Kody korekcyjne, czyli jak można wykryć i poprawić błąd w przesyłanej informacji

MiNI Akademii Matematyki

Wydział Matematyki i Nauk Informacyjnych PW
Agata Pilitowska

20 listopad 2010

Cyfrowa transmisja danych

Schemat

Informacja $c_1 c_2 \ldots c_k$ $\overset{\textit{Transmisja}}{\Longrightarrow}$ *Informacja odebrana* $w_1 w_2 \ldots w_k$

Cyfrowa transmisja danych

Schemat

$$\underset{c_1 c_2 \ldots c_k}{\textit{Informacja}} \xrightarrow{\textit{Transmisja}} \underset{w_1 w_2 \ldots w_k}{\textit{Informacja odebrana}}$$

Informacja o kierunku wiatru

$$\begin{aligned} W &= 00 \\ Z &= 10 \\ Pn &= 01 \\ Pd &= 11 \end{aligned}$$

Schemat

$$\begin{array}{ccc} \textit{Informacja} & & \textit{Informacja odebrana} \\ c_1 c_2 \ldots c_k & \overset{\textit{Transmisja}}{\Longrightarrow} & w_1 w_2 \ldots w_k \end{array}$$

Informacja o kierunku wiatru

$$\begin{array}{rcl} W & = & 00 \\ Z & = & 10 \\ Pn & = & 01 \\ Pd & = & 11 \end{array} \quad \overset{\textit{Transmisja}}{\Longrightarrow} \quad 11$$

Przykłady funkcji kodujących

$\mathcal{C}_1$: Wiadomość powtórzona

$$
\begin{array}{lcll}
W & = & 00 & \mapsto & 0000 \\
Z & = & 10 & \mapsto & 1010 \\
Pn & = & 01 & \mapsto & 0101 \\
Pd & = & 11 & \mapsto & 1111
\end{array}
$$

Przykłady funkcji kodujących

$\mathcal{C}_1$: Wiadomość powtórzona

$$
\begin{array}{lcll}
W & = & 00 & \mapsto & 0000 \\
Z & = & 10 & \mapsto & 1010 \\
Pn & = & 01 & \mapsto & 0101 \\
Pd & = & 11 & \mapsto & 1111
\end{array}
\quad \overset{Transmisja}{\Longrightarrow} \quad 1101
$$

Przykłady funkcji kodujących

$\mathcal{C}_1$: Wiadomość powtórzona

$$\begin{array}{lcll} W & = & 00 & \mapsto & 0000 \\ Z & = & 10 & \mapsto & 1010 \\ Pn & = & 01 & \mapsto & 0101 \\ Pd & = & 11 & \mapsto & 1111 \end{array} \quad \overset{Transmisja}{\Longrightarrow} \quad 1101$$

$\mathcal{C}_2$: Wiadomość z jednym symbolem kontrolnym

$$\begin{array}{lcll} W & = & 00 & \mapsto & 000 \\ Z & = & 10 & \mapsto & 101 \\ Pn & = & 01 & \mapsto & 011 \\ Pd & = & 11 & \mapsto & 110 \end{array}$$

Przykłady funkcji kodujących

$\mathcal{C}_1$: Wiadomość powtórzona

$$\begin{array}{rcccl} W & = & 00 & \mapsto & 0000 \\ Z & = & 10 & \mapsto & 1010 \\ Pn & = & 01 & \mapsto & 0101 \\ Pd & = & 11 & \mapsto & 1111 \end{array} \qquad \stackrel{Transmisja}{\Longrightarrow} \qquad 1101$$

$\mathcal{C}_2$: Wiadomość z jednym symbolem kontrolnym

$$\begin{array}{rcccl} W & = & 00 & \mapsto & 000 \\ Z & = & 10 & \mapsto & 101 \\ Pn & = & 01 & \mapsto & 011 \\ Pd & = & 11 & \mapsto & 110 \end{array} \qquad \stackrel{Transmisja}{\Longrightarrow} \qquad 111$$

Kodowanie z jednym symbolem kontrolnym

Numer PESEL

$$X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}$$

$X_1X_2X_3X_4X_5X_6$	–	*data*	*urodzenia*
$X_7X_8X_9$	–	*liczba*	*porządkowa*
X_{10}	–	K – *parzysta*	M – *nieparzysta*
X_{11}	–	*cyfra*	*kontrolna*

Kodowanie z jednym symbolem kontrolnym

Numer PESEL

$$X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}$$

$X_1X_2X_3X_4X_5X_6$	–	*data*	*urodzenia*
$X_7X_8X_9$	–	*liczba*	*porządkowa*
X_{10}	–	K – *parzysta*	M – *nieparzysta*
X_{11}	–	*cyfra*	*kontrolna*

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
1	3	7	9	1	3	7	9	1	3	1

Kodowanie z jednym symbolem kontrolnym

Numer PESEL

$$X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}$$

$X_1X_2X_3X_4X_5X_6$	–	*data*	*urodzenia*
$X_7X_8X_9$	–	*liczba*	*porządkowa*
X_{10}	–	K – *parzysta*	M – *nieparzysta*
X_{11}	–	*cyfra*	*kontrolna*

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
1	3	7	9	1	3	7	9	1	3	1

Funkcja kontrolna:

$$X_1\cdot 1+X_2\cdot 3+X_3\cdot 7+X_4\cdot 9+X_5\cdot 1+X_6\cdot 3+X_7\cdot 7+X_8\cdot 9+X_9\cdot 1+X_{10}\cdot 3+X_{11}\cdot 1$$

Numer PESEL

Przykład: 02070803628

Numer PESEL

Przykład: 02070803628

$$0\cdot 1+2\cdot 3+0\cdot 7+7\cdot 9+0\cdot 1+8\cdot 3+0\cdot 7+3\cdot 9+6\cdot 1+2\cdot 3+8\cdot 1$$

Numer PESEL

Przykład: 02070803628

$$0\cdot 1+2\cdot 3+0\cdot 7+7\cdot 9+0\cdot 1+8\cdot 3+0\cdot 7+3\cdot 9+6\cdot 1+2\cdot 3+8\cdot 1=140$$

Przykłady funkcji kodujących cd.

$\mathcal{C}_3$: Wiadomość dwukrotnie powtórzona

$$
\begin{array}{lcll}
W & = & 00 & \mapsto & 000000 \\
Z & = & 10 & \mapsto & 101010 \\
Pn & = & 01 & \mapsto & 010101 \\
Pd & = & 11 & \mapsto & 111111
\end{array}
$$

Przykłady funkcji kodujących cd.

$\mathcal{C}_3$: Wiadomość dwukrotnie powtórzona

$$
\begin{array}{lcll}
W & = & 00 & \mapsto 000000 \\
Z & = & 10 & \mapsto 101010 \\
Pn & = & 01 & \mapsto 010101 \\
Pd & = & 11 & \mapsto 111111
\end{array}
\quad \overset{Transmisja}{\Longrightarrow} \quad 110101
$$

Przykłady funkcji kodujących cd.

