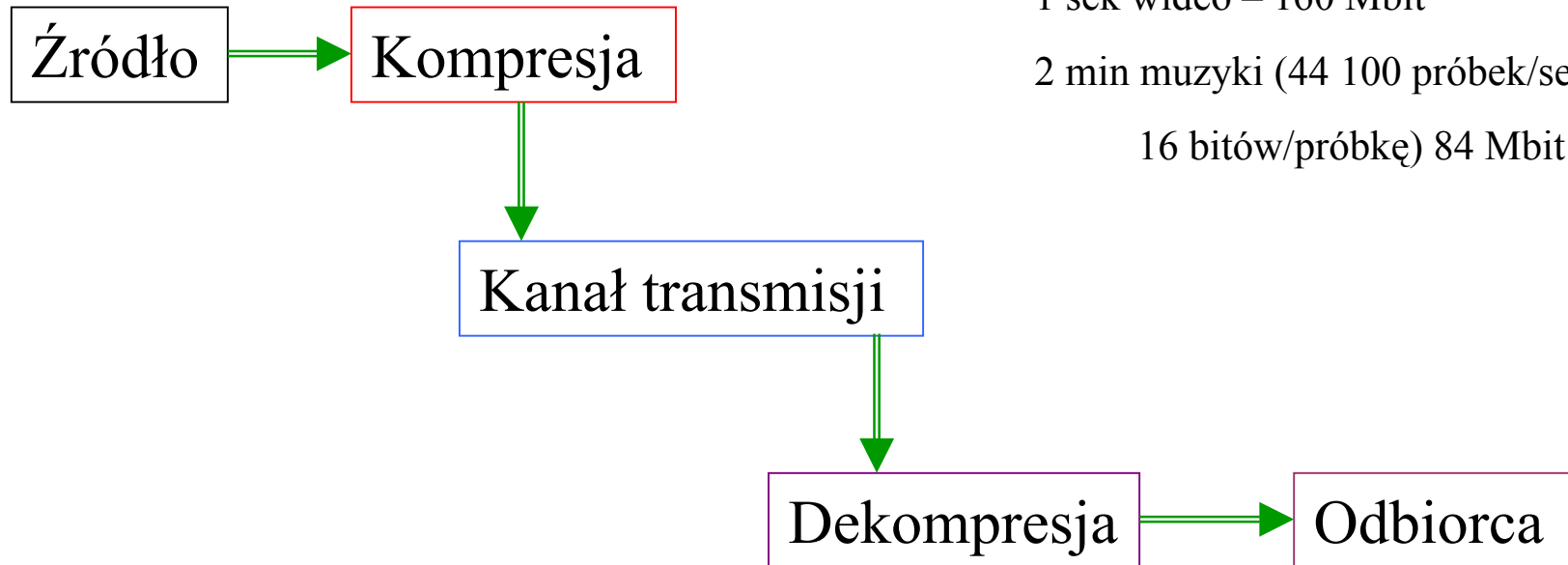
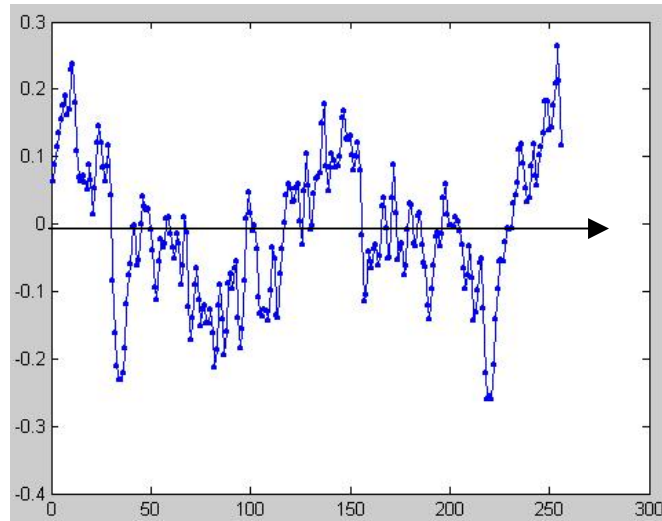


