ARHITECTURA SISTEMELOR DE CALCUL - CURS 0x02

INTRODUCERE ÎN TEORIA INFORMAȚIEI

Cristian Rusu

DATA TRECUTĂ

- **am văzut cum transformăm dintr-o bază numerică în alta**
- **reprezentarea în complement față de doi**
- **am văzut operații de adunare și logice în binar**
- **dar de unde vin acești biți pe care facem operații?**
- **azi, vom studia conceptul de informație propriu-zis ...**

CUPRINS

- **teoria informației**
    - ce este informația
    - cum putem măsura cantitatea de informație
    - codarea datelor
    - detectarea și corectarea erorilor

- **referințe bibliografice**

TEORIA INFORMAȚIEI

- **ce este “informația”?**
  - niște date care afectează (în general, reduc) incertitudinea pe care o avem despre un eveniment/fenomen/etc.

- **un exemplu: se dă un pachet de cărți de joc cu 52 de cărți, care eveniment aduce cea mai mare cantitate de informație?**
  - extragem din pachet o carte care este inimă roșie
  - extragem din pachet o carte care este un rege
  - în care situație avem informație mai exactă despre carte?

TEORIA INFORMAȚIEI

- **ce este "informația"?**
  - niște date care afectează (în general, reduc) incertitudinea pe care o avem despre un eveniment/fenomen/etc.

- **un exemplu: se dă un pachet de cărți de joc cu 52 de cărți, care eveniment aduce cea mai mare cantitate de informație?**
  - extragem din pachet o carte care este inimă roșie (13/52)
  - extragem din pachet o carte care este un rege (4/52)

- **de ce? probabilitatea de a extrage un rege din pachet este mult mai mică, deci dacă extragem această carte primim mai multă informație (evenimentele rare produc multă informație)**
- **alt mod de a vedea problema: carți de inimă roșie sunt 13, regi sunt doar 4 deci e mai ușor să ghicim ce carte a fost extrasă**

TEORIA INFORMAȚIEI

- **cum măsurăm cantitatea de informație?**

- **introducem o variabilă aleatoare X:**
  - această variabilă poate lua *N* valori distincte (x_1, x_2, ..., x_N)
  - fiecare valoare dictinctă apare cu probabilitate (p_1, p_2, ..., p_N)

- **câtă informație primim dacă observăm că variabila X a luat valoarea x_i?**

$$I(x_i) = \log_2\left(\frac{1}{p_i}\right)$$

  - cu cât p_i este mai mic cu atât cantitatea de informație este mai mare
  - în exemplul anterior: $I(\mathrm{rege}) = \log_2\left(\frac{1}{\frac{4}{52}}\right) = \log_2\left(\frac{52}{4}\right) = 3.7 \text{ biti}$

TEORIA INFORMAȚIEI

- **cazurile anterioare:**

$$I(\text{rege}) = \log_2\left(\frac{1}{\frac{4}{52}}\right) = \log_2\left(\frac{52}{4}\right) = 3.7 \text{ biti}$$

$$I(\text{inima rosie}) = \log_2\left(\frac{1}{\frac{13}{52}}\right) = \log_2\left(\frac{52}{13}\right) = 2 \text{ biti}$$

- **câtă informație avem dacă știm exact cartea selectată din pachet?**

$$I(\text{o carte particulara}) = \log_2\left(\frac{1}{\frac{1}{52}}\right) = \log_2\left(\frac{52}{1}\right) = 5.7 \text{ biti}$$

- **aruncați o privire pe valorile de mai sus: 5.7 = 3.7 + 2**
  - e o concidență?

TEORIA INFORMAȚIEI

- **unele experimente nu ne oferă informația completă**
- **informația se poate acumula: se pot realiza mai multe experimente (putem pune mai multe întrebări)**
- **informația nu este creată sau distrusă, este constantă**

- **am folosit câteva concepte din teoria probabilităților**
  - care este probabilitatea ca un eveniment să se întâmple?
  - $$P(\text{eveniment A}) = \frac{\text{număr situații în care se întâmplă A}}{\text{număr total situații}}$$
  - $P(\text{eveniment A și eveniment B}) = P(\text{eveniment A}) \times P(\text{eveniment B})$
    - doar dacă cele două evenimente sunt independente
    - independența evenimentelor se presupune în multe situații pentru a simplifica rezultatele

https://en.wikipedia.org/wiki/Probability

TEORIA INFORMAȚIEI

- **să presupem că avem *N* evenimente, la fel de probabile**
- **rulăm un experiment (imperfect din punct de vedere informațional) care ne spune că doar *M* evenimente au fost posibile din totalul de *N***
- **ex: *N = 52* (pentru că avem 52 de cărți posibile), *M = 13* (pentru că aflăm că am extras o carte de inimă roșie, dar nu știm care exact e cartea)**

- **câtă informație primim?**

$$I = \log_2\left(\frac{1}{M\frac{1}{N}}\right) = \log_2\left(\frac{N}{M}\right)$$

- **ce se întâmplă dacă:**
  - *M = 1*
  - *N = M*
  - *M > N*

.