$\mathcal{C}_3$: Wiadomość dwukrotnie powtórzona

$$
\begin{array}{lcll}
W & = & 00 & \mapsto 000000 \\
Z & = & 10 & \mapsto 101010 \\
Pn & = & 01 & \mapsto 010101 \overset{Transmisja}{\Longrightarrow} 110101 \\
Pd & = & 11 & \mapsto 111111
\end{array}
$$

$$
Pn = 01 \mapsto 010101 \overset{Transmisja}{\Longrightarrow} 111101
$$

Schemat kodowania

Informacja

$c_1 c_2 \ldots c_k$

Schemat kodowania

Informacja

$c_1 c_2 \ldots c_k$ $\overset{\text{Kodowanie}}{\mapsto}$

Schemat kodowania

Informacja		*Inf .zakodowana*
$c_1 c_2 \ldots c_k$	$\overset{Kodowanie}{\mapsto}$	$c_1 c_2 \ldots c_k c_{k+1} \ldots c_n$

Schemat kodowania

Informacja $c_1 c_2 \ldots c_k$ $\overset{\textit{Kodowanie}}{\mapsto}$ *Inf .zakodowana* $c_1 c_2 \ldots c_k c_{k+1} \ldots c_n$ $\overset{\textit{Transmisja}}{\Longrightarrow}$

Schemat kodowania

Informacja

$c_1 c_2 \ldots c_k$

$\overset{\textit{Kodowanie}}{\mapsto}$

Inf .zakodowana

$c_1 c_2 \ldots c_k c_{k+1} \ldots c_n$

$\overset{\textit{Transmisja}}{\Longrightarrow}$

Inf .odebrana

$w_1 w_2 \ldots w_n$

Schemat kodowania

Informacja

$c_1 c_2 \ldots c_k$

$\overset{\textit{Kodowanie}}{\mapsto}$

Inf .zakodowana

$c_1 c_2 \ldots c_k c_{k+1} \ldots c_n$

$\overset{\textit{Transmisja}}{\Longrightarrow}$

Inf .odebrana

$w_1 w_2 \ldots w_n$

$\overset{\textit{Dekodowanie}}{\longrightarrow}$

Schemat kodowania

$$\underset{c_1 c_2 \ldots c_k}{\textit{Informacja}} \xrightarrow{\textit{Kodowanie}} \underset{c_1 c_2 \ldots c_k c_{k+1} \ldots c_n}{\textit{Inf .zakodowana}} \xRightarrow{\textit{Transmisja}}$$

$$\underset{w_1 w_2 \ldots w_n}{\textit{Inf .odebrana}} \xrightarrow{\textit{Dekodowanie}} \underset{f_1 f_2 \ldots f_k \overset{???}{=} c_1 c_2 \ldots c_k}{\textit{Inf .otrzymana}}$$

Pomysłodawca teorii kodowania

Richard Hamming

1915 - ur. w Chicago
1937 - dyplom University of Nebraska
1942 - dokorat na University of Illinois
II Wojna Światowa - Profesor na Universytecie w Louisville
1945 - Projekt Manhattan
1946-76 - Bell Telephone Laboratories
1950 - "Error detecting and error correcting codes"
1968 - Nagroda Turinga (Association for Computing Machinery)
1976-97 - Naval Postgraduate School
1998 - zm. w Monterey (Kalifornia)

KODY BINARNE

$c_i \in \{0, 1\}$

$$\underset{c_1 c_2 \ldots c_k}{\textit{Informacja}} \xrightarrow{\textit{Kodowanie}} \underset{c_1 \ldots c_k c_{k+1} \ldots c_n}{\textit{Wektor kodowy}}$$

KODY BINARNE

$c_i \in \{0, 1\}$

$$\underset{c_1 c_2 \ldots c_k}{\textit{Informacja}} \xrightarrow{\textit{Kodowanie}} \underset{c_1 \ldots c_k c_{k+1} \ldots c_n}{\textit{Wektor kodowy}}$$

Wektory kodowe

$\mathcal{C}_1 : 0000, 1010, 0101, 1111$

$\mathcal{C}_2 : 000, 101, 011, 110$

$\mathcal{C}_3 : 000000, 101010, 010101, 111111$

KODOWANIE

$c_i \in \{0, 1\}$

$$\underset{c_1 c_2 \ldots c_k}{\textit{Informacja}} \xrightarrow{\textit{Kodowanie}} \underset{c_1 \ldots c_k c_{k+1} \ldots c_n}{\textit{Wektor kodowy}}$$

KODOWANIE

$c_i \in \{0, 1\}$

$$\underset{c_1 c_2 \ldots c_k}{\textit{Informacja}} \xrightarrow{\textit{Kodowanie}} \underset{c_1 \ldots c_k c_{k+1} \ldots c_n}{\textit{Wektor kodowy}} \xRightarrow{\textit{Transmisja}} ????$$

KODOWANIE

$c_i \in \{0, 1\}$

$$\underset{c_1 c_2 \ldots c_k}{\textit{Informacja}} \quad \underset{\longrightarrow}{\textit{Kodowanie}} \quad \underset{c_1 \ldots c_k c_{k+1} \ldots c_n}{\textit{Wektor kodowy}} \quad \underset{\Longrightarrow}{\textit{Transmisja}} \quad ????$$

- Co to znaczy, że kod może wykryć lub poprawić błąd w przesyłanej informacji?

KODOWANIE

$c_i \in \{0, 1\}$

$$\underset{c_1 c_2 \ldots c_k}{\textit{Informacja}} \quad \underset{\longrightarrow}{\textit{Kodowanie}} \quad \underset{c_1 \ldots c_k c_{k+1} \ldots c_n}{\textit{Wektor kodowy}} \quad \underset{\Longrightarrow}{\textit{Transmisja}} \quad ????$$

- Co to znaczy, że kod może wykryć lub poprawić błąd w przesyłanej informacji?
- Dlaczego jedne kody tylko wykrywają błędy a inne mogą je również poprawić?

KODOWANIE

$c_i \in \{0, 1\}$

$$\underset{c_1 c_2 \ldots c_k}{\textit{Informacja}} \xrightarrow{\textit{Kodowanie}} \underset{c_1 \ldots c_k c_{k+1} \ldots c_n}{\textit{Wektor kodowy}} \xRightarrow{\textit{Transmisja}} ????$$

- Co to znaczy, że kod może wykryć lub poprawić błąd w przesyłanej informacji?
- Dlaczego jedne kody tylko wykrywają błędy a inne mogą je również poprawić?
- Dlaczego różne kody wykrywają lub poprawiają różne ilości błędów?

Waga

Wagą wektora $\mathbf{c} = c_1 \dots c_n$ nazywamy liczbę jedynek wśród symboli $c_1, \dots, c_n$ i oznaczamy $wt(\mathbf{c})$.

Waga

Wagą wektora $\mathbf{c} = c_1 \dots c_n$ nazywamy liczbę jedynek wśród symboli $c_1, \dots, c_n$ i oznaczamy $wt(\mathbf{c})$.

Przykład

$wt(1101) = 3$
$wt(0000) = 0$

Waga

Wagą wektora $\mathbf{c} = c_1 \dots c_n$ nazywamy liczbę jedynek wśród symboli $c_1, \dots, c_n$ i oznaczamy $wt(\mathbf{c})$.

