Кибернетический сборник

НОВАЯ СЕРИЯ

ВЫПУСК

11

Сборник переводов

Под редакцией

О. Б. ЛУПАНОВА

ИЗДАТЕЛЬСТВО „МИР“

Москва 1974

УДК 519.95

Научный совет по кибернетике
Академии наук СССР

Продолжение новой серии кибернетических сборников, публикация которой начата издательством «Мир» в 1965 г. В данном выпуске большой интерес представит обзорная статья известных ученых Хартманиса и Хопкрофта по теории сложности вычислений, а также статья Сэлтона, в которой излагаются некоторые новые результаты из области информационного поиска. В сборник вошли также работы зарубежных авторов по теории кодирования, теории автоматов и по различным приложениям кибернетики.

Сборник рассчитан на научных работников, инженеров, аспирантов и студентов различных специальностей, занимающихся и интересующихся кибернетикой в ее теоретическом аспекте.

Редакция литературы по математическим наукам

К $\frac{20205-024}{041\,(01)-74}$ 24—74

КИБЕРНЕТИЧЕСКИЙ СБОРНИК
Новая серия
выпуск 11

Редактор Л. Н. Базынина. Художник Н. К. Сапожников.
Художественный редактор В. И. Шаповалов. Технический редактор З. И. Резник.
Корректор Т. С. Лаврова

Сдано в набор 5/V 1974 г. Подписано к печати 2/X 1974 г. Бумага № 2 60×90 1/16. 6,25 бум. л. 12,5 печ. л. Уч.-изд. л. 12,19. Изд. № 1/7806. Цена 1 р. 43 к. Зак. 210

ИЗДАТЕЛЬСТВО «МИР»
Москва, 1-й Рижский пер., 2

Ордена Трудового Красного Знамени Ленинградская типография № 2 имени Евгении Соколовой Союзполиграфпрома при Государственном комитете Совета Министров СССР по делам издательств, полиграфии и книжной торговли. 198052, Ленинград, Л-52, Измайловский проспект, 29

Базисы для эквациональных теорий полугрупп[1)]

П. Перкинс

1. ВВЕДЕНИЕ

Удивительно трудными могут оказаться некоторые вопросы о конечных алгебраических структурах, даже с одной бинарной операцией и всего с несколькими элементами. В [4] Линдон поставил вопрос: всякая ли конечная алгебра (в смысле Г. Биркгофа) обладает конечным множеством тождеств, из которых выводимы все остальные? В той же работе он показал, что для всех двуэлементных алгебр ответ положительный. Затем, однако, он доказал [5], что в общем случае ответ отрицательный, указав конкретный семиэлементный группоид. Позднее четырехэлементный группоид, тождества которого не имеют конечного базиса, описал В. В. Вишин [11][2)].

Этот же вопрос о конечном базисе рассматривался в связи с группами. В 1937 г. было доказано (Б. Нейман [7]), что тождества любой абелевой группы имеют конечный базис. Такой же результат для нильпотентных групп был получен Линдоном [6] и обобщен Хигманом [3]. Самый, вероятно, значительный результат на сегодняшний день — это положительное решение вопроса для всех конечных групп, данное Оутс и Пауэллом в их весьма «теоретико-групповой» статье [8]. Интересный вопрос о том, существует ли группа, не имеющая конечного базиса тождеств, все еще открыт[3)].

Цель настоящей статьи — получить положительный ответ на вопрос о конечной базируемости тождеств для всех коммутативных полугрупп и всех равномерно периодических полугрупп,

---

[1)] Peter Perkins, Bases for equational theories of semigroups, *Journal of Algebra*, **11** (1968), № 2, 298—314.

[2)] Трехэлементный группоид с этим свойством построен В. Л. Мурским в [12]. — *Прим. перев.*

[3)] А. Ю. Ольшанский, С. И. Адян и М. Р. Воан-Ли доказали, что такие группы существуют. См. [13, 14, 15]. — *Прим. перев.*

удовлетворяющих какому-либо «соотношению перестановочности». С другой стороны, мы указываем шестиэлементную полугруппу, тождества которой не имеют конечного базиса. Приводимые доказательства по идее чрезвычайно элементарны и основаны большей частью на пошаговом анализе выводов в определенной эквациональной системе.

Эти результаты составляют часть докторской диссертации автора (Калифорнийский университет, Беркли, 1966).

2. ПРЕДВАРИТЕЛЬНЫЕ ПОНЯТИЯ

Эквациональные теории полугрупп и связываемая с ними структура вывода могут трактоваться как подтеории теорий первого порядка с равенством, наделенные дедуктивной структурой исчислений первого порядка. С другой стороны, их легко определить и независимо, причем многими эквивалентными между собой способами. В целях большей самостоятельности изложения мы выбираем один из таких способов.

Под *полугруппой* S понимается множество S_0 вместе с бинарной ассоциативной операцией на S_0, которая обозначается конкатенацией. С полугруппой S связываются описываемые ниже язык и дедуктивная структура.

Язык. 1) Множество *переменных* — это $\{w, x, y, z, w_1, x_1, \ldots\}$.

2) Множество *термов* — это наименьшее множество, содержащее все переменные и такое, что если s и t — термы, то st — терм. Множество *подтермов* переменной v — это $\{v\}$. *Подтермы* терма t — это сам терм и всевозможные подтермы термов s_1 и s_2, где $s_1 s_2$ есть t. Отсутствие вхождения иногда обозначается словами «пустой терм».

3) Множество всех *тождеств* — это множество $\{s = t \mid s$ и t — термы$\}$.

Правила вывода. Пусть r, s, t, u — термы. Разрешаются следующие шаги вывода.

E1: вывести $s = t$ из $r = u$, если $s = t$ — это результат подстановки некоторого терма на место всех вхождений некоторой переменной в $r = u$.

E2: вывести $s = s$ из пустого множества.

E3: вывести $s = t$ из $t = s$.

E4: вывести $rs = us$ из $r = u$.

E5: вывести $sr = su$ из $r = u$.

E6: вывести $s = t$ из $s = r$ и $r = t$.

Пусть E — некоторое множество тождеств. Конечная последовательность тождеств $e_1, e_2, \ldots, e_n$ называется *выводом* тождества e_n из E, если каждое e_i либо является элементом множества E, либо выведено из предшествующих тождеств данной последовательности по одному из указанных правил вывода. Тождество e — *тео-*

рема в E, или e, *выводимом из* E, если существует вывод e из E. В этом случае мы пишем $E \vdash e$.

Мы предполагаем, что с понятиями, используемыми в последующих определениях и обозначениях, читатель знаком (см., например, [1]). Для удобства читателя ниже приведен список обозначений, которые будут использоваться в дальнейшем.

v — переменные.

s, t — термы.

E — система тождеств.

$s \leqslant t$ — означает: s — подтерм терма t. Для собственного подтерма используется $<$.

$\mathrm{var}(s)$ — множество всех переменных, встречающихся в s.

$\mathrm{occ}(v, s)$ — число вхождений v в s.

S — полугруппа.

f, g, h — функции назначений — естественные продолжения отображений переменных на элементы полугруппы до отображений термов на элементы этой полугруппы [1]).

$s = t$ *верно в* S, или S — *модель для* $s = t$, если для всех f, соответствующих S, $f(s) = f(t)$; S есть модель системы тождеств E, если она является моделью тождества e для всех $e \in E$.

$I(S)$ — *система тождеств полугруппы* S — множество всех тождеств, верных в S, т. е. *эквациональная теория* системы S.

$I_n(S)$ — система всех тождеств в S с не более чем n переменными.

FS_p/E — *относительно свободная полугруппа* с p образующими, определенная системой E, т. е. полугруппа, элементы которой суть классы смежности термов для эквивалентности $s \equiv {}_E t \leftrightarrow E \vdash s = t$, с естественной операцией. Рассматриваемые термы содержат переменные лишь из заданного множества мощности p.

$c(E)$ — *замыкание* системы E — множество всех теорем в E. Заметим, что $I(FS_\omega/E) = c(E)$.

E *конечно базируема*, если существует конечная система E_0, такая, что $c(E_0) = c(E)$. Если E конечно базируема, то существует конечная система $E_0 \subseteq E$, такая, что $c(E_0) = c(E)$.

S *имеет конечный базис тождеств*, если $I(S)$ конечно базируема.

CFS_ω — FS_ω/ϕ.

Дадим теперь один критерий выводимости, полезный при установлении свойств $c(E)$ по данному E.

Теорема 1. $E \vdash s = t$ *тогда и только тогда, когда существует последовательность тождеств* $M_i s_i N_i = M_i t_i N_i$, $i = 1, \ldots, n$, *такая что:*

[1]) Другими словами, такая f определяется набором значений переменных в S, скажем $v_1 = a_1$, $v_2 = a_2$, ..., и сопоставляет каждому терму его значение при $v_1 = a_1$, $v_2 = a_2$, ... — *Прим. перев.*

1) s_i, t_i — *непустые термы*; M_i, N_i — (*возможно пустые*) *термы*;
2) *для каждого* i $s_i = t_i$ *или* $t_i = s_i$ *есть результат подстановки в какое-нибудь тождество из* E (*или же* s_i *есть* t_i);
3) $M_1 s_1 N_1$ *есть* s;
4) $M_n t_n N_n$ *есть* t;
5) $M_i t_i N_i$ *есть* $M_{i+1} s_{i+1} N_{i+1}$, $i = 1, \ldots, n-1$.

Доказательство. Назовем такую последовательность *T-последовательностью для тождества* $s = t$. Теорема легко доказывается в обе стороны с помощью индуктивного теоретико-доказательственного рассуждения. Индукция ведется в одну сторону по длине данного вывода, в другую — по длине данной T-последовательности.

3. ПОЛУГРУППЫ БЕЗ КОНЕЧНОГО БАЗИСА ТОЖДЕСТВ

Используя теорему 1, нетрудно построить систему E (полугрупповых) тождеств, не являющуюся конечно базируемой, а из этого, переходя к FS_ω/E, — полугруппу, не имеющую конечного базиса тождеств.

Теорема 2. *Пусть* $E = \{xyzw = xzyw,\ yx^k y = xyx^{k-2}yx,\ k = 2, 3, \ldots\}$. *Тогда* E *не конечно базируема.*

Доказательство. Предположим, что $E_0 \subseteq E$ — конечный базис. Пусть n превосходит все показатели из E_0. Если $yx^{n!+1}y$ есть $M_i s_i N_i$, где s_i — результат подстановки в левую или правую часть тождества из E_0, то t_i должно быть s_i. Таким образом, $M_i s_i N_i$ есть $M_i t_i N_i$, и с помощью E_0 он может быть переведен только в себя. Теорема доказана.

Лишнее на первый взгляд тождество $xyzw = xzyw$ мы включили в E, чтобы впоследствии сослаться на FS_ω/E, как на пример «пермутативной» полугруппы без конечного базиса тождеств.

Поиски конечной полугруппы без конечного базиса тождеств можно вести как в направлении от полугрупп к тождествам, так и от тождеств к полугруппам. Первый подход оказался бесплодным, потому что по заданной таблично полугруппе трудно получить удобную характеристику системы ее тождеств, не говоря уже об утомительной проверке ассоциативности. В соответствии с этим мы построили некоторое множество тождеств M с бесконечным числом переменных; искомый пример после этого задается как гомоморфный образ полугруппы FS_2/M. При этом, как часто бывает, возникают более общие, хотя и менее наглядные рассмотрения, к изложению которых мы сейчас переходим.

Говорят, что t — *изотерм относительно* E, если его класс смежности $[t]_E$, или просто $[t]$, определенный системой E на CFS_ω, есть

$\{t\}$. Заметим, что если t — изотерм относительно E, то таковы же все его подтермы.

Система E *замкнута относительно вычеркивания*, если из $E \vdash s = t$ следует, что $\mathrm{var}(s) = \mathrm{var}(t)$ и что при вычеркивании всех вхождений любой переменной получающееся тождество, если оно не «пусто», выводимо из E. Например, система всех тождеств любой полугруппы с единичным элементом, не определимым алгебраически (т. е. не являющимся постоянным значением при любых значениях переменных, некоторого терма), замкнута относительно вычеркивания.

E *специальна*, если $xyzyx$ и $xzyxy$ — изотермы и из E не выводимо ни одно из тождеств $xxz = zxx$, $xyxy = xyyx$ и $xyxy = yxyx$ [1]).

В дальнейшем, до конца этого раздела, E предполагается специальной и замкнутой относительно вычеркивания. Таким образом, $xyzxy$ и все его подтермы также являются изотермами относительно E.

Определения. Переменная v *бесповторна в* t, *если* $\mathrm{occ}(v, t) = 1$ и t *линеен*, если все его переменные бесповторны в t. Терм t *2-ограничен*, если $\mathrm{occ}(v, t) \leqslant 2$ для всех v. Система E — *упрощенная*, если всякий раз, когда $s = t \in E$ и s, t 2-ограничены, бесповторные в s или в t переменные не встречаются на концах s и t. Если t есть $v_1 v_2 \ldots v_n$, то его *обращение* t^R — это терм $v_n v_{n-1} \ldots v_1$; $\mathrm{end}(t)$ — это переменная, стоящая на правом конце t.

Докажем теперь четыре леммы, устанавливающие некоторые свойства специальных систем, замкнутых относительно вычеркивания.

Лемма 3. *Если* $E \vdash Bv = t$ $(vB = t)$, B *и* t 2-*ограничены и* $v \nleqslant B$, *то* t *есть* $Cv(vC)$, *где* $v \nleqslant C$.

Доказательство. Несомненно, $\mathrm{occ}(v, t) = 1$. Предположим, что в t справа от v встречается v_1. При вычеркивании всех переменных, кроме v и v_1, в левой части остается либо $v_1 v$, либо $v_1 v_1 v$. Из того, что $v_1 v$ и $v_1 v v_1$ — изотермы, а тождество $v_1 v_1 v = v v_1 v_1$ из E не выводимо, вытекает требуемое.

Лемма 4. *Если* E *имеет конечный базис, то* E *имеет упрощенный конечный базис.*

Доказательство. *Пусть* $E_0 \subseteq E$ — конечный базис для E. Пусть $s = t \in E_0$, s и t 2-ограничены и s имеет вид Bv, где $v \nleqslant B$ В силу предыдущей леммы t имеет вид Cv с $v \nleqslant C$, так что в E_0 мы можем заменить $Bv = Cv$ на $B = C$. Аналогично рассуждаем для трех оставшихся возможных положений v. Конечное число таких

---

[1]) Последнее тождество добавлено при переводе. Без него теряет силу утверждение последней фразы следующего абзаца. — *Прим. перев.*

замен приведет к конечному упрощенному базису для E. Лемма доказана.

Лемма 5. *Пусть* $E \vdash s = t$, s *и* t 2-*ограничены*, s *имеет вид* BvC (CvB), *где* C *линеен и* $v \notin B, C$. *Тогда* t *есть* DvC (CvD).

Доказательство. Пусть C есть $v_1, \ldots, v_n$, где все v_i различны. Терм t должен иметь вид DvD'. Вычеркивая переменные из $\mathrm{var}(C)$ и применяя лемму 3, имеем $\mathrm{var}(D') \subseteq \mathrm{var}(C)$. Вычеркивая все переменные из $\mathrm{var}(D')$, имеем $\mathrm{var}(C) \subseteq \mathrm{var}(D')$, так что $\mathrm{var}(D') = \{v_1, v_2, \ldots, v_n\}$. Для каждых двух целых i, j, $1 \leqslant i < j \leqslant n$, вычеркнем из $BvC = DvD'$ все переменные, кроме v, v_i, v_j. Левая часть перейдет в один из термов vv_iv_j, $v_ivv_iv_j$, $v_jvv_iv_j$, $v_iv_jvv_iv_j$ или $v_jv_ivv_iv_j$, каждый из которых — изотерм относительно E. Этим установлены не только линейность D', но и тот факт, что в D' все v_k идут в том же порядке, что и в C, так что D' есть C. Лемма доказана.

В рассматриваемые системы E мы хотим теперь включать определенные тождества, содержащие термы вместе с их обращениями, так что возможности применения подстановки при их выводе минимальны. Будем говорить, что t обладает свойством Q_n, если он имеет вид $vBvB^R$, где $v \notin B$ и B — линейный терм с n переменными.

Лемма 6. *Пусть* $E_0 \subseteq E$ — *упрощенная система, содержащая менее* n *переменных, и пусть* $E_0 \vdash s = t$. *Если* s *обладает свойством* Q_n, *то* t *тоже обладает этим свойством, причем* $\mathrm{end}(s) = \mathrm{end}(t)$.

Доказательство. Нам надо лишь показать, что i-й «шаг» в T-последовательности для $s = t$, т. е. переход от $M_is_iN_i$ к $M_it_iN_i$, $i = 1, \ldots, m$, сохраняет Q_n и последнюю переменную. Предположим, что $s_i = t_i$ является результатом подстановки σ в тождество $s_i' = t_i' \in E_0$. Поскольку E_0 упрощенная, переменные на концах s_i', а значит и s_i, не бесповторные.

Если M_i не пусто, то s_i может иметь лишь вид

$$v_k \ldots v_n v v_n \ldots v_k,$$

и в этом случае t_i есть s_i по лемме 5.

Если M_i пусто, то, чтобы не кончаться бесповторной переменной, терм s_i должен содержать n различных переменных. Таким образом, подстановка σ не должна просто сводиться к переименованию переменных; учитывая еще, что s имеет вид $v'Bv'B^R$, получаем, что хотя бы одна из переменных, вместо которых делается подстановка, скажем переменная v, бесповторна в s_i'. Предположим, что уже выполнена подстановка на место всех переменных, кроме одной, последней, бесповторной переменной v, и что M_i и N_i уже приписаны, так что получилось $\bar{s} = \bar{t}$. Итак, $\mathrm{occ}(v, \bar{s}) = \mathrm{occ}(v, \bar{t}) = 1$.

Случай I. Терм $\bar{s}$ есть BvC, где B и C линейны или пусты. Вы-

черкивая все переменные, кроме двух, одна из них v, мы видим, что $\bar{t}$ должен быть 2-ограничен, и согласно лемме 5 $\bar{t}$ есть $\bar{s}$.

Случай II. Терм $\bar{s}$ имеет вид

$$wx_1 \dots x_p y_1 \dots y_q z_1 \dots z_r w z_r \dots z_1 v x_p \dots x_1.$$

Хотелось бы установить, что $\bar{t}$ также 2-ограничен, чтобы можно было применить лемму 5. Заметим, что переменная y_q, так же как и v, бесповторна в $\bar{s}$. Опуская все переменные, кроме w, y_q, находим, что $\operatorname{occ}(w, \bar{t}) = 2$. Для каждого i в отдельности, опуская все переменные, кроме x_i, y_q, показываем, что $\operatorname{occ}(x_i, \bar{t}) = 2$. Опуская все переменные, кроме w, y_q, z_i, получаем $\operatorname{occ}(z_i, \bar{t}) = 2$. Таким образом, по лемме 5 терм $\bar{t}$ должен иметь вид

$$wx_1 \dots x_p y_1 \dots y_q B w C v x_p \dots x_1,$$

причем фактически мы доказали, что B и C линейны и $\operatorname{var}(B) = \operatorname{var}(C) = \{z_1, \dots, z_n\}$. Наконец, если бы C не совпадал с B^R, то нашлись бы такие i и j, что, оставляя только z_i и z_j, мы получили бы $z_i z_j z_j z_i = z_i z_j z_i z_j$, что невозможно, так как система E специальна.

Случай III. Терм $\bar{s}$ имеет вид

$$wx_1 \dots x_p v z_1 \dots z_r w z_r \dots z_1 y_q \dots y_1 x_p \dots x_1$$

или

$$wvz_1 \dots z_r w z_r \dots z_1 y_q \dots y_1.$$

Этот случай не отличается от случая II, если поменять ролями v и y_q.

Случай IV. Переменная v встречается в $\bar{s}$ с краю. Вычеркивая переменные, легко убедиться, что терм $\bar{t}$ также должен быть 2-ограниченным, что противоречит упрощенности E_0.

Итак, в каждом из случаев I—III, которые только и возможны, сохраняются как свойство Q_n, так и правая переменная. Лемма доказана.

Определим теперь *квазизеркальное* тождество как любое из тождеств

$$wx_1 \dots x_n w x_n \dots x_1 = w x_n \dots x_1 w x_1 \dots x_n, \quad n = 1, 2, \dots$$

Теорема 7. *Если система E специальна, замкнута относительно вычеркивания и содержит все квазизеркальные тождества, то E не конечно базируема.*

Доказательство. Будь E конечно базируема, существовал бы конечный упрощенный базис $E_0 \subseteq E$, содержащий меньше, чем скажем, n переменных. В силу леммы 6 n-ое квазизеркальное тождество не было бы выводимо из E_0. Теорема доказана.

Построим теперь шестиэлементную полугруппу Z, тождества которой удовлетворяют условиям теоремы 7. Полугруппа Z — это полугруппа относительно матричного умножения, образованная следующими вещественными матрицами второго порядка:

$$\begin{pmatrix}0&0\\0&0\end{pmatrix},\ \begin{pmatrix}1&0\\0&1\end{pmatrix},\ \begin{pmatrix}0&1\\0&0\end{pmatrix},\ \begin{pmatrix}0&0\\1&0\end{pmatrix},\ \begin{pmatrix}1&0\\0&0\end{pmatrix},\ \begin{pmatrix}0&0\\0&1\end{pmatrix}.$$

$$0 \qquad 1 \qquad a \qquad b \qquad ab \qquad ba$$

	0	1	a	b	ab	ba
0	0	0	0	0	0	0
1	0	1	a	b	ab	ba
a	0	a	0	ab	0	a
b	0	b	ba	0	b	0
ab	0	ab	a	0	ab	0
ba	0	ba	0	b	0	ba

Теорема 8. *Тождества полугруппы Z не конечно базируемы.*

Доказательство. $I(Z)$ замкнута относительно вычеркивания, поскольку 1 — неалгебраический единичный элемент.

Теперь покажем, что $I(Z)$ специальна.

Пусть $xyzyx = t \in I(Z)$. Придавая переменным y и z значения $f(y) = a$, $f(z) = b$, легко получаем, что yzy — изотерм относительно $I(Z)$. Поэтому t должен быть одним из термов $yxzxy$, $yxzyx$, $xyzxy$ или $xyzyx$. «Назначая» $f(x) = f(z) = a$ и $f(y) = b$, отвергаем первые три возможности.

Пусть $xzyxy = t \in I(Z)$. Опускаем переменную y и видим, что $\operatorname{occ}(x, t) = 2$. Опускаем переменную x и, полагая $f(z) = a, f(y) = ba$, видим, что в t все переменные y должны стоять справа от z. Но в t справа от z не может быть больше двух y-ов, иначе t имел бы подтерм yy, что невозможно, как показывает назначение $f(x) = a$, $f(y) = b$, $f(z) = ba$. Это же назначение в действительности показывает, что t — это сам $xzyxy$.

Пусть $I(Z) \vdash xxz = zxx$. Противоречие получается, если положить $f(x) = ab$, $f(z) = a$.

Пусть $I(Z) \vdash xyxy = xyyx$. Противоречие получаем, полагая $f(x) = a$, $f(y) = b$.

Пусть $I(Z) \vdash xuxy = yuyx$[1]). Противоречие получаем, полагая $f(x) = a$, $f(y) = b$.

Дадим, наконец, полное индуктивное доказательство (хотя нетрудно провести и неформальную проверку) верности *в Z всех квазизеркальных тождеств.*

Будем называть *n-тождеством квазизеркального типа* (n-т. к. т.) любое тождество вида

$$v_0v_1 \dots v_nv_0v_n \dots v_1 = v_0v_n, \dots, v_1v_0v_1 \dots v_n,$$

а *n-тождеством зеркального типа* (n-т. з. т.) — любое тождество вида

$$v_1 \dots v_nv_n \dots v_1 = v_n \dots v_1v_1 \dots v_n.$$

Докажем индукцией по n, что в Z верны все n-т. к. т. и n-т. з. т.

Для $n = 1$ это утверждение верно тривиальным образом. Предположим, что оно верно для $n = k$, и рассмотрим какое-либо $(k+1)$-т. з. т., скажем $s = t$. Если f — произвольная функция назначений, полагающая хотя бы одну переменную из $s = t$ равной 1, то $f(s) = f(t)$ по предположению индукции. В противном случае либо $f(s) = f(t) = 0$, либо $f(v) = ab$ для всех встречающихся v или же $f(v) = ba$ для всех встречающихся v, и тогда $f(s) = f(t) = ab$ или, соответственно, $f(s) = f(t) = ba$. Итак, в Z верны все n-т. з. т.

Теперь рассмотрим какое-нибудь $(k+1)$-т. к. т., скажем $s = t$. Если одной из его переменных придано значение 1, то $f(s) = f(t)$ либо по предположению индукции, либо в силу уже доказанного результата для $(k+1)$-т. з. т. Если же это не так, рассмотрим два случая.

1) $k+1$ четно. Здесь для f имеются в точности те же возможности, что и выше.

2) $k+1$ нечетно. Помимо рассмотренных, есть еще только два способа получить ненулевое $f(s)$ или $f(t)$. Либо

$$f(v_i) = \begin{cases} b & \text{при } i \text{ нечетном,} \\ a & \text{при } i \text{ четном,} \end{cases}$$

либо

$$f(v_i) = \begin{cases} a & \text{при } i \text{ нечетном,} \\ b & \text{при } i \text{ четном.} \end{cases}$$

В этих случаях $f(s) = f(t) = ab$ или $f(s) = f(t) = ba$ соответственно. Теорема доказана.

Используя Z, а также теорему 7, нетрудно показать, что свойство конечной базируемости тождеств, даже в классе полугрупп, не сохраняется при образовании прямых произведений, гомоморфных образов или же при переходе к подалгебрам. Еще одно, несколько более неожиданное явление несохранения может иметь место, как

[1]) Этот абзац добавлен при переводе. См. сноску на стр. 9. — *Прим. перев.*

мы сейчас покажем, при присоединении к полугруппе единичного элемента.

Определим на CFS_ω следующее отношение $*: s*t$, если s есть t или ни s, ни t не есть подтерм в $xyzyx$, $xzyxy$, $xyxy$ или xxz. Ясно, что $*$ — это конгруенция с числом классов на 1 большим, чем число различных подтермов приведенных выше четырех термов. Заметим, что $H = CFS_\omega/*$ удовлетворяет тождеству $x_1x_2x_3x_4x_5x_6 = y_1y_2y_3y_4y_5y_6$, и поэтому ее тождества конечно базируемы. С другой стороны, если H_1 — полугруппа, образованная присоединением к H единичного элемента, то, как легко проверить, $I(H_1)$ удовлетворяет условиям теоремы 7 и не конечно базируема.

Припомните теперь замечание во введении, что вопрос о существовании группы без конечного базиса тождеств все еще открыт [1]). Заметив особенности системы квазизеркальных тождеств по отношению к конечным базисам, можно предложить в качестве одного из подходов к этой теоретико-групповой задаче изучить FG_ω/J — относительно свободную группу с ω образующими, определенную какой-нибудь системой J, подобной I.

4. КОММУТАТИВНЫЕ ПОЛУГРУППЫ

В этом разделе наша цель — показать, что тождества всякой коммутативной полугруппы конечно базируемы, и получить оценки числа переменных, небходимых для любого конечного базиса.

Полугруппа S называется *равномерно периодической* (р. п.), если существуют такие целые m, $k > 0$, что $x^m = x^{m+k} \in I(S)$. Если m_0 — наименьшее среди таких m и для этого m_0 число k_0 — наименьшее из таких k, то мы говорим, что S есть (m_0, k_0)-р. п. полугруппа.

В этом разделе мы будем предполагать, что S коммутативна, а $E \vdash e$ будет означать, что $E \cup \{c\} \vdash e$, где c — закон коммутативности. Заметьте, что либо $I(S)$ тривиальна в том смысле, что для всех $s = t \in I(S)$ и всех v справедливо $\mathrm{occ}(v, s) = \mathrm{occ}(v, t)$, либо S равномерно периодическая. В первом случае тождества S тривиальным образом имеют конечный базис, поэтому мы будем предполагать в дальнейшем, что $S(m_0, k_0)$-р. п.

Терм t называется *приведенным коммутативным*, или, когда не возникает неясностей, просто *приведенным*, если он имеет вид $v_1^{k_1} \ldots v_n^{k_n}$, где все v_i различны и $k_1 < m_0 + k_0$. Тождество $s = t$ приведенное, если s и t приведенные. Мы сейчас покажем, что в приведенных тождествах для показателей возможно лишь конечное число различных «ситуаций».

Термом с показателем 0 будем называть пустой терм. Пусть $0 \leqslant p, q < m_0 + k_0$. Множество $B = \{v_1, \ldots, v_n\}$ будет называться

[1]) См. сноску на стр. 5. — *Прим. перев.*

p-блоком приведенного терма s, если каждая переменная $v_i \in B$ имеет в s показатель p. Множество B назовем (p, q)-*блоком приведенного тождества* $s = t$, если B есть p-блок для s и q-блок для t, причем из чисел p, q хотя бы одно отлично от 0. *Длина* блока B — это число n.

Рассмотрим, например, тождество $x_1^2x_2^3x_3^5x_4^5x_5^2 = x_1^6x_5^6x_6^4$. Считая, что оно приведенное, т. е. что $6 < m_0 + k_0$, мы видим, что $\{x_1, x_5\}$, $\{x_1\}$, $\{x_5\}$ — это $(2, 6)$-блоки, а $\{x_6\}$ — $(0, 4)$-блок.

Если для конкретных p и q тождество e вообще имеет какой-нибудь (p, q)-блок, то оно имеет и (единственный) максимальный блок, содержащий все остальные (p, q)-блоки для e. Чтобы избежать употребления двойных индексов, предположим, что все пары (p, q) занумерованы, так что можно говорить о 1-блоке, 2-блоке, ..., r-блоке в e, где $r = (m_0 + k_0)^2 - 1$.

Пусть e — произвольное приведенное тождество, и пусть $1 \leqslant k \leqslant r$. Пусть $\lambda_k(e)$ будет длиной максимального k-блока тождества e, если такой существует, и 0 — в противном случае.

Теорема 9. *Каждая коммутативная полугруппа имеет конечный базис тождеств.*

Доказательство. Пусть это не так. Тогда должна существовать последовательность $e_1, e_2, \ldots$, такая, что для любого $i > 0$ $e_i \in I(S)$, но неверно $\{e_1, \ldots, e_i\} \vdash e_{i+1}$. Каждому e_i сопоставим «блок-вектор» $\lambda^{(i)}$, где $\lambda_k^{(i)} = \lambda_k(e_i)$, $1 \leqslant k \leqslant r$. Индукцией по r легко показать, что при частичном упорядочении, индуцированном естественным порядком на каждой из координат, бесконечное множество r-мерных векторов с целыми неотрицательными координатами обязательно содержит бесконечную возрастающую последовательность. Но если $\lambda_k^{(m)} \leqslant \lambda_k^{(n)}$, $1 \leqslant k \leqslant r$, то мы можем, подставляя в блоках, где неравенство строгое, вместо любой переменной линейный терм из новых переменных, получить из e_m некоторое тождество e'_m, для которого $\lambda_k(e'_m) = \lambda_k(e_n)$, $1 \leqslant k \leqslant r$ [1]). Затем простым переименованием переменных мы можем вывести e_n из e'_m, что противоречит исходному предположению о тождествах e_i. Теорема доказана.

Рассмотрим, например, тождества

$$e\colon\ x^2y^3 = x^4$$

и

$$e'\colon\ x^2y^2z^3 = x^4y^4.$$

Здесь имеются (2,4)-блок и (3,0)-блок, длины которых при переходе от e к e' возрастают, и $e \vdash e'$.

---

[1]) Предполагается, что m достаточно велико, так что $\lambda_k^{(m)} < \lambda_k^{(n)}$ лишь для тех k, для которых $\lambda_k^{(m)} \neq 0$. — *Прим. перев.*

Перейдем к получению оценки числа переменных, необходимых для конечного базиса, в случае когда S имеет лишь конечное число простых элементов.

Элемент $a \in S$ называется *простым*, если для всех $b, c \in S$ имеем $a \neq bc$. Заметьте, что, как вытекает из этого определения, если в S есть единичный элемент или она идемпотентна, то в ней вообще нет простых элементов. Если S конечно порожденная, то в ней простых элементов лишь конечное число. Символ [ ] будет обозначать, как обычно, целую часть.

Лемма 10. *Пусть* $e \in I_d(S)$, *пусть* e *не выводимо из* $I_{d-1}(S)$, *и пусть* e *имеет блок длины* $n+1 > 1$. *Тогда* S *содержит по крайней мере* $\left[\frac{n}{m_0}\right] + 1$ *различных простых элементов.*

Доказательство. Предположим, что e эквивалентно тождеству

$$(v_1 v_2 \ldots v_{n+1})^p G = (v_1 v_2 \ldots v_{n+1})^q H,$$

где данные переменные v_i не входят ни в G, ни в H, а p и q не равны одновременно 0. Вычеркивая из $e: s = t$ переменную v_{n+1}, получим новое тождество $e': s' = t'$. Разумеется, $e' \vdash e$ и в e' переменных всего лишь $d-1$, поэтому по условию леммы $e' \in S$ неверно. Значит, существует конкретная функция назначений f, которая приписывает термам s' и t' разные значения $f(s')$ и $f(t')$ из S. Каждый элемент $f(v_i)$, $i = 1, \ldots, n$, должен быть простым в S, ибо если бы выполнялось, скажем, $f(v_n) = bc$, то мы могли бы определить новую функцию назначений g следующим образом:

$$g(v_n) = b$$

$$g(v_{n+1}) = c$$

$$g(v) = f(v) \text{ для остальных переменных.}$$

Тогда было бы $f(s') = g(s)$ и $f(t') = g(t)$. Но $e \in I(S)$, так что $g(s) = g(t)$ в противоречии с $f(s') \neq f(t')$.

Мы показали, что все $f(v_i)$ простые; но, возможно, не все они различны. Однако среди этих $f(v_i)$ каждый простой элемент S должен встречаться менее m_0 раз. Если бы для некоторого элемента a это было не так, мы могли бы определить h следующим образом:

$$h(v_{n+1}) = a^{k_0},$$

$$h(v) = f(v) \text{ для остальных переменных.}$$

То же рассуждение, что и для g, снова приводит к противоречию. Следовательно, согласно «принципу ящиков», среди этих n простых должно быть $\left[\frac{n}{m_0}\right] + 1$ различных простых. Лемма доказана.

Теорема 11. *Пусть число простых элементов в S конечно и равно p. Тогда $I(S)$ имеет конечный базис с не более чем $(pm_0+2)((m_0+k_0)^2-1)$ переменными.*

Доказательство. Тождество с d переменными должно иметь блок длины по крайней мере $q=d/r$, где $r=(m_0+k_0)^2-1$ — число типов блоков. Далее если оно не выводимо из $I_{d-1}(S)$, то, согласно лемме 12, $\left[\frac{q-1}{m_0}\right]+1\leqslant p$, или $q\leqslant pm_0+1$. Таким образом, $(d/r)\leqslant pm_0+2$ и

$$d\leqslant(pm_0+2)((m_0+k_0)^2-1).$$

Теорема доказана.

Во второй части разд. 4 мы получаем еще одну оценку числа переменных, необходимых для конечного базиса $I(S)$, в терминах числа образующих системы S. Попутно мы получим также значительную информацию о решетке эквациональных классов коммутативных полугрупп.

Лемма 12. *Пусть $N, D>0$. Тогда $x^N=x^{N+D}$ верно в S тогда и только тогда, когда $N\geqslant m_0$ и $k_0\mid D$.*

Доказательство. Предположим, что $N=m_0+p$ и $D=ck_0$. Тогда требуемое утверждение легко доказать индукцией по c. Если, наоборот, в S верно $x^N=x^{N+D}$, то $N\geqslant m_0$, поскольку m_0 выбиралось минимальным. Пусть

$$N-m_0=q_1k_0+r_1, \qquad 0\leqslant r_1<k_0,$$
$$D=q_2k_0+r_2, \qquad 0\leqslant r_2<k_0.$$

„Умножая“ обе части тождества $x^N=x^{N+D}$ на $x^{k_0-r_1}$, получим

$$x^{m_0+(q_1+1)k_0}=x^{m_0+(q_1+1)k_0+q_2k_0+r_2},$$

и, таким образом, $x^{m_0}=x^{m_0+r_2}$ выводимо как тождество системы S. Возможность $r_2\neq 0$ противоречит минимальному выбору k_0. Итак, $k_0\mid D$. Лемма доказана.

Лемма 13. *Пусть $e\in I(S)$. Если в e есть (p,q)-блок, то $k_0\mid p-q$.*

Доказательство. Пусть x — переменная этого блока; вместо всех остальных переменных подставим y. Тогда в S верно $x^py^B=x^qy^C$, а значит, $x^{p+B}=x^{q+C}$ и $x^{2p+B}=x^{2q+C}$. По лемме 14, разности показателей в последних двух тождествах должны делиться на k_0, т. е. $k_0\mid(p-q)+(B-C)$ и $k_0\mid(2p-2q)+(B-C)$. Таким образом, $k_0\mid p-q$. Лемма доказана.

Из леммы 13 видно, что оценка в теореме 11 улучшаема, поскольку для «непустых» типов блоков возможны не все r различных пар (p,q), а только пары с $k_0\mid p-q$.

Лемма 14. *Предположим, что существует* $d > 0$, *такое, что в* S *выполнено тождество с* $(0, d)$*-блоком, и пусть* d_0 *— наименьшее такое* d. *Тогда* $d_0 | d$, *и* S *удовлетворяет тождеству* $x^{m_0} = x^{m_0} y^d$.

Доказательство. Пусть y — переменная из $(0, d)$-блока такого тождества. Подставляя в него x вместо всех остальных переменных, получаем $x^B = x^C y^d$. Пусть $B \geqslant C$. (Случай $B < C$ рассматривается так же). Мы знаем, что для некоторого неотрицательного c имеем $B - C = ck_0$, так что в S выполнены $x^{B+ck_0} = x^{B-C} x^C y^d$ и $x^B = x^B y^d$. Далее, B должно быть больше m_0, скажем, $B = qk_0 + r$, $0 \leqslant r < k_0$. Умножая в $x^B = x^B y^d$ обе части на x^{k_0+r}, получаем верное в S тождество $x^{m_0+(q+1)k_0} = x^{m_0+(q+1)k_0} y^d$, а из него $x^{m_0} = x^{m_0} y^d$. В частности, выполнено $x^{m_0} = x^{m_0} y^{d_0}$. Пусть $d = qd_0 + r$, $0 \leqslant r < d_0$. В S верны тождества $x^{m_0} = x^{m_0} y^{qd_0+r}$ и $x^{m_0} = x^{m_0} y^r$, откуда $r = 0$. Итак, $d_0 | d$. Лемма доказана.

Сейчас мы увидим, какую роль играет число образующих в S. Для терма t будем обозначать через $|t|$ мощность множества $\mathrm{var}(t)$.

Лемма 15. *Пусть* S *имеет конечное множество образующих мощности* N *и пусть* $KL = PQ \in I(S)$ *— приведенное тождество* [1]) *с* $|K| \geqslant m_0 N$.

a) *Если* $\mathrm{var}(L) \cap \mathrm{var}(PQ) = \varnothing$, *то* $K = PQ \in I(S)$.

b) *Если мощность* $\mathrm{var}(K) \cap \mathrm{var}(P)$ *больше или равна* $m_0 N$, *причем* [2]) $\mathrm{var}(LQ) \cap \mathrm{var}(KP) = \varnothing$, то $K = P \in I(S)$.

c) *Если существует* d_0 *из леммы* 14, *то* $K = Kv^{d_0} \in I(S)$ *при любой переменной* v.

Доказательство. *Пусть* f *— произвольная функция назначений;* $f(K)$ *представимо в виде произведения образующих, которое содержит* (после «собирания» образующих) *некоторую из них, скажем* a, *с показателем не менее* m_0.

a) Выберем функцию назначения g так, что $g(w) = f(w)$ для $w \in \mathrm{var}(PQ) \cup \mathrm{var}(K)$ и $g(w) = a^{k_0}$ для остальных переменных.

$$f(PQ) = g(PQ) = g(KL) = g(K)\, g(L) = f(K)\, g(L) = \\ = Ba^{m_0} a^{ck_0} = Ba^{m_0} = f(K).$$

Поскольку f произвольна, $K = PQ \in I(S)$.

b) Выберем g так, что $g(w) = f(w)$ для $w \in \mathrm{var}(K) \cap \mathrm{var}(P)$ и $g(w) = a^{k_0}$ для остальных переменных. Аналогично a) легко про-

---

[1]) Здесь и в доказательстве теоремы 16 подразумевается, что последняя переменная в K отлична от первой в L; то же для P и Q. — *Прим. перев.*

[2]) Это условие добавлено при переводе. — *Прим. перев.*

верить, что $f(K) = f(P)$; таким образом, $K = P \in I(S)$.

c) $f(Kv^{d_0}) = f(K)f(v^{d_0}) = Ba^{m_0}f(v)^{d_0} = Ba^{m_0} = f(K)$.

Лемма доказана.

Теорема 16. *Если S имеет конечное множество из N образующих, то $I(S)$ имеет конечный базис, содержащий не более $2Nm_0 + 1$ переменных.*

Доказательство. Пусть $s = t \in I(S)$ — приведенное тождество. Мы будем считать, что задан порядок переменных, при котором в s и t сначала идут их общие переменные (в одинаковом порядке). Разобьем s и t на куски так, что данное тождество можно будет записать в виде

$$G_1G_2 \ldots G_kG = H_1H_2 \ldots H_kH,$$

где первое вхождение переменной, не входящей в обе части, попадает в G_k, и $|G_i| = |H_i| = Nm_0$ для $i = 1, \ldots, k-1$. Если возможно, пусть $|G_k|$ и $|H_k| = Nm_0$. (G и H могут быть пусты; возможно, $|G_k| < Nm_0$ или $|H_k| < Nm_0$.) Рассмотрим тождества e_i: $G_i = H_i (i < k)$ и e_k: $G_{k-1}G_kG = H_{k-1}H_kH$. (При $k = 1$ в e_k G_{k-1} и H_{k-1} считаем пустыми). Повторными применениями леммы 17b получаем, что e_i и e_k верны в S.

Если $k = 1$, то по лемме 15b в S выполнено тождество e'_k: $G_kv_1^{d_0} = H_kv_2^{d_0}$, где в зависимости от $|G_k|$ и $|H_k|$ мы допускаем и случай, когда эти v_i фактически отсутствуют. Тождество e_k выводимо из e'_k, поскольку по лемме 14 все переменные в G и в H должны иметь показатель, кратный d_0, так что G и H выразимы в виде B^{d_0}.

Если $k > 1$, мы „подразбиваем“ e_k, представляя его в виде $KL = PQ$, где $\mathrm{var}(K) = \mathrm{var}(P)$ и $\mathrm{var}(L) \cap \mathrm{var}(Q) = \varnothing$. Заметим, что $Nm_0 \leqslant |K| < 2Nm_0$. Лемма 15 снова дает, что в S выполнено e''_k: $Kv_1^{d_0} = Pv_2^{d_0}$, с тем же замечанием, что для e'_k. Тождество e_k выводимо из e''_k. Поскольку $\mathrm{var}(K) = \mathrm{var}(P)$, число переменных, входящих в e'_k или e''_k, не превосходит $2Nm_0 + 1$. Склеивая теперь $e_1, \ldots, e_{k-1}$ и, в зависимости от рассматриваемого случая, e'_k или e''_k, мы можем вывести исходное $s = t$. Теорема доказана.

Разумеется, можно себе представить, что существует равномерная граница b, такая, что всякая коммутативная полугруппа имеет конечный базис тождеств с менее чем b переменным. Чтобы показать, что дело обстоит не так, мы доказываем следующее утверждение.

Теорема 17. *Для всякого целого $n > 1$ можно построить коммутативную полугруппу S_n, такую, что $I(S_n)$ не имеет базиса с менее чем n различными переменными.*

Доказательство. Полугруппа S_n имеет две образующие a и b, для которых $a^n = a^{n+1}$ и $ab = ba = b^2 = b$. Таким образом, все элементы S_n представимы как $a, a_2, \ldots, a^n, b$. Скажем, что тождество обладает свойством L_k, если одна из его частей имеет вид $v_1 \ldots v_k$, где v_i различны, а другая часть содержит хотя бы одну переменную более одного раза. Непосредственно проверяются следующие утверждения: 1), 2), 3).

1) В тождестве полугруппы S_n переменная входит в одну из частей только тогда, когда она входит и в другую.

2) Тождество, обладающее свойством L_k для $k < n$, не верно в S_n.

3) Тождество e_n: $x_1 x_2 \ldots x_n = x_1 x_2 \ldots x_n^2$ верно в S_n и обладает свойством L_n.

4) e_n не выводимо из $I_{n-1}(S_n)$.

Докажем 4). Пусть такой вывод существует. Применяя теорему 1, мы видим, что если $M_i s_i N_i$ линеен и $s_i = t_i$ есть результат подстановки в $r = u \in I_{n-1}(S_n)$, то r линеен. В силу 2) u также линеен и должен содержать те же переменные, что и r. Таким образом, $M_i t_i N_i$ линеен, и вывод e_n существовать не может. Теорема доказана.

5. ПЕРМУТАТИВНЫЕ ПОЛУГРУППЫ

Наиболее естественное обобщение закона коммутативности — это, пожалуй, вводимые ниже «тождества перестановочности». Если предположить еще и равномерную периодичность, то, как мы сумеем доказать, «пермутативные» полугруппы имеют конечный базис тождеств.

Пусть π — подстановка натуральных чисел, сдвигающая лишь конечное множество M_π. Пусть $m_\pi = \max M_\pi$, $k_\pi = \min M_\pi$ и $d_\pi = m_\pi - k_\pi$. Пусть $n \geqslant m_\pi$, а C_π — представление π в виде произведения циклов без общих элементов. Введем в рассмотрение тождество

$$ {}_nC_\pi\colon\ x_1 x_2 \ldots x_n = x_{\pi(1)} x_{\pi(2)} \ldots x_{\pi(n)}. $$

например, ${}_4(1,3)$ обозначает тождество $x_1x_2x_3x_4 = x_3x_2x_1x_4$. Для всякого $d > 0$ определим следующую подстановку $\pi + d$:

$$ [\pi + d](k) = \begin{cases} k, & \text{если } k \leqslant d, \\ \pi(k-d) + d, & \text{если } k > d. \end{cases} $$

Другими словами, $\pi + d$ — это перенос π на d. Если, например, π задана как (245)(31), то $d_\pi = 4$ и $\pi + d_\pi$ задается как (689)(75). Легко проверить, что $\{{}_nC_\pi,\ {}_nC_\sigma\} \vdash {}_nC_{\pi\circ\sigma},\ {}_{n+d}C_{\pi},\ {}_{n+d}C_{n+d}$ для всех $d > 0$. Назовем S *пермутативной*, если она удовлетворяет какому-нибудь тождеству вида

$$ x_1 \ldots x_n = x_{\sigma(1)} \ldots x_{\sigma(n)}, $$

где σ — произвольная подстановка натуральных чисел. Ясно, что S пермутативна в том и только в том случае, если она удовлетворяет некоторому ${}_nC_\pi$.

Если, кроме того, M_π состоит ровно из трех чисел, мы говорим, что S *трициклична*, а если M_π имеет вид $\{k, k+1\}$, то S называем *почти коммутативной*. Сейчас мы покажем с помощью цепочки лемм, связывающих эти понятия, что всякая пермутативная полугруппа является почти коммутативной. В действительности мы установим, что если операция умножения в пермутативной полугруппе отображает *на* S, то S удовлетворяет тождеству $xyzw = xzyw$. Заключительный шаг будет состоять в доказательстве конечной базируемости тождеств всякой почти коммутативной равномерно периодической полугруппы.

Лемма 18. *Если S пермутативна, то S трициклична.*

Доказательство. Пусть в S верно ${}_cC_\pi$. Если $\sigma = \pi + d_\pi$, а $\bar{n} = n + d_\pi$, то в S должны быть верны также ${}_{\bar{n}}C_{\pi,\,\bar{n}}C_{\pi^{-1}},\ {}_{\bar{n}}C_\sigma$, ${}_{\bar{n}}C_{\sigma^{-1}}$ и ${}_{\bar{n}}C_\tau$, где $\tau = \pi \circ \sigma \circ \pi^{-1} \circ \sigma^{-1}$. Но заметьте, что $M_\pi \cap M_\sigma = \{m_\pi\} = \{k_\sigma\}$, а $M_\tau = \{\pi^{-1}(m_\pi),\ m_\pi,\ \sigma^{-1}(m_\pi)\}$. В действительности τ задается в виде $(m_\pi, \sigma^{-1}(m_\pi), \pi^{-1}(m_\pi))$. Лемма доказана.

Лемма 19. *Если S трициклична, то S почти коммутативна.*

Доказательство. Можно считать, что числа, образующие цикл, соответствующий данному тождеству T, стоят в цикле в их естественном порядке.

Случай I. T есть ${}_n(p, p+1, p+2)$. Рассмотрим сначала случай $n = 3$, $p = 1$, т. е. предположим, что в S верно тождество $x_1x_2x_3 = x_2x_3x_1$. Заменяя x_2 на x_2x_3, а x_3 на x_4, получим (1) $x_1x_2x_3x_4 = x_2x_3x_4x_1$. В том же тождестве ${}_3(123)$ заменим теперь x_1 на x_2x_3, x_2 на x_4, x_3 на x_1; получим $x_2x_3x_4x_1 = x_4x_1x_2x_3$. Сочетая это с (1), получаем в S $x_1x_2x_3x_4 = x_4x_1x_2x_3$, или ${}_4(1432)$. Теперь, исходя из ${}_3(123)$, мы легко доказываем, что в S верны ${}_4(123)$ и отсюда ${}_4[(123)(1432)]$, а это и есть ${}_4(34)$. Ясно, что в действительности это рассуждение применимо и в общем случае и что

$$n(p, p+1, p+2) \vdash_{n+1} (p+2, p+3).$$

Случай II. T есть ${}_n(p, q, q+1)$, где $p+1 < q$. Из T вытекает

$${}_{n+1}[(p+1, q+1, q+2)(p, q+1, q+2)(p, q, q+1) \times \times (p+1, q+1, q+2)],$$

а это есть ${}_{n+1}(q+1, q, q+2)$, из чего следует ${}_{n+1}(q, q+1, q+2)$, и все свелось к случаю I.

Случай III. T есть ${}_n(r, p, q)$, где $p + 1 < q$. Из T вытекает

$${}_{n+1}[(r,\ p,\ q)(r,\ p,\ q+1)(r,\ p,\ q+1)];$$

а это есть ${}_{n+1}(p, q, q+1)$, и мы возвратились к случаю II.
Лемма доказана.

Лемма 20. *Если S почти коммутативная и равномерно периодическая, то тождества S конечно базируемы.*

Доказательство. Пусть S (m_0, k_0)-р.п. и почти коммутативна. Тогда каждое ее тождество $s_1 = s_2$ эквивалентно[1]) тождеству «нормального вида» $A_1B_1C_1D_1 = A_2B_2C_2D_2$, в котором:

1) нет показателей, больших $m_0 + k_0$;
2) A_i — это первые m букв терма s_i (x^q считается за q букв);
3) D_i — это последние n букв терма s_i (с вычеркнутым A_i);
4) B_i — терм в приведенной коммутативной форме, содержащей только переменные из A_1, A_2, D_1 или D_2;
5) C_i — терм в приведенной коммутативной форме, не содержащий переменных, встречающихся в A_1, A_2, D_1 или D_2.

С точностью до обозначения переменных существует лишь конечное число таких шестерок $A_1B_1D_1A_2B_2D_2$. Предположим теперь, что тождества S не конечно базируемы. Тогда найдется последовательность $e_1, e_2, \ldots$ «нормальных» тождеств из $I(S)$, такая, что ни одно e_i не выводимо из предшествующих тождеств этой последовательности. Следовательно, должна найтись подпоследовательность $f_1, f_2, \ldots$, обладающая тем же самым свойством, в которой «ABD-шестерки» у всех тождеств одни и те же (с точностью до переименования переменных). Но рассуждение с помощью (p, q)-блоков в применении к C_i, совершенно такое же, как использованное в теореме 9 для коммутативных полугрупп, оказывается и здесь достаточным для приведения к противоречию. Лемма доказана.
Комбинируя приведенные выше леммы, имеем еще один положительный результат.

Теорема 21. *Всякая равномерно периодическая пермутативная полугруппа имеет конечный базис тождеств.*
Заметьте, что условие равномерной периодичности здесь существенно, поскольку FS_ω/E из теоремы 2 — это пример пермутативной не р.п. полугруппы, тождества которой не конечно базируемы.

Следствие 22. *Все трехэлементные полугруппы имеют конечный базис тождеств.*

---

[1]) В силу системы из двух тождеств $\{x^{m_0} = x^{m_0+k_0},\ x_1 \ldots x_m yzw_1 \ldots w_n = x_1 \ldots x_m zyw_1 \ldots w_n\}$. — *Прим. перев.*

Доказательство. Все типы неизоморфных и неантиизоморфных полугрупп с тремя, четырьмя и пятью элементами перечислены [2, 10]. Из восемнадцати типов трехэлементных полугрупп семнадцать, как легко проверить, пермутативны, удовлетворяя (по крайней мере) тождеству $xyzw = xzyw$. Один остающийся тип [1]) удовлетворяет тождеству $xyx = xy$ и, как можно показать, имеет конечный базис тождеств.

СПИСОК ЛИТЕРАТУРЫ [2])

1. Cohn P. M., Universal Algebra, Harper & Row, New York, 1965. (Русский перевод: Кон П. Универсальная алгебра, «Мир», М., 1968.)
2. Forsythe G. E., SWAC computes 126 distinct semigroups of order 4, *Proc. Amer. Math. Soc.*, **6** (1955), 443—447.
3. Higman G., Some remarks on varieties of groups, *Quarterly J. of Math. Oxford Ser.*, **10** (1959), 165—178.
4. Lyndon R. C., Identities in two-valued calculi, *Trans. Amer. Math. Soc.*, **71** (1951), 457—465. (Русский перевод: Линдон Р. К., Тождества в двузначных исчислениях, Киб. сб. 1, М., ИЛ, 1960, 234—245).
5. Lyndon R. C., Identities in finite algebras, *Proc. Amer. Math. Soc.*, **5** (1954), 8—9. (Русский перевод: Линдон. Р. К., Тождества в конечных алгебрах, Кибернетический сборник, вып. I, ИЛ, М., 1960, стр. 246—248.)
6. Lyndon R. C., Two notes on nilpotent groups, *Proc. Amer. Math. Soc.*, **3** (1952), 579—583.
7. Neumann B. H., Identical relations in groups, *Math. Ann.*, **114** (1937), 506—525.
8. Oates S., Powell M. B., Identical relations in finite groups, *Journal of Algebra*, **1** (1964), No. 1, 11—39.
9. Scott D., Equationally complete extensions of finite algebras, *Nederl. Akad. Wetensch. Proc.*, Ser. A, **59** (1956), 35—38.
10. Selfridge J. L., On finite semigroups, Ph. D. dissertation, University of California, Los Angeles, June 1958.
11. Вишин В. В., Тождественные преобразования в четырехзначной логике, *Докл. АН СССР*, **150** (1963), № 4, 719—721.

*12. Мурский В. Л., Существование в трехзначной логике замкнутого класса с конечным базисом, не имеющего конечной полной системы тождеств, *Докл. АН СССР*, **163** (1965), № 4, 815—818.

*13. Ольшанский А. Ю., О проблеме конечного базиса тождеств в группах, *Изв. АН СССР*, **34** (1970), № 2, 376—384.

*14. Адян С. И., Бесконечные неприводимые системы групповых тождеств, *Изв. АН СССР*, **34** (1970), № 4, 715—734.

*15. Vaughan-Lee M. R., Uncountably many varieties of groups, *Bull. London Math. Soc.*, **2** (1970), 3, 280—286.

[1]) Речь идет о полугруппе с элементами, скажем, 0, 1, 2 и умножением $0x \equiv 0$, $1x \equiv x$, $2x \equiv 2$. — *Прим. перев.*

[2]) Работы, добавленные при переводе, отмечены звездочкой. — *Прим. ред.*

Веса многочленов и кодовые конструкции [1)]

Дж. Л. Месси, Д. Дж. Кастелло, Й. Юстесен

I. ВВЕДЕНИЕ

В этой статье будет показано, что многочлены $(x-c)^i$, $i=0,1,2,\ldots$, где c — любой ненулевой элемент $GF(q)$, обладают тем фундаментальным свойством (которое мы называем свойством сохранения веса), что любая линейная комбинация этих многочленов с коэффициентами из $GF(q)$ имеет хэммиговский вес, не меньший чем вес входящего в нее многочлена минимальной степени. Оно доказано отдельно в разд. II. А для случая, когда $GF(q)$ имеет характеристику $p=2$, так как это самый простой случай и так как он имеет наиболее интересные применения. Такие применения (1) к кодам Рида — Маллера, (2) к новому классу двоичных циклических кодов с кратными корнями, (3) к двум новым классам двоичных сверточных кодов, полученных из двоичных циклических кодов, и (4) к двум новым классам двоичных сверточных кодов, полученным из кодов Рида — Соломона, содержатся в разд. II. Б, II. В, II. Г и II.Д соответственно.

В разд. III. А мы приведем новый класс констациклических кодов над полями $GF(q)$ с характеристикой p, большей чем 2, которые являются кодами с достижимыми максимальными расстояниями и имеют простой алгебраический алгоритм декодирования. Затем этот класс кодов используется в разд. III. Б в качестве основы для доказательства по индукции свойства сохранения веса для произвольного конечного поля $GF(q)$. Наконец, в разд. III. В будет приведено новое p-ичное обобщение кодов Рида — Маллера и будет дан новый класс констациклических подкодов этих p-ичных кодов, имеющих то же самое минимальное расстояние, что и порождающие коды.

II. ДВОИЧНЫЙ СЛУЧАЙ

A. Свойство сохранения веса в полях характеристики 2

Пусть c — ненулевой элемент конечного поля $GF(q)$, где $q=2^r$ для некоторого целого r. Так как многочлены $(x+c)^i$, $i=0,1,2,\ldots$ содержат в точности по одному многочлену каждой степени, то они

[1)] Massey J. L., Costello D. J., Jr., Justesen J., Polynomial weights and code constructions, *IEEE Transactions on Information Theory*, **IT-19**, № 1, (1973), 101—110.

образуют базис векторного пространства всех многочленов над $GF(2^r)$, и, следовательно, каждый многочлен $P(x)$ над $GF(2^r)$ может быть единственным образом выражен в виде линейной комбинации этих многочленов. После этого обозначим через $W[P(x)]$ *хэмминговский вес* многочлена $P(x)$, т. е. число его ненулевых коэффициентов. Следующая теорема связывает $W[P(x)]$ с разложением $P(x)$ в указанном выше базисе.

Теорема 1.1. *Пусть I — какое-либо конечное непустое множество неотрицательных целых чисел, среди которых минимальным является целое число $i_{\min}$, и пусть*

$$P(x) = \sum_{i \in I} b_i (x + c)^i,$$

где c и каждое b_i — ненулевые элементы поля $GF(2^r)$. Тогда

$$W[P(x)] \geqslant W\left[(x + c)^{i_{\min}}\right]. \tag{1}$$

Доказательство. Проведем индукцию по наибольшему целому $i_{\max}$ из I. Простая проверка показывает, что (1) справедливо для $i_{\max} < 2^2$. Предположим теперь, что (1) имеет место для $i_{\max} < 2^n$, и покажем, что (1) справедливо для $i_{\max} < 2^{n+1}$.

Разобьем I на множества I_0 и I_1, где I_0 содержит те и только те i из I, для которых $i < 2^n$. Тогда

$$\sum_{i \in I_1} b_i (x + c)^i = (x + c)^{2^n} \sum_{i \in I_1} b_i (x + c)^{i - 2^n},$$

что после обозначения через $P_1(x)$ суммы в правой части, которая представляет собой многочлен степени, меньшей чем 2^n, принимает вид

$$\sum_{i \in I_1} b_i (x + c)^i = x^{2^n} P_1(x) + c^{2^n} P_1(x). \tag{2}$$

Аналогично обозначим через $P_0(x)$ многочлен степени, меньшей чем 2^n, который задается равенством

$$P_0(x) = \sum_{i \in I_0} b_i (x + c)^i.$$

Тогда из (2) и определений I_0 и I_1 получаем

$$P(x) = [P_0(x) + c^{2^n} P_1(x)] + x^{2^n} P_1(x). \tag{3}$$

Предположим сначала, что $P_0(x) = 0$. Тогда из (3) имеем $W[P(x)] = 2W[P_1(x)]$. Так как $P_1(x)$ имеет степень, меньшую чем 2^n, то, согласно индукции, получаем

$$W[P_1(x)] \geqslant W\left[(x + c)^{i_{\min} - 2^n}\right]. \tag{4}$$

Кроме того,

$$W\left[(x+c)^{i_{\min}}\right] = W\left[(x+c)^{2^n}(x+c)^{i_{\min}-2^n}\right] = \\ = W\left[(x^{2^n}+c^{2^n})(x+c)^{i_{\min}-2^n}\right] = 2W\left[(x+c)^{i_{\min}-2^n}\right],$$

так что с помощью (4) получаем $W[P(x)] = 2W[P_1(x)] \geqslant \geqslant W\left[(x+c)^{i_{\min}}\right]$, что и следовало доказать. Предположим теперь, что $P_0(x) \neq 0$. Тогда из (3) теперь имеем

$$W[P(x)] \geqslant W[P_0(x)], \tag{5}$$

так как любой ненулевой член в $P_0(x)$, сокращенный при добавлении $c^{2^n}P_1(x)$, должен вновь появиться как ненулевой член в $x^{2^n}P_1(x)$. Поскольку $P_0(x)$ имеет степень, меньшую чем 2^n, то, согласно индукции, $W[P_0(x)] \geqslant W[(x+c)^{i_{\min}}]$, что совместно с (5) дает (1), и теорема доказана.

Отметим, что в частном случае $r = 1$ теорема 1.1 эквивалентна тому, что двоичная матрица $2^n \times 2^n$, у которой $(i+1)$-я строка является последовательностью коэффициентов в $(x+1)^i$, т. е. $(i+1)$-й строкой треугольника Паскаля, приведенного по модулю 2, обладает тем свойством, что любая сумма ее строк имеет хэммингов-ский вес, по меньшей мере равный весу самой верхней строки, которая входит в сумму. При $n = 3$ эта матрица имеет вид

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Эти матрицы имеют определенное значение в теории переключательных схем; в этой теории Препарата [1] указал некоторые другие интересные свойства этих матриц, в частности то, что они совпадают с обратными себе.

Б. Двоичные коды Рида — Маллера

Мы часто будем использовать следующий результат.

Лемма 1. *Пусть c — ненулевой элемент конечного поля $GF(q)$ с характеристикой p, и пусть i — неотрицательное целое число в p-ичной записи $[i_{m-1}, \ldots, i_1, i_0]$. Тогда*

$$W[(x+c)^i] = \prod_{j=0}^{m-1}(i_j+1), \tag{6a}$$

или для частного случая, когда $p = 2$,

$$W[(x+1)^i] = 2^{w(i)}, \tag{6б}$$

где $w(i)$ *обозначает число единиц в двоичной записи* i.

Доказательство. Чтобы доказать эту лемму, заметим вначале, что $W[(x+c)^i]$ представляет собой просто число целых чисел k, $0 \leqslant k \leqslant i$, таких, при которых биномиальный коэффициент $\binom{i}{k}$ не равен нулю по модулю p. Но, согласно теореме Лукаса [2, стр. 121],

$$\binom{i}{k} = \prod_{j=0}^{m-1} \binom{i_j}{k_j} \bmod p, \tag{7}$$

где i_j и k_j символы в p-ичных записях i и k соответственно и где по определению биномиальный коэффициент, у которого нижнее число превышает верхнее, равен нулю. Теперь из (7) следует, что имеется в точности $i_j + 1$ возможностей выбрать k_j, а именно $0, 1, 2, \ldots, i_j$, так, чтобы соответствующий биномиальный коэффициент в (7) был отличен от нуля по модулю p. Таким образом, отсюда следует (6а), и лемма доказана.

