено из состояния \mathfrak{B} заменой нулем состояния элемента v . Если состояние \mathfrak{B} наступает в момент $t-1$, то это состояние таково, что все $x \in \Pi(w)$, для которых $L(x) > t-1$, находятся в покое. Из предложений б) и в) вытекает, что если состояние \mathfrak{B} наступает в момент $t-1$ при условии, что в моменты $1, 2, \dots, t-1$ вход имеет состояния

$$u_1, u_2, \dots, u_{t-1}, \quad (5)$$

то состояние \mathfrak{B}' наступает в момент $t-1$ при условии, что имеет место система состояний \mathfrak{B} , получаемая из системы состояний (5) заменой состояния u_{t-n+1} состоянием u'_{t-n+1} , при котором элемент $u(v, S)$ находится в покое, а остальные элементы имеют то же состояние, что и в u_{t-n+1} . Если имеет место система состояний (5), то в момент t в нервной системе (S, A, F) элемент w находится в состоянии a . При наличии же системы состояний \mathfrak{B} этот элемент в момент t находится в состоянии a' . Если цепь $C_n^T(u, w)$ отсутствует, то из предложения в) вытекает, что в нервной системе (T, A, G) состояние множества $\Pi(w)$ в момент $t-1$ будет одним и тем же как при наличии системы состояний (5), так и при наличии системы состояний \mathfrak{B} . Таким образом, при наличии этих систем состояний состояние элемента w в момент t будет одним и тем же, что противоречит Э2.

д) Лабиринты S и T изоморфны.

Пусть $x \in S \setminus W(S)$ и $C(x)$ — цепь, определенная в ходе доказательства предложения б). Если эту цепь нельзя вложить в цепь $C_n^S(u(x, S), w)$ для подходящих n и w , то элемент $u(x, S)$ оказывается фиктивным

входным элементом. Поскольку в (S, A, F) таких элементов нет, то цепь $C_n^S(u(x, S), \omega)$, содержащая x , существует. Согласно предложениям а) и г) существует единственная цепь $C_n^T(u(x, S), \omega)$. Обозначим через $\psi(x)$ элемент этой цепи, для которого $L(\psi(x)) = L(x)$. Ясно, что $\psi(S \setminus W(S)) \subseteq T \setminus W(T)$. Из предложения а) следует, что $\psi(x)$ не зависит от выбора n и ω . Положим

$$\varphi(x) = \begin{cases} x, & \text{если } x \in W(S), \\ \psi(x) & \text{в противном случае.} \end{cases}$$

Пусть $x \triangleright y$ в S . Если $x \notin W(S)$ и $L(x) = m$, то ввиду предложения а) первые $m-1$ элементов цепи из S , служащих для определения $\psi(x)$ и $\psi(y)$, совпадают. Значит, совпадают и первые $m-1$ элементов соответствующих цепей в T . Отсюда следует $\varphi(x) = \psi(x) \triangleright \psi(y) = \varphi(y)$. Если $x \in W(S)$ и $L(y) = m$, то ввиду а) $\psi(y) \in C_{m+1}^T(u(y, S), x)$. Так как $L(\psi(y)) = m$, то $\varphi(x) = x \triangleright \psi(y) = \varphi(y)$. Поскольку аналогичным способом можно построить отображение, обратное φ , предложение д) полностью доказано.

е) Отображение φ обладает свойством ИЗ'.

Справедливость этого предложения нуждается в проверке лишь для $\omega \in W(S)$. Если \mathfrak{B} — допустимое состояние множества $\Pi(\omega)$, то, рассуждая, как при доказательстве предложения г), найдем систему состояний входа (5), обеспечивающую наступление состояния \mathfrak{B} множества $\Pi(\omega)$ в момент $t-1$. Из условия К1 и из предложений в) и д) вытекает, что $\varphi(\Pi(\omega))$ имеет в момент $t-1$ состояние \mathfrak{B} . Но тогда из Э2 вытекает равенство $f_w(\mathfrak{B}) = g_w(\mathfrak{B})$, доказывающее предложение е).

Из предложений д) и е) вытекает, что нервные системы (S, A, F) и (T, A, G) слабо изоморфны. Чтобы убедиться в справедливости последнего утверждения теоремы, достаточно установить, что в этом случае для каждого $\omega \in W(S)$ любое состояние множества $\Pi(\omega)$ является допустимым. Но этот факт сразу следует из предложения в). Теорема 4 доказана.

Конструкция 1. Пусть (S, A, F) — нервная система, не содержащая ни фиктивных входных, ни фиктивных выходных элементов. Обозначим через P_0 множество таких элементов x , что всякое покрытие вида $y \triangleright x$ фиктивно. Далее обозначим через P_λ множество таких невыходных элементов x , что из $y \triangleright x$ вытекает или что это покрытие фиктивно, или что $y \in \bigcup_{\mu < \lambda} P_\mu$. Объединение всех P_λ обозначим через P и назовем ниж-

ним множеством. Далее обозначим через Q_0 множество таких элементов x , что всякое покрытие вида $x \triangleright y$ фиктивно. Затем соберем в множество Q_λ все такие невыходные элементы x из S , что из $x \triangleright y$ вытекает или фиктивность этого покрытия, или же включение $y \in \bigcup_{\mu < \lambda} Q_\mu$. Множество

$\bigcup_\lambda Q_\lambda$ обозначим через Q и назовем верхним множеством. Теперь построим лабиринт T , элементами которого служат элементы множества $S \setminus (P \cup Q)$, причем сохраняются все покрытия, имевшиеся в S , не являющиеся фиктивными. На множестве A построим новую систему операций G . Если $x \in T$, то положим

$$g_x(x_1, \dots, x_\alpha, \dots) = f_x(0, \dots, 0, \dots, x_1, \dots, x_\alpha, \dots),$$

где нули стоят на тех местах операции f_x , которые занимали элементы из $P \cup Q$, а также элементы, покрывавшие элемент x фиктивно. Нервную систему (T, A, G) будем считать результатом конструкции 1.

а) Какова бы ни была система состояний входа, верхнее множество находится в покое.

Для Q_0 это вытекает из определения фиктивного покрытия и свойств нуля. Для остальных элементов выводится из тех же соображений с помощью трансфинитной индукции.

б) Если $x \in T$, то операции f_x и g_x дают один и тот же результат, если только общие элементы x -оснований в лабиринтах S и T находятся в одинаковом состоянии.

Если $x \in T$, то в (S, A, F) все элементы из $P \cap \Pi(x)$ покрываются элементом x фиктивно. Поэтому предложение б) вытекает из предложения а), из определения фиктивного покрытия и из определения операции g_x .

в) $U(S) = U(T)$.

Из доказательства б) вытекает, что пересечение $U(S) \cap P$ состоит из фиктивных входных элементов и, следовательно, пусто. Так как пересечение $U(S) \cap Q$ пусто по определению, то $U(S) \subseteq U(T)$. Если $u \in U(T) \setminus U(S)$, то u — такой невыходной элемент лабиринта S , что $u \triangleright x$ влечет или фиктивность этого покрытия, или же включение $x \in P \cup Q$. Если это покрытие не фиктивно, то из $u \notin P$ вытекает $x \notin P$, т. е. $x \in Q$. Значит, $u \in Q$, что не совместно с $u \in T$. !

г) $W(S) = W(T)$.

Если $\omega \in W(S) \cap (P \cup Q)$, то $\omega \in Q$ и из а) вытекает, что ω — фиктивный выходной элемент. Значит, $W(S) \subseteq W(T)$. Если $\omega \in W(T) \setminus W(S)$, то ω — такой невыходной элемент лабиринта S , что $x \triangleright \omega$ влечет или фиктивность этого покрытия, или же включение $x \in P \cup Q$. Если это покрытие не фиктивно, то из $\omega \notin Q$ вытекает $x \notin Q$, т. е. $x \in P$. Значит, $\omega \in P$, что не совместно с $\omega \in T$.

д) Если в моменты $1, 2, \dots, t, \dots$ одинаковые входные элементы нервных систем (S, A, F) и (T, A, G) находятся в одном и том же состоянии, то в любой момент n состояния одинаковых элементов в этих нервных системах одинаковы.

Справедливость предложения д) для $n = 1$ очевидна. Если это предложение справедливо для $n - 1$, то из б) вытекает его справедливость для n .

е) В (T, A, G) нет фиктивных покрытий.

Допустим, что $x \triangleright y$ — фиктивное покрытие в (T, A, G) . Тогда

$$g_x(a, a_1, \dots, a_\alpha, \dots) = g_x(0, a_1, \dots, a_\alpha, \dots) \quad (6)$$

для любой допустимой системы значений $a, a_1, \dots, a_\alpha, \dots$, если только первое место в операции g_x соответствует элементу y . Но ввиду предложения б) из равенства (6) вытекает

$$\begin{aligned} f_x(b_1, \dots, b_\beta, \dots, a, a_1, \dots, a_\alpha, \dots) = \\ = f_x(b_1, \dots, b_\beta, \dots, 0, a_1, \dots, a_\alpha, \dots) \end{aligned}$$

для любой допустимой системы значений $b_1, \dots, b_\beta, \dots, a, a_1, \dots, a_\alpha, \dots \in A$, если только первые места в операции f_x заняты теми же элементами, что и в определении операции g_x . Значит, покрытие $x \triangleright y$ фиктивно в S . Но тогда в T должно быть $x \triangleright y$. Противоречие.

Так как в T не возникает никаких новых покрытий по сравнению с имевшимися в S , то из предложений в), г) и е) следует:

ж) Если (S, A, F) удовлетворяет условиям K1 и K2, то (T, A, G) каноническая.

Из предложений в) — ж) вытекает

Теорема 5. Нервная система (T, A, G) , полученная из (S, A, F) с помощью конструкции 1, эквивалентна (S, A, F) и не содержит фиктивных покрытий. Если (S, A, F) удовлетворяет условиям K1 и K2, то (T, A, G) каноническая.

§ 5. Критерий эквивалентности

Назовем высотой элемента s лабиринта S максимум длин цепей, соединяющих вход с элементом s . Если в лабиринте S нет ни петель, ни бесконечно убывающих цепей, то каждый элемент s имеет определенную конечную или бесконечную высоту. В настоящем параграфе доказывается теорема, которая вместе с теоремой 4 может рассматриваться как критерий эквивалентности нервных систем без петель и без элементов бесконечной высоты. Если лабиринты конечны и есть правило, позволяющее установить совпадение операций, то приводимое доказательство можно рассматривать как алгоритм, решающий проблему эквивалентности.

Теорема 6. Для каждой нервной системы (S, A, F) , не содержащей ни петель, ни фиктивных входных, ни фиктивных выходных элементов, ни элементов бесконечной высоты, существует эквивалентная ей каноническая система без петель.

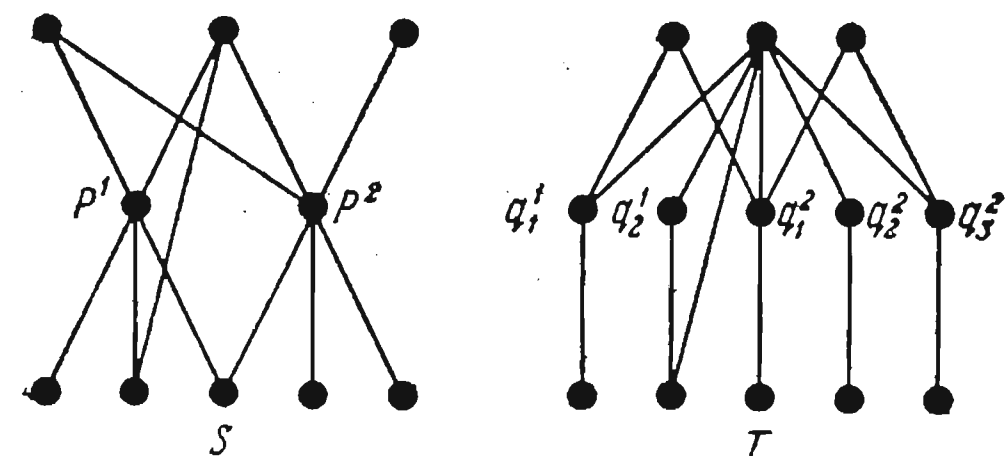


Рис. 3.

Для доказательства опишем сначала следующую конструкцию.

Конструкция 2. Пусть в произвольной нервной системе (S, A, F) выделено множество $P = \{p^\alpha\}$ внутренних элементов, причем $p^\alpha \triangleright p^\beta$ не имеет места ни для каких α и β , а элементу p^α соответствует операция $f_{p^\alpha}(x_1^\alpha, \dots, x_{\lambda_\alpha}^\alpha, \dots)$. Удалим из S множество P и вместо каждого p^α присоединим множество $q_1^\alpha, \dots, q_{\lambda_\alpha}^\alpha, \dots$. Положим $q_\mu^\alpha \triangleright s$ тогда и только тогда, когда s является элементом из $\Pi(p^\alpha)$ с номером μ , а $q_\mu^\alpha \triangleleft s$ тогда и только тогда, когда $p^\alpha \triangleleft s$. Сохранив для остальных элементов покрытия, имевшиеся в S , получим лабиринт, который обозначим через T (рис. 3). Превратим T в нервную систему (T, A, G) , ставя в соответствие всем элементам из $T \cap S$, которые не покрывают какой-либо элемент из P , операции, соответствовавшие им в (S, A, F) . Кроме того, положим

$$g_{q_\mu^\alpha}(x) = x, \quad (7)$$

а для остальных s , принадлежащих T :

$$\begin{aligned} g_s(x_1^1, \dots, x_{\lambda_1}^1, \dots, x_1^2, \dots, x_{\lambda_2}^2, \dots, x_1^\alpha, \dots, x_{\lambda_\alpha}^\alpha, \dots \mid y_1, \dots, y_\beta, \dots) = \\ = f_s(f_{p^1}(x_1^1, \dots, x_{\lambda_1}^1, \dots), f_{p^2}(x_1^2, \dots, x_{\lambda_2}^2, \dots), \dots, \\ \dots, f_{p^\alpha}(x_1^\alpha, \dots, x_{\lambda_\alpha}^\alpha, \dots), \dots \mid y_1, \dots, y_\beta, \dots), \end{aligned} \quad (8)$$

где места до вертикальной черты в операции g_s соответствуют элементам $q_1^1, \dots, q_{\lambda_1}^1, \dots, q_1^2, \dots, q_{\lambda_2}^2, \dots, q_1^\alpha, \dots, q_{\lambda_\alpha}^\alpha, \dots$, а в операции f_s — элементам $p^1, p^2, \dots, p^\alpha, \dots$. Нервную систему (T, A, G) будем считать результатом конструкции 2, примененной к множеству P .

Из способа построения лабиринта T непосредственно следует:

а) Если в S не было петель, то их нет и в T .

Докажем еще одно предложение.

б) Каковы бы ни были системы состояний входа нервных систем (S, A, F) и (T, A, G) (входы этих нервных систем, очевидно, совпа-

дают) в моменты $1, 2, \dots, t, \dots$, состояния одинаковых элементов этих нервных систем в любой момент n одинаковы.

Так как это, очевидно, справедливо для момента 1, то допустим, что предложение б) справедливо для моментов $1, 2, \dots, n-1$. Если $s \in T$ таков, что s -основания в лабиринтах S и T одинаковы, то совпадение состояний элемента S в момент n очевидно. В противном случае этот результат вытекает из формул (7) – (8) и совпадения состояний одинаковых элементов в моменты $n-2$ и $n-1$.

Цепь $s_1 \triangleleft \dots \triangleleft s_m$ назовем *тривиальной*, если $\Pi(s_i) = s_{i-1}$ и $f_{s_i}(x) = x$ для всех $i > 1$.

Теперь обозначим через (S_1, A, F_1) нервную систему (S, A, F) и допустим, что построены нервные системы $(S_1, A, F_1), \dots, (S_{n-1}, A, F_{n-1})$, обладающие следующими свойствами:

1. (S_i, A, F_i) не содержит петель и эквивалентна (S, A, F) .
2. Если внутренний элемент s лабиринта S_i имеет высоту, не превосходящую i , то существует единственная цепь $C(s)$, соединяющая вход с этим элементом; эта цепь $C(s)$ тривиальна.
3. Элементы из S_{i-1} , высота которых меньше, чем i , принадлежат S_i , а операции, соответствующие этим элементам в (S_{i-1}, A, F_{i-1}) и в (S_i, A, F_i) , совпадают.
4. Высоты одинаковых выходных элементов в лабиринтах S и S_i совпадают.

Пусть P — множество внутренних элементов лабиринта (S_{n-1}, A, F_{n-1}) , имеющих высоту n . Ввиду отсутствия петель никакой элемент из P не покрывает сам себя. Невозможность покрытия $x \triangleright y$, где $x, y \in P$, $x \neq y$, вытекает из определения высоты. Обозначим через (S_n, A, F_n) результат применения конструкции 2 к множеству P . Из предложений а) и б) вытекает, что (S_n, A, F_n) обладает свойством 1. Из описания конструкции 2 видно также, что элементы лабиринта S_{n-1} , высота которых меньше n , можно считать лежащими в S_n . При этой операции, соответствующие этим элементам в (S_n, A, F_n) и в (S_{n-1}, A, F_{n-1}) , совпадают. Таким образом, (S_n, A, F_n) обладает свойством 3. Пусть теперь $s \in S_n$ и высота s не превосходит n . Если $s \in S_{n-1}$, то справедливость свойства 2 для этого элемента вытекает из свойства 3 для (S_n, A, F_n) и индуктивного предположения. В противном случае элемент s возник в процессе построения. Значит, s покрывает единственный элемент из S_{n-1} , высота которого не превосходит $n-1$. При этом $f_s(x) = x$. Воспользовавшись индуктивным предположением, убеждаемся в справедливости свойства 2 и в этом случае. Пусть, наконец, выходной элемент w лабиринта S имеет высоту m . Если все цепи из S_n , соединяющие вход с элементом w , лежат в S_{n-1} , то справедливость свойства 4 для элемента w очевидна. Если же в S_n появилась новая цепь C' , то она содержит элемент p' , заменивший некоторый элемент p из P при выполнении конструкции 2. Итак как в S_{n-1} нет петель, то, заменив в цепи C' элемент p' элементом p , мы получим цепь C , соединяющую вход с элементом w в лабиринте S_{n-1} . И так, длины цепей C и C' совпадают, значит справедливость свойства 4 для S_n нами доказана.

Обозначим через S'_i множество элементов лабиринта S_i , высота которых не превосходит i . Положим $T' = \bigcup S'_i$. Превратим T' в лабиринт, полагая $x \triangleright y$ тогда и только тогда, когда x покрывает y в некотором S_i (а значит, ввиду свойства 3 и во всех следующих). Очевидно, $U(T') = U(S)$. Так как высоты всех элементов лабиринта S конечны, то из свойства 4 вытекает $W(T') = W(S)$. Превратим T' в нервную систему (T', A, G') , принимая за g_x операцию, соответ-

ствующую элементу $x \in T'$ в некотором S_i . Свойство 3 обеспечивает корректность определения. Так как всякая петля из T' и всякий элемент w из $W(T')$ располагаются в некотором S_i , то из свойства 1 следует отсутствие в T' петель и эквивалентность нервных систем (T', A, G') и (S, A, F) . Из свойства 2 вытекает, что (T', A, G') обладает свойством K1.

Теперь обозначим через (T_1, A, G_1) нервную систему (T', A, G') и допустим, что построены нервные системы $(T_1, A, G_1), \dots, (T_{n-1}, A, G_{n-1})$, обладающие следующими свойствами:

1". (T_i, A, G_i) не содержит петель и эквивалентна (S, A, F) .

2". (T_i, A, G_i) обладает свойством K1.

3". Для каждого элемента x из T_i , высота которого не превосходит $i-1$, существует не более одного внутреннего элемента, покрывающего x .

4". Элементы из T_{i-1} , высота которых меньше, чем i , принадлежат T_i , а операции, соответствующие этим элементам в (T_i, A, G_i) и (T_{i-1}, A, G_{i-1}) , совпадают.

5". Высоты одинаковых выходных элементов в лабиринтах T' и T_i совпадают.

Каждому элементу z из T_{n-1} , имеющему высоту $n-1$, соответствует множество $P(z)$ покрывающих его внутренних элементов. Каждое такое непустое множество заменим одним элементом \bar{z} . Положим, что всегда справедливо $\bar{z} \triangleright z$, а $y \triangleright \bar{z}$ — если имеет место $y \triangleright p$ для некоторого $p \in P(z)$. Сохранив остальные покрытия, получим лабиринт T_n . Далее, положим

$$g_{\bar{z}}(x) = x$$

и

$$g_y(a_1, \dots, a_\alpha, \dots, a'_1, \dots, a'_\beta, \dots) = \\ = f_y(\underbrace{a_1, \dots, a_1, \dots, \dots}_{\text{если } y \text{ покрывает } a_1}, \dots, \underbrace{a_\alpha, \dots, a_\alpha, \dots}_{\text{если } y \text{ покрывает } a_\alpha}, \dots, a'_\beta, \dots, a'_\beta, \dots),$$

если y покрывает какое-либо z . В операции f_y места, подчеркнутые фигурными скобками, соответствуют элементам, принадлежащим $\bigcup P(z)$. То, что на этих местах стоят одинаковые элементы, вытекает из свойства 2" для (T_{n-1}, A, G_{n-1}) . В операции g_y первые места соответствуют элементам, возникшим при построении лабиринта T_n *). Во всех остальных случаях положим $g_y = f_y$. Так возникает нервная система (T_n, A, G_n) . Рассуждая, как при доказательстве предложений а) и б) в конструкции 2, убедимся, что (T_n, A, G_n) обладает свойством 1". Непосредственно из построения видно, что для (T_n, A, G_n) сохраняются свойства 2", 4" и 5". Свойство 3" нуждается в проверке лишь для элементов, имеющих высоту $n-1$. Но для них справедливость этого свойства обеспечивается самим построением.

Теперь из нервных систем (T_i, A, G_i) , $i = 1, 2, \dots$, построим нервную систему (T, A, G) так же, как нервная система (T', A, G') была построена из нервных систем (S_i, A, F_i) . Эта нервная система (T, A, G) эквивалентна (S, A, F) и обладает свойствами K1 и K2. Применив к (T, A, G) конструкцию 1, получим согласно теореме 5 каноническую нервную систему, эквивалентную (S, A, F) . Теорема 3 позволяет выбросить из нее петли, чем и заканчивается доказательство теоремы 6.

*) Если $y \notin W(T_{n-1})$, то операция f_y , а значит, и g_y , одноместная.

§ 6. Один пример

В заключение построим пример, показывающий целесообразность рассмотрения в теории нервных систем бесконечно-местных операций. Лабиринты, изображенные на рис. 4 и 5, превратим в нервные системы

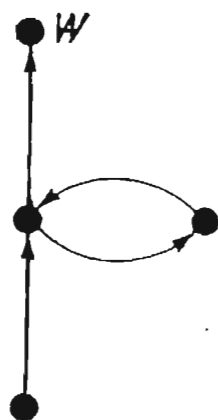


Рис. 4.

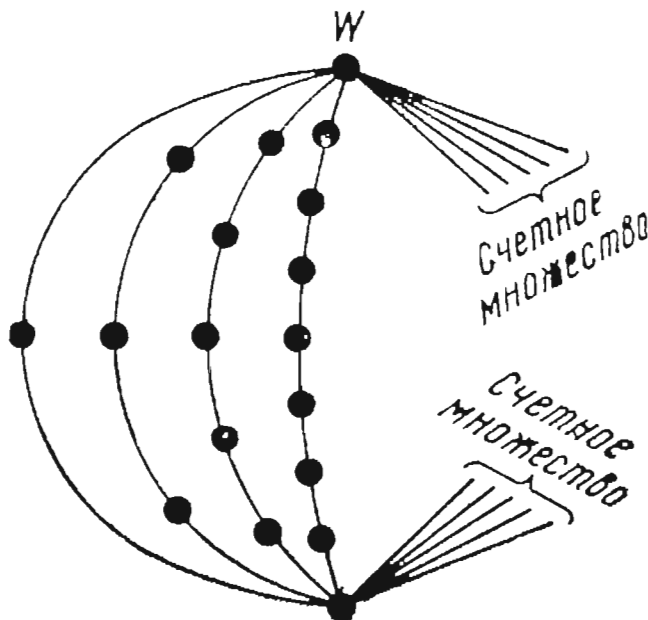


Рис. 5.

считая A состоящим из 0 и 1 и определяя операции следующей формулой:

$$f_s = \begin{cases} 1, & \text{если хотя бы один элемент из } \Pi(s) \text{ равен } 1, \\ 0 & \text{в противном случае.} \end{cases}$$

Эти нервные системы эквивалентны. Первая из них содержит петли. Во второй петель нет, но зато операция f_w бесконечно-местна. Это позволяет надеяться свести некоторые вопросы теории нервных систем с петлями на теорию нервных систем без петель, но с бесконечно-местными операциями.

ЛИТЕРАТУРА

- [1] Биркгоф Г., Теория структур, М., 1952.
- [2] Клини С. К., Представление событий в нервных сетях и конечных автоматах, «Автоматы», сб. статей под редакцией Шеннона К. Э. и Мак-Карти Дж., М., 1956, стр. 15—67.
- [3] Мальцев А. И., Об алгебрах с тождественными определяющими соотношениями, Матем. сборник 26 (1950), стр. 19—33.
- [4] Мак-Каллок У. С., и Питтс У., Логическое исчисление идей, относящихся к нервной активности, «Автоматы», Сб. статей под редакцией Шеннона К. Э. и Мак-Карти Дж., М., 1956, стр. 362—389.

Поступило в редакцию 3 I 1958.

II. ТЕОРИЯ УПРАВЛЯЮЩИХ СИСТЕМ

ЭЛЕКТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ ГОМЕОСТАТИЧЕСКИХ СИСТЕМ *)

Г. В. САВИНОВ

(МОСКВА)

Одной из наиболее общих и интересных задач кибернетики является создание электромеханических систем устойчивого и целенаправленного действия.

В этой связи значительный интерес приобретают системы, названные гомеостатическими [1], или самонастраивающимися [2].

В основе работы этих устройств лежит принцип отбора, по которому из всего многообразия возможных динамических процессов системы поддерживаются (разрешаются) лишь те, которые приводят к наперед заданному, желаемому действию. Принцип отбора предполагает наличие поиска, что обеспечивается дискретно-изменяемыми параметрами системы. Каждому сочетанию параметров системы соответствует свое многообразие динамических процессов, поэтому отбор процессов, приводящих к заданному действию, можно осуществить, прекратив дальнейшее изменение параметров, как только протекание процесса в системе примет необходимый (целенаправленный) характер. Целенаправленность динамического поведения системы формулируется в виде некоторого условия η , задаваемого обычно в форме неравенства. Таким образом, схема работы всего устройства выглядит следующим образом.

В системе непрерывно контролируется выполнение условия η .

В случае невыполнения этого условия обеспечивается изменение параметров. В случае выполнения условия η изменение параметров прекращается. Несмотря на такую простоту схемы работы следует отметить, что те комбинации параметров, которые приводят к выполнению условия η , не могут быть заранее предугаданы конструктором, так же как и не может быть предугадан сам динамический процесс в системе. Из сказанного также ясно, что успешная работа всего устройства в сильной степени зависит от формулировки условия η . Чем полнее это условие характеризует целенаправленность действия, тем успешнее происходит работа самонастраивающегося устройства. Для обеспечения лучшего поиска дискретное изменение параметров обычно происходит по случайному закону. Одно из таких самонастраивающихся устройств было названо Эшби *гомеостатом* [1].

В этом электромеханическом приборе целенаправленность динамических процессов заключалась в обеспечении устойчивости схемы четырех поворачивающихся магнитов, связанных друг с другом системой

*) Работа доложена на семинаре МГУ по кибернетике в апреле 1958 г.

обратных связей. Условие η сводилось к условиям нахождения всех четырех магнитов в средних положениях.

При проектировании подобных механизмов возникает целый ряд вопросов. Оказывается, что даже для устройств с очень простой целенаправленностью огромную роль играет правильный выбор условия η . Быстрота, с которой дискретно изменяются параметры системы, также влияет на работу устройства.

Гибким методом исследования самонастраивающихся устройств является *метод электрического моделирования* [4]. Именно поэтому и была предпринята попытка моделирования таких систем на электронном моделирующем устройстве «Оператор», построенном для кафедры прикладной механики МГУ. Эта электромодель не является серийным устройством, поэтому мы дадим краткое описание ее, необходимое для понимания дальнейшего. «Оператор» представляет собой устройство, позволяющее моделировать динамические системы, описываемые не более чем десятью линейными дифференциальными уравнениями первого порядка вида

$$\dot{x}_i = \sum_{k=1}^{10} a_{ik} x_k, \quad i = 1, 2, \dots, 10. \quad (1)$$

Таким образом, полная матрица системы содержит 100 коэффициентов.

Отдельное устройство для ввода переменных коэффициентов дает возможность моделировать линейные системы, содержащие 34 переменных коэффициента (любые из 100 элементов матрицы $\|a_{ik}\|$), из которых 10 могут задаваться в виде графиков, а 24 вводятся с помощью перфорированной ленты.

Из-за технических соображений лентопротяжные механизмы для ввода 24 переменных коэффициентов объединены в группы по 4 коэффициента, так что все устройство для ввода переменных коэффициентов состоит из 6 отдельных блоков переменных коэффициентов (в дальнейшем обозначаемых БПК), каждый из которых одновременно вводит в электромодель 4 различных коэффициента. Специальные электромагнитные муфты могут включать БПК либо вместе, либо отдельно.

Последнее обстоятельство особенно удобно при моделировании гомеостатических систем.

Индивидуальное включение БПК производилось отдельными реле, включаемыми сигналами управления, формирование которых производилось в отдельном устройстве—блоке команд (в дальнейшем обозначаемом БК).

Моделирование гомеостатической системы осуществлялось следующим образом. На электрической модели моделировалась линейная система в соответствии с уравнением (1). Некоторые из коэффициентов a_{ik} были постоянными. Другая часть коэффициентов могла изменяться в процессе моделирования. Каждый переменный коэффициент изменялся в соответствии с заготовленной заранее программой, которая обеспечивала задание следующих значений коэффициента: $-0,8$; $-0,4$; 0 ; $+0,4$; $+0,8$ (максимальная допустимая величина коэффициента устанавливается равной единице). Последовательность указанных значений в программе выбиралась случайной.

Такой характер изменения коэффициентов устанавливается для всех переменных коэффициентов. Таким образом, на блоках БПК задавалось некоторое случайное сочетание значений для переменной части матрицы системы. Включение лентопротяжных механизмов обеспечивало ввод в систему новой случайной комбинации коэффициентов. Поскольку это

включение лентопротяжных механизмов производилось автоматически сигналами с БК, то система как бы «сама» изменяла переменную часть матрицы коэффициентов.

Вся схема в целом может рассматриваться как некоторая динамическая система с обратными связями, воздействующими на параметры системы (рисунок).

Различное формирование сигналов управления в БК позволяет изучать на электромодели устройства с различным характером целенаправленности. В БК технически проверяется выполнение условия η . Целенаправленность действия изучаемого нами устройства заключалась в обеспечении устойчивости системы вне зависимости от первоначальной матрицы коэффициентов системы. Такой характер работы обуславливался автоматическим отбрасыванием состояний, приводящих к неустойчивым процессам, и сохранением состояний в системе, при которых течение процессов было устойчиво.

Как уже было указано, для осуществления подобной целенаправленности необходимо сформулировать должным образом условие η .

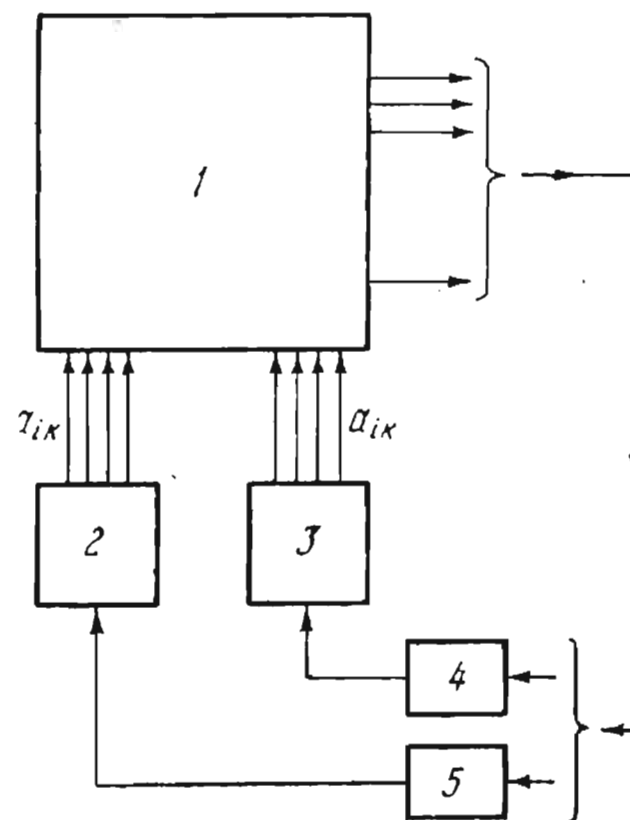
Здесь следует заметить, что динамическая система, описываемая n дифференциальными уравнениями (каждое первого порядка), полностью характеризуется точкой в n -мерном фазовом пространстве. Поэтому для обеспечения устойчивости достаточно окружить точку устойчивого равновесия (в рассматриваемом случае это начало координат) некоторой поверхностью в n -мерном фазовом пространстве и при достижении фазовой точки границы поверхности, что, вообще говоря, может рассматриваться, как факт неустойчивости процесса, производить переключение коэффициентов.

В качестве первой попытки реализации условия η можно образовать функцию $V = \sum_{k=1}^n |x_k|$ и при $V > V_0$ производить переключение коэффициентов.

Однако предварительные опыты показали, что в этом случае можно получить удовлетворительное поведение всей системы только при сильном демпфировании (т. е. в системах с малыми скоростями). Для обеспечения этого условия Эшби помещал магниты гомеостата в масло. Для системы, где скорости изменения координат не малы, более удобна функция вида

$$V = \sum_{k=1}^n (\alpha_k |x_k| + \beta_k |\dot{x}_k|).$$

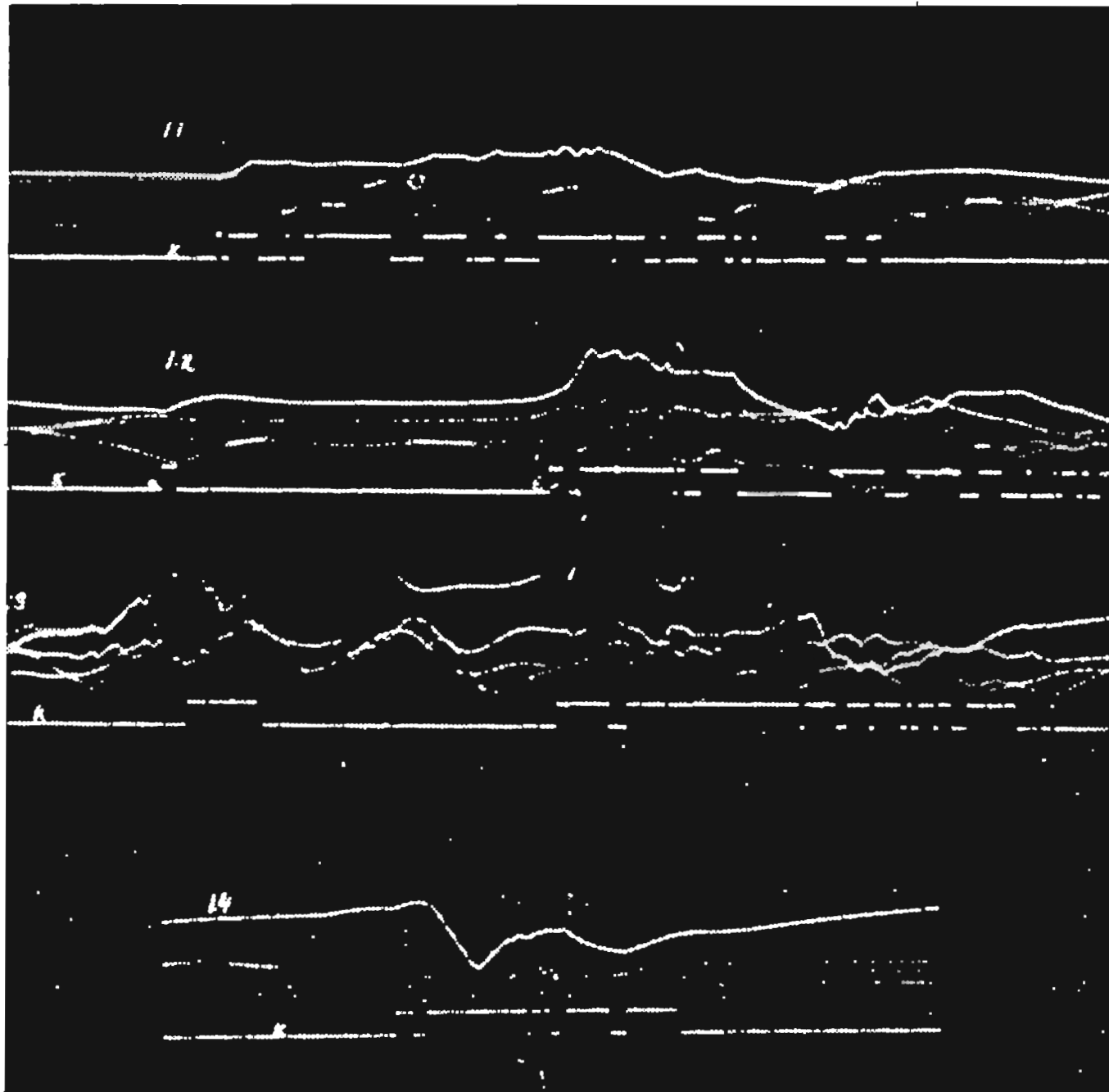
Однако может случиться, что даже и при таком сигнале потребуется слишком много времени для того, чтобы при переключениях коэффициентов (т. е. во время операции поиска) получить комбинацию коэффициентов, приводящую систему к равновесию. В этом случае для ускорения поиска Эшби рекомендует *разбить систему на подсистемы и поиск производить по каждой подсистеме отдельно*. Очевидно, этот метод даст положительный результат, если разбиение будет произведено на подсистемы, слабо взаимодействующие друг с другом. Эшби называет такие системы *приводимыми* [3].



1 — электронная модель-аналог динамической системы, 2, 3 — блоки ввода переменных коэффициентов (БПК), 4, 5 — блоки формирования сигналов команд для включения БПК.

Моделирование гомеостатической системы с разбиением только на две подсистемы подтвердило правильность этого высказывания.

Результаты моделирования системы, описываемой четырьмя уравнениями, при скорости переключения коэффициентов один раз в секунду



представлены на осциллограммах 1.1—1.4. Матрица коэффициентов имела следующий вид:

$$\begin{vmatrix} -0,3 & +0,1 & a_{13} & a_{14} \\ +0,1 & -0,2 & a_{23} & a_{24} \\ a_{31} & a_{32} & -0,15 & +0,1 \\ a_{41} & a_{42} & +0,1 & -0,25 \end{vmatrix}$$

Коэффициенты a_{13} , a_{14} , a_{23} , a_{24} , a_{31} , a_{32} , a_{41} , a_{42} изменялись дискретно. Функция имела вид

$$V = 0,25 (x_2 + x_4) + 0,5 (\dot{x}_2 + \dot{x}_4) *$$

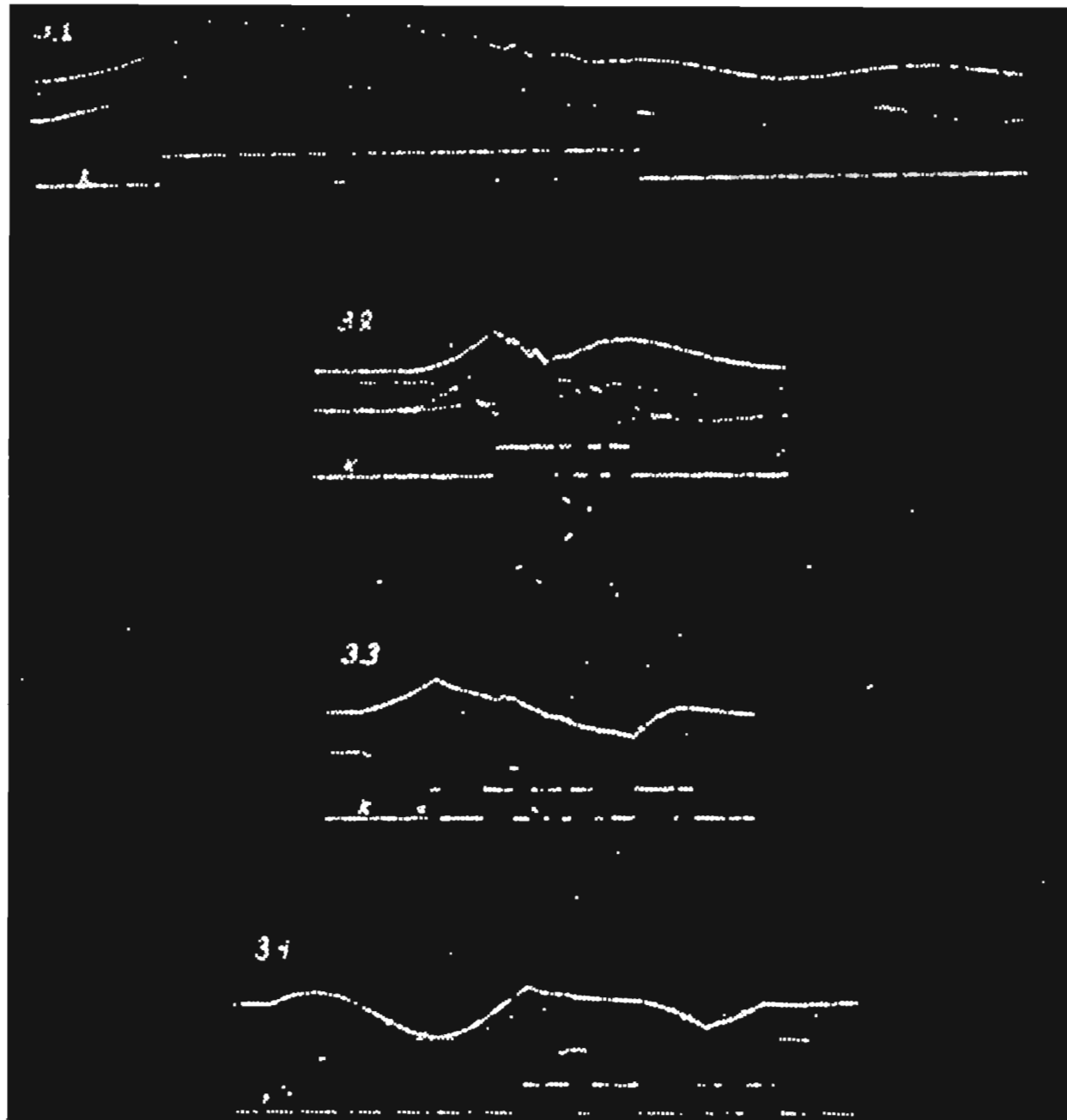
На осциллограммах записаны изменения во времени для всех координат (снизу вверх x_1, x_2, x_3, x_4), а также работа БПК (линия K).

Осциллограмма 1.1 является типичной для работы системы.

Интересно, что в результате поиска устойчивого режима был найден режим, хотя и не устойчивый, но близкий к нему (с медленным нарастанием). При достижении системой контура переключений (точка «а» на осциллограмме 1.2) происходит переключение коэффициентов, приводящее опять к медленно нарастающему процессу, что вызывает новое срабатывание БПК (точка «в» осциллограммы 1.2). После этого следует

*) По техническим причинам нельзя было образовать сумму от модулей величин, что было бы лучше.

целый цикл срабатываний ПБК, заканчивающийся отысканием устойчивого режима. Осциллограмма 1.3 показывает, как реагирует система в случае, когда включение БПК произошло тогда, когда отклонения в системе уже достигли больших значений. Как и следовало ожидать, в этом случае системе трудно справиться и требуется большее время для



нахождения необходимого режима (здесь найден режим с медленно нарастающими процессами).

Осциллограмма 1.4 дает продолжение процесса. Влияние скорости переключения коэффициентов на качество поисков показывают осциллограммы 3.1—3.4. Здесь система та же. Скорость—два значения коэффициента в секунду.

Функция имеет вид

$$V = 0,14(x_2 + x_4) + 1,4(\dot{x}_2 + \dot{x}_4).$$

(Здесь увеличено влияние производных от координат.)

Все осциллограммы свидетельствуют о более быстром отыскании системой устойчивых (или близких к ним) режимов. Кстати, на осциллограмме 3.3 видно немедленное изменение характера процесса при введении новой удачной комбинации коэффициентов (точка «а»).

На последующих осциллограммах представлены результаты моделирования системы седьмого порядка. В этом случае переменными коэффициентами матрицы были: a_{12} , a_{14} , a_{31} , a_{35} , a_{53} , a_{56} , a_{67} , a_{72} . Кроме того, система разбивалась на две подсистемы, которые взаимодействовали друг с другом довольно сильно. Функции имели вид:

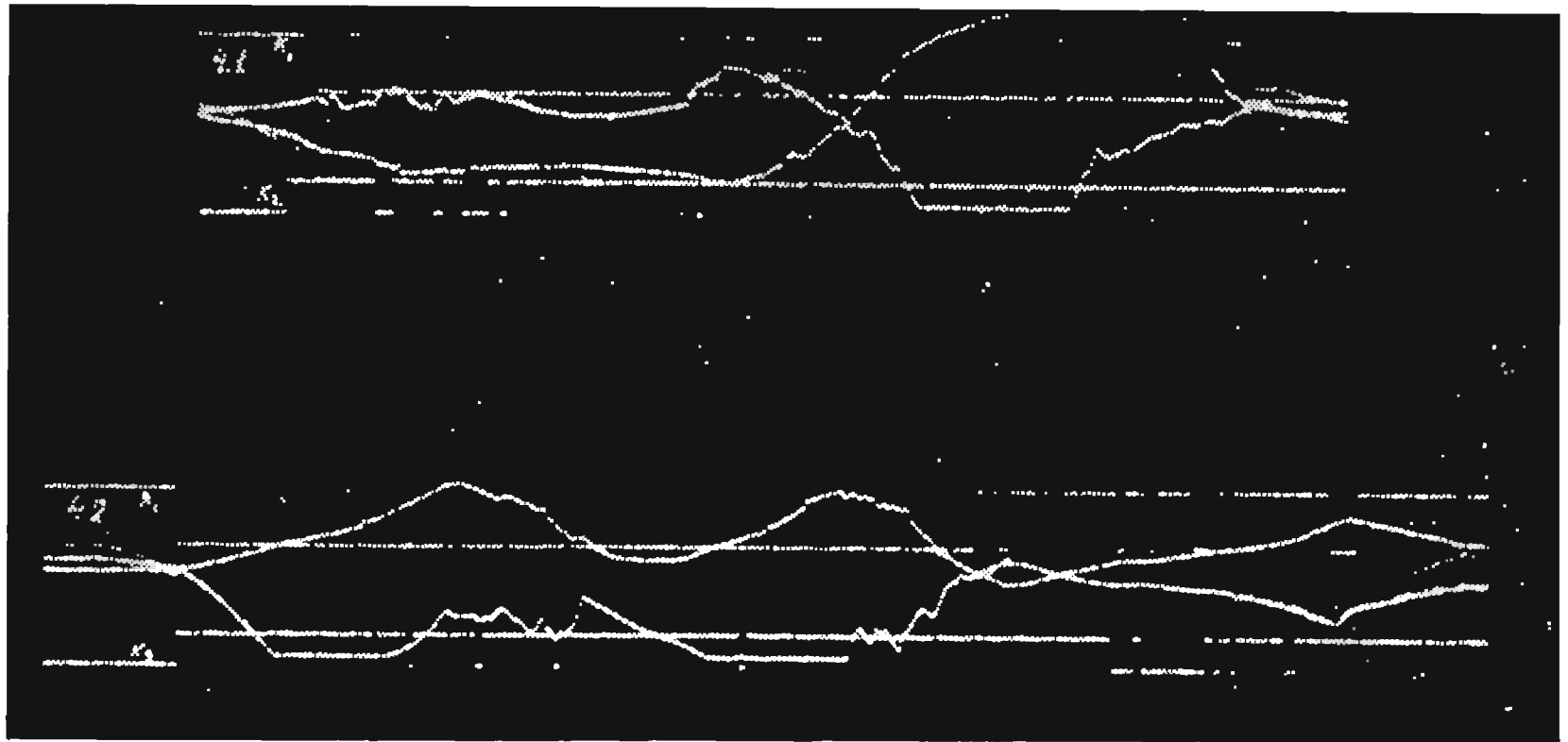
$$V_1 = 0,16(x_1 + x_2 + x_3) + 1,6(\dot{x}_1 + \dot{x}_2 + \dot{x}_3),$$

$$V_2 = 0,16(x_5 + x_6 + x_7) + 1,6(\dot{x}_5 + \dot{x}_6 + \dot{x}_7).$$

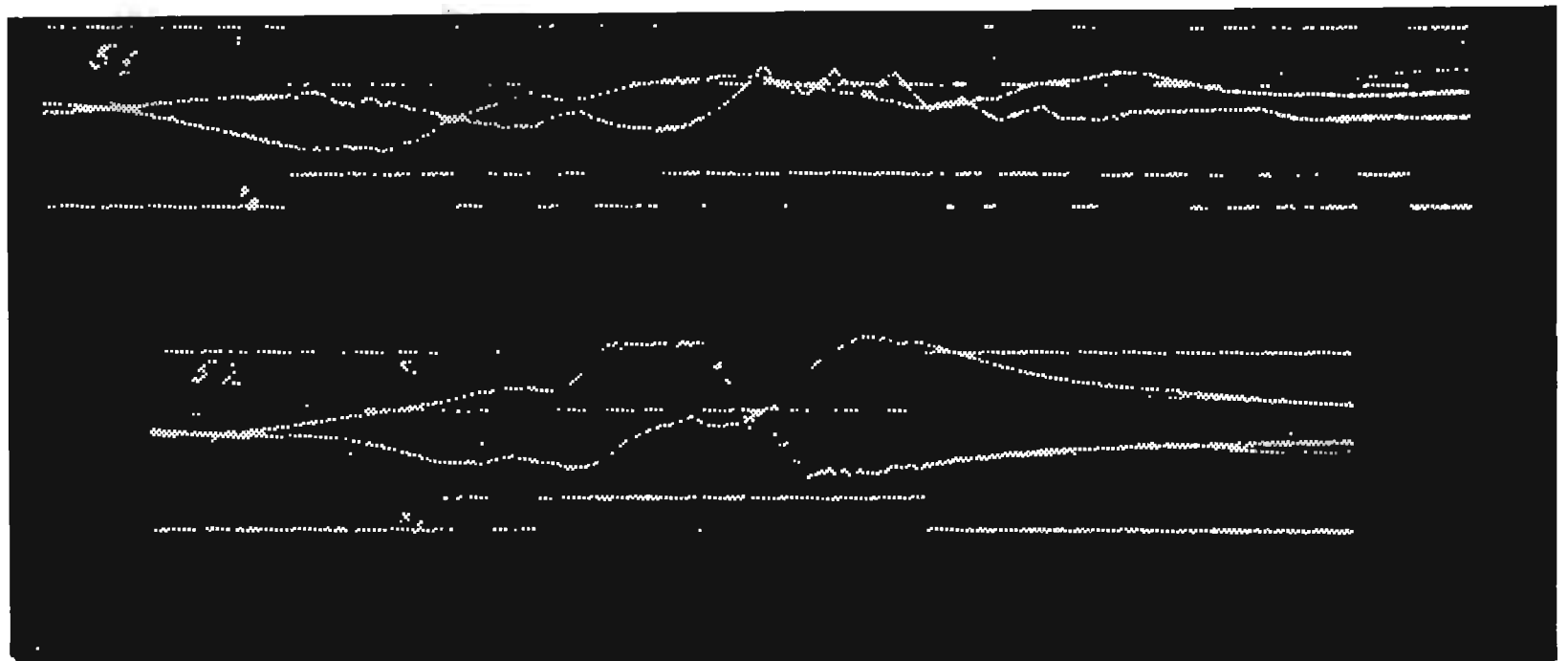
Функция V_1 управляла группой переменных коэффициентов

$$a_{12}, a_{14}, a_{31}, a_{35}.$$

Функция V_2 — группой $a_{53}, a_{56}, a_{67}, a_{72}$. Осциллограммы 4.1 и 4.2 дают представление о ходе процессов (записаны снизу вверх $x_1, x_2, x_3, x_5, x_6, x_7$)



Линия K_2 отражает ход переключения от сигнала V_1 . Линия K_1 отражает ход переключения от сигнала V_2 . Осциллограммы 5.1 и 5.2 дают запись процессов (снизу вверх $x_1, x_2, x_3, x_5, x_6, x_7$) в случае



общего управления, т. е. от одной функции $V = V_1 + V_2$. Сравнение осциллограмм серий 4 и 5 показывает лучший характер отбора в случае системы, не разделенной на подсистемы. Этот результат связан с тем, что системы в случае отдельного управления состояли из подсистем, сильно взаимодействующих друг с другом.

С целью проверки этого положения была промоделирована система с матрицей, содержащей нулевые миноры.

Матрица системы имела вид (коэффициенты a_{ik} являются переменными)

$$\begin{vmatrix} -0,2 & a_{12} & -0,2 & a_{14} & +0,2 & -0,1 & +0,2 \\ +0,2 & -0,4 & -0,5 & +0,1 & +0,1 & -0,2 & +0,1 \\ a_{31} & +0,2 & -0,3 & -0,2 & a_{35} & +0,2 & -0,2 \\ 0 & 0 & 0 & -0,4 & -0,2 & +0,15 & -0,1 \\ 0 & 0 & 0 & a_{54} & -0,3 & a_{56} & -0,2 \\ 0 & 0 & 0 & +0,3 & +0,4 & -0,3 & a_{67} \\ 0 & 0 & 0 & +0,4 & a_{75} & +0,2 & -0,2 \end{vmatrix}.$$

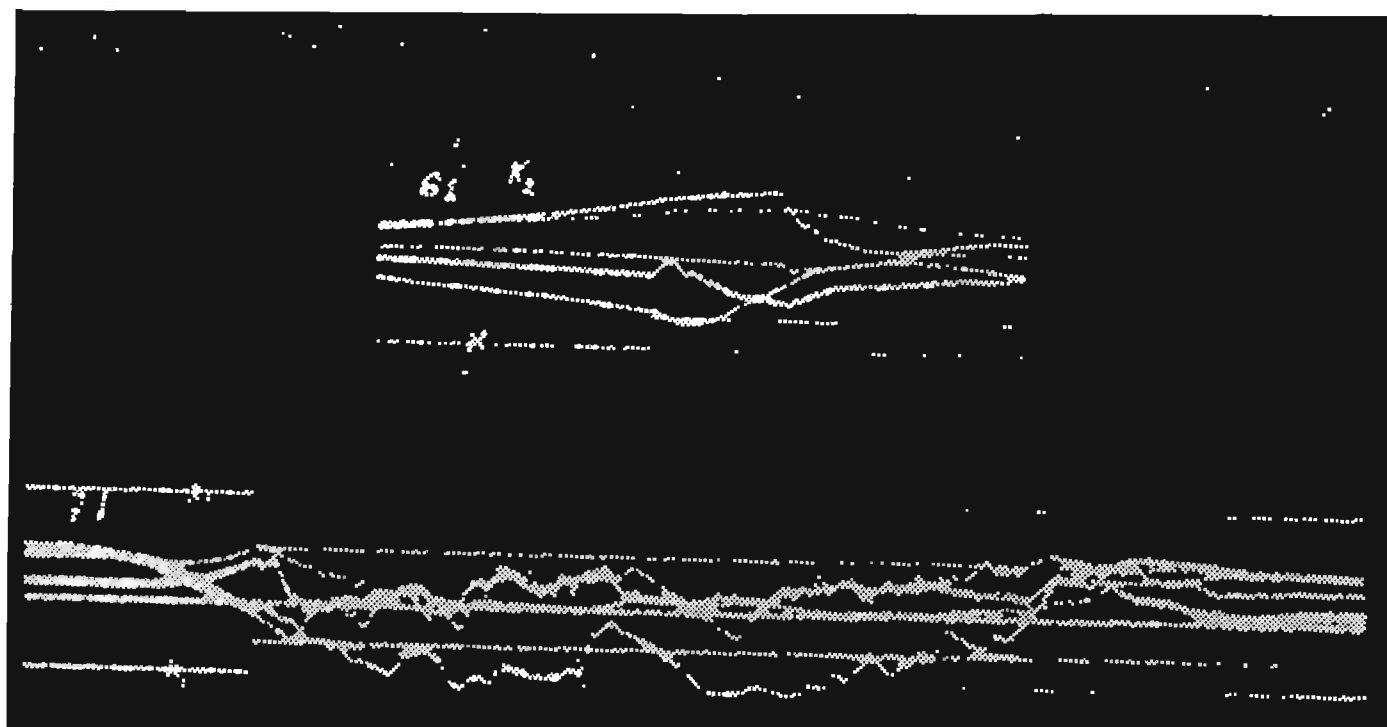
Система разбивалась на две подсистемы. Коэффициенты $a_{12}a_{14}a_{31}a_{35}$ управлялись функцией

$$V_1 = 0,16(x_1 + x_2 + x_3) + 1,6(\dot{x}_1 + \dot{x}_2 + \dot{x}_3).$$

Коэффициенты $a_{54}a_{56}a_{67}a_{75}$ управлялись функцией

$$V_2 = 0,16(x_5 + x_6 + x_7) + 1,6(\dot{x}_5 + \dot{x}_6 + \dot{x}_7).$$

На осциллограмме 6.1 виден характер процесса в этом случае. Осциллограмма 7.1 дает процесс установления для случая, когда все 8 коэффициентов управляются одной функцией $V = V_1 + V_2$.



Как и следовало ожидать, процесс поиска в этом случае длится дольше, т. е. разбиение всей системы на соответствующие (слабо взаимодействующие) подсистемы дает существенное улучшение характера отбора.

Анализ полученных кривых динамического поведения показывает, что принятый закон для командного сигнала все же не совсем удовлетворителен, поскольку процесс поиска происходит достаточно долго и, кроме того, в результате такого поиска допускаются движения с медленным нарастанием координат (случай 1.3; 1.4; 3.4 и др.). Для устранения указанных недостатков проводился эксперимент с системой четвертого порядка, причем в качестве командного сигнала выбиралась величина, пропорциональная производной от длины вектора, проведенного в точку четырехмерного фазового пространства, характеризующую состояние системы

$$V = \frac{d}{dt} \sum_{k=1}^4 x_k^2 = 2 \sum_{k=1}^4 x_k \dot{x}_k.$$

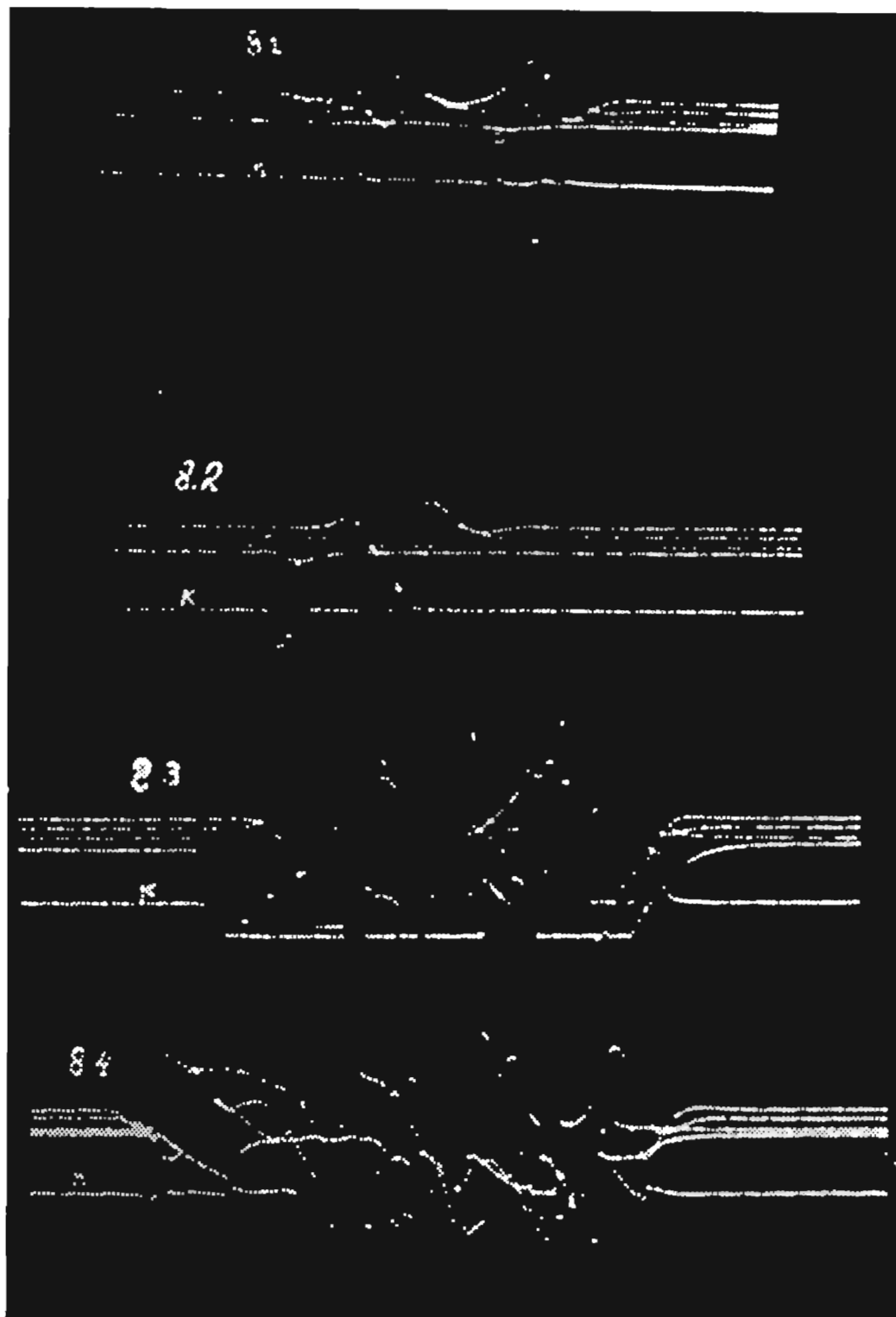
При $V > 0$ (что соответствовало неустойчивости) происходил поиск устойчивой комбинации коэффициентов. При $V < 0$ переключение коэффициентов приостанавливалось.

Поведение системы с таким командным сигналом характеризуется кривыми на осциллограммах 8.1—8.4.

Следует заметить, что знаки диагональных элементов матрицы системы существенным образом влияют на устойчивость, и во всех предыдущих случаях все диагональные коэффициенты выбирались отрицательными.

В случае системы с командным сигналом последнего вида оказалось возможным автоматически находить устойчивые состояния и при положительных диагональных коэффициентах.

Осциллограмма 8.1 дает представление о динамических процессах в системе с отрицательными диагональными коэффициентами (случай соответствует примерно осциллограмме 1.1).



Результат работы системы при 2-х положительных диагональных коэффициентах представлен на осциллограмме 8.2, при 3-х положительных коэффициентах—на осциллограмме 8.3, при 4-х положительных коэффициентах—на осциллограмме 8.4 (при прежнем законе для командного сигнала система не «смогла» бы найти устойчивого режима даже при одном положительном диагональном коэффициенте).

Приведенные в настоящей статье результаты, конечно, далеки от законченности, однако они могут свидетельствовать о возможности исследования гомеостатических систем с помощью электронных моделей. В частности, указанный здесь метод может быть применен в задачах автоматического отыскания коэффициентов систем автоматического управления, обеспечивающих устойчивость всей системы.

В заключение автор выражает благодарность сотрудникам лаборатории колебаний Р. А. Велерштейн и Г. М. Жаркову, а также студентке Р. Павловой, оказавшим помощь при выполнении эксперимента.

ЛИТЕРАТУРА

- [1] Ashby W. Ross., Design for a Brain, 1952.
- [2] Цянь Сюэ-сэнь, Техническая кибернетика, ИЛ, 1956.
- [3] Автоматы, Сб. статей под ред. Шеннона и Мак-Карти, ИЛ, 1956.
- [4] Корн Г. и Корн Т., Электронные моделирующие устройства, ИЛ, 1955.

Поступило в редакцию 21 XI 1958.

К ВОПРОСУ ОБ ОПРЕДЕЛЕНИИ ВЕРОЯТНОСТИ ОТКАЗА В ОДНОЛИНЕЙНЫХ СИСТЕМАХ МАССОВОГО ОБСЛУЖИВАНИЯ СМЕШАННОГО ТИПА

В. И. МУДРОВ

(КАЛИНИН)

Как известно, системы массового обслуживания разделяются на два принципиально различных вида. В системах первого вида поступившие заявки на обслуживание занимают свободные устройства (обслуживаются свободным оператором), а в случае, если все устройства заняты, заявка на обслуживание получает отказ. Такие системы называются *системами с отказами*. Основной характеристикой таких систем является *процент отказов в обслуживании*.

В системах второго вида в случае, если все устройства заняты, поступившая заявка (вызов) становится в очередь, т. е. сохраняется как претендент на обслуживание в дальнейшем. Такие системы называются *системами с ожиданием*. Основной характеристикой таких систем является *среднее время ожидания обслуживания*.

Кроме того, существуют системы обслуживания *смешанного типа*. В таких системах поступивший вызов становится в очередь и ожидает обслуживания. Если же число ожидающих или, в других случаях, если время пребывания в очереди превышает некоторый предел, то этот вновь поступивший вызов получает отказ.

В последнее время решен ряд задач, относящихся к таким системам [3], [4], [5]. В частности, решена задача отыскания процента отказов для следующих случаев:

а) Пуассоновский поток вызовов, однолинейная система, показательное распределение времени обслуживания. Вызов получает отказ, если общее время его обслуживания, включая пребывание в очереди, превышает постоянную величину τ . Обслуживание предполагается как упорядоченное (т. е. из числа ожидающих вызовов выбирается тот, который дольше всех находится в очереди), так и неупорядоченное (т. е. на основе случайного выбора среди ожидающих).

б) Пуассоновский поток вызовов, система с n линиями, показательное распределение времени обслуживания, вызов получает отказ, если время пребывания его в очереди превышает некоторую постоянную величину τ . Обслуживание упорядоченное.

Вызовы такого типа в этих статьях называются «нетерпеливыми клиентами».

в) Пуассоновский поток вызовов, система с n линиями, показательное распределение времени обслуживания. Вызов получает отказ, если в момент его поступления все линии заняты и, кроме того, в очереди имеется m клиентов. Дисциплина выбора клиентов из очереди роли не играет.

В настоящей статье рассматривается аналогичная однолинейная система с постоянным временем обслуживания и простейшим (пуассоновским) входящим потоком вызовов. Вызов получает отказ, если время пребывания его в очереди должно превысить постоянную величину τ . Обслуживание предполагается упорядоченным.

Вероятность отказа p в такой системе можно найти, используя то очевидное обстоятельство, что при большой длительности процесса среднее относительное число необслуженных вызовов сходится по вероятности к вероятности отказа. Так как система не дает немедленного отказа вызовам, то в ней может иметься очередь. При этом, если к моменту окончания обслуживания одного из вызовов есть ожидающие, то обслуживание очередного вызова из числа ожидающих начинается сразу после освобождения оператора; если же в этот момент ожидающих нет, то наступает «простой» оператора без работы до тех пор, пока не подойдет очередной вызов.

Учитывая показательный характер распределения интервалов между последовательными вызовами в простейшем потоке и связанную с этим независимость распределения $f(l)$ времени l ожидания оператором очередного вызова с момента начала этого ожидания

$$f(l) = \lambda e^{-\lambda l}, \quad (1)$$

легко убедиться, что математическое ожидание времени такого «простоя» (если он имеет место) равно

$$t_{\text{прост}} = \frac{1}{\lambda}. \quad (2)$$

Обозначим:

$t_{\text{обсл}}$ — время обслуживания одного вызова,

τ — максимальное время пребывания вызова в очереди.

Начало обслуживания очередного вызова, следующего за периодом простоя, является началом образования группы циклов обслуживания.

Можно найти вероятность p_k образования группы из k циклов ($k=1, 2, 3, \dots$) и среднюю длину такой группы. Каждая такая группа

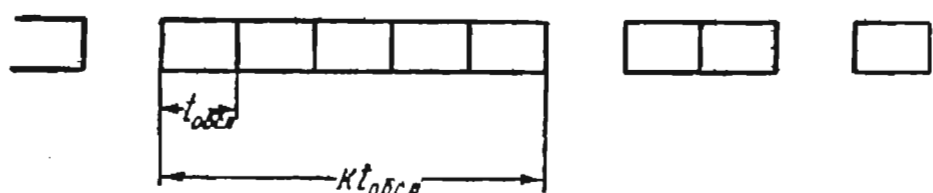


Рис. 1.

некоторой средней длины L будет в среднем сопровождаться простоем длиной $t_{\text{прост}} = 1/\lambda$. Поэтому за время $L + 1/\lambda$ будет обслуживаться в среднем $L/t_{\text{обсл}}$ вызовов.

Среднее число вызовов, обслуживаемых при большой длительности процесса в единицу времени, будет в пределе равно $\frac{L/t_{\text{обсл}}}{L + 1/\lambda} = \frac{1}{t_{\text{обсл}} + (t_{\text{обсл}}/\lambda L)}$ и, наконец, среднее относительное число обслуженных вызовов равно

$$\frac{1}{\left(\lambda + \frac{1}{L}\right) t_{\text{обсл}}}.$$

Учитывая сделанное ранее замечание о сходимости по вероятности среднего относительного числа необслуженных вызовов к вероятности отказа p , можно утверждать, что в установившемся режиме

$$p = 1 - \frac{1}{\left(\lambda + \frac{1}{L}\right) t_{\text{обсл}}}.$$

Не останавливаясь на доказательстве этих, вообще говоря, очевидных положений, перейдем к отысканию величины L .

$$\begin{aligned}
\pi_{k+1, m-1} = \pi_{k, 0} & \left\{ e^{-\lambda v} \frac{(\lambda u)^m}{m!} e^{-\lambda u} + \lambda v e^{-\lambda v} \frac{(\lambda u)^{m-1}}{(m-1)!} e^{-\lambda u} + \dots \right. \\
& \dots + \left[1 - e^{-\lambda v} - \lambda v e^{-\lambda v} - \dots - \frac{(\lambda v)^{m-1}}{(m-1)!} e^{-\lambda v} \right] e^{-\lambda u} \left. \right\} + \dots \\
& \dots + \pi_{k, m-2} \left\{ e^{-\lambda v} \frac{(\lambda u)^2}{2!} e^{-\lambda u} + \lambda v e^{-\lambda v} \lambda u e^{-\lambda u} + \right. \\
& + [1 - e^{-\lambda v} - \lambda v e^{-\lambda v}] e^{-\lambda u} \left. \right\} + \pi_{k, m-1} \{ e^{-\lambda v} \lambda u e^{-\lambda u} + [1 - e^{-\lambda v}] e^{-\lambda u} \} + \pi_{k, m} e^{-\lambda u} \\
\pi_{k+1, m} = \pi_{k, 0} & \left\{ e^{-\lambda v} \left[1 - e^{-\lambda u} - \dots - \frac{(\lambda u)^m}{m!} e^{-\lambda u} \right] + \right. \\
& + \lambda v e^{-\lambda v} \left[1 - e^{-\lambda u} - \dots - \frac{(\lambda u)^{m-1}}{(m-1)!} e^{-\lambda u} \right] + \dots \\
& \dots + \frac{(\lambda v)^{m-1}}{(m-1)!} e^{-\lambda v} [1 - e^{-\lambda u} - \lambda u e^{-\lambda u}] + \\
& + \left[1 - e^{-\lambda v} - \lambda v e^{-\lambda v} - \dots - \frac{(\lambda v)^{m-1}}{(m-1)!} e^{-\lambda v} \right] \times \\
& \times [1 - e^{-\lambda u}] \left. \right\} + \dots + \pi_{k, m-2} \left\{ e^{-\lambda v} \left[1 - e^{-\lambda u} - \lambda u e^{-\lambda u} - \frac{(\lambda u)^2}{2!} e^{-\lambda u} \right] + \right. \\
& + \lambda v e^{-\lambda v} [1 - e^{-\lambda u} - \lambda u e^{-\lambda u}] + [1 - e^{-\lambda v} - \lambda v e^{-\lambda v}] [1 - e^{-\lambda u}] \left. \right\} + \\
& + \pi_{k, m-1} \{ e^{-\lambda v} [1 - e^{-\lambda u} - \lambda u e^{-\lambda u}] + [1 - e^{-\lambda v}] [1 - e^{-\lambda u}] \} + \pi_{k, m} (1 - e^{-\lambda u}).
\end{aligned}$$

Пусть A_{ij} — коэффициент при π_{kj} в формуле для вычисления $\pi_{k+1, i}$, а

$$\alpha = \begin{vmatrix} A_{0,0} & A_{0,1} & 0 & 0 & \dots & 0 \\ A_{1,0} & A_{1,1} & A_{1,2} & 0 & \dots & 0 \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ A_{m-1,0} & A_{m-1,1} & A_{m-1,2} & A_{m-1,3} & \dots & A_{m-1,m} \\ A_{m,0} & A_{m,1} & A_{m,2} & A_{m,3} & \dots & A_{m,m} \end{vmatrix} \quad (7)$$

— соответствующая матрица этих коэффициентов.

Матрица перехода от вероятностей $\pi_{1,i}$ к вероятностям $\pi_{k,i}$ равна $(k-1)$ -й степени матрицы α . Обозначая эту матрицу через β , имеем

$$\beta = \alpha^{k-1}. \quad (8)$$

Вероятности $\pi_{1,i}$ на первом шаге равны

$$\pi_{1,0} = 1, \pi_{1,1} = \pi_{1,2} = \pi_{1,3} = \dots = \pi_{1,m} = 0. \quad (9)$$

Нас интересуют величины $\pi_{k,0}$, с которыми непосредственно связаны вероятности p_k образования групп из k циклов.

Так как

$$\pi_{k,0} = B_{0,0}^{(k)} \pi_{1,0} + B_{0,1}^{(k)} \pi_{1,1} + \dots + B_{0,m}^{(k)} \pi_{1,m}, \quad (10)$$

то, учитывая (9), получаем

$$\pi_{k,0} = B_{0,0}^{(k)}, \quad (11)$$

где $B_{0,0}^{(k)}$ — верхний левый элемент матрицы β_k . После этого легко находится величина

$$L = t_{\text{обсл}} \sum_{k=1}^{\infty} p_k k \quad (12)$$

и искомая вероятность отказа

$$p = 1 - \frac{1}{\lambda t_{\text{обсл}} + \frac{t_{\text{обсл}}}{L}}.$$

Рассмотрим частный случай, когда $\tau < t_{\text{обсл}}$. В этом случае $m = \lfloor \tau/t_{\text{обсл}} \rfloor = 0$. Система уравнений (6) принимает вид

$$\left. \begin{aligned} p_k &= \pi_{k,0} e^{-\lambda \tau}, \\ \pi_{k+1,0} &= (1 - e^{-\lambda \tau}) \pi_{k,0}. \end{aligned} \right\} \quad (13)$$

Матрица α состоит из одного элемента $A_{0,0} = 1 - e^{-\lambda \tau}$, матрица β также состоит из одного элемента

$$B_{0,0}^k = (1 - e^{-\lambda \tau})^{k-1}.$$

Вероятность p_k образования группы из k циклов, следующих без перерыва один за другим, дается формулой

$$p_k = (1 - e^{-\lambda \tau})^{k-1} e^{-\lambda \tau}.$$

Далее,

$$\begin{aligned} L &= t_{\text{обсл}} \sum_{k=1}^{\infty} k (1 - e^{-\lambda \tau})^{k-1} e^{-\lambda \tau} = \\ &= t_{\text{обсл}} \sum_{k=1}^{\infty} \frac{d}{d(\lambda \tau)} (1 - e^{-\lambda \tau})^k = t_{\text{обсл}} \frac{d}{d(\lambda \tau)} \sum_{k=1}^{\infty} (1 - e^{-\lambda \tau})^k = \\ &= t_{\text{обсл}} \frac{d}{d(\lambda \tau)} \frac{1}{1 - (1 - e^{-\lambda \tau})} = t_{\text{обсл}} \frac{d}{d(\lambda \tau)} e^{\lambda \tau} = t_{\text{обсл}} e^{\lambda \tau}. \end{aligned}$$

Поэтому вероятность отказа в такой системе можно найти по формуле

$$p = 1 - \frac{1}{\lambda t_{\text{обсл}} + e^{-\lambda \tau}}. \quad (14)$$

При $\tau=0$ получаем формулу Эрланга—Севастьянова для однолинейной системы

$$p = \frac{\lambda t_{\text{обсл}}}{1 + \lambda t_{\text{обсл}}}. \quad (15)$$

При помощи вышеизложенного элементарного приема можно находить вероятность отказа в более сложных задачах.

Рассмотрим одну из них.

Пусть оператор располагает небольшим количеством одинаковых инструментов, каждый из которых после использования некоторое время непригоден к работе. Такая ситуация весьма часто возникает в жизни.

Например, парикмахер после бритья очередного клиента отправляет бритвенную принадлежность на дезинфекцию, которая длится T_1 минут. Зубной врач-хирург аналогичным образом поступает с хирургическим инструментом. При ремонте сломавшихся деталей станков тупится резец, который направляется на заточку и т. п.

Пусть τ —максимальное время ожидания обслуживания «нетерпеливым клиентом» в такой системе; T_1 —время обработки инструмента, s —число инструментов, которыми располагает оператор. При этом делается естественное предположение, что при наличии нескольких свободных инструментов, к обслуживанию привлекается тот, который освободился раньше.

Ограничимся рассмотрением частного случая, когда $\tau \leq t_{\text{обсл}}$. Обозначим $t_{\text{обсл}} + T_1 = T$. Если $st_{\text{обсл}} \geq T$, то к началу обслуживания любого очередного вызова в распоряжении оператора всегда имеется готовый инструмент, и если оператор свободен, то обслуживание этого вызова может быть немедленно начато. В этом случае задача решается применением формулы (14).

Если $st_{\text{обсл}} < T$, то дело обстоит сложнее. Здесь оператор простаивает не только вследствие того, что по окончании цикла обслуживания не обязательно имеется вызов, но и вследствие того, что при частом следовании циклов обслуживания работа оператора приостанавливается, так как нет готовых к работе инструментов.

Наибольшее число вызовов будет обслужено, если подготовленные инструменты не простаивают. В этом случае за время T будет обслуживаться s вызовов. На самом же деле, возможны простои инструмента.

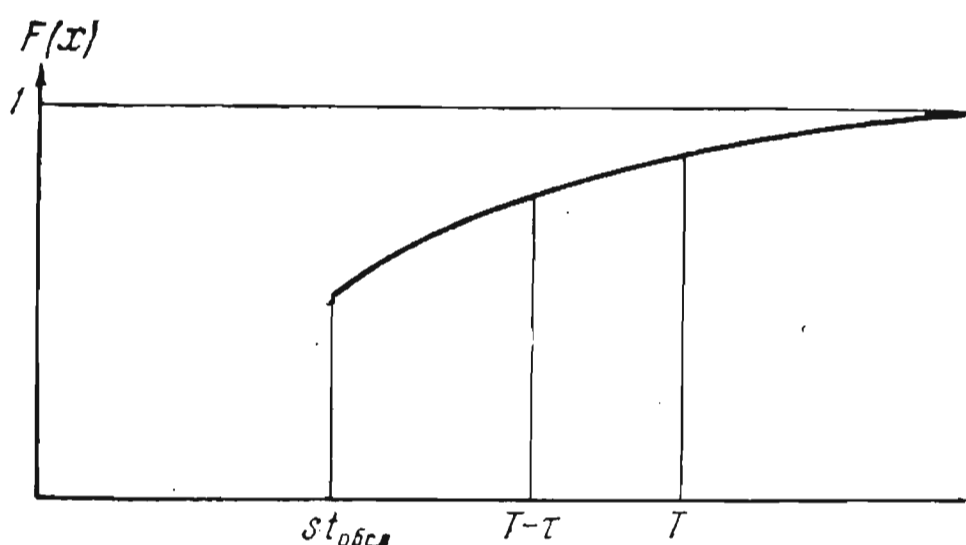


Рис. 2.

s вызовов в среднем будет обслуживаться за время, большее T .

Пусть ξ — отрезок времени между началом обслуживания i -го обслуженного вызова и окончанием обслуживания $i + (s-1)$ -го обслуженного вызова

$$\xi = st_{\text{обсл}} + \sum_{i=1}^{s-1} \xi_i,$$

где ξ_i — время простоя оператора после i -го цикла обслуживания.

Если $\xi \leq T$, то простой инструмента зависит от того, попали ли в интервал $[T-\tau, T]$ вызовы или нет*). Если в этом интервале был вызов, то простой инструмента не наступает, если нет, то имеет место простой инструмента средней длины $1/\lambda$; если $\xi > T$, то имеет место простой инструмента до начала обслуживания следующего после окончания s циклов вызова ($t_{\text{пр}} = \xi - T + \xi_s$ **).

Далее остается произвести надлежащие осреднения. Пусть $F(x)$ — интегральная функция распределения величины ξ . Вероятность того, что среди $s-1$ цикла k будут сопровождаться простоями оператора, а $s-1-k$ нет, равна

$$g_k = C_{s-1}^k (1 - e^{-\lambda\tau})^{s-k-1} e^{-k\lambda\tau}. \quad (16)$$

Случайная величина ξ при условии, что между циклами обслуживания было k промежутков, имеет следующее распределение:

$$F_0(x) = \begin{cases} 0, & \text{если } x < st_{\text{обсл}}; \\ 1, & \text{если } x \geq st_{\text{обсл}}, \end{cases}$$

$$F_k(x) = \begin{cases} 0, & \text{если } x < st_{\text{обсл}}; \\ \int_0^{x-st_{\text{обсл}}} \frac{\lambda^k}{(k-1)!} t^{k-1} e^{-\lambda t} dt, & \text{если } x \geq st_{\text{обсл}} \text{ (при } k > 0). \end{cases} \quad (17)$$

*) За начало отсчета времени принимается начало обслуживания i -го обслуженного вызова.

**) ξ_s — простой оператора и инструмента до начала обслуживания следующего после окончания s циклов вызова.

Подынтегральная функция в (17) есть функция плотности суммы k величин, каждая из которых имеет показательное распределение. Эта функция легко находится путем использования характеристических функций.

Безусловное распределение величины ξ имеет следующий вид:

$$F(x) = g_0 F_0(x) + g_1 F_1(x) + \dots + g_{s-1} F_{s-1}(x). \quad (18)$$

Вероятности попадания величины ξ в интервалы $(\xi \leq T)$ и $(\xi > T)$ равны соответственно

$$\Omega_1 = F(T) \text{ и } \Omega_2 = 1 - F(T). \quad (19)$$

Математическое ожидание η величины простоя инструмента при условии, что $\xi > T$, равно

$$\eta = \frac{1}{\Omega_2} \int_T^\infty (x - T) dF(x) + \frac{1}{\lambda} e^{-\lambda T} = \frac{B}{\Omega_2} + \frac{1}{\lambda} e^{-\lambda T}, \quad (20)$$

где

$$B = \int_T^\infty (x - T) dF(x).$$

Второе слагаемое есть математическое ожидание величины простоя после окончания обслуживания s -го вызова.

Математическое ожидание длины простоя подготовленного инструмента определяется по формуле

$$t_{\text{пр}} = \Omega_1 \frac{1}{\lambda} e^{-\lambda T} + \Omega_2 \eta = \Omega_1 \frac{1}{\lambda} e^{-\lambda T} + B + \Omega_2 \frac{1}{\lambda} e^{-\lambda T} = B + \frac{1}{\lambda} e^{-\lambda T}. \quad (21)$$

Среднее относительное число вызовов, обслуживаемых в единицу времени при достаточно большой длительности процесса, равно

$$\frac{1}{\lambda} \frac{s}{T + t_{\text{пр}}},$$

а среднее число необслуженных вызовов (а следовательно, и вероятность отказа) в такого рода системах равно

$$p = 1 - \frac{s}{\lambda \left(T + \frac{1}{\lambda} e^{-\lambda T} + B \right)}.$$

Например, в однолинейной системе, когда в распоряжении оператора имеется три инструмента,

$$p = 1 - \frac{3}{\lambda \left(T + \frac{1}{\lambda} e^{-\lambda T} + B \right)},$$

$$B = g_1 I_1 + g_2 I_2,$$

$$g_1 = 2(1 - e^{-\lambda T}) e^{-\lambda T},$$

$$g_2 = e^{-2\lambda T},$$

$$I_1 = \int_T^\infty (x - T) dF_1(x) = \frac{1}{\lambda} e^{-\lambda(T - 3t_{\text{обсл}})},$$

$$I_2 = \int_T^\infty (x - T) dF_2(x) = e^{-\lambda(T - 3t_{\text{обсл}})} \left[T - 3t_{\text{обсл}} + \frac{2}{\lambda} \right].$$

Точно такими же приемами может быть решена аналогичная задача для произвольного τ .

В заключение настоящей статьи автор благодарит Б. А. Севастьянова за внимательный просмотр рукописи статьи и за указания на имевшиеся в ней существенные неточности.

ЛИТЕРАТУРА

- [1] Х и н ч и н А. Я., Математические методы теории массового обслуживания, Труды Математического института им. Стеклова, т. 49, 1955 г.
- [2] С е в а с т ь я н о в Б. А., Эргодическая теорема для марковских процессов и ее применение к телефонным системам с отказами, Теория вероятностей и ее применение, т. 2, в. 1, 1957.
- [3] B a r r e r D. I., Quening with Impatient Customers and Indifferent Clerks, Operations Research, № 5, 1957.
- [4] B a r r e r D. I., Quening with Impatient Customers and Ordered Service, Operations Research, № 5, 1957.
- [5] Z i t e k F. Příspěvek k theorii smíšených systémů hromadně obsluhy. Aplikace mat., 1957 2, 2, 154—159. (См. также реферат 4025 в РЖМат, 1958, 5, 101.)

Поступило в редакцию 10 VI 1958.

III. ТЕОРИЯ ИНФОРМАЦИИ

О ЦЕННОСТИ ИНФОРМАЦИИ

А. А. ХАРКЕВИЧ

(МОСКВА)

1. Введение

Теория информации в ее теперешнем виде игнорирует смысл информации и тем более ценность информации для получателя. Это обстоятельство, вытекающее из принятых определений, настойчиво подчеркивается во всех руководствах по теории информации.

Количество информации определяется как мера неопределенности данной ситуации. Если число возможных равновероятных исходов составляло вначале P_0 , а после получения информации сократилось до P_1 , то количество полученной информации определяется как

$$I = \log_2 P_0 - \log_2 P_1 = \log_2 \frac{P_0}{P_1}. \quad (1)$$

2. Задачи с определенной целью

Представляется, однако, возможным обсуждать вопрос о ценности информации в рамках существующей теории, для чего нужно лишь немного видоизменить принятые определения. Все последующее рассуждение основано на представлении о том, что информация собирается для достижения некоторой определенной цели. В пояснение рассмотрим несколько примеров.

а) *Расследование*. Цель, стоящая перед следователем, состоит в том, чтобы обнаружить и изобличить преступника. После ознакомления с основными фактами составляется обычно несколько версий; число подозреваемых может быть велико. Информация, поступающая в виде свидетельских показаний, прямых и косвенных улик, уменьшает число версий и первоначальную неопределенность.

б) *Игра в карты*. Цель игры—выигрыш. Большинство карточных игр основано на том, что вначале неизвестно, какие карты имеются на руках у партнера. По ходу партии информация об этом постепенно возрастает, так что последние ходы опытный игрок основывает уже на полном или почти полном знании чужих карт.

Этот пример ясно показывает, что обсуждаемые вопросы могли бы с самого начала трактоваться в терминах теории игр.

в) *Охота по следу*. Цель охоты—настичь зверя. Информация о его пути и состоянии черпается из оставляемых зверем следов (речь идет не только об отпечатках лап на почве); эти следы должны быть распознаны

и отличены от следов других зверей; след может быть утерян и должен быть найден вновь.

г) *Разработка вакцины.* Цель разработки—получение действенной и безопасной вакцины (или вообще лечебного средства). Информация получается в результате длительной серии экспериментов, сперва над подопытными животными, а затем в клинических условиях.

д) *Стрельба по неподвижной цели с корректировкой.* В этом случае информация получается в виде данных о местах фактических попаданий; на основе этой информации исправляется первоначальная наводка.

е) *Стрельба по движущейся цели.* Имеется в виду стрельба с упреждением, т. е. с предвычислением точки встречи. Информация состоит в данных о положении цели в предшествующие моменты времени. Первые разности дают скорость, вторые—ускорение, и по всем этим данным можно с той или иной достоверностью предсказать траекторию цели.

Подобного рода примеров можно привести сколько угодно; все они сходны в том отношении, что информация собирается для достижения некоторой определенной цели. Информация ценна, поскольку она способствует достижению поставленной цели. Одна и та же информация может иметь различную ценность, если рассматривать ее с точки зрения использования для различных целей. Так, сообщение о погоде имеет значительную ценность для охотника, но не представляет обычно никакого интереса для игрока в карты.

Нужно, впрочем, заметить, что не все случаи получения информации укладываются в эту простую схему. Так, например, те виды информации, которые вызывают эмоции, в частности эстетические, остаются пока вне рассмотрения.

