Podstawy Teorii Informacji

Wykład 1
Wprowadzenie

Wojciech Kordecki
wojciech.kordecki@collegiumwitelona.pl

Collegium Witelona
Wydział Nauk Technicznych i Ekonomicznych
Zakład Informatyki

Semestr zimowy 2025/26

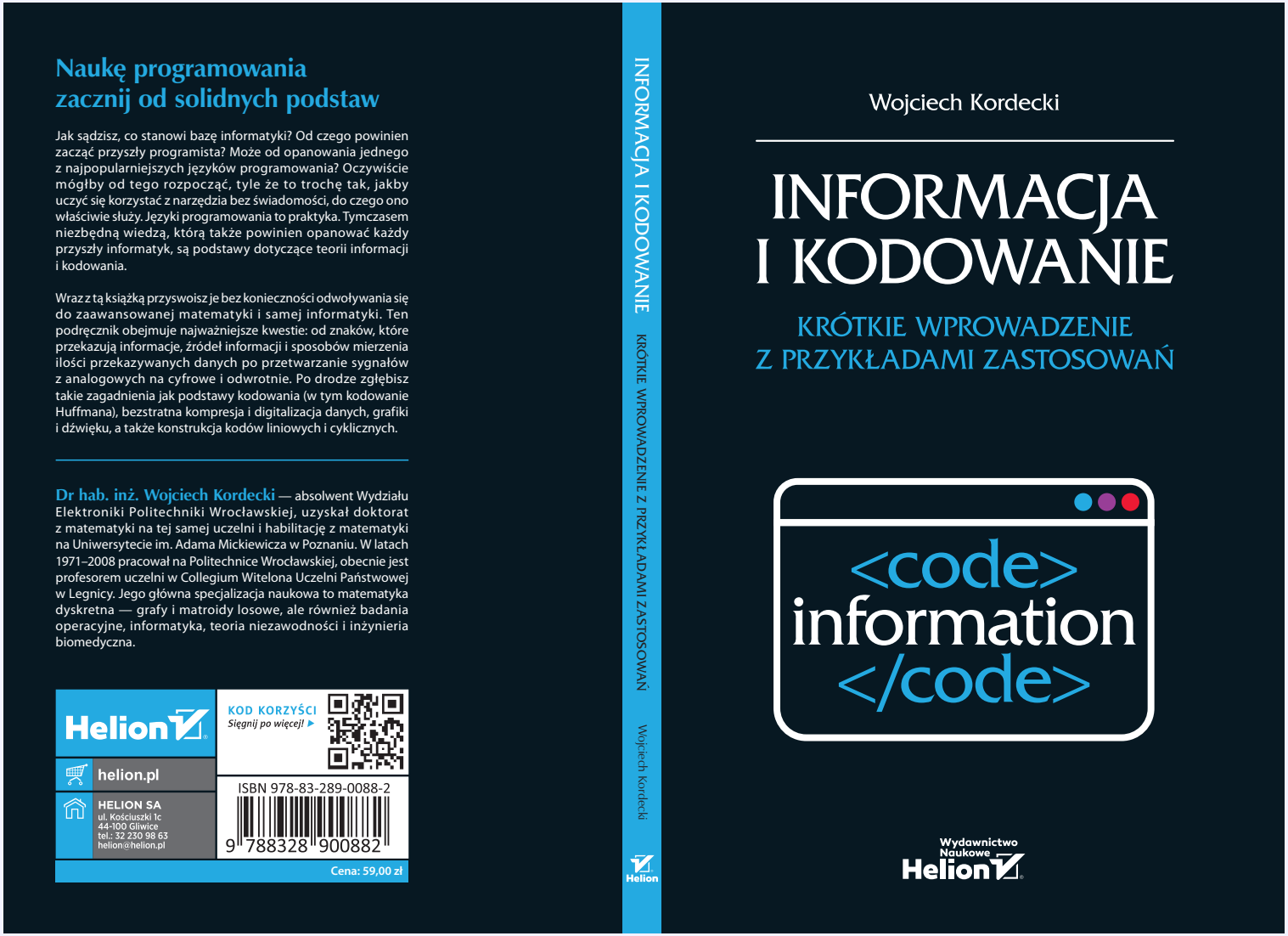
Zagadnienia poruszane na wykładzie

1. Informacja i jej źródła.
2. Entropia. Kody prefiksowe. Kod Huffmana.
3. Kodowanie wiadomości. Kody wykrywające i poprawiające błędy. Kody Hamminga.
4. Kody liniowe. Kody cykliczne i wielomiany. Kody dualne. Kody CRC.
5. Przetwarzanie sygnałów. Sygnały analogowe i cyfrowe.
6. Kompresja danych. Kodowanie multimediów. Formaty audio i wideo.
7. Macierze generujące kody.

Literatura podstawowa

- W. Kordecki, *Informacja i kodowanie*, Helion 2024.
- N. Abramson, *Teoria informacji i kodowania*, PWN 1969.
- I. Bułatowa. *Ćwiczenia z przedmiotu Wprowadzenie do informatyki. Kody liczbowe*, Oficyna Wydawnicza Politechniki Białostockiej, Białystok 2022.
- J. G. Brookshear, D. Brylow, *Informatyka w ogólnym zarysie*, PWN, Warszawa 2022.

Informacja i kodowanie [IK]



Literatura uzupełniająca


A. Drozdek, *Wprowadzenie do kompresji danych*, Wydawnictwa Naukowo-Techniczne, Warszawa 1999.


D. Karwowski, *Zrozumieć kompresję obrazu*, Poznań 2019. https://www.zrozumieckompresje.pl/ (dostęp 10.01.2024).


P. Przybyłowicz, Wstęp do teorii informacji i kodowania, 2008. https://www.scribd.com/document/49766365/wstep-do-teorii-informacji (dostęp 10.01.2024).

Bity i bajty [IK – 1.1]

Bit (z ang. kawałek, także skrót od *binary digit*) – najmniejsza ilość informacji potrzebna do określenia, który z dwóch stanów przyjął układ.
1 bit – oznaczenie 1b.

Bity i bajty [IK – 1.1]

Bit (z ang. kawałek, także skrót od *binary digit*) – najmniejsza ilość informacji potrzebna do określenia, który z dwóch stanów przyjął układ.
1 bit – oznaczenie 1b.

Dwa stany.

Bity i bajty [IK – 1.1]

Bit (z ang. kawałek, także skrót od *binary digit*) – najmniejsza ilość informacji potrzebna do określenia, który z dwóch stanów przyjął układ.
1 bit – oznaczenie 1b.

Dwa stany.

- Zero i jeden: 0, 1.

Bity i bajty [IK – 1.1]

Bit (z ang. kawałek, także skrót od *binary digit*) – najmniejsza ilość informacji potrzebna do określenia, który z dwóch stanów przyjął układ.
1 bit – oznaczenie 1b.

Dwa stany.

- Zero i jeden: 0, 1.
- Fałsz i prawda: F, T.

Bity i bajty [IK – 1.1]

Bit (z ang. kawałek, także skrót od *binary digit*) – najmniejsza ilość informacji potrzebna do określenia, który z dwóch stanów przyjął układ.
1 bit – oznaczenie 1b.

Dwa stany.

- Zero i jeden: 0, 1.
- Fałsz i prawda: F, T.
- Napięcie: stan niski i wysoki, np. 0 – 0.8V, 2.4 – 5V (TTL – Transistor-transistor logic).

Bity i bajty [IK – 1.1]

Bit (z ang. kawałek, także skrót od *binary digit*) – najmniejsza ilość informacji potrzebna do określenia, który z dwóch stanów przyjął układ.
1 bit – oznaczenie 1b.

Dwa stany.

- Zero i jeden: 0, 1.
- Fałsz i prawda: F, T.
- Napięcie: stan niski i wysoki, np. 0 – 0.8V, 2.4 – 5V (TTL – Transistor-transistor logic).
- Częstotliwość: mała i duża.

