Detekcja i korekcja błędów. Kody liniowe. Kody Hamminga

Kodowanie i kompresja informacji

Maciek Gębala

31 maja 2010

Kodowanie

- Macierz generująca Macierz przez którą mnożymy bity informacji aby otrzymać kod.

Kodowanie

- Macierz generująca Macierz przez którą mnożymy bity informacji aby otrzymać kod.
- Macierz parzystości Macierz przez którą mnożymy kod aby sprawdzić czy jest poprawny (wynik mnożenia daje wynik zerowy).

Kodowanie

- Macierz generująca Macierz przez którą mnożymy bity informacji aby otrzymać kod.
- Macierz parzystości Macierz przez którą mnożymy kod aby sprawdzić czy jest poprawny (wynik mnożenia daje wynik zerowy).
- Syndrom Niezerowy wynik pomnożenia przez macierz parzystości wraz z ewentualnym opisem jak skorygować powstały błąd.

Kod parzystości

- Macierz generująca rozmiaru $(n+1) \times n$

$$G = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

Kod parzystości

- Macierz generująca rozmiaru $(n+1) \times n$

$$G = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

- Dla wektora informacji x kod k liczymy następująco:

$$k = Gx$$

Kod parzystości

- Macierz generująca rozmiaru $(n+1) \times n$

$$G = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

- Dla wektora informacji x kod k liczymy następująco:

$$k = Gx$$

- Macierz parzystości rozmiaru $1 \times (n+1)$

$$H = [1\ 1\ \cdots\ 1]$$

Kod parzystości

- Macierz generująca rozmiaru $(n+1) \times n$

$$G = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

- Dla wektora informacji x kod k liczymy następująco:

$$k = Gx$$

- Macierz parzystości rozmiaru $1 \times (n+1)$

$$H = [1\ 1\ \cdots\ 1]$$

- Jeden syndrom (1) oznaczający błąd w kodzie.

Kod powtórzeniowy długości 5

- Macierz generująca

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Kod powtórzeniowy długości 5

- Macierz generująca

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

- Macierz parzystości

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Kod powtórzeniowy długości 5

- Macierz generująca

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

- Macierz parzystości

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Korekcja na zasadzie większości.

Binarne kody liniowe

- Kod binarny K nazywamy liniowym jeśli

Binarne kody liniowe

- Kod binarny K nazywamy liniowym jeśli
  1. suma dwóch słów kodowych z K także należy do K

Binarne kody liniowe

- Kod binarny K nazywamy liniowym jeśli
  1. suma dwóch słów kodowych z K także należy do K
  2. pomnożenie słowa kodowego przez skalar także należy do kodu (skalar jest równy 0 lub 1).

Binarne kody liniowe

- Kod binarny K nazywamy liniowym jeśli
  1. suma dwóch słów kodowych z K także należy do K
  2. pomnożenie słowa kodowego przez skalar także należy do kodu (skalar jest równy 0 lub 1).
- Każdy kod liniowy zawiera słowo $0 \ldots 0$.

Binarne kody liniowe

- Kod binarny K nazywamy liniowym jeśli
  1. suma dwóch słów kodowych z K także należy do K
  2. pomnożenie słowa kodowego przez skalar także należy do kodu (skalar jest równy 0 lub 1).
- Każdy kod liniowy zawiera słowo $0 \ldots 0$.

Waga minimalna

Wagą słowa kodowego nazywamy odległość Hamminga tego słowa od słowa $0 \ldots 0$. Minimalną wagą kodu K nazywamy najmniejszą wagę słowa kodowego należącego do K i różnego od $0 \ldots 0$.

Kody liniowe – własności

Własność

Dla binarnego kodu liniowego minimalna waga jest równa minimalnej odległości.

Kody liniowe – własności

Własność

Dla binarnego kodu liniowego minimalna waga jest równa minimalnej odległości.

Dowód

Niech $d(K)$ - minimalna odległość i $w(K)$ - minimalna waga. Niech a ma wagę $w(K)$. Wtedy $w(K) = d(a, 0) \geqslant d(K)$.
W drugą stronę, niech a i b - słowa kodowe takie, że $d(a, b) = d(K)$. $a + b$ jest także słowem kodowym. Mamy $w(a + b) = d(a, b)$, a stąd $d(K) = d(a, b) \geqslant w(K)$.

