



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale  
2007-2013



# Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

## Transmisia datelor multimedia in rețele de calculatoare

### 2. Codificare

# Modelare si codificare

- Cerințele reconstrucției sunt cele ce impun dacă compresia este cu sau fără pierderi, schema exactă depinzând de un număr de factori
- Unii din cei mai importanți sunt impuși de caracteristicile datelor destinate compresiei, de exemplu, o tehnică de compresie poate fi eficientă pentru compresia unui text, dar total ineficientă pentru imagini
- Fiecare aplicație prezintă particularități specifice
- Dezvoltarea algoritmilor de compresie pentru o varietate de date cuprinde două faze:
  - prima fază se referă de obicei la modelare, când se încearcă extragerea informației despre orice redundanță din date și descrierea acesteia sub forma unui model
  - a doua fază este codarea.
- Diferența dintre date și model se numește secvență reziduală sau reziduu.

# Modelare si Codificare. Exemplul 1

- Se considera secventa:
  - $S_n = 9, 11, 11, 11, 14, 13, 15, 17, 16, 17, 20, 21$
  - Codificare binara necesita **5** bits/sample
- Se considera modelul:
  - $\hat{S}_n = n + 8$ :
  - 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
- Reziduu
  - $e_n = S_n - \hat{S}_n$ : 0, 1, 0, -1, 1, -1, 0, 1, -1, -1, 1, 1
  - Ex. codificare: '00'  $\Leftrightarrow$  -1, '01'  $\Leftrightarrow$  0, '10'  $\Leftrightarrow$  1
  - *2 bits/sample*
- Schema de compresie = model + reziduu
  - Obs. Modelul trebuie sa fie codificat de asemenea (de obicei in cadrul algoritmului)

# Modelare si Codificare. Exemplul 2

- Se considera secventa:
  - $S_n = 27, 28, 29, 30, 30, 32, 31, 31, 29, 28, 27$
- Se considera modelul:
  - $\hat{S}_1 = 0; \hat{S}_n = S_{n-1}$  for  $n > 1$
  - $0, 27, 28, 29, 30, 30, 32, 31, 31, 29, 28$
- Reziduul
  - $\{e_n\} = 27, 1, 1, 1, 0, 2, -1, 0, -2, -1, -1$
  - Poate fi codificat cu mult mai putini biti
- ***Codificare predictiva***
  - Folosirea datelor precedente pentru a calcula date viitoare.
  - Foarte folositor pentru exploatarea redundantei temporale
    - Ex. in audio si video

# Codificare in lumea reala: **Braille** (1821)

|     |      |       |        |         |          |           |            |             |              |               |              |               |
|-----|------|-------|--------|---------|----------|-----------|------------|-------------|--------------|---------------|--------------|---------------|
| •   | ••   | •••   | ••••   | •••••   | ••••••   | •••••••   | ••••••••   | •••••••••   | ••••••••••   | •••••••••••   | •••••••••••• | ••••••••••••• |
| a   | b    | c     | d      | e       | f        | g         | h          | i           | j            | k             | l            | m             |
| ••• | •••• | ••••• | •••••• | ••••••• | •••••••• | ••••••••• | •••••••••• | ••••••••••• | •••••••••••• | ••••••••••••• | ••••••••••~  | ••••••••••~•  |
| n   | o    | p     | q      | r       | s        | t         | u          | v           | w            | x             | y            | z             |

|        |       |        |        |         |          |           |            |             |             |              |               |                |
|--------|-------|--------|--------|---------|----------|-----------|------------|-------------|-------------|--------------|---------------|----------------|
| •      | ••    | •••    | ••••   | •••••   | ••••••   | ••••~     | •••••      | ••••••      | •••••••     | ••••••••     | •••••••••     | ••••••••••     |
| a      | but   | can    | do     | every   | from     | go        | have       | just        | knowledge   | like         | more          | not            |
| •••    | ••••  | •••••  | •••••• | ••••••• | •••••••• | ••••••••• | •••••••••• | ••••••••••• | ••••••••••~ | ••••••••••~• | ••••••••••~•• | ••••••••••~••• |
| people | quite | rather | so     | that    | us       | very      | will       | it          | you         | as           | and           | for            |



“Be nice to others”