Bity i bajty [IK – 1.1]

Bit (z ang. kawałek, także skrót od *binary digit*) – najmniejsza ilość informacji potrzebna do określenia, który z dwóch stanów przyjął układ.
1 bit – oznaczenie 1b.

Dwa stany.

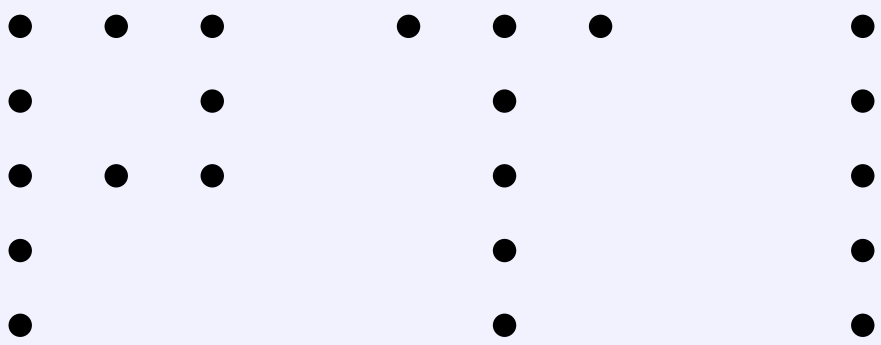
- Zero i jeden: 0, 1.
- Fałsz i prawda: F, T.
- Napięcie: stan niski i wysoki, np. 0 – 0.8V, 2.4 – 5V (TTL – Transistor-transistor logic).
- Częstotliwość: mała i duża.
- Ciśnienie: małe i duże.

Najprostsze kodowanie

1 bajt – oznaczenie 1B. 1 bajt, to 8 bitów.
Kodowanie informacji w postaci ciągu bitów – mapa bitowa:

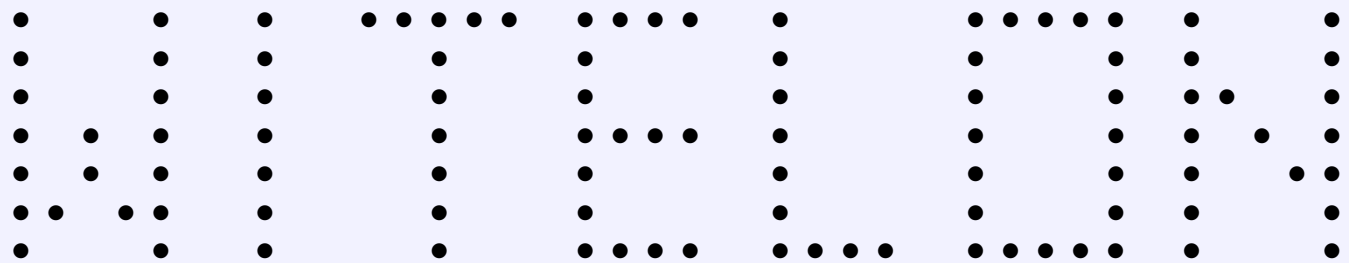
Najprostsze kodowanie

1 bajt – oznaczenie 1B. 1 bajt, to 8 bitów.
Kodowanie informacji w postaci ciągu bitów – mapa bitowa:



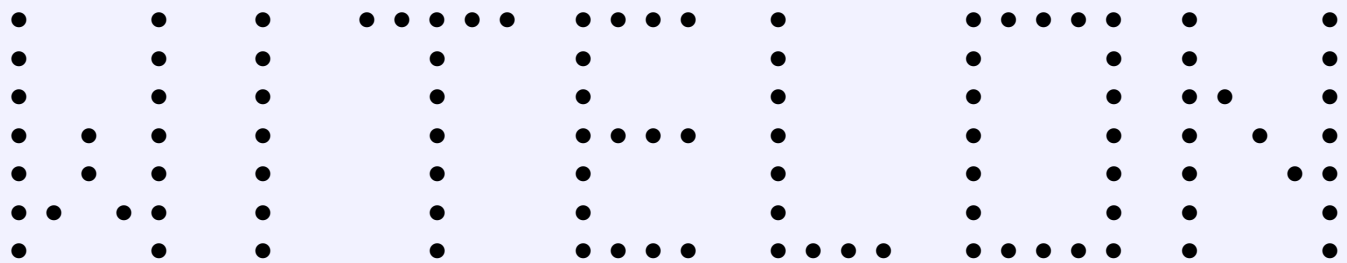
Jedna litera, to 15 bitów. Trzyliterowy napis – 45 bitów.

Najprostsze kodowanie



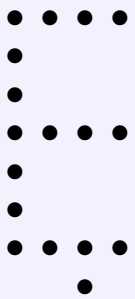
Jedna litera, to 35 bitów lub 5 bajtów. Siedmioliterowy napis – 35 bajtów.

Najprostsze kodowanie

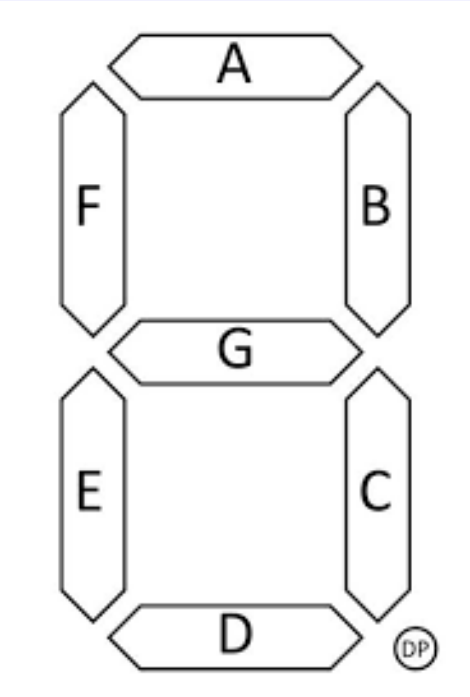


Jedna litera, to 35 bitów lub 5 bajtów. Siedmioliterowy napis – 35 bajtów.

Problem z polskimi literami.



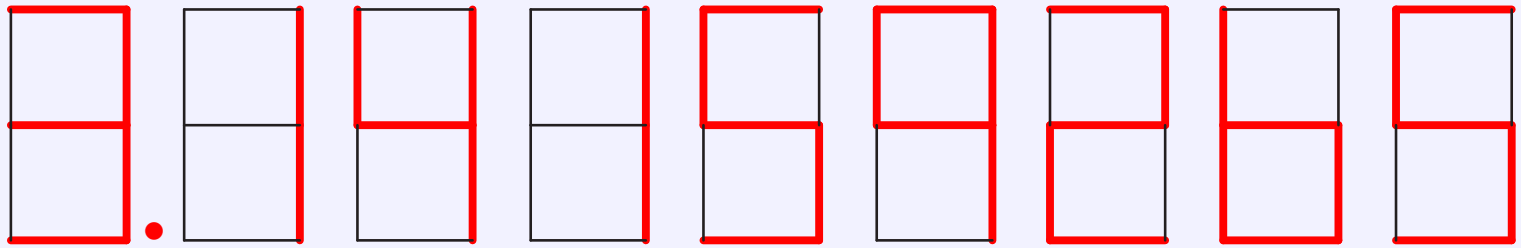
Wyświetlacz 7-segmentowy



cyfra	kod
0	0111111
1	0000110
2	1011011
3	1001111
4	1100110
5	1101101
6	1111100
7	0000111
8	1111111
9	1100111

Wyświetlacz 7-segmentowy c.d.

Liczba $\pi \approx 3.14159265$



Systemy pozycyjne [IK] – 1.2, str. 10

Podstawa systemu $p \geq 2$ jest liczbą całkowitą.
Niech $x_i \in [0, \ldots, p-1]$.

