

C.2 CUANTIFICAREA INFORMAȚIEI ȘI DETECȚIA DE STRUCTURI MALWARE

PAUL A. GAGNIUC



Academia Tehnică Militară „Ferdinand I”

PRINCIPALELE PĂRȚI ALE PREZENTĂRII

C.2 Cuantificarea informației și detecția de structuri malware:

- C.2.1 EVOLUȚIA LIMBAJULUI ȘI STRUCTURA CODULUI
- C.2.2 ÎNȚELEGEREA ENTROPIEI PRIN PRISMA COMPRESIEI
- C.2.3 CUANTIFICAREA INFORMAȚIEI
- C.2.4 DIMENSIUNI ȘI SISTEME DE REFERINȚĂ

C.2.1

EVOLUȚIA LIMBAJULUI ȘI STRUCTURA CODULUI



Inteligența biologică (IB): Descrierea obiectului



Peștera *Altamira* – Spania
(acum 35.600 de ani)

- Codificarea informațiilor prin simboluri, obiecte desenate.
- Prea multe situații complexe generează un număr mare de simboluri.
- Interpretari subiective.

IB: Descrierea evenimentelor (I)

BC, î.Hr = Before Christ, Înainte de Hristos

Pestera *Măgura* din Bulgaria (6300 BC – 3000 BC)

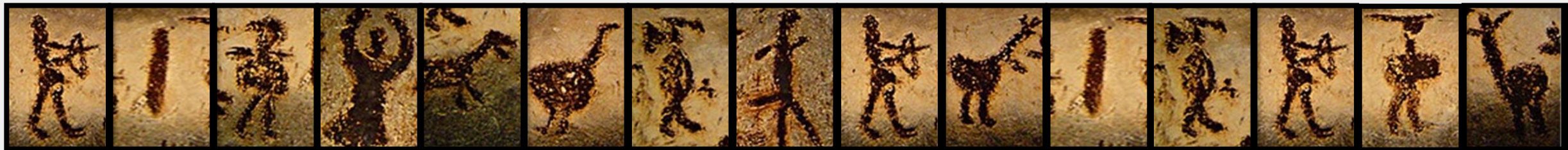


- În timp s-a făcut trecerea la simboluri fundamentale, a căror ordine succesivă poate codifica situații complexe pe combinații de simboluri.



A B C D E F G H I J K

Simboluri **unice observate** cu **semnificație necunoscută** reprezentate folosind simboluri ASCII.



B A G I D C F E B H A F B J K

Simboluri cu semnificație necunoscută observate in secvența:

Inapoi in trecut! Unde observam cod malițios?

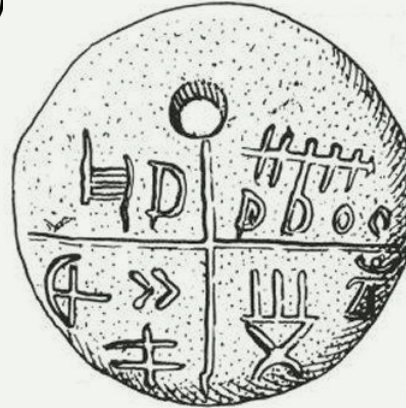
IB: Descrierea evenimentelor (II)

BC, Î.Hr = Before Christ, Înainte de Hristos

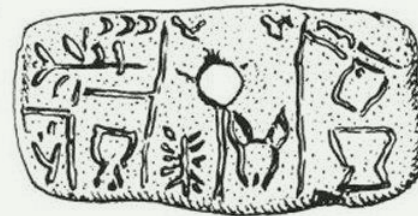
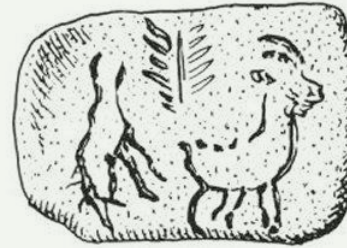
Fiecare simbol cu o însemnătate (mai avem simboluri de acest fel în China și Japonia)

100k
simboluri

50k
simboluri



0 3 cm



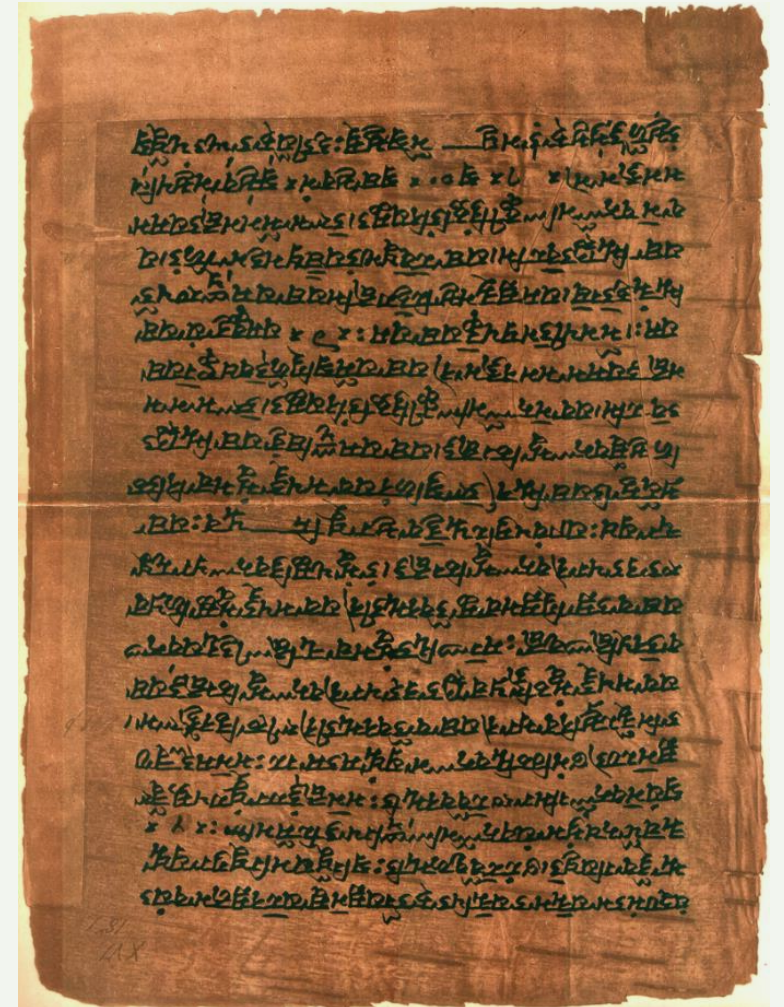
Tabletele Kish (3500 BC)
1920 - Irak



Tabletele de la Tărtăria (5300 BC)
1961 - Romania

Codificarea vorbirii prin simboluri fundamentale (gândire matematică)

Atharva Veda de la Hinduși (1500 BC)





Limba română este o limbă modernă!

Textul cu sens negativ poate fi emergent!

Semnificația construcțiilor imaginare !

- Care este punctul de plecare pentru descifrarea unei limbi străine?
- Ce deosebește o înjurătură de un cuvânt de laudă?
- Înjurăturile au altă entropie decât laudele?
- Dacă nu sunt folosite cuvinte negative, poate exista un text negativ din punct de vedere al contextului? (toate cuvintele pot fi pozitive, dar în context conotația este negativă).
- Toate construcțiile imaginare sunt restricționate de legi universale (pattern, repetiții).

Diferență între:

- sens negativ
- sens pozitiv

Codul mașină este un limbaj ca oricare altul!

Codul malițios este emergent !



- Codul rău intenționat are o entropie diferită de codul normal?
- Există un dialect malware?
- Cum facem diferența dintre malware și codul normal?

Codul mașina este un limbaj imperativ!

Diferența dintre:

- cod malware (non-self)
- cod normal (self)

C.2.2

ÎNTELEGEREA ENTROPIEI PRIN PRISMA COMPRESIEI

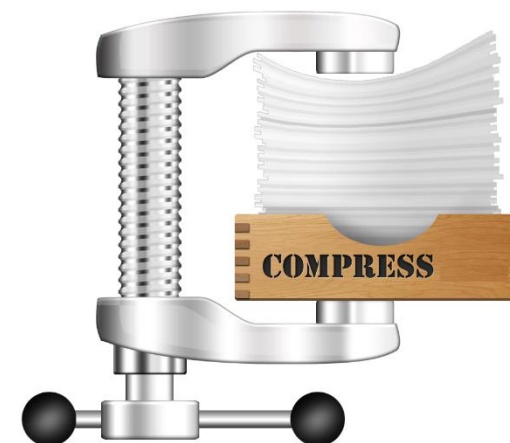


Legile universale: **COMP**PRESIA



Esentă

(redundanță)
distilare = optimizare



Esentă

Legi universale: Pornim de la entropia *Shannon*

Cum se masoara esenta?