3. Определение ценности информации

Ограничимся рассмотрением той категории случаев, в которых цель, ради достижения которой собирается информация, может быть ясно определена. Тогда ценность информации может быть, естественно, выражена через приращение вероятности достижения цели. Если до получения информации эта вероятность равнялась p_0 , а после получения информации она стала равна p_1 , то ценность полученной информации в указанном выше смысле можно определить*) как

$$\tilde{I} = \log_2 p_1 - \log_2 p_0 = \log_2 \frac{p_1}{p_0}. \quad (2)$$

Выбор логарифмической меры определяется обычным условием аддитивности. Сравнивая (2) и (1), мы видим, что оба определения совпадают, если считать

$$p_0 = \frac{1}{P_0}, \quad p_1 = \frac{1}{P_1}.$$

Таким образом, ценность информации измеряется в единицах информации, и, более того, различие между обеими категориями вообще как бы стирается. Однако определение (2) является в некотором смысле более общим. Дело в том, что соотношение (1) предполагает наличие только одного благоприятного исхода из общего числа P_0 или P_1 ; вообще же говоря, число благоприятных исходов может быть и больше одного. Ве-

*) Такое же определение вводит Вудворд для величины, называемой им «прирост информации» (information gain) (см. [1], гл. 3 § 3).

роятность же определяется как отношение числа благоприятных исходов к общему их числу. Таким образом, (2) можно рассматривать как результат нормировки числа исходов. В пояснение приведем три схемы рис. 1, а, б и в, на которых приняты одинаковые значения $P_0=2$, $P_1=6$. Исходное положение — точка O . На основании полученной информации

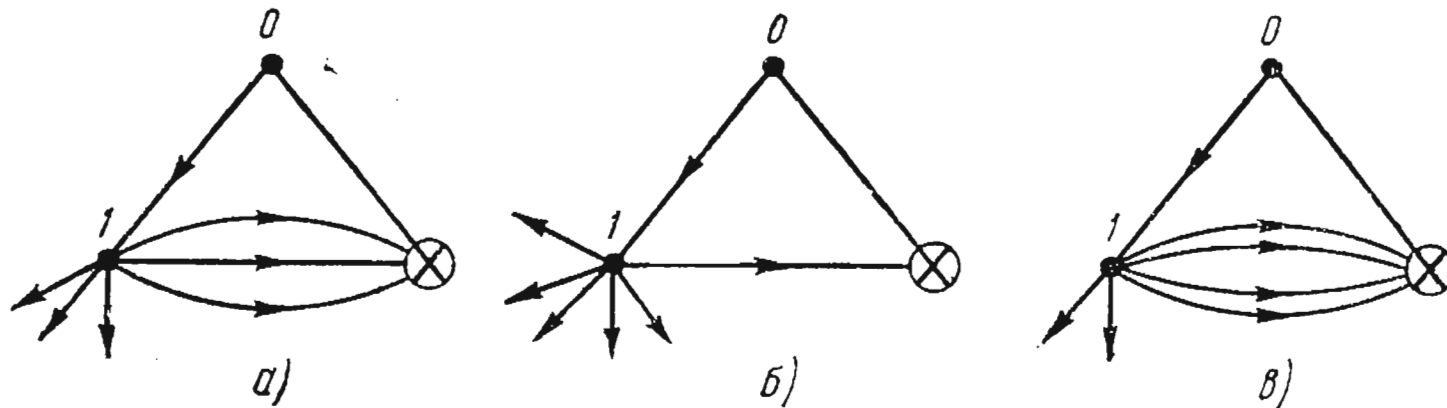


Рис. 1.

совершается переход в точку 1. Цель обозначена крестиком. Благоприятные исходы изображены линиями, ведущими к цели. Определим ценность полученной информации во всех трех случаях.

а) Число благоприятных исходов равно трем: $p_0 = \frac{1}{2}$, $p_1 = \frac{3}{6} = \frac{1}{2}$ и, следовательно, $\tilde{I} = \log_2 \frac{p_1}{p_0} = \log_2 1 = 0$.

б) Имеется один благоприятный исход:

$$p_0 = \frac{1}{2}, \quad p_1 = \frac{1}{6}, \quad \tilde{I} = \log_2 \frac{\frac{1}{6}}{\frac{1}{2}} = -\log_2 3 = -1,58.$$

в) Число благоприятных исходов равно четырем:

$$p_0 = \frac{1}{2}, \quad p_1 = \frac{4}{6} = \frac{2}{3}, \quad \tilde{I} = \log_2 \frac{\frac{2}{3}}{\frac{1}{2}} = \log_2 \frac{4}{3} = 0,42.$$

Как видим, мы совершенно естественно приходим к представлению об отрицательной ценности информации, или просто к отрицательной информации. Отрицательную ценность имеет такая информация, которая, увеличивая исходную неопределенность, уменьшает вероятность достижения цели*). Не придумывая новых слов, можно назвать такую отрицательную информацию дезинформацией; таким образом, в примере б) мы получаем дезинформацию в 1,58 двоичных единиц. В дальнейшем мы будем отождествлять ценность информации с информацией, определяемой согласно (2).

4. Одна простая модель

Очередная задача состоит в том, чтобы составить аналитическую схему рассматриваемой ситуации. Это является пока что очень трудной задачей, общие пути решения которой совершенно не ясны. Мы ограничимся здесь рассмотрением простейшей искусственной схемы, сходной со схемой одномерного случайного блуждания.

Пусть дана прямая с блуждающей точкой, расположенной в исходном положении в начале координат (точка O). Блуждающая точка может совершать единичные шаги вправо и влево с равной вероятностью. Цель,

*) Заметим, что Бриллюэн начисто отрицает возможность появления отрицательной информации в рамках существующей теории (см. [2], стр. 295—296).

которую должна достигнуть движущаяся точка, имеет координату m . В точку с координатой m ведут $\binom{n}{q}$ путей, где n —общее число шагов, q —число шагов вправо. Таким образом,

$$m = q - (n - q) = 2q - n$$

Всего же при n шагах имеется 2^n возможных путей. Следовательно, вероятность p достижения цели ровно за n шагов равна

$$p = 2^{-n} \binom{n}{q} = 2^{-n} \frac{n!}{q! (n-q)!} = 2^{-n} \frac{n!}{\left(\frac{n+m}{2}\right)! \left(\frac{n-m}{2}\right)!}.$$

Вся схема представляется треугольником Паскаля, изображенным на рис. 2. Цифры около узлов сетки представляют собой биномиальные коэффициенты и выражают число путей, ведущих в данный узел при соответствующем числе шагов. Очевидно, что n и m имеют одинаковую четность.

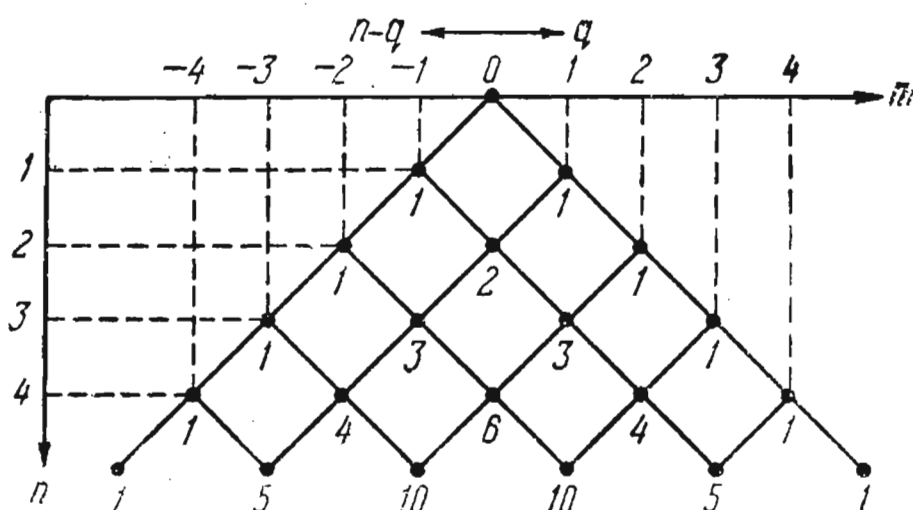


Рис. 2.

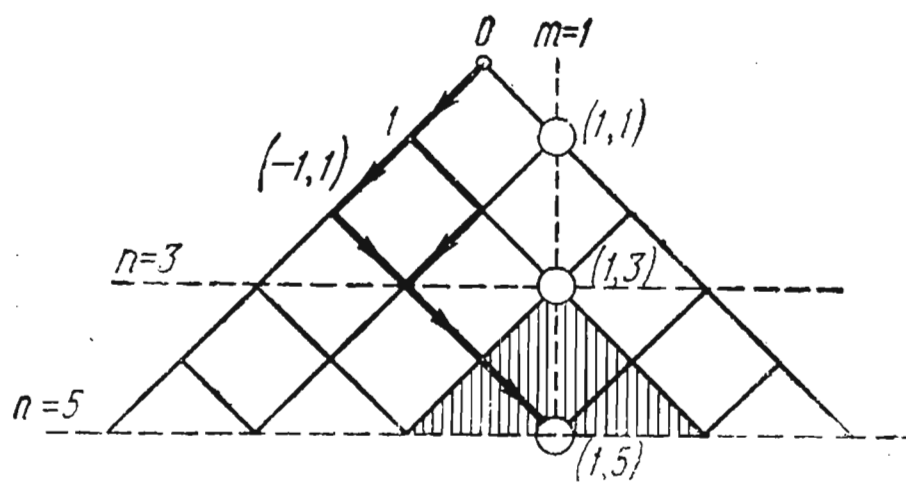


Рис. 3.

Движение точки происходит не вполне случайным образом: каждый очередной ее шаг может быть направлен получаемой информацией; на каждый шаг требуется одна единица информации (так как имеется только два равновероятных исхода: шаг вправо и шаг влево). Будем рассматривать только частный случай $m=1$. В исходном положении для достижения цели достаточно сделать только один шаг вправо; вероятность достижения цели равна $p_0 = 1/2$. Но пусть на основе полученной информации первый шаг сделан влево, так что движущаяся точка оказалась в узле I' (рис. 3). Теперь, чтобы достичь цели, нужно сделать еще не менее двух шагов. Можно поставить задачу двояко:

а) Цель должна быть достигнута ровно за n шагов (включая и первый); $n = 3, 5, 7, \dots$

б) Цель должна быть достигнута числом шагов, не превосходящим n_0 ; $n_0 = 3, 5, 7, \dots$

В принципе возможна еще и такая постановка задачи, когда число шагов не ограничивается; однако эта постановка не представляет интереса, во-первых, потому, что любая практическая задача должна решаться конечной процедурой, а во-вторых, потому, что, как можно показать, вероятность достижения цели стремится к единице при $n_0 \rightarrow \infty$.

Задача а). Вероятности попасть из точки $n = -1$, $n = 1$ в точку $m = 1$ после ровно n шагов (включая первый) выражаются табличкой:

n	3	5	7	...
p_1	$\frac{1}{4}$	$\frac{2}{16}$	$\frac{5}{64}$...

Эта табличка составлена следующим образом: знаменатели представляют собой общее число возможных путей (2^{n-1}); числители выражают число путей к точке $m = 1$; однако из них исключены пути, ведущие через узлы, соответствующие меньшим значениям n . Будем обозначать узлы значениями их координат (m, n) . Тогда из узла $(-1, 1)$ в узел $(1, 3)$ ведет только один путь; из $(-1, 1)$ в $(1, 5)$, ведут всего 4 пути, но два из них проходят через $(1, 3)$, остальные два отмечены на рис. 3. Из числа путей, ведущих из $(-1, 1)$ в $(1, 7)$, сохранены только 5, не проходящих через $(1, 5)$, и так далее. Теперь мы можем подсчитать дезинформацию первого шага, учитывая, что $p_0 = \frac{1}{2}$. Получаем:

n	3	5	7	...
$-\tilde{I}$	1,0	2,0	2,68	...

З а д а ч а б). Комбинаторная задача в этом случае видоизменяется. Число путей определяется следующим образом: при $n_0 = 3$ имеется только один путь. При $n_0 = 5$ нужно учесть все пути, проходящие через $(1, 3)$ (заштрихованный треугольник на рис. 3), куда бы они не шли далее, таких путей 4. К этому нужно добавить пути, ведущие к $(1, 5)$, минуя $(1, 3)$; таких путей 2 (отмечены на рис. 3). Всего, следовательно, к цели ведет 6 путей из 16 возможных, и $p_1 = \frac{6}{16}$. Действуя аналогично, получаем вероятности и дезинформацию

n_0	3	5	7	...
p_1	$\frac{1}{4}$	$\frac{6}{16}$	$\frac{29}{64}$...
$-\tilde{I}$	1,0	0,42	0,14	

Описанное рассуждение основано на приведении к ансамблю n_0 -шаговых путей. Но можно поступать и иначе, а именно: рассматривать вероятности попадания в точку $m = 1$ ровно за 3, 5, 7... шагов как вероятности независимых событий, и попросту суммировать эти вероятности, беря их из таблички задачи а).

Итак, в задаче б) (в отличие от задачи а)) дезинформация убывает с увеличением числа разрешенных шагов. При некотором n_0 / переходит через нуль, так как p_1 все время возрастает, а исходная вероятность $p_0 = \frac{1}{2}$.

Эта простая схема не должна рассматриваться как модель какой-либо реальной ситуации; она приведена здесь только для того, чтобы продемонстрировать возможность введения количественных соотношений в вопросе о ценности информации.

ЛИТЕРАТУРА

- [1] В у д в о р д Ф. М., Теория вероятностей и теория информации с применением к радиолокации, «Советское радио», М., 1955.
- [2] B r i l l o i n L., Science and information theory, 1956.

Поступило в редакцию 21 XI 1958.

IV. ПРОГРАММИРОВАНИЕ

О НЕКОТОРЫХ ПРЕОБРАЗОВАНИЯХ СХЕМ ПРОГРАММ

Н. Г. АРСЕНТЬЕВА

(МОСКВА)

В работе рассматриваются некоторые задачи линейной алгебры, в частности, умножение матриц и нахождение характеристического многочлена матрицы. Приводится преобразование схем программ, которое дает возможность перейти от схем, составленных для решения этих задач в общем случае, к схемам, использующим особенности исходных данных (в частности, симметричность матриц). Такое преобразование позволяет более рационально использовать память машины.

В схемы программ часто входят операторы, зависящие от некоторых параметров. В рассматриваемых нами задачах операторы являются функциями номеров строк и столбцов матриц. Иногда бывает удобно перейти от старых параметров к новым, причем так, что новые (старые) параметры являются функциями старых (новых) параметров и логических условий.

Пусть дана квадратная матрица $A = \|a_{ij}\|$ порядка n . При решении на вычислительных машинах различных задач линейной алгебры в памяти машины приходится хранить всю эту матрицу. Если же матрица A симметрическая, то при решении некоторых задач достаточно хранить лишь часть ее элементов: $\frac{n(n+1)}{2}$ элементов вместо n^2 . При этом мы вводим новые параметры, устанавливаем связь между старыми и новыми параметрами, а затем делаем соответствующие преобразования в схеме программы, составленной для решения данной задачи в общем случае. В качестве этих новых параметров i' и j' можно взять номера строк и столбцов элементов матрицы A , стоящих на главной диагонали и под ней. При этом i и j связаны с i' и j' следующим образом:

$$\left. \begin{aligned} i &= p(i \geq j) \uparrow_1 i' \downarrow_1 j'; \\ j &= p(i \geq j) \uparrow_1 j' \downarrow_1 i', \end{aligned} \right\} \quad (1)$$

т. е. если условие, стоящее в скобках, выполнено, то $i = i'$, если не выполнено, то $i = j'$.

Рассмотрим задачу нахождения произведения двух матриц. Пусть заданы квадратные матрицы $A = \|a_{ij}\|$ и $B = \|b_{kl}\|$. Их произведение обозначим через $C = \|c_{mr}\|$:

$$C = AB.$$

Согласно правилу умножения матриц

$$c_{mr} = \sum_{j=1}^n a_{mj} b_{jr}.$$

Из этой формулы видно, что при вычислении произведения матриц параметры i, j, k, l, m, r должны меняться согласованно, а именно: $j = k, i = m, l = r$.

Пусть оператор A_{ijl} находит произведение $a_{ij}b_{jl}$ и прибавляет его к числу, стоящему на месте элемента c_{il} . При этом предполагаем, что вначале во всех ячейках, где хранится матрица C , стоят нули. Тогда схема счета запишется так:

$$\prod_{l=1}^n \prod_{i=1}^n \prod_{j=1}^n A_{ijl}.$$

Здесь сначала выполняется цикл по j , потом цикл по i , потом по l .

Составим схему программы.

Здесь приняты следующие обозначения: прямыми прописными латинскими буквами обозначаются операторы. В некоторых случаях для обозначения операторов будут применяться и малые курсивные латинские буквы. Как правило, малыми латинскими буквами обозначаются параметры. Слово «Ост.» означает конец работы программы.

При изменении параметра мы должны изменять операторы, зависящие от этого параметра, и счетчик, который служит для проверки конца цикла по этому параметру.

Обозначим через $F(n \cdot i)$ оператор, который изменяет на n единиц значение параметра i , функцией которого является оператор A_i . $F(1 \cdot i)$ мы будем кратко обозначать через $F(i)$, а $F(-1 \cdot i)$ через $F(-i)$. Пусть, далее, оператор $f(i)$ прибавляет единицу к счетчику по i .

$p(\dots)$ — оператор проверки условия, записанного в скобках на месте многоточия; стрелка вверх с индексом k отсылает к той части программы, перед которой стоит стрелка вниз с индексом k , если не выполнено условие, проверяемое оператором p , стоящим перед стрелкой вверх с индексом k . Если условие, проверяемое оператором p , выполнено, то надо переходить к выполнению следующего за p оператора.

Если стрелки вверх с индексом k стоит после оператора, отличного от оператора проверки некоторого условия, то она означает передачу управления тому оператору, перед которым стоит стрелка вниз с индексом k .

Оператор $(1 \rightarrow j)$ означает занесение единицы в счетчик по j .

Схема программы запишется тогда так:

$$\begin{aligned} & \downarrow_{1, 2, 3} A_{ijl} F(j) f(j) p(j > n) \uparrow^1 F(-nj) F(i) f(i) (1 \rightarrow j) p(i > n) \uparrow^2 (1 \rightarrow i) \times \\ & \times F(-ni) F(l) f(l) p(l > n) \uparrow^3 F(-nl) (1 \rightarrow l) \text{Ост.} \end{aligned} \quad (2)$$

(начальные значения параметров: $i = j = l = 1$).

Будем считать, что элементы матрицы записываются в памяти машины по строкам; если при этом элемент a_{11} хранится в ячейке с номером a_1 , n — порядок матрицы, то элемент a_{1n} хранится в ячейке с номером a_{1+n} , элемент a_{21} в ячейке a_{1+n+1} и т. д.

Тогда можно установить связь параметров схемы программы (параметров i, j, l) с параметрами памяти (параметрами s, t и v), причем под

параметрами памяти понимаются величины, характеризующие номера групп ячеек, где хранятся рассматриваемые элементы. Как легко проверить, элементы a_{ij} , b_{kl} и c_{mr} хранятся соответственно в ячейках с номерами

$$\begin{aligned} a_s &= a_1 + j - 1 + n(i - 1); \\ b_t &= b_1 + l - 1 + n(k - 1); \\ c_v &= c_1 + r - 1 + n(m - 1), \end{aligned}$$

где a_1 , b_1 и c_1 — номера ячеек, в которых хранятся соответственно элементы a_{11} , b_{11} , c_{11} . Так как в нашей задаче $k = j$, $m = i$, $r = l$, то

$$\left. \begin{aligned} a_s &= a_1 + j - 1 + n(i - 1); \\ b_t &= b_1 + l - 1 + n(j - 1); \\ c_v &= c_1 + l - 1 + n(i - 1). \end{aligned} \right\} \quad (3)$$

Из (3) следует, что при увеличении параметра i на единицу параметры s и v увеличиваются на n единиц, т. е. что

$$F(i) = F(ns) F(nv). \quad (4)$$

Аналогично

$$\left. \begin{aligned} F(j) &= F(s) F(nt); \\ F(l) &= F(t) F(v). \end{aligned} \right\} \quad (4')$$

Подставляя формулы (4) и (4') в схему программы (2) и учитывая, что

$$F(-n^2v) F(v) = F(-(n^2 - 1)v) \text{ и } F(-ns) F(ns) = E(s),$$

где $E(s)$ — единичный оператор *), получим следующее **):

$$\begin{aligned} & \downarrow_{1,2,3} A_{stv} F(s) F(nt) f(j) p(j > n) \uparrow_1^1 F(-n^2t) F(nv) f(i) (1 \rightarrow j) p(i > n) \uparrow_1^2 \times \\ & \times F(-n^2s) F(t) F(-(n^2 - 1)v) f(l) (1 \rightarrow i) \times \\ & \times p(l > n) \uparrow_1^3 F(-nt) F(-nv) (1 \rightarrow l) \text{Ост.} \quad (2') \end{aligned}$$

Рассмотрим случай, когда матрицы **A** и **B** симметрические и перестановочные. Тогда матрица **C** тоже симметрическая. Введем новые параметры, заменяя i , j на i' , j' ; k , l на k' , l' ; m , r на m' , r' . При этом из формул (1) имеем:

$$\left. \begin{aligned} i &= p(i > j) \uparrow_1^1 i' \downarrow_1^1 j'; \quad k = p(k > l) \uparrow_1^1 k' \downarrow_1^1 l'; \quad m = p(m > r) \uparrow_1^1 m' \downarrow_1^1 r', \\ j &= p(i > j) \uparrow_1^1 j' \downarrow_1^1 i'; \quad l = p(k > l) \uparrow_1^1 l' \downarrow_1^1 k'; \quad r = p(m > r) \uparrow_1^1 r' \downarrow_1^1 m'. \end{aligned} \right\} \quad (5)$$

*) Единичным оператором мы называем оператор, не меняющий параметров; единичные операторы в схемах программ можно опускать.

**) Здесь оператор A_{stv} берет числа из ячеек с номерами a_s и b_t , перемножает их и ставит произведение в ячейку с номером c_v .

Операторы переадресации после этой замены примут следующий вид:

$$\left. \begin{aligned}
 F(i) &= p(i \geq j) \uparrow^1 F(i') \uparrow^2 \downarrow_1 F(j') \downarrow_2; & F(k) &= p(k \geq l) \uparrow^1 F(k') \uparrow^2 \downarrow_1 F(l') \downarrow_2; \\
 F(j) &= p(i > j) \uparrow^1 F(j') \uparrow^2 \downarrow_1 F(i') \downarrow_2; & F(l) &= p(k > l) \uparrow^1 F(l') \uparrow^2 \downarrow_1 F(k') \downarrow_2; \\
 F(m) &= p(m \geq r) \uparrow^1 F(m') \uparrow^2 \downarrow_1 F(r') \downarrow_2; \\
 F(r) &= p(m > r) \uparrow^1 F(r') \uparrow^2 \downarrow_1 F(m') \downarrow_2; \\
 F(-i) &= p(i \geq j) \uparrow^1 F(-i') \uparrow^2 \downarrow_1 F(-j') \downarrow_2; \\
 F(-m) &= p(m \geq r) \uparrow^1 F(-m') \uparrow^2 \downarrow_1 F(-r') \downarrow_2; \\
 F(-j) &= p(i > j) \uparrow^1 F(-j') \uparrow^2 \downarrow_1 F(-i') \downarrow_2; \\
 F(-r) &= p(m > r) \uparrow^1 F(-r') \uparrow^2 \downarrow_1 F(-m') \downarrow_2; \\
 F(-k) &= p(k \geq l) \uparrow^1 F(-k') \uparrow^2 \downarrow_1 F(-l') \downarrow_2; \\
 F(-l) &= p(k > l) \uparrow^1 F(-l') \uparrow^2 \downarrow_1 F(-k') \downarrow_2.
 \end{aligned} \right\} \quad (5')$$

Как легко проверить, верны следующие выражения операторов, увеличивающих и уменьшающих параметры на α единиц через операторы переадресации, зависящие от новых параметров:

$$\begin{aligned}
 F(\alpha j) &= p(i > j_0) \uparrow^1 p(\alpha > i - j_0) \uparrow^2 F((i - j_0) j') \times \\
 &\quad \times F((\alpha - i + j_0) i') \uparrow^3 \downarrow_2 F(\alpha j') \uparrow^3 \downarrow_1 F(\alpha i') \downarrow_3,
 \end{aligned}$$

причем $\alpha + j_0 \leq n$, а j_0 — значение параметра j , которое надо увеличить на α единиц;

$$\begin{aligned}
 F(-\alpha j) &= p(i > j_0) \uparrow^1 p(\alpha > i - j_0) \uparrow^2 F(-(i - j_0) i') \times \\
 &\quad \times F(-(i - j_0) j') \uparrow^3 \downarrow_2 F(-\alpha j') \uparrow^3 \downarrow_1 F(-\alpha i') \downarrow_3,
 \end{aligned}$$

причем здесь $j_0 - \alpha \geq 1$, а j_0 — значение параметра j , которое должно быть уменьшено на α единиц.

Аналогично записываются операторы $F(-\alpha k)$, $F(-\alpha i)$, $F(-\alpha m)$, $F(-\alpha r)$ и $F(-\alpha l)$. Нас интересуют эти операторы при $\alpha = n$, $i_0 = j_0 = l_0 = 1$, т. е. операторы восстановления. Как видно из приведенных выше формул, эти операторы очень сложны, поэтому мы будем пользоваться операторами формирования, которые хранят первоначальный вид модифицируемых команд и в нужный момент восстанавливают эти команды. Обозначим через $[1 \rightarrow j]$ оператор, полагающий $j = 1$ в операторах, зависящих от j , и в счетчике по j .

Так как в нашей задаче $k = j$, то изменение параметров j влечет за собой не только изменение параметров i' и j' , но и изменение пара-

метров k' и l' такое, которое вызывается изменением параметра k , поэтому, используя формулы (5'), мы получим

$$F(j) = p(i > j) \overset{1}{\uparrow} F(j') \overset{2}{\uparrow} \downarrow_1 F(i') \downarrow_2 p(j \geq l) \overset{3}{\uparrow} F(k') \overset{4}{\uparrow} \downarrow_3 F(l') \downarrow_4. \quad (6)$$

Аналогично, так как $m = i$, $r = l$,

$$\left. \begin{aligned} F(i) &= p(i \geq j) \overset{1}{\uparrow} F(i') \overset{2}{\uparrow} \downarrow_1 F(j') \downarrow_2 p(i \geq l) \overset{3}{\uparrow} F(m') \overset{4}{\uparrow} \downarrow_3 F(r') \downarrow_4; \\ F(l) &= p(j > l) \overset{1}{\uparrow} F(l') \overset{2}{\uparrow} \downarrow_1 F(k') \downarrow_2 p(i > l) \overset{3}{\uparrow} F(r') \overset{4}{\uparrow} \downarrow_3 F(m') \downarrow_4. \end{aligned} \right\} \quad (6')$$

Теперь, используя формулы (6), (6') и операторы формирования и учитывая, что во внешних циклах некоторые логические условия тождественно выполняются или не выполняются, мы перепишем схему (2) в таком виде:

$$\begin{aligned} & \downarrow_{1,2,3} A_{ijl} p(i > j) \overset{4}{\uparrow} F(j') \overset{5}{\uparrow} \downarrow_4 F(i') \downarrow_5 p(j \geq l) \overset{6}{\uparrow} F(k') \overset{7}{\uparrow} \times \\ & \times \downarrow_6 F(l') \downarrow_7 f(j) p(j > n) \overset{1}{\uparrow} [1 \rightarrow j] F(i') p(i \geq l) \overset{8}{\uparrow} F(m') \overset{9}{\uparrow} \times \\ & \times \downarrow_8 F(r') \downarrow_9 f(i) p(i > n) \overset{2}{\uparrow} [1 \rightarrow i] F(k') F(m') f(l) p(l > n) \overset{3}{\uparrow} [1 \rightarrow l] \text{Ост.} \quad (7) \end{aligned}$$

Рассмотрим связь новых параметров i' , j' , k' , l' , m' , r' с параметрами памяти s , t , v . Будем считать, что элементы матриц записываются в памяти машины по строкам и в случае, когда мы храним часть матрицы. Тогда легко видеть, что элементы $a_{i'j'}$, $b_{k'l'}$, $c_{m'r'}$ хранятся соответственно в ячейках с номерами

$$\begin{aligned} a_s &= a_1 + j' - 1 + \frac{i'(i' - 1)}{2}; \\ b_t &= b_1 + l' - 1 + \frac{k'(k' - 1)}{2}; \\ c_v &= c_1 + r' - 1 + \frac{m'(m' - 1)}{2}, \end{aligned}$$

где a_1 , b_1 , c_1 — номера ячеек, в которых хранятся соответственно a_{11} , b_{11} , c_{11} .

Рассматривая эти связи, мы находим, что

$$\begin{aligned} \Delta_{i'} s &\equiv s(i' + \Delta i', j') - s(i', j') = i' \Delta i' + \frac{\Delta i'(\Delta i' - 1)}{2}; \\ \Delta_{j'} s &= \Delta j'. \end{aligned}$$

При $\Delta i' = 1$, $\Delta j' = 1$ получаем, что $F(i') = F(i's)$, $F(j') = F(s)$. Аналогично находим, что

$$\left. \begin{aligned} F(k) &= F(k't), F(l') = F(t) \\ F(m') &= F(m'v), F(r') = F(v). \end{aligned} \right\} \quad (8)$$

Как легко видеть,

$$\begin{aligned} i' &= p(i \geq j) \overset{1}{\uparrow} i \downarrow_1 j; & k' &= p(k \geq l) \overset{1}{\uparrow} k \downarrow_1 l; & j' &= p(i \geq j) \overset{1}{\uparrow} j \downarrow_1 i; \\ l' &= p(k \geq l) \overset{1}{\uparrow} l \downarrow_1 k; & m' &= p(m \geq r) \overset{1}{\uparrow} m \downarrow_1 r; & r' &= p(m \geq r) \overset{1}{\uparrow} r \downarrow_1 m. \end{aligned}$$

Но у нас $k = j$, $m = i$, $r = l$, поэтому

$$\begin{aligned} i' &= p(i \geq j) \overset{1}{\uparrow} i \overset{1}{\downarrow} j; \\ k' &= p(j \geq l) \overset{1}{\uparrow} j \overset{1}{\downarrow} l; \\ m' &= p(i \geq l) \overset{1}{\uparrow} i \overset{1}{\downarrow} l. \end{aligned}$$

Тогда

$$\left. \begin{aligned} F(i') &= p(i \geq j) \overset{1}{\uparrow} F(is) \overset{2}{\uparrow} \overset{1}{\downarrow} F(js) \overset{2}{\downarrow}; \\ F(k') &= p(j \geq l) \overset{1}{\uparrow} F(jt) \overset{2}{\uparrow} \overset{1}{\downarrow} F(lt) \overset{2}{\downarrow}; \\ F(m') &= p(i \geq l) \overset{1}{\uparrow} F(iv) \overset{2}{\uparrow} \overset{1}{\downarrow} F(lv) \overset{2}{\downarrow}. \end{aligned} \right\} \quad (8')$$

Подставляя формулы (8) и (8') в (6) и (6') и учитывая, что некоторые логические условия тождественно выполняются или не выполняются, получаем:

$$\left. \begin{aligned} F(j) &= p(i > j) \overset{1}{\uparrow} F(s) \overset{2}{\uparrow} \overset{1}{\downarrow} F(js) \overset{2}{\downarrow} p(j \geq l) \overset{3}{\uparrow} F(jt) \overset{4}{\uparrow} \overset{3}{\downarrow} F(t) \overset{4}{\downarrow}; \\ F(i) &= p(i \geq j) \overset{1}{\uparrow} F(is) \overset{2}{\uparrow} \overset{1}{\downarrow} F(s) \overset{2}{\downarrow} p(i \geq l) \overset{3}{\uparrow} F(iv) \overset{4}{\uparrow} \overset{3}{\downarrow} F(v) \overset{4}{\downarrow}; \\ F(l) &= p(j > l) \overset{1}{\uparrow} F(t) \overset{2}{\uparrow} \overset{1}{\downarrow} F(lt) \overset{2}{\downarrow} p(i > l) \overset{3}{\uparrow} F(v) \overset{4}{\uparrow} \overset{3}{\downarrow} F(lv) \overset{4}{\downarrow}. \end{aligned} \right\} \quad (9)$$

Таким образом, используя (9) и схему (2), получаем схему программы в параметрах памяти:

$$\begin{aligned} & \overset{1}{1, 2, 3} A_{str} p(i > j) \overset{4}{\uparrow} F(s) \overset{5}{\uparrow} \overset{4}{\downarrow} F(js) \overset{5}{\downarrow} p(j \geq l) \overset{6}{\uparrow} F(jt) \overset{7}{\uparrow} \times \\ & \times \overset{6}{\downarrow} F(t) \overset{7}{\downarrow} f(j) p(j > n) \overset{1}{\uparrow} [1 \rightarrow j] F(is) p(i \geq l) \overset{8}{\uparrow} F(iv) \overset{9}{\uparrow} \overset{8}{\downarrow} F(v) \overset{9}{\downarrow} f(i) p(i > n) \overset{2}{\uparrow} \times \\ & \times [1 \rightarrow i] F(lt) F(lv) f(l) p(l > n) \overset{3}{\uparrow} [1 \rightarrow l] \text{Ост.} \end{aligned} \quad (10)$$

Метод Левеерье для нахождения характеристического многочлена матрицы

Пусть дана матрица

$$A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}.$$

Надо найти ее характеристический многочлен

$$D_n(\lambda) = (-1)^n (\lambda^n - p_1 \lambda^{n-1} - p_2 \lambda^{n-2} \dots - p_n).$$

Пусть $\lambda_1, \lambda_2, \dots, \lambda_n$ — корни этого многочлена. Введем следующее обозначение:

$$s_k = \sum_{u=1}^n \lambda_u^k.$$

Известно, что s_k равно следу матрицы A^k и

$$\begin{aligned} kp_k &= s_k - p_1 s_{k-1} - \dots - p_{k-1} s_1, \\ k &= 1, 2, \dots, n. \end{aligned}$$

Из этой системы уравнений последовательно находятся коэффициенты характеристического многочлена.

Составим схему счета.

Пусть оператор A_u находит произведение $A^u = A A^{u-1}$, причем $A^0 = E$. Пусть оператор B'_u вычисляет сумму диагональных элементов матрицы A^u , хранит все эти суммы и записывает A^u на место A^{u-1} . Пусть, наконец, оператор C_u находит p_u по формуле

$$p_u = \frac{1}{u} (s_u - p_1 s_{u-1} - \dots - p_{u-1} s_1).$$

Тогда схема счета коэффициентов характеристического многочлена выглядит следующим образом:

$$\prod_{u=1}^n A_u B'_u C_u.$$

Составим схему программы.

В задачу нахождения характеристического многочлена матрицы, как часть ее, входит уже рассмотренная нами задача об умножении матриц. Поэтому мы будем использовать результаты, полученные ранее. Пусть $A = \|a_{ij}\|$, $B = A^{u-1} = \|b_{kl}\|$, $C = A^u = \|c_{mr}\|$. Тогда $A_u = \prod_{l=1}^n \prod_{i=1}^n \prod_{j=1}^n A_{ijl}$. Пусть B_u вычисляет сумму $c_{11} + c_{22} + \dots + c_{nn}$ и хранит ее, а D_{mr} ставит элемент c_{mr} на место элемента b_{mr} . Тогда $B'_u = B_u \prod_{r=1}^n \prod_{m=1}^n D_{mr}$. Воспользовавшись схемой (2), мы получим следующую схему программы:

$$\begin{aligned} & \downarrow_{1, 2, 3, 6} A_{ijl} F(j) f(j) p(j > n) \uparrow^1 F(-nj) (1 \rightarrow j) F(i) f(i) p(i > n) \uparrow^2 \times \\ & \times F(-ni) (1 \rightarrow i) F(l) f(l) p(l > n) \uparrow^3 F(-nl) (1 \rightarrow l) B_u \times \\ & \times \downarrow_{4, 5} D_{mr} F(m) f(m) p(m > n) \uparrow^4 F(-nm) (1 \rightarrow m) F(r) f(r) p(r > n) \uparrow^5 \times \\ & \times F(-nr) (1 \rightarrow r) C_u F(u) f(u) p(u > n) \uparrow^6 [1 \rightarrow u] \text{ Ост. } \quad (11) \end{aligned}$$

(начальные значения $i = j = l = m = r = u = 1$).

Чтобы написать схему программы в параметрах памяти, воспользуемся формулами (4). Так как $m = k$, $r = l$, то

$$\left. \begin{aligned} F(m) &= F(nv) F(nt); \\ F(r) &= F(v) F(t). \end{aligned} \right\} \quad (12)$$

Подставляя (4), (4') и (12) в схему (11) и делая упрощения, мы получаем следующую схему:

$$\begin{aligned}
 & 1, \downarrow_{2,3,6} A_{stv} F(s) F(nt) f(j) p(j > n) \uparrow^1 F(-n^2 t) F(nv) f(i) (1 \rightarrow j) p(i > n) \uparrow^2 \times \\
 & \times F(-n^2 s) F(t) F(-(n^2 - 1)v) f(l) (1 \rightarrow i) p(l > n) \uparrow^3 F(-nt) F(-nv) (1 \rightarrow l) \times \\
 & \times B_{u,4,5} \downarrow D_{vt} F(nv) F(nt) f(m) p(m > n) \uparrow^4 F(-(n^2 - 1)v) F(-(n^2 - 1)t) \times \\
 & \times (1 \rightarrow m) f(r) p(r > n) \uparrow^5 F(-nv) F(-nt) (1 \rightarrow r) C_u F(u) f(u) p(u > n) \uparrow^6 \times \\
 & \times [1 \rightarrow u] \text{Ост. (13)}
 \end{aligned}$$

Если данная матрица A симметрическая, то и A^u — симметрическая матрица, так как $A^u = A \cdot A^{u-1} = A^{u-1} \cdot A$. Поэтому здесь также возможна сокращенная запись матрицы в памяти машины.

Делаем замену параметров по формулам (5). Используя формулы (6), (6'), схемы (7) и (11), мы получаем схему программы после перехода к новым параметрам:

$$\begin{aligned}
 & 1, \downarrow_{2,3,6} A_{ijl} p(i > j) \uparrow^7 F(j') \uparrow^8 \downarrow_7 F(i') \downarrow_8 p(j \geq l) \uparrow^9 F(k') \uparrow^{10} \downarrow_9 F(l') \downarrow_{10} f(j) p(j > n) \uparrow^1 \times \\
 & \times [1 \rightarrow j] F(i') p(i \geq l) \uparrow^{11} F(m') \uparrow^{12} \downarrow_{11} F(r') \downarrow_{12} f(i) p(i > n) \uparrow^2 [1 \rightarrow i] F(k') F(m') \times \\
 & \times f(l) p(l > n) \uparrow^3 [1 \rightarrow l] B_{u,4,5} \downarrow D_{mr} p(m \geq r) \uparrow^{13} F(k') F(m') \uparrow^{14} \downarrow_{13} F(l') F(r') \downarrow_{14} f(m) \times \\
 & \times p(m > n) \uparrow^4 [1 \rightarrow m] F(k') F(m') f(r) p(r > n) \uparrow^5 \times \\
 & \times [1 \rightarrow r] C_u F(u) f(u) p(u > n) \uparrow^6 [1 \rightarrow u] \text{Ост. (14)}
 \end{aligned}$$

Чтобы написать схему (14) в параметрах памяти, воспользуемся формулами (8), (8') и схемой (10).

Мы получаем следующее:

$$\begin{aligned}
 & 1, \downarrow_{2,3,6} A_{stv} p(i > j) \uparrow^7 F(s) \uparrow^8 \downarrow_7 F(js) \downarrow_8 p(j \geq l) \uparrow^9 F(jt) \uparrow^{10} \downarrow_9 F(t) \downarrow_{10} f(j) p(j > n) \uparrow^1 \times \\
 & \times [1 \rightarrow j] F(is) p(i \geq l) \uparrow^{11} F(iv) \uparrow^{12} \downarrow_{11} F(v) \downarrow_{12} f(i) p(i > n) \uparrow^2 [1 \rightarrow i] F(lt) F(lv) \times \\
 & \times f(l) p(l > n) \uparrow^3 [1 \rightarrow l] B_{u,4,5} \downarrow D_{tv} p(m \geq r) \uparrow^{13} F(mv) F(mt) \uparrow^{14} \downarrow_{13} F(v) F(t) \times \\
 & \times \downarrow_{14} f(m) p(m > n) \uparrow^4 [1 \rightarrow m] F(rv) F(rt) f(r) p(r > n) \uparrow^5 [1 \rightarrow r] C_u F(u) \times \\
 & \times f(u) p(u > n) \uparrow^6 [1 \rightarrow u] \text{Ост. (15)}
 \end{aligned}$$

Метод Леверрье с видоизменением Фаддеева

По методу Леверрье с видоизменением Фаддеева коэффициенты характеристического многочлена матрицы A находятся последовательно при помощи некоторых вспомогательных матриц. Вычисления прово-

дятся в следующем порядке:

$$\begin{array}{lll} \mathbf{A}_1 = \mathbf{A}; & Sp \mathbf{A}_1 = p_1; & \mathbf{B}_1 = \mathbf{A}_1 - p_1 \mathbf{E}; \\ \mathbf{A}_2 = \mathbf{A} \mathbf{B}_1; & \frac{Sp \mathbf{A}_2}{2} = p_2; & \mathbf{B}_2 = \mathbf{A}_2 - p_2 \mathbf{E}; \\ \dots & \dots & \dots \\ \mathbf{A}_{n-1} = \mathbf{A} \mathbf{B}_{n-2}; & \frac{Sp \mathbf{A}_{n-1}}{n-1} = p_{n-1}; & \mathbf{B}_{n-1} = \mathbf{A}_{n-1} - p_{n-1} \mathbf{E}; \\ \mathbf{A}_n = \mathbf{A} \mathbf{B}_{n-1}; & \frac{Sp \mathbf{A}_n}{n} = p_n; & \mathbf{B}_n = \mathbf{A}_n - p_n \mathbf{E}. \end{array}$$

При этом доказывается, что

- а) \mathbf{B}_n — нулевая матрица;
 б) если $|\mathbf{A}| \neq 0$, то $\mathbf{A}^{-1} = \frac{\mathbf{B}_{n-1}}{p_n}$.

Мы ограничимся лишь нахождением ^{$F(n)$} характеристического члена.

Схема счета

Пусть оператор A_u находит произведение AB_{u-1} , $B_0 = E$; оператор S_u подсчитывает p_u , т. е. складывает элементы, стоящие на главной диагонали матрицы AB_{u-1} , и сумму делит на u , а оператор B_u — находит матрицу $B_u = AB_{u-1} - p_u E$ и ставит ее на место матрицы B_{u-1} . Схема счета тогда выглядит так:

$$\prod_{u=1}^n A_u S_u B_u.$$

Составим схему программы.

Пусть $A = \|a_{ij}\|$, $B = B_{u-1} = \|b_{kl}\|$, $C = AB_{u-1} = \|c_{mr}\|$. Тогда

$$A_u = \prod_{i=1}^n \prod_{j=1}^n \prod_{k=1}^n A_{ijk}.$$

Пусть оператор D_{gi} находит разность $c_{gi} - p_i$ и ставит ее на место c_{gi} , а оператор E_{mr} — ставит элемент c_{mr} на место элемента b_{mr} . Тогда

$$B_u = \prod_{g=1}^n D_{gu} \prod_{r=1}^n \prod_{m=1}^n E_{mr}.$$

Воспользовавшись схемой (2), получим схему программы в параметрах i, j, l, m, r :

$$\begin{aligned}
& \downarrow_{1, 2, 3, 7} A_{iu} F(j) f(j) p(j > n) \overset{1}{\uparrow} F(-nj) (1 \rightarrow j) F(i) f(i) p(i > n) \overset{2}{\uparrow} F(-ni) \times \\
& \times (1 \rightarrow i) F(l) f(l) p(l > n) \overset{3}{\uparrow} F(-nl) (1 \rightarrow l) S_u \downarrow_4 D_{gu} F(g) f(g) p(g > n) \overset{4}{\uparrow} \times \\
& \times F(-ng) (1 \rightarrow g) \downarrow_{5, 6} E_{mr} F(m) f(m) p(m > n) \overset{5}{\uparrow} F(-nm) (1 \rightarrow m) F(r) f(r) \times \\
& \times p(r > n) \overset{6}{\uparrow} F(-nr) (1 \rightarrow r) F(u) p(u > n) \overset{7}{\uparrow} [1 \rightarrow u] \text{Oct.} \quad (16)
\end{aligned}$$

(в начале счета $i = j = l = m = r = g = u = 1$).

С помощью формул (4) и (4'), а также схемы (16) мы найдем вид схемы программы в параметрах памяти. Заметим, что

$$F(g) = F((n+1)v), \quad F(-ng) = F(-n(n+1)v),$$

так как элемент c_{gg} хранится в ячейке $c_v = c_1 + g - 1 + n(g - 1)$. Следо-

вательно, схема программы в параметрах памяти имеет вид:

$$\begin{aligned}
 & \downarrow_{1,2,3,7} A_{stv} F(s) F(nt) f(j) p(j > n) \uparrow^1 F(-n^2 t) F(nv) f(i) (1 \rightarrow j) p(i > n) \uparrow^2 \times \\
 & \times F(-n^2 s) F(t) F(-(n^2 - 1)v) f(l) (1 \rightarrow i) p(l > n) \uparrow^3 F(-nt) F(-nv) \times \\
 & \times (1 \rightarrow l) S_u \downarrow_4 D_{vu} F((n+1)v) f(g) p(g > n) \uparrow^4 F(-n(n+1)v) (1 \rightarrow g) \times \\
 & \times \downarrow_{5,6} E_{vt} F(nt) F(nv) f(m) p(m > n) \uparrow^5 F(-(n^2 - 1)t) F(-(n^2 - 1)v) \times \\
 & \times (1 \rightarrow m) f(r) p(r > n) \uparrow^6 F(-nt) F(-nv) (1 \rightarrow r) F(u) f(u) \times \\
 & \times p(u > n) \uparrow^7 [1 \rightarrow u] \text{ Ост.} \quad (17)
 \end{aligned}$$

Если A — симметрическая матрица, то при умножении ее на A^u и на E получаем симметрические матрицы, поэтому можно и в этом случае хранить в машине только части матриц A , B_{u-1} и AB_{u-1} .

Вводя новые параметры по формулам (5) и полагая $g = g'$, с помощью схем (7), (16) мы получаем схему программы в новых параметрах:

$$\begin{aligned}
 & \downarrow_{1,2,3,7} A_{ijl} p(i > j) \uparrow^8 F(j') \uparrow^9 \downarrow_8 F(i') \downarrow_9 p(j \geq l) \uparrow^{10} F(k') \uparrow^{11} \downarrow_{10} F(l') \downarrow_{11} f(j) \times \\
 & \times p(j > n) \uparrow^1 [1 \rightarrow j] F(i') p(i \geq l) \uparrow^{12} F(m') \uparrow^{13} \downarrow_{12} F(r') \downarrow_{13} f(i) p(i > n) \uparrow^2 \times \\
 & \times [1 \rightarrow i] F(k') F(m') f(l) p(l > n) \uparrow^3 [1 \rightarrow l] S_u \downarrow_4 D_{gu} F(g') f(g) p(g > n) \uparrow^4 \times \\
 & \times [1 \rightarrow g] \downarrow_{5,6} E_{mr} p(m \geq r) \uparrow^{14} F(m') F(k') \uparrow^{15} \downarrow_{14} F(r') F(l') \downarrow_{15} f(m) \times \\
 & \times p(m > n) \uparrow^5 [1 \rightarrow m] F(m') F(k') f(r) p(r > n) \uparrow^6 [1 \rightarrow r] F(u) f(u) \times \\
 & \times p(u > n) \uparrow^7 [1 \rightarrow u] \text{ Ост.} \quad (18)
 \end{aligned}$$

Отметим, что $F(g') = F((g+1)v)$, так как элемент $c_{g'g'}$ стоит в ячейке $c_v = c_1 + g' - 1 + \frac{g'(g'-1)}{2}$.

С помощью формул (8) и (8'), схем (10) и (18) мы находим схему программы в параметрах памяти:

$$\begin{aligned}
 & \downarrow_{1,2,3,7} A_{stv} p(i > j) \uparrow^8 F(s) \uparrow^9 \downarrow_8 F(js) \downarrow_9 p(j \geq l) \uparrow^{10} F(jt) \uparrow^{11} \downarrow_{10} F(t) \downarrow_{11} f(j) \times \\
 & \times p(j > n) \uparrow^1 [1 \rightarrow j] F(is) p(i \geq l) \uparrow^{12} F(iv) \uparrow^{13} \downarrow_{12} F(v) \downarrow_{13} f(i) p(i > n) \uparrow^2 \times \\
 & \times [1 \rightarrow i] F(lt) F(lv) f(l) p(l > n) \uparrow^3 [1 \rightarrow l] S_u \downarrow_4 D_{vu} F((g+1)v) f(g) \times \\
 & \times p(g > n) \uparrow^4 [1 \rightarrow g] \downarrow_{5,6} E_{tv} p(m \geq r) \uparrow^{14} F(mv) F(mt) \uparrow^{15} \downarrow_{14} F(v) F(t) \times \\
 & \times \downarrow_{15} f(m) p(m > n) \uparrow^5 [1 \rightarrow m] F(rv) F(rt) f(r) p(r > n) \uparrow^6 [1 \rightarrow r] \times \\
 & \times F(u) f(u) p(u > n) \uparrow^7 [1 \rightarrow u] \text{ Ост.}
 \end{aligned}$$

Поступило в редакцию 20 XII 1957.

МЕТОДЫ АВТОМАТИЗАЦИИ ПРОГРАММИРОВАНИЯ НА ВЫЧИСЛИТЕЛЬНЫХ МАШИНАХ

В. А. ФЕДОСЕЕВ

(МОСКВА)

ВВЕДЕНИЕ

Один из способов решения задач в математике сводится к построению алгоритма, т. е. однозначного процесса вычисления искомых величин.

Как правило, конечным числом операций алгоритм решает задачу приближенно, но зато с любой степенью точности. Однако количество операций (сложений, вычитаний, умножений) для достижения требуемых точностей бывает иногда так велико, что на их проведение не хватило бы человеческой жизни; в то же время разделение работы между группой вычислителей не всегда возможно из-за необходимости непрерывного обмена информацией.

Выход был найден в создании быстродействующих машин, реализующих десятки тысяч операций в секунду. Работа с такой скоростью требует полной автоматизации счетной машины. Машина управляется программой, состоящей из приказов (команд). Каждый приказ указывает, какую операцию и над какими числами необходимо произвести. Числа и приказы хранятся в ячейках «памяти»*) машины, откуда их можно извлекать и куда их можно записывать с большой скоростью. В зависимости от характера результатов машина может делать выбор, передавая управление той или иной группе приказов; операцию передачи управления производит приказ «условной передачи управления» или «условного перехода». Условием для перехода является некоторый признак (например, знак) полученного результата.

Такое установление обратной связи между результатами и ходом решения позволяет реализовать автоматически любой алгоритм, что дало право называть эти машины универсальными.

Одновременно появился интерес к теории алгоритмов. Было обнаружено, что возможно алгоритмическое решение не только «счетных», но и так называемых «неарифметических» задач.

Это прежде всего задачи управления: управление станком легко разбить на последовательность элементарных операций (установка детали; подача инструмента или детали вправо-влево на единицу измерения и т. п.) и задать алгоритм обработки детали.

Аналогичным образом, разработав соответствующие алгоритмы, можно поручить машине управлять работой целых цехов, выполнять роль

*) В ячейке может храниться одно число или один приказ, словом, элемент цифровой информации. Ячейки различаются своими номерами или адресами. В приказе указываются не сами числа, а адреса ячеек, в которых они хранятся.

диспетчера на аэродроме, переводить с одного языка на другой, играть в шахматы и т. д.

Всякий интеллектуальный процесс, после того, как он алгоритмизирован, может быть и автоматизирован, так что человек освобождается для решения новых, более принципиальных задач. К тому же огромное быстроедействие машин позволяет им выполнять ряд алгоритмов, недоступных человеку по своему объему.

Одним из первых применений машин в неарифметических целях было применение их для автоматического составления программ решения задач.

Настоящая статья освещает именно вопрос автоматизации программирования для универсальных машин с помощью этих же машин.

В первом параграфе намечается основной круг вопросов, которые при этом должны быть затронуты; во втором и третьем кратко излагается сущность, возможности и недостатки двух наиболее употребительных методов автоматического программирования.

Изложение ведется на основании опыта трехлетней работы на машине «Стрела» с учетом опубликованных материалов советских и зарубежных авторов. Там, где не оговорено противное, критерии и рекомендации будут даны применительно к логике работы этой машины и могут иметь несколько меньший вес при работе на других машинах.

§ 1. Постановка задачи автоматизации программирования

1. Совокупность приказов—программа—есть описание алгоритма решения в терминах машины, т. е. в терминах доступных ей математических операций.

Приказы, а в приказах—адреса и коды операций—это и есть «термины» машины.

Однако составление программы—программирование—имеет ряд специфических черт, на которых необходимо теперь же остановиться. Человек, совершая выкладку на арифмометре, попутно производит ряд умозаключений; в машине мы все это должны предварительно описать приказами. В этом смысле говорят, что «машинная операция» мельче (или проще) «человеческой».

Поэтому даже алогоритмы средней величины описываются обычно довольно длинными программами: в расплату за разложение сложного алгоритма на простейшие шаги мы вынуждены пользоваться большим количеством этих шагов.

Следствием этого является *малая обозримость* программы.

Необходимость всемерного сокращения программы диктуется в первую очередь ограниченностью объема памяти (средний объем быстроедействующей памяти колеблется от 500 до 4 тысяч ячеек), во вторую—соображениями скорости программирования. На «Стреле», например, нередко решаются задачи, требующие нескольких миллиардов операций; написать такое количество приказов вручную было бы практически невозможно.

К счастью, алгоритмы большинства задач являются циклическими, т. е. содержат повторяющиеся куски, состоящие из одних и тех же операций, совершающихся над новыми числами. Примером может служить задача прямого суммирования рядов, где каждый следующий член получается из предыдущего по одной и той же формуле и прибавляется к общей сумме. Такой алгоритм записывается кратко в виде циклически повторяющегося участка программы или просто «цикла», состоящего (и это существенно) из трех частей:

I. Собственно программа вычисления очередного члена по счетной формуле (программа одного шага).

II. Программа, подготавливающая счет следующего шага (например, изменяющая отдельные команды в I).

III. Программа, выясняющая, закончен ли расчет. Если нет, то реализуется переход к выполнению I.

Таким образом, вместо длинной программы многократно используются одни и те же приказы I части, быть может, только с некоторыми изменениями, производимыми всякий раз частью II. Ясно, что это сокращает программу; однако при этом ее структура усложняется в такой мере, что обзорность от такого сокращения только проигрывает.

Здесь любопытно отметить, что сокращение объема программы получается не за счет сокращения, а, напротив, за счет расширения исходного алгоритма (частями II и III). Таким образом, программа есть описание расширенного (а не исходного) алгоритма. Процесс расширения существенно неоднозначен; от того, насколько удачно произведено расширение, сильно зависит время решения задачи на машине.

2. Трудности, вызванные малой обзорностью программы, еще многократно усиливаются, так как программа утрачивает важное свойство исходного алгоритма—*свойство локальности*. Это свойство состоит в следующем: *если участок алгоритма (например, реализующий счет выражения $\frac{a}{b} \cdot \frac{c}{d}$) выбросить и заменить эквивалентным ему (например, счетом выражения $\frac{ac}{bd}$), то новый алгоритм, как легко видеть, будет эквивалентен прежнему*.

В программе же такое действие невозможно в силу адресного строения памяти машины: и команды программы, и числа, с которыми она оперирует, хранятся в ячейках памяти, различающихся своими адресами, т. е. порядковыми номерами, конструктивно закрепленными за каждой ячейкой.

Единая нумерация памяти накладывает формальную связь на числа и команды всей программы в целом, и локальность по существу независимых частей утрачивается.

Отсутствие локальности, выраженное тем сильнее, чем сложнее структура программы (т. е. чем больше она искусственно сокращена за счет, например, циклов), делает и без того малообозримую программу совсем необозримой; это заставляет программиста при написании, например, пятисотого приказа, держать в голове (или просматривать) четыреста девяносто девять предыдущих и добрую сотню последующих, еще не написанных, помнить мелочи наряду с принципиальными вещами.

Поэтому почти невозможно написать программу без ошибок, отыскание их очень сложно, исправление часто вызывает новые ошибки. Тем более, что часто встречаются ошибки «неустраняемые», т. е. такие, которые заставляют изменять большую часть, если не всю, программу, а иногда даже—и исходный алгоритм. Это случается при программировании в очень ограниченной памяти, когда приказы исправления просто некуда ставить.

Трудности, описанные выше, заставили искать систематический подход к задаче программирования. Так как необходимость расширения алгоритма и нелокальность программы определяются конструктивными свойствами современных машин и, следовательно, не зависят от методов программирования, то первой задачей стало создание обозримого описания программы.

С этой целью по образу схемы счета, применяемой для наглядности описания алгоритма, начали применять *схему программы—описание*

расширенного алгоритма, но не в терминах приказов, а в некоторых более широких терминах «участков программы» или «блоков».

Мелкие детали программы пропускаются и не загромождают описания, благодаря чему такое описание легко обозримо.

В то же время сами «блоки» определяются таким образом, чтобы схема оказалась также и локальной. Это возможно потому, что на схеме отражается только порядок выполнения блоков (в зависимости от тех или иных условий), тогда как адресное строение памяти не отражено совсем.

Составив схему и продумав взаимодействие блоков, можно составлять программы отдельных блоков (сравнительно короткие и, следовательно, обозримые), заботясь лишь об их внутренней непротиворечивости.

После этого можно компоновать программу из блоков уже в окончательном виде; непротиворечивость программы, как будто, должна следовать из непротиворечивости программ блоков и непротиворечивости блок-схемы.

Фактически на этом этапе часто обнаруживаются ошибки, ранее не замеченные из-за того, что схема—описание неполное. Их исправление сводится к дописыванию некоторых из блоков.

Таким образом, опыт доказал рациональность следующего хода работы:

- I этап: а) расширение исходного алгоритма,
- б) разбиение расширенного алгоритма на мелкие части (блоки),
- в) установление связей между ними, т. е. составление схемы программы,
- II этап: г) программирование отдельных блоков,
- д) компоновка единой программы.

При этом особую роль играет вопрос, как (по каким признакам) разбивать расширенный алгоритм на блоки. Различные ответы на этот вопрос могут привести и к различным методам автоматизации программирования.

3. Часть расширенного алгоритма, образующая блок, равно как и описывающая ее группа приказов*), должна, очевидно, иметь некоторое общее свойство, характеризующее блок.

Ведь наша первая цель—создание наглядной (или хотя бы просто обозримой) схемы, а неудачно выбранный признак может эту схему, напротив, только усложнить.

Наиболее естественно в качестве такого общего свойства выдвигать требования связности и логического, формального или содержательного единства, а иногда—и соединения последних свойств (двух или трех).

Под этим понимается следующее:

(А) *Логическое (или функциональное) единство блока означает, что либо все приказы блока выполняются, либо же все они не выполняются: управление тогда сразу передается одному из следующих блоков.*

Логически единый блок можно рассматривать, как обобщенный приказ. Схема из логически единых блоков в явном виде содержит все условные переходы будущей программы.

(В) *Формальное единство требует, чтобы все приказы, реализующие данный блок, были бы одного типа, т. е. производили операции либо над числами (I тип), либо над приказами (II тип), либо над ходом решения*

*) Эту группу также принято называть блоком. Различие между приказами и каким-либо другим описанием блока обычно несущественно, а из контекста всегда ясно, о каком из них идет речь.

(условные переходы, III тип). Блок тогда называется блоком того типа, каковы составляющие его приказы. (При желании можно было бы сформулировать определение формального единства и непосредственно по отношению к расширенному алгоритму.)

Пример. Цикл, рассмотренный в пункте 1, состоит из трех формально единых блоков всех трех типов, тогда как функционально—это один блок.

(С) *Наконец, участок алгоритма может объединять содержательная общность:* блок, например, может реализовать решение данного уравнения и сам по себе быть достаточно сложной программой.

Такое определение, правда, несколько дальше от обычных терминов программирования, чем первые два, однако также бывает очень удобно, так как отражает сущность задачи, для которой составляется программа.

4. Составление схемы, как начальная стадия программирования, суждение о достоинствах и недостатках программы непосредственно по схеме и рассмотрение эквивалентных (по результатам, но не по «качеству») схем привели к мысли о возможности создания своеобразного «исчисления» схем, основы которого были заложены А. А. Ляпуновым еще в 1953 году (см. [5]).

Блоки, из которых составлялись его схемы, удовлетворяли требованиям (А) и (В) предыдущего раздела, т. е. представляли собой логическое и формальное целое. Блок такого типа принято теперь называть оператором *); этого названия мы в дальнейшем и будем придерживаться.

Исследование содержательных (Ляпунов [5]) и формальных (Янов [6]) преобразований схем породили идею автоматического программирования, т. е. программирования с помощью машин.

Здесь сразу же нужно отметить два различных подхода к проблеме автоматизации. Один из них состоит в автоматизации хотя бы какой-нибудь стадии II этапа программирования (I этап предполагается законченным, так как составлена схема программы). Такой подход характерен в основном для работ зарубежных авторов. Напротив, в СССР, на базе идей операторного метода Ляпунова, возникло другое направление: создание универсальной программирующей программы (ПП), т. е. программы, которая в принципе полностью и окончательно решает проблему автоматизации всего второго этапа.

Такая программа была создана в 1955 г. сотрудниками Математического института им. Стеклова Любимским, Камыниным и другими под руководством М. Р. Шура-Бура [3]. К этому же времени и исходя из тех же идей создаются аналогичные программы и для других машин «Стрела».

Несколько позже сотрудники ВЦ АН СССР Ершов [1] и др. составили вариант программирующей программы (ПП) для машины БЭСМ. В принципе также универсальная, она была особенно приспособлена для программирования счетных задач; от программ для «Стрелы» она отличалась типами допустимых операторов.

Более поздние работы в этом направлении, в частности системы ПП, по которым еще только ведутся разработки (например, ПП в ВЦ МГУ и ПП в ВЦ АН СССР для «Стрелы»), преследуют цель, с одной стороны, расширения возможностей ПП, с другой,—улучшения алгоритмов программирования, что даст как увеличение скорости работы ПП, так и написание ею более качественных программ.

*) Точнее, Ляпунов определял оператор как функцию над сообщениями (числами или приказами), определенную на множестве сообщений; таким образом операторы преобразуют информацию, содержащуюся в памяти машины.

Обзор работ в направлении частичной автоматизации приведен в [7] и [8].

Более подробно проблемы, пути их решения и результаты применения автоматизации в обоих смыслах рассмотрены в двух последующих параграфах. Однако, прежде чем перейти к детальному рассмотрению этих вопросов, необходимо упомянуть еще об одной чрезвычайно важной проблеме автоматизации—проблеме контроля программ, т. е. обнаружения в них ошибок.

5. Под *правильностью* программы понимается не только внутренняя ее непротиворечивость, но (что особенно важно) *адекватность программы исходному алгоритму*.

Для того, чтобы исключить ошибки того и другого сорта, готовая программа проверяется одним или двумя программистами. Тщательная проверка нередко занимает столько же времени и сил, как и само программирование, и обычно малоэффективна. Трудности, вызываемые необозримостью и нелокальностью, имеют место при проверке в той же мере, как и при программировании.

Исторически первые попытки автоматизации относятся именно к контролю программ (так называемые программы контроля—ПК), но практически эта область автоматизирована меньше всего.

Принципиально здесь возможны два метода:

1. *Проверка правильности переработки информации при работе программы.*

2. *Проверка правильности структуры программы.*

Первый метод требует, чтобы результаты были известны заранее.

Для сокращения работы часто просчитывается вручную не вся программа, а лишь некоторые ее участки и притом для специально подобранных начальных данных (способ «тестов»). В других случаях идут дальше, и просто просчитывают несколько первых шагов решения в расчетном бюро на клавишных машинках: тождественное совпадение результатов гарантирует правильность работы и программы и вычислителей бюро. Процесс выполнения проверяемой программы и сличения результатов легко может быть автоматизирован. Недостатком метода является необходимость ручных просчетов, а также громоздкость самих программ контроля, затрудняющая их применение.

Второй способ предполагает, что по программе тем или иным способом автоматически восстанавливается ее алгоритм. Вопрос о том, эквивалентен ли этот алгоритм исходному, должен решать уже человек.

Этот способ является более перспективным (ручных просчетов не нужно), но требует, кроме разработки и создания крайне сложной программы анализа, еще окончательной отработки терминов, в которых описывается алгоритм, иначе вопрос об их сравнении остается открытым *).

В этой связи приобретают особенный интерес работы Янова по преобразованию схем программ.

В заключение следует отметить, что к проблеме контроля можно подойти еще с такой точки зрения: само наблюдение за ходом решения задачи на машине могло, казалось бы, дать нам уверенность в правильности результатов, подобно тому, как мы уверены в них при счете на клавишных машинах.

Однако полный автоматизм работы машины и ее скорость приводят к тому, что следить за ходом выкладок невозможно. Попытки обойти это

*) Впрочем, принципиально лучше было бы, конечно, другое решение: полная автоматизация программирования (и I и II этапов); правильное описание алгоритма (схема счета) автоматически порождала бы тогда правильную программу, и вопрос о контроле программ отпал бы совсем. Но это—дело будущего.

затруднение приводят нас к третьему решению проблемы контроля: нужно создать искусственную демонстративность работы программы. Этот вопрос принципиально легко решить программным путем (программы «прокрутки», выполняющие основную программу и анализирующие ее ход), однако практически такое решение неудобно: по самому замыслу—это программы, облегчающие отладку задачи, а план отладки сильно зависит от характера выявляемых ошибок. Поэтому нужно или создать программу с очень богатыми возможностями (чему препятствует ограниченный объем памяти, содержащей еще и исследуемую программу), или иметь возможность управлять работой программы контроля с пульта управления машиной, что также не просто.

Поэтому самым рациональным путем в этом направлении следует считать следующий:

а) *математическая разработка приемов и методов создания демонстративности работы машины;*

б) *проверка всех этих приемов и методов на «программах контроля» и последующее введение их в логику машины,* так как, на наш взгляд, демонстрирующие устройства гораздо удобней и экономичней демонстрирующих программ.

Такие устройства постепенно вводятся в конструкцию существующих машин и, очевидно, будут предусмотрены в создаваемых машинах. Эти устройства останавливают машину в моменты, когда происходит определенный этап работы, по предположению вызывающий ошибку. Такие остановки в узловых моментах, при почти мгновенном выполнении всех остальных команд, позволяют человеку постоянно быть в курсе решения и тем ускоряют отладку.

Еще лучше было бы не останавливать машину, а автоматически передавать управление к дополнительной программе анализа. Такая комбинация позволила бы вести отладку задачи в темпе работы машины, сохраняя в то же время все преимущества программы контроля.

§ 2. Универсальная программирующая программа

В этом параграфе наряду с более подробным изложением проблем II этапа мы рассмотрим основные возможности и особенности метода универсальных программирующих программ. При этом нужно будет четко различать программу, которая составляет, и программу, которую составляют; первую мы будем называть «программирующей программой» или сокращенно—ПП.

1. Задачей ПП является автоматизация работ II этапа. Но II этап состоит из

- а) расписывания программ отдельных операторов,
- б) компоновки этих программ.

При работе методом ПП результаты I этапа (описание схемы) вводятся в машину в виде «исходной информации», максимально приближенной к математическому описанию алгоритма: схемой программы и формулами ее операторов. Последние расписываются по каждому оператору отдельно; порядок расположения этой информации при вводе задает собой схему программы (в явном виде она обычно не вводится).

В связи с необходимостью цифровой кодировки всей информации вместо букв и вспомогательных понятий (например, «тип оператора») вводятся «условные числа» *), а операции обозначают цифровыми кодами, принятыми в логике данной машины.

*) Из них большинство имеет размер адреса и впоследствии, в программе, играет роль «условных адресов» (см. ниже).

Существуют приемы, позволяющие однозначно переводить математическую запись в информацию такого рода; производя перевод «в две руки», мы практически исключаем ошибки на этом этапе.

Автоматический перевод информации в программу требует довольно детального анализа этой информации. Первое и очевидное затруднение представляют скобки, изменяющие порядок действий в формулах; и это не единственная трудность.

Команды операторов II типа, т. е. операторов, перерабатывающих другие операторы, могут быть составлены только при наличии готовых программ тех операторов, которые перерабатываются ими (например, операторов I типа), так как переработка оператора сводится к переработке части его приказов.

Здесь мы встречаемся со своеобразной «иерархией» операторов, отчасти облегчающей работу III.

Важно при этом, что информация об операторе определяет его программу неоднозначно. Эта неоднозначность очень удобна: она позволяет доопределять программу таким образом, чтобы обеспечить ее «экономность».

Экономия может преследовать две цели: (1) *уменьшение времени работы* будущей программы, или (2) *уменьшение объема памяти*, занимаемой программой.

Эти два пути, вообще говоря, противоречат друг другу: улучшая программу в одном направлении, мы, как правило, ухудшаем ее в другом. Например, цикл, описанный в п. 1 § 1, экономя ячейки памяти, увеличивает время работы за счет многократного выполнения частей II и III программы, нужных не для счета, а для его подготовки.

В реализованных III выбрано «среднее»: производится экономия приказов с целью устранить повторение операций при вычислении, например, подобных членов. Затем экономятся рабочие ячейки и константы с целью сокращения загрузки памяти. То, что III производит экономию, имеет огромное значение: даже неполная экономия, проводимая систематически, практически может дать большие результаты, чем остроумная, но не систематически проводимая экономия программиста.

Создание и исследование алгоритмов экономии представляет большой интерес и в теоретическом отношении, особенно для теории алгоритмов и смежных наук.

Задача компоновки написанных программ операторов, достаточно сложная даже для программиста, для машины доступна только тогда, когда эти операторы расписывались специально для такой компоновки; задача компоновки состоит тогда лишь в явном указании связей между операторами, связей, ранее заданных неявно (схемой программы и специальными таблицами).

Так как явно эта связь выражается адресными частями команд компонуемых программ, то этот этап оказалось удобно совместить с этапом «присвоения истинных адресов» взамен условных, т. е. с окончательным оформлением программы.

Но присвоение адресов означает распределение памяти машины. Поэтому все алгоритмы экономии и распределения памяти относятся к этапу «присвоения». Впрочем, полного распределения памяти III не делает, да вряд ли это и нужно при автоматизации только II этапа. Основная трудность присвоения — сильная загруженность памяти, в которой накопилась вся будущая программа, и наличие множества типов условных адресов, каждый из которых «присваивается» по особому правилу.

Таким образом, при создании III приходится решать следующие проблемы:

1. Разработка удобной терминологии (или символики) для описания алгоритмов счетных задач.

2. Разработка программирующих алгоритмов (алгоритмов перевода информации в программу и алгоритмов экономии).

3. Разработка компанующих алгоритмов.

4. Технические проблемы.

Последний пункт вызван тем, что ПП должна работать достаточно быстро (порядка минут), и к тому же уместиться в памяти машины. А между тем объем как исходной информации *), так и получающейся программы (примерно одинаковые) может достигать размеров полной памяти. Привлечение вспомогательной памяти (магнитных лент, перфокарт) резко снижает скорость работы, а оперативность—основное преимущество ПП.

2. Понятие оператора, приведенное в § 1, нуждается еще в уточнении того, что мы считаем «типом приказа» (соответственно—«типом оператора»), или иначе—какие типы допустимы. А. А. Ляпунов делил приказы (соответственно—операторы) на два типа: счетные (арифметические) и управляющие (все остальные). Однако класс «управляющих операторов» явно неоднороден. Поскольку разный характер работы приказов приводит к тому, что они конструируются на основании разной информации и по разным алгоритмам, постольку при разработке ПП пришлось с полной четкостью определить типы допустимых (т. е. обрабатываемых в ПП) операторов.

Этих типов немного:

1. Арифметический оператор, или оператор типа А.

2. Оператор изменения команд, или оператор типа F.

3. Оператор их восстановления, или оператор типа О.

4. Оператор проверки логических условий, или оператор типа Р.

Следуя А. А. Ляпунову, оператор в схеме изображают латинской буквой; произведение двух операторов означает, что они выполняются подряд слева направо; изменение порядка выполнения вследствие условных переходов принято изображать стрелкой или парой уголков, снабженных индексами.

В отличие от этого, мы будем изображать условный переход одним уголком, «прямым» (\sqsubset) в случае, когда переход происходит при выполнении логического условия (на «Стреле» при $\omega=1$), «перевернутым» (\sqsupset)—в противном случае. В качестве же индекса будем писать символ оператора, к которому происходит передача. Именно такой способ записи применяется фактически в созданных вариантах ПП.

Принимается, что в случае отсутствия передачи, выполняется очередная по схеме оператор.

Записывая не всю схему, а лишь ее часть, мы будем на месте обрыва проставлять многоточие.

Пример. Очевидно, цикл (п. 1 § 1) изобразится следующей схемой:

$$\dots AFP \sqsubset_A \dots \quad (1)$$

Если цикл (1) является в свою очередь частью более крупного цикла, то при повторном входе в цикл или при выходе из него нужно «восстановить» все переменные команды (т. е. привести их к исходному виду).

Эту операцию производит оператор восстановления О. Такой цикл изобразится схемой

$$\dots OAFP \sqsubset_A \dots \quad (2)$$

*) Здесь и далее, говоря об объеме информации, мы имеем в виду не известное понятие теории информации, а просто количество занятых ячеек памяти машины.

или

$$\dots AFP \underset{A}{\sqsubset} O \dots \quad (2)$$

Первая схема по ряду соображений предпочтительней.

Операторы типа А, F, O и P являются основными; их достаточно, вообще говоря, для описания любой программы.

Из соображений практического удобства в ряде ПП допускаются еще некоторые операторы, реализующие специальные приемы программирования. В программирующей программе ВЦ АН СССР, составленной для БЭСМ, операторы типов F и O отсутствуют, но зато предусмотрен специальный оператор типа «цикл», равносильный схеме (2).

Такая замена для счетных задач практически не уменьшает универсальности ПП.

Для всех ПП обычно допускается еще *нестандартный оператор*, или *оператор типа N*. Это—готовый кусок программы. Такой оператор не обрабатывается, но присоединяется к общей программе на правах остальных операторов. Примером нестандартного оператора может явиться «маленький цикл», каждый «оператор» которого состоит только из одного приказа. Формально он укладывается в схему (1), фактически же его удобнее считать одним оператором. Дело в том, что маленькие, равно как и очень большие операторы, практически неудобны, поэтому всегда хочется иметь известный произвол в определении оператора; это, однако, противоречит идее четкого определения оператора и усложняет структуру ПП. Попыткой достигнуть компромиссного решения и является оператор N.

3. В рамках данной статьи трудно осветить все алгоритмы, реализация которых составляет программирующую программу. В этом и следующем параграфе будут охарактеризованы основные алгоритмы, работающие в ней; одновременно мы получим пусть грубое, но качественное представление о работе ПП.

Работа ПП начинается с переработки информации об отдельных операторах. Благодаря «иерархии» операторов всю программирующую программу удастся разбить на *блоки*, каждый из которых обрабатывает определенный тип операторов. Вследствие этого эти блоки далее обозначаются буквами, соответствующими типам обрабатываемых ими операторов.

Для наглядности дальнейшее изложение ведется в том порядке, в каком блоки расположены в ПП. Блоки А и Р могут быть переставлены.

3.1 Информацию типа А обрабатывает арифметический блок (блок А). Исходной информацией по оператору типа А является запись формул в условных числах. Для удобства кодирования формулы расписываются в строку. Например, формула

$$x^2 - \frac{a}{b} = y \quad (3)$$

записывается в виде

$$x \cdot x - a : b = y. \quad (4)$$

В правой части должен быть ровно один символ, отсутствующий в левой.

Первые алгоритмы расшифровки формул не учитывали порядка действий; другими словами, действия выполнялись не в обычном порядке (сложение после умножения), а в порядке их написания. Формула (4) в этом случае воспринялась бы следующим образом:

$$(x \cdot x - a) : b = y. \quad (4')$$

Такая условность была вызвана наличием в машине ряда операций (сдвиг числа, изменение знака одного числа в зависимости от знака другого и т. п.), которые не имеют общепринятого порядка.

Алгоритм расшифровки «без учета порядка действий» состоял в следующем:

а) В формуле отыскивается первая слева тройка стоящих подряд символов, из которых крайние—символы чисел, а средний—символ операции. Назовем такую тройку правильной.

б) Формируется команда на операцию, задаваемую правильной тройкой. Совокупность этих команд впоследствии даст программу решаемой задачи.

в) Тройка, породившая приказ, заменяется в формуле одним символом адреса его результата (совпадающим с III адресом изготовленного в б) приказа).

г) Если теперь формула не приведена к виду

$$a = b, \quad (5)$$

переходим к а). В противном случае расшифровка закончена и нужно только III адрес последнего изготовленного приказа заменить адресом окончательного результата «b».

Изменение порядка действий задается скобками: они означают, что расшифровку нужно начинать не от начала формулы, а изнутри самой внутренней скобки.

Соответствующий алгоритм состоит в следующем:

а) Отыскиваем первую слева закрывающуюся скобку. Если ее нет, применить алгоритм расшифровки.

б) Двигаясь от нее влево, отыскиваем первую открывающуюся скобку.

в) К заключенной в скобках части формул применяем алгоритм расшифровки, в котором пункт г) изменен следующим образом:

г') Если часть в скобках не приведена к виду

$$\dots (x) \dots, \quad (6)$$

то переходим к началу алгоритма расшифровки. В противном случае расшифровка скобок закончена и обе скобки в (6) следует стереть. Перейти к пункту а) алгоритма скобок. Указанный алгоритм сложен и не необходим.

Лучший алгоритм состоит в том, что применяется просто описанный алгоритм расшифровки, у которого первый пункт изменен:

а') Ищем первую слева правильную тройку или комбинацию (6). Если сначала найдена тройка—переходим к в), если же комбинация (6), то скобки в ней стираются и снова переходим к а').

Легко проверить, что указанный алгоритм обеспечивает расшифровку скобок.

Если желательно учесть еще и порядок действий, доопределенный каким-либо удобным способом, то, кроме правильной тройки, нужно рассматривать еще и следующий за ней знак операции: он может и запрещать изготовление приказа.

Пример. В выражении

$$a + b \times (a + b + c) = y \quad (6)$$

первая правильная тройка $a + b$ запрещена стоящим справа знаком умножения, тогда как вторая—разрешена.

Алгоритм расшифровки усложняется тем, что в пункте а) будет отыскиваться первая слева разрешенная тройка (или, как и прежде, комбинация со скобками).

Далее, в формулах часто встречаются повторяющиеся комбинации, порождающие одинаковые приказы. Экономия, которую проводит ПП, состоит в том, что вместо повторного построения таких приказов, сразу используется «результат» (его условное число) первого из них.

Экономить приказы можно либо «по формуле», отыскивая в ней похожую комбинацию символов, либо по программе, сравнивая только что изготовленный приказ с написанными ранее.

Первый способ сложнее, но перспективнее, так как позволяет экономить более глубоко.

Пример. Пусть имеем формулу

$$b + c + x(a + b + c) = y. \quad (7)$$

Тройка $b + c$ будет сэкономлена во втором случае только при экономии по формулам, так как при экономии по приказам будет сначала сделан приказ « $a + b$ ».

Алгоритм экономии работает сразу после построения приказа и очень похож на алгоритм расшифровки.

Разница здесь только в том, что знак операции, запрещающий приказ, может стоять не только позади, но и перед правильной тройкой.

Пример. В формуле

$$a + b + x(c - a + b + x) = y. \quad (8)$$

Тройка $a + b$ не может быть сэкономлена, хотя правый знак во втором случае и разрешает экономию.

Второе отличие в том, что пункт б), т. е. формирование приказа, выпадает.

Для сокращения объема исходной информации (т. е. количества символов в формулах) хотелось бы, чтобы ПП обладало возможностью расшифровки сокращенных выражений, которые задают функции, непосредственно на машине не реализуемые.

Однако программирование этих функций сильно осложняет ПП, так что такое расширение возможностей покупается пока слишком дорогой ценой.

Во всяком случае, работы в этом направлении представляют большой интерес и со временем дадут свой эффект.

3.2. Перейдем к блоку логических условий (блок Р). Как мы отмечали во введении, возможность изменять ход выкладок в зависимости от характера полученных результатов, означающая по существу обратную связь, является основой универсальности машин.

Основную роль играет здесь понятие *проверки логического условия* или «пробы». Это несколько приказов (а на некоторых машинах может быть и один приказ), которые проверяют наличие определенного признака (например, знака) у результата, и в зависимости от него производят изменение хода выполнения программы (например, передачу управления блоку, непосредственно за данным не следующему).

Важно, что иногда альтернатива передачи управления решается в зависимости не от одного, а от нескольких логических условий.

Пример. Пусть некоторый расчет требуется выполнить для точек $Q(x, y)$, лежащих внутри контура C_1 , притом внутри пересекающих его контуров C_2 или C_3 . Предполагая, что результат z_i подстановки координат точки Q в уравнение контура C_i ($i=1, 2, 3$) положителен внутри контура, и исследуя знаки результатов z_1, z_2 и z_3 , убеждаемся, что передачу управления к программе расчета нужно производить лишь в случаях:

$$z_1 > 0 \text{ и } z_2 > 0$$

или

$$z_1 > 0 \text{ и } z_3 > 0. \quad (9)$$

Истинность логических условий устанавливается по правилам алгебры логики, являющейся в сущности особой арифметикой двух чисел 1 и 0, в которой дополнительно принимается соглашение $1 + 1 = 1$. Единица соответствует выполненному условию, ноль — невыполненному, знак « \wedge » (читается «и») соответствует умножению, а знак « \vee » (читается «или») «сложению» в нашей «арифметике». Эти правила позволяют определять, выполнено ли условие или нет. В отличие от арифметики обычной в алгебре логики есть еще операция отрицания «не» (обозначается чертой над буквой или выражением, например, « \overline{p} » — читается «не- p »); « \overline{p} » равно единице при « $p = 0$ » и нулю при « $p = 1$ ».

Пример (продолжение). Условие, что точка лежит в данной области, запишется в виде формулы

$$P = (p_1 \wedge p_2) \vee (p_1 \wedge p_3), \quad (10)$$

где p_i равно 1 (условие p_i выполнено), если $z_i > 0$ ($i = 1, 2, 3$).

Блок P предназначен для программирования логических условий. Здесь возможны два пути:

1. Вычисление значения логических функций и передача управления в зависимости от результата этого вычисления (в нашем примере нужно просчитать значение P в (10)).

2. Построение программы, в которой передачи делаются после пробы каждого логического условия. Всякий раз передача происходит или окончательная («главная»), или к следующей пробе.

Пример (продолжение). Обозначим через A оператор требуемого счета, а через B — остальные операторы. Схема, соответствующая формуле (10), имеет вид

$$\dots P_1 \overset{P'_1}{\vdash} P_2 \underset{A}{\vdash} P'_1 \overset{B}{\vdash} P_3 \overset{B}{\vdash} AB, \quad (11)$$

где P_i — оператор, проверяющий логическое условие p_i ; оператор P_1 вошел в схему дважды (P_1 и P'_1).

Первый путь представляет значительный интерес, но простые алгоритмы приводят здесь к программам, более длинным, чем по второму способу (по крайней мере в системе команд машин «Стрела» и БЭСМ).

Алгоритмы построения программы по второму способу напоминают алгоритмы блока A , но проще их, так как символов здесь гораздо меньше.

Нужно отметить, что в функции блока могло бы входить и преобразование логических формул к более краткому виду.

Пример (окончание).
Очевидно,

$$P = p_1 \wedge (p_2 \vee p_3), \quad (10')$$

как легко убедиться подсчетом для всех восьми возможных значений троек (p_1, p_2, p_3) . Соответствующая формуле (10') схема

$$\dots P_1 \overset{B}{\vdash} P_2 \underset{A}{\vdash} P_3 \overset{B}{\vdash} AB, \quad (11')$$

короче, как и породившая ее формула. (Операция, произведенная в соотношении (10), алогична вынесению за скобку общего множителя.)

Нужно, однако, заметить, что проблема упрощения логических формул сложна и, насколько известно, еще никем не была запрограммирована.

В заключение нужно отметить, что блок P (с добавлением возможности преобразования), очевидно, будет иметь большую ценность при окончательной автоматизации программирования, т. е. когда и I этап будет полностью автоматизирован. Пока же блок P практически почти не применяется и имеет ценность только теоретическую.

3.3. После расписывания основной программы, т. е. операторов типа А и Р, можно приступить к построению команд операторов типа F, назначением которых является изменение адресов команд. Такая переадресация применяется, главным образом, в циклах. Задача цикла — счет i -го шага и подготовка программы к следующему $i+1$ -му шагу, приводит нас к важному понятию параметра i . В формулах параметр, как правило, служит индексом, по которому производится суммирование, умножение и т. п.

Пример. Из формулы умножения матриц

$$c_{ik} = \sum_j a_{ij} b_{jk}$$

видно, что вычисление элемента c_{ik} удобно производить циклом по параметру j .

Если при этом формула имеет несколько меняющихся индексов или, по условию задачи, нужно просчитать несколько однотипных формул, зависящих от индексов, то в программе это оформляется несколькими циклами, «вложенными» друг в друга.

Пример (продолжение). Просчет элементов одной строки содержит два цикла: внутренний — по параметру j , внешний — по i . Просчет элементов всей матрицы — три вложенных цикла — по j , по i и по k .

Внутри каждого цикла должны меняться адреса приказов, зависящие только от его параметра; остальные переменные адреса или не изменены в этом цикле или пробегают все свои значения.

Пример (продолжение). При работе цикла по i адреса, зависящие только от k , неизменны, тогда как адреса, зависящие от j , пробегают при каждом шаге по параметру i все свои значения.

Разные адреса могут зависеть от одного или нескольких разных параметров; они могут с изменением на единицу своего параметра меняться либо также на единицу, либо на любое другое целое положительное или отрицательное число m : мы будем говорить, что зависимость от параметра в этом случае равна m (или: адрес зависит от параметра i на m). Адреса, зависящие от нескольких параметров сразу, как правило, зависят от них на разные числа; к тому же эта зависимость часто бывает более сложной, чем зависимость от соответствующего индекса в исходной формуле (одно из следствий расширения алгоритма). Поэтому мы будем строго различать индекс и параметр, хотя принципиально у них много общего, а изображаются они теми же буквами.

Пример (окончание). Пусть адреса величин, зависящих в формуле только от индекса j , увеличиваются с каждым шагом по j на единицу. Тогда (если матрицы квадратные, порядка n) при каждом окончании цикла j (которое происходит при каждом шаге по i) эти адреса должны меняться на $-n$; другими словами, хотя величина по формуле зависит лишь от индекса j , ее адрес зависит от параметра j на $+1$, а от i на $-n$.

Указанные особенности должны быть учтены в алгоритме построения приказов переадресации операторов F.

Информация к каждому оператору F должна указывать, какие операторы и по какому параметру переадресуются (например, «операторы с номерами от N_0 до N_k переадресуются по параметру i »). Но так как разные величины зависят от параметра i по-разному, то, кроме троек (N_0, N_k, i) , должны быть заданы таблицы зависимостей от параметров (общие для всех операторов переадресации, фигурирующих в данной задаче). В каждой таблице указывается, какая величина и на сколько зависит от параметра. Такой характер информации F принят был первоначально еще в ПП Математического института.

Однако зависимость от параметра можно трактовать иначе, а именно так, как подходил к ней Ляпунов, создавая свой операторный метод. Там понятия индекса и параметра разделяются в другом смысле: считается, что все адреса, зависящие от параметра i , меняются на $+1$ при увеличении i на единицу. Если адрес меняется на $-n$, то считается, что и параметр изменился на $-n$. Таким образом, параметр связан здесь не столько с индексом, сколько с числами, зависящими от индекса.

При этом способе прежняя таблица зависимости от параметров уже не нужна, и информация получается более краткая; правда, возможности здесь уже, но для практического программирования их обычно хватает.

Такой подход к зависимости от параметра и такого рода информация были применены в ПП ВЦ АН СССР для машины БЭСМ. Как уже говорилось, вместо операторов типа F, там есть оператор «цикл», но это дела не меняет. Эта ПП допускала программы, в которых адреса зависят не более чем от трех параметров (что соответствует не менее чем трем вложенным циклам).

Алгоритм программирования в обоих случаях одинаков: просматриваются адреса готовых программ операторов, нуждающихся в переадресации; найдя адрес нужного типа, выясняем, на сколько его нужно переадресовать, и записываем соответствующую константу; попутно формируется команда, прибавляющая или вычитающая запасенную константу из переадресуемого приказа.

Экономии состоят в следующем:

а) Экономия команд. Если в одном приказе одновременно переадресуются разные адреса, команды переадресации, равно как и соответствующие константы, объединяются.

б) Экономия ячеек. Каждая вновь полученная константа сравнивается с уже имеющимися (а на «Стреле» сравнение производится и с константами накопителя констант); если подобная константа обнаружена — используется она, а новая не запасается.

Нужно отметить, что блок F, так же как и блок восстановления O (алгоритм работы которого совершенно аналогичен алгоритму блока F), являются наиболее эффективными в том смысле, что отношение объема изготовленных приказов к объему исходной информации*) о них для него максимально (тогда как у блока A оно порядка единицы, а у блока P в ряде случаев даже меньше).