Przykład

- Kod parzystości Każdy kod musi zawierać parzystą liczbę jedynek.

Przykład

- Kod parzystości Każdy kod musi zawierać parzystą liczbę jedynek.
  1. Suma dwóch kodów z parzystą ilością jedynek zawiera parzystą liczbę jedynek. (Dlaczego?)

Przykład

- Kod parzystości Każdy kod musi zawierać parzystą liczbę jedynek.
  1. Suma dwóch kodów z parzystą ilością jedynek zawiera parzystą liczbę jedynek. (Dlaczego?)
  2. Słowo kodowe o minimalnej wadze musi mieć co najmniej 2 jedynki stąd waga kodu wynosi 2.

Przykład

- Kod parzystości Każdy kod musi zawierać parzystą liczbę jedynek.
  1. Suma dwóch kodów z parzystą ilością jedynek zawiera parzystą liczbę jedynek. (Dlaczego?)
  2. Słowo kodowe o minimalnej wadze musi mieć co najmniej 2 jedynki stąd waga kodu wynosi 2.
- Kod powtórzeniowy Jest oczywiste, że jest kodem liniowym i że ma minimalną wagę równą swojej długości.

Kody Hamminga

- Kody doskonałe dla korekcji jednego błędu.

Kody Hamminga

- Kody doskonałe dla korekcji jednego błędu.
- Długość kodu: $n = 2^m - 1$

Kody Hamminga

- Kody doskonałe dla korekcji jednego błędu.
- Długość kodu: $n = 2^m - 1$
- Ilość bitów informacji: $k = 2^m - m - 1$

Kody Hamminga

- Kody doskonałe dla korekcji jednego błędu.
- Długość kodu: $n = 2^m - 1$
- Ilość bitów informacji: $k = 2^m - m - 1$
- Minimalna odległość: $d = 3$

Kody Hamminga

- Kody doskonałe dla korekcji jednego błędu.
- Długość kodu: $n = 2^m - 1$
- Ilość bitów informacji: $k = 2^m - m - 1$
- Minimalna odległość: $d = 3$

Dlaczego kod jest doskonały?

Każde słowo kodowe jest otoczone przez dokładnie n słów z jednym błędem. Stąd dla każdego słowa kodowego mamy 2^m słów które go reprezentują. Ilość słów kodowych to $2^{2^m-m-1} = \frac{2^n}{2^m}$. Czyli nie ma ciągu który nie byłby związany z jakimś słowem kodowym.

Kody Hamminga

Własność

Binarny kod liniowy K koryguje jeden błąd wtedy i tylko wtedy gdy każda macierz parzystości K ma niezerowe i parami różne kolumny.

Kody Hamminga

Własność

Binarny kod liniowy K koryguje jeden błąd wtedy i tylko wtedy gdy każda macierz parzystości K ma niezerowe i parami różne kolumny.

- Dla $n = 2^m - 1$ najprostszą taką macierzą jest zawierająca jako kolumny zapis binarny liczb od 1 do $2^m - 1$. Dla $m = 3$

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Kody Hamminga

- Zapisując to jako układ równań mamy

$$\begin{array}{ccccccccccccccc}
 & & & & & & x_4 & + & x_5 & + & x_6 & + & x_7 & = & 0 \\
 & & x_2 & + & x_3 & + & & & & & x_6 & + & x_7 & = & 0 \\
x_1 & + & & & x_3 & + & & & x_5 & + & & & x_7 & = & 0
\end{array}$$

Kody Hamminga

- Zapisując to jako układ równań mamy

$$\begin{array}{ccccccccccccccc} & & & & & & x_4 & + & x_5 & + & x_6 & + & x_7 & = & 0 \\ & & x_2 & + & x_3 & + & & & & & x_6 & + & x_7 & = & 0 \\ x_1 & + & & & x_3 & + & & & x_5 & + & & & x_7 & = & 0 \end{array}$$

- A to możemy zapisać równorzędnie jako

$$\begin{aligned} x_5 &= x_2 + x_3 + x_4 \\ x_6 &= x_1 + x_3 + x_4 \\ x_7 &= x_1 + x_2 + x_4 \end{aligned}$$

Kody Hamminga

- Macierz generująca dla tego kodu

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Kody Hamminga

- Macierz generująca dla tego kodu

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

- Łatwo sprawdzić, że dla tego kodu syndrom czytany jako numer kolumny oznacza który bit w kodzie poprawić.

