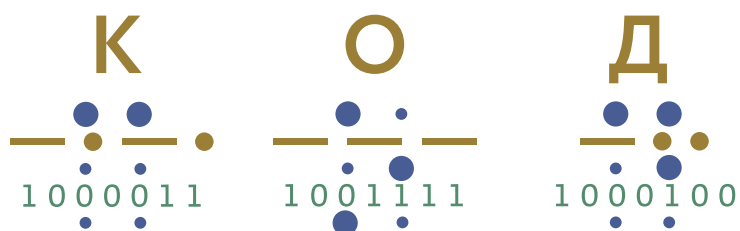


Тайный язык информатики



Чарльз Петцольд

Оглавление

Предисловие к оригинальному изданию	9
Глава 1. Лучшие друзья	12
Глава 2. Коды и комбинации	18
Глава 3. Брайль и двоичные коды	23
Глава 4. Устройство фонарика	30
Глава 5. Заглядывая за угол	41
Глава 6. Телеграфы и реле	50
Глава 7. Наши десять цифр	57
Глава 8. Альтернативы десятке	64
Глава 9. За битом бит	80
Глава 10. Логика и переключатели	98
Глава 11. Логические вентили	117
Глава 12. Двоичный сумматор	147
Глава 13. А как насчет вычитания?	160
Глава 14. Обратная связь и триггеры	174
Глава 15. Байты и шестнадцатеричные числа	202
Глава 16. Сборка памяти	213
Глава 17. Автоматизация	231
Глава 18. От счетов к микросхемам	268
Глава 19. Два классических микропроцессора	294
Глава 20. Набор символов ASCII	326
Глава 21. Шины	345
Глава 22. Операционная система	368
Глава 23. Фиксированная точка, плавающая точка	387
Глава 24. Языки высокого и низкого уровня	403
Глава 25. Графическая революция	421
Благодарности	444
Об авторе	445

Предисловие к оригинальному изданию

Замысел «Кода» я вынашивал лет десять. И тогда, и во время работы над рукописью, и даже когда книга вышла из типографии многие спрашивали: «О чем она?»

Я всегда отвечал уклончиво, бормотал что-нибудь в духе: «Необычная экскурсия по истории цифровых технологий, сформировавших современную эпоху» — в надежде, что этого будет достаточно. Но в какой-то момент мне пришлось признать: «Код» — это книга о том, как устроены компьютеры.

Как я и опасался, отклики были неблагоприятными. На возражение в духе: «А-а, у меня уже есть такая книга» — я немедленно парировал: «Отнюдь, такой — нет». И по-прежнему так считаю. «Код» не похож на прочие книги «о компьютерах». В нем нет больших цветных иллюстраций с дисковымидами, где стрелками показано, как данные поступают в компьютер, нет рисунков, где паровозик в товарных вагончиках везет нули и единички. Метафоры и сравнения чудесны в своей буквальности, но они ни на что не годны, лишь затмевают красоту технологий.

Мне говорили: «А кому интересно, как работают компьютеры?» Верное замечание. Мне, например, нравится вникать в устройство приборов, но я хочу сам решать, когда это делать. Так, описать, как работает мой холодильник, я смогу лишь под пыткой.

Однако окружающие часто задают вопросы, свидетельствующие об их интересе к внутреннему устройству компьютера. Типичный пример: «Чем отличается оперативная память от дисковой?» Естественно, это важный вопрос. Такие понятия составляют основу маркетинга ПК. Даже начинающему пользователю требуется знать, сколько *мегов* одного и *гигов* другого потребуется для конкретного приложения. Кроме того, новичок должен представлять, что такое файл, как он загружается с диска в память, а затем сохраняется там.

На вопрос о дисковой и оперативной памяти принято отвечать: «Память похожа на столешницу, а диск — на ящики стола». В принципе неплохой ответ, но мне он кажется неудовлетворительным. Создается впечатление, будто архитектура компьютера разрабатывалась по образу и подобию бюро. На самом деле

разница между оперативной и дисковой памятью — искусственная и обусловлена отсутствием единого энергонезависимого и при этом быстро работающего носителя. Так называемая архитектура фон Неймана, доминирующая в компьютерной индустрии уже более 50 лет, возникла в результате этого технического изъяна. Когда меня спрашивали, как запускать программы для Macintosh под Windows, я впадал в ступор, осознавая, что для ответа придется затронуть массу технических тонкостей, которые собеседник явно сразу не поймет.

Хочу, чтобы с помощью «Кода» вы научились разбираться во всех этих вещах настолько, чтобы смогли потягаться с электротехниками и программистами. Надеюсь, вы оцените, каким достижением является компьютер среди технологий XX века, и прочувствуете его красоту саму по себе, без метафор и сравнений.

По сути, компьютеры иерархичны: на самом нижнем уровне располагаются транзисторы, а венчает все информация, которая выводится на монитор. В книге мы будем придерживаться этой иерархии. В принципе, книга и структурирована от уровня к уровню. И этот путь не столь сложен, как может показаться. Да, в современном компьютере происходит масса всякой всячины, но это самые обычные и простые операции.

Хотя в настоящее время компьютеры сложнее, чем четверть или полвека назад, они не изменились фундаментально. Вот почему изучать историю техники так здорово: чем сильнее углубляешься в прошлое, тем проще становятся технологии. Именно поэтому легко добраться до точки, где понятно решительно все.

В книге «Код» я заглянул настолько далеко в прошлое, насколько смог. Сам поразился, что удалось добраться до XIX века и на примере первых телеграфных машин объяснить устройство компьютера. Теоретически все, о чем говорится в первых 17 главах, легко собирается из простейших электрических компонентов, которые в ходу уже более века.