Сейчас мы можем использовать теорему 1.1 для простого вывода двоичных кодов Рида — Маллера. Пусть m и u — какие-либо два положительных целых числа, такие, что $u \leqslant m$. Рассмотрим двоичную матрицу G с $n = 2^m$ столбцами, строки которой содержат последовательность коэффициентов в $(x+1)^i$ для всех i, таких, что $i < n$ и $w(i) \geqslant u$. Число таких i, т. е. число строк в G, в точности равно

$$k = \sum_{j=u}^{m} \binom{m}{j} = \sum_{j=0}^{m-4} \binom{m}{j}.$$

Например, при $m = 3$ и $u = 2$ имеем

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Возьмем теперь G в качестве порождающей матрицы двоичного (m, k)-кода с проверками на четность. Из теоремы 1.1 следует, что каждое ненулевое кодовое слово, т. е. каждая сумма одной или большего числа строк матрицы G, имеет хэмминговский вес, по меньшей мере равный 2^u, так как наш выбор G гарантирует, что такая сумма соответствует сумме многочленов $(x+1)^i$, для которой $w(i_{\min}) \geqslant u$, и, следовательно, согласно лемме 1,

$W[(x+1)^{i_{\min}}] \geqslant 2^u$. Более того, некоторые строки G имеют хэмминговский вес, в точности равный 2^u, так что минимальное расстояние кода равно $d = 2^u$. Этот код в точности является кодом Рида — Маллера u-го порядка длины $n = 2^m$, и на самом деле строки, которые мы выбрали для G, являются теми же самыми, которые были выбраны Ридом [3] (с точностью до тривиальной инверсии каждой строки).

Вычисление k и d для двоичных кодов Рида — Маллера, которое приведено здесь, является существенным упрощением известных ранее методов.

B. Новый класс двоичных циклических кодов с кратными корнями

При $2 \leqslant u \leqslant m$ рассмотрим двоичный многочлен

$$g(x) = (x+1)^i, \tag{8}$$

где

$$i = 2^m - 2^{m-u+1} + 1. \tag{9}$$

Двоичная запись i имеет вид

$$[i_{m-1}, \ldots, i_1, i_0] = [1\ 1 \ldots 1\ 0\ 0 \ldots 0\ 1], \tag{10}$$

где серия из нулей имеет длину $m - u$. Заметим, что $2^{m-1} < i < 2^m$; это влечет за собой то, что $g(x)$ — делитель $x^n + 1$ при $n = 2^m$ (так как в этом случае $x^n + 1 = (x+1)^n$), но не при меньших n. Следовательно, $g(x)$ порождает двоичный циклический (n, k)-код с $n = 2^m$ и $n - k = i$, который, согласно (9), дает $k = 2^{m-u+1} - 1$. Строки порождающей матрицы G для этого циклического кода могут быть выбраны в виде $g(x)(x+1)^j$ для $j = 0, 1, \ldots, k-1$ или эквивалентно в виде $(x+1)^j$ для $i \leqslant j < n$. Но из (10) следует, что $w(j) \geqslant w(i)$ для $i \leqslant j < n = 2^m$, так как двоичная запись j должна иметь $u - 1$ символов 1 и в первых позициях и по меньшей мере один символ 1 среди своих последних $m - u + 1$ позиций. Таким образом, строки G являются подмножеством строк G, как было в разд. II. Б. Следовательно, циклический код, порождаемый $g(x)$, является подкодом двоичного кода Рида — Маллера u-го порядка, имеющим то же самое минимальное расстояние, что и порождающий его код, так как $W[g(x)] = 2^u = d$. (В дальнейшем циклический код, у которого неприводимые делители и, следовательно, корни $g(x)$ имеют кратность, большую чем единица, будем называть циклическим кодом с *кратными корнями*.) Тем самым мы доказали следующую теорему.

Теорема 2. *При $2 \leqslant u \leqslant m$ двоичный циклический $(n = 2^m, k = 2^{m-u+1} - 1)$-код с кратными корнями, порождаемый $g(x) = (x+1)^{n-k}$, является подкодом двоичного кода Рида — Маллера*

u-го *порядка, имеющим такое же минимальное расстояние* $d = 2^u$, *что и порождающий его код.*

Хотя циклические коды с кратными корнями из теоремы 2 в общем случае хуже сравнимых с ними кодов Боуза—Чоудхури—Хоквингема (БЧХ-кодов), они имеют два интересных свойства, которые могут обусловить их использование в некоторых практических ситуациях.

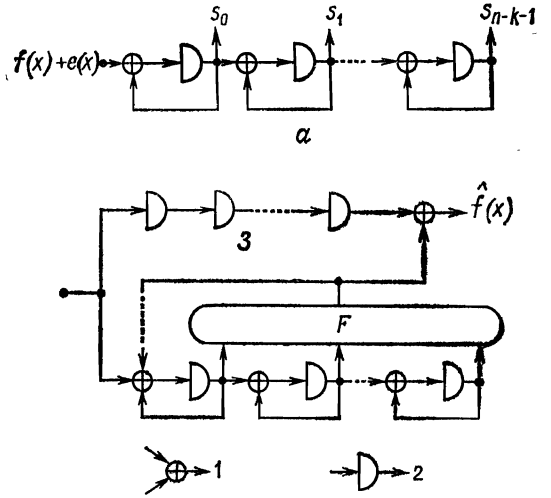


Рис. 1. *а* — схема, формирующая синдром для двоичного циклического кода с кратными корнями из теоремы 2; *б* — декодер.
1—сумматор по модулю 2; *2*—единичная задержка; *3*—*n* единичных задержек.

Первое свойство состоит в том, что для кодов с кратными корнями очень легко можно построить схемы вычисления синдрома и декодирования. Эти схемы используют логические элементы, соответствующие множителю $(x+1)$ в таких комбинациях, которые сами по себе приводят к реализациям на интегральных цепях. Рассмотрим схему, изображенную на рис. 1, *а*. Легко проверить, что если многочлен $P(x) = P_{n-1}x^{n-1} + \ldots + P_1x + P_0$ записан в этой схеме (коэффициенты при высших степенях расположены сначала), то содержимое

$$s(x) = s_0 + s_1x + \ldots + s_{n-k-1}x^{n-k-1},$$

когда P_j — единственный ненулевой коэффициент, будет

$$P_j(x+1)^j \bmod (x^{n-k});$$

следовательно, в силу линейности реакций на произвольный многочлен $P(x)$ будет

$$s(x) = P(x+1) \bmod (x^{n-k})$$

(здесь и в дальнейшем через $P(x) \bmod Q(x)$ обозначается остаток от деления многочлена $P(x)$ на многочлен $Q(x)$). В частности, когда $P(x) = f(x) + e(x)$, где $f(x) = a(x)g(x) = a(x)(x+1)^{n-k}$ — кодовое слово кода с кратными корнями, а $e(x) = e_0 + e_1 x + \ldots + e_{n-1}x^{n-1}$ — ошибка в канале, имеем

$$s(x) = e(x+1) \bmod (x^{n-k}) = \sum_{j=0}^{n-1} e_j (x+1)^j \bmod (x^{n-k}), \qquad (11)$$

откуда видно, что $s(x)$ зависит лишь от ошибок и, таким образом, является истинным синдромом. Предположим теперь, что реализована логическая функция F, которая производит при декодировании оценку $\hat{e}_{n-1}$ первого ошибочного символа e_{n-1} по синдрому $s(x)$. После дальнейших i сдвигов логической схемы, изображенной на рис. 1, *а*, содержимым регистра синдрома будет

$$\text{„}s(x)\text{“} = \sum_{j=0}^{n-1} e_{j-i} (x+1)^j \bmod (x^{n-k})$$

(где считается, что $e_{-h} = e_{n-h}$), так что та же самая функция F в дальнейшем будет производить соответствующую оценку $\hat{e}_{n-i-1}$ значения e_{n-i-1}. Таким образом, подобно методу, впервые предложенному Меггит [4] для циклических кодов, весь декодер для кода с кратными корнями может быть выполнен так, как показано на рис. 1, *б*. Соединение, изображенное точками в этом декодере, производится, если требуется устранить влияние правильно декодированных ошибочных символов на синдром так, чтобы его содержимое было нулевым после успешного декодирования полного блока.

Второе свойство состоит в следующем: так как коды с кратными корнями являются подкодами кодов Рида — Маллера, имеющими то же самое минимальное расстояние, что и порождающие их коды, мажоритарный алгоритм декодирования Рида [3] может быть использован в качестве простой реализации декодирующей функции F (рис. 1, *б*). Например, для (8,3)-кода с кратными корнями и с $d = 2^2 = 4$ легко проверить, что s_3, s_4 и $s_0 + s_3 + s_4$ образуют множество трех проверок на четность, ортогональных относительно $e_{n-1} = e_7$ [5], так что F можно реализовать на основе следующего правила декодирования: $\hat{e} = 0$, если ни одна проверка или только одна из этих проверок имеет значение 1; $\hat{e} = 1$, если все три проверки имеют значение 1; и обнаружение двух или более ошибок объявляется, если две из этих проверок имеют значение 1.

Г. Построение двоичных сверточных кодов по двоичным циклическим кодам

Теперь мы покажем, что теорема 1.1 дает ключ к использованию известных циклических кодов для построения сверточных кодов

с большим свободным расстоянием. Для того чтобы облегчить рассмотрение, вначале мы представим теорему 1.1 в следующих двух эквивалентных формах.

Теорема 1.2. *Для любого многочлена $Q(x)$ над $GF(2^r)$, любого ненулевого элемента c из $GF(2^r)$ и любого неотрицательного целого N имеет место неравенство*

$$W[Q(x)(x+c)^N] \geqslant W[(x+c)^N] \cdot W[Q(c)]. \tag{12}$$

Доказательство. Представим сначала $Q(x)$ в виде

$$Q(x) = \sum_{i=0}^{t} b_i (x+c)^i$$

и заметим, что $Q(c) = b_0$. Если $b_0 = 0$, то $W[Q(c)] = 0$ и (12) тривиально выполняется. Если $b_0 \neq 0$, то $W[Q(c)] = 1$, и, полагая $P(x)$ в теореме 1.1 равным $(x+c)^N Q(x)$, получаем $i_{\min} = N$, так что (12) снова справедливо.

Теорема 1.3. *Для любого многочлена $P(x)$ над $GF(2^r)$, любого ненулевого элемента c из $GF(2^r)$ и любых неотрицательных целых чисел n и N*

$$W[P(x)(x^n+c)^N] \geqslant W[(x+c)^N]\, W[P(x) \bmod (x^n+c)]. \tag{13}$$

Доказательство. Полагая $P(x) = Q_1(x^n) + xQ_2(x^n) + \ldots + x^{n-1}Q_n(x^n)$,
получаем

$$W[P(x)(x^n+c)^N] = \sum_{i=1}^{n} W[Q_i(x^n)(x^n+c)^N]. \tag{14}$$

Теперь, отождествляя x^n в правой части (14) с x в теореме 1.2, будем иметь

$$W[P(x)(x^n+c)^N] \geqslant \sum_{i=1}^{n} W[Q_i(c)] \cdot W[(x+c)^N] =$$

$$= W[(x+c)^N] \cdot \sum_{i=1}^{n} W[Q_i(c)] = W[(x+c)^N] \cdot W\left[\sum_{i=1}^{n} x^{i-1} Q_i(c)\right],$$

и условие (13) теперь легко получается, если заметить, что последняя сумма точно равна многочлену $P(x) \bmod (x^n + c)$. Теорема доказана.

Для того чтобы описать сверточные коды со скоростью $R = 1/v$, воспользуемся обозначениями, использованными Месси [6]. Последовательность $i_0, i_1, i_2, \ldots$ информационных битов описывается с помощью ее D-преобразования

$$I(D) = i_0 + i_1 D + i_2 D^2 + \ldots$$

и сверточный код определяется многочленом

$$G(D) = G_1(D^v) + DG_2(D^v) + \ldots + D^{v-1}G_v(D^v). \qquad (15)$$

(Составляющие многочлены $G_j(D)$ теперь обычно называются «многочленами, порождающими код» для сверточного кода [5].) Если M — максимальная степень многочленов, порождающих код, то M называется *памятью* кода, а $n_A = (M+1)v$ — *ограничительной длиной.* Закодированной последовательностью является последовательность $t_0, t_1, t_2, \ldots$, для которой D-преобразование задается равенством

$$T(D) = I(D^v)\,G(D),$$

которое, конечно, является многочленом всегда, когда $I(D)$ — многочлен.

Свободное расстояние d_{free} сверточного кода определяется как минимум веса $W[T(D)]$, взятый по всем $I(D) \neq 0$. Говорят, что код является *катастрофическим*, или имеет катастрофическое распространение ошибки, если величина $I(D)$, не являющаяся многочленом, может привести к многочлену $T(D)$ [7, 8]. Известным необходимым и достаточным условием [7] того, что код с $R = 1/v$ некатастрофический, является условие

$$\text{НОД}\,\{G_1(D), G_2(D), \ldots, G_v(D)\} = 1, \qquad (17)$$

где НОД обозначает наибольший общий делитель и где предполагается (без существенной потери общности), что по меньшей мере один из многочленов, порождающих код, имеет ненулевой постоянный член. Было убедительно показано, что d_{free} представляет собой основной параметр вероятности ошибки при декодировании по методу максимального правдоподобия (Витерби) сверточных кодов [9] и при декодировании по методу «почти» максимального правдоподобия, подобном последовательному декодированию [10].

Используем теперь теорему 1.3 для построения двоичных сверточных кодов с большим d_{free} на основе двоичных циклических кодов. Далее здесь мы будем всегда обозначать через $g(x)$ порождающий многочлен циклического кода, через d_g — минимальное расстояние такого кода, через $h(x) = (x^n+1)/g(x)$ — дуальный многочлен, через d_h — минимальное расстояние дуального кода и через n — длину как того, так и другого кода.

Теорема 3. *Если* $g(x)$ *порождает циклический код над* $GF(2^r)$ *нечетной длины* n, *то для любого положительного целого* m 2^r*-ичный сверточный код с* $R = 1/v$, $v = 2m$, *определяемый с помощью* $G(D) = g(D)$, *является некатастрофическим и имеет* $d_{\text{free}} \geqslant \geqslant \min\{d_g, 2d_h\}$.

Доказательство. Так как n нечетное, то $g(x)$ не имеет кратных корней. Но

$$G(D) = g(D) = \sum_{j=1}^{v} D^{j-1} G_j(D^{2m}) = \sum_{j=1}^{v} D^{j-1} \hat{G}_j(D^m)^2, \qquad (18)$$

где $\hat{G}_j(D)$ обозначает многочлен, полученный из $G_j(D)$ при замене каждого коэффициента на его корень квадратный; корень квадратный существует и является однозначным для каждого элемента $GF(2^r)$. Из (18) получаем, что любой неприводимый многочлен, который делит каждый многочлен, порождающий код, приводит к неприводимому делителю $g(x)$ с кратностью по меньшей мере 2. Следовательно, НОД $\{G_1(D), G_2(D), \ldots, G_v(D)\} = 1$, так что сверточный код является некатастрофическим.

Для любого многочлена $I(D) \neq 0$ имеем

$$T(D) = I(D^{2m}) G(D) = \hat{I}(D^m)^2 g(D), \qquad (19)$$

где снова коэффициенты $\hat{I}(D)$ являются корнями квадратными коэффициентов $I(D)$. Из (19) теперь следует, что

$$T(D) = P(D)\, g(D)^{2i+1} h(D)^{2j}, \qquad (20)$$

где $i \geqslant 0$, $j \geqslant 0$, а $P(D)$ — ненулевой многочлен, который не делится ни на $g(D)$, ни на $h(D)$.

Предположим сначала, что $i \geqslant j$. Тогда из (20)

$$T(D) = P(D)\, g(D)^{2(i-j)+1} (D^n + 1)^{2j},$$

что по теореме 1.3 приводит к

$$W[T(D)] \geqslant W[(D+1)^{2j}] \cdot W[P(D)\, g(D)^{2(i-j)+1} \bmod (D^n + 1)].$$

Первый сомножитель в правой части по меньшей мере равен 1; далее, $P(D) g(D)^{2(i-j)+1} \bmod (D^n + 1)$ является ненулевым кодовым словом циклического кода, порождаемого $g(x)$ и, таким образом, имеет хэмминговский вес, по меньшей мере равный d_g, так что

$$W[T(D)] \geqslant d_g. \qquad (21)$$

Предположим теперь, что $i < j$. Из (20) имеем

$$T(D) = P(D)\, h(D)^{2(j-i)-1} (D^n + 1)^{2i+1},$$

что по теореме 1.3 приводит к

$$W[T(D)] \geqslant [(D+1)^{2i+1}] \cdot W[P(D)\, h(D)^{2(j-i)-1} \bmod (D^n + 1)].$$

Первый сомножитель в правой части по меньшей мере равен 2; аргумент второго сомножителя представляет собой ненулевое кодо-

Таблица 1

Некоторые двоичные сверточные коды, полученные с помощью конструкции, приведенной в теореме 3

Использованный циклический код	R	n_A	d_{free}
КВ (7,4)-код $d_g = 3$, $d_h = 4$	$1/2$	4	$\geqslant 3$
	$1/4$	4	$\geqslant 3$
КВ (7,3)-код, $d_g = 4$, $d_h = 3$	$1/2$	6	$\geqslant 4$
	$1/4$	8	$\geqslant 4$
БЧХ (15,8)-код, $d_g = 4$, $d_h = 5$	$1/2$	8	$\geqslant 4$
	$1/4$	8	$\geqslant 4$
КВ (17,8)-код, $d_g = 6$, $d_h = 5$	$1/2$	10	$\geqslant 6$
	$1/4$	12	$\geqslant 6$
БЧХ (15,5)-код, $d_g = 7$, $d_h = 4$	$1/2$	12	$\geqslant 7$
	$1/4$	12	$\geqslant 7$
КВ (23,12)-код, $d_g = 7$, $d_h = 8$	$1/2$	12	$\geqslant 7$
	$1/4$	12	$\geqslant 7$
КВ (23,11)-код, $d_g = 8$, $d_h = 7$	$1/2$	14	$\geqslant 8$
	$1/4$	16	$\geqslant 8$
БЧХ (31,11)-код, $d_g = 11$, $d_h = 6$	$1/2$	22	$\geqslant 11$
	$1/4$	24	$\geqslant 11$
КВ (47,24)-код, $d_g = 11$, $d_h = 12$	$1/2$	24	$\geqslant 11$
	$1/4$	24	$\geqslant 11$
КВ (47,23)-код, $d_g = 12$, $d_h = 11$	$1/2$	26	$\geqslant 12$
	$1/4$	28	$\geqslant 12$
БЧХ (63,30)-код, $d_g = 13$, $d_h \geqslant 8$	$1/2$	34	$\geqslant 13$
	$1/4$	36	$\geqslant 13$
БЧХ (63,24)-код, $d_g = 15$, $d_h \geqslant 8$	$1/2$	40	$\geqslant 15$
	$1/4$	40	$\geqslant 15$
КВ (79,40)-код, $d_g = 15$, $d_h = 16$	$1/2$	40	$\geqslant 15$
	$1/4$	40	$\geqslant 15$
КВ (79,39)-код, $d_g = 16$, $d_h = 15$	$1/2$	42	$\geqslant 16$
	$1/4$	44	$\geqslant 16$
КВ (89,44)-код, $d_g = 18$, $d_h = 17$	$1/2$	46	$\geqslant 18$
	$1/4$	48	$\geqslant 18$
КВ (103,52)-код, $d_g = 19$, $d_h = 20$	$1/2$	52	$\geqslant 19$
	$1/4$	52	$\geqslant 19$
КВ (103,51)-код, $d_g = 20$, $d_h = 19$	$1/2$	54	$\geqslant 20$
	$1/4$	56	$\geqslant 20$

вое слово циклического кода, порожденного $h(x)$, и, таким образом, имеет хэмминговский вес, по крайней мере равный d_h, так что

$$W[T(D)] \geqslant 2h_h. \tag{22}$$

Теперь утверждение теоремы следует из (21) и (22).

Необходимо заметить, что нижняя граница для d_{free}, задаваемая теоремой 3, не зависит от m и, следовательно, от скорости R сверточного кода, полученного из циклического кода. Поэтому граница будет наиболее точной при $m = 1$, т. е. при $R = 1/2$, так как действительное d_{free} может лишь увеличиваться с увеличением m. Наилучшие сверточные коды получаются при выборе циклического кода так, чтобы $d_g \approx 2d_h$. В таблице 1 перечислены несколько двоичных ($r = 1$) сверточных кодов, полученных с помощью теоремы 3 как для $R = 1/2$, так и для $R = 1/4$, и в ней указан тот циклический код, который использован при построении.

Следующая теорема указывает до некоторой степени менее очевидный способ построения сверточных кодов по циклическим кодам.

Теорема 4. *Если $g(x)$ порождает циклический код над $GF(2^r)$ нечетной длины n, то для любого положительного целого m сверточный 2^r-ичный код со скоростью $R = 1/v$, $v = 4m$, определяемый с помощью $G(D) = g(D)^2 + Dh(D)^2$, является некатастрофическим и имеет $d_{\text{free}} \geqslant \min\{d_g + d_h, 3d_g, 3d_h\}$.*

Доказательство. Отметим снова, что так как n нечетное, то $g(x)$ не имеет кратных корней. Более того, согласно нашему выбору $G(D)$, в силу (15) получаем

$$g(D)^2 = \sum_{j=1}^{2m} D^{2(j-1)} G_{2j-1}(D^{4m}) = \sum_{j=1}^{2m} D^{2(j-1)} \hat{\hat{G}}_{2j-1}(D^m)^4,$$

где $\hat{\hat{G}}_j(D)$ обозначает многочлен, полученный из $G_j(D)$ при замене каждого коэффициента на его корень четвертой степени; корень четвертой степени существует и однозначен для каждого элемента $GF(2^r)$. Следовательно, общий делитель для $G_1(D), G_3(D), \ldots, G_{v-1}(D)$ означал бы, что $g(x)$ имеет некоторые корни кратности, большей 1. Отсюда получаем, что НОД$\{G_1(D), G_3(D), \ldots, G_{v-1}(D)\} = 1$ и, следовательно, НОД$\{G_1(D), G_2(D), \ldots, G_v(D)\} = 1$, так что сверточный код является некатастрофическим.

Для любого многочлена $I(D) \neq 0$ можно написать

$$T(D) = I(D^{4m}) G(D) = \hat{\hat{I}}(D^m)^4 [g(D)^2 + Dh(D)^2],$$

где коэффициенты в $\hat{\hat{I}}(D)$ — корни четвертой степени от коэффициентов в $I(D)$. Теперь можно записать это в виде

$$T(D) = P(D)\, g(D)^{4l}\, h(D)^{4j} [g(D)^2 + Dh(D)^2],$$

где $P(D)$ — ненулевой многочлен, который не делится ни на $g(D)$, ни на $h(D)$; отсюда следует, что

$$W[T(D)] = W[P(D)\,g(D)^{4l+2}\,h(D)^{4j}] + W[P(D)\,g(D)^{4l}\,h(D)^{4j+2}]. \quad (23)$$

Предположим вначале, что $i > j \geqslant 0$. Применяя теорему 1.3 ко второму члену правой части (23), получим

$$W[P(D)\,g(D)^{4(i-j)-2}\,(D^n+1)^{4j+2}] \geqslant$$
$$\geqslant W[(D+1)^{4j+2}] \cdot W[P(D)\,g(D)^{4(i-j)-2} \bmod (D^n+1)].$$

Первый сомножитель правой части по меньшей мере равен 2, а второй сомножитель по меньшей мере равен d_g, так как его аргументом является ненулевое кодовое слово циклического кода, порождаемого $g(x)$. Поэтому второе выражение в правой части (23) по меньшей мере равно $2d_g$. Подобные же рассуждения показывают, что первое выражение по меньшей мере равно d_g, так что

$$W[T(D)] \geqslant 3d_g. \quad (24)$$

Используя те же рассуждения, что и выше, при $j > i \geqslant 0$ имеем

$$W[T(D)] \geqslant 3d_h. \quad (25)$$

Наконец, предположим, что $i = j \geqslant 0$. Применяя теорему 1.3 к первому члену правой части (23), получаем

$$W[P(D)\,g(D)^2\,(D^n+1)^{4l}] \geqslant$$
$$\geqslant W[(D+1)^{4l}] \cdot W[P(D)\,g(D)^2 \bmod (D^n+1)] \geqslant d_g.$$

Подобные рассуждения показывают, что второй член правой части (23) ограничен снизу величиной d_h, так что будем иметь

$$W[T(D)] \geqslant d_g + d_h. \quad (26)$$

Утверждение теоремы теперь следует из (24), (25) и (26)'.

Отметим вновь, что нижняя граница для d_{free}, которая приведена в теореме 4, не зависит от m и, следовательно, от скорости R сверточного кода, полученного из циклического кода. Поэтому граница будет наиболее точной при $m = 1$, т. е. для $R = 1/4$. Наилучшие сверточные коды получаются при выборе циклического кода так, что $d_g \approx d_h$. В таблице 2 приведено несколько двоичных (т. е. с $r = 1$) сверточных кодов, полученных с помощью теоремы 4 для $R = 1/4$, и указан тот циклический код, который использован при построении. Сравнение таблиц 1 и 2 показывает, что коды с $R = 1/4$, полученные с помощью теоремы 4, часто оказываются лучше кодов с $R = 1/4$, полученных с помощью теоремы 3 (или по крайней мере граница снизу для d_{free} будет больше).

Таблица 2

Некоторые двоичные сверточные коды, полученные с помощью конструкции, приведенной в теореме 4

Использованный циклический код	R	n_A	d_{free}
КВ (7,3)-код, $d_g = 4$, $d_h = 3$	$1/4$	12	$\geqslant 7$
КВ (17,8)-код, $d_g = 6$, $d_h = 5$	$1/4$	20	$\geqslant 11$
КВ (23,11)-код, $d_g = 8$, $d_h = 7$	$1/4$	28	$\geqslant 15$
КВ (41,20)-код, $d_g = 10$, $d_h = 9$	$1/4$	44	$\geqslant 19$
КВ (47,23)-код, $d_g = 12$, $d_h = 11$	$1/4$	52	$\geqslant 23$
БЧХ (63,30)-код, $d_g = 13$, $d_h \geqslant 8$	$1/4$	68	$\geqslant 21$
КВ (79,39)-код, $d_g = 16$, $d_h = 15$	$1/4$	84	$\geqslant 31$
КВ (89,44)-код, $d_g = 18$, $d_h = 17$	$1/4$	92	$\geqslant 35$
КВ (103,51)-код, $d_g = 20$, $d_h = 19$	$1/4$	108	$\geqslant 39$

Для того чтобы понять, насколько хороши сверточные коды, полученные с помощью указанных выше конструкций, код из таблицы 1 с $n_A = 40$ и $R = 1/2$ был сравнен с «дополнительным» кодом Бола — Джелинека [11] с $n_A = 40$ и $R = 1/2$, а также с быстро просматриваемым кодом Месси — Кастелло с $n_A = 40$ и $R = 1/2$ [10] при использовании алгоритма Фано [12] последовательного декодирования в моделированных двоичных симметричных каналах и каналах с аддитивным белым гауссовским шумом. В результате обширного моделирования в случае, когда вычислительная скорость R_{comp} в канале была близка к кодовой скорости $1/2$, было установлено, что код таблицы 1 был несколько хуже с точки зрения вероятности необнаруживаемой ошибки по отношению к коду Бола — Джелинека, но был значительно лучше кода Месси — Кастелло. По вероятности стирания код таблицы 1 был хуже кода Месси — Кастелло, но лучше кода Бола — Джелинека. При этом представляется разумным считать, что коды, полученные с помощью теорем 3 и 4, будут конкурентноспособными по отношению к лучшим известным кодам, построенным на основании других конструкций.

Очевидно, что общий характер теоремы 1.3 допускает построение многих новых классов сверточных кодов путем смешивания $g(x)$ и $h(x)$ из нескольких кодов и т. д. Мы предоставляем сделать такие обобщения читателю. Следует также заметить, что в двоичном ($r = 1$) частном случае при $R = 1/2$ теорема 1.3 была независимо доказана Рудольфом и Микзо [13] с помощью совершенно других рассуждений.

Д. Построение двоичных сверточных кодов с помощью кодов Рида — Соломона

Выбирая $g(x)$ в теореме 3 (или в теореме 4) так, чтобы они были порождающими многочленами 2^r-ичного кода Рида — Соломона (РС) [2, стр. 318], мы можем построить некоторые удивительно хорошие двоичные сверточные коды с очень большими ограничительными длинами. Для того чтобы получить двоичные коды, каждый символ РС-кода представляется в виде двоичной r-последовательности так, что 2^r-ичный сверточный код ($R = 1/2$) с одним информационным символом и двумя кодовыми символами в подблоке преобразуется в двоичный код ($R = 1/2$) с r информационными битами и $2r$ кодовыми битами в подблоке. Ограничительная длина n_{Ab} двоичного кода равна умноженной на r ограничительной длине n_A 2^r-ичного кода.

Для РС $(n = 2^r - 1, k)$-кода $d_g = n - k + 1$ $d_h = k + 1$. Таким образом, наилучшая граница для d_{free} в теореме 3 получается при выборе $k = \lfloor n/3 \rfloor$, где $\lfloor \cdot \rfloor$ обозначает целую часть внутреннего выражения. При таком выборе получаем

$$d_{\text{free}} \geqslant \lfloor (2n + 4)/3 \rfloor$$

и

$$n_{Ab} = \begin{cases} r(n - k + 1), & n - k \text{ нечетное}, \\ r(n - k + 2), & n - k \text{ четное}, \end{cases}$$

так что

$$d_{\text{free}}/n_{Ab} \gtrsim 1/r. \tag{27}$$

Неравенство (27) при $r = 10$ показывает, что свободное расстояние этих двоичных кодов все еще остается по меньшей мере в пределах 10% от ограничительной длины при $n_{Ab} \approx 7000$.

Еще лучшие коды могут быть получены с помощью добавления одного проверочного символа к r-последовательности, которая используется для представления символов $GF(2^r)$. В этом случае двоичный код все еще имеет лишь r информационных битов на подблок, но длина подблока увеличивается до $2(r + 1)$, и поэтому скорость двоичного кода снижается до

$$R = \frac{1}{2}\frac{r}{r+1}; \tag{28}$$

эта величина стремится к $1/2$ с ростом r. Так как имеются по меньшей мере два ненулевых бита в каждом ненулевом символе $GF(2^r)$ в этом новом представлении двоичного кода, имеем

$$d_{\text{free}} \geqslant 2 \lfloor (2n + 4)/3 \rfloor,$$

а также

$$n_{Ab} = \begin{cases} (r + 1)(n - k + 1), & n - k \text{ нечетное}, \\ (r + 1)(n - k + 2), & n - k \text{ четное}, \end{cases}$$

так что

$$d_{\text{free}}/n_{Ab} \gtrsim 2/(r+1). \tag{29}$$

Неравенство (29) при $r = 19$ и, следовательно, при R, довольно близком к $^1/_2$, показывает, что свободное расстояние этих двоичных кодов все еще находится в пределах по меньшей мере 10% ограничительной длины при $n_{Ab} \approx 6\,700\,000$.

Наиболее сильная из известных нижних границ для d_{free} для двоичных сверточных кодов была получена Нейманом [14], но существует улучшенная нижняя граница, принадлежащая Кастелло [15] для сверточных кодов, меняющихся со временем. При $R = {}^1/_2$ эти нижние границы для d_{free}/n_A дают 0,22 и 0,40 соответственно для больших n_A. Нижняя граница для d_{free}/n_{Ab} для второго класса кодов, рассмотренных в этом параграфе, остается выше этих величин при $n_{Ab} \leqslant 900$ и $n_{Ab} \leqslant 100$ соответственно.

III. НЕДВОИЧНЫЙ СЛУЧАЙ

В дальнейшем будем обозначать через p простое число, большее чем 2, через c ненулевой элемент $GF(p^r)$, через n длину p^r-ичного блокового кода, через k число информационных символов и через d минимальное расстояние рассматриваемого кода.

A. Новый класс p^r-ичных констациклических кодов с кратными корнями, имеющих алгебраический алгоритм декодирования

Следуя терминологии Берлекэмпа [2, стр. 310], будем говорить, что многочлен $g(x)$ над $GF(p^r)$ степени $n-k$, который делит $x^n - c^p$, порождает *констациклический* (n, k)-код, все кодовые слова которого делятся на $g(x)$ и имеют степень, меньшую чем n. Код является циклическим тогда и только тогда, когда $c = 1$, и *негациклическим* [2, стр. 220] тогда и только тогда, когда $c = -1$. Следующая теорема дает новый класс констациклических кодов с кратными корнями и в ее доказательстве будет развит алгебраический алгоритм декодирования этих кодов. Циклические коды в этом классе ранее были получены Ассмусом и Мэттсоном [16] и Берманом [17].

Теорема 5. *Многочлен $g(x) = (x-c)^{p-k}$ при $1 \leqslant k < p$ порождает p^r-ичный констациклический $(n = p, k)$-код с $d = n-k+1$ (т. е. код с достижимым максимальным расстоянием* [2, стр. 317]).

Доказательство. Заметим вначале, что $(x-c)^p = x^p - c^p$, так что $g(x)$ является делителем $x^p - c^p$ и поэтому порождает констациклический код длины $n = p$. Более того, по лемме 1 $W[g(x)] = p-k+1 = n-k+1$, так что $d \leqslant n-k+1$.

Пусть $f(x)$ — кодовое слово этого констациклического кода, и пусть $e(x) = e_0 + e_1 x + \ldots + e_{p-1}x^{p-1}$ обозначает ошибку в канале.

То же доказательство, что и при выводе формулы (11), теперь показывает, что после того как $f(x)+e(x)$ будет записано в схеме, представленной на рис. 2, с членами более высокого порядка вначале, результирующий синдром будет задаваться равенством

$$s(x) = e(x+c) \bmod (x^{p-k}), \tag{30а}$$

или, что эквивалентно,

$$s(x) = \sum_{l=0}^{p-1} e_l (x+c)^l \bmod (x^{p-k}). \tag{30б}$$

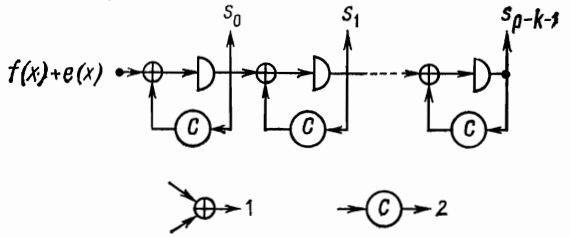


Рис. 2. Схема, формирующая синдром для p^r-ичных констациклических кодов с кратными корнями из теоремы 5.
1 — сумматор $GF(p^r)$, *2* — константный мультипликатор.

Из (30б) видно, что синдром $s(x) = s_0 + s_1 x + \ldots + s_{p-k-1}x^{p-k-1}$ может быть выражен в виде

$$s_i = \sum_{l=0}^{p-1} \binom{l}{i} c^{l-i} e_l, \quad 0 \leqslant i < p-k, \tag{31а}$$

или, что эквивалентно,

$$s_i = (i!)^{-1} \partial^i e(c), \quad 0 \leqslant i < p-k, \tag{31б}$$

где через $\partial^i e(c)$ обозначена i-я формальная производная $e(x)$ в точке $x = c$.

Определим теперь модифицированные символы синдрома $S_0, S_1, \ldots, S_{p-k-1}$ как следующие линейные комбинации:

$$S_i = \sum_{j=1}^{i} \left\{ {i \atop j} \right\} (j!)\, c^j s_i, \quad 1 \leqslant i < p-k \tag{32}$$

и

$$S_0 = s_0, \tag{33}$$

где $\left\{ {i \atop j} \right\}$ обозначает число Стирлинга второго рода [19]. Из (31а) и (32) получаем

$$S_i = \sum_{j=1}^{t} \sum_{m=0}^{p-1} c^m (j!) \binom{m}{j} e_j, \quad 1 \leqslant i < p-k,$$

так что можно записать

$$S_i = \sum_{j=0}^{p-1} e_j c^j j^i, \quad 0 \leqslant i < p-k, \tag{34}$$

где следует понимать, что $0^0 = 1$.

Предположим, что ошибка $e(x)$ имеет вес t, так что ее можно записать в виде

$$e(x) = \sum_{j=1}^{t} Y_j (x/c)^{i_j},$$

где $Y_j \neq 0$ является «модифицированным» *значением* j-й ошибки (оно связано с истинным значением ошибки e_{i_j} соотношением $e_{i_j} = Y_j c^{-i_j}$). Если определить $X_j = i_j$ как *локатор* j-й ошибки, то равенство (34) можно теперь переписать в виде

$$S_i = \sum_{j=1}^{t} Y_j X_j^i, \quad 0 \leqslant i < p-k, \tag{35}$$

Но (35) как раз представляет собой обычное синдромное соотношение для БЧХ-кода с конструктивным расстоянием [2, стр. 282] $p-k+1$, которое означает, что $d \geqslant p-k+1$.

Отсюда получаем, что $d = p-k+1$, что доказывает теорему.

Более того, из (35) следует, что значения ошибок и их локаторы для констациклических кодов из теоремы 5 могут быть определены с помощью любой процедуры декодирования для БЧХ-кодов, такой, как, например, итеративный алгоритм Берлекэмпа [2, стр. 227—229]. В частности, когда p — мерсеновское простое число, т. е. когда $p = 2^m - 1$ для некоторого целого m, так что операции $GF(p)$ просто являются арифметикой «дополнения единицы», процедура декодирования для p-ичных кодов легко реализуется, в особенности когда $c = 1$ (циклические коды) или когда $c = -1$ (негациклические коды), и формирователь синдрома на рис. 2 очень прост. Эти коды могут найти практическое применение в каскадных схемах кодирования [19].

Далее отметим, что параметры n, k и d констациклических кодов из теоремы 5 совпадают с параметрами удлиненных [2, стр. 338] p-ичных кодов Рида — Соломона, а при $c = 1$ как раз являются перестановками этих кодов. Интересное отличие состоит в том, что коды с кратными корнями позволяют использовать числа $0, 1, \ldots,$

$\ldots, p-1$ в качестве локаторов ошибок, в то время как в РС-кодах локаторы ошибок берутся в виде 0 и степеней $\alpha^0, \alpha, \ldots, \alpha^{p-2}$ примитивного элемента α поля $GF(p)$. Использование аддитивной группы поля $GF(p)$ вместо мультипликативной группы в качестве локаторов ошибок приводит как к естественному включению нулевой позиции, так и к уменьшению числа умножений, которые требуются для декодирования с помощью итеративного алгоритма.

Б. Свойство сохранения веса $(x-c)^i$ над $GF(p^r)$

Отметим еще раз, что многочлены $(x-c)^i$, $i=0,1,2,\ldots$, образуют базис векторного пространства всех многочленов над $GF(p^r)$. Теперь мы хотим показать, что свойство сохранения веса, сформулированное в теореме 1.1 для $p=2$, имеет место в общем случае. Мы считаем более удобным доказать вначале p-ичный аналог теоремы 1.2 и после этого вывести p-ичный аналог теоремы 1.1.

Теорема 6.2. *Для любого многочлена $Q(x)$ над $GF(p^r)$, любого ненулевого c из $GF(p^r)$ и любого неотрицательного целого N имеет место соотношение*

$$W[Q(x)(x-c)^N] \geqslant W[(x-c)^N] \cdot W[Q[c]]. \tag{36}$$

Доказательство. В дальнейшем мы часто будем использовать тот факт, что для любого i и для любого многочлена $P(x)$ справедливо $W[P(x)] \geqslant W[P(x) \bmod (x^i - c)]$.

Покажем вначале, что (36) справедливо для $N < p$. Если $Q(c)=0$, то (36) тривиально справедливо. Если $Q(c) \neq 0$, то $Q(x)$ не делится на $(x-c)$, так что $Q(x)(x-c)^N \bmod (x^p - c^p)$ является ненулевым кодовым словом констациклического кода, порождаемого $g(x)=(x-c)^N$, и поэтому по теореме 5 имеет хэмминговский вес, по меньшей мере равный $N+1$. Таким образом,

$$W[Q(x)(x-c)^N] \geqslant W[Q(x)(x-c)^N \bmod (x^p - c^p)] \geqslant$$
$$\geqslant N+1 = W[(x-c)^N],$$

где последнее неравенство следует из леммы 1. Поэтому (36) справедливо для $N < p$.

Предположим теперь, что (36) справедливо для $N < Kp^i$, $1 \leqslant K < p$ и проведем индукцию по K, которая также содержит в себе индукцию по i, так как $(K+1)p^i = p^{i+1}$, когда $K = p-1$. Уже было показано, что (36) справедливо для $i=0$, так что был установлен базис индукции. Остается показать, что (36) справедливо для $N < (K+1)p^i$ или, что эквивалентно, для $N = Kp^i + L$ при всех целых L, таких, что $0 \leqslant L < p^i$.

Начнем с того замечания, что

$$W[Q(x)(x-c)^N] = W[Q(x)(x-c)^L(x-c)^{Kp^i}] = \\ = W[Q(x)(x-c)^L(x^{p^i}-c^{p^i})^K]. \quad (37)$$

Теперь, записывая

$$P(x) = Q(x)(x-c)^L = \sum_{j=0}^{p^i-1} x^j P_j(x^{p^i}), \quad (38)$$

из (37) получаем

$$W[Q(x)(x-c)^N] = \sum_{j=0}^{p^i-1} W[P_j(x)(x-c^{p^i})^K], \quad (39)$$

где просто была произведена замена x^{p^i} на x из формулы (38). Так как $K < p$, то (36) справедливо для каждого слагаемого в правой части (39), и по лемме 1 получаем

$$W[Q(x)(x-c)^N] \geqslant (K+1) \sum_{j=0}^{p^i-1} W[P_j(c^{p^i})],$$

что с помощью (38) можно записать виде

$$W[Q(x)(x-c)^N] \geqslant (K+1)\, W[Q(x)(x-c)^L \bmod (x^{p^i}-c^{p^i})]. \quad (40)$$

Теперь также имеем

$$Q(x)(x-c)^L \bmod (x^{p^i}-c^{p^i}) = Q(x)(x-c)^L \bmod (x-c)^{p^i} = \\ = [Q(x) \bmod (x-c)^{p^i-L}](x-c)^L. \quad (41)$$

Но $L < p^i$, так что (36) можно применить к правой части (41), чтобы получить

$$W[Q(x)(x-c)^L \bmod (x^{p^i}-c^{p^i})] \geqslant W[(x-c)^L] \cdot W[Q(c)];$$

здесь было использовано то, что $Q(x) \bmod (x-c)^L$, вычисленное при $x = c$, в точности равно $Q(c)$. Из (40), (41) и (42) получаем

$$W[Q(x)(x-c)^N] \geqslant (K+1)\, W[(x-c)^L] \cdot W[Q(c)],$$

что с помощью леммы 1 и того, что $L < Kp^i$ и $N = Kp^i + L$, можно, наконец, записать в виде

$$W[Q(x)(x-c)^N] \geqslant W[(x-c)^N] \cdot W[Q(c)],$$

что совпадает с (36); теорема доказана.

Теперь мы приведем как тривиальное следствие теоремы 6.2 следующую теорему.

Теорема 6.1. *Пусть I является каким-либо непустым конечным множеством неотрицательных целых чисел, среди которых наименьшим является $i_{\min}$, и пусть*

$$P(x) = \sum_{i \in I} b_i (x - c)^i,$$

где c и каждое b_i являются ненулевыми элементами $GF(p^r)$. Тогда

$$W[P(x)] \geqslant W[(x - c)^{i_{\min}}]. \tag{43}$$

Эта теорема следует из теоремы 6.2, если заметить, что

$$P(x) = Q(x)(x - c)^{i_{\min}}, \quad \text{где} \quad Q(c) = b_{i_{\min}} \neq 0.$$

Теорема 6.1 представляет собой требуемый p-ичный аналог теоремы 1.1. Несмотря на то что в дальнейшем мы не используем этого, для полноты мы сформулируем сейчас p-ичный аналог теоремы 1.3, который следует из теоремы 6.2 в точности так же, как теорема 1.3 следует из теоремы 1.1.

Теорема 6.3. *Для любого многочлена $P(x)$ над $GF(p^r)$, любого ненулевого элемента c поля $GF(p^r)$ и любых неотрицательных целых n и N справедливо*

$$W[P(x)(x^n - c)^N] \geqslant W[(x - c)^N] \cdot W[P(x) \bmod (x^n - c)].$$

В. Новый класс p-ичных кодов Рида — Маллера

Поступая аналогично тому, как в разд. II.Б, обозначим через m произвольное целое положительное число и рассмотрим матрицу G с $n = p^m$ столбцами, строки которой представляют собой последовательности коэффициентов многочленов $(x - c)^i$ при всех $i < n$, таких, что $W[(x - c)^i] \geqslant d$, где d — некоторое целое, которое выбирается так, что равенство имеет место для по меньшей мере одного такого i. При заданных n и d значения k, соответствующие целым i, могут быть найдены с помощью леммы 1. Для простоты обычно можно брать $c = 1$ или $c = -1$, но это не является необходимым. Из теоремы 6.1 сразу же следует, что G — порождающая матрица p-ичного $(n = p^m, k)$-кода с минимальным расстоянием d. Будем называть эти коды p-ичными кодами Рида — Маллера в силу их сходства с двоичными кодами Рида — Маллера, описанными так, как в разд. II.Б. Краткий перечень этих кодов приведен в табл. 3.

Коды в табл. 3 в большинстве случаев имеют те же самые параметры n, k и d, что и обобщенные p-ичные коды Рида — Маллера, описанные Казами и др. [20] (которые далее мы будем называть КЛП-кодами), всегда, когда более редкие КЛП-коды существуют;

Таблица 3

Краткий перечень p-ичных кодов Рида — Маллера из нового класса, приведенного в разд. III. В

p	n	k	d		p	n	k	d
3	9	8	2		5	5	4	2
	9	6	3			5	3	3
	9	4	4			5	2	4
	9	3	6			5	1	5
	9	1	9			25	24	2
	27	26	2			25	22	3
	27	23	3			25	20	4
	27	20	4			25	17	5
	27	17	6			25	15	6
	27	11	8			25	13	8
	27	10	9			25	11	9
	27	7	12			25	10	10
	27	4	18			25	8	12
	27	1	27			25	6	15
						25	4	16
						25	3	20
						25	1	25

т. е. коды табл. 3 в общем имеют при том же самом k увеличенные на единицу параметры n и d по сравнению с соответствующими величинами для КЛП-кодов. Однако 5-ичный (25,15)-код из табл. 3 имеет $d = 6$, в то время как d равно лишь 4 у 5-ичного (24, 15) КЛП-кода, так что p-ичные коды Рида — Маллера, представленные здесь, не являются просто перестановками (более редких) КЛП-кодов.

Отметим также, что можно получить констациклические подкоды с кратными корнями p-ичных кодов Рида — Маллера, определенных здесь, которые аналогичны циклическим кодам из разд. II. В. Порождающий многочлен констациклического кода выбирается в виде $g(x) = (x - c)^{n-k}$, где $k = p^{m-u+1} - 1$. Эти констациклические подкоды имеют то же самое минимальное расстояние $d = 2p^{u-1}$, что и порождающие их p-ичные коды Рида — Маллера. Констациклический код является циклическим тогда и только тогда, когда $c = 1$, и является негациклическим тогда и только тогда, когда $c = -1$.

IV. ЗАКЛЮЧИТЕЛЬНЫЕ ЗАМЕЧАНИЯ

Выше было показано, что свойства сохранения веса многочленов $(x - c)^i$ допускают их использование при построении как блоковых, так и сверточных кодов. Эта общность до некоторой степени

удивительна и представляется возможным утверждать, что в этой статье были обнаружены не все следствия свойства сохранения веса.

Нам хочется отметить, что получение вторым автором этой статьи двоичных кодов из теоремы 3 с $R = {}^1/_2$ было отправным пунктом для проведенного здесь исследования. Двоичные сверточные коды из теоремы 3 для $m = 2^i$ и из теоремы 4 для $m = 1$ ранее были отмечены в устном сообщении первых двух авторов этой статьи [21]. В первоначальном варианте этой статьи результаты были получены лишь для простых полей $GF(p)$, а не для $GF(p^r)$, как представлено здесь. Наиболее значительным следствием этого обобщения было последующее обнаружение Юстесеном сверточных кодов большой ограничительной длины, представленных в разд. II. Д.

Авторы признательны доктору Г. Д. Форни за проявленный им интерес к нашему исследованию и за предложенное им простое доказательство леммы 1. (Другое доказательство леммы 1 можно найти у Бермана [17].)

СПИСОК ЛИТЕРАТУРЫ

1. Preparata F. P., State-logic relations for autonomous sequential networks, *IEEE Trans. Electron. Comput.*, **EC-13**, 542—548, Oct., 1964.
2. Berlekamp E. R., Algebraic Coding Theory, New York, Mc Graw-Hill, 1968. (Русский перевод: Берлекэмп Э., Алгебраическая теория кодирования, «Мир», М., 1974.)
3. Reed I. S., A class of multiple-error-correcting codes and the decoding scheme, *IRE Trans. Inform. Theory*, **PGIT-4**, 38—49, Sept. 1954.
4. Meggitt J. E., Error correcting codes and their implementation for data transmission systems, *IRE Trans. Inform. Theory*, **IT-7**, 234—244, Oct. 1961.
5. Massey J. L., Threshold Decoding, Cambridge, Mass., M. I. T. Press, 1963.
6. Massey J. L., Majority decoding of convolutional codes, Res. Lab. Electron. Mass. Inst. Technol., Cambridge, Quart. Prog. Rep., 64, 183—188, Jan. 15, 1962.
7. Massey J. L., Sain M. K., Inverses of linear sequential circuits, *IEEE Trans. Comput.*, **C-17**, 330—337, Apr. 1968.
8. Forney G. D., Jr., Convolutional codes I: Algebraic structure, *IEEE Trans. Inform. Theory*, **IT-16**, 720—738, Nov. 1970.
9. Viterbi A. J., Convolutional codes and their performance in communication systems, *IEEE Trans. Commun. Technol.*, **COM-19**, 751—772, Oct. 1971.
10. Massey J. L., Costello D. J., Jr., Nonsystematic convolutional codes for sequential decoding in space applications, *IEEE Trans. Commun. Technol.*, **COM-19**, 806—813, Oct. 1971.
11. Bahl L. R., Jelinek F., Rate ${}^1/_2$ convolutional codes with complementary generators, *IEEE Trans. Inform. Theory*, **IT-17**, 718—727, Nov. 1971.
12. Fano R. M., A heuristic discussion of probabilistic decoding, *IEEE Trans. Inform. Theory*, **IT-9**, 64—74, Apr. 1963.
13. Rudolph L. D., Miczo A., Some results on the distance properties of convolutional codes, Syracuse Univ., Syracuse, N. Y., Final Rep. NSF Grant GK-4737, Oct. 1970.

14. Neumann B., Distance properties of convolutional codes, S. M. thesis, Dep. Elec. Eng., Mass. Inst. Technol., Cambridge, Aug. 1968.
15. Costello D. J., Jr., Construction of convolutional codes for sequential decoding, Dep. Elec. Eng., Univ. Notre Dame, Notre Dame, Ind., Tech. Rep. EE-692, Aug. 1969.
16. Assmus E. F., Jr., Mattson H. F., Jr., New 5-designs, J. Combinatorial Theory, **6**, 122—151, 1969.
17. Берман С. Д., К теории групповых кодов, *Кибернетика*, **3**, № 1, 31—39, 1967.
18. Riordan J., An Introduction to Combinatorial Analysis, New York, Wiley, 33, 1958.
19. Forney G. D., Jr., Concatenated Codes, Cambridge, Mass., M. I. T. Press, 1966.
20. Kasami T., Lin S., Peterson W. W., New generalizations of the Reed-Muller codes, *IEEE Trans. Inform. Theory*, **IT-14**, 189—198, Jan. 1968.
21. Costello D. J., Massey J. L., Constructing good convolutional codes from cyclic block codes, presented at IEEE Int. Symp. Information Theory, Asilomar, Calif., Jan. 1972.

Новые конструкции сверточных кодов и класс асимптотически хороших кодов, меняющихся со временем [1)]

Йёрн Юстесен

1. ВВЕДЕНИЕ

Эта статья содержит ряд уточнений и обобщений результатов, относящихся к связи между циклическими кодами и сверточными кодами, полученных ранее Месси и др. [1].

В разд. 2 выведена нижняя граница для свободного расстояния некоторых сверточных кодов, полученных из циклических кодов. Эта граница дается в терминах минимального расстояния циклического кода и корней порождающего многочлена.

В разд. 3 посторены сверточные коды, аналогичные блоковым кодам Рида — Соломона, а в разд. 4 показано, что, если ограничить эти коды, рассматривая их над подполями, то получаются хорошие сверточные коды, тесно связанные с блоковыми БЧХ-кодами.

Наконец, с помощью метода, очень похожего на тот, который использовался ранее Юстесеном [2] для получения асимптотически хороших блоковых кодов из кодов Рида — Соломона, построены асимптотически хорошие периодически меняющиеся со временем сверточные коды на основе сверточных кодов из разд. 3.

2. НИЖНЯЯ ГРАНИЦА ДЛЯ СВОБОДНОГО РАССТОЯНИЯ НЕКОТОРЫХ СВЕРТОЧНЫХ КОДОВ

Мы широко будем использовать следующую теорему, которая была доказана Месси и др. [1].

Теорема 1. *Для любого многочлена $P(x)$ над $GF(p^r)$, любого ненулевого элемента c из $GF(p^r)$ и любых неотрицательных целых чисел n и N справедливо неравенство*

$$W[P(x)(x^n - c)^N] \geqslant W[(x - c)^N]\, W[P(x) \bmod (x^n - c)].$$

Здесь $W[P(x)]$ обозначает хэмминговский вес многочлена $P(x)$, а через $P(x) \bmod Q(x)$ обозначается остаток при делении $P(x)$ на многочлен $Q(x)$.

Мы опишем некоторый класс фиксированных несистематических сверточных кодов со скоростью κ/ν, используя следующие обозначения, обобщающие обозначения Месси [3].

[1)] Justesen J., New convolutional code constructions and a class of asymptotically good time-varying codes, *IEEE Trans. Inform. Theory*, **IT-19**, № 2, 220—225, (March, 1973).

Информационная последовательность с нулями, помещенными на проверочных позициях, имеет D-преобразование

$$I(D)=i_0+i_1D+i_2D^2+\ldots+i_{\varkappa-1}D^{\varkappa-1}+i_\nu D^\nu+\ldots=\sum_{j=1}^{\varkappa}D^{j-1}I_j(D^\nu).$$

В этой работе сверточный код определяется с помощью одного-единственного порождающего многочлена $G(x)$, и D-преобразование закодированной последовательности может быть записано в виде $T(D)=I(D)G(D)$.

Определим ограничительную длину кода как $n_A=$ степень $(G)+1$. Таким образом, ограничительная длина равна числу кодовых символов, на которые влияет один информационный символ.

В общем случае фиксированный сверточный код со скоростью $\varkappa/\nu$ определяется порождающей матрицей вида [4]

$$\mathscr{G}=\begin{Bmatrix} G_{11}(D) & G_{12}(D) & \ldots & G_{1\nu}(D)\\ G_{21}(D) & G_{22}(D) & \ldots & G_{2\nu}(D)\\ \ldots & \ldots & \ldots & \ldots\\ G_{\varkappa 1}(D) & G_{\varkappa 2}(D) & \ldots & G_{\varkappa\nu}(D)\end{Bmatrix}.$$

Рассматриваемый здесь подкласс имеет порождающие матрицы

$$\mathscr{G}'=\begin{Bmatrix} G_1(D) & G_2(D) & \ldots & G_\nu(D)\\ DG_\nu(D) & G_1(D) & \ldots & G_{\nu-1}(D)\\ \ldots & \ldots & \ldots & \ldots\\ DG_{\nu-\varkappa+2}(D) & DG_{\nu-\varkappa+3}(D) & \ldots & G_{\nu-\varkappa+1}(D)\end{Bmatrix}.$$

Этот класс сверточных кодов достаточно широк для того, чтобы допустить доказательство нижней границы Гильберта для минимального расстояния с помощью следующих стандартных рассуждений [5].

Рассмотрим усеченную кодовую последовательность T_m длины $m\nu$, начинающуюся ненулевым символом, и последовательность $m\varkappa$ информационных символов I_m, также имеющую ненулевой символ, на первой позиции. (Напомним, что минимальное расстояние по определению равно минимальному весу среди весов всевозможных таких T_m.) Тогда легко заметить, что существует ровно один код в классе с ограничительной длиной $n_A\leqslant m\nu$, который производит кодовую последовательность, первые $m\nu$ символов которой дают T_m в качестве реакции на информационную последовательность, первые $m\varkappa$ символов которой равны I_m. Следовательно, по меньшей мере один код над $GF(q)$ с ограничительной длиной, меньшей чем $m\nu$, имеет минимальное расстояние d_G, удовлетворяющее

$$\sum_{i=0}^{d_G-1}\binom{m\nu}{i}(q-1)^i<q^{m\nu(1-k/\nu)}, \qquad (1)$$

Сверточный код называется катастрофическим [6], если функция $I(D)$, не являющаяся многочленом, может привести к многочлену $T(D)$ [4].

Свободное расстояние d_{free} сверточного кода равно минимуму $W[T(D)]$, взятому по всем $I(D) \neq 0$. Для некатастрофических кодов d_{free} может быть взята как минимум $W[T(D)]$ по всем многочленам $I(D) \neq 0$. Очевидно, что граница Гильберта (1) является нижней границей как для свободного расстояния, так и для минимального расстояния.

Мы изучим сверточные коды, порождаемые порождающими многочленами $g(x)$ циклических кодов. Будем обозначать через $h(x) = (x^n - 1)/g(x)$ порождающий многочлен циклического дуального кода; через d_g минимальное расстояние исходного циклического кода; через d_h минимальное расстояние дуального кода и через n длину обоих этих циклических кодов.

В [1] получен следующий результат.

Теорема 2. *Если $g(x)$ порождает циклический код над $GF(2^r)$ нечетной длины n, то для любого положительного целого m 2^r-ичный сверточный код со скоростью $R = 1/\nu$ и с $\nu = 2m$, определяемый с помощью $G(D) = g(D)$, является некатастрофическим и имеет $d_{\text{free}} \geqslant \min\{d_g, 2d_h\}$.*

В разд. 3 будет дано обобщение этой теоремы на поля нечетной характеристики.

В силу того, что дуальные коды многих хороших циклических кодов, например БЧХ-кодов, имеют малые минимальные расстояния, желательно получить границу для свободного расстояния, которая зависит только от d_g. Следующее определение будет полезно при выводе такой границы.

Определение. Пусть p — простое число, n — целое число, взаимнопростое с p, и ν — целое число, которое является делителем n. Отношение

$$\alpha \equiv \beta \text{ тогда и только тогда, когда } \alpha^{\nu} = \beta^{\nu}$$

представляет собой отношение эквивалентности среди корней n-й степени из единицы в $GF(p^{rs})$. Будем говорить, что ν эквивалентных корней n-й степени из единицы образуют ν-класс или что они ν-эквивалентны.

Если β является первообразным корнем n-й степени из единицы, то $\gamma = \beta^{n/\nu}$ и α является корнем n-й степени из единицы; ν-класс, который содержит α, представляет собой $\{\alpha, \alpha\gamma, \alpha\gamma^2, \ldots, \alpha\gamma^{\nu-1}\}$.

Определение сверточного кода, введенного в этом разделе, делает возможным получение естественным образом сверточных кодов со скоростью $\varkappa/\nu$, $\varkappa \geqslant 1$, из циклических кодов.

Теорема 3. *Если* $g(x)$ *порождает циклический код над* $GF(p^r)$ *с длиной* n, *простой по отношению к* p, ν *является произвольным положительным целым делителем* n *и* $g(x)$ *имеет самое большее* $\nu - \varkappa$ ν-*эквивалентных корней, то сверточный код со скоростью* $R = \varkappa/\nu$ *над* $GF(p^r)$, *порождаемый* $G(x) = g(x)$, *является некатастрофическим и имеет свободное расстояние* $d_{\text{free}} \geqslant d_g$.

Доказательство будет основано на следующей лемме.

Лемма 1. *Если* $\varkappa$ *или более* ν-*эквивалентных корней* n-*й степени из единицы являются корнями многочлена вида*

$$P(x) = \sum_{j=1}^{k} x^{j-1} P_j(x^\nu),$$

то все элементы ν-*класса являются корнями* $P(x)$.

Доказательство. Пусть α — корень $P(x)$, т. е.

$$P_1(\alpha^\nu) + \alpha P_2(\alpha^\nu) + \ldots + \alpha^{\varkappa-1} P_\varkappa(\alpha^\nu) = 0. \tag{2}$$

Составляющие многочлены $P_j(x^\nu)$ принимают те же самые значения на всех ν-эквивалентных значениях x, и, следовательно, можно интерпретировать (2) как уравнение

$$P_1 + zP_2 + z^2 P_3 + \ldots + z^{\varkappa-1} P_\varkappa = 0. \tag{3}$$

Если какое-либо P_j не равно нулю, то (3) имеет самое большее $\varkappa - 1$ решений относительно z. Многочлен, который имеет $\varkappa$ или более ν-эквивалентных корней, может делить $P(x)$ только тогда, когда $P_j(\alpha^\nu) = 0$ для всех j. Но в этом случае все элементы ν-класса являются корнями многочлена $P(x)$.

Доказательство теоремы 3. Чтобы показать, что сверточный код, порождаемый $g(x)$, является некатастрофическим, мы должны доказать, что информационная последовательность, не представляющая собой многочлен,

$$I(D) = \sum_{j=1}^{\varkappa} D^{j-1} \frac{P_j(D^\nu)}{Q_j(D^\nu)}$$

не может дать многочлен на выходе.

Заметим, что наименьший общий делитель для $Q_j(D^\nu)$ является многочленом от D^ν; обозначим его через $Q(D^\nu)$. Таким образом,

$$I(D) = [1/Q(D^\nu)] \sum_{j=1}^{\varkappa} D^{j-1} R_j(D^\nu) = R(D)/Q(D^\nu).$$

Можно предположить, что любые общие делители для $R(D)$ и $Q(D^\nu)$, которые являются многочленами от D^ν, были сокращены, но они могут иметь другие общие делители. Для того чтобы выход

$T(D) = G(D)I(D)$ представлял собой многочлен, все делители $Q(D^\nu)$ должны быть делителями либо $G(D)$, либо $R(D)$. Но если β — корень для $Q(D^\nu)$, то все элементы ν-класса, содержащего β, являются корнями. Однако мы предположили, что самое большее $\nu - \varkappa$ этих элементов являются корнями для $G(D)$, и, следовательно, по меньшей мере $\varkappa$ элементов должны быть корнями для $R(D)$, чтобы функция $I(D)$, не являющаяся многочленом, приводила к многочлену $T(D)$. Здесь можно использовать лемму 1, чтобы показать, что это возможно только тогда, когда все элементы ν-класса являются корнями для $R(D)$, что противоречит предположению, что $R(D)$ и $Q(D^\nu)$ не имеют многочленов от D^ν в качестве общих делителей.

Для того чтобы ограничить свободное расстояние сверточного кода, заметим вначале, что так как ν является делителем n, то $(D^n - 1)^i$ — многочлен от D^ν.

Следовательно, $I(D)/(D^n - 1)^i$ — многочлен того же самого вида, что и $I(D)$ при любом t, таком, что $(D^n - 1)^t$ является делителем $I(D)$.