Kody Hamminga – przykład korekty

- Chcemy wysłać informację 1111.

Kody Hamminga – przykład korekty

- Chcemy wysłać informację 1111.
- Obliczamy kod

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Kody Hamminga – przykład korekty

- Chcemy wysłać informację 1111.
- Obliczamy kod

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

- Wysyłamy: 1111111

Kody Hamminga – przykład korekty

- Na skutek błędu otrzymujemy: 1110111

Kody Hamminga – przykład korekty

- Na skutek błędu otrzymujemy: 1110111
- Obliczamy syndrom

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Kody Hamminga – przykład korekty

- Na skutek błędu otrzymujemy: 1110111
- Obliczamy syndrom

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

- Czyli musimy poprawić 4 bit. Stąd informacja to 1111

Kod Hamminga długości 4

- Wszystkie słowa kodowe

Informacja	Kod	Informacja	Kod
0000	0000000	1000	1000011
0001	0001111	1001	1001100
0010	0010110	1010	1010101
0011	0011001	1011	1011010
0100	0100101	1100	1100110
0101	0101010	1101	1101001
0110	0110011	1110	1110000
0111	0111100	1111	1111111

Kod Hamminga długości 4

- Wszystkie słowa kodowe

Informacja	Kod	Informacja	Kod
0000	0000000	1000	1000011
0001	0001111	1001	1001100
0010	0010110	1010	1010101
0011	0011001	1011	1011010
0100	0100101	1100	1100110
0101	0101010	1101	1101001
0110	0110011	1110	1110000
0111	0111100	1111	1111111

- Łatwo zauważyć że minimalna waga tego kodu to 3.

Kody Hamminga – długości

- Ilość bitów informacji i długość kodu

m	n długość kodu	k bity informacji
2	3	1
3	7	4
4	15	11
5	31	26
6	63	57
7	127	120

Kody Hamminga – długości

- Ilość bitów informacji i długość kodu

m	n długość kodu	k bity informacji
2	3	1
3	7	4
4	15	11
5	31	26
6	63	57
7	127	120

- A co zrobić jak mamy inną ilość bitów informacji?

Nietypowa ilość bitów informacji

- Mamy 3 zamiast 4 bitów informacji.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow$$

Nietypowa ilość bitów informacji

- Mamy 3 zamiast 4 bitów informacji.
- W macierzy G nie jest więc potrzebna 4 kolumna.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow$$

Nietypowa ilość bitów informacji

- Mamy 3 zamiast 4 bitów informacji.
- W macierzy G nie jest więc potrzebna 4 kolumna.
- Ale 4 wiersz zawiera same zera więc też może być usunięty.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Nietypowa ilość bitów informacji

- Ostatecznie mamy

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Nietypowa ilość bitów informacji

- Ostatecznie mamy

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

- Macierz parzystości także modyfikujemy wykreślając 4 kolumnę (tak jak czwarty wiersz w G).

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Nietypowa ilość bitów informacji

- Ostatecznie mamy

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

- Macierz parzystości także modyfikujemy wykreślając 4 kolumnę (tak jak czwarty wiersz w G).

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

- Niestety syndromy już nie numerują dokładnie miejsca błędu. 110 oznacza poprawienie 5 bitu.

Rozszerzone kody Hamminga

- Do kodów Hamminga dodajemy jeszcze jeden bit – parzystość.

Rozszerzone kody Hamminga

- Do kodów Hamminga dodajemy jeszcze jeden bit – parzystość.
- Rozszerzone kody Hamminga korygują dalej 1 błąd ale wykrywają 2 błędy.