Думаю, благодаря всей этой винтажной технике при чтении вы испытаете некоторую ностальгию. Книгу «Код» невозможно было бы озаглавить «Еще быстрее, еще технологичнее» или «Сверхскоростной бизнес на цифровых нейронах»: определение бита дается лишь на 79-й странице, байта — на 199-й. Транзисторы впервые упоминаются на 156-й странице, и то вскользь.

Итак, пусть «Код» и весьма основательно объясняет устройство компьютера (найдется немного других книг, где описано, например, как именно работает процессор), стиль книги вполне развлекательный. Несмотря на глубину темы, я старался устроить читателю максимально комфортную прогулку. Без всяких вагончиков с нулями и единицами.

*Чарльз Петцольд
16 августа 2000 года*

[Код]

- 3а. Система сигналов для представления букв и цифр при передаче сообщений.
- б. Система символов, букв или слов, которым присваиваются некоторые произвольно подобранные значения. Используется для передачи сообщений в случаях, когда требуется добиться конфиденциальности или краткости.
4. Система символов, применяемая для представления компьютерных команд...

Словарь английского языка американского наследия

Глава 1

Лучшие друзья

Вам десять лет. Ваш лучший друг живет на другой стороне улицы, напротив. Даже окна ваших спален обращены друг к другу. Каждый вечер родители объявляют отбой в безбожно ранний час, а вам еще хочется пообщаться, поделиться мыслями, наблюдениями, секретами, сплетнями, шутками и мечтами. Никто не вправе вас за это упрекнуть. В конце концов, стремление к коммуникации — одно из наиболее характерных человеческих качеств.

Пока в спальнях горит свет, можно помахать друг другу из окон и, полагаясь на примитивный язык тела, жестикулируя, обменяться парой мыслей. Однако передавать таким образом сложную информацию вряд ли удастся. И как только родители скамандуют: «Погаси свет!» — ситуация кажется безнадежной.

Как общаться? Может, по телефону? А был ли у вас в комнате телефон, когда вам было десять? Даже если так, то, где бы он ни находился, вас подслушают. Если ваш домашний компьютер подключен к телефонной линии, возможно, через него удастся поболтать бесшумно, но, опять же, компьютера в комнате нет.

Однако у вас с другом есть карманные фонарики. Все знают, что такой фонарик изобрели специально для чтения книжки под одеялом, но он отлично подходит для ночной коммуникации. Такая связь практически бесшумна, а луч света бьет прицельно, и, пожалуй, его не заметишь в щель под дверью. Бдительные домашние ничего не заподозрят.

Можно ли общаться при помощи вспышек? Попробовать точно стоит. В первом классе вы учились писать на бумаге слова и буквы, поэтому кажется уместным экстраполировать эти знания на обмен сигналами. Просто встаньте у окна и попытайтесь рисовать буквы светом. Чтобы написать *O*, включите фонарик, опишите им круг в воздухе, а потом выключайте. *I* — это вертикальная палочка. Но, как вы вскоре убедитесь, этот метод просто не работает. Наблюдая за фонариком друга, которым тот выводит в воздухе буквы, вы поймете, как сложно мысленно скомпоновать эти штрихи во что-то цельное. Завитушки и мазки света не слишком *точные*.

Наверняка вы видели в фильмах, как два морехода сигнализировали друг другу над водой мерцающими фонариками, один шпион покачивал зеркальцем, направляя свет сообщнику в другую комнату. Кажется, вот решение.

Сначала разработаем простой метод. Каждая буква алфавита соответствует последовательности бликов. Таким образом, один блик будет означать *А*, два — *Б*, три — *В* и т. д. Для *Я* уже понадобятся 33 блика. Слово «ГДЕ» — 4 блика + 5 бликов + 6 бликов, которые передаются с небольшими паузами, чтобы не перепутать эту серию с 15 бликами, то есть с *Н*. Паузы-пробелы между словами должны быть чуть длиннее.

Но вот что скажу: махать фонариком в воздухе больше не понадобится. Достаточно направить его куда нужно и нажимать на кнопочку. Но здесь возникает другая проблема: одно из первых сообщений, которое вы решите отправить («Как дела?»), растянется на 44 вспышки! Более того, придется забыть о пунктуации, ведь неизвестно, сколько бликов соответствуют вопросительному знаку.

Однако вы уже у цели. Вы предполагаете, что кто-то уже сталкивался с такой проблемой. Рано утром вы отправляетесь в библиотеку на поиски и узнаете о чудесном изобретении под названием «азбука Морзе». Вот то, что нужно, пусть даже теперь придется переучиваться, как пишутся все буквы алфавита.

В чем разница: в изобретенной вами системе каждой букве алфавита соответствует определенное количество бликов, от 1 для *А* до 33 для *Я*. В азбуке Морзе два вида бликов: краткие и длинные. Разумеется, при этом код Морзе получается сложнее, но на практике оказывается гораздо эффективнее. Теперь словосочетание «Как дела?» состоит всего из 24 бликов, а не из 44, причем с учетом кода вопросительного знака.

Обсуждая принцип работы азбуки Морзе, принято говорить не о долгих и кратких бликах, а о точках и тире, поскольку при помощи этих знаков удобно изображать код на печатной странице. В азбуке Морзе каждой букве алфавита соответствует краткая серия точек и тире, показанная в таблице на следующей странице.

Пусть азбука Морзе и не связана с компьютером, она помогает познать суть кода, а это важная предпосылка для глубокого понимания тайных языков и внутреннего устройства компьютерного харда и софта.

