Сжатие с учётом контекста. Словарные методы с отдельным словарём (дерево/таблица) — семейство кодов Зива—Лемпеля, основанных на идее 1978 г., LZ78

Александра Игоревна Кононова

МИЭТ

17 ноября 2024 г. — актуальную версию можно найти на `https://gitlab.com/illinc/otik`

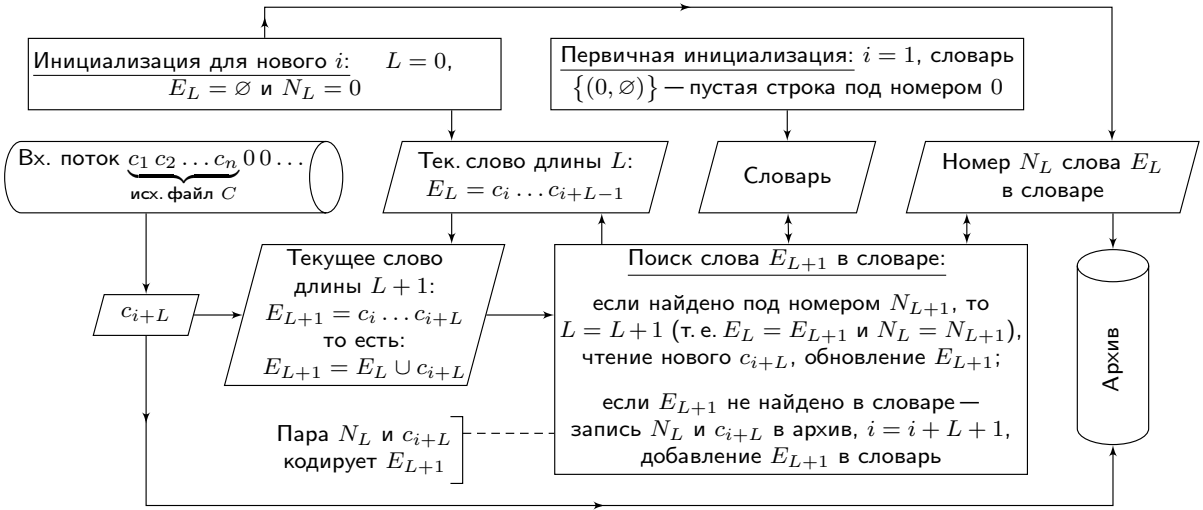
Алфавит и сообщение

Символ=байт (для доски байт = триада = 3 бита):

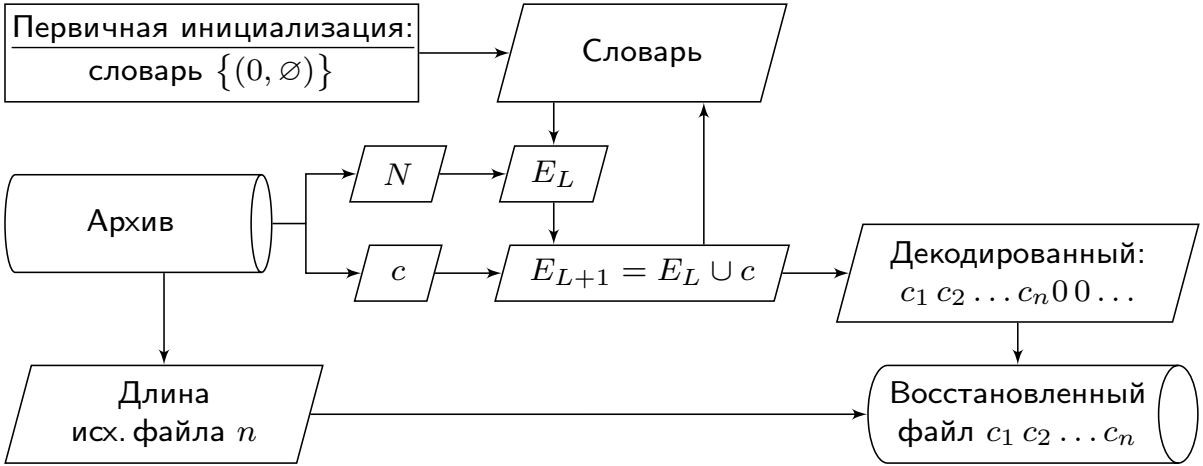
значение байта	0	1	2	3	4	5	6	7
глиф	б	н	о	п	р	с	т	ь