Przykład

$wt(1101) = 3$
$wt(0000) = 0$

Odległość

Odległością między dwoma wektorami $\mathbf{c} = c_1 \dots c_n$ i $\mathbf{f} = f_1 \dots f_n$ nazywamy liczbę współrzędnych, na których wektory te się różnią i oznaczamy $d(\mathbf{c}, \mathbf{f})$.

Waga

Wagą wektora $\mathbf{c} = c_1 \dots c_n$ nazywamy liczbę jedynek wśród symboli $c_1, \dots, c_n$ i oznaczamy $wt(\mathbf{c})$.

Przykład

$wt(1101) = 3$
$wt(0000) = 0$

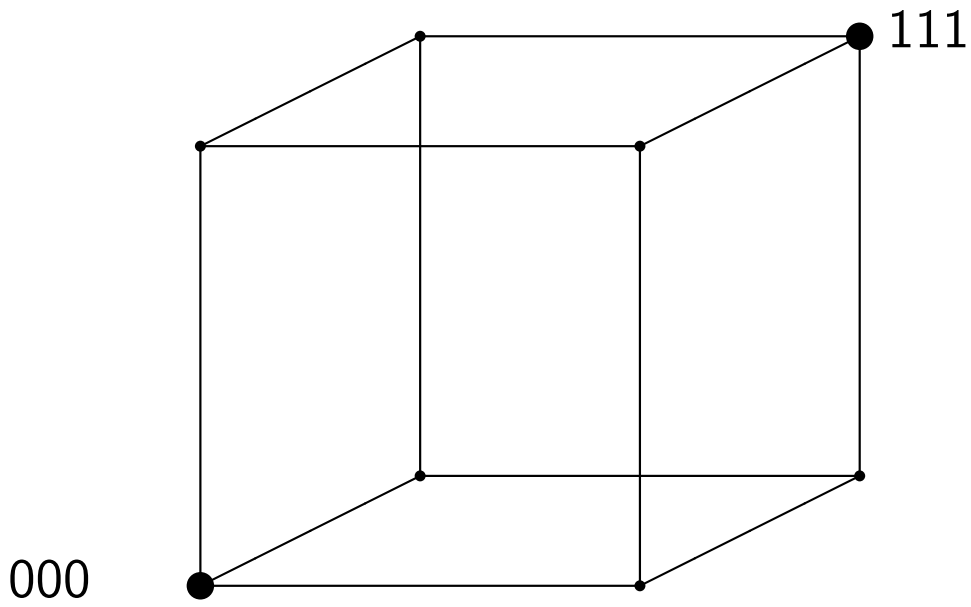
Odległość

Odległością między dwoma wektorami $\mathbf{c} = c_1 \dots c_n$ i $\mathbf{f} = f_1 \dots f_n$ nazywamy liczbę współrzędnych, na których wektory te się różnią i oznaczamy $d(\mathbf{c}, \mathbf{f})$.

Przykład

$d(1110, 1100) = 1$
$d(1010, 0000) = 2$

Kod długości $n = 3$



Odległość kodu

Odległością kodu $\mathcal{C}$ nazywamy liczbę

$$d := min\{d(\mathbf{c}, \mathbf{f}) \mid \mathbf{c}, \mathbf{f} \in \mathcal{C}, \mathbf{c} \neq \mathbf{f}\}.$$

Odległość kodu

Odległością kodu $\mathcal{C}$ nazywamy liczbę

$$d := min\{d(\mathbf{c}, \mathbf{f}) \mid \mathbf{c}, \mathbf{f} \in \mathcal{C}, \mathbf{c} \neq \mathbf{f}\}.$$

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\}$, $d = 2$

Odległość kodu

Odległością kodu $\mathcal{C}$ nazywamy liczbę

$$d := min\{d(\mathbf{c}, \mathbf{f}) \mid \mathbf{c}, \mathbf{f} \in \mathcal{C}, \mathbf{c} \neq \mathbf{f}\}.$$

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\},\ d = 2$
$\mathcal{C}_2 : \{000, 101, 011, 110\},\ d = 2$

Odległość kodu

Odległością kodu $\mathcal{C}$ nazywamy liczbę

$$d := min\{d(\mathbf{c}, \mathbf{f}) \mid \mathbf{c}, \mathbf{f} \in \mathcal{C}, \mathbf{c} \neq \mathbf{f}\}.$$

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\},\ d = 2$
$\mathcal{C}_2 : \{000, 101, 011, 110\},\ d = 2$
$\mathcal{C}_3 : \{000000, 101010, 010101, 111111\},\ d = 3$

Kule $K(\mathbf{c}, t)$

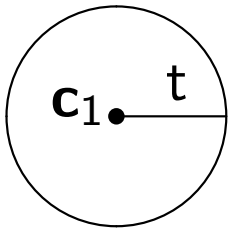
Jeśli odległość kodu $\mathcal{C}$ wynosi $d = 2t + 1$ to, że dowolne dwa słowa kodowe różnią się na co najmniej d miejscach.

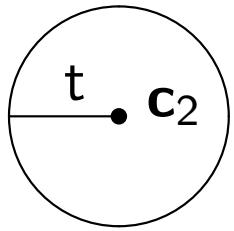
Kule $K(\mathbf{c}, t)$

Jeśli odległość kodu $\mathcal{C}$ wynosi $d = 2t + 1$ to, że dowolne dwa słowa kodowe różnią się na co najmniej d miejscach.
Kule $K(\mathbf{c}, t)$ o promieniu t wokół słów kodowych $\mathbf{c} \in \mathcal{C}$ są rozłączne.

Kule $K(\mathbf{c}, t)$

Jeśli odległość kodu $\mathcal{C}$ wynosi $d = 2t + 1$ to, że dowolne dwa słowa kodowe różnią się na co najmniej d miejscach.
Kule $K(\mathbf{c}, t)$ o promieniu t wokół słów kodowych $\mathbf{c} \in \mathcal{C}$ są rozłączne.

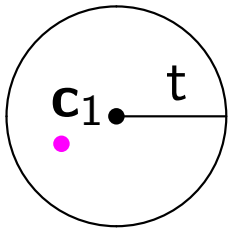


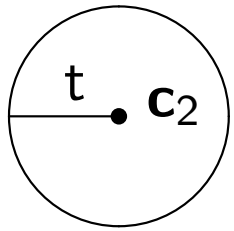


$\mathbf{c}_1 \overset{Transmisja}{\Longrightarrow}$????

Kule $K(\mathbf{c}, t)$

Jeśli odległość kodu $\mathcal{C}$ wynosi $d = 2t + 1$ to, że dowolne dwa słowa kodowe różnią się na co najmniej d miejscach.
Kule $K(\mathbf{c}, t)$ o promieniu t wokół słów kodowych $\mathbf{c} \in \mathcal{C}$ są rozłączne.





$\mathbf{c}_1 \overset{Transmisja}{\Longrightarrow}$????

Kule $K(\mathbf{c}, t)$

Jeśli odległość kodu $\mathcal{C}$ wynosi $d = 2t + 1$ to, że dowolne dwa słowa kodowe różnią się na co najmniej d miejscach.
Kule $K(\mathbf{c}, t)$ o promieniu t wokół słów kodowych $\mathbf{c} \in \mathcal{C}$ są rozłączne.