Rozszerzone kody Hamminga

- Do kodów Hamminga dodajemy jeszcze jeden bit – parzystość.
- Rozszerzone kody Hamminga korygują dalej 1 błąd ale wykrywają 2 błędy.
- Modyfikacje macierzy generującej i parzystości są proste.

Rozszerzone kody Hamminga

- Do kodów Hamminga dodajemy jeszcze jeden bit – parzystość.
- Rozszerzone kody Hamminga korygują dalej 1 błąd ale wykrywają 2 błędy.
- Modyfikacje macierzy generującej i parzystości są proste.
- Bardziej skomplikowany opis syndromów.

Cykliczne binarne kody liniowe

Definicja

Kod liniowy nazywamy cyklicznym wtedy i tylko wtedy gdy dla każdego słowa kodowego $v_0 v_1 \ldots v_{n-1}$ cykliczne przesunięcie $v_{n-1} v_0 v_1 \ldots v_{n-2}$ też jest słowem kodowym.

Cykliczne binarne kody liniowe

Definicja

Kod liniowy nazywamy cyklicznym wtedy i tylko wtedy gdy dla każdego słowa kodowego $v_0 v_1 \ldots v_{n-1}$ cykliczne przesunięcie $v_{n-1} v_0 v_1 \ldots v_{n-2}$ też jest słowem kodowym.

- Kod parzystości jest kodem cyklicznym.

Cykliczne binarne kody liniowe

Definicja

Kod liniowy nazywamy cyklicznym wtedy i tylko wtedy gdy dla każdego słowa kodowego $v_0 v_1 \ldots v_{n-1}$ cykliczne przesunięcie $v_{n-1} v_0 v_1 \ldots v_{n-2}$ też jest słowem kodowym.

- Kod parzystości jest kodem cyklicznym.
- Kod powtórzeniowy jest kodem cyklicznym.

Reprezentacja kodów przez wielomiany

Definicja

Kod $a_0 a_1 \ldots a_{n-1}$ będziemy reprezentować jako wielomian nad ciałem Z_2 w postaci
$a(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$.

Reprezentacja kodów przez wielomiany

Definicja

Kod $a_0 a_1 \ldots a_{n-1}$ będziemy reprezentować jako wielomian nad ciałem Z_2 w postaci
$a(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$.

- Dodawanie słów kodowych jest równoważne dodawaniu wielomianów je reprezentujących.

Reprezentacja kodów przez wielomiany

Definicja

Kod $a_0 a_1 \ldots a_{n-1}$ będziemy reprezentować jako wielomian nad ciałem Z_2 w postaci
$a(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$.

- Dodawanie słów kodowych jest równoważne dodawaniu wielomianów je reprezentujących.
- Mnożenie przez skalar również.

Reprezentacja kodów przez wielomiany

Definicja

Kod $a_0 a_1 \ldots a_{n-1}$ będziemy reprezentować jako wielomian nad ciałem Z_2 w postaci
$a(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$.

- Dodawanie słów kodowych jest równoważne dodawaniu wielomianów je reprezentujących.
- Mnożenie przez skalar również.
- Dodatkowo wielomiany można mnożyć przez siebie. (Po pomnożeniu wielomianu $a(x)$ przez $b(x)$ stopień wielomianu $a(x)b(x)$ jest równy sumie stopni wielomianów $a(x)$ i $b(x)$.)

Kody cykliczne - własności

Każdy cykliczny kod K długości n ma następującą własność:

$$g(x) \in K \Rightarrow q(x)g(x) \in K$$

gdzie wszystkie wielomiany $q(x)$ są stopnia takiego że stopień wielomianu $q(x)g(x)$ jest mniejszy niż n.

Kody cykliczne - własności

Każdy cykliczny kod K długości n ma następującą własność:

$$g(x) \in K \Rightarrow q(x)g(x) \in K$$

gdzie wszystkie wielomiany $q(x)$ są stopnia takiego że stopień wielomianu $q(x)g(x)$ jest mniejszy niż n.

Dowód

Niech $g(x) = g_0 + g_1 x + \ldots + g_s x^s$. Dla każdego $i < n - s$ wielomian $x^i g(x)$ odpowiada cyklicznemu przesunięciu słowa długości n o i pozycji. Pomnożenie $g(x)$ przez $q(x)$ odpowiada więc dodaniu do siebie przesuniętych cyklicznie kodów więc też jest słowem kodowym.