4. Далее следуют компанующие блоки, которые перерабатывают результаты работы предыдущих блоков с целью получить окончательную единую программу. Часть этих блоков носит вспомогательный характер. Например, всякого рода блоки перестановки операторов для удобства их обработки; при другой организации ПП они могут быть выброшены. Напротив, принципиально важными являются блоки экономии рабочих ячеек (блок r) и блок присвоения (блок π).

4.1. *Наиболее полная экономия памяти производится блоком r .* Блоки A (и частично F), расписывая программу, занимают адреса рабочих ячеек под промежуточные результаты подряд, не экономя их, так как впоследствии эти результаты могут понадобиться, например, для экономии приказов. Экономя после написания программы рабочие ячейки, мы получаем возможность свести их число к минимуму.

Возможность экономии базируется на том, что, использовав промежуточный результат в последний раз, мы считаем ячейку как бы освободившейся и можем ее использовать под новый результат.

*) См. сноску на стр. 77.

Простейший алгоритм экономии состоит в просмотре программы сверху вниз, причем для каждой рабочей ячейки дополнительным просмотром выясняем, освободилась она уже или нет (т. е. встречается ли она ниже).

Этот алгоритм был опробован практически. Недостатком его является низкая скорость работы (требуется порядка $\frac{n^2}{2}$ просмотров, где n — число приказов в операторе).

Сотрудниками Математического института АН СССР предложен другой алгоритм, состоящий *в просмотре снизу вверх*. Ввиду того, что последнее появление рабочей ячейки обнаруживается здесь попутно, требуется всего порядка n просмотров. Такой алгоритм реализован теперь в большинстве ПП.

Вторая часть экономии, проводимая блоком r , может состоять в отыскании места для рабочих ячеек (после их экономии). В программе для каждого оператора находится (если он есть) такой оператор из числа пройденных, к которому уже нет возврата. Ячейки, занятые этим оператором, можно использовать под рабочие. Для первых операторов (для которых таких «ненужных» операторов нет) в качестве рабочих ячеек можно попытаться использовать ячейки под окончательные результаты (которые еще не вычислены).

Указанный анализ программы (без последнего добавления) производился ПП Математического института. Поскольку, однако, программа анализа сложна, а экономия ячеек сводит их число к 3—7, то целесообразность анализа вызывает сомнения, ввиду чего в большинстве ПП он отсутствует.

4.2. Задача блока π состоит в переводе программы из условных адресов в истинные, т. е. окончательные. Иначе можно сказать, что блок π производит переработку адресов, или, с другой точки зрения, распределение памяти для работы будущей программы.

В первых вариантах ПП фактическое распределение делал человек, а блок π лишь учитывал его, присваивая истинные адреса. Распределение задавалось в виде таблицы, указывающей, какой группе условных чисел какие адреса соответствуют.

В улучшенном варианте в этой таблице указываются лишь те условные адреса, распределением которых человек интересуется. Остальные адреса блок π распределяет компактно на свободных местах памяти. Прежний алгоритм работы дополнился алгоритмом «дописывания» таблички, данной человеком, но не изменился.

Алгоритм замены адресов согласно указаниям таблицы присвоения был бы не сложнее обычного алгоритма выборки значений функции одного переменного из таблицы с интерполяцией (математически эти задачи эквивалентны), если бы не одно обстоятельство.

Разбивая адреса на классы условных чисел, естественно придерживаться двух правил:

а) условные числа должны отличаться от истинных адресов, а их классы — друг от друга, так, чтобы это можно было легко обнаружить при помощи машинных операций;

б) каждый класс условных чисел должен содержать практически бесконечное (обычно порядка $\frac{1}{4}$ емкости накопителя) число элементов.

Попытки выполнить оба условия приводили к противоречиям, так как часть условных чисел совпадала с истинными адресами*).

*) В машинах, где адрес не имеет запасных разрядов (на «Стреле» они есть), вообще не может быть условных чисел, не похожих на истинные, и вся проблема еще более усложняется.

Благодаря этому происходила путаница, и адрес приходилось присваивать, исходя не только из таблицы присвоения, но и из контекста программы (в основном из кода операции в приказе, где находится исследуемый адрес).

Далее, приказы переадресации указывают адрес приказа, с которым работают, парой чисел—номером оператора, где стоит приказ, и номером приказа, считая от начала оператора. На основании этих данных необходимо производить дополнительные пересчеты значений, полученных из таблицы.

Такое разнообразие задач привело к очень громоздкому блоку л. Сейчас становится ясным, что излишняя сложность присвоения является, по-видимому, следствием не совсем удачного разбиения условных чисел на классы.

Пересмотрев это разбиение в духе идей, изложенных в § 3, очевидно, можно будет значительно упростить как блок л, так и исходную кодировку величин путем уничтожения жестких границ каждого класса. Работы в этом направлении уже проводятся.

5. В заключение рассмотрим вопрос о «качестве» ПП. Принципиально даже самые первые, экспериментальные варианты ПП были универсальными и решали задачу автоматизации II этапа полностью. Однако практически ведется систематическая работа во многих организациях по улучшению действующих и созданию новых, более качественных ПП.

Качество ПП следует определять, учитывая следующие факторы:

- а) *«квалификацию»* ПП,
- б) *скорость работы* ПП,
- в) *качество исходной информации*,
- г) *максимально возможный объем получаемой программы*.

Во вторую очередь нужно учитывать такие свойства, как

- д) *объем программы* ПП,
- е) *простоту применяемых внутренних обозначений и алгоритмов*.

«Квалификация» ПП состоит в том, что она должна «понимать» (т. е. расшифровать) возможно большее количество общепринятых сокращений, в частности основные случаи опускания скобок, порядок действий, знак «минус» перед скобкой и т. п. Она должна, с другой стороны, максимально использовать логику машины, в частности групповые операции, имеющиеся на «Стреле». Учет их достаточно сложен, но частичное решение вопроса уже имеется.

«Квалификация» проявляется и в «умении» экономить; здесь, как указывалось, следует применять экономию по информации, дающую больший эффект.

Важным достижением было бы введение в алгоритмы программирования поисков оптимальной комбинации приказов или хотя бы пробы некоторых случайных наборов их с целью сокращения длины или времени счета программируемой задачи, но это крайне сложно. Квалификация ПП резко повышается и при всех попытках автоматизировать распределение памяти.

Скорость работы ПП, вообще говоря, играет меньшую роль, чем скорость решения задач, так как программирование занимает от нескольких минут до получаса. Конечно, слишком медленная работа снижает оперативность использования ПП, что нежелательно. Однако увеличение скорости работы последних вариантов ПП достигнуто не снижением ее возможностей, а выбором экономных алгоритмов просмотров и экономий.

Громадную роль играет качество исходной информации. Желательно, чтобы объем ее был невелик, но это требование не является основным. Гораздо важнее, чтобы исходная информация была простой по структуре,

имела легко запоминающиеся обозначения и (самое важное!) была бы вполне локальна.

Последнее качество, при быстро и оперативно работающей ПП, позволит во время отладки вносить исправления непосредственно в исходную информацию (что пока, к сожалению, затруднительно).

Желательно, чтобы операторы имели некоторую среднюю длину, а «нестандартных операторов» было бы поменьше. В этом направлении у нас сделан некоторый шаг, а именно различного рода нестандартности не обязательно оформлять в отдельный оператор, так как каждый оператор допускает «нестандартную информацию» (в виде приказов).

Для практических целей имеет серьезное значение также и максимально возможный объем получаемой программы. Так, некоторые задачи не могли быть запрограммированы с помощью прежних ПП по той причине, что получаемая программа превосходила максимально допустимый объем.

Даже неискушенному человеку ясно, что программа ПП очень обширна; действительно, она значительно превышает объем памяти (даже не считая объема исходной информации, который также может иметь порядок объема памяти). Благодаря «иерархии» операторов применяют такую организацию работы ПП, при которой в памяти находится только работающий блок, обрабатывающий последовательно все операторы своего типа.

Принципиально возможно загрузку памяти ограничить не только одним работающим блоком, но и одним, а для операторов типа F и O — двумя операторами, обрабатываемыми блоком. Остальная информация и блоки размещаются на магнитной ленте. Такая организация была опробована автором, но она пока дает некоторый проигрыш во времени работы ПП (из-за медленной работы ленты) и нуждается в доработке.

Говоря о перспективах развития ПП, можно наметить несколько путей.

Первый — автоматизация работ I этапа, приводящая нас к задачам теории алгоритмов.

Второй — усовершенствование автоматизации II этапа, повышение быстродействия и квалификации ПП, разработка новых, более удобных методов описания расширенных алгоритмов, решение ряда технических проблем, которые сейчас остаются открытыми.

Наконец, третий путь — автоматизация контроля программ, написанных при помощи ПП, «самоконтроль» ПП, и в перспективе — отказ от контроля программы, написанной автоматически (и I, и II этап), и, следовательно, без ошибок.

§ 3. Метод библиотеки стандартных подпрограмм

В этом параграфе мы рассмотрим второй из наиболее употребительных методов автоматизации.

Он значительно проще и доступнее, чем метод ПП, но в Советском Союзе применялся сравнительно мало.

Различные варианты автоматизации по этому методу приводят к программам, автоматизирующим различные моменты программирования (в основном конец II этапа); задача полной автоматизации здесь не ставится. Однако и такая автоматизация дает значительные преимущества, что в соединении с простотой метода дает основания применять его наряду с методом ПП.

1. При небольших изменениях в программе (особенно частых в период постановки задачи) по методу ПП задачу приходится всякий раз

программировать заново, так как готовая программа не локальна. При программировании похожих задач также оказывается желательным и возможным отыскать стандартные подпрограммы, которые входили бы в программы большинства этих задач единым куском.

Но, заготовив стандартную программу в виде участка информации ПП, мы еще не снимаем повторяемости работы: нужно создать такие подпрограммы, которые легко присоединялись бы к любой другой программе без предварительной ручной обработки.

Такие подпрограммы, написанные раз навсегда, можно было бы писать с полным учетом всех возможностей машины, с максимальной экономностью, недоступной никакой ПП.

Библиотека таких подпрограмм позволила бы нам набирать программу из готовых кусков (блоков) и тем самым свести время программирования задач, связанных единой тематикой, к минимуму.

Части, отсутствующие в библиотеке, можно писать, например, с помощью ПП. Но ввиду того, что объем таких частей обычно невелик, их можно писать и вручную сразу в виде, удобном для присоединения: это систематически бы пополняло библиотеку.

В этом случае, подходя к вопросу с точки зрения «качества» исходной информации, мы видим, что исходная информация в виде программы хотя и сложнее той, которая используется в ПП, но единообразней, а главное непосредственно может быть использована при отладке, тогда как по формулам задачу отлаживать затруднительно.

В конце концов, поскольку основным затруднением программирования является необозримость и нелокальность, а эти два затруднения при методе стандартных подпрограмм почти снимаются, то исходная информация тем удобней, чем она ближе к окончательной. В маленьких операторах расписать информацию немногим проще, чем написать программу, так что при очень удачной автоматизации конца II этапа автоматизации его начала может и не потребоваться.

2. В этом методе еще сильнее, чем в методе ПП, противоречие между желанием стандартизовать подпрограммы (в дальнейшем мы будем их называть блоками) и желанием сделать их возможно универсальнее. Компромиссное решение можно найти, уменьшая объем блока и сужая его функции, но так мы приходим снова к приказам, пусть более широким, чем в машине, так называемым «псевдокомандам». Но сложные псевдокоманды неуниверсальны и малоупотребительны, простые же мало помогают делу, так как программы из них все же малообозримы.

Программа, выполняющая на машине последовательность псевдокоманд—«псевдопрограмму», называется *интерпретирующей* программой. Ее крупнейшим недостатком является неэкономное использование машинного времени: реализуя циклические вычисления, интерпретатор должен расшифровать одни и те же псевдокоманды столько раз, сколько их нужно повторить; ясно, что это невыгодно.

Гораздо выгоднее сначала сделать обычную программу, а потом заставить ее работать уже без всякой расшифровки.

Для этого применяют неуниверсальные самонастраивающиеся блоки. Такие блоки содержат в первой части программу переработки остального, и по небольшой информации подстраиваются к остальной программе. Однако для систематического пользования методом стандартных подпрограмм такие блоки неудобны.

Проще и легче выделить «настраивающую» часть всех блоков в отдельную *компилирующую* программу. Такие компилирующие программы широко распространены за рубежом. У нас в этом направлении работает группа ленинградских программистов под руководством Л. В. Канто-

ровича. Их программа объединяет в себе свойства компилирующие и интерпретирующие: приказы, построенные при расшифровке, выполняются обычно сразу, и только в случае цикла происходит предварительно расписывание его программы.

Учитывая неэкономность интерпретирующих программ, в Москве за последнее время разрабатывались в основном программы компилирующие.

Ниже описаны два варианта метода автоматизации конца II этапа на базе компилирующих программ.

3. Первый метод разработан Е. А. Жоголевым*) и применяется в ВЦ МГУ. Суть метода состоит в следующем:

Различные блоки программы (сюда относятся для единообразия и «числовые блоки» — массивы чисел) при ее работе взаимодействуют и, следовательно, связаны.

Эта связь неодинакова. Состояние адресных частей одних блоков не зависит от положения в памяти всех остальных массивов и даже их самих; такие блоки назовем *инвариантными* (к таким блокам относятся, например, числовые блоки). Адресная часть других зависит лишь от их собственного положения, их мы назовем *замкнутыми*. Наконец, те блоки, состояние адресной части которых зависит от расположения других блоков, называются *открытыми*.

Аналогично разделим адреса, составляющие адресную часть команд: независимые назовем *постоянными* (например, адреса накопителя констант), зависящие от положения своего блока — *внутренними*, зависящие от положения других блоков — *внешними*. Например, инвариантный блок может содержать только постоянные адреса, замкнутый — только постоянные и внутренние, открытый — адреса всех трех типов.

Цель анализа — доказать, что возможны (с этой точки зрения) всего три указанных типа адресов. Используем теперь старшие разряды адреса, чтобы обозначить эти три типа; другими словами, кодируем каждый адрес следующим образом:

- а) постоянный — нулями в старших разрядах адреса,
- б) внутренний — I в разряде «две тысячи» (все для «Стрелы»).
- в) внешний — I в разряде «четыре тысячи».

Таким образом, постоянные адреса остаются истинными, а внутренние и внешние — условные.

Отсчет внутренних адресов ведем всякий раз с нуля от начала блока. Внешние адреса распределяются в произвольном порядке.

Программирование ведется сразу в условных адресах (для постоянных адресов они совпадают с истинными), а роль компилирующей программы состоит в последовательной переработке условных адресов.

При этом адреса истинные (постоянные) не рассматриваются совсем, а внутренние заменяются суммой адреса первой строки блока и младших разрядов условного адреса. Операция замены производится специальной программой накопителя стандартных программ (НСП) «Стрелы». Программы НСП встроены в машину конструктивно, так что имеют характер элементарных операций. Наконец, присвоение внешних адресов производится специальной компилирующей программой ССП (стандартной соединительной программой), по задачам, сложности и громоздкости напоминающую блок π в методе ПП.

В настоящее время на базе этого «блока π » в ВЦ МГУ создается универсальная ПП, аналогичная описанным в предыдущей главе.

*) Подробно метод Жоголева изложен в его отчете, хранящемся в библиотеке ВЦ МГУ (Москва).

4. На наш взгляд, *самой ценной идеей описанного метода является идея отметки в самом адресе характера его условности*. При этом рассмотрение кодов операций, прежде казавшееся неизбежным, оказывается ненужным. Само же деление на внутренние и внешние адреса представляется излишним: если отвлечься от рассмотрения данного блока, ясно, что гораздо симметричнее будет все адреса, кроме постоянных, на одинаковых правах обозначать условными адресами, не уточняя, является ли он внутренним для данного блока или для другого.

Преимущества такого слияния следующие:

1. Уменьшается число классов, так что достаточно *одного* разряда для различения условных и истинных адресов.

2. Снимается требование, чтобы внутренние адреса отсчитывались от начала блока; это позволяет производить нумерацию команд в *произвольном порядке*, а стало быть, и дописывать приказы *внутри* блока без необходимости переделки его частей.

3. Благодаря простоте и симметрии обозначений сильно упрощается и ускоряется работа компилирующей программы. Так, автору удалось сократить ее в такой степени, что она поместилась в накопителе стандартных программ «Стрелы», приобретя, таким образом, характер *элементарной операции*.

Основанный на нем метод условных адресов (УА) учитывает все приведенные соображения и состоит в следующем.

Диапазон адресов «Стрелы» разбит на два класса:

от 0000 до 3777 — истинные адреса,
от 4000 до 7776 — условные адреса.

Компилирующая программа сохраняет без изменения истинные адреса и допускает произвольное преобразование условных: оно задается таблицей исправлений ТИ.

ТИ состоит из троек чисел (A_n, A_k, α) , означающих, что всякий условный адрес A_i , заключенный в диапазоне $(A_n, A_k)^*$, преобразуется по закону:

$$\text{если } A_n \leq A_i \leq A_k, \quad (12)$$

$$\text{то } a_i = \alpha + A_i - A_n. \quad (13)$$

ТИ может содержать произвольное число таких троек, адрес α может быть как истинным, так и снова условным.

Программа задачи переводится в истинные адреса так же, как и числа—в двоичную систему при помощи группового выполнения программы НСП; скорость перевода очень велика (порядка скорости перевода чисел), она сильно зависит от длины таблицы исправлений и от количества условных адресов в программе.

Программируя задачу из *содержательно единых* и небольших (порядка 2—3 перфокарт) по величине блоков, мы получаем возможность быстро писать программу (не информацию!), удобно ее отлаживать и легко изменять при частичном изменении задачи.

5. Описанный метод УА может применяться для двух целей.

Его первое, основное назначение—задача компиляции готовых программ блоков, т. е. автоматизация второй стадии второго этапа. При этом для метода безразлично, как производится первая стадия: вручную или автоматически, например методом ПП (в этом случае блоку л

*) A_n и A_k —соответственно начальное и конечное значения условного адреса.

нужно дать фиктивную табличку присвоения, что предусмотрено логикой ПП).

Метод УА должен давать значительную экономию при наличии библиотеки типовых блоков—подпрограмм, так как он позволит быстро составлять программы задач, уравнения которых совпадают или близки к уравнениям задач, уже решенных.

Составление самой библиотеки при этом требует мало времени: в качестве основного фонда библиотеки может быть применена программа типовой задачи, разбитая разумным образом на несколько частей (сообразно с числом и сложностью уравнений). Впоследствии, при изменении задания, программа будет пополняться новыми блоками, которые со временем расширят фонд библиотеки до нужного объема.

Можно, однако, подойти к методу УА и с другой точки зрения: этот метод можно применить и для автоматизации (хотя и не полной) II этапа—для, собственно, расписывания программ блоков.

Прежде всего заметим, что расписывать блок (небольшой по объему) сравнительно несложно, так как программа невелика, ее структура проста, т. е. отсутствует одна из главных трудностей программирования—необозримость программы. Это не значит, конечно, что при этом полностью исключается возможность ошибок, но отыскивать их легко, а устранять безопасно благодаря причинам, изложенным ниже.

Дело в том, что отыскание ошибок значительно облегчается, если выводить на печать часть промежуточных результатов. Этот факт широко известен, однако при отладке такие дополнительные выводы применялись лишь в исключительных случаях: это объяснялось тем, что добавление приказов в программу, как мы знаем, чревато опасностями; поэтому считалось, что лучше знать меньшее количество результатов, чем рисковать внесением новых ошибок.

В случае использования метода УА мы можем добавлять вывод результатов после каждого блока. Так как к самим блокам при этом мы не прикасаемся, мы уже не рискуем испортить программу.

Отладив программу, вспомогательные отладочные блоки можно выкинуть, снова не опасаясь за целостность остального: оставшиеся части соединяются автоматически.

Упрощается не только отыскание, но и исправление ошибок. Исправление ошибок сводится в основном к выкидыванию ненужных приказов и к вписыванию необходимых. В случае, если выкидывается столько же приказов, сколько вписывается, это не представляет почти никаких затруднений и без метода УА. Если же эти два числа не равны, то адреса последующих команд меняются, т. е. меняется вся последующая программа. У нас они могут быть присвоены в произвольном порядке (для ССП ВЦ МГУ это не так, но наличие ошибки там заставляет переписывать только конец блока), так что исправление сводится лишь к зачеркиванию и дописыванию, но не переделке всего блока.

Таким образом, метод УА не только автоматизирует вторую стадию II этапа, но и упрощает его первую стадию в такой мере, что метод УА может быть иногда предпочтительней метода ПП, особенно при невысокой математической квалификации программистов.

ЗАКЛЮЧЕНИЕ

В ряде организаций в настоящее время успешно освоены как метод ПП, так и метод стандартных подпрограмм в той или иной их форме.

Как уже говорилось, исторически метод ПП возник первым (в Москве). Появление и распространение после этого «полуавтоматических»

методов представляет, следовательно, шаг назад в теоретическом отношении, а в то же время не секрет, что многие программисты предпочитают их методу ПП.

Чтобы понять причины этого, необходимо сравнить эти два метода также и в практическом отношении. Ниже изложены впечатления именно от практического их применения.

1. Несмотря на то, что даже экспериментальный вариант универсальной ПП в принципе автоматизировал до конца весь II этап, ее практические варианты еще и сейчас нуждаются в доработке (здесь уместно провести аналогию с той «доработкой», которую претерпела идея универсальной машины Тьюринга, прежде чем она была воплощена в металл и электронные схемы). Поэтому применение универсальной ПП сопряжено пока с известными трудностями.

Часть из них носит технический характер (отсутствие буквенного ввода, неприспособленность машинных операций для применяемых алгоритмов программирования и т. п.), другие же более принципиальны.

Многие проблемы, возникающие при программировании, еще не решены совсем или решены только в частных случаях (это и препятствует пока автоматизации I этапа). Сюда относятся прежде всего вопросы рационального распределения памяти машины, а также проблема экономии ячеек памяти по всей программе в целом, включая и числовой материал к ней. Далее, алгоритмы ПП, основанные на таком неполном знании, не всегда приводят к цели, что заставляет вводить в инструкции по пользованию ряд искусственных ограничений.

Кроме того, алгоритмы программируемых задач описываются в терминах, плохо «переводимых» на «язык» машины, особенно если это нужно делать обязательно в терминах ПП; здесь, видимо, предстоит еще работа по формализации основных приемов счета.

Наконец, многих затрудняет необходимость начинать программирование с составления подробной и непротиворечивой схемы программы, тогда как при ручном программировании схема является родом перспективного плана, недостатки которого выявляются постепенно в ходе программирования.

Если же еще учесть сложность и громоздкость самой ПП, десятки «аварийных остановов», которыми она сигнализирует о явных ошибках в исходной информации, а также возможность различных режимов работы ПП, то ясно, что для ее успешной эксплуатации необходима отдельная группа программистов, обслуживающая ПП подобно тому, как обслуживают универсальную машину *).

В результате суммарная производительность труда при методе ПП оказывается (особенно в первое время) несколько ниже, чем можно было бы ожидать, хотя и превышает ручную. Глубокое разделение труда, которое обеспечивает метод ПП (составители схемы программы почти совсем освобождаются от технической работы, тогда как группа ПП совсем не вникает в смысл задачи), должно дать максимальный эффект при большом потоке разнообразных заданий.

Работы по усовершенствованию метода ПП позволят, вероятно, ускорить работу ПП и упростить ее применение в такой мере, что со временем можно будет отказаться от специальной группы ПП.

2. Метод библиотеки стандартных подпрограмм, вообще говоря, не исключает, а дополняет метод ПП. Он является в сущности одной из

*) В качестве курьеза заметим, что здесь необходимы даже текущий и капитальный ремонт так как ПП более всякой машины подвержена «моральному износу».

форм накопления опыта программирования, а именно его документального накопления, в форме готовых или почти готовых к использованию программ. В этом—основное значение метода.

Но поскольку метод базируется на компилирующей программе, постольку он имеет еще и неосновное, хотя и тоже очень важное значение как метод автоматизации программирования, позволяющий составлять сложную программу из коротких и простых для написания блоков (2—3 десятка приказов каждый).

Хотя в этом своем значении метод и значительно беднее метода ПП, но, пока последний не освоен, может также применяться, как это делалось, например, в ВЦ МГУ. Как уже говорилось, их компилирующая программа чрезвычайно сложна; однако, разумным образом поступившись некоторыми возможностями, можно создать очень компактную и удобную программу (метод УА у нас; близкая по идеям, но более громоздкая ПАПА—программа автоматического присвоения адресов в МИАН). Бедность метода в значительной мере окупается тогда простотой использования такой сокращенной программы.

Метод УА, например, не налагающий на программиста практически никаких ограничений, не требующий привлечения к работе магнитной ленты или дополнительных комплектов перфокарт, простой по своим идеям и обозначениям, был освоен меньше чем за месяц и сейчас успешно применяется в качестве полезного подсобного средства.

Метод облегчает отладку программы и особенно удобен при решении однотипных задач.

3. Поскольку современные методы автоматизации программирования не гарантируют отсутствия ошибок в программе, и даже в перспективе—отсутствия ошибок в исходном задании, постольку автоматизация контроля программы по крайней мере столь же важна, как и автоматизация программирования.

Идея «свертывания» программы «в формулы» (точнее—получение по программе исходного алгоритма) приводит к задаче об эквивалентности двух алгоритмов. При современных терминах, в которых формулируется исходный алгоритм, эта задача вряд ли имеет решение.

«Метод» ручных просчетов, заставляющий машинно-счетное бюро «состязаться» с электронной машиной, тоже, очевидно, не может считаться перспективным.

Пока, на наш взгляд, наиболее рациональны устройства контроля, создающие возможность демонстрировать решение. Над их созданием должны потрудиться математики и конструкторы; в будущем, при сплошной автоматизации программирования, вопрос контроля программы, может быть, и потеряет свою остроту, поскольку исходный алгоритм будет однозначно определять программу, и ошибки нужно будет искать только в нем.

ВЫВОДЫ

1. Наиболее мощным и перспективным методом автоматизации программирования является *метод программирующей программы* (ПП), созданный в 1955 г. в Москве.

Метод автоматизирует составление программы *по описанию расширенного алгоритма* (II этап) и будет окончательно завершен после автоматизации составления расширенного алгоритма *по описанию исходного* (I этап).

2. Сейчас, пока метод ПП совершенствуется, большую пользу может принести применение возможно более простой и компактной *компилирующей программы*.

Она позволит накапливать опыт программирования документально, в виде готовых к работе подпрограмм, а также в отдельных случаях облегчит и собственно программирование.

3. Автоматизация программирования не снимает пока еще вопроса *контроля правильности программ*.

В настоящее время контроль осуществляется сравнением результатов работы машины с результатами, вычисленными вручную. Очевидные недостатки такого метода требуют разработки новых, более эффективных способов контроля. Одним из направлений является, например, создание «демонстративных» устройств контроля, позволяющих следить за ходом решения и сразу обнаружить его ошибки.

ЛИТЕРАТУРА

- [1] Е р ш о в А. П., Программирующая программа для быстродействующей электронной счетной машины, М., АН СССР, 1958.
- [2] Е р ш о в А. П., К а м ы н и н С. С., Л ю б и м с к и й Э. З., Автоматизация программирования, Тр. 3-го Всес. матем. съезда, 2, М., АН СССР, 1956, стр. 74—76.
- [3] К а м ы н и н С. С., Л ю б и м с к и й Э. З., Ш у р а - Б у р а М. Р., Об автоматизации программирования при помощи программирующей программы. (Проблемы кибернетики, вып. 1, Физматгиз, М., 1958, стр. 135—171.)
- [4] К е л д ы ш М. В., Л я п у н о в А. А., Ш у р а - Б у р а М. Р., Математические вопросы теории счетных машин, Вестн. АН СССР, № 11, 1956, стр. 16—37.
- [5] Л я п у н о в А. А., О логических схемах программ. (Проблемы кибернетики, вып. 1, Физматгиз, М., 1958, стр. 46—74.)
- [6] Я н о в Ю. И., О логических схемах алгоритмов. (Проблемы кибернетики, вып. 1, Физматгиз, М., 1958, стр. 75—127.)
- [7] A d a m s C. W., Development in programming research. Proc. East. Joint Computer Conf., 1955, New York, 1956, p. 75—78.
- [8] Н о р р е r G. M., Automatic coding for digital computers. Computers and Automat., 4, № 9, 1955, p. 21—24.

Поступило в редакцию 27 II 1958.

О ПОНЯТИИ АДРЕСНОГО АЛГОРИТМА

В. С. КОРОЛЮК

(КИЕВ)

§ 1. Введение

Настоящая статья непосредственно примыкает к работе Л. А. Калужнина [1]. Более того, введение статьи Л. А. Калужнина можно рассматривать как введение и к данной работе. Только ради цельности изложения повторим здесь некоторые положения, важные для понимания дальнейшего.

Поиски общих методов и приемов подготовки математических и логических задач для решения их на цифровых автоматических машинах (ЦАМ) в настоящее время приобретают все более актуальное значение. Основная трудность, с которой приходится иметь дело при использовании ЦАМ для решения задач, состоит в недостаточной формализации как исходной постановки задачи, так и процессов их решения. В связи с этим возникает проблема алгоритмизации — формализации математических и логических заключений, обеспечивающих их реализацию на ЦАМ. Для решения этой проблемы могут быть использованы существующие теории алгоритмов, развитые в связи с центральными вопросами основ математики. Не вдаваясь в детали построения общей теории алгоритмов, сформулируем основные элементы формализации задач. Для описания задачи и ее решения вводится специальный «язык»: исходным материалом служит определенная система объектов, между которыми установлены некоторые соотношения. Природа объектов остается неопределенной, за исключением того, что она согласуется со структурой соотношений, определенных в системе. В общей теории алгоритмов природа рассматриваемых исходных объектов является максимально неопределенной. Между объектами системы устанавливается только соотношение различия: для любых двух объектов системы определено, равны они друг другу или нет. Конечная система объектов, между которыми установлено соотношение различия, называется алфавитом. Отдельные объекты алфавита называются буквами. Содержательная информация о задаче кодируется при помощи букв некоторого алфавита. При этом строится новая система объектов, которые называются условиями задачи. Существуют различные способы кодирования информации о задачах. Каждый способ кодирования основан на определенном представлении о характере задания и преобразовании информации в конкретных содержательных теориях. Например, в теории нормальных алгоритмов А. А. Маркова [2] информация о задаче определяется как слово в некотором алфавите, обладающее определенными свойствами, относящимися к записи слова безотносительно содержательного смысла информации. Соотношения, которые определяют объекты как слова в некотором

алфавите, позволяют задать правила действий над словами. Эти правила действий дают возможность преобразовывать одни слова в другие.

Таким образом, мы приходим к следующему наглядному представлению о построении алгоритмической теории, сформулированному А. Н. Колмогоровым [3]:

1. Рассматривается система объектов, определяемых структурой соотношений между ними. Отдельные объекты системы называются «условиями» или «состояниями».

2. Определяется совокупность допустимых преобразований состояний. Отдельный этап преобразований определяется как переход от данного состояния к другому состоянию.

3. Алгоритмический процесс состоит из последовательности допустимых преобразований. При этом на каждом этапе процесса происходит преобразование состояния, полученного на предыдущем этапе.

4. Алгоритм определяется точным предписанием порядка применения допустимых преобразований.

Решение задачи, которая определяется начальным состоянием A , при помощи алгоритма \mathcal{A} есть заключительное состояние B , которое получается в результате применения алгоритма \mathcal{A} к состоянию A .

Различные способы формализации задач и процессов их решения, т. е. различные определения системы объектов, называемых условиями задач, а тем самым и различные определения системы допустимых операций, приводят к построению различных алгоритмических теорий. Если дана теория, то проблема алгоритмизации состоит в оформлении содержательных задач и процессов их решения на языке данной теории. Каждый тип универсальной ЦАМ имеет свой собственный язык, зависящий от технических данных машины. Поэтому формализация задач и процессов их решения на языке данной ЦАМ содержит не только специфические особенности алгоритмов, но зависит также от технических свойств ЦАМ. Техническая ограниченность возможностей ЦАМ усложняет процесс алгоритмизации задач, затрудняет обмен опытом программирования. С другой стороны, языки существующих теорий алгоритмов также обладают рядом существенных недостатков с точки зрения простоты построения алгоритмов в них и перевода алгоритмов на язык конкретной ЦАМ.

Работа по алгоритмизации задач приводит к необходимости построения алгоритмического языка, удовлетворяющего следующим требованиям:

1. Алгоритмический язык должен в возможно более простой форме выражать общие принципы построения алгоритмов: детерминированность, массовость, результативность.

2. Система допустимых преобразований информации должна быть достаточно простой, чтобы обеспечить обозримость алгоритмов, и достаточно гибкой, чтобы допускать простое построение алгоритмов для достаточно широкого круга задач.

3. Алгоритмы должны быть эффективными, т. е. должны допускать не только потенциальную, но и реальную возможность применения алгоритмов. Другими словами, число отдельных этапов действия алгоритма должно быть практически осуществимым.

4. Перевод алгоритмов с алгоритмического языка на языки существующих ЦАМ должен осуществляться просто, по возможности автоматически.

Анализ алгоритмов, разработанных участниками семинаров по автоматизации программирования и теории алгоритмов в г. Киеве (руководители Л. А. Калужнин, В. С. Королук, Е. Л. Ющенко), дает возможность рассмотреть следующий вариант алгоритмического языка.

§ 2. Основные элементы алгоритмического языка

1. Кодирование информации. Как было указано во введении, информация о всякой содержательной задаче кодируется при помощи определенного набора элементарных символов, называемых буквами некоторого алфавита. При этом строится новая система объектов, которые называются условиями задачи. Выбор способа кодирования информации мы осуществим на основании представлений о характере задания и преобразования информации в современных ЦАМ. Как известно, подготовка информации о задаче для ЦАМ состоит в кодировании отдельных элементов информации при помощи чисел и в размещении этих чисел в ячейках запоминающего устройства машины в определенном порядке. При кодировании отдельных элементов информации учитывается не только различие между ними: как правило, имеет место определенная связь между отдельными элементами информации. Иначе говоря, некоторые элементы информации обладают общими свойствами. Например, при кодировании алгебраических выражений используются коды, изображающие числа, коды, изображающие операции, и т. д. Связь между кодами—свойства кодов—определяется при помощи их классификации.

Описание алгоритма преобразования исходной информации о данной задаче обычно задается в такой форме, чтобы это описание было пригодным для решения целого класса однотипных задач. Например, приказ трехадресной ЦАМ $\boxed{+ \mid a \mid b \mid c}$ осуществляет сложение двух чисел, заданных в ячейках a и b . Помещая в ячейки a и b различные числа, можно при помощи одного и того же приказа сложить различные числа. Таким образом, описание алгоритма осуществляется в символической форме, которая обеспечивает применимость алгоритма к целому ряду однотипных задач, чем и обеспечивается массовость алгоритма. В описании алгоритма участвуют не сами элементы информации конкретной задачи, а их символические наименования, в машинной терминологии—адреса. При этом адреса, которые используются для описания алгоритма, кодируются так же, как и элементы информации. Поэтому можно рассматривать и адреса элементов информации и сами элементы информации как коды из одного и того же множества. В ЦАМ, например, и адреса и числа, которыми кодируются отдельные элементы информации, являются числами в определенной системе счисления (часто в двоичной). Коды, используемые для кодирования информации, являются словами в некотором конечном алфавите. Множество кодов, таким образом, является конструктивным (вообще говоря, бесконечным) множеством.

Размещение чисел, которыми кодируются отдельные элементы информации, в ячейках запоминающего устройства ЦАМ можно рассматривать как задание соответствия между кодами (двоичными числами). Преобразование информации в ЦАМ при этом состоит в изменении содержания ячеек запоминающего устройства, т. е. в изменении соответствия, определяющего начальные условия задачи.

Мы приходим, таким образом, к следующей схеме кодирования информации о задаче.

Пусть \mathfrak{A} —конструктивное множество элементов, называемых кодами.

Исходные условия задачи в данном множестве \mathfrak{A} определяются отображением A кодов данного множества \mathfrak{A} в это же множество. Отображение A определяется конструктивно либо конечной системой соотношений вида $Aa = {}^1a$ (a и ${}^1a \in \mathfrak{A}$), либо правилами построения таких соотношений. В соответствии с существующей интерпретацией кодов в ЦАМ назовем код a адресом кода 1a , который определяется отображением

$Aa = {}^1a$. В свою очередь код 1a называется содержимым адреса a . Область определения отображения A не всегда совпадает со всем множеством \mathfrak{A} . Однако можно ввести «пустой» код Λ и доопределить данное отображение A на все множество \mathfrak{A} таким образом, что если код a не принадлежит области определения отображения A , считать $Aa = \Lambda$. Таким образом, если область определения отображения A совпадает со всем множеством \mathfrak{A} , то каждый код множества \mathfrak{A} может рассматриваться как адрес некоторого другого кода этого же множества.

Пусть b есть содержимое адреса a , т. е. $Aa = b$. Рассматривая b как адрес, определим содержимое его c из соотношения $Ab = c$. Код a будет тогда адресом адреса кода c : $A^2a = A(Aa) = Ab = c$. Назовем в этом случае код a адресом 2-го ранга кода c . Подобным же образом можно определить адрес k -го ранга при любом натуральном k . Итак, каждый код множества \mathfrak{A} , в котором задано отображение A , определяющее условия задачи, может рассматриваться как адрес некоторого другого кода того же множества, и притом адрес некоторого ранга. Содержимое адреса a , рассматриваемого как адрес k -го ранга (k —целое положительное число), определяется из соотношения $A^k a = A(A^{k-1}a) = {}^k a$. Код ${}^k a$ является содержимым адреса a k -го ранга. Обычно в универсальных ЦАМ для описания алгоритмов в виде программ используются только адреса 1-го ранга. В ЦАМ операции, выполняемые отдельными приказами машины, осуществляются над содержимым ячеек запоминающего устройства, номера которых указаны в приказах. Использование адресов 2-го ранга существенно сокращает информацию об алгоритмах многих логических задач. Рассмотрим, например, приказ $\boxed{+ \mid \varphi \mid a \mid b}$, где φ —адрес 2-го ранга, т. е. номер ячейки, содержащей номер ячейки, в которой содержится число, участвующее в операции. При помощи такого приказа, не меняя его вида, а изменяя лишь содержимое ячейки φ , можно сложить с числом из ячейки a число, которое находится в любой ячейке запоминающего устройства.

2. Преобразование информации. В предыдущем пункте была определена система объектов, которые являются условиями задачи или состояниями. Условия задачи или состояния определяются отображением A из множества \mathfrak{A} в данное множество \mathfrak{A} . В соответствии с этим преобразование состояний при помощи допустимых алгоритмических операций состоит в изменении отображения A , определяющего условия данной задачи, в новое отображение в множестве \mathfrak{A} .

Описание алгоритмических операций должно осуществляться по мере накопления опыта алгоритмизации различных задач. Различные наборы допустимых алгоритмических операций могут оказаться удобными для различных классов задач. При этом не следует ограничивать сложность вводимых алгоритмических операций. Важно только, чтобы действие таких операций было вполне определенным и в то же время обеспечивало их универсальность, т. е. применимость отдельных операций для решения различных задач. Универсальность алгоритмических операций означает еще, что они не зависят от специфических особенностей исходного отображения A , определяющего условия задачи.

Как было отмечено в п. 1, в алгоритмических операциях коды рассматриваются как адреса, а операции выполняются над содержимым соответствующих адресов. При этом одни и те же коды могут использоваться как адреса различных рангов. Каждый код, рассматриваемый в алгоритмических операциях как некоторый адрес, содержит индекс—ранг этого адреса. Индекс, определяющий ранг адреса, будем помещать вверху слева от кода. Например, запись ${}^k a$ означает, что рассматривается (обозре-

вается) код $A^k a$. Запись $^1 a$ является в этих условиях сокращением записи Aa . Условимся, что отсутствие индекса ранга адреса означает, что рассматривается код нулевого адреса, т. е. обозревается данный код. Отсутствие индекса ранга адреса означает, что вместо $^0 a$ пишется для простоты a .

Понятие адреса различных рангов обеспечивает массовость алгоритмических операций. Разнообразие условий задач (состояний) определяется структурой соотношений в системе объектов, которые могут служить условиями задач. В нашем рассмотрении состояния определяются отображениями в данном множестве кодов. Возможности определения условий задач только при помощи отображений в некотором множестве кодов \mathfrak{A} , вообще говоря, ограничены. Уже было указано, что отдельные элементы информации могут обладать общими свойствами. Кроме того, могут существовать некоторые соотношения между элементами информации, например соотношения порядка.

Линейная запись информации означает, что для каждого элемента определен соседний справа и слева элемент; при задании матрицы каждый элемент матрицы определяется номером строки и столбца, на пересечении которых он стоит, и т. п. Поэтому необходимо предположить, что структура соотношений в рассматриваемом множестве кодов \mathfrak{A} определяется не только отображением A . В дальнейшем предполагаем, что в множестве \mathfrak{A} задана система операций C_1, C_2, \dots, C_n , являющихся отображениями \mathfrak{A} в \mathfrak{A} , т. е. для некоторых кодов $a \in \mathfrak{A}$ определены коды $C_i a$, также принадлежащие \mathfrak{A} . Отметим следующее соотношение: $C_i^k a = C_i A^k a$. C_1, C_2, \dots, C_n являются одноместными операциями в множестве \mathfrak{A} . Очевидно, что операции C_i не коммутативны с исходным отображением A , т. е. что

$$C_i A^k a \neq A^k C_i a.$$

Определим теперь классификацию свойств кодов множества \mathfrak{A} . Назовем *классификацией кодов множества \mathfrak{A}* любую систему \mathfrak{M} подмножеств \mathfrak{A} . Каждый элемент $\mu \in \mathfrak{M}$ называется *свойством кодов множества \mathfrak{A}* . Будем говорить, что код a обладает (не обладает) свойством μ , $\mu \in \mathfrak{M}$, если $a \in \mu$ ($a \notin \mu$). Заметим, что классификация \mathfrak{M} свойств кодов множества \mathfrak{A} может быть задана при помощи системы отображений, которые являются характеристическими функциями множеств.

Теперь структура соотношений системы объектов, определяющих условия задач, полностью определена. Подведем итог. Исходным служит конструктивное множество \mathfrak{A} элементов, называемых кодами. В множестве \mathfrak{A} задается система операций C_1, C_2, \dots, C_n между кодами, каждая со своей областью определения. Области значений C_i также принадлежат \mathfrak{A} . Система подмножеств \mathfrak{M} кодов из \mathfrak{A} определяет классификацию свойств кодов множества \mathfrak{A} , так что для любого $\mu \in \mathfrak{M}$ каждый код $a \in \mathfrak{A}$ обладает свойством μ , если $a \in \mu$, и свойство μ отсутствует у кода a , если $a \notin \mu$. Условие задачи в множестве \mathfrak{A} определяется некоторым отображением A кодов множества \mathfrak{A} в множество \mathfrak{A} . Построение отображения A , определяющего условия конкретной содержательной задачи, осуществляется с учетом системы операций C_1, C_2, \dots, C_n и классификации свойств \mathfrak{M} . Это построение производится содержательно, т. е. оно не формализовано.

Определим теперь алгоритмическую операцию преобразования условий задачи, т. е. преобразования отображения A , задающего условия задачи в множестве \mathfrak{A} с системой операций C_1, \dots, C_n . Такой операцией является операция переноса, определяемая следующим

образом. Пусть C — любая из операций C_1, \dots, C_n ; $k, r \geq 0$; $C_i^k a = \zeta$, ${}^r b = \xi$, где $a, b \in \mathfrak{M}$. Тогда действие операции переноса

$$C_i^k a \Rightarrow {}^r b \quad (1)$$

на отображение A состоит в том, что в этом отображении соотношение

$$A\xi = \eta$$

заменяется соотношением

$$A\xi = \zeta.$$

Содержательно это означает, что код $C_i^k a$ засылается по адресу ${}^r b$.

Таким образом, применение операции переноса изменяет только одно соотношение отображения, определяющего условия задачи. По правой части операции переноса (1) мы находим преобразуемое соотношение, левая часть этой операции задает новую правую часть преобразованного соотношения.

3. Схемы алгоритмов. Алгоритмический процесс состоит из последовательности применения операций переноса, являющихся единственными допустимыми преобразованиями состояний. При каждом применении операции переноса происходит преобразование отображения A , полученного на предыдущем этапе как результат применения предыдущей операции переноса. Для определения алгоритма необходимо сформулировать принцип построения алгоритмов — порядок действий — так, чтобы на каждом этапе было однозначно определено, какую операцию переноса следует применять. В наиболее гибкой форме порядок действий выражается в *граф-схемах алгоритмов*, предложенных Л. А. Калужниным [1].

Пусть $D = \{D_1, D_2, \dots, D_m\}$ — система операторов действия, преобразующих информацию о задаче, и $P = \{P_1, P_2, \dots, P_r\}$ — система распознавателей свойств информации.

$(D-P)$ -*граф-схемой* называется система точек, соединенных стрелками, обладающая следующими свойствами. Существует точка, называемая входом граф-схемы; в нее не входит ни одна стрелка. Обозначим эту точку буквой **В**. Существует точка, называемая выходом граф-схемы; из нее не выходит ни одна стрелка, обозначим ее знаком **В**. Из остальных точек схемы выходит либо только одна стрелка (такие точки мы будем называть *D-точками*), либо две стрелки: одна с отметкой 1, другая с отметкой 0 (такие точки называются *P-точками*). В любую точку граф-схемы может входить любое число стрелок. Каждой *D-точке* ставится в соответствие некоторый оператор действия D_k ($1 \leq k \leq m$); каждой *P-точке* ставится в соответствие некоторый распознаватель P_s ($1 \leq s \leq r$). Действие $(D-P)$ -граф-схемы как алгоритма осуществляется следующим образом. Из точки входа по стрелке поступает сигнал действия в следующую точку схемы, обозначим ее T_1 . Если точка T_1 есть *D-точка*, то информация о задаче преобразуется согласно описанию соответствующего ей оператора действия, после чего сигнал действия поступает в следующую точку схемы по стрелке, выходящей из точки T_1 . Если точка T_1 есть *P-точка*, то выясняется наличие в информации свойства согласно описанию соответствующего ей распознавателя; если информация обладает искомым свойством, то сигнал действия поступает по стрелке с отметкой 1, в противном случае сигнал действия переходит по стрелке с отметкой 0 и т. д. Окончание работы алгоритма происходит тогда, когда сигнал действия поступает в точку выхода схемы **В**.

$(D-P)$ -граф-схема определяет только порядок действия без каких-либо ограничений на сложность схемы или характер операторов действия и распознавателей.

В качестве операторов действия в нашем случае могут быть взяты операции переноса. Для определения распознавателей воспользуемся классификацией свойств \mathfrak{M} кодов множества \mathfrak{A} , в котором кодируется условие задачи A . Введем операцию разветвления.

Операция разветвления

$$P_{\mu}^k a \begin{array}{l} | \rightarrow \\ \circ \rightarrow \end{array}$$

определяет наличие свойства μ по адресу $^k a$, т. е. принадлежность кода $A^k a$ множеству μ , и передает управление по стрелке $| \rightarrow$, если $A^k a \in \mu$, и по стрелке $\circ \rightarrow$, если $A^k a \notin \mu$.

Определим теперь адресный алгоритм. Фиксируем множество кодов \mathfrak{A} , систему операций в нем C_1, C_2, \dots, C_n и классификацию свойств кодов \mathfrak{M} . Пусть A — произвольное отображение в \mathfrak{A} , принимаемое в качестве исходного условия задачи.

Адресным алгоритмом называется $(D-P)$ -граф-схема, в которой каждой D -точке ставится в соответствие операция переноса, а каждой P -точке — операция разветвления.

Применение адресного алгоритма к исходному отображению A приводит к отображению B , которое называется решением задачи.

Обратим внимание на то, что исходное отображение A и промежуточные результаты преобразования его участвуют в алгоритмическом процессе, однако описание адресных алгоритмов не зависит от вида отображений *), которые преобразуются в алгоритмическом процессе.

Изложенный выше способ описания алгоритмов использует существенные алгоритмические особенности современных ЦАМ. Поэтому следует ожидать, что программирование адресных алгоритмов должно быть несложным (см. § 4).

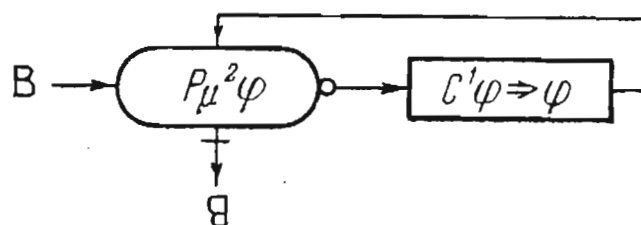
Другой особенностью предлагаемого описания алгоритмов является возможность избежать изменения схемы алгоритма в процессе работы алгоритма, как это имеет место в случае программного описания алгоритмов. Не обсуждая этот вопрос в деталях, приведем один пример. Рассмотрим элементарную алгоритмическую операцию, часто используемую во многих логических (и математических) задачах, — операцию поиска элемента, обладающего данным свойством μ , в последовательности элементов a_1, a_2, \dots . Введем адреса элементов последовательности k_1, k_2, \dots :

$$Ak_i = a_{i+1}, \quad i = 1, 2, \dots,$$

и соотношения порядка между адресами: $Ck_i = k_{i+1}$, $i = 1, 2, \dots$. Если коды k_i являются натуральными числами, то обычно

$$k_{i+1} = k_i + 1 = k_1 + i.$$

Пусть код φ есть адрес первого элемента последовательности a_1 : $A\varphi = k_1$ ($A^2\varphi = a_1$), т. е. φ является адресом 2-го ранга первого элемента a_1 . Граф-схема поиска элемента последовательности, обладающего свойством μ , имеет вид:



*) Содержательно это означает, что запись алгоритма не зависит от исходного распределения памяти, чем по сути дела является исходное отображение.

Обратим внимание на следующие характерные черты данной граф-схемы. Граф-схема поиска не изменяется в процессе работы. Она не зависит также от адресов рассматриваемых элементов, так что она применима и для другой последовательности элементов, имеющих иные адреса. Для сравнения приведем пример программы поиска для трехадресной машины.

$p+1$	ПУ	μ	k_1	n
$p+2$	+	$p+1$	c	$p+1$
$p+3$	БП	—	—	$p+1$

Здесь для простоты первый приказ осуществляет проверку свойства μ содержимого ячейки k_1 , номер которой находится во 2-м адресе приказа; второй приказ изменяет 2-й адрес первого приказа на единицу (c —номер ячейки с соответствующей константой переадресации); третий приказ передает управление первому приказу. При нахождении содержимого со свойством μ поиск оканчивается передачей управления приказу n . Приведенная программа зависит от адресов рассматриваемой последовательности элементов и изменяется в процессе работы. Если же ввести адрес 2-го ранга φ , т. е. считать, что ячейка с номером φ содержит номер ячейки k_1 , содержащей первый элемент последовательности, тогда программа поиска элемента со свойством μ в соответствии с приведенной выше граф-схемой поиска будет:

$p+1$	ПУ	μ	${}^2\varphi$	n
$p+2$	+	φ	c	φ
$p+3$	БП	—	—	$p+1$

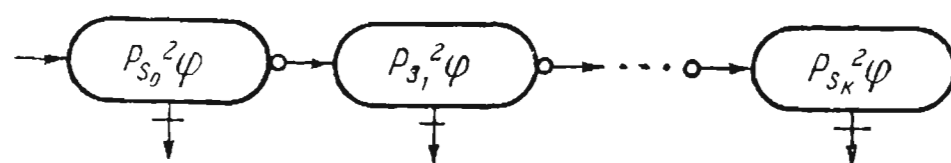
Здесь первый приказ осуществляет проверку свойства μ содержимого по адресу 2-го ранга φ , т. е. содержимого ячейки, номер которой содержится в ячейке φ . Вторым приказом изменяется содержимое ячейки φ на единицу.

§ 3. Адресная интерпретация машины Тьюринга

Можно легко построить интерпретацию машин Тьюринга [4] в виде адресных алгоритмов. Действительно, введем адреса ячеек ленты машины Тьюринга следующим образом. Пусть α_0 —адрес некоторой фиксированной ячейки. Адреса соседних ячеек определим следующими операциями: $S\alpha_0=\alpha_1$, $\bar{S}\alpha_0={}_1\alpha$; α_1 —адрес ячейки, следующей за α_0 , ${}_1\alpha$ —адрес предыдущей ячейки. Читающему элементу машины Тьюринга поставим в соответствие код φ , который в алгоритме будет адресом 2-го ранга. Исходное множество кодов \mathfrak{A} , содержащее адреса ячеек ленты машины Тьюринга и код φ , дополним кодами букв внешнего алфавита машины Тьюринга $s_0, s_1, s_2, \dots, s_k$ и кодами букв q_0, q_1, \dots, q_r , обозначающими состояния машины. Для фиксирования состояния машины в процессе работы введем код q , который служит адресом состояния машины. Функциональная схема машины Тьюринга может быть теперь представлена в виде граф-схемы адресного алгоритма. Условия задачи и начальное состояние машины определяются отображением A в рассматриваемом множестве кодов \mathfrak{A} . Каждому адресу ячейки α ставится в соответствие ее содержимое, т. е. некоторая буква внешнего алфавита s_i : $A\alpha=s_i$. Коду φ соответствует адрес α_0 начальной обозреваемой ячейки: $A\varphi=\alpha_0$; адресу q соответствует начальное состояние q_0 : $Aq=q_0$. Граф-схема адресного алгоритма, соответствующая

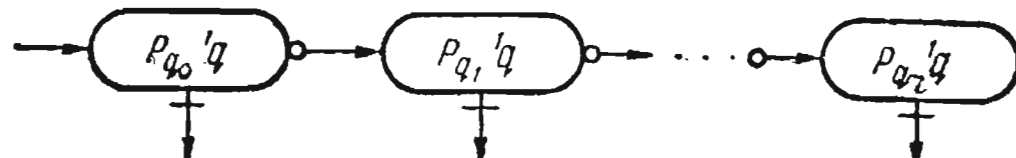
щего функциональной схеме машины Тьюринга, состоит из следующих узлов:

1. Система распознавателей букв внешнего алфавита по адресу 2-го ранга



проверяет, какая буква внешнего алфавита содержится в ячейке, номер которой является содержанием адреса φ .

2. Система распознавателей по адресу q состояний машины



выясняет по адресу q , в каком состоянии находится машина, и передает управление по стрелке с отметкой 1 от распознавателя, соответствующего данному состоянию машины.

Каждый выход системы распознавателей букв внешнего алфавита со стрелкой \rightarrow передает управление на вход системы распознавателей состояния машины. Каждый выход системы распознавателей состояния машины передает управление схеме действия, соответствующей отдельному узлу функциональной схемы машины Тьюринга. Отдельный узел функциональной схемы машины Тьюринга, «приказ», имеет вид

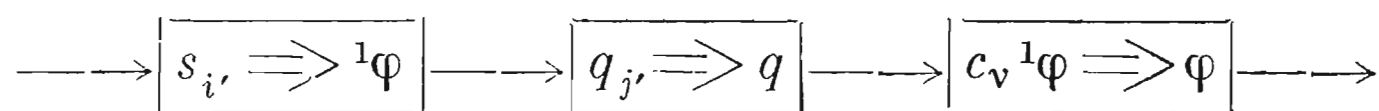
$$s_i q_j | s_i, q_j, c_v,$$

что означает: при наличии буквы s_i внешнего алфавита в ячейке, которая обозревается читающим элементом, и в состоянии q_j машины в обозреваемой ячейке пишется буква $s_{i'}$, машина переходит в состояние $q_{j'}$ и обозревает ячейку с адресом c_n , c_l или c_n (следующую справа, предыдущую слева или ту же самую ячейку).

3. Граф-схема действия, соответствующая приказу машины Тьюринга

$$s_i q_i | s_i, q_j, c_v,$$

имеет вид



Составление полной граф-схемы адресного алгоритма, интерпретирующего машину Тьюринга с заданной функциональной схемой, теперь очевидно. Таким образом, машина Тьюринга может быть представлена в виде адресного алгоритма. Обратное утверждение, что для любого адресного алгоритма может быть построена машина Тьюринга, реализующая адресный алгоритм, также может быть доказано непосредственным построением. Однако это доказательство уже значительно сложнее, так как необходимо интерпретировать в машине Тьюринга введенные выше алгоритмические операции, участвующие в адресном алгоритме. Мы сошлемся здесь лишь на основную гипотезу теории алгоритмов, признание которой означает возможность построения для любого адресного алгоритма соответствующей машины Тьюринга: всякий алгоритм может быть задан посредством тьюринговой функциональной схемы и реализован в соответствующей машине Тьюринга.

Адресная интерпретация машин Тьюринга нам кажется естественным представлением о характере задания информации и способе ее обработки в машинах Тьюринга.

§ 4. Построение алгоритма логической задачи

Рассмотрим пример построения адресного алгоритма для перевода формул с бинарными операциями в запись без скобок. Введем коды. Величины будем кодировать буквами a, b и т. д.; бинарные операции кодируются буквами \circ_1, \circ_2 и т. д.; скобка открывающая кодируется знаком $($, закрывающая—знаком $)$; введем также пустой код «—».

Формулами в записи со скобками (*скобочными формулами*) называются: 1) величины a, b и т. д.; 2) если A и B формулы, то выражение $(A)\circ(B)$ также формула.

В бесскобочной записи формулами называются: 1) величины; 2) если A и B формулы, то и $\circ AB$ также формула. Других формул нет.

Например, формула в скобочной записи с бинарными операциями

$$(((a) + (b)) \times (c)) - ((a) \times ((b) + (c)))$$

в бесскобочной записи имеет вид:

$$-- \times + abc \times a + bc.$$

Переход от записи формул со скобками к бесскобочной записи можно осуществить следующим образом: данная скобочная формула рассматривается как выражение $(A) \circ_k (B)$; т. е. выделяется завершающая операция в формуле; затем переписывается эта формула в запись без скобок $\circ_k AB$. Далее, каждое выражение A и B является в свою очередь формулами вида $(A_1) \circ_i (A_2)$ и $(B_1) \circ_j (B_2)$. Они переписываются в запись без скобок и т. д. Приведенный пример формулы преобразуется в бесскобочную запись следующим образом:

1-й шаг: $--((a) + (b)) \times (c) (a) \times ((b) + (c))$

2-й шаг: $-- \times (a) + (b)c \times a(b) + (c)$

3-й шаг: $-- \times + abc \times a + bc$ —искомый результат.

Алгоритм разобьем на две части:

1. Поиск завершающей операции.

2. Переход к бесскобочной записи формулы.

Для построения первой части алгоритма заметим, что перед завершающей операцией число открывающих скобок равно числу закрывающих скобок. Исходная информация о задаче кодируется и определяется следующим отображением:

$$A\alpha_k = a_k, \quad k = 1, 2, \dots;$$

α_k — адрес k -й буквы в формуле.

$$A\alpha = \alpha_1;$$

α — адрес 2-го ранга первой буквы.

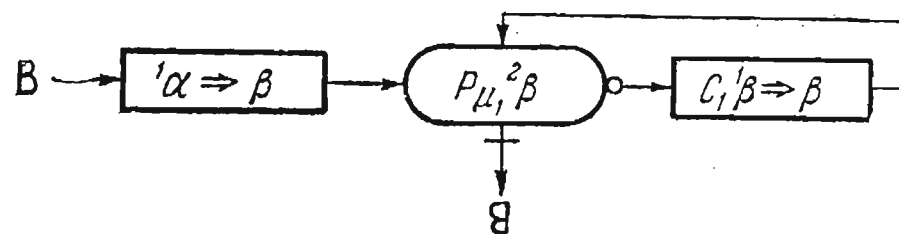
$$C_1\alpha_k = \alpha_{k+1},$$

$$\bar{C}_1\alpha_k = \alpha_{k-1}, \quad k = 1, 2, \dots$$

} Операции, задающие порядок просмотра адресов букв.

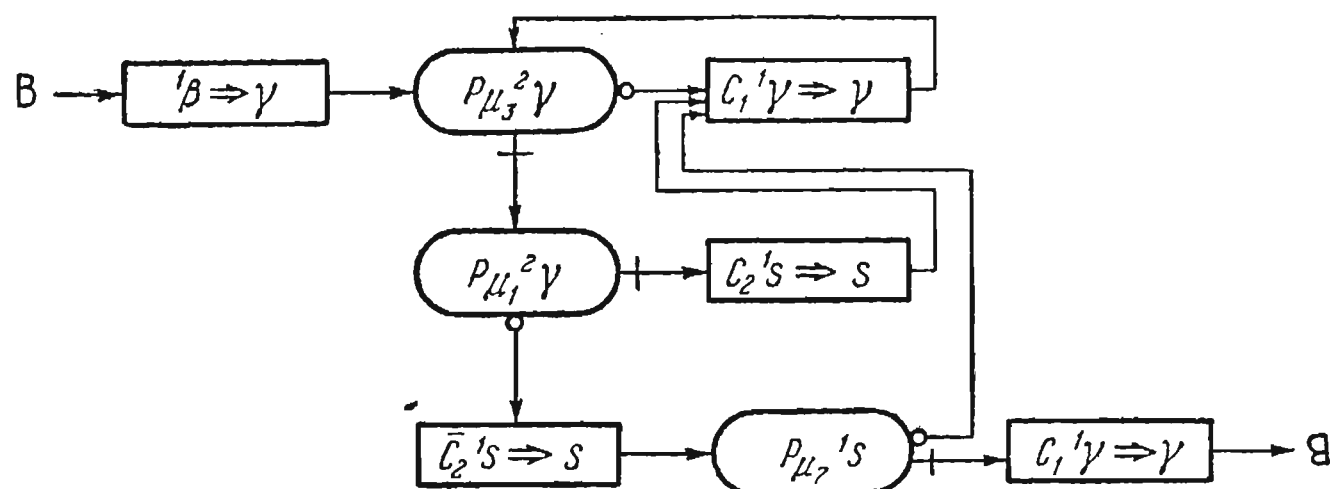
Введем еще вспомогательные коды. Код $!$ ставится в конце формулы; код s — адрес числа открывающих и закрывающих скобок в формуле; целые числа n будем использовать для счета числа скобок. Операции счета целых чисел обозначим C_2 ($C_2n = n + 1$) и \bar{C}_2 ($\bar{C}_2n = n - 1$). Отображение A дополним соотношением $As = 0$, определяющим начальное заполнение счетчика скобок. Введем классификацию свойств кодов: μ_1 — открывающая скобка; μ_2 — закрывающая скобка; μ_3 — открывающая или закрывающая скобка; μ_4 — величины; μ_5 — знаки операций; μ_6 — заключительный знак $!$; μ_7 — число нуль. Строим теперь граф-схему алгоритма.

Поиск первой открывающей скобки в формуле осуществляется по схеме



Операция ${}^1\alpha \Rightarrow \beta$ заполняет адрес β : $A\beta = A\alpha = \alpha_1$. Затем проверяется свойство μ_1 — открывающая скобка — по адресу β . Если содержимое ячейки β обладает свойством μ_1 (или, как мы будем говорить, μ_1 выполняется), то управление передается на выход схемы; в противном случае содержимое β сдвигается на единицу: $A\beta = \alpha_2$.

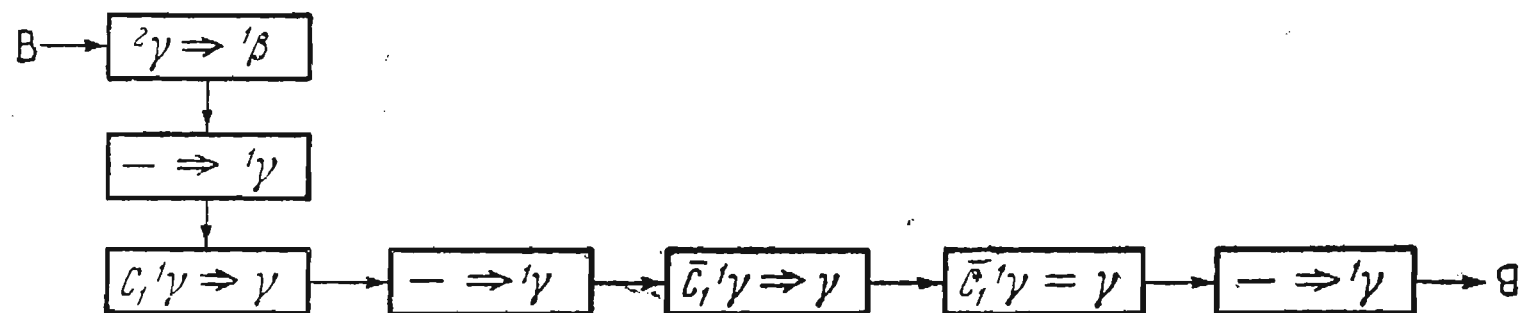
Выход этой схемы соединяется со входом схемы, осуществляющей счет числа открывающих и закрывающих скобок в формуле после ${}^1\beta$:



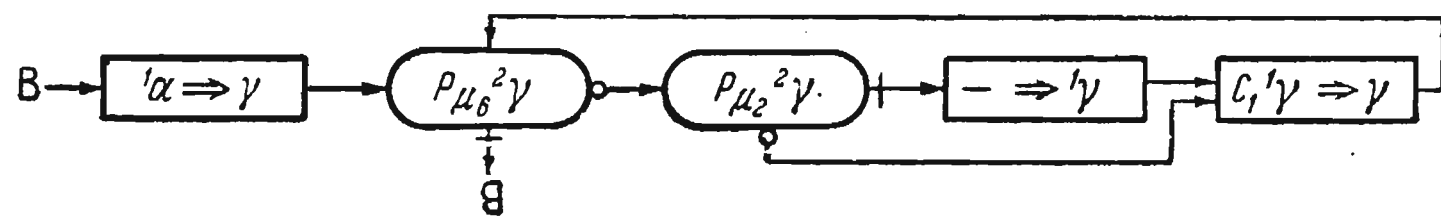
Операция ${}^1\beta \Rightarrow \gamma$ заполняет адрес γ : $A\gamma = A\beta$. Затем проверяется свойство μ_3 — скобка — по адресу γ и передается управление проверке свойства μ_2 — открывающая скобка — по γ , если μ_3 выполняется, или к сдвигу содержимого γ , если μ_3 не выполняется. Сдвиг γ производится операцией $C_1{}^1\gamma \Rightarrow \gamma$, после чего снова проверяется свойство μ_3 по γ операцией $P_{\mu_3}{}^2\gamma$. Проверка свойства μ_1 — открывающая скобка — по γ осуществляется операцией $P_{\mu_1}{}^2\gamma$. Если μ_1 выполняется, то прибавляется единица в s операцией $C_2{}^1s \Rightarrow s$; если μ_1 не выполняется, то вычитается единица в s операцией $\bar{C}_2{}^1s \Rightarrow s$. После вычитания единицы в s проверяется свойство μ_7 — нуль в s . Если μ_7 выполняется, то следующий элемент в формуле является завершающей операцией данной формулы, и поиск закончен; если μ_7 не выполняется, то поиск продолжается переходом к сдвигу γ . После прибавления единицы также происходит переход к сдвигу γ .

Вторую часть алгоритма — переход к записи без скобок — осуществим в два этапа. На первом шаге происходит переход от выражения $(A) \circ (B)$ к выражению $\circ A \text{ — — — } B$. После этого происходит стирание всех оставшихся закрывающих скобок (замена кодов этих скобок кодом «—»). Признаком окончания первого этапа является отсутствие открывающих скобок в формуле; поиск открывающих скобок ведется до тех пор, пока по адресу β не будет обнаружен знак ! (свойство μ_6 — конец формулы). Проверка этого свойства, т. е. операция $P_{\mu_6}{}^2\beta$, ставится после сдвига β . Граф-схема второй части алгоритма не требует специальных пояснений.

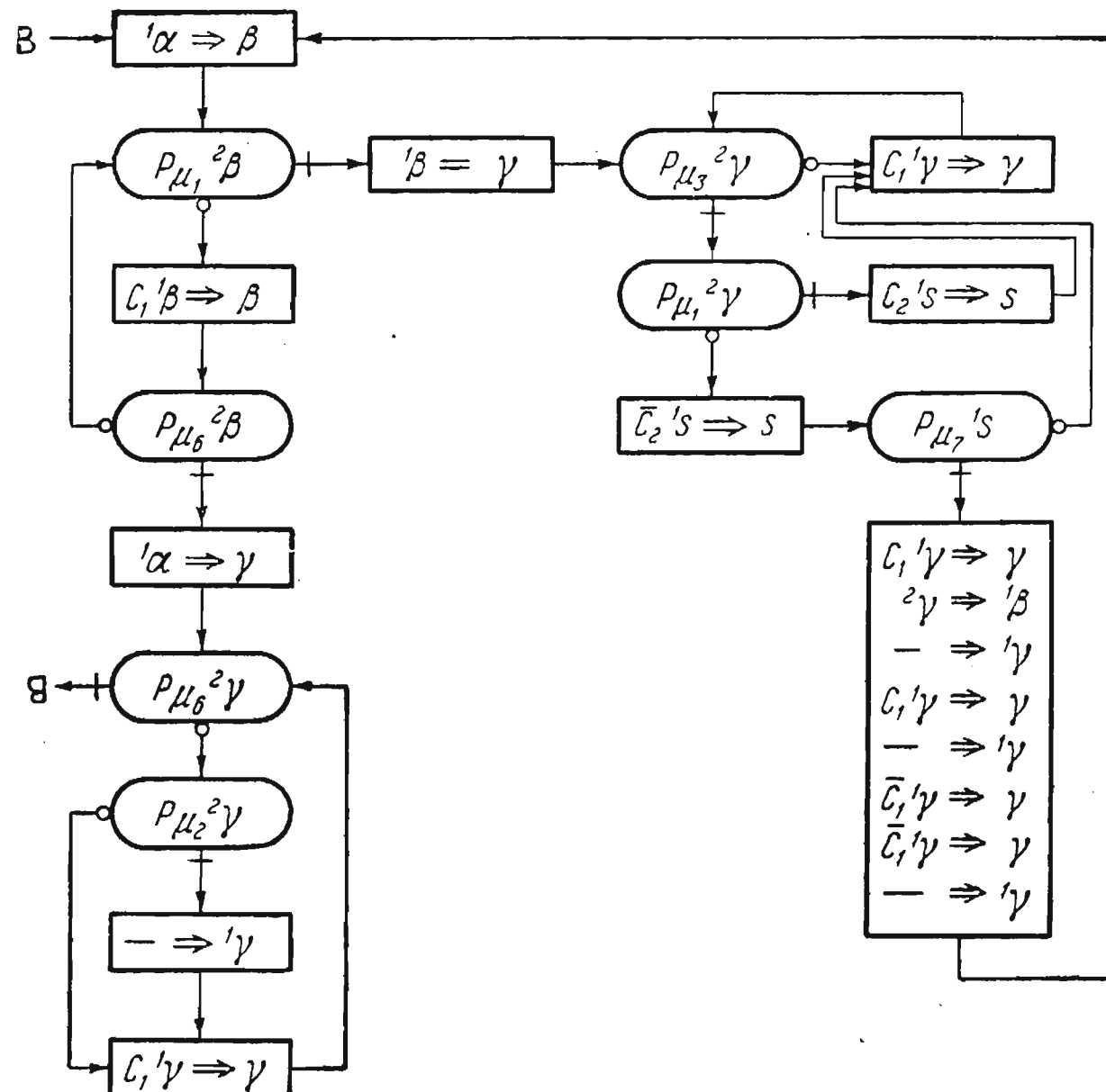
Переход от формулы $(A) \circ (B)$ к выражению $\circ A \text{ — — — } B$:



Стирание оставшихся закрывающих скобок:



Полная граф-схема алгоритма приведения формул с бинарными операциями в запись без скобок имеет вид:



§ 5. Программирование основных алгоритмических операций

В примерах, приведенных в предыдущем параграфе, в алгоритмических операциях используются только адреса 1-го и 2-го ранга. Анализ алгоритмов других более сложных логических задач показывает, что уже использование адресов 1-го и 2-го рангов создает возможности довольно простого описания алгоритмов сложных логических задач и при этом граф-схемы алгоритмов не изменяются в процессе работы алгоритмов.

С другой стороны, переход от записи граф-схемы алгоритма, использующей адреса 2-го ранга, к программе алгоритма в приказах существующих универсальных цифровых автоматических машин не представляется сложной задачей. В настоящем параграфе рассмотрим вопрос программирования алгоритмических операций, использующих адреса 1-го и 2-го рангов, при условии, что они являются обычными адресами ячеек запоминающего устройства машины. Иначе говоря, предполагается, что отдельные элементы информации о задаче кодируются в отдельных ячейках ЗУ. Естественно, что программы рассматриваемых алгоритмических операций значительно усложняются, если в каждой ячейке ЗУ может помещаться несколько кодов, т. е. когда адреса 1-го и 2-го рангов указывают не только номер ячейки ЗУ, но и часть ячейки, содержащей отдельный код.

Будем предполагать машину трехадресной. Предположим для определенности, что если ячейка служит адресом 2-го ранга, т. е. ее содержимым

является адрес 1-го ранга, то содержащийся в ней код адреса 1-го ранга помещается в разрядах, соответствующих первому адресу в трехадресном приказе. Остальные разряды—место кода операции и места кодов 2-го и 3-го адресов—содержат нули, т. е. если содержимое адреса 2-го ранга α есть α , то ячейка ϕ имеет вид: $\phi \mid 0 \dots 0 \mid \alpha \mid 0 \dots 0 \mid 0 \dots 0 \mid$

В дальнейшем нам понадобятся вспомогательные константы. Введем для них следующие обозначения. Константу переадресации, изменяющую первый адрес приказа на одну единицу, т. е. содержащую единицу в первом адресе, обозначим

$$\delta' \mid \begin{array}{|c|c|c|c|} \hline - & 1 & - & - \\ \hline \end{array}$$

Константы, содержащие единицы в разрядах кода операций 1-го, 2-го или 3-го адресов, обозначим буквой δ с двоичным номером, состоящим из четырех разрядов, причем единицы ставятся в разрядах, соответствующих отдельным частям приказа. Так, например, константа, содержащая единицы в разрядах кода операций и в разрядах 3-го адреса, имеет номер 1001; константа, содержащая единицы только в разрядах 1-го и 3-го адресов приказа, имеет номер 0101.

Будем использовать следующий набор операций трехадресной машины:

$$\mid \begin{array}{|c|c|c|c|} \hline + & a & b & c \\ \hline \end{array}$$

—сложить число из ячейки a с числом из ячейки b и результат поместить в ячейку c .

$$\mid \begin{array}{|c|c|c|c|} \hline - & a & b & c \\ \hline \end{array}$$

—вычесть из числа, содержащегося в ячейке a , число из ячейки b и результат поместить в ячейку c .

$$\mid \begin{array}{|c|c|c|c|} \hline \wedge & a & b & c \\ \hline \end{array}$$

—поразрядное логическое умножение числа из ячейки a на число из ячейки b , результат помещается в c .

$$\mid \begin{array}{|c|c|c|c|} \hline \rightarrow & n & a & c \\ \hline \end{array}$$

—сдвиг содержимого ячейки a вправо на число разрядов n ; в частности, запись содержимого a в c (при $n=0$).

Операция

$$\mid \begin{array}{|c|c|c|c|} \hline \rightarrow & 2A & a & c \\ \hline \end{array}$$

означает сдвиг содержимого ячейки a вправо на число разрядов, равное числу разрядов двух адресов приказа.

$$\mid \begin{array}{|c|c|c|c|} \hline \leftarrow & n & a & c \\ \hline \end{array}$$

—сдвиг содержимого ячейки a влево на число разрядов n ; в частности, запись содержимого a в c (при $n=0$).

Операция

$$\mid \begin{array}{|c|c|c|c|} \hline \leftarrow & 2A & a & c \\ \hline \end{array}$$

означает сдвиг влево содержимого ячейки a на число разрядов, равное числу разрядов двух адресов приказа.

Заметим, что мы не делаем различия между сложением чисел и сложением, изменяющим адреса приказов. Если в машине эти приказы различаются, то соответствующие изменения необходимо ввести в приводимые ниже программы.

Условимся адрес φ второго ранга кода a (${}^2\varphi = a$) называть фиксатором кода a . Иначе говоря, содержимым фиксатора φ кода a является адрес кода a :

$${}^1\varphi = b, \quad {}^1b = a.$$

Программы алгоритмических операций.

1. Увеличение содержимого фиксатора на единицу:

операция $C {}^1\varphi \Rightarrow \varphi$; программа

+	φ	δ'	φ
---	-----------	-----------	-----------

.

2. Перенос содержимого фиксатора φ в адрес ψ :

операция ${}^1\varphi \Rightarrow \psi$; программа

+	φ	-	ψ
---	-----------	---	--------

.

3. Перенос содержимого фиксатора φ со сдвигом на единицу вперед (назад) в адрес ψ :

операция $C {}^1\varphi \Rightarrow \psi$; программа

+	φ	δ'	ψ
---	-----------	-----------	--------

;

операция $\bar{C} {}^1\varphi \Rightarrow \psi$; программа

-	φ	δ'	ψ
---	-----------	-----------	--------

.

4. Перенос кода b , фиксатор которого φ , в адрес a (чтение по φ в a):

операция ${}^2\varphi \Rightarrow a$; программа

$k+1$	\wedge	$k+3$	δ_{1001}	$k+3$
$k+2$	+	$k+3$	φ	$k+3$
$k+3$	+			a

5. Перенос кода, адрес которого a , по адресу 2-го ранга φ (запись из a по φ):

операция ${}^1a \Rightarrow {}^1\varphi$; программа

$k+1$	\wedge	$k+4$	δ_{1100}	$k+4$
$k+2$	\rightarrow	$2A$	φ	c
$k+3$	+	$k+4$	c	$k+4$
$k+4$	+	a		

6. Перенос кода, фиксатор которого φ , по адресу 2-го ранга ψ :

операция ${}^2\varphi \Rightarrow {}^1\psi$; программа

$k+1$	\wedge	$k+5$	δ_{1000}	$k+5$
$k+2$	+	$k+5$	φ	$k+5$
$k+3$	\rightarrow	$2A$	ψ	c
$k+4$	+	$k+5$	c	$k+5$
$k+5$	+			

Последние три операции, в которых используются адреса 2-го ранга, содержат по несколько приказов. Следовательно, если в алгоритме ис-

пользуются часто адреса 2-го ранга, то соответствующие приказы программы приходится многократно повторять. Этого можно избежать, построив стандартные программы переноса, информацию для которых задавать в некоторых стандартных ячейках. При этом число приказов в стандартных программах увеличится, что приведет к увеличению числа тактов работы машины, но зато в программе алгоритма соответствующие алгоритмические операции могут быть запрограммированы одним приказом, осуществляющим перенос информации об операции в стандартную ячейку и передачу управления соответствующей стандартной программе алгоритмической операции.

Мы не приводим здесь программы операции передачи управления P_{μ^2a} , так как программа передачи управления по свойству содержимого адреса 1-го ранга очевидна, а программирование передачи управления по свойству содержимого адреса 2-го ранга может быть сведено к программе чтения в адрес 1-го ранга и программе передачи управления по адресу 1-го ранга.

§ 6. Требования к алгоритмической машине

Как показывает анализ алгоритмов программирующей программы, алгоритмов приведения формул исчисления высказываний к нормальной форме, дифференцирования функций, заданных в элементарных операциях, и др., использование адресов 2-го ранга значительно облегчает описание таких алгоритмов, а значит, обмен опытом алгоритмизации. Следует ожидать, что расширение использования современных цифровых автоматических машин для осуществления сложных алгоритмов потребует конструктивного усовершенствования машин. В этом отношении нам представляется целесообразным введение в машину адресов 2-го и старших рангов. Эта задача может быть технически решена различными способами. Например, можно предусмотреть возможность использования в машине каждой ячейки оперативного запоминающего устройства как адреса 1-го, 2-го и т. д. рангов. Для этого в приказах такой машины должны быть выделены разряды для указания ранга адреса. Тогда любая операция в машине может в зависимости от этого указания рассматривать номер ячейки оперативного ЗУ как адрес того или иного ранга. Такой подход требует увеличения числа разрядов в приказах машины по сравнению с обычным или же сокращения объема ЗУ по сравнению с возможностями, которые обеспечиваются разрядностью приказов. Может быть предложено другое решение задачи введения в машину адресов старших рангов. Следует отметить, что адреса 2-го ранга и старших рангов преимущественно используются в операциях переноса. Поэтому можно ввести различные операции переноса, которые рассматривают адреса как адреса различных рангов. Остальные операции машины могут оперировать с адресами только 1-го ранга. Можно, наконец, выделить различные массивы ячеек оперативного ЗУ под адреса различных рангов. При этом важно определить соотношение между количествами ячеек, которые отводятся под адреса различных рангов. Последняя задача может быть решена по мере накопления опыта алгоритмизации задач.

Подробный анализ различных путей конструирования алгоритмической машины должен быть предметом специального исследования. Как показывают предварительные исследования возможностей реализации адресов 2-го ранга в ЦАМ, проведенные Е. А. Шкабарой, незначительное изменение обычного устройства управления вычислительной машины позволяет ввести специальные операции, в которых адрес рассматривается как адрес 2-го ранга.

В заключение выражаю глубокую благодарность Л. А. Калужнину, А. П. Ершову и Е. Л. Ющенко, беседы с которыми способствовали существенному улучшению содержания настоящей статьи.

ЛИТЕРАТУРА

- [1] К а л у ж н и н Л. А., Об алгоритмизации математических задач. Сб. Проблемы кибернетики, вып. 2, М., Физматгиз, 1959, стр. 51—67.
- [2] М а р к о в А. А., Теория алгорифмов, Труды Матем. ин-та им. В. А. Стеклова, т. 42, 1954.
- [3] К о л м о г о р о в А. Н., О понятии алгоритма, УМН, т. 8, в. 4 (1953), стр. 175.
- [4] П е т е р Р., Рекурсивные функции. М., ИЛ, 1954.

Поступило в редакцию 11 IV 1958.

V. ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ

ЛОГИЧЕСКИЕ МЕТОДЫ УСКОРЕНИЯ УМНОЖЕНИЯ В ЦИФРОВЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИНАХ

М. А. КАРЦЕВ

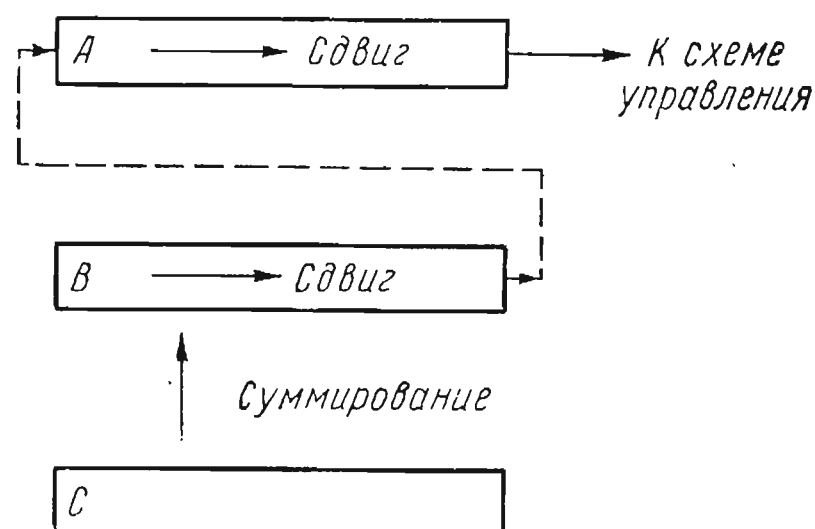
(МОСКВА)

1. Введение

Одной из основных операций для электронных цифровых вычислительных машин является операция умножения. Хотя при решении многих задач сложение и вычитание приходится выполнять в общем чаще, чем умножение, в большинстве случаев оказывается, что главную часть времени машина занята выполнением умножений. Это связано с тем, что длительность выполнения одного умножения больше длительности выполнения одного сложения или вычитания.

В течение ряда лет усилия исследователей и конструкторов электронных цифровых машин направлены на разработку наиболее рациональных методов выполнения умножения и на отыскание других средств (в частности, подходящих способов кодирования чисел), позволяющих ускорить эту операцию. В настоящей статье рассматривается влияние выбора основания позиционной системы счисления на эффективность применения логических методов ускорения умножения.

Предполагается, что умножение выполняется по схеме, которая чаще всего применяется в электронных цифровых машинах (см. рисунок).



Арифметическое устройство содержит 3 числовых регистра (A , B и C) и сумматор. К началу умножения множимое устанавливается в регистре C , множитель—в регистре A ; регистр частичных произведений (B) гасится, т. е. устанавливается на нуль.

Если никакие логические методы ускорения умножения не применяются, то умножение производится следующим образом. Сначала в схему управления поступает младший разряд множителя. Множимое добавляется к числу, находящемуся в регистре B , столько раз, сколько единиц содержится в этой цифре; при этом в регистре B образуется частичное произведение множимого на младшую цифру множителя. Затем выполняется

сдвиг на 1 разряд вправо в регистрах A и B . После выполнения сдвига к схеме управления оказывается присоединенным следующий по весу разряд множителя; в регистре B находится первое частичное произведение, разделенное на n (n —основание системы счисления). Младшая цифра этого частичного произведения, которая при выполнении сдвига вправо в регистре выходит за пределы регистра, является готовой младшей цифрой результата; при сдвиге она может передвигаться в освобождающийся слева разряд регистра A . Далее множимое добавляется к содержимому регистра B столько раз, сколько единиц содержится во второй цифре множителя, снова производится сдвиг вправо на 1 разряд в регистрах A и B и т. д. Если исходные числа содержат по m разрядов, то процесс этот повторяется m раз. В итоге в регистре B образуются m старших разрядов произведения, а в регистре A — m младших разрядов.

Предположим, что в качестве цифр системы счисления с основанием n выбраны целые числа $0, 1, 2, \dots, n-1$. Очевидно, что максимальное количество тактов сложения, необходимое при выполнении умножения m -разрядных чисел описанным методом, равно $m(n-1)$. При сравнении различных оснований системы счисления n следует учитывать, что количество разрядов m , с которыми необходимо оперировать для достижения заданной точности, обратно пропорционально $\log n$. Удобным критерием оценки могла бы служить, например, функция

$$f_1(n) = \frac{n-1}{\log n} : \frac{1}{\log 2} = \frac{n-1}{\log_2 n},$$

показывающая, во сколько раз количество тактов сложения, необходимых при выполнении умножения в системе счисления с основанием n , больше, чем количество тактов сложения, необходимых при выполнении умножения в двоичной системе при одинаковой точности. С ростом n функция $f_1(n)$ монотонно возрастает. В частности, для десятичной системы $f_1(10) \cong 2,71$.

При вычислении полного времени умножения иногда необходимо учитывать также и такты сдвига, которых должно быть столько, сколько разрядов содержат перемножаемые числа.

2. Логические методы ускорения умножения

Если не говорить о возможности усовершенствования физических элементов, используемых в схеме вычислительной машины, то всякое увеличение скорости операций при заданной системе кодирования чисел и заданной точности достигается, конечно, за счет увеличения количества оборудования. Логическими методами (в отличие от аппаратных) мы будем называть те методы ускорения умножения, для которых количество дополнительного оборудования не зависит от количества разрядов в сомножителях. Иными словами, ниже рассматриваются те методы, которые позволяют ускорить умножение путем усложнения только схемы управления, но без увеличения количества оборудования в основной схеме арифметического устройства, показанной на рисунке.

а) Сокращение среднего времени выполнения умножения проще всего достигается при таком построении устройства управления, когда в цикле умножения на некоторый разряд множителя отводится не максимально возможное количество тактов сложения ($n-1$), а ровно столько, сколько единиц содержится в цифре данного разряда. Если все значения цифр в любом из разрядов множителя равновероятны, то среднее количество тактов сложения, необходимых в процессе выполнения умножения, вдвое меньше, чем максимальное. Сокращение количества тактов сложе-

ния вдвое при использовании этого метода получается при любом основании системы счисления n .

б) Для больших оснований системы счисления ($n > 2$) можно применить, однако, и несколько более сложный логический метод ускорения умножения.

Пусть цифра некоторого разряда множителя n_i превышает $n/2$. Представив число n_i в виде $n_i = n - (n - n_i)$, можно при умножении на данный разряд множителя произвести вместо n_i добавлений множимого $n - n_i$ вычитаний; величину $+n$ нужно будет затем учесть в следующем цикле умножения путем увеличения на 1 очередной цифры множителя. Ясно, что при этом среднее количество тактов сложения или вычитания, необходимых в процессе выполнения умножения m -разрядных чисел, для четного n окажется равным $(m/4)n$, а для нечетного $n - (m/4)(n - 1/n)$. Этот метод эквивалентен использованию в качестве цифр системы счисления с основанием n вместо чисел $0, 1, 2, \dots, (n-1)$ соответственно либо чисел $-(n-1)/2, -(n-3)/2, \dots, -1, 0, +1, \dots, +(n-1)/2$ для нечетного n , либо чисел $-(n/2) + 1, -(n/2) + 2, \dots, -1, 0, +1, \dots, (n/2) - 1, n/2$ для четного n .

В частности, для десятичной системы получаем при использовании этого метода в среднем 2,5 m тактов сложения или вычитания на выполнение одного умножения. Для двоичной системы этот метод использовать нельзя, так как среди цифр двоичной системы (0 и 1) нет таких, для которых выполняется условие $n_i > (n/2)$. Поэтому, сопоставляя результат, полученный для десятичной системы при использовании метода б), с тем результатом, который был получен для двоичной системы по методу а), найдем, что среднее количество тактов сложения-вычитания при выполнении умножения в десятичной системе всего примерно в 1,5 раза больше, чем в двоичной системе. Если при этом учесть, что количество тактов сдвига, необходимых для выполнения умножения в десятичной системе, примерно втрое меньше, чем в двоичной системе (потому что при одинаковой точности необходимо иметь двоичных разрядов в $\log_2 10$ раз больше, чем десятичных), то создается впечатление, что в десятичной системе достигается примерно такая же скорость, как в двоичной.