STANDARDOWE TECHNIKI KOMPRESJI SYGNAŁÓW



1. Metody bezstratne
2. Metody stratne

Kodowanie predykcyjne



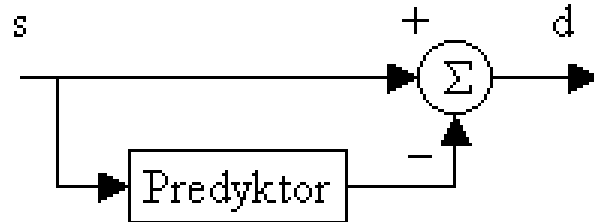
$$p(n) = \sum_{i=1}^r \alpha_i s(n-i)$$

np. z autokorelacji

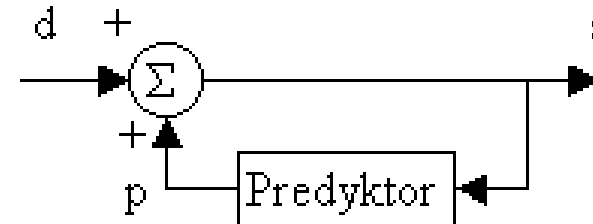
$$\beta_i = \sum_n s(n-i) s(n)$$

normalizujemy, aby $\sum_{i=1}^r \alpha_i = 1$

czyli
$$\alpha_i = \frac{\beta_i}{\sum_{i=1}^r \beta_i}$$



$$d(n) = s(n) - p(n)$$



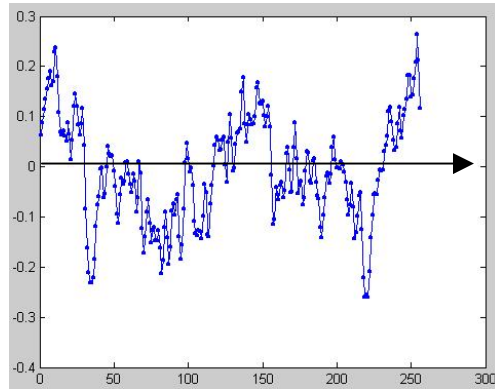
$$s(n) = p(n) + d(n)$$

NADAWCA

ODBIORCA

Kodowanie entropowe

EC od ang. Entropy Coding



$$s(n) \in \{s_m\}_{m=1}^M$$

Ilość informacji stowarzyszona z komunikatami $\{s_m : m = 1, \dots, M\}$

wynosi $I(m) = -\log_2 p_m$, gdzie p_m jest prawdopodobieństwem wystąpienia s_m , a jej miarą jest ilość bitów.

Średnia ilość bitów potrzebnych do bezstratnego zakodowania komunikatu (**entropia**) jest wartością oczekiwaną

$$H = -\sum_{m=1}^M p_m \log_2 p_m$$

Alfabet Morse'a

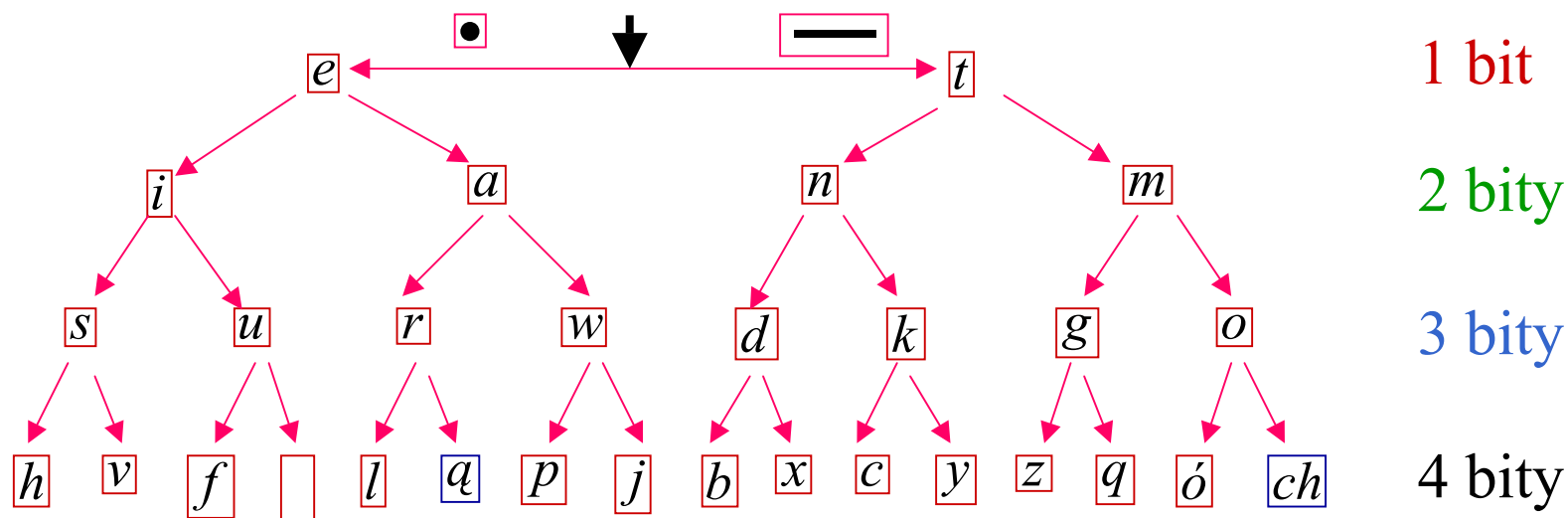
Samuel Finley Breese MORSE (1791-1872) –
amerykański malarz i wynalazca

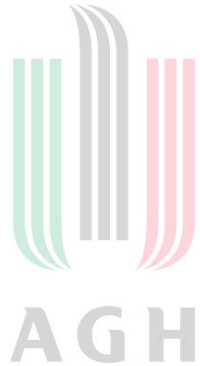
1837 – aparat telegraficzny

1840 - alfabet telegraficzny

1844 – pierwsza na świecie linia

Baltimore - Washington





Sprawność kodowania

$$H = -\sum_{m=1}^M p_m \log_2 p_m$$

Oczekiwana ilość bitów

$$H_w = \sum_{m=1}^M b_m p_m$$

gdzie p_m jest prawdopodobieństwem komunikatu zakodowanego przez symbol posiadający b_m bitów.

Sprawność kodowania

$$\eta = \frac{H}{H_w} 100\%$$

może być co najwyżej równa 100%, bo entropia jest dolną granicą średniej liczby bitów wymaganych do reprezentacji komunikatów.