$\mathbf{c}_1 \overset{\textit{Transmisja}}{\Longrightarrow}$????

Kule $K(\mathbf{c}, t)$

Jeśli odległość kodu $\mathcal{C}$ wynosi $d = 2t + 1$ to, że dowolne dwa słowa kodowe różnią się na co najmniej d miejscach.
Kule $K(\mathbf{c}, t)$ o promieniu t wokół słów kodowych $\mathbf{c} \in \mathcal{C}$ są rozłączne.

$\mathbf{c}_1 \overset{\textit{Transmisja}}{\Longrightarrow}$????

Własności detekcyjne kodów

Twierdzenie

Warunkiem koniecznym i dostatecznym na to, aby kod $\mathcal{C}$ umożliwiał wykrycie t (lub mniej) błędów jest, aby odległość kodu była równa co najmniej $t + 1$.

Własności detekcyjne kodów

Twierdzenie

Warunkiem koniecznym i dostatecznym na to, aby kod $\mathcal{C}$ umożliwiał wykrycie t (lub mniej) błędów jest, aby odległość kodu była równa co najmniej $t + 1$.

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\}$, $d = 2$, wykrywa błędy pojedyncze

Własności detekcyjne kodów

Twierdzenie

Warunkiem koniecznym i dostatecznym na to, aby kod $\mathcal{C}$ umożliwiał wykrycie t (lub mniej) błędów jest, aby odległość kodu była równa co najmniej $t + 1$.

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\}$, $d = 2$, wykrywa błędy pojedyncze
$\mathcal{C}_2 : \{000, 101, 011, 110\}$, $d = 2$, wykrywa błędy pojedyncze

Własności detekcyjne kodów

Twierdzenie

Warunkiem koniecznym i dostatecznym na to, aby kod $\mathcal{C}$ umożliwiał wykrycie t (lub mniej) błędów jest, aby odległość kodu była równa co najmniej $t+1$.

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\}$, $d = 2$, wykrywa błędy pojedyncze
$\mathcal{C}_2 : \{000, 101, 011, 110\}$, $d = 2$, wykrywa błędy pojedyncze
$\mathcal{C}_3 : \{000000, 101010, 010101, 111111\}$, $d = 3$, wykrywa błędy podwójne

Własności korekcyjne kodów

Własności korekcyjne kodów

Twierdzenie

Kod $\mathcal{C}$ może poprawić co najwyżej t błędów wtedy i tylko wtedy, gdy jego odległość wynosi co najmniej $d = 2t + 1$.

Własności korekcyjne kodów

Twierdzenie

Kod $\mathcal{C}$ może poprawić co najwyżej t błędów wtedy i tylko wtedy, gdy jego odległość wynosi co najmniej $d = 2t + 1$.

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\}$, $d = 2$, nie poprawia błędów

Własności korekcyjne kodów

Twierdzenie

Kod $\mathcal{C}$ może poprawić co najwyżej t błędów wtedy i tylko wtedy, gdy jego odległość wynosi co najmniej $d = 2t + 1$.

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\}$, $d = 2$, nie poprawia błędów
$\mathcal{C}_2 : \{000, 101, 011, 110\}$, $d = 2$, nie poprawia błędów

Własności korekcyjne kodów

Twierdzenie

Kod $\mathcal{C}$ może poprawić co najwyżej t błędów wtedy i tylko wtedy, gdy jego odległość wynosi co najmniej $d = 2t + 1$.

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\}$, $d = 2$, nie poprawia błędów
$\mathcal{C}_2 : \{000, 101, 011, 110\}$, $d = 2$, nie poprawia błędów
$\mathcal{C}_3 : \{000000, 101010, 010101, 111111\}$, $d = 3$, poprawia błędy pojedyncze

Własności korekcyjne kodów

Twierdzenie

Kod $\mathcal{C}$ może poprawić co najwyżej t błędów wtedy i tylko wtedy, gdy jego odległość wynosi co najmniej $d = 2t + 1$.

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\}$, $d = 2$, nie poprawia błędów
$\mathcal{C}_2 : \{000, 101, 011, 110\}$, $d = 2$, nie poprawia błędów
$\mathcal{C}_3 : \{000000, 101010, 010101, 111111\}$, $d = 3$, poprawia błędy pojedyncze

Kod $\mathcal{C}_3$

$$010101 \overset{Transmisja}{\Longrightarrow} 110101$$

Własności korekcyjne kodów

Twierdzenie

Kod $\mathcal{C}$ może poprawić co najwyżej t błędów wtedy i tylko wtedy, gdy jego odległość wynosi co najmniej $d = 2t + 1$.

Przykład

$\mathcal{C}_1 : \{0000, 1010, 0101, 1111\}$, $d = 2$, nie poprawia błędów
$\mathcal{C}_2 : \{000, 101, 011, 110\}$, $d = 2$, nie poprawia błędów
$\mathcal{C}_3 : \{000000, 101010, 010101, 111111\}$, $d = 3$, poprawia błędy pojedyncze

Kod $\mathcal{C}_3$

$$010101 \overset{Transmisja}{\Longrightarrow} 110101$$

$$d(110101, 000000) = 4, \quad d(110101, 101010) = 5$$
$$d(110101, 010101) = 1, \quad d(110101, 111111) = 3$$

Przykłady kodów binarnych

Kod powtórzeniowy

Binarny kod powtórzeniowy (długości n) ma tylko dwa słowa kodowe:

$$0 \mapsto \underbrace{00\ldots0}_{n-razy}$$

$$1 \mapsto \underbrace{11\ldots1}_{n-razy}$$

Przykłady kodów binarnych

Kod powtórzeniowy

Binarny kod powtórzeniowy (długości n) ma tylko dwa słowa kodowe:

$$0 \mapsto \underbrace{00\ldots0}_{n-razy}$$

$$1 \mapsto \underbrace{11\ldots1}_{n-razy}$$

Odległość: $d = n$

Przykłady kodów binarnych

Kod powtórzeniowy

Binarny kod powtórzeniowy (długości n) ma tylko dwa słowa kodowe:

$$0 \mapsto \underbrace{00\ldots0}_{n-razy}$$

$$1 \mapsto \underbrace{11\ldots1}_{n-razy}$$

Odległość: $d = n$
Wykrywa: $n - 1$ błędów

Przykłady kodów binarnych

Kod powtórzeniowy

Binarny kod powtórzeniowy (długości n) ma tylko dwa słowa kodowe:

$$0 \mapsto \underbrace{00\ldots0}_{n-razy}$$

$$1 \mapsto \underbrace{11\ldots1}_{n-razy}$$

Odległość: $d = n$
Wykrywa: $n-1$ błędów
Poprawia: dla $n = 2k+1 \to k$ błędów; dla $n = 2k \to k-1$ błędów

Przykłady kodów binarnych

Kod powtórzeniowy

Binarny kod powtórzeniowy (długości n) ma tylko dwa słowa kodowe:

$$0 \mapsto \underbrace{00\ldots0}_{n-razy}$$

$$1 \mapsto \underbrace{11\ldots1}_{n-razy}$$

Odległość: $d = n$
Wykrywa: $n-1$ błędów
Poprawia: dla $n = 2k+1 \to k$ błędów; dla $n = 2k \to k-1$ błędów
Dekodowanie: Jako informację wysłaną przyjmuje ten ciąg, w którym dany symbol wystąpił na większej liczbie pozycji

Przykłady kodów binarnych cd.