Systemy pozycyjne [IK] – 1.2, str. 10

Podstawa systemu $p \geq 2$ jest liczbą całkowitą.
Niech $x_i \in [0, \ldots, p-1]$.
Nieujemną liczbę całkowitą N można jednoznacznie przedstawić w postaci

$$N = x_n p^n + x_{n-1} p^{n-1} + \cdots + x_2 p^2 + x_1 p + x_0,$$

gdzie N jest liczbą całkowitą nieujemną, $0 \leq x_i \leq p-1$.

Systemy pozycyjne [IK] – 1.2, str. 10

Podstawa systemu $p \geq 2$ jest liczbą całkowitą.
Niech $x_i \in [0, \ldots, p-1]$.
Nieujemną liczbę całkowitą N można jednoznacznie przedstawić w postaci

$$N = x_n p^n + x_{n-1} p^{n-1} + \cdots + x_2 p^2 + x_1 p + x_0,$$

gdzie N jest liczbą całkowitą nieujemną, $0 \leq x_i \leq p-1$.
Najbardziej znane są systemy:

- dziesiętny, $p = 10$,
- dwójkowy (binarny), $p = 2$,
- szesnastkowy (hex), $p = 16$.

Liczby dwójkowo

$$
\begin{array}{ccccccccccc}
n & & n-1 & & \dots & & 2 & & 1 & & 0 \\
\downarrow & & \downarrow & & \dots & & \downarrow & & \downarrow & & \downarrow \\
x_n & & x_{n-1} & & \dots & & x_2 & & x_1 & & x_0 \\
x_n 2^n & + & x_{n-1} 2^{n-1} & + & \dots & + & x_2 2^2 & + & x_1 2 & + & x_0
\end{array}
$$

Liczby dwójkowo

$$\begin{array}{ccccccccccc}
n & & n-1 & & \dots & & 2 & & 1 & & 0 \\
\downarrow & & \downarrow & & \dots & & \downarrow & & \downarrow & & \downarrow \\
x_n & & x_{n-1} & & \dots & & x_2 & & x_1 & & x_0 \\
x_n 2^n & + & x_{n-1} 2^{n-1} & + & \dots & + & x_2 2^2 & + & x_1 2 & + & x_0
\end{array}$$

Zapis

$$(x_n x_{n-1} \dots x_2 x_1 x_0)_2 = x_n 2^n + x_{n-1} 2^{n-1} + \dots + x_2 2^2 + x_1 2 + x_0 \,.$$

Liczby dwójkowo

$$\begin{array}{ccccccccccc} n & & n-1 & & \dots & & 2 & & 1 & & 0 \\ \downarrow & & \downarrow & & \dots & & \downarrow & & \downarrow & & \downarrow \\ x_n & & x_{n-1} & & \dots & & x_2 & & x_1 & & x_0 \\ x_n 2^n & + & x_{n-1} 2^{n-1} & + & \dots & + & x_2 2^2 & + & x_1 2 & + & x_0 \end{array}$$

Zapis

$$(x_n x_{n-1} \dots x_2 x_1 x_0)_2 = x_n 2^n + x_{n-1} 2^{n-1} + \dots + x_2 2^2 + x_1 2 + x_0 \,.$$

Przykład.

$$(100101)_2 = 2^5 + 2^2 + 2^0 = 32 + 4 + 1 = 37.$$

Kod szesnastkowy – hex

Cyfry: 1,. . . 9,A,B,C,D,E,F.

bin	0000	0001	0010	0011	0100	0101	0110	0111
dec	0	1	2	3	4	5	6	7
hex	0	1	2	3	4	5	6	7
bin	1000	1001	1010	1011	1100	1101	1110	1111
dec	8	9	10	11	12	13	14	15
hex	8	9	A	B	C	D	E	F

Przykład.

$$(10110011)_2 = (B3)_{16} = 16 \cdot 11 + 3 = 179.$$

Bajt – znak

1 bajt = 8 bitów = 2 cyfry szesnastkowe. Jeden bajt pozwala przedstawić liczby w zakresie [0...FF], czyli $[0 \ldots 2^8 - 1] = [0 \ldots 255]$.

Bajt – znak

1 bajt = 8 bitów = 2 cyfry szesnastkowe. Jeden bajt pozwala przedstawić liczby w zakresie [0...FF], czyli $[0 \ldots 2^8 - 1] = [0 \ldots 255]$.

Jeden bajt = jeden znak (litera)

Bajt – znak

1 bajt = 8 bitów = 2 cyfry szesnastkowe. Jeden bajt pozwala przedstawić liczby w zakresie [0...FF], czyli $[0 \ldots 2^8 - 1] = [0 \ldots 255]$.

Jeden bajt = jeden znak (litera)

Kod ASCII: https://www.promotic.eu/pl/pmdoc/Directions/TableASCII.htm

Hex	30	...	39	41	...	5A	61	...	7A
Znak	0	...	9	A	...	Z	a	...	z

Pozostałe numery na inne znaki takie jak , . ( ) ! ? itd. oraz na znaki niedrukowalne jak spacja, tabulator, znak nowego wiersza (LF), powrót do początku wiersza (CR) itd.

Kodowanie ASCII tekstu

Napis PTI w kodzie ASCII jest ciągiem

$$\text{PTI} \iff 50\ 54\ 49 \iff 01010000\ 01010100\ 01001001$$

co daje 24 bity.

Polskie litery

	ą	ć	ę	ł	ń	ó	ś	ź	ż
ISO 8859-2	B1	E6	EA	B3	F1	F3	B6	BC	BF
CP 1250	B9	E6	EA	B3	F1	F3	9C	9F	BF
CP 852	A5	86	A9	88	E4	A2	98	AB	BE

	Ą	Ć	Ę	Ł	Ń	Ó	Ś	Ź	Ż
ISO 8859-2	A1	C6	CA	A3	D1	D3	A6	AC	AF
CP 1250	A5	C6	CA	A3	D1	D3	8C	8F	AF
CP 852	A4	8F	A8	9D	E3	E0	97	8D	BD

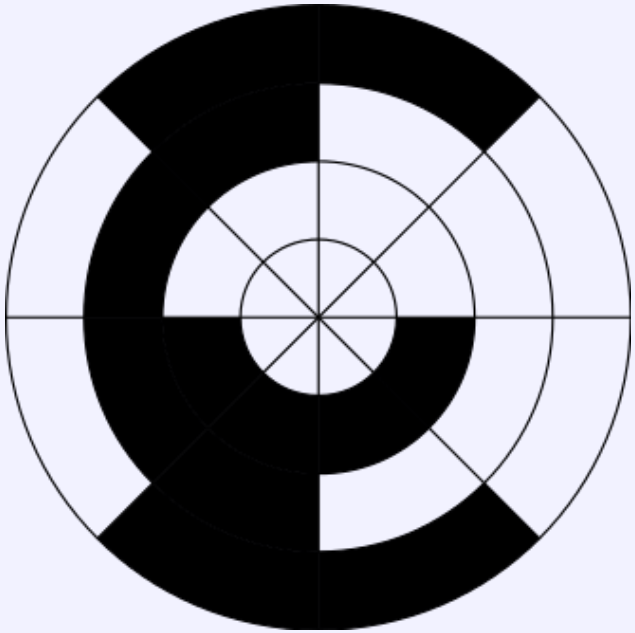
Kod Graya [IK] – 1.2, str. 17

Kod dający 2^n pozycji obrotu. Przy zmianie pozycji o 1, zmiana tylko jednego bitu.

Kod Graya [IK] – 1.2, str. 17

Kod dający 2^n pozycji obrotu. Przy zmianie pozycji o 1, zmiana tylko jednego bitu.
Na rysunku tarcza mierząca 8 pozycji, czyli z dokładnością do 45°.

