

Сжатие с учётом контекста. Словарные методы с отдельным словарём (дерево/таблица) — семейство LZ78

Александра Игоревна Кононова

МИЭТ

24 ноября 2023 г. — актуальную версию можно найти на
<https://gitlab.com/illinc/otik>

Алфавит и сообщение

В норме для кодов семейства LZ78, как и для любого кода, алфавит — набор байтов, исходный текст — последовательность байтов.

Рассмотрим сообщение «Обороноспособность» (18 символов всего, 8 разных) в 8-символьном алфавите из 3-битных байтов (в сообщении встречаются все 8 возможных символов):

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

Код Зива–Лемпеля, LZ78/LZ2 (концепция)

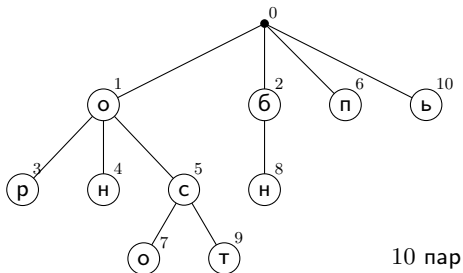
1978 г., Якоб Зив (Jacob Ziv) и Абрахам Лемпель (Abraham Lempel):

- 1 Скользящее окно не используем — кодируем в один проход вперёд \Rightarrow высокая скорость кодирования-декодирования.
- 2 Словарь = дерево, узел — номер и символ (n, c) , корень — (0, пустая строка), слово читается от корня.
- 3 Вначале словарь пуст (только корень).
- 4 На каждом шаге
 - к словарю добавляется узел (лист);
 - в выходной поток — номер родителя и символ нового листа (P, c) .
- 5 Когда кончается ёмкость номера листа, дерево:
 - либо уничтожается и раститесь заново;
 - либо ветви уничтожаются выборочно (сложно);
 - либо фиксируется и не растёт (нет прикорневого узла \Rightarrow сбой);
 - либо **увеличивается разрядность номера**.
- 6 При необходимости вх-й поток дополняется (либо конец обр-ся особо).

«Обороноспособность» (18, 8 разных)

- 1 Вначале словарь = корень (пустая строка),
 $n = 1$ (\mathbb{N} добавляемого узла, с 1), $i = 0$ (\mathbb{N} символа во вх. потоке, с 0).
- 2 $P = 0$ (текущий узел — корень), c_i (текущий символ входного потока);
- 3 Если c_i — дочерний P , $P = c_i$ и читаем c_{i+1} ($++i$)
- 4 Если c_i нет в дочерних узлах P :
 - добавляем P дочерний узел (n, c_i) , $++n$ и читаем c_{i+1} ($++i$);
 - в выходной поток пишем (P, c_i) .

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



Код Зива–Лемпеля, LZ78/LZ2 (концепция) — Минимально возможная длина кода

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

- разрядность $|c|$ символа постоянна и равна разрядности символа в исходном тексте;
- разрядность $|P|$ номера узла-родителя не равна $|c|$:
 - может быть постоянной: $|P| \geq \log_2(n_{\max} - 1)$, здесь $|P| \geq 4$ бита, обычно $|P| \gg |c|$;
 - может быть разной для разных пар: **минимальная длина** кода достигается при побитовом увеличении $|P|$ (тогда поток пар (P, c) — битовый, а не байтовый).

Рассчитаем эту **минимальную длину** $|code|$.

№	Возможные P	$\min(P)$, бит	
1	только 0	0 (не сохр.)	Суммарная длина (в символах) полей c во всех 10 парах постоянна и равна 10 символов (байтов).
2	0 или 1	1	
3	0, 1, 2	2	Минимальная суммарная длина (в битах) полей P во всех 10 парах:
4	0, 1, 2, 3	2	$ P _{\Sigma} = 1 + 2 \cdot 2 + 3 \cdot 4 + 4 \cdot 2 = 25$ бит.
5	0, 1, ... 4	3	
6	0, 1, ... 5	3	Минимальная общая длина $ code $ (в символах=байтах) кода при
7	0, 1, ... 6	3	$ c = 3$ (3-битный байт): $ code = 10 + \frac{25}{3} = 10 + 8 + \frac{1}{3} \cong 19$;
8	0, 1, ... 7	3	(для записи «лишнего» бита необходим целый байт).
9	0, 1, ... 8	4	
10	0, 1, ... 9	4	Длина исходного текста — 18 символов=байтов.

