Komprese dat
(Komprimace dat)

Př.: zakódovat slovo ARARAUNA

	$\mathcal{K}_1$	$\mathcal{K}_2$	četnost	
			absolutní	relativní
A	0 0	0	4	0,500
N	0 1	1 1 1	1	0,125
R	1 0	1 0	2	0,250
U	1 1	1 1 0	1	0,125

kód $\mathcal{K}_1$: 00 10 00 10 00 11 01 00 . . . **16 bitů**
kód $\mathcal{K}_2$: 0 10 0 10 0 110 111 0 . . . **14 bitů**

***prefixový kód*: žádné kódové slovo není prefixem** (začátkem) **jiného kódového slova**

Př.: $\mathcal{K}_1$ — blokový kód (viz též bezpečnostní kódy)
$\mathcal{K}_2$ — prefixový kód

postfixový kód: žádné kódové slovo není postfixem (koncovkou) **jiného kódového slova**

afixový kód: kód, který je prefixový i postfixový

komprese **(**též **komprimace): kódování** (přesněji: překódování) **— snížení počtu bitů**
dekódování: dekomprese (dekomprimace)

kompresní poměr $\quad \text{KP} = \dfrac{\text{nová délka}}{\text{původní délka}}$

Př.: předp.: $\mathcal{K}_1$: původní kód
$\mathcal{K}_2$: nový kód — komprimovaná data
KP = 14/16 = 87,5% (pro „ARARAUNA“)

zpráva = řetěz dílčích zpráv
např.: text = řetěz písmen a dalších znaků

zdrojová jednotka S_j — (původní) dílčí zpráva
kódová jednotka C_j — zakódovaná dílčí zpráva

informační entropie

Čím menší je pravděpodobnost p zprávy, tím je zpráva cennější — tím větší je její informační obsah E, tzn.:

$$\frac{1}{p} \text{ roste} \implies \text{roste } E$$

logaritmická míra: $\quad E = \log_2 \frac{1}{p} = -\log_2 p$

$$\boxed{E = -\log_2 p} \quad \textbf{(shannon)}$$

E . . . informační entropie, informační obsah, míra množství informace aj. (viz též bezpečnostní kódy)

stejně pravděpodobná $n$bitová čísla:
počet čísel: $N = 2^n$
pravděpodobnost: $p = \frac{1}{N} = 2^{-n}$
entropie: $\boxed{E = n}$ (shannon $\sim$ bit)
v $n$bitovém čísle může být n shannonů

---

p_i . . . pravděpodobnost $i$té z k možných zpráv

průměrná entropie $\quad \boxed{H = -\sum_{i=1}^{k} (p_i \cdot \log_2 p_i)}$

Př.: příkl. „ARARAUNA“;
předp.: pravděpodobnost = relativní četnost
entropie písmen A, N, R, U: $\underset{4/8}{1}, \underset{1/8}{3}, \underset{2/8}{2}, \underset{1/8}{3}$
průměrná entropie 1 písmene:
H = 0,5·1 + 0,125·3 + 0,25·2 + 0,125·3 = 1,75

p_j . . . pravděpodobnost zdrojové jednotky S_j
E_j . . . entropie zdrojové jednotky S_j
H_j . . . průměrná entropie $j$té dílčí zprávy
zpráva S_1,S_2, . . . , S_m:
pravděpodobnost $p = p_1 \cdot p_2 \cdot \cdots \cdot p_m$
entropie $E = -\log_2 p$, tzn.: $\quad \boxed{E = \sum_{j=1}^{m} E_j}$

průměrná entropie (analogicky): $\quad \boxed{H = \sum_{j=1}^{m} H_j}$

Př.: entropie zprávy ARARAUNA:
$E = 1+2+1+2+1+3+3+1 = 14$ (shannon)
průměrná entropie 8písmenového slova:
$H = 8 \cdot 1{,}75 = 14$ (shannon)

redundance
(nadbytečnost)

E ... **entropie zprávy**

L ... **délka zápisu = max. možná entropie**

redundance: $\boxed{R = L - E}$

p_i ... **pravděpodobnost $i$té z k možných zpráv**

E_i ... **entropie $i$té zprávy**

L_i ... **délka zápisu $i$té zprávy**

H ... **průměrná entropie**

průměrná redundance: $Q = \sum_{i=1}^{k} p_i \cdot (L_i - E_i)$

$$\boxed{Q = \sum_{i=1}^{k} p_i \cdot L_i - H}$$

Př.: příkl. „ARARAUNA“; **kód $\mathcal{K}_1$**
předp.: pravděpodobnost = relativní četnost
$R = Q = 16 - 14 = 2$ (shannon)

komprese dat: snížit redundanci na minimum!