в) С другой стороны, для двоичной системы предлагался также следующий логический метод ускорения умножения.

Пусть в схему управления поступает не одна младшая цифра из регистра A , а одновременно q очередных цифр. Перед 1-м, $(q+1)$ -м, $(2q+1)$ -м и т. д. циклами умножения будем расшифровывать комбинацию, имеющуюся в q очередных разрядах множителя, и вырабатывать на q циклов вперед программу сложения или вычитания с таким расчетом, чтобы количество тактов сложений и вычитаний при выполнении этих циклов было минимально. При этом имеется возможность, если необходимо, увеличить на единицу цифру в $(q+1)$ -м разряде (т. е. прибавить единицу младшего разряда к следующей группе из q разрядов).

Например, при $q=4$ комбинация $\cdot 0111$ расшифровывается в виде

$$\begin{aligned} \cdot 0111 &= \cdot 1000 - \\ &\quad - \cdot 0001, \end{aligned}$$

что означает «вычесть множимое в 1-м из четырех ближайших циклов умножения и добавить его в 4-м цикле»; комбинация $\cdot 1111$ расшифровывается в виде

$$\begin{aligned} \cdot 1111 &= 1 \cdot 0000 - \\ &\quad - \cdot 0001, \end{aligned}$$

что означает «вычесть множимое в 1-м из четырех ближайших циклов умножения и увеличить на единицу следующую четверку разрядов множителя» и т. д. Если бы мы пользовались только методом ускорения умножения а), то в первой из приведенных комбинаций нам пришлось бы затратить в течение 4 циклов умножения 3 такта сложения (вместо 1 такта вычитания и 1 такта сложения), во второй комбинации — 4 такта сложения (вместо 1 такта вычитания).

В предыдущей работе автора*) показано, что при использовании описанного метода среднее количество тактов сложения или вычитания, приходящееся на 1 двоичный разряд множителя, оказывается равным

$$\frac{A_q}{q} = \frac{1}{3} + \frac{1 - (-2)^{-q}}{9q} \quad (1)$$

(здесь A_q — среднее количество тактов сложения или вычитания, необходимых при выполнении q циклов умножения, если расшифровываются группы из q разрядов). В частности, при $q=2$ и $q=3$ $A_2/2 = A_3/3 = 0,375$, при $q=4$ $A_4/4 \cong 0,359$, в дальнейшем с ростом q величина A_q/q убывает очень медленно и $\lim_{q \rightarrow \infty} A_q/q = 0,333...$

Таким образом, расшифровывая большие группы разрядов множителя, можно получить среднее количество тактов сложения или вычитания при выполнении умножения m -разрядных двоичных чисел близким к $m/3$. При равной точности это примерно в 2,3 раза меньше, чем в десятичной системе с использованием метода б).

В действительности, однако, рассматриваемые результаты несопоставимы, так как и в десятичной системе в принципе возможно применение логических методов ускорения умножения, аналогичных методу в) для двоичной системы.

Ниже исследуется возможность сокращения среднего количества тактов сложения или вычитания при использовании логических методов ускорения умножения в позиционной системе счисления с произвольным основанием n и сравниваются с этой точкой зрения различные значения величины n .

3. Формулировка задачи

Пусть умножение чисел, представленных в позиционной системе счисления с основанием n , выполняется в арифметическом устройстве, которое построено в соответствии с рисунком. Перед 1-м, $(q+1)$ -м, $(2q+1)$ -м и т. д. циклами умножения в устройстве управления расшифровывается группа из q очередных разрядов множителя и на q циклов вперед вырабатывается программа сложений или вычитаний множимого с таким расчетом, чтобы минимизировать необходимое количество тактов сложения-вычитания.

Полагая, что все комбинации цифр равновероятны, нужно определить математическое ожидание $A_q(n)$ количества тактов сложения-вычитания, которые при этом потребуются для выполнения q циклов умножения, и среднее количество тактов сложения-вычитания, приходящихся на 1 разряд множителя, $A_q(n)/q$. При этом формула (1) будет получена как частный случай решения для $n=2$.

Далее, для оценки того или другого основания системы счисления n с точки зрения скорости выполнения умножения можно будет восполь-

*) М. А. Карцев, «Арифметические устройства электронных цифровых машин», Физматгиз, Москва, 1958 г.

зоваться функцией

$$f_{\infty}(n) = \left[\lim_{q \rightarrow \infty} \frac{A_q(n)}{A_q(2)} \right] \frac{1}{\log_2 n},$$

показывающей, во сколько раз количество тактов сложения-вычитания, необходимых при выполнении умножения в позиционной системе с основанием n при использовании наиболее сильных логических методов ускорения, больше, чем аналогичная величина в двоичной системе при условии, что точность вычислений одинакова.

Строго говоря, переход к пределу в оценочной функции $f_{\infty}(n)$ не вполне правомерен, так как величина q не может возрастать неограниченно. Прежде всего она не может быть, конечно, больше, чем количество разрядов m в перемножаемых числах. Но если бы мы выбрали $q=m$, то рассматриваемый метод ускорения нельзя было бы причислять к логическим методам, так как количество добавочного оборудования (в дешифраторе комбинаций) сильно зависело бы от m . Однако, как мы увидим из дальнейшего (и как это было видно в разделе 2, в) для случая $n=2$), величина $A_q(n)/q$ уже при сравнительно небольших значениях q мало отличается от предельного значения. С другой стороны, было бы неправильно пользоваться вместо функции $f_{\infty}(n)$ функциями вида $f_q(n) = (A_q(n)/A_q(2)) \cdot (1/\log_2 n)$, так как оборудование в схеме управления, необходимое для расшифровки групп из q разрядов множителя в системе счисления с основанием n , может резко отличаться от количества оборудования, необходимого для расшифровки групп из q двоичных разрядов.

В более общей форме поставленная задача может быть сформулирована следующим образом. Имеется набор чисел $0, 1, 2, \dots, n^q - 1$ (n и q — целые числа). Каждое из чисел α этого набора представлено в виде

$$\alpha = \sum_{i=1}^q a_i n^{i-1} + a_0 n^q,$$

где $a_0 = 0, 1$; a_i — целые числа, положительные или отрицательные; коэффициенты a_0 и a_i выбраны так, чтобы величина $\sum_{i=1}^q |a_i|$ была минимальна.

Требуется определить математическое ожидание $A_q(n)$ величины $\sum_{i=1}^q |a_i|$, а также отношение $A_q(n)/q$, если все числа $0, 1, 2, \dots, (n^q - 1)$ равновероятны.

На самом деле, как ясно из предыдущего, в рассматриваемый набор чисел следовало бы включить также число n^q , причем суммарная вероятность чисел 0 и n^q должна была бы приниматься равной вероятности любого из остальных чисел. Но это обстоятельство не играет никакой роли, потому что и для числа 0 и для числа n^q имеем $\sum_{i=1}^q |a_i| = 0$.

4. Вычисление величины $A_q(n)$ для четного n

В разделе 2,б) нами была подсчитана непосредственно величина $A_q(n)$ для случая $q=1$, которая при четном n оказалась равной

$$A_1(n) = \frac{n}{4}.$$

Сравним далее этот случай со случаем расшифровки пар разрядов множителя ($q=2$).

Если бы каждый из пары разрядов множителя расшифровывался в отдельности, то математическое ожидание количества тактов сложения-вычитания при выполнении двух циклов умножения равнялось бы $2A_q(n)$. Однако при расшифровке пар разрядов множителя можно выиграть по одному такту сложения-вычитания в половине из тех комбинаций, в которых правая цифра расшифровываемой пары равна $n/2$.

Когда расшифровывается один разряд множителя, то при наличии в нем цифры $n/2$ безразлично, как вести ближайший цикл умножения: добавлять ли множимое $n/2$ раз, ничего не перенося в следующий разряд множителя, или вычитать множимое $n/2$ раз и переносить единицу в следующий разряд множителя. Например, в разделе 2, б) мы условились в этом случае всегда добавлять множимое $n/2$ раз.

Если теперь расшифровываются пары разрядов множителя, то в тех комбинациях, когда правая цифра равна $n/2$, а левая меньше $n/2$, выгодно в первом цикле добавлять множимое $n/2$ раз, когда же левая цифра равна или больше $n/2$ —вычитать. В первом случае в левый разряд ничего не переносится, и количество тактов сложения-вычитания такое же, как при расшифровке каждого разряда в отдельности; во втором случае увеличение на единицу левой цифры расшифровываемой пары приводит к уменьшению на единицу количества тактов вычитания при выполнении второго из пары циклов умножения.

Например, в десятичной системе при расшифровке каждого разряда в отдельности комбинация «45» представляется в виде $45 = 4 \cdot 10 + 5$, а комбинация «65»—в виде $65 = (10 - 4) \cdot 10 + 5$; в обоих случаях получаем по 9 тактов сложения-вычитания при выполнении пары циклов умножения (для комбинации «45»—5 тактов сложения в первом цикле и 4 такта сложения во втором; для комбинации «65»—5 тактов сложения в первом цикле и 4 такта вычитания во втором). Если же расшифровываются пары разрядов, то для комбинации «45» выгодно сохранить прежнюю расшифровку, а комбинацию «65» представить в виде $65 = (10 - 3) \cdot 10 - 5$, в результате чего количество тактов сложения-вычитания при этой комбинации уменьшится на 1.

Так как суммарная вероятность комбинаций, в которых при расшифровке пар разрядов множителя можно сэкономить по 1 такту сложения-вычитания, равна $1/2 \cdot 1/n$, то

$$A_2(n) = 2A_1(n) - \frac{1}{2n} = \frac{n}{2} - \frac{1}{2n}.$$

Пусть теперь нам известна величина $A_{p-1}(n)$, найдем разность $A_p(n) - A_{p-1}(n)$.

Будем полагать, что расшифровываемая группа из p цифр получена в результате того, что к соответствующей группе из $p-1$ цифр была приписана одна цифра слева. Кроме того, предположим, что соответствующая группа из $p-1$ цифр была расшифрована оптимальным образом. Далее, для различных случаев мы вычислим, сколько лишних тактов сложения-вычитания потребуется при умножении на данную группу из p разрядов по сравнению с соответствующей группой из $p-1$ разрядов.

Рассмотрим сначала те комбинации из p цифр, в которых левая цифра есть 0.

Прежде всего очевидно, что во всех тех случаях, когда следующая (вторая слева) цифра меньше, чем $n/2$, количество тактов сложения-вычитания при умножении на p цифр будет таким же, как при умножении на соответствующую группу из $p-1$ цифр.

Легко убедиться также, что если вторая слева цифра больше $n/2$, то количество тактов сложения-вычитания при переходе от группы из

$p-1$ разрядов к группе из p разрядов должно увеличиться на 1. Действительно, если старшая цифра в группе из $p-1$ разрядов больше $n/2$, то, каковы бы ни были остальные разряды этой группы, в ее оптимальном представлении должна обязательно присутствовать единица, переносимая в следующий по весу разряд (т. е. слагаемое $1 \cdot 2^{p-1}$). Если расшифровываются группы из $p-1$ разрядов, то эта единица добавляется к следующей группе из $p-1$ разрядов. При этом вероятности всех комбинаций в этой следующей группе, кроме комбинации $00\dots 0$, сохраняются неизменными, комбинация $00\dots 0$ становится невозможной, но зато с той же вероятностью

может появиться комбинация $1.00\dots 0$ (для которой также $\sum_{i=1}^{p-1} |a_i| = 0$);

поэтому математическое ожидание количества тактов сложения-вычитания при умножении на эту группу разрядов не изменяется. Однако когда мы, переходя от группы из $p-1$ разрядов к группе из p разрядов, добавляем к ней слева один разряд и ставим в этот разряд цифру «0», то переносимая в этот разряд из младших разрядов единица потребует лишнего такта сложения в последнем из p циклов умножения на группу из p разрядов. Попытка изменить расшифровку младших разрядов группы с тем, чтобы не было переноса в разряд с весом 2^{p-1} , привела бы к тому, что в $(p-1)$ -м цикле умножения на данную группу разрядов количество тактов сложения-вычитания возросло бы по меньшей мере на 1.

Итак, если левая из p цифр есть 0, то в $n/2$ случаях (в комбинациях $00\dots, 01\dots, 02\dots, \dots, 0(n/2-1)\dots$) количество тактов сложения-вычитания при умножении на эту группу цифр такое же, как при умножении на соответствующую группу из $p-1$ цифр, а в $(n/2)-1$ случаях (в комбинациях $0(n/2+1)\dots, 0(n/2+2)\dots, \dots, 0(n-1)\dots$) — на 1 больше, чем для соответствующих групп из $p-1$ цифр.

Теперь рассмотрим случай, когда левая из p цифр есть «0», а следующая за ней равна в точности $n/2$. Если $p=2$, то количество тактов сложения-вычитания здесь такое же, как при умножении на соответствующую группу из $p-1$ разрядов (при этом $p-1=1$; «соответствующая группа из $p-1$ разрядов» представляет собой просто 1 разряд, в котором записана цифра $n/2$ и при умножении на который требуется $n/2$ тактов сложения; при умножении на группу из $p=2$ цифр 0 ($n/2$) тоже требуется $n/2$ тактов сложения). Если $p>2$, то изменение (количества тактов сложения-вычитания при переходе от группы из $p-1$ разрядов вида $(n/2)\dots$ к группе из p разрядов вида $0(n/2)\dots$ зависит от следующей цифры. Аналогично предыдущему можно показать, что в $n/2-1$ случаях (в комбинациях $0(n/2)0\dots, 0(n/2)1\dots, 0(n/2)(n/2-2)\dots$ (количество тактов сложения-вычитания при умножении на группу из p разрядов остается таким же, как при умножении на соответствующую группу из $p-1$ разрядов, в $n/2$ случаях (в комбинациях $0(n/2)(n/2)\dots, 0(n/2)(n/2+1)\dots, \dots, 0(n/2)(n-1)\dots$) она возрастает на 1, а в одном случае — для комбинаций вида $0(n/2)(n/2-1)\dots$ — требуется особое рассмотрение. В частности, при $p=3$ умножение на группу разрядов $0(n/2)(n/2-1)$ требует такого же количества тактов, как умножение на соответствующую группу из $p-1$ разрядов $(n/2)(n/2-1)$; при $p>3$ изменение в количестве тактов сложения-вычитания для комбинаций вида $0(n/2)(n/2-1)\dots$ зависит от четвертой цифры. Приведенное рассуждение можно было бы продолжить и дальше.

Теперь выпишем все те комбинации, начинающиеся с 0, для которых количество тактов сложения-вычитания больше на 1, чем для соответствующих комбинаций из $p-1$ разрядов. Сюда войдут: комбинации вида $0(n/2+1)\dots, 0(n/2+2)\dots, \dots, 0(n-1)\dots$;

их суммарная вероятность равна $(n/2-1) \cdot 1/n^2$; комбинации вида $0(n/2)(n/2) \dots$; $0(n/2)(n/2+1) \dots$, ..., $0(n/2)(n-1) \dots$; их суммарная вероятность $— n/2 \cdot 1/n^3$; комбинации вида $— 0(n/2)(n/2-1)(n/2) \dots$; $0(n/2)(n/2-1)(n/2+1) \dots$, ..., $0(n/2)(n/2-1)(n-1) \dots$; их суммарная вероятность $— (n/2-1) \frac{1}{n^4}$; и т. д.

Таким образом, комбинации, начинающиеся с 0, в общей сложности, внесут в разность $A_p(n) - A_{p-1}(n)$ следующую величину:
при p четном —

$$(\text{чет}) \Delta_{[A_p(n) - A_{p-1}(n)]}^{(0)} = \left(\frac{n}{2} - 1\right) \frac{1}{n^2} + \frac{n}{2} \cdot \frac{1}{n^3} + \left(\frac{n}{2} - 1\right) \frac{1}{n^4} + \dots + \left(\frac{n}{2} - 1\right) \frac{1}{n^p},$$

при p нечетном —

$$(\text{неч}) \Delta_{[A_p(n) - A_{p-1}(n)]}^{(0)} = \left(\frac{n}{2} - 1\right) \frac{1}{n^2} + \frac{n}{2} \frac{1}{n^3} + \left(\frac{n}{2} - 1\right) \frac{1}{n^4} + \dots + \frac{n}{2} \frac{1}{n^p}.$$

Отсюда для любого $p > 2$ можно записать

$$\Delta_{[A_p(n) - A_{p-1}(n)]}^{(0)} = \frac{n^{p+1} - n^p - n^2 + 1 - (-1)^p (n-1)}{2(n^2 - 1)n^p}.$$

Рассмотрим далее комбинации, начинающиеся с цифры 1. Не трудно показать, что в тех случаях, когда в комбинациях, начинающихся с цифры 0, количество тактов сложения-вычитания при переходе от групп по $p-1$ разрядов к группам по p разрядов не возросло, здесь будем иметь увеличение количества тактов на 1; в тех случаях, когда в комбинациях, начинающихся с 0, мы имели увеличение тактов на 1, в комбинациях, начинающихся с 1, увеличение количества тактов составит 2. Учитывая, кроме того, что суммарная вероятность комбинаций, начинающихся с 1, равна $1/n$, найдем величину, которая вносится в разность $A_p(n) - A_{p-1}(n)$ всеми этими комбинациями:

$$\Delta_{[A_p(n) - A_{p-1}(n)]}^{(1)} = \Delta_{[A_p(n) - A_{p-1}(n)]}^{(0)} + \frac{1}{n}.$$

Точно так же:

$$\Delta_{[A_p(n) - A_{p-1}(n)]}^{(2)} = \Delta_{[A_p(n) - A_{p-1}(n)]}^{(0)} + \frac{2}{n},$$

$$\Delta_{[A_p(n) - A_{p-1}(n)]}^{(3)} = \Delta_{[A_p(n) - A_{p-1}(n)]}^{(0)} + \frac{3}{n},$$

.....

$$\Delta_{[A_p(n) - A_{p-1}(n)]}^{\left(\frac{n}{2}-1\right)} = \Delta_{[A_p(n) - A_{p-1}(n)]}^{(0)} + \frac{\frac{n}{2}-1}{n}.$$

Для комбинаций, начинающихся с цифры $n/2$, можно было бы провести рассмотрение, аналогичное тому, которое было проведено для комбинаций, начинающихся с цифры 0. Опуская его, запишем прямо конечный результат:

$$\Delta_{[A_p(n) - A_{p-1}(n)]}^{\left(\frac{n}{2}\right)} = \Delta_{[A_p(n) - A_{p-1}(n)]}^{(0)} + \frac{\frac{n}{2}-1}{n} + \frac{1}{n^p}.$$

Затем, снова аналогично предыдущему

$$\Delta_{[A_p(n)-A_{p-1}(n)]}^{\left(\frac{n}{2}+1\right)} = \Delta_{[A_p(n)-A_{p-1}(n)]}^{\left(\frac{n}{2}\right)} - \frac{1}{n} = \Delta_{[A_p(n)-A_{p-1}(n)]}^{(0)} + \frac{\frac{n}{2}-2}{n} + \frac{1}{n^2},$$

.....

$$\Delta_{[A_p(n)-A_{p-1}(n)]}^{(n-1)} = \Delta_{[A_p(n)-A_{p-1}(n)]}^{(0)} + \frac{1}{n^p}.$$

Теперь можем получить (для $p > 2$)

$$A_p(n) - A_{p-1}(n) = \sum_{i=0}^{n-1} \Delta_{[A_p(n)-A_{p-1}(n)]}^{(i)} = \frac{n}{4} + \frac{(-n)^{1-p}-1}{2(n+1)}.$$

Наконец,

$$A_q(n) = A_2(n) + \sum_{p=3}^q [A_p(n) - A_{p-1}(n)].$$

Подставляя вычисленные выше значения $A_2(n)$ и $A_p(n) - A_{p-1}(n)$, после некоторых преобразований найдем

$$A_q(n) = \frac{1}{2(n+1)} \left\{ \frac{n}{n+1} [1 - (-n)^{-q}] + \frac{n^2+n-2}{2} q \right\}.$$

5. Величина $A_q(n)$ для нечетного n

Рассмотрев рассуждение, проведенное в п. 4 для четного n , нетрудно убедиться, что в том случае, когда n нечетно, расшифровка групп разрядов множителя не может дать никакой экономии в количестве тактов сложения-вычитания сверх того, что получается при оптимальном представлении каждого разряда множителя в отдельности.

Действительно, если цифрам n -ичной системы счисления приписаны значения $0, 1, \dots, (n-1)$, то при нечетном n среди них нет такой цифры n_i , которую можно было бы представить в виде $n_i = n - n_i$. Поэтому если бы мы решили изменить расшифровку цифры какого-либо разряда так, чтобы возник (или, наоборот, был предотвращен) перенос в следующий разряд, то при умножении на данный разряд потребовался бы по крайней мере один лишний такт сложения или вычитания. Между тем, при умножении на следующий разряд единица, переносимая из расшифровки предыдущего разряда множителя, может сократить не более одного такта сложения-вычитания. Поэтому никакой экономии в количестве тактов сложения-вычитания при нечетном n таким путем получить нельзя.

Следовательно, для нечетного n

$$A_q(n) = qA_1(n) = \frac{q}{4} \left(n - \frac{1}{n} \right)$$

(см. раздел 2,б).

6. Общие выводы

Обобщая результаты разделов 4 и 5, получим (для любых n и q):

$$\frac{A_q(n)}{q} = \frac{1}{4(n+1)} \left\{ n^2 + n - 2 + \frac{n}{q(n+1)} [1 + (-1)^n] [1 - (-n)^{-q}] + \frac{n-1}{2n} [1 - (-1)^{(n)}] \right\}. \quad (2)$$

Отсюда:

$$\lim_{q \rightarrow \infty} \frac{A_q(n)}{q} = \frac{1}{4(n+1)} \left\{ n^2 + n - 2 + \frac{n-1}{2n} [1 - (-1)^n] \right\}. \quad (3)$$

Выражение (3) определяет среднее количество тактов сложения-вычитания, приходящихся на 1 разряд перемножаемых чисел, для случая, когда весь бесконечно-значный множитель представлен оптимальным способом в виде суммы степеней числа n с некоторыми коэффициентами. Но в арифметическом устройстве, показанном на рисунке, и не могут выполняться никакие другие операции, кроме сложения множимого с суммой частичных произведений или вычитаний из нее и сдвигов частных произведений (т. е. умножений их на степени n); по определению же логические методы ускорения умножения не должны затрагивать структуры арифметического узла. Поэтому можно утверждать, что выражением (3) указывается предел возможностей сокращения времени умножения путем применения логических методов ускорения. Каков бы ни был метод ускорения умножения, если состав оборудования арифметического устройства не изменяется, а ускорение достигается только за счет усложнения схемы управления, не удастся получить меньшего количества тактов сложения-вычитания, чем это определяется выражением (3).

В разделе 3 говорилось уже, что величина q в действительности не только не может возрастать неограниченно, но должна быть существенно меньше количества разрядов в перемножаемых числах m . Однако из выражения (2) видно, что уже при сравнительно небольших значениях q величина $\frac{A_q(n)}{q}$ достаточно близка к пределу. Например, для того чтобы величина $\frac{A_q(n)}{q}$ отличалась от предельного значения меньше чем на 10%, при $n=2$ достаточно иметь $q=4$, при $n=4$ достаточно $q=2$, для всех остальных n достаточно $q=1$.

В заключение оценим различные основания системы счисления n с точки зрения скорости выполнения умножения при использовании наиболее сильных из возможных для данной системы логических методов ускорения умножения. В таблице приведены значения функции $f_\infty(n)$ для нескольких начальных значений n . Видно, что как и в предыдущих случаях (раздел 2), наиболее выгодна двоичная система. При возрастании n функция целочисленного аргумента $f_\infty(n)$ монотонно возрастает.

Т а б л и ц а

n	2	3	4	5	6	7	8	9	10
$f_\infty(n)$	1,00	1,26	1,35	1,55	1,66	1,84	1,95	2,10	2,22

Поступило в редакцию 21 XI 1958.

VI. ПРОЦЕССЫ УПРАВЛЕНИЯ В ЖИВЫХ ОРГАНИЗМАХ

ОСНОВЫ ЭВОЛЮЦИОННОГО ПРОЦЕССА В СВЕТЕ КИБЕРНЕТИКИ

И. И. ШМАЛЬГАУЗЕН

(МОСКВА)

Применение принципов кибернетики получило в биологии уже довольно широкое распространение. Первые попытки в этом направлении были сделаны Н. Винером (1948), обобщившим в своей книге основные принципы устройства автоматически регулируемых механизмов не только в машинах, но и в живых организмах. Понятие наследственной информации встречается уже в представлениях Э. Шредингера об организации хромосом (1945), хотя общая теория информации была разработана позднее К. Шенноном (1947—1948). В приложении к биологии кибернетика и теория информации использовались главным образом для анализа регуляторных процессов, осуществляющихся в организме через посредство нервной системы. В известной степени новой точкой зрения охватываются также эндокринная система и все регуляторные механизмы организма в целом (гомеостат). Всеобщее применение получил термин «наследственная информация» [11]. Были также попытки использовать понятие наследственной информации для описания процессов индивидуального развития. Что же касается вопроса о закономерностях эволюционного процесса, то мне известна только одна, на мой взгляд крайне неудачная, попытка приписать и в этом случае (как и в индивидуальном развитии) регуляторные функции наследственному материалу с его генами.

К сожалению, распространение термина «наследственная информация» в биологии происходит без достаточной критики, и его применение не всегда оправдано. Это приводит к тому, что понятие «информация», ценное именно своей определенностью, быстро теряет свои границы и становится таким же неопределенным, как и большинство биологических понятий, являющихся поэтому источником бесконечных и совершенно бесплодных «дискуссий». Чтобы избежать последних, представляется совершенно необходимым твердо держаться того содержания понятия «информация», которое принято в общей теории информации.

Биологи теперь часто применяют термин «наследственная информация» как простой синоним наследования. Такая замена терминов ничем не оправдана, она вносит ненужные ограничения в понятие наследования и может привести только к путанице. Еще чаще термин употребляется для обозначения специфического действия наследственного материала в развивающемся и зрелом организме. Предполагается использование химических средств передачи «информации» от наследственных единиц, генов, через посредство ферментов и гормонов к развивающимся и функционирующим частям организма (в виде локального действия в клетке или

дистантной индукции). И в том и в другом случае термин используется лишь для образного описания различных способов передачи некоторых активных веществ. В первом случае речь идет о передаче материала от одной особи к другой — дочерней. Во втором — предполагается реализация наследственного материала в индивидуальном развитии особи. Последнее относится, однако, не к передаче наследственной информации, а к ее преобразованию, что должно, конечно, изучаться, но в других терминах.

Специфика понятия информации, применяемого в кибернетике, при указанном его расширении теряется, и вместе с тем теряется основное его преимущество — возможность количественного учета информации. Между тем вся ценность понятия «информация» заключается в его определенности, в возможности измерения количества информации и оценки ее качества. Это возможно, однако, только на основе разработанной Шенноном статистической теории информации.

1. Необходимые предпосылки для эволюции организмов

Основными факторами эволюции организмов обычно считаются: 1) мутабельность, ведущая к изменению генетического состава популяции, 2) случайные колебания концентрации различных мутаций вследствие колебаний численности популяций во времени и в пространстве, 3) ограничения панмиксии при различных формах изоляции и 4) естественный отбор, связанный с различной жизнеспособностью различных генотипов в данных условиях существования. К этому иногда прибавляют в качестве пятого фактора вселение в свободную экологическую нишу.

В значении всех этих факторов для эволюции не может быть никаких сомнений. Однако нельзя не видеть, что перечисленные факторы не равноценны. Мутабельность является, конечно, обязательной предпосылкой. Колебания численности могут способствовать случайному увеличению концентрации одних мутаций и выпадению других. Они, однако, не могут рассматриваться в качестве необходимого фактора, определяющего возможность эволюции. Точно так же и изоляция, являясь необходимым фактором внутривидовой дифференциации и видообразования, не может рассматриваться в качестве обязательного условия эволюции вообще. Кроме наследственной изменчивости (мутабельности) только естественный отбор является совершенно необходимой предпосылкой эволюционного процесса.

Имеется, однако, еще одна предпосылка, о которой не говорят, так как она, по-видимому, считается само собой разумеющейся. Это — способность к поддержанию своей жизни и к самовоспроизведению [13]. Если попытаться проследить историю организмов назад до истоков самой жизни, то становится очевидным, что именно в поддержании своей жизни и в самовоспроизведении лежат как основы существования организмов, так и все необходимые условия для их эволюции. В самовоспроизведении имеется и источник мутабельности, а способность к поддержанию своей жизни является наиболее примитивной основой естественного отбора.

Способность к самовоспроизведению заложена в химическом строении молекул дезоксирибонуклеиновой кислоты (ДНК), являющихся основой всех форм жизни (включая вирусы). Способность нуклеотидов к полимеризации обусловила возможность наращивания огромных молекул нуклеиновых кислот, обладающих поэтому глубоко индивидуальным строением и свойствами. Строение молекул дезоксирибонуклеиновой кислоты в виде парной цепочки со слабыми водородными связями между обоими партнерами допускает продольное расщепление молекулы. После расщепления каждая цепь синтезирует недостающего ей партнера и вос-

становливая таким образом свое исходное парное строение [14]. При этом самовоспроизведении в точности повторяется индивидуальная структура материнской молекулы. Молекулы ДНК несомненно являются регуляторами процессов внутриклеточного синтеза индивидуальных молекул рибонуклеиновых кислот, а через это и синтеза специфических белковых тел и ферментов. Способность к самовоспроизведению более сложных биологических структур—хромосом и клеток—представляет уже результат дальнейшей эволюции первичных живых существ, в основе которой лежит возможно точное, равномерное и синхронное распределение всех наиболее существенных элементов живого вещества. Механизм митотического деления клеток представляет, очевидно, результат длительной эволюции, достигнутый, однако, живыми организмами еще задолго до оформления основных ветвей растительного и животного мира. Несомненно, что первые живые существа были весьма малоустойчивыми и при их воспроизведении очень часто возникали самые разнообразные отклонения [9]. Только приобретение указанного механизма равномерного распределения всех наиболее ответственных элементов клетки привело к более точной репродукции материнских свойств в дочерних особях.

Поэтому мы считаем изменчивость первичным явлением, выражением неизбежного несовершенства репродукции на первых этапах эволюции живых существ. По мере выработки более точных механизмов, обеспечивающих точное воспроизведение унаследованных свойств, первичная изменчивость организмов постепенно вводилась во все более узкое русло ограниченной мутабельности.

Способность к поддержанию своей жизни при меняющихся условиях существования означает безотказную работу синтезирующих механизмов при известном минимуме необходимых жизненных средств, т. е. доступных внешних источников веществ и энергии. Существа, которые не могут использовать средства окружающей среды для поддержания своей жизни и размножения, гибнут. Те особи, которым удается использовать наличные средства, сохраняют свою жизнь и распространяют ее путем самовоспроизведения. В этом простом выживании жизнеспособных особей находит свое выражение наиболее примитивная форма естественного отбора. В течение прогрессивной эволюции непрерывно усложнялись условия существования организмов и значительно изменялись формы естественного отбора.

Мы здесь отмечаем пока только наиболее элементарные основы эволюционного процесса и видим, что они заложены в химических основах строения организмов и могут быть прослежены до самых истоков жизни. В способности к самовоспроизведению мы видим и начало наследственности, а в несовершенстве этой способности—источник наследственной изменчивости. В способности к поддержанию своей жизни мы видим основы авторегуляции и объект для действия простейших форм естественного отбора.

2. Исходные формы и закономерности эволюции

Основным выражением эволюционного процесса на всех этапах развития органического мира является непрерывное приспособление живых существ к условиям окружающей среды. Это приспособление осуществляется через посредство естественного отбора, и поэтому неудивительно, что адаптация и отбор имеют одни и те же корни в первичных формах жизни. Живые организмы всегда требовали наличия известных условий, без которых не были возможны ни поддержание жизни, ни самовоспроизведение. Формы, в которых проявлялась жизнь, были различны.

Уже первые живые существа различались между собой как по источникам питания, так и по источникам используемой энергии. Таким образом, и условия существования различных организмов были всегда различными. В известных условиях могли существовать только некоторые определенные организмы. В этой способности жить, расти и размножаться в определенных условиях среды и выражалась их приспособленность. Она проявлялась просто в способности поддерживать свою жизнь и в способности к самовоспроизведению в данных условиях. И естественный отбор в простейших условиях выражался в простом выживании и размножении одних, более приспособленных особей, и гибели или отсутствии самовоспроизведения у других, менее приспособленных особей.

В основе таких различий, которые связаны с жизнеспособностью в разных условиях, лежат прежде всего различия в обмене веществ и энергий, что в свою очередь определяется структурой основных химических веществ, регулирующих этот обмен. Таким образом, и естественный отбор вел одновременно к отбору известных химических структур. Таковыми являются молекулы ДНК и их производные молекулы РНК, белков и энзимов. Все это—сложные полимерные органические соединения с высокой индивидуальностью. При самовоспроизведении молекулы ДНК повторяются в точной копии, а это влечет за собой и аналогичное воспроизведение производных веществ. При взаимодействии с различными химическими или физическими агентами возможны все же нарушения структуры молекулы ДНК—стереохимические перестройки. Такие перестройки и являются источником индивидуальных различий и материалом для естественного отбора. Надежность точного воспроизведения естественно зависит от числа распределяемых при делении сходных биологических структур. Чем их больше, тем более вероятно их равномерное распределение. Число биологических единиц определяется их способностью к самовоспроизведению, в основе которой лежит самовоспроизведение полимерных химических соединений (ДНК). При самовоспроизведении элементарных биологических единиц, без последующего разделения всего организма (особи), получается увеличение их общего числа. Таким образом, биологическая «полимеризация» ведет к усложнению организации уже на доклеточной стадии развития живых существ. Увеличивалось число генов, число митохондрий, увеличивалось и число молекул важнейших химических веществ. Увеличение числа сходных единиц допускало в известной степени изменение отдельных элементов без нарушения жизнеспособности организма в целом. Такие изменения вели к возникновению вариантов, служивших объектом для деятельности естественного отбора и дальнейшей эволюции организмов на пути *дифференциации* их элементарных структур и функций. Однако обособление и специализация функций требовали и согласования и объединения их. Дифференциация естественно дополнялась *интеграцией*.

Таким образом, в результате полимеризации, дифференциации и интеграции генов возникли хромосомы как наиболее ответственные структуры на клеточном уровне организации. В дальнейшей эволюции самовоспроизведение клеточных структур без разделения клеток приводило к новым типам усложнения организации—к многохромосомным и многоядерным простейшим организмам. Неполное разделение клеток приводило к образованию колоний и к возникновению многоклеточных организмов. И *многоклеточность* можно рассматривать как особую форму биологической полимеризации, которая таит в себе условия для возможности дальнейшей эволюции путем изменения и специализации отдельных элементов сложного организма. И здесь полимеризация послужила начальным этапом усложнения, ведущего к прогрессивной дифференциации и интегра-

ции специализированных клеток. Однако и специализированные клетки сохраняли способность к самовоспроизведению, а это вело к умножению числа сходных клеток, объединенных в тканях сложного организма. Дальнейшая специализация вела к еще большему усложнению организации. Именно увеличение числа сходных элементов, т. е. биологическая полимеризация, обычно лежит в основе прогрессивного развития сложных систем и органов, так как является предпосылкой для возможности дальнейшей дифференциации и интеграции клеточных структур (например, в эволюции глаза и центральной нервной системы, особенно у насекомых и у позвоночных). Вместе с тем увеличение числа сходных элементов является простейшим средством для увеличения надежности воспроизведения, для интенсификации функций и расширения связей с внешней средой. И в половом размножении необходимость известной страховки от случайных неудач приводила к увеличению числа органов размножения как у растений, так и у животных. У сидячих животных это явилось основой повторности этих органов в радиальной ориентации (антимерия). У подвижных животных с билатеральной симметрией органы размножения располагались последовательно вдоль оси. Это отражалось затем на всей организации, послужив источником возникновения метамерии. И в этом случае повторность в расположении органов по сегментам тела, т. е. биологическая полимеризация, дала возможность внести изменения в строение отдельных метамеров, достигнуть известной специализации. Таким образом, *полимеризация привела к дифференциации и интеграции многоклеточных структур*, органов и систем на высшем уровне организации.

В связи с дальнейшими вопросами можно здесь же отметить, что на тех же принципах строилась и эволюция материальной основы наследственного воспроизведения организмов—структуры хромосом и всего генома в целом.

3. Регулирующие механизмы эволюции

Известная степень авторегуляции является необходимым условием поддержания жизни в меняющихся условиях существования даже на низших уровнях организации живых систем. Соответственно и клетка имеет свой регуляторный аппарат, и сложный многоклеточный организм обладает сложной системой регуляторных механизмов (гомеостат). Наличие этих механизмов дает организму известную степень устойчивости, позволяющей ему поддерживать свое существование, расти и размножаться при обычных изменениях в факторах той внешней среды, к которой данный организм приспособлен [3, 4].

Все высшие организмы проходят более или менее сложный цикл изменений в течение своей индивидуальной жизни: они растут, развиваются, меняют свои функции, а также и условия своего существования, переходят из одной среды в другую, иногда резко отличную. Все эти изменения, включаемые в понятие индивидуального развития или онтогенеза, происходят в тесном и непрерывном взаимодействии с факторами внешней среды, доставляющими необходимые условия для этого развития. В этих изменениях выражаются видовые и индивидуальные наследственные свойства организма в виде определенных, специфических норм реагирования на внешние факторы, что указывает на значение внутренних факторов во всех этих преобразованиях. Индивидуальное развитие протекает «нормально» для данного вида также и при некоторых отклонениях во внешних факторах, и очень часто сопровождается адекватными ответами на эти отклонения, так что организм может сохранять свою жизнеспособность и приспособленность в различных условиях существования в преде-

лах обычных их колебаний. Все это указывает на наличие регуляторных механизмов, определяющих течение процессов индивидуального развития.

В историческом развитии организмов, их эволюции, основным является прогрессивная адаптация, т. е. непрерывное приспособление организмов к меняющимся условиям существования. Эти изменения протекают, однако, уже не в особи, а в ряде ее потомков или, точнее, в ряде последовательных поколений, составляющих некоторую более или менее обособленную популяцию данного вида организмов. Закономерность изменений, ведущих к дальнейшей и все более тонкой приспособленности организмов к различным факторам внешней среды, заставляет и здесь говорить о наличии механизма, регулирующего ход таких изменений. Основа этой регуляции хорошо известна, и мы уже говорили о ней. Это — естественный отбор.

Таким образом, мы познакомились уже с двумя механизмами, регулирующими преобразование живых организмов. Один из них действует в масштабе индивидуального развития особи, другой — в масштабе исторического развития популяции и даже вида в целом. Оба механизма должны быть связаны между собой, так как всякая популяция состоит из особей и перестраивается только через посредство их размножения и развития. Действие регуляторного механизма развития особи явно зависит от унаследованных норм реакции на изменения во внешних факторах. Действие регуляторного механизма преобразования популяции, т. е. естественный отбор, также зависит от унаследованных свойств популяции и форм ее взаимодействия с факторами внешней среды. Оба механизма обнаруживают теснейшую взаимозависимость с внешними факторами, однако эта взаимозависимость проявляется в различных формах, так как относится к разным уровням организации и различным процессам. Гомеостат, как выражение унаследованных норм реакций на изменения во внешних факторах, является регулятором индивидуального развития особи. Естественный отбор, как выражение сложных взаимоотношений между особями данного вида и условиями внешней среды, является механизмом исторического преобразования популяций и видов в целом, т. е. эволюции.

Таким образом, мы считаем, что основным регулирующим механизмом эволюции являются те сложные взаимоотношения между организмом и средой, которые Дарвин назвал борьбой за существование, и результат этих взаимоотношений — естественный отбор.

Возможны, правда, и другие точки зрения. Единственная известная мне попытка связать закономерности эволюции с принципами кибернетики исходит из предположения о том, что основной регулирующий механизм эволюции лежит в геноме и что естественный отбор является лишь средством для осуществления обратной связи от внешней среды к организму, т. е. «информирует» его о положении во внешней среде. В этом случае организм сам является регулятором своей эволюции. Автогенетический механизм этой эволюции, по существу, остается непонятым.

Можно было бы предположить и нечто весьма сходное. Регулирующий механизм лежит в самом геноме, а обратная информация подается непосредственно факторами внешней среды. Геном отвечает на изменения во внешних факторах адекватными изменениями, т. е. такими, которые проявляются в развитии приспособительных признаков организации. Такое по внешности материалистическое объяснение является, по существу, также идеалистическим, так как привлекает для «объяснения» неизвестные и непонятные явления — адекватность изменения наследственной основы и столь же адекватное изменение процессов развития, ведущее к приспособительным преобразованиям организации. Это — сложная цепь невероятностей. Она построена на совершенно необоснованном предположении о возможности непосредственного влияния внешних факторов и адекватной перестройки аппарата наследования. Этот аппарат развивался в течение всей эволюции организмов в направлении возможно большей точности самовоспроизведения и возможно большей защиты от ненормальных внешних воздействий. В результате он весьма надежно защищен от посторонних влияний целой системой регуляторных механизмов клетки и всего организма в целом.

Если рассматривать процесс эволюции с точки зрения кибернетики, то становится совершенно ясным, что регуляция этого процесса осуществляется в сложном взаимодействии внешних и внутренних факторов. Точная приспособленность организма к неорганическим факторам климата (в широком смысле) и в особенности к сложным биологическим соотношениям данного «биогеоценоза», в котором он нормально живет, показывает, что все эти факторы имеют свое значение в регуляции эволюционного процесса.

В роли регулятора эволюции выступает, следовательно, весь *биогеоценоз* [2], включающий рассматриваемый вид (популяцию) организмов. Биогеоценоз как регулятор должен быть связан с развивающейся популяцией организмов двумя линиями связи—прямой линией передачи управляющих сигналов от биогеоценоза к популяции и линией обратной связи, передающей в биогеоценоз информацию о действительном состоянии популяции. Наличие таких линий связи нетрудно показать. По линии прямой связи происходит передача *наследственной информации* от биогеоценоза к популяции через посредство зигот каждого нового поколения. Наследственная структура этих зигот отражает влияние (через естественный отбор и комбинирование признаков родительских особей) биогеоценоза и реализуется в развивающихся особях, вливающих в состав популяции. *Обратная информация* передается биогеоценозу через посредство специфических форм жизнедеятельности отдельных особей данной популяции. Эта информация преобразуется в биогеоценозе (т. е. в «регуляторе») в результате борьбы за существование и естественного отбора и передается в популяцию в виде нового поколения преобразованных зигот.

Таким образом, осуществляется двусторонняя связь между биогеоценозом и включенной в его состав популяцией. Однако между обеими линиями передачи информации нет непосредственной связи, так как они находятся на разных уровнях. Наследственная информация передается на молекулярном и во всяком случае внутриклеточном уровне организации, а обратная информация—только на уровне организации целой особи.

Переход от одной линии связи к другой осуществляется здесь (как обычно и в технике) через посредство очень сложного механизма *преобразования*. Наследственная информация преобразуется в *процессах индивидуального развития* в средства передачи обратной информации—в фенотип особи, являющийся полноценным носителем жизни и активным участником наступления на жизненные ресурсы биогеоценоза (борьба за существование). Это преобразование протекает в условиях непрерывного взаимодействия с факторами внешней среды, однако не в простой зависимости от них, а в виде реализации унаследованных норм реакций под контролем сложных регуляторных механизмов клетки и всего организма в целом.

В биогеоценозе происходит новое *преобразование обратной информации в наследственную* с переходом от уровня организации особи (фенотипа) на уровень внутриклеточной организации (в половых клетках и зиготе). Это совершается в результате естественного отбора и полового процесса. Оно начинается с контроля организации (в борьбе особей за существование) и естественного отбора особей и завершается созреванием половых клеток и образованием зигот с новыми комбинациями наследственных свойств двух особей, прошедших все жизненные испытания в данном биогеоценозе. Этим и замыкается полный круг преобразований в элементарном цикле эволюции [6, 7].

Для осуществления более совершенной регуляции эволюционного процесса необходима, очевидно, безупречная работа основных звеньев

всего регулирующего механизма, а именно, безошибочная передача информации по обоим каналам связи и адекватное внешним условиям преобразование этой информации в двух его механизмах: в индивидуальном развитии и в процессе естественного отбора. Так как регулирующие механизмы индивидуального преобразования и эволюционных изменений популяции не являются замкнутыми системами, а связаны с внешней средой, то необходимо считаться с возможностью различных случайных внешних влияний, оказывающихся в роли «помех», искажающих передачу информации и нарушающих нормальное течение преобразований.

4. Передача наследственной информации и ее защита от помех

Для передачи наследственной информации использован уже описанный механизм самовоспроизведения молекул ДНК и равномерного распределения биологических единиц—генов и хромосом—при митотических клеточных делениях. Строение молекул ДНК в виде двойной цепочки нуклеотидов, различающихся по входящим в их состав основаниям пуринового (аденин или гуанин) или пиримидинового ряда (тимин или цитозин), обеспечивает почти неограниченную возможность создания новых индивидуальных комбинаций. В парной цепочке ДНК указанные основания образуют дополняющие друг друга пары: аденин с тимином или цитозин с гуанином. Такие пары чередуются в определенной последовательности по всей молекуле и определяют индивидуальные ее свойства. При большом числе звеньев, исчисляемом тысячами, комбинированное чередование четырех различных знаков (2 пары оснований и 2 возможных положения) обеспечивает возможность «записи» наследственной информации любой сложности в виде условного кода (точно так же азбука Морзе использует только четыре знака для передачи какого угодно сообщения). Химической единицей этого кода является, очевидно, пара оснований в молекуле ДНК. Расщепление и восстановление этой молекулы гарантирует точное ее копирование при воспроизведении [14]. Биологической единицей наследственной информации является более сложное образование—ген, в состав которого входит не только ДНК, определяющая его индивидуальность, но и РНК и протеины. Дробимость гена во многих случаях уже доказана, и ген является в сущности блоком информации. Гены являются элементарными и при том дискретными биологическими единицами наследственной информации. Их дискретность доказывается возможностью изолированного их изменения в мутациях и фактическим разделением соседних генов при разрывах хромосом и обмене их частями. И все же они нормально не обособлены друг от друга. Они располагаются в определенной последовательности в виде более или менее длинной цепочки, образующей отдельную хромосому. Внутри хромосомы они связаны силами «сцепления». Комбинирование элементов информации—генов—хотя и возможно, но строго ограничено системными связями. С точки зрения теории информации каждая хромосома представляет собой длинный блок связанной информации или целое закодированное сообщение со своими «логическими» связями. Интересно, что при передаче наследственной информации используется самая распространенная из применяемых в технике связи систем кодирования: составление длинных блоков. В данном случае—это блоки генов, из которых каждый имеет только одно из двух возможных значений (простые аллеломорфы; в пределах диплоидной особи множественные аллеломорфы невозможны). Передача наследственной информации осуществляется только через посредство митотических клеточных делений, при которых происходит репродукция наследственного кода в виде точной копии исходного.

При клеточных делениях происходит одновременное разделение всех двойных (в результате репродукции) хромосом и их расхождение в дочерние клетки. Этот удивительно сложный механизм равнокачественной передачи наследственной информации по всем продуктам последовательных клеточных делений обеспечивает полную ее надежность. При этом вся информация передается как нечто целое, без всякого перекомбинирования, от оплодотворенной яйцеклетки (зиготы) до первичных половых клеток зрелого организма (в соматических клетках возможны перестройки и в особенности умножения наследственных структур, связанные с процессами дифференцировки, на чем мы здесь не останавливаемся). Эта точность и надежность передачи отражает изумительно тонкий механизм воспроизведения и равномерного распределения результатов многократного копирования (последовательной передачи копии с копии). Такой механизм—результат длительного совершенствования, происходившего еще на заре эволюции живых существ. Это понятно именно в связи с необходимостью безотказного функционирования регулирующего механизма. Без этого не была бы осуществлена преемственность организации, был бы невозможен ее контроль, невозможна была бы ни эволюция, ни само существование даже очень простых форм организации.

Вместе с тем наследственная информация не представляет собой чего-либо неизменяемого. Наоборот, имеется специальный механизм, ведущий к постоянным изменениям, которые (после их жизненной апробации) используются для внесения «поправок» в строение наследственного кода. Прежде всего отметим, что под влиянием случайных внешних воздействий могут возникать также объективно случайные нарушения в строении кода, которые обнаруживаются как наследственные изменения фенотипа—мутации. Такие изменения передаются через клеточные деления с такой же надежностью, как и неизменные наследственные свойства. Обычные генные мутации представляют изменения молекулярной структуры хромосом, ограничивающиеся одним геном и только в одной из пары гомологичных хромосом. Изменение происходит лишь в малом масштабе, и это показывает, что и причина, вызвавшая это изменение, действовала в молекулярном масштабе. Источники мутационных изменений довольно хорошо изучены. Поскольку хромосома состоит из химически активных субстанций, естественно предположить, что под влиянием химических агентов может произойти та или иная перестройка в молекуле ДНК. Значительные изменения приведут организм к гибели, а локальная перестройка может привести к мутации. Такие мутации уже получались. Возможно, что и в естественных условиях некоторые мутации возникают под влиянием каких-либо продуктов обмена (на это указывают факты накопления мутаций при хранении семян). Обычным средством получения мутаций являются, однако, лучи Рентгена. Механизм их действия довольно ясен; они вызывают ионизацию, которая является источником химических перестроек в какой-либо точке хромосомы или в ее ближайшем соседстве. Наконец, мутации могут возникать и спонтанно, т. е. без видимой причины. Зависимость спонтанного мутирования от температуры (по закону Вант-Гоффа) показывает, что его источником может быть случайная концентрация теплового движения в одной точке, которая приводит к местной химической перестройке.

Все источники мутирования являются в природе объективно случайными, и все они действуют в молекулярном масштабе. Случайные причины, вызывающие нарушение передачи, называются в теории информации помехами.

Генные мутации ограничиваются нарушением строения лишь одной элементарной биологической единицы информации. Однако встречаются

и более крупные изменения, возникающие в результате разрыва хромосом с последующим их воссоединением. Так как разрывы могут происходить одновременно в разных хромосомах и даже в разных местах одной хромосомы, то вместо нормального воссоединения разорванной хромосомы возможен обмен частями между хромосомами (особенно часто между гомологичными хромосомами во время их конъюгации).

При этом возможны различные перестановки частей (транслокации, инверсии). Многие из них оказываются столь грубыми нарушениями, что ведут организм к гибели. Некоторые же из таких перестроек не нарушают заметно функций клеток и развивающегося из них организма или его частей и играют роль обычных мутаций. Особое значение в эволюции имеют, по-видимому, дупликации, т. е. удвоение частей хромосомы (что сопровождается *делецией*, т. е. выпадением части в гомологичной хромосоме; такие дефекты губительны). Дупликация несет удвоенное число целого ряда генов и дает тем самым возможность дифференцировки и интеграции ряда новых наследственных свойств.

Все такие изменения происходят вследствие действия случайных факторов (особенно часто под влиянием понижающей радиации) и относятся также к мутациям в широком смысле. Это своего рода «ошибки» в передаче информации. Их источниками являются объективно случайные внешние воздействия, которые мы должны назвать *помехами* в передаче наследственной информации. Закономерности во влиянии помех и наличие специфических средств борьбы с помехами дают нам *метод для проверки* всей изложенной концепции. Помехи как случайные факторы могут оказать лишь нарушающее влияние, и поэтому неудивительно, что отдельные мутации в большинстве оказываются вредными, а при их бесконтрольном накоплении ведут к дезинтеграции и распаду организации. Так как все вредные отклонения подвергаются элиминации (в процессе жизненного состязания), то в результате естественного отбора нормальных фенотипов (стабилизирующей формы отбора) происходит выработка более устойчивых механизмов передачи и преобразования наследственной информации. Это означает развитие защитных средств, т. е. увеличение помехоустойчивости. С другой стороны, умеренные и малые наследственные отклонения, т. е. мутации, даже необходимы для возможности перестройки наследственного кода и эволюционного преобразования всего организма. Как примирить эти два противоположных требования—надежность наследственной передачи и индивидуальную устойчивость—с возможной мобильностью популяции и эволюционной пластичностью вида в целом? В многоклеточных организмах с половым размножением эта задача разрешена удивительным образом путем разделения их жизненного цикла на две фазы: гаплоидную с простым набором хромосом и диплоидную с двойным набором (полученным путем объединения хромосом материнской и отцовской половых клеток). У животных гаплоидная фаза очень коротка и представлена только половыми клетками (у растений соотношение сложнее и интереснее). Фаза половых клеток и особенно период их созреваия использованы для возможности возникновения новых мутаций и перекомбинирования уже существующих. Это—*чувствительный период*, в течение которого наследственный код относительно менее защищен от влияния внешних факторов, т. е. от помех. Вместе с тем гаплоидный набор хромосом в половых клетках дает благоприятные условия для предварительной оценки мутаций и их комбинаций и устранения (путем естественного отбора половых клеток) слишком больших нарушений. При оплодотворении происходит восстановление диплоидного набора хромосом в новой комбинации, и из полученной таким образом зиготы развивается новая глубоко индивидуальная по своим наследственным

свойствам особь. В этой главной фазе своей жизни наследственный код оказывается прекрасно защищенным от нарушающего влияния «помех» на передачу информации.

Мутации, т. е. нарушения этой передачи, правда, возможны, но они встречаются исключительно редко, а их перекомбинирование совершенно невозможно. *Помехоустойчивость* механизма передачи наследственной информации оказывается очень высокой в течение всего жизненного цикла особи, начиная от образования зиготы и до первичных половых клеток включительно. Этим обеспечивается точная передача наследственной информации всем клеткам организма и *возможность контроля* наследственных свойств организма в течение всего жизненного пути диплоидной особи. Это происходит в сложном взаимодействии с факторами внешней среды («борьба за существование») и ведет к естественному отбору наиболее приспособленных и наиболее устойчивых особей.

Таким образом, мы видим, что разделение жизненного цикла на две фазы (гаплоидную и диплоидную) позволило удивительным образом сочетать необходимую для эволюции лабильность (в половых клетках) с необходимой для особи (и для контроля эволюционных изменений) стабильностью (в соме и в первичных половых клетках), т. е. с надежностью передачи наследственной информации.

Подобные сочетания выполнения обоих требований—*лабильности в эволюции и стабильности в особи*—осуществляются и в организации самого наследственного кода.

В технике надежность передачи информации обеспечивается весьма разнообразными средствами: во-первых, надлежащей организацией механизма и средств передачи и преобразования, во-вторых, изоляцией линии передачи от посторонних воздействий, в-третьих, системой самой передачи не в виде самостоятельных импульсов, а путем их определенного связывания (частотная модуляция в радиосвязи, слова речевой передачи, блоки закодированной информации), в-четвертых, повторением информации. В коде наследственной информации надежность ее передачи достигается теми же средствами борьбы с помехами [5]. Прежде всего для «записи» информации использован достаточно *устойчивый материал*—молекулы ДНК—и введен очень *надежный механизм передачи* этого материала путем воспроизведения точной копии каждой хромосомы с ее индивидуализированными молекулами ДНК. Этот материал совершенно равномерно распределяется в составе одинаковых хромосом при митотическом клеточном делении. Он доставляется таким образом во всей полноте каждой клетке организма и в первую очередь его первичным половым клеткам.

Материал наследственного кода очень хорошо *изолирован* от случайных внешних воздействий наличием защитной белковой оболочки, положением внутри клетки с ее многочисленными регуляторными механизмами и защитными приспособлениями всего организма в целом (вспомним только тонкую регуляцию рН крови и терморегуляцию высших позвоночных).

Наследственный код обнаруживает *высокую степень организации* в своем составе из элементарных химических единиц информации—звеньев молекулы ДНК, и элементарных морфологических единиц—генов, которые представляют индивидуализированные комплексы химических единиц информации. Гены качественно различны и занимают определенное положение вдоль хромосомы. Хотя это несомненно дискретные единицы, которые могут комбинироваться (как и буквы алфавита), они фактически при обычной передаче от клетки к клетке не комбинировются. Они связаны силами «сцепления» в различные комплексы—супергены, устойчивые части хромосом и целые хромосомы (точно так же

как буквы в письменной передаче связываются в слова, а слова—логической связью в целые предложения). Наконец, и простейшее средство повышения надежности передачи информации—ее *повторение*—нашло применение в организации наследственного кода (как самое простое средство, оно применяется всегда на начальных этапах интеграции). В некоторых случаях ген оказывается дробимым на малые единицы со сходным значением (субгены). Очевидно, мы имеем здесь повторности некоторых единиц, которые нормально связываются очень прочно. В хромосомах нередко встречаются многочисленные сходные гены (полигены) и повторения целых участков со сходным строением. Известен и механизм возникновения таких повторных участков. Это дупликации, т. е. хромосомные перестройки, нередко наблюдающиеся в результате разрыва хромосом при созревании половых клеток. Будучи нарушениями нормальной структуры хромосом, они таят в себе широкие возможности дальнейшей дифференцировки и интеграции новых структур, которые могут получить большое значение в прогрессивной эволюции. Наконец, всеобщим выражением повторности информации является *парность хромосом* (в диплоидном организме), которая может вести и к дальнейшим повторностям (в полиплоидном организме). Во всех случаях повторение элементов и блоков наследственной информации не только повышает надежность информации, но и является вместе с тем очень хорошей основой прогрессивной эволюции. Повторность информации допускает отдельные варианты каждого отдельного сообщения без нарушения всей информации в целом.

Таким образом повышается лабильность наследственного кода и возможности его перестройки в процессе эволюции. Совпадение различных методов повышения надежности информации и помехоустойчивости в технике связи и в аппарате передачи наследственных свойств организма не может быть простой случайностью (такую случайность можно было бы назвать чудом). Ясно, что как техника, так и организмы, имея перед собой аналогичные задачи, шли по сходным путям прогрессивного развития. Очевидно, это—единственные возможные пути усовершенствования средств передачи некоторых сведений и их защиты от случайных помех. Во всяком случае, основные принципы организации наследственного материала путем связывания разных дискретных единиц (генов) в длинные блоки (хромосомы) и средства достижения помехоустойчивости при его передаче удовлетворяют всем требованиям теории информации. Из этого следует, что к явлениям наследственной информации применимы и общие методы оценки качества и количества этой информации.

5. Количество наследственной информации в зиготе и в первичных половых клетках особи

Теория информации является статистической теорией, определяющей место единичного события в ряду возможных. Она предполагает [1] дискретность элементов информации и возможность их комбинирования. Известно, что с помощью L различных символов при общем числе n знаков можно получить $k=L^n$ различных комбинаций. Так как при логарифмировании этого равенства логарифм числа возможных равновероятных комбинаций оказывается прямо пропорциональным числу примененных знаков, то наиболее удобной мерой количества информации является именно логарифм числа возможных комбинаций $\log k = n \log L$. Все это справедливо для случая свободного комбинирования. При ограничении свободы комбинирования количество информации уменьшается. Оно измеряется тогда суммой отрицательных логарифмов вероятности отдельных сообщений $H = -\log p - \log p_1 - \log p_2 - \dots$ (что при равновероятности

сообщений $p=p_1=p_2=\dots=1/L$ приводит к предыдущей формуле). При 32 буквах алфавита и общем числе 75 знаков (строка машинописи) можно получить $k=32^{75}$, т. е. необозримое число комбинаций, которые могут быть использованы для такого же числа разных сообщений. В человеческой речи нет, однако, свободного комбинирования звуков, а поэтому и в письме нет свободного комбинирования букв, нет и равновероятности символов—одни звуки и буквы встречаются часто (*о, е, н, т*), другие—редко (*ю, э, щ, ф*). Буквы редко встречаются изолированно—обычно они связываются в слова. Однако и слова не комбинируются свободно—они связываются логической связью, вне которой комбинации слов не имеют смысла и не используются для обычных сообщений. Как видно, число букв алфавита собственно избыточно и если бы не удобства фонетической передачи, то можно было бы ограничиться значительно меньшим числом символов. Это и выполнено в азбуке Морзе, применяющей лишь 4 разных знака (точка, тире, перерыв и двойной перерыв между словами).

Наличие связей между буквами и словами ограничивает свободу комбинирования, а следовательно, уменьшает количество передаваемой информации. Однако вместе с тем резко увеличивается надежность информации. При свободном комбинировании букв ошибка в одной букве означала бы полное искажение всей информации. При передаче сообщений посредством фиксированных комбинаций—слов—ошибка в одной букве обычно легко распознается и, следовательно, не нарушает информации. Другое очень простое средство повышения надежности информации состоит в ее повторении. В технике для повышения надежности информации применяются, кроме системных связей и повторения, также и другие средства борьбы с помехами—изоляция и усовершенствование каналов связи.

Если мы учтем возможность передачи одного и того же сообщения путем применения различного числа разных символов (например, 32 буквы алфавита или 4 знака Морзе) и соответственно различного общего числа знаков (букв или знаков Морзе), а также разного числа блоков информации (например, слов) при разных системах кодирования, то становится ясно, что количество информации ни в какой мере не зависит от числа символов или знаков (при отсутствии свободного комбинирования). Так, например, сообщение соседей после благополучных родов, что у них родился мальчик, содержит как раз единицу информации независимо от того, сколько звуков и слов было использовано для этого сообщения. Здесь возможен был выбор только между двумя одинаково вероятными исходами (мальчик или девочка— $\log_2 1/2 = \log_2 2 = 1$). Сообщение же соседей после состоявшихся родов, что у них родился ребенок, не содержит никакой информации, так как ничего, кроме ребенка, и не могло родиться ($-\log 1 = 0$). С другой стороны, сводка погоды, при всей ее лаконичности, содержит всегда очень большое количество информации, так как составляется из большого числа отдельных сообщений с очень малой вероятностью каждого из них.

Из этих примеров видно, что количество информации не зависит от числа примененных символов и знаков. Последние зависят от способа передачи, от примененного шифра и избранной системы кодирования. Они не отражают сущности, т. е. содержания сообщения. Таким образом, количество информации определяется только числом *новых* сообщений или, иными словами, «*невероятностью*» каждого данного сообщения (отрицательным логарифмом его вероятности).

Наследственная информация должна поэтому измеряться по числу мутантных генов, а не по общему числу всех генов. Это чрезвычайно важно для популяционной генетики и при изучении эволюционных преобра-

зований, во-первых, потому, что в этих случаях нас действительно интересует только «новое», т. е. *уклонения* от нормы, и их дальнейшая судьба, и, во-вторых, потому, что общего числа генов мы никогда не знаем, а число мутантных генов может быть учтено.

Количество наследственной информации в зиготе и клетках отдельной особи определяется чрезвычайно просто. Предположим, что особь, как это бывает обычно, гомозиготна по огромному большинству генов и содержит некоторое число гетерозиготных генов Aa, Bb, Cc, \dots (т. е. генов, которые в гомологичных хромосомах не идентичны; нормальный ген обозначается большой буквой, а измененный ген той же пары аллеломорфов — малой). Каждый из этих генов возможен (в отдельной диплоидной особи) лишь в двух вариантах, обозначаемых большой и малой буквами. При образовании половых клеток эти пары разойдутся, и каждый из аллеломорфов войдет в состав одинакового числа половых клеток. Вероятность каждого аллеломорфа равна $1/2$.

Гены напоминают бинарные знаки, и их вероятность дает при двоичном логарифмировании *единицу информации* ($H = -\log_2 1/2 = \log 2 = 1$). *Наследственная информация особи определяется* поэтому просто *числом гетерозиготных генов*. Что же касается большинства генов, которые находятся в гомозиготном состоянии, то они ($LL, MM, NN \dots$) расщепляются на идентичные единицы и никакой возможности выбора для половых клеток не дают. Каждый из этих генов имеется только в одном варианте, и поэтому его информация $H = -\log 1 = 0$. Все нормальные гомозиготные гены не несут никакой информации (гомозиготный мутант aa, bb, \dots также не несет никакой информации, так как не дает каких-либо вариантов в половых клетках; однако в популяцию он вносит свою долю информации). Вся наследственная информация особи определяется только числом гетерозиготных генов.

6. Преобразование информации в индивидуальном развитии и количество обратной информации в фенотипе особи

В индивидуальном развитии наследственная информация передается без каких-либо преобразований первичным половым клеткам, и только при созревании половых клеток происходит перестройка, о которой мы уже говорили.

Соматические клетки, как правило, получают также всю наследственную информацию. Здесь возможны, однако, некоторые перестройки, связанные с дифференцировкой данных частей тела. Нередко наблюдается, например, эндомиоз, т. е. разделение хромосом без разделения клетки, что ведет к увеличению пloidности, т. е. числа наборов хромосом. Иногда при этом гомологичные хромосомы не расходятся, а остаются в продольной связи. Так получают гигантские (политенные) хромосомы клеток слюнных желез двукрылых насекомых, в которых поэтому так хорошо видна строгая последовательность структур. И в соматических клетках возможно возникновение случайных мутаций и значительных перестроек всего наследственного материала (например, в опухолях). Все это не меняет того обстоятельства, что наследственный код совершенно точно передается непосредственно от зиготы через ряд клеточных делений первичным половым клеткам и через посредство вполне закономерных перестроек — половым клеткам и зиготам следующего поколения.

В этом разделе мы будем говорить, однако, не об этом, а о характерных преобразованиях, ведущих к формированию зрелого организма, т. е. отдельной особи, ведущей самостоятельную жизнь. В каком отношении находится эта особь и ее развитие к наследственному коду? Вся сумма

имеющихся данных показывает, что развитие особи (ее фенотипа) при данных внешних условиях полностью определяется полученным ею наследственным материалом (ее генотипом). При развитии различных особей в одинаковых внешних условиях их индивидуальные различия определяются только различиями в строении их наследственного кода. В огромном количестве случаев уже точно известна связь между изменениями в известных признаках фенотипа и изменениями (мутациями) определенных генов. Иногда изучено и расположение этих генов в определенных хромосомах. Связь между начальным изменением и конечным результатом совершенно ясна. Однако промежуточные ступени в цепи этих изменений в большинстве еще неизвестны. Их изучение ведется с разных концов: со стороны факторов, определяющих выявление мутантных признаков в фенотипе (феногенетика), со стороны изучения закономерных зависимостей между процессами формообразования (экспериментальная эмбриология, механика развития) и со стороны биохимических основ индивидуального развития. Здесь имеется еще необозримое поле для дальнейшего научного исследования. Разрешение этих задач—дело будущего. Однако для наших целей достаточно и скромных достижений настоящего времени.

Отметим прежде всего двойственную функцию всего наследственного аппарата, который определяет, с одной стороны, точную передачу наследственной информации от зиготы к половым клеткам и, с другой стороны, развитие фенотипа особи как единственного реального носителя жизни. В основе этой двойственности лежит такая же двойственная функция молекулы ДНК, которая, с одной стороны, способна к точному самовоспроизведению (через расщепление и репликацию недостающего партнера) и, с другой стороны, синтезирует молекулы РНК с такой же индивидуализированной структурой, которые в свою очередь определяют синтез столь же определенных белковых веществ, включая специфические ферменты, регулирующие весь клеточный обмен веществ. В случае неклеточных или простейших одноклеточных организмов таким образом и реализуется весь фенотип особи. У высших организмов процессы протекают много сложнее, однако биохимическая их основа в сущности одна и та же—самовоспроизведение молекул ДНК, детерминация синтеза полимерных органических структур и регуляция через их посредство внутриклеточного обмена веществ. Специфические продукты этого обмена оказывают не только локальное, но и дистантное влияние на другие группы клеток. Они являются в виде стимуляторов или эвекторов, индукторов или гормонов источниками для процессов прогрессивной дифференцировки в различных частях развивающегося организма. Результатом оказываются более или менее локализованные изменения в реакциях целых клеточных масс—в скорости их роста, в перемещениях и в специфических продуктах их дифференцировки и обмена веществ. Во многих случаях обнаруживаются явные зависимости между изменениями отдельных частей (морфогенетические корреляции—множественные и многостепенные выражения плейотропии [4]). Все это—чрезвычайно сложная цепь взаимозависимостей, которая приводит к необозримо сложной системе признаков сформированного организма. Система признаков фенотипа является, таким образом, конечным результатом последовательного преобразования наследственной информации. С другой стороны, эта же система признаков является носителем обратной информации (от особей данной популяции к биогеоценозу, включающему данную популяцию). Обратная информация записана в виде символов—«признаков» в фенотипе особи, совершенно так же, как прямая информация записана в виде символов—генов в наследственном коде. Между этими разными символами—признаками

и генами нет ничего общего (так же, как трудно найти что-либо общее между буквами фонетического алфавита, знаками Морзе и китайскими иероглифами), но связь между ними несомненна. Связь эта—результат преобразования, т. е. перехода с одной системы кодирования на другую (как, например, при переводе с русского языка на китайский). Однако этот переход столь сложен, что мы не имеем никакого права говорить, что определенный ген вызывает развитие известного признака. Это было бы недопустимым упрощением. Хотя изменение отдельного гена иногда действительно кажется сосредоточенным в виде изменения отдельного признака, это впечатление все же обманчиво, так как всегда имеются, на глаз, быть может, малозаметные, но нередко гораздо более существенные физиологические изменения, которые сказываются на жизнеспособности мутантов в разных условиях температуры, влажности, питания и т. п.

Гораздо правильнее поэтому утверждение, что изменение каждого гена влияет на все признаки и на всю организацию, и, с другой стороны, любой признак фенотипа зависит в конце концов от всего генотипа в целом. Таким образом, весь механизм преобразования оказывается как будто безнадежно сложным. В ходе онтогенетического преобразования играют свою роль все гены и именно весь интегрированный геном. Наследственная информация преобразуется в целом, т. е. все сообщение (точно так же как при точном переводе на иностранный язык переводится все сообщение независимо от количества содержащейся в нем информации и числа использованных символов и знаков). Если мы говорили, что гомозиготные гены не несут никакой информации, то это относится только к вопросу об учете количества информации, но не к вопросу о ее преобразовании. Несомненно, что *в преобразовании гомозиготные гены несут самую ответственную функцию, определяя развитие всех основ нормальной организации. Однако они не несут в себе условий для развития наследственных вариантов этой организации.* Между тем нас при рассмотрении вопросов эволюции (как и в теории информации) интересуют именно только эти варианты. Это дает нам ключ к разрешению всего вопроса о преобразовании ценной для нас информации и вопроса об измерении количества обратной информации.

Мы можем здесь затронуть вопрос о характере обратной информации лишь в самой общей форме [6, 7]. Обратная информация выражается в формах жизнедеятельности организма, что является характеристикой фенотипа, т. е. особи в целом (точно так же как наследственная информация выражается в процессах внутриклеточного синтеза и регуляции обмена, т. е. в формах жизнедеятельности клетки). Количество же этой информации определяется только количеством вариантов—уклонений, несущих в себе что-либо «новое», т. е. какие-либо отличия от нормы.

Определение общего числа признаков организации невозможно да и не нужно. Определение числа возможных вариантов различных признаков для наших целей также не нужно. В передаче обратной информации через фенотипы особей, в их оценке в биогеоценозе (контроль фенотипов в их жизненном соревновании) и в естественном отборе каждая особь является единственным самостоятельным неразложимым носителем жизни. Поэтому при рассмотрении эволюционных проблем нас интересуют только варианты всей организации в целом. Вместе с тем, только эти варианты и могут служить основой для оценки качества и количества обратной информации. Однако непосредственное *определение количества возможных вариантов фенотипа* также недостижимо без сложных фенотипических исследований.

Практически более легким является *путь теоретических расчетов по количеству наследственной информации на основании известных уже закономерностей преобразования.* В конкретных случаях это требует

конечно, генетического анализа состава популяции (т. е. определения степени ее насыщения различными мутациями).

Мы начнем рассмотрение закономерностей преобразования с самого простого несколько абстрактного случая чисто *мозаичного развития*, т. е. развития, полностью детерминированного строением зиготы, независимого от обычных отклонений во внешних факторах и не обладающего механизмами морфогенетической регуляции. В полном согласии с последним условием мы принимаем также, что все мутантные гены полудоминантны, т. е. проявляются в гетерозиготе в своем половинном выражении (по сравнению с гомозиготой). После этого рассмотрения мы внесем поправки в отношении возможной зависимости развития от факторов внешней среды и регуляторного типа развития.

При отсутствии регуляции изменение любого гена вызывает какое-то изменение фенотипа (является ли это локализованным морфологическим изменением в одном или во многих признаках или физиологическим изменением частного или общего характера, в данном случае совершенно безразлично). Вместе с тем изменение одного гена ведет при отсутствии меняющего влияния внешних факторов только к одному изменению фенотипа. Явление так называемой *плейотропии* не меняет положения, так как, на скольких бы признаках ни сказывалась мутация, она всегда означает только один вариант всего фенотипа в целом. С другой стороны, наличие полигенных систем также не меняет указанного соотношения, так как изменение каждого отдельного гена такой системы вызывает хотя бы и очень незначительное изменение фенотипа, т. е. опять-таки один вариант. Таким образом, число возможных вариантов фенотипа вполне однозначно определяется числом вариантов генотипа. В этом случае *количество обратной информации равно количеству наследственной информации*.

Мозаичное развитие, конечно не в столь резкой форме, весьма распространено среди беспозвоночных животных и, в частности, характерно для насекомых. О полной независимости от изменений во внешних факторах и об отсутствии всякой регуляции говорить, однако, не приходится. Попробуем поэтому учесть изменяющее влияние внешних факторов и регуляторных систем самого организма.

При рассмотрении *значения внешних факторов* нам нужно разграничивать действие закономерных факторов локального или сезонного характера и влияние случайных факторов. В первом случае все особи данной популяции, находящиеся в данном месте, одинаково подвергаются действию этих факторов, и если между ними имеются различия, то они обусловлены индивидуальными различиями в норме реакции, т. е. генетическими различиями. Так как в борьбе за существование между собой сталкиваются (конкурируют) в основном именно такие бок о бок живущие особи, то решающее значение при этом имеют генетические различия. В одинаковых условиях генетически идентичные особи не дадут никаких вариантов, и поэтому внешние факторы не окажутся в роли источников информации.

Однако *случайные* различия во внешних факторах могут оказаться причиной случайных фенотипических различий даже между генетически идентичными особями. Это увеличивает разнообразие форм, а следовательно, и количество обратной информации. Но эта информация не является в нормальных условиях среды (по исчезновении случайного отклонения) правильным отражением генотипа. Это будет «ложная» информация, которая должна вычитаться из общего количества информации как величина отрицательная. При дальнейшем контроле фенотипов в биогеоценозе и в процессе естественного отбора рано или поздно выявится

несоответствие фенотипа данным условиям существования, и такие случайно неблагоприятно реагировавшие особи с большой вероятностью будут элиминироваться. Во всяком случае эволюция не строится на результатах таких индивидуальных реакций.

Как мы видим, увеличение количества информации в результате действия случайных внешних факторов является кажущимся. Это—*нарушающее влияние помех, искажающее информацию*, а не увеличивающее ее количество. Эволюция может идти лишь по пути развития «помехоустойчивости», так как точное соответствие фенотипических различий генотипическим в определенных условиях внешней среды является обязательной предпосылкой возможности правильного контроля и отбора в биогеоценозе [7]. Гораздо большее значение для учета количества обратной информации имеет *наличие регуляторных механизмов*.

Мы поставили в числе условий нашего идеализированного случая мозаичного развития полное отсутствие регуляции. Этого никогда не бывает. Одним из наиболее ясных выражений регуляции является почти всеобщее доминирование нормы.

При полном доминировании нормы гетерозиготные мутации не имеют своего выражения и, следовательно, являются фенотипически нормальными. В этом случае количество обратной информации в таких особях падает до нуля и резко расходится с количеством наследственной информации. Это, однако, также идеализированный случай. Обычно действительно новые мутации бывают полудоминантными. В процессе эволюции вредные мутации становятся постепенно рецессивными. Это происходит в результате отбора других мутаций, подавляющих их выражения. С другой стороны, полезные мутации становятся постепенно доминантными в результате отбора более активных аллелей или генов-модификаторов, усиливающих их выражение [10]. Такие мутации быстро включаются в состав нормы. Мутации, фактически встречающиеся в популяции как отклонения от нормы, обычно рецессивны. Абсолютной рецессивности, однако, вероятно, никогда не бывает. Учитывая наличие в популяции разных мутаций—очень редких полудоминантных и весьма частых рецессивных, мы должны прийти к выводу, что количество обратной информации в фенотипе особи во всяком случае всегда значительно ниже наследственной информации той же особи.

Такую же роль, как доминирование нормы, играют и все другие регуляторные механизмы. Благодаря существованию известного запаса в пороговом уровне нормальной реактивности тканей развивающегося организма, преодоление этого уровня требует значительной концентрации ненормальных агентов, чтобы вызвать явно выраженное отклонение. Поэтому небольшие мутации могут не получить своего выражения даже в гомозиготном состоянии. Они вызовут лишь подпороговый сдвиг, который при суммировании мутаций со сходным действием может привести при преодолении порога сразу к хорошо выраженной мутации.

Таким образом, все регуляторные механизмы формообразовательной системы организма приводят к значительному снижению количества информации в фенотипе отдельной особи. *При регуляторном типе развития (а всякое развитие организма есть в значительной мере авторегуляция) количество обратной информации в фенотипе всегда значительно ниже количества наследственной информации той же особи.* Это означает высокую стабильность организма и в особенности характерно для высших позвоночных животных.

Все сказанное относится, однако, только к отдельной особи. Количество информации в популяции определяется другими величинами и подчиняется иным закономерностям.

7. Количество информации в популяции и закономерности его изменения

Вопрос о количестве наследственной информации в популяции разрешается довольно просто. Большинство генов гомозиготно и распределяется в качестве таковых по всем особям популяции. Эти гены не являются основой многообразия форм, они не расщепляются, не несут никакой информации и в расчет не принимаются. Мутантные гены распределяются неравномерно. Некоторые встречаются часто, другие—очень редко. Количество информации, приходящееся на долю каждого мутантного гена (a, b, c, d, \dots), определяется отрицательным двоичным логарифмом вероятности нахождения его в популяции: $J = -\log_2 p$.

Как видно, информация, которую несет мутантный ген, наиболее велика у редких генов (в них максимум новизны или невероятности). При концентрации одного мутантного гена на 1 000 000 гамет *) $J = 19,93$. По мере накопления одних и тех же мутаций в популяции количество информации довольно быстро падает (при увеличении концентрации одних и тех же генов в 10 раз количество информации падает на 3,32 единицы). При 1% концентрации мутантного гена $J = 6,64$. При 50% встречаемости данной мутации в популяции количество информации падает до 1 и при 100% насыщении—до 0. В последнем случае новый ген вошел в состав устойчивой нормы и расщепления не дает. Конечно, такое увеличение концентрации определенного гена и уменьшение количества информации указывает на положительное его значение и всегда представляет результат интегрирующего действия естественного отбора.

При наличии в популяции известного числа *разных* мутантных генов количество информации по этим генам просто суммируется, и это дает очень большие величины информации, если гены довольно редки (при 10 мутациях с концентрацией каждого гена в 1 на 1 000 000 гамет $J = 199,32$). Редкость отдельных мутаций указывает на их элиминацию в процессе стабилизирующего отбора, а большое число разных мутаций—на дезорганизующую роль мутационного процесса. Неустойчивое состояние популяции является показателем нарушения нормальных соотношений с внешней средой. Оно может быть хорошей основой для восстановления нарушенных соотношений путем накопления некоторых мутаций и их комбинаций в результате действия естественного отбора. Это всегда сопровождается снижением количества информации (в пределе, т. е. при полной гомозиготности и максимальной иммобильности популяции,—до 0). Полная гомозиготность возможна, однако, только по наиболее существенным генам, лежащим в основе развития жизненно важных свойств организма. По другим генам всегда сохраняется известный уровень их встречаемости в популяции, определяющий наличие индивидуальных вариантов организации. Эти варианты являются объектом постоянного контроля в биогеоценозе и через посредство естественного отбора служат основой эволюционной пластичности данного вида организмов.

Мы видели, что при строго мозаичном развитии, т. е. при отсутствии какой-либо регуляции, количество фенотипической информации равно количеству наследственной информации. Это возможно, однако, только в гаплоидном организме, так как в диплоидности содержатся условия для увеличения многообразия форм и, следовательно, для увеличения количества информации (за счет появления редких гомозигот). Но другие

*) Концентрация мутантного гена исчисляется по отношению его численности к общей численности данного гена. Поэтому 1% концентрации означает нахождение одного мутантного гена в 100 гаметах, в 100 гаплоидных особях или в 50 диплоидных особях.

регуляторные механизмы, вызывающие снижение фенотипического проявления мутаций, приводят также к уменьшению количества обратной информации в популяции.

При наличии регуляторных механизмов индивидуального формообразования количество информации в отдельных фенотипах резко сокращается. Но в популяции это сокращение незначительно. В случае неполного доминирования мутаций, когда все они обладают каким-то выражением в фенотипе, информация, которую несут гетерозиготы, равняется наследственной информации, определяемой отрицательным логарифмом вероятности нахождения соответствующего гена в популяции. Однако появляющиеся изредка гомозиготы имеют иной фенотип, несут, следовательно, иную информацию и должны учитываться отдельно. Как редкое явление, они значительно увеличивают общее количество обратной информации.

При известной концентрации гетерозиготных мутантов в популяции можно вычислить вероятное число гомозиготных мутантов по формуле Пирсона—Гарди. Точно так же при эмпирически найденной встречаемости гомозиготных мутаций с полным выражением ее признаков можно вычислить по той же формуле степень насыщенности популяции гетерозиготными мутантами (со сдвигом, вызываемым действием естественного отбора, можно пока не считаться).

При неполном доминировании обратная информация, вносимая данной мутацией в популяцию, определяется суммой отрицательных логарифмов концентраций (т. е. вероятности нахождения) гетерозиготной и гомозиготной мутаций. При полной рецессивности мутации вносимая ею обратная информация определяется одним только отрицательным логарифмом концентрации гомозиготной мутации (так как гетерозиготы имеют нормальный фенотип). Количество информации, вносимой разными мутациями, просто суммируется. В обоих случаях это дает значительное превышение обратной информации (по фенотипам) по сравнению с наследственной информацией по тем же генам.

Приведем некоторые примеры. При 2% концентрации гетерозигот встречаемость гомозиготной мутации будет около 1 на 10 000 особей. Концентрация мутантного гена—101 на 10 000 гамет. Количество наследственной информации (по данному гену):

$$J = -\log_2 0,0101 = \log_2 10\,000 - \log_2 101 = 6,63.$$

Количество фенотипической информации при полудоминировании:

$$J_1 = -\log_2 0,02 - \log_2 0,0001 = 5,65 + 13,29 = 18,94.$$

Количество обратной информации при полной рецессивности:

$$J_2 = -\log_2 0,0001 = 13,29.$$

Мы взяли пример исключительно высокой концентрации определенной мутации. Если взять пример более обычной концентрации—2 гетерозиготы на 10 000 особей и соответственно 1 гомозигота на 100 000 000 особей, то количество наследственной информации по данному гену $J = \log_2 10\,000 = 13,29$. Количество обратной информации при неполном доминировании $J_1 = \log_2 5\,000 + \log_2 100\,000\,000 = 12,29 + 26,59 = 38,88$. При полной рецессивности та же концентрация мутации дает информацию по фенотипам: $J_2 = \log_2 100\,000\,000 = 26,59$. В обоих случаях количество обратной информации значительно выше, чем количество наследственной информации. Оно особенно велико при неполном доминировании, и это отражает многообразие фенотипов в этом случае.

Наследственная информация определяется по количеству гетерозиготных особей в популяции (наличие гомозигот на этом почти не отра-

жается; в первом примере они учтены—они дают уменьшение информации на 0,01). При определении обратной информации в случае неполного доминирования учитывались как гетерозиготы, так и гомозиготы, а в случае полной рецессивности—только гомозиготы. Разница между обоими величинами равняется поэтому величине наследственной информации:

$J_1 - J_2 = J$, в последнем примере $38,88 - 26,59 = 12,29$.

В диплоидном организме количество фенотипической информации по гетерозиготам на единицу меньше количества наследственной информации, так как первая исчисляется по концентрации гетерозиготных особей в популяции, а вторая—по концентрации мутантных генов в хромосомах или гаметах. Концентрация мутантных генов вдвое ниже, чем концентрация гетерозигот в особях популяции, так как каждая гетерозиготная особь содержит в двух гомологичных хромосомах два гена: один мутантный и один нормальный. При воспроизведении такая особь дает равное число мутантных и нормальных половых клеток. Хотя общее количество обратной информации в популяции выше количества наследственной информации, рецессивность, связанная с подавлением выражения мутаций в гетерозиготе, снижает количество фенотипической информации.

Это подавление выражения мутаций и снижение количества информации есть результат действия *стабилизирующей формы естественного отбора*, которая ведет к полному доминированию нормы (рецессивности мутаций) и развитию других регуляторных механизмов, защищающих развитие нормы при наличии повреждающих влияний (помех). При высоком развитии регуляторных систем подавляется выражение небольших мутаций даже в гомозиготном состоянии, и это ведет к дальнейшему *снижению* количества фенотипической информации. По малым мутациям количество информации бывает ниже, так как они легко достигают более высоких концентраций.

Увеличение или уменьшение концентрации полностью рецессивных генов идет в значительной мере путем «дрейфа». Однако возможно и действие естественного отбора, который имеет точку своего приложения не на отдельных мутациях, а на интегральном их эффекте (даже в гетерозиготном состоянии), выражающемся в фенотипе целой особи.

Естественный отбор в его движущей форме ведет к включению в состав нормы новых мутаций и их комбинаций, приобретающих в данных условиях существования положительное значение. Постепенное насыщение популяции такими мутациями приводит к общему *уменьшению количества информации как наследственной, так и обратной*. Большое количество наследственной информации указывает на мобильность популяции, на ее легкую изменяемость, по меньшей мере путем «дрейфа». Большое количество обратной (фенотипической) информации является показателем больших возможностей для деятельности естественного отбора.

При отсутствии всяких регуляторных механизмов (и особенно в гаплоидном организме) элиминация всех наследственных отклонений от нормы привела бы к крайней редкости мутантов в популяции и к значительному уменьшению эволюционной пластичности (мобильности) данного вида организмов. Однако стабилизирующая форма естественного отбора, основанная на элиминации всех случайных нарушений, возникающих под влиянием внешних факторов (помех), ведет именно к развитию разнообразных регуляторных механизмов, выражающемуся прежде всего в доминировании нормы. Это хотя и снижает количество обратной информации, однако способствует накоплению мутаций в скрытом виде и тем самым повышает эволюционную пластичность популяции. Вместе с тем повышается устойчивость фенотипов отдельных особей (и уменьшается количество информации в особи).

Таким образом, стабилизирующая форма естественного отбора приводит в популяции и в особи к двум противоположным результатам: с одной стороны, к значительной мобильности, т. е. эволюционной пластичности популяции, и, с другой стороны, к максимальной устойчивости отдельных особей.

При известной концентрации мутантных генов в популяции нетрудно вычислить *среднюю информацию* на один ген и на наследственный код одной особи. Средняя информация на один символ определяется формулой $H_i = -p \log p_i$. Средняя информация целого сообщения $H = -\sum p_i \log p_i$, где p_i есть вероятность каждого элементарного сообщения (символа). Вероятность отдельного гена в популяции определяется его концентрацией. При концентрации известного гена 1 на 1 000 000 гамет $H_i = -0,000001 \log_2 0,000001 = 0,00002$. При концентрации 1 на 10 000 гамет $H_i = 0,0013$. При 1% концентрации $H_i = 0,0664$, при 10% $H_i = 0,332$, при 25% и при 50% $H_i = 0,5$, при 70% $H_i = 0,357$, при 90% $H_i = 0,135$ и при 100% $H_i = 0$. Однако значение средней информации на ген достигает максимума (0,531) при его концентрации, равной 36,8% общего числа гамет. При повышении концентрации до 100% количество средней информации на ген падает до 0. Однако и при понижении концентрации гена количество средней информации на ген падает в пределе до 0.

По всему наследственному коду средняя информация дается суммированием средних информаций по отдельным генам $H = -\sum p_i \log p_i$. Разумеется, эта сумма будет максимальной при максимальной величине по всем генам.

Средняя информация на сообщение представляет энтропию этого сообщения. Таким образом, *энтропия наследственного кода особи достигает максимума, когда все гены находятся в гетерозиготном состоянии* (концентрация каждого мутантного гена равна 50%). В этом случае неопределенность индивидуальной структуры наследственного кода достигает наибольшей величины (по любому гену каждый аллеломорф одинаково вероятен)—имеется полная свобода выбора и комбинирования. Вместе с тем имеются наиболее благоприятные условия для эффективного действия естественного отбора. Энтропия наследственного кода достигает минимума, когда все гены в особях всей популяции гомозиготны ($H=0$), или, по меньшей мере, если мутантные гены очень редки. Это всегда есть результат действия естественного отбора.

На основании закономерностей преобразования информации во время индивидуального развития можно от количества средней информации на наследственный код особи перейти и к количеству средней информации в фенотипе отдельной особи.

Количество средней информации на фенотип особи подчиняется, очевидно, той же закономерности, что и количество средней наследственной информации, т. е. ее максимум совпадает с максимальной гетерозиготностью особей, когда все варианты фенотипа становятся одинаково вероятными. Количество средней информации на особь, а следовательно, и энтропия фенотипа падают, когда концентрация некоторых вариантов подымается выше 36,8%, т. е. они начинают входить в состав нормы. Это—результат действия движущей формы естественного отбора, которая меняет генотип и фенотип нормы. С другой стороны, количество средней информации и энтропия фенотипа снижаются также, когда вероятность некоторых вариантов падает ниже 36,8%, т. е. они становятся более редкими. Это—результат действия стабилизирующей формы отбора, которая ведет к большей устойчивости данной нормы. Обе формы естественного отбора приводят, следовательно, к уменьшению количества средней информации на особь и к снижению количества энтропии в ее фенотипе.