Так как мы предположили, что $g(x)$ имеет самое большее $\nu - \varkappa$ ν-эквивалентных корней и так как все корни n-й степени из единицы являются корнями $x^n + 1 = g(x)h(x)$, то отсюда следует, что по меньшей мере $\varkappa$ элементов каждого ν-класса являются корнями $h(x)$. Поэтому лемма 1 показывает, что $h(D)$ является делителем $I(D)$ только тогда, когда $(D^n - 1)$ является делителем $I(D)$. Повторяя это рассуждение, получим, что $I(D) = (D^n - 1)^N p(D)$, где $h(D)$ не является делителем $p(D)$. Следовательно, с помощью теоремы 1 получаем

$$W[T(D)] \geqslant W[(D-1)^N]\, W[p(D)\, g(D) \bmod (D^n - 1)] \geqslant d_g.$$

Это завершает доказательство теоремы 3.

Условие, налагаемое на корни многочлена $g(x)$, может показаться довольно ограничительным, но, как установлено в разд. 3 и 4, оно выполняется в некоторых интересных случаях.

Двоичные коды, полученные с помощью теоремы 3, не включают коды со скоростью $1/2$. Наиболее интересными двоичными кодами будут коды со скоростями $1/3$ и $2/3$.

3. СВЕРТОЧНЫЕ КОДЫ, У КОТОРЫХ $d_{free} = n_A$

Так как $W[G(D)] \leqslant n_A$, то для любого сверточного кода имеем $d_{free} \leqslant n_A$. Коды, для которых $d_{free} = n_A$, можно рассматривать как аналоги блоковых кодов с достижимым максимальным расстоянием [7, стр. 317]. Коды Рида — Соломона [7, стр. 318] являются классом циклических кодов с достижимым максимальным расстоянием, которые существуют над любым конечным полем $GF(q)$.

Пусть $n = q - 1$ — длина кода Рида — Соломона над $GF(q)$, и пусть ν — делитель n. Порождающий многочлен кода Рида — Соломона со скоростью $k/n = \varkappa/\nu$ может быть взят в виде

$$g(x) = (x - 1)(x - \alpha)(x - \alpha^2) \ldots (x - \alpha^{n-k-1}),$$

где α — примитивный элемент из $GF(q)$. Из ν-класса, содержащего α^s, $0 \leqslant s < n/\nu$, элементы

$$\alpha^s,\ \alpha^{s+n/\nu},\ \alpha^{s+2n/\nu},\ \ldots,\ \alpha^{s+(n/\nu)(\nu-\varkappa-1)}$$

являются корнями многочлена $g(x)$. Таким образом, $g(x)$ имеет ровно $\nu - \varkappa$ корней из каждого ν-класса, и, следовательно, условие теоремы 3 выполняется.

Теорема 4. *Если $g(x)$ — порождающий многочлен кода Рида — Соломона со скоростью $k/n = \varkappa/\nu$ над $GF(q)$, то q-ичный сверточный код со скоростью $\varkappa/\nu$, порождаемый $G(x) = g(x)$, является некатастрофическим и имеет $d_{\text{free}} = n_A = n - k + 1$.*

Коды со скоростью $1/2$, полученные в [1] с помощью применения теоремы 2 к кодам Рида — Соломона над $GF(2^r)$ со скоростями $1/3$, также имеют $d_{\text{free}} = n_A$, но они обладают ограничительными длинами, приближенно на 50% большими, чем коды теоремы 4 для скоростей, близких к $1/2$. Доказательство теоремы 2 в [1] основывается существенным образом на соотношении между ν и характеристикой поля, и мы могли бы обобщить эту теорему очевидным образом на коды со скоростью $1/p$ для полей с характеристикой p. Мы предпочитаем доказать менее очевидное, но более сильное обобщение теоремы 2.

Теорема 5. *Если $g(x)$ порождает циклический код нечетной длины n над $GF(q)$, $q = p^r$ и p — нечетное простое число, то задаваемый с помощью $G(D) = g(D)$ сверточный код со скоростью $1/2$ является некатастрофическим и имеет $d_{\text{free}} \geqslant \min\{d_g, 2d_n\}$.*

Доказательство. Обозначим через $\tilde{P}(x)$ многочлен, корни которого равны корням $P(x)$ со знаком минус. Таким образом, $\tilde{P}(x) = \pm P(-x)$ и $W[\tilde{P}(x)] = W[P(x)]$. Любой многочлен $Q(x^2)$ может быть разложен в произведение вида $P(x)\tilde{P}(x)$, так как все корни появляются парами $\pm\sqrt{r^2}$. В частности, $x^{2n} - 1 = (x^n - 1)(x^n + 1)$, и если α — корень $x^n - 1$, то $-\alpha$ — корень $x^n + 1$. При нечетном n как α, так и $-\alpha$ не могут одновременно быть корнями $x^n - 1$, и, следовательно, они одновременно не могут быть корнями многочлена $g(x)$.

Теперь имеем $G(D) = G_1(D^2) + DG_2(D^2)$; таким образом, если α — общий корень многочленов $G_1(D^2)$ и $G_2(D^2)$, порождающих код, то $-\alpha$ также является общим корнем. Отсюда получаем, что $\text{HOD}\{G_1, G_2\} = 1$, следовательно, код — некатастрофический [6].

Для любого многочлена $I(D) \neq 0$ можно записать

$$T(D) = I(D^2)\, g(D) = I_1(D)\, \tilde{I}_1(D)\, g(D) = \\ = P(D)\, g(D)^{l+1}\, \tilde{g}(D)^i\, h(D)^j\, \tilde{h}(D)^j, \qquad (4)$$

где $P(D)$ не делится ни на какой из многочленов $g(D)$, $\tilde{g}(D)$, $h(D)$, $\tilde{h}(D)$.

Предположим, что $i \geqslant j$. Тогда (4) приводит к

$$T(D) = P(D)\, g(D)^{i-j+1}\, \tilde{g}(D)^{i-j} (D^{2n} - 1)^j,$$

так как $(\tilde{D^n} - 1) = D^n + 1$. Ни один из корней $\tilde{g}(x)$ не является корнем $(x^n - 1)$, так что $h(x)$ не может быть делителем $\tilde{g}(x)^{i-j}$. Применяя теорему 1, находим

$$W[T(D)] \geqslant W[(D-1)^j]\, W[P(D)\, g(D)^{i-j+1}\, \tilde{g}(D)^{i-j} \bmod (D^{2n} - 1)] \geqslant \\ \geqslant W[P(D)\, g(D)^{i-j+1}\, \tilde{g}(D)^{i-j} \bmod (D^n - 1)] \geqslant d_g. \qquad (5)$$

Предположим теперь, что $i < j$. Тогда из (4) имеем

$$T(D) = P(D)\, h(D)^{j-i-1}\, \tilde{h}(D)^{j-i} (D^n - 1)(D^{2n} - 1)^i.$$

Применяя снова теорему 1, получим

$$W[T(D)] \geqslant \\ \geqslant W[(D-1)^i]\, W[P(D)\, h(D)^{j-i-1}\, \tilde{h}(D)^{j-1} (D^n - 1) \bmod (D^{2n} - 1)].$$

Здесь заметим, что $\tilde{h}(x)(x^n - 1)$ является порождающим многочленом циклического кода длины $2n$ с минимальным расстоянием $2d_h$, так как $\tilde{h}(x)$ — делитель $x^n + 1$ и $W[\tilde{P}(x)] = W[P(x)]$. Следовательно,

$$W[T(D)] \geqslant 2d_h. \qquad (6)$$

Теперь утверждение теоремы следует из (5) и (6).

Теорема 5 не применима к порождающим многочленам кодов Рида — Соломона в их обычной форме, так как эти коды имеют четные длины при нечетной характеристике поля. Однако p-ичные циклические коды с достижимым максимальным расстоянием, порождаемые $g(x) = (x-1)^i$ [8], [9], [1], могут быть использованы в теореме 5 для того, чтобы получить хорошие p-ичные сверточные коды.

4. СВЕРТОЧНЫЕ КОДЫ НАД НЕБОЛЬШИМИ ПОЛЯМИ

В этом разделе будет изучено ограничение для кодов, определенных в теореме 4, состоящее в рассмотрении небольших полей. Так как блоковые БЧХ-коды могут быть получены просто как подкоды кодов Рида — Соломона с коэффициентами из подполя [10],

то мы должны рассмотреть здесь дополнительное условие, накладываемое теоремой 3. Таким образом, по (примитивному или непримитивному) БЧХ-коду с достаточно большой скоростью $R \geqslant \varkappa/\nu$ можно построить сверточный код со скоростью $\varkappa/\nu$ со свободным расстоянием, ограниченным снизу минимальным расстоянием БЧХ-кода, но с более короткой ограничительной длиной $n_A \approx \approx n(1-R)$.

Следующая теорема показывает, что, когда сверточные коды образуются из некоторых непримитивных БЧХ-кодов, условие теоремы 3 может быть проверено без детального просмотра корней порождающего многочлена.

Теорема 6. *Пусть $n = \mu\nu$ является делителем $q^{\mu-1}-1$ и μ — простое число, которое не является делителем никакого числа вида q^r-1 при $r < \mu-1$. Если $g(x)$ — порождающий многочлен q-ичного циклического кода длины n, самое большее $\nu-\varkappa$ корней ν-й степени из единицы в $GF(q^{\mu-1})$ являются корнями $g(x)$ и самое большее $\nu-\varkappa$ неприводимых многочленов степени $\mu-1$ являются делителями $g(x)$, то сверточный код, определяемый $G(x)=g(x)$, является некатастрофическим и имеет свободное расстояние $d_{\text{free}} \geqslant d_g$.*

Доказательство. Оно следует из предположений, что неприводимыми делителями x^n-1 [7, § 4.4] являются ν многочленов степени $\mu-1$, $(x-1)$ и один или большее число минимальных многочленов корней ν-й степени из единицы α^{μ}, $\alpha^{2\mu}$, ..., $\alpha^{(\nu-1)\mu}$, где α является первообразным корнем n-й степени из единицы в $GF(q^{\mu-1})$.

Для того чтобы два корня одного и того же неприводимого многочлена были ν-эквивалентными, должно быть

$$\alpha^{tq^i} = \alpha^{t+j\mu}, \quad i < \mu-1,$$

что означает, что

$$t(q^i-1) \equiv j\mu \bmod n, \quad i < \mu-1,$$

и так как μ — делитель n, то

$$t(q^i-1) = j'\mu, \quad i < \mu-1. \tag{7}$$

Так как μ не является делителем (q^i-1), то (7) удовлетворяется только тогда, когда t кратно μ. Таким образом, можно заключить, что, в то время как все корни минимальных многочленов корней ν-й степени из единицы принадлежат ν-классу, содержащему 1, корни каждого неприводимого делителя степени $\mu-1$ включают ровно один элемент из каждого из остающихся ν-классов. Следовательно, условие теоремы 3 удовлетворяется, если самое большее $\nu-\varkappa$ неприводимых делителей степени $\mu-1$ являются делителями

многочлена $g(x)$ и самое большее $\nu - \varkappa$ корней ν-й степени из единицы являются корнями многочлена $g(x)$.
Применим теорему 6 в следующих двух примерах.

Пример 1. Так как $65 = 5 \cdot 13$ является делителем $2^{12} - 1$ и 13 не является делителем никакого меньшего числа вида $2^r - 1$, то $x^{65} + 1$ имеет 5 неприводимых делителей степени 12 и другими делителями являются $x + 1$ и минимальный многочлен α^{13}. Мы получаем сверточные коды, приведенные в табл. 1. Интересно отметить [11], что код со скоростью $1/5$ может быть улучшен с помощью использования минимального многочлена α^{13} вместо $x + 1$. Этот порождающий многочлен будет содержать ровно 4 элемента каждого ν-класса.

Таблица 1

Сверточные коды, полученные из циклических кодов длины 65. Порождающий многочлен является произведением минимальных многочленов α^i, где α — первообразный корень 65-й степени из единицы, а i принимает значения, указанные в столбце «корни для $g(x)$».

$\varkappa/\nu$	Корни для $g(x)$	d_{free}	n_A
$4/5$	0, 1	$\geqslant 6$	14
$3/5$	0, 1, 3	$\geqslant 10$	26
$2/5$	0, 1, 3, 5	$\geqslant 14$	38
$1/5$	0, 1, 3, 5, 7	$\geqslant 22$	50
$1/5$	1, 3, 5, 7, 13	$\geqslant 25$	53

Пример 2. Число $20 = 4 \cdot 5$ является делителем $3^4 - 1$ и 5 не является делителем никакого меньшего числа вида $3^r - 1$. Неприводимые делители многочлена $x^{20} - 1$ представляют собой 4 многочлена степени 4, $x + 1$, $x - 1$ и минимальный многочлен α^5.

С помощью порождающего многочлена, имеющего корни α α^2, α^4, α^5, α^{10}, получаем троичный код со скоростью $1/4$, у которого $n_A = 16$ и $d_{\text{free}} \geqslant 11$.

К сожалению, применимость теоремы 6 довольно ограничена, но она проливает некоторый свет на связь между делителями n и максимальной степенью порождающих многочленов, которые удовлетворяют условию теоремы 3. В общем случае эта связь является более сложной, но наилучшие порождающие многочлены могут быть легко определены с помощью последовательного вычисления корней минимальных многочленов.

В табл. 2 приведен ряд двоичных и троичных кодов. Величины d_{free}/n_A для длинных двоичных кодов из этой таблицы сравнимы с асимптотическими значениями границы Гильберта и более сильной нижней границей для свободного расстояния в несистематиче-

ских кодах, полученной Нейманом [12]. Двоичный код со скоростью $^2/_3$, у которого $n_A = 102$, имеет $d_{\text{free}}/n_A = 0{,}22$, в то время как значение неймановской границы всего лишь равно 0,12. Код со скоростью $^1/_3$, у которого $n_A = 517$, имеет $d_{\text{free}}/n_A = 0{,}22$, в то время как значения гильбертовской и неймановской границ равны 0,17 и 0,34.

Таблица 2

Сверточные коды над $GF(q)$, полученные из циклических кодов длины n. Порождающий многочлен является произведением минимальных многочленов α^i, где α — первообразный корень n-й степени из единицы, а i принимает значения, указанные в столбце «корни для $g(x)$»

q	n	$\varkappa/\nu$	Корни для $g(x)$	d_{free}	n_A
2	129	$^2/_3$	0, 1, 3	$\geqslant 10$	30
2	129	$^1/_3$	0, 1, 3, 5, 7, 9, 11	$\geqslant 26$	86
2	255	$^2/_3$	$-1, -3, 0, 1, 3, 5, 7, 9$	$\geqslant 16$	58
2	255	$^1/_3$	$-17, -15, \ldots, -1, 0, 1, \ldots, 17$	$\geqslant 38$	130
2	023	$^2/_3$	$-9, -7, \ldots, -1, 0, 1, \ldots, 7, 9$	$\geqslant 22$	102
2	023	$^1/_3$	$-31, -29, \ldots, 0, \ldots, 77, 79$	$\geqslant 116$	517
3	242	$^1/_2$	$-11, -10, \ldots, 0, \ldots, 10, 11$	$\geqslant 26$	82
3	244	$^3/_4$	0, 1, 2, 4, 5	$\geqslant 14$	42
3	244	$^2/_4$	0, 1, 2, 4, ..., 13, 14	$\geqslant 32$	102
3	244	$^1/_4$	0, 1, 2, ... 17, 19, 20	$\geqslant 44$	144

5. АСИМПТОТИЧЕСКИ ХОРОШИЕ ПЕРИОДИЧЕСКИЕ СВЕРТОЧНЫЕ КОДЫ

В этом разделе мы построим для любого R, $0 < R < 1$, последовательность сверточных кодов со скоростями $R_{n_A} \geqslant R$ при растущем n_A, такую, что свободные расстояния и ограничительные длины будут удовлетворять условию

$$\liminf_{n_A \to \infty} d_{\text{free}}/n_A > 0. \tag{8}$$

Для того чтобы упростить доказательство, рассмотрим лишь двоичные коды.

Пусть $g_1(x)$ — порождающий многочлен сверточного кода со скоростью $R_m = \varkappa/\nu$ над $GF(2^m)$, полученного с помощью теоремы 4. Запишем D-преобразование кодовой последовательности:

$$T_1(D) = t_0 + t_1 D + t_2 D^2 + \ldots + t_u D^u + \ldots.$$

Методом, подобным тому, который использован при построении в [2], определим сверточный код, меняющийся со временем и имею-

щий скорость $\varkappa/2\nu$, с помощью кодовой последовательности

$$T(D) = t_0 + t_0 D + t_1 D^2 + \alpha t_1 D^3 + \ldots + t_u D^{2u} + \alpha^u t_u D^{2u+1} + \ldots, \quad (9)$$

где α — примитивный элемент из $GF(2^m)$.

Иначе мы могли бы определить этот сверточный код с помощью двух многочленов, порождающих код, фиксированного многочлена $g_1(x)$ и изменяющегося со временем многочлена, но мы предпочли более простое определение непосредственно в терминах кодовой последовательности (9).

Выразим элементы $GF(2^m)$ как двоичные m-компонентные векторы и будем интерпретировать код, определяемый (9), как изменяющийся со временем двоичный сверточный код. Период кода равен $2m(2^m - 1)$, так как $\alpha^{2^m-1} = 1$.

При доказательстве теоремы 3 было сделано замечание, что кодовая последовательность $T_1(D)$ могла быть записана в виде

$$T_1(D) = g_1(D)\, P(D)\, (D^n - 1)^N,$$

где $g_1(D)P(D)$ не делится на $(D^n - 1)$. Нам потребуются, кроме веса, некоторые другие сведения относительно $T_1(D)$, поэтому приведем несколько этапов доказательства теоремы 1 [1].

Для того чтобы построить границу для хэмминговского веса многочлена вида $q(x) \cdot (x^n + 1)^N$, запишем

$$q(x) = q_0(x^n) + x q_1(x^n) + \ldots + x^{n-1} q_{n-1}(x^n).$$

Тогда

$$W[q(x)(x^n + 1)^N[= \sum_{i=0}^{n-1} W[q_i(x^n)(x^n + 1)^N],$$

и теорема легко доказывается, если заметить, что для каждого слагаемого

$$W[q_i(x^n)(x^n + 1)^N] \geqslant W[q_i(1)]\, W[(x + 1)^N].$$

Однако вес каждого из этих слагаемых равен весу символов, которые умножаются на α^i, когда составляется последовательность (9), и поэтому

$$W[P(D)\, g(D) \bmod (D^n - 1)] \geqslant d_g$$

представляет собой нижнюю границу для числа ненулевых символов в $T_1(D)$, которые умножаются на различные степени α, когда составляется $T(D)$.

Теперь мы в состоянии использовать лемму, выведенную в [2].

Лемма 2. *Пусть* $o_2(L) \to 0$ *при* $L \to \infty$. *Тогда для любого* γ, $0 < \gamma < 1$, *и любого* δ, $0 < \delta < 1$, *общий хэмминговский вес* W *различных* $M_L = [\gamma - o_2(L)][2^{L\delta} - 1]$ *ненулевых двоичных* L*-последовательностей удовлетворяет неравенству*

$$W \geqslant \gamma L [H^{-1}(\delta) - o_1(L)](2^{L\delta} - 1).$$

Здесь H^{-1} обозначает функцию, обратную к двоичной энтропии и $o_1(L) \to 0$ при $L \to \infty$. Применим лемму 2 при $L = 2m$, $\delta = 1/2$ и $\gamma = 1 - R_m$. Будем иметь

$$d_{\text{free}} \geqslant 2m(1 - R_m)\left[H^{-1}(1/2) - o_1(2m)\right](2^m - 1).$$

Теперь $n_A = 2m(2^m - 1)(1 - R_m)$ и, следовательно,

$$\liminf_{n_A \to \infty} d_{\text{free}}/n_A \geqslant H^{-1}(1/2) \tag{10}$$

для любой скорости $R < 1/2$.

Так же, как и в [2], получим коды с большими скоростями с помощью отбрасывания последних s двоичных символов каждого произведения $t_u\alpha^u$, при этом обозначим через $[t_u\alpha^u]_s$ получающийся $(m - s)$-компонентный вектор. Видоизменим кодовую последовательность, взяв ее в виде

$$T_s: t_0, \quad [t_0]_s, \quad t_1, \quad [\alpha t_1]_s, \quad \ldots, \quad t_u, \quad t_u[\alpha^u]_s, \quad \ldots. \tag{11}$$

Имеются по меньшей мере $(2^m - 1)(1 - R_m)2^{-s}$ различных ненулевых векторов вида $\{t_u, [t_u\alpha^u]_s\}$, и мы можем вновь применить лемму 2, положив $L = 2m - s$, $\delta = (m - s)/(2m - s)$ и $\gamma = 1 - R_m$. Общий вес ненулевых векторов удовлетворяет неравенству

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

Отметим, что скорость этого двоичного кода равна $R_{n_A} = R_m m/(2m - s)$, но наилучшая граница получена для R_m, близких к 1, так что $R_{n_A} \approx m/(2m - s)$. Ограничительная длина равна

$$n_A = (2m - s)(2^m - 1)(1 - R_m).$$

Таким образом,

$$\liminf_{n_A \to \infty} d_{\text{free}}/n_A \geqslant H^{-1}(1 - R). \tag{12}$$

Объединим (10) и (12) для того, чтобы получить следующую теорему.

Теорема 7. *Двоичные периодические сверточные коды, определяемые посредством* (9) *и* (11), *имеют свободные расстояния и ограничительные длины, удовлетворяющие неравенству*

$$\liminf_{n_A \to \infty} d_{\text{free}}/n_A \geqslant \begin{cases} H^{-1}(1/2), & 0 < R < 1/2, \\ H^{-1}(1 - R), & 1/2 \leqslant R < 1. \end{cases}$$

Граница теоремы 7 нанесена на рис. 1 вместе с соответствующей границей для блоковых кодов, полученной в [2]. Для скоростей, больших чем $^1/_2$, граница, полученная в теореме 7, равна нижней границе Гильберта для блоковых кодов. С помощью каскадирования сверточных кодов из теоремы 4 с хорошими фиксированными блоковыми кодами, имеющими скорости, меньшие чем $^1/_2$, можно получить класс сверточных кодов, свободные расстояния ко-

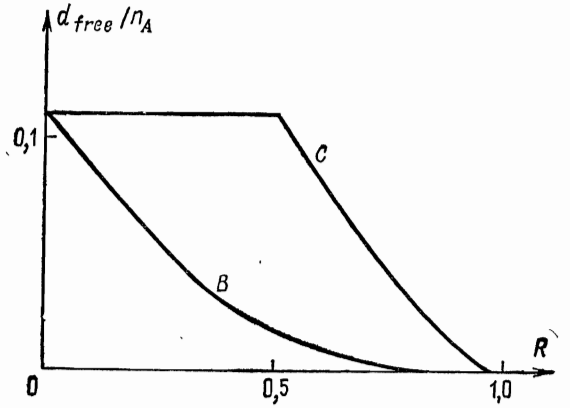


Рис. 1. Сравнение границы для d_{free}/n_A для сверточных кодов, построенных в разд. 5 (кривая C), и границы, полученной ранее для класса блоковых кодов (кривая B).

торых ограничены снизу границей Гильберта для всех скоростей $0 < R < 1$ [10], [13].

Автору хотелось бы поблагодарить профессора Дж. Л. Месси и рецензентов за некоторые полезные предложения.

СПИСОК ЛИТЕРАТУРЫ

1. Massey J. L., Costello D. J., Justesen J., Polynomial weights and code constructions, *IEEE Trans. Inform. Theory*, **IT-19**, (Jan. 1973) 101—110. (Русский перевод: Месси Дж. Л., Кастелло Д. Дж., Юстесен Й., Веса многочленов и кодовые конструкции. Кибернетический сборник вып. 11 «Мир», М., 1970, стр. 24—27).
2. Justesen J., A class of constructive asymptotically good algebraic codes, *IEEE Trans. Inform. Theory*, **IT-18**, (Sept. 1972) 652—656. (Русский перевод: Юстенсен Й., Класс конструктивных асимптотически хороших конструктивных кодов, Кибернетический сборник, вып. 10. «Мир», М., 1973, стр 39—50.)
3. Massey J. L., Majority decoding of convolutional codes, Res. Lab. Electron Quart. Prog. Rep. 64, Massachusetts Inst. Technol. Cambridge, (Jan. 15, 1962) 183—188.
4. Forney G. D., Jr., Convolutional codes I: Algebraic structures, *IEEE Trans. Inform. Theory*, **IT-16** (Nov. 1970), 720—738.

5. Massey J. L., Some algebraic and distance properties of convolutional codes, in H. B. Mann, Error Correcting Codes, New York, Wiley, 1968.
6. Massey J. L., Sain M. K., Inverses of linear sequential circuits, *IEEE Trans. Comput.*, **C-17**, (Apr. 1968) 330—337.
7. Berlekamp E. R., Algebraic Coding Theory, New York, McGraw-Hill, 1968. (Русский перевод: Берлекэпм Э., Алгебраическая теория кодирования, «Мир», М., 1971.)
8. Assmus E. F., Mattson H. F., Jr., New 5-designs, *J. Comp. Theory*, **6** (1969), 122—151.
9. Берман С. Д., К теории групповых кодов, Кибернетика, 3, № 1 (1967), 31—39.
10. Forney G. D., Jr., Concatenated Codes, Cambridge, Mass., M. I. T. Press, 1966. (Русский перевод: Форни Д., Каскадные коды, «Мир», М., 1970.)
11. Chen C. L., Computer results on the minimum distance of some binary cyclic codes, *IEEE Trans. Inform. Theory* (Corresp.), **IT-16**, (May 1970), 359—360.
12. Neumann B., Distance properties of convolutional codes, M. S. thesis, Massachusetts Inst. Technol., Cambridge, (Aug. 1968).
13. Зяблов В. В., Оценка сложности построения двоичных линейных каскадных кодов, Проблемы передачи информации, 7, вып. 1 (1971), 5—13.

Широковещательные каналы [1)]

Т. М. Ковер

1. ВВЕДЕНИЕ

В этой работе делается попытка развить некоторые интуитивные представления по проблеме одновременной передачи информации от одного источника к нескольким приемникам. Примерами одновременной передачи информации служат радиовещание, телепередачи от передатчика к многочисленным приемникам в данном районе, лекции для группы слушателей с различной подготовкой.

Будет найдено, что предлагаемая нами модель применима также к случаю, когда рассматриваются составные каналы, в которых передатчик не знает истинные характеристики канала, но хочет вести передачу информации приемнику с некоторой заранее заданной скоростью.

Общий широковещательный канал с k приемниками изображен на рис. 1. Подробнее его описание приведено в разд. 3. Основная проблема состоит в отыскании множества одновременно достижимых скоростей передачи $(R_1, R_2, \ldots, R_k)$.

Предположим, что передающие каналы от передатчика к приемникам имеют соответственно пропускные способности $C_1, C_2, \ldots, C_k$ бит в сек. Первое предложение, которое можно выдвинуть, использует максиминный подход: вести передачу со скоростью $C_{\min} = \min\{C_1, C_2, \ldots, C_k\}$. Даже эта скромная цель достижима лишь тогда, когда каналы в некотором смысле совместны (общее выражение дается в разд. 9). Если каналы совместны, то каждый приемник принимает без ошибок со скоростью $R = C_{\min}$ бит/сек. В этом случае скорость передачи лимитирована наихудшим каналом. В другом крайнем случае информацию можно было бы передавать со скоростью $R = C_{\max}$, со скоростями $R_i = 0$, $i = 1, 2, \ldots, k-1$ (во всех каналах кроме наилучшего) и скоростью $R_k = C_{\max}$ в наилучшем канале.

Следующая идея состоит в разделении времени. Разделим время так, чтобы на интервалах $\lambda_1, \lambda_2, \ldots, \lambda_k$, $\lambda_i \geqslant 0$, $\Sigma\lambda_i = 1$, передача велась со скоростями $C_1, C_2, \ldots, C_k$ соответственно. Предполагая совместимость каналов и предполагая, что $C_1 \leqslant C_2 \leqslant \ldots \leqslant$

[1)] Cover T. M., Broadcast channels, *IEEE Transactions on Information Theory*, IT-18, № 1, (1972), 2—14.

$\leqslant C_k$, находим выражение для скорости передачи информации по i-му каналу:

$$R_i = \sum_{j \leqslant i} \lambda_j C_j, \quad i = 1, 2, \ldots, k.$$

Насколько нам известно, в литературе не обсуждались никакие другие схемы передачи и не была сформулирована задача передачи по широковещательным каналам.

В этой работе будет показано, что можно превысить даже это семейство скоростей. В частности, будет показано, что при небольшом уменьшении скорости в наихудшем канале можно получить

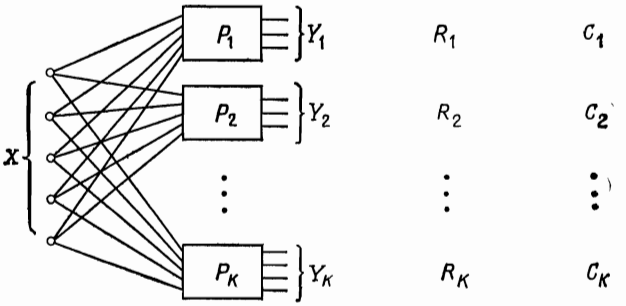


Рис. 1. Широковещательный канал; R — скорость, C — пропускная способность.

большее приращение скорости передачи для лучших каналов. Эвристическое понимание, которое возникает после нашего рассмотрения, состоит в том, что не следует одновременно передавать по нескольким каналам со скоростью, равной скорости наихудшего канала, и не следует пытаться передавать информацию методом разделения времени или уплотнения по времени, а скорее следует распределить высокоскоростную информацию среди низкоскоростных сообщений.

Будут приведены примеры хороших методов кодирования для множества двоичных симметричных каналов и для множества гауссовских каналов. Рассмотрен экстремальный случай ортогональных каналов, в котором несущественно, что можно пытаться передать два сообщения сразу двум различным людям, а также рассмотрен другой экстремальный случай несовместных каналов, в котором передача информации одному приемнику делает невозможной передачу информации другому.

2. ДВА ДВОИЧНЫХ СИММЕТРИЧНЫХ КАНАЛА

Прежде чем перейти к точным формулировкам для широковещательного канала (разд. 3), рассмотрим эвристически случай двух двоичных симметричных каналов. Необходимые определения можно найти у Эша [1] и в разд. 3.

Пусть $X = \{1, 2\}$ — входной алфавит, а $Y_1 = \{1, 2\}$ и $Y_2 = \{1, 2\}$ — выходные алфавиты приемников 1 и 2. Пусть матрицы

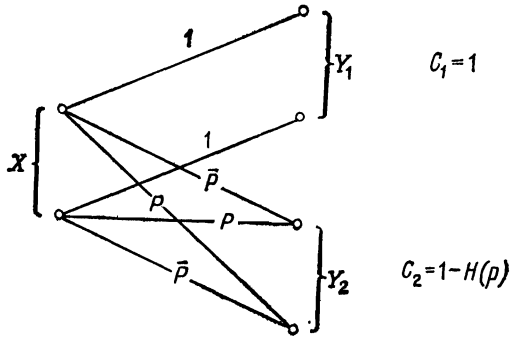


Рис. 2. Два двоичных симметричных канала.

переходных вероятностей каналов имеют вид (рис. 2)

$$P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} \bar{p} & p \\ p & \bar{p} \end{bmatrix}. \tag{1}$$

Это значит, что канал 1 бесшумный, а канал 2 является двоичным симметричным каналом (ДСК) с вероятностью ошибки p. Соответствующие пропускные способности каналов равны $C_1 = 1$ бит

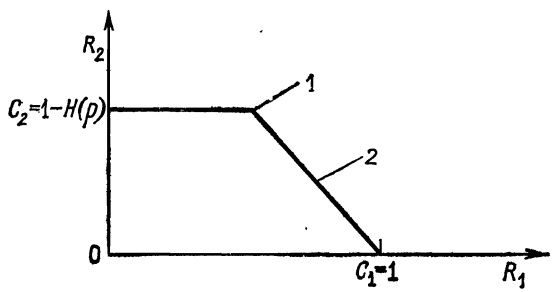


Рис. 3. Некоторые достижимые скорости для ДСК; *1* — минимаксные скорости, *2* — скорости при разделении времени.

на одну посылку и $C_2 = C(p) = 1 - H(p)$ [1]) бит на одну посылку.

Максиминный подход дает скорости передачи $(R_1, R_2) = (C_2, C_2)$ (рис. 3). (Максиминные точки по небрежности названы минимаксными точками на рисунках. Хотя в общем случае это не

[1]) $H(p) = -p \log_2 p - (1-p)\log_2(1-p)$. — *Прим. ред.*

одно и то же, во всех примерах, изображенных на рисунках, минимакс и максимин означают одно и то же). Эти скорости на самом деле могут быть одновременно достигнуты с помощью использования стандартного $(2^{n(C_2-\varepsilon)}, n, \lambda_n)$-кода для канала P_2 (см. Вольфовиц [2]).

В другом крайнем случае можно передавать со скоростью $R_1 = 1$ при нулевой вероятности ошибки на приемнике 1 и возникающей при этом скоростью $R_2 = 0$ для канала 2. Затем, передавая долю времени λ со скоростями (C_2, C_2) и долю времени $1-\lambda$ со скоростями $(1, 0)$, получим множество скоростей, изображенных на рис. 3 прямой линией. Будем называть эту линию нижней границей для множества достижимых скоростей, полученной с помощью разделения времени.

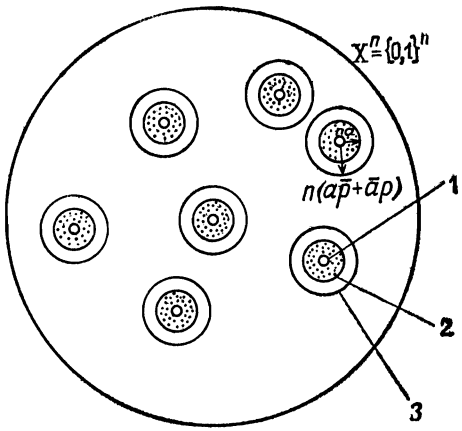


Рис. 4. Пространство кодовых слов для ДСК: *1* — кодовое слово, *2* — облако вспомогательных кодовых слов, *3* — сфера, в которой с большой вероятностью находится принятое y.

Теперь посмотрим, как сделать лучше. Из доказательства методом случайного кодирования известно, что хороший $(2^{n(C_2-\varepsilon)}, n, \lambda_n)$-код можно создать, выбирая случайно подмножество S с $2^{n(C_2-\varepsilon)}$ элементами из множества 2^n двоичных n-последовательностей $X^n = \{1, 2\}^n$ и используя метод декодирования, который сопоставляет принятый вектор $y = (y_1, y_2, \ldots, y_n)$ элементу S, находящемуся в пределах хэммингговского расстояния $n(p+\varepsilon)$ от y.

Выберем код такого типа для канала, который в некотором смысле является более шумным; а именно — для последовательного соединения ДСК с параметром p и ДСК с параметром α, что дает в результате ДСК с параметром $\alpha\bar{p} + \bar{\alpha}p$, где $\bar{\alpha} = 1-\alpha$. Таким образом, в этом множестве будет лишь $2^{n(C(\alpha\bar{p}+\bar{\alpha}p)-\varepsilon)}$ кодовых слов, но можно будет допустить больший шум порядка $n(\alpha\bar{p} + \bar{\alpha}p)$.

Используем теперь эту возможность путем размещения некоторой добавочной передаваемой информации, предназначенной лишь для безошибочного приемника Y_1.

Поставим в соответствие каждому кодовому слову x из $S \subseteq \subseteq X^n = \{1, 2\}^n$ множество всех кодовых слов, находящихся на рас-

стоянии Хэмминга $[\alpha n]$, что изображено с помощью облаков из точек на рис. 4.

Эта кодовая конструкция позволяет передать произвольное целое число $r \in \{1, 2, \ldots, 2^{n(C(\alpha\bar{p}+\bar{\alpha}p)-\varepsilon)}\}$ к обоим приемникам 1 и 2 и произвольное целое число

$$s \in \{1, 2, \ldots, C_n^{[\alpha n]}\}$$

приемнику 1. (См. разд. 3 для более подробного ознакомления.) Сообщение (r, s) посылается следующим образом. Целое число r определяет облако, а целое число s определяет точку $x \in \{1, 2\}^n$ в облаке. Затем эта n-последовательность x передается. Бесшумный канал имеет на выходе $y_1 = x$ и, таким образом, декодирует верно как r, так и s. Так как в облаке находятся

$$C_n^{[\alpha n]} \approx 2^{nH(\alpha)}$$

точек, скорость передачи для канала 1 равна

$$R_1 = \frac{1}{n} \log 2^{nH(\alpha)} 2^{n(C(\alpha\bar{p}+\bar{\alpha}p)-\varepsilon)} = C(\alpha\bar{p} + \bar{\alpha}p) + H(\alpha) - \varepsilon. \qquad (2)$$

Канал 2 воспринимает центр облака так, как если бы он был передан по добавочному ДСК с параметром α (из-за выбора s). Однако, поскольку центры облаков были выбраны так, чтобы быть различимыми в ДСК с параметром $\alpha\bar{p} + \bar{\alpha}p$, это означает, что r правильно декодируется приемником Y_2. Таким образом,

$$R_2 = C(\alpha\bar{p} + \bar{\alpha}p) - \varepsilon. \qquad (3)$$

Проведенное рассуждение позволяет предположить, что (R_1, R_2) достижимы совместно; это доказано в приложении. Заставляя α меняться между 0 и 1, получаем новое достижимое множество скоростей, изображенное на рис. 5. Мы вполне уверены, что на рис. 5 получена оптимальная область достижимых скоростей.

Полученная кривая охватывает кривую разделения времени. Заметим также, что наклон равен нулю около минимаксной точки. Таким образом, бесконечно малое уменьшение скорости в худшем канале дает инфинитезимально бесконечное увеличение скорости в лучшем канале. Следовательно, по крайней мере в случае двух ДСК, наложение информации лучше, чем разделение времени.

Этот пример естественно приводит к гипотезе относительно области пропускной способности для частного класса широковещательных каналов, в котором один канал является ухудшенным вариантом другого.

Определение. Пусть P и Q — матрицы переходных вероятностей канала порядка $|X|\times|Y_1|$ и $|X|\times|Y_2|$ соответственно. Будем говорить, что Q является *ухудшенным вариантом* P, если существует стохастическая матрица M, такая, что $P = QM$. Шен-

нон [6] показал, что пропускная способность канала Q не больше, чем пропускная способность канала P.

Гипотеза 1. Пусть S — произвольная матрица порядка $|X| \times |X|$ переходных вероятностей канала, соответствующая плотности распределения для канала $p(x|s)$, и пусть $p(s)$ — произвольное распределение вероятности на X. Пусть $p(s)$ приводит к совместному распределению $p(y_1, y_2, s, x) = p(s)\,p(x|s)\,p(y_1, y_2|x)$ на (y_1, y_2, s, x). Пусть P_2 — ухудшенный вариант P_1. Тогда множество достижимых пар (R_1, R_2) для широковещательного канала

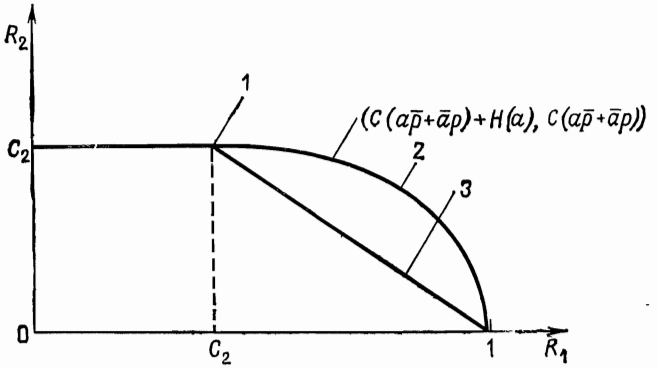


Рис. 5. Множество достижимых скоростей для ДСК; *1* — минимакс, *2* — наложение информаций, *3* — разделение времени.

$(X, p(y_1, y_2|x),\ Y_1 \times Y_2)$ задается парами $(I(S; Y_2) + I(X; Y_1|S),\ I(S; Y_2))$ для всех каналов S и всех распределений вероятностей $p(s)$.

Пример двух ДСК из этого раздела является частным случаем этой гипотезы. Код, который достигает скорости (R_1, R_2), строится аналогичным образом. Во время написания этой статьи Бергманс из Станфорда достиг некоторых успехов в доказательстве этой гипотезы. Фактически рассмотрения Бергманса позволили мне изменить первоначальную слишком сильную гипотезу, в которой фигурировал более широкий класс каналов, упомянутых в [6]. У меня нет больше никаких оснований верить в более сильную гипотезу.

3. ОПРЕДЕЛЕНИЯ И ОБОЗНАЧЕНИЯ

Определим *широковещательный канал* без памяти с двумя приемниками, который мы будем обозначать $(X, p(y_1, y_2|x),\ Y_1 \times Y_2)$ или $p(y_1, y_2|x)$, как три конечных множества X, Y_1, Y_2 и совокупность распределений вероятностей $p(\cdot, \cdot|x)$ на $Y_1 \times Y_2$, одно распределение для каждого $x \in X$. Интерпретация определения

состоит в том, что x — вход канала, а y_1 и y_2 — соответственно входы приемников 1 и 2, как показано на рис. 6. Задача состоит в одновременной передаче информации приемникам 1 и 2 с возможно большей эффективностью.

В этой работе нам нужны лишь одномерные распределения

$$p_1(y_1|x) = \sum_{y_2 \in Y_2} p(y_1, y_2|x),$$
$$p_2(y_2|x) = \sum_{y_1 \in Y_1} p(y_1, y_2|x), \tag{4}$$

которые мы ввели в примерах в виде матриц P_1 и P_2 порядка $|X| \times |Y_1|$ и $|X| \times |Y_2|$ соответственно. Возможная зависимость

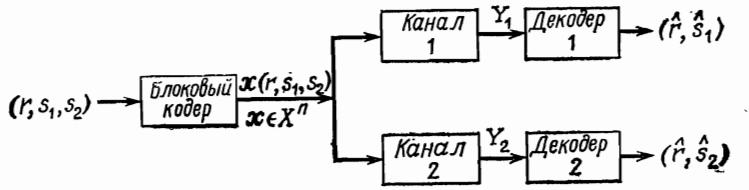


Рис. 6. Кодер и декодер для широковещательного канала.

$p_1(e) = \Pr\{(\hat{r}, \hat{s}_1) \neq (r, s_1)\}$
$p_2(e) = \Pr\{(\hat{r}, \hat{s}_2) \neq (r, s_2)\}$.

или независимость Y_1 и Y_2 при заданных X не существенна при условии, что декодирование в обоих приемниках должно выполняться независимо.

Назовем n-м расширением для широковещательного канала широковещательный канал

$$(X^n, p(\boldsymbol{y}_1, \boldsymbol{y}_2|\boldsymbol{x}), Y_1^n \times Y_2^n), \tag{5}$$

где $p(\boldsymbol{y}_1, \boldsymbol{y}_2|\boldsymbol{x}) = \prod_{j=1}^{n} p(y_{1j}, y_{2j}|x_j)$ для $\boldsymbol{x} \in X^n$, $\boldsymbol{y}_1 \in Y_1^n$, $\boldsymbol{y}_2 \in Y_2^n$.

Назовем $((M_1, M_2, M_{12}), n)$-кодом для широковещательного канала три множества целых чисел

$$R = \{1, 2, \ldots, M_{12}\}, \quad S_1 = \{1, 2, \ldots, M_1\}, \quad S_2 = \{1, 2, \ldots, M_2\},$$

функцию, задающую кодирование,

$$\boldsymbol{x}: R \times S_1 \times S_2 \to X^n$$

и две функции, задающие декодирование,

$$g_1: Y_1^n \to R \times S_1; \quad g_1(\boldsymbol{y}_1) = (\hat{r}, \hat{s}_1),$$
$$g_2: Y_2^n \to R \times S_2; \quad g_2(\boldsymbol{y}_2) = (\hat{r}, \hat{s}_2).$$

Множество $\{x(r, s_1, s_2) \mid (r, s_1, s_2) \in R \times S_1 \times S_2\}$ называется множеством кодовых слов. Как показано на рис. 6, целые числа s_1 и s_2 произвольно выбираются передатчиком для посылки приемникам 1 и 2 соответственно. Целое число r также выбирается передатчиком и предназначено для приема как тем, так и другим приемником. Таким образом, r является «общей» частью сообщения, а s_1 и s_2 дают «независимые» части сообщения.

Скажем, что i-й приемник совершает ошибку, если $g_i(y_i) \neq (r, s_i)$. Пусть передается сообщение (r, s_1, s_2); обозначим через

$$\lambda_i(r, s_1, s_2) = \Pr\{g_i(y_i) \neq (r, s_i)\}, \quad i = 1, 2, \tag{6}$$

вероятности ошибок для этих двух каналов; отметим здесь, что y_1, y_2 — единственные случайные величины в написанном выше выражении.

Обозначим (среднеарифметическую) вероятность ошибки при декодировании (r, s_1), усредненную по всем значениям s_2, через

$$\bar{\lambda}_1(r, s_1) = \frac{1}{M_2} \sum_{s_2=1}^{M_2} \lambda_1(r, s_1, s_2). \tag{7}$$

Аналогично для канала 2 определим

$$\bar{\lambda}_2(r, s_2) = \frac{1}{M_1} \sum_{s_1=1}^{M_1} \lambda_2(r, s_1, s_2). \tag{8}$$

Наконец, определим общие *среднеарифметические вероятности ошибок* кода для каналов 1 и 2 как

$$\bar{p}_1(e) = \frac{1}{M_1 M_{12}} \sum_{r, s_1} \bar{\lambda}_1(r, s_1) = \frac{1}{M} \sum_{r, s_1, s_2} \lambda_1(r, s_1, s_2), \tag{9}$$

$$\bar{p}_2(e) = \frac{1}{M_1 M_{12}} \sum_{r, s_2} \bar{\lambda}_2(r, s_2) = \frac{1}{M} \sum_{r, s_1, s_2} \lambda_2(r, s_1, s_2), \tag{10}$$

где

$$M = M_1 M_2 M_{12}. \tag{11}$$

Черта над $\bar{p}_i(e)$ сохраняется в качестве напоминания о том, что эта вероятность ошибки вычислена при некотором частном распределении, а именно при равномерном распределении на кодовых словах.

Для нас также будут представлять интерес *максимальные вероятности ошибок*

$$\lambda_i = \max_{r, s_1, s_2} \Pr\{g_i(y_i) \neq (r, s_i) \mid (r, s_1, s_2)\}, \quad i = 1, 2, \tag{12}$$

соответствующие наихудшему кодовому слову для каждого канала. Заметим, что $\lambda_i \geqslant \bar{p}_i(e)$.

Определим *скорость* (R_1, R_2, R_{12}) для $((M_1, M_2, M_{12}), n)$-кода следующим образом:

$$R_1 = \frac{1}{n} \log M_1 M_{12}, \quad R_2 = \frac{1}{n} \log M_2 M_{12}, \quad R_{12} = \frac{1}{n} \log M_{12}, \quad (13)$$

где все скорости выражены в битах на одну посылку. Таким образом, R_i — полная скорость передачи информации к приемнику i, $i = 1, 2$, а R_{12} — часть информации, общей для обоих приемников.

Замечание. Когда λ_i и $\bar{p}_i(e)$ будут относиться к n-му расширению широковещательного канала, мы часто будем обозначать это как $\lambda_i^{(n)}$, $\bar{p}_i^{(n)}(e)$.

Определение. Скорости (R_1, R_2, R_{12}) называются *достижимыми* в широковещательном канале, если для любого $\varepsilon > 0$ и для всех достаточно больших n существует $((M_1, M_2, M_{12}), n)$-код с

$$M_1 M_{12} \geqslant 2^{nR_1}, \quad M_2 M_{12} \geqslant 2^{nR_2}, \quad M_{12} \leqslant 2^{nR_{12}}, \quad (14)$$

такой, что $\bar{p}_1^{(n)}(e) < \varepsilon$, $\bar{p}_2^{(n)}(e) < \varepsilon$.

Замечание. Заметим, что общее число $M = M_1 M_2 M_{12}$ кодовых слов для кода, удовлетворяющего (14), должно быть больше, чем $2^{n(R_1+R_2-R_{12})}$.

Определение. *Областью пропускных способностей* $\mathfrak{R}^*$ для широковещательного канала называется множество всех достижимых скоростей (R_1, R_2, R_{12}).

Целью настоящей работы является определение $\mathfrak{R}^*$ для возможно более широкого класса каналов.

Замечание. Иногда через $\mathfrak{R}^*$ мы будем также обозначать множество достижимых пар (R_1, R_2). Однако при нашем настоящем понимании проблемы представляется, что рассмотрение только (R_1, R_2) и исключение из рассмотрения R_{12} приведут к загрублению и неясности исследования.

Замечание. Обобщение определения широковещательного канала при двух приемниках на случай k приемников является громоздким в смысле обозначений, но довольно очевидным, если иметь в виду следующее замечание. Множества целых чисел R, S_1, S_2 следует заменить $2^k - 1$ множествами $I(\theta)$, $\theta \in \{0, 1\}^k$, $\theta \neq 0$, и считать, что целое число $i(\theta)$, выбранное из множества $I(\theta) = \{1, 2, \ldots, M(\theta)\}$, требуется правильно принять (с помощью соответствующего выбора кода) каждым приемником j, для которого $\theta_j = 1$, $\theta = (\theta_1, \theta_2, \ldots, \theta_k)$. Тогда, например, скорость передачи по

n-му расширению широковещательного канала к i-му приемнику задается в виде

$$R_i = \frac{1}{n} \log \prod_{\substack{\theta \in \{0,1\}^k \\ \theta_i = 1}} M(\theta) = \frac{1}{n} \sum_{\substack{\theta \in \{0,1\}^k \\ \theta_i = 1}} \log M(\theta). \qquad (15)$$

В широковещательном канале с двумя приемниками соответствующие множества в новых обозначениях будут $R = I(1,1)$, $S_1 = I(1,0)$, $S_2 = I(0,1)$.

В разд. 4 рассматривается наилучшая ситуация с двумя каналами, а в разделе 5 — наихудшая ситуация.

4. ОРТОГОНАЛЬНЫЕ КАНАЛЫ

В этом разделе будет исследован широковещательный канал, в котором эффективная передача к одному из приемников никоим образом не влияет на передачу к другому приемнику. Примером такой ситуации служит обычный кинофильм, который можно одновременно демонстрировать слепому и глухому.

Рассмотрим широковещательный канал с $X = \{1, 2, 3, 4\}$, $Y_1 = \{1, 2\}$, $Y_2 = \{1, 2\}$ и с

$$P_1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \qquad P_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad (16)$$

изображенный на рис. 7. Так же, как раньше,

$$(P_k)_{ij} = \Pr\{y_k = j \mid x = i\}, \quad k = 1, 2; \quad j = 1, 2; \quad i = 1, 2, 3, 4.$$

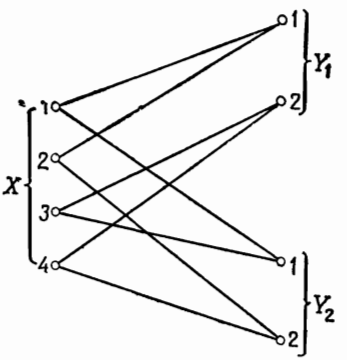


Рис. 7. Ортогональные каналы.

Легко найти, что $C_1 = C_2 = 1$ бит на посылку. Очевидно, что с точки зрения приемника Y_1 входы $x = 1$ и $x = 2$ приводят оба к $y_1 = 1$ с вероятностью 1 и поэтому могут быть объединены. Анализируя подобным образом дальше, находим, что Y_1 может определить лишь $x \in \{1, 2\}$ по отношению к $x \in \{3, 4\}$, в то время как Y_2 может определить лишь $x \in \{1, 3\}$ по отношению к $x \in \{2, 4\}$.

В этом примере $C_1 = 1$ и $C_2 = 1$, и это достигается соответственно при $\Pr\{x = 1\} + \Pr\{x = 2\} = 1/2$ и $\Pr\{x = 1\} + \Pr\{x = 3\} = 1/2$. Решая эти уравнения совместно, находим, что $(I(X|Y_1), I(X|Y_2)) = (1, 1)$ можно достичь на $\Pr\{x = i\} = 1/4$, $i = 1, 2, 3, 4$. Само по себе этим не гарантируется возможность достижения

(C_1, C_2). Однако для этого канала существует теорема кодирования. Пусть $u_1 \in \{1, 2\}$, $u_2 \in \{1, 2\}$ обозначают биты сообщения, которые требуется передать к Y_1 и Y_2 соответственно.

Установим соответствие пар u входным символам

$$\begin{aligned}(u_1, u_2) &= (1, 1) \to 1,\\ (u_1, u_2) &= (1, 2) \to 2\\ (u_1, u_2) &= (2, 1) \to 3,\\ (u_1, u_2) &= (2, 2) \to 4\end{aligned} \tag{17}$$

и пошлем соответствующий входной символ x. Тогда $y_1 = u_1$ и $y_2 = u_2$ и пропускные способности C_1 и C_2 достигаются одновременно. В силу того, что u_1 и u_2 можно выбрать независимо, можно также достичь $R_{12} = 1$ с помощью этой схемы. На рис. 8 изображено множество достижимых скоростей. Теорема о верхней границе (разд. 8) устанавливает, что эта область является оптимальной.

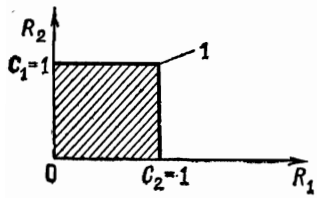


Рис. 8. Достижимые скорости для ортогональных каналов; *1* — скорости $(R_1, R_2) = (1, 1)$, достижимые при любых $0 \leqslant R_{12} \leqslant 1$.

То, что каналы бесшумные, не существенно. Широковещательный канал остается ортогональным в том смысле, что $(R_1, R_2) = (C_1, C_2)$ можно достичь, даже если определить новые каналы

$$P_1 = \begin{bmatrix} r_1 & \bar{r}_1 \\ r_1 & \bar{r}_1 \\ \bar{r}_1 & r_1 \\ \bar{r}_1 & r_1 \end{bmatrix}, \qquad P_2 = \begin{bmatrix} r_2 & \bar{r}_2 \\ \bar{r}_2 & r_2 \\ r_2 & \bar{r}_2 \\ \bar{r}_2 & r_2 \end{bmatrix}. \tag{18}$$

В этом случае $C_1 = 1 - H(r_1)$ и $C_2 = 1 - H(r_2)$. Кроме того, C_1 и C_2 можно одновременно достичь на последовательностях $(2^{n(C_1-\varepsilon)}, n, \lambda_1^{(n)})$, $(2^{n(C_2-\varepsilon)}, n, \lambda_2^{(n)})$-кодов со словами из $\{0, 1\}^n$, таких, что $\lambda_1^{(n)} \to 0$, $\lambda_2^{(n)} \to 0$ при $n \to \infty$, если выбирать $x_i \in \{1, 2\}$ или $x_i \in \{3, 4\}$ в соответствии со значением i-го бита кодового слова, выбранного для посылки из первого кода, и если выбирать $x_i \in \{1, 3\}$ или $x_i \in \{2, 3\}$ в соответствии со значением i-го бита кодового слова, выбранного из второго кода. Здесь также можно достичь любое R_{12} из интервала $0 \leqslant R_{12} \leqslant \min\{C_1, C_2\}$. Ничего большего ожидать нельзя, и каждый канал работает в присутствии другого не хуже, чем если бы он работал один.

5. НЕСОВМЕСТНЫЕ ШИРОКОВЕЩАТЕЛЬНЫЕ КАНАЛЫ

В поисках наихудшего случая несовместимости одновременной передачи обратимся к следующему практическому примеру ка-

нала, который по очевидным соображениям назовем каналом с переключением разговора.

Пример 1. *Переключение разговора.* Пусть

$$X = X_1 \cup X_2,$$

$$Y_1 = \tilde{Y}_1 \cup \{\varphi_1\},$$

$$Y_2 = \tilde{Y}_2 \cup \{\varphi_2\}$$

и

$$P_1 = \begin{array}{c} \\ X_1 \\ \\ \\ X_2 \\ \\ \end{array} \overbrace{\begin{array}{ccccc|c} x & x & x & \dots & x & 0 \\ & & & & & 0 \\ x & & & \dots & x & 0 \\ \hline 0 & 0 & & \dots & 0 & 1 \\ & & & & & 1 \\ 0 & 0 & & \dots & 0 & 1 \end{array}}^{\tilde{Y}_1 \qquad \varphi_1}, \tag{19}$$

$$P_2 = \begin{array}{c} \\ X_1 \\ \\ \\ X_2 \\ \\ \end{array} \overbrace{\begin{array}{cccc|c} 0 & 0 & \dots & 0 & 1 \\ & & & & 1 \\ 0 & 0 & \dots & 0 & 1 \\ \hline x & x & \dots & x & 0 \\ & & & & 0 \\ x & & \dots & x & 0 \end{array}}^{\tilde{Y}_2 \qquad \varphi_2},$$

как показано на рис. 9.

Каждый приемник имеет индикатор, который включается, когда посылающий передает сообщение другому приемнику. Идея состоит в том, что, когда посылающий хочет передавать сообщение к Y_1, он использует $x \in X_1$, что приводит к $y_2 = \varphi_2$, показывая приемнику 2, что посылающий передает сообщение к Y_1. Аналогично для передачи к Y_2 посылающий использует $x \in X_2$, что приводит к $y_1 = \varphi_1$. Это соответствует, например, ситуации, когда докладчик, свободно владеющий испанским и голландским языками, должен одновременно беседовать с двумя слушателями, один из которых понимает только голландский, а другой только испанский.

Пусть канал 1 имеет пропускную способность $C_1^{(0)}$, а канал 2 имеет пропускную способность $C_2^{(0)}$. Используя известный резуль-

тат для суммы каналов (см. Шеннон [3]), находим, что

$$C_1 = \log\left(1 + 2^{C_1^{(0)}}\right)$$

и

$$C_2 = \log\left(1 + 2^{C_2^{(0)}}\right).$$

Проведем неформальное обсуждение этого примера. Конечно, $(R_1, R_2) = (C_1 0)$ достигается и $(R_1, R_2) = (0, C_2)$ также достигается, и, следовательно, при разделении времени достигается любая пара

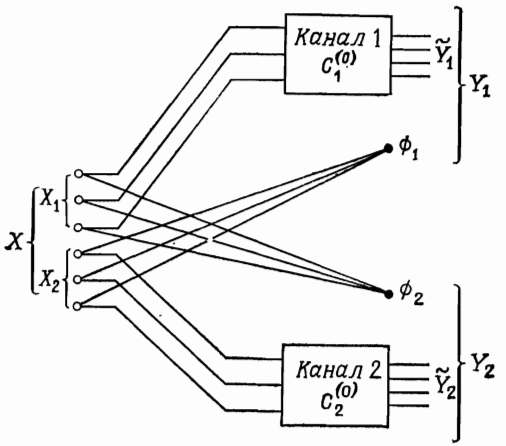


Рис. 9. Канал с переключением разговора.

скоростей $(R_1, R_2) = (\lambda C_1, \bar{\lambda} C_2)$, $0 \leqslant \lambda \leqslant 1$. Добавочная информация содержится, однако, в φ, и можно использовать подходящее кодирование с помощью моментов начала передач к Y_1 и Y_2 для того, чтобы передать дополнительную информацию по обоим каналам. Если канал 1 использует долю времени α, то приемник Y_1 получает $\alpha C_1^{(0)}$ битов на посылку. Однако дополнительно $H(\alpha)$ бит на посылку достигается, если выбирать канал для передачи независимо каждый раз, подбрасывая монету со смещением α. Другими словами, модуляция кнопкой переключения разговора при условии, что имеется ограничение на долю времени α, делает возможной безошибочную передачу одного из $2^{nH(\alpha)}$ добавочных сообщений двум приемникам Y_1 и Y_2.

Таким образом, можно достичь всех скоростей (R_1, R_2) вида $(R_1, R_2) = (\alpha C_1^{(0)} + H(\alpha), \bar{\alpha} C_2^{(0)} + H(\alpha))$, если выбирать подмножество n моментов, отведенных для использования канала 1, одним из $2^{nH(\alpha)}$ возможных способов. Эту границу нельзя достичь до тех пор, пока скорость передачи информации R_{12}, общая двум кана-

лам, не будет удовлетворять условию $R_{12} \geqslant H(\alpha)$. Результаты суммируются на рис. 10.

Простым следствием из разд. 8 является то, что рис. 10 соответствует области пропускных способностей для этого канала, и поэтому эта схема кодирования оптимальна для канала с переключением разговора.

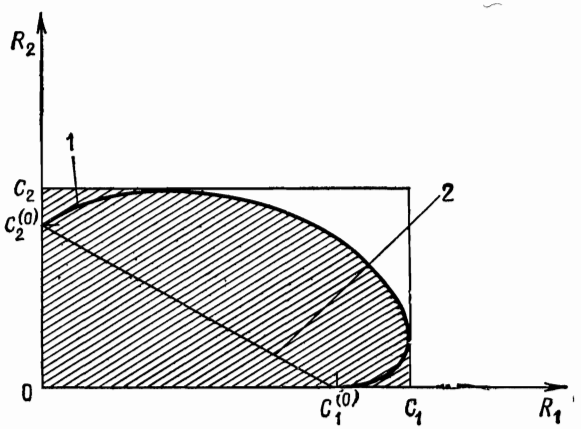


Рис. 10. Достижимые скорости для каналов с переключением разговора; $1 - (R_1 \quad R_2) = (\alpha C_1^0 + H(\alpha), \bar{\alpha} C_2^0 + H(\bar{\alpha})), R_{12} \geqslant H(\alpha)$, 2 — наивное разделение времени.

Следующий пример иллюстрирует наихудший случай, который имеет место при одновременных передачах.

Пример 2. *Случай несовместности.* Пусть

$$X = \{1, 2, 3, 4\}, \quad Y_1 = \{1, 2\}, \quad Y_2 = \{1, 2\},$$

и пусть

$$P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}, \qquad P_2 = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{20}$$

как показано на рис. 11. Таким образом, если X хочет передавать к Y_1 по каналу без шума $x \in \{1, 2\} \to Y_1$, он должен передать чистый шум к Y_2, т. е. $\Pr\{y_2 = 1 \mid x \in \{1, 2\}\} = 1/2$. Аналогичное утверждение справедливо, когда X передает к Y_2.

В разд. 8 будет найдена верхняя граница для области пропускных способностей путем отыскания множества всех достижимых пар $(I(X|Y_1), I(X|Y_2))$. Предвосхищая эти результаты, произведем вычисления для этого примера. Пусть $\Pr\{x = i\} = p_i$, $i = 1, 2, 3, 4$.

Определим $\alpha = p_1 + p_2$, $\bar{\alpha} = p_3 + p_4$. Тогда $H(Y_1) = H(p_1 + \bar{\alpha}/2)$ и $H(Y_1|X) = \bar{\alpha}$, откуда $I(X|Y_1) = H(p_1 + \bar{\alpha}/2) - \bar{\alpha}$. Аналогично $I(X|Y_2) = I(X|Y_2) = H(p_4 + \alpha/2) - \alpha$.

Сначала, фиксируя α, $\bar{\alpha}$ и производя максимизацию по $0 \leqslant p_1 \leqslant \alpha$, $0 \leqslant p_4 \leqslant \bar{\alpha}$, найдем максимальные значения

$$I(X|Y_1) = 1 - \bar{\alpha} = \alpha,$$
$$I(X|Y_2) = 1 - \alpha = \bar{\alpha}, \qquad (21)$$

которые достигаются при $p_1 = p_2 = \alpha/2$ и $p_3 = p_4 = \bar{\alpha}/2$. Это представляет собой верхнюю границу для достижимых пар $(I(X|Y_1), I(X|Y_2))$.

