Сжатие данных. Сжатие без учёта контекста. Разделимые энтропийные коды

Александра Игоревна Кононова

МИЭТ

29 сентября 2025 г. — актуальную версию можно найти на https://gitlab.com/illinc/otik

Сжатие (компрессия, упаковка)

— кодирование • $|code(C)| < |C|$, • C однозначно и полностью восстанавливается по $code(C)$.

Любой алгоритм сжатия сжимает частые блоки данных за счёт увеличения более редких.

Свойства алгоритмов сжатия: 1. степень сжатия; 2. **степень увеличения размера в наихудшем случае;** 3. скорость сжатия и разжатия.

Источник X генерирует **входную последовательность** $C = c_1 c_2 \dots c_i \dots$ — символы пронумерованы (есть «предыдущий» и «последующий»); $c_i \in A$. Типы входной последовательности:

1. **блок** — конечная входная последовательность (произвольный доступ);
2. **поток** — с неизвестными границами (последовательный доступ).

Алгоритмы сжатия по её типу:

1. блочные — статистика всего блока добавляется к сжатому блоку;
2. поточные (адаптивные) — статистика вычисляется только для уже обработанной части потока, «на лету».

Алфавит; разрядность n и $count(a_j)$; порядок байтов

У современных ЭВМ общего назначения (не МК и не ЦСП) • байт 8-битный (октет), символ кодирования = октет = байт, первичный алфавит $A_1 \subseteq A = \{00{,}01, \ldots \mathtt{FF}\}$.	На доске для компактности будем считать • байт 3-битным (триадой), символ кодирования = байт = триада, первичный алфавит $A_1 \subseteq A = \{0{,}1, \ldots 7\}$.
Современные файловые системы 64-битны: • длина n файла в байтах (октетах) 64-битна ($long\,long$, 8 восьмибитных байтов) $\Longrightarrow$ ненормированные частоты символов $count(a_j)$ и пар $count(a_k a_j)$ тоже 64-битны (могут достигать n и $n-1$ соответственно).	Примем для доски: • длина n файла в трёхбитных байтах (триадах) 24-битна (8 трёхбитных байтов) $\Longrightarrow$ ненормированные частоты символов $count(a_j)$ и пар $count(a_k a_j)$ тоже 24-битны.
Порядок байтов в многобайтовых числах в памяти определяется архитектурой ЭВМ (для x86/amd64 — Intel), в файлах — обычно следует памяти.	Для многобайтовых чисел примем порядок байтов Intel: $n = 13_{10} = 15_8$ записывается как 5100 0000.

Блочные разделимые энтропийные коды для сжатия без учёта контекста

$code(c_1 c_2 \dots c_n) = code(c_1)\ code(c_2)\ \dots\ code(c_n)$, оптимальные разделимые — Хаффмана.

Каждый символ $a_j \in A$ (байт) заменяется кодом $code(a_j) \in \{0,1\}^+$:

- **разделимый** $\Longrightarrow$ префиксный (дерево кодов);
- блочный без учёта контекста $\Longrightarrow$ дерево кодов одно и то же для всего файла;
- сжатие $\Longrightarrow$ короткие $code(a_j)$ для частых a_j, длинные для редких (код по $p(a_j)$);
- **энтропийное** кодирование $\Longrightarrow$ $|code(a_j)| \to I_X(a_j)$. Модель X?

Блочный код без учёта контекста $\Leftrightarrow$ стационарная **модель без памяти** (БП, источник $X_{\text{БП}}$):

- $code(a_j)$ строится по $p(a_j)$ и постоянен $\Longrightarrow$ оцениваем $p(a_j)$ как постоянную;
- оптимальное сжатие конкретного файла $\Longrightarrow$ оцениваем по частотам в файле:

$$p_{\text{БП}}(a_j) = \frac{count(a_j)}{\sum count} = \frac{count(a_j)}{n}.$$

Другие модели $\Leftrightarrow$ код не блочный или с учётом контекста — но опт. р. тоже Хаффмана

Схема данных блочного кодирования Хаффмана (ШФ, Ш) без учёта контекста

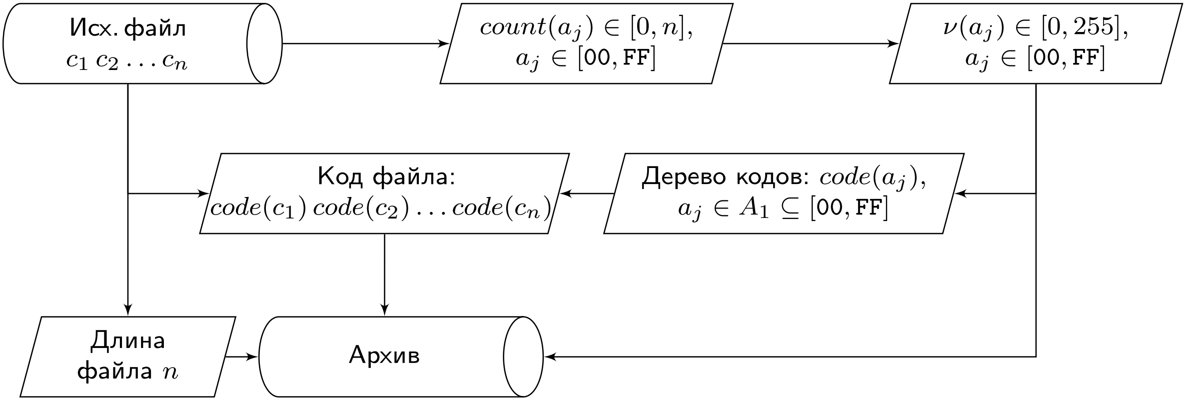
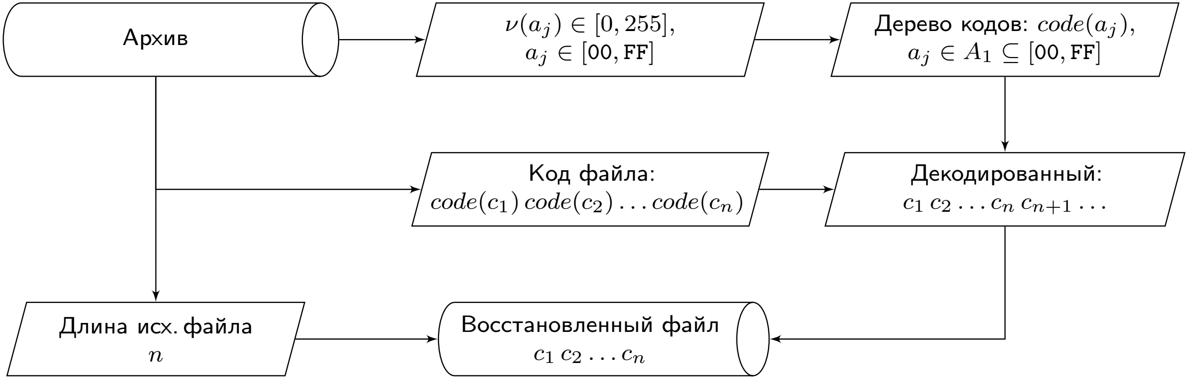