Сообщение $C =$ «обороноспособность» ($n = 18$ символов всего, 8 разных).

LZ78/LZ2: концепт Зива–Лемпеля 1978 г. — схема данных кодирования



LZ78/LZ2: концепт Зива–Лемпеля 1978 г. — схема данных декодирования



LZ78/LZ2: концепт Зива–Лемпеля 1978 г.

1978 г., Якоб Зив (Jacob Ziv) и Абрахам Лемпель (Abraham Lempel):

- словарь = дерево, каждый узел имеет номер N и символ $c \in A$: узел = пара (N, c);
- корень — пара $(0,$ пустая строка$)$ — имеет номер 0, но не имеет символа;
- слово E с номером N читается от корня вниз до узла (N, c) — символ c последний в E.

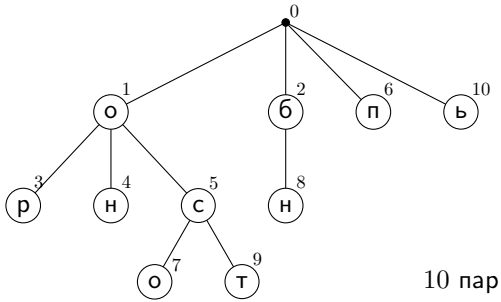
1. Вначале словарь пуст (в словаре $m = 1$ пустое слово с номером 0), позиция в файле $i = 1$.
2. На каждом шаге m (в словаре m слов), текущая позиция в файле i:
   1. разыскивается такая длина текущего слова L, что:
      - $E_L = c_i \dots c_{i+L-1}$ в словаре уже есть — узел (N_L, c_{i+L-1}), $0 \leqslant N_L \leqslant m-1$;
      - $E_{L+1} = c_i \dots c_{i+L-1}\, c_{i+L}$ ещё нет;
   2. в словарь добавляется новый лист (m, c_{i+L}), его родитель имеет номер $P = N_L$;
   3. в выходной поток записываются номер родителя P и символ нового листа c_{i+L} — для компактности далее (P, c_{i+L}) (это не узел!).
3. При необходимости входной поток дополняется (либо конец обрабатывается особо).

Вместо возврата назад по файлу — быстрый поиск в дереве $\Longrightarrow$ высокая скорость кодирования.

LZ78/LZ2: концепт Зива–Лемпеля 1978 г. — «обороноспособность» (18, 8 разных)

1. Первичная инициализация: словарь = корень (пустая строка), $m = 1$ (№ добавляемого узла, с 1).
2. Инициализация для нового i: $P = 0$ (текущий узел — корень), c (текущий символ входного потока).
3. Если у P есть дочерний (N, c), меняем текущий узел ($P = N$), читаем новый текущий символ c, к 3.
4. Если c нет в дочерних узлах P:
   - добавляем P дочерний (m, c), i увеличиваем на длину слова, $++m$, в выходной поток пишем (P, c);
   - читаем новый текущий символ c, к 2.

1	(0,о)	о
2	(0,б)	б
3	(1,р)	ор
4	(1,н)	он
5	(1,с)	ос
6	(0,п)	п
7	(5,о)	осо
8	(2,н)	бн
9	(5,т)	ост
10	(0,ь)	ь



10 пар

Как записать пару (P, c) на шаге m?