0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100

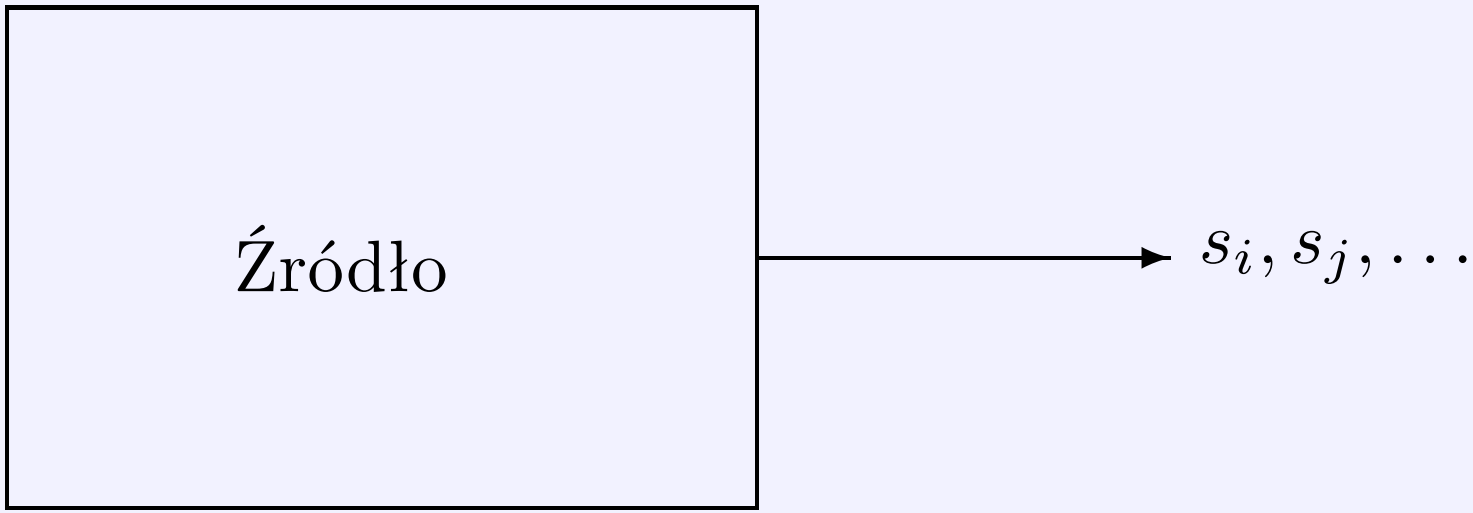


Człowiek pogryzł psa

Magnat prasowy z przełomu XVIII i XIX wieku – Alfred Harmsworth, uznawany jest za autora nagłówka idealnego: „Man bites dog" czyli „Człowiek pogryzł psa". Historia, która ma się „sprzedać" nie może być, zdaniem Brytyjczyków, zwyczajna. „Wściekły pitbull rzucił się do gardła właścicielowi" – nuda. już „Man bites dog and escapes jail" („Mężczyzna pogryzł psa i uniknął więzienia") – przyciąga uwagę. Choć nie jest to wiedza tajemna dostępna tylko dziennikarzom na Wyspach, to właśnie tam „gutter press", czyli "brukowce" święcą największe trumfy.

Źródło wiadomości dyskretnej [IK] – 2.1

Model źródła wiadomości dyskretnej:



Źródło wiadomości – obiekt, który ze skończonego ustalonego zbioru wiadomości elementarnych

$$S = \{s_1, s_2, \ldots, s_n\}$$

tworzy ciąg wiadomości elementarnych.

Telegraf Morse'a [IK] – 1.2, str. 9

Źródło:

$$S = \{s_1, s_2, s_3\}$$

gdzie:

- s_1 – kropka
- s_2 – kreska
- s_3 – odstęp

kropka, kreska, odstęp

Samuel Finley Breese Morse

ur. 27 kwietnia 1791, zm. 2 kwietnia 1872.

Amerykański wynalazca, malarz marynista, rzeźbiarz, prekursor telegrafii. Jeden z twórców alfabetu Morse'a.
W latach sześćdziesiątych XIX wieku Morse zasłynął jako aktywny obrońca instytucji niewolnictwa. Uważał je za zgodne z wolą boską.

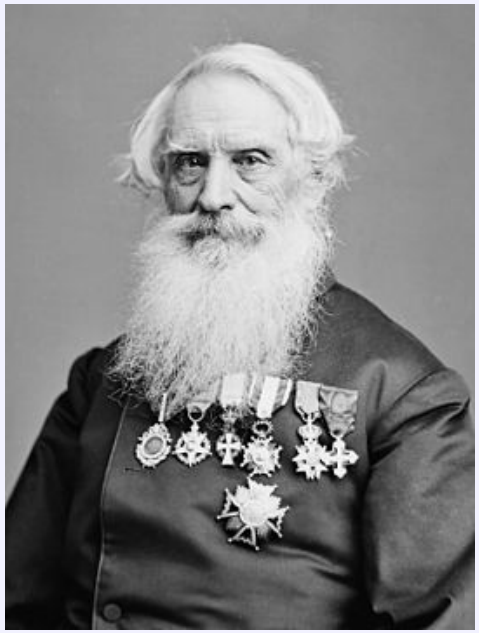
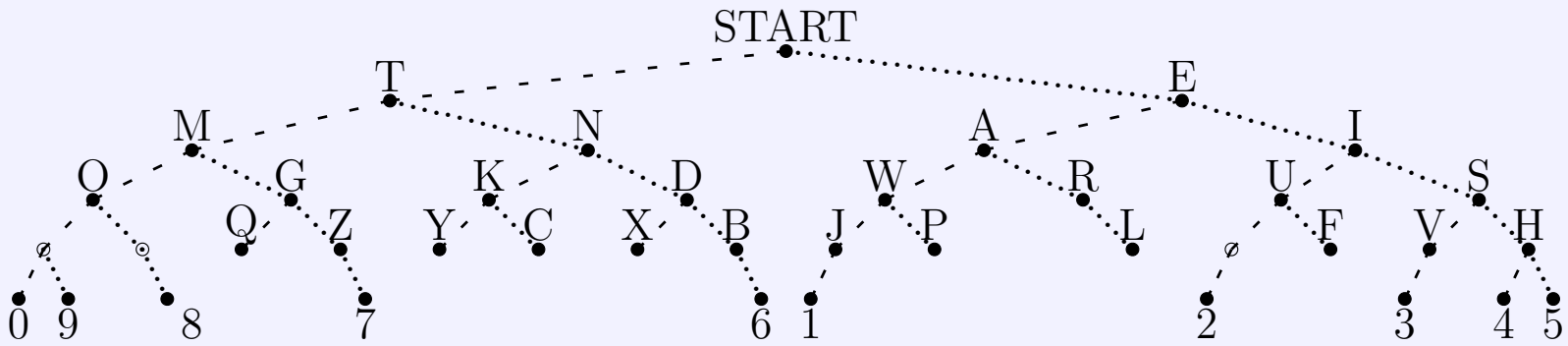


Diagram odczytu



Jest to drzewo binarne – będzie na *Matematyce Dyskretnej* w przyszłym semestrze.

Człowiek pogryzł psa c.d. [IK] – 1.3

Rozważmy zdanie „x pogryzł y”.
Jeśli oszacujemy prawdopodobieństwo p jako stosunek liczby zdarzeń pies pogryzł człowieka do liczby wszystkich pogryzień człowiek – pies, to p jest liczbą bardzo małą.
Oznacza to, że zdarzenie „człowiek pogryzł psa” jest bardzo ciekawe dla szukającego sensacji czytelnika tabloidu, czyli niesie dużo informacji.
Zdarzenie przeciwne „pies pogryzł człowieka” jest z punktu widzenie tego czytelnika nieinteresujące – nie niesie prawie żadnej informacji.