TEORIA INFORMAȚIEI

- **să presupem că avem *N* evenimente, la fel de probabile**
- **rulăm un experiment (imperfect din punct de vedere informațional) care ne spune că doar *M* evenimente au fost posibile din totalul de *N***
- **ex: *N = 52* (pentru că avem 52 de cărți posibile), *M = 13* (pentru că aflăm că am extras o carte de inimă roșie, dar nu știm care exact e cartea)**

- **câtă informație primim?**

$$I = \log_2\left(\frac{1}{M\frac{1}{N}}\right) = \log_2\left(\frac{N}{M}\right)$$

- **ce se întâmplă dacă:**
  - *M = 1* am redus incertitudinea la o singură variantă, perfect
  - *N = M* nu am rezolvat nimic, incertitudinea este aceeași
  - *M > N* mai rău, incertitudinea este mai mare

TEORIA INFORMAȚIEI

- **alte exemple**
  - aruncăm un ban (cap/pajură)
    - $N = ?, M = ?, I = ?$

TEORIA INFORMAȚIEI

- **alte exemple**
  - aruncăm un ban (cap/pajură)
    - $N = 2, M = 1, I = \log_2 (2/1) = 1$ bit
  - extragem o carte și vedem că e inimă roșie
    - $N = ?, M = ?, I = ?$

TEORIA INFORMAȚIEI

- **alte exemple**
  - aruncăm un ban (cap/pajură)
    - $N = 2, M = 1, I = \log_2 (2/1) = 1$ bit
  - extragem o carte și vedem că e inimă roșie
    - $N = 52, M = 13, I = \log_2 (52/13) = 2$ biți
  - aruncăm două zaruri
    - $N = ?, M = ?, I = ?$

TEORIA INFORMAȚIEI

- **alte exemple**
  - aruncăm un ban (cap/pajură)
    - $N = 2, M = 1, I = \log_2 (2/1) = 1$ bit
  - extragem o carte și vedem că e inimă roșie
    - $N = 52, M = 13, I = \log_2 (52/13) = 2$ biți
  - aruncăm două zaruri
    - $N = 36, M = 1, I = \log_2 (36/1) = 5.17$ biți
  - în filmele “Batman”, există un rău-făcător care are o monedă cu “cap” pe ambele părți
    - $N = ?, M = ?, I = ?$

TEORIA INFORMAȚIEI

- **alte exemple**
  - aruncăm un ban (cap/pajură)
    - $N = 2, M = 1, I = \log_2 (2/1) = 1$ bit
  - extragem o carte și vedem că e inimă roșie
    - $N = 52, M = 13, I = \log_2 (52/13) = 2$ biți
  - aruncăm două zaruri
    - $N = 36, M = 1, I = \log_2 (36/1) = 5.17$ biți
  - în filmele “Batman”, există un rău-făcător care are o monedă cu “cap” pe ambele părți
    - $N = 1, M = 1, I = \log_2 (1/1) = 0$ biți

ENTROPIA

- **entropia**
  - valoarea medie de informației primită despre o variabilă X

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^{N} -p_i \log_2 p_i$$

  - *H(X)* se numește entropia lui X
  - *I(X)* este informația despre X
  - *E* este “expected value”, operația care calculează valoarea medie

  - exemplu: X = {A, B, C, D} cu probabilități {1/3, ½, 1/12, ?}

.

ENTROPIA

- **entropia**
  - valoarea medie de informației primită despre o variabilă X

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^{N} -p_i \log_2 p_i$$

  - *H(X)* se numește entropia lui X
  - *I(X)* este informația despre X
  - *E* este “expected value”, operația care calculează valoarea medie

  - exemplu: X = {A, B, C, D} cu probabilități {1/3, ½, 1/12, 1/12}
    - care este entropia?
    -