Cum se masoara concentratia esentei?

$$p(A) = \frac{f(A)}{\text{lungime secventa}}$$

$$p(B) = 1 - p(A)$$

Exemplu - entropiile calculate pentru fiecare secvență, rotunjite la două zecimale:

ABABABABABABABABABAB	- [1.00]	Alternanță perfectă între A și B.
AABBAABBAABBAABBAABB	- [1.00]	Perechi de AA și BB, repetate.
AAABBBAAABBBAAABBBAA	- [0.99]	Grupuri de trei A-uri urmate de trei B-uri.
AAAABBBBAAAABBBBAAAA	- [0.97]	Patru A-uri urmate de patru B-uri.
AAAAAABBBBBBAAAAAABBB	- [0.97]	Șase A-uri urmate de cinci B-uri.
AAAAAABBBBBBAAAAAAA	- [0.90]	Șapte A-uri urmate de șase B-uri.
AAAAAABBBBBBAAAAAA	- [0.93]	Opt A-uri urmate de șapte B-uri.
AAAAAABBBBBBAAAAA	- [1.00]	Zece A-uri urmate de zece B-uri.
AAAAAAAAAAAAAAAAAAAAA	- [0.00]	Doar A-uri.
BBBBBBBBBBBBBBBBBBBB	- [0.00]	Doar B-uri.

Entropia măsoară gradul de aleatoriu într-o secvență. O valoare mai mare a entropiei indică o mai mare varietate sau imprevizibilitate în secvență.

Entropia maxima (model nul):

$$e = - \sum_{i=1}^n p_i \times \log_2(p_i)$$

$$e = - \left[p(A) \times \log_2(p(A)) + p(B) \times \log_2(p(B)) \right]$$

$$e = - \left[0.5 \times \log_2(0.5) + 0.5 \times \log_2(0.5) \right]$$

$$e = - \left[0.5 \times -1 + 0.5 \times -1 \right]$$

$$e = - \left[-0.5 + -0.5 \right]$$

$$e = -[-1] = 1$$

Entropia pe secvențe (I)

Legi universale: Entropia

Dacă înlocuim alfabetul secvenței anterioare cu numere?

Entropiile calculate pentru secvențele cu “A” înlocuit cu “1” și “B” cu “0”, rotunjite la două zecimale:

10101010101010101010	- [1.00]	Alternanță perfectă între 1 și 0.
11001100110011001100	- [1.00]	Perechi de 11 și 00, repetate.
11100011100011100011	- [0.99]	Grupuri de trei 1-uri urmate de trei 0-uri.
11110000111100001111	- [0.97]	Patru 1-uri urmate de patru 0-uri.
11111100000111110000	- [0.97]	Șase 1-uri urmate de cinci 0-uri.
11111110000001111111	- [0.90]	Șapte 1-uri urmate de șase 0-uri.
11111111000000011111	- [0.93]	Opt 1-uri urmate de șapte 0-uri.
11111111110000000000	- [1.00]	Zece 1-uri urmate de zece 0-uri.
11111111111111111111	- [0.00]	Doar 1-uri.
00000000000000000000	- [0.00]	Doar 0-uri.

Entropia rămâne aceeași ca și în cazul secvențelor originale, deoarece schimbarea caracterelor nu afectează distribuția lor.

Entropia pe secvențe (II)

Entropia maxima (model nul):

$$e = - \sum_{i=1}^n p_i \times \log_2(p_i)$$

$$e = - \left[\begin{array}{l} p(0) \times \log_2(p(0)) + \\ p(1) \times \log_2(p(1)) \end{array} \right]$$

$$e = - \left[\begin{array}{l} 0.5 \times \log_2(0.5) + \\ 0.5 \times \log_2(0.5) \end{array} \right]$$

$$e = - \left[\begin{array}{l} 0.5 \times -1 + \\ 0.5 \times -1 \end{array} \right]$$

$$e = - \left[\begin{array}{l} -0.5 + \\ -0.5 \end{array} \right]$$

$$e = -[-1]=1$$

Legi universale: Entropia – fara normalizare

O succesiune de două simboluri ne oferă o valoare maximă a entropiei egală cu 1.

Ce se întâmplă când numărul de simboluri din alfabetul secvenței crește?

Valorile entropiei ne-normalizate pentru secvențele cu trei caractere (A, B, și C), rotunjite la două zecimale:

ABCABCABCABCABCAB	- [1.58]	Alternanță regulată între A, B și C.
AABBCCAABBCCAABBCCAA	- [1.57]	Perechi de AA, BB și CC, repetate.
AAABBBCCAABBBCCAAB	- [1.51]	Grupuri de trei A-uri, B-uri și două C-uri.
AAAABBBBCCCCAAAABBBB	- [1.52]	Patru A-uri, B-uri și C-uri.
AAAAAABBBBBBCCCCAAAA	- [1.50]	Șase A-uri, B-uri și patru C-uri.
AAAAAAABBBBBBCCCCCA	- [1.57]	Șapte A-uri, șase B-uri și C-uri.
AAAAAAABBBBBBBCCCCC	- [1.56]	Opt A-uri, șapte B-uri și patru C-uri.
AAAAAAAAAABBBBBBBBBB	- [1.00]	Zece A-uri urmate de zece B-uri.
AAAAAAAAAAAAAAAAAAAAA	- [0.00]	Doar A-uri.
BBBBBBBBBBBBBBBBBBBB	- [0.00]	Doar B-uri.

Model nul (la ce ne așteptam):

$$p = \frac{1}{\text{nr. caractere in alfabet}}$$

Model observant:

$$p = \frac{f(x_i)}{\text{lungime secventa}}$$

Valorile entropiei ne-normalizate sunt calculate folosind același principiu, dar fără a fi împărțite la logaritmul numărului de caractere posibile. Aceste valori oferă o măsură a incertitudinii în secvență, dar nu sunt scalate pentru a fi direct comparabile între diferite seturi de caractere sau lungimi de secvență.

Iată entropiile normalizate calculate pentru fiecare secvență cu trei caractere (A, B, și C), rotunjite la două zecimale:

ABCABCABCABCABCAB	- [1.00]	Alternanță regulată între A, B și C.
AABBCCAABBCCAABBCCAA	- [0.99]	Perechi de AA, BB și CC, repetate.
AAABBBCCAAABBBCCAAAB	- [0.95]	Grup de trei A-uri, B-uri și două C-uri.
AAAABBBBCCCCAAAABBBB	- [0.96]	Patru A-uri, B-uri și C-uri.
AAAAAABBBBBBCCCCAAAA	- [0.95]	Șase A-uri, B-uri și patru C-uri.
AAAAAAABBBBBBBCCCCCA	- [0.99]	Șapte A-uri, șase B-uri și C-uri.
AAAAAAAABBBBBBBBCCCCC	- [0.98]	Opt A-uri, șapte B-uri și patru C-uri.
AAAAAAAAAABBBBBBBBBBB	- [0.63]	Zece A-uri urmate de zece B-uri.
AAAAAAAAAAAAAAAAAAAAA	- [0.00]	Doar A-uri.
BBBBBBBBBBBBBBBBBBBBB	- [0.00]	Doar B-uri.

Entropia normalizată variază între 0 și 1, unde 0 indică o lipsă totală de incertitudine (posibilitatea de secvențe uniforme), iar 1 reprezintă maximul de variabilitate și aleatoriu în secvență.

Entropia este normalizată?

Pentru a normaliza entropia, aceasta este de obicei împărțită la logaritmul numărului de caractere posibile (în acest caz, $\log_2(3) = 1.58$, deoarece sunt doar trei caractere posibile, A, B și C).

Astfel, valorile entropiei pe care le-am calculat sunt de fapt entropii normalizate, deoarece variază între 0 (pentru o secvență complet predictibilă) și 1 (pentru o secvență complet aleatoare sau echilibrată între A, B și C).

