Алгоритм LZ-78 ≈ LZW

5 января 2023 г. 1:25

На первом шаге кодер формирует словарь из esc-символа, который помещается в ячейку с номером i=0, объем словаря становится = 1.

Считаем, что символы $x_1, \ldots, x_{n-1}$ уже закодированы. Кодер смотрит на следующие символы: начиная с x_n и дальше, ищет самое длинное слово в словаре, совпадающее с началом последовательности, которую надо закодировать. То есть он не бежит по всей предыдущей строке и ищет, как LZ-77, а ищет именно по элементам словаря, находит самый длинный совпадающий.

- Если ничего не нашлось, декодеру передается ссылка на esc-символ (собственно, для этого мы его и завели). Каким образом:
  a. В равномерном коде передаем номер ячейки словаря, хранящей esc-символ: $\lceil log(C-1) \rceil$. Длина этого номера зависит от количества символов в словаре.
  b. И следом передаем ASCII-код нового встретившегося символа (8 бит).
- Если нашлось, то:
  a. декодеру передается индекс ячейки словаря, в которой хранится этот символ $\lceil log(C-1) \rceil$ (т.е. bin(i), где i - номер найденного слова).
  b. словарь дополняется новым словом, которое получается дописыванием к слову с номером i следующей буквы, объем словаря увеличивается на 1.
  c. декодер сможет добавить эту следующую букву только на следующем шаге. Поэтому кодер не использует последнее слово словаря (т.к. "следующая буква" еще даже не была продекодирована декодером, а в словарь в кодере мы его уже занесли, надо запретить кодеру использовать последнее слова словаря).

Пример

Шаг	Словарь	i	Код	Биты
0	esc	–	–	–
1	I	0	bin(I)	0+8=8
2	F	0	bin(F)	log(1)+8=8
3	_	0	0bin(_)	$\lceil \log(2) \rceil + 8 = 9$
4	W	0	00bin(W)	$\lceil \log(3) \rceil + 8 = 10$
5	E	0	00bin(E)	$\lceil \log(4) \rceil + 8 = 10$
6	_C	3	011	3
7	C	0	000bin(C)	3+8=11
8	A	0	000bin(A)	3+8=11
9	N	0	000bin(N)	3+8=11
10	NO	0	000bin(N)	4+8=12
11	O	0	0000bin(O)	4+8=12
12	T	0	0000bin(T)	4+8=12
13	_D	3	0011	4
14	D	0	0000bin(D)	4+8=12
15	O_	11	1011	4

Шаг	Словарь	i	Код	Биты
16	_A	3	0011	4
17	AS	8	1000	4
18	S	0	00000bin(S)	5+8
19	_W	3	00011	5
20	WE	4	00100	5
21	E_	5	00101	5
22	_WO	19	10011	5
23	OU	11	01011	5
24	U	0	00000bin(U)	5+8=13
25	L	0	00000bin(L)	5+8=13
26	D_	14	01110	5
27	_WE	19	10011	5
28	E_S	21	10101	5
29	SH	18	10010	5

Шаг	Словарь	i	Код	Биты
30	H	0	00000bin(H)	5+8=13
31	OUL	23	10111	5
32	LD	25	11001	5
33	D_D	26	11010	5
4	DO	14	001110	6
35	O_A	15	001111	6
36	AS_	17	010001	6
37	_WE_	27	011011	6
38	_CA	6	000110	6
39	AN	8	001000	6
40	N	9	001001	6
Всего бит:				291

Избыточность LZW с увеличением C определяется как: чем больше словарь, тем меньше избыточность

$$\bar{R} \approx H_\infty(X) + \frac{\log\log C}{\log C}$$