ENTROPIA

- **entropia**
  - valoarea medie de informației primită despre o variabilă X

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^{N} -p_i \log_2 p_i$$

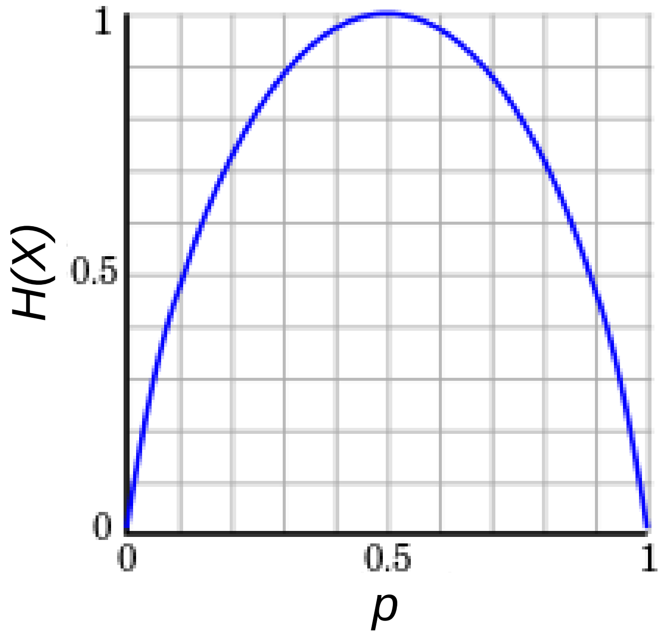
  - *H(X)* se numește entropia lui X
  - *I(X)* este informația despre X
  - *E* este "expected value", operația care calculează valoarea medie

  - exemplu: X = {A, B, C, D} cu probabilități {1/3, ½, 1/12, 1/12}
    - $H(X) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{1}{2}\log_2\frac{1}{2} - \frac{1}{12}\log_2\frac{1}{12} - \frac{1}{12}\log_2\frac{1}{12} = 1.626 \text{ biti}$
    - variabila X are 4 opțiuni, deci în mod normal am avea nevoie de 2 biți să memorăm toate posibilitațile, dar (pentru că probabilitățile nu sunt egale) putem să codăm mai bine de 2 biți

ENTROPIA

$$H(X) = E(I(X)) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^{N} -p_i \log_2 p_i$$

- **considerăm o monedă pe care o putem arunca**
- **probabilitatea de a vedea cap este *p* (iar pentru pajură este *1-p*)**



https://en.wikipedia.org/wiki/Entropy_(information_theory)

ENTROPIA

- **exemplu: X = {A, B, C, D} cu probabilități {1/3, ½, 1/12, 1/12}**
- **entropia lui X este 1.626**
- **ce se întâmplă?**
  - pot memora variabila X folosind câte 2 biți (codarea este: 00, 01, 10, 11) pentru fiecare eveniment posibil
    - este în regulă, dar ineficient
  - de ce este ineficient? pentru că entropia este 1.626 deci ne spune că putem fi mai eficienți, adică codarea de mai sus nu este cel mai eficient mod în care putem coda informația
    - în loc de 2 biți per eveniment putem să avem doar 1.626 biți
  - entropia ne spune că nu putem coda variabila de mai sus sub 1.626 biți per eveniment fără să pierdem informație
    - de exemplu, am putea coda X cu un singur bit (0 sau 1) dar atunci putem distinge doar între doua evenimente, nu patru
    - deci nu putem “decoda” ce s-a întamplat
- **entropia este limita de compresie posibilă**

https://en.wikipedia.org/wiki/Entropy_(information_theory)

CODAREA DATELOR

- **cum atingem acel 1.626 biți pentru cele patru evenimente?**
- **folosim o codare diferită de cea standard**
  - codarea standard A = 00, B = 01, C = 10, D = 11
  - cu această codare avem ABBC = 00 01 01 10
  - decodarea este directă: luăm câte 2 biți și fiecare e un eveniment

- **o altă codare (mai eficientă):**
  - dimensiune variabilă a codului: A = 01, B = 1, C = 000, D = 001
  - acum avem ABBC = 01 1 1 000
  - acum sunt 7 biți, față de 8 înainte (deci e mai bine)
  - decodarea trebuie să fie unică! trebuie să ne putem întoarce

- **o altă codare (și mai eficientă, dar incorectă):**
  - dimensiune variabilă a codului: A = 1, B = 0, C = 10, D = 11
  - acum avem ABBC = 1 0 0 10
  - 5 biți, și mai bine
  - problema? decodarea: dacă primim 10010 cum îl decodăm?