Kod kontroli parzystości

Binarny kod kontroli parzystości długości n powstaje przez dodanie na końcu każdej wysyłanej wiadomości $c_1 \dots c_{n-1}$, jednego dodatkowego symbolu c_n:

$$c_n := \begin{cases} 1, \text{ gdy suma } \sum_{i=1}^{n-1} c_i \text{ jest nieparzysta}, \\ 0, \text{ gdy suma } \sum_{i=1}^{n-1} c_i \text{ jest parzysta} \end{cases}$$

Przykłady kodów binarnych cd.

Kod kontroli parzystości

Binarny kod kontroli parzystości długości n powstaje przez dodanie na końcu każdej wysyłanej wiadomości $c_1 \dots c_{n-1}$, jednego dodatkowego symbolu c_n:

$$c_n := \begin{cases} 1, \text{ gdy suma } \sum_{i=1}^{n-1} c_i \text{ jest nieparzysta}, \\ 0, \text{ gdy suma } \sum_{i=1}^{n-1} c_i \text{ jest parzysta} \end{cases}$$

Przykład dla $n = 3$

Kod $\mathcal{C}_2$ jest kodem kontroli parzystości długości $n = 3$:
$00 \mapsto 000$, $10 \mapsto 101$, $01 \mapsto 011$, $11 \mapsto 110$

Przykłady kodów binarnych cd.

Kod kontroli parzystości

Binarny kod kontroli parzystości długości n powstaje przez dodanie na końcu każdej wysyłanej wiadomości $c_1 \dots c_{n-1}$, jednego dodatkowego symbolu c_n:

$$c_n := \begin{cases} 1, \text{ gdy suma } \sum_{i=1}^{n-1} c_i \text{ jest nieparzysta}, \\ 0, \text{ gdy suma } \sum_{i=1}^{n-1} c_i \text{ jest parzysta} \end{cases}$$

Przykład dla $n = 3$

Kod $\mathcal{C}_2$ jest kodem kontroli parzystości długości $n = 3$:
$00 \mapsto 000$, $10 \mapsto 101$, $01 \mapsto 011$, $11 \mapsto 110$
Odległość: $d = 2$

Przykłady kodów binarnych cd.

Kod kontroli parzystości

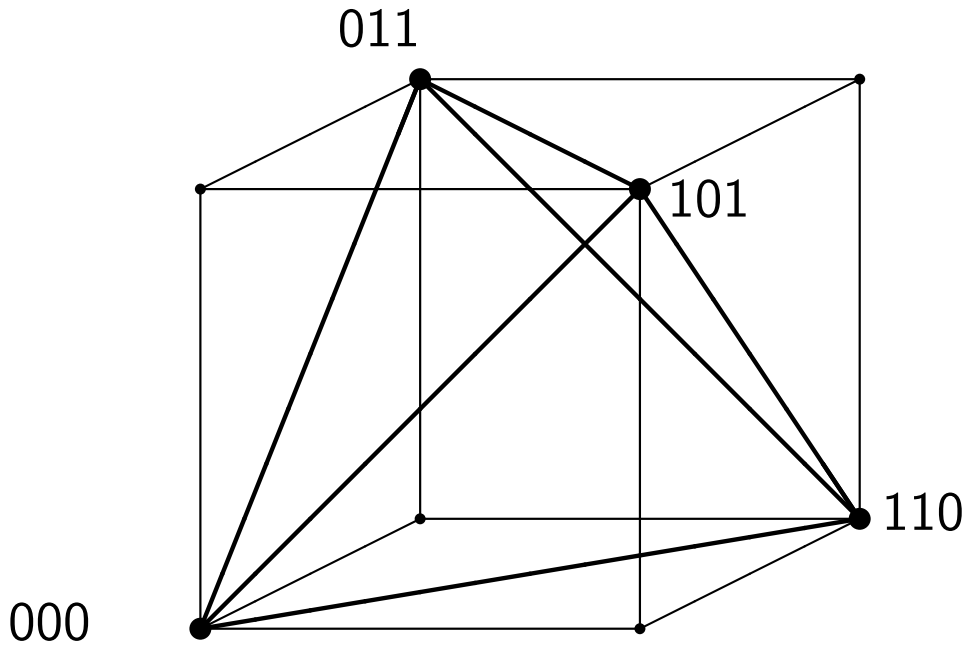
Binarny kod kontroli parzystości długości n powstaje przez dodanie na końcu każdej wysyłanej wiadomości $c_1 \dots c_{n-1}$, jednego dodatkowego symbolu c_n:

$$c_n := \begin{cases} 1, \text{ gdy suma } \sum_{i=1}^{n-1} c_i \text{ jest nieparzysta}, \\ 0, \text{ gdy suma } \sum_{i=1}^{n-1} c_i \text{ jest parzysta} \end{cases}$$

Przykład dla $n = 3$

Kod $\mathcal{C}_2$ jest kodem kontroli parzystości długości $n = 3$:
$00 \mapsto 000$, $10 \mapsto 101$, $01 \mapsto 011$, $11 \mapsto 110$
Odległość: $d = 2$
Wykrywa błędy pojedyncze

Kod kontroli parzystości



Binarna reprezantacja liczby n

$m \in \mathbb{N}$

Każdą liczbę naturalną $0 \leq n \leq 2^m - 1$ można przedstawić w postci:

$$n = a_0 \cdot 2^0 + a_1 \cdot 2^1 + \ldots + a_{m-1} \cdot 2^{m-1}.$$

Binarna reprezantacja liczby n

$m \in \mathbb{N}$

Każdą liczbę naturalną $0 \leq n \leq 2^m - 1$ można przedstawić w postci:

$$n = a_0 \cdot 2^0 + a_1 \cdot 2^1 + \ldots + a_{m-1} \cdot 2^{m-1}.$$

Binarny ciąg długości m

$$a_0 a_1 \ldots a_{m-1}$$

nazywamy *binarną reprezantacją liczby n* (długości m)

Binarna reprezantacja liczby n

$m \in \mathbb{N}$

Każdą liczbę naturalną $0 \leq n \leq 2^m - 1$ można przedstawić w postci:

$$n = a_0 \cdot 2^0 + a_1 \cdot 2^1 + \ldots + a_{m-1} \cdot 2^{m-1}.$$

Binarny ciąg długości m

$$a_0 a_1 \ldots a_{m-1}$$

nazywamy *binarną reprezantacją liczby n* (długości m)

Binarna reprezentacja liczby $6 \leq 2^3 - 1$ długości $m = 3$

$$6 = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^4$$

Binarna reprezantacja liczby n

$m \in \mathbb{N}$

Każdą liczbę naturalną $0 \leq n \leq 2^m - 1$ można przedstawić w postci:

$$n = a_0 \cdot 2^0 + a_1 \cdot 2^1 + \ldots + a_{m-1} \cdot 2^{m-1}.$$

Binarny ciąg długości m

$$a_0 a_1 \ldots a_{m-1}$$

nazywamy *binarną reprezantacją liczby n* (długości m)