Схема данных декодирования блочного кода Хаффмана (ШФ, Ш) без учёта контекста



Модель без памяти: $I_{БП}(C)$ (суммарное) и $I(X_{БП})$ (среднее на символ)

$C = 7431\ 6500\ 4444\ 7$

$|C| = 13$ [триад] $= 39$ [бит]

символы считаем независимыми

$a_j \in A$	0	1	2	3	4	5	6	7
$\nu(a_j)$	2	1	0	1	5	1	1	2
$p_{БП}(a_j)$	$\frac{2}{13}$	$\frac{1}{13}$	0	$\frac{1}{13}$	$\frac{5}{13}$	$\frac{1}{13}$	$\frac{1}{13}$	$\frac{2}{13}$
$I_{БП}(a_j)$ [бит]	2,7	3,7	–	3,7	1,4	3,7	3,7	2,7

$I_{БП}(2) = +\infty$,

но

$p_{БП}(2) \cdot I_{БП}(2) = 0.$

$$p_{БП}(C) = p_{БП}(7) \cdot p_{БП}(4) \cdot p_{БП}(3) \cdot p_{БП}(1) \cdot p_{БП}(6) \cdot p_{БП}(5) \cdot p_{БП}(0) \cdot p_{БП}(0) \cdot p_{БП}(4) \cdot p_{БП}(4) \cdot p_{БП}(4) \cdot p_{БП}(4) \cdot p_{БП}(7) =$$

$$= \prod_{i=1}^{n} p_{БП}(c_i) = p_{БП}(0)^2 \cdot p_{БП}(1)^1 \cdot p_{БП}(2)^0 \cdot p_{БП}(3)^1 \cdot p_{БП}(4)^5 \cdot p_{БП}(5)^1 \cdot p_{БП}(6)^1 \cdot p_{БП}(7)^2 = \prod_{j=1}^{|A|} p_{БП}(a_j)^{count(a_j)}.$$

Далее для краткости вводим обозначение «a^ν» вместо «символ a с частотой ν»: $0^2, 1^1, 2^0, 3^1, 4^5, 5^1, 6^1, 7^2$.

Суммарное: $I_{БП}(C)$ [бит] $= -\log_2\left(p_{БП}(C)\right) = -\sum_{i=1}^{n} \log_2\left(p_{БП}(c_i)\right) = \sum_{i=1}^{n} I_{БП}(c_i) = \sum_{j=0}^{|A|} count(a_j) \cdot I_{БП}(a_j) =$

$= 2 \cdot I_{БП}(0) + 1 \cdot I_{БП}(1) + 1 \cdot I_{БП}(3) + 5 \cdot I_{БП}(4) + 1 \cdot I_{БП}(5) + 1 \cdot I_{БП}(6) + 2 \cdot I_{БП}(7) \approx 32{,}5$ бита $\approx 10{,}8$ триады.

$|code(C)| \to I_{БП}(C)$ — блочный код Хаффмана, **дерево кодов одно** на весь файл.

Среднее на символ: $I(X_{БП}) = \frac{I_{БП}(C)}{n} = \sum_{j=1}^{|A|} \frac{count(a_j)}{n} \cdot I_{БП}(a_j) = \sum_{j=1}^{|A|} p(a_j) \cdot I_{БП}(a_j) \approx 2{,}5$ бита $= 0{,}8$ триады

Коды Шеннона: предпосылки (исторически первый энтропийный код)

1. $\begin{cases} |code(X)| \to I(X) \text{ для оптимального кода,} \\ |code(X)| \geqslant I(X) \text{ всегда} \end{cases}$ $\implies$ длина кода Шеннона символа $\big|Ш(a_j)\big| = \big\lceil I_{БП}(a_j)\big\rceil$;
   - обозначим эту длину $l_j = \big\lceil I_{БП}(a_j)\big\rceil = \big\lceil -\log_2\big(p_{БП}(a_j)\big)\big\rceil$, тогда $l_j - 1 < -\log_2\big(p_{БП}(a_j)\big) \leqslant l_j$;
   - то есть $2^{-l_j} \leqslant p_{БП}(a_j) < 2^{-l_j+1}$: l_j — номер старшего ненулевого двоичного разряда $p_{БП}(a_j)$
2. символам с нулевыми частотами не нужен код $\implies$ исключение их из алфавита:
   $A = \{0^2, 1^1, 2^0, 3^1, 4^5, 5^1, 6^1, 7^2\}$, $|A| = 8 \quad \to \quad A_1 = \{0^2, 1^1, 3^1, 4^5, 5^1, 6^1, 7^2\}$, $|A_1| = 7$; • теперь все $p \neq 0$
3. сам код $Ш(a_j)$ строим как первые l_j двоичные цифры после запятой некоторого числа $0 \leqslant x(a_j) < 1$:
   - $x(a_j)$ должны быть для разных a_j разные; • пусть они монотонно возрастают от $x(a_1) = x_1 = 0{,}000\ldots$;
   - **накопленные вероятности** $x(a_j) = \sum\limits_{\ell=0}^{j-1} p_{БП}(a_\ell)$ — подходят;
   - тогда $Ш(a_1)$ — l_1 нулей ($x_1 = 0$); • $Ш(a_2)$ — l_2 цифр $x_2 = p(a_1)$; • № ст. ненул. дв. разр. $p(a_1)$ — это l_1;
   - если $l_2 < l_1$, то $Ш(a_2)$ тоже будет из всех нулей $\implies$ требуем $l_2 \geqslant l_1$ и далее $l_{j+1} \geqslant l_j$;
   - для этого **достаточно** сортировать A_1 **по убыванию частот:**
     $a_1 = 4^5$, $a_2, a_3 \in \{0^2, 7^2\}$, $a_4, a_5, a_6, a_7 \in \{1^1, 3^1, 5^1, 6^1\}$.