În exemplul nostrum precedent, deoarece baza logaritmului era 2 și numărul de caractere posibile este de asemenea 2, entropia calculată era deja într-o formă normalizată.

$$e = 0$$



$$e = \max = 1$$



100% Alb

$$p(A) = 1$$

$$p(N) = 0$$

50% Alb; 50% Negru

$$p(A) = 0.5$$

$$p(N) = 0.5$$

Orice imagine alb/negru poate fi reprezentată printr-o matrice binară, de aceea urmează aceleași reguli ca și o secvență unidimensională.

$$e = - \sum_{i=1}^n p_i \times \log_2(p_i)$$

$$e = - \left[p(A) \times \log_2(p(A)) + p(N) \times \log_2(p(N)) \right]$$

$$e = - \left[0.5 \times \log_2(0.5) + 0.5 \times \log_2(0.5) \right]$$

$$e = - \left[0.5 \times -1 + 0.5 \times -1 \right]$$

$$e = - \left[-0.5 + -0.5 \right]$$

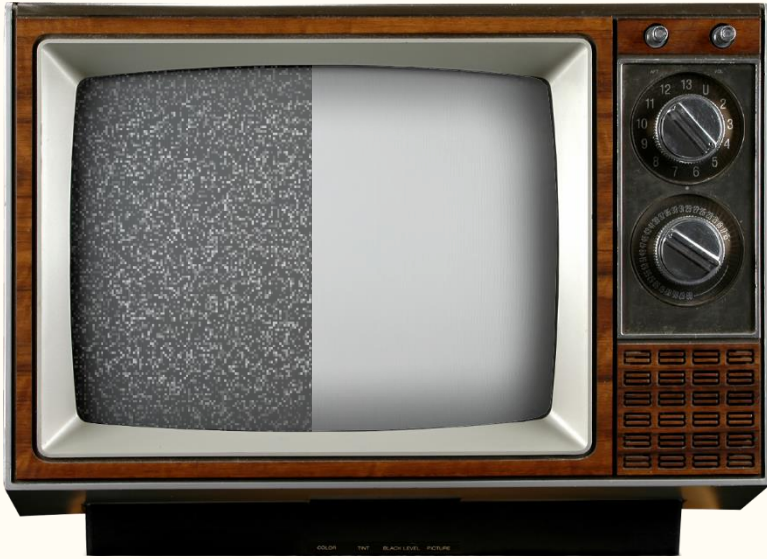
$$e = -[-1] = 1$$

Entropia pe două dimensiuni

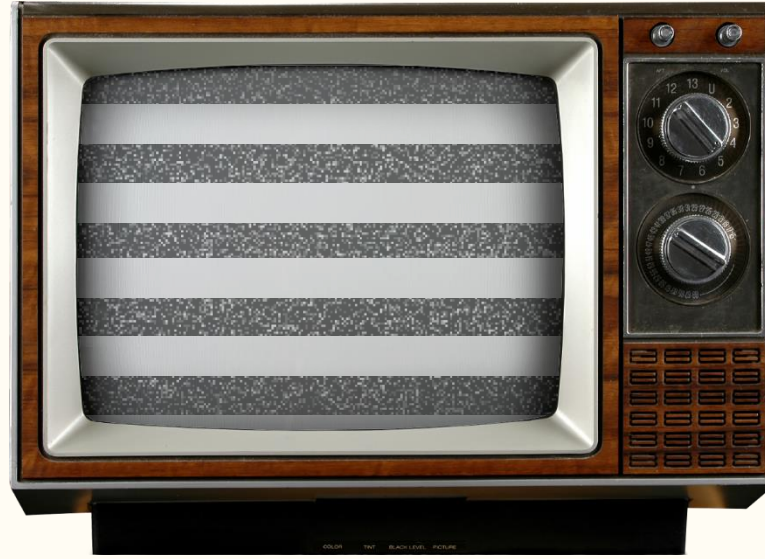
Ex stări: 00
Encodare culori 01 $\frac{1}{4}$
sau nuante 10
de gri. 11

$$e = - \sum_{i=1}^{n=4} \frac{1}{4} \times \log_2 \left(\frac{1}{4} \right)$$

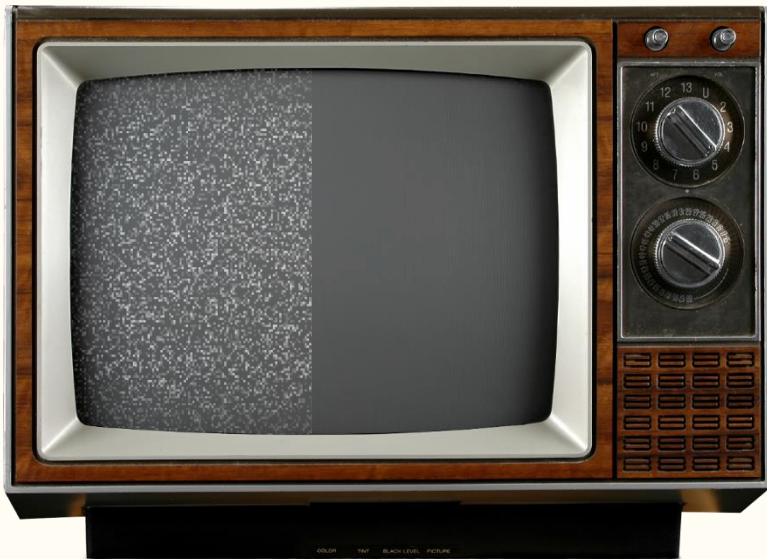
75% alb; 25% negru



75% alb; 25% negru



25% alb; 75% negru



25% alb; 75% negru



Compresie vs. Entropie

$$e = - \sum_{i=1}^n p_i \times \log_2(p_i)$$

$$e = - \left[p(A) \times \log_2(p(A)) + p(N) \times \log_2(p(N)) \right]$$

$$e = - \left[0.75 \times \log_2(0.75) + 0.25 \times \log_2(0.25) \right]$$

$$e = - \left[0.75 \times -0.41 + 0.25 \times -2 \right]$$

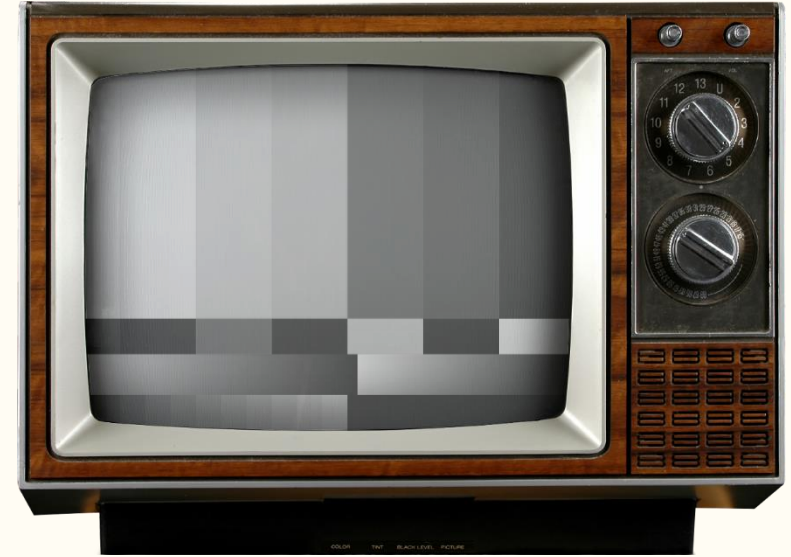
$$e = - \left[-0.31 + -0.5 \right]$$

$$e = -[-0.81] = 0.81$$

Mitul entropiei *Shannon* și cuantificarea informației!

Entropie și intuiție!