Binarna reprezentacja liczby $6 \leq 2^3 - 1$ długości $m = 3$

$$6 = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^4$$

Binarna reprezentacja: $a_0 a_1 a_2 = 011$

Kod Hamminga $\mathcal{H}_m$

$m \in \mathbb{N},\ n = 2^m - 1,\ 1 \leq i \leq 2^m - 1$

$$H_m = \begin{matrix} w_{11} & \dots & w_{1i} & \dots & w_{1n} \\ w_{21} & \dots & w_{2i} & \dots & w_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ w_{j1} & \dots & w_{ji} & \dots & w_{jn} \\ \dots & \dots & \dots & \dots & \dots \\ w_{m1} & \dots & w_{mi} & \dots & w_{mn} \end{matrix}$$

$\Uparrow$

Kod Hamminga $\mathcal{H}_m$

$m \in \mathbb{N},\ n = 2^m - 1,\ 1 \leq i \leq 2^m - 1$

$$
H_m = \begin{array}{ccccccc}
w_{11} & \dots & w_{1i} & \dots & w_{1n} & \leftarrow & \mathbf{w}_1 \\
w_{21} & \dots & w_{2i} & \dots & w_{2n} & \leftarrow & \mathbf{w}_2 \\
\dots & \dots & \dots & \dots & \dots & & \\
w_{j1} & \dots & w_{ji} & \dots & w_{jn} & \leftarrow & \mathbf{w}_j \\
\dots & \dots & \dots & \dots & \dots & & \\
w_{m1} & \dots & w_{mi} & \dots & w_{mn} & \leftarrow & \mathbf{w}_m \\
 & & \Uparrow & & & &
\end{array}
$$

Kod Hamminga $\mathcal{H}_m$

$m \in \mathbb{N},\ n = 2^m - 1,\ 1 \leq i \leq 2^m - 1$

$$
H_m = \begin{array}{ccccccc}
w_{11} & \ldots & w_{1i} & \ldots & w_{1n} & \leftarrow & \mathbf{w}_1 \\
w_{21} & \ldots & w_{2i} & \ldots & w_{2n} & \leftarrow & \mathbf{w}_2 \\
\ldots & \ldots & \ldots & \ldots & \ldots & & \\
w_{j1} & \ldots & w_{ji} & \ldots & w_{jn} & \leftarrow & \mathbf{w}_j \\
\ldots & \ldots & \ldots & \ldots & \ldots & & \\
w_{m1} & \ldots & w_{mi} & \ldots & w_{mn} & \leftarrow & \mathbf{w}_m \\
 & & \Uparrow & & & &
\end{array}
$$

$$
\begin{aligned}
w_{11}c_1 + w_{12}c_2 + \ldots + w_{1n}c_n &= 0 \\
\ldots\ldots\ldots\ldots &= 0 \\
w_{j1}c_1 + w_{j2}c_2 + \ldots + w_{jn}c_n &= 0 \\
\ldots\ldots\ldots\ldots &= 0 \\
w_{m1}c_1 + w_{m2}c_2 + \ldots + w_{mn}c_n &= 0
\end{aligned}
$$

Kod Hamminga $\mathcal{H}_m$

$$\begin{aligned} w_{11}c_1 + w_{12}c_2 + \ldots + w_{1n}c_n &= 0 \\ \ldots\ldots\ldots\ldots &= 0 \\ w_{j1}c_1 + w_{j2}c_2 + \ldots + w_{jn}c_n &= 0 \\ \ldots\ldots\ldots\ldots &= 0 \\ w_{m1}c_1 + w_{m2}c_2 + \ldots + w_{mn}c_n &= 0 \end{aligned} \qquad (1)$$

Definicja

Kodem Hamminga $\mathcal{H}_m$ nazywamy kod długości $n = 2^m - 1$ o m symbolach sprawdzających: $c_1, c_2, c_4, \ldots, c_{2^{m-1}}$, dla którego $\mathbf{c} = c_1 \ldots c_n \in \mathcal{H}_m$ wtedy i tylko wtedy, gdy $c_1 \ldots c_n$ spełniają każde z równań (1).

Kod Hamminga $\mathcal{H}_m$

Podstawowe własności

Kod Hamminga $\mathcal{H}_m$

Podstawowe własności

- m symboli kontrolnych: $c_1, c_2, c_4, \ldots, c_{2^{m-1}}$

Kod Hamminga $\mathcal{H}_m$

Podstawowe własności

- m symboli kontrolnych: $c_1, c_2, c_4, \ldots, c_{2^{m-1}}$
- $n - m$ symboli wiadomości

Kod Hamminga $\mathcal{H}_m$

Podstawowe własności

- m symboli kontrolnych: $c_1, c_2, c_4, \ldots, c_{2^{m-1}}$
- $n - m$ symboli wiadomości
- 2^{n-m} słów kodowych

Kod Hamminga $\mathcal{H}_m$

Podstawowe własności

- m symboli kontrolnych: $c_1, c_2, c_4, \ldots, c_{2^{m-1}}$
- $n - m$ symboli wiadomości
- 2^{n-m} słów kodowych
- Wykrywa błędy podwójne oraz poprawia błędy pojedyncze ($d = 3$)

Kod Hamminga $\mathcal{H}_2$

$m = 2,\ n = 2^2 - 1 = 3$

Kod Hamminga $\mathcal{H}_2$

$m = 2$, $n = 2^2 - 1 = 3$

Binarne reprezentacje długości $m = 2$ liczb:

$1 \rightsquigarrow 10$, $2 \rightsquigarrow 01$, $3 \rightsquigarrow 11$

Kod Hamminga $\mathcal{H}_2$

$m = 2$, $n = 2^2 - 1 = 3$

Binarne reprezentacje długości $m = 2$ liczb:

$1 \rightsquigarrow 10$, $2 \rightsquigarrow 01$, $3 \rightsquigarrow 11$

$$H_2 = \begin{array}{ccccc} 0 & 1 & 1 & = & \mathbf{w}_1 \\ 1 & 0 & 1 & = & \mathbf{w}_2 \end{array}$$

Kod Hamminga $\mathcal{H}_2$

$m = 2$, $n = 2^2 - 1 = 3$

Binarne reprezentacje długości $m = 2$ liczb:

$1 \rightsquigarrow 10$, $2 \rightsquigarrow 01$, $3 \rightsquigarrow 11$

$$H_2 = \begin{array}{ccccc} 0 & 1 & 1 & = & \mathbf{w}_1 \\ 1 & 0 & 1 & = & \mathbf{w}_2 \end{array}$$

c_3 - symbol wiadomości; c_1, c_2 - symbole kontrolne

$$\begin{array}{ccccccc} 0 \cdot c_1 & + & 1 \cdot c_2 & + & 1 \cdot c_3 & = & 0 \\ 1 \cdot c_1 & + & 0 \cdot c_2 & + & 1 \cdot c_3 & = & 0 \end{array} \Leftrightarrow \begin{array}{ccccc} c_2 & + & c_3 & = & 0 \\ c_1 & + & c_3 & = & 0 \end{array}$$