Расхождение в количестве средней информации в наследственном коде и в фенотипе особи начинается с развитием регуляторных механизмов, снижающих выражение наследственных изменений в фенотипах. При полном доминировании нормы, когда рецессивные мутации не имеют своего выражения, максимум средней информации (и максимум энтропии) лежит на более высоком уровне концентрации мутантных генов, именно на уровне 36,8% выраженных вариантов, т. е. гомозиготных мутаций. Это соответствует приблизительно 60% концентрации мутантного гена в гаметах. В этом случае имеется максимум свободы выбора вариантов фенотипов и наибольшая возможность эффективного действия естественного отбора.

Из всех приведенных методов учета количества информации *наибольший теоретический интерес имеет определение количества средней наследственной информации на особь*. Вместе с тем именно эта величина практически наиболее доступна для точного учета.

Индивидуальная характеристика особи в виде количества ее наследственной информации не имеет общего интереса, так как всегда случайна и может сильно уклоняться от средней нормы, а величина информации во всей популяции слишком сильно зависит от различных редких мутаций, которые иногда не могут быть учтены. Между тем величины средней наследственной информации и энтропии наследственного кода, с одной стороны, *характеризуют степень организованности и устойчивости особи* и, с другой стороны, как средние величины отражают и состояние всей популяции в целом.

Количество средней информации на особь зависит в основном от концентрации обычных мутаций, которые имеют наибольшее значение в генетической динамике популяции и вместе с тем учитываются с наибольшей легкостью. Очень редкие мутации оказывают на эту величину весьма незначительное влияние.

8. Эволюция наследственного кода, фенотипа особи и структуры популяции

Мутабельность, т. е. процесс возникновения новых мутаций, ведет к увеличению количества как наследственной, так и обратной информации. Это является показателем *дезорганизации* данной системы, т. е. наследственного кода, фенотипа особи и как генетической, так и фенотипической структуры всей популяции.

Движущая форма естественного отбора вызывает обратный эффект. Она ведет к накоплению лишь некоторых мутаций и к включению в состав нормального фенотипа таких изменений, которые в условиях данного биогеоценоза получили благоприятную оценку. Тем самым естественный отбор приводит также к насыщению популяции некоторыми гетерозиготными и гомозиготными наследственными изменениями. Строение наследственного кода при этом усложняется, а количество наследственной информации (по мере отбора, т. е. накопления положительных уклонений) как в особи, так и в популяции снижается.

В новых условиях данного биогеоценоза положение особей и всей популяции в целом становится более устойчивым.

Стабилизирующая форма естественного отбора ведет к совершенно иным результатам как в преобразовании наследственного кода, так и в организации фенотипа отдельной особи и всей популяции в целом. Прежде всего жесткая элиминация наследственных уклонений от нормы делает их более редкими. Это означает *нормализацию особей*, т. е. уменьшение количества информации в наследственном коде и в фенотипе. Это означает также *нормализацию популяции*, в которой, однако, количество как

наследственной, так и фенотипической информации увеличивается (за счет редкости уклонений от нормы). Кроме того, через элиминацию всех случайных уклонений, т. е. в результате отбора в пользу большей устойчивости нормы против всех влияний, нарушающих передачу и преобразование наследственной информации, возникают разные *защитные механизмы*, увеличивающие *помехоустойчивость* наследственного кода.

Основные средства защиты от помех сводятся к: 1) установлению повторностей в строении (полигены, повторения участков хромосом) как в отдельных хромосомах (гетероплоидия), так и в особенности в целых наборах их (диплоидия и иногда полиплоидия); 2) упорядочению системных связей, регулируемых силами «сцепления», внутри хромосом; 3) установлению максимально точного механизма митотического деления клетки; 4) защите наследственного кода от случайных помех его положением в ядре клетки и организацией регуляторных систем клетки и всего организма в целом. Борьба с помехами означает упорядочение всей системы передачи наследственной информации и общее увеличение ее надежности.

Стабилизирующая форма естественного отбора ведет также к установлению более надежного механизма преобразования информации в индивидуальном развитии. Точность преобразования наследственной информации приводит в обычных условиях развития к полному равенству количества наследственной и фенотипической информации в отдельной особи.

Через развитие регулирующих механизмов стабилизирующая форма естественного отбора ведет к защите нормального формообразования от нарушающего влияния случайных изменений во внешних факторах. Это означает *автономизацию* индивидуального развития и стабилизацию фенотипа особи. Особь приобретает максимальную устойчивость, и ее фенотип является в этом случае наиболее надежным выразителем наследственной информации.

Высокое развитие регуляторных механизмов, обеспечивающее прежде всего доминирование нормы, допускает, однако, более свободное накопление разнообразных мутаций, по меньшей мере в гетерозиготном состоянии. Это вызывает прежде всего увеличение количества наследственной информации в отдельных особях. Количество обратной информации, конечно, также увеличивается, но чем выше развитие регуляторных систем, тем больше расхождение между количеством наследственной информации и обратной информации в отдельных особях. В результате развития защитных механизмов, подавляющих проявление неблагоприятных наследственных изменений (мутаций), *количество обратной информации в фенотипах всегда значительно ниже количества наследственной информации* и это указывает на высокую степень *устойчивости* особи.

В *популяциях* элиминирующее влияние отбора в пользу нормы приводит прежде всего к тому, что мутации становятся редкими, а следовательно, количество как наследственной, так и обратной информации не уменьшается (как при положительном отборе), а увеличивается. В диплоидном организме каждая мутация может получить два выражения соответственно гомозиготе и гетерозиготе. Это означает увеличение многообразия фенотипов и, следовательно, *увеличение количества информации*.

В гетерозиготе вследствие наличия нормального аллеломорфа мутация не имеет полного выражения. Поэтому редкие гомозиготы являются носителями нового фенотипа, и это резко повышает количество обратной информации в популяции. Количество фенотипической информации тогда значительно превышает количество наследственной.

Однако при полной рецессивности мутаций, что является непосредственным результатом деятельности стабилизирующего отбора, *количество*

обратной информации несколько снижается за счет включения фенотипа гетерозиготных особей в состав нормального фенотипа. При дальнейшем повышении порогового уровня нормальной реактивности тканей, т. е. еще более совершенной регуляции, небольшие гомозиготные мутации могут не получать своего выражения, и тогда количество обратной информации еще снизится. Наконец, высокое развитие регуляции способствует и накоплению многих одинаковых мутаций в популяции, что приводит к уменьшению наследственной информации. В обоих случаях это означает увеличение эволюционной пластичности популяции (вида).

Таким образом, стабилизирующая форма естественного отбора ведет к *максимальной устойчивости особи* (при очень небольшом количестве обратной информации, которая поэтому в фенотипе всегда значительно ниже наследственной) и вместе с тем к *максимальной пластичности* (мобильности) *популяции и вида в целом* (причем в популяции количество наследственной информации может и не возражать, а количество обратной информации поддерживается в диплоидном организме на уровне, превышающем количество наследственной информации).

Устойчивость означает упорядоченность системы. Таким образом, стабилизирующая форма естественного отбора ведет к упорядоченности в строении наследственного кода и к упорядоченности в строении всего организма в целом. С другой стороны, тот же стабилизирующий отбор поддерживает высокий уровень многообразия в структуре всей популяции.

Количество средней информации как мера многообразия, свободы выбора, т. е. неопределенности и беспорядка, является вместе с тем и мерой энтропии. Э. Шредингер указывал, что в живых организмах энтропия поддерживается на низком уровне, и полагал, что это достигается использованием источников полезной энергии окружающей среды (химической энергии органических веществ животными и световой энергии растениями). Если это может служить объяснением возможности поддержания энтропии на низком уровне при жизни организма—особи, то это все же не объясняет нам происхождения самой организованности, т. е. очевидного снижения энтропии в процессе эволюции живых существ.

Если рассматривать второй закон термодинамики в его статистической интерпретации, то энтропия является количественной мерой атомной, молекулярной или любой другой неупорядоченности, к которой стремится всякая изолированная система. Количество информации измеряется логарифмом возможного многообразия. Этой же величиной измеряется и энтропия. Она также выражает возможное многообразие. Энтропия, как и количество информации, является показателем свободы выбора, т. е. мерой неопределенности, неорганизованности и беспорядка. Вместо термина «количество информации» поэтому нередко употребляется термин «энтропия информации». Вместо «количество информации» мы можем прямо поставить «уровень энтропии», добавив лишь для перехода коэффициент пропорциональности. Таким образом, можно говорить об энтропии наследственного кода (как носителя прямой информации) и об энтропии фенотипа (как носителя обратной информации).

Процесс *мутирования* нарушает организованность живых систем — он вносит беспорядок, дезорганизацию. Степень этой дезорганизации и учитывается путем измерения количества информации или уровня энтропии. Несомненно, что естественный и закономерный процесс мутирования как дезорганизующий фактор увеличивает энтропию организмов. Точно так же влияют и все случайные внешние факторы, которые мы обозначали как помехи. Дезорганизующему влиянию мутирования противодействует, однако, естественный отбор, не допускающий накоп-

ления большого количества разных мутаций в популяции или, во всяком случае, снижающий его отрицательное влияние на жизненную устойчивость отдельных особей.

Мутации, вредное влияние которых отражается на жизнеспособности особи даже в гетерозиготном состоянии, т. е. полудоминантные мутации, усиленно элиминируются и потому всегда остаются весьма редкими. Огромное большинство особей оказывается по фенотипам вполне нормальными и устойчивыми. Количество информации в них невелико. Но само наличие именно редких мутаций поддерживает количество наследственной (и обратной) информации в них на высоком уровне (особенно при большом их многообразии).

Вместе с тем малые мутации с незначительным выражением могут свободно накапливаться. Не являясь объектом непосредственного контроля, их концентрация подвергается случайным изменениям, особенно при колебаниях численности популяций в пространстве и во времени (генетико-автоматические процессы или «дрейф»).

Повторно встречающиеся мутации с заметным выражением при постоянном комбинировании их с другими мутациями постепенно теряют свое выражение вследствие элиминации особей с более значительным выражением неблагоприятных особенностей данной мутации. Таким образом, стабилизирующая форма отбора приводит к рецессивности мутаций, т. е. к полному доминированию нормального фенотипа (эволюция доминантности по Фишеру, Холдену и Райту).

Так как большинство мутаций обладают более или менее ярко выраженным плейотропным действием, то возможен положительный отбор таких мутаций при условии подавления их вредных выражений. При полной рецессивности происходит, однако, свободное накопление разных, даже очень вредных, мутаций в скрытом виде. Количество наследственной информации в особях и в популяции при этом может увеличиваться. Однако количество фенотипической информации в популяции хотя и снижается, но остается все же на более высоком уровне по сравнению с наследственной информацией. Это объясняется тем, что, кроме гетерозиготных мутаций, появляются редкие гомозиготы с отличающимся от нормы фенотипом, которые поддерживают количество информации на высоком уровне. Интенсивная элиминация гомозигот препятствует дальнейшему накоплению таких мутаций.

Таким образом, причиной многообразия форм в популяции является, конечно, процесс мутирования. *Стабилизирующая форма естественного отбора* препятствует накоплению одинаковых мутаций, переводит наследственное многообразие особей в скрытое состояние и всегда *поддерживает количество наследственной информации в популяции на довольно высоком уровне. На еще более высоком уровне поддерживается и количество обратной информации в фенотипах популяции.* Следовательно, энтропия популяции остается высокой. Популяция является мало организованной биологической системой, и этот низкий уровень организации, т. е. некоторый беспорядок и неопределенность, поддерживается действием стабилизирующего отбора. Этим самым поддерживается высокая эволюционная пластичность популяции и вида в целом. В случае изменения соотношений между популяцией (видом) и внешней средой (биогеоценозом) нормальные особи теряют свою приспособленность. Стабилизирующий отбор в известных отношениях (по признакам, утратившим свое значение) прекращается, и это ведет к увеличению числа разнообразных мутаций. Резко увеличивается количество информации в отдельных особях, организация расшатывается. Однако некоторые мутации и их комбинации могут получить в новых условиях среды положительную ценку. Это ведет к свободному

их накоплению под руководящим влиянием *движущей формы естественного отбора*.

Накопление одних и тех же мутаций означает снижение количества как наследственной, так и обратной информации в популяции. По мере включения таких мутаций в состав нормы, т. е. перехода их в гомозиготное состояние, резко *снижается количество наследственной информации в особях* и обратной информации в их фенотипах. При установлении новой нормы вновь повышается значение стабилизирующей формы естественного отбора, которая ведет к дальнейшему снижению количества наследственной и обратной информации в отдельных особях (за счет развития регуляторных систем), однако поддерживает количество наследственной, и особенно обратной, информации во всей популяции на относительно высоком уровне.

Стабилизирующая форма естественного отбора ведет через развитие регуляторных механизмов (включая доминирование нормы) к высокой устойчивости как наследственного кода, так и всей особи (фенотипа) в целом. Но наличие тех же механизмов способствует накоплению разных мутаций в гетерозиготном состоянии. В случае переоценки значения мутаций при изменении соотношений с внешней средой в распоряжении движущей формы естественного отбора сразу окажутся относительно высокие концентрации необходимых уклонений. Это обеспечивает возможность достаточно высоких темпов эволюции.

Таким образом, *стабилизирующая форма отбора ведет* собственно к двум разным, но одинаково важным результатам: *к максимальной устойчивости особи и возможной мобильности, т. е. эволюционной пластичности популяции*. Эти два результата коренятся в двух принципиально различных методах повышения помехоустойчивости наследственного кода и фенотипа особи в целом. Таковыми являются:

I. *Системная организация* наследственного кода из генов, связанных силами сцепления в целостных хромосомах, и *механизм митоза* с его точным распределением материала по дочерним клеткам. Организация автономно-«мозаичного» механизма индивидуального развития.

II. *Повторности* в организации наследственного кода—повторение одинаковых генов (полигены), одинаковых участков хромосом и особенно *диплоидия* (а иногда также и полиплоидия).

Второе средство повышения надежности информации, допускающее уже известную изменчивость без нарушения нормальных функций особи, привело в условиях межгруппового состязания в темпах эволюции к созданию дальнейших механизмов, обеспечивающих мобильность популяции. Диплоидность допускала возможность нарушения системных связей наследственного кода путем разрыва сцепления и организованность этих разрывов в мейозе (перекрест). Она же привела к обеспечению комбинирования в половом процессе. Вместе с тем диплоидность дала возможность накопления мутаций в гетерозиготах и послужила основой для развития регуляторных механизмов клетки (начиная с простого доминирования нормы). На этой же базе развился и автономно-регуляторный тип развития организма.

Э. Шредингер указывал на низкий уровень энтропии в живых организмах. Он предполагал, что в жизни особи низкий уровень энтропии поддерживается за счет внешних источников ценной энергии. Другими словами, низкий уровень энтропии живых существ достигается за счет его значительного повышения в окружающей среде.

Каким образом, однако, создались такие изумительные механизмы преобразования энергии, которые используют только ценные формы энергии и выбрасывают обесцененную энергию во внешнюю среду? И каким

образом эти механизмы совершенствуются—переходят на высший уровень упорядоченной организации, т. е. снижают уровень своей энтропии по мере своей прогрессивной эволюции? Это совершенствование живых систем происходит, несомненно, в результате *естественного отбора*.

Более *активные* особи, лучше использующие ресурсы внешней среды для роста, жизни и размножения, вытесняют путем своего размножения менее активных особей. Более *устойчивые* особи, т. е. лучше противостоящие различным вредным влияниям, также вытесняют путем преимущественного размножения менее устойчивых особей. В обоих случаях более упорядоченные формы организации с более низким уровнем энтропии вытесняют менее упорядоченные формы организации с более высоким уровнем энтропии.

Естественный отбор более устойчивых форм организации, т. е. стабилизирующий отбор, приводит к усложнению регуляторных систем. *Регуляторные механизмы* ведут, как мы видели, к увеличению устойчивости, т. е. падению энтропии особей.

Те же регуляторные механизмы (уже простое доминирование нормы) допускают накопление обезвреженных мутаций в скрытом виде, и мы видели, что в этом случае количество обратной информации снижается и в популяции в целом. Дальнейшее развитие регуляторных механизмов, при котором подавляется неблагоприятное выражение малых мутаций даже в гомозиготе, приводит к дальнейшему снижению обратной информации. Это означает *увеличение организованности и устойчивости популяции* (т. е. падение энтропии) при данных условиях существования, при сохранении высокой степени ее эволюционной пластичности. Популяция в этом случае насыщается гетерозиготами, а также многими малыми мутациями, которые свободно комбинируются. Мобильность популяции максимальна, и каждая особь по своей наследственной информации глубоко индивидуальна, хотя фенотипически и не выходит за пределы нормы.

Так как все особи такой популяции гетерозиготны по большому количеству генов, то стабилизирующая форма естественного отбора приводит к стабилизации именно таких гетерозиготных особей. Большинство этих гетерозигот являются носителями плеiotропных генов, благоприятные выражения которых послужили материалом положительного отбора, и потому полностью проявляются, а неблагоприятные в гетерозиготе подавлены.

Стабилизирующая форма естественного отбора приводит, следовательно, через развитие регуляции к повышению устойчивости не только особей, но отчасти и популяции в целом. Можно тогда говорить не только о саморегуляции в развитии особи (индивидуальный гомеостазис), но и саморегуляции в строении популяции (популяционный или генетический гомеостазис) [12]. Появление гомозиготных форм означает поэтому не только проявление неблагоприятных выражений отдельных мутаций, но и нарушение всей генетической системы, которая приобрела свою устойчивость именно путем комбинирования многих гетерозигот.

Таким образом, как мобильность популяции, так и устойчивость ее особей поддерживаются стабилизирующей формой естественного отбора при значительной и стабильной гетерозиготности их генетической структуры. Не только отдельные особи, но и популяция (и вид в целом) приобретают некоторую организованность. Возникают даже регуляторные механизмы, поддерживающие устойчивость состава популяции при данных условиях существования. В эволюции форм организации популяций и видов в целом значительную роль играют, однако, межгрупповое соревнование и групповой отбор, которые в данной статье не рассматриваются.

ЛИТЕРАТУРА

- [1] Г о л д м а н С., Теория информации, М., ИЛ., 1957.
- [2] С у к а ч е в В. Н., Биогеоценология и фитоценология, ДАН СССР, т. 47, № 6, 1945, 447—449.
- [3] Ш м а л ь г а у з е н И. И., Проблема устойчивости органических форм в процессе эволюции, Журн. общ. биол., т. 6, № 1, 1945, 3—25.
- [4] Ш м а л ь г а у з е н И. И., Факторы эволюции, М.—Л., Изд. АН СССР, 1946.
- [5] Ш м а л ь г а у з е н И. И., Наследственная информация и ее преобразования, ДАН СССР, т. 120, № 1, 1958, 187—190.
- [6] Ш м а л ь г а у з е н И. И., Регулирующие механизмы эволюции, Зоол. журн., т. 37, вып. 9, 1958, 1291—1306.
- [7] Ш м а л ь г а у з е н И. И., Контроль и регуляция в эволюции, Бюлл. Моск. о-ва исп. прир., отд. биол., т. 63, вып. 5, 1958, 93—121.
- [8] Ш р е д и н г е р Э., Что такое жизнь? М., ИЛ, 1947.
- [9] B o y d e n A. A., Comparative evolution with special reference to primitive mechanisms, *Evolution*, v. 7, № 1, 1953, 21—30.
- [10] F i s h e r R. A., The genetical theory of natural selection, Oxford, 1930.
- [11] K a l m u s H., A cybernetical aspect of genetics, *Journ. Heredity*, v. 41, 1950.
- [12] L e r n e r I. M., Genetic homeostasis, Edinburg—London, 1954.
- [13] M a d i s o n K. M., The organism and its origin, *Evolution*, v.7, № 3, 1953, 211—227.
- [14] W a t s o n J. D. and C r i c k F. H. C., The structure of DNA, Cold Spring Harbor Symposia Quant, Biol., v. 18, 1953, 123—131.

Поступило в редакцию 11/VI 1958.

ТИПЫ УПРАВЛЯЮЩИХ БИОЛОГИЧЕСКИХ СИСТЕМ И ИХ ПРИСПОСОБИТЕЛЬНОЕ ЗНАЧЕНИЕ

А. А. МАЛИНОВСКИЙ

(ОДЕССА)

1. Введение

В биологических образованиях—клетках, органах, организмах, видах и т. д.—можно выделить различные типы управляющих систем, различающиеся теми взаимоотношениями, в которых находятся составляющие их звенья—органы, процессы, нервные или гормональные связи и т. д.

Можно установить, что определенный тип управляющих систем, как правило, выполняет также определенную приспособительную функцию.

Так, например, изучаем ли мы идущее при смене поколений эволюционное приспособление организма к среде, или его кратковременное приспособление путем комбинаций элементарных рефлекторных реакций, или, наконец, морфологические черты структуры определенного органа, но если во всех этих случаях задачей организма является приспособление к ненаправленно изменяющейся, постоянно новой среде, то и тип управляющих систем, обеспечивающих это приспособление, точно так же будет в основном один и тот же. Другой, тоже довольно однородный тип систем мы обнаружим, изучая формы связи, обеспечивающие приспособление организма к тем изменениям среды, которые происходят регулярно и в которых повторяются одни и те же, постоянно возникающие комбинации условий. И здесь мы также можем обнаружить общие черты, например, в особенностях сложной рефлекторной или гуморальной реакции и т. д. Одни типы управляющих систем обеспечивают тенденцию к подвижному равновесию, другие—к дифференциации органов, тканей или реакций и т. д. Задачей настоящей работы является указать на некоторые из таких типов и на их приспособительную функцию.

Ниже мы рассматриваем несколько типов биологических систем в связи с определенными условиями, в которых существуют живые организмы (например, постоянно и неожиданно меняющимися, меняющимися регулярно, устойчивыми), и с теми процессами, которые протекают в самих организмах (устойчивыми, циклическими, нарастающими и пр.).

Наша задача заключается в том, чтобы выяснить, какие системы более благоприятны для приспособления организма (или вида и т. д.) к тем или иным условиям или для выполнения необходимых для организма функций.

Попытки найти общие принципы строения систем, управляющих развитием живых организмов, делались в биологии уже давно. В ряде статей начиная с 1914 г., а затем в своей книге 1924 г. «Физиология типов» Н. А. Белов [1] развил мысль, что основным типом взаимоотношений в организме является то, что теперь обычно называется отрицательной обратной связью: если орган *A* стимулирует орган *B*, то орган *B* угнетает орган *A*.

Н. А. Белов приводил много примеров подобных соотношений, взятых из литературных данных, но прямых экспериментальных доказательств у него не было.

В тридцатых годах с дальнейшим развитием этой идеи выступил М. М. Завадовский, доказавший ее экспериментами. В 1941 г. им была издана специальная монография по этому вопросу [3].

В [12] было показано, что в процессах дифференциации, а также в регуляции некоторых физиологических процессов основную роль играют положительные обратные связи.

Однако только с развитием кибернетики вопрос о строении и типах управляющих систем в живой природе стал рассматриваться в биологии сколько-нибудь систематически. Надо отметить, что основным объектом изучения здесь оказалась нервная система и лишь в гораздо меньшей степени—другие управляющие системы. Между тем процессы управления широко распространены и у организмов, лишенных нервной системы (например, у растений), и в ряде явлений, хотя и подчиняющихся влиянию нервной системы, но имеющих одновременно и другие пути управления. Нам представляется даже, что при современном состоянии наших знаний изучение схемы управления в процессах раннего эмбрионального развития и в эндокринной системе, а также в простейших процессах, которыми управляют низшие отделы нервной системы, может дать благодаря большей ясности отношений не меньше, чем изучение схем управления у высших отделов.

Биологические системы, даже самые простые, в ряде отношений гораздо сложнее созданных в настоящее время искусственных управляющих систем. Это обстоятельство порождает два на первый взгляд противоположных следствия:

Во-первых, в биологических системах мы можем ожидать выявления таких форм и типов взаимоотношения частей, которые или не встречаются в искусственных системах, или играют в них сравнительно меньшую роль.

Во-вторых, в биологических системах мы пока вынуждены ограничиваться более грубым изучением управляющих систем, чем в искусственно созданных механизмах. Причина отчасти заключается в том, что строение искусственных механизмов определяется самим человеком, и оно известно во всех его деталях. В случаях, взятых из биологии, соотношения настолько сложны, что мы имеем возможность выделить лишь основные их черты. Это неизбежно ведет к выделению только общих и сравнительно простых форм, что, однако, уже в настоящее время позволяет сделать определенный шаг вперед в познании биологических явлений.

Настоящая работа посвящена изучению и систематизации биологических управляющих систем.

2. Приспособление к ненаправленным изменениям среды

При изучении изменений среды, воздействующей на организм, можно выделить два основных типа. В первом случае имеется повторяющаяся (регулярно или нерегулярно) смена одного определенного комплекса условий другим. Так, например, живя постоянно в одной местности, животное или растение испытывает каждый год одни и те же сезонные изменения, образующие тесно связанную группу условий (температура, условия питания, окраска окружающей среды и т. п.). Все эти условия возникают почти одновременно и одновременно же исчезают.

Во втором типе изменений среды условия меняются без повторений. Примером могут быть вековые изменения климата и рельефа, когда

животные и растения испытывают совершенно новые комбинации условий, которых они не испытывали никогда ранее. Еще резче новые комбинации условий организмам приходится испытывать при переселениях. Точно так же и в индивидуальной жизни животному приходится постоянно сталкиваться с новыми сочетаниями условий.

Как же идет приспособление к этим условиям?

Рассмотрим два случая приспособлений к ненаправленным изменениям среды: механизм ядерной наследственности, обеспечивающий эволюционное приспособление вида, и систему рефлекторных реакций нервной системы при приспособлении организма к новым ситуациям.

а) Механизм ядерной наследственности

Одним из наиболее общих случаев такого приспособления является механизм ядерной наследственности. Некоторые стороны его были уже ранее изучены автором [9], [10], [11], поэтому мы здесь остановимся на них возможно короче.

Основной интерес для нас представляют три особенности этого механизма. Первая: каждое отдельное видоизменение наследственных задатков (мутация) оказывает отчетливо выраженное влияние лишь на сравнительно небольшое число признаков организма. Вторая: мутации различных генов в подавляющем большинстве случаев проявляются независимо друг от друга. Исключения бывают, но не часто (см., например, [22]). Третья: линейное расположение генов и хромосом.

Следует подчеркнуть простоту перечисленных особенностей механизма ядерной наследственности. Ниже будет показана целесообразность такого строения этого механизма, явившегося конечным результатом длительной эволюции. Эволюция в сторону целесообразного упрощения структуры в филогенезе животных и растений часто встречается. Так, например, движение суставов конечностей высших животных является в основе движением шарнирных механизмов и с механической точки зрения много проще, чем возникшие гораздо раньше сложнее движения червя или амебы.

Перейдем теперь к анализу особенностей ядерной наследственности. Действие отдельной генной мутации может охватывать больше или меньше признаков (быть более или менее плеiotропным). Строго говоря, ни одна мутация, видимо, не ограничивается в своем проявлении одним признаком. Однако, как правило, влияние мутаций проявляется резко лишь на одном или на немногих признаках, а на остальных или совсем не сказывается или сказывается сравнительно слабо.

Чтобы понять биологическую целесообразность такого явления, следует прежде всего учесть, что подавляющее большинство мутационных признаков является вредным. Вопрос этот детально разобран различными авторами (Фишер [30], Малиновский [9]), и, не останавливаясь на нем подробно, мы отметим лишь, что всякий организм в результате длительной эволюции хорошо приспособлен к условиям своего существования, поэтому случайное изменение его свойств может скорее всего нарушить его приспособленность к среде.

Затем, если мутация вызвала изменения сразу нескольких признаков, то в подавляющем большинстве случаев приспособительное значение изменения одних признаков не будет зависеть от значения изменения других признаков [9]. Если мы учтем оба эти условия: 1) преобладание среди вновь возникающих признаков вредных и 2) случайный характер сочетания вредных и полезных признаков, вызываемых одной мутацией (когда она вызывает их несколько), то становится понятным, что для эволюции вида выгоднее, когда мутации затрагивают меньше признаков, так как возраста-

ние числа признаков при каждой мутации влечет за собой связывание полезных признаков с вредными, что приводит особь к гибели. Поэтому можно ожидать, что в процессе борьбы за существование имели преимущество те виды, у которых действие каждой мутации было связано с меньшим числом признаков (поскольку этому не препятствовали противоположные эволюционные тенденции).

Мы не будем останавливаться на анализе взаимодействия мутаций, который по сути сводится к тем же основным положениям, что и анализ плейотропии.

Несколько иначе выглядит вопрос о линейном расположении генов [11].

Линейное расположение было установлено первоначально благодаря наблюдению того факта, что некоторые мутации имеют тенденцию передаваться совместно.

Не излагая подробно вопроса, мы постараемся пояснить лишь основное, важное для целей настоящей статьи. Как известно из элементарного курса генетики, наследственные единицы—гены—расположены в хромосомах, обычно продолговатых образованиях, заключенных в ядре. При этом в каждой клетке тела у высших животных и растений каждый вид хромосом представлен двумя экземплярами, из которых один произошел от отцовского организма, другой—от материнского. Каждый ген в нормальном случае занимает всегда одно, строго определенное место в хромосоме. Таким образом, каждый ген (как и каждая хромосома) представлен всегда двумя экземплярами, из которых один иногда может быть нормальным (например, ген, связанный у дрозофилы с обычным красным цветом глаз), а другой—мутацией этого гена (например, мутацией, обуславливающей белый цвет глаз). Тогда эту структуру изображают как Aa (где a —мутация).

Если какой-то другой ген (например, окраски или формы тела и т. д.) расположен в другой хромосоме, то он передается независимо от первого и вступает в свободные сочетания с ним. Иначе дело обстоит, когда оба гена находятся в одной хромосоме. Так, например, мутантные гены коричневой окраски глаза (*seria*) и усиленного развития щетинок на теле (*hairy*) расположены в одной и той же хромосоме. В этом случае мы не наблюдаем у потомков свободных сочетаний мутаций между собой и с нормальными соответствующими им генами. У внуков той особи, которая несла обе эти мутации, они опять окажутся в 99,5% случаев у одной и той же особи и только в 0,5%—у разных. Очевидно, что определенный участок хромосомы обычно передается как целое, и вместе с ним передаются эти почти неразлучные мутации. Однако в 0,5% случаев они разлучаются и впоследствии их трудно объединить. Чем это объясняется? Оказывается, что когда хромосомы обоих родителей попали в одно яйцо и из яйца развился организм, то в дальнейшем, в определенный момент (перед образованием половых клеток), парные хромосомы ложатся рядом и обмениваются соответствующими кусками. Такой обмен может происходить в разных местах хромосомы, но всегда так, что отдельный кусок одной хромосомы заменяется строго соответствующим ему куском другой. После обмена в каждой хромосоме опять остается по одному гену из каждой пары, при этом, однако, гены могут быть уже другие. Например, хромосома, которая несла ген коричневых глаз, может отдать его другой хромосоме и взамен получить парный ему ген обычных для дрозофилы красных глаз. Естественно, что гены, расположенные близко друг от друга, чаще всего остаются или передаются в другую хромосому совместно. Так бывает и с генами, дающими мутации коричневых глаз и усиленного развития щетинок. Но если разрыв хромосом все же происходит между ними (в 0,5% случаев), то одна мутация попадает с обменивающимся куском в другую хромосому, а другая мутация остается в прежней. Понятно, что чем дальше расположены гены друг от друга, тем чаще происходит между ними разрыв и тем более свободно они комбинируются. Частота разрывов и позволила определить взаимное расположение генов и даже составить целые карты хромосом, показывающие расположение генов по одной линии. Впоследствии эти карты были подтверждены микроскопическими наблюдениями разрыва хромосом.

Всякое сцепление генов чрезвычайно невыгодно с эволюционной точки зрения. В самом деле, представим себе, например, что у какого-то вида животных возникли два благоприятных мутационных признака. Что было бы, если бы гены, обуславливающие эти признаки, оказались расположенными очень близко друг к другу в одной хромосоме, причем мутации возникли в разных экземплярах гомологичных между собой хромосом, находящихся у разных особей? Такие признаки лишь с большим трудом могли бы

совместно попасть в обе хромосомы к одному животному и закрепиться у него.

Вместо того чтобы путем комбинирования при скрещивании приобрести оба признака, животные, обладающие одним преимуществом, почти не могли бы получить другого и боролись бы за существование с теми, которые его имеют. Таким образом, распространение одного признака не вело бы к суммированию его преимуществ с преимуществами другого, а только мешало бы распространению этого другого признака. Но новые полезные признаки обычно возникают порознь у разных особей, причем в обширных видах их может возникать одновременно довольно много. Понятно, что полное или почти полное сцепление генов чрезвычайно мешало бы эволюционному развитию вида.

Однако и совершенно отдельное наследование гена (распадение хромосом на отдельные гены), давая широкие возможности комбинирования, тоже имело бы свои отрицательные стороны. Во-первых, каждая хромосома имеет известный аппарат, обуславливающий правильное распределение хромосом при делении клеток: по одной хромосоме каждого типа в каждую дочернюю клетку (кинетическое тельце и т. д.). Если бы каждый из десятков тысяч генов наследовался отдельно от других, то аппарат клеточного ядра чрезвычайно усложнился бы, что, конечно, могло неблагоприятно отразиться на клетках, а следовательно, и на всем организме в целом. Во-вторых, при образовании половых клеток иногда наблюдаются случаи, когда правильное расхождение хромосом нарушается и в одну половую клетку попадают сразу две хромосомы одного типа, а в другую—ни одной. Известно, однако, что нехватка даже одного гена обычно очень неблагоприятно отзывается на организме. В несколько меньшей степени это относится и к присутствию излишнего гена. Потомство, образовавшееся из таких половых клеток, или сразу гибнет, или оказывается малоприспособленным и в свою очередь дает нарушения при следующем поколении.

Таким образом, если бы каждый из генов наследовался как независимая хромосома, их нерасхождение в процессе мейоза имело бы весьма тяжелые последствия для организма и его потомства. Но если нерасхождение наблюдается в определенном проценте случаев даже тогда, когда в ядре имеется немного хромосом, то ясно, насколько чаще оно встречалось бы, если бы таких генов—хромосом были десятки тысяч. Очевидно, что и с этой стороны полная независимость наследования генов была бы крайне невыгодна для вида.

С этой точки зрения линейное расположение генов в хромосоме представляет, по-видимому, наиболее выгодную форму. Вся масса генов разбита на небольшое число групп—хромосом—со всеми преимуществами, которые обеспечиваются таким ограниченным количеством (упрощение аппарата деления, редкость нерасхождения). В то же время связь генов в хромосоме сведена к минимуму. Каждый ген связан лишь непосредственно с двумя соседними. Вероятность разрывов здесь растет по мере увеличения расстояния. Уже при сравнительно небольших расстояниях между генами сцепление между ними невелико. Практически оно не мешает комбинированию отдельных полезных мутаций. Любая другая форма расположения требовала бы или большего числа связей гена с соседними, или вела бы к уменьшению расстояний, т. е. опять к более тесному сцеплению генов *). Поэтому отдельные отклонения от обычного расположения генов (не нарушающие, однако, линейного их характера) встречаются сравнительно редко.

Подводя итоги сказанному о рассмотренных сторонах механизма наследственности, можем сделать следующие выводы:

*) Это показано для искусственно созданных кольцевых хромосом.

1. Поскольку при изменении условий отдельные характеристики среды могут сочетаться в самых разнообразных комбинациях, постольку в эволюции наиболее совершенным аппаратом приспособления является аппарат, дающий возможность гибкого и свободного создания ответной комбинации признаков организма, соответствующих этим условиям.

2. Рассмотренные особенности ядерной наследственности, создавая взаимную независимость мутационных признаков, обеспечивают возможность их свободного комбинирования.

б) Система элементарных рефлекторных реакций

В процессе эволюции основной движущей силой, создающей новые сочетания признаков, является естественный отбор. Однако и в индивидуальной жизни отдельного животного, где приспособление идет другими путями, но где тоже приходится приспосабливаться к новым, неповторяющимся условиям, основные черты управляющей системы остаются теми же. Они должны обеспечить максимальную подвижность и взаимную независимость отдельных ее звеньев. Здесь снова проявляется корпускулярность.

В этом отношении наиболее ярким примером является расчлененная рефлекторная реакция нервной системы на те или иные новые ситуации, возникающие перед организмом.

Исследуя организм с богатой системой безусловных рефлексов, частью дополненных усовершенствующими их условнорефлекторными реакциями, мы можем отметить, что наряду с какой-то основной линией поведения его существование и поведение обеспечивается огромным количеством отдельных, практически почти независимых друг от друга реакций. На одном и том же фоне может одновременно протекать множество различных реакций, причем многие из них не связаны между собой. Объединяет их лишь одно общее свойство: все воздействия на организм имеют отношение к его существованию, и все реакции должны способствовать сохранению этого существования.

В животном организме общая структура нервных реакций, с их большой взаимной независимостью и способностью к одновременному разворачиванию целого ряда отдельных цепей рефлексов в различных направлениях, в значительной степени обусловлена этой потребностью.

Таким образом, мы и здесь (как и в генетических механизмах) имеем систему реакций в значительной степени «корпускулярного» типа. Роль независимых «корпускул» в данном случае играют отдельные реакции или даже их комплексы. Управление этой системы (если не вмешиваются более сложные интегрирующие механизмы нервной системы) заключается в целесообразном реагировании каждой отдельной рефлекторной цепи, с известной независимостью цепей друг от друга. Таким образом, наиболее простое комбинирование этих реакций основано, с одной стороны, на целесообразном характере каждой из них в соответствии с вызывающим ее условием *), а с другой стороны — на свободном сочетании таких реакций. В результате любая новая комбинация условий вызывает ответную комбинацию реакций, обеспечивающую приспособление организма к сложной среде.

Однако дело этим не ограничивается. И в объединении реакций в единое целое, протекающем на более высоком уровне поведения, расчлененность отдельных реакций дает возможность высшим центрам производить отбор и комбинирование их в соответствии с более сложными задачами организма. Сюда относятся все формы высшей нервной деятельности,

*) Конечно, целесообразность выработана ранее в процессе естественного отбора.

начиная от простейших условнорефлекторных комбинаций и кончая самыми сложными процессами поведения.

В подтверждение этого можно привести онтогенетическую эволюцию рефлекторных реакций у амфибий. Раннюю нерасчлененную форму поведения личинок амблистом наблюдал Дж. Э. Когхилл [5]. Этот автор установил, что на ранней стадии развития у переходящих к самостоятельной жизни личинок амфибий нельзя наблюдать отдельных элементарных рефлексов. Первичной, по Когхиллу, является целостная реакция, вначале сводящаяся к простому изгибу тела, а позже усложняющаяся и превращающаяся в сложные плавательные движения. Здесь звенья реакции связаны между собой, и нельзя выделить отдельные элементарные реакции. На более высоких ступенях развития амблистомы происходит уже видимое расчленение этих реакций на отдельные рефлексы. Когда задачи, стоящие перед животными, становятся сложнее, тогда выявляется неспособность первичного, нерасчлененного поведения к прямой перестройке в соответствии с условиями. Только отдельные дробные реакции оказываются подходящим материалом для построения более высоких форм нервной деятельности. В результате эволюционного процесса примитивные недифференцированные формы поведения животного заменяются более гибкими, дискретными формами. Еще большая гибкость достигается при развитии условнорефлекторных аппаратов, где любой исполнительный механизм может сочетаться с любым индифферентным раздражителем или даже с той или иной комбинацией раздражителей*).

Таким образом, *основной формой приспособления к неопределенной, различным образом изменяющейся среде является дискретность, раздробленность и способность к созданию свободных комбинаций тех единиц, за счет которых осуществляется приспособление к среде*. Эта тенденция всегда существует в живых организмах и прогрессирует в процессе естественного отбора. Но развитию этой тенденции постоянно противостоят два фактора: 1) ограниченные возможности изменения живых структур, обусловленные их физико-химическим строением, и 2) противоположная тенденция к созданию определенных полезных связей, которые обеспечивают организму в каждый данный момент наиболее полное и экономное приспособление к той сравнительно устойчивой среде, в которой он в настоящее время находится (включая и внутреннюю среду самого организма).

Так, например, известно, что при развитии глаза у амфибий выделившийся из зачатка нервной системы зачаток будущей сетчатки глаза («глазной бокал») так влияет на расположенные рядом клетки наружного слоя зародыша, что последние образуют утолщение, превращающееся в дальнейшем в хрусталик. При пересадке глазного бокала в другое место зародыша он и там может вызвать («индуцировать») образование хрусталика.

*) Дж. Э. Когхилл находит, что и в дальнейшем развитии поведения действуют те же закономерности: хотя поведение животного дифференцируется, но распадение на отдельные реакции и рефлексы является лишь видимым; даже в местной реакции участвует вся нервная система в целом. Остальные части реакции не проявляются и недоступны нашему прямому наблюдению. Отмечая аналогию описанной картины с развитием поведения высших животных до человека включительно, Дж. Э. Когхилл делает вывод, что ошибаются те психологи (и физиологи), которые исходят в своих представлениях о поведении из наличия отдельной реакции или рефлекса. Как мы говорили выше, выделение расчлененных реакций действительно может иметь вторичный характер, и можно допустить, что отдельные рефлекторные реакции являются уже вторичным результатом торможения более обширной реакции организма. Однако с точки зрения поведения нам важно, что при столкновении со средой животное способно действовать именно путем вычленения таких практически отдельных реакций. В этом отношении представление о рефлексе, как элементарной и универсальной единице поведения, сохраняет полную силу, и данные Когхилла, вопреки его мнению, только подтверждают представления И. П. Павлова.

В свою очередь получившаяся система (бокал — хрусталик) влияет на развитие других, позднее развивающихся частей глаза, и постепенно создается целая цепь зависимостей $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$ и т. д.

Нетрудно видеть, что такая цепь связей в значительной степени ограничивает возможности эволюции организма. Мутация, даже дающая полезные изменения ранних звеньев— A или B ,—имеет много шансов как-то отразиться и на дальнейших звеньях C , D , E или F , а такие дополнительные изменения, как правило, окажутся скорее вредными, чем полезными. Но этим будет обесценена и польза, приносимая мутацией в первом звене. В результате полезными могут быть чаще всего только те мутации, которые затрагивают последнее звено. И, действительно, именно эти звенья и могут легче всего эволюционировать, изменяясь или надставляясь новыми звеньями. Так, в значительной степени и представлял себе ход эволюции академик А. Н. Северцов [20]. С этой точки зрения легко объяснить огромный консерватизм ранних эмбриональных стадий развития, которые могут быть очень сходны у видов, уже давно разошедшихся в процессе эволюции: последняя шла у них в основном за счет изменений и надставок поздних стадий, а мутации, затрагивавшие ранние стадии, не были благоприятны и не закреплялись. Поэтому ранние стадии сохранились до настоящего времени со сравнительно малыми изменениями по сравнению с той эпохой, когда разошедшиеся виды были еще близки друг другу (см. И. И. Шмальгаузен [27] и др.).

3. Приспособление к постоянным условиям или правильной смене условий

Что же вынуждает организм к созданию связей, упомянутых нами в примере с индукцией хрусталика глазным бокалом? В данном случае, очевидно, помимо ряда других причин, для организма весьма ценно то, что такой зависимостью автоматически создается соответствие в расположении сетчатки и хрусталика: хрусталик развивается прямо перед сетчаткой, как это необходимо для их совместного функционирования. Хотя такая связь, вероятно, в известной мере ограничивает возможность эволюции, однако она является столь ценной для каждого организма, что приносимая ею польза более важна, чем некоторое снижение эволюционной гибкости вида.

Таким образом, эволюционные интересы вида в известной степени приходят в противоречие с ценностью приспособительных связей для каждого индивида в каждый данный момент, и естественный отбор ведет к созданию некоторого среднего оптимального положения, где интересы вида в целом (т. е. потенциальная эволюционная гибкость) и интересы индивида (т. е. механизмы, обеспечивающие координацию частей и приспособленность особи в любой данный момент) взаимно ограничивают и уравнивают друг друга.

Как же идет эта эволюция связей и каковы оптимальные их формы? Рассмотрим конкретный пример подобной эволюции.

На Больших Зондских островах на плоских берегах, то далеко заливаемых приливом, то обнажающихся при отливах, оказалась чрезвычайно благоприятная обстановка для постепенного выхода водных организмов на сушу, и целый ряд морских животных в различной степени приспособился к новым, полусухопутным условиям.

В числе других наиболее приспособились к сухопутным условиям два родственных между собою рода рыб из семейства Gobiidae: *Boleophthalmus* и *Periophthalmus*. Эти рыбы приобрели целую группу признаков, способствующих сухопутной жизни: своеобразные роговые «очки», прикрывающие глаза, дыхательные мешки, содиненные с носовой и жаберной полостью, и перепонки, предохраняющие жабры. При этом глаза в значительной степени переместились на верхнюю поверхность головы, так как на земле именно сверху надо ожидать и мелкой добычи в виде насекомых и нападения

возможных врагов. Плавники удлинились и дали возможность рыбе передвигаться, отталкиваясь от земли. Изменились и внешние покровы и ряд других признаков. В личиночном состоянии рыбки эти живут в воде, обладают обычным для рыб расположением глаз и рядом других аналогичных признаков, а затем происходит превращение, изменяющее их внешний облик и делающее их более сухопутными, чем водными животными. В опытах [31] *Periophthalmus* могли жить на суше во влажной атмосфере несколько суток, а погруженные в воду погибали уже нередко через 30 минут. Оказалось, что выход на сушу можно ускорить путем стимуляции рыбок гормоном щитовидной железы. При этом и у *Periophthalmus* и у *Voleophthalmus* происходило известное изменение покровов. Однако другие признаки реагировали на гормон щитовидной железы уже неодинаково: в то время как у *Periophthalmus* они в основном также изменялись в сторону приспособления к сухопутной жизни, у *Voleophthalmus* значительная часть из них под влиянием гормона щитовидной железы не прогрессировала в этом отношении.

Из изложенного ясно, что большинство признаков, связанных с жизнью на суше, принципиально может развиваться, не испытывая необходимости в усиленном снабжении гормоном щитовидной железы. Однако такая самостоятельность развития этих признаков имеет свои недостатки: не связанные единым сигналом, эти признаки в отдельных случаях могут развиваться в разноразличную; например, в то время как изменение покровов и поведения уже делает желательным переход на сушу, развитие глаз и плавников может быть еще не таково, чтобы обеспечить этот переход. Может быть и обратное. Главное, такой разноразличностью, не приспособив животных достаточно к новой среде, может сделать их маложизнеспособными и в прежней. В этом случае становится чрезвычайно важным, чтобы момент превращения управлялся для всех этих признаков одним сигналом. Таким сигналом становится у *Periophthalmus* усиление функций щитовидной железы. Совершенно аналогичную картину эволюции (конечно, давно уже завершенную) мы имеем и у амфибий *).

В случае *Periophthalmus* мы можем определить эволюцию связей, как объединение признаков, имеющих сходное приспособительное значение (содействие выходу на сушу), вокруг одного сигнала—усиления продукции гормона щитовидной железы.

Приведем еще ряд примеров подчинения однородных приспособительных признаков одному сигналу.

Гормоны целого ряда желез внутренней секреции, несомненно, имеют значение сигнала, управляющего изменениями, играющими в организме сходную роль в приспособлении. Так, адреналин, имеющий действие, сходное с действием симпатической нервной системы (т. е. являющийся симпатомиметическим веществом), в надлежащей концентрации мобилизует к максимальному действию (например, для борьбы или бегства) большинство функций организма (см. [6], [15], [33], [19]). Он усиливает деятельность сердца, работоспособность мышц, остроту восприятия некоторых органов чувств и тонус центральной нервной системы, повышает содержание сахара в крови, перераспределяет кровь и, наконец, даже повышает фагоцитарную активность лейкоцитов.

Точно так же половой гормон одновременно стимулирует развитие целого ряда вторичных половых признаков и специфического поведения, обеспечивая таким образом координированное появление всех этих особенностей, необходимых животному для выполнения его роли в размножении. Ростовый гормон гипофиза, действуя на рост различных органов с определенной неравномерностью, способствует тому изменению пропорций, которое необходимо, например, у человека при переходе от новорожденного с его слабо развитыми конечностями и лицевой частью черепа к взрослому организму с длинными сильными конечностями и развитым жевательным аппаратом.

Для более полного представления о координирующем значении гормональных систем полезно сопоставить их с системами признаков, завися-

*) Интересно, что это далеко не единственный случай, когда органы, которые вполне могут развиваться при разных условиях, начинают развиваться лишь при одном определенном условии, как бы подчиняясь его сигнализации. В виде примера можно привести рога оленей, которые у одних видов развиваются у всех особей, а у других—только у самцов, при наличии мужского полового гормона.

щими от витаминов. Здесь нетрудно увидеть глубокие различия гормонов от витаминов, с которыми их так часто сравнивают. При авитаминозах выявляется влияние витаминов на функции, имеющие между собой мало общего. Так, витамин А влияет на сумеречное зрение, на роговицу глаза, на сопротивляемость организма инфекции и на общий рост тела [7]. Авитаминоз D сказывается на росте костей, на функции кишечника и на высшей нервной деятельности у человека, вызывая иногда так называемую рахитофрению [13]. Одним словом, гормоны, иногда и не будучи необходимыми для развития определенных признаков, приобретают влияние на них, управляя моментом их появления, если эти признаки входят в единый приспособительный комплекс, связанный с данной железой. Напротив витамины, по-видимому, отзываются только на тех признаках, которые никак не могут развиваться без них, но которые хотя и зависят от одного витамина, однако имеют приспособительное значение в совершенно иной плоскости. В целом, в то время как витамины являются лишь необходимым материалом или условием для ряда важных реакций, гормоны являются организующими центрами, связывающими в ряде случаев однородные приспособительные функции.

Указанное различие связано и с другим. В большинстве случаев гормоны при избытке их сразу вызывают в организме целый ряд нарушений, противоположных нарушениям при недостатке этих гормонов. Напротив, избыток витаминов в большинстве случаев долго не сказывается на нормальном организме, и только огромные дозы их могут вызвать расстройства физиологических функций.

Это—следствие различной эволюции отношения тканей к витаминам и гормонам. Гормоны вырабатываются самим организмом, и их количество легко может быть сокращено. Поэтому избыток их не угрожает организму, и последний не должен быть приспособлен к компенсации его. С другой стороны, изменившиеся условия могут сделать даже желательным изменение состояния организма путем усиления продукции гормона. Чтобы гормоны могли быть орудием регуляции функций организма, они должны быть активными как при понижении их содержания в организме, так и при повышении.

Напротив, витамины такой организующей роли не играют. Кроме того, они вводятся извне с пищей, и если их содержание в пище велико, то всегда имеется опасность, что они будут введены в организм в избытке. Если бы такой избыток мог нарушать функции организма, мы постоянно встречались бы с заболеваниями организма на почве даже небольшого гипервитаминоза. Поэтому организму приходилось приспособливаться, противодействуя этим нарушениям либо путем выведения избытка витамина, либо путем его разрушения, либо путем откладывания его в качестве запасного питательного вещества, либо, наконец, просто путем ослабления реакции на избыток витамина.

Специфика действия гормонов, в отличие от локально действующей нервной системы и от эмбриональных факторов типа «индуктора», состоит в том, что гормоны, одинаково проникая во все части организма, не могут порождать различия в прежде одинаковых органах и тканях. Роль гормонов сводится к тому, чтобы на основе имеющихся уже особенностей вызвать в организме целый ряд координированных изменений. Так вызывается половое созревание у высших животных, так вызывается одновременное превращение различных органов личинки при перестройке ее во взрослое животное, так происходит одновременная мобилизация всех функций организма во время опасности и т. д. При этом гормоны являются прекрасным примером выработки оптимальной формы управления. Связь органов и функций, как мы говорили, должна быть, с одной стороны, прочной,

а, с другой стороны, изменение в одном звене должно возможно меньше затрагивать другие звенья, так как иначе эволюция последних была бы задержана.

Цепная связь типа $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$ и т. д., как мы видели, почти исключает возможность эволюции первых звеньев, так как она обычно неблагоприятно сказывается на последующих звеньях.

В гормональных системах осуществляется, как правило, другая, более совершенная форма связи—форма, которую можно назвать звездно-лучевой (рис. 1). В этой форме центральное звено—гормон—(А) одновременно связано с целой группой признаков (В, С, D, E, F, G и т. д.), зависящих от него, но не зависящих непосредственно друг от друга. При этом достигается весьма тесная связь всех «лучей» этой звезды между собой: все они связаны друг с другом только через одно звено. В то же время, являясь конечным звеном, от которого не зависит остальная часть цепи, каждый «луч» может эволюционировать и даже исчезать, почти не отразившись на остальных, сильно коррелированных с ним признаках. Напротив, сам гормон может эволюционировать только с большим трудом, так как его изменения, даже полезные сами по себе, могли бы неблагоприятно отразиться на целом ряде зависящих от него признаков организма. И действительно, мы наблюдаем, в отличие от других признаков организма, огромную эволюционную устойчивость гормонов. Это сказывается, например, в том, что гормоны млекопитающих, как правило, оказывают на рыб, амфибий или птиц то же действие, как и соответствующие гормоны их собственных желез.

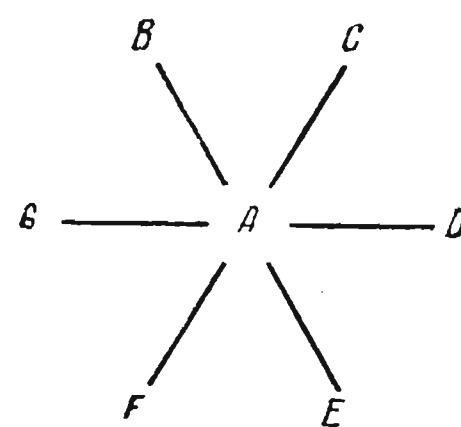


Рис. 1.

Особенно часто возникает координация в управлении многими признаками при повторяющемся переходе организмов из одних условий существования в другие. Постоянно повторяющиеся комбинации условий, которые ведут к возникновению прочных связей, аналогичных рассмотренным, могут встречаться и однократно и многократно при жизни одной особи и только один раз за несколько поколений. Так, перестройка организма и мобилизация его функций при приспособлении к опасности может происходить несколько раз в день. Приспособление к определенным сезонным изменениям происходит раз в год. Переход у амфибий и сухопутных рыб от жизни в водной среде к жизни в воздушной среде и связанный с ним метаморфоз происходит раз в жизни животного, так же, как и переход от бесполого существования растущего организма к созревшей в половом отношении взрослой форме. Здесь бросается в глаза большая аналогия превращений у выходящих на сушу амфибий и у ряда насекомых, у которых происходит метаморфоз, превращающий личинку во взрослую форму. Здесь также происходит многообразное, но повторяющееся из поколения в поколение изменение условий жизни, а часто и среды, с которой они связаны, например, водное или наземное существование личинок может сменяться переходом к условиям жизни взрослой летающей формы. При этом оказывается, что у насекомых, развивавшихся независимо от позвоночных, выработались другие, но принципиально сходные гормональные системы, управляющие явлениями метаморфоза и другими функциями организма. Это указывает на общие закономерности, приводящие именно к такой форме корреляционных связей.

Наконец, в ряде случаев мы наблюдаем чередование поколений, размножающихся партеногенетическим путем и поколений с половым размножением (например, у тлей [24]). В других случаях, в частности у растений, имеется несколько возможных форм развития в зависимости от того, попало ли растение на сушу или в воду (например, *Ranunculus multifidus*

и пр.), попало ли оно в условия высокогорья или в долину и т. д. Здесь мы видим те же устойчивые формы перестройки организма, происходящие уже не у определенной особи, а раз в несколько поколений, но позволяющие также предполагать достаточно прочную связь признаков между собою, осуществляющуюся, по-видимому, путями, сходными с разобранными выше.

Все описанные виды корреляций создаются путем установления физиологических связей между признаками, ранее не связанными между собой непосредственно и обладавшими той или иной степенью независимости. Но, как мы указывали выше, и независимость отдельных признаков далеко не всегда возникает первично. Она часто является продуктом эволюции и раздробления тех связей между признаками, которые возникли вне приспособления, как наиболее простое и вероятное следствие прежних процессов развития и жизнедеятельности. В других случаях эти связи когда-то имели приспособительное значение, но впоследствии, при перемене условий, утратили его и стали уже лишь тормозом в дальнейшей эволюции и приспособлении организма к новой обстановке.

Как в индивидуальном развитии, так и в эволюции мы наблюдаем возрастающую дифференциацию, разрыв прежних связей и выделение новых (или старых) самостоятельных единиц, способных к более свободному комбинированию. В индивидуальном развитии мы видели отчетливый пример этого на дифференциации рефлекторных реакций из первоначально слитных форм реакций [5]. То же мы видим в дифференциации тканей и их функций, эволюционное развитие которых было изучено А. А. Заварзиным [4]. Дальнейшая эволюция может создать у многих из них способность более или менее независимо от других признаков приспособительно изменяться при переходе в новую среду или в другие условия существования. Возможно, что подобный случай мы видели на примере *Boleophthalmus*, где переход к полусухопутному образу жизни связан с изменением целой группы признаков (так же, как и у *Periophthalmus*), но прямой причины, непосредственно координирующей их изменение, обнаружено не было. В дальнейшем, однако, параллельное изменение ряда признаков легко подчиняется одному координирующему центру (в случае *Periophthalmus* — щитовидной железе), который максимально обуславливает одновременность изменений и их надежность.

Таким образом, полная схема эволюции в этих случаях должна выглядеть так:

а) Вынужденная связь комплекса признаков (являющаяся побочным следствием сравнительно примитивных процессов развития или оставшаяся от предыдущих этапов эволюции), в значительной степени неадекватная требованиям приспособления.

б) Расчленение в эволюции всего комплекса на отдельные независимые признаки.

в) Создание под влиянием отбора приспособительных изменений целого комплекса признаков, не связанных еще непосредственной физиологической связью («историческая» или отдаленная физиологическая корреляция через самостоятельную реакцию каждого признака).

г) Закрепление ее путем создания прямых физиологических связей (физиологическая корреляция).

Подобный ход эволюции характерен для признаков, приспособляющих организм к устойчиво существующим или постоянно повторяющимся условиям. При этом легко возникает цепная форма связи, отражающая последовательность возникновения органов или признаков во времени. Однако эта форма, создавая систему, необходимую для одних определенных условий, резко ограничивает эволюционную гибкость данной системы

в дальнейшем. Более совершенной формой является «звездно-лучевая», обеспечивающая и прочность связей признаков у отдельного организма, и возможность дальнейшей эволюции системы.

4. Приспособительное значение некоторых типов систем с обратными связями

а) Процессы стабилизации функций

Как мы указали во введении, Н. А. Беловым и М. М. Завадовским было выяснено, что очень многие механизмы регуляции в организме обеспечиваются благодаря особым системам с обратной связью, которые М. М. Завадовский условно назвал типом «плюс-минус взаимодействия» (отрицательные обратные связи). В этих системах первый орган (или функция) стимулирует второй, а второй угнетает первый. Такие системы М. М. Завадовский изображает схемой $A \overset{+}{\underset{-}{\rightleftharpoons}} B$, означающей, что A стимулирует B , а B угнетает A . В качестве типичного примера такой связи приводится, например, взаимная зависимость половой железы и гипофиза. Гипофиз (мозговой придаток) стимулирует рост и функцию половой железы, а половая железа—угнетает функцию мозгового придатка. В результате всякий толчок, усиливающий работу половой железы, угнетает действие гипофиза, а половая железа при меньшей стимуляции со стороны гипофиза вновь снижает свою функцию, возвращаясь к прежнему состоянию. М. М. Завадовский экспериментально установил, что у петухов гребень, развитие которого стимулируется половой железой, угнетает функцию половой железы, и если срезать гребень, то размеры половой железы увеличиваются. Н. А. Белов и М. М. Завадовский показали, что аналогичными механизмами «плюс-минус взаимодействия» можно объяснить и регуляцию сахара в крови и явления компенсации, когда, например, при утрате одной почки другая частично берет на себя работу, ранее выполнявшуюся потерянному органом. Формально этот же механизм можно приписать самым различным приспособлениям к внешним условиям, которые уравнивают их вредные влияния. Так, избыток солнечных лучей стимулирует развитие пигмента в коже, который препятствует проникновению лучей в глубокие слои наружных покровов. Аналогично можно рассматривать взаимодействие патогенного микроорганизма и антител, образующихся под его влиянием и угнетающих его деятельность и т. д. Обширные материалы по взаимодействию органов и функций внутри организма дают достаточно оснований, чтобы, обобщив их, прийти к заключению о широком распространении и большом значении указанного типа связей в жизни организма, в частности в регуляции различных функций и сохранении их на определенном уровне. Таким образом, наличие обратных отрицательных связей способствует сохранению в организме устойчивого режима.

б) Процессы прогрессирующего нарастания и дифференциации

Для процессов развития организма характерны иные типы связей. В [12] было указано, что здесь основную роль играют положительные обратные связи. Так, усиленное развитие матки во время беременности, стимулированное влияниями, идущими со стороны плода, отнюдь не стремится к какому-либо равновесию, а идет нарастая, пока акт родов не нарушит эту связь. И действительно, матка в свою очередь стимулирует рост

плода, и мы имеем здесь взаимно усиливающееся взаимодействие, т. е. положительную обратную связь или, выражаясь аналогично термину

М. М. Завадовского, связь по типу «плюс-плюс» $A \xrightleftharpoons{+} B$. Другой пример

представляет собой так называемая ассимилятивная индукция, наблюдающаяся у некоторых тканей на ранних стадиях эмбрионального развития [2]. Представим себе, что в окружение некоторой ткани посажен небольшой участок другой, родственной ей и происходящей из того же источника. В этом случае форма, представленная более обширным участком, по-видимому, какими-то веществами, выделяемыми ею, может изменить развитие подсаженного к ней «меньшинства» клеток (выделяющих меньше специфических для них веществ) и повернуть это развитие в свою сторону, так сказать, «ассимилировать» их. Иначе говоря, клетки определенной структуры выделяют соответственные вещества, а последние в свою очередь способствуют образованию клеток такой же структуры из других клеток. Аналогичное явление, хотя и не доходящее до полного завершения, наблюдается при особых условиях, когда у коровы имеются в матке два теленка-близнеца разного пола, если между ними устанавливается частичная связь через кровообращение. В этом случае у самца мужская железа сохраняется нормальной, а у самки женская частично передифференцируется в мужскую. Здесь клетки железы мужского типа выделяют гормон, который стимулирует развитие таких же структур в клетках, их не имеющих (женской половой железы). Еще более яркий случай обнаружен В. Данчаковой [29], которая, вводя в куриные яйца фолликулин (женский половой гормон), добилась превращения петушков в кур (женская половая

железа $\xrightleftharpoons{+}$ фолликулин). Это явление чрезвычайно интересно и, видимо, имеет глубокие эволюционные корни.

Витчи (цитир. по [28]), спивая попарно самца и самку тритона боками, обнаружил, что их половые железы взаимно угнетают друг друга ($A \xrightleftharpoons{-} B$), но дело почти всегда кончается «победой» мужской половой железы — ее восстановлением и дальнейшим угнетением женской половой железы. Здесь, очевидно, имеется механизм, предохраняющий от возможности гермафродитизма известные виды, где определение пола частично зависит от внешних условий (как это обнаружено, например, у некоторых лягушек); взаимное угнетение двух желез ведет вследствие случайных отклонений к нарушению их равновесия. В этом случае менее угнетаемая железа, продолжая подавлять противоположную форму, сама в достаточной степени освобождается от давления и восстанавливается, уже совершенно без конкуренции противоположного пола. Этот тип ($A \xrightleftharpoons{-} B$) создает неустойчивое равновесие, нарушение которого, раз начавшись, идет до конца.

Следует заметить, что взаимодействие и типа $A \xrightleftharpoons{+} B$ и типа $A \xrightleftharpoons{-} B$ можно с равным правом назвать положительной обратной связью. Заменяя, например, B на $(-B)$, мы во втором случае также получим соотношение типа положительной стимуляции: $A \xrightleftharpoons{+} (-B)$. В обоих случаях изменения звена A будут иметь сходный характер. Если принять простейшие предпосылки, то с определенного момента A будет или только нарастать, насколько позволят условия организма, в которых находится A (это

будет после вступления системы A и B в зону $+A$, $+B$), или только убывать (в зоне $-A$, $-B$, рис. 2).

Однако в биологических системах взаимодействия типов $A \xrightleftharpoons{+} B$ и $A \xrightleftharpoons{-} B$ все же иногда удобнее и правильнее рассматривать как различные.

Действительно, если A и B — определенные органы, то их взаимная стимуляция обычно связана с развитием организма и при нормальных условиях нарастание обоих звеньев может быть обязательным, всегда однотипным этапом этого развития. Иначе обстоит дело с взаимным угнетением двух органов. На примере взаимного угнетения мужской и женской половой железы мы видим, что для организма одинаково возможны оба состояния: и преобладание мужского начала, и преобладание женского. Иначе говоря, в отличие от типичных «плюс-плюс» систем, в этих случаях, во-первых, допустимо развитие системы (A , B) в любом из двух направлений, и, во-вторых, важно не абсолютное увеличение развития обоих звеньев, а их взаимоотношение в системе. Аналогичным образом можно объяснить явление возникновения и устойчивости так называемых градиентных систем (см. [26], [2]).

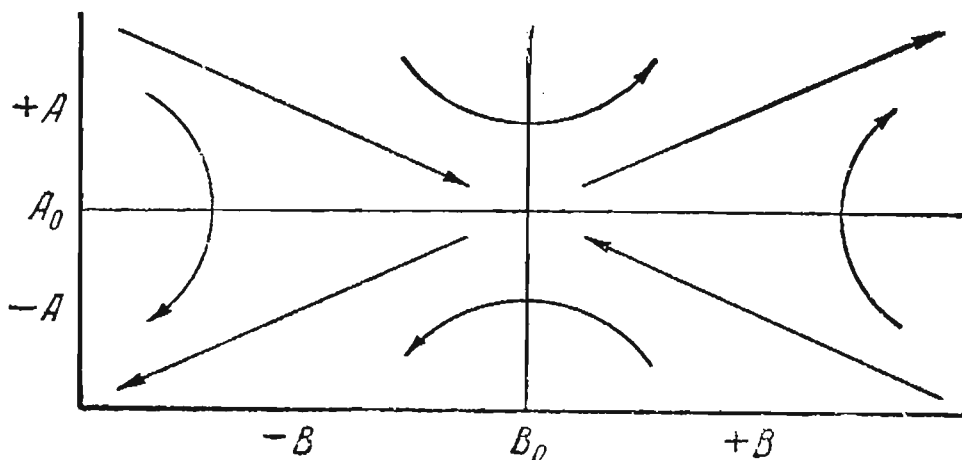


Рис. 2.

A_0 — уровень A , при котором B не меняется; ниже A_0 (т. е. при $-A$) стимуляция B со стороны A настолько мала, что B уменьшается; выше A_0 (при $+A$) стимуляция достаточна для нарастания B . Аналогичные значения в отношении A имеют B_0 , $-B$, $+B$.

«...Если какое-либо животное из низших, например планарию, — пишут Дж. Гексли и Г. де Бер [2], — разрезать на два кусочка поперечно, то обычно передний кусочек из своего заднего конца образует хвост, а задний из своего переднего конца — голову. Но если поперечный разрез сделать немного дальше назад по длине тела, то те клетки, которые в предыдущем эксперименте принадлежали к заднему кусочку и регенерировали голову, теперь будут принадлежать к переднему куску и регенерировать хвост. Следовательно, качество возникающего образования не может зависеть от локализации специфического клеточного материала или вещества, так как в таком случае нельзя было бы понять, каким образом из одинаковых тканей в одном случае может образоваться голова, а в другом — хвост». Рядом специальных исследований, главным образом Чайльда, было установлено, что в подобных случаях имеется «градиент» — постепенное убывание ряда физиологических показателей вдоль тела животного, обычно от головы к хвосту. При разрезе голову образует область с наиболее высоким в данном кусочке уровнем физиологических процессов, а хвост — с наиболее низким. При этом выяснилось, что область с наиболее высоким уровнем (головная, в данном случае), как правило, угнетает другие области («доминирует» над ними) и препятствует им подняться до такого же уровня, какой имеет она. В результате они достигают этого уровня лишь после удаления такой доминирующей части. Поэтому после удаления головы (доминирующей части) остающаяся область с наиболее высоким из оставшихся уровнем физиологических показателей не угнетается уже более ничем и сама достигает наиболее высокого возможного уровня, а потому при регенерации тоже может образовать голову.

Экспериментально было показано, однако, что и там, где одна зона (аналогично головной у планарии) угнетает другие, принципиально

в ряде случаев можно переопределить эти зоны, насильственно приведя их сильным воздействием к обратному положению. В тех случаях, когда градиентная система еще не определилась,— а это бывает в начале развития таких систем,— она возникает под влиянием столь слабых воздействий, что создается впечатление, будто их действие заключалось лишь в нарушении неустойчивого равновесия (типа $A \xrightleftharpoons{-} B$), а дальнейшее нарастание разностей идет уже автоматически*). По-видимому, многие процессы дифференцировки тканей могут быть объяснены таким же путем. Возможно, что и процессы «канализации» Уоддингтона [23] должны быть объяснены с этой же точки зрения.

в) А л ь т е р н а т и в н а я с м е н а с о с т о я н и й

Особая форма взаимодействий наблюдается в процессах, которые можно назвать альтернативными.

Мы можем наблюдать процессы, нормально протекающие в двух резко разграниченных формах. Такова, например, смена сна и бодрствования. У высших животных мы наблюдаем регулярную смену этих состояний. При этом, какой бы гипотезы ни придерживаться относительно самого механизма сна, остается несомненным, что физиологические предпосылки для сна (устомление высших отделов нервной системы) существуют во второй половине бодрствования в большей мере, чем во второй половине сна. В самом деле, с каждым часом бодрствования степень утомления нервной системы возрастает. Здесь не важна точная зависимость этого нарастания от времени. Важно лишь, что нарастание утомления идет монотонно. Точно так же во время сна идет более или менее монотонное снижение утомления до самого прекращения сна. Но если это так, то совершенно ясно, что в какой-нибудь момент к концу дня утомление нервных клеток должно быть значительно выше, чем в конце сна, под утро. И тем не менее в конце дня еще продолжается бодрствование, а в конце ночи—сон. Значит, дело не только в самих предпосылках, но и в каком-то физиологическом механизме, который, раз он пущен в ход, доводит процесс бодрствования до определенного предела, после которого пускается в ход такой же относительно устойчивый механизм сна.

Подобных механизмов в организме имеется целый ряд. Как на наиболее простой пример, можно указать хотя бы на процесс накопления мочи и ее удаления. Этот процесс протекает совершенно по тому же типу, как и смена сна и бодрствования, с тем только отличием, что степень утомления нервной системы заменяется степенью наполнения мочевого пузыря, а бодрствование и сон—периодами накопления мочи и мочеиспусканием. Приспособительный смысл подобного альтернативного хода процессов совершенно ясен. Необходимость слишком частой смены сна и бодрствования ставила бы животных в невыгодное положение. Тем более было бы невыгодно промежуточное состояние неполного торможения нервных процессов (реально встречающееся при некоторых патологических состояниях). То же касается и мочеиспускания: и непрерывное вытекание мочи и просто слишком частое мочеиспускание представляли бы ряд неудобств для приспособления организма к условиям его существования.

Как же осуществляется регуляция таких альтернативных процессов? Наиболее ясен случай периодического накопления и удаления мочи. Мы оставим пока в стороне случай сознательного регулирования мочеиспуска-

*) Здесь можно говорить о «пусковом механизме», но особого рода, где и направление действия этого пускового механизма зависит от направления исходного толчка.

ния, обычный у взрослых (хотя и здесь, за редкими исключениями особенно длительных волевых задержек *), прямая регуляция происходит в довольно узких пределах). Рассмотрим случай автоматической регуляции накопления и опорожнения пузыря. Он наблюдается у детей в раннем возрасте и у взрослых животных после перерезки спинного мозга выше поясничной области. В последнем случае, после снятия первоначального шокового состояния, устанавливается определенный ритм, также связанный с более или менее длительным накоплением мочи и затем полным опорожнением пузыря (Гольц, цит. по [8]). При этом известно, что переход к мочеиспусканию связан не с механическим преодолением сфинктера, запирающего мочевой пузырь, но вызывается рефлекторно путем раздражения растягивающейся слизистой и вызванным этим раздражением (через центр спинного мозга) сокращением пузыря и расслаблением сфинктера.

Очевидно, одно из двух чередующихся состояний не требует специального объяснения. Затруднение возникает только при объяснении относительно устойчивой смены первого состояния вторым состоянием. Так, например, накопление мочи при закрытом сфинктере протекает совершенно естественно за счет поступления мочи из почек. Известная степень наполнения приводит к тому растяжению слизистой, которое вызывает раздражение, уже превосходящее определенный порог. Наступившее «сверхпороговое» раздражение приводит к сокращению пузыря и расслаблению сфинктера. Когда часть мочи уже вытекла и тем самым «сверхпороговое» растяжение слизистой снято, процесс, казалось бы, мог прекратиться, а он продолжается до полного опорожнения пузыря. Такой характер процесса не может быть отнесен за счет простой длительности мышечного сокращения (тем более, что и она требует объяснения): произвольно, хотя и с усилием, вполне можно остановить неоконченное мочеиспускание, и позывы в большинстве случаев прекращаются значительно раньше, чем кончился бы нормальный процесс. То же еще отчетливее наблюдается и в отношении сна: если прервали сон, то уже легко сравнительно длительно сохранять бодрствование, иногда буквально после получасового сна.

Указания на механизм, поддерживающий процесс, дает изучение некоторых конкретных отношений. Оказывается, что тонус мускулатуры пузыря может подвергаться рефлекторному действию со стороны всех центростремительных нервов (за исключением *n. vagus*). Рефлекторная чувствительность пузыря оказывается в соответствующих местах чрезвычайно большой; слабые раздражения, которые еще не влияют на кровяное давление, могут вызвать энергичное сокращение пузыря. Как уже упомянуто выше, раздражение других чувствительных нервов может также вызвать сокращение пузыря, причем и это происходит при посредстве центральных аппаратов в нижней части спинного мозга. Этим объясняется, почему щекотка, согревание коленной области во сне может вызвать мочеиспускание [8].

В этих условиях совершенно ясно, что механизм поддержания процесса может быть очень прост. Как известно, небольшое раздражение уретры (мочеиспускательного канала) может вызвать позывы на мочеиспускание (чрезвычайно сильное раздражение иногда препятствует им). Это показывает, что центростремительные волокна, идущие от уретры, не являются исключением и тоже способствуют сокращению пузыря и расслаблению сфинктера. Если это так, то полное опорожнение пузыря обеспечивается совершенно автоматически: начавшееся мочеиспускание раздражает уретру, это в свою очередь стимулирует мочеиспускание и так до тех пор, пока полное опорожнение пузыря не сделает дальнейшее

*) Возможна и несознательная регуляция со стороны коры головного мозга — путем изменения местной реактивности и т. д.

мочеиспускание невозможным. В связи с этим раздражение уретры прекращается, и весь нервно-мышечный аппарат приходит в спокойное состояние. Возможно, что наряду с раздражением уретры быстрый переход слизистого пузыря из растянутого состояния к нормальному также дает раздражение, достаточное для стимуляции дальнейшего сокращения пузыря. Таким образом, весь ход процесса можно иллюстрировать схемой: сверхпороговое наполнение пузыря—первое раздражение слизистой—«запальная» порция мочи при мочеиспускании—раздражение уретры—мочеиспускание—раздражение уретры и т. д. (рис. 3). Когда после полного опорожнения

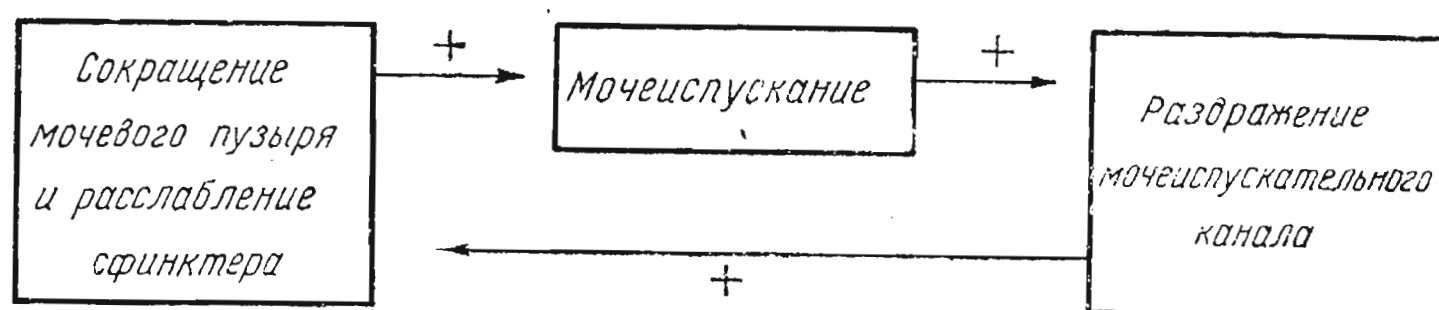


Рис. 3.

среднее звено (мочеиспускание) неизбежно выпадает, прекращается и вызываемое им раздражение уретры и, следовательно, возбуждение всего аппарата. В более полной форме процесс может быть представлен схемой на рис. 4, где весь механизм распадается на две системы обратных связей: одну (наполнение—мочеиспускание), дающую колебания вокруг среднего состояния по типу «плюс-минус», и другую (мочеиспускание—раздражение уретры), усиливающую эти колебания по типу «плюс-плюс» и доводящую раз начавшийся процесс (наполнение или опорожнение) до полного завершения.

Менее уверенно можно говорить о механизмах, поддерживающих альтернативную смену сна и бодрствования. Однако и там можно видеть известные аналогии описанному механизму, хотя, вероятно, далеко не исчерпывающие всей сложности явления.

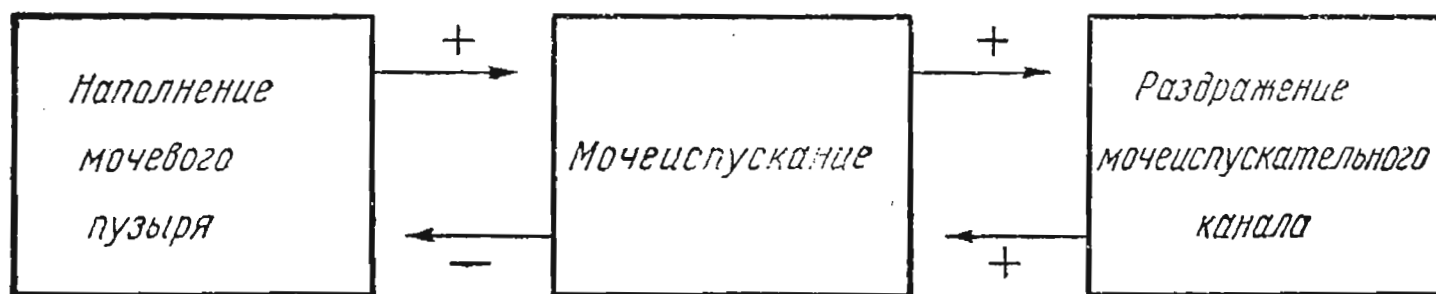


Рис. 4.

Как известно, во время сна автоматически выключается или снижается деятельность органов чувств, частью механически (зрение), частью путем, очевидно, центрального торможения. С другой стороны, известно, что полное выключение внешних раздражений легко приводит к сонному торможению. Известен случай с субъектом с нарушением большинства рецепторов, кроме зрения, где искусственное выключение зрения быстро приводило его ко сну (акад. И. П. Павлов—случай Штрюмпеля). Исходя из этих фактов, можно указать (не настаивая, впрочем, на этой гипотезе), что одним из существенных механизмов, поддерживающих сон, даже после значительного отдыха и, следовательно, снижения вызвавшего сон утомления, является опять-таки взаимная поддержка двух явлений: сонного деяможения высших отделов нервной системы и вызываемого им снижения тортельности органов чувств (в свою очередь исключаяющего внешние раздражения и поддерживающего этим торможение). Напротив, во время бодрствования мы имеем обратное соотношение, где возбуждение высших

отделов нервной системы поддерживает деятельность органов чувств, которые в свою очередь (через падающие на них раздражения) поддерживают тонус коры (рис. 5).

При произвольном вмешательстве в тот или иной процесс человек использует те же пути. Так, в случае необходимости опорожнения мочевого пузыря прежде естественно наступающих позывов человек (путем напряжения брюшного пресса для раздражающих сокращений произвольных мышц дна таза) вызывает лишь появление «запальной» порции мочи и дальше уже процесс течет так же автоматически, как и в случае непроизвольного хода [8]. Точно так же при необходимости уснуть или, наоборот, разбудить спящего, пускается в ход выключение внешних раздражений или, наоборот, включение их в особенно большой дозировке, и дальше достигнутое состояние поддерживается уже автоматически *).

Изложенный тип функций обеспечивающий альтернативное течение процессов, представляет довольно распространенное явление, хотя в настоящее время мы не во всех таких процессах можем вскрыть даже общие



Рис. 5.

очертания их механизма. Представляется вероятным, что к этому типу принадлежит регуляция женского полового цикла, а также, может быть, и ряд процессов, идущих по типу «все или ничего».

Таким образом, здесь определенное сочетание обратных связей типа «плюс-минус» и «плюс-плюс» (или «минус-минус», что в данном случае то же) создает механизм, дающий в организме альтернативную смену тех состояний, промежуточное положение между которыми является нежелательным.

В целом, подводя итоги рассмотрению систем с обратными связями, можно сказать, что в организме наблюдаются различные типы обратной связи и что каждому типу соответствует определенная форма функций: 1) тип «плюс-минус» способствует стабилизации функций (и поэтому легче выявляется при изучении организма, так как связан с длительно существующими функциями); 2) тип «плюс-плюс» способствует нарастанию функций; 3) тип «минус-минус» **) — дифференциации и уходу от среднего, неблагоприятного состояния (эмбриональное состояние тканей, гермафродитизм и т. д.); 4) определенные сочетания типа равновесия и «плюс-плюс» способствуют изменению порогов реакции органа и этим создают вместо сглаженных циклов альтернативные колебания органа между двумя крайними состояниями.

Изложенная классификация простейших биологических систем и их приспособительной роли представляет, разумеется, лишь самые первые шаги в этом вопросе. Чтобы яснее выявить особенности каждого типа, мы должны были в известной степени схематизировать явления, подчеркивая в каждом из них то, что в нем представлялось наиболее важным.

*) Мы оставляем в стороне конкретные механизмы, пускаемые здесь в ход (в первую очередь участие вегетативной нервной системы), поскольку нашей задачей является лишь выделение общего типа явлений, а не детальный анализ отдельных звеньев в различных физиологических процессах (сон, мочеиспускание и др.).

**) О различии типов «плюс-плюс» и «минус-минус» см. выше на стр. 164.

Однако, находя определенную систему на одном уровне интеграции некоторых явлений*), мы не должны забывать, что на других уровнях тех же явлений всегда имеются и другие типы взаимоотношений. Например, если мы правомерно рассматриваем систему рефлексов как дискретную (и то лишь функционально, а не анатомически), то отдельный рефлекс приходится рассматривать как систему цепных связей, притом часто осложненную обратными связями. «Звездно-лучевые» системы эндокринных корреляций почти всегда осложнены для каждого «луча» обратными связями, так же, как почти все звенья системы обратных связей являются центрами таких «звездных» подсистем. Например, половая железа является центром обширной звездной системы корреляции (вторичные половые признаки) и в то же время звеном в системе обратной связи с гипофизом и т. д. В альтернативных процессах мы видели примеры сочетания отрицательных («плюс-минус») и положительных («плюс-плюс») обратных связей и т. д. Одним словом, ни один тип систем не стоит особняком, а в каждом конкретном явлении так или иначе связан с другими.

Мы проанализировали здесь лишь весьма немногочисленные типы элементарных систем. Надо думать, что фактически даже таких элементарных форм гораздо больше. Тем больше должно быть число их сочетаний.

Наконец, следует учитывать, что биологически сходные результаты могут достигаться иногда путем применения неодинаковых механизмов. Так, простое на вид явление «регуляции» или компенсации может достигаться, как описали Н. А. Белов и М. М. Завадовский, путем «плюс-минус» взаимодействия. Но в других случаях такое восстановление нарушенного нормального состояния может идти путем простого процесса «текущей замены» тканей. Например, выстриженная шерсть млекопитающего, снятый поверхностный слой эпителия на коже или слизистой, обломанный кончик резца у грызуна могут постепенно восстановиться только за счет того, что все эти ткани, сменяясь, непрерывно возобновляются. Таким образом, удаленная сразу путем травмы часть ткани может возобновиться так, как если бы она была сменена постепенно естественным путем. Этот пассивный тип компенсации встречается обычно лишь при небольших травмах, но все же создает впечатление такой же «регуляции», как и активная «регуляция» путем включения обратной связи. Подобных примеров можно было бы привести много. Это все чрезвычайно важно иметь в виду, если ставить себе задачей раскрытие биологических механизмов и их синтеза в целом организме.

5. Некоторые общие соотношения управляющих систем в организмах и в машинных устройствах

В предыдущем изложении мы рассматривали различные типы систем с точки зрения их приспособленности к выполнению тех или иных функций. Надо, однако, помнить, что далеко не все связи имеют приспособительный характер. Мы привыкли а priori считать все явления в организме целесообразными, полученными в результате длительной приспособительной эволюции. На деле же определенная категория признаков и процессов в организме является следствием недостаточной длительности и успешности эволюции. Примером подобного признака, легко выявляемым при сравнении с более совершенными формами, является трехкамерное сердце двояко-

*) Под «уровнем интеграции» в биологии часто понимают степень сложности элементов, охваченных данной системой связей. Например, организм представляет собой наиболее высокий уровень интеграции, охватывающий все входящие в него органы. Связи внутри органа (между его частями, тканями)—более низкий уровень, между клетками одной ткани—еще более низкий и т. д.

дышащих рыб, амфибий и многих рептилий *). В других случаях, где мы не можем сравнивать уже развившиеся более совершенные формы, мы также должны помнить, что далеко не все структуры организма достигли максимального совершенства. Это важно иметь в виду и при сравнении систем организма с системами в различных машинных устройствах. Живая материя качественно отличается от неживой природы. В своей эволюции она достигла высокого совершенства выработанных естественным отбором приспособлений. Уже Н. Винер подчеркивал [35] огромное различие в габаритах управляющих систем машин и животных. Управляющие системы организмов в тысячи раз компактнее, чем соответствующие механизмы машин. С другой стороны, число элементов, например, в нервной системе, во много раз больше, чем у лучших современных машин. Сопоставляя количество нервных клеток мозга (порядка 10^{10} у человека) с числом элементов счетных машин (10^4), мы получили бы разницу в миллион раз. Иначе говоря, если бы мы могли свести размеры элементов счетной машины к размерам нервных клеток, мы получили бы машинный мозг всего в 1,5 миллиграмма весом **). Кроме того, и некоторые качественные особенности связей в нервной системе отличаются несравненно большим совершенством. Так, при увеличении объема запоминающих устройств у имеющихся машин поиски требуемых сведений («воспоминание») так осложняются, что требуют больше времени, чем в действующей медленнее, но более совершенной центральной нервной системе человека. Правда, и при этих условиях вычислительные машины в некоторых отношениях, в частности по скорости процессов, уже превосходят наши возможности. Однако в настоящий момент даже чисто формальные их возможности (кроме скорости) несоизмеримо малы по сравнению с мышлением человека.

Но если и качественно, и по ряду количественных показателей (богатство элементов, габариты) машины далеко уступают живому организму, то нельзя отрицать более широких возможностей в дальнейшем развитии машин, по крайней мере для некоторых их конструктивных особенностей, по сравнению с организмом.

Действительно, ряд биологов (Дж. Б. С. Холдэн [25], С. Райт [36] и др.) подчеркивали в различной форме тот факт, что живые организмы могут эволюционировать далеко не во всех направлениях. Реально эволюция идет не в том направлении, которое обещает наибольшие результаты в отдаленном будущем, а в том, которое дает наибольшее улучшение судьбы вида в ближайший период времени, хотя бы в дальнейшем это и было менее благоприятно. Животное и растение на каждом этапе своей эволюции должно быть максимально приспособленным к среде. Это ведет к тому, что даже довольно простые полезные сочетания мутаций не могут закрепиться в виде, если в отдельности эти мутации приносят вред (так как возникают они обычно в отдельности, а не в готовом сочетании). Если машину можно переконструировать, заменив в ней сразу группу частей, проанализировав еще в проекте все преимущества новой комбинации, то в организме эволюция позволяет, как правило, одновременную замену только одной, редко двух частей, и если они на первом же этапе не согласуются достаточно хорошо с остальными частями, то эволюция их отбрасывает, и новая конструкция, возможная и ценная только при замене сразу целой группы частей, вовсе не осуществляется. Это, несомненно, ограничивает возможности эволюции живых организмов и дает некоторые, хотя,

*) Трехкамерное сердце имеет два предсердия, но не имеет полной перегородки между двумя желудочками, вследствие чего возможно частичное смешение артериальной и венозной крови.

**) Фактически еще меньше, так как в состав мозга входят и связующие элементы, число которых растет быстрее, чем число основных элементов.

конечно, второстепенные, преимущества в конструировании машин. Грубым примером этой разницы является, вероятно, применение в технике с первых шагов ее развития поступательно-вращательного движения в виде колеса, оси, пропеллера и т. п., которые механически очень выгодны, но никогда не возникали в этой форме в живых организмах.