Выбор-1: порядок сортировки при равных частотах

- Энтропийные алгоритмы включают сортировку A_1 по убыванию частот (Шеннона и АС — принципиально по убыванию; Шеннона—Фано и Хаффмана — можно по возрастанию),
- при $\nu(a_i) = \nu(a_j)$ порядок не определён $\implies$ необходимо доопределить.

Обозначим «$a_i \succ a_j$» = «при $\nu(a_i) = \nu(a_j)$ сортируем как если $\nu(a_i) > \nu(a_j)$».

Всего $|A|!$ вариантов порядка, из них два удобных: $0 \succ 1 \succ \ldots \succ 7$ и $7 \succ 6 \succ \ldots \succ 0$.
Для лекции выберем $0 \succ 1 \succ 2 \succ \ldots \succ 7$.

Для алгоритма Хаффмана (сортируются не только символы $a_j \in A_1$, но и составные узлы S_i) выберем $\ldots \succ S_2 \succ S_1 \succ 0 \succ 1 \succ 2 \succ \ldots \succ 7$ (из восьми удобных вариантов).

Изменение порядка сортировки при равных частотах: • меняет коды; • не меняет общую длину.

Кодирование C: коды Шеннона, $0 \succ 1 \succ \ldots \succ 7$

$C = 7431\,6500\,4444\,7$, $|C| = 13$ [триад] $= 39$ [бит]. Код Шеннона строится не как дерево [но является деревом]:

1. символы сортируются по убыванию частот (выбор-1: при равных частотах $0 \succ 1 \succ \ldots \succ 7$);
2. код a_j — первые $l_j = \lceil I_{\text{БП}}(a_j) \rceil = \lceil -\log_2 p_{\text{БП}}(a_j) \rceil$ двоичных цифр $x_j = \sum_{\ell=0}^{j-1} p_{\text{БП}}(a_\ell)$ после запятой.

a_j	$p_{\text{БП}}(a_j)$	$I_{\text{БП}}(a_j)$, бит	l_j, бит	x_j	$Ш(a_j)$
4	$\frac{5}{13} \approx 0{,}01100\ldots$	$1{,}38\ldots$	2	$0 = 0{,}00000\ldots_2$	00
0	$\frac{2}{13} \approx 0{,}00100\ldots$	$2{,}70\ldots$	3	$\frac{5}{13} \approx 0{,}01100\ldots_2$	011
7	$\frac{2}{13} \approx 0{,}00100\ldots$	$2{,}70\ldots$	3	$\frac{7}{13} \approx 0{,}10001\ldots_2$	100
1	$\frac{1}{13} \approx 0{,}00010\ldots$	$3{,}70\ldots$	4	$\frac{9}{13} \approx 0{,}10110\ldots_2$	1011
3	$\frac{1}{13} \approx 0{,}00010\ldots$	$3{,}70\ldots$	4	$\frac{10}{13} \approx 0{,}11000\ldots_2$	1100
5	$\frac{1}{13} \approx 0{,}00010\ldots$	$3{,}70\ldots$	4	$\frac{11}{13} \approx 0{,}11011\ldots_2$	1101
6	$\frac{1}{13} \approx 0{,}00010\ldots$	$3{,}70\ldots$	4	$\frac{12}{13} \approx 0{,}11101\ldots_2$	1110

$Ш(C = 7431\,6500\,4444\,7) = 100\,00\,1100\,1011\quad 1110\,1101\,011\,011\quad 00\,00\,00\,00\quad 100$

$|Ш(C)| = 2 \cdot 5 + 3 \cdot 2 \cdot 2 + 4 \cdot 1 \cdot 4 \;=\; 44$ бита $= \lceil 14\frac{2}{3} \rceil$ триады $= 15$ триад (триада$\equiv$байт $\implies$ округление).

Не лучше Шеннона—Фано. Не используются.

Коды Шеннона: к вопросам 2025.09.16

1. Код Ш символа a_j — двоичные цифры накопленной вероятности $x_j = \sum_{\ell=0}^{j-1} p_\ell$, а не просто вероятности p_j, так как **коды разных** символов должны быть **разными,** а p_j разных символов a_j могут совпадать.

2. Зачем сортировка по убыванию частот — а) см. слайд 8 внизу; б) отсортируем неправильно (4 в конец):

a_j	$p_{\text{БП}}(a_j)$	$I_{\text{БП}}(a_j)$, бит	l_j, бит	неправильные x_j	неправильный Ш(a_j)
0	$\frac{2}{13} \approx 0{,}00100\ldots$	$2{,}70\ldots$	3	$0 = 0{,}00000\ldots_2$	000
7	$\frac{2}{13} \approx 0{,}00100\ldots$	$2{,}70\ldots$	3	$\frac{2}{13} \approx 0{,}00100\ldots_2$	001
1	$\frac{1}{13} \approx 0{,}00010\ldots$	$3{,}70\ldots$	4	$\frac{4}{13} \approx 0{,}01001\ldots_2$	0100
3	$\frac{1}{13} \approx 0{,}00010\ldots$	$3{,}70\ldots$	4	$\frac{5}{13} \approx 0{,}01100\ldots_2$	0110
5	$\frac{1}{13} \approx 0{,}00010\ldots$	$3{,}70\ldots$	4	$\frac{6}{13} \approx 0{,}01110\ldots_2$	0111
6	$\frac{1}{13} \approx 0{,}00010\ldots$	$3{,}70\ldots$	4	$\frac{7}{13} \approx 0{,}10001\ldots_2$	1000
4	$\frac{5}{13} \approx 0{,}01100\ldots$	$1{,}38\ldots$	2	$\frac{8}{13} \approx 0{,}10011\ldots_2$	10 — код не префиксный: начало Ш(6)

Накопленные вероятности $x(6)$ и $x(4)$ различаются в четвёртом знаке, но в Ш(4) этот знак не попал

Коды Шеннона–Фано: предпосылки (исторически первый близкий к оптимальному), построение

Коды Шеннона—Фано строятся как бинарное дерево:

- листья упорядочены по убыванию частот (выбор-1: при равных частотах $0 \succ 1 \succ \ldots \succ 7$);
- дочерние ветви промаркированы как 0 и 1 (выбор-2: порядок маркировки?);
- сбалансированное (суммы частот обеих дочерних ветвей равны; выбор-3: деление?).