Kodowanie ze zmienną długością słowa

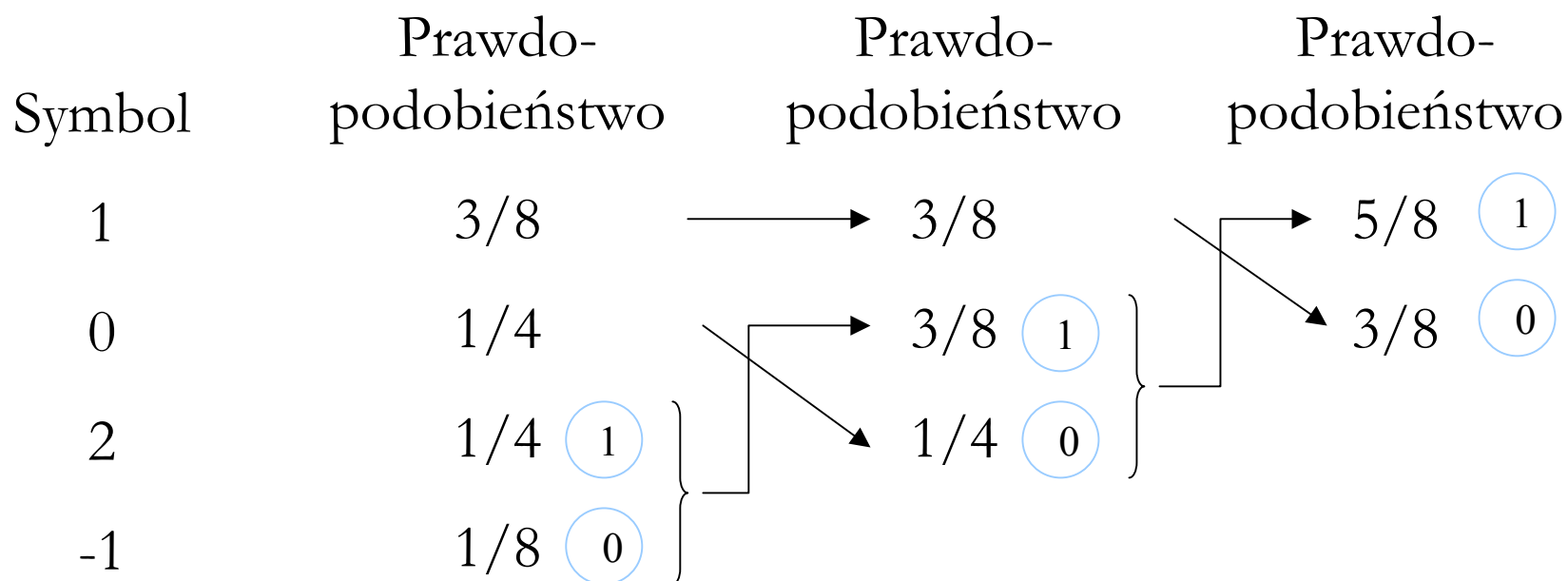
VLC ang. **V**ariable **L**ength **C**oding

s_m	p_m
-1	1/8
0	1/4
1	3/8
2	1/4

$$H = -\frac{1}{8} \log_2 \frac{1}{8} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{8} \log_2 \frac{3}{8} - \frac{1}{4} \log_2 \frac{1}{4} \approx 1,9$$



Przykład algorytmu kodowania metodą Huffmana 1952 rok





Kontynuacja przykładu kodu Huffmana

Symbol	-1	0	1	2
Prawdopodobieństwo	0,125	0,250	0,375	0,250
Kod binarny	110	10	0	111

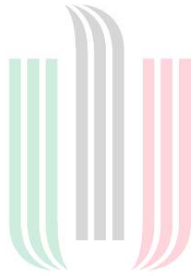
Przeciętna ilość bitów na symbol $H_w = 1 \cdot \frac{3}{8} + 2 \cdot \frac{1}{4} + 3(\frac{1}{4} + \frac{1}{8}) = 2$

Sprawność kodowania $\eta = \frac{1,9}{2} 100\% = 95\%$

111|111|110|0|0|10|0|0|110

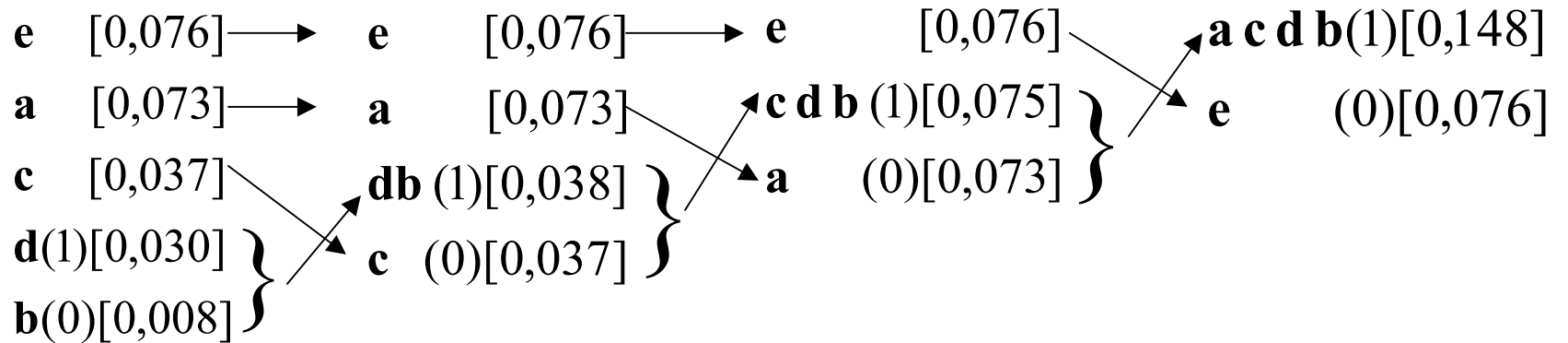
2 2 -1 1 1 0 1 1 -1

100 % sprawność kodowania gdy prawdopodobieństwa są potęgami 1/2



AGH

Kolejny przykład kodowania Huffmana



Symbol	a	b	c	d	e
Prawdopodobieństwo	0,073	0,008	0,037	0,030	0,076
Kod binarny	10	1110	110	1111	0