Metody komprese (vybrané:)

- ***speciální***
  - ◇ **čísla**
    - ○ **Elias**
    - ○ **Fibonacci**
  - ◇ **obrazové informace (přírůstky)**
  - ◇ **databáze (jména, data narození apod.)**
  - ◇ ...
- ***obecné I.***
  - ◇ **statické — týž kód pro různá data**
    - ○ **Shannon – Fano**
    - ○ **Huffman**
    - ○ **aritmetické k.**
  - ◇ **hybridní = semiadaptivní — kód ∈ data**
  - ◇ **dynamické = adaptivní — týž algoritmus pro kompresi i dekompresi**
    - ○ **FGK** (Faller – Gallager – Knuth)
    - ○ **V** (Vitter)
- ***obecné II. — slovníkové metody***
  - ◇ **statické**
  - ◇ **hybridní = semiadaptivní**
  - ◇ **dynamické = adaptivní**
    - ○ **Lempel – Ziv** (LZ77, LZ78)
    - ○ **BSTW** (Bentley – Sleator – Tarjan – Wei)

Eliasův kód I.

kódování přirozených čísel (větších než 0)
předpoklad: menší čísla — častější výskyt
princip: n**bitový zápis čísla, začínající 1**
před tento zápis umístit $n-1$ nul
$\Longrightarrow$ prefixový kód

1	1
2	*0* 1 0
3	*0* 1 1
4	*0 0* 1 0 0
5	*0 0* 1 0 1
6	*0 0* 1 1 0
7	*0 0* 1 1 1
8	*0 0 0* 1 0 0 0
9	*0 0 0* 1 0 0 1
...	...

Eliasův kód II.

před $n$bitový zápis čísla umístit n v Eliasově kódu I.
z $n$bitového zápisu čísla vypustit první bit (jedničku)
Př.: $386_{10} = 1\,1000\,0010_2 \Rightarrow n = 9 \sim$ *000 1001*
výsledné kódování: *000 1001* **1000 0010**

Fibonacciho kód

kódování přirozených čísel (větších než 0)
předpoklad: menší čísla — častější výskyt

Fibonacciho čísla (řádu 2)**:** $F_{-1} = F_0 = 1$,
$F_i = F_{i-1} + F_{i-2}$ **pro** $i = 1, 2, 3, \ldots,$

$F_0, F_1, F_2, \ldots$ **použijeme jako „váhy“ zápisu čísel:**

i	...	5	4	3	2	1	0	-1
F_i	...	13	8	5	3	2	1	1
1							1	
2						1	0	
3					1	0	0	
4					1	0	1	
5				1	0	0	0	
6				1	0	0	1	
7				1	0	1	0	
8			1	0	0	0	0	
9			1	0	0	0	1	
...	...							

v zápisech nejsou 2 jedničky vedle sebe

Fibonacciho kód (řádu 2)**:**

- **obrátit pořadí bitů** (v předchozích zápisech)
- **přidat *1* na konec $\Longrightarrow$ prefixový kód**

	1	2	3	5	8	…
1	**1**	***1***				
2	**0**	**1**	***1***			
3	**0**	**0**	**1**	***1***		
4	**1**	**0**	**1**	***1***		
5	**0**	**0**	**0**	**1**	***1***	
6	**1**	**0**	**0**	**1**	***1***	
7	**0**	**1**	**0**	**1**	***1***	
8	**0**	**0**	**0**	**0**	**1**	***1***
9	**1**	**0**	**0**	**0**	**1**	***1***
...	...					

Shannon – Fanoův kód

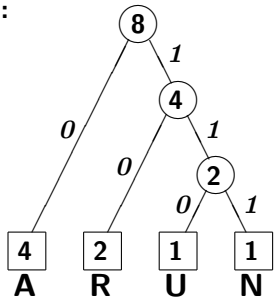
- **seřadit znaky podle četnosti (abs. nebo rel.)**
- **dělit podle četnosti na poloviny (pokud možno)**
- **0 ← → 1 (nebo naopak)**

Př.: příkl. „ARARAUNA“

A	R	U	N
4	2	1	1
1/2	1/4	1/8	1/8
0	1..		
	10	11.	
		110	111

A	**0**
R	**10**
U	**110**
N	**111**

tzv. *kódový strom*:

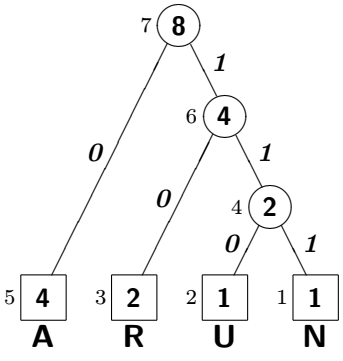


Huffmanův kód

***kódový strom*:**

- **seřadit znaky podle četnosti (abs. nebo rel.)**
- **znaky → uzly ohodnocené četnostmi**
- **nové uzly:**
  dvojice minimálně ohodnocených uzlů
  ohodnotit součtem ohodnocení