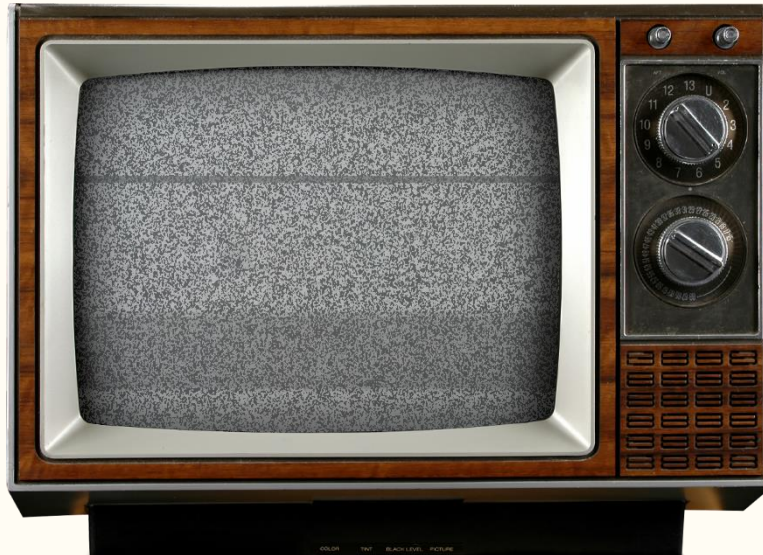
$E = ?$



$E = \max$



$E = \max-q$



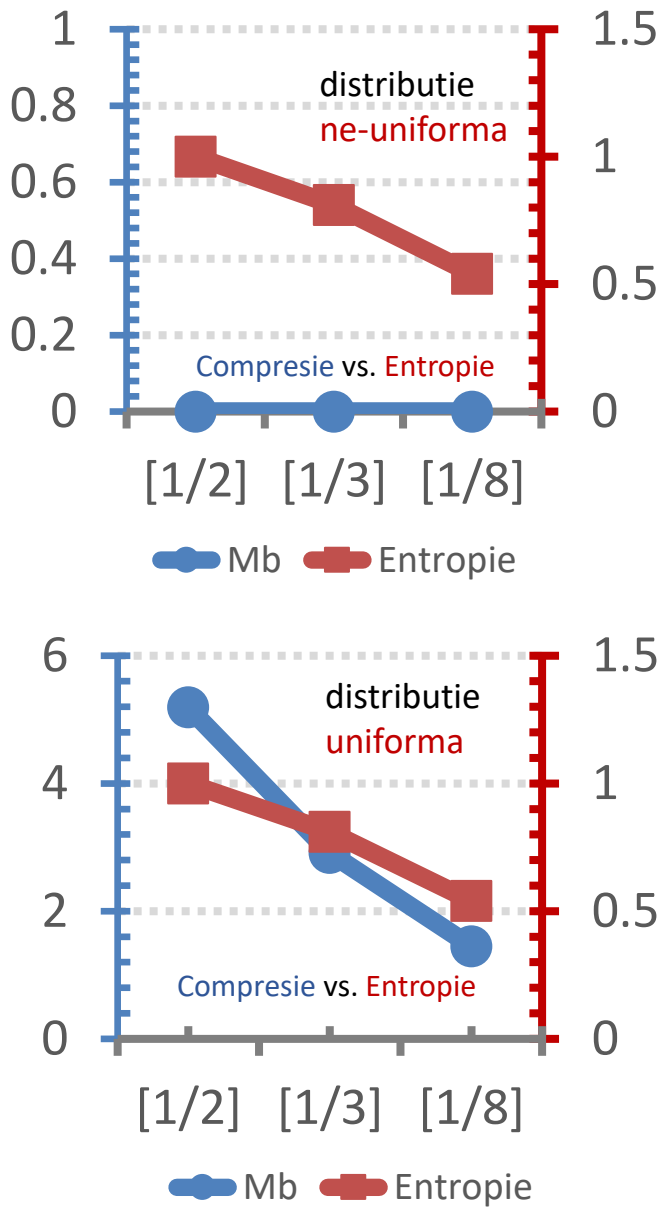
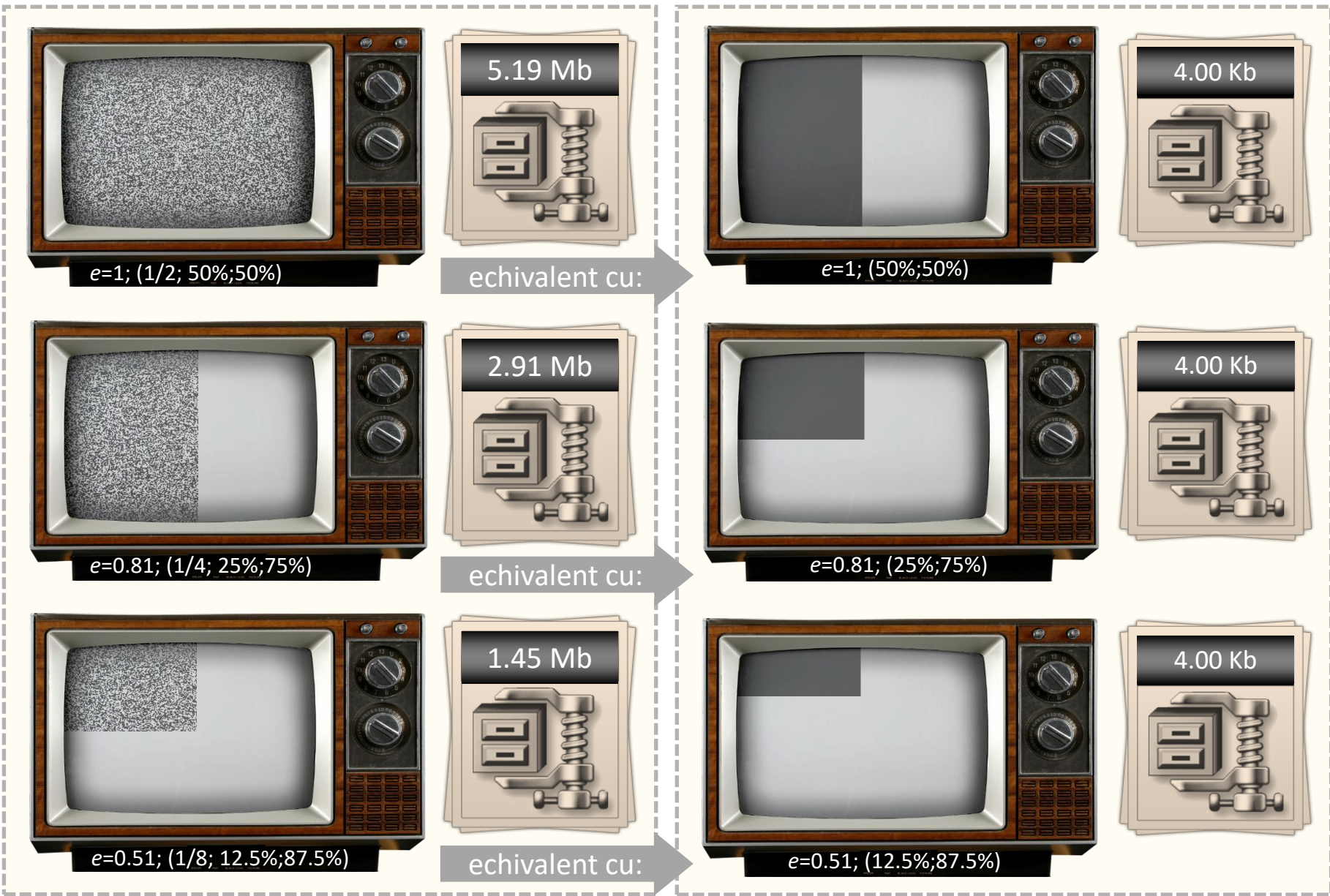
$E = ?$



Compresie vs. Entropie: **uniforma** vs **ne-uniforma** !

Distributie **uniforma** (4000x4000 pixeli; *.png)

Distributie **ne-uniforma** (4000x4000 pixeli; *.png)



Putem măsura cantitatea de informație folosind entropia *Shannon*?

Nu !

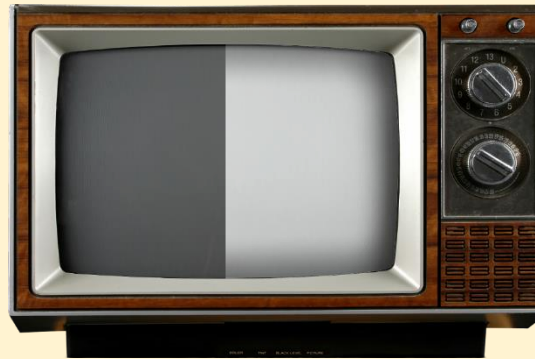
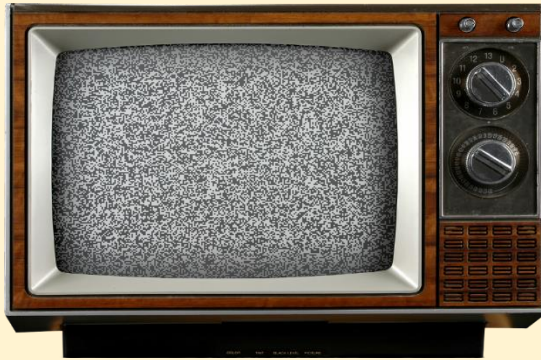
Ordinea nu este luata in considerare

1D

10101010101010101010



11111111110000000000

2D



nD

(obiecte multidimensionale)


$$e = 1$$


La ce se referă exact predictibilitatea?



Cantitatea de informație posibilă, **nu cea existentă!**

Când entropia este mare, predictibilitatea este scăzută:

ABCDABCDABCDABCD - [2]

ABCD {0.25, 0.25, 0.25, 0.25}

Când entropia este scăzută, predictibilitatea este ridicată:

ABADAAAAABAADBBA - [1.3]

ABCD {0.60, 0.20, 0.10, 0.10}

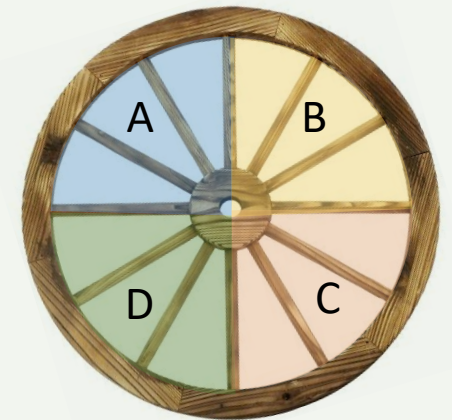
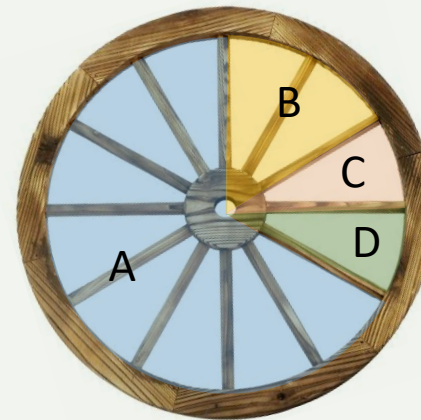
```
import random

def generate_sequence(total_chars=16, distribution=None):

    if distribution is None:
        distribution = \
            {'A': 0.6, 'B': 0.2, 'C': 0.1, 'D': 0.1}

    chars = ''.join(random.choices(
        population=list(distribution.keys()),
        weights=distribution.values(),
        k=total_chars)
    )

    return chars
sequence = generate_sequence()
print(sequence)
```



Ce masoara entropia? Concluzie !