Дерево Шеннона—Фано строится **сверху вниз** (от корня к листьям):

1. все символы сортируются по частоте;
2. упорядоченный ряд символов в некотором месте делится на две части так, чтобы в каждой из них сумма частот символов была примерно одинакова (без пересортировки!);
3. новое деление.

Выбор-2: маркировка ветвей

Маркировка ветвей для Шеннона—Фано и Хаффмана (для лекций выбираем 0/1):

1. 0/1) 0 со стороны бо́льших частот (слева), 1 со стороны меньших частот (справа);
2. 1/0) 1 со стороны бо́льших, 0 — меньших; 3-∞ сложные схемы — возможно, но неудобно.

Изменение схемы маркировки: • меняет коды; • не меняет их длину.

Выбор-3 (только Шеннона—Фано): правило деления, если нельзя точно пополам

Основные варианты деления s на $s_1 + s_2$: 1 ШФД1) $\min\limits_{s_1 \leqslant s_2} |s_2 - s_1|$;

2 ШФД2) $\min |s_2 - s_1|$, если он достигается в одной точке; если в двух: $\min\limits_{s_1 \leqslant s_2} |s_2 - s_1|$.

Изменение правила: • меняет коды; • **меняет длину кода** (нет однозначно лучшего правила).

Для лекций выбираем ШФД2. Для $C = 7431\,6500\,4444\,7$ ШФД1 лучше (код с ШФД1 короче).

Кодирование C: Шеннона–Фано, $0 \succ 1 \succ \ldots \succ 7$, 0/1, ШФД2

$C = 7431\,6500\,4444\,7$, $|C| = 13$ триад $= 39$ бит, $I_{\text{БП}}(C) = 32{,}5$ бита $= 10{,}8$ триады.

1) $\left(4^5, 0^2, 7^2, 1^1, 3^1, 5^1, 6^1\right)^{13} \to \underbrace{\left(4^5, 0^2\right)^7}_{\text{коды начинаются с 0}} + \underbrace{\left(7^2, 1^1, 3^1, 5^1, 6^1\right)^6}_{\text{коды начинаются с 1}}$

2) $\underbrace{\left(4^5, 0^2\right)^7}_{\text{коды начинаются с 0}} \to \underbrace{4^5}_{00} + \underbrace{0^2}_{01}$ и т. д.:

<table>
<tr><td>4^5</td><td>0^2</td><td>7^2</td><td>1^1</td><td>3^1</td><td>5^1</td><td>6^1</td></tr>
<tr><td colspan="2">0</td><td colspan="5">1</td></tr>
<tr><td>0</td><td>1</td><td colspan="2">0</td><td colspan="3">1</td></tr>
<tr><td></td><td></td><td>0</td><td>1</td><td>0</td><td colspan="2">1</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>0</td><td>1</td></tr>
<tr><td>00</td><td>01</td><td>100</td><td>101</td><td>110</td><td>1110</td><td>1111</td></tr>
</table>

ШФ$(C = 7431\,6500\,4444\,7) = 100\,00\,110\,101 \quad 1111\,1110\,01\,01 \quad 00\,00\,00\,00 \quad 100$

$|$ШФ$(C)| = 2\cdot5 + 2\cdot2 + 3\cdot2 + 3\cdot1\cdot2 + 4\cdot1\cdot2 = 34$ бит $= \left\lceil 11\tfrac{1}{3} \right\rceil$ триады $= 12$ триад.

Не лучше Хаффмана (при аналогичной скорости). Не используются.

Коды Хаффмана: предпосылки (длинные коды — редким символам), построение

- если в дереве оптимального кода максимальная длина кода $l_{\max}$, то есть и второй код длины $l_{\max}$, с тем же родительским узлом (иначе можно укоротить $\implies$ не оптимальный);
- самые длинные коды ($l_{\max}$ и $l_{\max}$) должны достаться двум самым редким символам;
- пусть A_1 отсортирован по убыванию частот: самые редкие $a_{|A_1|}$ и $a_{|A_1|-1}$ $\implies$ у $a_{|A_1|}$ и $a_{|A_1|-1}$ общий родительский узел (обозначим S_1)...

Коды Хаффмана строятся как бинарное дерево **снизу вверх** (от листьев к корню):

1. все символы алфавита A_i (узлы) сортируются по частоте (по убыванию);
2. два последних (самых редких) узла $a_{|A_i|-1}$ и $a_{|A_i|}$ отсортированного A_i заменяются на новый узел S_i с частотой, равной сумме исходных: $\nu(S_i) = \nu(a_{|A_i|-1}) + \nu(a_{|A_i|})$;
3. новый алфавит A_{i+1} (короче старого: $|A_{i+1}| = |A_i| - 1$) $\implies$ новая сортировка.

Узел, полученный на последнем, $(|A_1| - 1)$-м шаге — корень.

Выбор-1 (сортировка при равных частотах) для Хаффмана — восемь, а не две!

