Cơ sở Lý thuyết Truyền tin-2004

Chương 4: Mã hiệu

Hà Quốc Trung[1]

[1]Khoa Công nghệ thông tin
Đại học Bách khoa Hà nội

Chương 6: Mã hóa kênh

1. Khái niệm cơ bản
2. Mã tuyến tính
3. Mã vòng (CRC)
4. Mã chập

1. Khái niệm cơ bản

1. Khái niệm cơ bản
   - Giới thiệu
   - Khoảng cách Hamming
2. Mã tuyến tính
3. Mã vòng (CRC)
4. Mã chập

1.1.Giới thiệu

- Định lý Shannon 2 về mã hóa kênh có nhiễu: Nếu thông lượng kênh lớn hơn tốc độ lập tin của nguồn thì có thể truyền tin với sai số nhỏ tùy ý.
- Định lý chỉ ra với một độ dư dương, sai số truyền tin có thể nhỏ tùy ý.
- Định lý chỉ ra cách thức mã hóa để có sai số đó
- Các phương pháp mã hóa này đòi hỏi bảng đối chiếu (từ điển mã) khổng lồ, kích thước tăng theo hàm mũ của chiều dài từ mã
- Các phương pháp mã hóa thực tế còn cách xa giới hạn của Shannon (Xem phần mã hiệu)

Nguyên tắc sửa sai và phát hiện sai

- Sửa lỗi và phát hiện lỗi phụ thuộc vào tính chất thống kê của kênh và lỗi
- Phân biệt hai loại lỗi
  - Lỗi độc lập thống kê: các lỗi xuất hiện riêng lẻ, không liên quan lẫn nhau
  - Lỗi chùm: lỗi liên quan chặt chẽ với nhau, thường xuất hiện cùng một lúc (đĩa cứng hỏng)
- Cấu trúc của mã kênh phụ thuộc vào phân bố xác suất của lỗi

Nguyên tắc phát hiện lỗi

- Số từ mã nhỏ hơn số các tổ hợp mã có thể
- Sử dụng các tổ hợp cấm để phát hiện việc truyền tin sai
  - Cần lựa chọn các từ mã và các tổ hợp bị cấm để
  - Hiệu quả: số lượng tổ hợp mã có thể không quá nhiều
  - Chính xác: đảm bảo sai số luôn sinh ra một tổ hợp cấm
- Đảm bảo một từ mã không bị truyền sai thành một từ mã khác
- Khả năng phát hiện lỗi: tỷ lệ có tổ hợp cấm khi có lỗi
  - Phát hiện lỗi: chuyển đổi từ mã thành tổ hợp cấm L(M-L)
  - Tổng số lỗi: chuyển đổi một từ mã thành một từ mã bất kỳ LM
  - Vậy khả năng phát hiện lỗi là: $\frac{L(M-L)}{LM} = 1 - \frac{L}{M}$
  - Để khả năng phát hiện sai lớn, $M \gg L$, hay nói cách khác từ mã phải có độ dài lớn hơn nhiều so với chiều dài tối ưu

Nguyên tắc sửa lỗi

- Mục đích sửa sai: đảm bảo sai nhầm tối thiểu
- Nguyên tắc: các ký hiệu được ánh xạ vào một từ mã. Từ mã này do sai số biến đổi sẽ tạo ra các tổ hợp mã bị cấm.
- Khi nhận được một tổ hợp mã, xác định tổ hợp mã này thuộc về tập hợp các tổ hợp mã có thể của một ký hiệu đầu nào để xác định ký hiệu đầu vào
- Cần có điều kiện là lỗi không chuyển một từ mã này sang tổ hợp mã (lỗi) của một từ mã khác