В этой книге слово «код» обычно означает систему передачи информации между людьми и машинами. Иными словами, код обеспечивает коммуникацию. Иногда покажется, что код — это шифр, но большинство кодов таковыми не являются, хотя и должны быть понятными, поскольку лежат в основе человеческого общения.

В начале романа «Сто лет одиночества» Габриэль Гарсия Маркес вспоминает времена, когда «мир был еще таким новым, что многие вещи не имели

Код

названия, и на них приходилось показывать пальцем». Создается впечатление, что лексемы присваиваются понятиям совершенно произвольно. Сложно понять, почему собаку называют собакой, а кошку — кошкой. Можно сказать, что словарь — это своеобразный код.

A	• —	R	• — •	A	• —	P	• — •
B	— • • •	S	• • •	Б	— • • •	С	• • •
C	— • — •	T	—	В	• — —	Т	—
D	— • • •	U	• • —	Г	— — •	У	• • —
E	•	V	• • • —	Д	— • • •	Ф	• • — •
F	• • — •	W	• — —	Е	•	Х	• • • •
G	— — •	X	— • • —	Ё	•	Ц	— • — •
H	• • • •	Y	— • — —	Ж	• • • —	Ч	— — — •
I	• •	Z	— — • •	З	— — • •	Ш	— — — —
J	• — — —			И	• •	Щ	— — • —
K	— • —			Й	• — — —	Ъ	— — • — —
L	• — • •			К	— • —	Ы	— • — —
M	— —			Л	• — • •	Ь	— • • —
N	— •			М	— —	Э	• • — • •
O	— — — —			Н	— •	Ю	• • — —
P	• — — •			О	— — — —	Я	• — • —
Q	— — • —			П	• — — •		

Звуки, которые мы произносим и складываем в слова, — код, понятный любому, кто слышит наш голос и понимает язык, на котором мы говорим. Этот код называется говорением, или речью. Существуют и другие коды для записи слов на бумаге (камне, дереве, в воздухе, например когда самолет выводит рекламные надписи в небе). Такой код — это и рукописные и печатные символы, которые мы видим в книгах, журналах или газетах. Мы называем его письменной речью, текстом. Во многих языках речь и текст согласуются друг с другом. Например, в английском буквы и буквосочетания (в большей или меньшей степени) соответствуют произносимым звукам.

Для глухих или немых был разработан иной код, облегчающий межличностное общение, — язык жестов, состоящий из движений рук, передающих отдельные буквы, слова или целые концепции. Для слепых письменный текст заменяется азбукой Брайля — системой выпуклых точек, соответствующих буквам, буквосочетаниям или целым словам. Когда приходится быстро записывать речь, удобно пользоваться стенографией или сокращениями.

При общении мы пользуемся различными кодами, поскольку одна кодировка удобнее других. Например, устную речь невозможно хранить на бумаге,

и ее заменяет письмо. Тихо передавать информацию на расстоянии невозможно ни при помощи речи, ни на бумаге. Удобная альтернатива — азбука Морзе. Далее мы увидим, что в компьютерах применяются различные типы кодов для передачи чисел, звуков, музыки, изображений и видео. Компьютер не может работать непосредственно с человеческими кодами: машина не в состоянии симитировать работу человеческих глаз, ушей, рта и пальцев. Недавно* в компьютерной технике наметилась такая тенденция: настольные ПК собирают и хранят различные виды информации, используемой при человеческом общении, и имеют возможность манипулировать такой информацией и ее отображениями. Это визуальная (текст, картинки) и акустическая (речь, звуки, музыка) информация, их комбинация (анимация или кино). Для всех этих типов требуются собственные коды, точно так же как при разговоре используются одни органы (рот и уши), а при письме и чтении — другие (руки и глаза).

Даже сама таблица с азбукой Морзе (с. 14) — в некотором роде код. В таблице каждая буква представлена последовательностью точек и тире. Но как передать точки и тире? Получается, они соответствуют бликам. Для обозначения точки мы быстро перещелкиваем кнопку фонарика (короткий блик), тире — задерживаем фонарик включенным чуть дольше. Так, чтобы передать А, мы быстро перещелкиваем фонарик, а потом включаем и выключаем его более медленно. Перед отправкой следующего символа делаем небольшую паузу. Принято, что тире должно быть примерно втрое длиннее точки. Так, если точка длится одну секунду, то тире — три (на самом деле азбука Морзе транслируется гораздо быстрее). Адресат видит короткий сигнал, затем длинный и понимает, что это А.

Паузы между точками и тире в азбуке Морзе критически важны. Так, при передаче А фонарик должен быть выключен между точкой и тире в течение периода, по длительности примерно равного одной точке. (Если точка длится одну секунду, то промежуток между точкой и тире также длится секунду.) Между буквами в слове выдерживаются более долгие паузы, сравнимые по длительности с тире (в данном случае по три секунды). Например, вот так на азбуке Морзе будет «привет» (обратите внимание на паузы между буквами).

• — — • • — • • • • — — • —

Между словами выдерживается период длительностью примерно два тире (шесть секунд, если тире — три секунды). Вот код фразы «как дела».

— • — • — — • — — • • • • — • • • —

* Книга написана в 2000 году, и автор отталкивается от реалий того периода. *Прим. перев.*

Длительность периодов, в течение которых фонарик остается включен или выключен, не фиксируется. Все периоды отсчитываются относительно длительности точки, а эта длина зависит от того, как быстро удастся переключивать фонарик, насколько быстро отправитель азбуки Морзе успевает вспомнить код для той или иной буквы. Тире у быстрого отправителя может получиться таким же коротким, как точка у неторопливого. Из-за этой небольшой проблемы расшифровка сообщений может усложняться, но после первых двух-трех букв адресат обычно успевает сориентироваться, где точка, а где тире.