Człowiek pogryzł psa c.d. [IK] – 1.3

Rozważmy zdanie „x pogryzł y”.
Jeśli oszacujemy prawdopodobieństwo p jako stosunek liczby zdarzeń pies pogryzł człowieka do liczby wszystkich pogryzień człowiek – pies, to p jest liczbą bardzo małą.
Oznacza to, że zdarzenie „człowiek pogryzł psa” jest bardzo ciekawe dla szukającego sensacji czytelnika tabloidu, czyli niesie dużo informacji.
Zdarzenie przeciwne „pies pogryzł człowieka” jest z punktu widzenie tego czytelnika nieinteresujące – nie niesie prawie żadnej informacji.
Zdarzenie mało prawdopodobne niesie więcej informacji niż zdarzenie bardzo prawdopodobne.

Logarytm [IK] – A.1

Definicja logarytmu:

$$\log_r x = y \iff r^y = x.$$

Liczba $r > 0$ i $r \neq 1$ jest podstawą logarytmu.

Logarytm [IK] – A.1

Definicja logarytmu:

$$\log_r x = y \iff r^y = x.$$

Liczba $r > 0$ i $r \neq 1$ jest podstawą logarytmu.
Tutaj zawsze będziemy zakładać, że $r > 1$, a najczęściej, $r = 2$.

Logarytm [IK] – A.1

Definicja logarytmu:

$$\log_r x = y \iff r^y = x.$$

Liczba $r > 0$ i $r \neq 1$ jest podstawą logarytmu.
Tutaj zawsze będziemy zakładać, że $r > 1$, a najczęściej, $r = 2$.

Przykład. Ile potrzeba bitów, aby zakodować 16 wartości?
Odpowiedź: $\log_2 16 = 4$, bo $2^4 = 16$.

Logarytmy o podstawie 2

$$\log_2 1 = \log_2 (00001)_2 = 0$$
$$\log_2 2 = \log_2 (00010)_2 = 1$$
$$\log_2 4 = \log_2 (00100)_2 = 2$$
$$\log_2 8 = \log_2 (01000)_2 = 3$$
$$\log_2 16 = \log_2 (10000)_2 = 4$$

Logarytmy o podstawie 2

$$\log_2 1 = \log_2 (00001)_2 = 0$$
$$\log_2 2 = \log_2 (00010)_2 = 1$$
$$\log_2 4 = \log_2 (00100)_2 = 2$$
$$\log_2 8 = \log_2 (01000)_2 = 3$$
$$\log_2 16 = \log_2 (10000)_2 = 4$$

$$\log \frac{1}{x} = -\log x.$$

Logarytmy o podstawie 2

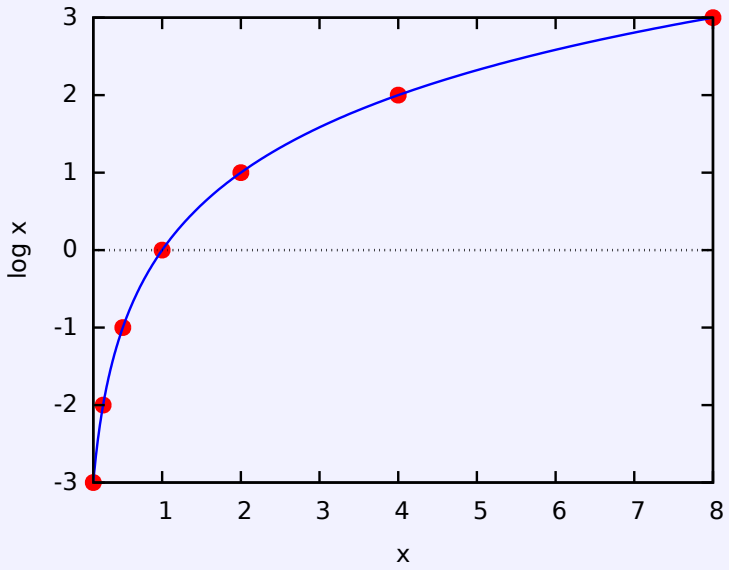
$$\log_2 1 = \log_2 (00001)_2 = 0$$
$$\log_2 2 = \log_2 (00010)_2 = 1$$
$$\log_2 4 = \log_2 (00100)_2 = 2$$
$$\log_2 8 = \log_2 (01000)_2 = 3$$
$$\log_2 16 = \log_2 (10000)_2 = 4$$

$$\log \frac{1}{x} = -\log x.$$

Stąd

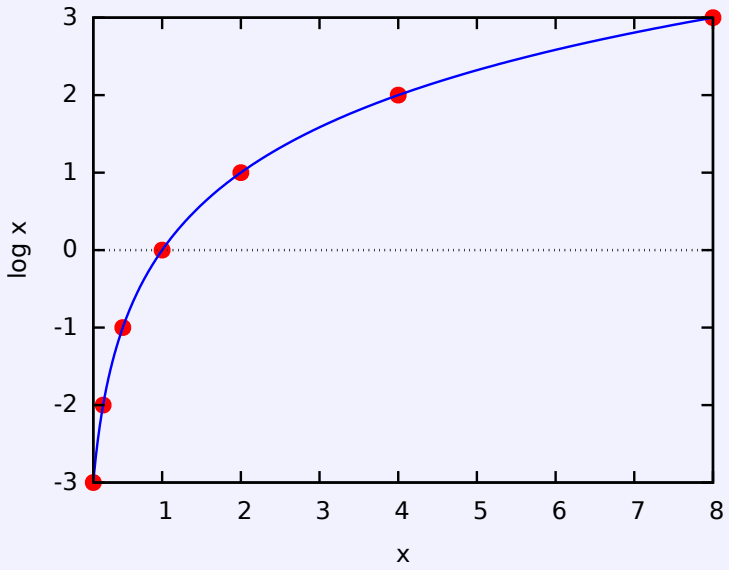
$$\log_2 1/2 = -1$$
$$\log_2 1/4 = -2$$
$$\log_2 1/8 = -3$$

Logarytmy o podstawie 2 c.d.



Wnioski – oszacowania:

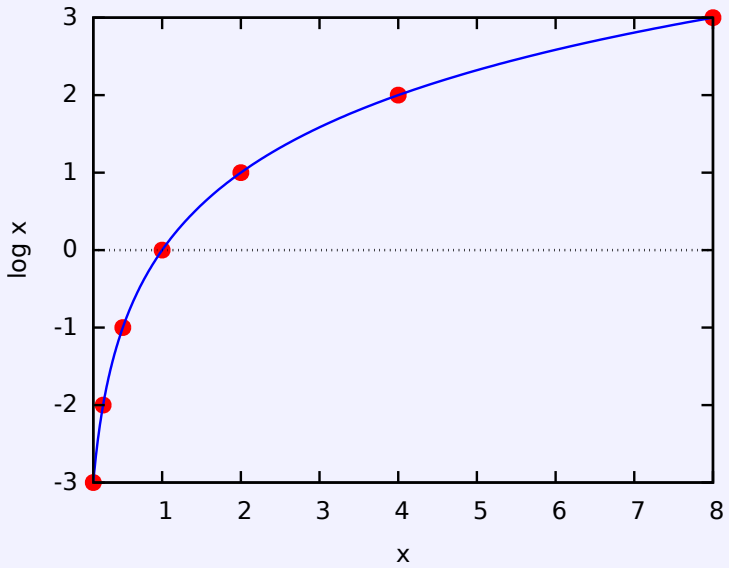
Logarytmy o podstawie 2 c.d.



Wnioski – oszacowania:

$$2 < \log_2 5 < \log_2 6 < \log_2 7 < 3.$$

Logarytmy o podstawie 2 c.d.