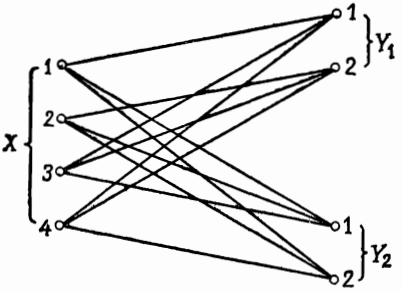


Рис. 11. Несовместные щироковещательные каналы.

Можно также проверить, что для любого $\alpha \in [0, 1]$ существуют p_1, p_2, p_3, p_4, на которых достигается любая пара $(I(X|Y_1), I(X|Y_2))$, меньшая $(\alpha, 1-\alpha)$. Таким образом, получаем множество I достижимых пар $(I(X|Y_1), I(X|Y_2))$, изображенное на рис. 12.

В разд. 8 будет показано, что эта область одновременно достижимых пар $(I(X|Y_1), I(X|Y_2))$ является верхней границей области пропускных способностей. Одако мы можем тривиально достичь любую пару скоростей (R_1, R_2), лежащую на верхней границе области $\mathscr{R}$, с помощью простого разделения времени между двумя бесшумными каналами $x \in \{1, 2\} \to Y_1$ и $x \in \{3, 4\} \to Y_2$. Если каналом $x \in \{1, 2\}$ используется доля времени α, то без всякого добавочного кодирования можно достичь скоростей $R_1 = \alpha$ и $R_2 = \bar{\alpha} = 1 - \alpha$. Таким образом, верхняя граница может быть достигнута на тривиальных процедурах кодирования, и поэтому рис. 12 изображает область пропускных способностей.

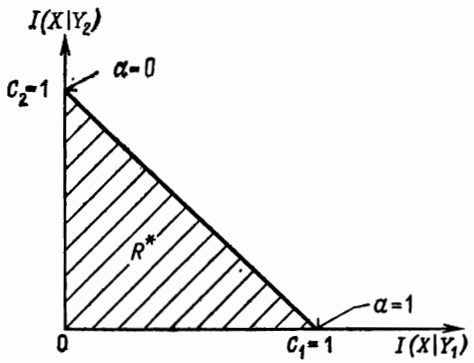


Рис. 12. Область пропускных способностей несовместных каналов.

Таким образом, здесь приведен пример, в котором два канала столь несовместны, что нельзя придумать ничего лучшего, чем разделение времени, т. е. использовать один канал эффективно в течение доли времени и другой канал — в оставшееся время.

К счастью для тех, кто любит получать что-либо из ничего, это скорее исключение, а не правило.

6. УЗКИЙ КАНАЛ

Рассмотрим широковещательный канал, в котором оба канала имеют одну и ту же структуру, т. е.

$$p_1(y_1|\,x) = p_2(y_2|\,x), \quad \forall x \in X, \quad \forall y_1,\ y_2 \in Y_1 = Y_2 = Y,$$

как показано на рис. 13. Будем называть его узким каналом.

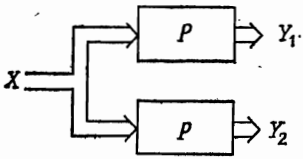


Рис. 13. Узкий канал.

Отметим здесь, что любой код для приемника Y_1 также является кодом с той же вероятностью ошибки для приемника Y_2. Таким образом, Y_1 и Y_2 воспринимают правильно переданную последовательность $\boldsymbol{x}$ с малой вероятностью ошибки.

Обозначим пропускную способность канала P через $C_1 = C_2 = C$ бит на посылку. Теперь, так как оба приемника получают одну и ту же информацию относительно X, то как приемник 1, так и 2 способны правильно воспринять r, s_1 и s_2 тогда и только тогда, когда (R_1, R_2, R_{12}) являются достижимыми скоростями. Находя число посылок за единицу времени, которые необходимы, чтобы передать (r, s_1, s_2) верно, получим следующее предложение (см. замечание после формулы (14)).

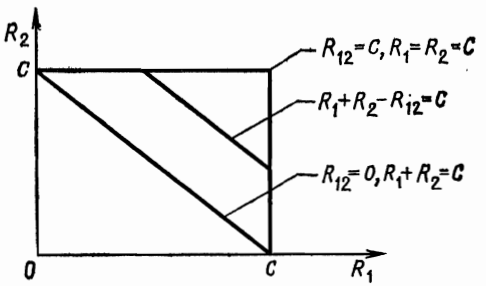


Рис. 14. Достижимые скорости для узкого канала.

Предложение. Скорости (R_1, R_2, R_{12}) достижимы в широковещательном узком канале с пропускной способностью C (см. рис. 14) тогда и только тогда, когда

$$R_1 + R_2 - R_{12} \leqslant C, \quad 0 \leqslant R_1 \leqslant C, \quad 0 \leqslant R_2 \leqslant C, \quad 0 \leqslant R_{12} \leqslant C. \quad (22)$$

В качестве важного применения изложенного предположим, что требуется передать случайный процесс $U = \{U_n : n = 1, 2, \ldots\}$ к

приемнику 1 и случайный процесс $V = \{V_n : n = 1, 2, \ldots\}$ к приемнику 2 по узкому каналу P со сколь угодно малой вероятностью ошибки (см. рис. 15).

Пусть $U = \{U_n\}$ и $V = \{V_n\}$ образуют совместно эргодический процесс со значениями в конечных алфавитах. Понятие «совместно эргодический» означает, что процесс $Z_n = (U_n, V_n)$ эргодический. Напомним, что энтропия эргодического процесса $\{Z_n\}$ определяется равенством

$$H(Z) = \lim_{n \to \infty} n^{-1} H(Z_1, Z_2, \ldots, Z_n). \tag{23}$$

Докажем следующее утверждение.

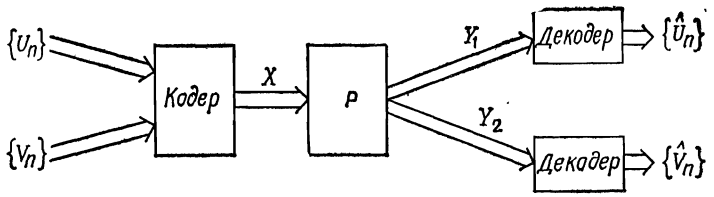


Рис. 15. Передача двух случайных процессов по одному и тому же каналу.

Утверждение. Асимптотически безошибочная передача $\{U_1, U_2, \ldots, U_n\} \to \{\hat{U}_1, \hat{U}_2, \ldots, \hat{U}_n\}$ и $\{V_1, V_2, \ldots, V_n\} \to \{\hat{V}_1, \hat{V}_2, \ldots, \hat{V}_n\}$ по узкому каналу с пропускной способностью C может быть произведена тогда и только тогда, когда

$$H(U, V) < C. \tag{24}$$

Доказательство. Хорошо известным способом кодирования является нумерация $2^{n(H(U, V)+\varepsilon)}$ ε-типичных последовательностей и передача номера в действительности появившейся последовательности $(z_1, z_2, \ldots, z_n)$ по каналу. Если $H(U, V) + \varepsilon < C$, то этот номер будет передан с вероятностью ошибки $\varepsilon/2$ для достаточно больших n. Так как вероятность того, что случайная последовательность $(z_1, z_2, \ldots, z_n)$ будет типичной, может быть сделана $\geqslant 1 - \varepsilon/2$ для достаточно больших n, то полная вероятность ошибки может быть сделана меньшей ε. Обращение доказывается с помощью обычных рассуждений для случая одного-единственного канала.

Обобщение этого результата на произвольные широковещательные каналы неизвестно.

Сравним теперь ортогональный канал с узким каналом. В ортогональном канале из разд. 4 достигается $(R_1, R_2) = (1, 1)$ при *любой* общей скорости $0 \leqslant R_{12} \leqslant 1$. Таким образом, абсолютно независимые сообщения ($R_{11} = 0$) или полностью зависимые сообщения ($R_{11} = 1$) можно посылать одновременно приемником 1 и 2.

В другом крайнем случае — узкий канал с пропускной способностью $C = 1$ — можно одновременно достичь $R_1 = 1$, $R_2 = 1$. Однако здесь можно заметить, что достижение $(R_1, R_2) = (1, 1)$ означает, что $R_{12} = 1$. Таким образом, сообщения, передаваемые к 1 и 2, должны быть полностью зависимыми и на самом деле идентичными.

7. ГАУССОВСКИЕ КАНАЛЫ

Рассмотрим дискретный по времени гауссовский широковещательный канал с двумя приемниками, изображенный на рис. 16.

Пусть $z_1 = (z_{11}, z_{12}, \ldots, z_{1n}, \ldots)$ — последовательность независимых одинаково распределенных (н. о. р.) нормальных случайных величин (с. в.) с нулевыми средними значениями и дисперсиями N_1, и пусть $z_2 = (z_{21}, z_{22}, \ldots, z_{2n}, \ldots)$ — н. о. р. нормальные с. в. со средними значениями, равными нулю, и дисперсиями N_2. Пусть $N_1 < N_2$. При i-й передаче посылается действительное число x_i и принимается $y_{1i} = x_i + z_{1i}$ и $y_{2i} = x_i + z_{2i}$. В нашем рассмотрении несущественно, являются ли z_{1i} и z_{2i} коррелированными или нет (хотя при наличии обратной связи это может оказаться существенным). Предположим, что существует ограничение на передаваемую мощность, которое для любого n задается неравенством

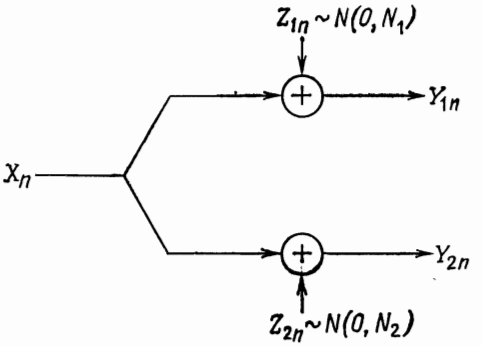


Рис. 16. Гауссовский широковещательный канал.

$$\frac{1}{n}\sum_{i=1}^{n} x_i^2 \leqslant S \tag{25}$$

для любого сигнала $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ с длиной блока n.

Хорошо известно, что отдельные пропускные способности равны $C_1 = {}^1/_2 \log(1 + S/N_1)$ и $C_2 = {}^1/_2 \log(1 + S/N_2)$ бит на посылку, где все логарифмы взяты по основанию 2.

При разделении времени достигается любая выпуклая комбинация (C_2, C_2) и $(C_1, 0)$, как показано на рис. 17.

Рассмотрим теперь, как можно улучшить эту характеристику. Пусть сигнал s_2 (предназначенный для приемника с высоким уровнем шума Y_2) представляет собой последовательность н. о. р. (нормальных) с. в. $N(0, \bar{\alpha}S)$. При передаче к этой последовательности прибавляется последовательность s_1, которую можно

рассматривать как последовательность н. о. р. с. в. $N(0, \alpha S)$. Здесь $0 \leqslant \alpha \leqslant 1$ и $\bar{\alpha} = 1 - \alpha$. Таким образом, последовательность $s = s_1 + s_2$ будет последовательностью н. о. р. с. в. $N(0, S)$. Принимаемые последовательности $y_1 = s_1 + s_2 + z_1$ и $y_2 = s_1 + s_2 + z_2$ изображены на рис. 18.

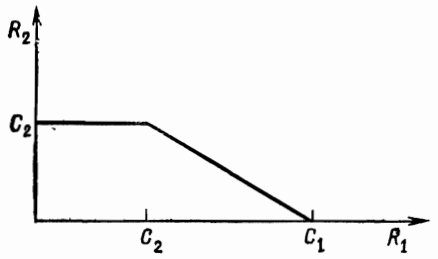


Рис. 17. Скорости при разделении времени в гауссовском широковещательном канале.

Рассмотрим теперь s_1 и z_2 в качестве шума для приемника 2. Заметим, что $s_{1i} + z_{2i}$ являются н. о. р. с. в. $N(0, \alpha S + N_2)$. Следовательно, сообщения можно посылать со скоростями, меньшими чем

$$\frac{1}{2}\log\left(1 + \frac{\bar{\alpha}S}{\alpha S + N_2}\right) = C_2(\alpha),$$

к приемнику Y_2 с вероятностью ошибки, близкой к нулю, при достаточно большой длине блока n. Это значит, что существует последовательность $(2^{n(C_2(\alpha)-\varepsilon)}, n)$-кодов со средней мощностью ограничений $\bar{\alpha}S$ и вероятностью ошибки $\bar{p}_2^{(n)}(e) \to 0$.

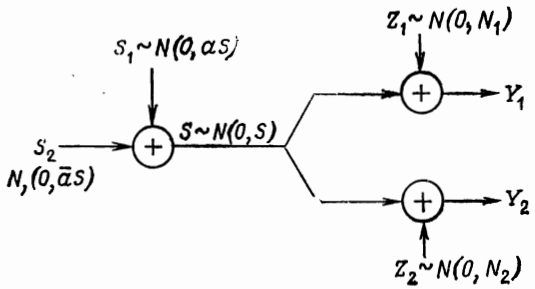


Рис. 18. Разложение сигнала.

Теперь так как $N_1 < N_2$, то приемник Y_1 также может верно определить переданную последовательность s_2 со сколь угодно малой вероятностью ошибки, декодируя s_2 при заданном y_1, приемник Y_1 затем вычитает s_2 из y_1, получая $y_1 = y_2 - s_2 = s_1 + z_1$. В этом месте канал 1 можно рассматривать как гауссовский канал с мощностью на входе, ограниченной αS, и аддитивным гауссовским шумом с нулевым средним и дисперсией N_1. Пропускная способность этого канала равна $\frac{1}{2}\log[1 + (\alpha S/N_1)] = C_1(\alpha)$ бит на посылку, и она достигается, грубо говоря, при выборе $2^{nC_1(\alpha)}$ независимых n-последовательностей н. о. р. с. в. $N(0, \alpha S)$ в качестве множества

возможных кодовых последовательностей s_1. Таким образом, приемник Y_1 верно принимает как s_1, так и s_2.

Эти неформальные рассуждения показывают, что одновременно возможно ε-достижение скоростей

$$R_1 = \frac{1}{2}\log\left(1 + \frac{\bar{\alpha}S}{\alpha S + N_2}\right) + \frac{1}{2}\log\left(1 + \frac{\alpha S}{N_1}\right),$$
$$R_2 = \frac{1}{2}\log\left(1 + \frac{\bar{\alpha}S}{\alpha S + N_2}\right) \qquad (26)$$

для любого $0 \leqslant \alpha \leqslant 1$. Эти пары скоростей изображены на рис. 19, и они охватывают скорости, получающиеся при разделении времени.

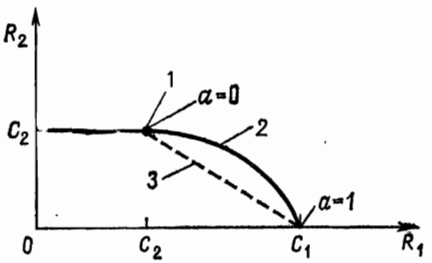


Рис. 19. Множество достижимых скоростей в гауссовском широковещательном канале; *1* — минимакс, *2* — суперпозиция, *3* — разделение времени.

Резюмируя рассмотрения, выберем множество $2^{n(C_2(\alpha)-\varepsilon)}$ случайных n-последовательностей н. о. р. с. в. $N(0, \alpha S)$ и множество $2^{n(\tilde{C}_1(\alpha)-\varepsilon)}$ случайных n-последовательностей н. о. р. с. в. $N(0, \bar{\alpha}S)$. После этого $2^{n(\tilde{C}_1(\alpha)+C_2(\alpha)-2\varepsilon)}$ n-последовательностей строятся с помощью сложения пар последовательностей, в которых первая последовательность выбирается из первого множества, а вторая последовательность — из второго множества, и эти пары выбираются всеми возможными способами. Сообщение

$$(r, s_1), \quad r \in \{1, 2, \ldots, 2^{n(C_2(\alpha)-\varepsilon)}\}, \quad s_1 \in \{1, 2, \ldots, 2^{n(\tilde{C}_1(\alpha)-\varepsilon)}\}$$

передается путем выбора n-последовательности, соответствующей сумме r-й последовательности из первого множества и s_1-й последовательности из второго множества. Приемник 1 должен декодировать верно (r, s_1), а приемник 2 должен декодировать верно r, и, таким образом, одновременно достигаются скорости

$$R_1 = \tilde{C}_1(\alpha) + C_2(\alpha) - 2\varepsilon,$$
$$R_2 = C_2(\alpha) - \varepsilon, \qquad (27)$$

что утверждалось в (26).

Исчерпывающее рассмотрение гауссовского канала увело бы нас в сторону. Непосредственное простое доказательство достижимости скоростей (27) было получено, но мы не будем приводить его здесь.

Закончим этот раздел одним замечанием. Если $N_1 = 0$, так что канал 1 идеальный, то имеем $C_1 = \infty$ и $C_2 = {}^1/_2 \log(1 + S/N_2)$.

Составной канал или максиминное приближение подразумевают передачу со скоростями $(R_1, R_2) = (C_2, C_2)$. Однако сколь угодно малое уменьшение в скорости для канала 2, соответствующее $0 < \alpha \ll 1$ в (26), приводит к $(R_1, R_2) = (\infty, C_2 - \varepsilon)$ как к паре достижимых скоростей. Хотя эта пара скоростей не мажорирует (C_2, C_2), она представляется более предпочтительной.

8. ВЕРХНЯЯ ГРАНИЦА ДЛЯ ДОСТИЖИМЫХ СКОРОСТЕЙ (R_1, R_2)

Предположим, что $p(x)$ (распределение вероятностей на X) порождает пару взаимных информаций $(I(X|Y_1), I(X|Y_2))$, где

$$I(X|Y_i) = \sum_{x \in X} \sum_{y \in Y_i} p(x)\, p_i(y|x) \log \frac{p_i(y|x)}{p_i(y)}, \qquad i = 1, 2. \tag{28}$$

Принимая во внимание свойства понятия взаимной информации, естественно предположить, что скорости $R_1 = I(X|Y_1)$, $R_2 = I(X|Y_2)$ одновременно достижимы. Оказывается, что это не так. (Контрпример можно получить при подробном рассмотрении примера двух ДСК из разд. 2 при $\Pr\{x = 1\} = 1/2$ и $I(X|Y_1) = 1$, $I(X|Y_2) = C_2$.) Однако множество совместно достижимых пар взаимных информаций с соответствующей модификацией, при которой принимается во внимание возможность разделения времени и отбрасывания информации, все-таки дает верхнюю границу $\mathfrak{R}$ для области пропускных способностей $\mathfrak{R}^*$. Эта верхняя граница в действительности достигается на ортогональном канале, канале с переключением разговора и на несовместных каналах.

Таким образом, мы подошли к тому, чтобы определить $\mathfrak{R}$ и установить $\mathfrak{R}$ в качестве верхней границы. Пусть

$$I = \{(I(X|Y_1), \quad I(X|Y_2)) \,|\, p(x) \geqslant 0, \quad \Sigma p(x) = 1\} \tag{29}$$

обозначает множество всех пар $(I(X|Y_1), I(X|Y_2))$, порождаемых $p(x)$ при вариации $p(\cdot)$ в симплексе возможных распределений вероятностей на X. Определим $\bar{I}$ как выпуклую оболочку I. Таким образом, $\bar{I}$ можно интерпретировать как среднюю совместную взаимную информацию, достижимую с помощью вариации $p(\cdot)$ во времени. Пусть

$$\mathfrak{R} = \{(R_1, R_2) \in E_2 \,|\, R_1 \leqslant I_1, \; R_2 \leqslant I_2, \\ \text{для некоторых } (I_1, I_2) \in \bar{I}\}. \tag{30}$$

Таким образом, $\mathfrak{R}$ интуитивно соответствует совместной взаимной информации, достижимой из $\bar{I}$ при отбрасывании информации. Эти множества изображены на рис. 20. Покажем теперь, что $\mathfrak{R}^* \subseteq \mathfrak{R}$.

Лемма 1. *При заданном произвольном* $((M_1, M_2, M_{12}), n)$*-коде для* n*-го расширения широковещательного канала, состоящем из слов* $\boldsymbol{x}(r, s_1, s_2) \in X^n$, $r \in R$, $s_1 \in S_1$, $s_2 \in S_2$, $|R| = M_{12}$, $|S_1| = M_1$, $|S_2| = M_2$, $M = M_{12}M_1M_2$, *пусть* (r, s_1, s_2) *будет случайной величиной, принимающей значения на множестве* $R \times S_1 \times S_2$. *Пусть* $(\boldsymbol{y}_1, \boldsymbol{y}_2) \in Y_1^n \times Y_2^n$ — *соответствующие случайные выходные* n*-последовательности, принимаемые приемниками* 1 *и* 2 *при передаче* $\boldsymbol{x}(r, s_1, s_2)$ *по каналу. Если* $p_1(e) = \Pr\{(\hat{r}_1, \hat{s}_1) \neq (r_1, s_1)\}$ *и* $p_2(e) =$

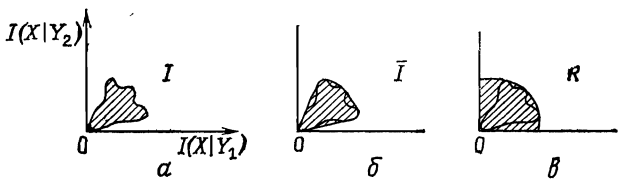


Рис. 20. Верхняя граница $\mathscr{R}$ для области пропускных способностей.

$= \Pr\{(\hat{r}_1, \hat{s}_2) \neq (r_1, s_2)\}$ — *вероятности ошибок при приеме этого кода, то*[1])

$$H(X|Y_1) \leqslant 1 + \log M_2 + p_1(e) \log M_{12}M_1, \tag{31}$$

$$H(X|Y_2) \leqslant 1 + \log M_1 + p_2(e) \log M_{12}M_2. \tag{32}$$

Доказательство. Пусть правила декодирования, соответствующие нашему коду, будут

$$\begin{aligned} g_1&: Y_1^n \to R \times S_1, \\ g_2&: Y_2^n \to R \times S_2, \end{aligned} \tag{33}$$

которые мы запишем в виде

$$g_k(\boldsymbol{y}_k) = (g_{k1}(\boldsymbol{y}_k), g_{k2}(\boldsymbol{y}_k)), \qquad k = 1, 2.$$

Таким образом, при заданном случайном сообщении (r, s_1, s_2) и последовательности $\boldsymbol{y}_k \in Y_k^n$ приемник k будет совершать ошибку тогда и только тогда, когда

$$\begin{aligned} g_1(\boldsymbol{y}_1) &\neq (r, s_1), \qquad k = 1, \\ g_2(\boldsymbol{y}_2) &\neq (r, s_2), \qquad k = 2. \end{aligned} \tag{34}$$

Таким образом,

$$\begin{aligned} p_1(e) &= \Pr\{g_1(\boldsymbol{y}_1) \neq (r, s_1)\}, \\ p_2(e) &= \Pr\{g_2(\boldsymbol{y}_2) \neq (r, s_2)\}. \end{aligned} \tag{35}$$

---

[1]) Здесь $H(X|Y_i)$ — условная энтропия, $H(X|Y_i) = -\sum\limits_{x \in X} \sum\limits_{y \in Y_i} p_i(x, y) \times$ $\times \log p_i(x|y)$. — *Прим. ред.*

Заметим, что

$$H(X\,|\,\boldsymbol{y}_1) \leqslant H(p_1(e\,|\,\boldsymbol{y}_1),\ 1-p_1(e\,|\,\boldsymbol{y}_1)) + \\ + (1-p_1(e\,|\,\boldsymbol{y}_1))\log M_2 + p_1(e\,|\,\boldsymbol{y}_1)\log(M-M_2), \tag{36}$$

где были использованы неравенство

$$H(a_1,\ a_2,\ \ldots,\ a_m) \leqslant \log m \tag{37}$$

и основное композиционное соотношение (см. Эш [1, стр. 8]). Конечно, считаются заданными события $g_1(\boldsymbol{y}_1) = (r, s_1)$ и $g_1(\boldsymbol{y}_1) \neq (r, s_1)$. Взяв математическое ожидание по Y_1^n и используя выпуклость $H(p, 1-p)$ по p, получаем

$$H(X\,|\,Y_1) \leqslant H(p_1(e),\ 1-p_1(e)) + (1-p_1(e))\log M_2 + \\ + p_1(e)\log(M-M_2). \tag{38}$$

Наконец, так как $H(p,\ 1-p) \leqslant 1$ и $M = M_{12}M_1M_2$, имеем

$$H(X\,|\,Y_1) \leqslant 1 + \log M_2 + p_1(e)\log(M-M_2)/M_2 \leqslant \\ \leqslant 1 + \log M_2 + p_1(e)\log M_{12}M_1. \tag{39}$$

Соответствующие рассмотрения для $H(X\,|\,Y_2)$ завершают доказательство.

Нам понадобится следующая лемма (см. Эш [1, стр. 81]).

Лемма 2. *Пусть $X_1, \ldots, X_n$ — последовательность случайных величин на входе широковещательного (дискретного) канала (без памяти), а $Y_{11}, \ldots, Y_{1n}$, $Y_{21}, \ldots, Y_{2n}$ — соответствующие принимаемые случайные величины на выходе каналов 1 и 2 соответственно. Тогда*

$$I(X_1, \ldots, X_n\,|\,Y_{k1}, \ldots, Y_{kn}) \leqslant \sum_{i=1}^{n} I(X_i\,|\,Y_{ki}), \quad k = 1,\ 2, \tag{40}$$

где равенство имеет место тогда и только тогда, когда $Y_{k1}, Y_{k2}, \ldots, Y_{kn}$ независимы.

Доказательство. Имеем

$$H(Y_{k1}, Y_{k2}, \ldots, Y_{kn}\,|\,X_1, \ldots, X_n) = -\sum p_k(\boldsymbol{x},\ \boldsymbol{y}_k)\log p_k(\boldsymbol{y}_k\,|\,\boldsymbol{x}),$$

но в силу того, что канал k является каналом без памяти, $p_k(\boldsymbol{y}_k\,|\,\boldsymbol{x})$ представляется в виде произведения $\Pi p_k(y_{ki}\,|\,x_i)$, давая

$$H(Y_{k1}, \ldots, Y_{kn}\,|\,X_1, \ldots, X_n) = \\ = \sum_{\boldsymbol{x},\ \boldsymbol{y}_k} p_k(\boldsymbol{x},\ \boldsymbol{y}_k) \sum_{i=1}^{n} \log p_k(y_{ki}\,|\,x_i) = \sum_{i=1}^{n} H(Y_{ki}\,|\,X_i).$$

Используя теперь известное неравенство, получаем

$$H(Y_{k1}, \ldots, Y_{kn}) \leqslant \sum_{i=1}^{n} H(Y_{ki}),$$

где равенство имеет место тогда и только тогда, когда Y_{ki} независимы при $i = 1, 2, \ldots, n$. Так как $I(\boldsymbol{X}|\boldsymbol{Y}_k) = H(\boldsymbol{Y}_k) - H(\boldsymbol{Y}_k|\boldsymbol{X})$, то лемма доказана.

Теперь мы хотим показать, что $\bar{p}_1^{(n)}(e)$, $\bar{p}_2^{(n)}(e)$ не могут одновременно стремиться к нулю для скоростей $(R_1, R_2) \notin \mathfrak{R}$. Это дает нам $\mathfrak{R}$ в качестве верхней границы области пропускных способностей широковещательного канала.

Пусть $R_1 = (1/n)\log M_1 M_{12}$ и $R_2 = (1/n)\log M_2 M_{12}$ — скорости передачи в битах на посылку для приемников Y_1 и Y_2 соответственно. (Напомним, что $R_{12} = \log M_{12}$ является скоростью передачи информации, общей для обоих каналов.) Доказательство очень напоминает то, которое было использовано Шенноном [4] для двухпутевого канала.

Теорема. *Для любой последовательности* $[(2^{nR_1}, 2^{nR_2}, 2^{nR_{12}}), n]$-*кодов при* $(R_1, R_2) \notin \mathfrak{R}$ *имеем*

$$(\bar{p}_1^{(n)}(e), \bar{p}_2^{(n)}(e)) \not\to (0, 0), \quad (\lambda_1^{(n)}, \lambda_2^{(n)}) \to (0, 0) \quad n \to \infty.$$

Таким образом, $\mathfrak{R}$ *является верхней границей области пропускных способностей широковещательного канала.*

Доказательство. При заданном произвольном $[(M_1, M_2, M_{12}), n]$-коде для n-го расширения широковещательного канала выберем кодовое слово $\boldsymbol{x}(r, s_1, s_2)$ случайным образом в соответствии с равномерным распределением $\Pr\{r, s_1, s_2\} = 1/M$, $(r, s_1, s_2) \in R \times S_1 \times S_2$, где $M = |R||S_1||S_2|$. Если кодовые слова $\boldsymbol{x}(r, s_1, s_2) \in X^n$ не являются различными, то для доказательства теоремы нужно лишь незначительно изменить приведенные ниже рассуждения. Поэтому, рассматривая случай, когда $\boldsymbol{x}(r, s_1, s_2)$ различны, получаем $H(X) = \log M$ и $I(X|Y_1) = \log M - H(X|Y_1)$ при условии заданного равномерного распределения на кодовых словах. Как и в разд. 3, пусть $\bar{p}_1^{(n)}(e)$ и $\bar{p}_2^{(n)}(e)$ обозначают вероятности ошибок для кода при этом распределении. Согласно лемме 2,

$$I(X|Y_1) \leqslant \sum_{i=1}^{n} I(X_i|Y_{1i}). \tag{41}$$

Таким образом,

$$I(X|Y_1) = \log M - H(X|Y_1) \leqslant \sum_{i=1}^{n} I(X_i|Y_{1i}). \tag{42}$$

Наконец, так как неравенство (31) в лемме 1 справедливо для любого распределения на кодовых словах, то, используя (31) и

(42), получаем

$$\log M - 1 - \log M_2 - \bar{p}_1^{(n)}(e) \log M_{12} M_1 \leqslant \sum_{i=1}^{n} I(X_i \mid Y_{1i}), \qquad (43)$$

что дает основное неравенство

$$R_1 = \frac{1}{n} \log M_{12} M_1 \leqslant \frac{(1/n) + (1/n) \sum_{i=1}^{n} I(X_i \mid Y_{1i})}{1 - \bar{p}_1^{(n)}(e)}. \qquad (44а)$$

Аналогично находим, что

$$R_2 = \frac{1}{n} \log M_{12} M_2 \leqslant \frac{(1/n) + (1/n) \sum_{i=1}^{n} I(X_i \mid Y_{2i})}{1 - \bar{p}_2^{(n)}(e)}. \qquad (44б)$$

Подводя итоги, получаем, что любой код для n-го расширения широковещательного канала должен иметь скорости (R_1, R_2),

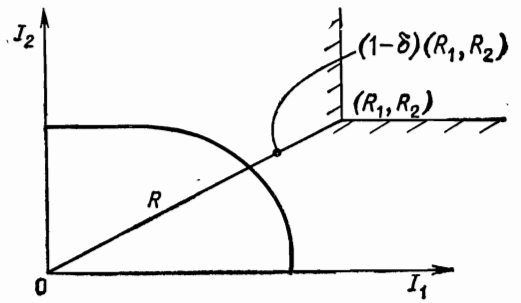


Рис. 21. Недостижимая пара скоростей (R_1, R_2); заштрихованная область: $\{(I_1, I_2) \mid I_1 < R_2$ или $I_2 < R_2\}$.

удовлетворяющие (44а) и (44б), где

$$\bar{p}_i^{(n)}(e) = \frac{1}{M} \sum_{r,\, s_1,\, s_2} \lambda_i(r,\, s_1,\, s_2), \quad i = 1,\, 2. \qquad (45)$$

Предположим теперь, что $(R_1, R_2) \notin \mathfrak{R}$, $R_1 \geqslant 0$, $R_2 \geqslant 0$, как изображено на рис. 21. Покажем, что $\bar{p}_i^{(n)}(e)$, $i = 1, 2$, не могут быть одновременно малы. В силу выпуклости $\mathfrak{R}$ и того, что $I \subseteq \mathfrak{R}$, получаем

$$\left(\frac{1}{n} \sum_{i=1}^{n} I(X \mid Y_{1i}),\ \frac{1}{n} \sum_{i=1}^{n} I(X \mid Y_{2i}) \right) \in \mathfrak{R}$$

для всех $p(x)$. Следовательно, как показано на рис. 21, либо

$$\frac{1}{n} \sum_{i=1}^{n} I(X \mid Y_{1i}) < R_1(1 - \delta), \qquad (46а)$$

либо

$$\frac{1}{n}\sum_{i=1}^{n} I(X\,|\,Y_{2i}) < R_2(1-\delta), \tag{46б}$$

где $\delta > 0$ — любое неотрицательное действительное число, такое, что $(1-\delta)(R_1, R_2) \notin \mathfrak{R}$.

Но (44) при $i=1, 2$ означает, что

$$\bar{p}_i^{(n)}(e) \geqslant 1 - \frac{1}{nR_i} - \frac{(1/n)\sum_{j=1}^{n} I(X\,|\,Y_{ij})}{R_i}. \tag{47}$$

Второй член правой части в (47) стремится к нулю с ростом n, а третий должен быть меньше, чем $(1-\delta)$, либо при $i=1$, либо при $i=2$, либо в обоих случаях. Поэтому

$$\lim_{n\to\infty} \max\{\bar{p}_1^{(n)}(e),\ \bar{p}_2^{(n)}(e)\} \geqslant \delta > 0, \tag{48}$$

и, следовательно, $\bar{p}_1^{(n)}(e)$ и $\bar{p}_2^{(n)}(e)$ не могут одновременно быть близкими к нулю. Точно так же из того, что вероятность ошибки для наихудшего кодового слова в каждом канале подчиняется неравенству $\lambda_i^{(n)} \geqslant \bar{p}_i^{(n)}(e)$, $i=1, 2$, можно заключить, что если $(R_1, R_2) \notin \mathfrak{R}$, то не существует последовательности $((2^{nR_1}, 2^{nR_2}, 2^{nR_{12}}), n)$-кодов для широковещательного канала, такой, что $(\lambda_1^{(n)}, \lambda_2^{(n)}) \to (0, 0)$.

9. СОСТАВНЫЕ КАНАЛЫ

Пусть $P_\beta(y|x)$, $\beta \in \mathfrak{B}$, представляет собой возможно бесконечную совокупность переходных вероятностей для канала. Индекс β выбирается природой, и последовательность n посылок $x_1, x_2, \ldots, x_n$ передается приемнику по дискретному каналу без памяти $P_\beta(y|x)$. Индекс β неизвестен посылающему, но может (без потери общности) предполагаться известным приемнику. (Простая передача $\sqrt{n}$ заранее указанных символов среди n посылок позволит приемнику определить β со сколь угодно малой вероятностью ошибки при конечном $\mathfrak{B}$, и это не повлияет на величину достижимой скорости R.) Вольфовиц [2] и Блекуэлл и др. [5] определили пропускную способность C составного канала как

$$C = C_{\max\min} = \sup_{p(x)} \inf_{\beta} I_\beta(X\,|\,Y). \tag{49}$$

Эта скорость C достигается при конечном $\mathfrak{B}$ с помощью использования кода для канала β^*, такого, что

$$C = \max_{p(x)} I_{\beta^*}(X\,|\,Y). \tag{50}$$

Максимальная скорость C достигается тогда независимо от значения индекса β.

Рассмотрим теперь линию связи, в которой неизвестно, будет ли эта линия бесшумным двоичным симметричным каналом или двоичным симметричным каналом с параметром p. Это значит, что описание канала $P_\beta(y|x)$, $\beta = 1, 2$, задается в виде

$$P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} \bar{p} & p \\ p & \bar{p} \end{bmatrix}. \tag{51}$$

Для этого составного канала находим, что

$$C = 1 - H(p). \tag{52}$$

С точки зрения настоящей работы мы определили множество $\mathfrak{R}^*$ всех достижимых пар скоростей (R_1, R_2) для обоих заданных каналов (см. рис. 22). Это дает полный спектр достижимых скоростей при различных случайных выборах, осуществляемых природой.

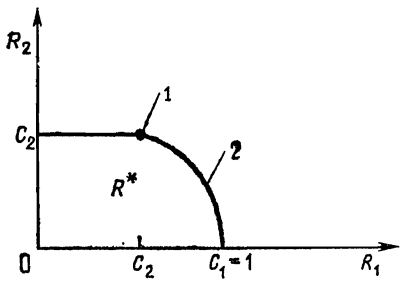


Рис. 22. Множество достижимых скоростей для составного канала; *1* — минимаксная скорость, *2* — другие допустимые скорости.

Таким образом, например, если известно, что

$$\Pr\{\beta = 1\} = \pi_1 = 1 - \Pr\{\beta = 2\}, \tag{53}$$

то можно найти максимальную ожидаемую скорость

$$R(\pi_1) = \max_{(R_1, R_2) \in \mathfrak{R}^*} (\pi_1 R_1 + \pi_2 R_2). \tag{54}$$

Интерпретация ее состоит в том, что при использовании наложенных кодов из разд. 2 можно достичь средних скоростей

$$R(\pi_1) = \max_{0 \leqslant \alpha \leqslant 1} [C(\alpha\bar{p} + \bar{\alpha}p) + \pi_1 H(\alpha)], \tag{55}$$

соответствующих точкам на границе $\mathfrak{R}^*$. Эти средние скорости строго больше, чем средние скорости, достижимые при разделении времени (кроме случая вырожденного распределения $\pi_1 = 0$ или 1). Наконец, часть сообщения со скоростью $C(\alpha\bar{p} + \bar{\alpha}p)$ будет уверенно принята независимо от того, какой канал был выбран природой.

Эти рассмотрения предполагают, что задачи, связанные с составными каналами, могут быть исследованы также с точки зрения широковещательных каналов, если интерпретировать распределение вероятностей по параметру β как распределение вероятности по номеру приемника, выбираемого в задаче широковещательного канала со многими приемниками. При этом рассмотре-

ние области пропускных способностей $\mathfrak{R}^*$ даст все достижимые распределения вероятностей на скоростях для составного канала. После этого можно выбрать самое лучшее распределение.

10. ЗАКЛЮЧЕНИЕ

Пусть, как и ранее, областью пропускных способностей $\mathfrak{R}^*$ будет множество всех совместно достижимых скоростей (R_1, R_2) для заданного широковещательного канала с двумя приемниками. Теперь нам известно следующее. Имеется некоторая теоретически определенная область $\mathfrak{R}$, порождаемая $(I(X|Y_1)\ I(X|Y_2))$ и приведенная в разд. 8, которая дает верхнюю границу для $\mathfrak{R}^*$. Кроме того, с помощью простого разделения времени строится внутренняя граница, скажем, $\mathfrak{R}_0$ для $\mathfrak{R}^*$, как изображено на рис. 23.

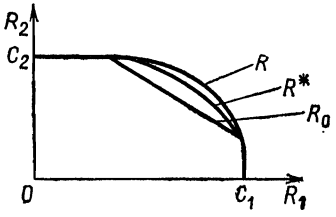


Рис. 23. Границы для области пропускных способностей $\mathfrak{R}^*$.

Иногда эти границы совпадают, как это имело место в случае несовместных каналов. При этом $\mathfrak{R} = \mathfrak{R}_0 = \mathfrak{R}^*$. В других примерах, таких, как ортогональные каналы, в которых границы не совпадают, простыми рассуждениями было показано, что верхняя граница может быть достигнута и, следовательно, что $\mathfrak{R} = \mathfrak{R}^*$. Во многих промежуточных случаях (например, в случае двух ДСК в разд. 2) у нас имеются разумные основания считать, что наши специальные коды достигают границы $\mathfrak{R}^*$, хотя доказательства обращений кажутся трудными.

Рассмотрение этой задачи стало целесообразным из-за того, что почти всегда с помощью соответствующим образом выбранного способа кодирования можно достичь скоростей $\mathfrak{R}^*$, строго больших тех, которые достигаются при простом разделении времени.

Главное соображение, к которому приводят эти исследования, состоит в том, что высокие совместные скорости передачи лучше всего достигаются при наложении высокоскоростной и низкоскоростной информаций, а не при использовании разделения времени. Рассказы, содержащие целый ряд уровней символизма, служат одним из примеров способа передачи, который допускает множество различных уровней восприятия со стороны различных людей.

Мне хочется поблагодарить Д. Сагаловича и К. Кейлерса за многочисленные полезные обсуждения этой статьи. Д. Сагалович помог также улучшить доказательство верхней границы, а К. Кейлерс оказал помощь при составлении некоторых примеров. Я также получил пользу от обсуждений с П. Бергмансом и А. Д. Вайнером.

ПРИЛОЖЕНИЕ

Здесь мы докажем главный результат из разд. 2. Пусть $C(p) = 1 - H(p)$.

Теорема. *В широковещательном канале из разд. 2 с ДСК, параметры которых* $p_1 = 0$ *и* $p_2 = p$, *соответственно достигаются скорости* $(R_1, R_2) = (C(\alpha\bar{p} + \bar{\alpha}p) + H(\alpha),\ C(\alpha\bar{p} + \bar{\alpha}p))$ *при любом* $0 \leqslant \alpha \leqslant 1$.

Доказательство. Пусть $M_{12} = 2^{nR_{12}}$ и $M_1 = 2^{n(R_1 - R_{12})}$ — целые числа, и пусть $R_2 = R_{12}$, $M_2 = 2^{nR_{12}}$. Рассмотрим следующий случайный код. Пусть $\boldsymbol{x}(r)$, $r \in R = \{1, 2, \ldots, M_{12}\}$ будут n-последовательностями н. о. р. в множестве $X^n = \{0, 1\}^n$, где $\boldsymbol{x}(r)$ выбираются в соответствии с равномерным распределением на X^n. Пусть при $\alpha < 1/2$ величина αn будет целым числом, и пусть $\boldsymbol{z}(s)$, $s \in S = \{1, 2, \ldots, M_1\}$ представляет собой нумерацию всех n-последовательностей $\boldsymbol{z} \in \{0, 1\}^n$, таких, что

$$\sum_{i=1}^{n} z_i = \alpha n.$$

Всего имеется

$$C_n^{\alpha n} = 2^{n\{H(\alpha) + O[(\ln n)/n]\}}$$

таких последовательностей. Положим $\boldsymbol{x}(r, s) = \boldsymbol{x}(r) \oplus \boldsymbol{z}(s)$, где сложение векторов производится покомпонентно по модулю 2. Без потери общности положим $p < 1/2$.

Правило декодирования g_1: $Y_1^n \to R \times S$ для n-го расширения для приемника 1 состоит в выборе значений $\hat{r} \in R$, $\hat{s} \in S$, таких, что $\boldsymbol{y}_1 = \boldsymbol{x}(\hat{r}, \hat{s})$. Будем говорить, что произошла ошибка, если имеется более чем одно значение пары $(\hat{r}, \hat{s})$, для которой это справедливо. (Так как канал 1 бесшумный, то событие, состоящее в том, что такая пара $(\hat{r}, \hat{s})$ не существует, не имеет места).

По правилу декодирования g_2: $Y_2^n \to R$ для канала 2 решение выносится в пользу значения $\hat{r} \in R$, такого, что $d(\boldsymbol{y}_2, \boldsymbol{x}(\hat{r})) \leqslant n(\alpha\bar{p} + \bar{\alpha}p) + n\varepsilon$ при заданном $\varepsilon > 0$, где d — хэмминговское расстояние. Ошибка в канале 2 будет происходить, если имеются более чем одно или если нет ни одного такого значения $\hat{r} \in R$.

Выберем теперь сообщение (r, s) с вероятностью $1/M_1M_{12}$ и найдем математическое ожидание суммы вероятностей ошибок $E\{\bar{p}_1(e) + \bar{p}_2(e)\}$ (см. (9), (10)), где математическое ожидание берется по случайному коду, определенному выше.

В силу того, что канал 1 является идеальным (т. е. $\boldsymbol{y}_1 = \boldsymbol{x}(r, s)$), ошибка при декодировании в канале 1 возникает, только если сам

(случайный) код приписывает некоторый другой номер (r', s') той n-последовательности, которая имеет номер (r, s).

В силу симметрии процесса построения кода можно рассматривать лишь передачу $x(1, 1)$. Таким образом,

$$E\bar{p}_1(e) = \Pr\{x(r, s) = x(1, 1) \text{ для некоторой пары } (r, s) \neq (1, 1)\}, \tag{57}$$

где вероятность определена на кодах при случайном выборе.

Теперь $x(1, 1) = x(1, s)$ означает, что $z(1) = z(s)$, что невозможно при любом $s \neq 1$ в силу построения $z(s)$. Таким образом, единственной возможностью ошибки является условие $x(1, 1) = x(r, s)$, $r \neq 1$, $s \in S$. Но $r \neq 1$ означает, что $x(r, s)$ и $x(1, 1)$ являются независимыми равномерно распределенными n-последовательностями в $\{0, 1\}^n$. Таким образом, при $r \neq 1$ имеем

$$\Pr\{x(1, 1) = x(r, s)\} = 2^{-n}. \tag{58}$$

Используя это вместе с границей для объединения событий, получаем

$$E\bar{p}_1(e) \leqslant \sum_{(r, s) \neq (1, 1)} \Pr\{x(1, 1) = x(r, s)\} \leqslant$$

$$\leqslant M_1 M_{12} 2^{-n} = 2^{-n(1-R_1)} \to 0, \quad R_1 < 1. \tag{59}$$

Таким образом, $E\{\bar{p}_1(e)\} \to 0$ при $n \to \infty$, если $R_1 < 1$, и в этом случае из конструкции следует, что

$$R_1 - R_{12} = (\log M_1)/n = H(\alpha) - O[(\ln n)/n]. \tag{60}$$

Рассмотрим теперь канал 2. Пусть $e = (e_1, e_2, \ldots, e_n)$—двоичный n-вектор н. о. р. бернуллиевых с. в. с параметром p. Таким образом, можно записать $y_2 = x(r, s) \oplus e$ и

$$y_2 = x(r) \oplus z(s) \oplus e. \tag{61}$$

Ошибка при декодировании может произойти в одном из двух случаев. E_1: верное значение $r = 1$ не удовлетворяет неравенству

$$d(y_2, x(1)) \leqslant n(\alpha\bar{p} + \bar{\alpha}p) + n\varepsilon, \tag{62}$$

и E_2: существует номер $r \neq 1$, $r \in R$, такой, что

$$d(y_2, x(r)) \leqslant n(\alpha\bar{p} + \bar{\alpha}p) + n\varepsilon.$$

Таким образом,

$$E\{\bar{p}_2(e)\} \leqslant \Pr\{E_1\} + \Pr\{E_2\}, \tag{63}$$

где вероятности учитывают случайность выбора кода и случайность выбора (r, s). Из (61) имеем

$$\Pr\{E_1\} = \Pr\{d(\boldsymbol{y}_2, \boldsymbol{x}(1)) > n(\alpha\bar{p} + \bar{\alpha}p + \varepsilon)\} = \\ = \Pr\left\{\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}(s)_i \oplus \boldsymbol{e}_i > \alpha\bar{p} + \bar{\alpha}p + \varepsilon\right\}. \tag{64}$$

Найдем математическое ожидание (по $\boldsymbol{e}$ и s)

$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}(s)_i \oplus \boldsymbol{e}_i = \frac{1}{n}\sum_{i=1}^{n} \Pr\{(\boldsymbol{z}(s)_i, \boldsymbol{e}_i) = (1, 0) \text{ или } (0, 1)\} \\ = \frac{1}{n}\sum_{i=1}^{n} (\alpha\bar{p} + \bar{\alpha}p) = \alpha\bar{p} + \bar{\alpha}p. \tag{65}$$

После некоторых вычислений имеем также

$$\operatorname{var} \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}(s)_i \oplus \boldsymbol{e}_i \leqslant \frac{p\bar{p}}{n}. \tag{66}$$

Это означает, что $d(\boldsymbol{y}, \boldsymbol{x}(r)) \to \alpha\bar{p} + \bar{\alpha}p$ по вероятности, и, следовательно, $\Pr\{E_1\} \to 0$ при $n \to \infty$.

Осталось оценить $\Pr\{E_2\}$. Запишем

$$\Pr\{E_2\} \leqslant \Pr\{d(\boldsymbol{x}(r), \boldsymbol{y}_2) \leqslant n(\alpha\bar{p} + \bar{\alpha}p + \varepsilon), \\ \text{для некоторого } r \neq 1 \mid \text{передано } \boldsymbol{x}(1)\} \leqslant \\ \leqslant 2^{nR_{12}} \Pr\{d(\boldsymbol{x}(2), \boldsymbol{y}_2) \leqslant n(\alpha\bar{p} + \bar{\alpha}p + \varepsilon)\}. \tag{67}$$

Но

$$d(\boldsymbol{x}(2), \boldsymbol{y}_2) = wt(\boldsymbol{x}(2) \oplus \boldsymbol{x}(1) \oplus \boldsymbol{z}(s) \oplus \boldsymbol{e}), \tag{68}$$

где wt обозначает число единиц в двоичной n-последовательности, а $\boldsymbol{x}(2)$ и $\boldsymbol{x}(1)$ являются независимыми бернуллиевыми n-последовательностями с параметром 1/2. Таким образом, для любого $\varepsilon > 0$ имеем

$$\Pr\{E_2\} \leqslant 2^{nR_{12}} 2^{n(H(\alpha\bar{p}+\bar{\alpha}p)+O[(\ln n)/n]+\varepsilon')} 2^{-n}, \tag{69}$$

где $2^{n(H(\alpha\bar{p}+\bar{\alpha}p)+O[(\ln n)/n]+\varepsilon')}$ обозначает число точек

$$\sum_{i=0}^{n(\alpha\bar{p}+\bar{\alpha}p+\varepsilon')} C_n^i$$

в декодирующей сфере вокруг $\boldsymbol{y}_2$. Следовательно, если

$$R_{12} < 1 - H(\alpha\bar{p} + \bar{\alpha}p) - \varepsilon', \tag{70}$$

то $\Pr\{E_2\} \to 0$ при $n \to \infty$. Используя (60) и (70), получаем, что если

$$R_2 = R_{12} < 1 - H(\alpha\bar{p} + \bar{\alpha}p), \tag{71}$$

$$R_1 < H(\alpha) + R_2 = 1 - H(\alpha\bar{p} + \bar{\alpha}p) + H(\alpha),$$

то

$$E\{\bar{p}_1^{(n)}(e) + \bar{p}_2^{(n)}(e)\} = E\{\bar{p}_1^{(n)}(e)\} + E\{\bar{p}_2^{(n)}(e)\} \to 0. \tag{72}$$

Так как наилучший код ведет себя лучше, чем средний, то существует последовательность $[(2^{nR_1}, 2^{nR_2}, 2^{nR_{12}}), n]$-кодов, $n = 1, 2, \ldots$, со скоростями

$$R_1 = C(\alpha\bar{p} + \bar{\alpha}p) + H(\alpha) - \varepsilon,$$

$$R_2 = C(\alpha\bar{p} + \bar{\alpha}p) - \varepsilon, \tag{73}$$

такая, что

$$\bar{p}_1^{(n)}(e) + \bar{p}_2^{(n)}(e) \to 0, \tag{74}$$

и, таким образом, $\bar{p}_1^{(n)}(e) \to 0$, $\bar{p}_2^{(n)}(e) \to 0$.

Вычислив предел (R_1, R_2) при $\varepsilon \to 0$, мы завершим доказательство теоремы.

СПИСОК ЛИТЕРАТУРЫ

1. Ash R. B., Information Theory, New York, Interscience, 1965.
2. Wolfowitz J., Coding Theorems of Information Theory, 2nd ed., Berlin, Springer, and Englewood Cliffs, N. J., Prentice Hall, 1964.
3. Shannon C. E., A mathematical theory of communication, *Bell. Syst. Tech. J.*, **27** (1948), 379—423 (pt. 1), 623—656 (pt. 2); Urbana, Ill.; Univ. Illinois Press, 1949. [1])
4. Shannon C. E., Two-way communication channels, in Proc. 4th Berkeley Symp. Probability and Statistics, I, Berkeley Calif., Univ. California Press, 1961, 611—644. [1])
5. Blackwell D., Breiman L., Thomasian A., The capacity of a class of channels, *Ann. Math. Statist.*, **30** (1959), 1229—1241.
6. Shannon C. E., A note on a partial ordering for communication channels, *Inform. Contr.*, **1** (1958), 390—397. [1])

[1]) Имеется русский перевод, см. Шеннон К., Работы по теории информации и кибернетике, «ИЛ», М., 1963. — *Прим. ред.*

Верхняя граница числа K для последовательностных машин без потери информации порядка K[1)]

Яхико Камбаяси и Сузо Ядзима

1. ВВЕДЕНИЕ

Последовательностная машина называется машиной без потери информации порядка k, если k — наименьшее целое число, такое, что начальное состояние и первые k выходов однозначно определяют начальный вход. Необходимое и достаточное условие, при котором последовательностная машина является машиной без потери информации порядка k, дано Ивеном [1]. Используя его результат, можно показать, что для машин с n состояниями $k \leqslant (1/2)n(n-1)+1$, однако машина, для которой эта оценка достигается, не была получена. В настоящей статье будет показано, что для любого n существует последовательностная машина с n состояниями, для которой верхняя граница достигается.

Последовательностная машина типа Мили может быть определена как система, состоящая из S, X, Y, δ и λ, т. е. $M = (S, X, Y, \delta, \lambda)$, где $S = \{s_1, s_2, \ldots, s_n\}$ — конечное множество состояний, $X = \{x_1, x_2, \ldots, x_p\}$ — конечное множество входных символов, $Y = \{y_1, y_2, \ldots, y_q\}$ — конечное множество выходных символов, δ — функция переходов и λ — функция выходов, удовлетворяющие следующим равенствам:

$$s(t+1) = \delta(s(t), x(t)), \tag{1}$$

$$y(t) = \lambda(s(t), x(t)), \tag{2}$$

где $s(t)$, $x(t)$ и $y(t)$ — состояние, вход и выход в момент времени t (t принимает целочисленные значения).

Определение 1 (Хаффмэн [2]). Последовательностная машина называется *машиной без потери информации порядка k*, если k — наименьшее целое число, такое, что начальное состояние и первые k выходов однозначно определяют начальный вход, т. е.

$$x(t-k) = g(y(t-1), \ldots, y(t-k), s(t-k)). \tag{3}$$

Если последовательное соединение машин M и M' соответствует линии задержки длительности h, то M' называется квазиобратной для M с задержкой h. Для любой машины M без по-

---

[1)] Yahiko Kambayashi and Shuzo Yajima, The upper bound of K in K-lossless sequential machines, *Information and Control*, 19, № 5 (1971), 432—438.

тери информации порядка k существует квазиобратная машина M' с задержкой $h \geqslant k-1$. Квазиобратная машина с задержкой 0 называется обратной машиной.

Последовательностная машина называется машиной без потери информации, если каждый раз входная последовательность определяется начальным состоянием, заключительным состоянием и выходной последовательностью. Очевидно, что машина без потери информации порядка k является машиной без потери информации. Однако известно (Хаффмэн [2]), что существуют последовательностные машины без потери информации, не имеющие квазиобратных последовательностных машин.

2. ВЕРХНЯЯ И НИЖНЯЯ ГРАНИЦЫ ЧИСЛА k ДЛЯ МАШИН БЕЗ ПОТЕРИ ИНФОРМАЦИИ ПОРЯДКА k

В этом разделе обсуждаются верхняя и нижняя границы числа k для машин с n состояниями без потери информации порядка k.

Определение 2. Граф пар $G_0(M)$ последовательностной машины M состоит из следующих частей:

(1) Вершина для каждой неупорядоченной пары $(s_i, s_j) = (s_j, s_i)$.

(2) Если $\delta(s_i, x) = s_h$, $\delta(s_j, x') = s_h$, $\lambda(s_i, x) = y$ и $\lambda(s_j, x') = y$, где x и x' могут и совпадать, то существует дуга из вершины (s_i, s_j) в вершину (s_h, s_h).

Обозначим через $G(M)$ множество вершин (s_i, s_j) графа $G_0(M)$, таких, что существуют $s_h \in S$ и $x, x' \in X$, $x \neq x'$, для которых $s_i = \delta(s_h, x)$, $s_i = \delta(s_h, x)$, $s_j = \delta(s_h, x')$, $\lambda(s_h, x) = \lambda(s_h, x')$.

Теорема 3 (Ивен [1]). *Последовательностная машина M является машиной без потери информации тогда и только тогда, когда граф $\tilde{G}_0(M)$, полученный из $G_0(M)$ отбрасыванием всех вершин, недостижимых ни из какой вершины множества $G(M)$, не содержит вершин-пар одинаковых состояний, т. е. вершин вида (s_i, s_i).*

Теорема 4 (Ивен [1]). *Последовательностная машина M является машиной без потери информации порядка k тогда и только тогда, когда выполняются следующие условия:*

(1) *M является машиной без потери информации.*

(2) *Граф $\tilde{G}_0(M)$ не содержит контуров, и путь наибольшей длины в нем содержит $k-1$ вершин.*

Так как для любой последовательностной машины M с n состояниями граф $G_0(M)$ содержит $(1/2)n(n-1)$ вершин[1]), то справедливо

[1]) Авторы имеют в виду, что граф $\tilde{G}_0(M)$ для машины без потери информации содержит не более $(1/2)n(n-1)$ вершин. — *Прим. перев.*

Следствие 5. *Если последовательностная машина с n состояниями является машиной без потери информации порядка k, то k удовлетворяет неравенству*

$$1 \leqslant k \leqslant (1/2)\, n\,(n-1) + 1. \qquad (4)$$

Путем построения соответствующих машин можно показать, что эти границы являются строгими.

Теорема 6. *Для любого n существует последовательностная машина с n состояниями без потери информации порядка* $(1/2)\,n\,(n-1)+1$ *с 3 входными и 6 выходными символами*[1]).

Таблица 1

Вход	0	1	2
1	2, 0	3, 1	2, 4
2	3, 0	4, 1	3, 4
.	. .	. .	. .
.	. .	. .	. .
.	. .	. .	. .
i	$i+1$, 0	$i+2$, 1	$i+1$, 4
.	. .	. .	. .
.	. .	. .	. .
.	. .	. .	. .
$[n/2]-1$	$[n/2]$, 0	$[n/2]+1$, 1	$[n/2]$, 4
$[n/2]$	$[n/2]+1$, 0	1, 3	2, 3
$[n/2]+1$	$[n/2]+2$, 0	$[n/2]+2$, 2	$[n/2]+2$, 4
.	. .	. .	. .
.	. .	. .	. .
.	. .	. .	. .
j	$j+1$, 0	$j+1$, 2	$j+1$, 4
.	. .	. .	. .
.	. .	. .	. .
.	. .	. .	. .
$n-2$	$n-1$, 0	$n-1$, 2	$n-1$, 4
$n-1$	n, 0	n, 2	n, 4
n	1, 1	1, 2	1, 5

Текущие состояния — Следующие состояния Выход

Доказательство. Покажем, что машина M, определенная с помощью табл. 1, является машиной без потери информации порядка $(1/2)\,n\,(n-1)+1$. В табл. через $[x]$ обозначена целая часть числа x. Для простоты s_i обозначается как i. Очевидно, что мно-

[1]) Теорема справедлива при $n \geqslant 3$ При $n = 1, 2$ легко могут быть построены таблицы соответствующих машин с 2 входными и выходными символами. — *Прим. перев.*

жество $G(M)$ состоит из единственной вершины (1, 2). Так как нет пары состояний (i, j) $(i \neq j)$, удовлетворяющей условиям

$$\exists x, x' \in X, \quad \delta(i, x) = \delta(j, x'), \quad \lambda(i, x) = \lambda(j, x'),$$

то ни из одной вершины (i, j) $(i \neq j)$ не проходит дуга в вершину-пару одинаковых состояний. Таким образом, M является машиной без потери информации. Остальные части доказательства следующие:

(1) В $G_0(M)$ существует путь, начинающийся в вершине (1,2) и содержащий все вершины.

(2) В $G_0(M)$ нет контуров.

Распределим дуги графа пар $G_0(M)$ по следующим типам.

[Тип I] Дуги, соответствующие выходам 0 и 4, т. е.

$$\delta(i_1, x) = i_2, \quad \delta(j_1, x') = j_2,$$
$$\lambda(i_1, x) = \lambda(j_1, x') = 0 \quad \text{(или 4)}.$$

Этот тип дуг соединяет вершины следующим образом:

$$(i, j) \to (i+1, j+1), \quad (i < j < n).$$

Число дуг этого типа равно

$$\frac{(n-1)(n-2)}{2}.$$

[Тип II] Дуги, соответствующие выходу 1, т. е.

$$\delta(i_1, x) = i_2, \quad \delta(j_1, x') = j_2,$$
$$\lambda(i_1, x) = \lambda(j_1, x') = 1.$$

Этот тип дуг соединяет вершины следующим образом:

$$\text{[II-a]} \quad (i, n) \to (1, i+2) \qquad \left(i < \left[\frac{n}{2}\right]\right),$$

$$\text{[II-b]} \quad (i, j) \to (i+2, j+2) \qquad \left(i < j < \left[\frac{n}{2}\right]\right).$$

Число дуг типа [II-a] равно

$$\left[\frac{n}{2}\right] - 1.$$

[Тип III] Дуги, соответствующие выходу 2, т. е.

$$\delta(i_1, x) = i_2, \quad \delta(j_1, x') = j_2,$$
$$\lambda(i_1, x) = \lambda(j_1, x') = 2.$$

Этот тип дуг соединяет вершины следующим образом:

$$\text{[III-a]} \quad (i, n) \to (1, i+1) \qquad \left(\left[\frac{n}{2}\right] < i < n\right),$$

$$\text{[III-b]} \quad (i, j) \to (i+1, j+1) \qquad \left(\left[\frac{n}{2}\right] < i < j < n\right).$$

Число дуг типа [III-a] равно

$$\left[\frac{n+1}{2}\right]-1.$$

Любая дуга графа принадлежит одному из вышеуказанных типов. Покажем, что дуги типов I, II-a и III-a образуют путь, содержащий все вершины (i, j) $(i=j)$, т. е.

$$(1, 2)\to(2, 3)\to \ldots \to(n-1, n) \qquad \text{(Тип I)}$$

$$(n-1, n)\to(1, n) \qquad \text{(Тип III-a)}$$

$$(1, n)\to(1, 3) \qquad \text{(Тип II-a)}$$

$$(1, 3)\to(2, 4)\to \ldots \to(n-2, n) \qquad \text{(Тип 1)}$$

$$(n-2, n)\to(1, n-1) \qquad \text{(Тип III-a)}$$

$$(1, n-1)\to(2, n) \qquad \text{(Тип I)}$$

$$(2, n)\to(1, 4) \qquad \text{(Тип II-a)}$$

$$(1, 4)\to(2, 5)\to \ldots \to(n-3, n) \qquad \text{(Тип I)}$$

$$\vdots$$

$$\left(1, \left[\frac{n+1}{2}\right]+1\right)\to \ldots \to\left(n-\left[\frac{n+1}{2}\right], n\right) \qquad \text{(Тип I)}$$

Такой путь при $n=8$ показан на рис. 1. Так как общее число дуг этих типов равно

$$\frac{(n-1)(n-2)}{2}+\left[\frac{n}{2}\right]-1+\left[\frac{n+1}{2}\right]-1=\frac{1}{2}n(n-1)-1,$$

то все вершины содержатся на этом пути. Остальные дуги — типа II-b и III-b. Так как любая дуга типа II-b соединяет вершины, которые соединены двумя дугами типа I, и любая дуга типа III-b соединяет вершины, которые соединены дугой типа I, то граф пар не имеет контуров. Таким образом, M является машиной без потери информации порядка $(1/2)n(n-1)+1$, ч. т. д.