CODAREA DATELOR

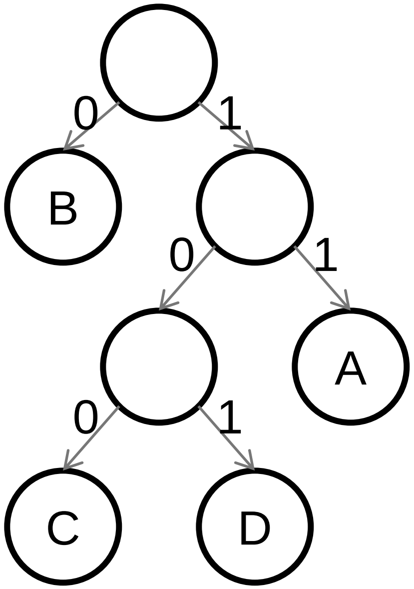
- **cum atingem acel 1.626 biți pentru cele patru evenimente?**
- **folosim o codare diferită de cea standard**
  - codarea standard A = 00, B = 01, C = 10, D = 11
  - cu această codare avem ABBC = 00 01 01 10
  - decodarea este directă: luăm câte 2 biți și fiecare e un eveniment

- **o altă codare (mai eficientă):**
  - dimensiune variabilă a codului: A = 01, B = 1, C = 000, D = 001
  - acum avem ABBC = 01 1 1 000
  - acum sunt 7 biți, față de 8 înainte (deci e mai bine)
  - decodarea trebuie să fie unică! trebuie să ne putem întoarce

- **o altă codare (și mai eficientă, dar incorectă):**
  - dimensiune variabilă a codului: A = 1, B = 0, C = 10, D = 11
  - acum avem ABBC = 1 0 0 10
  - 5 biți, și mai bine
  - problema? decodarea: dacă primim 10010 cum îl decodăm?
    - ABBC sau CBC sau …

CODAREA DATELOR

- **cum putem crea o codare eficientă și unică?**
  - un arbore binar
    - frunzele sunt codurile
    - stânga/dreapta e decis de 0/1
    - codarea este:
      - B = 0
      - A = 11
      - C = 100
      - D = 101
    - asta garantează codare eficientă
    și decodare unică

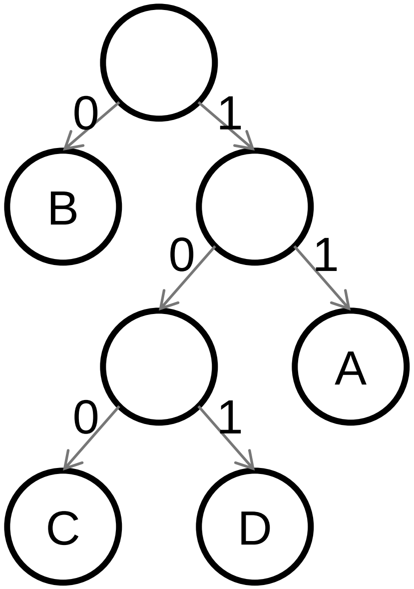


  - cum generăm codarea eficientă?
    - algoritmul Huffman
    - input: probabilitatea fiecărui eveniment {1/3, ½, 1/12, 1/12}
    - output: codurile care se citesc de pe un arbore binar (mai sus)
    - cheia: unele evenimente/simboluri apar mai des decât altele, deci acestea primesc o codare mai scurtă
    - dacă toate evenimente sunt equiprobabile, atunci nu putem face nimic

https://en.wikipedia.org/wiki/Huffman_coding

CODAREA DATELOR

- **cum putem crea o codare eficientă și unică?**
  - un arbore binar
    - frunzele sunt codurile
    - stânga/dreapta e decis de 0/1
    - codarea este:
      - B = 0
      - A = 11
      - C = 100
      - D = 101
    - asta garantează codare eficientă și decodare unică

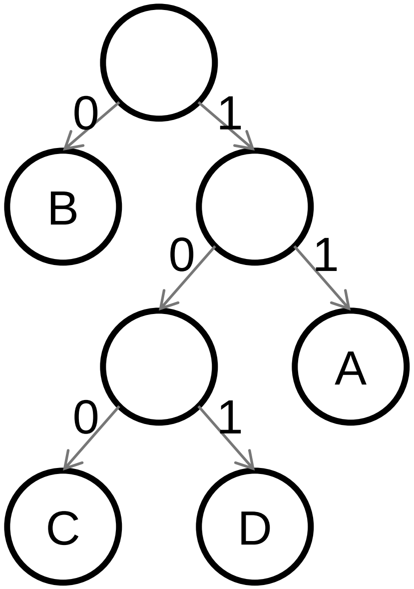


- exerciții: decodați 00100111010011

https://en.wikipedia.org/wiki/Huffman_coding

CODAREA DATELOR

- **cum putem crea o codare eficientă și unică?**
  - un arbore binar
    - frunzele sunt codurile
    - stânga/dreapta e decis de 0/1
    - codarea este:
      - B = 0
      - A = 11
      - C = 100
      - D = 101
    - asta garantează codare eficientă și decodare unică