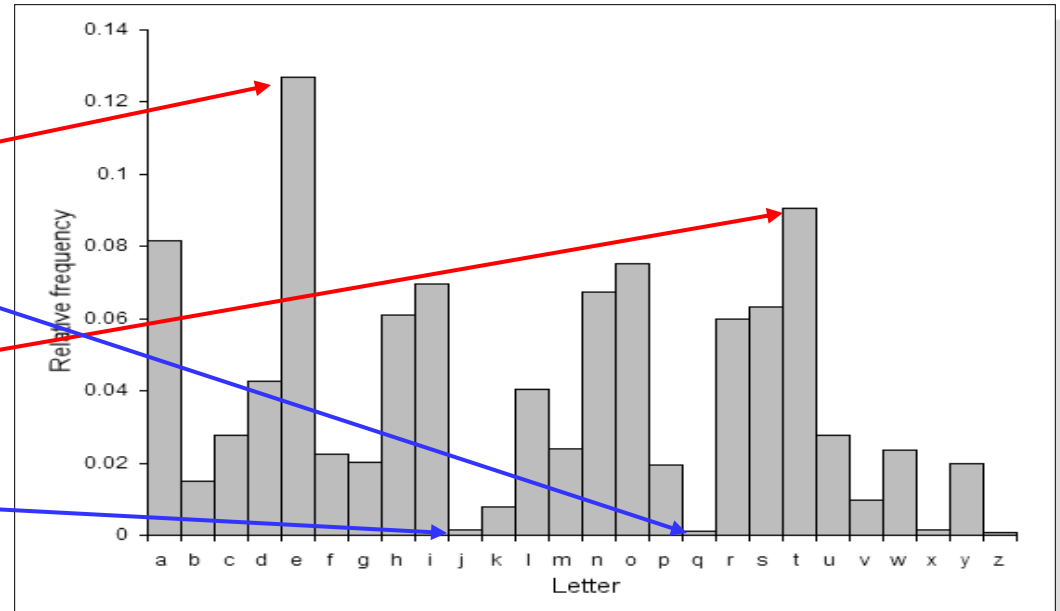
# Codificare in lumea reala: Morse (1844)

|          |       |          |       |           |        |          |         |
|----------|-------|----------|-------|-----------|--------|----------|---------|
| <b>A</b> | .-    | <b>M</b> | --    | <b>Y</b>  | -.--   | <b>6</b> | -....   |
| <b>B</b> | -...  | <b>N</b> | -. .  | <b>Z</b>  | --..   | <b>7</b> | ---...  |
| <b>C</b> | -.-.  | <b>O</b> | ---   | <b>Ä</b>  | .-.-   | <b>8</b> | ----..  |
| <b>D</b> | -..   | <b>P</b> | .-.-  | <b>Ö</b>  | ---.   | <b>9</b> | -----   |
| <b>E</b> | .     | <b>Q</b> | ---.- | <b>Ü</b>  | ..--   | .        | .-.-.-  |
| <b>F</b> | ..-.  | <b>R</b> | .-.   | <b>Ch</b> | ----   | ,        | --..--  |
| <b>G</b> | --.   | <b>S</b> | ...   | <b>0</b>  | -----  | ?        | ..-..   |
| <b>H</b> | ....  | <b>T</b> | -     | <b>1</b>  | .----- | !        | ..-..   |
| <b>I</b> | ..    | <b>U</b> | ..-   | <b>2</b>  | ..---- | :        | ----... |
| <b>J</b> | .---- | <b>V</b> | ...-  | <b>3</b>  | ...--  | "        | .-.-.   |
| <b>K</b> | -.-   | <b>W</b> | .--   | <b>4</b>  | ....-  | '        | .-....  |
| <b>L</b> | .-..  | <b>X</b> | -. -  | <b>5</b>  | .....  | =        | -...-   |