Покажем, что любая последовательностная машина с n состояниями без потери информации порядка k, для которой достигается верхняя граница числа k, является минимальной.

Допустим, что существует последовательностная машина с n состояниями без потери информации порядка k, для которой достигается верхняя граница числа k и которая не является мини-

мальной. Тогда мы можем получить эквивалентную последовательностную машину с n' состояниями ($n' < n$). Так как обе машины осуществляют один и тот же оператор, то обе являются машинами без потери информации порядка $(1/2)n(n-1)+1$.

Таким образом, существует последовательностная машина с n' состояниями без потери информации порядка $(1/2)n(n-1)+1$, что противоречит условию (4). Что касается нижней границы, то очевидно, что для любых положительных целых чисел n, p и

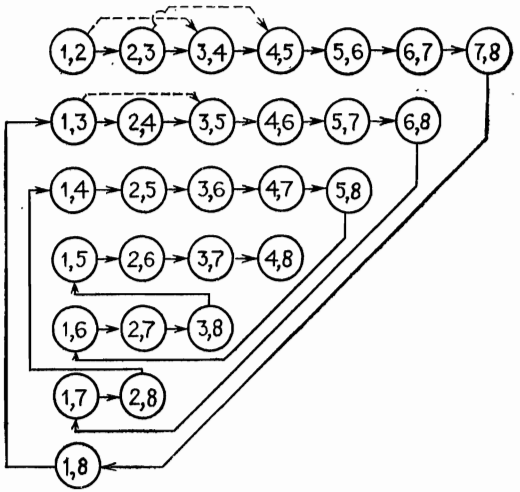


Рис. 1. Граф пар для последовательностной машины с 8 состояниями без потери информации порядка 29.

q ($q \geqslant p$) существует минимальная последовательностная машина с n состояниями без потери информации порядка 1 с p входными и q выходными символами. Следующая машина является одним из примеров.

Пусть $\{B_1, B_2, \ldots, B_p\}$ — разбиение множества Y. Функцию переходов определим следующим образом:

$$x \in X, \quad s_{i+1} = \delta(s_i, x), \quad s_1 = \delta(s_n, x).$$

Функцию выходов определим таким образом, что существуют только одно состояние s_i и два входа x_j и x_k, удовлетворяющие условиям

$$\lambda(s_i, x_j) \in B_k, \quad \lambda(s_i, x_k) \in B_j,$$

а во всех остальных случаях $\lambda(s, x_h) \in B_h$.

Можно сделать вывод, что для любого n существует минимальная последовательностная машина с n состояниями, удовлетворяющая границам (4).

Авторы выражают благодарность профессору К. Маэда и профессору Т. Кийоно из университета Киото за поддержку данного исследования.

СПИСОК ЛИТЕРАТУРЫ

1. Even S., On information lossless automata of finite order, *IEEE Trans. Electronic Computers*, **EC-14**, № 4 (1965), 561—569.
2. Huffman D. A., Canonical forms for information-lossless finite-state logical machines, *IRE Trans. Circuit Theory*, **CT-6**, Spec. suppl., (1959), 41—59.
3. Kambayashi Y., Yajima S., Finite memory sequential machines and information lossless sequential machines, Report A-6, Data processing center, Kyoto University, Kyoto, Japan, 1971.

Синтаксис и семантика с точки зрения теории категорий [1)]

Д. Б. Бенсон

1. ВВЕДЕНИЕ

Томпсон [15] предложил некоторое понятие семантики для формальных языков, порождаемых полутуэвскими системами, в котором вывод интерпретируется функцией. Такие же семантические понятия для контекстно-свободных языков и, в частности, при построении компиляторов использовались Кнутом [8]. Эти семантические понятия тесно связаны с семантикой исчислений (см. [1]). В теории моделей для логических исчислений интерпретации осуществляются в системах отношений и некоторые совокупности интерпретаций играют особо важную роль.

В языках программирования и в теории формальных языков семантика — это некоторая операция или функция над определенными множествами. Например, переработка информации или обращение к подпрограмме и ее выполнение являются такими операциями. Понятие истинности не является при этом основным, однако классы интерпретаций все же полезны.

Хотц [7] (см. также [13]) предложил использовать теорию категорий для изучения структуры выводов, налагаемой на формальный язык системой подстановок, порождающей этот язык. Категорный подход привлекателен своей ясностью. Структура выводов, называемая здесь синтаксисом, является определенным типом категории, порождаемой полутуэвской системой. Богатство синтаксиса с точки зрения выводимости зависит от цели исследования, как видно из следующего абзаца. Область семантики — это категория множеств и функций. Интерпретации синтаксиса являются кофункторами из синтаксиса в область семантики, строящимися с помощью сопоставления функций каждой продукции полутуэвской системы. Автор заметил здесь явную связь с «теориями» Лоувера (см. [13]).

Более подробно, объектами синтаксиса являются цепочки букв некоторого фиксированного алфавита и, если угодно, только те цепочки, которые выводимы из аксиомы полутуэвской системы. Морфизмы — это выводы одной цепочки из другой. Если выводы чисто синтаксические, или свободные, то считается, что между

[1)] Benson D. B., Syntax and semantics: a categorical view, *Information and Control*, 17 (1970), 145—160.

ними делаются несущественные различия. Можно сказать, что структура выводов слишком богата. Хотц [7] и Гриффитс [6] исследовали этот вопрос, рассматривая некоторое отношение между свободными выводами, называемое подобием. Доказано, что это отношение является отношением эквивалентности при более общих условиях, чем понадобятся нам. Действительно, оно является конгруэнтностью для операции композиции морфизмов. Классы эквивалентности Хотц также называет выводами. Семантическая теория показывает, что это вполне удобно, так как можно поставить в соответствие конгруэнтностям выводов классы интерпретации, подобно тому как делается в алгебраической логике. Показано, что отношение подобия соответствует классу всех интерпретаций.

Можно изучить даже более бедные системы выводов, ставя в соответствие меньшим классам интерпретаций большие классы эквивалентности. В качестве применения этой теории в данной статье исследуются три различных понятия контекстных систем. Каждая разновидность определяется в литературе без семантики. Эти разновидности мы получим, рассматривая сужающиеся подклассы интерпретаций.

2. ОБОЗНАЧЕНИЯ И ТЕРМИНОЛОГИЯ

Категория — это совокупность объектов и совокупность морфизмов. Морфизмы с областью определения a и образом b обозначаются $a \to b$ или снабжаются меткой, например, $x : a \to b$. Множество морфизмов с областью определения a и образом b обозначается (a, b). Для каждого объекта a в (a, a) имеется по крайней мере один морфизм — тождество на a. Если $x \in (a, b)$, $y \in (b, c)$, то произведение yx содержится в (a, c). Произведение морфизмов ассоциативно. Функтор из одной категории в другую — это пара функций: функция объектов отображает объекты в объекты, а функция морфизмов отображает морфизмы в морфизмы, сохраняя тождества и произведения морфизмов. Точные определения можно найти у Маклейна и Биркгоффа [9], Митчелла [10], Фрейда [4] или Кона [1]:

Строчными греческими буквами обозначаются цепочки в алфавите A. Исключениями является $\varnothing$, обозначающая пустое множество, а также $\in$, обозначающая принадлежность классу. Пустая цепочка обозначается через λ. Множество всех цепочек обозначается через A^*, множество всех непустых цепочек через $A^+ = A^* - \{\lambda\}$; $N = \{0, 1, 2 \ldots\}$ — множество натуральных чисел.

Полутуэвская система, как она рассматривается в этой статье, — это упорядоченная тройка объектов $G = (A, P, \delta)$, где A — алфавит, P — множество продукций, т. е. конечное бинарное отношение на A^+, а σ — аксиома G, $\sigma \in A^+$. Члены множества P за-

писываются в виде $\alpha \to \beta$, что согласуется с общепринятой терминологией и с нашим намерением рассматривать их как морфизмы некоторой категории.

Определение вывода, приведенное ниже, отличается от общепринятого и эквивалентно определению Гриффитса [6]. Решающим фактом является то, что так определенные выводы несут достаточную информацию, чтобы полностью и однозначно восстановить операции подстановки. Кроме того, используемая здесь особая форма выводов обладает теми преимуществами с алгебраической точки зрения, что подвывод является выводом, естественная композиция выводов снова представляет собой вывод и, если $\theta \to \psi$ — вывод, то это же можно сказать и о его распространении с помощью μ и ν — $\mu\theta\nu \to \mu\psi\nu$. Более того, выводы длины 0 естественным образом ставятся во взаимно однозначное соответствие цепочкам, образуя категорию тождеств, а продукции естественным образом совпадают с некоторыми выводами длины 1. Читатель, удовлетворенный приведенным выше описанием и не заинтересованный в подробностях, может пропустить дальнейшую часть этого раздела.

Выводом из θ в ψ называется упорядоченная тройка конечных последовательностей. Первый ее член является доказательством [12], т. е. последовательностью цепочек, получающихся друг из друга с помощью подстановок. В литературе по теории формальных языков обычно называют выводом одну эту последовательность. Второй член тройки является последовательностью продукций, применявшихся при переходе от одной цепочки к другой. Третий член является последовательностью левых и правых контекстов, в которых применялись эти продукции.

Определение 2.1. Выводом из θ в ψ называется упорядоченная тройка объектов

$$((\theta_0, \ldots, \theta_n), (r_0, \ldots, r_{n-1}), (\mu_0 - \nu_0, \ldots, \mu_{n-1} - \nu_{n-1}))$$

для некоторого $n \in N$, такая, что $\theta_0 = \theta$, $\theta_n = \psi$ и для всех $i < n$ имеем $\theta_i = \mu_i \alpha_i \nu_i$, $\theta_{i+1} = \mu_i \beta_i \nu_i$, где $r_i = \alpha_i \to \beta_i \in P$.

Определение 2.2. Если

$$x_1 = ((\theta_0, \ldots, \theta_n), (r_0, \ldots, r_{n-1}), (\mu_0 - \nu_0, \ldots, \mu_{n-1} - \nu_{n-1}))$$

— вывод из θ в ψ, а

$$x_2 = ((\psi_0, \ldots, \psi_m), (q_0, \ldots, q_{m-1}), (\pi_0 - \rho_0, \pi_{m-1} - \rho_{m-1})$$

— вывод из ψ в ξ, то произведением x_1 и x_2, которое мы обозначаем $x_2 x_1$, называется вывод (d, r, c) из θ в ξ, такой, что

$$d = (\theta_0, \ldots, \theta_n, \psi_1, \ldots, \psi_m), \quad r = (r_0, \ldots, r_{n-1}, q_0, \ldots, q_{m-1}),$$
$$c = (\mu_0 - \nu_0, \ldots, \mu_{n-1} - \nu_{n-1}, \pi_0 - \rho_0, \ldots, \pi_{m-1} - \rho_{m-1}).$$

Определение 2.3. Вывод $x_2 = (d, r, c)$ из $\mu\theta\nu$ в $\mu\psi\nu$ называется (μ, ν)-распространением вывода $x_1 = ((\theta_0, \ldots, \theta_n), q, (\pi_0 - \rho_0, \ldots, \pi_{n-1} - \rho_{n-1}))$ из θ в ψ тогда и только тогда, когда $d = (\mu\theta_0\nu, \ldots, \mu\theta_n\nu)$, $r = q$, $c = (\mu\pi_0 - \rho_0\nu, \ldots, \mu\pi_{n-1} - \rho_{n-1}\nu)$.

Определение 2.4. Вывод длины 0 $((\theta), ( ), ( ))$ называется θ-тождественным выводом.

Определение 2.5. Вывод длины 1 $((\alpha, \beta), (\alpha \to \beta), (\lambda - \lambda))$ называется $(\alpha \to \beta)$-выводом.

3. СИНТАКСИС

Будут определены три категории, называемые синтаксическими. Каждая из них описывает структуру свободных выводов, G-индуцируемую в A^+. Основная категория, **F**, является G-индуцируемой алгеброй отношений, рассматриваемой как категория. Объектами **F** являются все цепочки из A^+, а множество морфизмов (θ, ψ) — это множество выводов из θ в ψ. **F** — категория, поскольку (i) композиция выводов, которая ассоциативна, совпадает с произведением морфизмов и (ii) для каждого объекта θ θ-тождественный вывод является в категории тождеством для θ по отношению к операции композиции выводов.

Для всякой продукции из P $\alpha \to \beta$ $(\alpha \to \beta)$-вывод длины один является и морфизмом с областью определения α, и образом β, и применением продукции $\alpha \to \beta$. Мы намеренно не будем различать такой морфизм и продукцию $\alpha \to \beta$, так что продукции будут рассматриваться как морфизмы категории **F**.

Для всякого морфизма категории **F** $x : \theta \to \psi$ и всякой пары объектов μ, ν существует единственный морфизм категории **F** $\mu\theta\nu \to \mu\psi\nu$, являющийся (μ, ν)-распространением x. Распространение x преобразует θ в ψ, а μ и ν остаются неизменными в ходе вывода.

Морфизмы называют выводами, если хотят подчеркнуть лингвистическую структуру. Категория **F** представляет лингвистический интерес, поскольку позволяет определить следующую подкатегорию.

Синтаксис **A** — это полная подкатегория категории **F**, такая, что (θ, ψ) является множеством морфизмов **A** в том и только в том случае, если (σ, θ) в **F** непусто. Объекты **A** восстанавливаются по тождествам, попавшим в **A**, и являются цепочками из A^*, выводимыми из аксиомы σ.

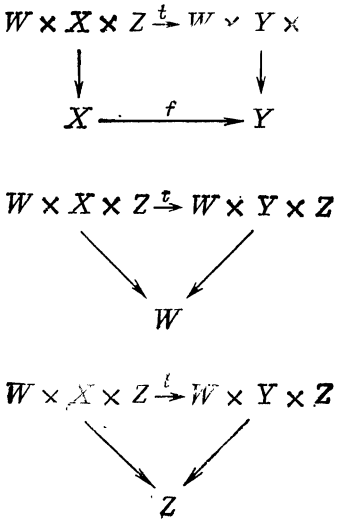
Если разбить алфавит A на непересекающиеся алфавиты I и E, внутренний и внешний соответственно, то внешний синтаксис **E** можно определить как полную подкатегорию синтаксиса **A**, такую, что (θ, ψ) является множеством морфизмов синтаксиса **E** в том и только том случае, если имеется $\omega \in E^+$, такая, что (ψ, ω) не-

пусто в **A**. Цепочки из E^+, являющиеся одновременно объектами синтаксиса **E**, образуют язык грамматики G в обычном смысле. Термин «внешний» используется, чтобы избежать путаницы с термином «терминальный» из теории категорий. В дальнейшем внешний синтаксис рассматриваться не будет.

Следующие утверждения получаются сразу же. Если G моногенна, то **A** упорядочена. Грамматика G без зацикливания тогда и только тогда, когда для каждого объекта θ из **A** (θ, θ) состоит из одного тождества. Если $.P$ — бинарное отношение на $(A^+ - E^+) \times A^+$, то всякий морфизм для **A**, образом которого является член E^+, является эпиморфизмом. В общем случае **A** не имеет терминальных объектов.

4. СЕМАНТИКА И ИНТЕРПРЕТАЦИИ

В качестве области семантики берется некоторая фиксированная категория множеств и функций **S**, в которой имеются прямые произведения. Взятие прямого произведения рассматривается как ассоциативная операция. При этом распространение функции определяется следующим образом: если W, X, Y и Z — множества, а f: $X \to Y$ — некоторая функция из X в Y, то t: $W \times X \times Z \to W \times Y \times Z$ является распространением f, сохраняющим W и Z, тогда и только тогда, когда t действует на Y как f и действует как тождественная функция на W и Z, т. е. t является распространением f, если следующие три диаграммы коммутативны (не отмеченные буквами стрелки обозначают проекции).



Интерпретация синтаксиса $\mathbf{A}$ — это кофунктор I: $\mathbf{A} \to \mathbf{S}$, переводящий цепочки в произведения, а выводы — в функции. Это будет легче представить, если рассматривать интерпретации для основной категории $\mathbf{F}$, а затем ограничить их на подкатегорию $\mathbf{A}$.

Определение 4.1. Кофунктор I: $\mathbf{F} \to \mathbf{S}$ является интерпретацией тогда и только тогда, когда выполнены п. (i) и (ii).

(i) Функция объектов кофунктора I удовлетворяет условиям $I(a) \neq \varnothing$ для всякого $a \in A$ и $I(\alpha a) = I(\alpha) \times I(a)$ для $\alpha \in A^+$, $a \in A$.

(ii) Функция морфизмов кофунктора I обладает тем свойством, что для всякого вывода $\theta \to \psi$ из $\mathbf{F}$ и всякой пары $\mu, \nu \in A^*$ интерпретацией (μ, ν)-распространения вывода $\theta \to \psi$ $(\mu\theta\nu \to \mu\psi\nu)$ является функция

$$I(\mu\theta\nu \to \mu\psi\nu)\colon I(\mu\psi\nu) \to I(\mu\theta\nu),$$

представляющая собой распространение функции $I(\theta \to \psi)\colon I(\psi) \to I(\theta)$, сохраняющее $I(\mu)$ и $I(\nu)$.

Для того чтобы задать некоторую интерпретацию, достаточно задать функцию, ставящую в соответствие буквам алфавита множества, и функцию I, ставящую в соответствие продукциям функции, причем, если $\alpha \to \beta \in P$, то $I(\alpha \to \beta)\colon I(\beta) \to I(\alpha)$. Если $I(\alpha)$ является произведением k множеств, а $I(\beta)$ — произведением n множеств, то $I(\alpha \to \beta)$ представляет собой набор из k n-арных функций. Если кофунктор I: $\mathbf{F} \to \mathbf{S}$ является интерпретацией, то ограничение кофунктора I на $\mathbf{A}$, I: $\mathbf{A} \to \mathbf{S}$, называется интерпретацией. Образ интерпретации называется семантикой интерпретации. В общем случае семантика не является подкатегорией категории $\mathbf{S}$.

Если $\omega \in E^+$, а $\sigma \to \omega$ — вывод в $\mathbf{A}$, где σ — аксиома, то ω называется предложением. Интерпретациями предложения являются функции из $I(\omega)$ в $I(\sigma)$. Значениями предложения ω называются образы интерпретаций предложения ω. Предложение ω является неоднозначным в обычном синтаксическом смысле, если у него имеется несколько интерпретаций, задаваемых I. Обратное утверждение неверно.

В качестве примера рассмотрим синтаксис для сложения натуральных чисел. Алфавит необычен, поскольку нужно избежать смешения синтаксических и семантических объектов. В качестве алфавита рассматривается {\$, #, @}, в качестве продукций

$$\$ \to \$@\$$$
$$\$ \to \#$$

а аксиомой является \$. Знак # должен обозначать цифру 1, @ — знак сложения, а \$ — синтаксический класс, выражение. Привычные символы, имеющие такие значения, сохраняются для обозначения соответствующих семантических объектов. Интерпретация

алфавита задается следующим образом: $I(\$) = N$, множеству всех натуральных чисел; $I(\#) = \{1\}$, множеству, единственным членом которого является натуральное число 1; $I(@) = \Lambda$, по определению, — тождество для операции прямого произведения. Можно обойтись без употребления Λ, но это облегчает изложение примера. Интерпретация правил задается следующим образом:

$$I(\$ \to \$@\$) : N^2 \to N = + : N^2 \to N,$$

обычное сложение натуральных чисел,

$$I(\$ \to \#) : \{1\} \to N,$$

вложение единицы в множество натуральных чисел.

Имеется два различных дерева вывода $\#@\#@\#$ из $, но каждый из этих выводов имеет интерпретацию $f: \{1\} \times \{1\} \times \{1\} \to N$, такую, что $f(1, 1, 1) = 3$. Используя нашу терминологию, интерпретацией предложения $\#@\#@\#$ является $f: \{1\} \times \{1\} \times \{1\} \to N$, а значением предложения $\#@\#@\#$ — образ f — $\{3\}$; в этом случае семантика интерпретации является подкатегорией категории **S**. Этот пример свидетельствует о сходстве наших формулировок с формулировками для автоматов на деревьях (см. [14]).

5. КОНГРУЭНТНОСТИ

Пусть $\sim$ — отношение эквивалентности, определенное на каждом множестве морфизмов (θ, ψ). Если два морфизма не имеют одинаковых областей определения и одинаковых образов, то они неэквивалентны. Предположим, что эквивалентность $\sim$ обладает тем свойством, что для всех η, θ, ψ, ω, $w \in (\eta, \theta)$, $x, y \in (\theta, \psi)$, $z \in (\psi, \omega)$ из $x \sim y$ следует $xw \sim yw$ и $zx \sim zy$. Тогда $\sim$ называется конгруэнтностью. Классы эквивалентности, на которые каждое множество (θ, ψ) разбивается конгруэнтностью $\sim$, являются морфизмами категории частных $\mathbf{A}/\sim$ ([10]).

Пусть Φ — класс интерпретаций синтаксиса **A**. Для всякой конгруэнтности $\sim$ определим $\Xi(\sim)$ как класс всех таких $I \in \Phi$, что для $x \sim y$ справедливо $I(x) = I(y)$. Для всякого подкласса $\Xi \subseteq \Phi$ можно определить подобие по модулю Ξ, $\sim(\Xi)$, следующим образом: $x \sim y(\Xi)$ тогда и только тогда, когда x и y имеют одну и ту же область определения и один и тот же образ и для всех $I \in \Xi$ справедливо $I(x) = I(y)$. Соответствия

$$\sim \to \Xi(\sim)$$

$$\Xi \to \sim(\Xi)$$

образуют соответствие Галуа (см. [1]) между отношениями конгруэнтности и классами интерпретаций, что отражено в следующем утверждении.

Теорема 5.1. *Пусть* $\sim_1$, $\sim_2$ — *конгруэнтности на* **A**, Ξ_1 и Ξ_2 — *подклассы* Φ. *Тогда*

$$\text{из } \sim_1 \subseteq \sim_2 \text{ следует } \Xi(\sim_1) \supseteq \Xi(\sim_2),$$
$$\text{из } \Xi_1 \subseteq \Xi_2 \text{ следует } \sim\Xi_1) \supseteq \sim(\Xi_2),$$
$$\sim_1 \subseteq \sim(\Xi(\sim_1)),$$
$$\Xi_1 \subseteq \Xi(\sim\Xi_1).$$

В интуитивном смысле два вывода конгруэнтны, если они обладают одинаковой структурой и, следовательно, определяют один и тот же набор возможных значений для выводимого предложения. Соответствие Галуа дает семантическое определение структуры. Структурное описание вывода — это класс эквивалентности, которому он принадлежит, а, согласно соответствию Галуа, он определяется специальным подклассом интерпретаций, которые признаются допустимыми, исходя из некоторых внешних критериев.

Неинтерпретированное понятие структурного описания происходит от конгруэнтности, определенной Гриффитсом [6] и названной им отношением подобия. Пусть в дальнейшем $\sim$ обозначает отношение подобия. В этом разделе мы покажем, что $\sim = \sim(\Phi)$, т. е. два вывода подобны тогда и только тогда, когда они одинаково интерпретируются при любой возможной интерпретации. Этот результат можно рассматривать как еще один довод в пользу изучения X-категорий Дж. Хотца. Свободные X-категории являются категориями частных основных синтаксисов по модулю подобия. Если $\Xi(\approx)$ образует собственный подкласс Φ для некоторой конгруэнтности $\approx$, то соответствующая категория частных является несвободной X-категорией.

Вначале требуется определить неинтерпретированное, или синтаксическое подобие. Подобие — это наименьшее отношение конгруэнтности на **A**, такое, что если

$$x\colon \theta \to \psi, \quad y\colon \theta' \to \psi'$$

— выводы синтаксиса **A** с распространениями

$$x_0\colon \theta\theta' \to \psi\theta', \quad x_1\colon \theta\psi' \to \psi\psi', \quad y_0\colon \theta\theta' \to \theta\psi', \quad y_1\colon \psi\theta' \to \psi\psi',$$

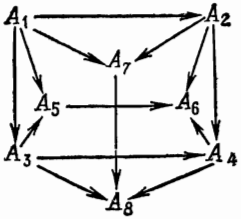
а $u = y_1x_0$, $v = x_1y_0$, то u подобно $v(u \sim v)$.

Для того чтобы обосновать это определение, приведем краткое описание рассуждений Гриффитса. Вывод может быть заменен другим с помощью перестановки двух применений продукций, если эти применения не взаимодействуют. Из того что написано выше видно, что x_1y_0 и y_0x_1 можно заменять друг на друга. Вывод может быть L-заменен, если его можно заменить выводом, который действует раньше на некоторое начало исходной цепочки. Так, x_1y_0 может быть L-заменен. В каждом классе подобия имеется ровно один канонический вывод. В случае контекстно-свободных систем

канонические выводы изоморфны левосторонним выводам [5], а классы подобия $[\theta \to \psi]$ изоморфны деревьям выводов ψ из θ.

Для того чтобы доказать, что из $u \sim v$ следует $u \sim v(\Phi)$ будет полезен следующий факт из теории категорий.

Теорема 5.2. *Если в двойной треугольной призме*



коммутативны все квадраты и все треугольники, за исключением, быть может, квадрата с вершинами A_1, A_2, A_3, A_4, а $\{A_4 \to A_6, A_4 \to A_8\}$ — произведение, то и оставшийся квадрат коммутативен.

Доказательство. Обозначим через f_{ij} морфизм $A_i \to A_j$. Тогда

$$f_{56}f_{15} = f_{56}f_{35}f_{13} = f_{46}f_{34}f_{13} = f_{26}f_{12} = f_{46}f_{24}f_{12}.$$

Аналогично $f_{78}f_{17} = f_{48}f_{34}f_{13} = f_{48}f_{24}f_{12}$. Так как $\{f_{46}, f_{48}\}$ — произведение, то морфизмы $f_{56}f_{15}$ и $f_{78}f_{17}$ единственным образом проходят через него и, следовательно, $f_{34}f_{13} = f_{24}f_{12}$.

Теорема 5.3.. *Из $u \sim v$ следует $u \sim v(\Phi)$.*

Доказательство. Предположим, что $u \sim v$. Если $u = y_1x_0$, а $v = x_1y_0$, где x_0, x_1, y_0, y_1 — описанные выше распространения x и y, то $I(u) = I(x_0)I(y_1)$, $I(v) = I(y_0)I(x_1)$ и $I(x_0)$, $I(x_1)$ являются распространениями $I(x)$, а $I(y_0)$, $I(y_1)$ — распространениями $I(y)$. Получаем двойную треугольную призму, где $I(\psi\psi') = A_1$, $I(\theta\psi') = A_2$, $I(\psi\theta') = A_3$, $I(\theta\theta') = A_4$, $I(\psi) = A_5$, $I(\theta) = A_6$, $I(\psi') = A_7$, $I(\theta') = A_8$. Все треугольники и квадраты в ней коммутативны согласно свойствам распространений и приведенному выше факту из теории категорий. В частности, $I(u) = I(v)$. Поскольку отношение, задающее подобие, является подмножеством $\sim(\Phi)$, то из определения подобия как наименьшей конгруэнтности следует, что $\sim \subseteq \sim(\Phi)$.

Для того чтобы доказать, что из $u \sim v(\Phi)$ следует $u \sim v$, требуется построить интерпретацию Q, которая дает возможность восстановить вывод с точностью до подобия. Поскольку $Q \in \Phi$, то желаемый результат получается, если показать, что из $Q(u)$ можно получить канонический вывод, подобный u. Интерпретацию Q

можно представить себе как наилучшую гёделевскую нумерацию выводов, которая может быть получена как интерпретация. Гёделевская нумерация проводится в следующей обобщенной арифметике $G(A,P)$ над алфавитом A и множеством продукций P.

(i) $A \subseteq G(A,P)$.

(ii) Если $a \in A$, $r \in P$ и $g \in G(A,P)$, то $(a,r,g) \in G(A,P)$.

(iii) Конечные последовательности членов $G(A,P)$ являются членами $G(A,P)$.

Нужные нам проекции из $G(A,P)$ обозначаются следующим образом:

Если $g=(a,\ r,\ g') \in G(A,\ P)$, то $p_0 g = a$, $p_1 g = r$, $p_2 g = g'$.

Определим интерпретацию Q на буквах алфавита так: для $a \in A$ имеем $Q(a) = \{a\} \cup \{(a,r,g) \mid r \in P \,\&\, g \in G(A,P)\}$. Интерпретация Q определяется на продукции $r = \alpha \to \beta$, где $\alpha = a_1, \ldots, a_k$, как $Q(r)$: $Q(\beta) \to Q(\alpha)$, так что для всякого $g \in Q(\beta)$ имеем $Q(r)(g) = ((a_1,r,g), \ldots, (a_k,r,g))$. Затем Q распространяется на **A**, как было описано в разд. 4.

Если u — вывод из θ в ψ, то можно определить гёделев номер вывода u, $Gd(u)$, следующим образом: $Gd(u) = Q(u)(\psi)$. Если дан гёделев номер $Gd(u) = (g_1, \ldots, g_m)$, то область определения вывода u можно восстановить в виде $a_1, \ldots, a_m$, где $a_i = g_i$, если g_i не является упорядоченной тройкой, и $a_i = p_0 g_i$, если g_i — тройка. Восстановление образа вывода u из $Gd(u)$ происходит в процессе восстановления канонического вывода, подобного u.

Теорема 5.4. *Выводы u и v подобны тогда и только тогда, когда совпадают их гёделевы номера.*

Доказательство. Из $u \sim v$ следует $Q(u) = Q(v)$, а значит, $Gd(u) = Gd(v)$. Предположим, что $Gd(u) = Gd(v)$. Достаточно дополнительно предположить, что v — канонический вывод. Применение следующей процедуры позволяет восстановить v по $Gd(u)$ за конечное число шагов.

Предположим, что $Gd(u) = (g_1, \ldots, g_m)$, $g_i \in G(A,P)$. Для наибольшего i, такого, что $g_1, \ldots, g_i$ все являются буквами, а значит, не тройками, никакая продукция не применялась к начальному отрезку $\gamma = g_1 \ldots g_i$. Если $i = m$, то построение закончено. Если $i \neq m$, то рассмотрим $g_{i+1} = (a,r,g')$, где $a \in A$, $r \in P$, $g' \in G(A,P)$. Если $r = \alpha \to \beta$, длина α равна k, α совпадает с конкатенацией $p_0 g_{i+1}, \ldots, p_0 g_{i+k}$, каждая из $p_1 g_{i+1}, \ldots, p_1 g_{i+k}$ совпадает с r, а $g' = p_2 g_{i+1} = \ldots = p_2 g_{i+k}$, то в этом месте в каноническом выводе v применяется продукция $r = \alpha \to \beta$. Следующая конечная последовательность, к которой применяется процесс редукции, получается, если заменить в $(g_1, \ldots, g_m)$ подпоследовательность $(g_{i+1}, \ldots, g_{i+k})$ последовательностью g'.

Если при описанной выше проверке применимости продукции $r = \alpha \to \beta$ для некоторого j, $1 \leqslant j \leqslant k$, $a_j \neq p_0 g_{i+j}$, или $r \neq p_1 g_{i+j}$, или $g' = p_2 g_{i+j}$, то следует отложить рассмотрение r, так как раньше применяется какая-то другая продукция, применение которой перекрывается с проверяемым применением на g_{i+j}. Здесь нужно начать проверку на $(i+j)$-м месте. Вывод u подобен некоторому каноническому выводу, например, x. Отсюда $Q(u) = Q(x)$ и $Gd(u) = Gd(x) = Gd(v)$. Так как v канонический, то, согласно описанной выше конструкции, он совпадает с x. Итак, $u \sim v$.

Из теорем 5.3 и 5.4 получаем теорему 5.5.

Теорема 5.5. *Для выводов u и v из* **A** $u \sim v$ *тогда и только тогда, когда* $u \sim v(\Phi)$.

Отметим в качестве следствия, что если Ω — класс интерпретаций и $Q \in \Omega$, то $\sim(\Omega) = \sim$ и $\Xi(\sim(\Omega)) = \Phi$. Конгруэнтность равенства на **A** имеет Φ в качестве соответствующего класса интерпретаций, так что $\sim(\Xi(=)) = \sim$.

6. КОНТЕКСТНЫЕ СИСТЕМЫ

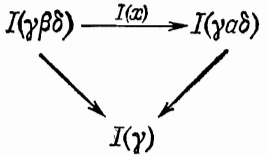
Имеется по крайней мере три различных понятия структурного описания предложений, порождаемых контекстной грамматикой. Первое носит чисто синтаксический характер, и в этом случае вместо допустимости подстановки с помощью контекстно-свободной продукции только в фиксированном (и локальном) контексте требуется только, чтобы длины левых частей продукций не превышали длин соответствующих правых частей. В этом смысле полутуэвская продукция является контекстной типа 1 в том случае, когда длина α не превышает длины β. Хорошо известно, что это условие слабо эквивалентно более строгому условию на продукции. Однако соответствующие структурные описания не обладают той силой, которая интуитивно кажется необходимой. Пусть P — множество контекстных продукций типа 1 и (A, P, s) порождает синтаксис **C**. Класс интерпретаций синтаксиса **C** обозначается через Φ и структурные описания выводов в **C** являются морфизмами в **C**/$\sim$. При чисто синтаксическом подходе никакой дополнительной структуры не налагается. Синтаксис **C** называется контекстным синтаксисом типа 1.

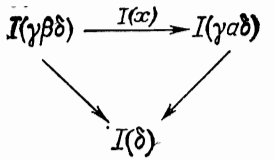
Второе и третье понятия возникают при рассмотрении контекстных продукций как контекстно-свободных продукций, применяемых только в определенном локальном контексте. Такие продукции часто записываются в виде $a \to \beta/\gamma\_\delta$, где a — внутренняя буква, $\beta \in A^+$, а $\gamma, \delta \in A^*$; γ и δ являются левым и правым контекстом, соответственно, ограничивающими применимость $a \to \beta$. Такие продукции можно рассматривать как полутуэвские, а соответствующий синтаксис будет контекстным типа 2 или 3 в зависимости от

подкласса интерпретаций. Тип 3 соответствует такому подходу, когда структурным описанием является построенное по выводу дерево. Если взять более широкий класс интерпретаций, то получатся структурные описания Куно ([11]), называемые здесь описаниями типа 2.

Контекстная продукция $\alpha \to \beta/\gamma\_\delta$ — это полутуэвская продукция $\gamma\alpha\delta \to \gamma\beta\delta$, для которой допустимы интерпретации, I, функциями, постоянными на $I(\gamma)$ и $I(\delta)$ в произведении $I(\gamma\alpha\delta) = I(\gamma) \times I(\alpha) \times I(\delta)$. Тип 2 от типа 3 отличается действием функции на $I(\alpha)$. Если структурные описания должны быть деревьями выводов, то интерпретация $\gamma\alpha\delta - \gamma\beta\delta$ должна зависеть только от интерпретации $\alpha \to \beta$. То есть функция $I(\gamma\alpha\delta \to \gamma\beta\delta)$: $I(\gamma\beta\delta) \to I(\gamma\alpha\delta)$ должна быть постоянной на $I(\gamma)$ и $I(\delta)$ и действовать как некоторая функция f: $I(\beta) \to I(\alpha)$ на прообразе $I(\gamma\alpha\delta)$. Для типа 2 соответствующая функция может иметь в качестве аргумента любой элемент из $I(\gamma\beta\delta) - f$: $I(\gamma\beta\delta) \to I(\alpha)$.

Формально говоря, контекстный синтаксис типа 2 — это такой контекстный синтаксис **C** типа 1, что каждой продукции $x, x \in \in (\gamma\alpha\delta, \gamma\beta\delta)$ полутуэвской системы, порождающей **C**, соответствует ее контекст γ_δ. Это обстоятельство обозначается так: $x = \alpha \to \beta/\gamma\_\delta$. Заметим, что множество $(\gamma\alpha\delta, \gamma\beta\delta)$ может содержать и больше одной продукции; такие кратные продукции различаются по своему контексту. Кроме того, класс допустимых интерпретаций Ξ — это наибольший подкласс класса Φ, такой, что для всякой продукции $x = \alpha \to \beta/\gamma\_\delta$ и для всякой $I \in \Xi$ будут коммутативными следующие диаграммы, где непомеченные буквами стрелки являются проекциями.





Контекстный синтаксис типа 3 — это контекстный синтаксис **C** типа 2, для которого классом допустимых интерпретаций Ω является наибольший подкласс класса Ξ, такой, что для всякой $I \in \Omega$

и всякой продукции $x = a \to \beta/\gamma\_\delta$ из полутуэвской системы, порождающей **C**, имеется функция

$$f(x, I)\colon I(\beta) \to I(a).$$

Ω и функции $f(x, I)$ обладают такими свойствами:

(i) Коммутативна диаграмма

$$\begin{array}{ccc} I(\gamma\beta\delta) & \xrightarrow{I(x)} & I(\gamma a\delta) \\ \downarrow & & \downarrow \\ I(\beta) & \xrightarrow{f(x,I)} & I(a) \end{array}$$

где проекции не отмечены буквами.

(ii) Если $x = a \to \beta/\gamma\_\delta$ и $y = a \to \beta/\pi\_\rho$ — продукции, то для всякого $I \in \Omega$ имеем $f(x, I)\colon I(\beta) \to I(a) = f(y, I)\colon I(\beta) \to I(a)$.

Последнее свойство соответствует тому, что $a \to \beta/\gamma\_\delta$ и $a \to \beta/\pi\_\rho$ рассматриваются как одна и та же контекстно-свободная продукция $a \to \beta$, ограниченная контекстом γ_δ или π_ρ. Без этого свойства нельзя построить по выводу дерево. Структурным описанием вывода x в **C** является класс эквивалентности $[x]_{\sim(\Omega)}$. Каждому такому классу эквивалентности соответствует единственным образом дерево вывода в соответствующем контекстно-свободном синтаксисе, как показывают следующие рассуждения.

Предположим, что P — это множество продукций, порождающее вместе со своими контекстами контекстный синтаксис **C** типа 3 с классом интерпретаций Ω. Тогда $R = \{a \to \beta \mid a \to \beta/\gamma\_\delta \in P\}$ — множество контекстно-свободных продукций, порождающее контекстно-свободный синтаксис **D**, деревья выводов которого образуют $\mathbf{D}/\sim$. Функцию $F\colon P \to R$, где $F(a \to \beta/\gamma\_\delta) = a \to \beta$, можно продолжить до освобождающего от контекста функтора $F\colon \mathbf{C} \to \mathbf{D}$. Для всякой $I\colon \mathbf{C} \to \mathbf{S}$ из Ω определим $J_I\colon \mathbf{D} \to \mathbf{S}$ следующим образом.

(i) Для всякого объекта θ имеем $J_I(\theta) = I(\theta)$.

(ii) Для морфизмов определим J_I по индукции, начиная с продукции из R: при $a \to \beta \in R$ полагаем $J_I(a \to \beta) = f(x, I)$ для всякого $x = a \to \beta/\gamma\_\delta \in P$, такого, что $F(x) = a \to \beta$.

Диаграмма

$$\begin{array}{ccc} \mathbf{C} & \xrightarrow{F} & \mathbf{D} \\ & {\scriptstyle I}\searrow \quad \swarrow{\scriptstyle J_I} & \\ & \mathbf{S} & \end{array}$$

коммутативна для любого $I \in \Omega$. Более того, $\{J_I \mid I \in \Omega\}$ совпадает со всем классом интерпретаций синтаксиса **D**, $\Phi_{\mathbf{D}}$, поскольку любую $K \in \Phi_{\mathbf{D}}$ можно продолжить до некоторой $I \in \Omega$.

Из описанного выше соответствия контекстно-свободной системы контекстному синтаксису **C** типа 3 можно заметить, что дерево вывода x из **C** является классом подобия $F(x)$, $[F(x)]$.

Теорема 6.1. *Коммутативна диаграмма*

$$\begin{array}{ccc} \mathbf{C} & \xrightarrow{F} & \mathbf{D} \\ \downarrow & & \downarrow \\ \mathbf{C}/\Omega & \xrightarrow{G} & \mathbf{D}/\sim \end{array}$$

где F: $\mathbf{C} \to \mathbf{D}$ — *освобождающий от контекста функтор для* **C**, $\mathbf{C}/\Omega$ — *категория частных синтаксиса* **C** *по модулю* $\sim(\Omega)$, $\mathbf{D}/\sim$ — *категория частных синтаксиса* **D** *по модулю подобия, а* G: $\mathbf{C}/\Omega \to \mathbf{D}/\sim$ — *структурное описание или функтор деревьев вывода, тождественный на объектах, а на морфизмах* $G([x]_\Omega) = F(x)]$.

Доказательство. G определен корректно, так как для всяких $x, y \in (\theta, \psi)$; если $x \sim y(\Omega)$, то для всякой $I \in \Omega$ имеем $J_I(F(x)) = I(x) = I(y) = J_I(F(y))$, а значит, $F(x) \sim F(y)$. Коммутативность диаграммы вытекает непосредственно из определения G.

Теорема 6.2. *G точен, т. е. инъективен на морфизмах.*

Доказательство. Предположим, что $G([x]_{\mathbf{D}}) = G([y]_{\mathbf{D}})$ или, что эквивалентно, $F(x) \sim F(y)$. Поскольку для всех $K \in \Phi_{\mathbf{D}}$ $K(F(x)) = K(F(y))$ и поскольку всякая $I \in \Omega$ является продолжением некоторой $K \in \Phi_{\mathbf{D}}$, то в этом случае для всех $I \in \Omega$ $I(x) = I(y)$, так что $x \sim y(\Omega)$.

Из доказанных выше утверждений получаем.

Теорема 6.3. *Если $x, y \in (\theta, \psi)$ являются выводами синтаксиса* **C**, *то x и y подобны по модулю Ω в том и только том случае, если они имеют одинаковые деревья. То есть $x \sim y(\Omega)$ тогда и только тогда, когда $F(x) \sim F(y)$.*

Куно [11] определяет деревья с добавленными четверками целых чисел в вершинах для структурного описания контекстных выводов. Соответствующее этому неинтерпретированное отношение конгруэнтности назовем нс-подобием. Нс-подобие, $\simeq$, определяется как наименьшая конгруэнтность на **C**, такая что

(i) из $x \sim y$ следует $x \simeq y$ и

(ii) если $x = a \to \beta/\gamma\_\mu\nu$ и $y = c \to \delta/\nu\pi\_\rho$ — продукции, порождающие **C** и имеющие распространения

$$x_0 : \gamma a \mu \nu \pi c \rho \to \gamma \beta \mu \nu \pi c \rho, \quad x_1 : \gamma a \mu \nu \pi \delta \rho \to \gamma \beta \mu \nu \pi \delta \rho,$$

$$y_0 : \gamma a \mu \nu \pi c \rho \to \gamma a \mu \nu \pi \delta \rho, \quad y_1 : \gamma \beta \mu \nu \pi c \rho \to \gamma \beta \mu \nu \pi \delta \rho,$$

то $y_1 x_0 \simeq x_1 y_0$.

Два нс-подобных вывода можно заменять друг другом, хотя применения продукций могут пересекаться, если пересечение происходит по контекстам, а не по соответствующим контекстно-свободным продукциям. Не удивительно, что нс-подобие совпадает с подобием по модулю Ξ, где Ξ класс интерпретаций типа 2.

Теорема 6.4. $\simeq = \sim(\Xi)$.

Доказательство. По существу доказательство такое же, как для теоремы 5,5, следует только заметить, что интерпретация Q с помощью гёделевской нумерации не входит в Ξ. Для доказательства строится аналогичная гёделевская нумерация Q', удовлетворяющая условиям, налагаемым на интерпретации типа 2. На буквах Q' определяется точно так же, как Q: для $a \in A$ $Q'(a) = \{a\} \cup \{(a, r, g) \mid r \in P \,\&\, g \in G(A, P)\}$. На продукциях дело обстоит иначе. Пусть $r = a \to \beta/\gamma\_\delta$ — продукция, причем длина γ равна l, длина β равна m, а длина δ равна n. Тогда $Q'(r)$ — это такая функция из $Q'(\gamma\beta\delta)$ в $Q'(\gamma a \delta)$, что для $g \in Q'(\gamma\beta\delta)$, $g = (g_1, \ldots, g_l, b_1, \ldots, b_m, d_1, \ldots, d_n)$, $Q'(r)(g) = (g_1, \ldots, g_l, (a, r, g), d_1, \ldots, d_n)$.

Понятия типа 2 легко обобщить на случай продукций вида $\alpha \to \beta/\gamma\_\delta$, где $\alpha, \beta \in A^+$, $\gamma, \delta \in A^*$, и на случай матричных продукций вида

$$\alpha_0, \alpha_1, \ldots, \alpha_{n-1} \to \beta_0, \beta_1, \ldots, \beta_{n-1}/\gamma_0 - \gamma_1 - \ldots \gamma_{n-1} - \gamma_n,$$

которые сводятся в случае $n = 1$ к первым.

СПИСОК ЛИТЕРАТУРЫ

1. Cohn P. M., Universal Algebra, Harper and Row, New York, 1965. (Русский перевод: Кон П. М., Универсальная алгебра, «Мир», М., 1968.)
2. Davis M., Computability and Unsolvability, McGraw-Hill, New York, 1958.
3. Eilenberg S., Wright J. B., Automata in general algebras, *Information and Control.*, **11** (1967), 452—470.
4. Freyd P., Abelian Categories, Harper and Row, New York, 1964.
5. Ginsburg S., The Mathematical Theory of Context-Free Languages, McGraw-Hill, New York, 1966. (Русский перевод: Гинзбург С., Математическая теория контекстно-свободных языков, «Мир», М., 1969.)
6. Griffiths T. V., Some remarks on derivations in general rewriting systems, *Information and Control*, **12** (1968), 27—54.
7. Hotz G., Eindeutigkeit und Mehrdeutigkeit formaler Sprachen, *EIK*, 2 (1966), 235—247.
8. Knuth D. E., Semantics of context-free languages, *Math. Systems Theory*, 2 (1968), 127—146.

9. MacLane S., Birkhoff G., Algebra, Macmillan, New York, 1967.
10. Mitchell B., Theory of Categories, Academic Press, Inc., New York, 1965.
11. Kuno S., A context-sensitive recognition procedure, in Math. Linguistics and Auto. Translation Rpt. NSF-18, Aiken Comp. Lab., Harvard, Cambridge, Mass. 1967.
12. Nelson R. J., Introduction to Automata, John Wiley and Sons, Inc., New York, 1968.
13. Schnorr C. P., Transformational classes of grammars, *Information and Control*, **14** (1969), 252—277.
14. Thatcher J. W., Characterizing derivation trees of context-free grammars through a generalization of finite automata theory, *J. Comp. System Sci.*, **1** (1967), 317—323.
15. Thompson F. B., English for the computer, *AFIPS Conf. Proc.*, **29** (1966), 349—356, FJCC, Spartan Books, Washington, D. C.

Использование списков в изучении проблем теории автоматов[1)]

Дж. Хартманис, Ф. Д. Льюис

1. ВВЕДЕНИЕ

Хорошо известно, что многие проблемы в теории автоматов[2)] и формальных языков алгоритмически неразрешимы и, следовательно, могут быть классифицированы по степеням неразрешимости. В этой работе мы покажем, во-первых, как можно естественным образом построить множества машин Тьюринга, которые являются полными множествами в любом заданном уровне арифметической степени неразрешимости. Конструкция этих множеств прямо соответствует кванторным последовательностям в арифметической иерархии. Во-вторых, описывается метод представления трудных проблем как более простых проблем над списками машин Тьюринга.

Например, проблема определения, допускает ли машина Тьюринга кофинитное множество, эквивалентна проблеме определения, будет ли рекурсивный список машин Тьюринга содержать такую машину, которая допускает любую входную последовательность. Проблема определения, будет ли машина Тьюринга допускать каждую входную последовательность, в свою очередь эквивалентна проблеме определения, будет ли каждая машина Тьюринга из рекурсивного списка допускать некоторую входную последовательность. При более подробном рассмотрении можно заметить, что проблема кофинитности также эквивалентна проблеме определения, будет ли рекурсивный список машин Тьюринга содержать только конечное множество машин, которые не допускают никакую входную последовательность.

Эти примеры основываются на некоторых общих результатах, которые указывают, как увеличивается трудность распознавания свойства P, когда мы задаем «количественный» вопрос о множестве машин со свойством P из рекурсивного списка машин. Используя наши специальные множества, мы установим несколько таких результатов. Эти результаты вносят ясность в различие между степенями неразрешимости проблем о списках и дают некото-

---

[1)] Hartmanis J., Lewis F. D., The use of lists in the study of undecidable problems in automata theory, *Journal of Computer and System Sciences*, 5, № 1 (1971), 54—66.

[2)] Под автоматами авторы понимают не только конечные автоматы. — *Прим. ред.*

рый обзор таких проблем. Более того, эти результаты дают простые методы для определения минимальной степени неразрешимости нескольких хорошо известных проблем и возможность естественным образом строить проблемы с высокой степенью неразрешимости.

2. ПРЕДВАРИТЕЛЬНЫЕ ОПРЕДЕЛЕНИЯ И ОТНОСИТЕЛЬНЫЕ МНОЖЕСТВА

Мы предполагаем, что читатель знаком с понятиями машины Тьюринга, рекурсивными и рекурсивно перечислимыми множествами (р. п.), как определено в [2, 5, 11], а также с неразрешимыми проблемами и их классификацией по относительной трудности [6, 9].

В этой части мы введем некоторые понятия и дадим обзор определений, которые потребуются далее, о сводимости и арифметической иерархии. Читатель, не знакомый с этими определениями, должен обратиться к работам, указанным выше. Пусть $M_0, M_1, M_2, \ldots$ — допустимая [10] нумерация машин Тьюринга или частично рекурсивных функций, и пусть $W_0, W_1, W_2, \ldots$ — множества, допустимые соответствующими машинами. Мы пишем $T(i, x, t) = 1$ тогда и только тогда, когда i-я машина останавливается после t шагов, если на вход ее поступает последовательность x. (Это аналогично T-предикату Клини.) Наконец, строчкой, состоящей из буквы θ и названия множества последовательностей, будем обозначать *множество индексов* (индексное множество) машин Тьюринга, допускающих эти последовательности. Например, θ*Пустое* обозначает множество индексов машин, не допускающих никаких последовательностей.

Множество A *одно-односводимо* к множеству B (обозначение $A \leqslant_1 B$) тогда и только тогда, когда существует взаимно однозначная рекурсивная функция g, такая, что для всех x

$$x \in A \Leftrightarrow g(x) \in B.$$

Множество A *одно-одноэквивалентно* множеству B (обозначение $A \equiv_1 B$) тогда и только тогда, когда $A \leqslant_1 B$ и $B \leqslant_1 A$. Эти определения были введены Постом [8].

Наметим теперь в общих чертах классификацию проблем, неразрешимых средствами арифметической или клиниевской иерархии [4].

Мы говорим, что предикат S является *рекурсивным*, если S рекурсивно разрешим. Говорят, что S — *арифметический предикат* тогда и только тогда, когда S получен из некоторого рекурсивного предиката навешиванием кванторов.

Множество A называется *арифметическим множеством* тогда и только тогда, когда существует арифметический предикат S, такой, что

$$A = \{x \mid S(x)\}.$$

Арифметический предикат S всегда может быть записан в *префиксной нормальной форме* как рекурсивный предикат R с некоторым префиксом кванторов [5, стр. 167; 11, стр. 308]. Этот кванторный префикс может быть представлен в виде чередующихся кванторов, и тогда предикаты помещаются в иерархии согласно числу чередования кванторов существования и общности в префиксе.

(a) Предикат S — это Σ_n-*предикат* тогда и только тогда, когда его префикс начинается с квантора существования $\exists$ и содержит n чередующихся кванторов.

(b) Предикат S — это Π_n-*предикат* тогда и только тогда, когда его префикс начинается с квантора общности $\forall$ и содержит n чередующихся кванторов.

Например, $S(x) = \exists u \forall v \exists w R(u, v, w, x)$ — это Σ_3-предикат при условии, что $R(u, v, w, x)$ — рекурсивный предикат.

Множество A — это Σ_n (Π_n)-*множество* тогда и только тогда, когда существует $\Sigma_n(\Pi_n)$-предикат S, такой, что

$$A = \{x \mid S(x)\}.$$

Для краткости будем писать $A \in \Sigma_n$ или $A \in \Pi_n$.

Множество A Σ_n-*полно* тогда и только тогда, когда $A \in \Sigma_n$ и каждое Σ_n-множество одно-односводимо к A. Аналогичным образом определяется Π_n-*полное* множество.

Определим следующим образом несколько хорошо известных индексных множеств машин Тьюринга, которые будут использоваться ниже:

(a) θ*Пустое* $= \{i \mid M_i$ не допускает никаких последовательностей$\}$

$$= \{i \mid \forall x \forall t [T(i, x, t) \neq 1]\}$$

это Π_1-полное множество.

(b) θ*Конечное* $= \{i \mid M_i$ допускает конечное множество входных последовательностей$\}$

$$= \{i \mid \exists x \forall y \forall t [y > x \Rightarrow T(i, y, t) \neq 1]\}$$

это Σ_2-полное множество.

(c) θ*Универсальное* $= \{i \mid M_i$ допускает все входные последовательности$\}$

$$= \{i \mid \forall x \exists t [T(i, x, t) = 1]\}$$

это Π_2-полное множество.

(d) θ*Кофинитное* $= \{i \mid M_i$ допускает все входные последовательности, кроме конечного числа их$\}$

$$= \{i \mid \exists x \forall y \exists t [y > x \Rightarrow T(i, y, t) = 1]\}$$

это Σ_3-полное множество.

При конструировании полных множеств более высокой степени встречаются большие трудности, и многие примеры, скажем пример Σ_5-полного множества, громоздки для описания. Более того, такие примеры имеют специфический вид и доказательство того, что они являются полными множествами данной степени, могут быть достаточно трудными.

Для преодоления этих и других трудностей мы вводим понятие полных относительных множеств арифметической иерархии. Это понятие естественно вытекает из понятий машин Тьюринга и чередующихся кванторов и будет использоваться позднее для доказательства результатов о списках машин.

Определение. n-м *допускаемым множеством* (обозначение A_n) называется множество

$$\{i \mid \forall x_1 \exists x_2 \ldots \forall x_{n-1} \exists x_n \; [M_i \text{ допускает последовательность } x_1 \# x_2 \# \ldots \# x_{n-1} \# x_n]\},$$

если n четно и

$$\{i \mid \exists x_1 \forall x_2 \ldots \forall x_{n-1} \exists x_n \; [M_i \text{ допускает входную последовательность } x_1 \# x_2 \# \ldots \# x_n]\},$$

если n нечетно (здесь x_i не содержит символа $\#$).

Определение. *n-м отклоняемым множеством* (обозначение R_n) называется множество

$$\{i \mid \exists x_1 \forall x_2 \ldots \forall x_n \; [M_i \text{ не допускает входную последовательность } x_1 \# \ldots \# x_n]\},$$

если n четно и

$$\{i \mid \forall x_1 \exists x_2 \ldots \forall x_n \; [M_i \text{ не допускает входную последовательность } x_1 \# \ldots \# x_n]\},$$

если n нечетно (как и прежде, x_i не содержит символа $\#$).

Например, множество A_3 содержит все индексы машин Тьюринга, для которых существует (бинарная) последовательность x, такая, что для каждой (бинарной) последовательности y существует последовательность z, такая, что машина допускает входную последовательность $x\#y\#z$. Очевидно, что $A_n = \bar{R}_n$, $R_n = \bar{A}_n$ и $R_1 = \theta$*Пустое*.

Поместим теперь таким образом определенные относительные множества в арифметическую иерархию и покажем, что они полны в соответствующей им степени неразрешимости.

Теорема (О полноте).

(а) *Если n нечетно, то множество A_n является Σ_n-полным, а множество R_n является Π_n-полным.*

(b) *Если n четно, то множество* A_n *является* Π_n*-полным, а множество* R_n *является* Σ_n*-полным.*

Доказательство. Будет доказана только часть (b) для A_n, так как доказательства остальных частей аналогичны.

Во-первых, справедливо включение $A_n \in \Pi_n$, так как отношение $i \in A_n$ можно представить как

$$\forall x_1 \exists x_2 \ldots \exists x_n \exists t \{T(i, x_1 \# \ldots \# x_n, t) = 1\}.$$

Далее рассмотрим произвольное множество $B \in \Pi_n$. Это множество можно представить как

$$B = \{z \mid \forall x_1 \exists x_2 \ldots \exists x_n R(x_1, \ldots, x_n, z)\}$$

для некоторого рекурсивного предиката R.

Рассмотрим машину Тьюринга $M_{g(z)}$, которая допускает входную последовательность вида $x_1 \# \ldots \# x_n$ тогда и только тогда, когда значение предиката $R(x_1, \ldots, x_n, z)$ есть истина. Далее

$$z \in B \Leftrightarrow \forall x_1 \exists x_2 \ldots \exists x_n R(x_1, \ldots, x_n, z)$$

$$\Leftrightarrow \forall x_1 \exists x_2 \ldots \exists x_n [x_1 \# \ldots \# x_n \in W_{g(z)}]$$

$$\Leftrightarrow g(z) \in A_n.$$

Функция g является рекурсивной (по тезису Черча) и взаимно однозначной, так как новая машина $M_{g(z)}$ строится для каждого значения z. Поэтому $B \leqslant_1 A_n$ и множество A_n является Π_n-полным.

Этот результат показывает, что мы можем определить расположение наших относительных множеств на каждой степени арифметической иерархии, и поэтому они являются эквивалентными тем относительным множествам, которые используются в теории рекурсивных функций. (Это множества $\varphi^{(n)}$ и $\overline{\varphi^{(n)}}$, описанные в [11].) Отсюда следуют все хорошо известные результаты, такие, например, как теорема о иерархии для A_n и R_n. Также верными являются стандартные отношения сводимости $A_n \leqslant_1 A_{n+1}$, $R_n \leqslant_1 R_{n+1}$, $A_n \leqslant_1 R_{n+1}$ и $R_n \leqslant_1 A_{n+1}$. Относительные множества можно использовать для определения минимальной степени неразрешимости проблем теории автоматов.

Например, легко показать, что θ*Универсальное* — это хорошо известное Π_2-полное множество — эквивалентно множеству A_2. Для этого достаточно свести множество θ*Универсальное* к множеству A_2 с помощью взаимно однозначной рекурсивной функции g, которая определяется так:

$M_{g(i)}$ допускает входную последовательность $x \# y$ тогда и только тогда, когда $T(i, x, y) = 1$.

Сведение множества A_2 к множеству θ*Универсальное* достигается с помощью функции f:

$M_{f(i)}$ допускает последовательность x тогда и только тогда, когда $\exists y \exists t T(i, x \# y, t) = 1$.

(Другие применения относительных множеств появятся в следующих разделах.)

3. СПИСКИ.

Вопросы о списках часто встречаются в теории автоматов, и многие проблемы, первоначально имеющие другую формулировку, могут быть лучше поняты, если они представлены как проблемы о списках.

В этом разделе мы установим несколько результатов, которые для некоторого свойства P характеризуют степень неразрешимости рекурсивно инвариантных проблем о списках машин в терминах множества, ассоциированного с P и структурой проблемы.

Определение. *Список* — это рекурсивно перечислимое множество. (Как прежде, W_i употребляется для обозначения множества, допускаемого машиной M_i.)

Определение. Свойство P называется *рекурсивно инвариантным* тогда и только тогда, когда оно сохраняется рекурсивным изоморфизмом или рекурсивной взаимно однозначной функцией на. (То есть, если множество A обладает свойством P и g — взаимно однозначная рекурсивная функция на, то $g(A)$ также обладает свойством P.)

Отметим, что обычные количественные свойства «множество A пусто», «множество A конечно», «множество A бесконечно» и «множество A конфинитно» рекурсивно инвариантны.

В дальнейшем большинство рассматриваемых списков будут состоять из индексов машин Тьюринга. Вопросы, которые мы будем рассматривать относительно списков, — это вопросы о пересечении списка и некоторого, хорошо известного семейства машин (или индексного множества семейства).

Следующая лемма показывает, что в рекурсивно инвариантных свойствах пересечения множеств со списком можно заменить любое полное множество соответствующим ему относительным множеством.

Лемма. *Если P — рекурсивно инвариантное свойство и $B \equiv {}_1A_n$, то*

$$C = \{i \mid P(A_n \cap W_i) = 1\} \equiv_1 \{i \mid P(B \cap W_i) = 1\} = D.$$

Доказательство. (а) Пусть соотношение $B \leqslant {}_1A_n$ установлено при помощи взаимно однозначной рекурсивной функции

g; определим функцию f, такую, что $W_{f(i)} = g(W_i)$. (Она легко может быть получена из функции g.) Смысл функций f и g иллюстрируется на рис. 1.

Так как функция g взаимно однозначна и функция на, она сохраняет рекурсивно инвариантные свойства (по теореме Майхилла [7]). Таким образом,

$$P(B \cap W_i) = P(g(B \cap W_i)) = P(A_n \cap W_{f(i)}),$$

и поэтому

$$\begin{aligned} i \in D &\Leftrightarrow P(B \cap W_i) = 1 \\ &\Leftrightarrow P(A_n \cap W_{f(i)}) = 1 \\ &\Leftrightarrow f(i) \in C. \end{aligned}$$

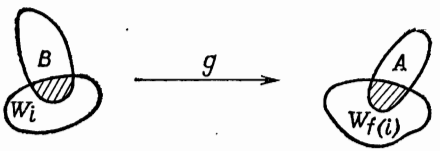


Рис. 1.

Следовательно, соотношение $D \leqslant_1 C$ устанавливается с помощью функции f.

(b) Соотношение $C \leqslant_1 D$ можно доказать аналогично.

Эта лемма верна также и для множеств, эквивалентных множеству R_n, и доказывается таким же образом.

Следующий результат показывает, как увеличивается трудность проблемы, когда мы задаем вопрос о выполнимости этой проблемы на списке машин Тьюринга. В силу предыдущей леммы можно формулировать вопросы относительно множеств A_n и R_n, а затем переносить результаты на любые эквивалентные проблемы.

Две приводимые ниже теоремы демонстрируют два метода доказательства. Первый из них является методом сведения, базирующемся на построении машины, в то время как второй является более изящным доказательством, основанным на результате Крейзела, Шёнфилда и Вана. Здесь даются оба типа доказательств для демонстрации обоих методов.

Теорема. *Если множество A является Σ_n-полным, то множество*

$$\{i \mid A \cap W_i = W_i\} = \{i \mid W_i \subseteq A\}$$

является Π_{n+1}-полным.

Доказательство. В силу предыдущей леммы вместо множества A можно использовать множество A_n. Поэтому пусть $B = \{i \mid A_n \cap W_i = W_i\}$.

(a) Множество B является Π_{n+1}-множеством, так как

$$\begin{aligned} i \in B &\Leftrightarrow A_n \cap W_i = W_i \Leftrightarrow \forall x\,[x \in W_i \Rightarrow x \in A_n] \\ &\Leftrightarrow \forall x\,[x \notin W_i \text{ или } x \in A_n] \\ &\Leftrightarrow \forall x\,[\forall t\,[T(i, x, t) \neq 1] \text{ или } x \in A_n] \\ &\Leftrightarrow \forall x \forall t\,[T(i, x, t) \neq 1 \text{ или } x \in A_n]. \end{aligned}$$

Так как $A_n \in \Sigma_n$, то $B \in \Pi_{n+1}$, и поэтому $B \leqslant_1 A_{n+1}$.

(b) Отметим, что

$$A_n = \{i \mid \exists x_1 \ldots \exists x_n\,[x_1 \# x_2 \# \ldots \# x_n \in W_i]\}$$

и

$$A_{n+1} = \{i \mid \forall m \exists x_1 \ldots \exists x_n\,[m \# x_2 \# \ldots \# x_n \in W_i]\}.$$

Для любой машины Тьюринга M_i построим список индексов машин $W_{g(i)} = \{k_0, k_1, \ldots\}$ следующим образом: машина M_{k_m} допускает лишь входные последовательности вида $x_1 \# x_2 \# \ldots \# x_n$. Если машина M_{k_m} получает на входе последовательность $x_1 \# \# x_2 \# \ldots x_n$, она ставит выражение $m \#$ в начале этой последовательности и перерабатывает последовательность $m \# x_1 \# \ldots \ldots \# x_n$ так же, как ее перерабатывала бы машина M_i.