Prosty przykład kodu Huffmana

p_m gęstość prawdopodobieństwa zmiennej losowej s_m

$$F_m = \sum_{i=1}^m p_i \quad \text{dystrybuanta}$$

Znak	Prawdopodobieństwo	Kod Huffmana
♂	0,95	1
♀	0,02	00
😊	0,03	01

$$H = 0,335 \text{ bit/symbol}$$

$$H_w = 1,05 \text{ bit/symbol}$$

Sprawność tylko

$$\eta = 31,9\%$$

Kod Huffmana sekwencji symboli

Sekwencja symboli	Iloczyn prawdopodobieństw	Kod Huffmana
♂♂	0,9025	1
♂♀	0,0190	000
♂😊	0,0285	011
♀♂	0,0190	0010
♀♀	0,0004	001100
♀😊	0,0006	001110
😊♂	0,0285	010
😊♀	0,0006	001101
😊😊	0,0009	001111

$$H = 0,611 \text{ bit/symbol}$$

$$H_w = 1,222 \text{ bit/symbol}$$

$$\text{sprawność } \eta = 50\%$$

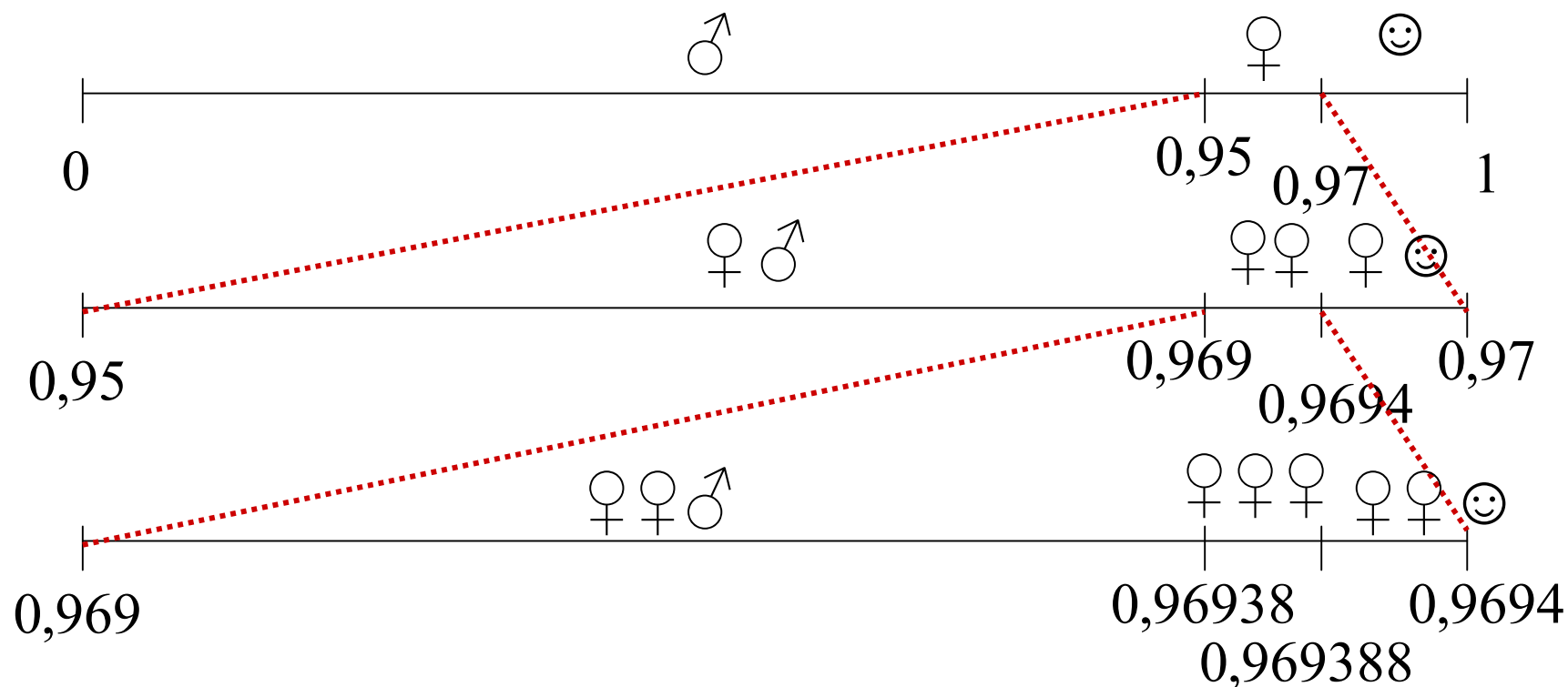
Poprzednio:

$$H = 0,335 \text{ bit/symbol}$$

$$H_w = 1,05$$

$$\eta = 31,9\%$$

Tworzenie sekwencji elementów dla kodowania arytmetycznego



Ilość bitów potrzebna do zakodowania komunikatu
 jest częścią całkowitą $1 - \log_2 p_n$