- Порядок $0 \succ 1 \succ 2 \succ \ldots \succ 7$ (по возрастанию) для символов:

РЧС1	$\ldots \succ S_2 \succ S_1 \succ$ 00 $\succ$ 01 $\succ$ 02 $\succ \ldots \succ$ FF	$\ldots \succ S_2 \succ S_1 \succ 0 \succ 1 \succ 2 \succ \ldots \succ 7$
РЧС2	$S_1 \succ S_2 \succ \ldots \succ$ 00 $\succ$ 01 $\succ$ 02 $\succ \ldots \succ$ FF	$S_1 \succ S_2 \succ \ldots \succ 0 \succ 1 \succ 2 \succ \ldots \succ 7$
РЧС3	00 $\succ$ 01 $\succ$ 02 $\succ \ldots \succ$ FF $\succ \ldots \succ S_2 \succ S_1$	$0 \succ 1 \succ 2 \succ \ldots \succ 7 \succ \ldots \succ S_2 \succ S_1$
РЧС4	00 $\succ$ 01 $\succ$ 02 $\succ \ldots \succ$ FF $\succ S_1 \succ S_2 \succ \ldots$	$0 \succ 1 \succ 2 \succ \ldots \succ 7 \succ S_1 \succ S_2 \succ \ldots$

- Порядок $7 \succ 6 \succ \ldots \succ 0$ (по убыванию) для символов:

РЧС5	$\ldots \succ S_2 \succ S_1 \succ$ FF $\succ$ FE $\succ$ FD $\succ \ldots \succ$ 00	$\ldots \succ S_2 \succ S_1 \succ 7 \succ 6 \succ \ldots \succ 0$
РЧС6	$S_1 \succ S_2 \succ \ldots \succ$ FF $\succ$ FE $\succ$ FD $\succ \ldots \succ$ 00	$S_1 \succ S_2 \succ \ldots \succ 7 \succ 6 \succ \ldots \succ 0$
РЧС7	FF $\succ$ FE $\succ$ FD $\succ \ldots \succ$ 00 $\succ \ldots \succ S_2 \succ S_1$	$7 \succ 6 \succ \ldots \succ 0 \succ \ldots \succ S_2 \succ S_1$
РЧС8	FF $\succ$ FE $\succ$ FD $\succ \ldots \succ$ 00 $\succ S_1 \succ S_2 \succ \ldots$	$7 \succ 6 \succ \ldots \succ 0 \succ S_1 \succ S_2 \succ \ldots$

В Кр1 и в отчёте о л/р лучше указывать человекочитабельный порядок ($\ldots \succ S_2 \succ S_1 \succ$ 00 $\succ$ 01 $\succ$ 02 $\succ \ldots \succ$ FF), а не случайно выбранный номер (РЧС1).

Выбор-2 (маркировка ветвей) — как для Шеннона—Фано: 0/1, 1/0 и неудобные

Кодирование C: Хаффмана, ... $\succ S_2 \succ S_1 \succ 0 \succ 1 \succ ... \succ 7$, 0/1

$C = 7431\,6500\,4444\,7$, $|C| = 13$ триад $= 39$ бит, $I_{БП}(C) = 32{,}5$ бита $= 10{,}8$ триады.

1) $4^5, 0^2, 7^2, 1^1, 3^1, \underbrace{5^1, 6^1}_{0 \quad S_1^2 \quad 1}$ — последний бит кода 5^1 — 0, последний бит кода 6^1 — 1

2) $4^5, S_1^2, 0^2, 7^2, \underbrace{1^1, 3^1}_{0 \quad S_2^2 \quad 1}$

3) $4^5, S_2^2, S_1^2, \underbrace{0^2, 7^2}_{0 \quad S_3^4 \quad 1}$

4) $4^5, S_3^4, \underbrace{S_2^2, S_1^2}_{0 \quad S_4^4 \quad 1}$

5) $4^5, \underbrace{S_4^4, S_3^4}_{0 \quad S_5^8 \quad 1}$

6) $\underbrace{S_5^8, 4^5}_{0 \quad S_6^{13} \quad 1}$

0^2	1^1	2^0	3^1	4^5	5^1	6^1	7^2
010	0000	—	0001	1	0010	0011	011

Хф$(C = 7431\,6500\,4444\,7) = 011\,1\,0001\,0000 \quad 0011\,0010\,010\,010 \quad 1\,1\,1\,1 \quad 011$
|Хф$(C)| = 1 \cdot 5 + 3 \cdot 2 \cdot 2 + 4 \cdot 1 \cdot 4 = 33$ бита $= 11$ триад.

Код Хаффмана имеет минимальную длину среди разделимых энтропийных кодов.

В худшем случае |Хф$(C)| = |C|$ (не увеличивает размера исходных данных, если не считать заголовка архива и частот).

АС — арифметические (интервальные) коды

Арифметический (интервальный) код, АС — **неразделимый** энтропийный код: код сообщения $C = c_1c_2...c_n$ не разделяется на $code(c_1)$, $code(c_2)$ и т. д.

$C = c_1c_2...c_n \quad \rightarrow \quad z \in [0, 1)$; $\qquad (0, 1) \simeq \mathbb{R}$

$I(z) \approx I(C)$, $\qquad$ и чаще всего $I(z) >> 64$ бит $> I(\texttt{double})$

Концепт АС — всегда не хуже Хаффмана и иногда лучше;
реализации АС могут быть хуже Хаффмана (искажение частот + потеря точности).

Запись кода в файл

На примере кода Шеннона—Фано со стр. 14 длины $11\frac{1}{3}$ триады (трёхбитного байта):
$code(C = 7431\,6500\,4444\,7) = 1000011010111111100101000000000100$

1. При записи в файл код дополняется до целого числа байтов (до 12 триад), обычно нулями:
   $code(C) = 100\,001\,101\,011 \quad 111\,111\,001\,010 \quad 000\,000\,010\,0\textcolor{red}{00} = 4153\,7712\,0020$
   при декодировании добавленные биты 00 будут прочитаны как лишний символ «4»:
   $decode(4153\,7712\,0020) = 7431\,6500\,4444\,74 \neq C \implies$ проверять исходную длину $n = 13$.
2. Для восстановления дерева необходимы исходные частоты байтов:
   $\vec{\nu} = \Big(\nu(0), \nu(1), \nu(2), \nu(3), \nu(4), \nu(5), \nu(6), \nu(7)\Big) = (2, 1, 0, 1, 5, 1, 1, 2) = 2101\,5112.$
3. Необходимо правильно выбрать алгоритм кодирования/декодирования.

Вырожденный случай

Для $C = 4444444444444$ из $n = 13_{10} = 15_8$ байтов $I_{\text{БП}}(C) = 0$:

- длина кода Шеннона символа 4 равна нулю, так как $I_{\text{БП}}(4) = 0$;
- длина кода Хаффмана и Шеннона—Фано символа 4 равна нулю, так как дерево состоит из одного узла (корня 4) и нуля ветвей.