Код Зива–Лемпеля–Велча, LZW

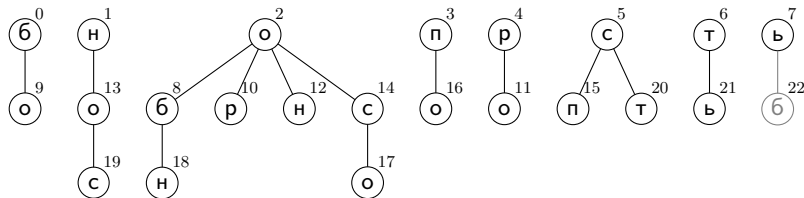
1984 г., Терри Велч (Terry Welch) по концепции LZ78:

- 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 с номером m и символом c_i :
 - оставляем c_i во входном потоке (c_i — последний текущего слова и первый следующего);
 - в выходной поток пишем P .
- ③ Декодирование: перед любым шагом, кроме первого, последний узел (с номером $m - 1$) — пустой (с неизвестным символом c_j); читаем $P \in \{0, 1, \dots, m - 1\} \Rightarrow$ пишем в выходной поток слово = ветвь P и одновременно заполняем пустой узел:
 - символ c_j , соответствующий первому (прикорневому \Rightarrow всегда заполненному) узлу ветви P пишем и в выходной поток, и в пустой узел $m - 1$ (первый слова = последний предыдущего);
 - [теперь нет пустых узлов] прочие символы ветви P , включая сам P — только в выходной поток;
 - добавляем узлу P пустой (с неизвестным c_i) дочерний с номером m .

Первый шаг ($m = 2^k$) отдельно: $P \in \{0, 1, \dots, 2^k - 1\}$, пустого нет.

- ④ $|m| \gg |c|$, во многих реализациях увеличивается по битам.
- ⑤ Дерево часто разворачивается в таблицу.
- ⑥ Вх-й поток всегда дополняется как минимум одним незначащим символом.

«Оборонеспособность» (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			

Код Зива–Лемпеля–Велча, LZW — Минимально возможная длина кода

В коде сообщения $m_{\max} - m_{\min} + 1 = 22 - 8 + 1 = 15$ значений P ;
рассчитаем **минимально возможную** их длину:

- 1 размером 3 бита ($m = 8: P \in \{0, 1, \dots, 7\}$);
- 8 размером 4 бита (от $m = 9: P \in \{0, 1, \dots, 7, 8\}$ до $m = 16: P \in \{0, 1, \dots, 15\}$);
- 6 размером 5 бит (от $m = 17: P \in \{0, 1, \dots, 16\}$ до $m = 22: P \in \{0, 1, \dots, 22\}$);
потенциально 5 бит хватило бы на 16 значений (до $m = 32: P \in \{0, 1, \dots, 31\}$),
но сообщение закончилось раньше.

Суммарная длина (в битах) $3 + 4 \cdot 8 + 5 \cdot 6 = 65$ бит.

Символ=байт при таком дереве может занимать только 3 бита (алфавит из 8 символов).

Суммарная длина (в символах) $\frac{65}{3} = 21\frac{2}{3} \cong 22$ байта.

Декодирование (обороноспособность)

$\textcircled{6}^0$ $\textcircled{н}^1$ $\textcircled{о}^2$ $\textcircled{п}^3$ $\textcircled{р}^4$ $\textcircled{с}^5$ $\textcircled{т}^6$ $\textcircled{ь}^7$

2 о
 0 6 8 (2, 6)
 2 о 9 (0, о)
 4 р 10 (2, р)
 2 о 11 (4, о)
 1 н 12 (2, н)
 2 о 13 (0, о)
 5 с 14 (2, с)

3 п 15 (5, с)
 14 ос 16 (3, о)
 8 об 17 (14, о)
 13 но 18 (8, н)
 5 с 19 (13, с)
 6 т 20 (5, т)
 7 ь 21 (6, ь)

Задачи для лекции

Закодируйте/декодируйте различными алгоритмами семейства LZ78 (концептом и LZW) сообщение

$m = 7770\ 0000\ 0123\ 4567\ 7770\ 7770\ 0000\ 0000$

Сравните длину кода с кодом семейства LZ77 с односимвольным префиксом.

Спасибо за внимание!

МИЭТ

www.miet.ru

Александра Игоревна Кононова

illinc@mail.ru

gitlab.com/illinc/raspisanie