На первый взгляд, определение кода Морзе — под *определением* в данном случае я понимаю соответствие различных последовательностей точек и тире буквам алфавита — кажется столь же произвольным, как и раскладка клавиатуры на пишущей машинке. Если присмотреться, не все так однозначно. Сравнительно простые и краткие коды присваиваются более частотным буквам алфавита, например *E* и *T**. Любители игр «Эрудит» и «Поле чудес» могли это сразу заметить. У редких букв (например, *Q* и *Z* на латинице, за которые в «Эрудите» присваивается по 10 очков) коды длиннее.

Практически каждый хоть немного знает азбуку Морзе. Три точки, три тире, три точки — SOS, международный сигнал бедствия. SOS не аббревиатура. Это просто код из азбуки Морзе, который легко запоминается. Во время Второй мировой войны Британская радиовещательная компания предвзяла некоторые передачи первыми нотами из Пятой симфонии Бетховена: ТА-ТА-ТА-ТАММММ! Сочиняя эту музыку, Людвиг ван Бетховен еще не мог знать, что именно такая последовательность сигналов (точка-точка-точка-тире) в азбуке Морзе будет соответствовать букве *V*, с которой начинается английское слово *victory* — «победа».

Один из недостатков азбуки Морзе в том, что в ней нет капитализации букв. Однако она позволяет передавать не только буквы, но и цифры, которым соответствуют свои последовательности по пять точек и тире.

1	• — — — —	6	— • • • •
2	• • — — —	7	— — • • •
3	• • • — —	8	— — — • •
4	• • • • —	9	— — — — •
5	• • • • •	0	— — — — —

Эти коды как минимум чуть более регулярны, чем буквенные. Для большинства знаков препинания используются по пять, шесть или семь точек и тире.

* В русском языке (по массивам текстов) это буквы *O*, *E*, *A*. *Прим. науч. ред.*

.	•••••	'	• — — — — •
,	• — • — • —	(— • — — • —
?	•• — — ••)	— • — — • —
:	— — — •••	=	— ••• —
;	— • — • —	+	• — • — •
-	— •••• —	\$	••• — •• —
/	— •• — •	¶	• — • — ••
"	• — •• — •	-	•• — — • —

Кроме того, существуют дополнительные коды для букв с диакритическими знаками из некоторых европейских языков и специальные последовательности-сокращения. Одно из таких сокращений — код SOS. Его следует посылать непрерывно, делая между каждой тройкой символов паузу в одну точку.

Вы убедитесь, что общаться с другом азбукой Морзе гораздо удобнее, если вооружиться специальным фонариком. Кроме обычного переключателя-ползунка, на такой фонарик монтируется кнопочный переключатель, который мы нажимаем и отпускаем, и фонарик зажигается и гаснет. Напрактиковавшись, вы, вероятно, научитесь передавать и принимать по пять-десять слов в минуту, что все равно гораздо медленнее, чем речь (при разговоре в минуту укладывается около 100 слов*), но вполне неплохо.

Когда вы с другом наконец-то выучите азбуку Морзе (а иначе общение при помощи этих сигналов не построить), вы сможете пользоваться таким словарем и в обычной речи. Для максимально быстрого общения произносите точку как «дих» («дит», если это последняя буква в слове), а тире — как «дах». Подобно тому как азбука Морзе позволяет сократить письмо до точек и тире, устный код редуцирует речь всего до двух слогов.

В данном случае ключевой элемент — *двойка*. Два типа бликов, два слога. Два любых феномена, если они разные, в правильных комбинациях подходят для передачи информации.

* При речи на английском языке. В русском языке темп речи (скорость произнесения ее элементов) медленнее, поскольку слова на 20–30% длиннее. *Прим. науч. ред.*

Глава 2

Коды и комбинации

Азбуку Морзе придумал Сэмюэл Финли Бриз Морзе (1791–1872). Это изобретение неотделимо от создания телеграфа, о работе которого нам также предстоит узнать. Азбука Морзе послужила хорошим вводным материалом для знакомства с сущностью кода, а телеграф — такой же удобный пример, иллюстрирующий аппаратное обеспечение компьютера.

Многим кажется, что азбуку Морзе проще передавать, чем принимать. Даже если вы не знаете ее на память, можете просто сверяться с таблицей, где буквы для удобства расставлены по алфавиту (с. 14).

Принимать азбуку Морзе и переводить ее в обычные слова значительно сложнее и дольше, поскольку вы работаете в обратном порядке: выясняете, какая буква соответствует конкретной кодовой последовательности точек и тире. Например, если вы получите сигнал «тире-точка-тире-тире», придется заглянуть в таблицу и просмотреть почти все буквы одну за другой, пока не выяснится, что перед вами *Ы*.

Проблема в том, что у нас есть таблица для следующего перевода:

*буква алфавита → последовательность азбуки Морзе,
состоящая из точек и тире.*

Однако нет обратной таблицы:

*последовательность азбуки Морзе, состоящая
из точек и тире, → буква алфавита.*

В начале изучения азбуки Морзе такая таблица, безусловно, пригодилась бы. Правда, не вполне понятно, как ее составить. Точки и тире не допускают никакого подобия алфавитного порядка.