Př.: příkl. „ARARAUNA“



A	**0**
R	**10**
U	**110**
N	**111**

tzv. *sourozenecká vlastnost*

Kódy Shannon – Fanoův a Huffmanův

představitelé statických metod

- **velmi malá až minimální redundance**

Př.:

znak	četnost	S-F	H
A	**11**	**00**	**0**
B	**5**	**01**	**100**
C	**5**	**10**	**101**
D	**4**	**110**	**110**
E	**3**	**111**	**111**

28 znaků: Shannon – Fano 63 bitů
Huffman 62 bitů

semiadaptivní (hybridní) kódování

- **dva průchody:**
  1. určení četností dílčích zpráv → kód
  2. vlastní kódování
- **s daty je nutno přenést/zapsat i kódovací tabulku**

metoda FGK

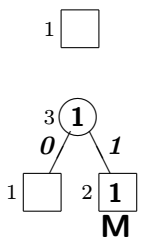
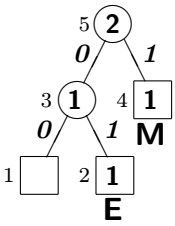
(Faller – Gallager – Knuth)

postupné vytváření a modifikace kódového stromu:

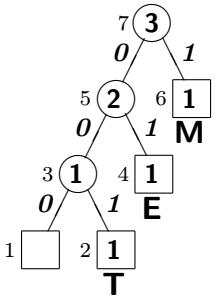
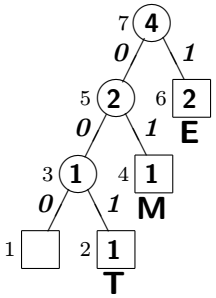
- **na počátku tvoří strom jen „zvláštní“ uzel — uzel (list) ohodnocený 0:** ***nulový list***
- **nový znak na vstupu:**
  - ◇ **vyšle se kód nulového listu (poprvé nic)**
  - ◇ **vyšle se vstupující znak** (např. v kódu ASCII)
  - ◇ **nulový list se nahradí binárním podstromem:**
    - ○ **kořen ohodnocen 1**
    - ○ **levý list: nulový list**
    - ○ **pravý list přísluší znaku — ohodnocen 1**
- **„známý“ znak na vstupu:**
  - ◇ **vyšle se kód znaku**
  - ◇ **pro list, který znaku přísluší, a postupně pro všechny jeho předchůdce** (směrem ke kořenu) **se zamění uzel s podstromem** (pokud existuje)**:**
    - ○ **jehož kořen je stejně ohodnocený uzel**
    - ○ **má vyšší číslo — uzly jsou číslovány zdola nahoru a zleva doprava**
  - ◇ **ohodnocení uzlu a všech jeho předchůdců se zvýší o 1**

metoda FGK — příklad I.

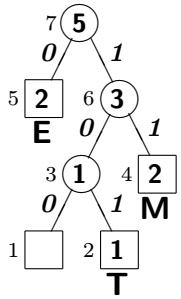
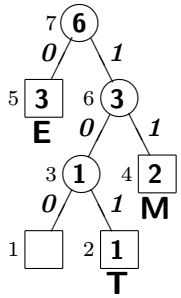
zakódovat řetěz znaků 'METEME_'

vstup	výstup	strom
'M'	'M'	
'E'	*0*'E'	
'T'	*00*'T'	

metoda FGK — příklad II.

vstup	výstup	strom
'T'	*00*'T'	
'E'	*01*	
'M'	*01*	

metoda FGK — příklad III.

vstup	výstup	strom
'M'	*01*	
'E'	*0*	
'_'	*100*'_'	

Slovníkové metody

- **orientovány na opakující se podřetězce, tzv. fráze, popř. digramy, trigramy, ...**
- **vytváří se „slovníky“ frází, ...**

metoda LZ78

- ◇ **položky ve slovníku číslovány od 1**
- ◇ **0 $\sim$ prázdný řetěz**
- ◇ **do slovníku se ukládá výstup $\sim$ podřetěz**
- ◇ **ve slovníku se hledá nejdelší podřetěz na vstupu**
- ◇ **výstup: číslo položky & následující znak**

Př.: zakódovat: ARA_ARARAUNA_ (13 znaků)

	vstup	výstup
1	A	0 A
2	R	0 R
3	A_	1 _
4	AR	1 R
5	ARA	4 A
6	U	0 U
7	N	0 N
8	A_	3