В каждой паре (P, c):

- разрядность $|c|$ символа c постоянна и равна разрядности k байта в исходном тексте;
- разрядность $|P|$ номера P узла-родителя в общем случае не равна k.

В памяти ЭВМ P типа $long\ long$ (его разрядность $|P| = 64$ бита) и не переполняется.

В выходном потоке $|P|$ может:

1. меняться — расти с ростом номера m шага: **минимальная длина** кода достигается при **побитовом увеличении** $|P|$:
   - тогда поток пар (P, c) — битовый, а не байтовый;
   - далее в лекции — побитовое увеличение $|P|$ и рассчитывается минимальная длина кода;

2. быть постоянной, $|P| \gg |c|$ (при $|P| = |c|$ всегда хуже кода фиксированной ширины, сжатия нет):

   а) при $m < 2^{|P|-1}$ длина больше минимальной 1; **б)** при $2^{|P|-1} \leqslant m < 2^{|P|}$ длина равна 1;

   в) на шаге $m = 2^{|P|}$ номер P узла-родителя переполняется, при этом дерево:
   - либо уничтожается и растится заново с нуля (следующее m после $(2^{|P|} - 1)$ — не $2^{|P|}$, а 1);
   - либо ветви уничтожаются выборочно (m уменьшается, но не до 1);
   - либо фиксируется и не растёт: **код меняется;** сбои, если на первом уровне не весь алфавит.

Концепт 1978 г., «обороноспособность», минимальная общая длина кода

В коде сообщения $m_{\max} = 10$ пар (P, c), то есть:

- 10 символов c ($|c| = k$ бит $= 1$ байт, суммарная длина $|c|_{\sum} = m_{\max} = 10$ байтов);
- и 10 номеров родительских узлов P, причём для каждого шага m разрядность $P \in \{0, 1, \dots m-1\}$ выбирается минимально возможной (тогда общая длина кода $|code|$ — минимальна):

m	Возможные P	$\min(\lvert P\rvert)$, бит
1	только 0	0 (не сохр.)
2	0 или 1	1
3	0, 1, 2	2
4	0, 1, 2, 3	2
5	0, 1, ... 4	3
6	0, 1, ... 5	3
7	0, 1, ... 6	3
8	0, 1, ... 7	3
9	0, 1, ... 8	4
10	0, 1, ... 9	4

Суммарная длина (в битах) полей P во всех 10 парах:
$|P|_{\sum} = 1 + 2 \cdot 2 + 3 \cdot 4 + 4 \cdot 2 = 25$ бит.
Зависит только от $m_{\max}$, но не от разрядности k байта.

При $m_{\max} = 2^Q + R$, $0 \leqslant R < 2^Q$: $|P|_{\sum} = \sum_{i=0}^{Q} i 2^{i-1} + (Q+1)R$ бит.

Общая длина кода 10 пар (в трёхбитных байтах):
$|code| = |c|_{\sum}$ [байтов] $+ \frac{|P|_{\sum}\text{ [бит]}}{k} = 10 + \frac{25}{3} = 18\frac{1}{3} \cong 19$ байтов.

Исходная длина — 18 символов=байтов: увеличение размера.
Задача: закодировать «обороноспособностьобороноспособность».
Будет ли сжатие?

Концепт 1978 г., минимальная длина кода P

Общая длина полей P в $m_{\max} = 2^Q + R$ парах $(0 \leqslant R < 2^Q)$ в коде концепта 1978 г.:

$$|P|_{\sum}(2^Q + R) = \sum_{i=0}^{Q} i \cdot 2^{i-1} + (Q+1)R \text{ бит}$$

— не зависит от разрядности k байта.

При $R = 0$ (то есть $m_{\max} = 2^Q$):

$$|P|_{\sum}(2^Q) = \sum_{i=0}^{Q} i \cdot 2^{i-1} \text{ бит.}$$

Слагаемые $i \cdot 2^{i-1}$: 0, 1, 4, 12, 32, 80, 192, 448, 1024, ... — последовательность OEIS A001787,

$\sum_{i=0}^{Q} i \cdot 2^{i-1} = (Q-1) \cdot 2^Q + 1$: 0, 1, 5, 17, 49, 129, 321, 769, 1793, ... — OEIS A000337.