Wnioski – oszacowania:

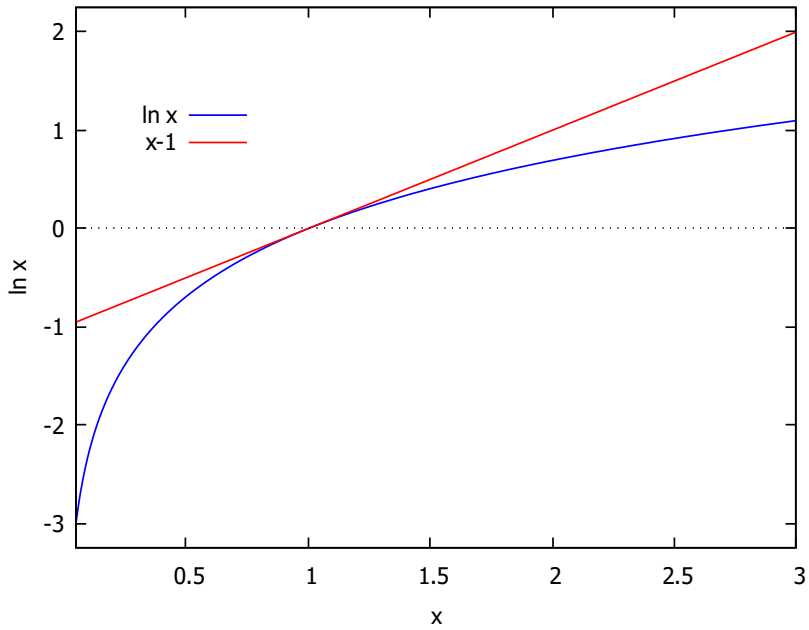
$$2 < \log_2 5 < \log_2 6 < \log_2 7 < 3.$$

$$-3 < \log_2 \frac{1}{7} < \log_2 \frac{1}{6} < \log_2 \frac{1}{5} < -2.$$

Logarytmy naturalne

Podstawa logarytmów naturalnych
$e = 2.718281828459045\cdots \approx 2.71.$

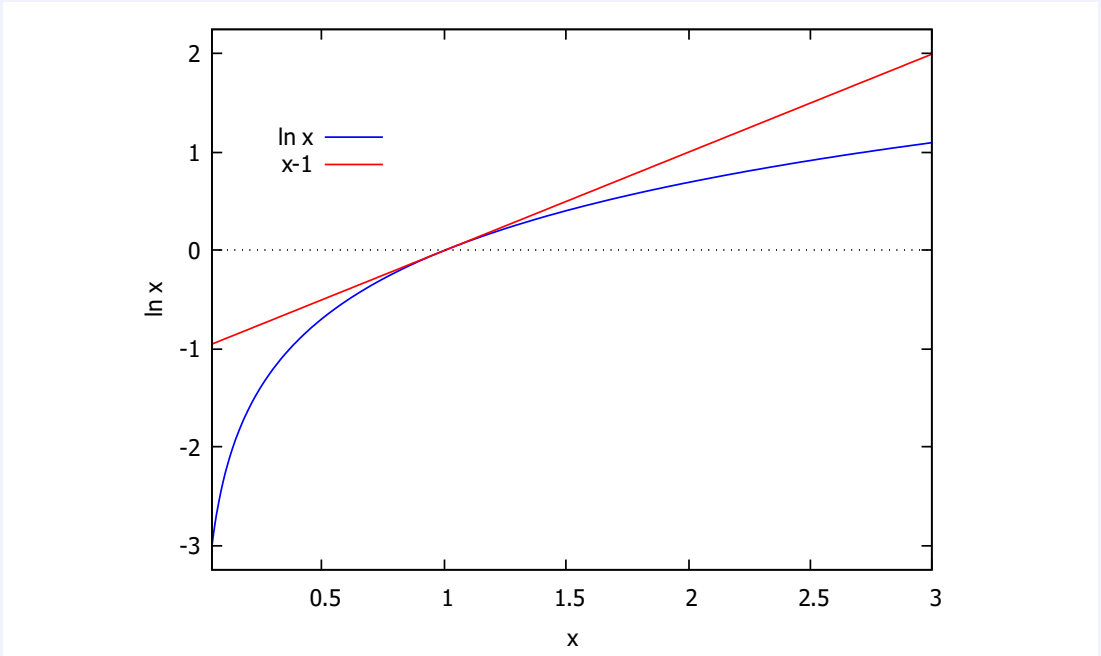
$$\log_e x = \ln x \quad \text{– logarytm naturalny.}$$



Logarytmy naturalne

Podstawa logarytmów naturalnych
$e = 2.718281828459045\cdots \approx 2.71.$

$$\log_e x = \ln x \quad \text{– logarytm naturalny.}$$



Wniosek:

$$\ln x \leq x - 1, \quad \ln \frac{1}{x} \geq 1 - x.$$

Własności logarytmów

Definicja:

$$\log_a b = c \iff a^c = b.$$

Stąd

$$\log_a 1 = 0,\ \log_a a = 1,\ \log_a a^b = b,\ a^{\log_a b} = b.$$

$$\log_a (bc) = \log_a b + \log_a c,$$
$$\log_a \frac{b}{c} = \log_a b - \log_a c,$$
$$\log_a b^c = c \log_a b.$$

Zmiana podstaw logarytmów

Z logarytmu o danej podstawie można otrzymać logarytm o dowolnej innej podstawie.

$$\log_a x = \frac{\log_b x}{\log_b a}\,.$$

Stąd na przykład

$$\log_2 x = \frac{\ln x}{\ln 2}\,,$$

gdzie $\ln x = \log_e x$, $e \approx 2.71$ jest podstawą logarytmów naturalnych.
W przybliżeniu $\ln 2 = 0.693147180559945 3\ldots$.

Logarytm w językach programowania

Język	Nazwa	$\log_a x$	Biblioteka
Pascal	`ln(x)`	$\ln x$	
C++	`log(x)`	$\ln x$	`#include cmath`
	`log10(x)`	$\log_{10} x$	`#include cmath`
	`log2(x)`	$\log_2 x$	`#include cmath`
Python	`math.log(x)`	$\ln x$	`import math`
	`math.log10(x)`	$\log_{10} x$	`import math`
	`math.log2(x)`	$\log_2 x$	`import math`
	`math.log(x,a)`	$\log_a x$	`import math`
Maxima	`log(x)`	$\ln x$	

C++

```
#include <iostream>
#include <cmath>
using namespace std;
double x;
int main ()
{

  x=5.0;
  cout<< "log(5) = " << log(x)<<endl;
  cout<< "log10(5) = " << log10(x)<<endl;
  cout<< "log2(5) = " << log2(x)<<endl;
  return 0;
}
```

C++ c.d.

```
log(5)=1.60944
log10(5)=0.69897
log2(5)=2.32193
```

Python

```
import math

x=5
print("\nlog(5)=")
print(math.log(x))
print("\nlog(5,e)=")
print(math.log(x,math.e))
print("\nlog10(5)=")
print(math.log10(x))
print("\nlog(5,10)=")
print(math.log(x,10))
print("\nlog2(5)=")
print(math.log2(x))
print("\nlog(x,2)=")
print(math.log(x,2))
```

Python c.d.

```
log(5)=
1.6094379124341003
log(5,e)=
1.6094379124341003
log10(5)=
0.6989700043360189
log(5,10)=
0.6989700043360187
log2(5)=
2.321928094887362
log(x,2)=
2.321928094887362
```

Maxima

```
(%i1) if numer#false then numer:false else numer:true;
(%o1) true
(%i2) log(5)
(%o2) 1.6094379124341
(%i3) log10(x):=log(x)/log(10);
(%o3) log10(x):=log(x)/log(10)
(%i4) log10(5);
(%o4) 0.69897000433601 87
(%i5) log2(x):=log(5)/log(2);
(%o5) log2(x):=log(5)/log(2)
(%i6) log2(5);
(%o6) 2.321928094887362
```

Pascal

```
implementation
var x:double;
function log2(x:double):double;
begin
     log2:=ln(x)/ln(2)
end;
function log10(x:double):double;
begin
      log10:=ln(x)/ln(10);
end;
begin
     writeln('ln(5)=',ln(5):0:5);
     writeln('log10(5)=',log10(5):0:5);
     writeln('log2(5)=',log2(5):8:5);
end.
```

Pascal c.d.

```
ln(5)=1.60944
log10(5)=0.69897
log2(5)=2.32193
```

Miara informacji [IK] – 1.2

$$I = -\log_r p,$$

gdzie

$I = I(p)$ – ilość informacji otrzymanej przy zajściu zdarzenia x,

p – prawdopodobieństwo zajścia zdarzenia x.