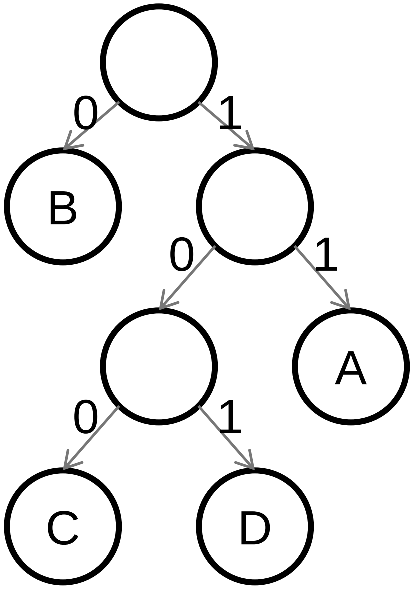


- exercițiu: decodați 0 0 100 11 101 0 0 11
  - soluția: BBCADBBA

https://en.wikipedia.org/wiki/Huffman_coding

CODAREA DATELOR

- **cum putem crea o codare eficientă și unică?**
  - un arbore binar
    - frunzele sunt codurile
    - stânga/dreapta e decis de 0/1
    - codarea este:
      - B = 0
      - A = 11
      - C = 100
      - D = 101
    - asta garantează codare eficientă și decodare unică

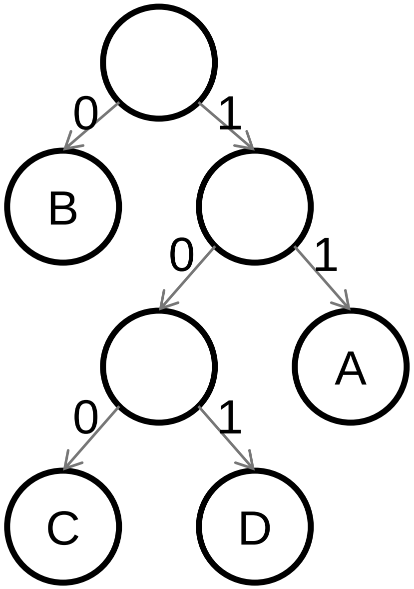


- exercițiu: cum calculăm eficiența acestei codări? dimensiunea in medie a unui mesaj este? probabilitățile sunt {1/3, ½, 1/12, 1/12}

https://en.wikipedia.org/wiki/Huffman_coding

CODAREA DATELOR

- **cum putem crea o codare eficientă și unică?**
  - un arbore binar
    - frunzele sunt codurile
    - stânga/dreapta e decis de 0/1
    - codarea este:
      - B = 0
      - A = 11
      - C = 100
      - D = 101
    - asta garantează codare eficientă
    și decodare unică



- exercițiu: cum calculăm eficiența acestei codări? dimensiunea in medie a unui mesaj este? probabilitățile sunt {1/3, ½, 1/12, 1/12}
    - 2x1/3 + 1x1/2 + 3x1/12 + 3x1/12 = 1.667 biți
    - comparat cu 1.626 biți care e optim

https://en.wikipedia.org/wiki/Huffman_coding

CODAREA DATELOR

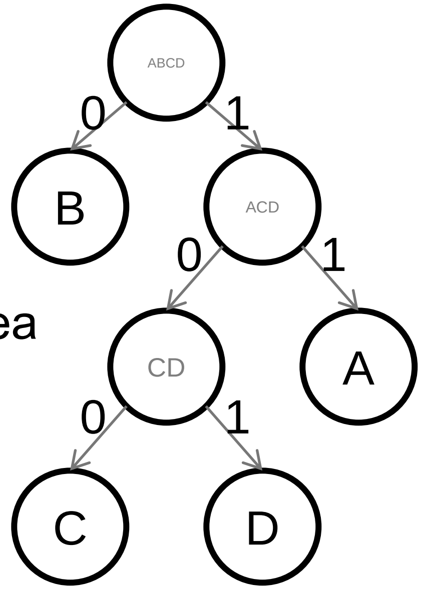
- **deja am văzut noi numere codate în sistemul binar**
  - codarea/memorarea numerelor naturale și întregi
  - ce fel de codare a fost aceasta? dimensiune fixă
  - indiferent de ce număr memorăm, folosim *N* biți
  - care sunt avantajele codării cu dimensiune fixă?
  - care sunt avantajele codării cu dimensiune variabilă?