Kod Hamminga $\mathcal{H}_2$

$m = 2$, $n = 2^2 - 1 = 3$

Binarne reprezentacje długości $m = 2$ liczb:

$1 \rightsquigarrow 10$, $2 \rightsquigarrow 01$, $3 \rightsquigarrow 11$

$$H_2 = \begin{array}{ccccc} 0 & 1 & 1 & = & \mathbf{w}_1 \\ 1 & 0 & 1 & = & \mathbf{w}_2 \end{array}$$

c_3 - symbol wiadomości; c_1, c_2 - symbole kontrolne

$$\begin{array}{ccccccc} 0 \cdot c_1 & + & 1 \cdot c_2 & + & 1 \cdot c_3 & = & 0 \\ 1 \cdot c_1 & + & 0 \cdot c_2 & + & 1 \cdot c_3 & = & 0 \end{array} \Leftrightarrow \begin{array}{ccccc} c_2 & + & c_3 & = & 0 \\ c_1 & + & c_3 & = & 0 \end{array} \Leftrightarrow$$

$$c_1 = c_2 = c_3$$

Kod Hamminga $\mathcal{H}_2$

$m = 2$, $n = 2^2 - 1 = 3$

Binarne reprezentacje długości $m = 2$ liczb:

$1 \rightsquigarrow 10$, $2 \rightsquigarrow 01$, $3 \rightsquigarrow 11$

$$H_2 = \begin{array}{ccccc} 0 & 1 & 1 & = & \mathbf{w}_1 \\ 1 & 0 & 1 & = & \mathbf{w}_2 \end{array}$$

c_3 - symbol wiadomości; c_1, c_2 - symbole kontrolne

$$\begin{array}{ccccccc} 0 \cdot c_1 & + & 1 \cdot c_2 & + & 1 \cdot c_3 & = & 0 \\ 1 \cdot c_1 & + & 0 \cdot c_2 & + & 1 \cdot c_3 & = & 0 \end{array} \Leftrightarrow \begin{array}{ccccc} c_2 & + & c_3 & = & 0 \\ c_1 & + & c_3 & = & 0 \end{array} \Leftrightarrow$$

$$c_1 = c_2 = c_3$$

$\mathcal{H}_2$

Kod Hamminga $\mathcal{H}_2$ jest kodem powtórzeniowym długości $n = 3$

Dekodowanie kodów Hamminga

$c_1 c_2 c_3 \in \mathcal{H}_2 \Rightarrow$

$$H_2 = \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{matrix} \quad \rightrightarrows \quad \begin{matrix} c_2 + c_3 = 0 \\ c_1 + c_3 = 0 \end{matrix}$$

Dekodowanie kodów Hamminga

$c_1 c_2 c_3 \in \mathcal{H}_2 \Rightarrow$

$$H_2 = \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{matrix} \quad \rightrightarrows \quad \begin{matrix} c_2 + c_3 = 0 \\ c_1 + c_3 = 0 \end{matrix}$$

$000 \rightsquigarrow 001 = f_1 f_2 f_3 = c_1 c_2 (c_3 + 1)$

Dekodowanie kodów Hamminga

$c_1 c_2 c_3 \in \mathcal{H}_2 \Rightarrow$

$$H_2 = \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{matrix} \quad \rightrightarrows \quad \begin{matrix} c_2 & + & c_3 & = & 0 \\ c_1 & + & c_3 & = & 0 \end{matrix}$$

$000 \rightsquigarrow 001 = f_1 f_2 f_3 = c_1 c_2 (c_3 + 1) \Rightarrow$

$$\begin{matrix} f_2 & + & f_3 & = & c_2 & + & c_3 + 1 \\ f_1 & + & f_3 & = & c_1 & + & c_3 + 1 \end{matrix}$$

Dekodowanie kodów Hamminga

$c_1 c_2 c_3 \in \mathcal{H}_2 \Rightarrow$

$$H_2 = \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ & & \Uparrow \end{matrix} \quad \rightrightarrows \quad \begin{matrix} c_2 + c_3 = 0 \\ c_1 + c_3 = 0 \end{matrix}$$

$000 \rightsquigarrow 001 = f_1 f_2 f_3 = c_1 c_2 (c_3 + 1) \Rightarrow$

$$\begin{matrix} f_2 + f_3 = c_2 + c_3 + 1 = 1 \\ f_1 + f_3 = c_1 + c_3 + 1 = 1 \end{matrix}$$

Kody praktycznie stosowane

Kody BCH (R.C. Bose, D.K. Ray-Chaudhuri, A. Hocquenghema): Rodzina kodów poprawiających błędy wielokrotne, uogólnienie kodów Hamminga.

Kody praktycznie stosowane

Kody BCH (R.C. Bose, D.K. Ray-Chaudhuri, A. Hocquenghema): Rodzina kodów poprawiających błędy wielokrotne, uogólnienie kodów Hamminga.

Kody BCH

Jeden z pierwszych praktycznie realizowanych ważnych kodów był kodem BCH, miał długość $n = 127$, w tym 35 symboli kontrolnych i składał się z 2^{92} słów kodowych. Miał odległość $d = 11$, czyli poprawiał do 5 błędów.

Kody praktycznie stosowane

Kody BCH (R.C. Bose, D.K. Ray-Chaudhuri, A. Hocquenghema): Rodzina kodów poprawiających błędy wielokrotne, uogólnienie kodów Hamminga.

Kody BCH

Jeden z pierwszych praktycznie realizowanych ważnych kodów był kodem BCH, miał długość $n = 127$, w tym 35 symboli kontrolnych i składał się z 2^{92} słów kodowych. Miał odległość $d = 11$, czyli poprawiał do 5 błędów.
Powszechnie stosowanym w europejskich systemach przesyłania danych jest binarny kod BCH długości $n = 255$, o 24 symbolach kontrolnych. Ten kod ma 2^{231} słów kodowych a jego odległośc wynosi $d = 7$, tzn. poprawia błędy potrójne.

Kody praktycznie stosowane - Kod Reeda - Mullera

1972 - Sonda Mariner: zdjęcia z Marsa

Zdjęcia czarno-białe: każdy "punkt" zdjęcia był opisany przez jeden z 64 odcieni szarości.
Każdy odcień szarości zapisano jako ciąg binarny długości 6. $(64 = 2^6)$
Kodowanie: do wiadomości długości 6 dopisano 26 symboli kontrolnych. $(n = 32)$
Kod poprawia 7 błędów.

Kody praktycznie stosowane - Kod Goley'a

1979 - Sonda Voyager: zdjęcia z Jupitera

Zdjęcia kolorowe: każdy "punkt" zdjęcia był opisany przez jeden z 4096 kolorów.
Każdy kolor zapisano jako ciąg binarny długości 12. ($4096 = 2^{12}$)
Kodowanie: do wiadomości długości 12 dopisano kolejne 12 symboli kontrolnych. ($n = 24$)
Kod poprawia 3 błędy.

