

LZ 77, LZ 77 bootstrap, LZ 78 кодовање и аритметичко кодовање

1. Кодовати секвенцу LZ77 алгоритмом и израчунати степен компресије ако је величина прозора $w=8$.

$$\begin{aligned} (0, \text{char}) &\rightarrow a - \text{дубина match-а} \\ (1, a, b) &\rightarrow b - \text{дужина match-а} \end{aligned} \quad \left\{ \begin{array}{l} \text{match-а} \\ \text{match-а} \end{array} \right\} \quad \boxed{b \leq a}$$

MANASTIR PASTIR PASTI! MAST! MAT! ...

[0, M] [0, A] [0, N] [1, 2, 1] [0, S] [0, T] [0, I] [0, R] [0, -] [0, P] [1, 7, 7]
[1, 7, 4] [1, 6, 1] [0, M] [1, 6, 3] [1, 5, 1] [1, 7, 1] [1, 6, 2] [1, 5, 1] [1, 4, 1]

- потребан број бита за стандардно ASCII кодовање: $32 \cdot 8 = 256b$

- потребан број бита у случају LZ77 кодовања: $10 \cdot 9 + 10 \cdot 7 = 90 + 70 = 160b$

$$\eta = \frac{160}{256} = 0,625$$

$$1 - 0,625 = 0,375 \Rightarrow 37,5\% \text{ износи степен компресије}$$

$$\begin{aligned} [0, \text{char}] &- 9 \text{ бита } (1+8) \\ [1, a, b] &- 7 \text{ бита } (\text{size}(a) = \text{size}(b) = \log_2 w) \end{aligned}$$

2. Кодовати следећу секвенцу LZ77 и LZ78 алгоритмом.

LZ77: / WED / WE / WEE / WEB.

[0, 1] [0, W] [0, E] [0, D] [1, 4, 3] [1, 3, 3] [1, 4, 4] [0, B]

$$\text{ASCII: } 15 \cdot 8 = 120b$$

$$\text{LZ77: } 5 \cdot 9 + 3 \cdot 7 = 45 + 21 = 66b$$

$$\eta = \frac{66}{120} = 0,55$$

$$1 - 0,55 = 0,45 \Rightarrow \text{степен компресије је } 45\%$$

LZ78:

/ WED / WE / WEE / WEB

адреса	сејмент	износ
1	/	(0, 1)
2	W	(0, W)
3	E	(0, E)
4	D	(0, D)
5	/W	(1, W)
6	E/	(3, 1)
7	WE	(2, E)
8	E/W	(6, W)
9	EB	(3, B)

LZW алгоритам (Lempel - Ziv - Welch)

- алгоритам за компресију без губитака у којем се кодна књига одређује адаптивно
- за конструкцију кодне књиге није потребно познавати статистику избора, већ, у току самог кодирања и декодирања, кодер и декодер на основу примљених података конструишу идентичну кодну књигу
- за разлику од Хафмановог кодирања код којем су кодови симбола променљиве дужине, LZW алгоритам придружује кодне речени фиксне дужине #изобила симбола променљиве дужине
- компресија се у овом случају постиже коришћењем мањег броја бита за кодирање низа симбола као уједине него сваког симбола појединачно
- пошто је LZW адаптиван алгоритам, нема потребе за претходном кодном табелом чиме се избегава увећање количине података који се преносе
- основна идеја LZW алгоритма је да се кодна табела у току кодирања проширује #изобила изборних симбола који ће при следећим обрађивањима бити кодирани једним кодом

* Нема је даи избор чи/а је листа симбола/ ш. алфавити $\{A, B, L, I\}$.
 Потредно је кодовати низ симбола ALIBABABAB LZW алгоритмом.

	код	симбол	излаз
	(1)	A	
	(2)	B	
	(3)	L	
	(4)	I	
A L I B A B A B A B	(5)	AL	(1)
A L I B A B A B A B	(6)	LI	(3)
A L I B A B A B A B	(7)	IB	(4)
A L I B A B A B A B	(8)	BA	(2)
A L I B A B A B A B	(9)	AB	(1)
A L I B A B A B A B	(10)	BAB	(8)
A L I B A B A B A B			(10)

на основу оригиналне кодне табеле излаз је: (1)(3)(4)(2)(1)(2)(1)(2)(1)(2)

LZW алгоритмом излаз је: (1)(3)(4)(2)(1)(8)(10)

7 кодних симбола

10 кодних симбола

$$\eta = \frac{7}{10} = 0,7$$

$$1 - 0,7 = \underline{\underline{0,3}}$$

\Rightarrow

30% степен компресије