Давайте забудем об алфавите. Пожалуй, разумнее сгруппировать коды таким образом, чтобы их расстановка зависела от количества точек и тире в той

или иной букве. Так, последовательность из азбуки Морзе, содержащая одну точку и одно тире, может означать всего одну из двух букв: *E* или *T*.

E	•	E	•
T	—	T	—

Комбинации, в которых содержится по два знака (либо точки, либо тире), дают нам уже четыре буквы: *I*, *A*, *H* и *M*.

I	••	И	••
A	•—	A	•—
N	—•	H	—•
M	— —	M	— —

Паттерн из трех символов, точек или тире, дает нам восемь букв: *C*, *D*, *U*, *K*, *P*, *G*, *O*, *B*.

S	•••	D	—••	C	•••	Д	—••
U	••—	K	—•—	У	••—	К	—•—
R	•—•	G	— —•	P	•—•	Г	— —•
W	• — —	O	— — —	B	• — —	O	— — —

Наконец (если мы хотим прекратить это упражнение, пока не перешли к цифрам и знакам препинания), четырехзначные последовательности точек и тире дают нам еще 16 символов.

H	••••	B	—•••	X	••••	Б	—•••
V	•••—	X	—••—	Ж	•••—	б	—••—
F	••—•	C	—•—•	Ф	••—•	Ц	—•—•
Ü	••— —	Y	—•— —	Ю	••— —	Ы	—•— —
L	•—••	Z	— —••	Л	•—••	З	— —••
Ä	•—•—	Q	— —•—	Я	•—•—	Щ	— —•—
P	• — —•	O	— — —•	П	• — —•	Ч	— — —•
J	• — — —	Ş	— — — —	Й	• — — —	Ш	— — — —

Всего в этих таблицах содержится $2 + 4 + 8 + 16$ кодов суммарно для 30 букв; это на четыре кода больше, чем требуется для полной латиницы, состоящей из 26 букв. Именно поэтому четыре кода в последней таблице отведены под буквы с диакритическими знаками.

Код

Эти четыре таблицы помогут с легкостью переводить любые сообщения, передаваемые азбукой Морзе. Получив код конкретной буквы, вы считаете, сколько в нем точек и тире, и решаете, с какой из таблиц сверяться. Каждая таблица устроена так, что код, состоящий из одних точек, располагается в верхнем левом углу, а код из одних тире — в нижнем правом углу.

Замечаете закономерность в *размерах* четырех таблиц? Обратите внимание: в каждой следующей таблице вдвое больше кодов, чем в предыдущей. Это логично: в последующей таблице содержатся все коды из предыдущей «плюс точка», а также все коды из предыдущей «плюс тире».

Эту тенденцию можно резюмировать следующим образом.

Количество точек и тире	1	2	3	4
Количество кодов	2	4	8	16

Каждая из четырех таблиц содержит вдвое больше кодов, чем предшествующая ей таблица, так что если в первой таблице 2 кода, то во второй — 2×2 кодов, в третьей — $2 \times 2 \times 2$ кодов. Вот как еще можно это представить.

Количество точек и тире	1	2	3	4
Количество кодов	2	2×2	$2 \times 2 \times 2$	$2 \times 2 \times 2 \times 2$

Разумеется, при умножении числа самого на себя можно использовать степени. Так, $2 \times 2 \times 2 \times 2$ можно записать как 2^4 (2 в четвертой степени). Числа 2, 4, 8 и 16 являются степенями двойки, поскольку представляют произведения, которые можно получить умножением двойки самой на себя. Итак, нашу таблицу можно переписать и так.

Количество точек и тире	1	2	3	4
Количество кодов	2^1	2^2	2^3	2^4

Таблица сильно упростилась. Количество кодов равно просто 2 в степени <количество точек и тире>. Можно резюмировать табличные данные в виде простой формулы:

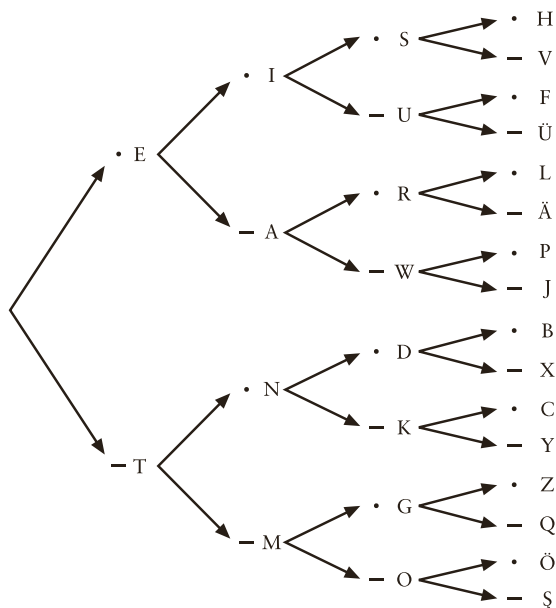
$$\text{Количество кодов} = 2^{\text{количество точек и тире}}$$

Степени двойки часто используются в различных кодах (другой пример рассмотрим в следующей главе).

Чтобы еще сильнее упростить расшифровку кода Морзе, давайте попробуем построить большую древовидную схему на следующей странице.

На схеме показано, какие буквы получаются при постепенном усложнении последовательностей точек и тире. Чтобы расшифровать конкретную последовательность, идите по стрелкам слева направо. Допустим, мы хотим выяснить, какая буква соответствует коду «точка-тире-точка». Начинаем слева, берем точку; далее идем по стрелкам, выбираем тире, а затем еще одну точку. Получаем букву *R*, расположенную около последней точки.