CODAREA DATELOR

- **deja am văzut noi numere codate în sistemul binar**
  - codarea/memorarea numerelor naturale și întregi
  - ce fel de codare a fost aceasta? dimensiune fixă
  - indiferent de ce număr memorăm, folosim *N* biți
  - care sunt avantajele codării cu dimensiune fixă?
    - e simplu, știm că fiecare simbol are același număr de biți (deci știm de cât spațiu avem nevoie, etc.)
    - putem accesa direct al i-lea simbol din șir (ABBA etc.)
    - implementarea în circuite electronice se face la fel pentru că toate elementele au același număr de biți (32 sau 64 de biți)
    - este optimă dacă simbolurile au probabilități egale
  - care sunt avantajele codării cu dimensiune variabilă?
    - codare eficientă (spațiu de stocare)

CODAREA DATELOR

- **algoritmul Huffman**
  - input: probabilitatea fiecărui eveniment {1/3, ½, 1/12, 1/12}



- **algoritmul:**
  - luați evenimentele cele mai improbabile
    - C și D, și le punem în arbore
    - creăm un nou eveniment CD, probabilitatea de apariție a acestuia este 1/6 (suma C și D)
    - noile evenimente sunt {A, B, CD} cu probabilități {1/3, ½, 1/6}
  - din nou evenimentele cele mai improbabile
    - A și CD, A merge pe cealaltă frunză
    - noile evenimente sunt {B, ACD} cu probabilități {½, ½}
  - din nou evenimentele cele mai improbabile
    - acum sunt doar două, {B, ACD}, B merge pe cealaltă frunză

https://en.wikipedia.org/wiki/Huffman_coding

CODAREA DATELOR

- **algoritmul Huffman este optim dacă considerăm un singur simbol pe rând**
- **dar putem unii simboluri, adică putem face același arbore binar pentru toate combinațile de câte două simboluri**
  - AA, AB, AC, AD, BA, ..., CA, CB, CC, ..., DA, DB, DC, DD

- **câteva întrebări:**
  - câte simboluri avem acum? 16
  - care este probabilitatea următoarelor evenimente?
    - AA 1/9
    - DD 1/144
    - DB 1/24
    - AD 1/36

  - cum calculăm entropia acum? o sumă de 16 termeni
    - noua lungime medie Huffman, cu câte două simboluri? 1.646 biți
    - mai bine decât înainte (Huffman cu un singur simbol, 1.667 biți)
    - mai aproape de entropia de 1.626 biți (optim)
    - deci, daca asociem mai multe simboluri convergem către 1.626 biți

CODAREA DATELOR

- **comprimăm un text (avem un zip, de exemplu)**
  - putem comprima: text, imagini, chiar și executabile

- **ce ne oprește să mai comprimăm o dată? (să comprimăm zip-ul)**

https://en.wikipedia.org/wiki/Entropy_(information_theory)

CODAREA DATELOR

- **comprimăm un text (avem un zip, de exemplu)**
  - putem comprima: text, imagini, chiar și executabile

- **ce ne oprește să mai comprimăm o dată? (să comprimăm zip-ul)**

- **conținutul zip-ului arată complet aleator**
  - ceva comprimat perfect arată ca zgomot
  - doar zgomotul nu poate fi comprimat
  - dacă pierdem ceva din zgomot nu mai putem recupera

https://en.wikipedia.org/wiki/Entropy_(information_theory)

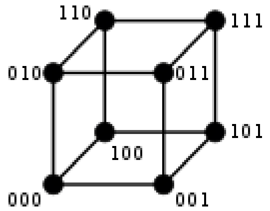
DETECTAREA / CORECTAREA ERORILOR

- **detectarea erorilor**
  - ce se întâmplă dacă memorăm un șir binar dar se întamplă ceva eroare: un 0 devine 1 sau un 1 devine 0?
  - de exemplu:
    - inițial avem: 0110 0101
    - datele sunt corupte și avem: 1100 0001
  - primul pas:
    - trebuie să definim o distanță între șirul corect și cel corupt
    - distanța Hamming între două șiruri binare: câți biți sunt diferiți (biți de pe aceleași poziții în preprezentarea binară)
    - șirurile trebuie să aibă aceeași lungime
    - distanța Hamming mai sus: 3