- Entropia măsoară proporțiile, unde obiectele prezente în egală măsură într-o secvență, reprezintă entropia cu valoarea maximă. Îndepărtarea de la proporțiile egale arată o valoare a entropiei care devine din ce în ce mai mică.
- Cu alte cuvinte, ordinea obiectelor din secvență **nu** contează în cazul entropiei.

- Mențiune: Prin menținerea entropiei localizate la niveluri mai mici decât mediul înconjurător, sistemele biologice sunt capabile să construiască informații și ordine.
- Acest lucru evidențiază faptul că sistemele biologice trebuie să disipeze entropia în mediul înconjurător pentru a construi ordinea locală. De exemplu, ordinea într-un oras arunca entropia la gunoi.
- De exemplu, un produs ambalat conține mai multe obiecte care se află în același sistem de referință. După desigilare, sistemul de referință comun dispare și obiectele din pachet devin independente, astfel, mediul conține mai multe tipuri de obiecte și entropia crește. Entropia scade atunci când obiectele secundare sunt aruncate la gunoi.

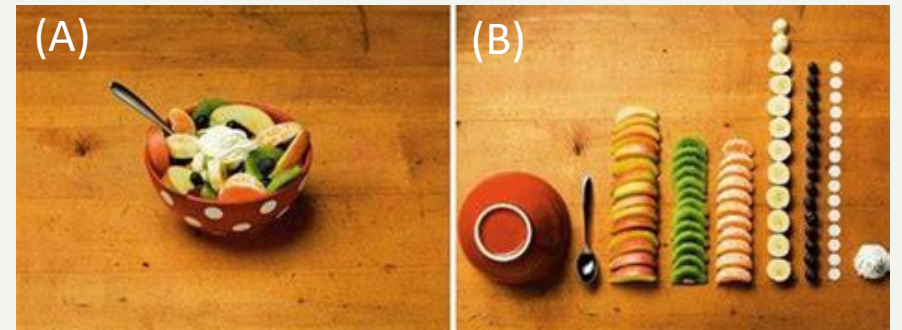
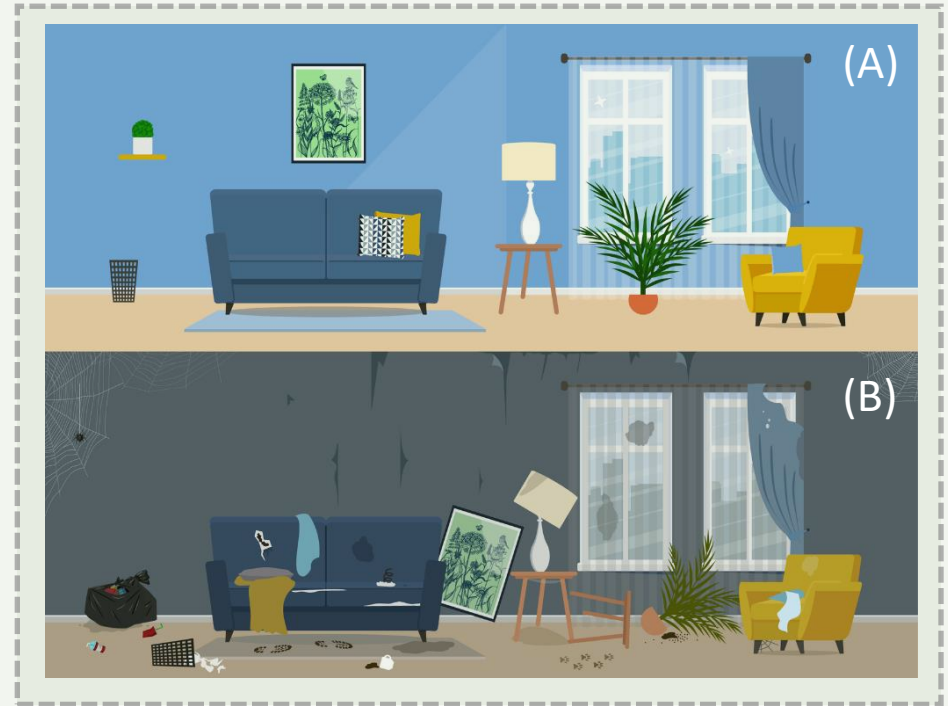
Este ca și cum valorile dintr-un array (sistem de referință comun) sunt atribuite fiecare unei variabile separate.

Unde apare asocierea dintre dezordine și entropie?

Exemplu:

- Când facem ordine într-o cameră, **ALINIEM** seturi de lucruri la un sistem de referință comun!
- Fiind aliniat la un sistem de referință comun, un obiect mai mare este emergent.
- Astfel, mai multe obiecte sunt reduse la un singur obiect mai mare.
- Prin urmare, entropia scade deoarece există mai puține tipuri de obiecte în cameră.
- Reducem numărul de sisteme de referință !
- Reducem entropia !

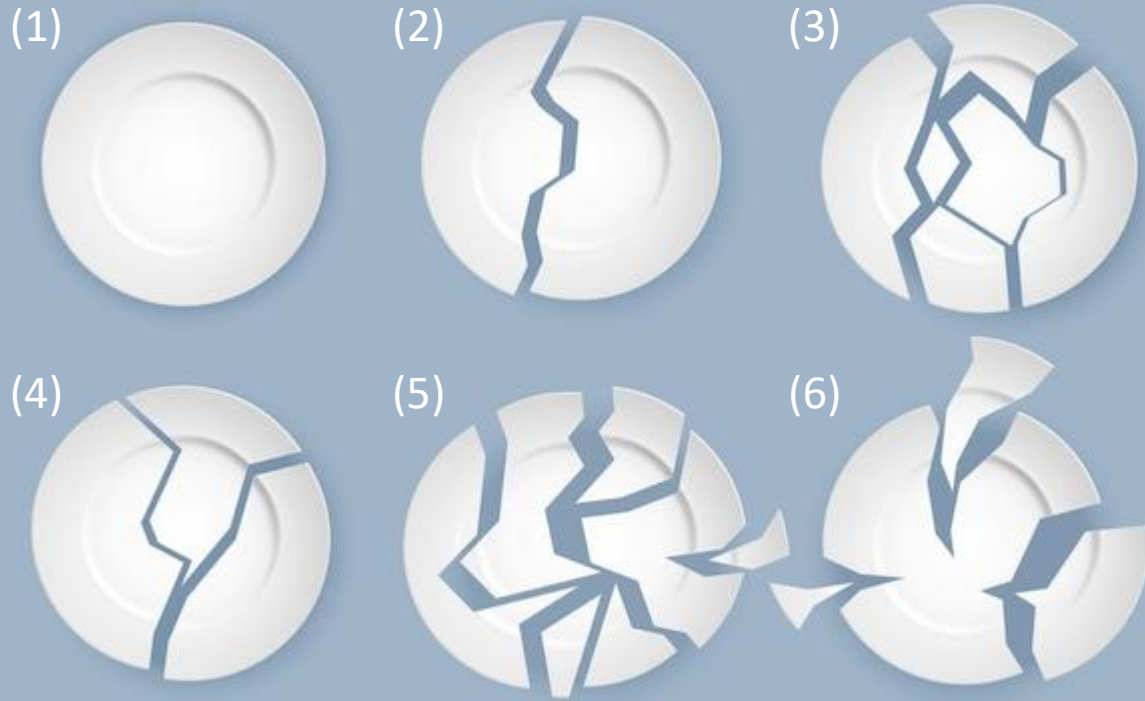
Sistem de referință:
- direcție
- recipient



Codul rău intenționat sau normal se află sub aceeași umbrelă, doar că sistemul de referință ia forme diferite !

Pe scurt ! Entropia si sistemul de referință !