Такая схема необходима прежде всего для того, чтобы определить код Морзе. Во-первых, она страхует от тупой ошибки: не дает присвоить двум разным буквам один и тот же код. Во-вторых, вы гарантированно задействуете все возможные коды, не выстраивая чрезмерно длинных последовательностей из точек и тире.



Рискуя получить схему, которая не поместится на печатной странице, мы могли бы расширить ее и добавить туда пятизначные коды из точек и тире. Последовательность из пяти точек и тире даст нам 32 ($2 \times 2 \times 2 \times 2 \times 2$, или 2^5) дополнительных кода. Как правило, этого достаточно не только для букв, но и для 10 цифр и 18 знаков препинания, включаемых в азбуку Морзе: цифры действительно кодируются пятизначными последовательностями точек и тире. Правда, многие другие пятизначные коды зарезервированы не за знаками препинания, а за буквами с диакритическими знаками.

Чтобы система учитывала все знаки препинания, в нее нужно включить последовательности из шести точек и тире. Таким образом получим

Код

64 ($2 \times 2 \times 2 \times 2 \times 2 \times 2$, или 2^6) дополнительных кода для суммарного множества из $2 + 4 + 8 + 16 + 32 + 64$, или 126, символов. Для азбуки Морзе этого слишком много, поэтому большинство таких длинных кодов остаются неопределенными. Слово «неопределенный» в данном контексте указывает на код, который ничего не означает. Если бы вы, принимая азбуку Морзе, получили неопределенный код, то могли бы почти не сомневаться, что кто-то просто допустил ошибку.

У нас хватило смекалки построить эту небольшую формулу:

$$\text{Количество кодов} = 2^{\text{количество точек и тире}}$$

Так давайте продолжим нашу таблицу и посмотрим, сколько кодов получится из более длинных последовательностей точек и тире.

Количество точек и тире	Количество кодов
1	$2^1 = 2$
2	$2^2 = 4$
3	$2^3 = 8$
4	$2^4 = 16$
5	$2^5 = 32$
6	$2^6 = 64$
7	$2^7 = 128$
8	$2^8 = 256$
9	$2^9 = 512$
10	$2^{10} = 1024$

К счастью, нет необходимости выписывать все возможные коды, чтобы определить, сколько их будет. Достаточно умножить двойку на себя нужное количество раз.

Код Морзе называется *двоичным* (что буквально означает «два на два»), поскольку любой его элемент включает только два компонента: точку и тире. Такой код подобен монете, которая может упасть только решкой или орлом. Двоичные объекты (например, монеты) и двоичные коды (например, азбука Морзе) всегда можно описать в виде степеней двойки.

Проделанный нами анализ двоичных кодов — это простое упражнение в одной математической дисциплине, которая называется *комбинаторикой*, или *комбинаторным анализом*. Традиционно комбинаторный анализ особенно активно используется в теории вероятностей и статистике, поскольку связан с выявлением количества вариантов комбинаций различных объектов (например, монет или игральные кости). Он также помогает понять, как составляются и разбираются коды.

Глава 3

Брайль и двоичные коды

Сэмюэл Морзе не был первым, кому успешно удалось транслировать буквы письменного языка в интерпретируемый код. Он не был первым и среди тех, чья фамилия запомнилась как название кода, а не имя собственное. Такая честь выпала слепому французскому подростку, родившемуся примерно через 18 лет после Морзе, но оставившему след в истории гораздо раньше. О жизни Луи Брайля известно немного, но это захватывающая история.

Луи Брайль родился в 1809 году во французском городке Кувре, в 40 километрах к востоку от Парижа. Отец мальчика был шорником. Будучи трех лет от роду (а в таком возрасте дети не должны играть в отцовской мастерской), Луи случайно ткнул себе в глаз шорным ножом. В ране начался процесс заражения, инфекция распространилась и на второй глаз, и мальчик полностью ослеп. Наверняка его ждала жизнь в невежестве и бедности (как и большинство слепцов в те времена), но Луи проявил незаурядный ум и тягу к знаниям. Благодаря участию деревенского пастора и школьного учителя Луи ходил в сельскую школу вместе с другими ребятами, а в возрасте десяти лет отправился в Парижский государственный институт для слепых детей.

Разумеется, одна из главных сложностей при обучении незрячих в том, что они не могут читать печатные книги. Основатель этой парижской школы Валентин Гаюи (1745–1822) изобрел систему тисненых выпуклых букв для чтения их на ощупь. Но пользоваться системой было сложно, и вышло только несколько книг, напечатанных таким методом. Гаюи не смог посмотреть глубже. Для него буква А оставалась буквой А. Она должна была выглядеть (ощущаться) как А. (Общаясь на языке световых сигналов, мы пробовали рисовать буквы в воздухе и убедились, что такой прием неработоспособен.) Вероятно, Гаюи не догадался, что некий код, сильно отличающийся от печатного алфавита, оказался бы для незрячих удобнее.

Прообраз такого альтернативного кода возник в достаточно необычном контексте. Шарль Барбье, капитан французской армии, изобрел систему записи под названием *écriture nocturne*, или «ночная азбука». В ней использовались

узоры выпуклых точек и тире на плотной бумаге. Предполагалось, что солдаты могли бы обмениваться в темноте такими записками, когда требовалось соблюдать тишину. Писать точки и тире можно было специальным стилусом, вроде шила. Затем выпуклые точки можно было читать на ощупь. Недостаток системы Барбье заключался в ее чрезмерной сложности. Комбинации точек и тире соответствовали звукам, а не буквам алфавита, поэтому одно слово часто могло шифроваться разными кодами. Система хорошо работала для обмена короткими сообщениями в полевых условиях, но решительно не подходила для сравнительно крупных текстов, тем более книг.