Przykład kodowania arytmetycznego $H_w = 1,5008 \quad \eta = 40,7\%$

Sekwencja symboli	Iloczyn prawdopodobieństw	Granice przedziału	Wartość środkowa w kodzie dziesiętnym	Wartość środkowa w kodzie binarnym	$1 - \log_2 p_n$
♂♂	0,9025	0 – 0,9025	0,45125	0,0111001110000101 0	1,15
♂♀	0,0190	0,9025 – 0,9215	0,912	0,1110100101111000 111010	6,72
♂😊	0,0285	0,9215 – 0,95	0,93575	0,1110111110001101 111011	6,13
♀♂	0,0190	0,95 – 0,969	0,9595	0,1111010110100001 111101	6,72
♀♀	0,0004	0,969 – 0,9694	0,9692	0,1111100000011101 1111100000011	13,29
♀😊	0,0006	0,9694 – 0,97	0,9697	0,1111100000111110 11111000001	11,70
😊♂	0,0285	0,97 – 0,9985	0,98425	0,1111101111110111 111110	6,13
😊♀	0,0006	0,9985 – 0,9991	0,9988	0,111111110110001 1111111101	11,70
😊😊	0,0009	0,9991 – 1	0,99955	0,1111111111100010 1111111111	11,12



Kodowanie ciągów

RLC ang. Run Length Coding

(v, r) gdzie v (od ang. value) powtarzający się symbol
 r (od ang. run) liczba powtórzeń

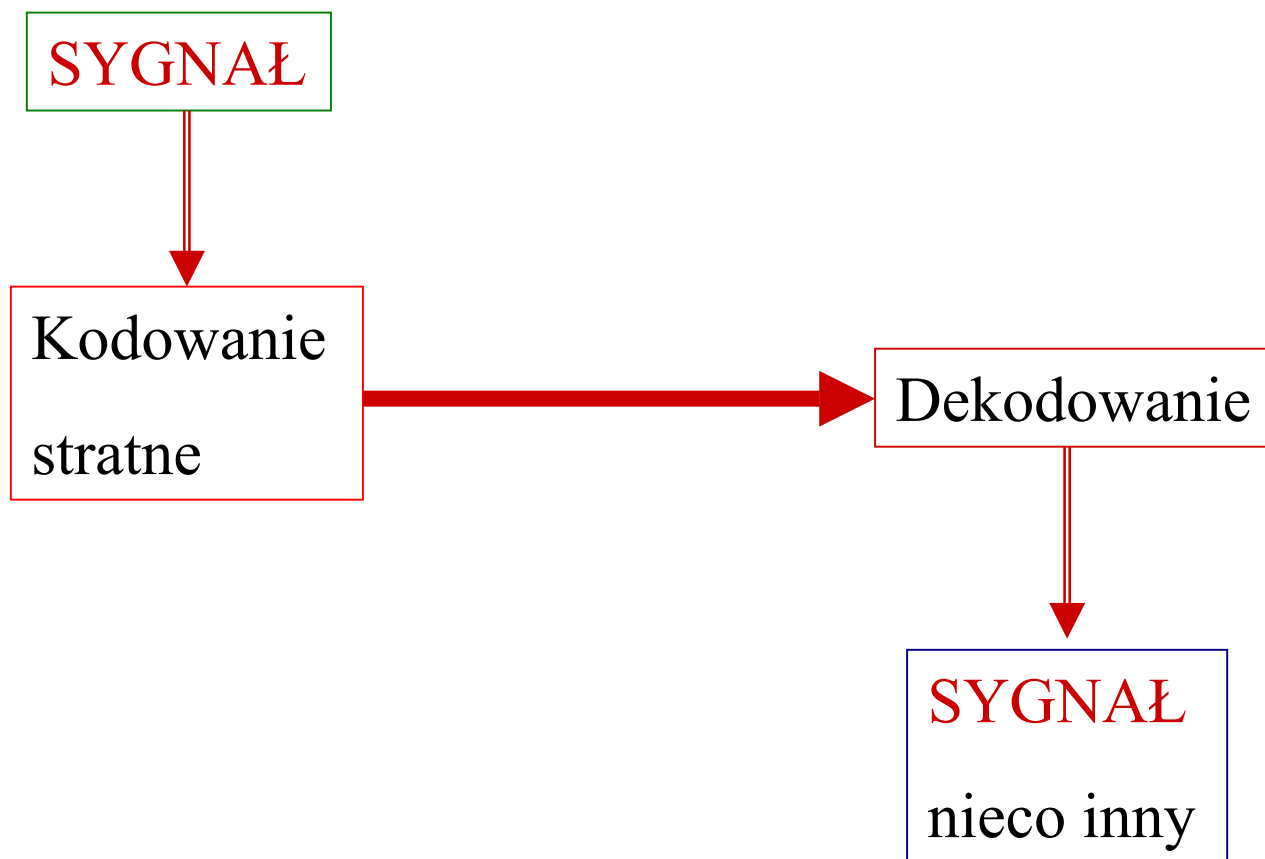
1111000111111110001111000000111111111111

(1,1) (0,1) (1,2) (0,1) (1,1) (0,2) (1,3)