Miara informacji [IK] – 1.2

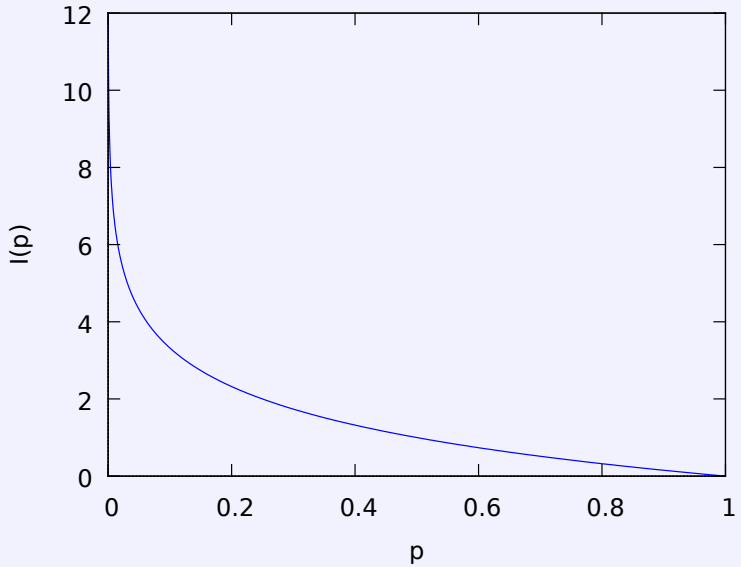
$$I = -\log_r p,$$

gdzie

$I = I(p)$ – ilość informacji otrzymanej przy zajściu zdarzenia x,

p – prawdopodobieństwo zajścia zdarzenia x.

Dla $r = 2$:



Średnia [IK] 1.4

Średnia statystyczna

$$\bar{x} = p_1 x_1 + p_2 x_2 + \cdots + p_n x_n \,,$$

gdzie p_i jest prawdopodobieństwem zajścia zdarzenia x_i.

Przykład. Rzucamy kostką do gry. Prawdopodobieństwo wyrzucenia szóstki wynosi $1/6$. Prawdopodobieństwo wyrzucenia czterech szóstek pod rząd wynosi

$$\begin{aligned}(1/6)^4 &= 1/1296 = 7.716049382716048 \cdot 10^{-4} \\ &= 0.0007716049382716048 < 0.001.\end{aligned}$$

Definicja entropii

Niech x_i będą wszystkimi możliwymi wynikami dla pewnego zdarzenia losowego, $1 \leq i \leq n$.

$$x = \{x_1, x_2, \ldots, x_n\}.$$

Niech p_i będzie prawdopodobieństwem otrzymania wyniku x_i.

Definicja entropii

Niech x_i będą wszystkimi możliwymi wynikami dla pewnego zdarzenia losowego, $1 \leq i \leq n$.

$$x = \{x_1, x_2, \ldots, x_n\}.$$

Niech p_i będzie prawdopodobieństwem otrzymania wyniku x_i.

Entropia:

Średnia

$$H(x) = -\left(p_1 \log_r p_1 + p_2 \log_r p_2 + \cdots + p_n \log_r p_n\right).$$

Tylko jeden bit

Przykład. x może przybrać tylko dwie wartości: 0 i 1:

$$\Pr(x = 0) = p,$$
$$\Pr(x = 1) = 1 - p.$$

Tylko jeden bit

Przykład. x może przybrać tylko dwie wartości: 0 i 1:

$$\Pr(x = 0) = p,$$
$$\Pr(x = 1) = 1 - p.$$

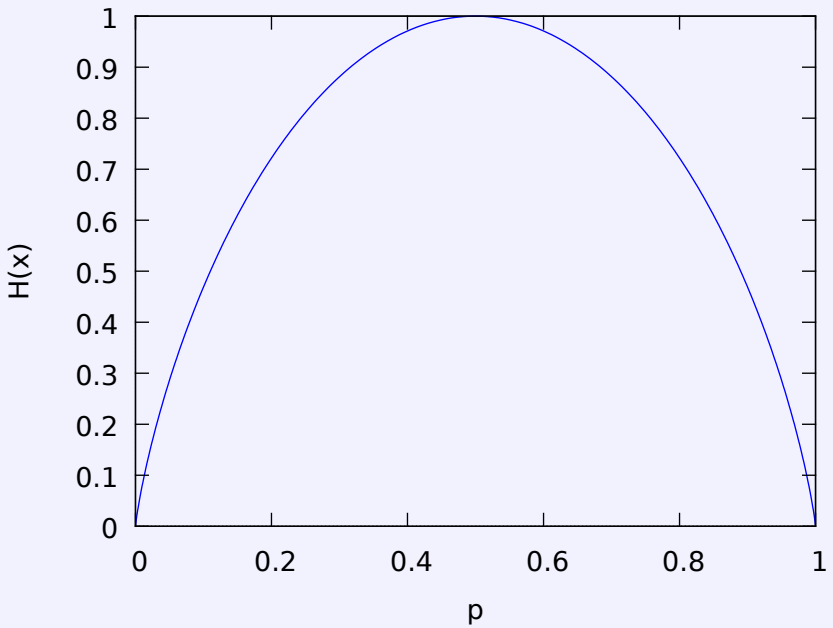
$$H_p(x) = -\left(p \log_2 p + (1 - p) \log_2 (1 - p)\right)$$

Tylko jeden bit

Przykład. x może przybrać tylko dwie wartości: 0 i 1:

$$\Pr(x = 0) = p,$$
$$\Pr(x = 1) = 1 - p.$$

$$H_p(x) = -\left(p \log_2 p + (1-p) \log_2 (1-p)\right)$$



Przykład dla $r = 2$ (1) [IK] – 1.3, str. 20

Przesyłamy wynik stanu urządzenia, które może znajdować się w czterech stanach: stan normalny N i jeden z trzech stanów awaryjnych A, B, C.

Przykład dla $r = 2$ (1) [IK] – 1.3, str. 20

Przesyłamy wynik stanu urządzenia, które może znajdować się w czterech stanach: stan normalny N i jeden z trzech stanów awaryjnych A, B, C. Potrzebujemy dwóch bitów do określenia stanu:

N	00
A	01
B	10
C	11

Przykład dla $r = 2$ (2)

Załóżmy, że urządzenie jest

- w stanie normalnym z prawdopodobieństwem 0.98,
- stanie A z prawdopodobieństwem 0.01,
- w każdym ze stanów B i C, z prawdopodobieństwami po 0.005.