Луи Брайль познакомился с системой Барбье в двенадцатилетнем возрасте. Ему понравились выпуклые точки не только потому, что они легко читались на ощупь, но и потому, что их было просто *писать*. Ученик в классе, вооружившись бумагой и стилусом, в самом деле мог записывать и читать такие сообщения. Луи Брайль постарался усовершенствовать эту систему, и через три года (когда ему было пятнадцать) в общих чертах составил собственную, основы которой применяются и сегодня. Много лет такая система использовалась лишь в школах, но постепенно вошла в широкое употребление. В 1835 году Брайль подхватил туберкулез, от которого и умер в 1852 году, в возрасте 43 лет.

Сегодня усовершенствованные варианты системы Брайля соперничают с аудиокнигами, обеспечивая незрячим доступ к письменной информации. Тем не менее шрифт Брайля по-прежнему незаменим и является единственной письменностью, доступной слепоглухим. Шрифт Брайля применяется даже в общественных местах, например в лифтах и банкоматах.

В этой главе мы препарлируем код Брайля и разберемся, как он работает. Мы не будем *учить* код Брайля или что-то запоминать. Мы лишь попробуем на этом примере лучше понять его природу.

В шрифте Брайля каждый символ, присутствующий в обычном письменном языке, то есть буквы, цифры и знаки препинания, кодируется в виде одной или нескольких точек в клетке размером две на три точки. Как правило, точки в клетке нумеруются от 1 до 6.

1	○	○	4
2	○	○	5
3	○	○	6

В настоящее время существуют специальные пишущие машинки — брайлевские принтеры, выбивающие точки брайлевского шрифта на бумаге.

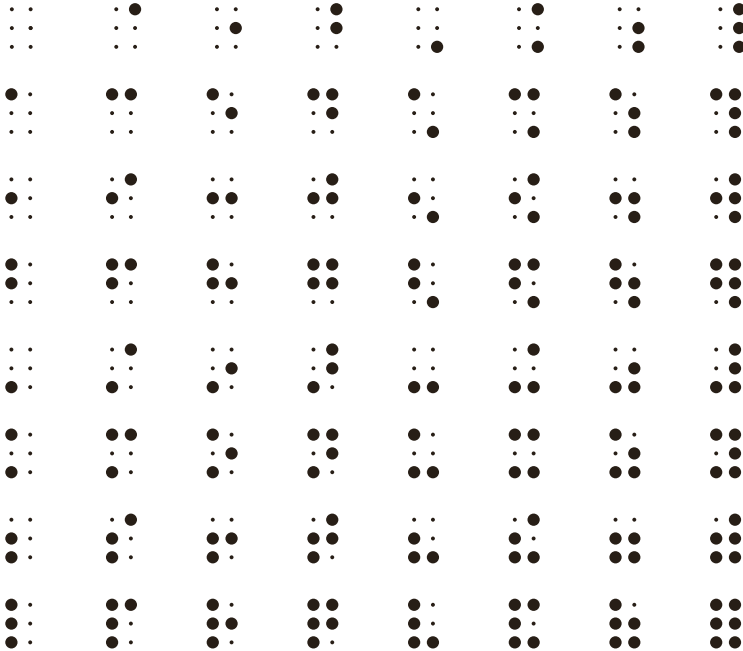
Поскольку книга получилась бы запредельно дорогой, если бы хоть пару страниц набрали шрифтом Брайля, я пользовался нотацией, традиционно применяемой для передачи азбуки Брайля при печати. В такой нотации отображаются

все шесть точек в клетке. Жирные точки — это выпуклости на бумаге, мелкие — плоские элементы клетки. Например, в следующем брайлевском символе точки 1, 3 и 5 выпуклые, а 2, 4 и 6 — нет.



На данный момент нас должно заинтересовать, что эти точки *двоичны*. Любая точка может быть либо выпуклой, либо плоской. Таким образом, шрифт Брайля подчиняется тем же принципам, которые знакомы нам из азбуки Морзе и комбинаторного анализа. Известно, что в клетке шесть точек, и каждая точка может быть плоской или выпуклой, поэтому общее число комбинаций, которые складываются из шести плоских или выпуклых точек, равно $2 \times 2 \times 2 \times 2 \times 2 \times 2$, или 2^6 , или 64.

Как видите, в системе Брайля можно представить 64 уникальных кода.



Если в шрифте Брайля используется менее 64 кодов, логично спросить, почему не все возможные варианты в ходу. Если в шрифте Брайля найдется более 64 возможных кодов, значит, сбоят либо наш разум, либо фундаментальные математические истины из разряда «два плюс два равно четырем».

Приступая к изучению шрифта Брайля, рассмотрим, как в нем записываются строчные буквы латиницы.

⠁	⠃	⠉	⠙	⠑	⠕	⠗	⠓	⠋	⠊
a	b	c	d	e	f	g	h	i	j

⠕	⠗	⠓	⠎	⠑	⠕	⠗	⠓	⠋	⠊
k	l	m	n	o	p	q	r	s	t

⠥	⠧	⠭	⠮	⠵
u	v	x	y	z

Например, английская фраза *You and me** записывается следующим образом.

⠠⠽⠠⠗⠠⠎⠠⠊⠠⠎⠠⠑
⠠⠽⠠⠗⠠⠎⠠⠊⠠⠎⠠⠑

Важно: между клетками, соответствующим буквам в слове, ставятся небольшие пробелы; более широкий пробел (в сущности, целая клетка, в которой нет выпуклых точек) соответствует пробелу между словами.