Kody cykliczne - własności

Każdy nietrywialny (n, k)-kod cykliczny (kod długości n kodujący k bitów informacji) zawiera słowo kodowe $g(x)$ stopnia $n - k$. Kod ma wtedy następującą macierz generującą ($kod = informacja \cdot G$):

$$G = \begin{bmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{bmatrix}$$

$$= \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{bmatrix}$$

$g(x)$ nazywamy wielomianem generującym kodu.

Przykład

- Kod parzystości długości n jest cykliczny (jest (n,k)=(n,n-1)-kodem). Jego wielomianem generującym jest $1 + x$. Stąd mamy macierz generującą postaci

$$\begin{bmatrix} 1 & 1 & 0 & & \cdots & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 1 & 0 \\ 0 & \cdots & & 0 & 1 & 1 \end{bmatrix}$$

Przykład

- Kod parzystości długości n jest cykliczny (jest (n,k)=(n,n-1)-kodem). Jego wielomianem generującym jest $1 + x$. Stąd mamy macierz generującą postaci

$$\begin{bmatrix} 1 & 1 & 0 & & \cdots & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 1 & 0 \\ 0 & \cdots & & 0 & 1 & 1 \end{bmatrix}$$

- Kod powtórzeniowy długości n ma wielomian generujący $g(x) = 1 + x + x^2 + \ldots + x^{n-1}$.

Kody cykliczne - własności

- Czy każdy wielomian może być generatorem?

Kody cykliczne - własności

- Czy każdy wielomian może być generatorem?
- Zobaczmy co to jest $x^k g(x)$?

Kody cykliczne - własności

- Czy każdy wielomian może być generatorem?
- Zobaczmy co to jest $x^k g(x)$?
- $x^k g(x)$ jest stopnia n więc nie jest słowem kodowym. Ale jeśli porównamy to z operacją przesunięcia cyklicznego to powinniśmy usunąć x^n a dodać 1. Stąd $x^k g(x) - (x^n + 1)$ jest słowem kodowym.

Kody cykliczne - własności

- Czy każdy wielomian może być generatorem?
- Zobaczmy co to jest $x^k g(x)$?
- $x^k g(x)$ jest stopnia n więc nie jest słowem kodowym. Ale jeśli porównamy to z operacją przesunięcia cyklicznego to powinniśmy usunąć x^n a dodać 1. Stąd $x^k g(x) - (x^n + 1)$ jest słowem kodowym.

Każdy wielomian generujący kodu cyklicznego długości n dzieli wielomian $x^n + 1$. Stąd generatorami mogą być tylko właściwe dzielniki $x^n - 1$.

Kody cykliczne - własności

Dla każdego cyklicznego kodu długości n generowanego przez wielomian $g(x)$, wielomian postaci

$$h(x) = \frac{x^n - 1}{g(x)}$$

nazywamy wielomianem parzystości tego kodu.

Kody cykliczne - własności

Dla każdego cyklicznego kodu długości n generowanego przez wielomian $g(x)$, wielomian postaci

$$h(x) = \frac{x^n - 1}{g(x)}$$

nazywamy wielomianem parzystości tego kodu.

Dla cyklicznego kodu długości n z wielomianem parzystości $h(x) = h_0 + h_1 x + \ldots + h_{k-1} x^{k-1} + x^k$ mamy następującą macierz parzystości $n \times k$

$$H = \begin{bmatrix} 0 & \cdots & & 0 & h_k & \cdots & h_0 \\ 0 & \cdots & 0 & h_k & \cdots & h_0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ h_k & \cdots & h_0 & 0 & & \cdots & 0 \end{bmatrix}$$

Przykład

- Weźmy $n = 7$ i $g(x) = 1 + x + x^3$.

Przykład

- Weźmy $n = 7$ i $g(x) = 1 + x + x^3$.
- Policzmy $h(x) = (x^7 + 1)/g(x)$.