Таким образом, мы можем сформулировать обязательный для живых организмов принцип *непрерывности максимальной приспособленности в эволюции*, приходящий нередко в противоречие с конечной максимальной приспособленностью организма. Если эти противоречия по какой-либо причине были особенно велики, то они могли заводить в тупик целые группы видов. Так, достигнув определенного размера, животные часто продолжают уже закономерно эволюционировать в сторону увеличения роста, так как в этом случае особи большего размера получают существенное преимущество в защите от хищников и в борьбе за самку. Но увеличение роста животных через определенное время вызывает падение темпов эволюции (вследствие уменьшения численности вида на той же территории и более медленной смены поколений) и, наконец, приводит к вымиранию медленно приспособляющихся к новым условиям гигантов. Такие или аналогичные противоречия в какой-то мере имеются в каждой из ветвей животного и растительного царства. Однако поскольку жизнеспособность вида определяется его конкуренцией с другими видами, а развитие всех видов является в той или иной мере противоречивым и медленным, то и в борьбе за существование гибнут не все виды с недостаточно совершенным приспособлением, а только те, которые особенно отклоняются в неблагоприятную сторону.

В машинных устройствах при всей их элементарности по сравнению с живыми организмами некоторые из этих противоречий могут быть сняты благодаря большей дальновидности человеческого разума по сравнению со слепо действующим естественным отбором.

Другое существенное отличие большинства современных машинных устройств от живых организмов заключается в сравнительно большем богатстве взаимно независимых реакций у живых организмов. Выше мы уже указали, что причиной такой дискретности является необходимость гибкого приспособления к изменяющимся внешним условиям. Вычислительные машины, поскольку они не ведут самостоятельного существования, как правило, не должны сами обеспечивать свою безопасность и приспособляемость и работают в каждый определенный момент по одной определенной программе.

В высшей нервной деятельности животных и тем более человека особенно надо учитывать множественность, одновременность и комбинированность реакций центральной нервной системы. В психической жизни высших животных, и особенно человека, часто наблюдается своего рода «статистическое» решение вопроса, которое под влиянием тех или иных внешних или внутренних условий может сильно варьировать. Изучение изменения вероятностей тех или иных процессов в центральной нервной системе, можно думать, открывает дальнейший путь к анализу типов характера, а также развития бреда и ряда других проявлений нормальной и патологической высшей нервной деятельности человека, основы изучения которой заложены академиком И. П. Павловым.

Мы так упорно подчеркиваем отличия живых организмов от машинных устройств потому, что основная задача кибернетики—выявление общих принципов управляющих систем в живых организмах и машинах—может быть осуществлена лишь тогда, когда будут учтены специфические черты тех и других. При этом здесь надо выделить два типа различий. Первый тип—это частично уже рассмотренные нами различия самих объектов:

живых организмов и машин. Второй тип заключается в различии нашего подхода к тем и другим объектам. Машины мы конструируем сами, кроме того, они гораздо доступнее анализу благодаря своей относительной простоте. В организмах, наоборот, мы в сложном переплетении разнообразных явлений нередко с трудом выделяем основные типы взаимоотношения частей и определенные их системы. Здесь нам иногда приходится идти от результата к конструкции.

В живых системах мы обычно имеем дело с многими совокупностями неизвестных или лишь частично известных элементов, с неточно определенными числами и с возможностью лишь приблизительного установления типа их взаимоотношений. Все это ставит перед биологом,— если он хочет применять в своих исследованиях общие принципы кибернетики,— задачи, несколько отличные от задач математика или техника, изучающего принципы конструирования управляющих устройств в машинах, хотя цель в обоих случаях остается одна и та же.

С одной стороны, как видно из изложенного, подход биолога должен быть в значительной степени качественным, т. е. ориентироваться не на точные количественные данные, а на выяснение общих типов связей, допускающих ряд количественных отклонений без существенного отличия в конечных результатах. Так, в значительной степени старались идти и мы в описании управляющих систем.

С другой стороны, имея дело с более сложными и менее ясными отношениями, чем те, которые существуют в машинных устройствах, биолог не может установить четкую грань между собственно управляющими механизмами и различными другими типами, скажем, выполняющими ту или иную частную функцию. В результате в биологии приходится исследовать любые отношения частей, складывающихся в системы, выясняя их специфику и уже затем определяя их значение для организации управления в живых организмах. Дж. Нидхэм [14], характеризуя задачи научного исследования в биологии, писал: «Единицы, которые как мы предполагаем, различны по своему существу, могут быть введены в область научного изучения путем анализа их составных частей. Мы можем тогда сказать, что внутренние различия связаны с числом, природой или с отношениями этих частей». Природа частей является специфическим объектом описательной биологии, а число (если оно требует уже специальных методов)—объектом математической биологии. Третья сторона вопроса—выяснение типов отношения частей в живых организмах,— очевидно, должна составлять особую задачу, и она связана с выяснением принципов строения управляющих систем в организме. Действительно, любая, самая простая конструкция определяется не только составляющими ее частями и их числом, но и их расположением. Выяснение общих принципов архитектуры живых систем и является ключом к раскрытию той целостности, которая так поражает в живых организмах, которая часто служила поводом к признанию невозможности научного познания в этой области, и сущность которой в настоящее время начинает раскрываться благодаря изучению некоторых общих принципов строения управляющих саморегулирующихся систем.

6. Некоторые возможные направления практических приложений представлений о типах биологических систем

Остановимся на том значении, которые могут иметь изложенные представления для медицины.

В настоящее время имеется целый ряд заболеваний неинфекционной природы, этиология и патогенез которых еще не выяснены до конца. Сюда

относятся: гипертония, злокачественные новообразования, глаукома, такие заболевания, как язва желудка и двенадцатиперстной кишки, нейродермиты и, наконец, целый ряд нервно-психических заболеваний от эпилепсии, шизофрении и циркулярного психоза до истерии и психастении.

Полный анализ этих заболеваний, конечно, возможен только при широком накоплении материала и установлении связей, которые определяют основное развитие патологического процесса. Однако на современном этапе исследования, когда накоплено уже большое количество данных, изложенные представления о типах систем могут в известной мере облегчить разрешение вопроса путем исключения некоторых решений или указания на наиболее вероятные типы взаимоотношения органов, принимающих основное участие в патологическом процессе.

Бесспорно, что многие заболевания самого элементарного характера осложняются созданием обратных связей того или иного типа. В этом отношении можно указать хотя бы на такое хорошо изученное заболевание, как рахит. Как известно, рахит вызывается недостатком витамина D в пище. Однако оказывается, что для его излечения далеко не всегда бывает достаточно восстановить нормальный состав пищи путем добавления в нее снова витамина D.

В ряде случаев в результате развития рахита нарушается нормальная работа кишечника, и витамин D уже не может нормально усваиваться, даже если он имеется в пище [13]. Получается система с положительной обратной связью: недостаток витамина вызывает нарушения нормального функционирования кишечника, а нарушение функции кишечника создает в свою очередь недостаток витамина D. Таких случаев «порочного круга» в патологии можно встретить очень много. Наблюдаются и другие формы взаимодействия звеньев, определяющих создание патологического процесса. В настоящее время можно установить, по крайней мере, четыре типа патологических процессов.

К первому типу следует отнести те, которые связаны с нарастающим, так сказать, «лавинообразным» развитием процесса. Такой характер развития можно отметить у гипертонии, глаукомы, злокачественных новообразований и шизофрении. Здесь можно предположить, что в основе процесса лежит возникновение «порочного круга» того же типа, какой известен для рахита, т. е. положительной обратной связи («плюс-плюс»). Указания на такой характер процессов при этих заболеваниях действительно имеются. Так, при гипертонии изменение сосудов, возникшее под влиянием повышенного давления, в некоторых органах достигает такой степени, что снижение давления до нормы может неблагоприятно сказаться на функции органа, и организм вынужден по возможности сохранять высокое давление. Точно так же при так называемом патологическом развитии личности сокращение контакта больного с окружающей средой бесспорно усиливает неадекватную его реакцию на внешнюю обстановку, что, в свою очередь, является причиной конфликтов и травм, усугубляющих патологическое развитие личности. Наличие таких взаимоотношений еще не дает нам права делать определенное заключение об их решающем значении, но во всяком случае позволяет поставить соответствующие исследования. Аналогичные данные можно привести и для глаукомы.

Сложнее вопрос в отношении злокачественных новообразований и артериосклероза, если последний развивается без существенного повышения давления крови. Лавинообразное развитие злокачественных новообразований может происходить просто от своего рода «цепной реакции», когда одна злокачественная клетка порождает путем делений все новые. Хотя этот механизм бесспорно имеет место, но, возможно, здесь играет роль и другой момент: все большее ослабление организма под влиянием

злокачественной опухоли уменьшает его сопротивляемость, и это способствует более интенсивному ее развитию. На это указывают некоторые факты. Так, для прививки опухоли от одного животного другому требуется известный минимум материала, ниже которого опухоль у животных, как правило, не прививается. С другой стороны, наличие в литературе данных о единичных случаях «самоизлечения» злокачественных опухолей тоже как будто говорит о возможности подавления развития опухоли организмом и заставляет допускать их взаимоотношения по типу «минус-минус».

Так же осторожно надо подойти и к вопросу о развитии атеросклероза (без гипертонических явлений). И тут, помимо «самоусиливающегося» процесса, может идти нарастание явлений просто путем постоянной аккумуляции необратимых изменений в результате непрерывно идущих патологических влияний, например, постоянной интоксикации. В этом случае постановка специальных исследований о типе процесса, вероятно, позволила бы ближе подойти к разрешению вопроса о развитии заболевания. Рассматривая первый тип патологических процессов в целом (там, где он подтвержден прямыми исследованиями), можно себе представить, что в ряде случаев вполне возможно излечение, основанное на учете характера взаимодействия звеньев, определяющих заболевание. Для этого было бы достаточно заставить одно из основных звеньев временно действовать по нормальному типу и этим возвратить к норме и другие связанные с ним звенья. Так поступают при рахите, когда энергичным вмешательством, хотя бы временно, добиваются нормального поступления витамина D в организм и этим в достаточной мере нормализуют все функции организма, в том числе и усвоение витамина D.

Точно так же, когда недостаточная функция сердца приводит прямо (путем ослабления коронарного кровообращения) или косвенно к ухудшению работы сердечной мышцы и тоже создается «порочный круг», его прерывают временной стимуляцией сердечной мышцы различными фармакологическими средствами, в результате чего создаются условия для дальнейшей сравнительно нормальной деятельности и самой сердечной мышцы и всего организма в целом.

Ко второму типу патологических явлений можно отнести те, которые имеют тенденцию к определенному равновесию, имеющему, однако, совершенно не соответствующий норме характер. Возможно, что сюда следует отнести упорно возобновляющиеся трофические расстройства (язва желудка или двенадцатиперстной кишки, экземы и нейродермиты и т. д.), некоторые стабильные отношения в состоянии эндокринной системы, а из нервно-психических заболеваний — психастению с неврозом навязчивых состояний и ряд психопатий, не имеющих тенденции к развитию.

В этом случае, чтобы вызвать обратный ход процесса, недостаточно временных изменений в отдельных компонентах, так как патологическая равновесная система снова может возвратиться к прежнему состоянию. Здесь требуется уже более устойчивое изменение хотя бы одного из компонентов, участвующих в процессе. Однако реально эта задача является, может быть, более простой, чем в первом типе, так как устойчивое равновесное состояние дает больше времени и возможности вмешиваться в процесс, а размеры отклонения от нормы здесь почти никогда не могут достигнуть той степени, которая наблюдается при направленном развертывании процесса в первом типе. К тому же многие функции органов могут быть изменены в небольших пределах (что здесь нередко только и требуется) простым изменением условий их работы в том числе изменением питания, режима и т. д.

Третий тип патологических процессов может быть назван циклическим. Наиболее яркой иллюстрацией его является, вероятно, циркулярный

относятся: гипертония, злокачественные новообразования, глаукома, такие заболевания, как язва желудка и двенадцатиперстной кишки, нейродермиты и, наконец, целый ряд нервно-психических заболеваний от эпилепсии, шизофрении и циркулярного психоза до истерии и психастении.

Полный анализ этих заболеваний, конечно, возможен только при широком накоплении материала и установлении связей, которые определяют основное развитие патологического процесса. Однако на современном этапе исследования, когда накоплено уже большое количество данных, изложенные представления о типах систем могут в известной мере облегчить разрешение вопроса путем исключения некоторых решений или указания на наиболее вероятные типы взаимоотношения органов, принимающих основное участие в патологическом процессе.

Бесспорно, что многие заболевания самого элементарного характера осложняются созданием обратных связей того или иного типа. В этом отношении можно указать хотя бы на такое хорошо изученное заболевание, как рахит. Как известно, рахит вызывается недостатком витамина D в пище. Однако оказывается, что для его излечения далеко не всегда бывает достаточно восстановить нормальный состав пищи путем добавления в нее снова витамина D.

В ряде случаев в результате развития рахита нарушается нормальная работа кишечника, и витамин D уже не может нормально усваиваться, даже если он имеется в пище [13]. Получается система с положительной обратной связью: недостаток витамина вызывает нарушения нормального функционирования кишечника, а нарушение функции кишечника создает в свою очередь недостаток витамина D. Таких случаев «порочного круга» в патологии можно встретить очень много. Наблюдаются и другие формы взаимодействия звеньев, определяющих создание патологического процесса. В настоящее время можно установить, по крайней мере, четыре типа патологических процессов.

К первому типу следует отнести те, которые связаны с нарастающим, так сказать, «лавинообразным» развитием процесса. Такой характер развития можно отметить у гипертонии, глаукомы, злокачественных новообразований и шизофрении. Здесь можно предположить, что в основе процесса лежит возникновение «порочного круга» того же типа, какой известен для рахита, т. е. положительной обратной связи («плюс-плюс»). Указания на такой характер процессов при этих заболеваниях действительно имеются. Так, при гипертонии изменение сосудов, возникшее под влиянием повышенного давления, в некоторых органах достигает такой степени, что снижение давления до нормы может неблагоприятно сказаться на функции органа, и организм вынужден по возможности сохранять высокое давление. Точно так же при так называемом патологическом развитии личности сокращение контакта больного с окружающей средой бесспорно усиливает неадекватную его реакцию на внешнюю обстановку, что, в свою очередь, является причиной конфликтов и травм, усугубляющих патологическое развитие личности. Наличие таких взаимоотношений еще не дает нам права делать определенное заключение об их решающем значении, но во всяком случае позволяет поставить соответствующие исследования. Аналогичные данные можно привести и для глаукомы.

Сложнее вопрос в отношении злокачественных новообразований и артериосклероза, если последний развивается без существенного повышения давления крови. Лавинообразное развитие злокачественных новообразований может происходить просто от своего рода «цепной реакции», когда одна злокачественная клетка порождает путем делений все новые. Хотя этот механизм бесспорно имеет место, но, возможно, здесь играет роль и другой момент: все большее ослабление организма под влиянием

злокачественной опухоли уменьшает его сопротивляемость, и это способствует более интенсивному ее развитию. На это указывают некоторые факты. Так, для прививки опухоли от одного животного другому требуется известный минимум материала, ниже которого опухоль у животных, как правило, не прививается. С другой стороны, наличие в литературе данных о единичных случаях «самоизлечения» злокачественных опухолей тоже как будто говорит о возможности подавления развития опухоли организмом и заставляет допускать их взаимоотношения по типу «минус-минус».

Так же осторожно надо подойти и к вопросу о развитии артериосклероза (без гипертонических явлений). И тут, помимо «самоусиливающегося» процесса, может идти нарастание явлений просто путем постоянной аккумуляции необратимых изменений в результате непрерывно идущих патологических влияний, например, постоянной интоксикации. В этом случае постановка специальных исследований о типе процесса, вероятно, позволила бы ближе подойти к разрешению вопроса о развитии заболевания. Рассматривая первый тип патологических процессов в целом (там, где он подтвержден прямыми исследованиями), можно себе представить, что в ряде случаев вполне возможно излечение, основанное на учете характера взаимодействия звеньев, определяющих заболевание. Для этого было бы достаточно заставить одно из основных звеньев временно действовать по нормальному типу и этим возвратить к норме и другие связанные с ним звенья. Так поступают при рахите, когда энергичным вмешательством, хотя бы временно, добиваются нормального поступления витамина D в организм и этим в достаточной мере нормализуют все функции организма, в том числе и усвоение витамина D.

Точно так же, когда недостаточная функция сердца приводит прямо (путем ослабления коронарного кровообращения) или косвенно к ухудшению работы сердечной мышцы и тоже создается «порочный круг», его прерывают временной стимуляцией сердечной мышцы различными фармакологическими средствами, в результате чего создаются условия для дальнейшей сравнительно нормальной деятельности и самой сердечной мышцы и всего организма в целом.

Ко второму типу патологических явлений можно отнести те, которые имеют тенденцию к определенному равновесию, имеющему, однако, совершенно не соответствующий норме характер. Возможно, что сюда следует отнести упорно возобновляющиеся трофические расстройства (язва желудка или двенадцатиперстной кишки, экземы и нейродермиты и т. д.), некоторые стабильные отношения в состоянии эндокринной системы, а из нервно-психических заболеваний — психастению с неврозом навязчивых состояний и ряд психопатий, не имеющих тенденции к развитию.

В этом случае, чтобы вызвать обратный ход процесса, недостаточно временных изменений в отдельных компонентах, так как патологическая равновесная система снова может возвратиться к прежнему состоянию. Здесь требуется уже более устойчивое изменение хотя бы одного из компонентов, участвующих в процессе. Однако реально эта задача является, может быть, более простой, чем в первом типе, так как устойчивое равновесное состояние дает больше времени и возможности вмешиваться в процесс, а размеры отклонения от нормы здесь почти никогда не могут достигнуть той степени, которая наблюдается при направленном развертывании процесса в первом типе. К тому же многие функции органов могут быть изменены в небольших пределах (что здесь нередко только и требуется) простым изменением условий их работы в том числе изменением питания, режима и т. д.

Третий тип патологических процессов может быть назван циклическим. Наиболее яркой иллюстрацией его является, вероятно, циркулярный

психоз. В данном случае, очевидно, при сохранении общего уровня равновесия в организме имеет место какое-то нарушение его регуляции, приводящее к тому, что вместо точной установки на среднее состояние некоторых процессов обмена происходит их «раскачка» вокруг среднего состояния. Это может быть вызвано как запаздыванием реакции*), так и чрезмерно сильной реакцией регулирующих механизмов на отклонения в обратную сторону. Поскольку организм, как система, ограниченная в возможностях своей реакции, не дает этим колебаниям выйти за определенные пределы, размах волн не возрастает, но последние повторяются нередко с довольно большой правильностью.

Несколько иной характер имеют кратковременные «циклы» при эпилепсии. Они скорее имеют характер описанных выше «альтернативных» процессов, которые, раз начавшись, имеют тенденцию дойти до своего исчерпания, и лишь после этого может произойти возвращение к норме. В связи с этим простейшим практическим способом борьбы с эпилептическими припадками обычно является повышение того порога, переход которого пускает в ход весь патологический механизм, в дальнейшем уже действующий по типу самоусиливающейся системы положительной обратной связи («плюс-плюс»). Это достигается в первую очередь снижением возбудимости центральной нервной системы.

Принципы борьбы с циклическими патологическими процессами, очевидно, должны быть весьма различны в зависимости от того, чем обусловлено нарушение регуляции—запаздыванием реакции, ее избыточностью или же пуском в ход частично самоусиливающегося механизма. Однако и здесь, можно думать, ясное представление о различных возможных типах процесса может способствовать более точному его анализу, а в дальнейшем—и изысканию тех или иных методов борьбы с болезнью.

Четвертым типом, который мы можем наметить, является наличие постоянно действующего внешнего или внутреннего фактора, вносящего патологические нарушения в нормальную жизнь организма. В одних случаях это может быть недостаток питания, общий или какого-либо специального вида (например, авитаминоз), в других—патогенный микроб (туберкулез, тиф и т. д.), наконец, постоянно действующий фактор интоксикации продуктами обмена, всегда присутствующий в организме даже при нормальном состоянии.

Понятно, что этот четвертый тип (к которому чаще всего и сводят большинство заболеваний) может имитировать тот или иной описанный выше тип**) или даже быть источником для дальнейшего развития патологического процесса путем создания различных обратных связей (как это иногда бывает при рахите).

Не следует также забывать и то, что при многообразии связей в организме всякое заболевание, по-видимому, *включает самые различные типы взаимодействий*. Так, например, шизофрения типа «развития личности» при наличии основного механизма, связанного с участием обратной положительной связи, по-видимому, иногда включает и расстройство альтер-

*) Тенденция к равновесию в случае отрицательной обратной связи может приобретать характер постоянных колебаний вокруг среднего устойчивого положения. Условием для этого может быть запоздание получения информации следящим механизмом. Подобный случай имеет место, например, в численных соотношениях особей хищника и жертвы. Увеличение числа хищников приводит к уменьшению числа особей жертвы, но не сразу же. Увеличение числа особей жертвы способствует росту и числа хищников (обратная отрицательная связь), но тоже не сразу. В результате могут возникать регулярные, незатухающие колебания численности тех и других [32].

**) Например, малярия, создавая за счет самого патогенного фактора циклический характер заболевания, могла бы, если бы мы не знали причины, заставить нас думать о нарушении регуляции равновесия в организме и т. д.

нативной смены сна и бодрствования, вследствие чего нарушаются нормальное функционирование и отдых центральной нервной системы. С другой стороны, всегда возможно вмешательство и более общих механизмов компенсации нарушений, и если оно оказывается своевременным и достаточно интенсивным, мы имеем дело с улучшением состояния больного или даже с выздоровлением.

Мы здесь ставили вопрос только о выявлении *основного* типа процесса, установление которого может с наибольшими шансами дать нам в руки ключ к пониманию заболевания и к борьбе с ним.

Практическое значение изложенных здесь представлений о типах биологических систем, конечно, не ограничивается только той областью медицины, на которой мы хотели проиллюстрировать вероятные пути их использования. Для ряда практических вопросов как медицинских, так и сельскохозяйственных может иметь значение и вопрос о «корпускулярности» наследственных задатков, рефлекторных элементов и т. д. В частности, например, таким вопросом является борьба с появлением устойчивых к ядам рас вредителей сельского хозяйства и болезнетворных микроорганизмов. Известно, что при применении какого-либо специального яда у вредителя нередко путем отбора со временем вырабатывается раса, устойчивая к этому яду. Тогда приходится переходить на другой яд. Однако это не гарантирует от того, что вредитель приобретает устойчивость и к нему, и т. д. Нечто аналогичное наблюдается и в отношении патогенных микроорганизмов, вырабатывающих устойчивость к фармакологическим средствам.

С изложенной выше точки зрения представляется целесообразным для избежания подобных результатов применение не менее чем двух ядов одновременно. Действительно, возникновение мутации, повышающей устойчивость вида к одному яду, не так уже маловероятно. Когда же устойчивые особи размножаются, то легко может возникнуть новая мутация, повышающая устойчивость к другому яду и т. д. *Последовательное* возникновение таких мутаций практически может быть безграничным. Но, как мы видели, *одновременное* возникновение у одной особи двух мутаций, нужных для устойчивости против двух ядов, крайне маловероятно (так же как и возникновение одной мутации, имеющей одновременно оба полезных признака). Следовательно, для эффективной борьбы с вредителями или микробами, вероятно, целесообразно применять комбинацию двух или трех ядов. В ряде случаев несколько бóльшие затраты и трудности, представляемые этими мероприятиями в первый момент, более чем окупятся эффектом в дальнейшем.

С другой стороны, можно думать, что при поисках полезных наследственных комбинаций у сельскохозяйственных растений и животных селекционеры оставляют до сих пор много неиспользованных комбинаций так же, как это делает и природа, поскольку возможно, что отдельные генотипы, дающие порознь плохой эффект, вместе могут дать благоприятную комбинацию [34]. В известном смысле с реализацией подобных сложных благоприятных комбинаций мы имеем, видимо, дело в гибридах инцухтной кукурузы, где отдельные линии, порознь малоэффективные, в некоторых комбинациях дают чрезвычайно высокий эффект. Принципиально возможно, что и в других случаях (даже при создании устойчивых гомозиготных генотипов) неудачные порознь генотипические блоки в комбинации могут дать нужный эффект. Эволюционное уменьшение взаимодействия мутаций, хотя и должно идти все время, но, конечно, полная ликвидация взаимодействия невозможна. Поэтому с возможностью взаимодействия мутаций в известной мере надо считаться как в теории, так и в практической деятельности.

Значение представлений о свойствах и роли дискретных систем может быть очень велико и в других отношениях. Еще большее значение для практики медицины и сельского хозяйства может иметь использование развитых представлений о корреляционных системах в развитии и функционировании организма. Последние, однако, заслуживают более широкого и детального обсуждения, и мы не будем здесь на них останавливаться.

Можно подчеркнуть одно: несмотря на то, что представления о типах биологических управляющих систем бесспорно развиты еще далеко не достаточно и в них имеется много пробелов, уже теперь с первых шагов развития намечаются определенные пути их применения для решения не только теоретических, но и весьма реальных практических задач, которые ставят перед нами биологические науки со всеми их разветвлениями, включающими медицинские и сельскохозяйственные дисциплины.

Заключение

Выше мы изучали биологические системы, характеризующиеся различными формами связи между отдельными звеньями. Основные рассмотренные нами типы систем были следующие:

1. Дискретные или корпускулярные системы, состоящие из ряда элементов, непосредственно почти не связанных друг с другом и легко вступающих между собой в любые комбинации.

2. Системы с фиксированными отношениями отдельных звеньев, в частности, характеризующиеся цепной последовательностью отдельных звеньев или «звездным» расположением их вокруг одного центрального звена.

3. Системы из двух или более взаимно влияющих (по типу обратной связи) звеньев при условии их взаимной стимуляции, взаимного торможения или «плюс-минус» взаимодействия и комбинация последнего типа с взаимно стимулирующим (или тормозящим).

При этом оказалось, что в живых организмах сходные по общему смыслу приспособительные задачи, хотя и относящиеся к разным функциям и органам, часто решаются принципиально сходным образом с помощью систем одного типа. Так, приспособление к постоянно меняющейся в любых направлениях среде, как в процессе эволюции, так и в процессе выработки оптимальных форм поведения, обеспечивается за счет дискретных («корпускулярных») систем: в эволюционном процессе—в виде отдельных генетических единиц; в развитии поведения—в виде рефлекторных реакций, способных к сравнительно свободному комбинированию и формированию различных сочетаний, адекватно отвечающих на любые новые комбинации условий среды. Приспособления к постоянным условиям или к ограниченной регулярной смене условий достигаются созданием корреляционных систем, где одни звенья (органы, функции) находятся в постоянной тесной связи с другими.

Второй тип, сталкиваясь с первым, в ряде случаев приобретает определенную оптимальную форму, характеризующуюся, в частности, «звездно-лучевыми» связями (например, в эндокринных железах), наиболее удовлетворяющими как требованию прочности связей, так и возможности сравнительно независимой эволюции отдельных звеньев («лучей») системы.

В целом можно предположить такой ход эволюции. 1. Раздробление первичных, наиболее естественных с физикохимической стороны связей (примером которых являются корреляции, обусловленные авитаминозами, первичные сложные реакции нервной системы и пр.) или связей, оставшихся от предыдущего эволюционного этапа. Раздробление связей обеспечивает более гибкое и независимое приспособление различных звеньев

к новым условиям. 2. Создание связей (преимущественно «лучевого» типа) различных органов и функций, имеющих сходную и взаимно дополняющую приспособительную роль (например, половые и вторично половые признаки—с половой железой, механизм мобилизации различных функций—с симпатической нервной системой и группой симпатомиметических гуморальных факторов и т. д.) для приспособления к постоянным внешним или внутренним условиям.

Определение того или иного хода процессов в организме (устойчивого, нарастающего, циклического и т. д.) обуславливается системами тех или иных обратных связей. Так, приспособление к стабилизации функций осуществляется в основном за счет отрицательной обратной связи («плюс-минус» связи М. М. Завадовского и Н. А. Белова), как это происходит во взаимодействии большинства эндокринных желез у взрослых животных. Нарастающее развитие систем и процессы, ведущие к созданию их однородности, осуществляются в основном путем установления взаимно стимулирующих отношений между звеньями системы («плюс-плюс»).

Определенные типы биологических процессов, в частности альтернативных (сон—бодрствование, накопление мочи—мочеиспускание и т. д.), осуществляются часто комбинацией равновесной системы (типа «плюс-минус») и системы взаимной стимуляции процессов («плюс-плюс»).

Создание дифференцировки, в первую очередь по типу градиентных систем, а также процессов, мешающих организму сохранить примитивное недифференцированное состояние (например, в некоторых случаях от гермафродитизма), а возможно, и создание «стрелочных» механизмов (Уоддингтон) достигается с помощью взаимоугнетающих отношений между звеньями системы («минус-минус»).

В эволюции наблюдается переход от систем менее благоприятных для выполнения данной функции к системам более благоприятным. С этой точки зрения дается единое объяснение эволюции путем разделения функций тканей (А. А. Заварзин) и дифференциации реакций нервной системы, возникновению мозаичных форм развития яйца и т. д.— как эволюции на приспособленность к все возникающим новым, принципиально не повторяющимся условиям среды.

Обратные формы эволюции наблюдаются при постоянно повторяющихся комбинациях условий (покой—мобилизация организма; существование в водной—в воздушной среде, защищенные формы детенышей—самостоятельное существование взрослых особей; выполнение функций материнских—отцовских; морфозы у растений долинные и высокогорные, водные и сухопутные и т. д.). При этом там, где возможен анализ самого механизма связей, поражает удивительный параллелизм создающихся типов связи у далеких форм животных, например у полусухопутных рыб, амфибий и даже насекомых, у которых возникают во многих отношениях параллельные эндокринные системы.

Самые различные механизмы развития и эндокринно-нервной регуляции, если они имеют сходную по типу задачу (равновесие, циклические процессы, дифференциация или нарастание процесса), осуществляются также поразительно однородными по характеру связей системами.

Так, равновесие большинства эндокринных систем и процессы регуляции (например, компенсаторной) в других органах обуславливаются в большинстве изученных форм системами «плюс-минус». Такие различные по содержанию, но сходные по смене состояний «альтернативные» процессы, как сон и мочеиспускание, вероятно, также обуславливаются сходной в обоих случаях комбинацией обратных связей и т. д.

При этом следует учитывать, что все изложенные взаимоотношения являются не единственными в каждой конкретной системе органов и

реакций, а лишь основными, выделенными из весьма обширного комплекса, осложненного рядом других детальных механизмов. В связи с этим сравнение общих задач изучения систем в биологии с задачами, поставленными создателями кибернетического направления в науке, позволяет думать, что при изучении биологических систем на настоящем этапе развития биологии, является неизбежным, во-первых, широкое выявление типов взаимоотношения частей в системах, зачастую без возможности достаточно точных представлений о числе звеньев и о количественных характеристиках их действия, и, во-вторых, рационально глубокое изучение типов систем, включающих не только собственно управляющие системы с обратными связями, но и других, играющих определенную роль в биологических приспособительных механизмах.

Можно думать, что на настоящем этапе наших знаний о живых организмах изучение менее сложных и легче анализируемых механизмов развития и гуморальных связей для понимания общих типов биологических систем может дать не меньше, а может быть, и больше, чем изучение более сложных и труднее поддающихся расшифровке связей в центральной нервной системе.

Интересно, что в эволюции живых организмов наряду с качественными отличиями от искусственно создаваемых систем наблюдаются и некоторые ограничения в возникновении определенных приспособительных устройств, почти не имеющих шансов на развитие в порядке естественного отбора и легко создаваемых при сознательном проектировании целесообразных механизмов.

Наконец, следует отметить, что, несмотря на пока еще очень ограниченные заключения, которые мы здесь могли сделать, намечается, по-видимому, возможность, используя выясненные соотношения типов систем и определяемых ими процессов, ближе подойти к анализу основных механизмов некоторых патологических явлений и к решению ряда практических задач, например, в борьбе с микроорганизмами, вредителями и т. д.

Все это заставляет думать, что дальнейшее изучение типов биологических систем во всей его широте должно быть одной из важных задач современной биологии и определенных разделов кибернетики.

ЛИТЕРАТУРА

- [1] Белов Н. А., Физиология типов, Орел, Красная книга, 1924.
- [2] Гексли Дж. и де Бер Г., Экспериментальная эмбриология, М.—Л., Биомедгиз, 1936.
- [3] Завадовский М. М., Противоречивое взаимодействие между органами, М., Изд. МГУ, 1941.
- [4] Заварзин А. А., Об эволюционной динамике тканей, Архив биологических наук, т. 36, сер. А, вып. 1, 1934, 3—64.
- [5] Когхилл Дж. Э., Анатомия и проблема поведения, Биомедгиз, М.—Л., 1934.
- [6] Кэннон Б., Физиология эмоций, Л., Прибой, 1927.
- [7] Кудряшов Б. А., Биологические основы учения о витаминах, М., Советская наука, 1948.
- [8] Ландуа Л., Руководство по физиологии человека. т. 1, Берлин, 1921.
- [9] Малиновский А. А., Роль генетических и фенотипических явлений в эволюции вида, Изв. АН СССР, сер. биол., № 4, 1939.
- [10] Малиновский А. А., Закономерности наследственности в свете дарвиновского учения об отборе, Успехи совр. биологии, т. 14, вып. 1, 1941, 171—176.
- [11] Малиновский А. А., Объединение полезных признаков в процессе естественного отбора, Рефераты работ учреждений Отделения биологических наук АН СССР за 1941—1943 гг., 1945, 291—292.
- [12] Малиновский А. А., Типы взаимодействия и их значение в организме, Рефераты работ учреждений Отделения биологических наук АН СССР за 1941—1943 гг., 1945, 292—293.
- [13] Медовиков П. С., Рахит и его лечение, М.—Л., Госиздат, 1927.

- [14] Н и д х э м Д ж., На рубеже морфологии и биохимии, Успехи совр. биологии, т. 5, вып. 1, 1936, 36—51.
- [15] О р б е л и Л. А., Лекции по вопросам высшей нервной деятельности, М.—Л., Изд. АН СССР, 1945.
- [16] П а в л о в И. П., Физиология высшей нервной деятельности, Доклад на XIV Международном физиологическом конгрессе в Риме 2/IX 1932.
- [17] П а в л о в И. П., Условия деятельности и спокойного состояния больших полушарий, Собр. соч., т. 3, ч. 1, 1951, 290—298.
- [18] П е р е д е л ь с к и й А. А., Эндокринология беспозвоночных, Успехи совр. биологии, т. 10, вып. 1, 1939, 51—95.
- [19] П у ч к о в Н. В., Вегетативная нервная система как фактор регуляции фагоцитарной деятельности лейкоцитов, VII Всесоюзный съезд физиологов, биохимиков и фармакологов, доклады, 1947, 305—308.
- [20] С е в е р ц о в А. Н., Морфологические закономерности эволюции, М.—Л., Изд. АН СССР, 1939.
- [21] С о б о л е в С. А., К и т о в А. И., и Л я п у н о в А. А., Основные черты кибернетики, Вопросы философии, № 4, 1955, 136—148.
- [22] Т р о ф и м о в И. Е., Взаимодействие аналогичных генов, Биологический журнал, т. 3, вып. 2, 1934, 346—385.
- [23] У о д д и н г т о н К. Х., Канализация развития и наследование приобретенных признаков, Успехи совр. биологии, т. 18, вып. 3, 1944, 393—396.
- [24] Ф и л и п ч е н к о Ю. А., Экспериментальная зоология, М.—Л., Медгиз, 1932.
- [25] Х о л д э н Д ж. Б. С., Факторы эволюции, М.—Л., Медгиз, 1932.
- [26] Ч а й л д Ч. М., Роль организаторов в процессах развития, М., ИЛ., 1948.
- [27] Ш м а л ь г а у з е н И. И., Организм как целое в индивидуальном и историческом развитии, М.—Л., Изд. АН СССР, 1942.
- [28] A l l e n E., Sex and Internal Secretion, Baltimor and London, 1932.
- [29] D a n c h a k o f f V., Sur les effects morphogenetiques de la folliculine dans l'ebauche testiculaire du poulet, Comptes rendus de séances de la Societe de biologie, C XIX p. 11, 1935, 1117.
- [30] F i s c h e r R., The Genetical Theory of Natural Selection, Oxford, 1930.
- [31] H a r m s J. N., Wandlungen des Artgefuges, Tübingen, 1934.
- [32] K o l m o g o r o f f A. N., Sulla teoria di Volterra della lotta per l'esistenza, Giornale dell 'Istituto Italiano degli Affuari, Anno VII, n. 1, gennaio 1936, 74—80.
- [33] N a g e l W., Sympathicotomie und Vagotonie als Symptome physiologischer Zustände, Verh. Schweiz. Med. Biol. ges. 3. Schweiz. Med. Wschr., 66, Jahrg. 1936.
- [34] T i m o f f e e f - R e s s o w s k y N. W., Über die Vitalität einiger Genomutationen und ihrer Kombinationen bei Drosophila funebris und ihre Abhängigkeit vom genotypischen und vom äusseren Milieu.
- [35] W i e n e r N., Cybernetics, New York, 1948. Русский перевод: М., «Советское радио», 1958.
- [36] W r i g h t S., The role in evolution of mutation, inbreeding, crossbreeding and selection, Proc. 6-th, Int. Congress of Genetics, 1, 356—366.

Поступило в редакцию 15 V 1957.

НЕКОТОРЫЕ ПРИНЦИПЫ РАБОТЫ ГОЛОВНОГО МОЗГА

А. В. НАПАЛКОВ

(МОСКВА)

ВВЕДЕНИЕ

При изучении высшей нервной деятельности выявляются некоторые специфические формы и закономерности переработки информации, поступающей к животному из внешней среды. Вопросы, стоящие перед физиологией при изучении этого явления, очень близки к тем проблемам, с которыми сталкивается техническая кибернетика при создании различных управляющих систем.

Настоящая статья посвящена изложению экспериментальных исследований, связанных с изучением переработки внешней информации живыми организмами. Работа проводилась на кафедре физиологии высшей нервной деятельности МГУ (зав. кафедрой проф. Л. Г. Воронин). В выполнении экспериментов принимали участие студенты Е. В. Волошинова, Е. В. Штильман, М. И. Бобнева.

І. Цепи условных рефлексов

Одним из основных понятий, которыми пользуются физиологи является понятие цепи условных рефлексов. В это понятие вкладывается тот же смысл, который содержит понятие «программа работы». Программа работы электронных систем управления представляет собой ряд последовательных приказов или запланированных действий. После того как выполняется один из приказов, система управления получает информацию о результатах. Эта информация служит сигналом для следующего действия.

По такому же принципу работает и животное. В ответ на условный раздражитель оно совершает рефлекторное движение, которое производит определенное изменение во внешней среде. Информация об этих изменениях служит сигналом для последующего действия и т. д. [1, 6, 7].

Цепь рефлексов представляет собой проявление определенной информации, сохраняющейся в памяти животного. Ясно, что в организме животного имеются материальные носители этой информации—некоторые специальные структуры. При этом следует различать два вида структур. К первому относятся наследственно закрепленные морфологические структуры, образующиеся с момента рождения и остающиеся относительно неизменными в течение всей жизни. Они являются носителями цепей безусловных рефлексов. Иначе говоря, это—программы, аналогичные тем жестким программам работы, которые закладываются в электронную систему управления человеком. Ко второму виду относятся структуры, формирующиеся в результате выработки сложных систем временных связей (цепей условных рефлексов). Это—«функциональные» структуры, которые

легко образуются в течение жизни каждого животного. Цепи условных рефлексов аналогичны тем новым программам, которые создаются в процессе работы самообучающихся кибернетических систем.

Формирование цепей условных рефлексов происходит за счет информации, поступающей из внешней среды.

Информация поступает к животному в различное время и в различных формах. Для того чтобы выработалась цепь рефлексов, эта информация должна быть особым образом переработана. Важное значение при этом имеет тот факт, что организм, осуществляя двигательные реакции, сам активно воздействует на внешнюю среду, провоцируя появление новых внешних раздражителей. В ходе выработки цепи рефлексов между организмом и окружающей его средой возникают определенные формы взаимодействия, имеющие свои специфические закономерности. Изучение этих закономерностей имеет значение для обсуждаемой здесь проблемы.

Следует подчеркнуть, что человек и высшие животные обладают способностью не только пассивно воспринимать информацию, поступающую из внешнего мира, но и активно «добывать» эту информацию, провоцируя ее появление во внешней среде при помощи двигательных реакций.

Во внешней среде существуют определенные сложные системы объективных закономерностей. Рассмотрим в качестве примера серию актов, ведущих к достижению животным некоторой цели α . В абстрактной форме эту серию можно представить следующим образом. Если в обстановке A будет произведено действие 1, то во внешней среде возникнет явление B . Если после этого произвести действие 2, то явление B превратится в явление B , если теперь будет осуществлено действие 3, то это приведет к достижению цели α (например, к получению пищи). Эта система закономерностей изображена в виде следующей схемы:

$$A \xrightarrow{1} B \xrightarrow{2} B \xrightarrow{3} \alpha \text{ (достижение цели).}$$

Для изучения закономерностей, связанных с выработкой цепей рефлексов, применяется разработанная И. П. Павловым [8] методика искусственной выработки условных рефлексов, основанная на многократном совпадении по времени двух явлений внешнего мира (двух раздражителей).

Многократное совпадение этих раздражителей указывает на то, что в данном случае организм столкнулся с реальной закономерностью внешнего мира. Это дает основание для фиксации ее в памяти в форме условного рефлекса.

Большую роль играет также принцип подкрепления [8]. Это — механизм, обеспечивающий закрепление в памяти только тех закономерностей внешнего мира, которые связаны с удовлетворением потребностей животного. Следует упомянуть также о явлении так называемых «проб и ошибок» [12, 13] или о принципе «эфферентной генерализации» [2, 3] — механизме, играющем большую роль в возникновении «целесообразного», «разумного» поведения животных. В определенных условиях у животного возникает большое количество случайных «неразумных» движений. Те движения, которые приводят к нужным результатам, закрепляются. Таким образом, из большого количества движений как бы отбираются «целесообразные» (в данных условиях внешней среды) реакции, в результате чего возникает «целесообразное», «разумное» поведение.

Однако не все вопросы, связанные с формированием сложных цепей условных рефлексов, оказываются в настоящее время полностью решенными. Известно, что у животных новые формы поведения часто формируются в условиях, когда непосредственное подкрепление пищей отдель-

ных входящих в систему условных рефлексов оказывается невозможным. Животное в первый раз может получить пищу только после того, как оно полностью осуществит всю цепь рефлексов. Само же формирование новой формы поведения идет без подкрепления пищей (В. Келер [4] и др.).

Вопрос о закономерностях формирования сложных форм поведения в этом случае остается в значительной степени открытым.

При изучении этого вопроса следует различать два случая: 1) случай, когда животное не обладает накопленной ранее информацией и сталкивается с данной ситуацией впервые; 2) случай, когда животное уже имеет определенный запас информации и формирование нового поведения идет за счет использования ранее накопленной информации.

При решении этих вопросов большое значение имеет методика искусственного формирования у животных систем условных рефлексов различной сложности. Интересные исследования в этом направлении были осуществлены Л. Г. Ворониным [2, 3], П. С. Купаловым [5], Н. А. Рокотовой [10] и другими исследователями.

II. Формирование сложных систем условных рефлексов

Один из путей формирования сложных систем условных рефлексов связан с последовательным присоединением к уже выработанной системе новых условных рефлексов. При этом в качестве подкрепления используется один из условных раздражителей ранее уже выработанных рефлексов системы. После завершения всей цепи животному дается пища (двойное подкрепление). Выработка осуществляется следующим образом.

Сначала у животного обычным способом [2, 3] вырабатывается простой условный рефлекс; например, кролик в ответ на включение белой лампочки дергает зубами за специальное кольцо. Методика выработки этого рефлекса была многократно описана [2, 3, 6] и заключается в том, что тем или иным способом провоцируется нужное экспериментатору движение животного. В данном случае, как только кролик дергает за кольцо, к которому привязан кусочек моркови, ему тотчас дается корм из кормушки. После 6—12 повторений животное начинает все время дергать за кольцо. Тогда эта реакция связывается с условным сигналом. Включается белая лампочка. Если кролик дергает за кольцо, когда горит белый свет, то он получает пищу. Если он дергает, когда свет не горит, то это движение не подкрепляется (пища не дается). После 10—15 сочетаний кролик начинает дергать за кольцо только в ответ на включение белой лампочки (у него вырабатывается условнорефлекторная реакция).

Выработка второго звена цепи проводится точно так же, как и первого, с той лишь разницей, что в качестве подкрепления дается уже не пища, а условный раздражитель первого рефлекса—белый свет. При выработке третьего звена цепи в качестве подкрепления используется условный раздражитель второго звена и так далее. При этом вся цепь рефлексов после ее полного завершения каждый раз подкрепляется пищей (применяется двойное подкрепление). Таким путем можно быстро формировать цепи условных рефлексов, состоящие из 9—10 звеньев. Цепи рефлексов удобно изобразить в виде схемы, следующим образом:

$$A \xrightarrow{1} B \xrightarrow{2} C \xrightarrow{3} D \xrightarrow{4} \alpha.$$

В этой схеме буквами обозначены различные условные раздражители, а цифрами—различные условнорефлекторные движения.

Важно отметить, что на базе каждого из условных раздражителей цепи (используя их в качестве подкрепления) можно выработать несколько

логический механизм, механизм «сокращения» цепи. Из цепи рефлексов один за другим начинали выпадать отдельные звенья. Если при этом цепь рефлексов подкреплялась, то выпавшие звенья больше не восстанавливались. Если же подкрепление не давалось, то они возникали вновь. Таким образом, как бы отбирались те элементы системы, которые точно соответствовали условиям внешней среды (создавалось изоморфное отражение).

Далее были проведены эксперименты, связанные с изучением закономерностей формирования более сложных систем условных рефлексов. С этой целью мы еще более усложняли окружающую животное экспериментальную внешнюю среду. Мы вводили в нее такой раздражитель, который делал невозможным осуществление всей ранее выработанной цепи рефлексов. Подобные соотношения можно очень часто наблюдать в окружающем нас внешнем мире. Эту систему закономерностей внешнего мира можно изобразить в виде схемы следующим образом:

$$\underbrace{A \xrightarrow{1} B \xrightarrow{2} C \xrightarrow{3} \alpha}_{K} \quad (\text{пища})$$

В наших опытах была создана экспериментальная модель этих отношений. Мы вводили особый раздражитель K , например синий свет. Когда горел синий свет, то цепь рефлексов никогда не подкреплялась пищей, отдельные условнорефлекторные движения никогда не подкреплялись включением последующих условных раздражителей. При этом было обнаружено, что животное способно к выработке новой программы (системы рефлексов), которая точно отражает особенности этой сложной системы внешних закономерностей. После 5—6 опытов животное начинает реагировать на отдельные раздражители новой цепи рефлексов только тогда, когда выключен синий свет (K). Физиологи обычно обозначают этот раздражитель (в данном случае синий свет) термином «условный тормоз». Они говорят, что у животного выработалось условное торможение, которое проявляется в том, что включение синего света затормаживает всю цепь условных рефлексов.

Например, у голубя № 6 была выработана следующая цепь условных рефлексов. Включалась маленькая черная вертушка—голубь спрыгивал вниз. Включался синий свет—голубь прыгал на жердочку. Включался метроном—птица спрыгивала вниз. Включалась белая лампочка—голубь бил клювом по площадке, после чего он получал корм.

В качестве условного тормоза служило включение приглушенного звонка. Когда звенел звонок, то голубь не реагировал на условные раздражители.

В ходе проведения экспериментов было обращено внимание на то, что если после какого-нибудь из движений животного мы выключали условный тормоз (например, звонок), то животное начинало многократно производить это движение. Оно как бы само активно выключало условный тормоз. Это свойство было использовано нами в целях дальнейшего усложнения системы условных рефлексов. Мы сделали попытку выработать новые условнорефлекторные реакции, а позднее и короткие цепи рефлексов в условиях, когда в качестве подкрепления применялось выключение условного тормоза. Выработка цепей рефлексов производилась так же, как и выработка обычной пищевой системы условных рефлексов. Однако в качестве подкрепления мы пользовались не пищей, а выключением условного тормоза.

Например, на голубе № 6 эксперименты были осуществлены следующим образом. Включался условный тормоз—звонки. После этого мы ждали, когда голубь случайно перепрыгнет в левое отделение камеры, и тотчас после этого выключали условный тормоз. Было выяснено, что после 18 таких

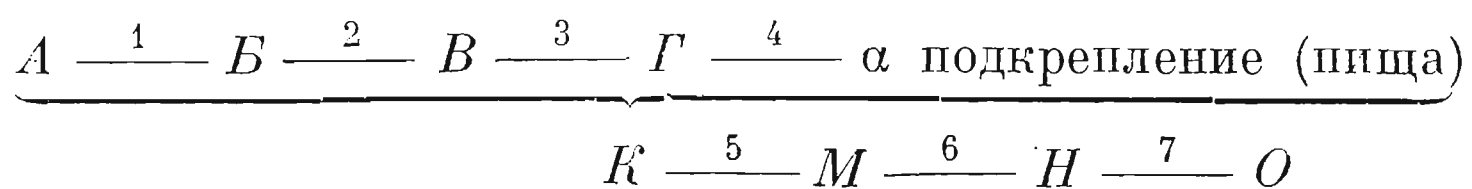
сочетаний у голубя выработалась новая условнорефлекторная реакция, и он стал каждый раз прыгать в левое отделение сразу же после того, как мы включали звонок. Таким образом, голубь как бы сам активно выключал условный тормоз. Следует при этом отметить, что для того, чтобы исключить возможность подкрепления этого движения одним из условных раздражителей цепи рефлексов, после того, как птица перепрыгивала в левое отделение, мы создали интервал от 45 сек. до 3 минут. Только после этого мы либо провоцировали обычную пищевую цепь рефлексов, либо снова включали условный тормоз (звонок).

Когда условнорефлекторное движение оказывалось прочно выработанным, мы приступали к формированию условнорефлекторной связи между этим движением и новым условным раздражителем, используя при этом обычную методику [2, 3]. Мы включали новый сигнал—свисток. Если голубь перепрыгивал в левое отделение в период, когда свистел свисток, то звонок выключался (давалось подкрепление). Если же голубь прыгал, когда свисток был выключен, то условный тормоз не выключался (подкрепление не давалось). После 64 сочетаний у голубя вырабатывался новый условный рефлекс.

Дальнейшее формирование цепи условных рефлексов производилось обычным способом [2, 3]. При выработке каждого нового из условных рефлексов цепи в качестве непосредственного подкрепления использовался один из условных раздражителей ранее выработанных звеньев системы. Однако после завершения всей цепи вместо подкрепления пищей выключался условный тормоз.

У всех животных удавалось выработать таким образом добавочные, боковые цепи условных рефлексов (растормаживающие цепи рефлексов), состоящие из 2—3 звеньев. Например, у голубя № 6 была выработана следующая растормаживающая цепь рефлексов. Включалось бульканье—голубь прыгал на проволоку. Тогда включался зрительный раздражитель в виде прямоугольника—голубь спрыгивал вниз. И, наконец, включался свисток—птица перепрыгивала в левое отделение камеры.

Таким образом, оказалось возможным формировать сложные системы, состоящие из двух цепей условных рефлексов. Эти цепи были в значительной степени независимы друг от друга. Одна из цепей условных рефлексов подкреплялась пищей и представляла собой пищевую цепь рефлексов. При выработке другой в качестве подкрепления использовалось выключение условного тормоза, и она могла быть обозначена как «растормаживающая» цепь рефлексов. Между обеими этими цепями мог быть создан значительный интервал времени. Обе цепи оказывались, однако, объединенными в одну единую систему, так как имели одно общее конечное подкрепление в виде пищи. Такие системы могут быть изображены следующей схемой:

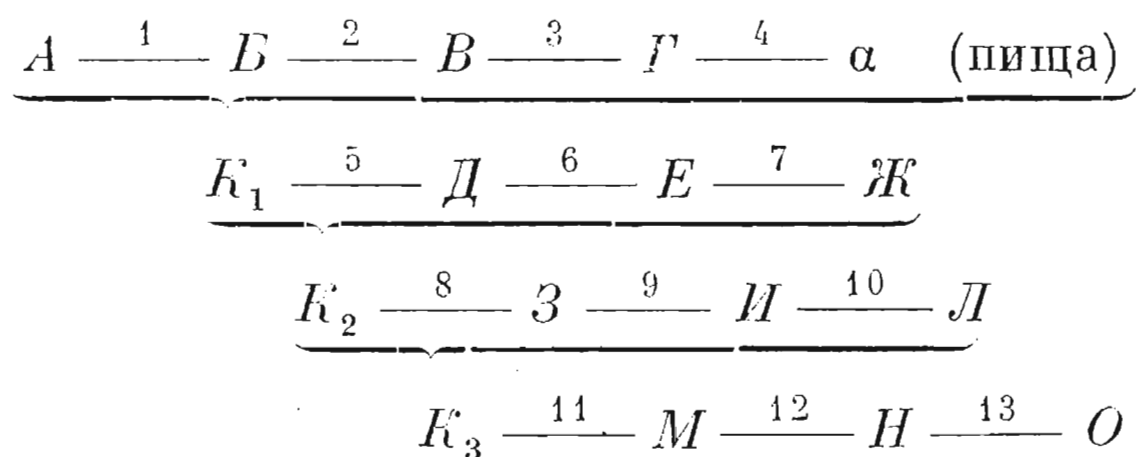


В этой схеме буквами обозначены различные условные раздражители, а цифрами различные двигательные реакции. Буквой *K* обозначен условный тормоз. $A \xrightarrow{1} B \xrightarrow{2} B \xrightarrow{3} \Gamma \xrightarrow{4}$ — пищевая цепь рефлексов, $\xrightarrow{5} M \xrightarrow{6} H \xrightarrow{7} O$ — растормаживающая цепь рефлексов.

Следует подчеркнуть, что описанные выше приемы открывают возможности для беспредельного усложнения системы. Это особенно ярко проявилось при проведении экспериментов на людях. На человеке оказалось возможным к боковой растормаживающей цепи рефлексов ($\xrightarrow{5} M \xrightarrow{6} H \xrightarrow{7} O$)

выработать свой собственный условный тормоз K_2 . Используя выключение этого тормоза как подкрепление, удалось выработать новую, вторую растормаживающую цепь рефлексов. К этой второй растормаживающей цепи удалось снова выработать условный тормоз K_3 и так далее.

Таким образом, удалось формировать «многоступенчатые», «многофазные» системы. Приводим схематическое изображение такой системы:



Приведенные экспериментальные данные позволяют сделать некоторые выводы относительно закономерностей формирования сложных систем условных рефлексов. Одним из путей выработки является путь последовательного усложнения системы. При этом решающее значение имеет тот факт, что в ходе самого формирования системы возникают такие раздражители, которые в дальнейшем могут служить подкреплением при выработке новых условных рефлексов и новых цепей условных рефлексов. Это правило создает предпосылки для быстрого разрастания и усложнения системы условных рефлексов. Если бы при выработке новых звеньев системы всегда было необходимо подкрепление пищей, то формирование сложных систем рефлексов оказалось бы невозможным. Как уже говорилось, было установлено, что в качестве подкрепления могут служить включение условных сигналов ранее выработанных рефлексов, а также выключение условного тормоза. Важно отметить, что при этом подкрепление делалось многократным или многофазным. Наряду с подкреплением отдельных условных рефлексов большое значение приобрело подкрепление цепей рефлексов, как единого целостного явления, и наконец, подкрепление всей системы (пища). Таким образом, для существования ее было необходимо двухкратное или трехкратное подкрепление.

Второй способ выработки отражал случай, когда последовательное усложнение системы оказывалось невозможным, и конечное подкрепление могло применяться только после полного завершения системы. В этом случае в основе формирования цепей рефлексов лежала ориентировочно-исследовательская деятельность. Появление нового раздражителя, вызывающее ориентировочную реакцию, стимулировало дальнейшие поиски и закрепило определенные формы поведения.

Приведенные экспериментальные данные позволяют сделать также некоторые выводы о принципах организации сложных систем условных рефлексов.

Описанные исследования, как нам кажется, могут иметь значение для разработки некоторых вопросов, связанных с изучением самонастраивающихся кибернетических систем.

III. Переработка вновь поступившей информации на основе ранее накопленной

Большой интерес представляет изучение закономерностей переработки вновь поступающей информации в системах, имеющих запас ранее накопленной информации. Мы попытались подойти к изучению этой проблемы, используя методику искусственной выработки условных рефлексов.

У животных вырабатывались сложные системы условных рефлексов. Затем в ходе эксперимента, включая тот или иной комплекс условных раздражителей, мы искусственно создавали различные по своей природе потоки информации, поступающей к животному. В ряде случаев создавалась также и новая потребность, например, у животных, у которых была выработана оборонительная цепь рефлексов, мы вызывали жажду или голод. В этих условиях изучался процесс формирования новых форм поведения, приводящий к удовлетворению новой потребности. При этом мы точно знали характер той информации, которая находилась в памяти животного, так как эта информация была нами ранее искусственно выработана в форме цепей условных рефлексов.

Мы могли создавать любые ситуации, произвольно включая те или иные сигналы, располагая, таким образом, удобной моделью, на которой могли быть исследованы некоторые закономерности, связанные с переработкой вновь поступающей к животному информации.

Как уже говорилось, выработанная в наших опытах система условных рефлексов могла быть изображена в виде следующей схемы:

$$\underbrace{A \xrightarrow{1} B \xrightarrow{2} B \xrightarrow{3} \Gamma \xrightarrow{4} \alpha}_{K \xrightarrow{5} M \xrightarrow{6} H \xrightarrow{7} O} \quad \text{подкрепление (пища)}$$

Эта система могла быть еще более усложнена за счет ее многократного ветвления.

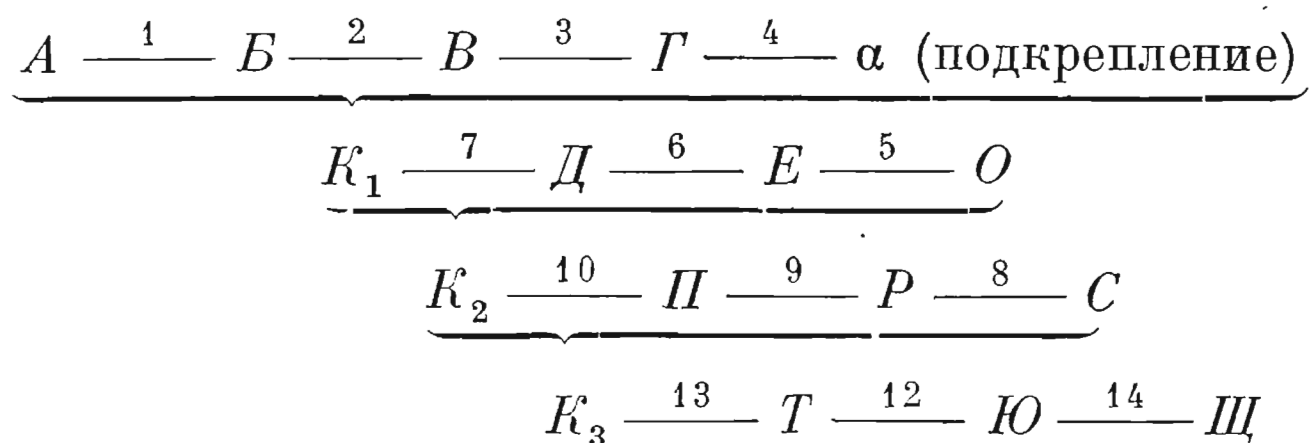
В первой серии опытов мы на длительное время включали условный тормоз K . Этот раздражитель сам по себе не вызывал никакой рефлексорной реакции. На фоне действия этого раздражителя мы начинали затем применять различные условные раздражители основной пищевой цепи рефлексов (A, B, B, Γ). При этом было обнаружено, что животное не реагирует на эти раздражители.

После этого на фоне действия раздражителя K мы начинали включать раздражители боковой растормаживающей цепи условных рефлексов (M, H, O). Было обнаружено, что животное каждый раз осуществляло соответствующую рефлексорную реакцию.

Во второй серии опытов мы выключали условный тормоз (K). После этого, так же как и в первой серии опытов, мы начинали включать условные раздражители основной, пищевой цепи рефлексов (A, B, B, Γ), а затем боковой растормаживающей цепи (M, H, O). При этом было обнаружено, что животные начинали реагировать на условные раздражители пищевой цепи (A, B, B, Γ) и в то же время переставали отвечать реакциями на условные сигналы растормаживающей цепи рефлексов (M, H, O). Условный тормоз K (звонок и др.) являлся, таким образом, условным сигналом, который, затормаживая один вид деятельности, переключал нервную систему на осуществление другой цепи рефлексов.

Следует специально обратить внимание на то, что, как показали эксперименты, человек и животные начинали реагировать на условные сигналы боковой «растормаживающей» цепи рефлексов (M, H, O) только после того, как мы включали условнотормозной раздражитель (K), причем в этом случае в ходе выработки системы рефлексов не существовало никаких предпосылок для развития процесса внутреннего торможения. Здесь мы сталкиваемся, очевидно, с новым явлением, когда один условный «включающий» раздражитель (K), который сам по себе не вызывает никакой определенной условнорефлексорной двигательной реакции, приводит в действие определенную цепь условных рефлексов (делает активной целую группу условнорефлексорных реакций) $O \xrightarrow{7} H \xrightarrow{6} M \xrightarrow{5}$.

В опытах на человеке, как уже говорилось, были выработаны еще более сложные системы рефлексов, имеющие форму:



При изучении этой сложной системы подтвердились те результаты, которые были ранее получены у животных на более простой системе условных рефлексов. Следует, однако, подчеркнуть, что такая многофазная система могла обеспечить уже значительно большее количество вариантов поведения при разнообразных формах внешней информации. В наших опытах мы создавали самые разнообразные варианты поступления внешней информации, например, включали раздражители K_1 , K_2 , K_3 , Ш, С, Е и А или K_2 , K_1 и Γ ; О и Υ и т. д. При этом каждый раз мы наблюдали формирование особых «целесообразных» форм поведения. Например, в том случае, если давался комплекс B, P, K_1, K_2 и О, поведение носило следующий характер. Осуществлялась цепь $P \xrightarrow{9} \Pi \xrightarrow{10} O \xrightarrow{5} E \xrightarrow{6} D \xrightarrow{7}$ и затем $B \xrightarrow{2} B \xrightarrow{3} \Gamma$.

В другой серии опытов создавались еще более сложные ситуации. У человека и животных создавалась новая потребность (цель). Например, испытуемому, у которого система рефлексов вырабатывалась на основе подкрепления включением белой лампочки, давалась новая инструкция — добиваться не того, чтобы горела белая лампочка, а того, чтобы зазвенел звонок. В экспериментах на животных, у которых была выработана оборонительная система условных рефлексов, мы вызывали у последних голод или жажду. Затем в ходе экспериментов создавались различные ситуации путем включения различных комплексов выработанной ранее системы рефлексов. В этих условиях было обнаружено, что выработанная ранее система рефлексов оказывалась использованной в той или иной степени при формировании нового поведения, служащего для удовлетворения другой потребности (жажды, голода и т. д.). Например, элементы оборонительной цепи рефлексов оказывались включенными в новую пищевую систему условных рефлексов и т. д.

Примененная нами форма опытов открывала определенные возможности для изучения некоторых закономерностей переработки информации. В ходе экспериментов испытуемому направлялась информация двух видов: 1) информация о новой потребности (цели работы), 2) информация о наличии во внешней среде тех или иных раздражителей (условиях деятельности). Испытуемым осуществлялась переработка этой информации на базе информации, накопленной ранее и сохраняющейся в памяти. Точно зная как характер поступающей информации, так и характер накопленной ранее информации и наблюдая за формированием новых форм поведения животного (новых программ работы), мы могли сделать некоторые выводы о характере сложных форм его аналитико-синтетической деятельности.

Мы видели, что в ходе наших экспериментов в ответ на создание различных ситуаций возникло большое количество разнообразных форм поведения (новых систем условных рефлексов или программ работы). При этом программы каждый раз включали элементы всех трех перечисленных выше видов информации и являлись сложным результатом их переработки; ни один из видов информации сам по себе не мог предопределить характера новой программы работы. Каждый раз формировалось такое

поведение, которое приводило к удовлетворению новой потребности и точно соответствовало характеру создаваемых нами внешних условий. Эксперименты указывали, таким образом, на то, что процесс формирования новых форм поведения не может быть сведен к простому осуществлению какой-либо уже выработанной ранее цепи условных рефлексов. В наших экспериментах при создании ситуаций, с которыми человек или животное встречались впервые, формировались совершенно новые формы «целесообразного» поведения. Это поведение включало, конечно, элементы прошлого опыта, однако сама комбинация этих элементов каждый раз возникала впервые.

Важно подчеркнуть также, что при этом новая форма поведения представляла собой единую систему, а не простую сумму рефлекторных ответов на условные сигналы, входящие в новый комплекс раздражителей. Отдельные раздражители внешней среды оказывали влияние на характер всей возникающей в ходе опыта новой формы поведения.

В экспериментах выявились две различные по своим свойствам группы условных раздражителей. При включении раздражителей первой группы (пусковые раздражители) наблюдалось появление определенных рефлекторных реакций. Включение раздражителей второй группы не приводило к появлению каких-либо конкретных рефлекторных ответов. Включение этих сигналов, однако, изменяло характер последующих реакций животного на «пусковые» условные сигналы.

Было выяснено, что в обычных условиях человек и животные не отвечают условными сигналами на включение «пусковых» раздражителей. Для того чтобы осуществилась реакция, необходимо наличие определенного «включающего» сигнала, который приводит в действие ту или иную цепь условных рефлексов. У каждой из систем рефлексов имелся свой собственный «включающий» раздражитель. Для осуществления некоторых форм поведения было необходимо наличие нескольких «включающих» раздражителей. Это явление имеет, видимо, большое значение. Учитывая очень большое количество вырабатывающихся у животных в течение их жизни цепей условных рефлексов, следует сделать вывод, что и количество условных раздражителей, действующих в каждый момент на нервную систему животного, в окружающей внешней среде должно быть очень большим. Многие из этих раздражителей присутствуют во внешней среде постоянно. Если бы животные реагировали на все условные раздражители, то их поведение представляло бы собой хаос непрерывных бессмысленных движений. О сложном, целенаправленном поведении не могло бы быть и речи. А между тем мы знаем, что поведение животных всегда имеет целенаправленный и целесообразный характер. Целенаправленность поведения обеспечивается, видимо, в результате системы субординации условных раздражителей разных категорий. Наличие одного «включающего» сигнала приводит к тому, что животное начинает реагировать на определенную группу «включающих» сигналов второй категории. Наличие одного из этих сигналов приводит к возникновению условнорефлекторных реакций на новую группу раздражителей и так далее.

Все это приводит к тому, что в каждый данный момент животное реагирует лишь на небольшую группу условных раздражителей. Таким образом, для осуществления какой-либо одной рефлекторной реакции часто оказывается необходимым наличие двух, трех и более сигналов. С подобным же явлением мы встречаемся при анализе работы некоторых кибернетических машин [9].

Далее можно сделать вывод, что живое существо как управляющая система может длительно находиться в различных состояниях, и именно состояние определяет реакции системы на различные «пусковые» условные сигналы. Мы видели в ходе наших опытов, что включение условного тор-

мозга (K) изменяло характер рефлекторных ответов системы на целый ряд раздражителей. При этом это «состояние» удерживалось в течение неограниченно долгого времени. Согласно представлениям А. А. Ухтомского [14] можно думать, что в основе этого явления лежит длительное состояние возбуждения определенных «констелляций нервных центров» (доминанта).

Наконец, следует указать на факт выработки новых условных рефлексов без непосредственного совпадения двух сигналов внешнего мира. Мы видели, что если один из сигналов оборонительной цепи рефлексов окажется связанным с пищей, то вся система используется животным для получения пищи, т. е. все другие раздражители, входящие в данную цепь рефлексов, оказываются связанными с получением пищи. Появляется целая группа новых пищевых условных рефлексов.

IV. Обсуждение экспериментальных данных

Одной из характерных особенностей животных является возможность осуществлять очень большое количество различных форм поведения. Простейшим объяснением этой особенности, казалось бы, могло явиться предположение, что для каждой из этих форм поведения существует своя собственная программа (цепь рефлексов), которая осуществляется под действием соответствующих условных раздражителей. При этом каждый из условных раздражителей должен был бы всегда вызывать соответствующую условнорефлекторную реакцию. Независимо друг от друга должно было существовать большое количество оборонительных, пищевых, половых и т. д. цепей условных рефлексов.

Согласно этой системе представлений широта возможностей в формировании новых форм поведения находилась бы в непосредственной зависимости от объема памяти. При этом интересующая нас проблема взаимоотношения вновь поступающей информации с накопленной ранее информацией находила бы очень простое разрешение.

Система управления, в основу деятельности которой были бы положены описанные выше принципы, обладала бы относительно жестким режимом работы. Формирование новых программ работы и переформирование старых, связанное с изменениями условий внешней среды, могло бы осуществляться только за счет выработки новых условнорефлекторных связей и угашения ранее выработанных условных рефлексов, что, как известно, требует значительного времени.

Существует другое решение описанной выше проблемы, связанной с обеспечением большого количества различных форм поведения животных. В системе управления могут отсутствовать готовые программы для каждого из возможных случаев поведения. В памяти такой системы может храниться минимум информации, на базе которой в каждом отдельном случае (при создании новых внешних условий) формируется новая программа. Информация в этом случае должна сохраняться в системе памяти в недействующем, неактивном состоянии. Животное должно отвечать на соответствующие условные сигналы только после того, как в результате сложных процессов переработки сохраняющейся в памяти информации возникнет та или иная программа.

Такое устройство делает режим работы системы управления менее жестким, более лабильным, и она может приспосабливаться к меняющимся условиям внешней среды быстрее. Объем памяти при этом может быть значительно меньшим.

Следует, однако, отметить, что описываемая система управления должна иметь ряд дополнительных механизмов, необходимых для того,

чтобы в нужный момент времени из памяти могла быть быстро извлечена необходимая информация. При этом встает вопрос об определенной структуре, в которой должна сохраняться вся накопленная животным информация. Без наличия такой структуры оказалось бы необходимым каждый раз пересматривать всю хранящуюся в памяти информацию, чтобы выявить ту, которая нужна в данный момент. Далее оказывается необходимым существование специальных механизмов, обеспечивающих пересмотр информации, и сопоставление вновь поступающей информации с накопленной ранее информацией, а также механизмов, лежащих в основе перегруппировки информации различного вида.

Широта возможностей выработки новых форм поведения описываемой системы управления определяется уже не только объемом памяти, но и совершенством перечисленных выше механизмов, которые могут быть охарактеризованы как механизмы аналитико-синтетической деятельности.

Нам кажется, что совокупность имеющихся в распоряжении физиологов сведений указывает на то, что животное работает по второй описанной схеме. Это подтверждается и приведенными нами экспериментальными данными. В связи с этим встает вопрос об изучении физиологических механизмов, обеспечивающих сложные формы переработки информации.

Примененная нами экспериментальная методика не дает возможности делать непосредственные выводы о характере нервных процессов, разыгрывающихся в головном мозгу.

Однако наши опыты могут дать некоторые косвенные данные для того, чтобы, исходя из современных достижений в области кибернетики и физиологии, была создана определенная рабочая гипотеза. Такая гипотеза должна, очевидно, объяснять все факты, известные в настоящее время физиологам. Она должна претерпевать соответствующие изменения в случае появления новых, противоречащих ей фактов.

Мы уже говорили, что при формировании новых форм поведения к животному поступает информация двух видов: 1) информация о новой потребности, 2) информация об условиях внешней ситуации (о комплексе внешних раздражителей). Выработка новых форм поведения осуществляется на базе информации, сохраняющейся в памяти. При этом важное значение, видимо, должно иметь сопоставление информации всех трех видов. В результате этого сопоставления из всей хранящейся в памяти информации должна быть выделена та, которая может оказаться необходимой для удовлетворения потребности организма и которая может быть применена в данных конкретных условиях.

Можно высказать предположение, что эту задачу выполняет механизм продвижения возбуждения по системам ранее выработанных условнорефлекторных связей. При этом сами системы нервных клеток, которые сохраняют информацию, остаются неподвижными. Однако в результате того, что волна возбуждения последовательно приводит эти клетки одну за другой в активное состояние, оказывается возможным сопоставление информации двух видов. Одновременно в эти же нервные клетки поступает возбуждение, вызываемое действием раздражителей внешней среды. Если две названные выше волны возбуждения попадают одновременно в нервную клетку, то в результате суммации возбуждения нервная клетка приходит в возбужденное состояние, и тем самым фиксирует совпадение наличия одного и того же компонента (раздражителя) во внешней среде и в накопленной ранее (хранящейся в памяти) информации. Процесс суммации играет при этом, видимо, ту же роль, которую в кибернетических машинах играет схема совпадения.

Можно предположить, что при возникновении той или иной потребности в соответствующих нервных клетках возникает волна возбуждения.

Эта волна начинает последовательно распространяться по тем системам условнорефлекторных связей, которые связаны с удовлетворением этой потребности. Таким образом обеспечивается отбор такой информации, которая может быть полезна именно для удовлетворения данной потребности.

Предлагаемая нами гипотеза объясняет все изложенные нами экспериментальные данные и может служить основой для проведения новых экспериментов.

ЛИТЕРАТУРА

- [1] Анохин П. К., Физиология и кибернетика, Вопросы философии, № 4, 1957, 142—158.
- [2] Воронин Л. Г., Анализ и синтез сложных раздражителей у высших животных, Медгиз, 1952.
- [3] Воронин Л. Г., Некоторые итоги сравнительно-физиологического исследования высшей нервной деятельности, Изв. АН СССР, сер. биол., № 5, 1954, 122—134.
- [4] Келер В., Исследование интеллекта человекообразных обезьян, М., Изд. Ком. акад., 1930.
- [5] Купалов П. С., Физиологическая организация процессов возбуждения и торможения в коре мозговых полушарий при условнорефлекторной деятельности, Журнал высшей нервной деятельности, т. V, вып. 4, 1955, 463—473.
- [6] Напалков А. В., Цепи двигательных условных рефлексов у голубей, Семнадцатое совещание по проблемам высшей нервной деятельности, Тезисы докладов, 1956, 80—81.
- [7] Напалков А. В., Физиологические механизмы, лежащие в основе формирования цепей двигательных условных рефлексов, Научные доклады высшей школы. Биологические науки, т. 1, № 2, 1958, 66—73.
- [8] Павлов И. П., Полное собрание сочинений, т. 3, кн. 1, АН СССР, 1951.
- [9] Полетаев И. А., Сигнал, М., «Советское радио», 1958.
- [10] Рокотова Н. А., Цепные двигательные условные рефлексy у собак, Журнал высшей нервной деятельности, 4, вып. 6, 1954, 833—841.
- [11] Соболев С. Л., Китов А. И., Ляпунов А. А., Основные черты кибернетики, Вопросы философии, № 4, 1955, 136—148.
- [12] Торндайк Е., Процесс обучения у человека, М., Медгиз, М—25.
- [13] Уотсон Д., Психология как наука о поведении, Одесса, Гос. изд. Украины, 1926.
- [14] Ухтомский А. А., Физиология нервной системы, вып. 3, кн. 1, М., Медгиз, 1952.

Поступило в редакцию 17 XII 1957

VII. ВОПРОСЫ МАТЕМАТИЧЕСКОЙ ЛИНГВИСТИКИ

НЕКОТОРЫЕ ДАННЫЕ О РАСПРЕДЕЛЕНИИ ФОРМ МНОГООСНОВНЫХ ГЛАГОЛОВ В СВЯЗИ С ПРОБЛЕМОЙ СОСТАВЛЕНИЯ СЛОВАРЯ ОСНОВ ДЛЯ МАШИННОГО ПЕРЕВОДА

Р. М. ФРУМКИНА

(МОСКВА)

ВВЕДЕНИЕ

При составлении словаря машинного перевода *a priori* мыслимы следующие два подхода.

1. В словаре записываются слова (или части слов) в том виде, в каком они встречаются в тексте, так что при идентификации слов текста с одной или несколькими единицами словаря никогда не возникает необходимости предварительного преобразования этих единиц. Тем самым операции анализа и синтеза слов с помощью словаря сводятся к операциям разбиения и составления.

Так, если в словаре записаны отдельные слова и морфемы, то в тексте слово «поискать» будет идентифицироваться в словаре со словом «поиск» и морфемой «ать», но слово «разыскать» не может по тому же принципу связываться со словом «розыск» и морфемой «ать».

Другими словами, при наличии чередований в единицах словаря каждый элемент с чередованием приходится рассматривать как самостоятельную единицу.

Такой подход, упрощая работу со словарем, приводит к значительному увеличению его объема. Примером словаря этого типа с отдельными словами в качестве единиц может служить автоматический словарь А. Эттингера [4]; в основном так же, но с другим выбором единиц словаря, построен словарь во французско-русском алгоритме О. С. Кулагиной и И. А. Мельчука [1].

2. Второй подход допускает преобразование единиц словаря и текста по некоторой программе в процессе анализа и синтеза. Такое преобразование позволяет, с одной стороны, существенно сократить объем словаря за счет отождествления взаимно преобразуемых по принятой программе единиц словаря, а с другой стороны, дает сравнительно небольшой проигрыш во времени по сравнению с первым подходом, если в словарь вносятся преимущественно часто встречающиеся в тексте единицы.

Конкретизируя такой подход, можно предложить строить словарь следующим образом.

В качестве единиц словаря берутся как основы слов и морфем, так и целиком некоторые слова. Среди разнообразия основ (морфем), возникающего в каждом языке за счет чередований, выделяется группа наиболее частых, а остальные основы (морфемы) получают из выделенных преобразованием по некоторой программе.

Окончательное решение вопроса о том, какой подход целесообразнее и какие основы при втором подходе считать редкими, зависит от многих моментов: объема машинной памяти, общей схемы перевода и прочих факторов, которые мы здесь не рассматриваем, а также от статистических закономерностей, которые обнаруживаются в распределении различных типов основ (или других частей слова, в которых имеет место чередование).

Статистические данные о распределении основ с чередованиями дают исходный материал для определения того, насколько может увеличиться объем словаря и насколько усложнится программа.

В настоящей работе с целью установления некоторых закономерностей такого рода анализируется частотность глагольных*) основ с чередованиями в математическом тексте на испанском языке. Испанский язык обладает большим разнообразием типов глагольных основ, и, хотя полученные выводы носят предварительный характер, основные принципы классификации основ могут быть использованы и для других языков, где имеются чередования в основах.

Учитывая недостаточную разработанность терминологии, укажем, в каком смысле употребляются ниже некоторые термины, относящиеся к объектам статистического обследования.

1. Глаголом будем считать часть речи, определенную так в традиционной грамматике, плюс ту его форму, которая в грамматике испанского языка называется *gerundio*—герундий (до некоторой степени соответствует русскому деепричастию). Причастие—*participio pasivo*—в понятие глагола не включается, ибо при составлении алгоритма его удобнее рассматривать как прилагательное.

2. Под формой слова мы будем понимать конкретный вид, который принимает данное слово в тексте. Неизменяемые слова, например наречия, имеют одну форму; изменяемые слова, например глаголы, имеют несколько форм.

3. Под основой глагола будем понимать часть глагола, выделяемую согласно следующим правилам:

а) Основа инфинитива (инфинитив понимается в том смысле, как он определен в традиционной грамматике) выделяется для испанского глагола путем отбрасывания от формы инфинитива двух букв справа: окончания инфинитива и тематической гласной, например,

$$\text{инфинитивы} \left\{ \begin{array}{l} \text{contar} \\ \text{leer} \\ \text{servir} \end{array} \right. \quad \begin{array}{c} \text{соответствующие} \\ \text{основы} \end{array} \left\{ \begin{array}{l} \text{cont-} \\ \text{le-} \\ \text{serv-} \end{array} \right.$$

б) Основы других глагольных форм, получаемых при изменении глагола по временам, наклонениям, лицам и числам, выделяются путем отбрасывания личных окончаний. Инвентарь личных окончаний известен из традиционной грамматики.

Полученные таким образом основы могут быть тождественны с основой инфинитива— тогда данный глагол имеет одну основу, или отличаться от нее— тогда данный глагол имеет более чем одну основу.

4. Глаголы, имеющие одну основу, будем называть *регулярными*, например «читать», «означать». Глаголы, имеющие более чем одну основу, будем называть *сингулярными*, например «брать» («бер-» и «бр-»), «переводить» («перевод-» и «перевож-»).

*) Глагол выбран потому, что, давая наиболее разнообразные чередования в основах, он составляет около 12—14% текста, являясь второй по распространенности частью речи.

5. Те формы сингулярных глаголов, основа которых тождественна основе инфинитива, назовем нормальными (например, «брал», «брала», «переводите», «переводил»). Формы сингулярных глаголов, основа которых отличается от основы инфинитива, будем называть *отклоняющимися* (например, «берем», «перевожу»).

6. Наклонения, отличные от изъявительного: сослагательное, условное, а также герундий, условимся рассматривать как отдельные времена глагола наряду с временами изъявительного наклонения.

Сформулируем теперь наши задачи:

1) Каково соотношение в тексте между регулярными и сингулярными глаголами?

2) Каково распределение типов чередований в основах (типов основ) и, в частности, каково распределение форм сингулярных глаголов? Преобладают ли при этом нормальные формы (т. е. с основой инфинитива) или отклоняющиеся (т. е. с другими основами)?

При втором подходе к построению словаря ответ на эти вопросы приобретает первостепенное значение, поскольку позволяет в сочетании с другими факторами (такими, например, как возможности машины) решать, какие основы считать частыми.

3) Намного ли увеличится объем словаря, если в него будут заноситься наряду с нормальными все остальные варианты основ сингулярных глаголов?

Ответ на этот вопрос позволит выяснить максимальные возможности сокращения словаря при втором подходе его построения.

§ 1. Выделение и классификация объектов подсчета

Как отмечалось выше, изменение формы глагола может выражаться в замене личных окончаний и в чередованиях в основе. Это позволяет рассматривать форму любого глагола как функцию переменных лица, числа, времени*) и типа чередования:

$$\Gamma = f(x, y, z, u),$$

где x —переменная лица, y —числа, z —времени и u —типа чередования.

Поскольку каждая переменная принимает конечное число значений, их можно перенумеровать, причем для испанского языка оказывается заведомо достаточным придавать переменной времени z 10 значений—от 0 до 9, а переменной типа чередования u —100 значений от 00 до 99, где 00 обозначает отсутствие чередования, т. е. нормальную основу. В обследованных текстах обнаружено 7 времен глагола и 37 типов чередований (см. приложение).

Тем самым каждая форма глагола вполне характеризуется набором значений указанных переменных, т. е. если известен глагол, то по заданному набору можно построить только одну его форму.

Поскольку нашей задачей является получение сведений о распределении форм нормальных и сингулярных глаголов и о распределении типов чередований в основах сингулярных глаголов, то в дальнейшем нас преимущественно будет интересовать форма как функция типа чередования. Тогда при рассмотрении текста можно ограничиться сопоставлением каждой форме глагола некоторого числа u , характеризующего тип чередования.

Известно, что тип чередования существенным образом зависит от времени глагола, хотя однозначно им не определяется. Если бы *a priori* было известно, что в математических текстах вероятность формы в прошедшем

*) Наклонение рассматривается как отдельное время (см. выше, пункт 6).

времени чрезвычайно мала, то можно было бы утверждать, что чрезвычайно мала вероятность чередований определенных типов, ибо они возможны только в прошедшем времени.

В силу этого представляет интерес попутно получить сведения о распределении времен для форм сингулярных глаголов. Это можно сделать, сопоставляя каждой форме Г глагола в тексте не одно число *u*, указывающее на тип чередования, а пару чисел *z, u*: $[Г] \longleftrightarrow \{z, u\}$ (влияние лица и числа на тип чередования не рассматривается). При таком подходе между формой и парой чисел нет взаимно однозначного соответствия: одна и та же пара чисел сопоставляется всем формам глагола, которые в данном времени имеют данный тип чередования в основе

$[tiene] = [vienen] = [siente] = [mienten] = \{z, u\}.$

Будем рассматривать совокупность форм глаголов в тексте как некоторое множество, элементами которого являются формы регулярных и сингулярных глаголов. Исключим из рассмотрения подмножество форм регулярных глаголов. Из подмножества форм сингулярных глаголов исключим подмножество форм вспомогательных глаголов: *ser, estar, haber*, так как большинство их форм образуется от разных одинаково частых основ, которые при обработке удобно считать разными словами.

Каждой форме сингулярного глагола мы сопоставили выше пару чисел: $[Г] \longleftrightarrow \{z, u\}$. Это сопоставление осуществляет отображение множества \mathfrak{A} форм всех сингулярных глаголов на некоторое множество пар чисел $[\mathfrak{A}]$.

Объединение пар чисел с одинаковыми вторыми числами в подмножества $[W_1], [W_2], \dots$ множества $[\mathfrak{A}]$ разбивает множество форм сингулярных глаголов на пересекающиеся подмножества W_1, W_2, \dots . Каждое подмножество W_i объединяет, таким образом, все формы сингулярных глаголов с данным типом чередования в основе. Например, по проводимой ниже классификации, глаголы: *sintió, vino, pidiendo* — попадут в один класс W , для которого $[W] = \{., 03\}$, поскольку для всех типов глаголов характерно чередование в основе $e \rightarrow i$. Подсчитав частотность элементов, входящих в $W_1 \dots W_i$, получим данные о распределении основ с данным типом чередования $\{., u\}$. Для удобства записи составим таблицу вида:

Т а б л и ц а 1

Значение переменной типа чередования <i>u</i>	Значение переменной времени <i>z</i>	0	1	2	3	...
0 0						
0 1						
0 2						
0 3						
...						

В одну клетку попадут только формы, которым сопоставлены одинаковые пары чисел $\{z, u\}$ множества $[X]$. Например, в клетку (1,03) попадут формы *sintiendo*, *viniendo*, *sirviendo*; в клетку (4,03)—формы *vino*, *sinti6*, *sirvi6* и т. д.

В горизонтальных графах расположатся формы, которым сопоставлены одинаковые вторые числа $\{., u\}$, так как каждый из классов $W_{01}, W_{02}, \dots, W_{99}$ займет горизонтальную графу (строку).

Таким образом, формы *sintiendo*, *vino*, *sinti6* и т. д. объединяются по общему типу чередования. Разделив число элементов в строке на общее число элементов в таблице, получаем частотность данного типа чередования, что является ответом на второй вопрос.

Некоторые клетки останутся пустыми в силу того, что определенные типы чередований имеют место не во всех временах; так, тип 06 возможен только в будущем времени (клетка 2 графы 06) и в условном наклонении (клетка 6 графы 06), так что в графе 06 клетки, кроме 2 и 6, останутся пустыми.

По данным в вертикальных столбцах можно судить также о частоте употребления глагольных времен для сингулярных глаголов.

§ 2. Объем выборки и характер текста

При статистических расчетах предполагалось, что текст, из которого производится выборка, однороден, и что частота x_n какого-либо признака глаголов A , полученная с помощью случайной выборки объема n , имеет нормальное распределение с параметрами $a = Mx_n = pn$ (p —вероятность признака A), и $\sigma = \sqrt{x_n(1-x_n)/n}$.

Это предположение позволило оценить вероятность p с помощью наблюдаемой частоты x_n . В качестве уровня значимости всегда бралось 0,99.

Для получения усредненных по языку результатов были взяты тексты работ, относящихся к разным отраслям математики.

Вообще говоря, частотность некоторой формы слова в математическом тексте в первом грубом приближении можно рассматривать как алгебраическую сумму средней частоты p употребления формы слова в математическом языке, поправки $\theta_1 p$ на зависимость от типа текста (для нашей работы—это текст по высшей алгебре, топологии, дифференциальному и интегральному исчислению и т. д.) и поправки $\theta_2 p$ на зависимость от стиля автора*), т. е.

$$\bar{p} = p(1 + \theta_1 + \theta_2),$$

где \bar{p} —частота данной формы в рассматриваемом тексте; p —среднее значение частоты данной формы в математическом языке вообще, которое может быть получено при обследовании достаточно большого числа разных текстов; θ_1 —тематический коэффициент; θ_2 —авторский коэффициент.

При такой записи θ_1 и θ_2 могут быть как положительными, так и отрицательными числами. Если мы будем обследовать некоторый текст по данной отрасли математики, принадлежащий перу одного автора, то θ_1 и θ_2 будут постоянными величинами и \bar{p} будет значительно отличаться от p . Если же мы будем брать тексты разных авторов по разным отраслям математики, то θ_1 и θ_2 , принимая то положительное, то отрицательное значение, при возрастании числа разнообразных текстов будут компенсировать

*) Если стать на более общую точку зрения и рассматривать вообще тексты данного языка, то там будет наблюдаться аналогичная картина с той только разницей, что число поправок увеличится.

друг друга. Исходя из этих соображений, для обследования были выбраны тексты статей по разным отраслям математики из журнала *Revista matemática hispano-americana* и текст книги по исчислению бесконечно малых (см. приложение).

§ 3. Результаты статистического обследования и некоторые рекомендации для составления словаря основ

В табл. 2 представлены результаты статистического обследования текста длиной в 75 000 слов. В соответствии с поставленными задачами мы исследовали все более редкие события, так что результаты в строках 2—5, где обследовались относительно более частые события, получались достаточно точными (т. е. относительная ошибка оцениваемой вероятности p $\epsilon/(p-\epsilon) \leq 0,046$), а результаты, записанные в строках 6—9, относящиеся к редким событиям, могут рассматриваться лишь как сугубо предварительные. Для получения более точных результатов следует значительно увеличить объем выборки.