Observati diferentele in lungimele codurilor

# Codul Morse vs. Frecventa literelor

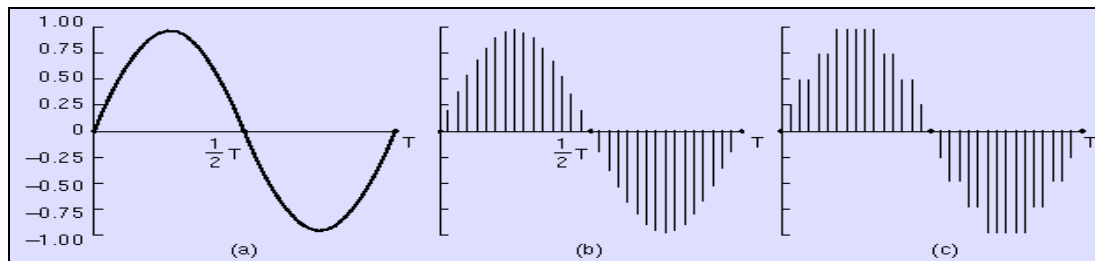
|   |      |   |        |
|---|------|---|--------|
| A | .-   | M | --     |
| B | -... | N | -. .   |
| C | -.-. | O | ---    |
| D | -..  | P | -. . . |
| E | .    | Q | -.-.-  |
| F | ..-. | R | -. .   |
| G | --.  | S | ... .  |
| H | .... | T | -      |
| I | ..   | U | ..-    |
| J | .--- | V | ...-   |
| K | -.-  | W | -. -   |
| L | -. . | X | -..-   |



- In general, frecventa mare/mica => cod scurt/lung
- Acesta este ideea in spatele schemei de codificare ce exploateaza redundanta statistica.

# Reprezentarea datelor

- Date analogice (continue)
  - Reprezentate de numere reale
  - Observatie: nu pot fi stocate in calculatoare
- Date digitale (discrete)
  - Valori intr-un set finit  $\{a_1, a_2, \dots, a_n\}$ ,
  - Toate datele sunt reprezentate ca siruri de simboluri din acest set.
  - Ex.:  $\{a,b,c,d,r\} \Rightarrow abc, car, bar, abracadabra, \dots$
  - Folosim date digitale pentru a aproxima date analogice





# Seturi de simboluri

- Alfabetul roman si semnele de punctuatie
- ASCII - 256 simboluri
- Braille, Morse
- Binar- {0,1}
  - 0 si 1 se numesc biti
  - Toate datele digitale pot fi reprezentate eficient cu biti
  - Ex.: {a, b, c, d}, reprezentare binara cu lungime fixa (2 bits/simbol):

|        |    |    |    |    |
|--------|----|----|----|----|
| Simbol | a  | b  | c  | d  |
| Binar  | 00 | 01 | 10 | 11 |

# Surse de informație si codificare

- Sursele de informatie pot fi analogice sau discrete
- Majoritatea surselor de informatie din domeniul calculatoarelor si al aplicatiilor internet sunt discrete
  - Pentru a descrie o sursă discretă fără memorie (SDFM) sunt necesare două mărimi: *alfabetul sursei și probabilitățile de furnizare a fiecărui simbol:*

$$S : \begin{pmatrix} s_1 & s_2 & \dots & s_N \\ p_1 & p_2 & \dots & p_N \end{pmatrix} \quad \sum_{k=1}^N p(s_k) = 1$$

# Surse de informație si codificare

- dacă numărul de simboluri este finit, sursa se numește *discretă*
- dacă la un moment dat se emite sigur un simbol atunci sursa este *completă*
- sursa este *fără memorie* dacă evenimentele  $s_k$  sunt independente, adică furnizarea unui simbol la un moment dat nu depinde de simbolurile furnizate anterior
- totalitatea simbolurilor unei surse formează *alfabetul sursei*
- orice succesiune finită de simboluri, în particular un singur simbol, se numește *cuvânt*
- totalitatea cuvintelor formate cu un anumit alfabet se numește *limbaj*