Care dintre următoarele cazuri prezintă cea mai mare valoare a entropiei din punct de vedere **spațial**?



Care dintre următoarele cazuri prezintă cea mai mare valoare a entropiei din punct de vedere al **informatiei**?

Ce valoare a entropiei putem observa din punct de vedere **spațial**?



Ce valoare a entropiei putem observa din punct de vedere al **informatiei**?

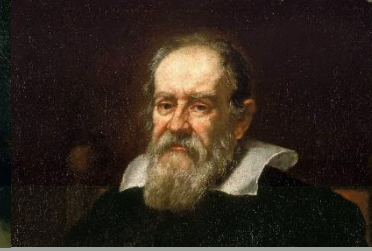
Sistemul de referință contează!

Dă-mi un punct și-ți voi muta pământul.

Dă-mi un loc unde să stau și voi muta pământul.



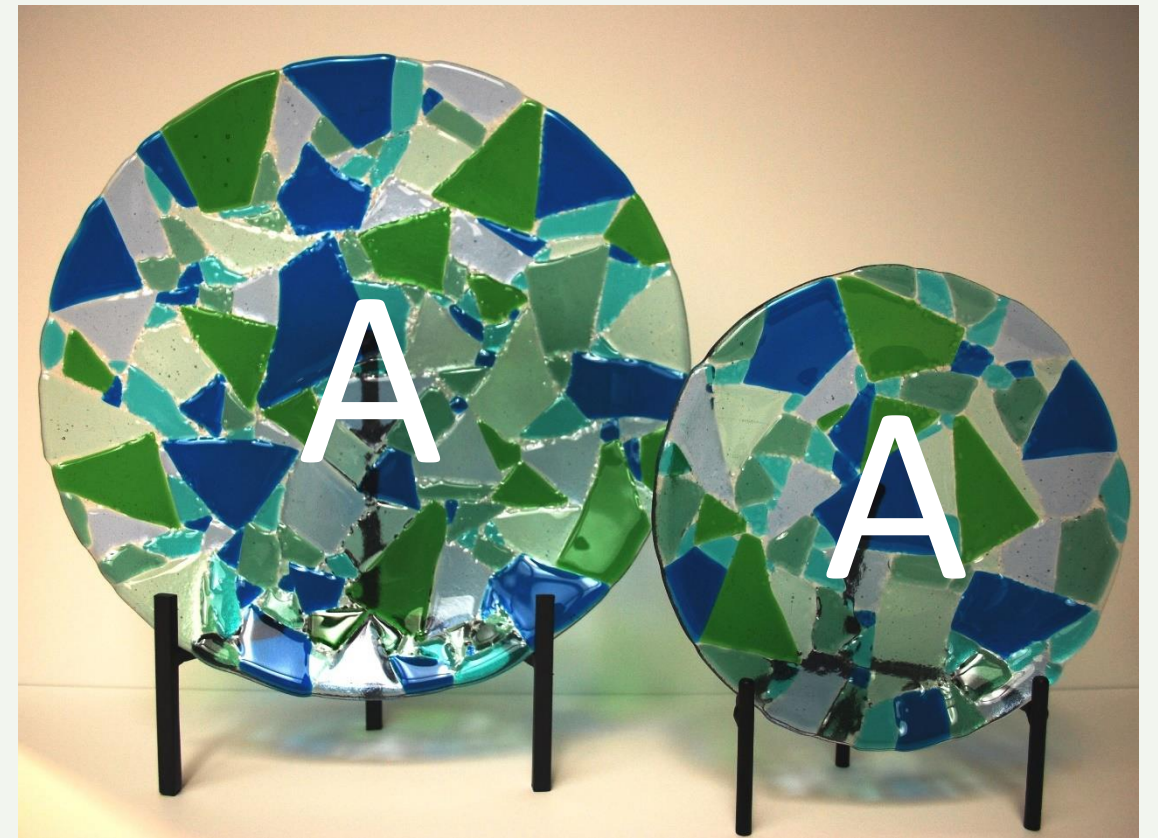
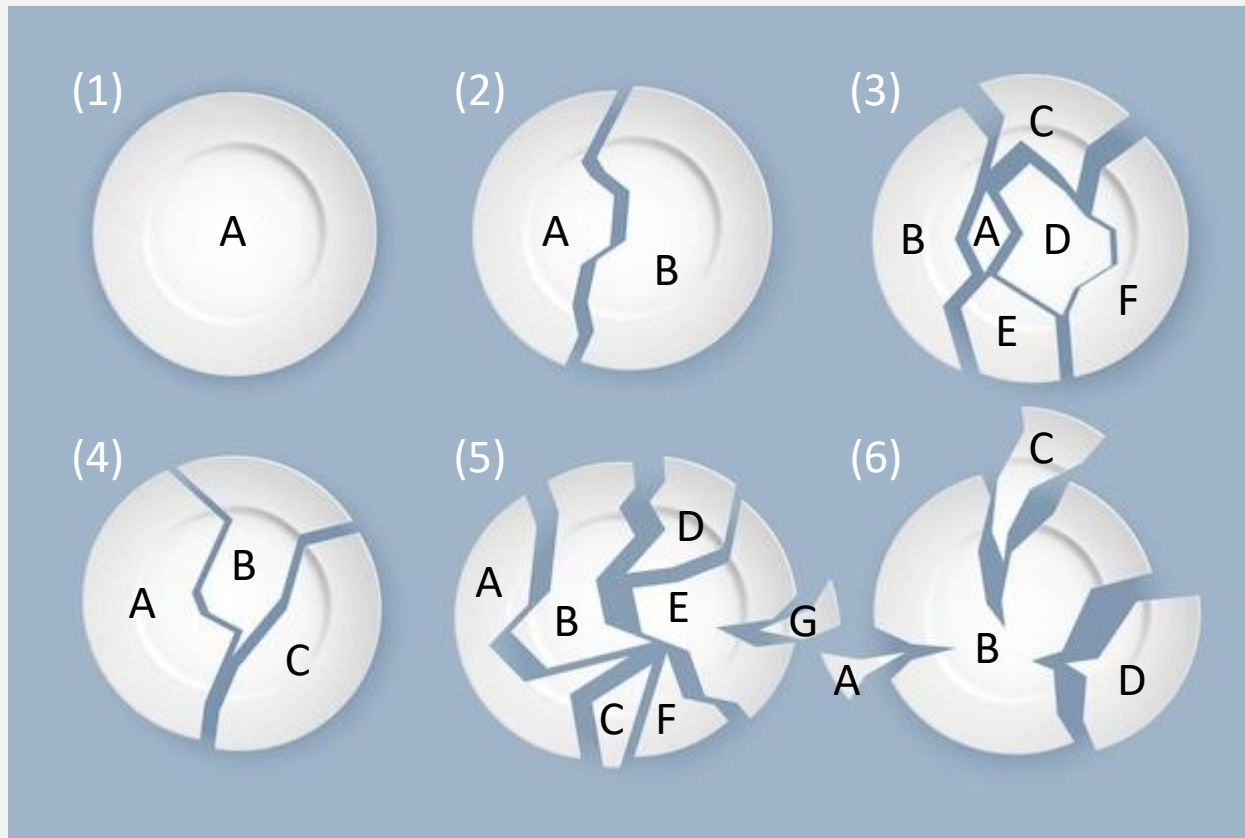
Arhimede



Galileo Galilei



Henri Poincaré



Poate entropia sa fie utilizată în detectarea malware?

Da ! (Aplicatiile malware converg catre o plaja de valori ale entropiei, indiferent de autorul malware)
Convergenta

Poate o detectie bazată pe entropie să fie păcălită?

Da ! (dilutia codului sau criptarea)

Este entropia fiabilă pentru detecție?