Т а б л и ц а 2

Оценка результатов (уровень значимости—0,99)

Номер испы- тания	Объем выборки n	Вероят- ность p	Абсолют- ное от- клонение ϵ	Относи- тельное откло- нение $\frac{\epsilon}{p-\epsilon}$	Характеристика задачи
1	2	3	4	5	6
1	75000	0,109	0,003	0,029	Частотность глаголов в математическом тексте
2	8200	0,426	0,014	0,034	Частотность сингулярных глаголов(все формы) по отношению к общему числу глаголов
3	3500	0,491	0,021	0,046	Частотность отклоняющихся форм сингулярных глаголов по отношению к общему количеству их форм
4	1718	0,766	0,025	0,035	Частотность форм с чередованием типа 01, 02, 03 и 04 по отношению к количеству отклоняющихся форм
5	3500	0,800	0,017	0,022	Частотность настоящего времени (для сингулярных глаголов)
6	3500	0,092	0,012	0,158	Частотность герундия (для сингулярных глаголов)
7	3500	0,063	0,010	0,202	Частотность сослагательного наклонения (для сингулярных глаголов)
8	3500	0,044	0,009	0,256	Частотность будущего времени (для сингулярных глаголов)
9	3500	0,010	0,004	0,711	Частотность <i>imperfecto de subjuntivo</i>

Из табл. 2 видно, что частотность сингулярных глаголов в тексте весьма значительна: на их долю приходится немногим менее половины—42,6%—глагольных форм текста. Однако среди различных форм сингулярных глаголов текста почти половину—49,1%—составляют нормальные

формы, т. е. формы с основой инфинитива. В тексте решительно преобладает настоящее время (данные о распределении времени получены только для сингулярных глаголов, но их можно распространить на весь текст, опираясь на обычные языковые закономерности).

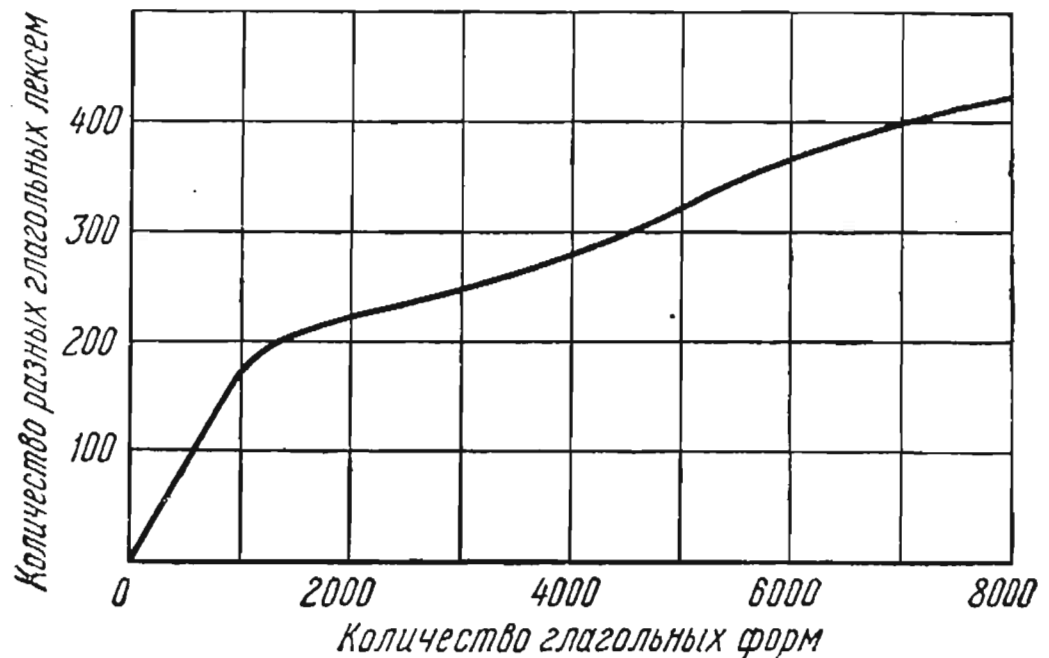
В связи с преобладанием настоящего времени среди отклоняющихся форм преобладают формы с чередованием типа 01, 02, 03 и 04. На остальные 33 типа чередования приходится всего 23,4% отклоняющихся форм.

Отметим, что все прошедшие времена встречаются редко, в частности почти полностью отсутствует *Presente perfecto* (в таблице оно не учтено).

Полученные статистические данные позволяют предложить некоторые рекомендации для составления словаря основ глаголов. Для этого вначале выясним, насколько увеличится объемом глагольной части словаря основ при записи в нее всех вариантов основ.

В нашем тексте 8200 глагольных форм представляют 402 разных глагола. При этом появление новых глаголов описывается графиком, приведенным на рисунке.

Прирост глаголов от седьмой до восьмой тысячи глагольных форм составляет всего 3,07% от количества разных глаголов, появившихся на первые семь тысяч глагольных форм. Поскольку с увеличением длины текста кривая приближается к горизонтали, можно принять в первом грубом приближении, что 402 глагола—это и есть глагольная часть словаря математического текста. Из соотношения между количеством глаголов



и количеством основ (табл. 3) ясно, что при записи всех вариантов основ объем словаря возрастает примерно в 1,6 раза.

Каковы же пути уменьшения объема словаря?

В нашем конкретном случае глагольная часть словаря основ включает 115 глаголов с двумя основами. Из них более половины—64—сингулярные глаголы с редким типом чередования в основе (это чередования с номерами

от 05 до 37), причем в тексте многие из этих глаголов встречаются только в виде нормальных форм (таких форм около 300, а отклоняющихся форм—30). Следовательно, если это соотношение не случайно*), можно уменьшить объем словаря основ на 64 единицы путем записи в него только нормальной основы, как более частой для каждого из указанных 64 глаголов.

Остальные 51 глагол с двумя основами и часть глаголов с тремя основами имеют основы с чередованием по распространенным типам—от 01 до 04. При этом обнаружилось, что в тексте эти глаголы встречаются преимущественно в виде отклоняющихся форм (табл. 4).

*) Выше отмечалось, что для ряда редких событий была взята недостаточная выборка.

Поэтому в данном случае следует записать в словарь отклоняющуюся основу, а нормальную строить от отклоняющейся. Это позволит уменьшить объем словаря основ еще на 70—80 единиц.

Т а б л и ц а 4

Тип чередования	Количество отклоняющихся форм	Количество нормальных форм
01	655	296
02	494	332
03	81	29
04	87	21

В общем случае можно с достаточной уверенностью утверждать, что для многоосновных глаголов (а также других частей речи) формы от различных основ будут резко различаться по частотности. Поэтому предлагается при составлении словаря основ отказаться от неперменной записи в него традиционной «начальной формы»—инфинитива, именительного падежа и т. п.,—а на основе статистического анализа вводить в словарь только наиболее частую основу (основы).

Автор искренне признателен канд. физ.-матем. наук В. М. Золотареву за ценные советы и помощь в проведении данной работы.

ПРИЛОЖЕНИЕ

I. Условные обозначения времен испанского глагола

0 — настоящее время	(изъявительное наклонение)
1 — герундий	
2 — будущее время	
3 — <i>presente de subjuntivo</i>	(настоящее время, сослагательное наклонение)
4 — <i>preferito perfecto simple</i>	(прошедшее время)
5 — <i>imperfecto</i>	(имперфект изъявительного наклонения)
6 — <i>condicional</i>	(условное наклонение)
7 — <i>imperfecto de subjuntivo</i>	(имперфект сослагательного наклонения)

II. Условные обозначения типов чередований в основах испанского языка

« / » знак чередования	18	<i>c / z</i>
« — » нуль буквы	19	<i>e / ie; — / u</i>
00 — отсутствие изменений	20	<i>o / ue; — / u</i>
01 <i>e / ie</i>	21	<i>ec / ig</i>
02 <i>o / ue</i>	22	<i>ir / va</i>
03 <i>e / i</i>	23	<i>ab / ep</i>
04 <i>— / y</i>	24	<i>e / i; u / —</i>
05 <i>— / g</i>	25	<i>e / i; g / j</i>
06 <i>e / d</i>	26	<i>ce / —</i>
07 <i>e / —</i>	27	<i>ec / —</i>
08 <i>o / u</i>	28	<i>c / j</i>
09 <i>c / zc</i>	29	<i>en / uv</i>
10 <i>z / c</i>	30	<i>on / us</i>
11 <i>c / qu</i>	31	<i>— / i</i>
12 <i>c / g</i>	32	<i>ab / up</i>
13 <i>— / ig</i>	33	<i>ec / ij</i>
14 <i>— / u</i>	34	<i>ac / iz</i>
15 <i>g / j</i>	35	<i>— / j</i>
16 <i>e / ie; z / c</i>	36	<i>er / is</i>
17 <i>— / e</i>	37	<i>— / i</i>

III. Список обследованных текстов

I. Revista matemática hispano-americana. Madrid. Ser. 4. 1947, t. 7.
№ 1. 1. Carrasco L. E. La derivada n-sima de una función polígena, p. 3—10.
2. Dwinas S. Una deducción de la ley de errores de Laplace — Gauss, p. 12—18.
3. Rodeja E. G. Generalización del teorema de Pascal a los hiperespacion, p. 23—45.

- № 2. 1. Casulleras J. Sobre la representación de un E_n complejo mediante un E_n real, p. 51—56.
 2. Cuesta N. Ordenación densa perfectamente escalonada, p. 57—71.
 3. Gaeta F. Sobre las superficies y las variedades del S_7 aritméticamente normales, p. 72—82.
- № 3. 1. Feller W. Los teoremas límites fundamentales en la teoría de la probabilidad, p. 95—126.
- II. Revista matemática hispano-americana. Madrid. Ser. 4. 1953, t. 13.
 № 1. 1. Doetsch G. Desarrollos asintóticos y transformación de Laplace, p. 5—60.
 2. Azpeitia A — G. Sobre una generalización de las integrales de Fresnel, p. 61—80.
 3. Belgrano Bremard J. C. Sobre una problema de cálculo funcional planteado por la nomografía, p. 87—89.
- III. Phillips. H. B. Elementos de cálculo infinitesimal. Mexico.

ЛИТЕРАТУРА

- [1] Кулагина О. С. и Мельчук И. А., Машинный перевод с французского на русский, Вопросы языкознания, М., 1956, № 5, стр. 111—121.
 [2] Молошная Т. Н., Пурто В. А., Ревзин В. Ю., Некоторые лингвистические вопросы машинного перевода, Вопросы языкознания, М., 1957, № 1, стр. 107—113.
 [3] V. García Hoz, Vocabulario usual, vocabulario común y vocabulario fundamental, Madrid, Inst. «San Jose de Calasanz», 1953.
 [4] Эттингер А. Г., Проект автоматического русско-английского технического словаря, Машинный перевод (сборник статей), перевод с английского, М., ИЛ, 1957, стр. 76—100.

Поступило в редакцию 18 VI 1958.

О МАШИННОМ ПЕРЕВОДЕ С ФРАНЦУЗСКОГО ЯЗЫКА НА РУССКИЙ. II АЛГОРИТМ ПЕРЕВОДА С ФРАНЦУЗСКОГО ЯЗЫКА НА РУССКИЙ

О. С. КУЛАГИНА

(МОСКВА)

Алгоритм перевода с французского языка на русский составлялся с целью перевода текстов математического характера.

Правила перевода рассчитаны на перевод текста отдельными фразами, т. е. при вводе в машину каждой фразы никаких сведений о переводе предыдущей фразы в памяти машины не остается. Из-за этого, правда, для некоторых местоимений не удастся выяснить, какие слова они заменяют (эти слова могут быть в другой фразе), и поэтому не удастся установить род этих местоимений. В таких случаях приходится выдавать несколько вариантов. Для выбора нужных переводов многозначных слов контекст из одной фразы не всегда оказывается достаточным. Однако в настоящее время приходится ограничиваться переводом отдельных фраз, так как совершенно неясно, какие сведения о переводе предыдущих фраз (и сколько) нужно запомнить для перевода последующих. Этот вопрос требует отдельной разработки.

Предполагается, что, как правило, одно слово французской фразы переводится одним словом русской фразы, однако возможны исключения в ту и другую сторону. Случаи, когда несколько французских слов переводятся одним русским словом, либо включены в словарь оборотов, либо оговорены в правилах (например, сложные формы глаголов). Французское слово может переводиться несколькими русскими словами в следующих случаях. Во-первых, когда французское слово многозначное и нет правил выбора одного из значений (например, *aire*—«площадь», «область»). Во-вторых, когда русские слова образуют неразделимое сочетание, в котором только последнее слово может изменяться (например, *car*—«так как»; *biunivoque*—«взаимно однозначный»), причем для машины такие сочетания представляют как бы одно слово, написанное с пропусками. В-третьих, когда французское слово не имеет русского эквивалента и переводится несколькими русскими словами, такие случаи специальным образом закодированы в словаре (например, *annuler*—«обращать в ноль»).

Перевод фразы ведется в следующем порядке. Слова переводимой фразы кодируются побуквенно, математические формулы помечаются специальным образом, кодируются также знаки препинания. После этого закодированная фраза вводится в машину и начинается ее обработка, которая ведется в следующем порядке:

- 1) поиск слов в словаре,
- 2) обработка оборотов,
- 3) различение омонимов,
- 4) анализ французской фразы,
- 5) синтез русской фразы.

Полученная русская фраза выдается машиной также в закодированной форме. Правила различения омонимов и анализа даются в приложении.

Для автоматизации процесса кодировки фразы, которая сейчас производится человеком, требуется разработать специальные устройства, способные читать текст с книги и кодировать его. Аналогично для декодирования полученного результата нужны специальные устройства. Вопросы автоматизации ввода фраз в машину и расшифровки полученного результата нами не рассматривались, так как это вопросы скорее инженерного, чем математического характера.

Рассмотрим теперь подробнее каждый из этапов работы алгоритма перевода.

1. Поиск слов в словаре

Под нахождением слова в словаре подразумевается нахождение в словаре максимальной по числу букв основы, целиком уместящейся в данном слове, а также выбор из словаря и запоминание для дальнейшей работы словарной статьи, находящейся при данной основе.

Благодаря принятому способу размещения основ в словаре нахождение максимальной по числу букв основы, уместящейся в некотором слове, сводится к нахождению первой такой основы.

Если основа найдена, то из словаря выбирается и запоминается для дальнейшей работы стоящая при ней информация. Часть слова, остающаяся после выделения из него основы, называется окончанием; окончания также запоминаются для дальнейшей обработки. Если основа не найдена, то слово остается непереведенным и будет без всяких изменений поставлено в переводящую фразу.

После того, как поиск в словаре окончен, слова переводимой фразы больше не нужны, они никогда не используются. Вся дальнейшая работа ведется только с французской информацией слов и с отсеченными от слов окончаниями, причем везде в дальнейшем мы отождествляем понятия слова и заменяющей его в машине информации, т. е., говоря «слово» или «основа», подразумеваем соответствующую им информацию, а говоря «фраза», подразумеваем последовательность словарных информации, соответствующих словам этой фразы.

2. Обработка оборотов

После окончания поиска слов в словаре переходим к обработке оборотов. Задача этой обработки—найти все имеющиеся во фразе обороты и заменить информацию к входящим в них словам, взятую из словаря основ, информацией к обороту из словаря оборотов.

Для этого просматривают фразу, отыскивая в ней слова с указанием об обороте. Найдя такое слово, по указанию об обороте обращаются к первому из оборотов, имеющих это слово в качестве основного [1]. Оборот как бы накладывают на фразу, совмещая основные слова. Если оборот совпадает с куском фразы, т. е. во фразе присутствуют и стоят в нужном порядке все слова оборота, то считаем, что оборот найден. Информацию к составляющим его словам заменяют информацией к обороту. При этом могут представиться различные случаи. Иногда оборот состоит из нескольких слов, но переводится одним словом (например, *par exemple*—«например», *à présent*—«теперь»). В других случаях заменяющих слов также несколько, причем обычно число заменяющих слов не больше числа слов, входящих в оборот. Исключением является случай, когда

одно французское слово заменяется несколькими русскими, которые не могут быть ничем отделены друг от друга и из которых только последнее может быть изменяемым. Такое сочетание, как уже говорилось выше, для машины есть как бы одно слово, написанное с пропуском (см. примеры ниже).

При обороте ставится специальное указание относительно того, какое из исходных слов каким именно следует заменить. Например, оборот *faire (adv) correspondre* переводится «ставить (наречие) в соответствие», причем глагол «ставить» заносится на место *faire*, а неизменное сочетание «в соответствии» — на место *correspondre*. Если оборот не совпал с куском фразы, то в случае целого оборота [1] переходим к следующему обороту с тем же основным словом. В случае нецелого оборота проверяем, не принадлежат ли слова фразы, не совпадающие со словами оборота, к числу пропускаемых в этом обороте слов. Если принадлежат, то пропускаем их и сравниваем слова оборота со следующими за пропускаемыми словами («следующий» при движении от основного слова в обе стороны). Например, нецелый оборот *à... près* имеет основное слово *près*. Влево от него должно стоять *à*, которое может отделяться другими частями речи. Оборот переводится «с точностью до», перевод ставится на место *à*. Слова оборота ищут описанным способом до тех пор, пока не получится совпадение или не встретится слово, которое в обороте пропускать нельзя. В этом последнем случае переходят к следующему обороту с тем же основным словом. Когда все обороты с данным основным словом просмотрены, переходят к следующему слову фразы, и так до тех пор, пока вся фраза не будет пройдена до конца.

3. Различение омонимов

После обработки оборотов начинается различение омонимов. Мы будем в дальнейшем понимать под омонимией первого типа случаи, когда одно и то же слово может выступать в роли различных частей речи и, следовательно, иметь в словаре несколько словарных информации; под омонимией второго типа — случаи, когда слово выступает в роли одной и той же части речи, но имеет несколько переводов (такие слова мы будем еще называть многозначными).

Слова с омонимами первого типа мы разбираем обязательно, т. е. у них после различения остается только одна словарная статья. Многозначные слова разбираются не всегда, т. е. у них может остаться несколько переводов. Это связано с тем, что наличие во фразе неразобранной омонимии первого типа влияет на анализ всей переводимой фразы, тогда как наличие многозначного слова не мешает формальному анализу остальных слов фразы. С другой стороны, вопрос о выборе одного из омонимов первого типа обычно можно решить, не прибегая к смыслу фразы, а выбор нужного перевода многозначного слова часто определяется только смыслом и не может быть сделан на основе формальных признаков.

Поэтому в этом первом варианте алгоритма перевода, который является опытным, предварительным, мы иногда оставляем для некоторых многозначных слов несколько переводов, предоставляя читателю выбор нужного из них. В дальнейшем и такие случаи можно будет различать. По-видимому, для этого придется установить классы сочетаемых друг с другом значений слов (или понятий) и определять значение многозначных слов в зависимости от наличия или отсутствия во фразе слов, допускающих сочетание с тем или иным значением разбираемого слова.

К различению омонимов мы условно относим также следующие операции.

Во-первых, установление для каждого прилагательного, не является ли оно субстантивированным в данной фразе, что сводится к отысканию для прилагательного того существительного, к которому оно относится, или установлению отсутствия такого существительного. В последнем случае прилагательное должно при дальнейшем анализе фразы расцениваться как существительное, поэтому его информацию заменяют информацией для существительного. Иначе говоря, все прилагательные, которые могут быть употреблены самостоятельно без соответствующего существительного, хотя и имеют одну словарную статью прилагательного, но рассматриваются как потенциальная омонимичная пара: прилагательное—существительное.

Во-вторых, установление для математических формул, употреблены ли они как самостоятельные слова или после существительных. Формула, употребленная самостоятельно, считается при анализе фразы существительным и ей приписывается информация существительного. Формула, употребленная после существительного, условно получает информацию наречия. Например, $f(x)$ в сочетании *considérons la fonction $f(x)$...* получает информацию наречия, а в сочетании *considérons $f(x)$...*—информацию существительного.

В-третьих, установление для глаголов, которые могут быть как переходными, так и непереходными, какой именно случай имеет место в данной фразе.

Следует заметить, что основа в словаре имеет пометку об омонимии (и соответственно несколько информаций) и в том случае, когда она является основой некоторой словоформы, которая может относиться к нескольким классам слов (частям речи). При этом от такой основы могут образовываться и другие словоформы, которые не имеют омонимов, однако при разборе омонимии приходится рассматривать все слова, получившие по несколько информаций. Поэтому среди правил различения омонимов некоторые относятся к словам, не имеющим омонимов, но образованным от основ, которые имеют несколько информаций.

Например, основа *moyen* может быть основой существительного («способ») и прилагательного («средний»), причем совпадают формы ед. ч. существительного и м. р. ед. ч. прилагательного и формы мн. ч. существительного и м. р. мн. ч. прилагательного. Если в тексте была форма *moienne*, то хотя она и не имеет омонима—существительного, но так как для нее найдена основа с двумя информациями, то приходится разбирать этот случай тоже как случай омонимии, с тем чтобы выбрать нужную информацию.

Отметим, что среди случаев омонимии, имеющих в словаре для машинного перевода, имеется несколько «искусственных», возникших только за счет того, что в словаре содержатся не слова целиком, а их основы. Так, например, для глагола *aborder* отсечена основа *abord-*, совпадающая с существительным *abord*, тогда как ни одна из форм глагола с существительным не совпадает; аналогично для глагола *pouvoir* для будущего времени есть основа *pour-*, совпадающая с предлогом *pour*, а для глагола *paraître* есть в словаре основа *par-*, совпадающая с предлогом *par*. В нашем словаре 8 случаев «искусственной» омонимии.

Разбор этих случаев не представляет труда.

Всего в нашем словаре насчитывается 189 различных случаев омонимии. Из них 94 случая омонимии первого типа, 83 случая второго типа и 12 случаев, принадлежащих одновременно обоим типам, т. е. таких, которые могут представлять несколько различных частей речи и при этом одна или несколько из них имеют больше одного перевода. Среди различных случаев омонимии есть случаи массовые, т. е. имеющие много

представителей, и есть такие, к которым относятся только одно-два конкретных слова.

Правила различения омонимов основаны на рассмотрении окончания обрабатываемого слова или окружения данного слова во фразе.

На основе анализа окончания правила различения работают в тех случаях, когда основа может являться основой различных частей речи в зависимости от того, присоединено ли к ней окончание или нет, и если присоединено, то какое именно. Однако только немногие случаи (в основном «искусственные» омонимы, примеры которых приведены выше) могут быть разобраны таким способом до конца.

В большинстве случаев для полного разбора омонимии оказывается недостаточно анализа окончания и приходится прибегать к рассмотрению контекста. Так же поступают в случаях, когда у разбираемого слова не бывает окончания. Именно, для каждой части речи, которую может представлять разбираемое слово, установлено на основе анализа большого числа фраз, содержащих данное слово, какое именно окружение для нее характерно. В зависимости от того, каково окружение разбираемого слова в переводимой фразе, устанавливают нужную часть речи и оставляют соответствующую информацию, а остальные стирают.

Правила, устанавливающие перевод для многозначных слов, также анализируют окружающие слова. Отличие состоит в том, что результатом их работы является не выбор нужной информации, а выработка номера перевода. В случае, если различные переводы многозначного слова собраны под одним номером перевода [1], вырабатывается частный номер перевода, в других случаях меняется основной номер перевода в информации.

В результате работы правил различения омонимов для каждого слова фразы остается одна словарная статья, или, иначе говоря, одна французская информация.

4. Анализ

После различения омонимов начинается анализ переводимой фразы с целью установления для каждого французского слова формы и места во фразе переводящего его русского слова.

Анализирующие правила распределены по группам таким образом, что правила каждой группы касаются обработки одной определенной части речи. Группы правил работают последовательно друг за другом и обрабатывают части речи в следующем порядке: глагол, предлог, существительное, местоимение, причастие, не вошедшее в сложную форму глагола. Что касается обработки прилагательных, то она разделена на две части. Первая часть делается совместно с различением омонимов: устанавливая, не субстантивировано ли данное прилагательное, в случае, если находят существительное, к которому это прилагательное относится, «привязывают» к нему прилагательное, т. е. заносят в информацию прилагательного направление и величину расстояния до найденного существительного. Во второй части после работы всех остальных анализирующих правил, т. е. после обработки причастий, прилагательные согласуются с теми существительными, к которым они привязаны, т. е. им просто переносятся данные, выработанные для существительного.

Указанная последовательность обработки частей речи не случайна. Начинают с глагола потому, что глагол во французском языке сохранил наибольшее (по сравнению с другими частями речи) число форм, различающихся графически, как это и нужно машине. Благодаря этому данные о глаголе (его число, лицо, время, наклонение) в большинстве слу-

чаев можно установить по его форме, точнее, по его окончанию, рассматривая этот глагол вне контекста. Перевод предлога зависит от перевода управляющего предлогом слова, которое чаще всего является глаголом. Для существительного и местоимения падеж часто определяется предшествующим предлогом, т. е. нужно, чтобы предлоги к моменту обработки существительных и местоимений были переведены. Причастия, употребленные в роли прилагательных, и прилагательные согласуются с существительными, которые, следовательно, должны быть обработаны раньше.

Остальные части речи, такие, как союзы, наречия и другие неизменяемые слова, никакой специальной обработки не требуют.

Опишем в общих чертах работу анализирующих правил. Обработка глагола разбивается на две части: предварительную и окончательную. Предварительная обработка состоит в том, что по окончанию глагола устанавливаются некоторые предварительные данные о нем, которые еще нуждаются в уточнении. Для этого по номеру группы глагола обращаются к нужной таблице окончаний глаголов и там отыскивают анализируемое окончание. В таблице каждое окончание снабжено информацией о лице, числе и времени глагола, причем эта информация касается русского глагола, а не французского. Иначе говоря, каждой форме французского глагола, характеризующейся определенным окончанием, поставлена в соответствие некоторая форма русского глагола, и информация о ней состоит при французском окончании. При установлении этого соответствия мы считали, что лицо и число русского глагола такие же, как лицо и число французского (исключением является русское прошедшее время, где нужно устанавливать род, о роде см. стр. 213). Предварительное соответствие времен такое: *présent de l'indicatif* переводится настоящим временем, *futur de l'indicatif*—будущим временем совершенного вида, *imparfait de l'indicatif*—прошедшим временем несовершенного вида, все остальные формы *passé de l'indicatif* переводятся прошедшим временем совершенного вида. *Subjonctif présent* переводится прошедшим временем несовершенного вида + «бы» (русское условное наклонение). *Conditionnel présent* мы переводим, вообще говоря, будущим временем, что достаточно хорошо оправдывается в математических текстах. Форме *infinitif* ставится в соответствие неопределенная форма совершенного вида. Некоторые из французских времен (*passé simple indicatif*, *imparfait* и *plus-que-parfait subjonctif*) не принимались во внимание совсем, т. е. их окончания не включены в таблицы и они не учтены в правилах, так как они практически не встречаются в математических текстах.

Для окончаний *infinitif* в таблице записано только указание на неопределенную форму. Для *participe présent*—число и род французского *participe présent* и информация деепричастия настоящего времени, вместо информации о времени и виде.

Для окончаний *participe passé* также указывается число и род французского *participe passé*, а вместо информации о времени ставится указание о том, что это *participe passé* (ввиду отсутствия предварительного соответствия для этой формы глагола).

Те формы отдельных глаголов, которые присутствуют в словаре как самостоятельные слова (*est*, *sont*, *a*, *fallait*), не нуждаются в установлении предварительных данных по окончаниям, их предварительные данные занесены в их словарную информацию.

Предварительные данные нуждаются в уточнении из-за того, что указанное выше соответствие времен не всегда оказывается правильным из-за различия правил употребления времен во французском и в русском языках (например, во французском языке в придаточном предложении, присоединенном союзом *si*, будущее время заменяется настоящим, а в рус-

ском языке нужно поставить будущее, если в главном предложении было будущее). Кроме того, картина осложняется тем, что многие окончания омонимичны. Например, окончания *présent indicatif* и *présent subjunctif* у всех глаголов первой группы (и некоторых других) совпадают для всех форм, кроме первого лица множественного числа. В таблице окончаний им поставлена в соответствие информация настоящего времени, а уточняющая часть правил по контексту выясняет, было ли действительно *présent indicatif* или же это *présent subjunctif*.

Уточнения выполняются второй частью правил обработки глагола. Эта вторая часть разбита на разделы в зависимости от того, какое время было получено по предварительным данным.

В правилах обработки глагола обрабатываются те *participe passé*, которые входят в аналитические формы *).

Остальные *participe passé* так же, как и *participe présent*, не переводимые деепричастием, обрабатываются позже специальными правилами для причастий.

Уточняющая часть правил обрабатывает и такие формы французского глагола, как *passé immédiat* и *futur immédiat*.

Несмотря на предусмотренные уточнения, мы получаем в русском языке несколько упрощенную картину времен (например, будущее время и неопределенная форма всегда совершенного вида), однако, как правило, такие огрубления не мешают правильной передаче смысла переводимого текста.

Если после уточнений получили для глагола прошедшее время, то в случае единственного числа надо установить род этого глагола. Род определяется по ближайшему слева существительному или местоимению в именительном падеже. Но так как глаголу нужно записать род русского существительного (или местоимения), а во время анализа французской фразы русская часть словаря, где этот род записан, еще не введена в память машины, то приписывание рода откладывается до того момента, когда после окончания анализа будет введена русская часть словаря. Поэтому при обработке глаголов тем из них, которые получили информацию прошедшего времени, только делается пометка о том, что в дальнейшем им следует приписать род.

После анализа глаголов переходим к анализу предлогов. Из предлогов в обработке нуждаются только неоднозначные. Таких предлогов во французском языке насчитывается семь (*à, dans, de, en, par, pour, sur*). Остальные, во всяком случае в пределах математических текстов, можно переводить однозначно. Неоднозначные предлоги допускают от пяти до пятнадцати различных переводов [1]. Каждый раз приходится устанавливать, какой именно перевод следует употребить.

Большинство из этих предлогов могут стоять либо перед существительным, либо перед глаголом в *infinitif* и в зависимости от этого обрабатываются по-разному.

Сначала обрабатываются предлоги, стоящие перед глаголом в *infinitif*, для них определяется перевод иногда в зависимости от предыдущих слов, иногда в зависимости от последующих. Нередко наличие такого предлога заставляет менять перевод глагола, стоящего за ним, или того глагола, который управляет этим предлогом. Заметим, что здесь и в дальнейшем термин «управление» употребляется в не совсем обычном смысле, именно, мы называем управлением всякий случай воздействия одного слова на другое.

*) Имеются в виду сложные формы французского глагола (*passé composé, plus-que-parfait, futur antérieur* и т. п.), которые переводятся на русский язык одним словом.

Для предлогов, стоящих перед существительными, нужно определить не только перевод, но и падеж, которого требует переводящий предлог. Имеются некоторые частные правила, касающиеся перевода индивидуальных предлогов (например, *à, de*; см. стр. 230). Общее правило обработки предлогов состоит в отыскании того слова, которое управляет в данном случае предлогом, и установлении данных о предлоге по таблице перевода предлогов, номер которой дается предложным кодом управляющего слова. Управляющим словом может быть существительное, прилагательное, наречие, но чаще всего им бывает глагол. Существительное может управлять как предлогом, стоящим перед ним (например, *à condition*), так и предлогом, стоящим после него (например, *théorème sur les fonctions entières*), причем, как легко видеть, это два различных случая управления, в первом предлог определяет падеж того слова, которое им управляет (*condition*), во втором—предлог управляется одним словом (*théorème*), а воздействует на другое (*fonctions*). Прилагательные и наречия управляют только предлогами, идущими после них (например, *égal à, indépendamment de*). Глагол может управлять как предлогом, стоящим позади него, так и предлогом, стоящим перед ним (например, *de ce théorème en peut déduire* и *on peut déduire decethéorème*), причем очевидно, что в случае глагола характер управления не меняется с изменением относительного расположения глагола и предлога. По окончании анализа предлогов начинается анализ существительных.

Для существительного нужно определить число и падеж. Сначала определяется число французского существительного. Почти всегда множественное число можно установить по наличию окончания (*-s, -x* и др.). Исключение составляют существительные, у которых формы единственного и множественного числа совпадают (*pas, cas* и др.). Для определения числа таких существительных приходится отыскивать артикль и устанавливать число по нему, если это возможно. Если и по артиклю установить число не удастся, то ставят единственное (возможно с ошибкой). Число русского существительного мы считаем совпадающим с числом французского, за исключением нескольких индивидуальных слов (например, *mathématiques*—«математика») и случаев, когда перед существительным стоит числительное (например, *21 équations* (pl.) переводится «21 уравнение» (ед. ч.)). После определения числа определяют падеж русского существительного в зависимости от положения французского существительного во фразе, точнее, от того, какие слова ему предшествуют. Подробно различные случаи изложены в приложении. После существительных анализируются местоимения.

Из местоимений в обработке нуждаются следующие: *nous, me, elle, elles, lui, ce, cela, ceci, qui, que, dont, on, lequel, laquelle, lesquels, lesquelles*. Общих правил обработки местоимений нет, местоимения разбиты на типы, каждый из которых разбирается самостоятельно (см. приложение).

После анализа местоимений переходим к анализу *participe passé* и *participe présent*. Специальные правила обработки причастий касаются тех *participe passé*, которые не вошли в аналитические формы глагола, и тех *participe présent*, которые не переводятся деепричастиями. Среди оставшихся причастий сначала выделяются те, которые должны переводиться глаголом третьего лица. Дело в том, что во французском языке возможна, например, конструкция *On va définir la courbe par son équation $y=f(x)$, x variant dans l'intervalle (0,1)...*, которая на русский язык переводится: «Мы определим кривую ее уравнением $y=f(x)$, причем x меняется в интервале (0,1)», т. е. *participe présent* при переводе переделывается в возвратный глагол настоящего времени третьего лица и того числа, какое имеет предшествующее существительное в именительном падеже.

Причастия, которые не нуждаются в переделке и являются определениями, обрабатываются аналогично прилагательным, т. е. такое причастие привязывается к существительному, с которым оно согласуется во французской фразе, и затем его перевод согласуется в русской фразе с переводом того существительного, к которому это причастие привязано.

Как уже было сказано, анализирующие правила устанавливают данные о форме каждого переводящего русского слова. Именно, для существительных устанавливается число и падеж; для глаголов—число, время и вид, лицо или род, возвратность; для местоимений—число и падеж, если можно, род; для прилагательных и причастий, выступающих в роли прилагательных,—число, падеж, показатель полной или краткой формы и род (род для глаголов прошедшего времени, прилагательных и причастий берется из словарной информации того русского существительного, с которым перечисленные слова должны быть согласованы), кроме того, для причастий еще устанавливается время, залог и возвратность.

Что касается установления места переводящих слов, то в основном мы сохраняем тот порядок, который был во французской фразе; исключение составляют прилагательные, причастия и местоимения *en* и *dont*.

Прилагательное или причастие, которое во французском языке стоит обычно после того существительного, к которому оно относится, ставится при переводе перед этим существительным вместе с относящимися к нему наречиями. Например, *Une dernière remarque relativement aux fonctions entières qui n'ont pas de racines* переведется: «Последнее замечание относительно целых функций, которые не имеют корней».

Однако этой перестановки не производится, если прилагательное или причастие управляет идущим за ним предлогом. Например, *Voici une remarque générale relative au cas où l'une des limites devient infinie* переведется: «Вот общее замечание, относящееся к случаю, когда один из пределов становится бесконечным». Прилагательное *relative* в данном случае не переставляется, так как оно управляет предлогом *à*.

Причастие не переставляется также в случае, если оно образовано от переходного глагола и за ним стоит существительное без предлога, например, *le contour C limitant l'aire A est convexe*—«контур С, ограничивающий область А, выпуклый».

Перестановки, связанные с местоимением *dont*, делаются только в том случае, если подлежащее придаточного предложения, присоединенного словом *dont*, не является прилагательным. Если при этом глагол придаточного предложения переходный, то перед *dont* ставится прямое дополнение этого предложения вместе с относящимися к нему прилагательными. Например, *Soit $x=a$ une droite rencontrant la courbe $f(x, y)$ en m points distincts, dont nous désignerons les coordonnées par y_1, y_2, \dots, y_m* переведется: «Пусть $x=a$ прямая, пересекающая кривую $f(x, y)$ в m различных точках, координаты которых мы обозначим через y_1, y_2, \dots, y_m ...». Если же глагол придаточного предложения непереходный, то перед *dont* ставится подлежащее придаточного предложения вместе с относящимися к нему прилагательными. Например, *Divisons l'arc $\bar{\alpha}\beta$ en un certain nombre d'intervalles par les points de subdivision $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$, dont les coordonnées seront $(x_1, y_1), (x_2, y_2), \dots, (x_{n-1}, y_{n-1})$* переведется: «Разделим дугу $\bar{\alpha}\beta$ на некоторое число интервалов точками деления $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$, координаты которых будут $(x_1, y_1), (x_2, y_2), \dots, (x_{n-1}, y_{n-1})$ ».

Если *dont* стоит перед прилагательным, то перестановка не производится. Например, *dont* в сочетании *dont le premier* переведется «из которых» и не переставится.

Полные анализирующие правила приводятся в приложении.

5. Синтез

После работы анализирующей части алгоритма начинается синтез переводящей русской фразы. Задача синтеза—образовать нужные формы русских слов на основе данных, полученных при анализе.

Последовательность обработки слов при синтезе не существенна, можно было бы просто обрабатывать слова в том порядке, как они стоят во фразе. Однако в целях удобства осуществления на машине синтезирующие правила также сгруппированы по классам слов, которых в русском языке мы различаем только пять [1].

Для каждого русского слова, имеющего несколько основ, нужно выбрать из них одну, пользуясь русской информацией, где указано, какая из основ используется для образования той или иной формы.

После того, как основы определены, к ним нужно добавить окончания. По указанию в русской информации, где дается тип изменения слова, или, иначе говоря, номер таблицы окончаний, которую нужно использовать, обращаются к этой таблице и из нее «вынимают» то окончание, которое служит для образования требуемой формы.

Если некоторое слово имеет несколько переводов, то все они будут выданы в обработанной форме в том порядке, как они стояли в словаре. Слова, оставшиеся непереуведенными, т. е. те, для которых не нашлось соответствующей основы в словаре, выдаются в неизмененном виде и ставятся в русской фразе в скобках.

Работой синтезирующих правил заканчивается перевод фразы.

ПРИЛОЖЕНИЕ

В приводимых ниже правилах для классов слов, соответствующих принятым у нас частям речи, используются следующие обозначения:

- S—существительное,
- V—глагол,
- Pr—предлог,
- Cc—сочинительный союз,
- Cs—подчинительный союз,
- Ad—прилагательное—детерминатив,
- A—прилагательное—не детерминатив,
- P—местоимение,
- adv—наречие,
- N—числительное,
- Pr—participe passé,
- Ppr—participe présent,
- A-S—субстантивированное прилагательное,
- Ф—формула,
- З—запятая,
- T—точка.

І Р а з д е л

А. Правила различения омонимии

Различение омонимии начинается с рассмотрения группы случаев, в которых один из омонимов—глагол. С них удобно начинать потому, что для правильного разбора остальных случаев омонимии очень важно знать расположение глаголов во фразе.

1. Г л а г о л—с у щ е с т в и т е л ь н о е.

1.1 Совпадение формы 3 *pers. sing. prés. ind.* глагола с формой единственного числа существительного. Этот случай обозначим VS₁.

Такой случай омонимии дают следующие основы нашего словаря: *base, borne, compte, figure, forme, formule, groupe, mesure, note, part, place, remarque, reste, somme*. Если встретилась одна из перечисленных основ, то сначала нужно проверить, имеется ли окончание у слова с данной основой.

1.11 Если имеется окончание *-s*, то VS_1 является существительным (окончание *-s* может быть также окончанием 2 *pers. sing. prés. ind.* глагола, но эта форма глагола в математических текстах не встречается).

1.12 Если есть любое другое окончание, то VS_1 является глаголом.

1.13 Если окончания нет, то посмотреть, что стоит перед VS_1 .

1.131 Если перед VS_1 стоит детерминатив (но не *le, la, les*) или предлог (но не *en*) или *le, la, les*, перед которым стоит любой предлог (в том числе и *en*), то VS_1 —существительное.

Оговорку о *le, la, les, en* приходится делать потому, что *le, la, les* могут быть прямым объектным местоимением при глаголе (например, *il la forme*), а *en* может быть приглагольным местоимением (например, *il en forme*); правда, в математических текстах это маловероятно.

1.132 Если перед VS_1 стоит *les* без предлога, то VS_1 —глагол.

1.133 Если перед VS_1 стоят *le, la*, то VS_1 является глаголом, когда фраза не начинается с *soit* и влево от VS_1 во фразе есть $S_N(P_N)^*$, к которому не отнесен ни один глагол фразы**), а вправо от VS_1 во фразе нет глагола множественного числа, который может быть отнесен к $S_N(P_N)$ и VS_1 , если их считать однородными подлежащими.

1.134 Если перед VS_1 стоит *en*, то VS_1 считать глаголом в следующих случаях: а) фраза начинается с *soit*; б) VS_1 стоит в придаточном, начинающемся с *qui*; в) после VS_1 идет $S_N(P_N)$ без предлога.

1.135 Если перечисленные условия для VS_1 после *le, la, en* не выполнены, то VS_1 —существительное.

1.136 В остальных случаях VS_1 —глагол.

1.2 Совпадение формы *participe passé* глагола с формой единственного числа существительного. Этот случай обозначим VS_2 . Такой случай омонимии дают следующие основы нашего словаря: *dérivée, donné, énoncé, étendu, exposé, ordonnée, procédé, résumé*.

1.21 После глагола *avoir* (может отделяться наречием) VS_2 —*participe passé*. Правило имеет в виду аналитические формы глагола, вне этих форм *participe passé* выступает обычно в функции определения.

1.22 Если глагола *avoir* перед VS_2 нет, то VS_2 разбирается по правилам различения омонимии существительного и прилагательного.

*) Под $S_N(P_N)$ здесь понимается существительное (местоимение), перед которым нет ни предлога, ни переходного глагола.

**) Отнесение глаголов к существительным (местоимениям), т. е. к $S_N(P_N)$, делается по следующим правилам: глагол считается относящимся к существительному, если имеет место один из случаев:

а) они согласованы в лице и числе и отделены только прилагательными, причастиями и наречиями;

б) они согласованы в лице и числе и хотя отделены не только прилагательными, причастиями и наречиями, но после $S_N(P_N)$ есть определительное придаточное предложение, вводимое одним из слов: *qui, que, dont, où, le quel* (и его формы), глагол которого отнесен по одному из этих правил к подлежащему этого придаточного, причем между $S_N(P_N)$ и относимым к нему V нет другого глагола, который нельзя отнести к подлежащему придаточного предложения и можно отнести к $S_N(P_N)$;

в) они согласованы в лице, глагол стоит во множественном числе, и имеется несколько существительных, к которым не отнесены другие глаголы, причем перед последним из этих существительных есть сочинительный союз;

г) они согласованы в лице и числе, имеется несколько глаголов, согласующихся с данным существительным, причем перед последним из них стоит сочинительный союз.

1.3 Совпадение форм 3 *pers. sing. prés. ind.* глагола, *participe passé* глагола и формы единственного числа существительного. Этот случай обозначим $V_{pp}S$, к нему относятся основы *fait, produit*.

1.31 С окончанием *-e, -es*—это *participe passé*.

1.32 С окончанием *-s*—это либо *participe passé*, либо существительное, разбор производится, как в п. 1.2.

1.33 Если окончания нет, то:

1.331 При наличии слева глаголов *avoir* или *être* (могут отделяться наречиями) $V_{pp}S$ —*participe passé*.

1.332 При наличии слева детерминатива (но не *le, la, les*), предлога (но не *en*) или любого детерминатива с любым предлогом $V_{pp}S$ —существительное.

1.333 При наличии *le, la* или *en* $V_{pp}S$ —либо глагол, либо существительное, разбор делается так же, как в случае 1.1.

1.334 При наличии слева существительного (может отделяться прилагательными и наречиями) $V_{pp}S$ —либо глагол, либо *participe passé*. $V_{pp}S$ —глагол, если фраза не начинается с *soit* и слева от $V_{pp}S$ есть $S_N(P_N)$, к которому не отнесен ни один глагол фразы. В противном случае $V_{pp}S$ —*participe passé*.

1.335 Во всех остальных случаях $V_{pp}S$ —глагол.

1.4 Совпадение основы глагола с формой единственного числа существительного—искусственная омонимия. Этот случай обозначим VS_4 .

К этому случаю относятся основы: *abord, appel, bas, calcul, rapport, écart*. Без окончания или с окончанием *-s* VS_4 —существительное, с другими окончаниями—это глагол.

2. Г л а г о л — п р е д л о г.

2.1 Слово *entre* может быть либо предлогом («между»), либо формой 3 *pers. sing. prés. ind.* глагола *entrer*. Различение делается по следующему правилу: *entre*—предлог, если непосредственно за ним следует формула, числительное или прилагательное (любое, как детерминатив, так и не детерминатив). В остальных случаях—это форма глагола.

Правило основано на том, что в математических текстах встретить *entrer* в качестве переходного глагола маловероятно, а после непереходного глагола существительное может стоять только с предлогом, т. е., двигаясь от *entre*—глагола вправо, мы, прежде чем дойти либо до формулы, либо до числительного или прилагательного, предшествующих существительному, наткнемся на предлог, тогда как за *entre*—предлогом должно, очевидно, следовать одно из перечисленных слов.

2.2 Искусственная омонимия: основы *par* и *pour* могут быть либо предлогами, либо основами глаголов *paraître* и *pouvoir*.

Правило различения: если есть окончание, это основа глагола, нет окончания—предлог.

3. Г л а г о л — н а р е ч и е.

3.1 Слово *soit* (*soient*), вообще говоря, может быть наречием («пусть»), союзом («либо..., либо...») и формой 3 *pers. prés. subj.* (*sing.* или *pl.*) глагола *être*. Однако, так как в качестве союза *soit* обычно соединяет однородные члены предложения (не является границей предложения), то такой союз без ущерба для правильности формального анализа фразы можно считать наречием. Поэтому в нашем словаре *soit*—либо форма глагола, либо наречие с двумя переводами «пусть» и «либо». Разбор этих случаев идет следующим образом.

3.11 *Soit*—наречие в начале предложения, после сочинительного союза и после запятой, если в промежутке от предыдущей запятой до *soit* есть глагол (не причастие и не деепричастие).

3.111 Наречие *soit* переводится «либо», если во фразе два *soit* подходят под правило определения наречия и они не соединены сочинительным союзом.

3.112 Наречие *soit* переводится «пусть», если оно не подходит под условия предыдущего пункта.

3.12 В остальных случаях *soit*—глагол.

3.2 Искусственная омонимия: *peu* и *se*.

Peu—это либо наречие («мало»), либо одна из основ глагола *pouvoir*. *Se* в нашем словаре—это либо наречие, указывающее на возвратность глагола, либо основа глагола *être* (для форм будущего времени).

Правило различения: с окончанием—это основа глагола, без окончания—наречие.

4. Г л а г о л (л и ч н а я ф о р м а) — п р и ч а с т и е.

Обозначение VPr. К этому случаю относятся слова: *dit, écrit, extrait, restreint*, которые могут быть либо формой *participe passé* глагола, либо формой 3 pers. sing. prés. ind.

4.1 Они являются *participe passé* в следующих случаях:

4.11 Когда есть окончание (например, *dite, écrits* и т. д.).

4.12 После глаголов *avoir* (сложное время) или *être* (пассивная форма), которые могут отделяться от VPr наречиями.

4.13 После существительного в случае, если фраза начинается с *soit*, или в случае, если влево от VPr нет $S_N(P_N)$, к которому не отнесен другой глагол.

4.2 В остальных случаях VPr—личная форма глагола.

5. Г л а г о л — п р и л а г а т е л ь н о е.

5.1 Основы: *égal, précis, présent*—могут быть основами либо глагола, либо прилагательного. Этот случай обозначим VA_1 .

5.11 Без окончания или с окончанием *-s, -es* VA_1 —прилагательное.

5.12 С окончанием *-e* VA_1 может быть как прилагательным (*feminin singulier*), так и глаголом (3 pers. sing. prés. ind.).

5.121 VA_1 считается прилагательным после глагола *être* (*la valeur de $F(x)$ est égale à Q*) и после существительного в условиях п. 4.13.

5.122 В остальных случаях VA_1 с окончанием *-e*—глагол.

5.13 С другими окончаниями VA_1 —глагол.

5.2 Основы: *fixe, alterne*. Этот случай обозначим VA_2 .

5.21 С окончанием *-s* VA_2 —прилагательное.

5.22 С любым другим окончанием VA_2 —глагол.

5.23 Если окончания нет, то VA_2 —либо прилагательное, либо форма 3 pers. sing. prés. ind. глагола. Разбор этого случая делается, как в п. 5.12.

6. Г л а г о л — с у щ е с т в и т е л ь н о е — п р и л а г а т е л ь н о е.

6.1 Основа *continui* может быть основой глагола *continuer* («продолжать»), основой прилагательного («непрерывный») и основой существительного («континуум»).

6.11 С окончанием, отличным от *-e, -s, -es*,—это глагол.

6.12 С окончанием *-es*—это прилагательное.

6.13 С окончанием *-e*—это либо прилагательное, либо глагол; разбирается по правилам п. 5.12.

6.14 С окончанием *-s* или без окончания—это либо прилагательное, либо существительное; разбирается так же, как слова типа SA (см. п. 16).

6.2 Основа *limite* может быть формой 3 pers. sing. prés. ind. глагола *limiter* («ограничивать») или основой других форм того же глагола, основой существительного («предел») и основой прилагательного («предельный»).

6.21 С окончанием *-s*—это либо существительное, либо прилагательное; разбирается так же, как другие случаи омонимии SA (см. п. 16).

6.22 С любым другим окончанием—это глагол.

6.23 Если окончания нет, то нужно искать влево глагол *être* (может отделяться наречиями).

6.231 После него *limite*—прилагательное.

6.232 Если глагола *être* нет, то искать влево детерминатив (не *le*, *la*), предлог (не *en*) или любой детерминатив с любым предлогом; далее разбор ведется, как в п. 1.33 с заменой в написанных там правилах *participe passé* на прилагательное.

7. Местоимение — предлог.

Сюда относится слово *en*.

7.1 Если непосредственно после него стоит второе *en* («*en en prenant*»—«беря из этого») или если справа (может отделяться наречиями и местоимениями: *nous*, *se*, *le*, *la*, *les*, *y*, *lui*, *leur*, *me*) стоит глагол, то *en*—местоимение.

7.2 В остальных случаях—это предлог.

8. Существительное — наречие.

Сюда относятся: *pas*, *point* и *ensemble*.

8.1 Для *pas* и *point* правило различения такое:

8.11 Непосредственно после глагола и *ne*—это наречие.

8.12 В остальных случаях—существительное.

Действительно, в качестве наречия—второго элемента отрицания—эти слова либо стоят непосредственно за глаголом, либо за наречием *ne* в сочетаниях типа: *être ou ne pas être*.

8.2 Для *ensemble* правило различения другое, так как это наречие может отделяться от глагола любыми словами (например, *on peut démontrer ces deux théorèmes ensemble*).

8.21 Если у *ensemble* есть окончание (*ensembles*), то это существительное.

8.22 Если окончания нет, то:

8.221 *Ensemble* считать существительным, когда перед ним стоит прилагательное—детерминатив мужского рода единственного числа или предлог.

8.222 В остальных случаях—это наречие.

9. Прилагательное — предлог.

Сюда относится слово *suivant*.

9.1 С окончаниями *-e*, *-s*, *-es*—это прилагательное.

9.2 При отсутствии окончания:

9.21 Оно считается предлогом, если непосредственно за ним следует прилагательное—детерминатив и, кроме того, имеет место один из следующих трех случаев.

9.211 Перед *suivant* (не переходя через глагол, подчинительный союз) нет существительного мужского рода единственного числа, например, *Suivant le théorème...*

9.212 Перед *suivant* стоит существительное без предлога. В этом случае имеется в виду, что два существительных без предлога, идущие подряд (однородные члены), должны разделяться запятой, т. е. если бы *suivant* было не предлогом, а прилагательным, относящимся к первому существительному, то после него стояла бы запятая. Например, *Nous pouvons déduire ce théorème suivant la règle... Nous pouvons déduire le théorème suivant, la règle...*

9.213 Перед *suivant* стоит существительное с предлогом и предшествующий ему глагол непереходный. Правило основано на том, что после непереходного глагола не может идти существительное без предлога.

Например, *Nous pouvons agir dans ce théorème suivant la règle...* Если же глагол переходный, то прямое дополнение без предлога должно стоять впереди косвенного с предлогом *suitant*, т. е. получаем случай 9.212.

9.22 В остальных случаях *suitant*—прилагательное.

10. Местоимение — прилагательное (детерминатив).

Сюда относятся *la, le, les* и *ce*. Они разбираются по-разному.

10.1 *Ce* считается местоимением, когда оно стоит перед глаголом (может отделяться наречиями), а в остальных случаях—прилагательным. Правило основано на том, что *ce*—местоимение выступает в роли подлежащего, значит, в утвердительном предложении должно стоять перед глаголом.

10.2 Местоимения *le, la, les* являются в предложении прямыми дополнениями, поэтому могут стоять как перед глаголом (*nous les prenons*), так и после него (*prenons les*). Поэтому предыдущего правила для них недостаточно.

10.21 Слова *le, la, les* являются местоимениями: а) если за ними следует глагол (может отделяться местоимениями *lui, leur, en, y*), б) если за ними следует союз, предлог, запятая, точка (могут отделяться наречиями, но только не наречиями *plus, moins*, имеющими омоним—прилагательное).

Это правило основано на том, что после стоящего за глаголом прямого дополнения может идти либо косвенное дополнение с предлогом, либо другое прямое дополнение, которое будет отделяться от первого запятой или сочинительным союзом, либо фраза на этом кончается (случай подчинительного союза и точки).

10.22 В остальных случаях—это прилагательное.

11. Прилагательное с тремя переводами — наречие с двумя переводами.

Сюда относится слово *même*, которое может быть либо прилагательным с переводом «тот же самый» (например, *le même segment*—«тот же самый отрезок», *segment de même longueur*—«отрезок той же самой длины»), либо прилагательным с переводом «один и тот же» (например, *Nous prenons chaque fois un même segment*—«Мы берем каждый раз один и тот же отрезок»), либо прилагательным с переводом «сам» (например, *lui-même*—«он сам»), либо наречием с переводом «даже» (например, *Nous pouvons même démontrer...*—«Мы можем даже доказать...»), либо в сочетании с предлогом *de* наречием с переводом «так же» (например, *on agit de même dans le cas...*—«так же действуют в случае...»).

Для различения надо искать слева существительное, местоимение, прилагательное—детерминатив, предлог.

11.1 При наличии слева детерминатива (не *le, la, les, un, une*), предлога (не *de*), существительного или местоимения *même*—прилагательное с переводом «сам».

11.2 При наличии слева детерминативов: *un, une*—*même*—прилагательное с переводом «один и тот же».

11.3 При наличии слева детерминативов: *le, la, les*—*même*—прилагательное с переводом «тот же самый».

11.4 При наличии слева предлога *de*:

11.41 Если вправо от *même* есть существительное, то *même*—прилагательное с переводом «тот же самый».

11.42 Если существительного нет, то *même*—наречие с переводом «так же».

11.5 В остальных случаях *même*—наречие с переводом «даже».

12. Прилагательное — наречие — подчинительный союз.

Сюда относится слово *plus*. Оно условно считается прилагательным в сочетании *le plus*, которое переводится словом «самый». *Plus* считается союзом (тоже совершенно условно, лишь для удобства анализа), когда оно, как математический знак «плюс» соединяет некоторые выражения. Наконец, в остальных случаях — это наречие «более».

12.1 При наличии слева от *plus* детерминатива *plus* — прилагательное.

12.2 При наличии слева от *plus* существительного без предлога (может отделяться прилагательными и наречиями) и справа тоже существительного без предлога (может отделяться детерминативами) *plus* — союз.

12.3 Если указанные условия не выполнены, то *plus* — наречие.

13. Наречие — подчинительный союз.

Сюда относится слово *si*.

13.1 Оно считается наречием, если стоит перед прилагательными или наречием, за которыми следует запятая, *que* или точка.

13.2 В остальных случаях *si* — подчинительный союз.

14. Прилагательное — наречие.

Сюда относятся слова: *vite, vrai, haut*.

14.1 Они считаются наречиями, если стоят после глагола (могут отделяться наречиями).

14.2 В остальных случаях — это прилагательное.

15. Местоимение — наречие — подчинительный союз.

Сюда относится слово *que*.

15.1 В начале фразы *que* — наречие, переводимое «пусть».

15.2 Предположим, *que* стоит после *même, moins, tel* (и его формы), *plus, aussi* или *moindre*, не отделенных глаголом или подчинительным союзом.

15.21 Если впереди стоит *même*, то *que* — наречие, переводимое «что и» (например, *La même équation que...* — «То же самое уравнение, что и...»).

15.22 Если впереди стоят *plus, moins* или *moindre*, то *que* — союз с переводом «чем» (например, *Prenons y plus grand que e* — «Возьмем y больше чем e»).

15.23 Если впереди стоит *tel*, то *que* — союз с переводом «что».

15.24 Если впереди — *aussi*, то *que* — наречие с переводом «как».

15.3 Если слева от *que* стоит *ne* (до союза или запятой) и если между *ne* и *que* не найдено ни одного из слов: *rien, jamais, personne, aucun, pas, point* — и влево от *ne* до подчинительного союза нет ни одного из слов: *rien, jamais, personne, aucun*, то *que* — наречие; оно переводится словом «только», а *ne* уничтожается.

15.4 При наличии слева запятой или сочинительного союза (могут отделяться наречиями) *que* переводится так же, как ближайшее слева другое *que*.

15.5 Если слева от *que* найден глагол, то *que* — союз с переводом «что» или «чтобы».

Правила выбора одного из этих значений разработаны не полностью. Проверяется, не входит ли глагол, стоящий перед *que*, в специальный список, где собраны слова, после которых перевод «чтобы» невозможен (например, *que* после глагола *supposer* всегда переводится словом «что»). Если глагол стоит в списке, то следует оставить перевод «что», если не стоит, то оставить оба перевода: «что» и «чтобы».

15.6 При наличии слева от *que* существительного без предлога, *que* — местоимение с переводом «который».

15.7 При наличии слева от *que* существительного с предлогом:

15.71 Если глагол справа от *que* непереходный, то *que*—союз «что» или «чтобы» (эти два перевода различаются, как было сказано выше в п. 15.5).

15.72 Если глагол справа от *que* переходный, то:

15.721 При наличии существительного (местоимения) без предлога между *que* и глаголом проверяют, есть ли вправо от глагола прямое дополнение (существительное или местоимение без предлога).

15.7211 Если нет, то *que*—местоимение с переводом «который».

15.7212 Если есть, то *que*—союз с переводом «что» или «чтобы».

15.722 При отсутствии существительного (местоимения) без предлога между *que* и глаголом *que*—местоимение с переводом «который». Например, фраза *La valeur au centre d'une sphère Σ d'une fonction $V(x, y, z)$ harmonique dans une région contenant Σ est la moyenne des valeurs que prend V sur la sphère* переведется: «Значение в центре сферы Σ функции $V(x, y, z)$ гармонической в области, содержащей Σ , есть среднее значений, которые принимает V на сфере», а фраза *Quant au terme—1, il provient du fait que l'opération de la dérivation supprime dans (21) une constante* переведется: «Что касается члена —1, то он происходит от того факта, что операция дифференцирования упраздняет в (21) константу».

16. С у щ е с т в и т е л ь н о е — п р и л а г а т е л ь н о е.

Сюда относятся слова: *absurde, arithmétique, carré, compact, composant, complexe, découverte, différentiel, droit, extérieur, équivalent, fin, infini, instant, intégral, intérieur, invariant, issue, mathématique, moyen, multiple, orthogonal, pair, parallèle, plan, potentiel, pratique, radical, rectangle, scalaire, statistique, tangente, vide*.

16.1 При наличии слева существительного (может отделяться наречиями и прилагательными) обрабатываемое слово—прилагательное.

16.2 При наличии слева предлога, детерминатива или числительного:

16.21 Если справа стоит существительное и оно не имеет омонима—прилагательного, то разбираемое слово—прилагательное, относящееся к этому существительному.

16.22 Если найденное существительное того же типа, что и разбираемое слово (т. е. существительное—прилагательное), то первое слово считается существительным, а второе прилагательным (поскольку во французском языке прилагательные, как правило, стоят после существительных, к которым они относятся).

16.3 При наличии слева глагола (может отделяться наречиями) обрабатываемое слово—прилагательное.

16.4 При наличии слева сочинительного союза (может отделяться наречиями) искать влево прилагательное или причастие (*participe présent* или *participe passé*).

16.41 Найдя, проверить, согласовано ли найденное слово с обрабатываемым.

16.411 Если они согласованы, то считать обрабатываемое слово прилагательным, относящимся к тому же существительному, что и найденное слово.

16.412 Если они не согласованы, то искать дальше влево прилагательное или причастие (вернуться к 16.41).

16.42 Если влево нет искомых слов, то искать влево существительное.

16.421 Найдя, проверить, согласовано ли с ним обрабатываемое слово.

16.4211 Если они согласованы, то считать обрабатываемое слово прилагательным, относящимся к этому существительному.

16.4212 Если они не согласованы, то снова искать существительное (вернуться к 16.421).

16.422 Если существительное не найдено, то считать обрабатываемое слово существительным.

По этим же правилам обрабатываются прилагательные и устанавливается, не являются ли они субстантивированными.

17. Прилагательное с четырьмя переводами.

Правило касается слова *tout*, которое может переводиться как «всякий», «целый», «весь» и «все».

17.1 Перед существительным *tout* переводится «всякий» (например, *tout nombre entier*—«всякое целое число»).

17.2. Перед *un, une* оно переводится «целый» (например, *tout une heure*—«целый час»).

17.3 Перед *le, la, les* оно переводится «весь» (например, *dans tout le plan*—«во всей плоскости»).

17.4 В остальных случаях *tout* переводится «все» (например, *c'est tout ce que je voulais vous dire*—«это все, что я хотел вам сказать»).

18. Существительное с двумя переводами.

Существительное *infinité* в сочетании с предлогом *de* (*infinité de*) переводится как «бесконечное число», в остальных случаях—как «бесконечность».

19. Глагол с двумя переводами.

Сюда относятся формы *est* и *sont* глагола *être* с переводом «быть», которые не переводятся, если за ними не следует детерминатив. Дело в том, что *être* перед детерминативом, т. е. перед существительным, в математических текстах обычно не опускается. Например, «...leur différence est une série trigonométrique...»—«...их разность есть тригонометрический ряд...».

20. Наречие с двумя переводами.

20.1 *Moins* переводится либо «наименее», либо «меньше». Если непосредственно влево от него стоит *le, la, les*, то *moins* переводится—«наименее». Имеется в виду сравнительная степень прилагательных (например, *le moins possible*—«наименее возможный»). В остальных случаях *moins*—«меньше».

20.2 *Etant* в нашем словаре условно названо наречием, потому что *étant* переводится неизменяемыми словами и не требует определения каких-либо данных о переводе, а выполняемая им роль в предложении такова, что такое отнесение к другому классу не приводит к ошибкам при анализе.

Etant переводится «где», если оно стоит в последнем предложении (т. е. вправо от него уже нет глагола), и «когда»—в остальных случаях. Например, фраза: *Il est clair que z étant inférieur à R , il existe un nombre H ... d'où $|I| < \varepsilon + (2g\delta/R)$, δ et ε étant tout deux arbitrairement petits* переводится: «Ясно, что когда z меньше R , существует число H ...»; переводится: «...откуда $|I| < \varepsilon + (2g\delta/R)$, где δ и ε оба произвольно малые».

21. Прилагательное с тремя переводами.

Прилагательное—детерминатив *un (une)* либо переводится «один», либо опускается при переводе, либо переводится «некоторый».

21.1 После оборота *à partir de* слово *un* переводится как «некоторый».

21.2 Если *un* стоит после сочинительного союза, то при наличии слева второго *un* рассматриваемое *un* переводится так же, как предыдущее.

21.3 Если перед *un (une)* стоит *l'*, то *un* переводится «один» (например, *l'un d'eux*—«один из них»).

21.4 Если перечисленные условия не имеют места, то ищем вправо существительное, пропуская прилагательные и наречия.

21.41 При наличии справа от *un* существительного (может отделяться прилагательными и наречиями) и справа от существительного оборотов *au plus* или *au moins un* переводится «один». Например, *On peut trouver une fonction au moins qui satisfait la condition* переведется: «Можно найти по крайней мере одну функцию, которая удовлетворяла условию».

21.42 При отсутствии справа от *un* существительного (может отделяться прилагательными и наречиями) *un* переводится «один».

21.5 В остальных случаях *un* при переводе опускается.

22. П р а в л о о б р а б о т к и ф о р м у л.

22.1 Если влево от формулы есть существительное (может отделяться прилагательными и наречиями), то такую формулу причислить к наречиям.

22.2 Если непосредственно слева стоит сочинительный союз и слева от него стоит формула, причисленная к наречиям, то обрабатываемую формулу считать наречием.

22.3 В остальных случаях оставить ее в шкале формул и обрабатывать наравне с существительными.

23. У с т а н о в л е н и е п е р е х о д н о с т и г л а г о л а.

Эти правила составлены для глаголов, которые могут быть и переходными и непереходными.

23.1 При наличии слева от глагола местоимений: *le, la, les*—глагол переходный.

23.2 При наличии справа от глагола существительного или местоимения (не отделенных подчинительным союзом или другим глаголом) без предлога (или с предлогом *de*, если перед глаголом есть отрицание *ne*) глагол переходный.

23.3 В остальных случаях глагол—непереходный.

По существу правило сводится к тому, что глагол считается переходным при наличии прямого дополнения, причем учитывается, что между глаголом и прямым дополнением может стоять обстоятельство или косвенное дополнение, выраженное существительным с предлогом. Например, во фразе: *Cette ligne peut continuer jusqu'au point A*—глагол *continuer*—непереходный. Во фразе: *On peut continuer cette ligne jusqu'au point A*—глагол *continuer*—переходный.

Б. Правила анализа

1. Глагол

П р а в и л а о к о н ч а т е л ь н о й о б р а б о т к и

1. По предварительным данным получено *настоящее время*.

1.1 Перед глаголом в 1 л. мн. ч. нет *nous* и нет сочинительного союза (могут быть отделены от глагола наречиями и местоимениями)—настоящее время заменить будущим совершенного вида.

Здесь имеется в виду форма 1-го лица мн. числа в *impératif*: *considérons la série...*—«рассмотрим ряд»...

1.2 Перед глаголом в 1 л. мн. ч. есть сочинительный союз (может отделяться наречиями и местоимениями)—искать слева другой глагол в 1 л. мн. ч. и переводить так же, как переводится тот.

1.3 Перед глаголом есть *que*, и оно переводится словом «чтобы»—настоящее время заменить прошедшим совершенного вида.

Здесь имеется в виду совпадение *subjonctif* и *indicatif* в I группе. Так как машина не различает графически совпадающие формы, то для нее во фразах: *nous savons qu'il parle* и *nous voulons qu'il parle*—глагол *parle* стоит в одной и той же форме; *-e*—помечено в таблице как настоящее время. Формальным признаком того, что глагол с окончанием *-e* стоит в фор-

ме *subjunctif*, является наличие слова *que*. *Subjunctif* обычно переводится прошедшим временем совершенного вида + «бы», причем «бы» убирается при наличии слева «чтобы».

Выбор совершенного вида в этом и в некоторых других случаях обусловлен причинами практического удобства. В подавляющем большинстве случаев перевод будет верным, искажения смысла не произойдет, а это основное, что нас интересует.

1.4 Перед глаголом есть *si* (может отделяться местоимением или существительным без предлога и наречиями), а в пределах фразы есть другой глагол в *futur*—настоящее время заменить на будущее совершенного вида.

Здесь имеется в виду условное придаточное, вводимое *si*. Например, *Si l'on développe, suivant les puissances de z l'expression..., on verra que le coefficient...* переведется: «Если мы разложим по степеням z выражение..., мы увидим, что коэффициент...».

1.5 При отсутствии условий, указанных в правилах 1.1, 1.2, 1.3, 1.4, оставить настоящее время.

2. Правило определения лица.

В случаях, когда окончание первого и третьего лица французских глаголов совпадают, этому окончанию в таблицах ставится в соответствие условное четвертое лицо. Русским временам, не нуждающимся в определении лица (прошедшее время, неопределенная форма), в таблице дается условное нулевое лицо.

Правило уточнения лица составлено только для четвертого лица.

Четвертое лицо заменить на первое при наличии слева местоимения *je* (может отделяться местоимениями: *le, la, les, lui, leur, en, y, me* и наречиями), при его отсутствии—на третье лицо. В остальных случаях остается лицо, полученное по предварительным данным.

3. По предварительным данным получено *будущее время* совершенного вида.

3.1 Если есть пометка окончательности, то уточнения не требуется. Пометка окончательности ставится при окончаниях *futur*, так как *futur* всегда переводится будущим временем.

3.2 Будущее время без пометки окончательности стоит в таблицах при окончаниях *conditionnel*; случай 3.2 имеет в виду *conditionnel* в главном предложении условного периода. Если пометки окончательности нет и в пределах фразы есть *si* с переводом «если», то будущее время заменить прошедшим несовершенного вида + «бы».

3.3 В остальных случаях оставить время, полученное по предварительным данным. Имеется в виду модальное употребление *conditionnel* в независимых предложениях, где оно будет переводиться будущим временем. Например, *La démonstration s'appliquerait sans modification à une fonction telle que...* переведется: «Доказательство применится без изменения к такой функции, что...».

4. По предварительным данным получено *прошедшее время*.

4.1 Прошедшее время несовершенного вида дается в таблицах для окончаний *imparfait*.

4.11 Если перед глаголом есть *si* с переводом «если», то поставить «бы» после «если» и записать глаголу прошедшее время совершенного вида.

4.12 Если *si* отсутствует, то оставить время и вид, полученные по предварительным данным.

4.2 Прошедшее время несовершенного вида + «бы» дается в таблицах для окончаний *subjunctif* в тех группах, где оно отличается от *présent indicatif*.

4.21 Если перед глаголом есть *que* с переводом «чтобы», то убрать «бы», а у глагола оставить вид и время, полученные по предварительным данным.

4.22 Если *que* отсутствует, то оставить время и вид по предварительным данным.

5. По предварительным данным получено *participe passé* или *participe présent*.

Причастия (и *participe passé*, и *participe présent*) могут играть в предложении двоякую роль: они либо ведут себя как форма глагола (образуя сказуемое) и тогда должны обрабатываться вместе с глаголом, либо ведут себя как прилагательные (являясь определениями) и тогда должны обрабатываться в конце анализа. Поэтому обработка причастий начинается с выяснения их роли в переводимой фразе.

5.1 *Participe passé*.

Если *participe passé* стоит после любой формы *avoir* или *s'être*, а также после *être* для глаголов, которые спрягаются с *être* (например, *aller*, *venir*), причем вспомогательный глагол может быть отделен от причастия наречиями, то *participe passé* считается глагольной формой, перевод которой зависит от того, какую форму имеет вспомогательный глагол.

5.11 Если вспомогательный глагол в неопределенной форме и перед ним стоит *après*, то *participe passé* переводить деепричастием прошедшего времени совершенного вида, а слово *après* и вспомогательный глагол опустить. Например, сочетание *après avoir construit* переводится «построив».

5.12 Если вспомогательный глагол в неопределенной форме и перед ним стоит *sans* (см. правило 6.1), то *participe passé* переводить деепричастием прошедшего времени, вспомогательный глагол стереть. Например, сочетание *sans avoir fait* переведется «не сделав».

5.13 Если вспомогательный глагол в *participe présent*, то *participe passé* переводить деепричастием прошедшего времени, а вспомогательный глагол стереть. Например, *ayant démontré* переведется «доказав».

5.14 Если вспомогательный глагол в будущем времени совершенного вида, то *participe passé* переводить глаголом будущего времени совершенного вида, приписать ему лицо и число вспомогательного глагола и стереть вспомогательный глагол. Это правило составлено для перевода *futur antérieur*.

5.15 Если вспомогательный глагол стоит в любом другом времени, то *participe passé* переводить глаголом прошедшего времени совершенного вида, число, отрицательность и возвратность взять от вспомогательного глагола, род определять по ближайшему слева существительному, или местоимению в именительном падеже, а вспомогательный глагол стереть.

5.16 Если перед *participe passé* нет глаголов *avoir* или *être* (для глаголов с пометкой о сопряжении с *être*), то оно обрабатывается правилами перевода причастий.

5.2 *Participe présent*.

5.21 *Participe présent* переводится деепричастием настоящего времени, если оно стоит либо в начале фразы, либо после глагола, либо после *en*.

5.22 В остальных случаях *participe présent* обрабатывается по правилам перевода причастий.

6. По предварительным данным получена неопределенная форма совершенного вида.

6.1 Если перед глаголом есть *sans* (может отделяться местоимениями и наречиями), то глагол следует переводить деепричастием настоящего времени, а *sans* переводить словом «не».

6.2 Если перед глаголом в *infinitif* есть глагол *aller* (может отделяться местоимениями и наречиями), то неопределенную форму заменить будущим временем совершенного вида, число, лицо и возвратность взять от глагола *aller*, а глагол *aller* опустить. Это правило составлено для перевода *futur immédiat*. Например, *nous allons faire...* переведется «мы сделаем...».

6.3 Если перед глаголом в *infinitif* есть форма глагола *venir*, которая может отделяться наречиями, местоимениями и предлогом *de*, то неопределенную форму заменить прошедшим временем совершенного вида, число и возвратность взять от глагола *venir*, род определять как в п. 5.15, перед глаголом вставить «только что», глагол *venir*—опустить. Это правило составлено для перевода *passé immédiat*. Например, *nous venons de faire* переведется «мы только что сделали».

7. Определение возвратности.

7.1 Если глагол имеет форму первого лица множественного числа и перед ним стоит *nous nous* или *nous ne nous* (может отделяться наречиями и местоимениями: *le, la, les, en, y*), глагол возвратный.

7.2 Если после глагола первого лица множественного числа есть *nous*, глагол возвратный.

7.3 Если глагол имеет форму первого лица единственного числа и перед ним есть *me* (может отделяться наречиями и местоимениями: *le, la, les, en, y*), глагол возвратный.

7.4 Если перед глаголом не первого лица есть *se* (может отделяться наречиями и местоимениями: *le, la, les, en, y*), глагол возвратный.

8. Определение отрицательности.

Если перед глаголом есть *ne* с переводом «не» (может отделяться местоимениями: *nous, me, le, la, les, lui, leur, en, se, y* и наречиями), глагол отрицательный.

II. Предлог

В обработке нуждаются неоднозначные предлоги: *à, dans, de, en, par, pour, sur*. Правила обработки предлогов включают также обработку местоимений *en, dont*, так как для них нужно установить данные, как для предлога *de*.

При обработке предлогов сначала выясняется, стоит ли предлог перед существительным (местоимением) или перед глаголом в инфинитиве.

1. Если предлог стоит перед инфинитивом глагола, то нужно установить перевод предлога. Рассматриваются различные случаи в зависимости от того, какой это предлог.

1.1 Предлог *pour* переводить «чтобы».

1.2 При наличии предлога *de* проверить наличие перед ним оборота *il s'agit*, если он есть, то переводить «о том, чтобы». В противном случае *de* перед инфинитивом глагола опустить.

1.3 Предлог *par* переводить «тем, что». Инфинитив глагола переделать в глагол того же лица, числа и времени, что и глагол, предшествующий предлогу. Например, *Nous finirons par démontrer ce théorème pour le cas...* переведется: «Мы кончим тем, что докажем эту теорему для случая...».

1.4 При наличии предлога *à* проверить, что стоит слева от него.

1.41 Если непосредственно слева стоит запятая или сочинительный союз, то искать влево другой предлог *à* перед инфинитивом глагола.

1.411 В случае нахождения переводить данный предлог так же, как найденный.

1.412 Если другого предлога *à* нет, то пройти до следующей запятой влево и посмотреть, что стоит перед этой запятой, как если бы эти слова стояли непосредственно перед предлогом.

1.42 При наличии непосредственно слева наречия проверить, какое это наречие.

1.421 Если наречие *jusque*, то предлог *à* переводить «того, чтобы»^{*)}.

1.422 Если наречие *soit*, то *à* переводить «нужно». Например, *soit à démontrer* переведется «пусть нужно доказать».

1.43 При наличии непосредственно слева глагола проверить, какой это глагол.

1.431 Если это один из глаголов: *arriver, conduire, revenir, venir*, то предлог *à* переводить «к тому, чтобы». Например, *Cela nous conduit à faire...* переведется: «Это приводит нас к тому, чтобы сделать...».

1.432 Если это глагол *borner*, то *à* переводить «тем, что». Например, *Nous nous bornerons à démontrer...* переведется: «Мы ограничимся тем, что докажем...».

1.433 Если это глагол *mettre* или *prendre*, то *à* переводить «за то, чтобы». Например, *Nous nous mettrons à démontrer...* переведется: «Мы примемся за то, чтобы доказать...». В этом случае перевод глагола тоже меняется: *prendre* и *mettre* переводятся «приниматься» вместо «брать» и «помещать».

1.434 Если это глагол *assujettir*, то *à* переводить «условию». Например, *...assujettir à être positif* переведется: «...подчинить условию быть положительным».

1.435 Если это глагол *apprendre*, то предлог *à* опустить. Перевод глагола изменить, вместо обычных для математических текстов переводов «узнавать», «сообщить» переводить «учить». Например, *apprendre à calculer* — «учить вычислять».

1.436 Если это глагол *savoir*, то предлог *à* опустить, глагол *savoir* переводить «уметь». Например, *savoir à commencer* — «уметь начать».

1.437 Если это глагол *servir*, то предлог *à* переводить «для того, чтобы». Например, *cette fonction nous servira à construire...* переведется: «эта функция нам послужит для того, чтобы построить...».

1.438 Если глагол *avoir*, то предлог *à* опустить, глагол *avoir* переводить словом «должен» (согласуется с ближайшим влево $S_N (P_N)$). Например, *nous avons à résoudre un problème* переведется: «мы должны решить задачу».

1.439 После остальных глаголов предлог *à* опустить.

1.44 В остальных случаях предлог *à* переводить словом «который», после него вставить слово «нужно». «Который» поставить в винительном падеже и согласовать в роде и числе с ближайшим существительным слева. Например, *le problème à résoudre* переведется: «задача, которую нужно решить».

2. Предлог не перед инфинитивом глагола. В этом случае надо установить перевод предлога и то, какого падежа он требует от последующего существительного.

2.1 Проверить, не является ли этот предлог предлогом *à* или *de*. Если является, то для этих предлогов сначала выполнить частные правила (см. п. 3).

2.2 Для остальных предлогов и для *à* и *de*, если частные правила уже выполнены, найти ближайшее существительное справа и проверить, есть ли у него сильный левый код; если есть, то перевод предлога определяется этим кодом.

^{*)} Переводы предлогов выбраны так, чтобы можно было не менять конструкцию фразы, поэтому иногда они очень необычны.

2.3 Если сильного левого кода нет, то нужно искать правый код.

2.31 Если за предлогом стоит одно из слов: *lequel, laquelle, lesquels, lesquelles, qui, que, quoi*, то искать только предложное управление стоящего справа глагола, так как наличие этих слов указывает на начало придаточного предложения, глагол которого управляет предлогом.

2.32 В остальных случаях искать сначала управляющее слово влево (это может быть глагол, существительное, прилагательное или наречие), искать до конца фразы или до подчинительного союза.

2.321 Если слово с правым кодом найдено, то перевод предлога определяется этим кодом.

2.322 Если слово с кодом влево не найдено, то искать вправо глагол с кодом; найдя глагол, проверить, не следует ли за ним другой глагол (может отделяться наречиями и местоимениями: *le, la, les, lui, leur*); если есть, брать код последнего вправо глагола, за которым не следует другой глагол (т. е. в сочетании *de ce théorème nous pouvons déduire* брать код у *déduire*, а не у *pouvoir*).

2.4 Если слово с правым кодом не найдено, то проверить, есть ли у существительного за предлогом слабый левый код; если есть, то перевод предлога определяется этим кодом.

2.5 Если никакого кода не нашли, то оставить тот основной перевод предлога, который заранее занесен в его информацию.

Во всех случаях, когда управляющий код найден, по номеру этого кода нужно обратиться к соответствующей таблице перевода предлогов, извлечь из нее данные в зависимости от того, какой это предлог, и занести их в информацию предлога.

3. Частные правила для предлогов *à* и *de*.

3.1 Предлог *à* между двумя существительными при отсутствии детерминатива у второго существительного переводить «с» (творит. падеж). Например, *série à membres positifs* переведется как «ряд с положительными членами».

3.2 Предлог *de* между двумя существительными требует родительного падежа второго существительного, а сам опускается, если только он не входит в сочетание *de...à*.

3.3 Предлог *de* после глагола с отрицанием опустить.

3.4 Предлог *de* перед *entre* переводить «из» (родит. падеж), а *entre* опустить. Например, *le premier d'entre eux* переведется «первый из них».

3.5 Если перед предлогом *de* стоит слово со специальной пометкой о *de...à*, то искать вправо предлог *à*, не пропуская слова с кодом на *à*, найдя, переводить *de* как «от» (родит. падеж), *à*—как «до» (родит. падеж). Если указанные условия не выполнены, переводить каждый из предлогов самостоятельно.

4. Правила для местоимений *en* и *dont*.

4.1 Если местоимение *en* стоит не перед *Ppr*, то найти глагол вправо, после которого нет другого глагола, отделенного только наречиями и местоимениями: *le, la, les, leur, lui*,—определить, как переводился бы и какого падежа требовал бы предлог *de*, управляемый этим глаголом, занести этот падеж в информацию местоимения и поставить предлог (с найденным переводом) и местоимение после этого глагола. Например, *nous en avons pris...* переведется «мы взяли из этого».

4.2 Для *dont* тоже искать глагол вправо, за которым не следует другой глагол, отделенный от предыдущего только наречиями или местоимениями: *le, la, les, lui, leur*,—определить, как этот глагол управляет предлогом *de*, предлог *de* вставить перед *dont*, занести ему перевод, определенный по предложному коду глагола, а падеж, которого требует в этом случае предлог *de*, занести в информацию местоимения. Например, *Mainte-*

nant le théorème dont nous avons parlé plus haut peut être démontré. Parlé de переводится «говорить о (pp. п.)», поэтому эта фраза переведется: «Теперь теорема, о которой мы говорили выше, может быть доказана».

5. У ряда слов изменяется перевод в зависимости от того, управляет ли это слово предлогом в переводимой фразе и каким именно. Например, *différent* переводится «различный», а *différent de*—«отличный от (род. п.)», *affirmer* переводится «утверждать», а *affirmer par*—«подтверждать (твор. падеж)». Поэтому, если найдено слово, управляющее предлогом, то надо проверить, не принадлежит ли оно к числу таких слов, и если принадлежит, то изменить у него номер перевода; если прилагательное управляет идущим за ним предлогом, то такому прилагательному ставится запрет перестановки.

Приведем список слов, меняющих перевод в зависимости от предлога:

- 1) *développer* —развертывать
—разлагать (*en*)
- 2) *développable*—развертывающийся
—разложимый (*en*)
- 3) *distinguer* —различать
—отличать (*de*)
- 4) *différent* —различный
—отличный (*de*)
- 5) *dérivable* —дифференцируемый
—выводимый (*de*)
- 6) *combien* —насколько
—сколько (*de*)
- 7) *inférieur* —нижний
—меньший (*à*)
- 8) *attacher* —соединять
—присоединять (*à*)
- 9) *relatif* —относительный
—относящийся (*à*)
- 10) *supérieur* —верхний
—большой (*à*)
- 11) *savoir* —знать
—уметь (*à*)
- 12) *affirmer* —утверждать
—подтверждать (*par*)
- 13) *apprendre* —сообщать, узнавать
—учить (*à + infinitif*)
- 14) *mettre* —помещать
—представлять (*à + S*)
—приниматься (*à + infinitif*).

III. Существительное

Для каждого французского существительного нужно определить число и падеж переводящего его русского существительного.

1. Так как число русского существительного, вообще говоря, совпадает с числом французского, то нужно определить число французского существительного. Для определения числа существительного устанавливают тип образования множественного числа этого существительного.

1.1 Сущ. 1-го типа: нет окончания или есть окончание *-e*—единственное число, есть окончание *-s* или *-es*—множественное число.

Окончание *-e* в единственном числе могут иметь субстантивированные прилагательные, которые отнесены к 1 типу.

1.2 Сущ. 2-го типа: нет окончания—единственное число, есть окончание *-x*—множественное число.

1.3 Сущ. 3-го типа: есть окончание *-l* или *-il*—единственное число, есть окончание *-ix*—множественное число.

1.4 Сущ. 4-го типа: искать влево детерминатив, определять число по нему, если можно; если нет, то взять единственное число.

1.5 Сущ. 5-го типа: число не надо определять, оно фиксировано в информации заранее.

2. В правилах *определения падежа* русского существительного будем обозначать обрабатываемое французское существительное через S_0 .

Для определения падежа искать влево от S_0 другое существительное, глагол, подчинительный союз или слово, приравненное подчинительному союзу (*qui*, *lequel* и др.), прилагательное—детерминатив, предлог, *participe présent*, числительное, сочинительный союз, пропуская прилагательные—не детерминативы, местоимения, наречия, *participe passé*, запяты.

2.1 Нашли прилагательное—детерминатив. Искать влево предлог, сочинительный союз, *participe présent*, пропуская только наречия и прилагательные—детерминативы.

2.11 Если нашли, обращаться к соответствующему случаю (п. 2.2, 2.7, 2.8).

2.12 Если не нашли, то искать влево существительное, глагол, подчинительный союз или слово, приравненное подчинительному союзу, числительное, пропуская прилагательные—не детерминативы, сочинительные союзы, наречия, предлоги, местоимения, запяты, *participe présent* и *participe passé*. Найдя что-либо, обращаться к соответствующему случаю (2.3, 2.4, 2.5, 2.6), не найдя ничего,—к случаю 2.9.

2.2 Нашли предлог (после прилагательного—детерминатива или сразу). Записать существительному падеж, которого требует этот предлог (он записан в информации предлога).

2.3 Нашли числительное. Запомнить, что было числительное, и искать остальные части речи дальше влево; определить падеж; если получили именительный или винительный, то переделать его в зависимости от типа числительного.

2.31 Если числительное кончается на 1 (но не 11), то записать существительному именительный падеж единственного числа.

2.32 Если числительное кончается на 2, 3, 4 (не 12, 13, 14)—родительный падеж, единственное число.

2.33 Если числительное кончается на одну из остальных цифр или 11, 12, 13, 14—родительный падеж, множественное число. Запомнить также первоначальные (до переделки по числительному) число и падеж.

2.4 Нашли подчинительный союз или слово, к нему приравненное. Падеж S_0 —именительный.

2.5 Нашли глагол.

Падеж существительного определяется в зависимости от того, какой это глагол.

2.51 Глагол *être* в настоящем времени—именительный падеж.

2.52 Глагол *être* не в настоящем времени—творительный падеж.

2.53 Глаголы: *devenir*, *paraître*, *s'appeler*, *se nommer*—творительный падеж.

2.54 Остальные непереходные глаголы—именительный падеж.

2.55 Глагол *empêcher*—дательный падеж.

2.56 Глагол *appeler* или *nommer*, и перед существительным есть детерминатив—винительный падеж, перед существительным нет детерминатива—творительный падеж. Сделать у существительного отметку о том, что оно стоит после этих глаголов.

2.57 Глаголы: *supposer*, *laisser*, *rendre*—винительный падеж (с отметкой).

2.58 Остальные переходные глаголы—искать влево от глагола *que*, не пропуская *Cs*.

2.581 Если *que* нет, то в случае утвердительной формы глагола—винительный падеж, в случае отрицательной формы глагола—родительный падеж.

2.582 Если *que* есть, то искать между *que* и глаголом $S_N (P_N)$: если их нет, то падеж S_0 —именительный, если они есть, падеж S_0 , как в п. 2.581.

2.6 Нашли существительное.

Проверить, есть ли перед ним предлог.

2.61 Если предлог есть, проверить, является ли предлог особым (особыми называются предлоги: *entre*, *avec*, *parmi*, которые не повторяются при однородных членах).

2.611 Если предлог особый, то падеж S_0 такой же, как у найденного существительного, и S_0 тоже считать существительным с особым предлогом.

2.612 Если предлог не особый, то такое существительное с предлогом пропустить и искать дальше влево существительное, глагол, подчинительный союз. Найдя, обращаться к соответствующему случаю.

2.62 Если предлога нет—проверить, не является ли S (через S будем обозначать найденное существительное или местоимение) существительным после *appeler*, *nommer*.

2.621 Если является, то при винительном падеже S у S_0 падеж творительный, при творительном падеже S у S_0 падеж винительный.

2.622 Если S —не существительное после *appeler*, *nommer*, то проверить его падеж.

2.6221 Если падеж S именительный, то падеж S_0 тоже именительный.

2.6222 Если падеж S не именительный, но был именительным и переделан из-за числительного, то падеж S_0 именительный.

2.6223 Если падеж S не именительный и не был именительным, проверить, есть ли перед S сочинительный союз (искать, пропуская Δ , Δd , adv).

2.62231 Если есть, то падеж S_0 именительный.

2.62232 Если перед S нет сочинительного союза, искать вправо от S_0 глагол в третьем лице или *participe présent*, которые можно отнести к S_0 (о правилах отнесения см. сноску на стр. 217). Если такой глагол или *participe présent* есть, то падеж S_0 именительный, если их нет, то падеж S_0 такой же, как у предыдущего S без предлога.

2.7 Нашли сочинительный союз (перед детерминативом или непосредственно).