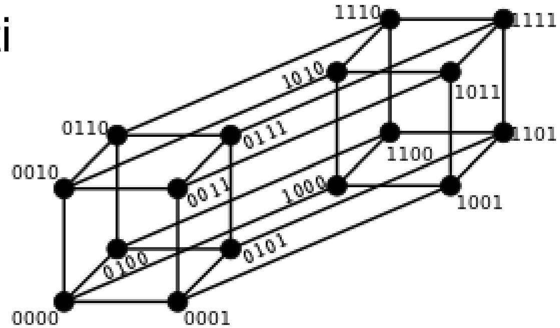
https://en.wikipedia.org/wiki/Hamming_distance

DETECTAREA / CORECTAREA ERORILOR

- **detectarea erorilor**
  - primul pas:
    - trebuie să definim o distanță între șirul corect și cel corupt
    - distanța Hamming între două șiruri binare: câți biți sunt diferiți (biți de pe aceleași poziții în preprezentarea binară)
    - șirurile trebuie să aibă aceeași lungime
    - distanța Hamming în exemplul anterior: 3
    - Hamming pentru șiruri de 3 biți



    - Hamming pentru șiruri de 4 biți



https://en.wikipedia.org/wiki/Hamming_distance

DETECTAREA / CORECTAREA ERORILOR

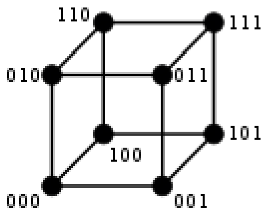
- **detectarea erorilor**
  - exemplul anterior:
    - inițial avem: 0110 0101
    - datele sunt corupte și avem: 1100 0001
  - problema: 0110 0101 și 1100 0001 sunt șiruri valide
  - adică, nu ne putem da seamă că o eroare s-a produs
  - ideea: nu toate șirurile binare vor fi posibile: în felul acesta, dacă apare un șir binar care nu este permis (care nu există) atunci știm că o eroare s-a produs undeva
  - ideea cea mai simplă: adăugăm un bit de paritate simbolurilor
    - 0 devine 00
    - 1 devine 11
    - doar aceste două noi simboluri sunt valide

DETECTAREA / CORECTAREA ERORILOR

- **detectarea erorilor**
  - ideea cea mai simplă: adăugăm un bit de paritate simbolurilor
    - 0 devine 00
    - 1 devine 11
    - doar aceste două noi simboluri sunt valide
  - dacă primim 01 sau 10, știm că o eroare s-a produs
  - dacă primim 00 sau 11, știm că totul e OK
  - ce am făcut? am adăugat redundanță: adică putem pierde ceva și tot ne putem da seama ce simbol avem
  - ce pierdem? suntem de două ori mai ineficienți (în loc de un bit trebuie acum să stocăm doi biți)
  - putem detecta erori multiple? nu. dacă sunt două erori atunci 00 se poate transforma în 11 din cauza erorilor

DETECTAREA / CORECTAREA ERORILOR

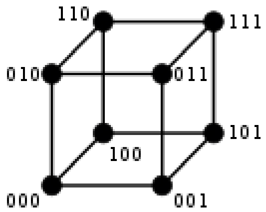
- **corecția erorilor**
  - distanță Hamming destul de mare poate să ducă și la corectarea erorilor (nu doar detectarea lor)
  - să considerăm șiruri de 3 biți



  - singurele coduri valide sunt “000” care e “0” și “111” care e “1”
    - dacă primim “001” sau “010” sau “100” putem suspecta că e “0”
    - dacă primim “110” sau “101” sau “011” putem suspecta că e “1”
  - o distanță Hamming de *2E+1* poate corecta *E* erori
    - care e intuiția?

DETECTAREA / CORECTAREA ERORILOR

- **corecția erorilor**
  - distanță Hamming destul de mare poate să ducă și la corectarea erorilor (nu doar detectarea lor)
  - să considerăm șiruri de 3 biți



  - singurele coduri valide sunt “000” care e “0” și “111” care e “1”
    - dacă primim “001” sau “010” sau “100” putem suspecta că e “0”
    - dacă primim “110” sau “101” sau “011” putem suspecta că e “1”
  - o distanță Hamming de *2E+1* poate corecta *E* erori
    - care e intuiția? majority vote
  - în general, dacă vrem să detectăm E erori avem nevoie de o distanță Hamming între coduri de E + 1: adică “0” poate să fie “000” iar “1” poate să fie “111” – suntem de 3 ori mai ineficienți acum ca stocare, dar putem detecta mai 2 erori și putem corecta o eroare

TEORIA INFORMAȚIEI

- **un exercițiu ...**

TEORIA INFORMAȚIEI

Eur*pa a în*eg*stt cel mai m**e n*m*r să**ăm*nal de c**uri de **ro**virus de până ac**, iar Or*******ia M***ială a Sa**tății a averti*** că bil**țu*ile zi**ic* ale de*es**or ar putea aju**e în ap**lie 2021 să fie de 4-5 ori mai m**i decât în pri****ra acestui an. Ş*ptes***zece ţări, între care şi R****ia, din to**lul celor 27 de s***e me***e UE p*** M**** B******e ***t ****ate cu r**u pe n**a h**** *****eană de **** ep******.**

.