$m_{\max}$	1	$2+R$	$4+R$	$8+R$	$16+R$	$32+R$	$64+R$	$128+R$	$256+R$	$512+R$
$\|P\|_{\sum}$, бит	0	$1+2R$	$5+3R$	$17+4R$	$49+5R$	$129+6R$	$321+7R$	$769+8R$	$1793+9R$	$4097+10R$

$m_{\max}$	1	2	$4-R$	$8-R$	$16-R$	$32-R$	$64-R$	$128-R$	$256-R$	$512-R$
$\|P\|_{\sum}$, бит	0	1	$5-2R$	$17-3R$	$49-4R$	$129-5R$	$321-6R$	$769-7R$	$1793-8R$	$4097-9R$

LZ78/LZ2: код Зива–Лемпеля–Велча, LZW

1984 г., Терри Велч (Terry Welch) по концепции LZ78:

1. Вначале словарь = первый уровень (все одиночные символы, 2^k штук для k-битного символа=байта). Тогда корень можно не нумеровать (прикорневые нумеруем с нуля).
2. Кодирование: при добавлении узлу P дочернего узла (m, c):
   - **оставляем c во входном потоке** (c — последний символ текущего слова и первый следующего);
   - в выходной поток пишем только P.
3. Декодирование: прочитан номер P — родительский для $(m, ?)$:
   1. символ c, соответствующий первому (прикорневому $\Longrightarrow$ всегда известному) узлу ветви P:
      - пишем в выходной поток;
      - пишем в узел $(m-1, ?)$ (первый символ слова = последний предыдущего), теперь это $(m-1, c)$;
   2. [теперь все узлы известны] прочие символы ветви P, включая сам P — только в выходной поток;
   3. добавляем узлу P дочерний $(m, ?)$ с номером m и пока неизвестным c, $\quad ++m$.

   Первый шаг (номер $m = 2^k$, $P \in \{0, 1, ..., 2^k - 1\}$ кодирует первый символ файла c_1) **необходимо обработать отдельно:** пред. слова нет; узел $(m-1, 2^k - 1)$ менять нельзя, в общем случае $c_1 \neq 2^k - 1$.
4. Дерево часто разворачивается в таблицу.
5. Входной поток **всегда** дополняется как минимум одним незначащим символом.