Искать влево от сочинительного союза запятую (непосредственно).

2.71 Если она есть, то искать влево существительное без предлога, не переходя через глагол; найдя существительное без предлога, проверить его падеж.

2.711 Если у него падеж именительный, то падеж S_0 тоже именительный.

2.712 Если падеж не именительный, то искать вправо глагол третьего лица или *participe présent* и дальше, как в 2.62232.

2.72 Если запятой перед сочинительным союзом нет, то искать влево существительное (пропуская прилагательные, наречия и причастия).

2.721 Найдя существительное, проверить, с предлогом ли оно.

2.7211 Если с предлогом, то проверить, не особый ли это предлог.

2.72111 Если предлог особый, то проверить найденное существительное на пометку о правом члене после особого предлога.

2.721111 Если пометки нет, то падеж S_0 такой же, как у найденного существительного; S_0 записать пометку о правом члене после особого предлога.

2.721112 Если найденное существительное имеет пометку, то искать влево существительное без предлога (см. 2.7212).

2.72112 Если предлог не особый, то такое существительное пропустить и искать дальше влево существительное без предлога или с особым предлогом. В зависимости от результата поиска обращаться к случаям 2.7212, 2.72111 или 2.722 соответственно.

2.7212 Если нашли существительное без предлога, то проверить, именительный ли у него падеж.

2.72121 Если именительный, то и падеж S_0 —именительный.

2.72122 Если падеж не именительный, то искать вправо глагол третьего лица или *participe présent* и дальше, как в п. 2.62232.

2.722 Если влево нет существительного, то падеж S_0 —именительный.

Правила п. 2. 6 и п. 2. 7 рассматривают существительное без предлога, которое может быть либо подлежащим, либо прямым дополнением, либо косвенным дополнением, входящим в комбинацию из нескольких существительных, стоящих после предлога, который не повторяется при однородных членах (особые предлоги).

Решение вопроса осложняется тем, что очень трудно установить границы простых предложений в пределах сложного и установить, в каком случае мы имеем дело с однородными членами, а в каком—с новым простым предложением. Для установления того факта, что очередное существительное без предлога есть подлежащее нового простого предложения, мы исходим из следующих предположений: а) в каждом простом предложении должно быть сказуемое (оно может быть выражено либо глаголом третьего лица, либо *participe présent*); б) если имеется несколько однородных членов, то перед последним из них должен стоять сочинительный союз.

Отыскание сказуемого осложнено еще тем, что оно может быть отделено от подлежащего определительным придаточным предложением, в котором есть свое подлежащее и сказуемое, причем в нем может быть даже несколько глаголов, однородных членов, и таких придаточных предложений может быть несколько. Кроме того, между словами, связь которых нужно установить, могут вклиниваться существительные с предлогами, которые также мешают разбору. Правила, сформулированные выше, рассчитаны на правильный перевод в большинстве случаев; однако если в сложном предложении нагромождено слишком много простых предложений и однородных членов—падеж может быть определен неправильно. Заметим, что даже в случае неправильного определения падежа русский перевод может тем не менее оказаться правильным из-за совпадения форм винительного и именительного падежа для большинства существительных (для всех, кроме существительных женского рода на -а, -я).

2.8 Нашли *participe présent*.

Проверить, есть ли непосредственно после него запятая.

2.81 Если нет, то такое *participe présent* рассматривать как глагол из п. 2. 5.

2.82 Если запятая есть, то такое *participe présent* пропустить и искать дальше влево существительное, глагол или подчинительный союз, пропуская остальное. Найдя, обращаться к соответствующему случаю.

2.9 Если ничего не нашли—записать существительному именительный падеж.

IV. Местоимение

Рассматриваются следующие местоимения:

1) *Pronoms personnels*: *je, il, elle, nous, ils, elles, eux, me, le, la, les, lui, leur, se*.

2) *Pronoms adverbiaux*: *en, y*.

3) *Pronoms démonstratifs*: *celui, celle, ce, ceux, celles*.

4) *Pronoms relatifs*: *lequel, laquelle, lesquels, lesquelles, qui, que, dont*.

5) *Pronoms indéfinis*: *on, chacun, personne, rien*.

Остальные местоимения в математических текстах практически не встречаются. Из перечисленных местоимений некоторые отнесены в нашем словаре к другим частям речи, так, *chacun* причислено к прилагательным, *celui, celle, ceux, celles, personne*—к существительным, *se*—к наречиям (так как оно нужно только как показатель возвратности глагола).

Местоимения *en* и *dont* разбираются сначала вместе с предлогами, так как они ведут себя в некотором отношении так, как слова с предлогами, они как бы включают в себя предлог *de*, причем *en* в дальнейшей обработке не нуждается, а сведения о *dont* еще нужно дополнить. Местоимение *y* в пределах рассматриваемых текстов встречается только в оборотах *il y a* (возможна другая форма глагола *avoir*), *y compris* и отдельно не анализируется. Местоимения: *le, la, les, ce*, имеющие омоним—прилагательное, предварительно обрабатываются правилами различения омонимов (см. п. А.10). Из перечисленных местоимений такие, как *je, il, ils, le, la, les, rien* (1 тип), не обрабатываются, так как переводятся однозначно, поэтому число и падеж перевода занесены заранее во французские информации этих местоимений.

Правила определения рода, т. е., иначе говоря, отыскания того слова, которое местоимение заменяет, в настоящее время еще не разработаны, поэтому для *il* дается 3 варианта перевода (он, она, оно), а для *le, la*—2 варианта (его, ее).

Обрабатываемые местоимения разделены на группы, которые обрабатываются по-разному. Эти группы таковы:

1) *nous, me* (II тип);

2) *elle, elles, eux, ce, cela, ceci* (III тип), *lui* (IV тип);

3) *on* (V тип);

4) *qui, que* (VI тип);

5) *lequel, laquelle, lesquels, lesquelles* (VII тип);

6) *dont* (VIII тип).

Рассмотрим правила для каждой из этих групп.

1. Местоимение *nous* может быть подлежащим (например, *nous allons considérer*), и тогда перевод ставится в именительном падеже («мы»); прямым дополнением (например, *cette difficulté ne nous arrêtera pas*), тогда перевод ставится в винительном падеже («нас»); и косвенным дополнением (например, *cette démonstration nous donne le moyen...*), тогда перевод ставится в дательном падеже («нам»).

Me может быть прямым дополнением (например, *Ce théorème ne m'intéresse pas*), тогда перевод ставится в винительном падеже («меня»), и косвенным дополнением (например, *cela me donne*), тогда перевод ставится в дательном падеже («мне»).

1.1 Если местоимение *nous* стоит перед глаголом 1 *pers. pl.* (может отделяться наречиями и местоимениями), то *nous* считается подлежащим, перевод ставится в именительном падеже.

1.2 Если *nous* стоит перед глаголом в другой форме (то же самое для *me*), то надо проверить, переходный ли это глагол.

1.21 Если глагол непереходный, то записать местоимению дательный падеж.

1.22 Если глагол переходный, то проверить, есть ли за ним прямое дополнение.

1.221 Если оно есть, то записать дательный падеж.

1.222 Если его нет, то записать винительный падеж. Последнее правило касается различения случаев: *il nous envoie* и *il nous envoie une lettre*.

2. Для местоимений, входящих в эту группу, число записано в информации заранее: у *eux, elles*—множественное, у остальных—единственное. Перевод *ce, cela, ceci* имеет средний род. Для *elle* и *lui* дается 2 варианта перевода (в 2 родах).

Падеж определяется по следующему правилу.

2.1 Если перед местоимением стоит предлог, то ставить тот падеж, которого этот предлог требует (например, *pour lui*—«для него (нее)», *pour cela*—«для этого», *sans elles*—«без них»).

2.2 Если предлога нет, то у *lui* ставить дательный падеж (ему, ей), у остальных—именительный (*elle*—«он, она, оно», *elles*—«они», *ce, cela, ceci*—«это»).

Правило основано на том, что употребленное без предлога *lui* будет в предложении косвенным дополнением, *elle, elles*—подлежащим, *ce, cela, ceci*—подлежащим или прямым дополнением (переводы в этих двух случаях совпадают).

3. Местоимение *on* (как уже указывалось раньше) переводится нами местоимением «мы», что для математического текста нам кажется удачнее, чем обычный перевод сочетания *on*+V(3 *pers. sing.*) глаголом третьего лица множественного числа без местоимения. Поэтому идущие за этим местоимением глаголы 3 *pers. sing.* (до подчинительного союза или другого подлежащего) переделываются в глаголы первого лица множественного числа.

Например, фраза: *On démontrerait de même la convergence de la série $B = b_0 + b_1 \sin \theta + b_2 \sin 2\theta + \dots + b_n \sin n\theta + \dots$* переведется: «Мы докажем также сходимость ряда $B = b_0 + b_1 \sin \theta + b_2 \sin 2\theta + \dots + b_n \sin n\theta + \dots$ », а не «Докажут также сходимость ряда

$$B = b_0 + b_1 \sin \theta + b_2 \sin 2\theta + \dots + b_n \sin n\theta + \dots$$

4. Местоимения *qui* и *que* переводятся словом «который».

4.1 При переводе *qui* оно ставится в именительном падеже, при переводе *que*—в винительном, так как *qui* выступает в роли подлежащего придаточного предложения, а *que* в роли прямого дополнения.

4.2 Число при переводе *qui* определяется по стоящему справа глаголу. Число при переводе *que* и род для обоих случаев определяются по ближайшему слева существительному, которое не является существительным с *de* без детерминатива (так как такое существительное выполняет функции определения наравне с прилагательными), если только перед местоимением нет сочинительного союза. Если перед местоимением стоит сочинительный союз, то надо искать влево местоимение того же типа и согласовать обрабатываемое местоимение с тем существительным, с которым согласовано предыдущее местоимение.

Например, *Le théorème qui vient d'être établi subsiste dans ces nouvelles conditions* переведется: «Теорема, которая только что была установлена, существует в этих новых условиях».

5. Местоимения: *lequel, laquelle, lesquels, lesquelles*—все переводятся словом «который». Местоимениям *lequel* и *laquelle* приписано заранее единственное число, местоимениям *lesquels* и *lesquelles*—множественное число; нужно определить род и падеж перевода.

5.1 Падеж, если перед местоимением стоит предлог (как это чаще всего бывает), определяется по этому предлогу; при отсутствии предлога в переводе ставится именительный падеж.

5.2 Для определения рода в единственном числе (при множественном числе род определять не нужно) ищется ближайшее существительное влево, которое не является существительным с *de* без детерминатива и с которым местоимение согласуется во французской фразе в роде и числе.

5.21 Если нашли, то перевод местоимения согласуют в роде с переводом этого существительного.

5.22 Если слева такого существительного не нашли, то ищется существительное вправо, но в этом случае согласования может не быть. Если и там нет, то род остается не определенным (в таких случаях при нашей кодировке получается женский род).

Например, *Soit une série $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n + \dots$ dans laquelle les coefficients a_i sont positifs* переведется: «Пусть имеется ряд $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n + \dots$, в котором коэффициенты a_i положительны».

6. Местоимение *dont* переводится словом «который», перед ним может быть вставлен перевод предлога *de*. Нужно определить число, падеж и род в единственном числе.

Ищется вправо от местоимения существительное, субстантивированное прилагательное или другое местоимение (пропуская прилагательные и наречия).

6.1 Справа найдено субстантивированное прилагательное (т. е. имеем конструкцию типа: *dont le premier est positif*)—*dont* переводить «из которых».

6.2 Справа найдено существительное или местоимение. Искать влево существительное (не существительное с *de* без детерминатива) и занести местоимению число и род (в случае единственного числа) такие же, как у этого существительного. Посмотреть, какой глагол стоит вправо от *dont*.

6.21 Если глагол не возвратный, то падеж перевода местоимения—родительный.

6.22 Если глагол возвратный, то перед *dont* при обработке предлогов должен быть вставлен предлог *de* и переведен так, как того требует глагол, стоящий после *dont*. Местоимению заносится падеж, которого требует этот предлог. Если глагол предлогом *de* не управляет, то *dont* заносится родительный падеж.

Правила перестановки слов таковы. Если глагол, найденный вправо от *dont*, переходный, то переставляется прямое дополнение придаточного предложения, присоединяемого *dont*; оно вместе с относящимися к нему прилагательными ставится перед *dont*.

Если глагол непереходный или прямого дополнения нет, то переставляется подлежащее придаточного предложения; оно вместе с относящимися к нему прилагательными ставится перед *dont*.

Например, фраза: *Si a est l'abscisse d'un point double, la droite $x=a$ rencontrera la courbe au point double, dont nous désignerons l'ordonnée par y_1 et en $m-2$ autres points simples dont nous désignerons comme ci-dessus les ordonnées par y_2, \dots, y_{m-1}* переведется: «Если a абсцисса двойной точки,

прямая $x=a$ пересечет кривую в двойной точке, ординату которой мы обозначили через y_1 , и в $m-2$ других простых точках, ординаты которых мы обозначим, как выше, через y_2, \dots, y_{m-1} ».

V. Причастие

A) *Participe passé*

Ищем влево от *participe passé* существительное, местоимение, глагол или сочинительный союз (можно пропускать прилагательные, наречия, *participe passé, participe présent*).

1. *Participe passé* после существительного в именительном падеже.

1.1 При наличии справа глагола в третьем лице, не отделенного другим существительным или местоимением в именительном падеже, с которым первое существительное не являются однородными членами предложения, и в случае, если фраза начинается с *soit (soient)*—*participe passé* переводить страдательным причастием прошедшего времени, согласованным в роде и числе с существительным в именительном падеже.

1.2 При отсутствии указанных в п. 1.1 условий (нет глагола или он отделен, нет *soit (soient)*) *participe passé* переводить страдательным причастием прошедшего времени краткой формы, в роде и числе существительного в именительном падеже. Перед существительным и относящимися к нему прилагательными поставить «после того, как», Например, *Cette définition bien comprise, nous allons montrer que...* переведется: «После того, как это определение вполне понято, мы покажем, что...».

2. Если *participe passé* стоит после существительного не в именительном падеже, то надо посмотреть, каковы число и род *participe passé*.

2.1 Если *participe passé* мужского рода единственного числа—см. п. 2.3, 2.4.

2.2 Если *participe passé* женского рода единственного числа или множественного числа или мужского рода множественного числа, то выполнять правила 2.3, 2.4, если этому не мешает согласование, иначе относить *participe passé* к ближайшему существительному в том же роде и числе, что и *participe passé*, в п. 2.3 искать его дальше влево, в п. 2.4—брать существительное после предлога *de*.

2.3 Если найденное существительное не является существительным с *de* без детерминатива, *participe passé* переводить страдательным причастием прошедшего времени в роде, числе и падеже найденного существительного.

2.4 После существительного с *de* без детерминатива, которому предшествует другое существительное, *participe passé* переводить страдательным причастием прошедшего времени в роде, числе и падеже второго существительного.

3. *Participe passé* после местоимения.

3.1 Если это местоимение *cela*, то *participe passé* переводить страдательным причастием краткой формы среднего рода единственного числа именительного падежа, а перед местоимением вставить «после того, как». Здесь имеется в виду оборот речи типа: *cela fait*, который по указанным правилам переведется «после того, как это сделано».

3.2 После других местоимений переводить так же, как после существительного.

4. *Participe passé* после сочинительного союза.

4.1 Переводить так же, как ближайшее слева *participe passé* (не отделенное глаголом, существительным, местоимением в именительном падеже или подчинительным союзом).

4.2 Если другого *participe passé* нет, то как ближайшее слева прилагательное или *participe présent* (искать, не переходя через глагол, существительное и местоимение в именительном падеже и подчинительный союз).

4.3 Если слева нет ни прилагательного, ни *participe présent*, то переводить как в п. 6.

5. *Participe passé* после глагола.

5.1 Если это глагол *être* в настоящем времени, то *participe passé* переделать в возвратный глагол настоящего времени третьего лица, число определить по ближайшему слева существительному или местоимению в именительном падеже. Глагол *être* при этом опустить.

Например, *Dans ce théorème $f(x,y)$ est considérée comme fonction seulement de x* переведется: «В этой теореме $f(x,y)$ рассматривается как функция только от x », а не «есть рассмотрена», как это было бы по общим правилам (см. ниже).

5.2 Если глагол *être* не в настоящем времени, то *participe passé* переводить страдательным причастием краткой формы, род и число определить по ближайшему слева существительному или местоимению в именительном падеже.

5.3 Если глагол не *être*, то *participe passé* переводить страдательным причастием полной формы творительного падежа, род и число определить, как в предыдущем случае.

6. *Participe passé*, стоящее в начале фразы или после подчинительного союза, переводить в зависимости от того, переходный или непереходный тот глагол, от которого это *participe passé* образовано.

6.1 Если глагол непереходный, то *participe passé* переводится деепричастием прошедшего времени.

6.2 Если глагол переходный, то *participe passé* переводится страдательным причастием прошедшего времени именительного падежа, род и число определяются по ближайшему справа существительному или местоимению в именительном падеже.

Например, *Arrivés à x , nous nous arrêterons...* переведется: «Прибыв в x , мы остановимся...», тогда как *arrêté par...* переведется как «Остановленный...».

Б) *Participe présent*

Искать влево от *participe présent* существительное (может отделяться прилагательными, наречиями, *participe passé*, *participe présent*) или сочинительный союз.

1. *Participe présent* после существительного в именительном падеже.

1.1 При наличии справа глагола в третьем лице, не отделенного другим существительным или местоимением в именительном падеже, которые не являются с первым существительным однородными членами предложения, и в случае, если фраза начинается с *soit* (*soient*), *participe présent* переводить действительным причастием настоящего времени именительного падежа в роде и числе предшествующего существительного.

1.2 При отсутствии вышеуказанных условий (нет глагола или он отделен, нет *soit* (*soient*)) *participe présent* переводить личной формой глагола в третьем лице, число такое же, как у существительного слева, время, как у ближайшего слева или справа глагола. Перед существительным поставить «так как», если слева до конца фразы или до подчинительного союза нет глагола, и «причем», если слева есть другой глагол. Например,

On obtient la relation (1), en posant

$$a_n = \frac{1}{\pi} \int_0^\pi f(\psi) \cos n\psi d\psi, \quad b_n = \frac{1}{\pi} \int_0^\pi f(\psi) \sin n\psi d\psi, \quad f(\psi)$$

désignant la valeur de V sur le contour переводится: «Мы получаем соотношение (1), полагая

$$a_n = \frac{1}{\pi} \int_0^\pi f(\psi) \cos n\psi d\psi, \quad b_n = \frac{1}{\pi} \int_0^\pi f(\psi) \sin n\psi d\psi,$$

причем $f(\psi)$ обозначает значение V на контуре».

2. После существительного не в именительном падеже. Проверить, каковы число и род *participe présent*.

2.1 Если *participe présent* мужского рода единственного числа, см. п. 2.3, 2.4.

2.2 Если *participe présent* женского рода единственного или множественного числа или мужского рода множественного числа, то выполнять правила 2.3, 2.4, если этому не мешает согласование; иначе относить к ближайшему существительному в том же роде и числе, что и *participe présent*, в п. 2.3 искать его дальше влево; в п. 2.4 брать существительное после предлога *de*.

2.3 Если найденное существительное не является существительным с *de* без детерминатива, *participe présent* переводить действительным причастием настоящего времени в роде, числе и падеже предшествующего существительного.

2.4 После существительного с *de* без детерминатива, которому предшествует другое существительное, *participe présent* переводить действительным причастием настоящего времени в роде и числе второго существительного.

3. *Participe présent* после сочинительного союза.

3.1 Переводить так же, как ближайшее слева *participe présent*, не отделенное глаголом, подчинительным союзом, существительным или местоимением в именительном падеже (если предшествующее *participe présent* переделано в глагол, то и это переделать).

3.2 Если *participe présent* слева нет, то переводить, как ближайшее слева *participe passé* или прилагательное, не отделенное глаголом, подчинительным союзом, существительным или местоимением в именительном падеже.

3.3 Если их тоже нет, то переводить деепричастием настоящего времени.

Participe passé и *participe présent*, переводимые причастиями, представляют как прилагательные за исключением случаев, когда это *participe* образовано от переходного глагола и после него есть существительное без предлога.

II раздел

Ниже приводится иная запись алгоритма, точнее, той его части, которая касается различения омонимов и анализа французской фразы. Именно алгоритм записывается в терминах некоторых стандартных простых правил, каждое из которых представляет собой некоторую команду, предписывающую совершить то или иное действие над переводимым текстом или над самими правилами. В каждом из этих простых правил (команд) указывается, какое именно действие следует произвести и над каким

объектом. Объектом действия может быть либо слово (в этом случае нужно указать его номер, о нумерации слов см. ниже), либо некоторые промежуточные данные (сведения, установленные в процессе работы), либо сами простые правила. Следует заметить, что при записи алгоритма в простых правилах мы различаем в слове основу и окончание. Мы считаем, что каждое слово характеризуется определенным набором признаков (информацией), и не интересуемся техникой того, каким образом эта совокупность признаков сопоставлена слову.

Слово имеет обычно одну информацию, но в случае омонимии оно может иметь несколько информации, по одной для каждой части речи, к которой оно может принадлежать.

Простые правила не связаны с реализацией алгоритма в конкретной машине, не связаны со способом записи слов и их признаков в машине.

Алгоритм (или его часть) записывается в виде последовательности простых правил, которые нумеруются. В каждом правиле указывается, к какому правилу следует переходить после выполнения данного. Число правил, к которым можно перейти после выполнения данного, называется числом выходов данного правила, а указанные номера—значениями выходов.

При записи простых правил используются следующие обозначения. Буквой C_γ^β обозначается любое слово, если мы ничего не можем или не хотим говорить о его принадлежности к некоторому классу слов (о значении индексов см. ниже).

Вспомогательные промежуточные данные обозначаются n_i , где $i=1, 2, 3, \dots$ —нумерация по мере их запоминания.

Специальные отметки, которые записываются словам и используются в работе, но не являются окончательными результатами анализа, мы будем обычно обозначать α_j , где $j=1, 2, 3, \dots$ —нумерация для их различения.

В правилах используются две различные нумерации слов.

1. *Абсолютная нумерация* (обозначение C^β , где β —абсолютный номер). Абсолютный номер β есть порядковый номер слова в исходной фразе, он не изменяется в процессе работы.

2. *Относительная нумерация* (обозначение C_γ , где γ —относительный номер слова)—это нумерация, которая вводится правилами поиска по фразе (см. правило 3). Любому слову может быть присвоен любой относительный номер, причем он сохраняется за ним до тех пор, пока тот же номер не будет присвоен некоторому другому слову.

Номер присваивается слову, как графическому комплексу, т. е. если слово может относиться к нескольким частям речи (омонимия), то оно может быть обозначено символом любой из этих частей речи с общим для них всех номером. Такое слово имеет несколько информации по числу частей речи, к которым оно может принадлежать. Например, если слово с омонимией A , S получает номер «2», то мы имеем информации A_2 и S_2 .

В некоторых правилах используем термин «группа S_δ »; под группой существительного подразумеваются относящиеся к существительному прилагательные и причастия с соответствующими им наречиями, не отделенными от S словами, не входящими в группу.

Опишем теперь простые правила.

1. *Проверка по окончанию*.

Запись «проверить C_γ на окончание $-x$ » означает, что надо проверить, есть ли у слова C_γ окончание $-x$. Частный случай этой проверки—проверка на наличие у слова какого бы то ни было окончания.

2. Проверка по признакам.

С помощью этого правила можно проверять наличие самых разнообразных признаков у слов.

Например, можно проверять, принадлежит ли слово к определенному классу слов. Запись «проверить C_γ на V » означает, что надо проверить, не является ли C_γ глаголом.

Можно проверить, принадлежит ли слово к некоторой группе слов, которая указывается в правиле, например, «проверить C_γ на принадлежность к группе: *appeler, nommer*».

Можно проверить наличие у слова определенного признака, либо приписанного ему с самого начала (в словаре), либо выработанного в процессе работы, например, «проверить C_γ на мужской род» или «проверить C_γ на множественное число», «проверить C_γ на α_k » (что означает проверку на наличие пометки α_k).

Можно проверить совпадение некоторого признака или группы признаков у двух слов, например, «проверить C_2 на согласование с C_0 в лице и числе».

Проверки могут касаться не только слов, но и промежуточных данных, например, если n_3 есть запоминание того факта, что во фразе есть глагол в *futur*, то можно записать команду: «проверить по n_3 наличие глагола в *futur*».

3. Поиск по фразе.

В определенном куске фразы ищется слово с определенными признаками. При этом в конкретном правиле должны быть указаны значения следующих параметров: а) место, от которого начинается поиск, б) направление поиска, в) объект поиска, г) условия поиска, д) номер найденного объекта.

Поясним перечисленные параметры и способы их обозначения.

а) Место, от которого начинается поиск.

Указывается номер (относительный или абсолютный) слова, от которого начинается поиск. Можно указать также, что поиск следует начинать с начала фразы.

Если поиск ведется от слова C_0 , то это указание опускается; в начальный момент, когда еще ни одно слово не получило относительный номер ноль, поиск начинается с начала фразы.

б) Направление поиска; пишется: «искать вправо (влево)».

в) Объект поиска.

Если ищется представитель некоторого класса слов, то пишется символ этого класса («искать V »), если ищется некоторое конкретное слово, то пишется это слово («искать *pous*»).

Если ищется представитель некоторой группы слов или классов слов, то пишется «искать $C \in K_1 \vee \dots \vee K_j \vee L_{j+1} \vee \dots \vee L_k$ », где K_i — символы классов слов, к которым может принадлежать искомое, L_i — конкретные слова.

г) Условия поиска.

Перечисляются слова или классы слов, которые можно пропускать при поиске (или, наоборот, те слова и классы слов, которые нельзя пропускать при поиске). Если пропускать ничего нельзя, то пишут «непосредственно», такой поиск сводится к проверке некоторых свойств слова, соседнего с тем, от которого начинается поиск.

Если можно пропускать все, то условия поиска не указываются.

д) Номер найденного объекта.

Указывается, какой относительный номер присваивается найденному слову, пишется «считать $C = C_\gamma$ ». Если искали представителя некоторого класса слов, то найденное слово в дальнейшем обозначается символом

этого класса с тем относительным номером, который ему присвоен («считать $V=V_5$ », в дальнейшем это слово фигурирует как V_5).

Если искомое слово C_γ могло принадлежать к нескольким классам слов, например $C \in A \vee S$, то в дальнейшем эти слова можно обозначать C_γ , а можно обозначать символом того класса, к которому найденное слово принадлежит с присвоенным ему номером, т. е. A_γ или S_γ .

4. Поиск в таблице.

Окончание слова отыскивается в таблице до тех пор, пока не найдется такое окончание, которое целиком содержится в искомом. Информация, которая была записана в таблице при найденном окончании, приписывается тому слову, окончание которого искали.

В правиле указывается номер слова, окончание которого ищется, и номер той таблицы, в которой следует искать. Например, «найти окончание C_γ в таблице № Т».

Правила 1—4 являются альтернативными, поэтому в каждом из них указываются два номера правил; первым указывается номер того правила, к которому следует переходить, если проверяемое условие оказалось выполнено (искомое найдено); вторым указывается номер того правила, к которому следует переходить в противном случае.

5. Запись.

С помощью правила записи можно осуществлять запись любых данных, выработанных в процессе анализа. В правиле следует указать значения следующих параметров: а) место записи, б) предмет записи.

Поясним эти параметры и способы их обозначения.

а) Место записи.

Если записывается некоторый признак какому-то слову, то следует указать, какому именно («записать C_3 »).

Если записываются некоторые промежуточные данные, то указывается, какой номер им при этом присваивается, пишут «записать в n_2 ».

б) Предмет записи.

Предмет записи может быть указан двумя способами: а) непосредственно («записать S_1 именительный падеж», «записать n_1 место V_5 », т. е. абсолютный номер V_5 и др.), б) косвенно («записать A_4 падеж S_2 », т. е. записать A_4 тот падеж, который записан у слова S_2 , и т. п.).

6. Выбор одной из словарных информации.

При различении омонимов из нескольких имеющихся у слова информации некоторые стираются. Различные информации слова даны для различных частей речи. Если слово C_1 могло быть либо прилагательным, либо существительным, то при выполнении правила «стереть A_1 » информация прилагательного сотрется, информация существительного останется.

7. Вставка слова.

По этому правилу можно вставить слово, причем следует указать место вставки (номер слова, перед которым или после которого вставляется) и какое слово следует вставить. Например, «вставить перед V_2 adv с переводом «только что».

8. Перестановка слов.

С помощью этого правила можно переставлять слова или группы слов, причем в правиле указывается, что именно надо переставлять (слово C_γ или группу слова S_δ) и место, на которое ставится (поставить перед C_γ , после группы S_δ).

9. Выбор информации из таблицы.

Это правило служит для выбора из таблицы записанных в ней данных, причем следует указывать номер таблицы, из которой производится выбор, и место в таблице.

Номер таблицы и место в таблице могут быть указаны как непосредственно («взять из таблицы № 3 вторую строку»), так и косвенно («взять из таблицы, номер которой равен предложному коду C_3 , данные, соответствующие предлогу Pr_0 »).

Взятые из таблицы данные переносятся в некоторое специальное стандартное место, откуда они могут быть взяты и записаны любому слову.

В правилах 5—9 указывается номер правила, к которому следует переходить после выполнения данного.

10. П р а в и л о в е т в л е н и я.

Правило ветвления применяется для сокращения числа проверок там, где происходит разветвление алгоритма на несколько путей в зависимости от числового значения некоторого признака слова (типа местоимения, группы спряжения глагола и др.). Такое разветвление можно осуществлять с помощью последовательности проверок («проверить C_1 на тип 1», «проверить C_1 на тип 2» и т. д.), но ветвление проще.

В правиле ветвления указывается номер слова и признак, по которому происходит ветвление (или номер промежуточного результата, по которому также можно устроить ветвление). Например, «ветвление по типу образования множественного числа S_0 ».

Номера правил, к которым следует переходить после данного, указываются следующим образом: на первом месте пишется номер правила, к которому переходим, если признак имеет числовое значение 1, на втором — номер правила, к которому переходим, если он имеет значение 2, и т. д.

Число выходов правила ветвления определяется числом возможных значений признака, по которому идет ветвление.

11. П р а в и л о п е р е д е л к и п р а в и л.

Объектом действия этого правила в отличие от всех остальных является алгоритм перевода, а не переводимый текст. Указывается номер изменяемого правила и то, какую переделку в нем надо сделать, например, «переделать 5 на поиск от S_2 », «переделать 6, поставив выходы (1,8)».

Различные случаи разбора омонимов объединены в несколько групп. Случаи, находящиеся в одной группе, обрабатываются за один просмотр фразы. Различные группы обрабатываются последовательно, т. е. каждый раз вся фраза просматривается снова, причем правила некоторой группы опираются на то, что предшествующие группы уже разобраны. Поэтому порядок, в котором рассматриваются различные случаи, здесь несколько иной, чем в 1 разделе.

ПРАВИЛА РАЗЛИЧЕНИЯ ОМОНИМОВ

1	1(2, II)	Искать выраво C с омонимами, считать $C = C_0$.
	2(3, 1)	Проверить C_0 на наличие V среди омонимов.
	3(4, 61)	Проверить C_0 на наличие S среди омонимов.
	4(118, 5)	Проверить C_0 на наличие A среди омонимов.
	1. 5(6, 44)	Проверить C_0 на принадлежность к группе VS_1 .
	6(7, 10)	Проверить C_0 на наличие окончания.
	7(8, 9)	Проверить C_0 на окончание $-s$.
	8(1)	Стереть V_0 .
	9(1)	Стереть S_0 .
	10(11, 9)	Искать влево $C \in Ad \vee Pr$, пропуская A , adv . считать $C = C_1$.
	11(12, 16)	Проверить C_1 на Ad .
	12(9, 13)	Проверить C_1 на les .
	13(14, 8)	Проверить C_1 на принадлежность к группе le, la .
	14(15, 9)	Проверить C_0 на согласование с C_1 в роде.
	15(8, 17)	Искать от C_1 влево Pr , пропуская Ad .
	16(17, 8)	Проверить C_1 на en .
	17(42, 18)	Проверить C_1 на $soit$.
	18(19, 8)	Искать влево $C \in S \vee P$, считать $C = C_2$.
	19(20, 8)	Проверить C_2 на единственное число.

- 20(21,22) Искать от C_2 влево $C \in Pr \vee V$, пропуская A , Ad , adv .
 21(18) Переделать 18 на поиск от C_2 .
 22(23,9) Искать от C_2 вправо V , считать $V = V_3$.
 23(31,24) Проверить V_3 на совпадение с V_0 .
 24(26,25) Проверить V_3 на отметку об отнесении α_1 .
 25(27,26) Проверить V_3 на согласование с C_2 в лице.
 26(25,9) Искать от V_3 вправо V , считать $V = V_3$.
 27(28,30) Проверить V_3 на единственное число.
 28(29,26) Проверить C_2 на единственное число.
 29(18) Записать V_3 отметку об отнесении α_1 .
 30(18,29) Искать от C_2 влево $C \in Cs \vee Z$, пропуская A , Ad , adv .
 31(32,9) Искать от C_1 влево $C \in Cs \vee Z$, считать $C = C_4$.
 32(33,9) Искать вправо V , считать $V = V_5$.
 33(35,34) Проверить V_5 на множественное число.
 34(32) Переделать 32 на поиск от V_5 .
 35(36,8) Искать от V_5 влево $C \in S \vee P$, считать $C = C_6$.
 36(8,38) Проверить C_6 на совпадение с C_0 .
 37(41,38) Проверить C_6 на отметку об отнесении α_2 .
 38(40,39) Искать от C_6 влево Cs , пропуская A , Ad , adv .
 39(32) Записать C_6 пометку об отнесении α_2 .
 40(39,8) Проверить C_5 на Cs .
 41(36,8) Искать от C_6 влево $C \in S \vee P$, считать $C = C_6$.
 42(8,43) Искать вправо S , пропуская A , Ad , adv .
 43(8,9) Искать влево *que*, не пропуская V , Cs .
 44(45,46) Проверить C_0 на принадлежность к группе VS_2 .
 45(9,1) Искать влево V (*avoir*), пропуская adv .
 46(47,59) Проверить C_0 на принадлежность к группе: *fait, produit*.
 47(48,51) Проверить C_0 на наличие окончания.
 48(49,50) Проверить C_0 на окончание *-s*.
 49(1) Записать V_0 информацию *participe passé*.
 50(49) Стереть S_0 .
 51(50,52) Искать влево $C \in V$ (*avoir* \vee *être*), пропуская adv .
 52(11,53) Искать влево $C \in Ad \vee Pr$, пропуская A , adv , считать $C = C_1$.
 53(56,54) Искать влево S , пропуская A , adv .
 54(55) Записать V_0 показатель $V_{3л}$.
 55(1) Стереть S_0 .
 56(49,57) Проверить C^1 на *soit*.
 57(18) Переделать 22, 26: вместо отсылки к 9 поставить отсылку к 58.
 58(54) Переделать 22, 26: вместо отсылки к 58 поставить отсылку к 58.
 59(60,8) Проверить C_0 на наличие окончания.
 60(8,9) Проверить C_0 на окончание *-s*.
 2. 61(62,68) Проверить C_0 на наличие среди омонимов Pr .
 62(63,66) Проверить C_0 на *entre*.
 63(65,64) Искать вправо $C \in A \vee Ad \vee \Phi \vee N$.
 64(1) Стереть Pr_0 .
 65(1) Стереть V_0 .
 66(67,68) Проверить C_0 на принадлежность к группе: *par, pour*.
 67(64,65) Проверить C_0 на наличие окончания.
 3. 68(69,84) Проверить C_0 на наличие среди омонимов adv .
 69(70,83) Проверить C_0 на *soit*.
 70(71,73) Проверить C_0 на C^1 .
 71(72) Записать место adv_0 в n_1 , отметить его \tilde{C}_0 .
 72(1) Стереть V_0 .
 73(74,78) Искать влево $C \in Cs \vee Cs \vee T$, пропуская adv , считать $C = C_1$.
 74(72,75) Проверить C_1 на Cs .
 75(76,71) Проверить по n_1 наличие \tilde{C}_0 .
 76(77) Записать C_0 перевод «либо».
 77(72) Записать \tilde{C}_0 перевод «либо».
 78(79,82) Искать влево Z непосредственно, считать $Z = Z_2$.
 79(80,82) Искать от Z_2 влево V , не пропуская Z , Cs , считать $V = V_3$.
 80(81,75) Проверить V_3 на *participe*.
 81(79) Переделать 79 на поиск от V_3 .
 82(1) Стереть adv_0 .
 83(82,72) Проверить C_0 на наличие окончания.
 4. 84(111,85) Проверить C_0 на A .
 85(86,87) Проверить C_0 на наличие окончания.
 86(1) Записать V_0 показатель *participe passé*.
 87(86,88) Искать влево $C \in V$ (*avoir, être*), пропуская adv .

- 88(86,89) Проверить C^1 на *soit*.
89(90,101) Искать влево $C \in S \vee P$, считать $C = C_1$.
90(91,92) Искать от C_1 влево $C \in Pr \vee V$, пропуская A , Ad , adv .
91(89) Переделать 89 на поиск от C_1 .
92(93,89) Проверить C_1 на единственное число.
93(94,101) Искать от C_1 вправо V , считать $V = V_2$.
94(98,95) Проверить V_2 на совпадение с V_0 .
95(86,97) Проверить V_2 на единственное число.
96(93) Переделать 93 на поиск от V_2 .
97(89,96) Проверить V_2 на согласование с C_1 в лице.
98(99,101) Искать вправо $C \in V \vee S \vee P \vee Cs$, считать $C = C_3$.
99(100,103) Проверить C_3 на Cs .
100(102,101) Проверить C_3 на принадлежность к группе: *que, qui, dont, où, lequel, laquelle, lesquels, lesquelles*.
101(1) Записать C_0 показатель V .
102(98) Переделать 98 команду на поиск от C_3 .
103(106,104) Проверить C_3 на V .
104(105) Искать от C_3 влево Cs , пропуская A , Ad , adv .
105(102) Записать в n_2 наличие $C_3 = S \vee P$.
106(108,107) Проверить по n_2 наличие $S \vee P$.
107(86,98) Проверить V_3 на согласование с C_1 в лице.
108(98,109) Проверить V_3 на согласование с данными в n_2 лице и числе.
109(110,107) Проверить V_3 на согласование с n_2 в лице.
110(86,98) Проверить V_3 на множественное число.
5. 111(116,112) Проверить C_0 на *fixe, alterne*.
112(113,8) Проверить C_0 на наличие окончания.
113(8,114) Проверить C_0 на окончание *-s*.
114(8,115) Проверить C_0 на окончание *-es*.
115(89,126) Проверить C_0 на окончание *-e*.
116(117,89) Проверить C_0 на наличие окончания.
117(8,126) Проверить C_0 на окончание *-s*.
6. 118(119,127) Проверить C_0 на *continu*.
119(120,8) Проверить C_0 на наличие окончания.
120(121,122) Проверить C_0 на окончание *-es*.
121(8) Стереть S_0 .
122(8,123) Проверить C_0 на окончание *-s*.
123(124,125) Проверить C_0 на окончание *-e*.
124(89) Стереть S_0 .
125(126) Стереть S_0 .
126(1) Стереть A_0 .
127(128,132) Проверить C на наличие окончания.
128(129,130) Проверить C_0 на окончание *-s*.
129(1) Стереть V_0 .
130(131) Стереть S_0 .
131(1) Стереть A_0 .
132(133,134) Искать влево *être*, пропуская adv .
133(129) Стереть S_0 .
134(18) Переделать 22,26: вместо отсылки к 9 поставить отсылку к 135.
135(130) Переделать 22,26: вместо отсылки к 135 поставить отсылку к 9.
II 22. 1(2,III) Искать вправо Φ , считать $\Phi = \Phi_0$.
2(3,4) Искать влево S , пропуская A , adv , Pr , Ppr .
3(1) Записать Φ_0 информацию наречия.
4(5,6) Искать влево Cs непосредственно, считать $Cs = Cs_1$.
5(3,6) Искать от Cs_1 влево Φ (наречие), пропуская A , adv .
6(1) Записать Φ_0 информацию существительного.
III 7. 1(2,IV) Искать вправо C с омонимами, считать $C = C_0$.
2(3,6) Проверить C_0 на *en*.
3(5,4) Искать вправо $C \in en \vee V$, пропуская adv .
4(1) Стереть P_0 .
5(1) Стереть Pp_0 .
8. 6(7,10) Проверить C_0 на принадлежность к группе: *pas, point*.
7(8,9) Искать влево $C \in V \vee ne$ непосредственно.
8(1) Стереть S_0 .
9(1) Стереть adv_0 .
10(11,14) Проверить C_0 на *ensemble*.
11(9,12) Проверить S_0 на наличие окончания.
12(13,8) Искать влево $C \in Ad \vee Pr$, пропуская A , считать $C = C_1$.
13(8,9) Проверить C_1 на принадлежность к группе: *la, les*.
9. 14(15,25) Проверить C_0 на *suivant*.

- 15(16,17) Проверить C_0 на наличие окончания.
 16(1) Стереть Pr_0 .
 17(18,16) Искать вправо Ad непосредственно.
 18(19,24) Искать влево S , не пропуская V , считать $C=C_1$.
 19(21,20) Проверить C_1 на м. род ед. ч.
 20(18) Переделать 18 на поиск от C_1 .
 21(22,24) Искать от C_1 влево Pr , пропуская Ad , A , adv , считать $Pr=Pr_2$.
 22(23,24) Искать от Pr_2 влево V , считать $V=V_3$.
 23(16,24) Проверить V_3 на переходность.
 24(1) Стереть A_0 .
 10. 25(26,29) Проверить C_0 на *se*.
 26(27,28) Искать вправо V , пропуская adv .
 27(1) Стереть A_0 .
 28(1) Стереть P_0 .
 29(30,1) Проверить C_0 на принадлежность к группе *le, la, les*.
 30(32,31) Искать вправо V , пропуская P (*lui, leur, en, y*).
 31(32,33) Искать вправо $C \in C_s \vee C_s \vee T \vee Z$, пропуская adv (не *plus, moins*).
 32(1) Стереть A_0 .
 33(1) Стереть P_0 .
 IV 11. 1(2,V) Искать вправо C с омонимами, считать $C=C_0$.
 2(3,16) Проверить C_0 на *même*.
 3(6,4) Искать влево $C \in S \vee P \vee Ad \vee Pr$, считать $C=C_1$.
 4(5) Записать adv_0 перевод «даже».
 5(1) Стереть A_0 .
 6(7,11) Проверить C_1 на Pr (*de*).
 7(8,10) Искать от C_1 вправо S непосредственно.
 8(9) Записать A_0 перевод «тот же самый».
 9(1) Стереть adv_0 .
 10(5) Записать adv_0 перевод «также».
 11(12,15) Проверить C_1 на $Ad \vee Pr$.
 12(13,14) Проверить C_1 на принадлежность к группе: *un, une*.
 13(9) Записать A_0 перевод «один и тот же».
 14(9) Записать A_0 перевод «тот же самый».
 15(9) Записать A_0 перевод «сам».
 12. 16(17,26) Проверить C_0 на *plus*.
 17(18,20) Искать влево Ad .
 18(19) Стереть adv_0 .
 19(1) Стереть Cs_0 .
 20(21,25) Искать влево $C \in S \vee \Phi$, пропуская A , adv , считать $C=C_1$.
 21(25,22) Искать от C_1 влево Pr , пропуская A , Ad , adv .
 22(23,25) Искать вправо $C \in S \vee \Phi$, пропуская Ad .
 23(24) Стереть adv_0 .
 24(1) Стереть A_0 .
 25(19) Стереть A_0 .
 13. 26(27,31) Проверить C_0 на *si*.
 27(28,30) Искать вправо $C \in A \vee adv$, считать $C=C_1$.
 28(29,30) Искать от C_1 вправо $C \in Z \vee que \vee T$.
 29(1) Стереть Cs_0 .
 30(1) Стереть adv_0 .
 14. 31(32,1) Проверить C_0 на принадлежность к группе: *vite, vrai, haut*.
 32(33,34) Искать влево V , пропуская adv .
 33(1) Стереть A_0 .
 34(1) Стереть adv_0 .
 V 15. 1(2,VI) Искать вправо C (*que*), считать $C=C_s$.
 2(3,6) Проверить C_0 на C^1 (начало фразы).
 3(4) Записать adv_0 перевод «пусть».
 4(5) Стереть P_0 .
 5(1) Стереть Cs_0 .
 6(7,16) Искать влево $C \in même \vee plus \vee aussi \vee moins \vee moindre \vee \vee tel$ (с формами), пропуская S , A , Ad , adv , считать $C=C_1$.
 7(8,9) Проверить C_1 на *même*.
 8(4) Записать adv_0 перевод «что и».
 9(10,13) Проверить C_1 на принадлежность к группе: *tel, telle, tels, telles*.
 10(11) Записать Cs_0 перевод «что».
 11(12) Стереть P_0 .
 12(1) Стереть adv_0 .
 13(14,15) Проверить C_0 на *aussi*.
 14(11) Записать Cs_0 перевод «чем».

	15(14)	Записать adv_0 перевод «как».
	16(17,21)	Искать влево $C(ne)$, не пропуская Cs , Cs , \exists , считать $C=C_1$.
	17(21,18)	Искать от C_1 вправо $C \in S (personne, aucun) \vee \text{adv} (rien, pas, jamais, point)$.
	18(21,19)	Искать от C_1 влево $C \in S (personne, aucun) \vee \text{adv} (rien, jamais)$.
	19(20)	Записать C_1 нулевой перевод.
	20(4)	Записать adv_0 перевод «только».
	21(22,29)	Искать влево $C \in Cs \vee \exists$, пропуская adv .
	22(23,29)	Искать влево $C(que)$, считать $C=C_1$.
	23(24,25)	Проверить C_1 на adv .
	24(4)	Записать adv_0 перевод adv_1 .
	25(26,28)	Проверить C_1 на P .
	26(27)	Записать P_0 перевод P_1 .
	27(12)	Стереть Cs_0 .
	28(11)	Записать Cs_0 перевод Cs_1 .
	29(30,10)	Искать влево $C \in S \vee V$, считать $C=C_1$.
	30(31,33)	Проверить C_1 на V .
	31(10,32)	Проверить C_1 на вхождение в список слов с пометкой о que .
	32(11)	Записать Cs_0 перевод «что», «чтобы».
	33(35,27)	Искать от C_1 влево Pr , пропуская A , Ad , adv .
	34(35,10)	Искать вправо V , считать $V=V_2$.
	35(35,36)	Искать от V_2 вправо V , пропуская adv , P , считать $V=V_2$.
	36(38,37)	Проверить V_2 на переходность.
	37(10,32)	Проверить V_2 на вхождение в список слов с пометкой о que .
	38(39,32)	Искать от V_2 влево $C \in S \vee P$, не пропуская que , считать $C=C_4$.
	39(40,41)	Искать от C_4 влево Pr , пропуская Ad , A , adv .
	40(38)	Переделать 38 на поиск от C_4 .
	41(42,27)	Искать от V_2 вправо $C \in S \vee P$, не пропуская V , Cs , считать $C=C_3$.
	42(43,32)	Искать от C_3 влево Pr , пропуская Ad , A , adv .
	43(41)	Переделать 41 на поиск от C_3 .
VI	16. 1(2,30)	Искать вправо $C \in A \vee Ad$, считать $C=C_0$.
	2(3,14)	Проверить C_0 на Ad .
	3(4,7)	Искать вправо S , пропуская A , Ad , adv , N , считать $S=S_1$.
	4(5)	Записать C_0 расстояние $\rho=\rho(C_0, S_1)$.
	5(6,1)	Проверить C_0 на наличие омонимии.
	6(1)	Стереть S_0 .
	7(8,9)	Проверить C_0 на наличие омонимии.
	8(11)	Стереть A_0 .
	9(10,13)	Проверить C_0 на отметку о субстантивации.
	10(11)	Записать C_0 информацию существительного.
	11(12,1)	Искать влево $C \in A \vee Ad$ с отметкой α_1 , пропуская A , Ad , N , adv , считать $C=C_2$.
	12(1)	Записать C_2 расстояние $\rho=\rho(C_2, C_0)$.
	13(1)	Записать C_0 отметку α_1 (о том, что справа нет S).
	14(25,15)	Искать влево Cs , пропуская adv .
	15(16,25)	Искать влево $C \in S \vee V \vee Pr \vee N \vee Ad$, пропуская A , Pr , Ppr , adv , считать $C=C_3$.
	16(17,23)	Проверить C_3 на S .
	17(18,19)	Проверить C_0 на согласование с C_3 .
	18(5)	Записать C_0 расстояние до C_3 , $\rho=\rho(C_0, C_3)$.
	19(20,28)	Искать от C_3 влево S , пропуская A , adv , Pr , Ppr , $Pr(de)$, считать $S=S_4$.
	20(21,22)	Проверить C_0 на согласование с S_4 .
	21(5)	Записать C_0 расстояние $\rho=\rho(C_0, S_4)$.
	22(19)	Переделать 19 на поиск от S_4 .
	23(5,24)	Проверить C_3 на V .
	24(4,7)	Искать вправо S , пропуская A , adv , считать $S=S_1$.
	25(26,28)	Искать влево $C \in A \vee Pr \vee Ppr$, считать $C=C_2$.
	26(27,25)	Проверить C_0 на согласование с C_2 .
	27(5)	Записать C_0 расстояние $\rho=\rho(C_0, C_2) + \rho(C_2, Y(C_2))$ *).
	28(29,7)	Искать влево S , считать $S=S_1$.
	29(4,28)	Проверить C_0 на согласование с S_1 .
	30(31, VII)	Искать вправо $C \in A$ (с отм. α_1) $\vee Ad$ (с отм. α_1), считать $C=C_0$.
	31(30)	Записать C_0 информацию существительного.

*) Через $Y(C_2)$ обозначено слово, управляющее словом C_2 .

- VII 17. 1(2,VIII) Искать вправо S с омонимами, считать $S = C_0$.
 2(3,1) Проверить C_0 на омонимию второго типа (многозначность).
 3(4,10) Проверить C_0 на *tout*.
 4(5,1) Искать вправо $C \in S \vee Ad$ непосредственно, считать $C = C_1$.
 5(6,7) Проверить C_1 на S .
 6(1) Записать C_0 перевод «всякий».
 7(8,9) Проверить C_1 на принадлежность к группе *le, la, les*.
 8(1) Записать C_0 перевод «весь».
 9(1) Записать C_0 перевод «целый».
 18. 10(11,13) Проверить C_0 на *infinité*.
 11(12,1) Искать вправо *de*, пропуская A .
 12(1) Записать C_0 перевод «бесконечное число».
 19. 13(14,16) Проверить C_0 на принадлежность к группе *est, sont*.
 14(1,15) Искать вправо Ad непосредственно.
 15(1) Записать C_0 нулевой перевод.
 20. 16(17,19) Проверить C_0 на *moins*.
 17(18,1) Искать влево $C \in le \vee la \vee les$ непосредственно.
 18(1) Записать C_0 перевод «наименее».
 19(20,22) Проверить C_0 на *étant*.
 20(21,1) Искать вправо V .
 21(1) Записать C_0 перевод «когда».
 21. 22(23,1) Проверить C_0 на принадлежность к группе: *un, une*.
 23(24,25) Искать влево оборот *à partir de* непосредственно.
 24(1) Записать C_0 перевод «некоторый».
 25(26,28) Искать влево Cs непосредственно.
 26(27,28) Искать влево C (*un*), пропуская Cs , считать $C = C_1$.
 27(1) Записать C_0 перевод C_1 .
 28(29,30) Искать влево *le* непосредственно.
 29(1) Записать C_0 перевод «один».
 30(31,29) Искать вправо S , пропуская A , adv , считать $S = S_1$.
 31(29,1) Искать от S_1 вправо обороты *au plus, au moins*, пропуская A , adv .
 VIII 23. 1(2,13) Искать вправо V , считать $V = V_0$.
 2(3,1) Проверить V_0 на *t-i*.
 3(4,5) Искать влево P (*le, la, les*), пропуская adv , V (*avoir*).
 4(1) Записать V_0 показатель переходного глагола.
 5(6,7) Искать вправо $C \in S \vee P \vee Pr \vee V \vee Cs$, считать $C = C_1$.
 6(7,8) Проверить C_1 на V .
 7(1) Записать V_0 показатель непереходного глагола.
 8(7,9) Проверить C_1 на Cs .
 9(10,4) Проверить C_1 на Pr .
 10(11,12) Проверить C_1 на *de*.
 11(4,12) Искать влево *ne*, пропуская adv , V .
 12(5) Переделать 5 на поиск от C_1 .
 13 Останов.

ПРАВИЛА АНАЛИЗА

1°. Глагол

Правила окончательной обработки

- 1(2,96) Искать вправо V , считать $V = V_0$.
 2(3,4) Проверить V_0 на *supposer*.
 3(4) Записать предварительные данные V_0 в n_1 .
 4(10,5) Проверить предварительные данные V_0 на настоящее время.
 5(26,6) Проверить предварительные данные V_0 на будущее время.
 6(31,7) Проверить предварительные данные V_0 на прошедшее время несовершенного вида.
 7(33,8) Проверить предварительные данные V_0 на прошедшее время несовершенного вида + «бы».
 8(38,9) Проверить предварительные данные V_0 на *participe passé*.
 9(61,56) Проверить предварительные данные V_0 на неопределенную форму.
 10(11,16) Проверить V_0 на первое лицо множественного числа.
 11(79,12) Искать влево *vous*, пропуская adv , P .
 12(13,15) Искать влево Cs , пропуская adv , P .
 13(14,81) Проверить по n_2 наличие ранее переделанного глагола.
 14(81) Записать V_0 будущее время совершенного вида.
 15(14) Записать в n_2 , что была переделка настоящего времени на будущее.
 16(17,20) Искать влево *que* («чтобы»), пропуская adv , P .

- 17(18) Записать V_0 прошедшее время совершенного вида.
 18(19,77) Проверить V_0 на единственное число.
 19(83) Записать в n_3 место глагола прошедшего времени для определения рода.
 20(21,22) Искать влево Cs (*si*), пропуская adv, A, N, Pr, Ppr, P, S, Z.
 21(22) Запомнить в n_4 место глагола в потенциальном будущем.
 22(23,77) Проверить V_0 на четвертое лицо.
 23(24,25) Искать влево *je*, пропуская adv, P.
 24(84) Записать V_0 первое лицо.
 25(85,3) Записать V_0 третье лицо.
 26(27,28) Проверить V_0 на окончательность.
 27(77) Записать в n_5 , что был глагол в *futur*.
 28(30,29) Искать влево Cs (*si*).
 29(30,27) Искать вправо Cs (*si*).
 30(18) Записать V_0 прошедшее время несовершенного вида + «бы».
 31(32,18) Искать Cs (*si*) влево, пропуская A, Ad, Pr, Ppr, adv, P, Z, S, N.
 32(18) Записать V_0 прошедшее время совершенного вида + «бы».
 33(34,35) Искать влево *que* («чтобы»), пропуская adv, P, Z.
 34(18) Записать V_0 прошедшее время несовершенного вида.
 35(36,18) Проверить по n_1 , был ли впереди глагол *supposer*.
 36(37) Записать V_0 время и вид глагола *supposer* из n_1 .
 37(18,77) Проверить V_0 на прошедшее время.
 38(39,41) Проверить V_0 на спряжение с *être*.
 39(42,40) Искать влево V (*être*), пропуская adv, P, считать $V = V_1$.
 40(77) Записать в n_6 место причастия для обработки правилами для причастий.
 41(42,40) Искать влево $V \in avoir \vee s'être$, пропуская P, adv, считать $V = V_1$.
 42(43,49) Проверить V_1 на *infinitif*.
 43(44,48) Искать от V_1 влево Pr (*après*), пропуская P, считать $Pr = Pr_2$.
 44(45) Записать Pr_2 нулевой перевод.
 45(46) Записать V_0 деепричастие прошедшего времени.
 46(47) Записать V_0 отрицательность и возвратность V_1 .
 47(65) Записать V_1 нулевой перевод.
 48(45,53) Искать от V_1 влево Pr (*sans*), пропуская P.
 49(45,50) Проверить V_1 на *participe présent*.
 50(51,53) Проверить V_1 на будущее время.
 51(52) Записать V_0 будущее время совершенного вида.
 52(47) Записать V_0 лицо, число, отрицательность, возвратность V_1 .
 53(54) Записать V_0 прошедшее время совершенного вида.
 54(55) Записать V_0 число, возвратность, отрицательность V_1 .
 55(18) Записать V_1 нулевой перевод.
 56(57,58) Проверить V_0 на отметку о деепричастии.
 57(76) Записать V_0 деепричастие настоящего времени.
 58(57,59) Искать начало фразы (влево), пропуская adv, Cc.
 59(57,60) Искать влево $C \in V \vee Cs \vee en$, пропуская adv, P.
 60(78) Записать в n_7 место причастия для обработки правилами для причастий.
 61(62,66) Искать влево Pr (*sans*), пропуская adv, P, считать $Pr = Pr_1$.
 62(63) Записать V_0 деепричастие настоящего времени.
 63(64) Записать V_0 отрицательность.
 64(65) Записать Pr_1 перевод «не».
 65(78) Записать V_0 пометку α_1 о том, что не надо определять отрицательность.
 66(67,70) Искать влево V (*aller*), пропуская adv, P, считать $V = V_2$.
 67(68) Записать V_0 будущее время совершенного вида.
 68(69) Записать V_0 лицо, число, отрицательность и возвратность V_2 .
 69(65) Записать V_2 нулевой перевод.
 70(71,78) Искать влево V (*venir*) пропуская adv, P, *de*, считать $V = V_2$.
 71(72,78) Искать вправо от V_2 Pr (*de*) непосредственно, считать $Pr = Pr_2$.
 72(73) Записать V_0 прошедшее время совершенного вида.
 73(74) Записать V_0 число, отрицательность и возвратность V_2 .
 74(75) Вставить перед V_0 «только что».
 75(78) Записать V_2 нулевой перевод.
 76(18) Записать Pr_2 нулевой перевод.
 77(86,78) Проверить V_0 на наличие возвратности (пометка перенесена от другого глагола).
 78(79,83) Проверить V_0 на первое лицо множественного числа.
 79(80,81) Искать влево C (*nous*), пропуская adv, P (*le, la, les, en, y*), считать $C = C_3$.
 80(82,71) Искать влево от C_3 P (*nous*), пропуская *ne*, считать $P = P_4$.
 81(82,92) Искать вправо C (*nous*) непосредственно, считать $C = C_3$.

82(86)	Записать V_0 возвратность.
83(84,85)	Проверить V_0 на первое лицо единственного числа.
84(82,99)	Искать влево C (<i>me</i>), пропуская <i>adv</i> , P (<i>le, la, les, en, y</i>), считать $C = C_3$.
85(82,92)	Искать влево C (<i>se</i>), пропуская <i>adv</i> , P (<i>le, la, les, en, y</i>), считать $C = C_3$.
86(87,88)	Проверить V_0 на <i>être</i> .
87(91)	Записать в n_7 место <i>s'être</i> .
88(89,90)	Проверить V_0 на принадлежность к группе <i>appeler, nommer</i> .
89(90)	Записать в n_8 место глагола этой группы.
90(91)	Записать V_0 , что он непереходный.
91(92)	Записать C_3 нулевой перевод.
92(1,93)	Проверить V_0 на отметку α_1 .
93(94,95)	Искать влево <i>adv</i> (<i>ne</i>), пропуская <i>adv</i> , P .
94(1)	Записать V_0 отрицательность.
95(1)	Записать V_0 утвердительность.
96(97,99)	Проверить по n_4 , были ли глаголы в потенциальном будущем.
97(98,99)	Проверить по n_5 , был ли глагол в <i>futur</i> .
98(99)	Записать глаголам в потенциальном будущем будущее время совершенного вида.
99	Останов.

2°. Предлог

1(2,107)	Искать вправо $C \in Pr$ V P (<i>en, dont</i>), считать $C = C_0$.
2(3,95)	Проверить C_0 на Pr .
3(1,4)	Проверить Pr_0 на однозначность.
4(5,55)	Искать вправо V , пропуская P , <i>adv</i> , считать $V = V_1$.
5(6,55)	Проверить V_1 на <i>infinitif</i> .
6(7,8)	Проверить Pr_0 на <i>pour</i> .
7(1)	Записать Pr_0 перевод «чтобы».
8(9,12)	Проверить Pr_0 на <i>de</i> .
9(10,11)	Искать влево оборот <i>il s'agit</i> , пропуская <i>adv</i> .
10(1)	Записать Pr_0 перевод «о том, чтобы».
11(1)	Записать Pr_0 нулевой перевод.
12(13,16)	Проверить Pr_0 на <i>par</i> .
13(14)	Записать Pr_0 перевод «тем, что».
14(15)	Искать влево V , считать $V = V_2$.
15(1)	Записать V_1 время, лицо, число V_2 .
16(17,1)	Проверить Pr_0 на <i>à</i> .
17(19,18)	Искать влево Z непосредственно, считать $Z = Z_1$.
18(19,23)	Искать влево Cc непосредственно, считать $Cc = C_1$.
19(20,24)	Проверить по n_1 наличие другого предлога <i>à</i> перед V в <i>infinitif</i> .
20(1)	Записать Pr_0 данные из n_1 .
21(22,29)	Искать от Z_1 влево Z , считать $Z = Z_2$.
22(23)	Переделать 22, 28, 51. в них во всех поиск вести от Z_2 .
23(24,29)	Искать влево <i>adv</i> непосредственно, считать $adv = adv_1$.
24(25,27)	Проверить adv_1 на <i>jusque</i> .
25(26)	Записать Pr_0 перевод «того, чтобы».
26(1)	Записать перевод Pr_0 в n_1 .
27(28,29)	Проверить adv_1 на <i>soit</i> .
28(26)	Записать Pr_0 перевод «нужно».
29(30,52)	Искать влево V , пропуская <i>adv</i> , Z , считать $V = V_3$.
30(31,32)	Проверить V_3 на принадлежность к группе: <i>arriver, conduire, ramener, regagner, venir</i> .
31(26)	Записать Pr_0 перевод «к тому, чтобы».
32(33,34)	Проверить V_3 на <i>borner</i> .
33(26)	Записать Pr_0 перевод «тем, что».
34(35,3)	Проверить V_3 на принадлежность к группе: <i>mettre, prendre</i> .
35(36)	Записать Pr_0 перевод «за то, чтобы».
36(26)	Записать V_3 перевод «приниматься».
37(38,39)	Проверить V_3 на <i>assujettir</i> .
38(26)	Записать Pr_0 перевод «условию».
39(40,42)	Проверить V_3 на <i>apprendre</i> .
40(41)	Записать Pr_0 нулевой перевод.
41(26)	Записать V_3 перевод «учить».
42(43,45)	Проверить V_3 на <i>savoir</i> .
43(44)	Записать Pr_0 нулевой перевод.
44(26)	Записать V_3 перевод «уметь».
45(46,47)	Проверить V_3 на <i>servir</i> .
46(26)	Записать Pr_0 перевод «для того, чтобы».

- 47(48,50) Проверить V_3 на *avoir*.
 48(49) Записать Pr_0 нулевой перевод.
 49(26) Записать V_3 информацию глагола прошедшего времени и перевод «должен».
 50(51) Записать Pr_0 перевод «который».
 51(52) Вставить перед V_3 *adv* с переводом «нужно».
 52(53,54) Искать влево S , считать $S = S_1$.
 53(54) Записать Pr_0 число и род S_1 .
 54(26) Записать Pr_0 винительный падеж.
 55(56,59) Проверить Pr_0 на \grave{a} .
 56(57,73) Искать влево S , пропуская A , *adv*.
 57(58,73) Искать вправо S , пропуская A , *adv*.
 58(1) Записать Pr_0 перевод «с» (твор. пад.).
 59(60,73) Проверить Pr_0 на *de*.
 60(61,63) Искать влево S , пропуская A , *adv*.
 61(62,63) Искать вправо S , пропуская A , *Ad*, *adv*.
 62(1) Записать Pr_0 нулевой перевод, родительный падеж.
 63(64,66) Искать влево V , пропуская *adv*, считать $V = V_4$.
 64(65,66) Проверить V_4 на отрицательность.
 65(1) Записать Pr_0 нулевой перевод.
 66(67,69) Искать вправо Pr (*entre*) непосредственно, считать $Pr = Pr_1$.
 67(68) Записать Pr_1 нулевой перевод.
 68(1) Записать Pr_0 перевод «из».
 69(70,73) Искать влево C со специальной пометкой о *de...à*.
 70(71,73) Искать вправо Pr (\grave{a}), не пропуская слов с кодом на \grave{a} , считать $Pr = Pr_2$.
 71(72) Записать Pr_0 перевод «от» (род. пад.).
 72(1) Записать Pr_2 перевод «до» (род. пад.).
 73(74,77) Искать вправо S , пропуская A , *Ad*, *adv*, считать $S = S_2$.
 74(75,77) Проверить S_2 на сильный левый код.
 75(76) Взять из таблицы, номер которой равен левому коду S_2 , перевод предлога и падеж.
 76(1) Записать Pr_0 взятые из таблицы данные.
 77(80,78) Искать вправо $C \in lequell \vee laquelle \vee lesquels \vee lesquelles \vee qui. \vee que \vee quoi$ непосредственно.
 78(79,80) Искать влево C с кодом, не пропуская Cs , считать $C = C_3$.
 79(85)] Взять из таблицы, номер которой равен коду C_3 , перевод предлога и падеж.
 80(81,83) Искать вправо V , считать $V = V_3$.
 81(81,82) Искать от V_3 вправо V , пропуская *adv*, P , считать $V = V_3$.
 82(85) Взять из таблицы, номер которой равен коду V_3 , перевод предлога и падеж.
 83(84,1) Проверить S_2 на слабый левый код.
 84(76) Взять из таблицы, номер которой равен левому коду S_2 , перевод предлога и падеж.
 85(86,76) Проверить C_3 или V_3 на принадлежность к группе слов, перевод которых меняется в зависимости от предлога.
 86(87,89) Проверить Pr_0 на *en*.
 87(88,76) Проверить C_3 (V_3) на принадлежность к группе: *développer, développable*.
 88(76) Записать C_3 (V_3) второй частный номер перевода (выбор 2-го значения).
 89(90,91) Проверить Pr_0 на *de*.
 90(88,76) Проверить C_3 (V_3) на принадлежность к группе: *distinguer, différent dérivable, combien*.
 91(92,93) Проверить Pr_0 на \grave{a} .
 92(88,76) Проверить C_3 (V_3) на принадлежность к группе: *attacher, inférieur relatif, supérieur*.
 93(94,76) Проверить Pr_0 на *par*.
 94(88,76) Проверить C_3 на *affirmer*.
 95(96,103) Проверить C_0 на *en*.
 96(1,97) Проверить C_0 на вхождение в сочетание *en + Ppr*, обработанное ранее.
 97(98,99) Искать вправо V , пропуская *adv*, P , считать $V = V_4$.
 98(98,99) Искать от V_4 вправо V , пропуская *adv*, P , считать $V = V_4$.
 99(100) Поставить C_0 (*en*) после V_4 .
 100(101) Вставить Pr перед C_0 , считать $Pr = Pr_3$.
 101(102) Взять из таблицы, номер которой равен коду V_4 , перевод и падеж, как для предлога *de*.
 102(106) Записать C_0 взятый из таблицы падеж.
 103(104,105) Искать вправо V , пропуская *adv*, P , считать $V = V_4$.

- 104(104,105) Искать от V_4 вправо V , пропуская adv , P , считать $V = V_4$.
 105(101) Вставить перед C_0 (*dont*), Pr (*de*), считать $Pr = Pr_3$.
 106(1) Записать Pr_3 взятый из таблицы перевод.
 107 Останов.

3°. Существительное

- 1(2,102) Искать вправо S , считать $S = S_0$.
 2(3,6,7,8,10) Ветвление по типу S_0 .
 3(4,5) Окончание S_0 равно $-s$.
 4(10) Записать S_0 множественное число.
 5(10) Записать S_0 единственное число.
 6(4,5) Окончание S_0 равно $-x$.
 7(4,5) Окончание S_0 равно $-ix$.
 8(9,5) Искать влево Ad , пропуская A , adv , считать $Ad = Ad_1$.
 9(10) Записать S_0 число Ad_1 .
 10(11,29) Искать влево $C \in S \vee V \vee Cs \vee Ad \vee Pr \vee Ppr \vee N \vee Cc \vee P$, пропуская A , adv , Pr , 3 , считать $C = C_2$.
 11(12,15) Проверить C_2 на Ad .
 12(13) Записать S_0 пометку α_1 о наличии Ad .
 13(15,14) Искать от C_2 влево $C \in Pr \vee Ppr \vee Cc$, пропуская adv , Ad , считать $C = C_2$.
 14(20,29) Искать от C_2 влево $C \in S \vee V \vee Cs \vee N$, пропуская A , Cc , adv , Pr , P , Pr , Ppr , 3 , считать $C = C_2$.
 15(16,20) Проверить C_2 на Pr .
 16(17) Записать S_0 падеж от Pr_2 .
 17(18,19) Проверить Pr_2 на особый.
 18(1) Записать S_0 пометку α_5 об особом предлоге.
 19(91) Записать S_0 пометку α_4 о не особом предлоге.
 20(21,24) Проверить C_2 на N .
 21(22) Записать в n_1 , какое было числительное.
 22(23) Переделать 10 команду на поиск влево от N_2 .
 23(10) Записать S_0 пометку α_6 о числительном.
 24(29,25) Проверить C_2 на Cs .
 25(26,44) Проверить C_2 на V .
 26(27,30) Проверить V_2 на *être*.
 27(28,29) Проверить V_2 на настоящее время.
 28(1) Записать S_0 творительный падеж.
 29(91) Записать S_0 именительный падеж.
 30(28,31) Проверить V_2 на принадлежность к группе: *devenir, paraître, s'appeler, se nommer*.
 31(32,29) Проверить V_2 на переходность.
 32(33,4) Проверить V_2 на *empêcher*.
 33(1) Записать S_0 дательный падеж.
 34(35,38) Проверить V_2 на принадлежность к группе: *appeler, nommer*.
 35(36) Записать S_0 пометку α_2 о том, что оно после этих глаголов.
 36(37,28) Проверить S_0 на пометку α_1 .
 37(91) Записать S_0 винительный падеж.
 38(39,40) Проверить V_2 на принадлежность к группе: *supposer, laisser, rendre*.
 39(37) Записать S_0 пометку α_3 о том, что оно после этих глаголов.
 40(41,42) Искать от V_2 влево *que*, не пропуская Cs , считать *que = que₁*.
 41(42,29) Искать от *que₁* вправо $C \in S_N \vee P_N$, не пропуская V_2 .
 42(43,37) Проверить V_2 на отрицательность.
 43(1) Записать S_0 родительный падеж.
 44(45,69) Проверить C_2 на S .
 45(46,47) Проверить S_2 на пометку α_5 об особом предлоге.
 46(1) Записать S_0 падеж S_2 и пометку α_3 .
 47(48,49) Проверить S_2 на пометку α_4 .
 48(14) Переделать 14 на поиск от S_2 .
 49(50,51) Проверить S_2 на пометку α_2 .
 50(28,37) Проверить S_2 на пометку α_1 .
 51(29,52) Проверить S_2 на именительный падеж.
 52(29,53) Проверить S_2 на пометку α_6 о переделанном именительном падеже.
 53(29,54) Искать от S_2 влево Cc , пропуская A , Ad , adv , считать $Cc = Cc_4$.
 54(56,55) Искать вправо V , не пропуская Cs , считать $V = V_5$.
 55(91) Записать S_0 падеж S_2 .
 56(63,57) Искать от V_3 влево $C \in S \vee P$, не пропуская S_0 , V , считать $C = C_4$.
 57(29,58) Проверить V_3 на третье лицо.
 58(59,55) Проверить V_3 на *participe présent*.

59(29,60)	Проверить V_3 на возвратность.
60(29,61)	Проверить V_3 на непереходность.
61(29,62)	Искать от Ppr_3 вправо $C \in A \vee Ad$, пропуская adv , считать $C = C_1$.
62(54)	Переделать 54 на поиск от Ppr_3 .
63(64,65)	Искать от C_4 влево Pr , пропуская A, Ad, adv .
64(56)	Переделать 56 на поиск от C_4 .
65(66,68)	Проверить C_4 на P .
66(67,55)	Искать от V_3 вправо V , считать $V = V_3$.
67(66,56)	Искать от V_3 влево Cs , пропуская adv .
68(69,73)	Искать от S_4 влево Cs , не пропуская S_0 , считать $Cs = Cs_1$.
69(71,70)	Проверить Cs_1 на запись в n_7 .
70(66)	Записать место Cs_1 в n_7 .
71(69,72)	Искать от Cs_1 влево Cs , считать $Cs = Cs_1$.
72(68,66)	Искать от Cs_1 влево S , считать $S = S_4$.
73(74,55)	Искать от S_4 влево Cs , пропуская A, Ad, adv .
74(75,78)	Искать от S_4 влево S , не пропуская V , считать $S = S_1$.
75(76,77)	Искать от S_1 влево Pr , пропуская A, Ad, adv .
76(74)	Переделать 74 на поиск от S_1 .
77(66,76)	Искать от S_1 влево Cs , пропуская A, Ad, adv .
78(29,79)	Проверить V_3 на Ppr .
79(29,55)	Проверить V_3 на множественное число.
80(81,85)	Искать от Cs_2 влево 3 непосредственно.
81(82,29)	Искать от Cs_2 влево S , не пропуская V , считать $S = S_1$.
82(83,84)	Проверить S_1 на пометку α_4 (о предлоге не особом).
83(81)	Переделать 81 на поиск от S_1 .
84(29,54)	Проверить S_1 на именительный падеж.
85(86,29)	Искать влево S , не пропуская V , считать $S = S_1$.
86(87,89)	Проверить S_1 на пометку α_5 (об особом предлоге).
87(91,88)	Проверить S_1 на пометку α_7 (о правом члене).
88(46)	Записать S_0 пометку α_7 (о правом члене).
89(90,84)	Проверить S_1 на пометку α_4 (о не особом предлоге).
90(85)	Переделать 85 на поиск от S_1 .
91(92,29)	Проверить C_2 на Ppr .
92(93,26)	Искать от Ppr_2 вправо 3 непосредственно.
93(24,29)	Искать от Ppr_2 влево $C \in S \vee V \vee Cs$.
94(95,1)	Проверить S_0 на пометку α_6 о числительном.
95(97,96)	Проверить S_0 на именительный падеж.
96(97,1)	Проверить S_0 на винительный падеж.
97(98,100)	Проверить N по n_1 на 1 тип.
98(99)	Записать S_0 именительный падеж, единственное число.
99(1)	Записать S_0 пометку о старом падеже и числе.
100(101,102)	Проверить N по n_1 на 2 тип.
101(99)	Записать S_0 родительный падеж, единственное число.
102(99)	Записать S_0 родительный падеж, множественное число.
103	Останов.

4°. Местоимение

1(2,61)	Искать вправо P , считать $P = P_0$.
2(1,3,13,13,1,16,29,39,1)	Ветвление по типу P_0 .
3(4,12)	Искать вправо V , пропуская adv , P , считать $V = V_1$.
4(5,7)	Проверить P_0 на <i>poss</i> .
5(6,7)	Проверить V_1 на первое лицо множественного числа.
6(1)	Записать P_0 именительный падеж.
7(7,8)	Искать от V_1 вправо V , пропуская adv , P , считать $V = V_1$.
8(10,9)	Проверить V_1 на переходность.
9(1)	Записать P_0 дательный падеж.
10(9,11)	Искать от V_1 вправо $C \in S_N \vee P_N$, не пропуская V, Cs .
11(1)	Записать P_0 винительный падеж.
12(14,6)	Искать влево Pr непосредственно, считать $Pr = Pr_1$.
13(14,15)	Искать влево Pr непосредственно, считать $Pr = Pr_1$.
14(1)	Записать P_0 падеж Pr_1 .
15(9,6)	Проверить P_0 на <i>lui</i> (IV тип).
16(17,24)	Проверить P_0 на <i>qui</i> .
17(18)	Записать P_0 именительный падеж.
18(19,21)	Искать вправо V , не пропуская Cs , считать $V = V_1$.
19(20)	Записать P_0 число V_1 .
20(1,21)	Проверить P_0 на множественное число.

21(22,25)	Искать влево Cs непосредственно.
22(23,25)	Искать влево P (VI тип), считать $P = P_0$.
23(27)	Записать P_0 расстояние $\rho = \rho(P_0, P_1) + \rho(P_1, Y(P_1))$.
24(21)	Записать P_0 винительный падеж.
25(26,1)	Искать влево не особое S , считать $S = S_1$.
26(27)	Записать P_0 расстояние $\rho = \rho(P_0, S_1)$.
27(28,1)	Проверить P_0 на <i>que</i> .
28(1)	Записать P_0 число S_1 .
29(30,31)	Искать влево Pt непосредственно, считать $Pt = Pt_1$.
30(32)	Записать P_0 падеж Pt_1 .
31(32)	Записать P_0 именительный падеж.
32(33,1)	Проверить P_0 на единственное число.
33(34,37)	Искать влево S не особое, считать $S = S_2$.
34(36,35)	Проверить P_0 на согласование с S_2 во французском роде и числе.
35(33)	Переделать 33 на поиск влево от S_2 .
36(1)	Записать P_0 расстояние $\rho = \rho(P_0, S_2)$.
37(38,1)	Искать вправо S не особое, считать $S = S_3$.
38(1)	Записать P_0 расстояние $\rho = \rho(P_0, S_3)$.
39(40,1)	Искать вправо $C \in S \setminus V \setminus A = S \setminus P$, считать $C = C_2$.
40(41,42)	Проверить C_2 на $A = S$.
41(1)	Записать P_0 перевод «из которых».
42(43,46)	Искать влево S не особое, считать $S = S_4$.
43(44)	Записать P_0 число S_4 .
44(45,46)	Проверить P_0 на единственное число.
45(46)	Записать P_0 расстояние $\rho = \rho(P_0, S_4)$.
46(47,51)	Проверить C_2 на S .
47(48,51)	Искать от S_2 вправо V , считать $V = V_3$.
48(51,49)	Проверить V_3 на возвратность.
49(50)	Записать P_0 родительный падеж.
50(1)	Переставить группу S_2 , поставить ее перед P_0 .
51(52,53)	Проверить P_0 на наличие некоторого падежа, выработанного при обработке предлогов.
52(53)	Записать P_0 родительный падеж.
53(55,54)	Проверить C_2 на V .
54(55,1)	Искать от C_2 вправо V , считать $V = V_2$.
55(55,56)	Искать от V_2 вправо V , пропуская <i>adv</i> , P , считать $V = V_2$.
56(57,59)	Проверить V_2 на переходность.
57(58,1)	Искать от V_2 вправо $C \in S_N \setminus P_N$, не пропуская V , Cs , считать $S = S_3$.
58(1)	Переставить группу S_3 , поставить перед P_0 .
59(60,1)	Искать вправо S_N , не пропуская V , Cs , считать $S = S_4$.
60(1)	Переставить группу S_4 , поставить ее перед P_0 .
61(62,66)	Искать вправо $P(on)$, считать $P = P_0$.
62(63,61)	Искать вправо V , не пропуская Cs , S_N , P_N , считать $V = V_1$.
63(64,62)	Проверить V_1 на третье лицо единственного числа.
64(65)	Записать V_1 первое лицо множественного числа.
65(63,61)	Искать от V_1 вправо V , не пропуская Cs , S_N , P_N , считать $V = V_1$.
66	Останов.

5°. Причастие

Participe passé

1(2,53)	Искать вправо Pp , считать $Pp = Pp_0$.
2(3,50)	Искать влево $C \in S \setminus V \setminus P \setminus Cs$, не пропуская Cs , считать $C = C_1$.
3(4,30)	Проверить C_1 на S .
4(5,16)	Проверить C_1 на именительный падеж.
5(6,13)	Искать вправо $C \in V_{3л} \setminus P_N \setminus S_N$, не пропуская Cs , считать $C = C_2$.
6(7,11)	Проверить C_2 на V .
7(8,9)	Проверить по n_1 наличие $S_N \setminus P_N$.
8(9,13)	Проверить V_2 на множественное число.
9(10)	Записать Pp_0 страдательное причастие прошедшего времени именительного падежа.
10(1)	Записать Pp_0 род и число C_1 .
11(12)	Записать в n_1 наличие $S_N \setminus P_N$.
12(5)	Переделать 5 на поиск от C_2 .
13(9,14)	Искать вправо от начала фразы <i>soit (soient)</i> , пропуская <i>adv</i> , 3.
14(15)	Записать Pp_0 страдательное причастие прошедшего времени краткой формы.

- 15(10) Вставить перед группой C_1 «после того, как».
 16(25,17) Проверить C_1 на особенность (S с de без Ad).
 17(18,20) Проверить Pr_0 на мужской род единственного числа.
 18(19) Записать Pr_0 род, число, падеж C_1 .
 19(1) Записать Pr_0 страдательное причастие прошедшего времени.
 20(18,21) Проверить C_1 на Φ .
 21(18,22) Проверить Pr_0 на согласование с C_1 .
 22(23,18) Искать от C_1 влево S , считать $S = S_3$.
 23(24,29) Проверить Pr_0 на согласование с S_3 .
 24(19) Записать Pr_0 род, число, падеж S_3 .
 25(26,18) Искать от C_1 влево S , пропуская A , Pr , Ppr , adv , $Pr(de)$, считать $S = S_3$.
 26(27,28) Проверить Pr_0 на согласование с S_3 .
 27(19) Записать Pr_0 род, число, падеж S_3 .
 28(18,29) Проверить Pr_0 на согласование с C_1 .
 29(22) Переделать 22 на поиск от S_3 .
 30(31,34) Проверить C_1 на P .
 31(32,4) Проверить P_1 на $cela$.
 32(33) Записать Pr_0 страдательное причастие краткой формы среднего рода единственного числа именительного падежа.
 33(1) Вставить перед P_1 «после того, как».
 34(35,46) Проверить C_1 на V .
 35(36,44) Проверить V_1 на $\acute{e}tre$.
 36(37,41) Проверить V_1 на настоящее время.
 37(38) Записать Pr_0 возвратный глагол настоящего времени третьего лица.
 38(39,40) Искать влево $C \in S_N \vee P_N$, считать $C = C_2$.
 39(1) Записать Pr_0 число C_2 .
 40(1) Записать Pr_0 единственное число.
 41(42) Записать Pr_0 страдательное причастие краткой формы.
 42(43,44) Искать влево $C \in S_N \vee P_N$, считать $C = C_2$.
 43(45) Записать Pr_0 род и число C_2 .
 44(42) Записать Pr_0 страдательное полное причастие творительного падежа.
 45(1) Записать Pr_0 запрет перестановки.
 46(47,48) Искать влево Pr , не пропуская V , Cs , S_N , P_N , считать $Pr = Pr_3$.
 47(1) Записать Pr_0 данные Pr_3 .
 48(49,50) Искать влево $C \in A \vee Ppr$, не пропуская V , Cs , S_N , P_N , считать $C = C_4$.
 49(1) Записать Pr_0 данные C_4 .
 50(52,51) Проверить Pr_0 на переходность.
 51(1) Записать Pr_0 деепричастие прошедшего времени.
 52(42) Записать Pr_0 страдательное причастие прошедшего времени.
 53 Останов.

Participe présent

- 1(2,51) Искать вправо Ppr , считать $Ppr = Ppr_0$.
 2(3,8) Искать влево Cs непосредственно.
 3(4,5) Искать влево Ppr , не пропуская V , Cs , S_N , P_N , считать $Ppr = Ppr_1$.
 4(1) Записать Ppr_0 данные Ppr_1 .
 5(6,7) Искать влево $C \in A \vee Pr$, не пропуская V , Cs , S_N , P_N , считать $C = C_2$.
 6(1) Записать Ppr_0 данные C_2 .
 7(1) Записать Ppr_0 деепричастие настоящего времени.
 8(9,7) Искать влево S , не пропуская V , Cs , P , Pr , считать $S = S_3$.
 9(10,31) Проверить S_3 на именительный падеж.
 10(11,19) Искать вправо $C \in V \vee S_N \vee P_N$, не пропуская Cs , считать $C = C_4$.
 11(12,18) Проверить C_4 на V .
 12(13,18) Проверить V_4 на третье лицо.
 13(14,16) Проверить по n_1 наличие $S_N \vee P_N$.
 14(15,19) Проверить V_4 на множественное число.
 15(16) Записать время V_4 в n_2 .
 16(17) Записать Ppr_0 действительное причастие настоящего времени именительного падежа.
 17(1) Записать Ppr_0 род и число S_3 .
 18(10) Записать наличие $S_N \vee P_N$ в n_1 .
 19(15,20) Искать вправо от начала фразы *soit (soient)*, пропуская adv , 3 .
 20(21) Записать Ppr_0 глагол третьего лица.

21(22)	Записать Prg_0 число S_3 .
22(23,25)	Искать вправо V , не пропуская S_3 , считать $V = V_2$.
23(24)	Записать Prg_0 время V_2 .
24(1)	Вставить перед группой S_3 «так как».
25(26)	Вставить перед группой S_3 «причем».
26(27,28)	Проверить время V_4 по n_2 .
27(1)	Записать Prg_0 время из n_2 .
28(29,30)	Искать влево V , считать $V = V_1$.
29(1)	Записать Prg_0 время V_1 .
30(29,31)	Искать вправо V , считать $V = V_1$.
31(1)	Записать Prg_0 настоящее время.
32(42,33)	Проверить S_3 на особенность (S с <i>de</i> без <i>Ad</i>).
33(34,36)	Проверить Prg_0 на мужской род единственного числа.
34(35)	Записать Prg_0 род, число, падеж S_3 .
35(1)	Записать Prg_0 действительное причастие настоящего времени.
36(34,37)	Проверить S_3 на Φ .
37(34,38)	Проверить Prg_0 на согласование с S_3 .
38(39,34)	Искать от S_3 влево S , считать $S = S_4$.
39(41,40)	Проверить Prg_0 на согласование с S_4 .
40(38)	Переделать 38 на поиск от S_4 .
41(35)	Записать Prg_0 род, число, падеж S_4 .
42(43,34)	Искать от S_3 влево S , пропуская A , Pp , Ppr , Pr (<i>de</i>), <i>adv</i> , считать $S = S_4$.
43(41,44)	Проверить Prg_0 на мужской род единственного числа.
44(41,45)	Проверить S_4 на Φ .
45(41,46)	Проверить Prg_0 на согласование с S_4 .
46(34,47)	Проверить Prg_0 на согласование с S_3 .
47(48,41)	Искать от S_4 влево S , считать $S = S_3$.
48(34,49)	Проверить S_3 на Φ .
49(34,50)	Проверить Prg_0 на согласование с S_3 .
50(47)	Переделать 47 на поиск от S_3 .
51	Останов.

ЛИТЕРАТУРА

- [1] Кулагина О. С., О машинном переводе с французского языка на русский. / (Проблемы кибернетики, М., 1960, № 3, стр. 181—208.)

Поступило в редакцию 4 VI 1958.

ГОСУДАРСТВЕННОЕ ИЗДАТЕЛЬСТВО
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
«ФИЗМАТГИЗ»

Москва, В-71, Ленинский проспект, 15.

ГОТОВЯТСЯ К ИЗДАНИЮ

Проблемы кибернетики, выпуск 5.

Автоматизация программирования, сборник переводов под ред. А. П. Ершова.

Бусленко Н. П. и Шрейдер Ю. А., Метод статистических испытаний (Монте-Карло) и его реализация на электронных цифровых машинах.

Гаазе-Рапопорт М. Г., Автоматы и живые организмы.

Гнеденко Б. В., Королюк Б. С. и Ющенко Е. Л., Элементы программирования.

Мак Кинси Дж., Введение в теорию игр, перевод с англ. под ред. Д. Б. Юдина.

Успенский В. А., Лекции о вычислимых функциях.

Фаддеев Д. К. и Фаддеева В. Н., Вычислительные методы линейной алгебры.

Предварительные заказы на эти книги принимают магазины Книго-торга. Оформив заказ на почтовой открытке лично в магазине, Вы получите извещение о поступлении книги в магазин. В случае отказа в приеме предварительного заказа просим сообщить Всесоюзному объединению книжной торговли по адресу: г. Москва, Ленинский проспект, 15.

ГОСУДАРСТВЕННОЕ ИЗДАТЕЛЬСТВО
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
«ФИЗМАТГИЗ»

Москва, В-71, Ленинский проспект, 15.

ИМЕЮТСЯ В ПРОДАЖЕ

Проблемы кибернетики, выпуск 1, 1958, 268 стр., ц. 12 р. 60 к.

Проблемы кибернетики, выпуск 2, 1959, 324 стр., ц. 14 р. 30 к.

Проблемы кибернетики, выпуск 3, 1960, 284 стр., ц. 12 р. 90 к.

Барсов А. С., Что такое линейное программирование, 1959, 104 стр., ц. 1 р. 60 к.

Березин И. С. и Жидков Н. П., Методы вычислений, т. I, 1959, 464 стр., ц. 11 р.

Березин И. С. и Жидков Н. П., Методы вычислений, т. II, 1959, 620 стр., ц. 14 р. 10 к.

Бут Э., Численные методы, 1959, 240 стр., ц. 7 р. 45 к.

Бут Э. и Бут Ю., Автоматические цифровые машины, 1959, 320 стр., ц. 10 р. 50 к.

Вентцель Е. С., Элементы теории игр, 1959, 68 стр., ц. 90 коп.

Китов А. И. и Крипцкий Н. А., Электронные цифровые машины и программирование, 1959, 572 стр., ц. 12 р. 50 к.

Книги продаются в книжных магазинах, а также высылаются наложенным платежом всеми республиканскими, краевыми и областными отделениями «Книга—почтой».