Индексы k_m машин и их список $W_{g(i)}$ легко определяется в терминах машины M_i.

Интуитивно ясно, что если $i \in A_{n+1}$, то все индексы k_m являются элементами множества A_n. Более формально:

$$\begin{aligned} i \in A_{n+1} &\Leftrightarrow \forall m \exists x_1 \ldots \exists x_n\,[m \# x_1 \# \ldots \# x_n \in W_i] \\ &\Leftrightarrow \forall m \exists x_1 \ldots \exists x_n\,[x_1 \# \ldots \# x_n \in W_{k_m}] \\ &\Leftrightarrow \forall m\,[k_m \in A_n] \\ &\Leftrightarrow A_n \cap W_{g(i)} = W_{g(i)} \\ &\Leftrightarrow g(i) \in B. \end{aligned}$$

Следовательно, $A_{n+1} \leqslant B$.

Аналогичная теорема может быть доказана и для множества $R_n \in \Sigma_n$.

Теперь множества более высокой степени можно естественным образом строить из множеств относительно низких степеней. Для любого Σ_n-полного множества A множество списков машин Тьюринга (в действительности их индексов), которые содержат только элементы множества A, является Π_{n+1}-полным. Приведем несколько примеров (следствий) к доказанной выше теореме:

(a) Множество списков (индексов) машин Тьюринга, которые допускают конечное множество входных последовательностей

$$\{i \mid W_i \cap \theta\mathit{Конечное} = W_i\},$$

является Π_3-полным.

(b) Множество списков машин Тьюринга, которые допускают кофинитные множества входных последовательностей

$$\{i \mid W_i \cap \theta \textit{Кофинитное} = W_i\},$$

является Π_4-полным.

Доказанная выше теорема не может быть обобщена на Π_n-полные множества. Оказывается, что множество списков, которые содержат только элементы из Π_n-полного множества A, вновь является Π_n-полным множеством. Этот факт устанавливается из анализа структуры множества

$$B = \{i \mid W_i \cap R_n = W_i\},$$

если $R_n \in \Pi_n$:

$$\begin{aligned} i \in B &\Leftrightarrow W_i \cap R_n = W_i \\ &\Leftrightarrow \forall x\,[x \in W_i \Rightarrow x \in R_n] \\ &\Leftrightarrow \forall x \forall t\,[T(i, x, t) \neq 1 \text{ или } x \in R_n] \end{aligned}$$

Так как отношение $X \in R_n$ является Π_n-предикатом, то последний предикат в приведенной эквивалентности является Π_n-предикатом.

Другое свойство списков позволяет порождать множества арифметической иерархии, лежащие на два уровня выше первоначального множества. Это достигается при решении такой проблемы: если некоторый список содержит бесконечное число элементов из некоторого множества, верно ли, что все элементы списка являются элементами этого множества.

Теорема. *Если множество A является Π_n-полным, то множество $\{i \mid A \cap W_i$ — бесконечно$\}$ является Π_{n+2}-полным.*

Доказательство. Как и прежде, пусть $B = \{i \mid A \cap W_i$ — *бесконечно*$\}$

(a) $B \in \Pi_{n+2}$, так как

$$\begin{aligned} i \in B &\Leftrightarrow A \cap W_i \text{ бесконечно} \\ &\Leftrightarrow \forall x \exists y\,[y > x \wedge y \in W_i \wedge y \in A] \\ &\Leftrightarrow \forall x \exists y \exists t\,[y > x \wedge T(i, y, t) = 1 \wedge y \in A] \\ &\Leftrightarrow \forall \exists \exists\,[y \in A \wedge (\text{рекурсивный предикат})]. \end{aligned}$$

В силу того, что $y \in A$ есть Π_n-предикат, выполнено соотношение $B \in \Pi_{n+2}$.

(b) Для доказательства этой части используется лемма Крейзела, Шёнфилда и Вана [6], касающаяся квантора Ux, который

обозначает «существует бесконечно много x». Этот результат говорит о том, что любой Π_{2n}-предикат можно представить в виде

$$Ux_1 \ldots Ux_n R(x_1, \ldots, x_n, y),$$

где R — некоторый рекурсивный предикат, а любой Π_{2n+1}-предикат можно представить в виде

$$Ux_1 \ldots Ux_n \forall x_{n+1} R(x_1, \ldots, x_{n+1}, y).$$

Пусть теперь C — любое Π_{n+2}-множество; тогда из приведенных выше рассуждений следует, что его можно представить в виде

$$x \in C \Leftrightarrow U_y S(x, y)$$

для некоторого Π_n-предиката S. Так как множество A — это Π_n-множество, то существует рекурсивная, взаимно однозначная функция f, такая, что

$$S(x, y) \Leftrightarrow f(x, y) \in A$$

Положим $M_{g(x)}(y) = f(x, y)$, и пусть множество $W_{h(x)}$ будет областью определения машины $M_{g(x)}$. Тогда

$$\begin{aligned} x \in C &\Leftrightarrow UyS(x, y) \\ &\Leftrightarrow Uy[f(x, y) \in A] \\ &\Leftrightarrow W_{h(x)} \cap A \text{ бесконечно} \\ &\Leftrightarrow h(x) \in B. \end{aligned}$$

Таким образом, из $C \leqslant_1 B$ следует, что множество B является Π_{n+2}-полным.

С помощью этого результата можно строить множества, уровень которых в арифметической иерархии на два больше, чем уровень первоначального множества.

Приведем несколько непосредственно получающихся примеров.

(a) Множество списков машин Тьюринга, в которых бесконечное число машин не допускает никакую входную последовательность:

$$\{i \mid W_i \cap \theta \textit{Пустое} \text{ бесконечно}\}$$

является Π_3-полным множеством.

(b) Множество списков машин Тьюринга, в которых бесконечное число машин допускает любые входные последовательности

$$\{i \mid W_i \cap \theta \textit{Универсальное} \text{ бесконечно}\},$$

является Π_4-полным множеством.

(c) Множество списков машин Тьюринга, в которых бесконечное число машин допускает кофинитное множество входных последовательностей

$$\{i \mid W_i \cap \theta \textit{Кофинитное} \text{ бесконечно}\},$$

является, как и предполагалось, Π_5-полным множеством.

Отметим, что когда эта процедура применяется к Σ_n-полным множествам, то новое множество получается уже только уровня Π_{n+1}. Это доказывается, как и прежде, с помощью представления предиката для нового множества. Предположив, что $A \in \Sigma_n$, и просмотрев часть (а) предыдущего доказательства, мы получим выражение

$$\forall\exists\exists [y \in A \wedge \ldots].$$

Так как $A \in \Sigma_n$, то это, очевидно, Π_{n+1}-предикат, и следовательно, может быть получено повышение только на единицу

Существует много естественных свойств списков (такие же, как два вышеупомянутых), и эти свойства можно задать множествами, как, например:

$$\{i \mid P(A, W_i) = 1\},$$

где $P(A, W_i)$ является отношением вида: «в списке W_i имеется элемент из множества A» или «в списке W_i имеется конечное число элементов из множества A». Некоторые из таких общих отношений собраны в приведенной ниже таблице. Доказательства для них получаются простым применением методов, используемых в предыдущих доказательствах.

Таблица

$P(A, W_i)$	$A-\Sigma_n$-полно	$A-\Pi_n$-полно
$A \cap W_i = \varphi$	Π_n-полно	Π_{n+1}-полно
$A \cap W_i \neq \varphi$	Σ_n-полно	Σ_{n+1}-полно
$A \cap W_i = W_i$	Π_{n+1}-полно	Π_n-полно
$A \cap W_i \neq W_i$	Σ_{n+1}-полно	Σ_n-полно
$A \cap W_i$ конечно	Σ_{n+1}-полно	Σ_{n+2}-полно
$A \cap W_i$ бесконечно	Π_{n+1}-полно	Π_{n+2}-полно
$\overline{A} \cap W_i$ конечно	Σ_{n+2}-полно	Σ_{n+1}-полно
$\overline{A} \cap W_i$ бесконечно	Π_{n+2}-полно	Π_{n+1}-полно

4. ДАЛЬНЕЙШИЕ ПРИМЕНЕНИЯ

Свойства списков, которые отмечены в приведенной выше таблице, упрощают технику нахождения нижних границ множества в степени неразрешимости. Например, вместо предварительно применимого сложного аппарата, как в [11], все, что теперь требуется

для того, чтобы показать, что множество θ*Кофинитное* является Σ_3-полным множеством, это простое сведение множества $\{i \mid W_i \cap \theta$*Пустое* конечно$\}$.

Приведем две теоремы, иллюстрирующие эту технику.

Теорема. *Множество* θ*Конечное* $= \{\mid W_i$ *конечно*$\}$ *является* Σ_2*-полным множеством.*

Доказательство. (а) Запишем отношение θ*Конечное* в префиксной нормальной форме.

$$\begin{aligned} i \in \theta\textit{Конечное} &\Leftrightarrow \exists x \forall y\,[y > x \Rightarrow y \notin W_i] \\ &\Leftrightarrow \exists x \forall y\,[y \leqslant x \vee y \notin W_i] \\ &\Leftrightarrow \exists x \forall y\,[y \leqslant x \vee \forall t T(i, y, t) \neq 1] \\ &\Leftrightarrow \exists \forall \forall \text{ [рекурсивный предикат].} \end{aligned}$$

Таким образом, множество θ*Конечное* является Σ_2-множеством и поэтому по теореме о полноте θ*Конечное* $\leqslant_1 R_2$.

(b) Покажем теперь, что $R_2 \leqslant_1 \theta$*Конечное*. Рассмотрим множество $B = \{i \mid W_i \cap A_1$ конечно$\}$, которое, как следует из таблицы свойств списков, является Σ_2-полным, и поэтому $B \equiv_1 R_2$. (A_1 является дополнением к множеству θ*Пустое*). Сведение множества B к множеству θ*Конечное* происходит следующим образом: по машине Тьюринга M_i строится другая машина $M_{g(i)}$, такая, что всякий раз, когда машина M_i допускает входную последовательность, в которой только конечное число элементов не является индексами для пустого множества, то $M_{g(i)}$ допускает конечное множество входных последовательностей. Более формально, пусть

$$W_i = \{a_0, a_1, \ldots\};$$

рассмотрим машину

$$M_{g(i)}(k) = \begin{cases} \text{останавливается, если } \exists a_k \in W_i \text{ и } \exists x \\ \qquad\qquad [M_{a_k}(x) \text{ останавливается}], \\ \text{не останавливается в противном случае.} \end{cases}$$

Поэтому всякий раз, когда a_k является индексом для пустого множества (или когда W_i конечно и не существует индекса a_k), тогда $M_{g(i)}(k)$ не останавливается. Однако, когда индекс a_k существует и является элементом множества A_1, тогда $M_{g(i)}(k)$ останавливается. Таким образом,

$$\begin{aligned} i \in B &\Leftrightarrow W_i \cap A_1 \text{ конечно} \\ &\Leftrightarrow \exists \text{ конечное } k, \text{ для которого } M_{g(i)}(k) \text{ останавливается} \\ &\Leftrightarrow g(i) \in \theta\textit{Конечное}. \end{aligned}$$

Следовательно, так как множество A_1 содержит все индексы для непустых множеств, если $W_i \cap A_1$ — конечно, то все, кроме конеч-

ного числа элементов множества W_i, являются индексами для пустого множества. Поэтому машина $M_{g(i)}$ не останавливается для всех, кроме конечного числа входных последовательностей.

Теорема. *Множество* θ*Кофинитное является* Σ_3*-полным множеством.*

Доказательство. (а) Во-первых, рассмотрим префиксную нормальную форму

$$\begin{aligned} i \in \theta\textit{Кофинитное} &\Leftrightarrow \exists x \forall y\,[y > x \Rightarrow y \in W_i] \\ &\Leftrightarrow \exists x \forall y\,[y \leqslant x \vee y \in W_i] \\ &\Leftrightarrow \exists x \forall y\,[y \leqslant x \vee \exists t\,(T(i, y, t) = 1)] \\ &\Leftrightarrow \exists \forall \exists\ [\text{рекурсивный предикат}], \end{aligned}$$

и поэтому θ*Кофинитное* $\in \Sigma_3$ и θ*Кофинитное* $\leqslant_1 A_3$.

(b) Следующая часть — сведение множества $B = \{i \mid W_i \cap \theta$*Пустое* конечно$\}$ (которое является Σ_3-полным) к множеству θ*Кофинитное*.

Рассмотрим любую машину Тьюринга M_i, пусть $W_i = \{a_0, a_1, \ldots\}$ употребляется в обычном своем значении; определим машину $M_{g(i)}$ следующим образом:

$$M_{g(i)}(k) = \begin{cases} \text{останавливается, если } \exists x\ [Ma_k(x) \text{ останавливается}], \\ \text{не останавливается в противном случае.} \end{cases}$$

Если a_k — индекс пустого множества, то $M_{g(i)}(k)$ не останавливается, и если только конечное число элементов a_i находятся в множестве θ*Пустое*, то $M_{g(i)}$ не останавливается для конечного числа входных последовательностей. Или более формально:

$$\begin{aligned} i \in B &\Leftrightarrow W_i \cap \theta\textit{Пустое} \text{ конечно} \\ &\Leftrightarrow W_{a_k} = \varphi \text{ для конечного набора значений } k \\ &\Leftrightarrow k \notin W_{g(i)} \text{ для конечного набора значений } k \\ &\Leftrightarrow g(i) \in \theta\textit{Кофинитное}. \end{aligned}$$

Таким образом, $B_1 \leqslant_1 \theta$*Кофинитное* и θ*Кофинитное* является Σ_3-полным множеством.

Все приведенные выше доказательства достаточно прямые, и в них используется одна и та же конструкция. При правильном выборе множества B это более продуктивный способ доказательства по сравнению с доказательствами такого типа.

Методика, развитая в этой работе, имеет различные применения в теории автоматов. Мы закончим обсуждение иллюстрацией того, как такой подход можно использовать при решении проблем о рекурсивно перечислимых списках контекстно-свободных (к. с.) грамматик. Проблемы такого типа изучались недавно Кудиа [1].

С помощью развитой в [3] методики можно легко показать, что

(a) Множество неоднозначных контекстно-свободных грамматик является Σ_1-полным.

(b) Множество контекстно-свободных грамматик, которые порождают регулярные языки (или кофинитные множества), является Σ_2-полным.

Из этого немедленно следует, что множество всех рекурсивно перечислимых списков, которые содержат к. с. грамматики, порождающие нерегулярные множества, является Σ_3-полным.

Множество рекурсивно перечислимых списков к. с. грамматик, в которых каждый список содержит бесконечно много грамматик, порождающих регулярные множества, является Π_3-полным. С другой стороны, множество списков к. с. грамматик, в которых каждый список содержит бесконечно много грамматик, порождающих нерегулярные множества, является Π_4-полным.

Множество рекурсивно перечислимых списков к. с. грамматик, которые содержат только конечное число неопределенных грамматик, является Σ_2-полным, с другой стороны, множество списков к. с. грамматик, которое содержит бесконечно много неопределенных грамматик, является Π_3-полным.

Большинство результатов о тьюринговой эквивалентности неразрешимых проблем, о списках контекстно-свободных грамматик выпадают из рассмотрений такого типа, и существующие различия легко объясняются видом последовательности кванторов, описывающих рассматриваемые множества.

СПИСОК ЛИТЕРАТУРЫ

1. Cudia D. F., The degree hierarchy of undecidable problems of formal grammars, in 2nd Ann. ACM. Symp. on Theory of Comput., pp. 10—21, ACM, New York, 1970.
2. Davis M., Computability and Unsolvability, McGraw-Hill, New York, 1958.
3. Hartmanis J., Context free languages and Turing machine computations, *Proc. Symp. Appl. Math.*, **19** (1967), 42—51.
4. Kleene S. C., Recursive predicates and quantifiers, *Trans. Amer. Math. Soc.*, **53** (1943), 41—73.
5. Kleene S. C., Introduction to Metamathematics, Van Nostrand, Princeton, N. J., 1952. (Русский перевод: Клини С. К., Введение в метаматематику, М., ИЛ, 1957.)
6. Kreisel G., Shoenfield J. R., Wang H., Number theoretic concepts and recursive well-orderings, *Arch. Math. Logik Grundlagenforsh*, **5** (1960), 42—64.
7. Myhill J., Creative sets, *Z. Math. Logik Grundlagen Math.*, **1** (1955), 97—108.
8. Post E. L., Recursively enumerable sets of positive integers and their decision problems, *Bull. Amer. Math. Soc.*, **50** (1944), 284—316.
9. Rogers H., Jr., Computing degrees of unsolvability, *Math. Ann.*, **138** (1959), 125—140.
0. Rogers H., Jr., Gödel numbering of partial recursive functions, *J. Symbolic Logic*, **23** (1958), 331—341.
1. Rogers H., Jr., Theory of Recursive Functions and Effective Computability, McGraw-Hill, New York, 1967. (Русский перевод: Роджерс Х., Теория рекурсивных функций и эффективная вычислимость, М., «Мир», 1972.)

Обзор теории сложности вычислений [1)]

Дж. Хартманис и Дж. Э. Хопкрофт

1. ВВЕДЕНИЕ

Очевидно, что жизнеспособная теория вычислений должна реалистически рассматривать количественные аспекты вычислений и содержать общую теорию, в которой изучаются свойства возможных мер трудности вычисления функций. Такая теория должна идти дальше классификации функций как вычислимых и невычислимых или элементарных и примитивно рекурсивных и т. д. Она должна рассматривать меры сложности вычислений, которые определены для всевозможных вычислений и которые приписывают сложность каждому заканчивающемуся вычислению. Более того, эта теория должна в конечном счете отражать некоторые аспекты реальных вычислений, чтобы оправдать себя соответствующим вкладом в общее развитие науки о вычислительных процессах. В течение последнего десятилетия в развитии такой теории, касающейся сложности вычислений, был сделан значительный прогресс. Мы убеждены, что в настоящее время эта теория является существенной частью теории вычислений и что в будущем она станет важной теорией, которая глубоко проникнет в теоретические работы в области науки о вычислительных процессах.

Цель этой статьи — дать представление о развитии теории сложности вычислений за последнее время, изложив ее главные понятия, результаты и методы. Статья касается прежде всего изучения мер сложности вычислений, определенных для всех вычислимых частичных функций. Мы не пытались ни дать исчерпывающий обзор всей области, ни представить материал в исторической последовательности. Скорее, мы сосредоточили внимание на демонстрации тех результатов и методов, которые нам кажутся важными, и мы хотим представить их с такой точки зрения, с которой их легче всего понять. Эта статья содержит некоторым образом то, что, как мы полагаем, должен знать о сложности вычислений каждый, кто работает в области науки о вычислительных процессах (или по крайней мере в области теории вычислений). С другой стороны, тот кто хочет заниматься дальнейшими исследованиями в этой области, должен прочесть значительно больше. В частности, он должен изучить результаты о специфических мерах сложности

[1)] Hartmanis J., Hopcroft J. E., An overview of the theory of computational complexity, *J. Assoc. Comp. Mach.*, 18 (1971), № 3, 444—475.

и отношениях между мерами, которые мотивировали основы общего подхода и которые остаются источником идей и контрпримеров.

Следует подчеркнуть, что этот обзор не включает очень интересные недавние результаты, в которых устанавливаются верхние и нижние границы сложности вычисления отдельных функций. Это важная область исследований по сложности вычислений, имеющая, по-видимому, и серьезное практическое значение. В то же время мы убеждены, что эта область очень быстро развивается и сейчас еще преждевременно пытаться дать полный обзор.

Во втором разделе статьи дается мотивировка определения мер сложности вычислений и приводится несколько примеров. Затем выводятся некоторые основные свойства, которые выполняются для всех мер сложности. Показывается, например, что для каждой меры сложности существуют сколь угодно сложные функции, принимающие только значения 0, 1, и что не существует рекурсивного отношения между величиной функции и ее сложностью. С другой стороны, показано, что любые две меры сложности рекурсивно ограничивают друг друга. В разд. 3 мы даем новое доказательство довольно неожиданного результата, согласно которому для любой меры сложности существуют функции, вычисление которых можно сколь угодно сильно ускорить, беря все более и более эффективные алгоритмы. Позднее будет показано, что это верно не для всякой рекурсивной функции. Наше доказательство основано на прямой диагональной процедуре и не опирается на теорему о рекурсии, которая использовалась в первоначальном доказательстве. Добиться этого можно, заметив, что теорема об ускорении не зависит от меры сложности (т. е. если она выполняется для какой-нибудь меры, то выполняется и для всех мер), и затем доказывая ее непосредственно для очень понятной меры сложности — емкости ленты при вычислении на машине Тьюринга. Для этой меры сложности трудность доказательства значительно уменьшается по сравнению с первоначальными трудностями. Теорема об ускорении имеет странное следствие: какие бы две универсальные машины мы ни взяли (неважно, насколько более быстрой и мощной является одна из них), существуют такие функции, которые нельзя вычислить быстрее на более мощной машине. Это следует из того, что для любого алгоритма, который мы используем для вычисления такой функции на более мощной машине, существует другой алгоритм, настолько быстрый, что даже на медленной машине он выполняется быстрее, чем первый алгоритм на быстрой машине.

Таким образом, существуют функции, которые не имеют наилучших программ, и так как мы не можем классифицировать функции по их минимальным программам, то мы обращаемся к изучению классов функций, сложность вычислений которых ограничена

некоторой рекурсивной функцией. Мы покажем, во-первых, что для любого класса сложности, сложность которого ограничена рекурсивной функцией f, можно эффективно построить строго больший класс, сложность которого задается рекурсивной функцией от сложности функции f (т. е. от времени вычисления f). Следующий результат, теорема о пробелах, утверждает, что это наилучший возможный результат, который мы можем получить, так как в ней показано, что существуют сколь угодно широкие пробелы между классами сложности. Именно, для каждой рекурсивной функции r существует возрастающая рекурсивная функция t, такая, что класс всех функций, вычислимых с границей сложности t, совпадает с классом функций, вычислимых с границей сложности $r \circ t$ [1]).

Таким образом, мы не всегда можем получить более широкие классы сложности, применяя рекурсивную функцию к старой границе сложности. Этот результат имеет еще такое интересное следствие. Когда мы рассматриваем универсальную вычислительную машину, то не важно, как сильно мы увеличиваем скорость вычисления и как много новых операций добавляем; все равно существует бесконечно много рекурсивных границ сложности, для которых старая и новая машины будут вычислять в точности те же самые функции. То есть в пределах бесконечного числа границ сложности нельзя получить никакого преимущества новой машины над старой, увеличивая скорость и вычислительную мощь машины. За обсуждением этого результата следует другой удивительный результат, который показывает, что аксиомы сложности допускают меры сложности с классами сложности, которые могут не быть рекурсивно перечислимыми. К счастью, для широких классов сложности эта ситуация не может иметь места. Показано, что для любой меры все достаточно широкие классы сложности являются рекурсивно перечислимыми.

В разд. 5 мы более детально рассматриваем процедуру построения новых классов сложности посредством диагонального процесса. Мы предлагаем новый подход, который позволяет разбить «цену диагонализации» любого класса сложности на «цену моделирования» и «цену параллельных вычислений». Из этой общей формулировки мы можем извлечь результаты о классах сложности для специальных мер, если мы знаем, насколько трудно при данной мере моделировать вычисления и выполнять их параллельно. Это иллюстрируется на трех довольно несхожих результатах для классов сложности вычислений на машинах Тьюринга с ограничением на емкость ленты, а также для вычислений на одноленточных и многоленточных машинах Тьюринга с ограничением на время вычисления. В каждом случае различия в структуре

---

[1]) $r \circ t$. — суперпозиция функций r и t, т. е. функция $f(x) = r(t(x))$. — *Прим. перев.*

результата прослеживаются на различиях в трудности моделирования и параллелизации вычислений для трех различных мер сложности.

В разд. 6 мы рассматриваем проблему «именования» классов сложности. Во-первых, мы доказываем теорему об объединении, которая утверждает, что объединение любой рекурсивно перечислимой последовательности возрастающих классов сложности снова является классом сложности. Отсюда следует, что многие изучавшиеся ранее подклассы рекурсивных функций естественным образом подходят под многие меры сложности. Например, существует рекурсивная граница $L(n)$ емкости ленты, такая, что класс функций, вычислимых на машинах Тьюринга, у которых длина ленты ограничена функцией $L(n)$, состоит точно из примитивно рекурсивных функций. Второй главный результат этой части — теорема об именовании — несколько ослабляет остроту теоремы о пробелах, показывая, что для любой меры существует (измеримое) множество функций, которые именуют все классы сложности, не оставляя сколь угодно широкие пробелы вверх. К сожалению, оказывается, что именование классов сложности может иметь сколь угодно широкие пробелы вниз.

Теорема об именовании является довольно техническим результатом, а ее доказательство — все еще весьма трудно. Читатель, возможно, захочет пропустить это доказательство и перейти к следующей части.

Мы заканчиваем этот обзор некоторыми рассуждениями (в разд. 7) об объеме алгоритмов (машин) для того, чтобы извлечь понятие о сложности описания алгоритма. Сначала мы даем формальное определение меры объема, а затем показываем, что между любыми двумя такими мерами существует рекурсивное отношение. Главный результат этой части показывает, что в любом рекурсивно перечислимом списке алгоритмов имеются сколь угодно неэффективные описания. Этот результат используется затем при рассмотрении экономности формализмов, представляющих различные алгоритмы. Например, показано, что если мы используем примитивно рекурсивные схемы для представления примитивно рекурсивных функций, то в некоторых случаях это описание примитивно рекурсивных функций очень неэффективно. Даже среди кратчайших программ, построенных из таких схем, мы можем найти программы, которые можно сократить в любое желаемое число раз, переходя к схемам общей рекурсии. Это означает, что, хотя для вычисления примитивно рекурсивных функций нам не нужны условные операторы и операторы перехода, их использование позволяет значительно сократить длину наших программ, и тем самым оно выявляет их важность для языков программирования.

В последней части очень кратко излагается история исследований, описанных в этой статье, и делается попытка указать тех,

кому принадлежат первоначальные работы. Мы включили также короткий рекомендательный список литературы для дальнейшего чтения.

2. МЕРЫ СЛОЖНОСТИ ВЫЧИСЛЕНИЙ

Теория сложности вычислений касается измерения трудности вычислений. Чтобы перейти к этому, мы должны обсудить, что понимается под мерой сложности вычислений.

В этой статье мы рассматриваем меры сложности вычислений, которые определены для всевозможных вычислений, т. е. для всех частично рекурсивных функций, отображающих натуральные числа в натуральные числа. Следовательно, чтобы определить некоторую меру сложности, нам нужен эффективный способ задания всевозможных вычислений или алгоритмов (для вычисления частично рекурсивных функций), а мера сложности тогда будет показывать, сколько «шагов» требуется любому из этих алгоритмов, чтобы по данному значению аргумента вычислить соответствующее значение функции.

Например, нашим списком алгоритмов или вычислительных устройств мог бы быть перечень всех одноленточных машин Тьюринга (которые, как мы знаем, в состоянии вычислить любую частично рекурсивную функцию), а мерой сложности для данной машины M_i (или алгоритма), работающей на аргументе n, могло бы быть число операций, которое она выполнит, прежде чем остановится.

Другая мера сложности получается, когда мы рассматриваем рекурсивный перечень всех программ, написанных на Алголе, и опять определяем сложность i-й программы на аргументе n числом команд, которые выполняются до того, как программа заканчивает работу.

Следует заметить, что эти меры сложности ассоциируются с алгоритмами, а не прямо с теми функциями, которые вычисляются алгоритмами. Основанием для этого служит то, что при вычислениях мы обычно имеем дело с алгоритмами, определяющими функции, и для каждой вычислимой функции существует бесконечно много алгоритмов, которые ее вычисляют. Кроме того, как будет позднее показано, существуют функции, у которых нет «наилучшего» алгоритма, поэтому мы не можем говорить о сложности функции как о сложности наилучшего вычисляющего алгоритма.

Из предыдущих примеров мы видим, что мера сложности состоит из рекурсивного списка алгоритмов, вычисляющих все частично рекурсивные функции, каждому из которых приписана

сигнализирующая функция[1]), дающая количество «средств», использованных данным алгоритмом на данном аргументе. Приписывание сигнализирующих функций, кроме того, удовлетворяет некоторым условиям. Если наш список алгоритмов обозначить через $\varphi_1, \varphi_2, \varphi_3, \ldots$, а соответствующие сигнализирующие функции — через $\Phi_1, \Phi_2, \Phi_3, \ldots$, то мы замечаем, что в наших примерах выполняются следующие два условия:

(1) алгоритм $\varphi_i(n)$ определен тогда и только тогда, когда $\Phi_i(n)$ определена;

(2) для каждого данного числа шагов m и каждого алгоритма φ_i, работающего на аргументе n, мы можем рекурсивно определить, заканчивается ли вычисление $\varphi_i(n)$ через m шагов, т. е. выполняется ли равенство $\Phi_i(n) = m$.

Другими словами, если i-я машина Тьюринга останавливается на входе n, то число шагов, которые она делает до остановки, вполне определено. С другой стороны, если i-я машина не останавливается на входе n, то мы не можем определить, насколько сложным является вычисление, так как мера не определена. Что мы можем сделать для любых i и n, так это определить, остановится ли i-машина на входе n через m шагов. Очевидно, что этого можно добиться в нашем первом примере, просто проделывая m шагов i-го вычисления на входе n и замечая, заканчивается ли вычисление на последнем шаге.

На меру сложности можно наложить дополнительные условия, чтобы более полно охватить некоторые специфические аспекты трудности вычисления, но установленные нами условия являются настолько естественными и основными для любого понятия сложности вычисления, что теперь общепринято, что они должны выполняться для любой меры сложности вычисления. Удивительный факт состоит в том, что они достаточны для доказательства многих интересных результатов о всех мерах сложности, для которых они выполняются. В остальной части статьи это будет некоторым образом проиллюстрировано, хотя мы рассмотрим и ряд специфических мер, чтобы расширить наши представления и проиллюстрировать некоторые специальные результаты.

В то же время следует также отметить, что многие специалисты в области применения вычислительных машин, возможно, гораздо более заинтересованы в результатах о специфических мерах сложности для специфических проблем. Тем не менее обрисованный выше подход достаточен, чтобы начать развивать общую теорию.

Теперь мы уточним понятие меры сложности вычисления. На протяжении всей статьи мы называем рекурсивной функцией всюду определенную вычислимую функцию, а слово «алгоритм» ис-

---

[1]) Этот термин употребляется в литературе на русском языке, в подлиннике — функция счета шагов (step-counting function). — *Прим. перев.*

пользуем для алгоритмической процедуры даже и тогда, когда она заканчивается не для всех значений аргументов.

Определение. *Мерой сложности вычисления*[1]) Φ называется допустимая нумерация[2]) частично рекурсивных функций $\varphi_1, \varphi_2, \varphi_3, \ldots$, которым сопоставлены частично рекурсивные *сигнализирующие функции* $\Phi_1, \Phi_2, \Phi_2, \ldots$, такие, что

(1) $\varphi_i(n)$ определена тогда и только тогда, когда $\Phi_i(n)$ определена.

$$(2) \qquad M(i, n, m) = \begin{cases} 0, & \text{если } \Phi_i(n) \neq m, \\ 1, & \text{если } \Phi_i(n) = m \end{cases}$$

есть рекурсивная функция.

Мы уже видели, что число тактов работы машины Тьюринга можно использовать в качестве сигнализирующей функции, чтобы получить меру сложности вычисления. Подобным же образом для определения меры можно использовать число ячеек ленты, прочитанных машиной Тьюринга (если мы условимся считать, что это число не определено, когда машина не останавливается). Фактически большинство других естественных мер сложности, с которыми приходится иметь дело, действительно удовлетворяют этому определению. Если дано некоторое множество сигнализирующих функций, то можно применить произвольную рекурсивную функцию $f(n) \geqslant n$ (или произвольную рекурсивную неограниченную монотонную функцию) к каждой сигнализирующей функции, чтобы получить новое множество сигнализирующих функций. Тем не менее будет показано, что определение мер сложности вычисления является достаточно ограничительным, чтобы исключить из множества сигнализирующих функций те функции, которые не имеют реального смысла меры сложности вычисления. Здесь поучительно рассмотреть несколько примеров, которые не являются мерами сложности.

(A) Число рекурсий, используемых для определения функции в схеме примитивной рекурсии, нельзя использовать в качестве сигнализирующих функций, так как этими схемами невозможно задать все частично рекурсивные функции, и, таким образом, мы не имеем допустимой нумерации всех алгоритмов.

(B) Функции $\{\Phi_i\}$, определенные равенством $\Phi_i(n) = 0$ для любых i и n, не удовлетворяют условию (1).

---

[1]) Употребителен еще термин «сигнализирующий оператор». — *Прим. перев.*

[2]) В американской литературе допустимой нумерацией называется вычислимая главная нумерация. Нумерация $\varphi_1, \varphi_2, \varphi_3, \ldots$, называется *вычислимой*, если существует частично рекурсивная («универсальная») функция $\varphi(i, n)$, такая, что для каждого фиксированного i функция $\varphi(i, n)$ совпадает с $\varphi_i(n)$. Она называется *главной*, если для любой вычислимой нумерации $\psi_1, \psi_2, \psi_3, \ldots$ существует рекурсивная («сводящая») функция f, такая, что для каждого i функция ψ_i совпадает с $\varphi_{f(i)}$. — *Прим. перев.*

(C) Функции $\{\Phi_i\}$, определенные соотношением

$$\Phi_i(n) = \begin{cases} 0, & \text{если } \varphi_i(n) \text{ определена,} \\ \text{не определена} & \text{в остальных случаях,} \end{cases}$$

не удовлетворяют условию (2), так как $\varphi_i(n)$ определена для каждых i и n тогда и только тогда, когда $M(i, n, 0) = 1$, и, таким образом, $M(i, n, m)$ не может быть рекурсивной (иначе мы могли бы решать проблему остановки).

Из определения меры сложности вычисления получается много результатов. Первый результат состоит в том, что для любой меры существуют сколь угодно сложные рекурсивные функции. Чтобы установить этот результат, мы покажем, что для любой рекурсивной функции f существует рекурсивная функция φ, обладающая тем свойством, что любой возможный способ вычисления φ требует более чем $f(n)$ шагов для бесконечно многих n. Чтобы построить φ, мы просто должны формализовать процедуру (диагональный процесс), которая с ростом n бесконечно часто обращается к каждому номеру i и полагает

$$\varphi(n) \neq \varphi_i(n), \quad \text{если} \quad \Phi_i(n) \leqslant f(n).$$

Замечание. Мы говорим, что i есть номер функции φ, если $\varphi_i(n) = \varphi(n)$ для всех n.

Теорема 1. *Пусть Φ — мера сложности вычислений и f — любая рекурсивная функция. Тогда существует рекурсивная функция φ, такая, что для любого номера i функции φ справедливо $\Phi_i(n) > f(n)$ для бесконечно многих n.*

Доказательство. Пусть $r(n)$ — рекурсивная функция, обладающая тем свойством, что для всех i, $i = 1, 2, 3, \ldots$, существует бесконечно много n, таких, что $r(n) = i$. Определим

$$\varphi(n) = \begin{cases} \varphi_{r(n)}(n) + 1, & \text{если} \quad \Phi_{r(n)}(n) \leqslant f(n), \\ 0 & \text{в противном случае.} \end{cases}$$

Так как f и r — рекурсивные функции, то (по второму условию, наложенному на меры сложности) мы можем вычислить, выполняется ли неравенство

$$\Phi_{r(n)}(n) \leqslant f(n),$$

поэтому $\varphi(n)$ — рекурсивная функция. Кроме того, если j есть номер φ, то для бесконечно многих n, таких, что $r(n) = j$, мы имеем $\Phi_j(n) > f(n)$, что и требовалось доказать.

Используя несколько более сложный диагональный процесс, мы теперь получим более сильный результат, который утверждает, что для любой рекурсивной f существуют рекурсивные функции,

сложность которых превышает f почти всюду [1]). Чтобы установить этот результат, мы просто формализуем утверждение: «если сложность $\Phi_i(n)$ i-й функции меньше, чем $f(n)$, для бесконечно многих n, то φ не является i-й функцией».

Теорема 2. *Пусть Φ — мера сложности. Тогда для любой рекурсивной функции f существует рекурсивная функция φ, такая, что для любого номера i функции φ*

$$\Phi_i(n) > f(n) \quad \textit{для почти всех} \quad n.$$

Доказательство. Пусть f — произвольная рекурсивная функция. Чтобы построить функцию φ, такую, что для любого номера j функции φ было $\Phi_j(n) > f(n)$ для почти всех n, мы поступаем следующим образом: для каждого n мы рассматриваем первую функцию с наименьшим номером i, такую, что $\Phi_i(n) \leqslant f(n)$, и делаем $\varphi(n)$ отличным от $\varphi_i(n)$ при условии, что это не было сделано раньше. Более точно, пусть $s(n)$ — наименьшее целое положительное число, меньшее чем n, такое, что

$$\Phi_{s(n)}(n) \leqslant f(n),$$

и ни для какого $m < n$ не выполняется

$$\Phi_{s(n)}(m) \leqslant f(m), \quad \varphi_{s(n)}(m) \neq \varphi(m).$$

Если такого числа не существует, то $s(n)$ не определено. Пусть

$$\varphi(n) = \begin{cases} 0, & \text{если } \varphi_{s(n)}(n) = 1, \\ 1 & \text{в противном случае.} \end{cases}$$

Ясно, что $\varphi(n)$ — рекурсивная функция. Предположим, что $\Phi_j(n) \leqslant f(n)$ для бесконечно многих n и $\varphi_j = \varphi$. Рано или поздно для некоторого значения n, скажем n_0, число j будет тем наименьшим числом k, для которого $\Phi_k(n_0) \leqslant f(n_0)$, и для $m < n_0$ не выполняется неравенство $\Phi_k(m) \leqslant f(m)$ с условием $\varphi_k(m) \neq \varphi(m)$. Тогда по определению φ имеем $\varphi(n_0) \neq \varphi_j(n_0)$ — противоречие. Следовательно, для каждого номера i функции φ имеем $\Phi_i(n) > f(n)$ для почти всех n.

Следствие. *Для всех мер существуют сколь угодно сложные функции, принимающие только два значения* 0 *и* 1.

Из теоремы 2 мы видим, что существуют сколь угодно сложные ограниченные функции. Отсюда мы немедленно заключаем, что не может быть рекурсивного отношения между функциями и их сложностями, так как из такого отношения получалось бы ограничение на сложность любой ограниченной функции.

[1]) Выражения «почти всюду» и «для почти всех» означают: для всех, за исключением, быть может, конечного числа. — *Прим. перев.*

Теорема 3. *Пусть* Φ — *мера сложности вычислений.*

(A) *Тогда не существует рекурсивной функции* k, *такой, что для каждого* i

$$k(n, \varphi_i(n)) \geqslant \Phi_i(n)$$

почти всюду (*п. в.*)

(B) *Существует рекурсивная функция* h, *такая, что для каждого* i

$$h(n, \Phi_i(n)) \geqslant \varphi_i(n) \text{ п. в.}$$

Доказательство. Предположим, что такая функция k существует. Тогда сложность любой двузначной функции $\varphi_i(n)$ должна удовлетворять условию

$$\Phi_i(n) \leqslant k(n, 0) + k(n, 1) \text{ п. в.},$$

что противоречит предыдущему следствию.

Чтобы показать, что существует желаемая функция h, положим

$$H(i, n, m) = \begin{cases} \varphi_i(n), & \text{если } \Phi_i(n) = m, \\ 1 & \text{в противном случае.} \end{cases}$$

Функция H рекурсивная, так как мы можем определить, выполняется ли $\Phi_i(n) = m$ (если да, то $\varphi_i(n)$ определена и мы можем вычислить это значение, в противном случае функция принимает значение 1). Функция h определяется так:

$$h(n, m) = \max_{i \leqslant n} H(i, n, m).$$

Ясно, что если $\varphi_i(n)$ определено и $i \leqslant n$, то

$$h(n, \Phi_i(n)) \geqslant \varphi_i(n),$$

что и завершает доказательство.

Вторая часть этого результата утверждает, что мы можем рекурсивно ограничить величину любой функции с помощью ее сложности. Отсюда мы заключаем, что «ужасно» быстро растущие вычислимые функции будут и «ужасно» сложными.

Хотя не существует рекурсивного отношения между величиной функции и ее сложностью, существует рекурсивное отношение между сложностями одного и того же алгоритма для любых двух мер. Другими словами, функцию, которую «легко» вычислять при одной мере, «легко» вычислять и при всех других.

Теорема 4. *Пусть* Φ *и* $\hat{\Phi}$ — *меры сложности. Тогда существует рекурсивная функция* r, *такая, что для любого* i

$$r(n, \hat{\Phi}_i(n)) \geqslant \hat{\Phi}_i(n) \quad \text{и} \quad r(n, \hat{\Phi}_i(n)) \geqslant \Phi_i(n)$$

для почти всех n.

Доказательство. Пусть

$$r(n, m) = \max\{\Phi_i(n), \hat{\Phi}_i(n) \mid i \leqslant n \text{ и } \Phi_i(n) = m \text{ или } \hat{\Phi}_i(n) = m\}.$$

Очевидно, что

$$r(n, \Phi_i(n)) \geqslant \hat{\Phi}_i(n) \quad \text{и} \quad r(n, \hat{\Phi}_i(n)) \geqslant \Phi_i(n)$$

для почти всех n. Функция r рекурсивна, так как существует эффективная процедура для определения, равно ли $\Phi_i(n)$ или $\hat{\Phi}_i(n)$ числу m. Если хотя бы одно из этих значений равно m, то оба должны быть определены. Следовательно, можно эффективно вычислить максимум.

Утверждение о том, что две меры связаны рекурсивным отношением, с практической точки зрения значит не слишком много, так как это отношение может быть сколь угодно грубым. Кроме того, рекурсивное отношение, даваемое теоремой 4, может не быть точной границей потому, что данное отношение зависит от нумерации. Так, если мы сравниваем число шагов одноленточной машины Тьюринга с числом использованных ею ячеек ленты, то функция r из теоремы 4 будет зависеть от порядка, в котором мы нумеруем машины Тьюринга. Однако в качестве следствия этой теоремы мы отметим, что сложность любого класса функций, которая рекурсивно ограничена при одной мере (например, полиномами, примитивно рекурсивными функциями и т. д.), будет рекурсивно ограничена при любой другой мере сложности.

Прежде чем перейти к изучению более впечатляющих результатов, мы докажем одну промежуточную лемму, которая показывает, что при любой мере, когда мы комбинируем вычисления, сложность нового вычисления рекурсивно ограничена сложностями составляющих вычислений. Эта лемма часто используется при изучении сложности вычислений.

Лемма 1 (лемма о комбинировании). *Пусть* Φ *— произвольная мера и* $c(i, j)$ *— рекурсивная функция, такая, что если* $\varphi_i(n)$ *и* $\varphi_j(n)$ *определены, то определено и* $\varphi_{c(i,j)}(n)$. *Тогда существует рекурсивная функция* h, *такая, что*

$$\Phi_{c(i,j)}(n) \leqslant h(n, \Phi_i(n), \Phi_j(n)) \text{ п. в.}$$

Доказательство. Определим

$$p(i, j, n, m, l) = \begin{cases} \Phi_{c(i,j)}(n), & \text{если } \Phi_i(n) = m \text{ и } \Phi_j(n) = l, \\ 1 & \text{в остальных случаях.} \end{cases}$$

Эта функция рекурсивна, и мы получаем желаемую функцию h, полагая

$$h(n, m, l) = \max_{i, l \leqslant n} p(i, j, n, m, l).$$

Очевидно, что для $n \geqslant i, j$ мы имеем

$$h(n, \Phi_i(n), \Phi_j(n)) \geqslant \Phi_{c(i,j)}(n);$$

таким образом, это неравенство выполняется почти всюду, что и требовалось показать.

3. ТЕОРЕМА ОБ УСКОРЕНИИ И ЕЕ ПРИЛОЖЕНИЯ

Теперь мы дадим новое доказательство (без использования теоремы о рекурсии [1]) довольно удивительного результата о существовании функций, не имеющих наилучших алгоритмов. На самом деле мы покажем, что для любой рекурсивной функции $r(n)$ существует рекурсивная функция $\varphi(n)$, такая, что для каждого номера i функции φ найдется номер j этой функции, для которого $\Phi_i(n) \geqslant r(\Phi_j(n))$, если n достаточно велико.

Например, если мы выберем $r(n) = 2^n$, то существует рекурсивная функция φ, такая, что если $\varphi_i = \varphi$, то для некоторого другого номера j функции φ мы имеем

$$\Phi_j(n) \leqslant \log \Phi_i(n) \text{ п. в.}$$

Более того, этот процесс можно повторить, и мы заключаем, что существует номер k функции φ, такой, что

$$\Phi_k(n) \leqslant \log \log \Phi_i(n) \text{ п. в.},$$

и это логарифмическое ускорение можно итерировать любое число раз.

Вместо того чтобы прямо доказывать самый общий случай, мы вначале установим этот результат для одной специфической и понятной меры с помощью прямых диагональных рассуждений, а затем для получения общей теоремы используем тот факт, что все меры связаны рекурсивным отношением. Мера сложности, которую мы используем, основана на емкости ленты, используемой одноленточной машиной Тьюринга. Машины модифицированы таким образом, что они никогда не зацикливаются на конечном куске ленты, и, таким образом, эта мера удовлетворяет двум условиям нашего определения. Кроме того, мы модифицируем машины так, чтобы $L_i(n)$ (число ячеек ленты, использованных i-й машиной Тьюринга на входе n) было больше, чем длина описания i-й машины. Наконец, мы используем такую нумерацию этих машин Тьюринга, что i-я машина имеет не больше i различных символов ленты. Преимущество выбора емкости ленты в качестве меры сложности свя-

---

[1]) Пусть $\varphi_1, \varphi_2, \varphi_3, \ldots$ — вычислимая главная нумерация. Тогда для каждой рекурсивной функции $f(i, n)$ существует i_0, такое, что $f(i_0, n) = \varphi_{i_0}(n)$ для всех n. Эквивалентное этому утверждение называется теоремой о неподвижной точке. — *Прим. перев.*

зано с тем, что ленту можно использовать повторно несколько раз для различных вычислений, которые мы будем выполнять в ходе нужного нам вычисления функции φ.

Мы говорим, что рекурсивная функция $f(n)$ является *емкостной сигнализирующей* функцией, если существует одноленточная машина Тьюринга, которая использует точно $f(n)$ ячеек ленты для вычисления при аргументе n, $n = 1, 2, \ldots$.

Основная идея следующего ниже доказательства очень проста. Мы строим функцию φ, которую «малые» машины быстро вычислять не могут, проводя диагональный процесс, в котором сила условий убывает с ростом номера машины. Более точно, для подходящим образом выбранной функции h мы формализуем следующую процедуру построения φ: «если i-я машина вычисляет $\varphi_i(n)$, используя менее $h(n-i)$ ячеек ленты, то φ не совпадает с φ_i», затем мы показываем, что этот диагональный процесс достаточно прост, чтобы для любого k мы могли с помощью достаточно большой машины вычислить φ, используя $h(n-k)$ ячеек ленты.

Лемма 2. *Пусть $r(n)$ — произвольная рекурсивная функция. Тогда существует рекурсивная функция $\varphi(n)$, такая, что для каждого i, для которого $\varphi_i(n) = \varphi(n)$, существует j, удовлетворяющее условиям*

$$\varphi_j(n) = \varphi(n) \quad \text{и} \quad L_i(n) \geqslant r(L_j(n))$$

для достаточно больших n.

Доказательство. Не теряя общности, мы можем предполагать, что $r(n)$ — строго возрастающая емкостная сигнализирующая функция. (В противном случае заменим $r(n)$ на $\hat{r}(n)$, где $\hat{r}(n)$ — строго возрастающая емкостная сигнализирующая функция и $\hat{r}(n) \geqslant r(n)$ для всех n.) Определим $h(n)$, полагая $h(1) = 1$ и для $n > 1$

$$h(n) = r(h(n-1)) + 1.$$

Тогда для всех $n > 1$

$$h(n) > r(h(n-1))$$

и, очевидно, h является емкостной сигнализирующей функцией, так как $r(k)$ и $r(k)+1$ — емкостные сигнализирующие, а следовательно, по индукции такова и $h(n)$.

Определим функцию $\varphi(n)$ таким образом, что

(1) $\varphi_i(n) = \varphi(n)$ влечет за собой то, что $L_i(n) \geqslant h(n-i)$ п. в.

(2) для каждого k существует номер j, такой, что

$$\varphi_j(n) = \varphi(n) \quad \text{и} \quad L_j(n) \leqslant h(n-k).$$

Это гарантирует, что для данного номера i функции φ существует номер j функции φ, такой, что

$$L_i(n) > r(L_j(n)) \text{ п. в.}$$

Чтобы получить это, выберем j так, чтобы

$$L_j(n) \leqslant h(n-i-1).$$

Тогда

$$L_i(n) \geqslant h(n-i) > r(h(n-i-1)) \geqslant r(L_j(n)).$$

Построение $\varphi(n)$. Положим

$$\varphi(1) = \begin{cases} \varphi_1(1)+1, & \text{если } L_1(1) < h(1), \\ 0 & \text{в противном случае.} \end{cases}$$

Если $L_1(1) < h(1)$, то вычеркиваем первую машину из списка машин Тьюринга. Для $n = 2, 3, 4, \ldots$ полагаем

$$\varphi(n) = \begin{cases} \varphi_i(n)+1, & \text{где } i < n \text{ наименьший, еще не вычеркнутый номер, такой, что } L_i(n) < h(n-i), \text{ и вычеркиваем номер } i; \\ 0, & \text{если такого } i \text{ не существует.} \end{cases}$$

Очевидно, что если i-я машина вычисляет φ, то

$$L_i(n) \geqslant h(n-i) \text{ п. в.},$$

так как для достаточно большого n_0 каждое $j < i$, которое рано или поздно вычеркивается, будет уже вычеркнуто, и, следовательно, если

$$L_i(n) < h(n-i)$$

для какого-нибудь $n > n_0$, то

$$\varphi(n) = \varphi_i(n) + 1$$

и, следовательно,

$$\varphi_i(n) \neq \varphi(n).$$

Более того, для каждого k существует машина Тьюринга j, которая вычисляет φ, используя

$$L_j(n) \leqslant h(n-k)$$

ячеек ленты. Пусть $i_k > k$ таково, что при любом фиксированном $i \geqslant i_k$ неравенство

$$ih(n-i) < h(n-k)$$

выполняется для всех $n > i$. (Такое i_k найдется, если с самого начала взять быстро растущую $\hat{r}$, например $\hat{r}(n) \geqslant 2^n$.) Пусть v — значение аргумента φ, при котором вычеркивается последний из тех номеров $i < i_k$, которые вычеркиваются в ходе построения. Машина j работает следующим образом. Если $n \leqslant v$, то она просто печатает значение $\varphi(n)$, которое хранится в ее конечной внутренней памяти. Для $n > v$ машина j следующим образом вычисляет наименьшее невычеркнутое i:

(1) j отмеривает $h(n-k)$ ячеек ленты;

(2) j запасает во внутренней памяти значения номеров, вычеркнутых для аргументов $n \leqslant v$;

(3) для $v < m \leqslant n$ машина j моделирует каждую машину i, $i_k \leqslant i \leqslant n$, определяет, какая машина вычеркивается при каждом из значений $m < n$, и затем находит наименьшее невычеркнутое i, такое, что

$$L_i(n) < h(n-i),$$

и записывает

$$\varphi(n) = \varphi_i(n) + 1.$$

Если такого i не существует, то $\varphi(n)$ полагается равным нулю. Это моделирование можно осуществить, используя $h(n-k)$ ячеек ленты, так как машина j моделирует только те машины, номера которых не меньше i_k, и моделируют машину i только до тех пор, пока она использует менее $h(n-i)$ ячеек ленты. Так как машина i имеет не больше i символов ленты, моделирование требует не больше

$$ih(n-i) < h(n-k)$$

ячеек ленты.

Теперь мы рассмотрим произвольные меры и покажем, что для всех мер справедлива теорема об ускорении.

Теорема 5 (теорема об ускорении). *Пусть* Φ *— мера сложности и* $r(n)$ *— рекурсивная функция. Существует рекурсивная функция* $\varphi(n)$, *такая, что для каждого* i, *такого, что* $\varphi_i(n) = \varphi(n)$, *существует* j, *для которого*

$$\varphi_j(n) = \varphi(n) \quad \text{и} \quad \Phi_i(n) \geqslant r(\Phi_j(n)) \text{ п. в.}$$

Доказательство. Не теряя общности, мы предполагаем, что $r(n)$ — монотонно возрастающая функция. Так как все меры связаны рекурсивным отношением, существует строго монотонная неограниченная рекурсивная функция R, такая, что

$$L_i(n) \leqslant R(\Phi_i(n)) \quad \text{и} \quad \Phi_j(n) \leqslant R(L_j(n)) \text{ п. в.}$$

при условии, что $\Phi_i(n)$ и $L_j(n)$ растут быстрее n. Положим

$$\hat{r}(n) = R(r(R(n))).$$

Выберем по лемме 2 функцию φ так, что для каждого i, такого, что $\varphi_i = \varphi$, существует номер j функции φ со свойством

$$L_i(n) \geqslant \hat{r}(L_j(n)) \text{ п. в.}$$

Тогда

$$R(r(\Phi_j(n))) \leqslant R(r(R(L_j(n)))) \leqslant L_i(n) \leqslant R(\Phi_i(n)) \text{ п. в.}$$

Но из

$$R(r(\Phi_j(n))) \leqslant R(\Phi_i(n)) \text{ п. в.}$$

и строгой монотонности функции R следует

$$r(\Phi_j(n)) \leqslant \Phi_i(n) \text{ п. в.},$$

что и нужно было показать.

В следующем разделе мы покажем, что не для всякой рекурсивной функции можно ускорить ее вычисление, доказав, что для любой меры существует возрастающая рекурсивная функция h, такая, что для каждой достаточно большой временной сигнализирующей функции Φ_i существует функция f, имеющая сложность Φ_i, вычисление которой нельзя ускорить в h раз. То есть f вычислима за Φ_i шагов, но не вычислима за Φ_j шагов, если

$$h \circ \Phi_j(n) < \Phi_i(n) \text{ п. в.}$$

Таким образом, для любой меры многие функции имеют h-наилучшие программы.

Следует также заметить, что ускорение не эффективно в том смысле, что величина v в конструкции леммы 2 определяется неэффективно. Отсюда сразу возникает вопрос, существует ли другая конструкция и другая функция f, для которой ускорение эффективно. Если требуется небольшое ускорение, скажем линейное ускорение при вычислениях на машинах Тьюринга с ограничением на емкость, то эффективная процедура существует. (Что значит «небольшое» — это, конечно, зависит от меры, и оно должно быть меньше, чем только что упоминавшееся h.) Однако большие ускорения не могут быть эффективными. Так как эффективность ускорения не зависит от меры, мы рассмотрим емкостную сигнализирующую функцию для одноленточных машин Тьюринга и покажем, что для этой меры большие ускорения неэффективны. В наброске доказательства снова будет использован тот факт, что мы можем повторно использовать ленту много раз для различных вычислений. Пусть φ — некоторая функция и $i_1, i_2, i_3, \ldots$ — список программ, вычисляющих φ и обладающих тем свойством, что если $\varphi_j = \varphi$, то существует k, такое, что $r(\Phi_{i_k}(n)) < \Phi_j(n)$ п. в. Здесь r — достаточно большая функция, и перед нами стоит проблема, можем ли мы предполагать, что этот список рекурсивно перечислим [1]).

Рассмотрим алгоритм, который для входа n отмечает две ячейки ленты, перечисляет столько номеров из списка $i_1, i_2, i_3, \ldots$, сколько поместится на половине куска ленты, заключенного между отмеченными ячейками, и затем моделирует (некоторым подходя-

---

[1]) Отрицательное решение этой проблемы еще не доказывает «неэффективности ускорения», которую естественно понимать так: не существует частично рекурсивной функции p, такой, что если $\varphi_j = \varphi$, то $p(j)$ определено и $\varphi_{p(j)}$ есть r-ускорение для функции φ (т. е. $r(\Phi_{p(j)}(n)) < \Phi_j(n)$ п. в.). Это утверждение доказано Блюмом [1]. — *Прим. перев.*

щим способом) последовательно каждый алгоритм φ_{i_1}, φ_{i_2}, ... до тех пор, пока моделирование не попытается выйти за пределы отмеченной части ленты. Если ни один из алгоритмов еще не вычислил $\varphi(n)$, то отмечаются две более удаленные друг от друга ячейки и процесс повторяется. Таким образом, каждый алгоритм φ_{i_j} из списка может быть промоделирован для достаточно большого n, и так как для моделирования φ_{i_j} нашим алгоритмом требуется емкость ленты, не более чем в постоянное число раз большая, чем емкость, использованная самим φ_{i_j}, то мы можем заключить, что этому алгоритму требуется приблизительно такая же емкость (почти всюду), как и любому из алгоритмов нашего списка. Следовательно, в этом списке нет программы, которая использовала бы значительно меньшую емкость, и поэтому для построенной нами программы мы не имеем r-ускорения. Таким образом, мы не можем рекурсивно перечислять какой-нибудь список алгоритмов, вычисляющих функцию f, который для любого алгоритма, вычисляющего f, содержит его r-ускорение.

Так как теорема об ускорении применима ко всем мерам сложности, то ее можно применять для ускорения программ вычислительных машин. Однако такие программы обычно используются для вычисления значений функции для некоторого конечного множества значений аргумента, а не для всевозможных значений. Уменьшение асимптотического времени работы программы почти всюду — это не совсем то, в чем заинтересован вычислитель-практик. Однако теорема говорит о том, что мы не можем классифицировать функции по сложности их вычисления, так как некоторые функции не имеют внутренней минимальной сложности. То, что мы будем делать вместо этого, — это определять классы сложности и изучать их в разд. 4.

Прежде чем перейти к изучению классов сложности, мы обсудим одно приложение теоремы об ускорении. Рассмотрим две вычислительные машины, каждая из которых может вычислять все частично рекурсивные функции. Предположим, что одна из вычислительных машин имеет очень богатое множество операций и, скажем, может выполнять $(10!^{10!})!$ операций в секунду. Предположим, что вторая машина имеет всего несколько операций первой машины и может выполнять только по одной операции за каждые сто лет. Мы можем теперь использовать эти вычислительные машины и их программы, чтобы определить две меры сложности вычислений, основанные на их времени работы, измеряемой в секундах. Кроме того, мы знаем из теоремы 4, что времена работы этих двух машин связаны рекурсивной функцией h. Очевидно, что h будет на самом деле очень большой функцией. Тем не менее теорема об ускорении утверждает, что если мы возьмем функции, которые имеют r-ускорение при $r > h$, то мы не сможем получить преиму-

щества в скорости вычисления, используя более быструю машину для вычисления этих функций, так как для каждой программы быстрой машины существует другая программа для медленной машины, вычисляющая ту же функцию быстрее (для больших значений аргумента). Отметим, что это заключение выполняется только для достаточно больших значений n и что не существует эффективного способа нахождения программы для медленной машины; однако мы знаем, что она существует.

Позднее мы получим связанный с этим результат, теорему о пробелах, которая утверждает, что можно указать сколь угодно большую рекурсивную функцию, такую, что множество функций, время вычисления которых ограничено этой функцией, одно и то же для обеих машин. В этом случае, как будет видно, доказательство не основано на том, что медленная машина использует лучшие программы.

4. КЛАССЫ СЛОЖНОСТИ

Во втором разделе этой статьи мы постулировали два свойства, которыми должна обладать любая мера сложности вычислений, а затем мы получили несколько результатов о мерах сложности, используя только эти постулаты. Мы видели, что для любой меры существуют сколь угодно сложно вычислимые функции, что не существует рекурсивного отношения между величиной функции и сложностью ее вычисления, что любые две меры сложности связаны рекурсивным отношением (они рекурсивно ограничивают друг друга) и, наконец, что существуют функции, вычисления которых могут быть как угодно «ускорены».

Мы теперь обращаемся к изучению классов функций, сложность вычисления которых ограничена рекурсивными функциями. Более точно, для любой меры сложности Φ и рекурсивной функции φ_i мы определяем класс сложности

$$C^{\Phi}_{\varphi_i} = \{\varphi_j \mid \Phi_j(n) \leqslant \varphi_i(n) \text{ п. в.}\}.$$

Таким образом, $C^{\Phi}_{\varphi_i}$ или C_{φ_i} состоит из всех вычислимых функций, сложность которых ограничена функцией φ_i почти всюду.

Рассмотрим первой проблему, состоящую в построении для данного C_{φ_i} нового класса сложности, который содержит некоторую новую функцию, не содержащуюся в C_{φ_i}.

Построение нового класса будет достигнуто с помощью диагонального процесса, охватывающего всюду определенные функции, которые вычисляются за $\varphi_i(n)$ шагов. Следует отметить, что в нашей диагонализации мы будем рассматривать каждый номер бесконечно часто, так как функция φ_j находится в C_{φ_i} лишь при усло-

вии, что $\Phi_j(n) \leqslant \varphi_i(n)$ для достаточно больших n. Таким образом, даже если мы найдем, что $\Phi_j(n) > \varphi_i(n)$ для некоторого n, мы все же должны проверять позднее, не выполнилось ли обратное неравенство и не должны ли мы сделать $\varphi \neq \varphi_j$.

Чтобы это сделать, выбираем, во-первых, рекурсивную функцию r, такую, что для каждого i, $r(n) = i$ для бесконечно многих n. Затем определяем

$$\varphi(n) = \begin{cases} \varphi_{r(n)}(n) + 1, & \text{если } \Phi_{r(n)}(n) \leqslant \varphi_l(n), \\ 1 & \text{в противном случае.} \end{cases}$$

Если $\varphi_j \in C_{\varphi_i}$, то для достаточно большого n имеем $r(n) = j$ и $\Phi_j(n) \leqslant \varphi_i(n)$, откуда следует, что $\varphi \neq \varphi_j$ по построению. (Заметим, что из $\varphi_j \in C_{\varphi_i}$ не обязательно следует, что $\Phi_j \leqslant \varphi_i$ п. в. Отсюда следует только, что существует номер k функции φ_j, т. е. $\varphi_j = \varphi_k$, такой, что $\Phi_k \leqslant \varphi_i$ п. в. Не теряя общности, мы будем предполагать везде в этой статье, что в таких ситуациях мы уже выбрали подходящий номер.) Таким образом, φ не принадлежит C_{φ_i}. С другой стороны, в силу леммы о комбинировании мы знаем, что для некоторой рекурсивной h и номера k функции φ

$$\Phi_k(n) \leqslant h(n, \Phi_{r(n)}(n), \Phi_i(n)) \text{ п. в.}$$

Таким образом, мы получили следующий результат.

Теорема 6. *Для любой меры* Φ *существует рекурсивная функция* H, *такая, что для каждой всюду определенной* φ_i *существует функция* f_i

$$f_i \notin C_{\varphi_i} \quad \text{и} \quad f_i \in C_{H(n, \Phi_i(n))}.$$

Эту формулу можно упростить, устранив первый аргумент из $H(n, \Phi_i(n))$, если $\Phi_i(n) \geqslant n$. Но в общем случае мы не можем это сделать, так как если одна из наших мер есть число ячеек рабочей ленты, использованных машиной Тьюринга со входом, и $\Phi_i(n)$ — константа (в действительности такая машина — это конечный автомат), то $H(\Phi_i(n))$ была бы константой и не могла бы ограничивать, скажем, число шагов обычной машины Тьюринга, которое растет по крайней мере линейно с ростом аргумента.