Длина кода (Хаффмана, Шеннона—Фано или Шеннона) всего сообщения из n одинаковых символов 4 также **нулевая.**

Файл архива должен содержать n и массив приведённых к байту частот $\vec{\nu}$:

$15, \quad 00007000$

этого достаточно для восстановления такого сообщения.

Код и архив с кодами

Смещение	Размер	Описание	
0	4	Сигнатура+версия формата	всегда 0711
4	1	№ алгоритма сжатия с контекстом	0 — нет сжатия
5	1	№ алгоритма сжатия без контекста	0 — нет сжатия, 1 — Шеннона со стр. 10, 2 — Шеннона—Фано со стр. 14, 3 — Хаффмана со стр. 17
6	1	№ алгоритма шифрования	0 — нет шифрования
7	1	№ алгоритма защиты от помех	0 — нет защиты от помех
8	8	Исходная длина файла n	беззнаковое 24-битное целое
16	8	Массив частот $\vec{\nu} = (\nu(0), \nu(1), \ldots, \nu(7))$	беззнаковые 3-битные целые
24	до конца	Сжатые данные	выравнивание на 1 байт

- Код Шеннона—Фано со стр. 14: № алгоритма 0200;
- исх. длина $n = 13_{10} = 15_8$ триад $(5100\,0000)$;
- частоты $\vec{\nu} \sim 2101\,5112$, код $4153\,7712\,0020$;

$\Longrightarrow$ архив $0711\,0200\,5100\,0000 \quad 2101\,5112\,4153\,7712 \quad 0020$.

Приведение частот: $count(a_j) \to \nu(a_j)$

Ненормированное $count(a_j)$ может превышать допустимое $\max(\nu(a_j)) \implies$ приведение:

$$\begin{cases} \nu_0 : \nu_1 : \ldots : \nu_N \approx count(0) : count(1) : \ldots : count(N), \\ \max(\nu_i) = \text{максимальное значение байта.} \end{cases}$$

Для $m = 7754\,4444\,4444\,4444\,4444\,3333\,3333\,3333\,3333\,1112$ длины $n = 40_{10} = 50_8$ $\overrightarrow{count} = (0, 3, 1, 16, 17, 1, 0, 2)$, но $\vec{\nu} = (0, 2, 1, 7, 7, 1, 0, 1)$.

В архив записываются:

- исходная длина $n = 40_{10} = 50_8$ разрядности длины файла — $0500\,0000$;
- приведённые к $\max(\nu_i) = 7$ частоты $0217\,7101$;
- код, рассчитанный по $\vec{\nu} = (0, 2, 1, 7, 7, 1, 0, 1)$, а не по исходным $\overrightarrow{count}$.

Нулевые частоты и приведение частот

1. По умолчанию байты с нулевыми ν_i отбрасываются и не получают кода. Тогда при приведении частот $count(i) \in [0, \max(count)] \;\to\; \nu_i \in [0, Max]$ необходимо, чтобы при $count(i) > 0$ было $\nu_i > 0$:
   - соотношения всех частот незначительно искажаются:
   $$\begin{cases} \nu_i = 0, & count(i) = 0, \\ \nu_i = \mathrm{round}\left(\frac{count(i)-1}{\max(count)-1} \cdot (Max-1)\right) + 1, & count(i) > 0; \end{cases} \qquad \text{(A)}$$
   - для $count(i) > \frac{\max(count)}{Max}$ передаются максимально точно; для малых полностью искажаются:
   $$\begin{cases} \nu_i = 0, & count(i) = 0, \\ \nu_i = 1, & 0 < count(i) \leqslant \frac{\max(count)}{Max}, \\ \nu_i = \mathrm{round}\left(\frac{count(i)}{\max(count)} \cdot Max\right), & count(i) > \frac{\max(count)}{Max}; \end{cases} \qquad \text{(B)}$$

   для октетов ($Max = 255$) и $\dfrac{\max(count)}{\min\limits_{\neq 0}(count)} \leqslant Max$ обе формулы дают приемлемый результат.

2. Если хочется $\nu_i = \mathrm{round}\left(\frac{count(i)}{\max(count)} \cdot Max\right)$ для всех (возможно $count(i) > 0 \;\to\; \nu_i = 0$), то:
   - необходимо модифицировать алгоритм, чтобы байты с $\nu_i = 0$ получили коды (возможно для Хаффмана и Шеннона—Фано, невозможно для АС и Шеннона);
   - тогда коды получат и байты с $count(i) = 0$, а коды $count(i) > 0$ удлинятся.

Марковская модель первого порядка-1, $Y_{\text{М1}}$: $I_{\text{М1}}(C)$ (суммарное); $I_{\text{М1}}(C) \neq I_{\text{БП}}(C)$

$C = 7431\ 6500\ 4444\ 7$, $|C| = 13$ [триад] $= 39$ [бит], считаем $p_{\text{М1}}(c_1 = 0) = p_{\text{М1}}(c_1 = 1) \ldots = \frac{1}{|A|} = \frac{1}{8}$.

$\nu(xa_j)$	0	1	2	3	4	5	6	7
x = 0	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0
4	0	0	0	1	3	0	0	1
5	1	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	1	0	0	0

$p_{\text{М1}}(a_j\|x)$	0	1	2	3	4	5	6	7
x = 0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0
4	0	0	0	$\frac{1}{5}$	$\frac{3}{5}$	0	0	$\frac{1}{5}$
5	1	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	1	0	0	0