Схема данных кодирования LZW (семейство LZ78), для октетов

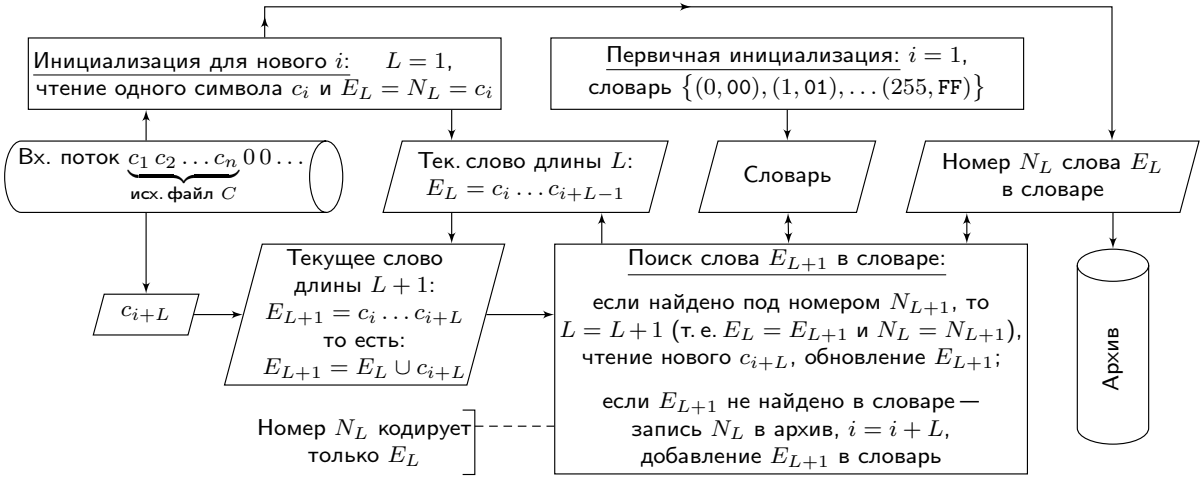
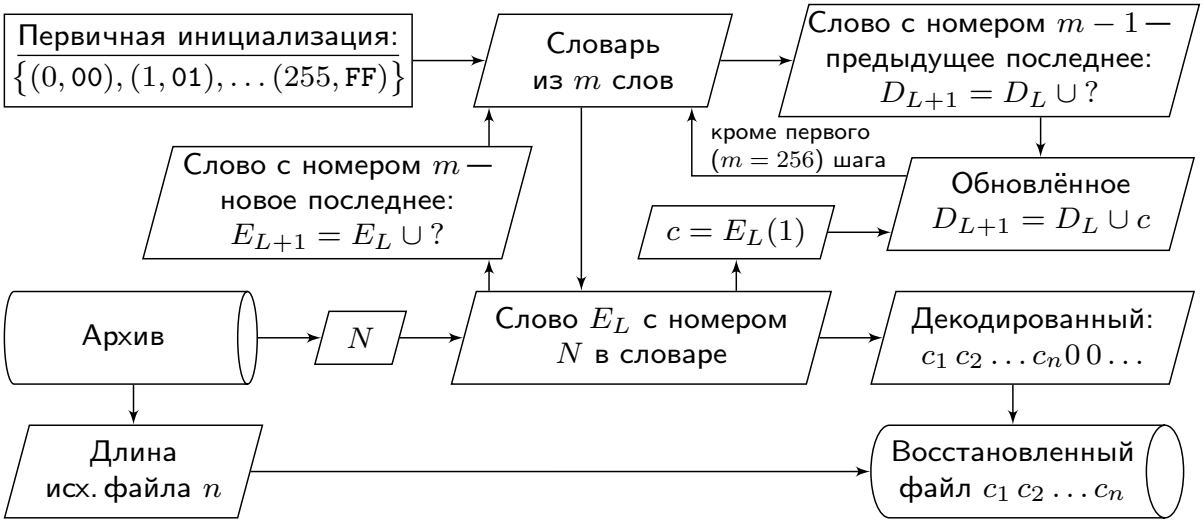
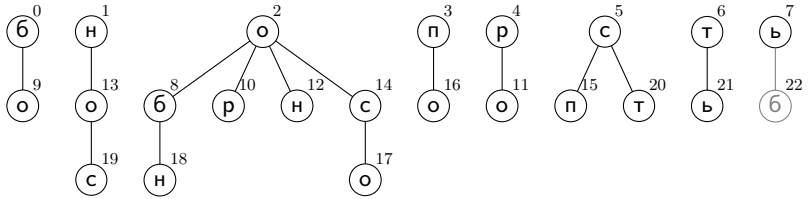


Схема данных декодирования LZW (семейство LZ78), для октетов



LZ78/LZ2: код Зива–Лемпеля–Велча, LZW— «обороноспособность» (18, алфавит из 8)



8	(2, б)	об	2
9	(0, о)	бо	0
10	(2, р)	ор	2
11	(4, о)	ро	4
12	(2, н)	он	2
13	(1, о)	но	1
14	(2, с)	ос	2
15	(5, п)	сп	5