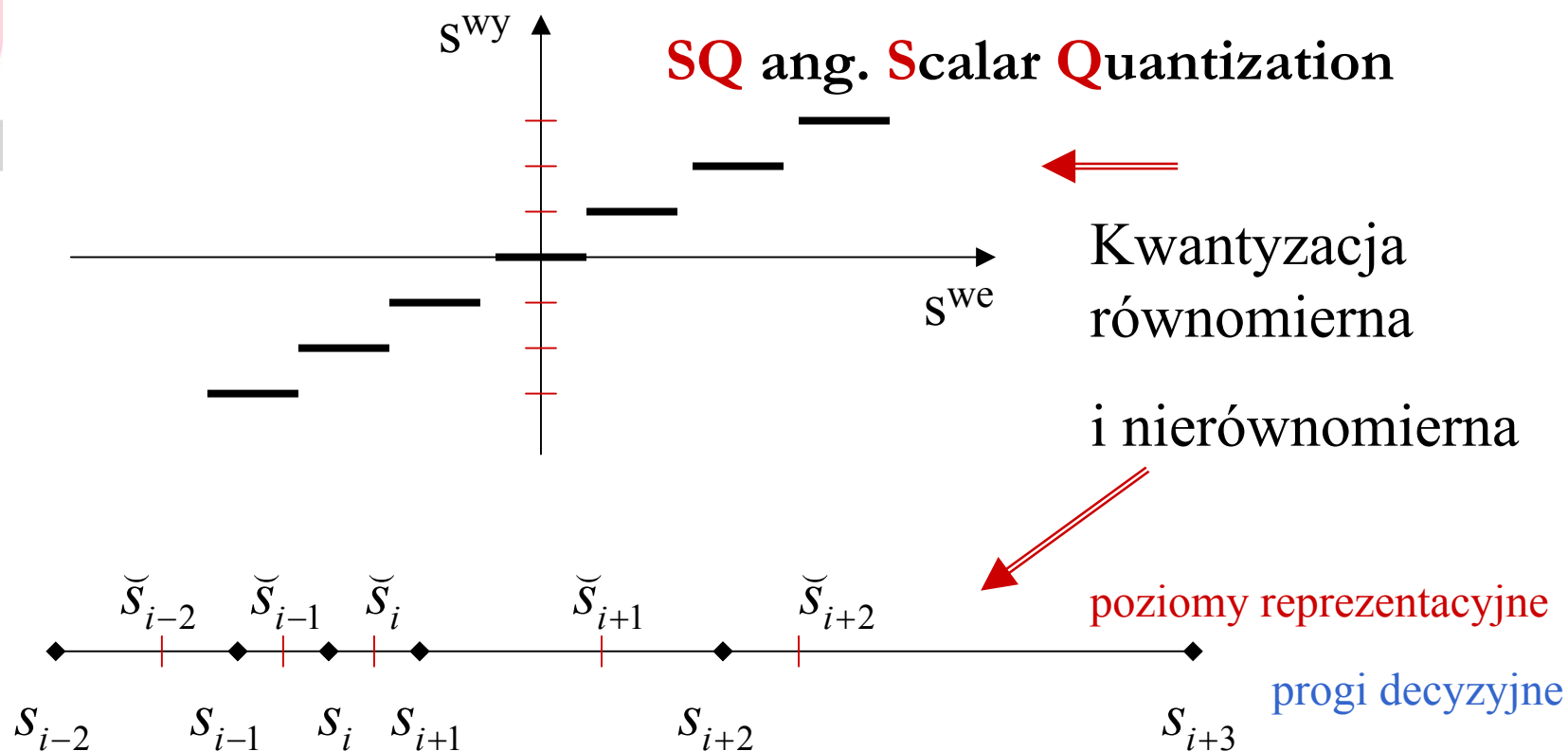
1 kod 1111

0 kod 000

Metody kompresji stratnej



Kwantyzacja skalarna



SQ ang. **S**calar **Q**uantization

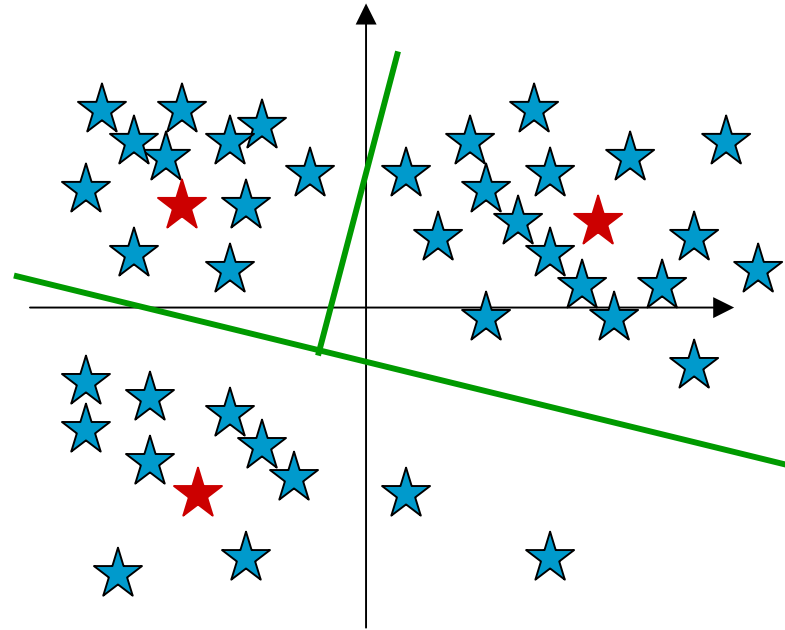
Kwantyzacja
równomierna
i nierównomierna

poziomy reprezentacyjne
progi decyzyjne

$$s \in \mathfrak{R}_i \subset \mathfrak{R} \Rightarrow Q(s) = \check{s}_i \quad \check{s}(i) \text{ poziom reprezentacyjny lub kod}$$

gdzie $\mathfrak{R}_i = [s_i, s_{i+1}]$ są rozdzielnymi zbiorami pokrywającymi podzbiór liczb rzeczywistych.