Przykład

- Weźmy $n = 7$ i $g(x) = 1 + x + x^3$.
- Policzmy $h(x) = (x^7 + 1)/g(x)$.
- $h(x) = x^4 + x^2 + x + 1$

Przykład

- Weźmy $n = 7$ i $g(x) = 1 + x + x^3$.
- Policzmy $h(x) = (x^7 + 1)/g(x)$.
- $h(x) = x^4 + x^2 + x + 1$
- Macierz generująca

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Przykład

- Weźmy $n = 7$ i $g(x) = 1 + x + x^3$.
- Policzmy $h(x) = (x^7 + 1)/g(x)$.
- $h(x) = x^4 + x^2 + x + 1$
- Macierz generująca

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

- Macierz parzystości

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Przykład

- Weźmy $n = 7$ i $g(x) = 1 + x + x^3$.
- Policzmy $h(x) = (x^7 + 1)/g(x)$.
- $h(x) = x^4 + x^2 + x + 1$
- Macierz generująca

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

- Macierz parzystości

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

- Łatwo zauważyć, że jest to kod Hamminga.

Przykład – kody cykliczne długości 7

$x^7 + 1 = (x+1)(x^3+x+1)(x^3+x^2+1)$

$g(x)$	$h(x)$	
$x+1$	$x^6+x^5+x^4$ $+x^3+x^2+x+1$	Kod parzystości
x^3+x+1	x^4+x^2+x+1	Kod Hamminga
x^3+x^2+1	$x^4+x^3+x^2+1$	Kod Hamminga
$x^4+x^3+x^2+1$	x^3+x^2+1	Kod dualny
x^4+x^2+x+1	x^3+x+1	Kod dualny
$x^6+x^5+x^4$ $+x^3+x^2+x+1$	$x+1$	Kod powtórzeniowy

Kody doskonałe

Binarny (n,k)-kod jest doskonały dla t błędów jeśli ma minimalną odległość równą $2t + 1$ oraz spełniona jest równość

$$2^{n-k} = \sum_{i=0}^{t} \binom{n}{i}.$$

(Lewa strona równości odpowiada ilości ciągów n-bitowych na jedno słowo kodowe, a prawa ilości ciągów odległych od danego odpowiednio o $0, 1, \ldots, t$ w mierze Hamminga.)

Kod Golay-a

- Kod Golay-a ma długość 23 bity.

Kod Golay-a

- Kod Golay-a ma długość 23 bity.
- Wielomianem generującym jest $g(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11}$.

Kod Golay-a

- Kod Golay-a ma długość 23 bity.
- Wielomianem generującym jest $g(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11}$.
- Kod Golay-a ma minimalną odległość równą 7.

Kod Golay-a

- Kod Golay-a ma długość 23 bity.
- Wielomianem generującym jest $g(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11}$.
- Kod Golay-a ma minimalną odległość równą 7.
- Kod Golay-a koryguje 3 błędy.

Kod Golay-a

- Kod Golay-a ma długość 23 bity.
- Wielomianem generującym jest $g(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11}$.
- Kod Golay-a ma minimalną odległość równą 7.
- Kod Golay-a koryguje 3 błędy.
- Kod Golay-a jest kodem doskonałym

$$
\begin{aligned}
2^{23-12} &= 2048 = \binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} = \\
&= 1 + 23 + \frac{23 \cdot 22}{2} + \frac{23 \cdot 22 \cdot 21}{2 \cdot 3} = \\
&= 1 + 23 + 253 + 1771
\end{aligned}
$$

Burst Errors

- Kod liniowy wykrywa *t burst errors* jeśli wykrywa *t* błędów położonych obok siebie w słowie kodowym (błędy na kolejnych *t* bitach, bez przerw).

Burst Errors

- Kod liniowy wykrywa *t burst errors* jeśli wykrywa *t* błędów położonych obok siebie w słowie kodowym (błędy na kolejnych *t* bitach, bez przerw).
- Kod liniowy koryguje *t burst errors* jeśli koryguje *t* błędów na położonych obok siebie bitach.

Przykład

- Kod dualny do kodu Hamminga z wielomianem generującym $g(x) = 1 + x^2 + x^3 + x^4$ jest (7,3)-kodem korygującym 2 *burst errors*.