16	(3, о)	по	3
17	(14, о)	осо	14
18	(8, н)	обн	8
19	(13, с)	нос	13
20	(5, т)	ст	5
21	(6, ь)	ть	6
22	(7, б)	ьб	7

15 значений

LZW, «обороноспособность», минимальная длина кода

В коде сообщения $m_{\max} - m_{\min} + 1 = 22 - 8 + 1 = 15$ значений P:

m	Возможные P	$\min(\lvert P\rvert)$, бит
8	0, 1, ... 7	3
9	0, 1, ... 8	4
10	0, 1, ... 9	4
11	0, 1, ... 10	4
12	0, 1, ... 11	4
13	0, 1, ... 12	4
14	0, 1, ... 13	4
15	0, 1, ... 14	4
16	0, 1, ... 15	4
17	0, 1, ... 16	5
18	0, 1, ... 17	5
19	0, 1, ... 18	5
20	0, 1, ... 19	5
21	0, 1, ... 20	5
22	0, 1, ... 21	5

Общая длина кода (в битах): $|P|_{\sum} = 3 + 4 \cdot 8 + 5 \cdot 6 = 65$ бит.

Общая длина кода (в трёхбитных байтах):
$|code| = |P|_{\sum} = \frac{65}{3} = 21\frac{2}{3} \cong 22$ символа=байта.

Исходная длина — 18 символов=байтов: увеличение размера.

Задача: закодировать:
а) «обороноспособностьобороноспособность»,
б) «обороноспособностьобороноспособностьобороноспособность».
Будет ли сжатие?

LZ78 (в том числе LZW) — для малоцветных **изображений.**

LZW, минимальная длина кода

При оптимальном кодировании P LZW: $|P|_{\sum}^{\text{LZW}}(m_{\max}) = |P|_{\sum}^{\text{к1978}}(m_{\max}) - |P|_{\sum}^{\text{к1978}}(m_{\min} - 1)$ бит,

где $m_{\min} = |A| = 2^k$ постоянно: $|P|_{\sum}^{\text{LZW}}(m_{\max}) = |P|_{\sum}^{\text{к1978}}(m_{\max}) - (k-1) \cdot 2^k - 1 + k$ бит,

в частности, $|P|_{\sum}^{\text{LZW}}(2^Q) = (Q-1) \cdot 2^Q - (k-1) \cdot 2^k + k$ бит.

Для доски $k = 3 \implies m_{\min} = 8 \implies |P|_{\sum}^{\text{LZW}}(2^Q) = (Q-1) \cdot 2^Q - 2 \cdot 8 + 3 = (Q-1) \cdot 2^Q - 13$

$m_{\max}$	$8+R$	$16+R$	$32+R$	$64+R$	$128+R$
$\|P\|_{\sum}$, бит	$3+4R$	$35+5R$	$115+6R$	$307+7R$	$755+8R$
$\|P\|_{\sum}$, триад	$1+R+\frac{R}{3}$	$11+\frac{2}{3}+R+\frac{2R}{3}$	$38+\frac{1}{3}+2R$	$102+\frac{1}{3}+2R+\frac{R}{3}$	$251+\frac{2}{3}+2R+\frac{2R}{3}$

Для x86/amd64 $k = 8 \implies m_{\min} = 256 \implies |P|_{\sum}^{\text{LZW}}(2^Q) = (Q-1) \cdot 2^Q - 1784$

$m_{\max}$	$256+R$	$512+R$	$1024+R$	$2048+R$
$\|P\|_{\sum}$, бит	$8+9R$	$2312+10R$	$7432+11R$	$18696+12R$
$\|P\|_{\sum}$, октетов	$1+R+\frac{R}{8}$	$289+R+\frac{2R}{8}$	$939+R+\frac{3R}{8}$	$2337+R+\frac{4R}{8}$

Спасибо за внимание!

МИЭТ
www.miet.ru

Александра Игоревна Кононова
illinc@mail.ru
https://gitlab.com/illinc/raspisanie