$I_{\text{М1}}(a_j\|x)$ [бит]	0	1	2	3	4	5	6	7
x = 0	1				1			
1							0	
2								
3		0						
4				2,3	0,7			2,3
5	0							
6						0		
7					0			

$$p_{\text{М1}}(C) = p_{\text{М1}}(7) \cdot p_{\text{М1}}(4|7) \cdot p_{\text{М1}}(3|4) \cdot p_{\text{М1}}(1|3) \cdot p_{\text{М1}}(6|1) \cdot \ldots \cdot p_{\text{М1}}(4|4) \cdot p_{\text{М1}}(7|4) = \frac{1}{8} \cdot 1 \cdot \frac{1}{5} \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{3}{5} \cdot \frac{3}{5} \cdot \frac{3}{5} \cdot \frac{1}{5}$$

$$I_{\text{М1}}(C) \text{ [бит]} = -\log_2\Big(p_{\text{М1}}(C)\Big) = -\log_2\Big(p_{\text{М1}}(7)\Big) - \log_2\Big(p_{\text{М1}}(4|7)\Big) - \log_2\Big(p_{\text{М1}}(3|4)\Big) - \ldots - \log_2\Big(p_{\text{М1}}(7|4)\Big) =$$
$$= I_{\text{М1}}(7) + I_{\text{М1}}(4|7) + \ldots + I_{\text{М1}}(7|4) \approx 3+0+2{,}3+0+0+0+0+1+1+0{,}7+0{,}7+0{,}7+2{,}3 \approx 11{,}8 \text{ бит} \approx 4{,}0 \text{ триады.}$$

$I(Y_{\text{М1}})$ — усреднять $I_{\text{М1}}(C)$ по всем сообщениям C (n-символьное $\Longrightarrow$ $|A|^n$ вариантов) и по длине $|C|$.

Оптимальный разделимый код М1: c_1 пишем как есть (байтом), далее блочный код Хаффмана, **дерево на каждом шаге перестраивается** по строке $\nu(c_{i-1}y)$ [оптимальный неразделимый = АС, частоты меняются]. Требует $|A|^2$ частот.

Марковская модель первого порядка-2, $Y_{\widetilde{\text{М1}}}$: не имеет смысла

$C = 7431\ 6500\ 4444\ 7$, $|C| = 13$ [триад] $= 39$ [бит],

считаем $p_{\widetilde{\text{М1}}}(c_1 = a_j) = p_{\text{БП}}(a_j)$, при $i \geqslant 2$: $p_{\widetilde{\text{М1}}}(c_i = a_j) = p_{\text{М1}}(a_j | c_{i-1})$

$$p_{\widetilde{\text{М1}}}(C) = p_{\text{БП}}(7) \cdot p_{\text{М1}}(4|7) \cdot p_{\text{М1}}(3|4) \cdot p_{\text{М1}}(1|3) \cdot p_{\text{М1}}(6|1) \cdot \ldots \cdot p_{\text{М1}}(4|4) \cdot p_{\text{М1}}(7|4)$$

$$I_{\widetilde{\text{М1}}}(C)\ [\text{бит}] = -\log_2\left(p_{\widetilde{\text{М1}}}(C)\right) = I_{\text{БП}}(7) + I_{\text{М1}}(4|7) + \ldots + I_{\text{М1}}(7|4) =$$
$$= I_{\text{БП}}(7) + (I_{\text{М1}}(C)\ [\text{бит}] - 3) = 2{,}7 + (11{,}8 - 3)\ \text{бит} \approx 11{,}6\ \text{бит}.$$

ν	0	1	2	3	4	5	6	7
0	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0
4	0	0	0	1	3	0	0	1
5	1	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	1	0	0	0
—	2	1	0	1	5	1	1	2

В общем случае: $I_{\widetilde{\text{М1}}}(C) = I_{\text{М1}}(C) - \log_2(|A|) + I_{\text{БП}}(c_1)$.
Если c_1 — редкий, то $I_{\widetilde{\text{М1}}}(C) > I_{\text{М1}}(C)$.

Оптимальный разделимый код $\widetilde{\text{М1}}$: блочный код Хаффмана, дерево кодов перестраивается на каждом шаге, включая первый.

Отличается от оптимального кода М1 только первым шагом:

- в лучшем случае экономим менее байта на c_1, но частоты $|A|^2 + |A|$ байтов (против $|A|^2$ для М1);
- в худшем случае (c_1 — редкий) теряем и на c_1, и на частотах.

Марковская модель первого порядка-3, $Y_{\widetilde{\widetilde{\text{M1}}}}$

$C = 7431\ 6500\ 4444\ 7, \quad |C| = 13$ [триад] $= 39$ [бит],

считаем $p_{\widetilde{\widetilde{\text{M1}}}}(c_1 = 7) = 1, \quad$ при $i \geqslant 2$: $p_{\widetilde{\widetilde{\text{M1}}}}(c_i = a_j) = p_{\text{M1}}(a_j | c_{i-1})$:

$$p_{\widetilde{\widetilde{\text{M1}}}}(C) = 1 \cdot p_{\text{M1}}(4|7) \cdot p_{\text{M1}}(3|4) \cdot p_{\text{M1}}(1|3) \cdot p_{\text{M1}}(6|1) \cdot \ldots \cdot p_{\text{M1}}(4|4) \cdot p_{\text{M1}}(7|4)$$

$$I_{\widetilde{\widetilde{\text{M1}}}}(C)\ [\text{бит}] = -\log_2\left(p_{\widetilde{\widetilde{\text{M1}}}}(C)\right) = 0 + I_{\text{M1}}(4|7) + \ldots + I_{\text{M1}}(7|4) =$$
$$= 0 + (I_{\text{M1}}(C)\ [\text{бит}] - 3) = 11{,}8 - 3\ \text{бит} \approx 10{,}8\ \text{бит}.$$

В общем случае: $I_{\widetilde{\widetilde{\text{M1}}}}(C) = I_{\text{M1}}(C) - \log_2(|A|) < I_{\text{M1}}(C)$.

ν	0	1	2	3	4	5	6	7
0	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0
4	0	0	0	1	3	0	0	1
5	1	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	1	0	0	0
–	0	0	0	0	0	0	0	1

Оптимальный разделимый код $\widetilde{\widetilde{\text{M1}}}$: блочный код Хаффмана, дерево кодов перестраивается на каждом шаге, первый шаг — ничего не пишем (вырожденный случай).

Оптимальные коды $\widetilde{\widetilde{\text{M1}}}$ и M1 отличаются только первым шагом:

- экономим байт на c_1, теряем $|A|$ байтов на частотах ($|A|^2 + |A|$ байтов для $\widetilde{\widetilde{\text{M1}}}$ против $|A|^2$ для M1).

если хранить только байт $c_1 = 7$, а не частоты $0000\,0001$ — получится оптимальный код M1.