Przykład

- Kod dualny do kodu Hamminga z wielomianem generującym $g(x) = 1 + x^2 + x^3 + x^4$ jest (7,3)-kodem korygującym 2 *burst errors*.
- Minimalną odległością tego kodu jest 4.

Przykład

- Kod dualny do kodu Hamminga z wielomianem generującym $g(x) = 1 + x^2 + x^3 + x^4$ jest (7,3)-kodem korygującym 2 *burst errors*.
- Minimalną odległością tego kodu jest 4.
- Aby pokazać, że rzeczywiście jest to kod korygujący 2 *burst errors* wystarczy pokazać, że złożenie dwóch takich błędów nie jest słowem kodowym. Czyli słowa
  1100000+0011000=1111000,
  1100000+0001100=1101100,
  1100000+0000110=1100110,
  1100000+0000011=1100011,
  nie są poprawnymi kodami (łatwo sprawdzić).
  Pozostałe przypadki to przesunięcia cykliczne tych 4 słów.

Przeplatanie

Własność

Dla każdego (n, k)-kodu korygującego *l burst errors* możemy stworzyć (nj, kj)-kod korygujący *lj burst errors* przez przeplot *j* słów kodowych, tj. dla słów kodowych $a_0^1 a_1^1 \dots a_{n-1}^1$, ..., $a_0^j a_1^j \dots a_{n-1}^j$ tworzymy słowo kodowe $a_0^1 a_0^2 \dots a_0^j a_1^1 a_1^2 \dots a_1^j \dots a_{n-1}^1 a_{n-1}^2 \dots a_{n-1}^j$

Przeplatanie

Własność

Dla każdego (n, k)-kodu korygującego l *burst errors* możemy stworzyć (nj, kj)-kod korygujący lj *burst errors* przez przeplot j słów kodowych, tj. dla słów kodowych $a_0^1 a_1^1 \dots a_{n-1}^1$, ..., $a_0^j a_1^j \dots a_{n-1}^j$ tworzymy słowo kodowe $a_0^1 a_0^2 \dots a_0^j a_1^1 a_1^2 \dots a_1^j \dots a_{n-1}^1 a_{n-1}^2 \dots a_{n-1}^j$

- Kod Hamminga (7,4) jest kodem korygującym 1 błąd więc również korygującym 1 *burst error*.

Przeplatanie

Własność

Dla każdego (n, k)-kodu korygującego l *burst errors* możemy stworzyć (nj, kj)-kod korygujący lj *burst errors* przez przeplot j słów kodowych, tj. dla słów kodowych $a_0^1 a_1^1 \dots a_{n-1}^1$, ..., $a_0^j a_1^j \dots a_{n-1}^j$ tworzymy słowo kodowe $a_0^1 a_0^2 \dots a_0^j a_1^1 a_1^2 \dots a_1^j \dots a_{n-1}^1 a_{n-1}^2 \dots a_{n-1}^j$

- Kod Hamminga (7,4) jest kodem korygującym 1 błąd więc również korygującym 1 *burst error*.
- Jeśli przepleciemy 3 słowa kodowe to otrzymamy (21,12)-kod korygujący 3 *burst errors*.

Przeplatanie

Własność

Dla każdego (n, k)-kodu korygującego l *burst errors* możemy stworzyć (nj, kj)-kod korygujący lj *burst errors* przez przeplot j słów kodowych, tj. dla słów kodowych $a_0^1 a_1^1 \dots a_{n-1}^1$, ..., $a_0^j a_1^j \dots a_{n-1}^j$ tworzymy słowo kodowe $a_0^1 a_0^2 \dots a_0^j a_1^1 a_1^2 \dots a_1^j \dots a_{n-1}^1 a_{n-1}^2 \dots a_{n-1}^j$

- Kod Hamminga (7,4) jest kodem korygującym 1 błąd więc również korygującym 1 *burst error*.
- Jeśli przepleciemy 3 słowa kodowe to otrzymamy (21,12)-kod korygujący 3 *burst errors*.
- Oszczędzamy dwa bity w porównaniu z kodem Golay-a ((23,12)-kod korygujący 3 błędy).