C.2.3 CUANTIFICAREA INFORMAȚIEI



Avem o metodă de măsurare a conținutului informațional? Sigur ca da !

```
<script>  
document.write(Sigma("AAAAAASDAAAAAAAAAAAA"));
```

```
function Sigma(s)  
{  
  var t = 0;  
  var m = 0;
```

```
  for (var u=1; u<=(s.length - 1); u++)  
  {  
    for (var i=0; i<=(s.length-u); i++)  
    {  
      m += f(s.substr(i,1),s.substr(u+i,1));  
    }  
    t += (m / (s.length-u) * 100);  
    m = 0;  
  }  
}
```

```
  return (100 - (t / (s.length - 1))).toFixed(2);  
}
```

```
function f(x,y){  
  if (x == y) {  
    return 1;  
  } else {  
    return 0;  
  }  
}
```

```
</script>
```

$$s = \{x_1, \dots, x_{|s|}\}$$

$$\sigma(s) = 100 - \frac{\sum_{u=1}^{|s|-1} \left(\frac{\sum_{i=1}^{|s|-u} f(x_i, x_{u+i})}{(|s| - u) \times 100} \right)}{(|s| - 1)}$$

$$f(x_i, x_{u+i}) = \begin{cases} +1, & x_i = x_{u+i} \\ 0, & x_i \neq x_{u+i} \end{cases}$$

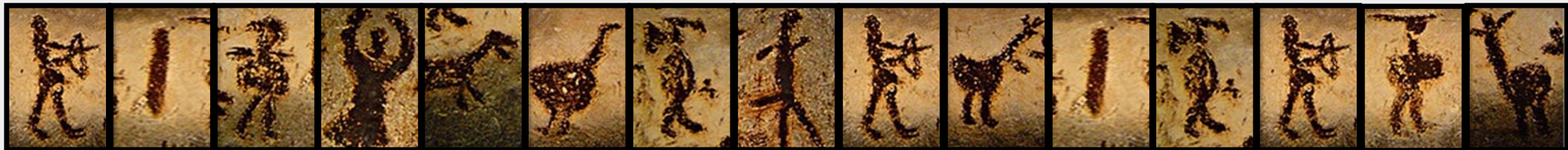
Auto-alinierea de secvente (metodă noua românească) |s|= "ABAAC"

Total steps of u: 4	Self alignment	Location	Match per step
Step u=1: $\sum_{u=1}^4 \frac{\sum_{i=1}^4 f(x_i, x_{u+i})}{(4 \times 100)}$ s =5 s -u=5-1=4	X[i] = ABAA X[u+i] = BAAC ABAA --■- BAAC	ABAA [i] ■■■■ ABAAC ■■■■ BAAC [u+i]	$f(\text{A}, \text{B})=0 \mid u=1 \mid i=1$ $f(\text{B}, \text{A})=0 \mid u=1 \mid i=2$ $f(\text{A}, \text{A})=1 \mid u=1 \mid i=3$ $f(\text{A}, \text{C})=0 \mid u=1 \mid i=4$ $\sum_{i=1}^4 f(x_i, x_{u+i}) = 1$ $\sum_{u=1}^4 \frac{1}{4 \times 100} = 25$
Step u=2: $\sum_{u=2}^3 \frac{\sum_{i=1}^3 f(x_i, x_{u+i})}{(3 \times 100)}$ s =5 s -u=5-2=3	X[i] = ABA X[u+i] = AAC ABA ■-- AAC	ABA [i] ■■■ ABAAC ■■■■ AAC [u+i]	$f(\text{A}, \text{A})=1 \mid u=2 \mid i=1$ $f(\text{B}, \text{A})=0 \mid u=2 \mid i=2$ $f(\text{A}, \text{C})=0 \mid u=2 \mid i=3$ $\sum_{i=1}^3 f(x_i, x_{u+i}) = 1$ $\sum_{u=2}^3 \frac{1}{3 \times 100} = \sim 33$
Step u=3: $\sum_{u=3}^2 \frac{\sum_{i=1}^2 f(x_i, x_{u+i})}{(2 \times 100)}$ s =5 s -u=5-2=2	X[i] = AB X[u+i] = AC AB ■- AC	AB [i] ■■ ABAAC ■■■■ AC [u+i]	$f(\text{A}, \text{A})=1 \mid u=3 \mid i=1$ $f(\text{B}, \text{C})=0 \mid u=3 \mid i=2$ $\sum_{i=1}^2 f(x_i, x_{u+i}) = 1$ $\sum_{u=3}^2 \frac{1}{2 \times 100} = 50$
Step u=4: $\sum_{u=4}^1 \frac{\sum_{i=1}^1 f(x_i, x_{u+i})}{(1 \times 100)}$ s =5 s -u=5-4=1	X[i] = A X[u+i] = C A - C	A [i] ■ ABAAC ■■■■ C [u+i]	$f(\text{A}, \text{C})=0 \mid u=4 \mid i=1$ $\sum_{i=1}^1 f(x_i, x_{u+i}) = 0$ $\sum_{u=4}^1 \frac{0}{1 \times 100} = 0$



A B C D E F G H I J K

Simboluri **unice observate** cu **semnificație necunoscută** reprezentate folosind simboluri ASCII.



B A G I D C F E B H A F B J K

Simboluri cu semnificatie necunoscuta observate in secventa.

Entropie | $ABCDEFGHIJK = 3.45$ |
 $BAGIDCFEBHAFBJK = 3.32$ |

Alfabet din 11 simboluri
 BAGIDCFEBHAFBJK

Cuantificarea informației!

Spre deosebire de entropia *Shannon*, metoda de mai sus ia în considerare ordinea simbolurilor din secvență!

Alfabet din 11 simboluri

BAGIDCFEBHAFBJK

$$e = - \sum_{i=1}^n p_i \times \log_2(p_i)$$

3.32

$$s = \{x_1, \dots, x_{|s|}\}$$

$$\sigma(s) = 100 - \left(\frac{\sum_{u=1}^{|s|-1} \left(\frac{\sum_{i=1}^{|s|-u} f(x_i, x_{u+i})}{(|s| - u) \times 100} \right)}{(|s| - 1)} \right)$$

94.04

$$f(x_i, x_{u+i}) = \begin{cases} +1, & x_i = x_{u+i} \\ 0, & x_i \neq x_{u+i} \end{cases}$$

Normalizarea!

Metodele cu intervale de valori diferite trebuie normalizate pentru a fi comparate:

Secvență observată:
BAGIDCFEBHAFBJK = 3.32

$$e = - \sum_{i=1}^n p_i \times \log_2(p_i)$$

$$N(e) = \left(\frac{100}{\max(e)} \right) \times e = \left(\frac{100}{3.45} \right) \times 3.32 = 28.9 \times 3.32 = 96.23$$

Alfabet secvență (11):
ABCDEFGHIIJK = 3.45

$$s = \{x_1, \dots, x_{|s|}\}$$
$$\sigma(s) = 100 - \left(\frac{\sum_{u=1}^{|s|-1} \left(\frac{\sum_{i=1}^{|s|-u} f(x_i, x_{u+i})}{(|s| - u) \times 100} \right)}{(|s| - 1)} \right)$$

$$f(x_i, x_{u+i}) = \begin{cases} +1, & x_i = x_{u+i} \\ 0, & x_i \neq x_{u+i} \end{cases}$$

$$N(\sigma(s)) = \sigma(s) = 94.04$$

(deja normalizat)

Notăți că valoarea „100” este sistemul de referință folosit aici, însă sistemul de referință poate fi orice valoare numerică.

Sistemul de referință „100” ne permite un interval de la 0 la 100 pentru ambele metode.

Experiment 1 – Alfabet = “AB” [2 chr]

Auto-aliniere:

- AAAAAAABBBBBBBB = 70.94
- ABABABABABABAB = 53.85
- ABBBBBBBBBBBBBBA = 33.54



Se ține cont de ordinea
obiectelor din secvență.

Entropie: (normalizat)

- AAAAAAABBBBBBBB = 1 = 100
- ABABABABABABAB = 1 = 100
- ABBBBBBBBBBBBBBA = 0.59 = 59



NU se ține cont de ordinea
obiectelor din secvență.

Experiment 2 – Alfabet = “ABC” [15 chr]

Auto-aliniere:

- AAAAABBBBBBCCCCC = 83.41
- ABCABCABCABCABC = 71.43
- ABBBBBBCBBBBBBBA = 39.79

Entropie:

(normalizat)

- AAAAABBBBBBCCCCC = 1.58 = 100
- ABCABCABCABCABC = 1.58 = 100
- ABBBBBBCBBBBBBBA = 0.90 = 56.96

Fișiere PE (Portable Executable) executabile! Care este entropia maximă care poate fi observată?

$$e = - \sum_{i=1}^{n=256} p_i \times \log_2(p_i)$$

$$e = - \sum_{i=1}^{n=256} \frac{1}{256} \times \log_2\left(\frac{1}{256}\right) = 8$$

Observam ca 256 simboluri (bytes) diferite pot fi codate intr-un **POTENTIAL** de 8 biti.