Именно такова основа шрифта Брайля в редакции самого Брайля — как минимум что касается латиницы. Луи Брайль также разработал коды для букв с диакритическими знаками (они часто встречаются во французском языке). Обратите внимание: здесь нет кода для буквы *w*, которая в классическом французском не используется. (Не волнуйтесь, и эта буква вскоре появится.) Пока мы учли всего 25 из 64 возможных кодов.

Внимательно присмотревшись к вышеприведенным строчкам, можно заметить, что в них прослеживается закономерность. В первой строчке (от *a* до *j*) в каждой клетке используются лишь четыре верхние точки: 1, 2, 4 и 5. Второй ряд точно такой же, как первый, но в нем есть и точка 3. Третий ряд подобен первым двум, но в нем мы видим не только точку 3, но и точку 6.

Со времени Луи Брайля его шрифт дополняли различным образом. Современная система, при помощи которой обычно записываются подобные английские тексты, называется «сокращенный Брайль». В сокращенном Брайле много упрощений, помогающих беречь деревья и ускорять чтение. Например, если код некоторой буквы стоит обособленно, то он означает распространенное слово. В следующих трех рядах приведены такие коды для целых слов.

* «Ты и я». *Прим. перев.*

(none) (отсутствует)	but но	can может	do делать	every каждый	from от, из	go идти	have иметь	(none)	just только, сейчас

knowledge знание	like как, любить	more больше	not не	(none)	people люди	quite весьма	rather вполне	so так	that так что

us нам, нас, нами	very очень	it оно	you ты	as как, в качестве	and и	for для	of (предлог)	the (артикль)	with с

Таким образом, фразу *You and me* сокращенным Брайлем можно записать так.

Вот мы и описали 31 код: пробел без точек, который ставится между словами, и три строки по десять кодов, используемых для обозначения букв и слов. Мы до сих пор и близко не израсходовали 64 теоретически доступных кода. Как мы убедимся, в сокращенном Брайле ни один не остался без дела.

Во-первых, можно использовать коды букв *a — j*, добавляя к каждому из них выпуклую точку 6. Эти коды применяются в основном для сокращения в слове букв, для буквы *w* и другого сокращения слов.

ch	gh	sh	th	wh	ed	er	ou	ow	w (или «will»*)

* *Will* — вспомогательный глагол для образования будущего времени.

Код

Например, слово about* можно записать сокращенным Брайлем вот так.



Во-вторых, можно взять коды букв *a — j* и «опустить» их так, чтобы использовались лишь точки 2, 3, 5 и 6. Этими кодами обозначаются некоторые знаки препинания и сокращения, в зависимости от контекста.

ea	bb	cc	dis	en	to	gg	his	in	was
,	;	:	.		к	!	()	“	”

Первые четыре приведенных кода — это запятая, точка с запятой, двоеточие и точка. Обратите внимание: как открывающая, так и закрывающая скобки обозначаются одним и тем же кодом, а вот коды для открывающей и закрывающей кавычки отличаются.

Пока мы использовали 51 код. Далее приведены шесть кодов, представляющих различные незадействованные комбинации точек 3, 4, 5 и 6. С их помощью записывают сокращения и некоторые дополнительные знаки препинания.

st	ing	ble	ar	'	com
/		#			-

Код *ble* очень важен: если это не часть слова, то он означает, что следующие далее коды должны интерпретироваться как числа. Числовые коды точно такие же, как и для букв *a — j*.

1	2	3	4	5	6	7	8	9	0

Следовательно, нижеприведенная последовательность означает 256.



* О, около, про. *Прим. перев.*

Если вы следите за нитью повествования, то помните, что до максимума (64) нам остается еще семь кодов. Вот они.



Первый код (выпуклая точка 4) — индикатор ударения. Остальные используются в качестве префиксов при некоторых сокращениях, а также в иных целях. Например, при выпуклых точках 4 и 6 (пятый код в этом ряду) код может означать либо десятичную запятую (для чисел), либо логическое ударение — в зависимости от контекста.

Наконец (если вам не терпится узнать, как в шрифте Брайля записываются заглавные буквы), у нас есть выпуклая точка 6. Это индикатор заглавной буквы. Следующая после такого символа буква будет в верхнем регистре. Например, имя создателя этой системы записывается так.



Здесь индикатор заглавной буквы, буква *l*, буквосочетание *oi*, буквы *i* и *s*, пробел, еще один индикатор заглавной буквы, а далее — буквы *b*, *r*, *a*, *i*, *l* и *e* (на практике эта запись может быть еще короче: отбрасываются две последние буквы, так как они не произносятся).

Итак, мы рассмотрели, как шесть двоичных элементов (точек) дают 64 возможных кода — и не больше. Получается, что многие из этих кодов выполняют двойную работу в зависимости от контекста. Особенно интересны «числовой» и «буквенный» индикаторы (при этом второй отменяет первый). Эти коды меняют семантику других кодов — тех, что следуют за ними: с букв на цифры и обратно с цифр на буквы. Подобные коды часто именуются кодами *старшинства* или *переключения*. Они меняют семантику всех последующих кодов до тех пор, пока переключение не будет отменено.

Индикатор заглавной буквы означает, что следующая (и только следующая) буква должна быть в верхнем, а не в нижнем регистре. Такой код принято называть *экранирующим*, и он «защищает» последовательность других кодов от банальной, рутинной семантики и обеспечивает им новую интерпретацию. Читая следующие главы, убедимся, что коды переключения и экранирующие коды постоянно используются в ситуациях, когда письменный язык нужно представить в двоичном виде.