Марковская модель порядка N и неэффективность оптимального кода

Оптимальный разделимый код марковской модели порядка N:

- $c_1 c_2 \ldots c_N$ пишем как есть (один символ одним байтом);
- на каждом шаге $i \geqslant N$ дерево Хаффмана перестраивается по частотам $\vec{\nu} = \Big(\nu(c_{i-N} \ldots c_{i-2} c_{i-1}\, 0),\;\; \nu(c_{i-N} \ldots c_{i-2} c_{i-1}\, 1),\;\; \nu(c_{i-N} \ldots c_{i-2} c_{i-1}\, 2), \ldots\Big)$.

Оптимальный неразделимый — аналогичный АС.

Оптимальный код требует $|A|^{N+1}$ частот (все $\nu(c_{i-N} \ldots c_{i-2} c_{i-1}\, y)$) $\implies$ невыгоден.
На практике энтропийное кодирование применяется только для сжатия без учёта контекста.

Методы сжатия с учётом контекста (семейства RLE, LZ77, LZ78):
длина кода C существенно больше, чем $I_{\mathsf{M}N}(C)$, но для декодирования не требуются частоты.

Модель S. Поточный код Хаффмана без учёта контекста

$C = 7431\ 6500\ 4444\ 7$, $|C| = 13$ [триад] $= 39$ [бит],

считаем $p_S(c_i = a_j) = \frac{\nu_i(a_j)}{\sum \nu_i(x)}$, уточняем на каждом шаге.

i	c_{i-1}	$\nu_i(a_j)$								
		0	1	2	3	4	5	6	7	$\sum \nu_i(x)$
1	–	1	1	1	1	1	1	1	1	$8 = \|A\|$
2	7	1	1	1	1	1	1	1	2	9
3	4	1	1	1	1	2	1	1	2	10
4	3	1	1	1	2	2	1	1	2	11
5	1	1	2	1	2	2	1	1	2	12
6	6	1	2	1	2	2	1	2	2	13
7	5	1	2	1	2	2	2	2	2	14
8	0	2	2	1	2	2	2	2	2	15
9	0	3	2	1	2	2	2	2	2	16
10	4	3	2	1	2	3	2	2	2	17
11	4	3	2	1	2	4	2	2	2	18
12	4	3	2	1	2	5	2	2	2	19
13	4	3	2	1	2	6	2	2	2	$20 = \|A\| + n - 1$

$$p_S(C) = \frac{1}{8} \cdot \frac{1}{9} \cdot \frac{1}{10} \cdot \frac{1}{11} \cdot \frac{1}{12} \cdot \frac{1}{13} \cdot \frac{1}{14} \cdot \frac{2}{15} \cdot \frac{2}{16} \cdot \frac{3}{17} \cdot \frac{4}{17} \cdot \frac{5}{19} \cdot \frac{6}{20}$$

$$I_S(C)\ [\text{бит}] = -\log_2 \left(p_S(C)\right) \approx 38{,}3\ \text{бит} \approx 12{,}8\ \text{триады}.$$

Оптимальный разделимый код S ($|code(C)| \to I_S(C)$): **поточный** код Хаффмана, где статистика уточняется и дерево кодов перестраивается на каждом шаге.

Такой код может быть длиннее $|C|$,

часто длиннее блочного Хаффмана,
не нашлось информации, может ли быть короче;

не требует хранить массив частот
(часто короче блочного+частот).

Альтернативные поточные (блочно-поточные) коды

Дерево перестраивается раз в N шагов (поблочно), • блок 1 $(1 \dots N)$ всегда без сжатия.

1. Статистика накапливается непрерывно:
   - блок 2 $((N+1)\dots 2N)$ — Хаффман по частотам блока 1;
   - блок 3 $((2N+1)\dots 3N)$ — Хаффман по частотам блоков 1+2;
   - блок 4 $((3N+1)\dots 4N)$ — Хаффман по частотам блоков 1+2+3...

   быстрее, но хуже кода модели S.

2. Статистика сбрасывается после каждого блока (частоты $\to$ $1111\dots1111$):
   - блок 2 по частотам блока 1;
   - блок 3 по частотам блока 2;
   - блок 4 по частотам блока 3;
   - блок 5 по частотам блока 4...

3. Статистика сохраняется для двух блоков (N может быть меньше, чем в 2):
   - блок 2 по частотам блока 1 или без сжатия;
   - блок 3 по частотам блоков 1+2;
   - блок 4 по частотам блоков 2+3;
   - блок 5 по частотам блоков 3+4...

и т. д. Для файла из неоднородных фрагментов 2 и 3 иногда лучше кода модели S.

Каждая реализация — своя модель.

Вопросы и задачи к семинару 1 (введение в коды без контекста и модель БП)

1. Символ [первичного алфавита]=2-битный байт ($A = \{0, 1, 2, 3\}$), вторичный алфавит — биты, рассматриваем блочный код без контекста $\{0, 1, 2, 3\}^+ \to \{0, 1\}^+$ на примере сообщения
2. пусть задан код K:

Вопросы и задачи к семинару 2 (энтропийные коды)

1. Символ=k-битный байт. Найдите для оптимальных разделимых кодов минимальную и максимальную длины кода символа.
2. Определите, при каких сочетаниях порядка при равн. част. и маркировки ветвей для $\vec{\nu} = 1111\ldots1111$ и метода Хаффмана $\forall a$: $code(a) = a$.
3. Символ=байт=триада. Дано сообщение $C = 6707\,4444\,4411\,55$:
   - найдите $I_{\text{БП}}(C)$;
   - закодируйте C методами: Хаффмана, Шеннона—Фано, Шеннона без учёта контекста (укажите выбранные: порядок при равн. част., маркировку ветвей, правило деления);
   - сравните длины кодов друг с другом и с $I_{\text{БП}}(C)$.
   - найдите $I_{\text{М1}}(C)$;
   - закодируйте C методом Хаффмана с учётом предыстории в 1 символ (оптимальным разделимым кодом М1), порядок при равн. част. и маркировка ветвей — те же;
   - сравните длину кода с $I_{\text{М1}}(C)$.

Спасибо за внимание!

МИЭТ
www.miet.ru

Александра Игоревна Кононова

ОТИК
https://gitlab.com/illinc/otik