C.2.4

DIMENSIUNI ȘI SISTEME DE REFERINȚĂ

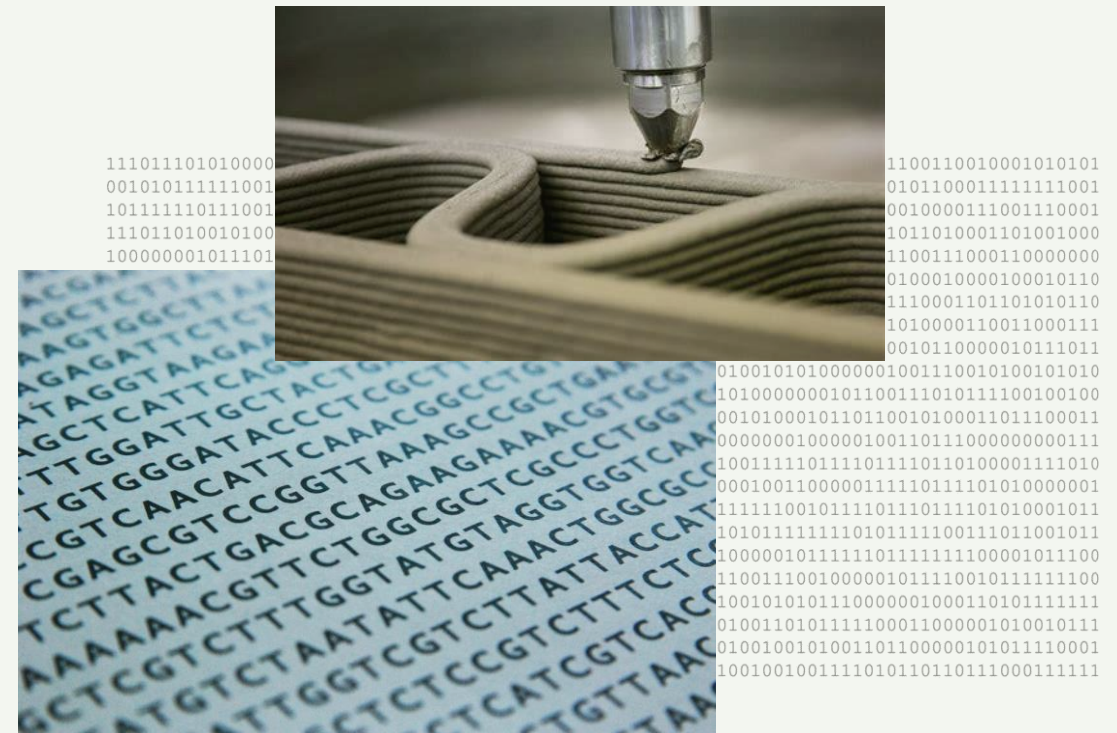


PUTEM MĂSURA INFORMAȚIA ÎN STRUCTURI N-DIMENSIONALE?

STRUCTURI DE DATE

- Toate structurile n -dimensionale sunt construite din structuri unidimensionale.
- **Exemplul 1:** O matrice este construită dintr-o succesiune de elemente pliate în n straturi.
- **Exemplul 2:** O imprimantă 3D folosește un fir de plastic pentru a construi structuri 3D (firul de plastic fiind asociat aici cu o secvență unidimensională).
- **Exemplul 3:** Omul este rezultatul unei informații unidimensionale, și anume, secvența ADN.
- **Exemplul 4:** Toate programele de calculator sunt unidimensionale.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}; [1 \quad 2 \quad 3 \quad 4]$$



SISTEMUL DE REFERINȚĂ

DISCRIMINARE OPTIMA

- Sistemul de referință este secretul tuturor tipurilor de discriminare optima!
(Uniformă - Armată; Sfânta Biblie - Creștinismul;)
- Normalizare (se face in functie de un sistem de referință)
- Învățare automată (sistem de referință – observare/așteptare)
- Semnificație (sistemul de referință – ajută la găsirea sensului)

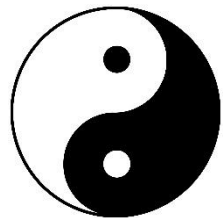
SISTEMUL DE REFERINȚĂ

PIESA DE REZISTENȚĂ

- Cum se leagă ceea ce am discutat până acum cu detectarea codului malițios?
- Mostre de fișiere normale (sistem de referință)
- Mostre de fișiere care conțin cod malițios.

CONCLUZII

- Simboluri preistorice.
- Abordarea unui mesaj necunoscut.
- Entropia nu măsoară informația.
- Cuantificarea conținutului informațional.
- Sistemul de referință și detecția de cod malițios.



Ătălia modelelor ! yin-yang !

- Cum facem ingineria inversă a unui sistem informatic necunoscut?
- Ce facem când găsim tehnologie militară cu hardware care nu seamănă cu nimic din ce știm?
- Care este punctul de plecare pentru descifrarea unei limbi străine?
- Cum facem diferența între malware și codul normal?
- Self sau Non-Self?

INTRODUCERE PRACTICĂ ÎN INGINERIE INVERSĂ

(3 FAZE):

- **INSTALAREA ȘI ÎNȚELEGEREA MEDIULUI DE DETONARE**
- **INSTALAREA ȘI ÎNȚELEGEREA INSTRUMENTELOR DE INGINERIE INVERSĂ**
- **DESCĂRCARE MOSTRE MALWARE & PROTOCOALE DE MANIPULARE**

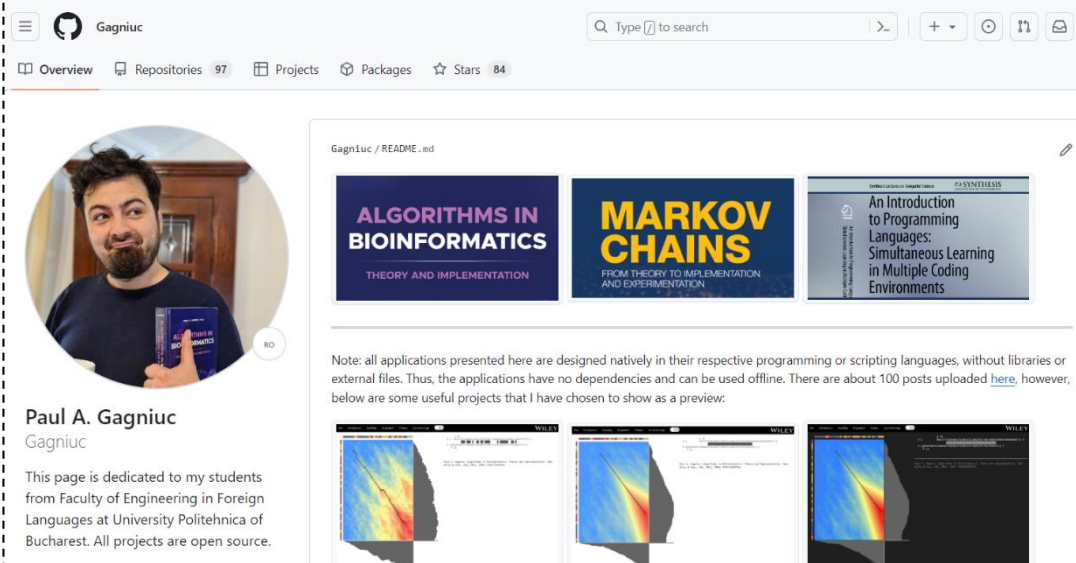
Detonare: Rularea unei aplicații malware (compromitere intenționată a sistemului de operare).

Malware : Termenul „malware” a fost folosit pentru prima dată de *Yisrael Radai* în 1990.

BIBLIOGRAFIE / RESURSE

- Paul A. Gagniuc. *Antivirus Engines: From Methods to Innovations, Design, and Applications*. Cambridge, MA: Elsevier Syngress, 2024. pp. 1-656.
- Paul A. Gagniuc. *An Introduction to Programming Languages: Simultaneous Learning in Multiple Coding Environments. Synthesis Lectures on Computer Science*. Springer International Publishing, 2023, pp. 1-280.
- Paul A. Gagniuc. *Coding Examples from Simple to Complex - Applications in MATLAB*, Springer, 2024, pp. 1-255.
- Paul A. Gagniuc. *Coding Examples from Simple to Complex - Applications in Python*, Springer, 2024, pp. 1-245.
- Paul A. Gagniuc. *Coding Examples from Simple to Complex - Applications in Javascript*, Springer, 2024, pp. 1-240.
- Paul A. Gagniuc. *Markov chains: from theory to implementation and experimentation*. Hoboken, NJ, John Wiley & Sons, USA, 2017, ISBN: 978-1-119-38755-8.

<https://github.com/gagniuc>



Gagniuc

Overview Repositories 97 Projects Packages Stars 84

Gagniuc / README.md

ALGORITHMS IN BIOINFORMATICS
THEORY AND IMPLEMENTATION

MARKOV CHAINS
FROM THEORY TO IMPLEMENTATION AND EXPERIMENTATION

An Introduction to Programming Languages: Simultaneous Learning in Multiple Coding Environments

Note: all applications presented here are designed natively in their respective programming or scripting languages, without libraries or external files. Thus, the applications have no dependencies and can be used offline. There are about 100 posts uploaded [here](#), however, below are some useful projects that I have chosen to show as a preview.

Paul A. Gagniuc
Gagniuc

This page is dedicated to my students from Faculty of Engineering in Foreign Languages at University Politehnica of Bucharest. All projects are open source.