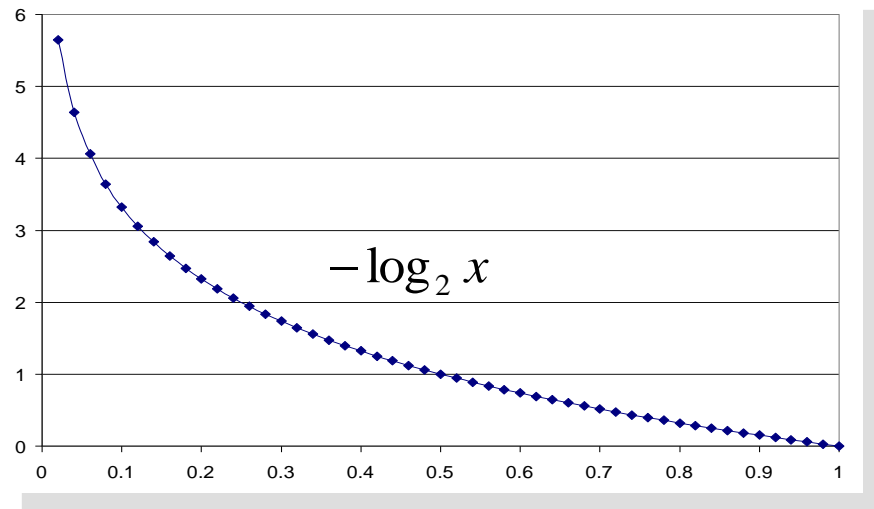
# Teoria informatiei

- Dezvoltata formal de Claude Shannon la Bell Labs in 1940-1950
- Explica limitele de codificare/transmisie folosind teoria probabilitatilor
- Informatia furnizata de un simbol A al unei surse de informatii este:
  - $P(A)$  reprezinta probabilitatea de aparitie a simbolului A

$$i(A) = \log_b \frac{1}{P(A)} = -\log_b P(A)$$

# Surse de informație si codificare

- Observatii
  - $P(A)$  mic  $\Rightarrow i(A)$  mare
  - $P(A)$  mare  $\Rightarrow i(A)$  mic
- Idee:
  - Simbolurile cu probabilitate mica (surpriza) contin mai multa informatie
- Daca A si B sunt independente atunci:
  - $i(AB) = i(A) + i(B)$



$$\begin{aligned}
 i(AB) &= \log_b \frac{1}{P(AB)} = \log_b \frac{1}{P(A)P(B)} = \\
 &= \log_b \frac{1}{P(A)} + \log_b \frac{1}{P(B)} = i(A) + i(B)
 \end{aligned}$$

# Exemplu: aruncarea monezii

- Moneda ideala
  - Fie H si T rezultatele aruncarii
  - Daca  $P(H) = P(T) = 1/2$ , atunci
  - $i(H) = i(T) = -1/\log_2(1/2) = \mathbf{1}$  bit
- Moneda masluita
  - Fie  $P(H) = 1/8$ ,  $P(T) = 7/8$
  - $i(H) = \mathbf{3}$  biti
  - $i(T) = \mathbf{0.193}$  biti
- Obs. ca  $P(H) + P(T) = 1$

# Entropie

- Entropia este *informația medie pe simbol* sau, altfel formulat, este *incertitudinea medie asupra simbolurilor sursei  $S$ , sau informația medie furnizată de un simbol*
  - Unitatea de măsură pentru entropie este **biti/simbol**
- Dacă experimentul generează simboluri, atunci (pentru  $\log_2$ )  $H$  este numărul mediu de simboluri binare necesare pentru codificare
- **Shannon: Nici un algoritm fără pierderi nu poate comprima mai bine**

$$H(S) = \sum_{i=1}^N p(s_i) i(s_i) = - \sum_{i=1}^N p(s_i) \log(p(s_i))$$

# Exemplul 1

- Se considera secventa:
  - 1 2 3 2 3 4 5 4 5 6 7 8 9 8 9 10
  - $P(1) = P(6) = P(7) = p(10) = 1/16$
  - $P(2) = P(3) = P(4) = P(5) = P(8) = P(9) = 2/16$
- *Presupunand ca secventa este iid (Independent & Identically Distributed):*

$$H = -\sum_{i=1}^{10} P(i) \log_2 P(i) = -4 \frac{1}{16} \log_2 \left( \frac{1}{16} \right) - 6 \frac{2}{16} \log_2 \left( \frac{2}{16} \right) = 3.25 \text{ bits}$$



## Exemplul 2

- Consideram secventa
  - 1 2 1 2 3 3 3 3 1 2 3 3 3 3 1 2 3 3 1 2
  - $P(1) = P(2) = 1/4$ ,  $P(3) = 1/2$ ,  $\mathbf{H} = \mathbf{1.5}$  bits/simbol
  - Total biti:  $20 \times 1.5 = 30$
- Reconsideram secventa
  - (1 2) (1 2) (3 3) (3 3) (1 2) (3 3) (3 3) (1 2) (3 3) (1 2)
  - $P(1\ 2) = 1/2$ ,  $P(3\ 3) = 1/2$
  - $\mathbf{H} = \mathbf{1}$  bit/simbol  $\times 10$  simboluri = 10 biti
- In teorie, se poate extrage o structura luand esantioane mai mari
- In realitate, ne trebuie un model corect, pentru ca de obicei nu este practic sa observam sursa prea mult timp