Kwantowanie wektorowe



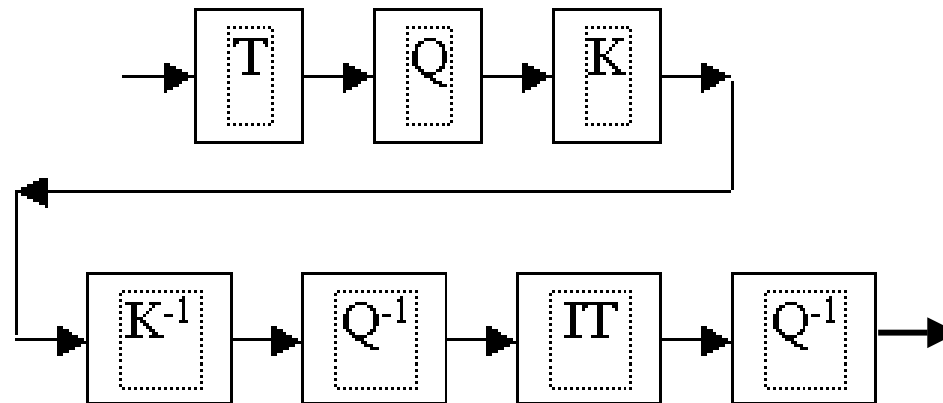
Przykład gdy $s, s_i \in \mathcal{R}^2$

Książka kodowa $Q(s) = s_i$

np. działanie w/g reguły $d(s, s_i) \leq d(s, s_j) \quad \forall j = 1, \dots, M$

M - ilość elementów książki kodowej

Kodowanie transformatowe



T - transformacja

Q - kwantowanie

K – kodowanie bezstratne

K⁻¹ - dekodowanie

Q⁻¹ - dekwantyzacja

IT- transformacja odwrotna

Transformacje częstotliwościowe

krótkoczasowa

Fouriera

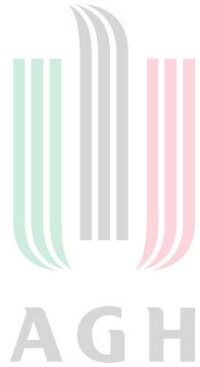
$$\hat{s}(f) = \int_{-\infty}^{+\infty} w(t) s(t) e^{-2\pi j f t} dt = \int_{-\infty}^{+\infty} w(t) s(t) (\cos(2\pi j f) - j \sin(2\pi j f)) dt$$

kosinusowa

$$\hat{s}(f) = \int_{-\infty}^{+\infty} s(t) \cos(2\pi f t) dt$$

falkowa

$$\tilde{s}_{\psi}(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(t) \psi\left(\frac{t-b}{a}\right) dt$$



Dyskretna transformacja kosinusowa

DCT od ang. **D**iscrete **C**osine **T**ransform

$$\hat{s}(k) = \sqrt{\frac{2}{N}} c(k) \sum_{n=0}^{N-1} s(n) \cos\left(\frac{(2n+1)k\pi}{2N}\right) \quad k = 0, 1, \dots, N-1$$

gdzie

$$c(k) = \begin{cases} 1/\sqrt{2} & \text{dla } k = 0 \\ 1 & \text{dla } k \neq 0 \end{cases}$$

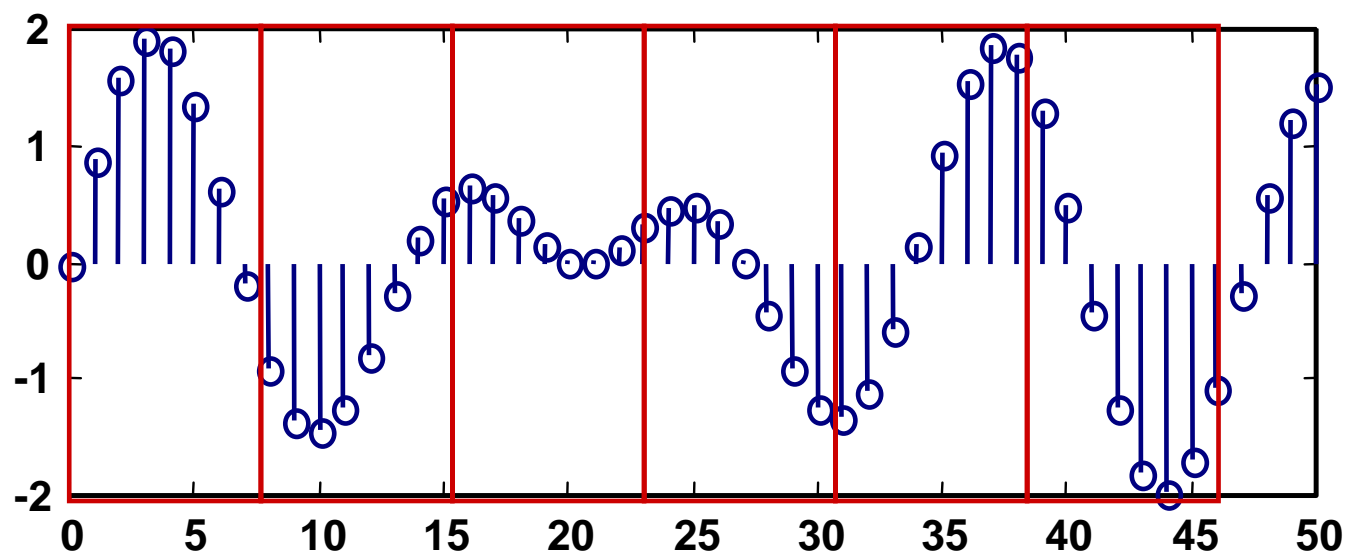
Transformacja odwrotna

$$s(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} c(k) \hat{s}(k) \cos\left(\frac{(2n+1)k\pi}{2N}\right)$$