TEORIA INFORMAȚIEI

Europa a înregistrat cel mai mare număr săptămânal de cazuri de coronavirus de până acum, iar Organizația Mondială a Sănătății a avertizat că bilanțurile zilnice ale deceselor ar putea ajunge în aprilie 2021 să fie de 4-5 ori mai mari decât în primăvara acestui an. Şaptesprezece ţări, între care şi România, din totalul celor 27 de state membre UE plus Marea Britanie sunt marcate cu roşu pe noua hartă europeană de risc epidemic.

https://www.hotnews.ro/stiri-coronavirus-24355802-coronavirus-lume-16-octombrie.htm

TEORIA INFORMAȚIEI

- **un alt exercițiu ...**

TEORIA INFORMAȚIEI

I cnduo't bvleiee taht I culod aulaclty uesdtannrd waht I was rdnaieg. Unisg the icndeblire pweor of the hmuan mnid, aocdcrnig to rseecrah at Cmabrigde Uinervtisy, it dseno't mttaer in waht oderr the lterets in a wrod are, the olny irpoamtnt tihng is taht the frsit and lsat ltteer be in the rhgit pclae. The rset can be a taotl mses and you can sitll raed it whoutit a pboerlm. Tihs is bucseae the huamn mnid deos not raed ervey ltteer by istlef, but the wrod as a wlohe. Aaznmig, huh? Yaeh and I awlyas tghhuot slelinpg was ipmorantt! See if yuor fdreins can raed tihs too.

https://www.ecenglish.com/learnenglish/lessons/can-you-read

TEORIA INFORMAȚIEI

- de ce am făcut aceste exerciții?
- care este “morala”?

TEORIA INFORMAȚIEI

- **de ce am făcut aceste exerciții?**
- **care este “morala”?**

- **limba română și engleză sunt redundante**
  - în plus, mașina de decodare pe care o avem (creierul) este destul de performantă în astfel de situații

- **putem extrapola și putem spune că toate limbile lumii sunt redundante**
- **altfel, nu am putea comunica din cauza zgomotului (în cazul acesta, zgomot este la propriu)**

CE AM FĂCUT ASTĂZI

- am discutat despre informație
- am discutat despre entropia lui Shannon
- am văzut cum codăm/decodăm informația
- detectarea și corecția erorilor

DATA VIITOARE ...

- **abstractizarea digitală**
- **începem să discutăm despre circuite electronice**
- **tranzistorul**
- **cum sunt implementate operații de pe un sistem de calcul cu circuite electronice**
- **circuite de bază**

LECTURĂ SUPLIMENTARĂ

- **David MacKay, Information Theory, Inference, and Learning Algorithms, 2003**
  - I Data Compression
  - cele 3 capitole introductive

- **Huffman Codes: An Information Theory Perspective, https://www.youtube.com/watch?v=B3y0RsVCyrw**

- **The Universe is Hostile to Computers, https://www.youtube.com/watch?v=AaZ_RSt0KP8**

- **video-urile lui MacKay (primele sunt cele relevante pentru noi)**
  - https://www.youtube.com/playlist?list=PLruBu5BI5n4aFpG32iMbdWoRVAA-Vcso6

https://www.inference.org.uk/itprnn/book.pdf

LECTURĂ SUPLIMENTARĂ (NU INTRA ÎN EXAMEN)

- **Sean Caroll, The Biggest Ideas in the Universe | 20. Entropy and Information, https://www.youtube.com/watch?v=rBPPOI5UIe0**

- **Computerphile, playlist despre entropie și informație, https://www.youtube.com/playlist?list=PLzH6n4zXuckpKAj1_88VS-8Z6yn9zX_P6**

- **3Blue1Brown, Hamming codes, h■w to ov■rco■e n■ise, https://www.youtube.com/watch?v=X8jsijhllIA**
- **3Blue1Brown, Hamming codes part 2, the elegance of it all, https://www.youtube.com/watch?v=b3NxrZOu_CE**

- **Reed-Solomon Encoding**
  https://www.youtube.com/watch?v=fBRMaEAFLE0
  https://www.youtube.com/watch?v=xE4jEKx9fTM