Следствие. *Для любой меры* Φ *существует рекурсивная функция* k, *такая, что для любой всюду определенной* $\varphi_i(n)$, *такой, что* $\Phi_i(n) \geqslant n$, *существует такая* $f_i(n)$, *что*

$$f_i \notin C_{\varphi_i} \quad \text{и} \quad f_i \in C_{k \circ \Phi_i}.$$

Слегка модифицировав предыдущее доказательство (заменив H возрастающей R), мы можем получить новый класс сложности, собственно содержащий в себе старый класс.

Следствие. *Для любой меры* Φ *существует рекурсивная функция* R, *такая, что для любой всюду определенной* φ_i

$$C_{\varphi_i} \subset C_{R(n, \Phi_i(n))}.$$

Как и раньше, если мы рассматриваем достаточно «трудную» рекурсивную функцию φ_i, *т. е.* $\Phi_i(n) \geqslant n$, *то мы можем выбросить* n *из предыдущего соотношения и получить* $C_{\varphi_i} \subset C_{R \circ \Phi_i}$.

Предыдущие результаты показывают, что для любой меры сложности мы можем получить новые классы сложности эффективно по сигнализирующей для любой рекурсивной φ_i, т. е.

$$C_{\varphi_i} \subset C_{R \circ \Phi_i}.$$

Доказательство этого результата было совсем простым и мы видели, что сигнализирующая Φ_i вошла в это выражение потому, что мы должны были вычислить φ_i, чтобы ограничить число шагов моделирования $\varphi_{r(n)}(n)$. Если бы мы могли величину сигнализирующей Φ_i рекурсивно ограничить величиной функции φ_i, мы могли бы получать новые классы сложности эффективно по φ_i вместо Φ_i. С другой стороны, мы знаем, что сложность функции нельзя рекурсивно ограничить ее величиной, и это вызывает подозрение, что не существует рекурсивной функции q, такой, что $C_{\varphi_i} \subset C_{q \circ \varphi_i}$ для всех достаточно больших φ_i. Следующий результат подтверждает это подозрение, устанавливая, что предыдущий результат является самым сильным результатом об эффективной равномерности, который мы можем получить, когда рассматриваем все всюду определенные функции.

На теорему о пробелах, которую мы докажем далее, можно смотреть как на утверждение о том, что для любой меры сложности сигнализирующие функции редко разбросаны среди рекурсивных функций. Причина этого состоит в том, что второе условие для мер сложности позволяет нам решать, выполняется ли $\Phi_i(n) \leqslant m$ для всех n, m и i. Это условие настолько сильное, что, используя его, мы можем построить для каждой рекурсивной функции r рекурсивную функцию t, такую, что ни одна сигнализирующая функция не попадает бесконечно часто между t и $r \circ t$, гарантируя таким образом, что все, что можно вычислить с границей $r \circ t$, можно также вычислить и с границей t.

Теорема 7 (теорема о пробелах). *Для любой меры сложности* Φ *и для любой рекурсивной функции* r, $r(n) \geqslant n$ *существует рекурсивная монотонно возрастающая функция* t, *такая, что*

$$C_t = C_{r \circ t}.$$

Доказательство. Пусть $\Phi_1, \Phi_2, \Phi_3, \ldots$ — нумерация сигнализирующих функций меры сложности Φ. Мы определяем же-

лаемую t индуктивно: пусть

$$t(1) = 1,$$
$$t(n+1) = t(n) + m_n,$$

где

$$m_n = \min \{m \mid \text{для каждого } i \leqslant n, \text{ либо}$$
$$\Phi_i(n+1) \geqslant r \circ (t(n) + m), \text{ либо } \Phi_i(n+1) \leqslant t(n) + m\}.$$

Последний предикат рекурсивен, так что мы можем проверять его для $m = 1, 2, 3, \ldots$. Чтобы убедиться, что эта процедура найдет желаемое m_n, заметим, что существует лишь конечное число значений $\Phi_i(n+1)$, $i \leqslant n$, и, таким образом, существует m, такое, что

$$\Phi_i(n+1) \leqslant t(n) + m$$

для всех тех i, $1 \leqslant i \leqslant n$, для которых $\Phi_i(n+1)$ определено; если $\Phi_i(n+1)$ не определено, то, очевидно,

$$\Phi_i(n+1) \geqslant r \circ (t(n) + m) \quad \text{для всех} \quad m.$$

Таким образом, желаемое m существует, и мы заключаем, что $t(n)$ — рекурсивная функция.

Чтобы получить, что $C_t \subset C_{rot}$, мы должны найти Φ_j, такую, что $\Phi_j(n) \leqslant r \circ t(n)$ п. в., и не верно, что $\Phi_j(n) \leqslant t(n)$ п. в. Это невозможно по построению t, так как $\Phi_j(n) \leqslant r \circ t(n)$ п. в. влечет $\Phi_j(n) \leqslant t(n)$ п. в.
Таким образом,

$$C_t = C_{rot}.$$

Теорема о пробелах показывает, что существуют сколь угодно широкие «пробелы» между рекурсивными функциями, в которых не содержится (с точностью до конечного числа значений) ни одной сигнализирующей функции. Классы сложности, определенные функциями, ограничивающими такой пробел, совпадают. Отсюда также следует, что мы не можем иметь рекурсивного способа увеличивать каждую достаточно большую всюду определенную функцию φ_i до $r \circ \varphi_i$, чтобы получить границу сложности для некоторого вычисления, не лежащего в C_{φ_i}.

В то же время стоит напомнить, что существует рекурсивный способ увеличения сигнализирующей Φ_i любой рекурсивной функции φ_i, дающий границу для вычисления, которое не принадлежит C_{φ_i}, т. е. $C_{\varphi_i} \subset C_{R(n, \Phi_i(n))}$. Мы используем теперь этот результат, чтобы показать, для каких подклассов рекурсивных функций мы можем рекурсивно увеличивать границу сложности, чтобы получать новые вычисления.

Отметим сначала, что емкостная и временная сигнализирующие машины Тьюринга являются границами для вычислений самих

себя. Более точно, существует рекурсивная функция σ, такая, что для каждой емкостной сигнализирующей L_i мы имеем

$$\varphi_{\sigma(i)}(n) = L_i(n) \quad \text{и} \quad \varphi_{\sigma(i)} \in C_{L_i}.$$

Функция $\sigma(i)$ по машине M_i дает новую машину, которая проверяет, сколько ленты использует M_i, и затем записывает число ячеек ленты в качестве выхода в подходящей (скажем, двоичной) форме. При подходящих соглашениях о записи выхода то же самое верно для временных сигнализирующих многоленточных машин Тьюринга. Это наводит на мысль о том, что меры, у которых сигнализирующие служат границами сложности собственных вычислений, заслуживают специального внимания.

Определение. Мера сложности вычислений Φ называется *собственной*, если существует рекурсивная функция σ, такая, что для всех i

$$\varphi_{\sigma(i)} = \Phi_i \quad \text{и} \quad \varphi_{\sigma(i)} \in C_{\Phi_i}.$$

Это приводит нас к следующему результату.

Следствие. *Для любой собственной меры Φ существует рекурсивная функция R, такая, что для всех i*

$$C_{\Phi_i} \subset C_{K(n, \Phi_i(n))}.$$

Доказательство. Так как Φ_i служит границей вычисления Φ_i, то мы можем заменить φ_i на Φ_i во втором следствии теоремы 6.

Предыдущий результат легко можно обобщить на все меры, если заметить, что для любой меры величина сигнализирующей рекурсивно ограничивает сложность своего собственного вычисления.

Лемма 3. *Для любой меры Φ существуют две рекурсивные функции σ и r, такие, что для всех i имеем $\Phi_i(n) = \varphi_{\sigma(i)}(n)$ и $r(n, \Phi_i(n)) \geqslant \Phi_{\sigma(i)}(n)$ п. в.*

Доказательство. Тот факт, что для всех i, m и n мы можем решить, выполняется ли $\varphi_{\sigma(i)}(n) = m$, позволяет нам дать доказательство, очень похожее на доказательство леммы о комбинировании [1]). Мы полагаем

$$p(i, n, m) = \begin{cases} \Phi_{\sigma(i)}(n), & \text{если } \varphi_{\sigma(i)}(n) = m, \\ 1 & \text{в противном случае.} \end{cases}$$

[1]) Существование функции σ вытекает из определения меры сложности вычислений. — *Прим. ред.*

и затем определяем

$$r(n, m) = \max_{i \leqslant n} p(i, n, m).$$

Очевидно, что r является требуемой ограничивающей функцией.

Объединяя этот результат с теоремой 6, мы получаем следующую теорему.

Теорема 8. *Для любой меры* Φ *существует рекурсивная функция* h, *такая, что для всех* i

$$C_{\Phi_i} \subset C_{h(n, \Phi_i(n))}.$$

Доказательство. Достаточно заметить, что результат леммы 3, подставленный в результат второго следствия теоремы 6, дает желаемое отношение.

Возвращаясь к последнему результату, мы видим, что сигнализирующие можно рекурсивно увеличивать, чтобы получать границы для новых вычислений. Этот результат связан прежде всего с тем фактом, что мы можем перечислять сигнализирующие Φ_i и определять, выполняется ли $\Phi_i(n) = m$. Сейчас мы обобщим это наблюдение.

Определение. Множество функций $\{\gamma_i\}$, которое можно рекурсивно перечислить и для которого можно при любых i, m, n решить, выполняется ли $\gamma_i(n) = m$, называется *измеримым множеством* функций.

Заметим, что сигнализирующие любой меры сложности образуют измеримое множество и, кроме того, свойство быть измеримым множеством не зависит от меры сложности.

Приведенное выше определение позволяет нам установить более общий результат.

Теорема 9. *Пусть* $\{\gamma_i\}$ — *измеримое множество функций. Тогда для любой меры сложности* Φ *существует рекурсивная функция* r, *такая, что*

$$C_{\gamma_i} \subset C_{r(n, \gamma_i(n))}.$$

Доказательство. Заметим, что для любого измеримого множества существуют рекурсивные σ и h, такие, что

$$\varphi_{\sigma(i)} = \gamma_i \quad \text{и} \quad \Phi_{\sigma(i)}(n) \leqslant h(n, \varphi_{\sigma(i)}(n)) \text{ п. в.},$$

и затем используем второе следствие теоремы 6.

Теперь мы рассмотрим проблему перечисления всех функций, принадлежащих классу сложности.

Мы говорим, что класс сложности C_t^{Φ} *рекурсивно перечислим*, если существует рекурсивно перечислимое множество номеров, которое содержит номер каждой функции из C_t^{Φ} и состоит только из

номеров функций из C_t^{Φ}. Заметим, что мы отнюдь не настаиваем на том, чтобы алгоритмы, которые появляются при этом пересчете, сами работали в пределах границы t, мы требуем только, чтобы они были именами (номерами) функций, для которых существует некоторый алгоритм, работающий в пределах t (почти всюду).

Оказывается, что три специфические меры сложности, обсуждавшиеся в этой статье, имеют рекурсивно перечислимые классы сложности. Мы покажем, что это имеет место для любых мер сложности, если классы сложности (их границы) достаточно велики. С другой стороны, мы покажем также, что существуют меры сложности, для которых классы сложности не могут быть рекурсивно перечислимыми. Это довольно удивительно, так как отсюда следует, что не существует эффективного способа описания того, какие функции лежат в этих классах. Это означает, что такие меры сложности являются весьма патологическими и что на меры сложности надо налагать дополнительные ограничения, чтобы устранить такие случаи.

Отметим сначала, что для любого рекурсивно перечислимого множества рекурсивных функций мы можем ограничить (п. в.) их сигнализирующие.

Лемма 4. *Пусть* Φ *— любая мера и* A *— рекурсивно перечислимое множество всюду определенных функций. Тогда существует рекурсивная функция* t*, такая, что* $A \subseteq C_t$.

Доказательство. Пусть $i_1, i_2, i_3, \ldots$ — рекурсивный пересчет A, тогда определим

$$t(n) = \max \{\Phi_{i_j}(n) \mid j \leqslant n\}.$$

Очевидно, что $t(n)$ рекурсивна и что для каждого j имеем $\Phi_{i_j}(n) \leqslant t(n)$ п. в. Таким образом, $A \subseteq C_t$.

В следующем результате мы используем множество функций с конечной основой:

$$\mathcal{F} = \{\varphi_i \mid \varphi_i(n) \text{ всюду определена и } \varphi_i(n) = 0 \text{ п. в.}\}.$$

Легко видеть, что $\mathcal{F}$ рекурсивно перечислимо, и, следовательно, для любой меры Φ существует рекурсивная t, такая, что $\mathcal{F} \subseteq C_t^{\Phi}$. Теперь мы покажем, что все классы сложности, которые содержат C_t, являются рекурсивно перечислимыми.

Теорема 10. *Пусть* Φ *— мера сложности и* t *такова, что* $\mathcal{F} \subseteq C_t^{\Phi}$. *Тогда для каждой рекурсивной* $\varphi_j(n) \geqslant t(n)$ *класс сложности* $C_{\varphi_j}^{\Phi}$ *рекурсивно перечислим.*

Доказательство. Пусть $\varphi_j(n) \geqslant t(n)$. Рассмотрим рекурсивно перечислимое множество функций

$$\{\varphi_{\sigma(i,p,w)} \mid i = 1, 2, 3, \ldots \text{ и } p, w = 0, 1, 2, 3, \ldots\},$$

где

$$\varphi_{\sigma(i,p,w)}(n) = \begin{cases} \varphi_i(n), & \text{если для каждого } k \leqslant p \ \Phi_i(k) \leqslant w \text{ и для} \\ & \text{каждого } k,\ p < k \leqslant n,\ \Phi_i(k) \leqslant \varphi_j(k), \\ 0 & \text{в остальных случаях.} \end{cases}$$

Таким образом, $\varphi_{\sigma(i,p,w)}$ равна φ_i, если сложность $\Phi_i(n) \leqslant \varphi_j(n)$ для всех $n > p$, а для $n \leqslant p$ имеем $\Phi_i(n) \leqslant w$, в остальных случаях $\varphi_{\sigma(i,p,w)}$ — функция с конечной основой. Если

$$\Phi_i(n) \leqslant \varphi_j(n) \text{ п. в.},$$

то для некоторых p и w

$$\varphi_{\sigma(i,p,w)} = \varphi_i,$$

и, таким образом, каждое φ_i из $C^{\Phi}_{\varphi_j}$ появится в нашем пересчете. Кроме того, так как все функции с конечной основой принадлежат $C^{\Phi}_{\varphi_j}$, то мы можем заключить, что имеем только функции из $C^{\Phi}_{\varphi_j}$. Таким образом, $\sigma(i, p, w)$ дает желаемый пересчет.

Следующий результат показывает, что существуют меры сложности, для которых «малые» классы сложности могут не быть рекурсивно перечислимыми.

Теорема 11. *Существует мера сложности* Φ *и рекурсивная* t, *такие, что* C^{Φ}_t *не является рекурсивно перечислимым.*

Доказательство. Пусть $\varphi_{i_1}, \varphi_{i_2}, \ldots$ — рекурсивный пересчет константных функций, такой, что $\varphi_{i_j}(n) = j$. Определим меру Φ следующим образом: для всех $k \neq i_j$, $j = 1, 2, \ldots$, пусть $\Phi_k(n) \geqslant n$[1]), и пусть

$$\Phi_{i_j}(n) = \begin{cases} 0, & \text{если } M_j(j) \text{ не останавливается за } h \text{ шагов,} \\ n & \text{в остальных случаях.} \end{cases}$$

Таким образом, C_0 состоит из всех тех константных функций $\varphi_{i_j}(n) = j$, для которых j-я машина Тьюринга M_j не останавливается на входе j. Следовательно, если $\varphi_{k_1}, \varphi_{k_2}, \ldots$ — пересчет C_0, то $\varphi_{k_1}(1), \varphi_{k_2}(1), \varphi_{k_3}(1), \ldots$ — пересчет множества $\{j \mid M_j(j)$ не останавливается$\}$. Получили противоречие, так как известно (и можно легко показать), что это множество не рекурсивно перечислимо.

Интересно отметить, что это доказательство существенно опирается на тот факт, что для этой меры конечное изменение функции φ_i может вызвать бесконечное изменение ее сложности Φ_i.

[1]) Для тех n, для которых $\varphi_k(n)$ определено. — *Прим. ред.*

Если это не имеет места, то все классы сложности рекурсивно перечислимы. Следующий результат использует это наблюдение.

Следствие. *Пусть* Φ — *мера, обладающая тем свойством, что если* $\varphi_i = \varphi_j$ *п. в., то* $\varphi_i \in C_i^{\Phi}$ *тогда и только тогда, когда* $\varphi_j \in C_i^{\Phi}$. *Тогда все классы сложности* C_i^{Φ} *рекурсивно перечислимы.*

Доказательство этого результата подобно доказательству теоремы 10.

5. МОДЕЛИРОВАНИЕ И ПАРАЛЛЕЛИЗМ

В этом разделе мы более детально рассмотрим диагональный процесс над классами сложности и выразим сложность этого процесса в терминах сложности «моделирования» и сложности «параллельного» выполнения двух вычислений. Этот подход позволит нам глубже проникнуть внутрь этого процесса и легко получить некоторые результаты о специфических мерах сложности.

В предыдущей части мы рассматривали следующий диагональный процесс, который работает для любых i, при условии, что φ_i — всюду определенная функция, давая функцию, не принадлежащую C_{φ_i}:

$$\varphi_{d(i)}(n) = \begin{cases} \varphi_{r(n)}(n) + 1, & \text{если } \Phi_{r(n)}(n) \leqslant \varphi_i(n), \\ 1 & \text{в противном случае,} \end{cases}$$

где $r(n)$ — любая вычислимая функция, обладающая тем свойством, что для каждого i существует бесконечно много n, таких, что $r(n) = i$. Теперь мы слегка изменим этот процесс, чтобы получить результаты, полученные ранее для специфических мер.

Вместо проверки того, что вычисление $\varphi_{r(n)}(n)$ использует не более $\varphi_i(n)$ шагов (т. е. верно ли, что $\Phi_{r(n)}(n) < \varphi_i(n)$), мы строим машину (находим алгоритм), которая вычисляет $\varphi_{r(n)}(n)$, и затем накладываем ограничение на то, как долго этой машине разрешается моделировать $\varphi_{r(n)}(n)$. Таким образом, в первом диагональном процессе мы ограничивали число шагов *моделируемого* вычисления $\varphi_{r(n)}(n)$. В новом процессе мы ограничиваем число шагов *моделирующего* вычисления $\varphi_{r(n)}(n)$. Более точно, пусть

$$\varphi_S(n) = \varphi_{r(n)}(n) + 1$$

и

$$\varphi_{D(i)}(n) = \begin{cases} \varphi_S(n), & \text{если } \Phi_S(n) < \varphi_i(n), \\ 1 & \text{в противном случае.} \end{cases}$$

Теперь мы выразим сложность моделирования и ограничения моделирования $\Phi_{D(i)}$ через сложности $\Phi_{r(n)}(n)$ и $\Phi_i(n)$.

Лемма 5. *Для любой меры* Φ *существуют рекурсивные функции* S *и* P, *такие, что*

$$S(n, \Phi_{r(n)}(n)) \geqslant \Phi_S(n)$$

и

$$P(n, \Phi_i(n)) \geqslant \Phi_{D(i)}(n)$$

для всех i *и почти всех* n.

Доказательство. Пусть

$$S(n, m) = \begin{cases} \Phi_S(n), & \text{если } \Phi_{r(n)}(n) = m, \\ 1 & \text{в противном случае.} \end{cases}$$

Чтобы построить P, поступим как в лемме о комбинировании, полагая

$$p(i, n, m) = \begin{cases} \Phi_{D(i)}(n), & \text{если } \Phi_i(n) = m, \\ 1 & \text{в противном случае,} \end{cases}$$

и затем положим

$$P(n, m) = \max_{i \leqslant n} p(i, n, m).$$

Следующий результат дает эффективный способ построения новых классов сложности.

Теорема 12. *Для любой меры существуют две рекурсивные функции* S *и* P, *такие, что для всех рекурсивных* φ_i *и* φ_j, *если*

$$S(n, \varphi_j(n)) < \varphi_i(n),$$

то существует рекурсивная f, *такая, что*

$$f \notin C_{\varphi_j}, \quad \text{но} \quad f \in C_{P(n, \Phi_i(n))}.$$

Комментарий. Интуитивно результат утверждает, что если потеря времени на моделирование вычислений, ограниченных функцией φ_j, не превышает φ_i (т. е. $S(n, \varphi_j(n)) < \varphi_i(n)$), то мы можем провести диагональ по этим вычислениям за время φ_i. Кроме того, затраты на ограничение моделирования после того, как оно использовало $\varphi_i(n)$ шагов, связаны с трудностью вычисления $\varphi_i(n)$ (а именно $\Phi_i(n)$) и задаются функцией $P(n, \Phi_i(n))$, которая описывает сложность привлечения механизма ограничения (или его выполнения параллельно с моделированием). Таким образом, сложность диагонального процесса не превосходит $P(n, \Phi_i(n))$, и, следовательно, существует функция $f \in C_{P(n, \Phi_i(n))}$, такая, что $f \notin C_{\varphi_j}$.

Доказательство. Пусть S и P — всюду определенные функции, удовлетворяющие лемме 5, и пусть S не убывает по вто-

рому аргументу (т. е. $k > l$ влечет за собой то, что для всех i $S(i, k) \geqslant S(i, l)$). Пусть $f(n) = \varphi_{D(i)}(n)$. Заметим, что

$$P(n, \Phi_l(n)) \geqslant \Phi_{D(l)}(n) \text{ п. в.},$$

и, следовательно, $\varphi_{D(l)}(n) \in C_{P(n, \Phi_l(n))}$.

Чтобы показать, что $\varphi_{D(l)}(n)$ не лежит в C_{φ_j}, рассмотрим любую φ_k из C_{φ_j}, сложность которой ограничена функцией φ_j, $\Phi_k(n) \leqslant \varphi_j(n)$ п. в. По лемме 5 для достаточно больших n

$$\Phi_S(n) \leqslant S(n, \Phi_{r(n)}(n)).$$

Кроме того, по построению r для сколь угодно больших n справедливо $r(n) = k$, следовательно, $\varphi_k(n) = \varphi_{r(n)}(n)$ и $\Phi_k(n) = \Phi_{r(n)}(n)$. Таким образом, для достаточно больших n

$$\Phi_S(n) \leqslant S(n, \Phi_k(n))$$

и так как $\Phi_k(n) \leqslant \varphi_j(n)$, то

$$S(n, \Phi_k(n)) \leqslant S(n, \varphi_j(n)).$$

По условию теоремы

$$S(n, \varphi_j(n)) < \varphi_l(n),$$

и, следовательно,

$$\Phi_S(n) < \varphi_l(n),$$

откуда следует, что

$$\varphi_{D(l)}(n) = \varphi_S(n) = \varphi_{r(n)}(n) + 1 = \varphi_k(n) + 1.$$

Но тогда $\varphi_k \neq \varphi_{D(l)}$. Отсюда, вспоминая, что $f(n) = \varphi_{D(l)}(n)$, имеем, что $f \in C_{P(n, \Phi_l(n))}$, $f \notin C_{\varphi_j}$, что и требовалось показать.

Слегка модифицируя доказательство теоремы 12, мы получаем следующий результат.

Следствие. *Для любой меры существуют две рекурсивные функции S и P', такие, что для любых рекурсивных φ_l и φ_j, если*

$$S(n, \varphi_j(n)) < \varphi_l(n),$$

то

$$C_{\varphi_j} \subset C_{P'(n, \Phi_l(n))}.$$

Этот результат устанавливает достаточное условие того, что один класс сложности собственно содержится в другом. К сожалению, эти результаты опять не эффективны по φ_i, а эффективны только по Φ_i, и теорема о пробелах утверждает, что это лучшее, что мы можем сделать.

Чтобы усилить наши представления и понимание общего подхода, мы рассмотрим теперь несколько специфических мер. Как будет видно, для специфических мер мы часто сможем получить

более тонкие границы для времени моделирования S и затрат P на проведение двух процессов параллельно.

Рассматриваемые специфические меры связаны с вычислениями на машинах Тьюринга. Чтобы упростить наши рассуждения и получить результаты в их первоначальной форме, мы немного модифицируем наши меры сложности. Будем рассматривать машины Тьюринга как распознаватели входных слов (таким образом, они вычисляют функции, принимающие значения 0 и 1), и параметром входа будет его длина l (а не число, представленное этим входом).

5.1. *Вычисления с ограничением на время.* Сначала мы опишем специфическую меру, основанную на времени вычисления на машинах Тьюринга.

Определение. Множество слов R называется *$T(l)$-допустимым*, если существует многоленточная машина Тьюринга, которая допускает множество R и для входов длины l использует не более $T(l)$ операций. Чтобы указать, что мы имеем дело с многоленточными машинами, мы обозначим этот класс через C_T^M.

(Следует отметить, что функция счета шагов, аргументом которой является длина входного слова (в k-буквенном алфавите, $k \geqslant 2$), строго говоря, не удовлетворяет аксиомам сложности, потому что машина может остановиться на одном входе длины l и не остановиться на другом. Таким образом, мы не можем однозначно определить время работы, основываясь только на длине входа. Наше определение преодолевает эту трудность, рассматривая C_T^M только для рекурсивных T и требуя, чтобы все входы длины l перерабатывались за время не более $T(l)$. Тот, кому требуется бо́льшая строгость, может избежать этих трудностей, ограничившись однобуквенным входным алфавитом, представляя n последовательностью из $n+1$ символов и полагая $l(n) = n+1$. При таком соглашении о входе все последующие результаты останутся неизменными.)

Чтобы использовать наш предыдущий результат, мы должны теперь определить «хорошую» границу моделирования S и «хорошие» затраты на ограничение P. При моделировании трудность состоит в том, чтобы моделировать машины с произвольным числом лент на машине с фиксированным числом лент. К счастью, существует остроумный способ моделирования, который дает хороший результат [2]. Доказательство этого результата весьма трудно, и, так как он используется в этой статье только один раз, мы это доказательство сюда не включили.

Лемма 6. *Существуют две вычислимые функции $r(n)$ и $c(n) = C(r(n))$ и двуленточная машина Тьюринга M, такая, что*

$$M(n) = M_{r(n)}(n) + 1,$$

и если $M_{r(n)}(n)$ останавливается, выполнив t операций, то $M(n)$ останавливается, выполнив не более

$$c(n)\,t\log t + c(n)$$

операций.

Комментарий.

(1) Для этой модели существует функция моделирования S, такая, что

$$S(n, t) \leqslant c(n)\,t\log t + c(n), \quad \text{где} \quad c(n) = C(r(n)).$$

(2) Функцию $r(n)$, скажем для трехбуквенного алфавита $\{0, 1, a\}$, можно выбрать так, что она зависит только от двоичного префикса слова, стоящего перед первой меткой a, и этот префикс интерпретируется как весьма непосредственный код программы машины Тьюринга. Таким образом, для каждой M_i существует слово x в алфавите $\{0, 1\}$, такое, что для каждого слова y в алфавите $\{0, 1, a\}$

$$r(xay) = i.$$

Эта декодирующая функция r имеет то преимущество, что, когда моделируется машина M_i, ее описание остается в одной и той же форме и операции, которые требуются для того, чтобы начать и выполнить один шаг моделирования, зависят только от префикса x, а не от длины всего входа. (Если рассуждать более строго, то следует заметить, что, когда мы используем однобуквенный входной алфавит, мы вынуждены использовать более тонкую технику кодирования, но, немного поразмыслив, мы смогли бы обеспечить ею три рассматриваемые модели.)

Теорема 13. *Пусть $T(l)$ — время работы некоторой многоленточной машины Тьюринга. Тогда для любой рекурсивной функции φ, если*

$$\lim_{l\to\infty} \frac{\varphi(l)\log\varphi(l)}{T(l)} = 0,$$

то

$$C^M_{\varphi(l)} \subset C^M_{T(l)}.$$

Доказательство. Мы дадим набросок доказательства, чтобы объяснить значение предельного условия и использование временной сигнализирующей. Из предельного условия следует, что для любого $c > 0$ и достаточно больших l

$$c\varphi(l)\log\varphi(l) < T(l).$$

Таким образом, $C^M_\varphi \subseteq C^M_T$ и для некоторых достаточно больших n

$$S(n, \varphi(l)) < T(l),$$

где n используется при вычислении $\varphi_{r(n)}(n)$, l — длина n. Но тогда мы можем провести диагональ по всем множествам R из класса C_φ^M за $T(l)$ операций, и, так как $T(l)$ является некоторой сигнализирующей для некоторой машины M_T, мы можем заставить работать M_T на отдельных лентах параллельно с моделированием и закончить процесс, когда M_T остановится. Таким образом, $P(n, T(l)) = T(l)$, и мы заключаем, что $\varphi_{D(i)} \in C_T^M$, но $\varphi_{D(i)} \notin C_\varphi^M$, что и требовалось показать ($\varphi_{D(i)}$ рассматривается как характеристическая функция некоторого множества слов).

5.2. *Одноленточные машины Тьюринга.* Теперь мы рассмотрим классы сложности, определенные через время работы одноленточных машин Тьюринга.

Определение. Множество слов R называется *$T(l)$-допустимым* одноленточной машиной Тьюринга, если существует одноленточная машина Тьюринга, которая допускает R и использует не более $T(l)$ операций при переработке входов длины l. Класс всех $T(l)$-допустимых множеств обозначается через C_T^1.

Для одноленточных машин проблема моделирования значительно проще, чем для многоленточных машин. На самом деле для простых функций $r(n)$ моделирование на одноленточной машине можно выполнить за время, не более чем в постоянное число раз превышающее время работы моделируемой машины, т. е.

$$S(n, t) \leqslant c(n)\, t + c(n), \quad \text{где} \quad c(n) = C(r(n)).$$

Это достигается тем, что копия описания $M_{r(n)}$ всегда находится рядом с местом, где происходит моделирование (скажем, на особой дорожке ленты). Так как длина описания фиксирована, так же, как и число символов ленты $M_{r(n)}$, мы видим, что каждый шаг вычисления машины $M_{r(n)}$ можно промоделировать за фиксированное число шагов машины M (включая и перемещение описания $M_{r(n)}$).

С другой стороны, прекращение процесса моделирования после того, как проделано $T(l)$ шагов моделирования, здесь выполнить труднее, чем на многоленточной модели (где мы просто запускали на особых лентах параллельно работающий счетчик остановки). Трудность проистекает из того факта, что мы должны выполнять два независимых вычисления на одной и той же ленте с помощью одной головки. Один из способов преодоления этой трудности состоит в том, чтобы проводить процессы на разных дорожках ленты и перемещать одну из них, если надо, для того, чтобы позиции головок для обоих процессов совпадали. Если мы поступаем таким образом, то мы, безусловно, заинтересованы в том, чтобы емкость дорожки, которую нам приходится передвигать, была не слишком большой. Это достигается выбором такой $T(l)$, которая является временной сигнализирующей одноленточной машины

Тьюринга, использующей не более $\log T(l)$ ячеек ленты. Легко видеть, что для такой $T(l)$

$$P(n, T(l)) \leqslant T(l) \log T(l).$$

Таким образом, мы получаем соответствующий результат для одноленточных машин.

Теорема 14. *Пусть $T(l)$ — время работы одноленточной машины Тьюринга, которая вычисляет $T(l)$ на $\log T(l)$ ячейках ленты. Тогда для любой рекурсивной φ, если*

$$\lim_{n\to\infty} \frac{\varphi(l)}{T(l)} = 0, \quad \text{то} \quad C_{\varphi}^{1} \subset C_{T \log T}^{1}.$$

Заметим, что эти два результата различны по структуре, так как для многоленточных машин моделирование дорого, а параллелизм бесплатный, тогда как для одноленточной модели моделирование дешево, а параллелизм дорог.

5.3. *Вычисление с ограничением на емкость ленты.* В заключение рассмотрим вычисления с ограничением на емкость ленты.

Определение. Множество слов R называется *допустимым с емкостью $L(l)$*, если существует машина Тьюринга M, которая допускает R, используя не более $L(l)$ ячеек ленты при переработке входов длины l. Класс всех множеств, допустимых с емкостью $L(l)$, обозначим через C_L^T.

Для вычислений с ограничением на емкость легко можно показать, что моделирование «стóит» только в постоянное число раз больше, чем исходное вычисление, т. е.

$$S(n, l) \leqslant c(n)\, l + c(n), \quad \text{где} \quad c(n) = C(r(n)),$$

а параллелизм бесплатный, т. е.

$$P(n, L(l)) \leqslant L(l)$$

(при условии, что $L(l)$ можно вычислить на $L(l)$ ячейках). Таким образом, для вычислений с ограничением на емкость ленты мы получаем следующий результат.

Теорема 15. *Если $L(l)$ вычислима с емкостью $L(l)$, то*

$$\lim_{n\to\infty} \frac{\varphi(l)}{L(l)} = 0 \quad \text{влечёт} \quad C_{\varphi}^{T} \subset C_{L}^{T}.$$

Таким образом, мы видим, что структура этого результата отражает тот факт, что для вычислений с ограничением на емкость ленты моделирование стóит дешево, а параллелизм — бесплатный.

6. ИМЕНОВАНИЕ КЛАССОВ СЛОЖНОСТИ

В этом разделе мы изучаем две взаимосвязанные проблемы. Первая возникает естественным образом, когда мы рассматриваем некоторые хорошо известные подклассы рекурсивных функций, вроде примитивно рекурсивных функций, и пытаемся локализовать их среди классов сложности при заданной мере. Обычно эти подклассы рекурсивных функций определяются через структуру своих алгоритмов, и это вполне убеждает в том, что они должны естественным образом располагаться среди классов сложности. Мы покажем на самом деле, в качестве приложения теоремы об объединении, что для многих мер сложности Φ существует рекурсивная функция t, такая, что C_t^{Φ} является в точности классом примитивно рекурсивных функций.

Вторая проблема возникает, когда мы ищем «хорошие» способы именования классов сложности. Вспомним, что теорема о пробелах утверждает, что при любой мере для любой рекурсивной r существует рекурсивная функция t, такая, что $C_t = C_{rot}$. Таким образом, мы можем построить функции, которые служат именами одного и того же класса сложности, но которые отстоят друг от друга так далеко, как мы пожелаем. Отсюда, по-видимому, следует, что в качестве имен классов сложности мы выбирали неподходящие функции. Оказывается, что так оно и есть на самом деле и что можно поступить гораздо лучше. Мы не можем всегда именовать классы сложности сигнализирующими функциями соответствующей меры, но мы покажем, что существует измеримое множество функций, которыми можно именовать все классы сложности.

Сначала мы покажем, что объединение любой рекурсивно перечислимой иерархии классов сложности (последовательности возрастающих классов сложности) само является классом сложности. Пусть $\{f_i \mid i = 1, 2, \ldots\}$ — рекурсивно перечислимое множество функций, таких, что для любых i и n

$$f_i(n) < f_{i+1}(n).$$

Достаточно показать, что существует рекурсивная функция, которая больше, чем $f_i(n)$ для каждого i и почти всех n, но в бесконечном числе точек меньше, чем каждая сигнализирующая функция, которая больше $f_i(n)$ для каждого i и почти всех n. Тогда класс сложности, определенный этой функцией, и будет $\bigcup_i C_{f_i}$.

Очевидно, что $t(n) = f_n(n)$ больше $f_i(n)$ для каждого i и почти всех n. Однако может существовать φ_j, для которой

(1) $\Phi_j(n) < t(n)$ для почти всех n и

(2) $\Phi_j(n) > f_i(n)$ для каждого i и бесконечно многих n.

Тогда φ_j лежит в C_t, но не в объединении C_{f_i}. Способ избежать этой трудности состоит в том, чтобы для каждого j делать предположение о том, что некоторая f_{i_j} мажорирует Φ_j. Если мы обнаруживаем, что для некоторого n имеем $\Phi_j(n) > f_{i_j}(n)$, то мы приписываем $t(n)$ значение, которое меньше, чем $\Phi_j(n)$, и предполагаем, что некоторая еще бо́льшая f_i мажорирует $\Phi_j(n)$. Если φ_j лежит в $\bigcup_i C_{f_i}$, то мы наконец найдем f_i, мажорирующую Φ_j, и t будет больше Φ_j почти всюду. С другой стороны, если φ_j не содержится в объединении, то t будет меньше Φ_j бесконечно часто (б. ч.) и, таким образом, φ_j не будет принадлежать C_t. Мы формализуем эту интуитивную идею в доказательстве теоремы об объединении.

Теорема 16 (теорема об объединении). Пусть $\{f_i \mid i = 1, 2, \ldots\}$ — *рекурсивно перечислимое множество рекурсивных функций, такое, что для любых* i *и* n *имеем* $f_i(n) < f_{i+1}(n)$. *Тогда существует рекурсивная функция* $t(n)$, *такая, что* $C_t = \bigcup_i C_{f_i}$.

Доказательство. Построим t, такую, что

(1) для каждого i, $t(n) \geqslant f_i(n)$ п. в.,

(2) для каждого j, если $\Phi_i(n) > f_j(n)$ б. ч., то $t(n) < \Phi_i(n)$ б. ч.

При построении t мы будем использовать список номеров $i_1, i_2, i_3, \ldots$. Список будет обновляться, и на n-м шаге, когда мы вычисляем $t(n)$, интерпретация состоит в следующем: $i_j = k$ означает, что в данный момент мы предполагаем, что $f_k \geqslant \Phi_j$ почти всюду. Чтобы вычислить $t(n)$, мы проверяем, справедливы ли все наши предположения. Проверка наших предположений происходит следующим образом: мы начинаем с наименьшего k, такого, что $k = i_j$, и определяем, выполняется ли $f_k(n) \geqslant \Phi_j(n)$; если это условие выполняется для всех k, то мы полагаем $t(n) = f_n(n)$ и делаем новое предположение, что $f_n \geqslant \Phi_n$ п. в., полагая $i_n = n$. С другой стороны, если одно из наших предположений ошибочно, $f_k(n) < \Phi_j(n)$, то мы полагаем $t(n) = f_k(n)$ и изменяем наше предположение на $\Phi_j \leqslant f_n$ п. в., полагая $i_j = n$. В этом случае мы также добавляем новое предположение $i_n = n$ и повторяем процесс для $n = n + 1$. Ниже этот процесс описан более формально.

Построение t. Вначале i_j не определено для каждого j, т. е. список пуст. Полагаем $n = 1$. Переходим к шагу n.

Шаг n. Пусть k — наименьшее число, такое, что существует j, для которого j-й элемент списка есть k (т. е. $i_j = k$) и $f_{i_j}(n) < \Phi_j(n)$. Если существует больше чем одно такое j, выбираем наименьшее. Определяем $t(n) = f_{i_j}(n)$ и полагаем $i_j = i_n = n$

и $n = n + 1$. Переходим к шагу n. Если такого j не существует, определяем $t(n) = f_n(n)$. Полагаем $i_n = n$ и $n = n + 1$. Переходим к шагу n. Докажем, что $C_t = \bigcup_i C_{f_i}$.

(1) Если $\varphi_g \in \bigcup_i C_{f_i}$, то существует i, такое, что $\varphi_g \in C_{f_i}$, и, следовательно,

$$\Phi_g \leqslant f_i \text{ п. в.}$$

Но $t \geqslant f_i$ почти всюду, так как в конце концов для каждого j либо i_j примет значение, большее чем i, либо i_j таково, что $f_{i_j}(n)$ мажорирует $\Phi_j(n)$. Начиная с этого места $t \geqslant f_i$, значит $\varphi_g \in C_t$.

(2) Если $\varphi_g \in C_t$, то $\Phi_g \leqslant t$ п. в., и, таким образом, существует f_k, такая, что $f_k \geqslant \Phi_g$ п. в. В противном случае бесконечное число раз i_g было бы наименьшим числом списка, таким, что $f_{i_g} < \Phi_g$, и тогда t в бесконечном числе точек было бы меньше, чем Φ_g — противоречие. Но если $\varphi_g \in C_{f_k}$, то $\varphi_g \in \bigcup_i C_{f_i}$.

Рассмотрим теперь примитивно рекурсивные функции и меру сложности, которая «считает шаги» одноленточной машины Тьюринга. Мы утверждаем, что если g примитивно рекурсивна, то существует примитивно рекурсивная t, такая, что $g \in C_t$. Причина этого состоит в том, что функции $s(x) = x + 1$, $O(x) = 0$ и $I_i^n(x_1, \ldots, x_n) = x_i$ принадлежат классу сложности, определенному некоторой примитивно рекурсивной функцией; рекурсивная функция, ограничивающая сложность суперпозиции и примитивной рекурсии, является примитивно рекурсивной, и класс примитивно рекурсивных функций замкнут относительно суперпозиции и примитивной рекурсии.

Мы утверждаем далее, что любой класс сложности, определенный примитивно рекурсивной функцией, содержит только примитивно рекурсивные функции. Основанием для этого служит то, что примитивной рекурсии достаточно, чтобы промоделировать машину Тьюринга, работающую с примитивно рекурсивным числом шагов.

Заметим, что нам осталось только понять, что существует рекурсивно перечислимая последовательность примитивно рекурсивных функций, такая, что каждая примитивно рекурсивная функция мажорируется некоторой функцией из-за этой последовательности. Отсюда будет следовать существование класса сложности, определяемого временной сигнализирующей и состоящего в точности из примитивно рекурсивных функций. Пусть $g_1, g_2, \ldots$ — такая последовательность, тогда последовательность $f_1, f_2, \ldots$, где

$$f_i(n) = \max\{g_1(n), g_2(n), \ldots, g_i(n)\} + 1,$$

удовлетворяет условию теоремы об объединении. Желаемый результат получается немедленно, и мы устанавливаем его в качестве следствия теоремы об объединении.

Следствие. *Существует рекурсивная функция t, такая, что множество функций, вычислимых на одноленточных машинах Тьюринга с границей времени t, в точности совпадает с множеством примитивно рекурсивных функций. Тот же результат выполняется для многоленточных машин Тьюринга при ограничении на время, а также для машин Тьюринга при ограничении на емкость.*

Заметим, что для любой меры, которая связана примитивно рекурсивной функцией с числом шагов одноленточной машины Тьюринга, примитивно рекурсивные функции образуют класс сложности. Другое интересное наблюдение заключается в том, что класс сложности, состоящий в точности из примитивно рекурсивных функций, нельзя именовать примитивно рекурсивной функцией. Если бы он был именован примитивно рекурсивной функцией t, то t лежала бы на некотором уровне в иерархии Гжегорчика и, таким образом, класс C_t не содержал бы более высоких уровней. Это означает, что функция, которая именует этот класс сложности, очень сложна.

Кроме того, мы видели в теореме о пробелах, что один и тот же класс сложности можно именовать совершенно разными функциями. А именно, для любой меры и для любой рекурсивной функции r мы можем построить рекурсивную функцию t, такую, что $C_t = C_{rot}$. Во всех таких случаях функции оказываются очень сложными в том смысле, что их сложность сильно отличается от их величины. Это приводит нас к проблеме именования всех классов сложности функциями, которые не слишком сложны.

Предыдущие наблюдения приводят нас к проблеме нахождения рекурсивно перечислимого множества функций, которые именуют все классы сложности заданной меры и которые обладают тем свойством, что сложность каждой функции рекурсивно ограничена ее величиной, т. е. „честного“ множества функций. Следующий результат, теорема об именовании утверждает, что всегда можно именовать все классы сложности с помощью некоторого измеримого множества функций. Наша стратегия состоит в том, чтобы для каждой рекурсивной φ_t найти $\varphi_{t'}$ из измеримого множества, такую, что $C_{\varphi_t} = C_{\varphi_{t'}}$. Определяя $\varphi_{t'} = \max(\Phi_t, \varphi_t)$, мы получаем гарантию, что $\varphi_{t'}$ — „честная“ функция. Однако может случиться, что для некоторого i функция φ_i принадлежит $C_{\varphi_{t'}}$, но не принадлежит C_{φ_t}. Чтобы преодолеть эту трудность, можно сделать $\varphi_{t'}(n)$ меньше $\varphi_t(n)$ для бесконечно многих n. Значения n выбираются так, чтобы $\Phi_{t'}(n)$ также уменьшалось, оставляя функ-

цию $\varphi_{t'}$ „честной". Делая $\varphi_{t'}(n)$ меньше $\varphi_t(n)$, мы должны гарантировать, что $\varphi_{t'}$ не окажется в бесконечном числе точек меньше некоторой Φ_j, которая почти всюду меньше φ_t. Иначе φ_j была бы элементом C_{φ_t}, но не $C_{\varphi_{t'}}$. Следующая теорема доказывается с помощью формализации этих идей.

Теорема 17 (Теорема об именовании). *Для каждой меры* Φ *существует измеримое множество, именующее все классы сложности.*

Доказательство[1]). Мы хотим показать, что существует рекурсивная функция r, такая, что для каждой рекурсивной φ_i можно построить рекурсивную $\varphi_{t'}$, обладающую свойствами:

(a) $C_{\varphi_t} = C_{\varphi_{t'}}$ (т. е. $\Phi_i \leqslant \varphi_t$ п. в. $\Leftrightarrow \Phi_i \leqslant \varphi_{t'}$ п. в.)

(b) $\Phi_{t'}(n) \leqslant r(n, \varphi_{t'}(n))$ п. в. (т. е. функции $\varphi_{t'}$ — „честные").

Условие «честности» (b) гарантирует, что функции $\varphi_{t'}$ содержатся в некотором измеримом множестве Ψ. Чтобы обеспечить выполнение этого условия, $\varphi_{t'}$ будет определена с помощью некоторой согласованной процедуры таким образом, что малые значения $\varphi_{t'}$ на больших аргументах будут, вообще говоря, определены раньше, чем большие значения $\varphi_{t'}$ на малых аргументах. Условию (a) трудно удовлетворить по той причине, что, когда определяется $\varphi_{t'}(y)$, может нехватить времени для вычисления $\varphi_{t'}(y)$. Для устранения этой трудности $\varphi_{t'}(y)$ определяется так, чтобы удовлетворить условиям, которые зависят от значения φ_t на некотором легче вычислимом аргументе, отличном от y. Величину $\varphi_{t'}(y)$ нужно выбирать так, чтобы удовлетворить двум конфликтующим условиям: во-первых, она должна быть больше, чем «число шагов», которые затрачивают на аргументе y те программы, которые оказываются ограниченными функцией φ_t; во-вторых, она должна быть меньше числа шагов тех программ, для которых обнаружится, что это число превосходит значения φ_t на некоторых аргументах. Этот конфликт разрешается с помощью описанного ниже механизма приоритета.

Вычисление $\varphi_{t'}$ происходит по этапам, начиная с этапа 0. На этапе x рассматриваются программы (номера функций) 0, 1, ..., x, которые упорядочиваются по приоритету. Это упорядочивание остается тем же, каким оно было на предыдущем шаге, если не меняется на данном шаге. На каждом этапе некоторые программы могут потребовать специальной пометки. Это означает, что их сигнализирующие где-то превысили φ_t, и требуется, чтобы они где-то превысили $\varphi_{t'}$. Если на данном этапе положение меток не

[1]) Доказательство, данное авторами обзора, некорректно. Приводимое здесь доказательство следует оригиналу [18]. — *Прим. перев.*

меняется, то они остаются на тех же местах, где были на предыдущем этапе. Пусть $\sigma(x) \leqslant x$ — рекурсивная функция, которая принимает в качестве значения каждое натуральное число бесконечно много раз.

Этап x: часть I. Приписываем низший приоритет программе x. Считаем, что x не отмечена. Проделываем x шагов вычислений, в ходе которых ищем z, такое, что $\varphi_t(z)$ можно вычислить на этом этапе, но еще не вычислили на предыдущих этапах. Если такое z нельзя найти за отведенное для этого время, переходим к части II. Иначе, моделируем программы $0, 1, \ldots, x$ на входе z в течение $\varphi_t(z)$ «шагов» (в смысле сигнализирующей Φ) и отмечаем те программы, которые не заканчивают вычисление. Переходим к части II.

Часть II. Пусть $y = \sigma(x)$. Смотрим, определено ли $\varphi_{t'}(y)$ на предыдущих этапах. Если да, то переходим к этапу $x + 1$. В противном случае смотрим, каковы приоритеты и отметки у программ $0, 1, \ldots, y$ в конце части I этапа y. Пытаемся найти программу P с наивысшим приоритетом (на этапе y), такую, что она на этапе y имеет метку и что все программы, которые

(1) на этапе y не имеют меток,

(2) на этапе y имеют приоритет выше, чем P, требуют для вычисления на входе y меньше шагов, чем требует P на том же входе y.

Если такую программу P можно найти, полагаем $\varphi_{t'}(y)$ равной наибольшему из чисел шагов, которые требуют программы, удовлетворяющие условиям (1) и (2), приписываем программе P низший приоритет (на этапе x), стираем с нее метку и переходим к этапу $x + 1$.

Однако описанную попытку найти P нужно прекратить, если

(3) она требует более x шагов,

(4) она требует шагов более, чем требуется для вычисления $\varphi_t(y)$.

Если первым выполнится условие (3), переходим к этапу $x + 1$. Если первым выполнится условие (4), делаем $\varphi_{t'}(y)$ больше, чем $\varphi_t(y)$ и $\Phi_t(y)$, переходим к этапу $x + 1$.

Описание процедуры окончено. Условие (4) гарантирует, что $\varphi_{t'}(y)$ будет определена для каждого y, так как $\varphi_t(y)$ определена. Более того, $\varphi_{t'}(y)$ не удается определить на этапе x, для которого $y = \sigma(x)$, только потому, что $\varphi_{t'}(y)$ слишком велико, чтобы вычислить его за время, отведенное на этапе x. Так как время, отведенное на этапе x, ограничено фиксированной (не зависящей от φ_t) рекурсивной функцией от x, можно показать, что число шагов, требующееся для вычисления $\varphi_{t'}$, рекурсивно ограничено ее величиной, т. е. $\Phi_{t'}(y) \leqslant r(y, \varphi_{t'}(y))$п. в. (Например, для машин Тьюринга время вычисления $\varphi_{t'}$ будет ограничено по крайней мере элементарной функцией, в которую подставлена $\varphi_{t'}$.)

Доказательство того, что $C_{\varphi_t} = C_{\varphi_{t'}}$, основано на следующих эквивалентностях: (α) $\Phi_i > \varphi_t$ б. ч. $\Leftrightarrow$ (β) программа i на бесконечно многих этапах вновь получает метку $\Leftrightarrow$ (γ) $\Phi_i > \varphi_{t'}$ б. ч.

Легко убедиться, что (β) $\Rightarrow$ (γ). Если на некотором этапе x программа i теряет метку, то для $y = \sigma(x)$ полагаем $\varphi_{t'}(y) = \max\{\Phi_j(y)\}$ для тех программ j, удовлетворяющих условиям (1) и (2), для которых $\Phi_j(y) < \Phi_i(y)$. Следовательно, $\varphi_{t'}(y) < \Phi_i(y)$. Но в силу (β) программа i бесконечно часто теряет метку. Отсюда получаем (γ).

Покажем, что (α) $\Rightarrow$ (β). Пусть, напротив, программа i, начиная с некоторого этапа, либо навсегда теряет метку, либо всегда сохраняет ее. В первом случае сразу имеем $\Phi_i(y) \leqslant \varphi_t(y)$ п. в. Во втором случае рассмотрим тот этап процедуры, после которого программа i навсегда сохраняет метку и приоритет, а программы из множества $\{0, 1, \ldots, i-1\}$, имевшие, но потерявшие приоритет, более высокий, чем этот приоритет программы i, уже его потеряли. Среди программ, сохранивших более высокий приоритет, нас будут интересовать те, которые не имеют меток (и, стало быть, далее их не получат). Пусть Π_i — множество таких программ. Для каждой $j \in \Pi_i$ имеем $\Phi_j(y) \leqslant \varphi_t(y)$ п. в. Тогда для каждого достаточно большого y найдется $j \in \Pi_i$, такое, что $\Phi_j(y) \leqslant \Phi_i(y)$. В противном случае пришлось бы вопреки предположению снять метку с программы i. Следовательно, $\Phi_i(y) \leqslant \varphi_t(y)$ п. в.

Аналогично проверяются остальные импликации.

7. ОБЪЕМ МАШИН

Мы заканчиваем изучение сложности вычислений установлением некоторых связей между объемом алгоритмов (машин) и их эффективностью. Так же, как мы абстрагировали понятие сложности алгоритма, мы можем абстрагировать понятие «объема» алгоритма. Мы имеем в виду извлечь понятие сложности описания алгоритма. Объем программы вычислительной машины можно измерять числом команд, а «объем» машины Тьюринга — произведением числа состояний на число символов ленты.

Определение. Пусть s — отображение натуральных чисел в натуральные числа. Мы говорим, что s является *мерой объема машин* для допустимой нумерации всех частично рекурсивных функций $\varphi_1, \varphi_2, \varphi_3, \ldots$, если выполняются условия:

(1) для каждого j существует конечное число номеров i, таких, что $s(i) = j$;

(2) существует рекурсивная функция, дающая число алгоритмов каждого объема [1]).

[1]) В оригинале имеется еще одна аксиома, которая делает все определение некорректным. — *Прим. ред.*

Рассмотрим изображение алгоритмов словами в некотором конечном алфавите. Пусть объем алгоритма — это число символов. Первая аксиома отражает тот факт, что существует конечное число слов любой данной длины. Вторая аксиома учитывает то обстоятельство, что мы можем проверить «формат» слова, чтобы определить, изображает ли оно алгоритм. Эти две аксиомы эквивалентны эффективной нумерации алгоритмов в порядке возрастания объема (среди изображений одного и того же объема порядок не имеет значения).

Легко показать, что все меры объема рекурсивно связаны друг с другом.

Теорема 18. *Пусть s и $\hat{s}$ — две меры объема. Существует рекурсивная функция g, такая, что*

$$g(s(i)) \geqslant \hat{s}(i) \quad \text{и} \quad g(\hat{s}(i)) \geqslant s(i)$$

для всех i.

Доказательство. Пусть

$$g(m) = \max_{i \in S_m} \{s(i), \hat{s}(i)\},$$

где $S_m = \{i \mid$ либо $s(i) \leqslant m$, либо $\hat{s}(i) \leqslant m\}$. Так как S_m для каждого m есть конечное множество, а функции s и $\hat{s}$ рекурсивны, то g — рекурсивная функция.

Очевидно, что

$$g(s(i)) \geqslant \hat{s}(i) \quad \text{и} \quad g(\hat{s}(i)) \geqslant s(i)$$

для всех i.

Поводом для рассмотрения объемов алгоритмов является изучение экономичности различных формализмов, представляющих алгоритмические процессы. При написании машинных программ для функций, возникающих в практических ситуациях, можно обойтись без операторов условного перехода и написать программу, процесс выполнения которой определяется очень просто вложенной цикловой структурой. Более того, время выполнения этой программы не будет сильно отличаться от времени выполнения произвольной программы. Возникает вопрос, зачем вообще мы используем условные переходы. Ответ заключается в эффективности представления. Сравним, например, объем изображения примитивно рекурсивной функции, использующего примитивно рекурсивную схему, с представлением посредством машины Тьюринга, которая вычисляет ту же функцию. Для произвольной рекурсивной функции f мы можем указать примитивно рекурсивную функцию φ, такую, что минимальное число символов в любой схеме примитивной рекурсии для φ больше $f(m)$, где m — число символов в описании некоторой машины Тьюринга, вычисляющей φ.

Чтобы получить этот результат, мы сначала покажем, что в любой бесконечной последовательности алгоритмов имеются неэффективные изображения[1]).

Теорема 19. *Пусть g — рекурсивная функция с бесконечным множеством значений (g перечисляет бесконечную последовательность номеров алгоритмов). Пусть f — рекурсивная функция. Существуют i и j, такие, что*

(1) $\varphi_i = \varphi_{g(j)}$,
(2) $f(s(i)) < s(g(j))$.

Замечание. Интуитивную идею, лежащую в основе теоремы, следует пояснить. Так как существует конечное число алгоритмов любого данного объема, то в любой бесконечной рекурсивно перечислимой (р. п.) последовательности алгоритмов существует бесконечная р. п. подпоследовательность, в которой объемы алгоритмов растут сколь угодно быстро. Пусть g перечисляет быстро растущую последовательность. По данному k функцию $\varphi_{g(k)}(n)$ можно вычислить с помощью программы фиксированного объема, а именно

$$\varphi(k, n) = \varphi_{g(k)}(n).$$

Таким образом, объем программ, необходимых для вычисления $\varphi_{g(k)}$, растет со скоростью, необходимой для вычисления k, тогда как объем соответствующих программ из р. п. списка растет очень быстро и разность между длинами двух описаний становится сколь угодно большой.

Доказательство. Так как уменьшение объема не зависит от меры (т. е. если существует произвольное уменьшение объема для одной меры, то существует произвольное уменьшение объема для всех мер), то нам достаточно доказать теорему только для случая, когда $s(i)$ есть длина описания i-й машины Тьюринга. Без потери общности предполагаем, что

$$s(g(n+1)) > f(s(g(n))).$$

(Так как существует только конечное число машин любого объема, просто удалим из последовательности, определяемой функцией g, некоторые машины, пока не останутся достаточно большие машины.) Рассмотрим машину $\varphi_{i(k)}$, которая записывает k на своей ленте, вычисляет $g(k)$ и затем вычисляет $\varphi_{g(k)}$. Объем $\varphi_{i(k)}$ есть некоторая константа плюс k (т. е. объем машины Тьюринга, вычисляющей g, плюс объем универсальной машины, моделирующей

[1]) Здесь эффективность изображения алгоритма понимается в смысле выбранной меры объема алгоритмов. — *Прим. ред.*

φ_g, плюс состояния, необходимые для записи k). Таким образом, $\varphi_{i(k)} = \varphi_{g(k)}$. Увеличение k на 1 увеличивает объем $i(k)$ на 1, а $g(k)$ — на f. Таким образом, для достаточно большого k

$$f(s(i(k))) < s(g(k)).$$

Чтобы завершить доказательство, положим $j = k$ и $i = i(k)$.

В качестве следствия теоремы 19 получаем, что существует примитивно рекурсивная функция, примитивно рекурсивное описание которой гораздо больше, чем общерекурсивный алгоритм, вычисляющий эту функцию. Каждая примитивно рекурсивная функция имеет по крайней мере одно наименьшее примитивно рекурсивное описание. Список наименьших описаний рекурсивно перечислим. (Первой записываем наименьшую схему. Начинаем вычислять функцию, определенную i-й схемой на входе n для все больших и больших i и n. Записываем схему, когда обнаружится, что она вычисляет функцию, отличную от всех функций, вычисляемых меньшими схемами.) Пусть g перечисляет наименьшие схемы. Пусть $f(n) = n^2$. Тогда, применяя теорему 19, мы получаем примитивно рекурсивную функцию φ, длина наименьшего примитивно рекурсивного описания которой не меньше квадрата числа символов некоторого общерекурсивного алгоритма, вычисляющего φ.

Заметим, что вместо примитивно рекурсивных функций мы могли бы взять любой рекурсивно перечислимый класс рекурсивных функций и получить тот же результат.

8. ИСТОРИЧЕСКИЕ ЗАМЕЧАНИЯ

Интерес к сложности вычислений можно обнаружить во многих областях математики, в которых алгоритмы определяются, анализируются и сравниваются по их эффективности. С другой стороны, вряд ли в этих областях математики предпринимались систематические попытки развить теорию сложности вычислений, в которой изучались бы количественные проблемы, связанные с вычислениями. Проблемы сложности первоначально не были достаточно хорошо определены, и даже в период быстрого развития конструктивной математики в первой половине нашего века на них не смотрели как на самостоятельную область. В то время были определены некоторые классификации подклассов рекурсивных функций, но главный интерес был скорее в единообразном построении все более и более широких подклассов рекурсивных функций, чем в изучении внутренней вычислительной сложности функций. Появление электронных вычислительных машин и общее развитие науки о процессах вычисления несомненно придало особый смысл

количественной теории вычислений, а теория рекурсивных функций предоставила этой теории формализмы и исходные модели.

Первая попытка аксиоматического подхода к измерению трудности вычислений была сделана Рабином [3, 4], который аксиоматизировал понятие «меры на доказательствах и длины вычисления функции» и получил некоторые начальные результаты для этих мер. Первое систематическое исследование одной специфической меры сложности вычислений и изучение соответствующих классов сложности принадлежит Хартманису и Стирнзу [5, 6], которые дали также название «сложность вычислений» этой новой области исследований. В докладе Кобхэма [7] обсуждалась важность исследования количественных аспектов вычисления и приводились некоторые дальнейшие результаты. В работах Рабина, Хартманиса и Стирнза и Кобхэма была ясно установлена важность этой новой области исследований, были получены результаты, достаточные, чтобы считать их многообещающими в изучении сложности вычислений, а сама область получила соответствующее название [1]).

Общий аксиоматический подход к сложности вычислений, использованный в этом обзоре, был сформулирован Блюмом [10] под влиянием работы Рабина. Блюм получил также большинство результатов, содержащихся в разд. 2 этой статьи, а некоторые из них являются обобщением результатов Хартманиса и Стирнза, полученных для времени вычислений на машине Тьюринга. Теорема об ускорении также принадлежит Блюму, хотя доказательство, данное в разд. 3, является новым. Оно отличается от первоначального доказательства тем, что в нем не используется теорема о рекурсии, которая затемняет (в чем мы уверены) простоту центрального диагонального процесса. Наблюдение, выраженное в форме теоремы 6, является новым и используется далее в разд. 5. Теорема о пробелах была открыта независимо Трахтенбротом [11] и Бородиным [12]. Она показывает, что теорему 6 нельзя улучшить. Теорема о пробелах дает изящное оправдание для использования введенных Блюмом измеримых множеств функций, которые входят в теоремы 8 и 9. Рекурсивная перечислимость классов сложности изучалась Хартманисом и Стирнзом [6] и Янгом [13]. Доказательство того, что для некоторых мер существуют классы сложности, которые не рекурсивно перечислимы, принадлежит Ф. Льюису [14] и Робертсону и Ландвеберу [15].

Материал о моделировании и параллелизме в разд. 5 является новым, а результаты этого раздела о классах сложности

---

[1]) В Советском Союзе начало исследованиям по теории сложности вычислений положили работы Г. С Цейтина [8] и Б. А. Трахтенброта [9], которые, по-видимому, не известны авторам обзора, так же как и более поздние работы этих и других советских авторов. — *Прим. перев.*

при ограничении на время и емкость принадлежат Хенни и Стирнзу [2], Хартманису и Стирнзу [6], Хартманису [16] и Стирнзу, Хартманису и Льюису [17].

Теорема об объединении и теорема об именовании из разд. 6 принадлежат Маккрайту и Мейеру [18]. Результат о том, что теорема об именовании оставляет все же произвольно широкие пробелы «снизу», принадлежит Констэблу [19]. Материал об объеме машин взят из работы Блюма [20].

На развитие теории сложности вычислений дальнейшее влияние оказывали многие работы и результаты, которые не были явно использованы в этой статье. Некоторые из них перечислены в нашей короткой библиографии[1]). Полную библиографию, охватывающую самые последние работы, смотри в «Библиографии по сложности вычислений», составленной Айлендом и Фишером и помещенной в нашей библиографии[2]).

СПИСОК ЛИТЕРАТУРЫ*

1*. Blum M., On effective procedures for speeding up algorithms, *J. ACM*, **18** (1971), 290—305.

2. Hennie F. C., Stearns R. E., Two-tape simulation of multi-tape Turing machines, *J. ACM*, **13** (1966), 533—546. (Русский перевод: Ф. К. Хенни, Р. Е. Стирнз, Моделирование многоленточной машины Тьюринга на двуленточной, Сб. пер. «Проблемы математической логики», М. (1970), 194—212.)

3. Rabin M. O., Speed of computation of functions and classification of recursive sets, Proc. 3-rd Conf. of Scientific Societies, Israel, 1—2, 1959.