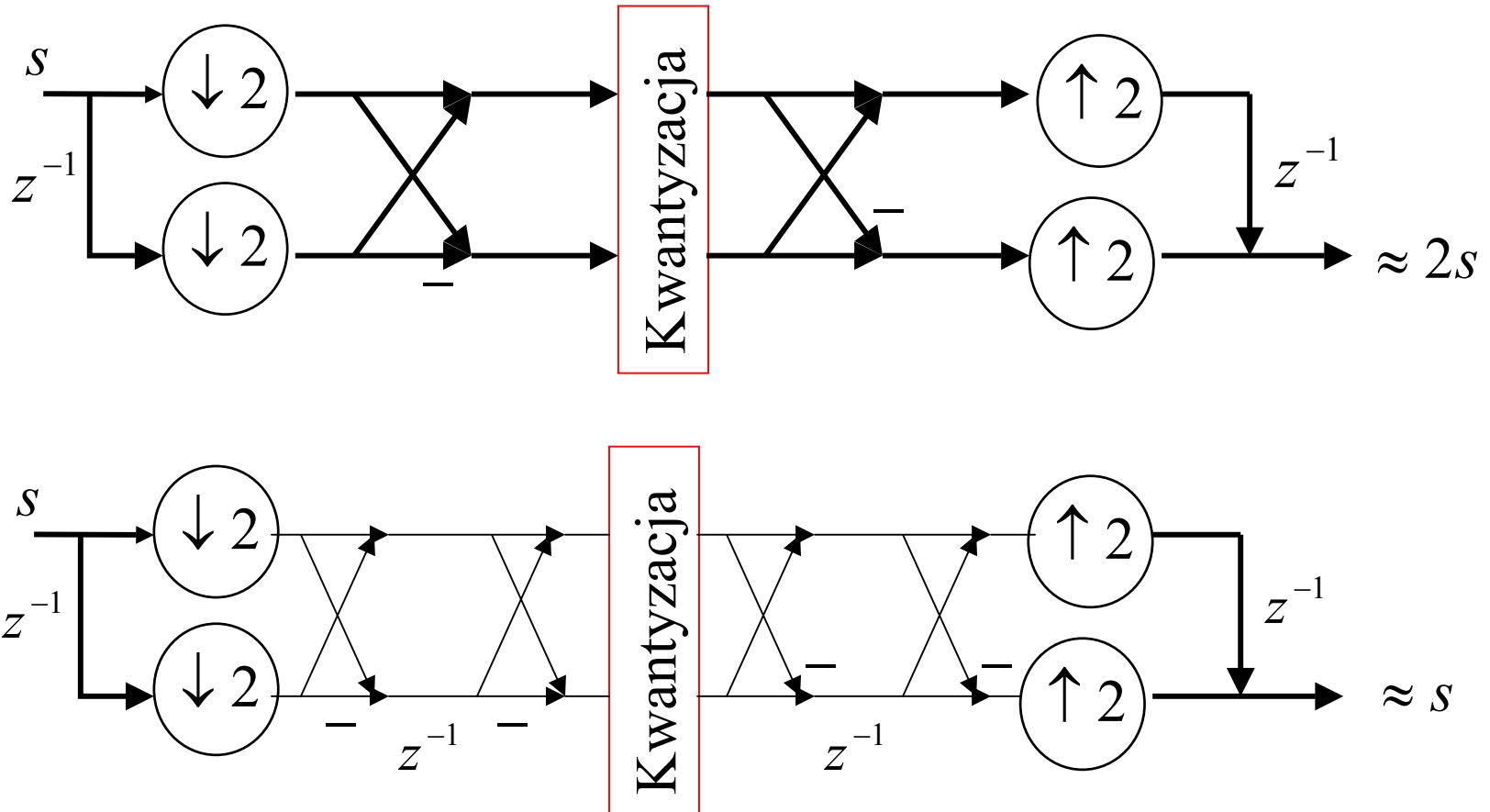
DCT dla bloków po 8 próbek

Dzieląc sygnał na bloki po 8 próbek posługujemy się transformacją

$$s(k) = 0,5 c(k) \sum_{n=0}^7 s(n) \cos((2n+1)k\pi / 16)$$



Banki filtrów cyfrowych





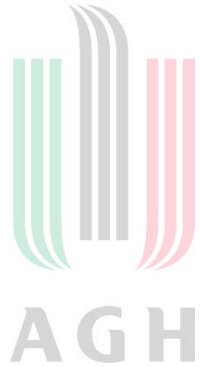
Sygnał audio:

44 100 próbek/sekundę po 16 bitów daje
705 600 bitów/s w jednym kanale

Sygnał telefoniczny:

- Filtrowanie do 4 kHz,
- Próbkowanie 8000 próbek/sekundę,
- Zamiana próbek na 13-bitowe pakiety -
strumień 104 kbit/s,
- KOMPRESJA do 13 kbit/s.

Kompresja $705\ 600 / 13\ 000 \sim 54,28$



Kompresja audio w GSM

- Sygnał dzielony na bloki po 20 ms,
- Każdy blok kodowany na 260 bitach,
- Biblioteka wzorców (sygnałów wzorcowych) i 104 bity na sygnał wzorcowy,
- 156 bitów opisuje różnicę między wzorcem a oryginałem.

$$260 \text{ bit} \cdot \frac{1}{20 \text{ ms}} = 13 \text{ kbit/s}$$