Wtedy

$$H(x) = -\left(0.98 \log_2 0.98 + \cdot 0.01 \cdot \log_2 0.01 + 2 \cdot 0.005 \cdot \log_2 0.005\right).$$

Przykład dla $r = 2$ (3)

Obliczamy logarytmy:

$$\log_2 0.980 = -0.02914634565951651$$
$$\log_2 0.010 = -0.643856189774724$$
$$\log_2 0.005 = -7.643856189774724$$

Przykład dla $r = 2$ (3)

Obliczamy logarytmy:

$$\log_2 0.980 = -0.02914634565951651$$
$$\log_2 0.010 = -0.643856189774724$$
$$\log_2 0.005 = -7.643856189774724$$

Wtedy

$$\begin{aligned} H(x) &= -\left(0.98 \log_2 0.97 + 0.01 \cdot \log_2 0.01 + 2 \cdot 0.005 \cdot \log_2 0.005\right) \\ &= 0.1714405425418207. \end{aligned}$$

Przykład dla $r = 2$ (4)

Nowy, oszczędniejszy sposób przesyłania danych:

N	0
A	101
B	110
C	111

Odczyt: 0 – stan N, jeśli 1, to odczytujemy dwa następne bity.
Teraz średnia liczba bitów na jeden odczyt danych:

$$0.98 + 3 \cdot 0.01 + 2 \cdot 3 \cdot 0.005 = 1.04.$$

Przykład dla $r = 2$ (4)

Nowy, oszczędniejszy sposób przesyłania danych:

N	0
A	101
B	110
C	111

Odczyt: 0 – stan N, jeśli 1, to odczytujemy dwa następne bity.
Teraz średnia liczba bitów na jeden odczyt danych:

$$0.98 + 3 \cdot 0.01 + 2 \cdot 3 \cdot 0.005 = 1.04.$$

To jest pierwszy przykład *kompresji danych*.

Częstość [IK] – 2.1

Urządzenie wysłało ciąg 50 znaków ze zbioru pięcioelementowego

$$S = \{a, b, c, d, e\} :$$

ababcdaeda

cdaaabcacc

bcdaacdabe

deabccbaab

bdbdaaacbd

Częstość c.d.

Liczby k wystąpień znaków i ich częstości f_k:

znak	k	f_k
a	17	0.34
b	11	0.22
c	10	0.20
d	9	0.18
e	3	0.06
	50	1.00

n-krotny rzut monetą

Rzucamy symetryczną monetą n razy, a pierwszy rzut ma numer 0. Przyjmijmy, że wyrzucenie orła to 1, a reszki to 0. Wobec tego seria takich rzutów jest równoważna liczbie $x = (x_{n-1} \dots x_1 x_0)_2$, $0 \leq x \leq 2^n - 1$.
Jakie jest prawdopodobieństwo przyjęcia przez x dowolnej ustalonej wartości z tego zakresu?

Jeżeli $n = 1$, prawdopodobieństwo to jest równe $1/2$ i równe prawdopodobieństwu wyrzucenia dowolnej strony monety.
Jeżeli $n = 2$, prawdopodobieństwo przyjęcia dowolnej wartości przez $x \in \{0, 1, 2, 3\}$ jest równe $(1/2)(1/2) = 1/4$ itd.
Dla n rzutów, prawdopodobieństwo to jest równe $(1/2)^n = 1/2^n$.
Jest tak, bo poszczególne rzuty są od siebie niezależne.

Rzut kostka i monetą

Rzucamy jednocześnie kostką do gry i monetą jeden raz. Jakie jest prawdopodobieństwo wyrzucenia pary (i, j), gdzie $i = 0$ lub $i = 1$ jest wynikiem rzutu monetą, a j jest liczbą wyrzuconych oczek na kostce?
Rozumując jak w poprzedni przykładzie otrzymujemy prawdopodobieństwo równe $(1/2)(1/6) = 1/12$. I znowu: jest tak bo rzucamy monetą i kostką niezależnie od siebie, to znaczy że wynik rzutu monetą nie wpływa na wynik rzutu kostką i odwrotnie.

Niezależność [IK] – 2.1, str. 30

Dwie wiadomości elementarne lub dwa zdarzenia x_i i x_j o prawdopodobieństwach ich pojawienia się p_i i p_j są niezależne, gdy prawdopodobieństwo ich łącznego pojawienia się jest równe $p_i p_j$.

Źródło bezpamięciowe [IK] – 2.1

Kolejne wiadomości elementarne

$$x_1, x_2, \ldots$$

pojawiają się niezależnie od poprzednich.

Źródło bezpamięciowe [IK] – 2.1

Kolejne wiadomości elementarne

$$x_1, x_2, \ldots$$

pojawiają się niezależnie od poprzednich.
Załóżmy, że wiadomość elementarna s, wysłana przez źródło, jest równa s_i z prawdopodobieństwem p_i. Zawsze:

$$p_1 + p_2 + \cdots + p_n = 1\,.$$

Źródło bezpamięciowe [IK] – 2.1

Kolejne wiadomości elementarne

$$x_1, x_2, \ldots$$

pojawiają się niezależnie od poprzednich.
Załóżmy, że wiadomość elementarna s, wysłana przez źródło, jest równa s_i z prawdopodobieństwem p_i. Zawsze:

$$p_1 + p_2 + \cdots + p_n = 1\,.$$

Niezależność:

$$\Pr\left(x = s_i, y = s_j\right) = p_i p_j\,,$$

gdzie x i y są dwiema dowolnymi wiadomościami wysłanym ze źródła.

Ilość informacji – ponownie

Ilość informacji $I(p)$ powinna spełniać warunki:

- $I(p) \geq 0$,
- $I(1) = 0$,
- Jeśli p jest bliskie zeru, to $I(p)$ jest bardzo duże, co można sformułować jako $I(0) = \infty$

Funkcjami spełniającymi te warunki są właśnie logarytmy.
Definicję taką wprowadził C. Shannon.

Claude Elwood Shannon

ur. 30 kwietnia 1916, zm. 24 lutego 2001.

Amerykański matematyk i inżynier, profesor Massachusetts Institute of Technology (MIT). Stworzył modele procesu komunikacyjnego wykorzystywane później przez psychologów. Jego najsłynniejsze dzieło, *A Mathematical Theory of Communication* opublikowane w 1948 roku, położyło podwaliny pod teorię informacji i kodowania.

Claude Shannon znany był z tego, że uprawianie nauki było dla niego nie tylko poważnym zajęciem, ale również radością i zabawą. Dla czystej przyjemności na przykład skonstruował maszynę o nazwie THROBAC-I, która liczyła, wykorzystując rzymski zapis liczb.

Shannon opracował zasady działania wyspecjalizowanego komputera do gry w szachy na blisko pół wieku przed głośnym meczem Garriego Kasparowa z komputerem Deep Blue. Był laureatem Nagrody Kioto w dziedzinie nauk podstawowych (1985).

Entropia źródła

Ilość informacji przenoszona przez wiadomość s_i:

$$I(s_i) = -\log p_i \,.$$

Entropia źródła

Ilość informacji przenoszona przez wiadomość s_i:

$$I(s_i) = -\log p_i \,.$$

Średnia ilość informacji przenoszona przez wiadomości ze źródła S, czyli entropia źródła:

$$\begin{aligned} H(S) &= p_1 I(s_1) + p_2 I(s_2) + \cdots + p_n I(s_n) \\ &= -\left(p_1 \log p_1 + p_2 \log p_2 + \cdots + p_n \log p_n\right) \\ &= -\sum_{i=1}^{n} p_i \log p_i = -\sum_{S} p_i \log p_i \end{aligned}$$

Przykład

Źródło

$$S = \{s_1, s_2, s_3\}$$

oraz

$$p_1 = \frac{1}{2}, \; p_2 = p_3 = \frac{1}{4}.$$

Entropia

$$H(S) = \frac{1}{2}\log_2 2 + \frac{1}{4}\log_2 4 + \frac{1}{4}\log_2 4 = \frac{1}{2} + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 = \frac{3}{2}.$$

Oszacowanie entropii

$$H(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \,.$$

Można wyprowadzić (proste zadanie do rozwiązania):

$$H(S) \leq \log_2 n.$$

Entropia jest zawsze mniejsza lub równa $\log_2 n$.

Oszacowanie entropii

$$H(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \,.$$

Można wyprowadzić (proste zadanie do rozwiązania):

$$H(S) \leq \log_2 n.$$

Entropia jest zawsze mniejsza lub równa $\log_2 n$.
Jest równa $\log_2 n$ (jest największa) tylko wtedy, gdy $p_i = 1/n$ dla wszystkich i.