Kody doskonałe

Przykład

Kod Hamminga $\mathcal{H}_3$ $(m = 3)$ długości $n = 2^3 - 1 = 7$ ma $2^{7-3} = 2^4 = 16$ słów kodowych.

Kody doskonałe

Przykład

Kod Hamminga $\mathcal{H}_3$ $(m = 3)$ długości $n = 2^3 - 1 = 7$ ma $2^{7-3} = 2^4 = 16$ słów kodowych.

Każda kula $K(\mathbf{c}, 1)$, $\mathbf{c} \in \mathcal{H}_3$ zawiera 8 wektorów binarnych długości 7.

Kody doskonałe

Przykład

Kod Hamminga $\mathcal{H}_3$ $(m = 3)$ długości $n = 2^3 - 1 = 7$ ma $2^{7-3} = 2^4 = 16$ słów kodowych.

Każda kula $K(\mathbf{c}, 1)$, $\mathbf{c} \in \mathcal{H}_3$ zawiera 8 wektorów binarnych długości 7.

We wszystkich takich kulach mieści się $8 \cdot 16 = 128 = 2^7$ wektorów binarnych długości 7.

Kody doskonałe

Przykład

Kod Hamminga $\mathcal{H}_3$ $(m = 3)$ długości $n = 2^3 - 1 = 7$ ma $2^{7-3} = 2^4 = 16$ słów kodowych.
Każda kula $K(\mathbf{c}, 1)$, $\mathbf{c} \in \mathcal{H}_3$ zawiera 8 wektorów binarnych długości 7.
We wszystkich takich kulach mieści się $8 \cdot 16 = 128 = 2^7$ wektorów binarnych długości 7.

Definicja

Binarny kod $\mathcal{C}$ długości n, który poprawia t błędów nazywamy **kodem doskonałym**, jeśli wszystkie binarne wektory długości n mieszczą się w kulach $K(\mathbf{c}, t)$, dla $\mathbf{c} \in \mathcal{C}$.

Kody doskonałe - przykłady

Przykład

Wszystkie binarne kody Hamminga są doskonałe.

Kody doskonałe - przykłady

Przykład

Wszystkie binarne kody Hamminga są doskonałe.

Trywialnymi kodami doskonałymi długości n są:

- kod zawierający dokładnie jedno słowo kodowe (poprawia wszystkie popełnione błędy)

Kody doskonałe - przykłady

Przykład

Wszystkie binarne kody Hamminga są doskonałe.

Trywialnymi kodami doskonałymi długości n są:

- kod zawierający dokładnie jedno słowo kodowe (poprawia wszystkie popełnione błędy)
- kod zawierający wszystkie binarne wektory długości n (nie poprawia żadnego błędu)

Kody doskonałe - przykłady

Przykład

Wszystkie binarne kody Hamminga są doskonałe.

Trywialnymi kodami doskonałymi długości n są:

- kod zawierający dokładnie jedno słowo kodowe (poprawia wszystkie popełnione błędy)
- kod zawierający wszystkie binarne wektory długości n (nie poprawia żadnego błędu)
- binarny kod powtórzeniowy dla n nieparzystego

Binarne nietrywialne kody doskonałe

Kod $\mathcal{C}$ (długości n) nazywamy *liniowym*, jeśli dla $\mathbf{c}, \mathbf{f} \in \mathcal{C}$, również

$$\mathbf{c} + \mathbf{f} := c_1 +_2 f_1 \ldots c_n +_2 f_n \in \mathcal{C}$$

Binarne nietrywialne kody doskonałe

Kod $\mathcal{C}$ (długości n) nazywamy *liniowym*, jeśli dla $\mathbf{c}, \mathbf{f} \in \mathcal{C}$, również

$$\mathbf{c} + \mathbf{f} := c_1 +_2 f_1 \ldots c_n +_2 f_n \in \mathcal{C}$$

- Binarne kody Hammiga $\mathcal{H}_m$ są jedynymi liniowymi kodami doskonałymi poprawiającymi błędy pojedyncze

Binarne nietrywialne kody doskonałe

Kod $\mathcal{C}$ (długości n) nazywamy *liniowym*, jeśli dla $\mathbf{c}, \mathbf{f} \in \mathcal{C}$, również

$$\mathbf{c} + \mathbf{f} := c_1 +_2 f_1 \ldots c_n +_2 f_n \in \mathcal{C}$$

- Binarne kody Hammiga $\mathcal{H}_m$ są jedynymi liniowymi kodami doskonałymi poprawiającymi błędy pojedyncze
- Istnieje ogromnie wiele nieliniowych kodów doskonałych poprawiających błędy pojedyncze

Binarne nietrywialne kody doskonałe

Kod $\mathcal{C}$ (długości n) nazywamy *liniowym*, jeśli dla $\mathbf{c}, \mathbf{f} \in \mathcal{C}$, również

$$\mathbf{c} + \mathbf{f} := c_1 +_2 f_1 \ldots c_n +_2 f_n \in \mathcal{C}$$

- Binarne kody Hammiga $\mathcal{H}_m$ są jedynymi liniowymi kodami doskonałymi poprawiającymi błędy pojedyncze
- Istnieje ogromnie wiele nieliniowych kodów doskonałych poprawiających błędy pojedyncze
- Nie istnieją binarne kody doskonałe poprawiające błędy podwójne

Binarne nietrywialne kody doskonałe

Kod $\mathcal{C}$ (długości n) nazywamy *liniowym*, jeśli dla $\mathbf{c}, \mathbf{f} \in \mathcal{C}$, również

$$\mathbf{c} + \mathbf{f} := c_1 +_2 f_1 \ldots c_n +_2 f_n \in \mathcal{C}$$

- Binarne kody Hammiga $\mathcal{H}_m$ są jedynymi liniowymi kodami doskonałymi poprawiającymi błędy pojedyncze
- Istnieje ogromnie wiele nieliniowych kodów doskonałych poprawiających błędy pojedyncze
- Nie istnieją binarne kody doskonałe poprawiające błędy podwójne
- Kod długości $n = 23$ zawierający 2^{12} słów kodowych (powstaje z kodu Goley'a o długości 24) jest jedynym binarnym kodem doskonałym poprawiającym błędy potrójne

Binarne nietrywialne kody doskonałe

Kod $\mathcal{C}$ (długości n) nazywamy *liniowym*, jeśli dla $\mathbf{c}, \mathbf{f} \in \mathcal{C}$, również

$$\mathbf{c} + \mathbf{f} := c_1 +_2 f_1 \ldots c_n +_2 f_n \in \mathcal{C}$$

- Binarne kody Hammiga $\mathcal{H}_m$ są jedynymi liniowymi kodami doskonałymi poprawiającymi błędy pojedyncze
- Istnieje ogromnie wiele nieliniowych kodów doskonałych poprawiających błędy pojedyncze
- Nie istnieją binarne kody doskonałe poprawiające błędy podwójne
- Kod długości $n = 23$ zawierający 2^{12} słów kodowych (powstaje z kodu Goley'a o długości 24) jest jedynym binarnym kodem doskonałym poprawiającym błędy potrójne
- Nie ma innych binarnych kodów doskonałych poprawiających błędy wielokrotne

Bardzo dziekuję za uwagę