4. Rabin M. O., Degrees of difficulty of computing a function and a partial ordering of recursive sets, Techn. Rep. 2, Hebrew U., Jerusalem, Israel, 1960.

5. Hartmanis J., Stearns R. E., Computational complexity of recursive sequences, IEEE Proc. 5th Ann. Symp. on Switching Circuit Theory and Logical Design, 1964, 82—90.

6. Hartmanis J., Stearns R. E., On the computational complexity of algorithms, *Trans. Amer. Math. Soc.*, **117** (1965), 285—306. (Русский перевод: Дж. Хартманис, Р. Е. Стирнз, О вычислительной сложности алгоритмов, Киб. сборник (новая серия), вып. 4, М. (1967), 57—85.)

7. Cobham A., The intrinsic computational difficulty of functions, Proc. 1964 Intern. Congr. for Logic, Methodology and Philosophy of Science, Amsterdam, 1964, 24—30.

8*. Цейтин Г. С., Оценка числа шагов при применении нормального алгоритма, Математика в СССР за сорок лет, т. I, М., 1959, стр. 44—45.

9*. Трахтенброт Б. А., Сигнализирующие функции и табличные операторы, Уч. записки Пензенского Гос. пед. ин-та, IV, 1956, стр. 75—87.

10. Blum M., A machine-independent theory of complexity of recursive functions, *J. ACM*, **14** (1967), 322—336. (Русский перевод: М. Блюм, Машинно-незави-

[1]) Настоящий обзор написан в середине 1970 г. — *Прим. ред.*

[2]) Библиография, составленная А. Д. Коршуновым [22], охватывает работы вплоть до 1971 г. Дополнительную литературу можно найти также в [11], [21] и обзоре Фишера, упомянутом в библиографии авторов. — *Прим. перев.*

* Звездочкой отмечена литература, добавленная переводчиком. Ссылки на краткие сообщения заменены более полными работами, если таковые имеются. — *Прим. перев.*

симая теория сложности рекурсивных функций, Сб. пер. «Проблемы математической логики», М. (1970), 401—422.)
11. Трахтенброт Б. А., Сложность алгоритмов и вычислений, Изд. НГУ, Новосибирск, 1967.
12. Borodin A., Computational complexity and existence of complexity gaps, *J. ACM*, **19** (1972), 158—174.
13. Young P. R., Toward a theory of enumerations, *J. ACM*, **16** (1969), 328—348. (Русский перевод: П. Р. Янг, К теории перечислений, Киб. сборник (новая серия), вып. 8, М., (1971), 201—231.)
14. Lewis F. D., Unsolvability considerations in computational complexity, Conf. Rec. 2nd Ann. ACM Symp. on Theory of Computing, 1970, 22—30.
15. Landweber H. H., Robertson E. L., Recursive properties of abstract complexity classes, Conf. Rec. 2nd Ann. ACM Symp. on Theory of Computing, 1970, 31—36.
16. Hartmanis J., Computational complexity of one-tape Turing machine computations, *J. ACM*, **15** (1968), 325—339. (Русский перевод: Дж. Хартманис, Сложность вычислений на одноленточных машинах Тьюринга, Сб. пер. «Проблемы математической логики», М., (1970), 282—300.)
17. Stearns R. E., Hartmanis J., Lewis P. M., II, Hierarchies of memory limited computations, 1965. IEEE Conf. Rec. on Switching Circuit Theory and Logical Design, 179—190. (Русский перевод: Р. Е. Стирнз, Дж. Хартманис, П. М. Льюис II, Иерархии вычислений с ограниченной памятью, Сб. пер. «Проблемы математической логики», М., (1970), 301—319.)
18. McCreight E. M., Meyer A. R., Classes of computable functions defined by bounds on computation: preliminary report. Conf. Rec. ACM Symp. on Theory of Computing, 1969, 79—88.
19. Constable R. L., Upward and downward diagonalization over axiomatic complexity classes, Tech. Rep., 69—32, Dep. of Computer Science, Cornell U., Ithaca, 1969.
20. Blum M., On the size of machines, *Inf. Contr.*, **11** (1967), 257—265. (Русский перевод: М. Блюм, Об объеме машин в сб. Проблемы математической логики, «Мир», М., 1970, стр. 423—431.)

Рекомендательный список литературы

1. Axt P., Enumeration and the Grzegorczyk hierarchy, *Z. Math. Logik und Grundlagen Math.*, **9** (1963), 53—65.
2. Becvar J., Real-time and complexity problems in automata theory, *Kybernetika*, **1** (1965), 475—497.
3. Blum M., On effective procedures for speeding up algorithms, Conf. Rec. ACM Symp. on Theory of Computing, 1969, 43—53.
4. Borodin A., Constable R. L., Hopcroft J. E., Dense and nondense families of complexity classes, IEEE Conf. Rec. 10th Ann. Symp. on Switching and Automata Theory, 1969, 7—19.
5. Cobham A., On the Hartmanis-Stearns problem for a class of tag machines, IEEE Conf. Rec. 9th Ann. Symp. on Switching and Automata Theory, 1968, 51—60.
6. Constable R. L., The operator gap, *J. ACM*, **19** (1972), 175—183.
7. Fischer P. C., Multi-tape and infinite-state automata — a survey, *Comm. ACM*, **8** (1965), 799—805. (Русский перевод: П. Фишер, Многоленточные и бесконечные автоматы, Киб. сборник (новая серия), вып. 5, М. (1968), 64—80.)
8. Fischer P. C., The reduction of tape reversals for off-line one-tape Turing machines, *J. Comp. System Sci.*, **2** (1968), 136—147.
9. Fischer P. C., Hartmanis J., Blum M., Tape reversal complexity hierarchies, IEEE Conf. Rec. 9th Ann. Symp. on Switching and Automata Theory, 1968, 373—382.

10. Grzegorczyk A., Some classes of recursive functions, *Rozprawy Math.*, **4**, Warsaw (1953), 1—45. (Русский перевод: А. Гжегорчик, Некоторые классы рекурсивных функций, Сб. пер. «Проблемы математической логики», М. (1970), 9—49.)
11. Hartmanis J., Tape reversal bounded Turing machine computations, *J. Comp. System Sci.*, 2 (1968), 117—135.
12. Hennie F. C., One-tape, off-line Turing machine computations, *Inf. Contr.*, **8** (1965), 553—578. (Русский перевод: Ф. К. Хенни, Вычисления на одноленточной машине Тьюринга с записью на ленте, Сб. пер. «Проблемы математической логики», М. (1970), 223—248.)
13. Hennie F. C., Crossing sequences and off-line Turing machine computations, IEEE Conf. Rec. on Switching Circuit Theory and Logical Design, 168—172.
14. Hopcroft J. E., Ullman J. D., Relations between time and tape complexities, *J. ACM*, **15** (1968), 414—427.
15. Hopcroft J. E., Ullman J. D., Some results on tape bounded Turing machines, *J. ACM*, **16** (1969), 168—177.
16. Irland M. J., Fischer P. C., A bibliography on computational complexity, Res. Rep. CSRR 2028, U. of Waterloo, Ontario, Canada, Oct., 1970.
17. Karp R. M., Some bounds on the storage requirements of sequential machines and Turing machines, *J. ACM*, **14** (1967), 478—489.
18. Lewis P. M., II, Stearns R. E., Hartmanis J., Memory bounds for recognition of context-free and context-sensitive languages, 1965, IEEE Conf. Rec. on Switching Circuit Theory and Logical Design, 191—202. (Русский перевод: П. М. Льюис II, Р. Е. Стирнз, Дж. Хартманис, Границы памяти для разрешения контекстно-свободных и контекстных языков, Сб. пер. «Проблемы математической логики», М. (1970), 320—338.)
19. McCreight E. M., Classes of computable functions defined by bounds on computations, Doctoral Th., Computer Sci. Dep., Carnegie — Mellon U., Pittsburh, Pa., 1969.
20. Meyer A. R., Ritchie D. M., The complexity of loop programs, Proc. ACM 22nd Nat. Conf., 1967, 465—469.
21. Rabin M. O., Real time computation, *Israel J. Math.*, **1** (1964), 203—211. (Русский перевод: М. Рабин, Вычисления в реальное время, Сб. пер. «Проблемы математической логики», М. (1970), 156—167.)
22. Ritchie R. W., Classes of predictably computable functions, *Trans. Amer. Math. Soc.*, **108** (1963), 139—173. (Русский перевод: Р. В. Ричи, Классы предсказуемо вычислимых функций, Сб. пер. «Проблемы математической логики», М. (1970), 50—93.)
23. Savitch W. J., Relationships between nondeterministic and deterministic tape complexities, *J. Comp. System* Sci., **4** (1970), 177—192.
24. Трахтенброт Б. А., Тьюринговы вычисления с логарифмическим замедлением, Алгебра и логика, Семинар, 3 (1964), № 4, 33—48.
25. Yamada H., Real-time computation and recursive-functions not real-time computable, *IRE Trans. Elec. Comp.* **EC-11** (1962), 753—760. (Русский перевод: Х. Ямада, Вычисления в реальное время и рекурсивные функции, не вычислимые в реальное время, Сб. пер. «Проблемы математической логики», М. (1970), 139—155.)

Алгоритм для минимизации конечного автомата[1)]

Дж. Хопкрофт

1. ВВЕДЕНИЕ

В большинстве книг по конечным автоматам приводятся алгоритмы для минимизации числа состояний в конечном автомате [1, 3]. Однако анализ применения этих алгоритмов в наихудшем случае показывает, что они требуют n^2 шагов, где n есть число состояний. Для конечных автоматов с большим числом состояний эти алгоритмы чрезвычайно неэффективны. В этой статье описывается алгоритм для минимизации числа состояний, время работы которого в наихудшем случае асимптотически растет как $n \log n$. Константа пропорциональности при этом линейно зависит от числа входных символов. Такой же алгоритм может быть, очевидно, использован для определения эквивалентности двух конечных автоматов[2)].

Сущность алгоритмов, опубликованных до сих пор, состояла в начальном разбиении состояний в соответствии с их выходами. Блоки этих разбиений затем повторно делились при рассмотрении следующего состояния при данном входном символе для каждого состояния блока. Состояния, которые под действием данного входного символа переходили в различные блоки, сами помещались в разные блоки. Когда дальнейшее подразбиение блоков оказывалось невозможным, все состояния одного и того же блока оказывались эквивалентными, что можно доказать. Рассмотрим пример (табл. 1). Начальным разбиением является (1, 2, 3, 4, 5) (6). Так как на входе 0 следующие состояния для состояний 1, 2, 3, 4 лежат в первом блоке разбиения, а состояние, в которое переходит состояние 5, лежит во втором блоке, то первая итерация приводит к разбиению (1, 2, 3, 4) (5) и (6). Последующие итерации дают (1, 2, 3) (4) (5) (6); (1,2) (3) (4) (5) (6) и (1) (2) (3) (4) (5) (6). Таким образом, на этом примере можно видеть, что может иногда потребоваться n повторных разбиений (итераций). Поскольку одна итерация требует рассмотрения каждого состояния, общее число ша-

---

[1)] Hopcroft J., An n log n Algorithm for Minimizing States in a Finite Automaton, Theory of Machines and Computations, Proc. Int. Symp., Haifa, 1971, pp. 189—196.

[2)] В работе [4*] предложенный здесь алгоритм рассматривается для входного алфавита с m входными символами и доказывается, что время его работы пропорционально $m\,n \log n$. — *Прим. ред.*

Таблица 1

Состояние	Вход 0	Вход 1	Выход
1	2	1	0
2	3	2	0
3	4	3	0
4	5	4	0
5	6	5	0
6	6	6	1

гов, которое может потребоваться для выполнения алгоритма, если его осуществлять непосредственно на вычислительной машине, есть n^2.

Предлагаемый в этой статье алгоритм также может потребовать n итераций, но результат суммирования времени работы за одну итерацию по всем n интерациям равен по порядку лишь $n \log n$. Мы проиллюстрируем этот алгоритм на примере, прежде чем дадим его детальное описание. Широкое использование обработки списков позволяет уменьшить время вычисления. Сначала табл. 1 переходов — выходов автомата преобразуется в табл. 2.

Таблица 2

Состояние	Вход 0	Вход 1	Выход
1	—	1	0
2	1	2	0
3	2	3	0
4	3	4	0
5	4	5	0
6	5,6	6	1

предыдущее состояние

Состояния распределены в соответствии с их выходами (1, 2, 3, 4, 5) (6). Выбираются блок и входной символ, с помощью которого продолжается разбиение. Предположим, что выбраны блок (6) и входной символ 0. Состояния каждого блока разбиваются дальше в зависимости от того, переходят ли они под действием нуля в блок (6) или нет. Таким образом, следующее разбиение есть (1, 2, 3, 4) (5) (6). Заметим, что если бы мы разбивали блок (1, 2, 3, 4, 5) на входе 0, то получили бы тот же результат. Более общо, если мы однажды произвели разбиение на блоке и входном символе, то нет необходимости снова производить раз-

биение на том же блоке при том же входном символе, пока этот блок не разделен дальше, а тогда необходимо совершить разбиение на одном из подблоков. Поскольку время, необходимое для разбиения блока, пропорционально числу переходов в этот блок и ввиду того, что мы всегда можем выбрать половину с меньшим числом переходов, общее число шагов алгоритма ограничено величиной $n \log n$.

2. ФОРМАЛЬНОЕ ОПИСАНИЕ АЛГОРИТМА

Пусть $A = (S, I, \delta, F)$ — конечный автомат, где S есть конечное множество состояний, I — конечное множество входных символов, δ — отображение из $S \times I$ в S, а $F \subseteq S$ — множество заключительных состояний. Начальное состояние не выделяется, так как это в дальнейшем не играет роли. Отображение δ расширяется на $S \times I^*$ обычным образом, где I^* обозначает множество всех цепочек конечной длины в алфавите I. Состояния s и t называются эквивалентными, если для каждого x в I^* отображение $\delta(s, x)$ принадлежит F тогда и только тогда, когда $\delta(t, x)$ принадлежит F. Ниже описывается алгоритм для нахождения классов эквивалентности в S.

Шаг 1. Для каждого s в S и каждого a в I строится

$$\delta^{-1}(s, a) = \{t \mid \delta(t, a) = s\}.$$

Шаг 2. Строится $B(1) = F$, $B(2) = S - F$ и (для каждого $a \in I$ и $1 \leqslant i \leqslant 2$) строятся множества

$$\hat{B}(B(i), a) = \{s \mid s \in B(i) \text{ и } \delta^{-1}(s, a) \neq \varnothing\}.$$

Шаг 3. Полагается $k = 3$.

Шаг 4. Для каждого a в I находится множество индексов или список

$$L(a) = \begin{cases} \{1\}, & \text{если } |\hat{B}(B(1), a)| \leqslant |\hat{B}(B(2), a)|, \\ \{2\} & \text{в противном случае.} \end{cases}$$

Шаг 5. Выбирается a в I и i в $L(a)$. Алгоритм прекращает работу, если $L(a) = \varnothing$ для каждого a в I.

Шаг 6. От списка $L(a)$ отделяется i.

Шаг 7. Для каждого $j < k$, такого, что существует t в $B(j)$ с $\delta(t, a)$ в $\hat{B}(B(i), a)$, осуществляются шаги 7a, 7b, 7c и 7d.

Шаг 7a. Делается разбиение $B(j)$ на множества (или блоки)

$$B'(j) = \{t \mid \delta(t, a) \in \hat{B}(B(i), a)\} \text{ и}$$

$$B''(j) = B(j) - B'(j).$$

Шаг 7b. $B(j)$ заменяется на $B'(j)$ и строится $B(k)=B''(j)$. Для каждого a в I находятся соответствующие

$$\hat{B}(B(j), a) \quad \text{и} \quad \hat{B}(B(k), a).$$

Шаг 7c. Для каждого a в I список $L(a)$ изменяется следующим образом:

$$L(a)=\begin{cases} L(a)\cup\{j\}, & \text{если } j\notin L(a) \text{ и} \\ 0<|\hat{B}(B(j), a)|\leqslant|\hat{B}(B(k), a)|; \\ L(a)\cup\{k\} & \text{в противном случае.} \end{cases}$$

Шаг 7d. Полагается $k=k+1$,
Шаг 8. Возвращаемся к шагу 5.

3. ПРАВИЛЬНОСТЬ АЛГОРИТМА

Утверждается, что по окончании работы алгоритма два состояния оказываются эквивалентными тогда и только тогда, когда они попадают в один и тот же блок. Пусть a принадлежит I.

Алгоритм должен закончить работу, так как моменты времени, в которые добавляется индекс к $L(a)$, имеют место лишь на шагах 4 и 7c. Шаг 4 выполняется только один раз. На шаге 7c индекс добавляется только после раздробления блока разбиения. Число таких раздроблений не превосходит n. Каждый раз при выполнении шага 6 для некоторого a из $L(a)$ удаляется индекс. Таким образом, алгоритм должен завершить работу.

Легко показать индукцией по числу шагов, на которых выполняется 7a, что если s принадлежит $B(i)$, а t принадлежит $B(j)$, $i\neq j$, то s не эквивалентно t. Очевидно, что это верно для первого применения 7a, так как при этом налицо только два блока: $B(1)$, содержащий только заключительные, т. е. отмеченные состояния, и $B(2)$, содержащий незаключительные состояния. Блоки подразбиваются на шаге 7a, только когда последующие состояния при данном входном символе, как ранее установлено, не эквивалентны. Чтобы показать, что два неэквивалентных состояния не могут находиться в одном и том же блоке, когда алгоритм заканчивает работу, предположим, что состояния s и t принадлежат $B(i)$ и что s и t неэквивалентны.

Без потери общности можно считать, что $\delta(s, a)\in B(j)$, а $\delta(t, a)\in B(k)$, где $j\neq k$. Если $\delta(s, a)$ и $\delta(t, a)$ находятся в одном и том же блоке, то существует кратчайшее слово x, такое, что $\delta(s, x)$ и $\delta(t, x)$ находятся в разных блоках. Очевидно, такое x существует, а следовательно, существует и кратчайшее x, так как

для некоторого x одно из состояний $\delta(s, x)$ или $\delta(t, x)$ является заключительным, но не оба одновременно, и каждый блок состоит только из заключительных или только из незаключительных состояний. Пусть a является последним символом x и запишем $x = ya$. Тогда $\delta(s, y)$ и $\delta(t, y)$ находятся в одном блоке, $\delta(s, y)$ и $\delta(t, y)$ не эквивалентны и $\delta(\delta(s, y), a)$ и $\delta(\delta(t, y), a)$ принадлежат разным блокам.

Заменим s на $\delta(s, y)$, а t на $\delta(t, y)$. Рассмотрим момент, в который блок, содержащий $\delta(s, a)$ и $\delta(t, a)$, разбивается так, что $\delta(s, a)$ и $\delta(t, a)$ впервые оказываются в различных блоках. В этот момент один из двух подблоков помещается в $L(a)$. Когда этот подблок удаляется из $L(a)$, блок, содержащий s и t, разбивается, причем s и t помещаются в различные подблоки. Таким образом, s и t не могут одновременно быть в $B(i)$, т. е. получилось противоречие.

4. ПОДСЧЕТ ВРЕМЕНИ РАБОТЫ АЛГОРИТМА

Время работы алгоритма зависит, очевидно, от его реализации. Алгоритм программируется на Алголе. Поскольку программа содержит приблизительно 300 операторов языка Алгол, мы просто проверим, как осуществляются отдельные шаги, и не формально оценим время их осуществления. Список программы можно найти в [2][1]).

Такие множества, как $\delta^{-1}(s, a)$, $L(a)$ и т. д., представлены связанными списками таким образом, что элемент мог быть добавлен или удален в начале списка за конечное число шагов. Фиксируются векторы, показывающие присутствие или отсутствие состояний в этом списке. Это избавляет от необходимости поиска списка просто с целью определения, имеется ли элемент в списке и является ли он существенным для шага 7c. Множества $B(i)$ и $\hat{B}(B(i), a)$ были бы представлены как дважды зацепленные списки, так что элемент мог бы быть добавлен или удален в любом месте списка за фиксированное число шагов, коль скоро задано его положение. Структура такова, что данное состояние s, его положение в $B(i)$ и $\hat{B}(B(i), a)$ могут быть определены за фиксированное число шагов.

Шаги 1, 2, 3 и 4 выполняются только по одному разу и требуют времени, пропорционального произведению числа состояний на число входных символов. Шаги от 5 до 8 образуют простой цикл. Время, необходимое для прохождения цикла для данного $a \in I$ и i в $L(a)$, пропорционально числу переходов состояний на входах, завершающихся состояниями в $B(i)$. Для того чтобы показать это,

---

[1]) См. также [4*], где подробно излагается алгоритм Хопкрофта и приводится программа на языке PL/I. — *Прим. перев.*

заметим, что шаги 5, 6 и 8 конечны. На шаге 7 не нужно проверять $B(j)$ для каждого $j < k$, чтобы убедиться, существует ли t в $B(j)$ с $\delta(t, a)$ в $\hat{B}(B(i), a)$. Вместо этого мы для каждого состояния в $\hat{B}(B(i), a)$ рассмотрим обращение (inverse) таблицы состояний и найдем все t, такие, что $\delta(t, a) \in \hat{B}(B(i), a)$. Каждый раз, когда находится новое t, обнаруживается блок, содержащий t, а t помещается в список состояний, которые должны быть отделены от блока. Блок помещается в списке блоков, которые подразбиваются, а до этого он еще не входил в этот список. Наконец, мы обращаемся к списку подразбиваемых блоков и фактически к их разбиению. Число блоков, которые следует просмотреть, не превосходит числа переходов на входном символе a, вынуждающем переходы в состояния $B(i)$. Обозначим через k коэффициент пропорциональности.

Рассмотрим время, затрачиваемое в цикле согласно шагам от 5 до 8 для данного входного символа a. Предположим, что на шаге 5 блоки разбиения суть $B(1), B(2), \ldots, B(m)$ и что $L(a) = \{i_1, i_2, \ldots, i_r\}$. Пусть $\{i_{r+1}, i_{r+2}, \ldots, i_m\} = \{1, 2, \ldots, m\} - L(a)$, и пусть $a_i = |\delta^{-1}(B(i), a)|$. Утверждается, что общее время, затрачиваемое на прохождение цикла, для которого входной символ a отобран на шаге 5, до окончания работы программы ограничено величиной

$$T = k\left(\sum_{j=1}^{r} a_{i_j} \log a_{i_j} + \sum_{j=r+1}^{m} a_{i_j} \log (a_{i_j}/2)\right).$$

Очевидно, что эта граница верна, если алгоритм заканчивает работу. Если цикл проходится при входном символе, отличном от a, то затраченное время не включается в T. Однако, поскольку блоки разбиваются и множество $L(a)$ изменено, нам надо показать, что новое значение T, назовем его $\hat{T}$, меньше или равно значению T. Если блок, индекс которого по-прежнему принадлежит $L(a)$, разбивается, то член вида $b \log b$ заменяется выражением $c \log c + (b - c) \log (b - c)$, что уменьшает значение T. Если разбивается блок, индекс которого не входит в $L(a)$, то член вида $b \log (b/2)$ заменяется выражением $c \log c + (b - c) \log ((b - c)/2)$, где $c \leqslant b - c$. Так как $c \leqslant b/2$ и $(b - c)/2 \leqslant b/2$, то $c \log c + (b - c) \log (b - c)/2) \leqslant c \log b/2 + (b - c) \log (b/2) \leqslant b \log (b/2)$.

Таким образом, в любом случае $\hat{T}$ меньше чем T. Предположим, наконец, что $r \neq 0$, $a \in I$ и некоторое $l \in L(a)$ отобрано на шаге 5. Как было показано раньше, время прохождения соответствующего цикла ограничено величиной ka_l. Следовательно, по индукции общее время ограничено величиной

$$k\left[a_l + a_l \log (a_l/2) + \sum_{\substack{j=1 \\ j \neq l}}^{r} a_{i_j} \log a_{i_j} + \sum_{j=r+1}^{m} a_{i_j} \log (a_{i_j}/2)\right].$$

Нам надо показать, что это выражение меньше или равно T. То есть требуется показать, что

$$a_l + a_l \log (a_l/2) \leqslant a_l \log a_l.$$

Очевидно, $a_l + a_l \log(a_l/2) = a_l(\log(a_l/2) + \log 2) = a_l \log a_l$. Этим завершено доказательство сделанного утверждения.

При первом осуществлении шага 5 формула для T ограничена величиной $kn \log n$. Многократное повторение по числу входных символов и прибавление времени, необходимого для выполнения шагов 1—4, дает общую границу времени, пропорциональную $n \log n$.

5. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ И ВЫВОДЫ

Для получения информации о необходимом времени алгоритм был применен к двум классам конечных автоматов. Автоматы первого класса задавались посредством $A(n) = (\{1, 2, \ldots, n\}, \{0, 1\}, \delta, \{1\})$, где $\delta(1, 0) = \delta(1, 1) = 1$, а $\delta(i, 0) = i - 1$ и $\delta(i, 1) = i$ для $2 \leqslant i \leqslant n$.

Автоматы второго класса определяются (для четного n) посредством $B(n) = (\{1, 2, \ldots, n\}, \{0, 1\}, \delta, \{i \mid 1 \leqslant i \leqslant n/2\})$, где $\delta(i, 0) = \delta(i, 1) = n/2 + 2i - 1$, $\delta(n/4 + i, 0) = \delta(n/4 + i, 1) = 2i - 1$ для $1 \leqslant i \leqslant n/4$, а $\delta(n/2 + i, 0) = \delta(n/2 + i, 1) = 2i - 1$ для $n/2 < i \leqslant n$.

Время, использованное на IBM 360/67 для автоматов этих двух классов, указано в табл. 3 (время в сек).

Таблица 3

n	$A(n)$	$B(n)$
100	$^{37}/_{60}$	$^{2}/_{3}$
1000	$5\ ^{50}/_{60}$	$6\ ^{3}/_{4}$
2000	$11\ ^{38}/_{60}$	$13\ ^{2}/_{3}$

Заметим, что $A(n)$ является примером, который требует n^2 шагов в ранее использовавшихся алгоритмах.

Наш алгоритм особенно хорошо подходит для $A(n)$, и детальный анализ показывает, что время линейно растет с ростом числа состояний, как видно из экспериментов. Наихудший случай для нашего алгоритма представляют автоматы класса $B(n)$, в которых блоки всегда разбиваются на равные части. Время вычисления для $B(n)$ растет, как $n \log n$, как для нашего алгоритма, так и для ранее известных алгоритмов. По-видимому, эти результаты пока-

зывают практическую полезность предложенного алгоритма для минимизации числа состояний конечных автоматов (или проверки эквивалентности конечных автоматов) вплоть до нескольких тысяч состояний.

СПИСОК ЛИТЕРАТУРЫ

1. Harrison M. A., Introduction to Switching and Automata Theory, McGraw-Hill, New York, 1965.
2. Hopcroft J. E., An n log n Algorithm for Minimizing States in a Finite Automaton, Technical Report CS-190, Stanford University, Stanford, California, 1970.
3. McCluskey E. J., Introduction to the Theory of Switching Circuits, McGraw-Hill, New York, 1965.
4*. Gries D., Describing an Algorithm by Hopcroft, *Acta Inform.*, **2**, f. 2 (1973), 97—109.

Эксперименты по автоматическому построению тезауруса для информационного поиска[1)]

Джерард Сэлтон

1. СОСТАВЛЕНИЕ СЛОВАРЕЙ ВРУЧНУЮ

Большинство систем как информационного поиска, так и систем обработки текстовой информации содержат в качестве основного компонента некоторую систему анализа, которая служит для выявления «содержания», или «значения» заданной единицы информации. В обычных системах такого рода, применяемых в библиотеках, этот анализ может осуществлять человек. При этом он использует заранее разработанные классификационные таблицы для определения того, какой идентификатор содержания больше подходит для данной единицы информации. Известны также системы так называемого автоматического индексирования, в которых идентификаторы содержания присваиваются автоматически, исходя из текстов документа и запроса.

Поскольку естественный язык по своей природе содержит различного рода нерегулярные явления, которые наблюдаются как в синтаксисе, так и в семантике, система смыслового анализа должна приводить входные тексты к некоторому нормализованному виду, преобразуя различные, возможно неоднозначные, структуры на входе в фиксированные, стандартные идентификаторы содержания. Такого рода процедуры нормализации языка часто используют словари и списки слов, содержащие допустимые идентификаторы содержания, причем для каждого идентификатора приводится соответствующее определение с тем, чтобы регулировать и контролировать его использование. Например, в автоматической информационно-поисковой системе SMART используются следующие основные типы словарей [1]:

[1)] Salton G., Experiments in automatic thesaurus construction for information retrieval, IFIP Congress 71, Foundations of information processing, p. 45—49.

а) *словарь запрещенных слов* («отрицательный» словарь), содержащий общеупотребительные слова, которые запрещено использовать при смысловом анализе;

б) *тезаурус, или словарь синонимов*, определяющий для каждой словарной единицы одну или более синонимичных категорий, или классов понятий;

в) *словарь словосочетаний*, задающий наиболее употребительные комбинации слов или понятий;

г) *иерархическая классификация* терминов или понятий, подобная по своей структуре обычной библиотечной классификации.

Хотя для правильного приписывания идентификаторов содержания, или понятий, совершенно необходимы хорошие словари, построение такого словаря всегда представляет собой весьма трудную задачу, особенно если этот словарь используется в области, лексика которой подвержена изменениям, или если рассматриваемая область относительно широка и неоднородна [2].

Ниже приводится список операций и средств, применяемых в системе SMART для построения словарей запрещенных слов и тезаурусов, причем бóльшая часть работы делается вручную [3]:

а) используется некоторый *список общеупотребительных слов*, которые должны быть исключены из словаря, — это так называемые служебные слова;

б) для множества документов, принадлежащих рассматриваемой области, строится *конкорданс*, или *список ключевых слов вместе с контекстом каждого слова* и частотой встречаемости данного слова в данных документах (указатель типа KWIC);

в) к списку общеупотребительных слов добавляются новые слова, которые берутся из конкорданса; как правило, слова, образующие *расширенный список общеупотребительных слов*, представляют собой либо слова с очень высокой частотой употребления (эти слова ничего не дают для выделения данного документа из документов рассматриваемой области), либо с очень низкой частотой (такие слова дают мало случаев совпадения при сравнении запросов и документов);

г) используется некоторый *стандартный список суффиксов*, содержащий основные суффиксы, употребляемые в английском языке;

д) затем программа автоматического отсечения суффиксов определяет основу каждого оставшегося (необщеупотребительного) слова, после чего человек может просмотреть получившийся в результате *словарь основ* с тем, чтобы обнаружить ошибки алгоритма выделения основы;

е) после этого отбираются основы значащих слов, имеющие наибольшую частоту; эти основы становятся центрами классов понятий рассматриваемого тезауруса;

ж) словарь основ просматривается в алфавитном порядке, при этом слова со средней частотой или добавляются к уже построенным классам, или образуют центры новых классов понятий;

з) оставшиеся основы слов добавляются к уже существующим классам слов, — в основном это слова с низкой частотой;

и) полученный в результате всех этих операций тезаурус проверяется человеком на внутреннюю непротиворечивость; после этого тезаурус печатается.

Как показал эксперимент, тезаурусы, полученные с помощью описанной выше процедуры, дают наиболее приемлемые результаты, если неоднозначные термины входят только в те классы понятий, которые представляют вероятный интерес для рассматриваемой предметной области. Например, термин bat (летучая мышь или бита) не нуждается в указании о принадлежности к классу животных, если рассматриваемые документы отражают спортивную тематику (игры с мячом). Более того, размеры получившихся классов должны быть сравнимы между собой в том смысле, что суммарная частота слов в данном классе понятий должна быть приблизительно одинаковой для всех классов, поэтому высокочастотные слова должны образовывать отдельные классы, а низкочастотные слова должны группироваться таким образом, чтобы происходило выравнивание суммарных частот [3, 4].

Было проведено несколько экспериментов в системе SMART для того, чтобы сравнить эффективность двух различных методов поиска информации. В первом случае использовались составленные вручную тезаурусы, учитывающие синонимию, во втором происходило простое сравнение основ слов, выбранных из запроса, с основами слов, выбранными из документа. При усреднении результатов поиска по многим запросам можно сделать следующий общий вывод: использование тезауруса, которое позволяет приписывать словам идентификаторы понятий, представляющие классы понятий, дает увеличение точности приблизительно на 10 процентов для данного значения полноты по сравнению с методом поосновного сравнения [1]) [5].

Чтобы определить, какие именно свойства тезауруса особенно желательны с точки зрения практического применения, целесообразно кратко рассмотреть основные процедуры, из которых складывается процесс построения тезауруса [6]:

[1]) *Полнота* есть отношение числа выданных релевантных документов к числу всех релевантных документов, имеющихся в массиве, а *точность* есть отношение числа выданных релевантных документов к числу всех выданных документов. Естественным является желание получить большое количество релевантных документов и малое количество нерелевантных документов, что соответствует высокой полноте и высокой точности. Наилучшие значения этих характеристик представляются кривой, которая приближается к правому верхнему углу типичного графика «полнота — точность», поскольку в точке с координатами (1, 1) как полнота, так и точность принимают свое наибольшее значение.

а) Получение основы слова;

i) тип используемой при этом процедуры отделения суффикса: эта процедура может быть полностью автоматической или в ней может использоваться заранее составленный словарь суффиксов;

ii) степень сложности средств, применяемых для выделения суффикса: можно учитывать морфологические свойства одного слова или рассматривать также и окрестность каждого слова;

б) Процесс построения класса понятий;

i) степень автоматизации при построении классов тезауруса;

ii) средний объем классов в тезаурусе;

iii) однородность классов по объему;

iv) однородность по частоте встречаемости отдельных членов класса (в пределах класса тезауруса);

v) степень перекрываемости классов тезауруса (т. е. число слов в пересечении);

vi) семантическая близость классов тезауруса;

в) Распознавание общеупотребительных слов;

i) степень автоматизации этого процесса;

ii) доля общеупотребительных слов во всем словаре;

г) Обработка случаев многозначности в языке;

i) степень автоматизации распознавания случаев неоднозначности;

ii) степень проникновения в глубину многозначных структур.

Во всех алгоритмах анализа языка, употребляющихся в системе SMART, используется программа автоматического отделения суффиксов, опирающаяся на составленный вручную словарь суффиксов. Кроме того, в системе SMART не рассматриваются в в явном виде случаи языковой многозначности, например когда в тексте встречаются омографы [1]). Поэтому при исследовании эффективности использования тезауруса как главные рассматриваются две процедуры: распознавание общеупотребительных слов и построение классов понятий. Эти две задачи рассматриваются ниже.

2. РАСПОЗНАВАНИЕ ОБЩЕУПОТРЕБИТЕЛЬНЫХ СЛОВ

Говоря о задаче распознавания общеупотребительных слов, прежде всего необходимо отличать общеупотребительные *служебные* слова, такие, как предлоги, союзы или артикли, от общеупотребительных значащих слов. Первые легко находятся в тексте, если использовать список служебных слов, причем этот список не меняется, какую бы предметную область мы ни рассматривали.

[1]) Хотя некоторые системы анализа языка используют детально разработанные процедуры распознавания языковой многозначности, оказывается, что даже в самых сложных случаях многозначность автоматически устраняется, если применять данный словарь к определенной, точно очерченной предметной области [7, 8].

Слова второго типа представляют собой термины, встречающиеся в текстах очень часто (или очень редко), которые не следует включать в обычные классы понятий, потому что они не выделяют данный документ среди документов рассматриваемой предметной области. Такой является, например, основа «автомат», очень часто встречающаяся в текстах по вычислительной технике. Необходимо, чтобы слова такого рода были опознаны, так как, будучи приписаны в качестве идентификаторов понятий, они будут давать высокий коэффициент подобия между единицами информации, которые на самом деле имеют мало общего между собой; кроме того, присутствие таких слов увеличивало бы как объем машинной памяти, так и стоимость обработки информации.

Для определения того, насколько важной является задача распознавания общеупотребительных слов, недавно было проведено соответствующее исследование. При этом сравнивалась эффективность применения при информационном поиске стандартной процедуры сравнения основ слов, обычного тезауруса, а также процедуры сравнения основ слов, в которой распознаются общеупотребительные значащие слова, обычно включаемые в тезаурус [9]. Для этой цели применялась процедура, с помощью которой из тезауруса был получен словарь основ. При этом каждый класс тезауруса разбивался на отдельные слова, и основы этих слов брались в качестве элементов нового словаря основ. Главным отличием этого нового *словаря значащих основ* от прежнего словаря основ является то, что в нем отсутствуют основы, соответствующие как служебным словам, так и общеупотребительным значащим словам, которые, как правило, приводятся в тезаурусе. Поэтому сравнение словаря значащих основ с обычным словарем основ даст нам доказательство того, насколько важно отбрасывать общеупотребительные слова при сравнении запроса и документа. Сравнение же словаря значащих основ и тезауруса позволит нам оценить методы построения этого тезауруса, а именно те методы (принципы), которые лежали в основе образования классов понятий и объединения терминов в класс.

На рис. 1, *а* представлена зависимость между полнотой и точностью при работе этих трех словарей, причем значения полноты и точности усреднены для 42 запросов и 200 документов из области аэродинамики. Как видно из рис. 1, *а*, использование тезауруса дает увеличение точности на несколько десятков процентов при заданной полноте по сравнению с соответствующей величиной для обычного словаря основ. Словарь значащих основ неожиданно дает дальнейшее улучшение точности по сравнению с тезаурусом, а это показывает, что основной характеристикой словаря, которая проверялась в эксперименте, является информация об общеупотребительных словах, а не объединение терминов в классы понятий. В рассматриваемом эксперименте словарь значащих основ содер-

жит примерно в два раза больше основ общеупотребительных слов, чем обычный словарь основ.

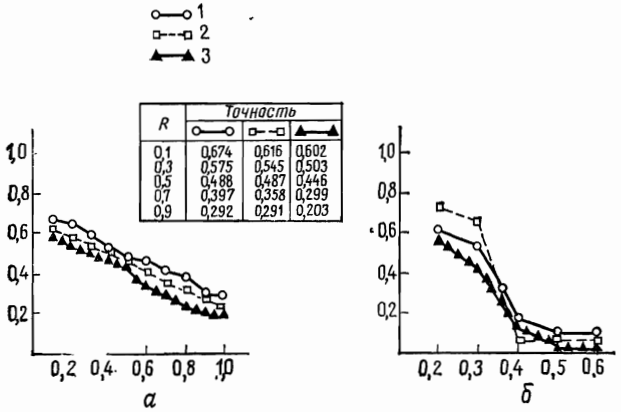


Рис. 1. Сравнение словаря значащих основ с тезаурусом и обычным словарем основ (Крэнфилдский массив документов).
1—словарь значащих основ; *2*—обычный тезаурус; *3* — обычный словарь основ.
а) График зависимости полнота — точность (200 документов, 42 запроса); по оси абсцисс — полнота; по оси ординат — точность; б) ранговая полнота для запросов, записанных при помощи 6 понятий; по оси абсцисс — число релевантных документов на запрос; по оси ординат — ранговая полнота.

Конечно, характеристики полнота — точность, представленные на рис. 1, *а*, не позволяют сделать вывода о том, что использование словарей синонимов, или тезаурусов, построенных путем объединения терминов в классы, ничего не дает для анализа содержания запроса и документа при информационном поиске. Довольно часто в отдельных запросах ставятся особые требования, например требуется очень высокая полнота или точность, — в таких случаях тезаурус может оказать весьма существенную помощь.

В качестве примера рассмотрим рис 1, *б*, на котором изображены значения *ранговой полноты* для десяти запросов (из сорока двух), каждый из которых описан с помощью шести понятий, взятых из тезауруса [1]). Очевидно, для запросов, которым соответствует очень малое количество документов из рассматриваемого множества документов, тезаурус дает значительно более высокую эффективность, чем любой из словарей. По мере того как число документов, релевантных данному запросу, растет, словари основ начинают соперничать с тезаурусом.

Исходя из очевидной необходимости распознавания общеупо-

[1]) Ранговая полнота представляет собой характеристику поиска, выражаемую числом, которое связано обратной зависимостью с рангами документов, полученными в процессе поиска [1].

требительных слов, можно задать следующий вопрос: нельзя ли выделять слова такого рода автоматически вместо того, чтобы вручную пользоваться процедурой, описанной в предыдущем разделе. Этот вопрос изучался с помощью следующей математической модели. Рассмотрим исходное множество терминов, или понятий, которые используются для описания некоторого множества запросов и документов, и пусть это множество терминов меняется путем *выборочного вычеркивания* некоторых определенных терминов из запросов и документов. Тогда мы получим одно из двух в зависимости от того, какие термины вычеркивались:

а) если вычеркиваемые термины используются по существу при анализе содержания, то именно они и дают возможность различать документы и удаление таких терминов приведет к тому, что пространство документов как бы сожмется, документы станут более похожими друг на друга, т. е. увеличится корреляция между парами документов;

б) с другой стороны, если вычеркиваемые термины представляют собой общеупотребительные слова, которые не играют роли при различении документов, то при удалении таких терминов пространство документов как бы расширится, а корреляция между парами документов уменьшится.

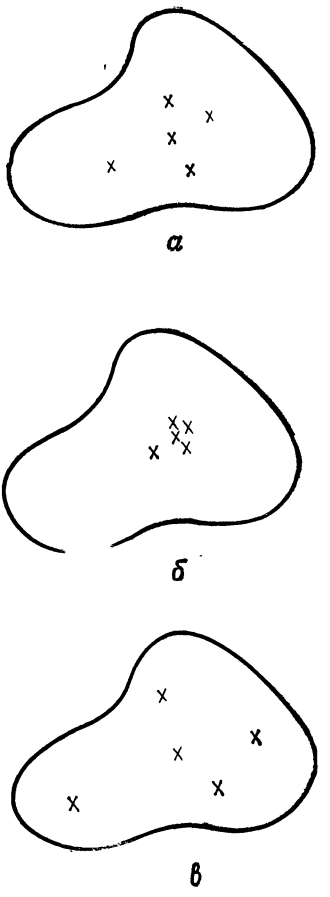


Рис. 2. Изменения, происходящие в пространстве документов в результате вычеркивания определенных терминов.
а) Исходное пространство документов; б) пространство документов после удаления словразличителей; в) пространство документов после удаления слов-неразличителей.

Это положение иллюстрируется упрощенной моделью, изображенной на рис. 2, где каждый документ обозначен крестиком. При этом предполагается, что близость документов (величина, выражающая сходство документов) обратно пропорциональна расстоянию между соответствующими крестиками. Тогда нужно проверить такое предположение: термин, который следует отнести к общеупотребительным словам и поэтому удалить из множества кандидатов в идентификаторы понятий (и из множества допусти-

мых понятий тезауруса), является термином, вызывающим расширение пространства документов путем уменьшения его компактности.

Для проверки этого предположения используется следующая процедура [10]. Рассмотрим множество из N документов. Пусть каждый документ представляется с помощью *вектора терминов*, или понятий $\mathbf{v}_j$, где $\mathbf{v}_{ij}$ выражает вес термина i в документе j. Пусть центроид $\mathbf{c}$ всех точек, соответствующих множеству документов, определяется как «средний документ», т. е.

$$\mathbf{c}_i = \frac{1}{N} \sum_{j=1}^{N} \mathbf{v}_{ij}.$$

Тогда, по существу, этот центроид является центром тяжести пространства документов. Если подобие пары документов k и j задано корреляцией $r(\mathbf{v}_k, \mathbf{v}_j)$, где r меняется от 1 (для абсолютно подобных документов) до 0 (для совершенно непохожих документов), компактность Q пространства документов можно определить как

$$Q = \sum_{j=1}^{N} r(\mathbf{c}, \mathbf{v}_j), \quad 0 \leqslant Q \leqslant N,$$

т. е. как сумма значений функции подобия между каждым документом и центроидом; чем больше значение Q, тем больше компактность пространства документов.

Рассмотрим теперь функцию Q_i, представляющую собой компактность пространства документов, если термин i удален. Если $Q_i > Q$, пространство документов стало еще более компактно, а термин i является различителем (дискриминатором); в противном случае $Q_i < Q$, пространство расширяется, а удаление термина i может дать лучшие результаты поиска. Поскольку вычеркивание слов-различителей увеличивает Q, а вычеркивание слов-неразличителей (общеупотребительных слов) уменьшает Q, должно существовать некоторое оптимальное множество терминов I, такое, что для него Q_I минимально.

Тогда в качестве экспериментальной можно предложить следующую процедуру:

а) рассмотреть по порядку все термины i и вычислить для каждого из них Q_i;

б) расположить все термины в порядке уменьшения Q_i (т. е. первыми будут те термины, которые вызывают наибольшее уменьшение компактности);

в) считать, по определению, множеством общеупотребительных терминов I, подлежащих удалению, такое множество терминов, которое дает наименьшее значение Q.

На рис. 3 представлены результаты, оценивающие работу этого алгоритма на примере 35 реальных запросов и 82 документов по

информатике. Сначала для описания документов употреблялось 1218 различных основ слов. Из рис. 3, *а* видно, что результаты эксперимента полностью соответствуют модели:

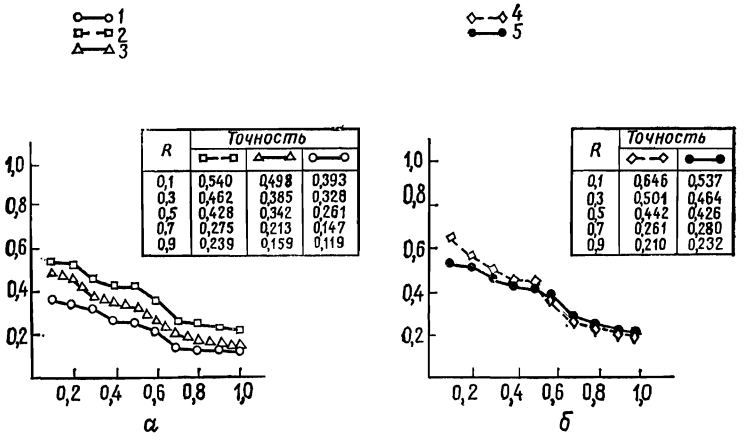


Рис. 3. Автоматическое распознавание общеупотребительных слов.
1 — исходное множество основ; *2* — вычеркнуто 252 термина; *3* — вычеркнуто 1120 терминов; *4* — тезаурус; *5* — вычеркнуто 28 основ слов.
а) Сравнение результатов с разными объемами словарей; по оси абсцисс — полнота, по оси ординат — точность; б) сравнение с тезаурусом.

а) сначала были удалены высокочастотные слова, не являющиеся различителями (252 термина), при этом пространство расширилось, а соответствующее значение точности при заданной полноте увеличилось примерно на 20 процентов;

б) когда дополнительно были удалены еще некоторые термины, компактность пространства начала увеличиваться по мере удаления различителей, а соотношение полнота — точность начало ухудшаться. На рис. 3, *а* кривая, проходящая посередине, представляет эту характеристику после удаления 1120 терминов (Q уменьшено на порядок), при этом эффективность поиска снижается примерно на десять процентов.

На рис. 3, *б* представлен результат сравнения соответствующих характеристик для обычного тезауруса и словаря основ, в котором удалены первые 28 общеупотребительных терминов[1]). Очевидно, что тезаурус дает лучшие результаты там, где полнота мала, причем эти две кривые почти сливаются в большей части области определения.

Таким образом, результаты, представленные на рис. 3, подтверждают результаты, представленные на рис. 1, в том смысле, что

[1]) Термины упорядочены в соответствии с пунктом б) процедуры. — *Прим. перев.*

использование словаря основ дает приблизительно те же значения числовых характеристик поиска, что и использование обычного тезауруса при условии, что выделены основы общеупотребительных слов и что эти основы исключены из множества идентификаторов понятий.

3. АЛГОРИТМЫ АВТОМАТИЧЕСКОГО ПОСТРОЕНИЯ КЛАССОВ ПОНЯТИЙ

Общая проблема классификации, заключающаяся в построении групп, или классов единиц, которые в некотором смысле подобны между собой, нашла отражение во многих исследованиях, ведущихся в течение многих лет в различных областях науки. В задачах информационного поиска документы часто разбивают на некоторые классы, упрощая таким образом процесс поиска информации. С другой стороны, термины или понятия объединяются в классы тезауруса таким образом, чтобы в один класс, характеризуемый некоторым номером, попали все синонимы данного слова и слова, связанные с данным словом по смыслу.

В разд. 1 настоящей работы были определены различные критерии построения классов тезауруса вручную. Однако поскольку построение тезауруса вручную требует от его создателей больших затрат времени, а также большого опыта, в течение многих лет ведутся эксперименты по автоматическому построению классов тезауруса. В основе этих экспериментов лежит изучение свойств рассматриваемого множества документов, т. е. изучение соответствия между терминами и документами. Этот процесс в общих чертах может быть описан следующим образом [11]:

а) сначала строится матрица термин — документ, в которой для каждого документа указываются все термины, ему приписанные; если терминам приписаны веса, они тоже указываются в этой матрице;

б) с помощью матрицы термин — документ строится матрица близости терминов, — для этого вычисляется коэффициент близости между каждой парой векторов терминов, все это делается на основе рассмотрения всех терминов и всех документов;

в) строится матрица связей между парами терминов: два термина считаются связанными (в соответствующее место матрицы связей ставится 1), если коэффициент близости между соответствующими векторами терминов больше некоторого порогового значения;

г) можно рассматривать эту двоичную матрицу связей как абстрактный граф, в котором каждая вершина изображает некоторый термин, а каждая установленная связь изображается в виде ветви между соответствующей парой вершин; для выделения классов терминов, или пучков, исследуется некоторая функция этого графа (в качестве такой функции можно взять связные компо-

ненты графа или максимальные полные подграфы данного графа) [1]).

Ряд исследователей проводили автоматическую классификацию, используя процедуры, подобные описанной выше [12—15]. К сожалению, построение матрицы связей в том случае, когда число терминов не очень мало, является довольно дорогостоящей процедурой. По этой причине на практике используются более дешевые методы автоматической классификации, когда некоторая существующая грубая классификация улучшается за счет выборочного изменения исходных классов [16, 17].

Ниже кратко описываются три эксперимента из области информационного поиска, проведенные для определения эффективности автоматической классификации такого рода. При этом изучались следующие методы классификации: автоматическое уточнение уже построенных классов, два способа полностью автоматической классификации и один метод полуавтоматической классификации.

Первый из этих способов заключается в следующем: берется уже существующая классификация или существующий тезаурус и производится чистка классов путем удаления тех классов, которые пересекаются по большому числу элементов [18]. Один алгоритм такого рода, опробованный в системе SMART, содержал следующие шаги в добавление к шагам а)—г), ранее описанным:

д) для данных классов терминов строится матрица близости классов, причем используется процедура, подобная той, которая применяется для построения матрицы близости терминов;

е) с помощью матрицы близости классов строится матрица связей классов, состоящая из нулей и единиц, при этом используется некоторое пороговое значение;

ж) каждый максимальный полный подграф задает новый, *объединенный класс понятий*;

з) затем удаляются те объединенные классы, которые входят в некоторые другие, более широкие классы; объединенные классы, оставшиеся после всех удалений, образуют новую классификацию.

Этот алгоритм был применен для уточнения тезауруса, первоначально использовавшегося для описания документов по аэродинамике (82 документа и 35 запросов). Были построены два «объединенных» тезауруса:

1) тезаурус 1 с общим числом классов 156, причем на класс в нем приходится около 3,9 понятия;

2) тезаурус 2 с общим числом классов 289, в нем на класс приходится в среднем 1,4 понятия [19].

---

[1]) *Связной компонентой* графа называется подграф, каждая пара вершин которого соединена некоторым путем (цепью ветвей); *максимальный полный подграф* — это такой подграф, в котором каждая пара вершин непосредственно связана некоторой ветвью, и никакая вершина вне его не обладает этим свойством по отношению к любой вершине из рассматриваемого подграфа.

Общие нормализованные показатели полноты — точности, усредненные для 35 запросов, которые приведены в табл. 1, отражают некоторое улучшение характеристик поиска, произведенного с помощью этих уточненных тезаурусов.

Таблица 1

Числовые характеристики поиска при работе с тезаурусами, полученными в результате объединения классов

Тип тезауруса	Нормализованная полнота	Нормализованная точность
Исходный тезаурус	0,800	0,610
Объединенный тезаурус 1	0,830	0,640
Объединенный тезаурус 2	0,830	0,650

Вторая группа экспериментов была посвящена решению более смелой задачи — полностью автоматической классификации, описанной в начале этого раздела. В одном из таких исследований использовалось много определений из теории графов, таких, как «цепочки терминов», «звезды», «клики» (cliques), «пучки», а также различные частотные и пороговые ограничения [20]. Вообще говоря, оказалось, что использование некоторых из этих автоматических классификаций позволяет получить бóльшую эффективность, чем использование ключевых слов, в которых классификация не проведена, в особенности если при этом используется отношение «сильной» близости (ему соответствует большое пороговое значение), и при этом разрешается классифицировать только те слова, которые имеют небольшую частоту. При этом исследователи даже не пытались сравнить эти автоматически полученные классы с тезаурусами, построенными вручную.

Недавно был закончен другой эксперимент, посвященный полностью автоматической классификации. В этом эксперименте на большом материале (11 500 аннотаций документов из области вычислительной техники) использовались алгоритмы, аналогичные описанным выше [21]. При этом происходило уточнение классов, а также изменение и опробование многих параметров. В конце концов, было достигнуто незначительное улучшение по сравнению с обычным методом, когда происходит сравнение основ слов.

Автор этой работы заявляет: «Что касается результатов, достигнутых при помощи различных (автоматических) ассоциативных стратегий, следует сделать вывод о том, что поиск путем простого сравнения основ ключевых слов обеспечивает очень хороший уровень соответствующих числовых характеристик» [21, стр. 61].

Последний описываемый эксперимент по классификации терминов основан на полуавтоматическом способе получения исходных векторов терминов (понятий), которые используются при построе-

нии матрицы близости терминов, а именно: человеком задается некоторое множество свойств термина путем ответа на вопросы о каждом термине, причем ответы кодируются соответствующим образом [1]). Затем для каждого термина определяется соответствующий характеристический вектор — множество ответов на 10 или 12 вопросов, заданных человеком. Когда для всех терминов уже построены соответствующие характеристические векторы, для получения классификации типа тезауруса можно воспользоваться одним из алгоритмов автоматической классификации [3, 22].

Словарь такого рода, построенный полуавтоматически, был получен для документов из области вычислительной техники. В табл. 2 приводятся его характеристики и характеристики тезауруса, построенного вручную. Очевидно, что в тезаурусе, построенном полуавтоматически, классы значительно менее однородны, чем в тезаурусе, построенном вручную: одни классы очень велики, другие очень малы. Более того, в тезаурусе, построенном полуавтоматически, содержится меньшее число общеупотребительных слов.

На рис. 4 представлены результаты использования этих двух тезаурусов при информационном поиске. Очевидно, что для почти всех значений полноты полуавтоматический тезаурус дает более низкие значения точности, чем тезаурус, построенный вручную. Только для очень больших значений полноты эффективность обоих этих словарей приблизительно одинакова.

Таблица 2

Сравнительные характеристики словаря, построенного полуавтоматически

Характеристики	Тезаурус, построенный вручную (Харрис)	Тезаурус, построенный полуавтоматически (Бенч)
Число классов понятий	863	2953
Число основ слов	2551	5197
Среднее число слов в классе	3	1,8
Число небольших по объему классов (по одному слову в классе)	468	2725
Число очень больших классов (от 32 до 101 слова)	2	12
Число слов, попавших в два или более классов	52	275
Отношение числа общеупотребительных слов к общему числу слов	37,3%	4,4%

[1]) Если некоторый заданный термин относится к области вычислительной техники, то типичными вопросами можно считать следующие: относится ли этот термин к устройству самой машины (1) или к области математического обеспечения (2), или данный вопрос поставлен некорректно для данного термина (3); выбранный ответ кодируется своим номером (n).

4. ВЫВОДЫ

В настоящей работе описывается и оценивается ряд алгоритмов, предназначенных как для автоматического, так и для ручного построения словарей, в частности, рассматриваются методы автоматического распознавания общеупотребительных слов и методы группирования терминов, автоматические и полуавтоматические.

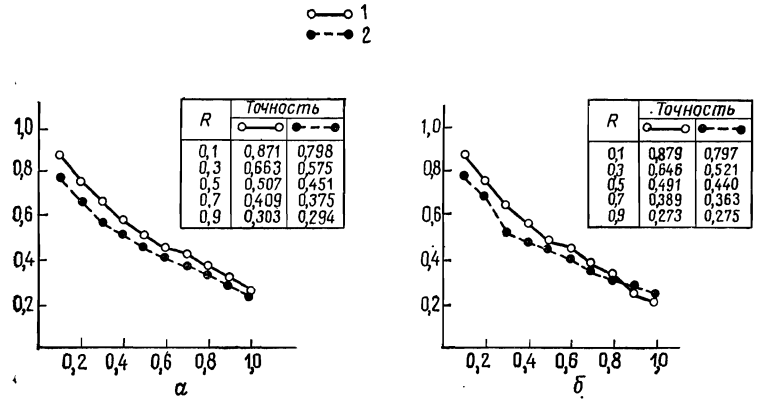


Рис. 4. Сравнение двух тезаурусов: построенного вручную и построенного полуавтоматически (по оси абсцисс — полнота, по оси ординат — точность).
Тезаурус: *1* — построенный вручную; *2* — построенный автоматически.
а) Тексты по вычислительной технике; б) аннотации по вычислительной технике.

Как оказалось, автоматические методы распознавания общеупотребительных слов могут успешно применяться в существующих системах анализа текстов. В самом деле, эффективность применения в информационном поиске пополненного словаря основ оказывается такой же, как и эффективность поиска, проводимого с помощью обычных тезаурусов.

Эффективность применения алгоритмов автоматической группировки слов отчасти находится под сомнением. Вероятно, методы автоматического объединения слов в группы могут использоваться более эффективно, чем дорогостоящие процессы ручного построения тезаурусов. Однако до сих пор автоматическими методами не было построено тезауруса, который бы доказал свое преимущество [23, 24].

Поэтому в настоящее время в практических приложениях наиболее перспективно комбинирование ручных и автоматических методов, в которое входит:

а) автоматическое распознавание общеупотребительных слов;
б) классификация терминов, осуществляемая вручную;
в) автоматическое уточнение классов, полученных вручную.

СПИСОК ЛИТЕРАТУРЫ

1. Salton G., Automatic Information Organisation and Retrieval, McGraw-Hill Book Co., New York, 1968.
2. Wall E., Vocabulary Building and Control Techniques, Amer. Doc., 20, 2 (1969).
3. Salton G., Lesk M. E., Information Analysis and Dictionary Construction, Report No. ISR-11 to National Science Foundation, Sect. IV, Dept. of Computer Science, Cornell University, June 1966.
4. Lesk M. E., Performance of Automatic Information Systems, *Information Storage and Retrieval*, **4** (1968), 201—218.
5. Salton G., Computer Evaluation of Indexing and Text Processing, *JACM*, **15**.
6. Moser A., Construction of Dictionaries for Text Analysis and Retrieval, неопубл. работа, Cornell University, 1970.
7. Coyaud M., Sio-Decauville N., L'Analyse Automatique des Documents Mouton and Co., Paris, 1967.
8. Д. Г. Лахути, Б. Л. Румшиский, Е. Б. Федоров, В. С. Чернявский, А. Л. Шумилина, Полуавтоматический перевод с естественного языка на дескрипторный язык систем типа «Пусто-Непусто» (автоматическое индексирование), Сб. «Информационно-поисковые системы и автоматизированная обработка НТИ», Труды III Всесоюзной конференции, т. 1, ИПС, М., 1967.
9. Bergmark D., The Effect of Common Words on Retrieval Performance, Report No. ISR-18 to National Science Foundation and to National Library of Medicine, Sect. V, Dept. of Computer Science, Cornell University, October 1970.
10. Bonwit K., Aste-Tonsman J., Negative Dictionaries, Report No. ISR-18 to National Science Foundation and to National Library of Medicine, Sect. VI, Dept. of Computer Science, Cornell University, October 1970.
11. Augustson J. G., Minker J., An Analysis of Some Graph Theoretical Cluster Techniques, *JACM*, **17** (1970), 571—588.
12. Borko H., The Construction of an Empirically Based Mathematically Derived Classification System, Report No. SP-588, Syst. Devel. Corp., October 1961.
13. Dennis S. F., The Design and Testing of a Fully Automatic Index Searching System for Documents Consisting of Expository Test, in G. Schester, (ed.), «Information Retrieval — A Critical View», Thompson Book Co., 1967.
14. Sparck Jones K., Jackson D., Current Approaches to Classification and Clump Finding, *Computer Journal*, **10**, 1 (1967), 29—37.
15. Giuliano V. E., Jones P. E., Linear Associative Information Retrieval, in P. Howerton (ed.), «Vistas in Information Handling», (Spartan Books, Inc., 1963).
16. Doyle L. B., Breaking the Cost Barrier in Automatic Classification, Report No. SP-2516, System Development Corp., Santa Monica, July 1966.
17. Dattola R. T., A Fast Algorithm for Automatic Classification Report No. ISR-14 to National Science Foundation, Sect. V, Comp. Sci. Dep., Cornell Univ., October 1968.
18. Gotlieb C. C., Kumar S., Semantic Clustering of Index Terms, *JACM*, **15** (1968), 493—513.
19. Dattola R. T., Murray D. M., An Experiment in Aautomatic Thesaurus Construction, Report No. ISR-13 to National Science Foundation, Sect. VIII, Dept. of Computer Science, Cornell University, December 1967.
20. Sparck Jones K., Jackson D. M., The Use of Automatically-Obtained Keyword Classifications for Information Retrieval, *Information Storage and Retrieval*, **5**, 4 (1970), 175—202.
21. Vaswani P. K. T., Cameron J. B., The NPL Experiments in Statistical Word Associations and their Use in Document Indexing and Retrieval, Report Com. Sci. 42, National Physical Laboratory, Teddington, England, April 1970.
22. Salton G., Information Dissemination and Automatic Information Systems, *Proc. IEEE*, **54**, 12 (1966), 1663—1678.
23. Cleverdon C. W., Keen E. M., Factors Determining the Performance of Indexing Systems, Aslib-Cranfield Research Project Report, Vol. 1 and 2, Cranfield, England, 1966.
24. Salton G., Automatic Text Analysis, *Science*, **168**, April 1970, 335—343.

СОДЕРЖАНИЕ

Математические вопросы

П. Перкинс. Базисы для эквациональных теорий полугрупп. *Перевод В. Л. Мурского* . . . . . .

Дж. Л. Месси, Д. Дж. Кастелло, Й. Юстесен. Веса многочленов и кодовые конструкции. *Перевод Б. С. Цыбакова* . . . . . .

Йёрн Юстесен. Новые конструкции сверточных кодов и класс асимптотически хороших кодов, меняющихся со временем. *Перевод Б. С. Цыбакова*

Т. М. Ковер. Широковещательные каналы. *Перевод Б. С. Цыбакова* . .

Яхико Камбаяси и Сузо Ядзима. Верхняя граница числа К для последовательностных машин без потери информации порядка К. *Перевод А. А. Курмита* . . . . . .

Д. Б. Бенсон. Синтаксис и семантика с точки зрения теории категорий *Перевод Е. С. Бургиной* . . . . . .

Дж. Хартманис, Ф. Д. Льюис. Использование списков в изучении проблем теории автоматов. *Перевод Т. И. Поповой* . . . . . .

Дж. Хартманис и Дж. Э. Хопкрофт. Обзор теории сложности вычислений. *Перевод В. Н. Агафонова* . . . . . .

Дж. Хопкрофт. Алгоритм для минимизации конечного автомата. *Перевод А. А. Мучника* . . . . . .

Вопросы информационного поиска

Джерард Сэлтон. Эксперименты по автоматическому построению тезауруса для информационного поиска. *Перевод Н. Г. Арсентьевой* . . . .

5
24
48
62
94
101
117
131
177
185