1.2.Khoảng cách Hamming

- Số lượng các bít khác nhau giữa hai tổ hợp mã có cùng độ dài
- Khoảng cách giữa một từ mã và từ mã 0 gọi là trọng số của một từ mã.
- Phản ánh sự "gần" nhau của hai tổ hợp mã khi có nhiễu
- Nhiễu biến một từ mã thành một tổ hợp mã cách từ mã một khoảng nào đó
- Nếu khoảng cách đủ nhỏ (số lỗi ít, số lượng các bít bị thay đổi ít) để tổ hợp mã không trùng với từ mã khác, mã hiệu có khả năng phát hiện lỗi. Khoảng cách giữa các từ mã lớn hơn số lỗi có thể
- Nếu khoảng cách đủ nhỏ, để có thể phân biệt tổ hợp mã thu được gần từ mã nào nhất, có thể sửa lỗi. Cần đảm bảo khoảng cách giữa hai từ mã lớn hơn (thực sự) 2 lần số lỗi có thể

1.2.Khoảng cách Hamming *(Tiếp)*

- Ví dụ: Mã 00,01,10,11 không có khả năng phát hiện lỗi
- Mã 0000,0011,1100,1111 có khoảng cách hamming giữa các từ mã là 2, có thể phát hiện được 1 lỗi
- Mã 000000,000111,111000,111111 có khoảng cách hamming giữa các từ mã là 3, vậy có thể phát hiện 2 lỗi và sửa một lỗi

1.3.Ví dụ về mã chống nhiễu

- Mã lặp
- Mã chẵn lẻ

1.4.Phân loại mã chống nhiễu

- Mã khối
- Mã luồng

1.5.Một số kiến thức toán học cơ bản

- Trường
- Không gian tuyến tính
- Không gian đa thức

2. Mã tuyến tính

1 Khái niệm cơ bản

2 Mã tuyến tính
- Định nghĩa, Phương pháp biểu diễn
- Nguyên lý giải mã tuyến tính
- Các giới hạn lý thuyết của mã tuyến tính
- Mã Hamming tuyến tính

3 Mã vòng (CRC)

4 Mã chập

2.1.Định nghĩa, Phương pháp biểu diễn

- Khái niệm: Mã hiệu gọi là tuyến tính nếu tập hợp các từ mã đóng đối với phép cộng các từ mã
  - Xét các mã hiệu nhị phân đồng đều
  - Thực hiện phép tính cộng nhị phân trên mỗi cặp hai từ mã 00100+010111=011011. Phép toán có tính kết hợp và giao hoán
  - Khi đó mã hiệu là tuyến tính nếu tổng hai từ mã luôn luôn là một từ mã

Biểu diễn bằng ma trận sinh

- Xét mã hiệu tuyến tính là một tập N từ mã, có độ dài n.
- Luôn có một tập con của mã hiệu để
  - Tất cả các từ mã đều là tổ hợp tuyến tính của các từ mã thuộc tập con này
  - Càc từ mã trong tập con độc lập tuyến tính (không là tổ hợp tuyến tính của nhau)
  - Tập từ mã có tính chất như vậy, có độ dài tối thiểu k gọi là cơ sở của mã hiệu
- Có tối đa 2^k tổ hợp tuyến tính của k từ mã. Do mã hiệu đóng với phép cộng, $N = 2^k$
- Mã hiệu được đặc trưng bởi ma trận các từ mã cơ sở: ma trận sinh, có k dòng và n cột. Mã hiệu được ký hiệu (k,n)
- Các từ mã của mã hiệu là các tổ hợp tuyến tính của các dòng trong ma trận sinh

Biểu diễn bằng ma trận sinh *(Tiếp)*

- Ví dụ mã tuyến tính (5,2) 00000, 01101, 10110,11011 sinh ra từ ma trận

$$G = \begin{bmatrix} 01101 \\ 10110 \end{bmatrix} \text{ hoặc } \begin{bmatrix} 10110 \\ 11011 \end{bmatrix} \text{ hoặc } \begin{bmatrix} 01101 \\ 11011 \end{bmatrix}$$

- Ví dụ mã (5,3): 00000,10011,01010,11001,00101,10110,01111,11100 có thể được biểu diễn bởi một trong các ma trận sinh

$$G = \begin{bmatrix} 10011 \\ 01010 \\ 00101 \end{bmatrix} \text{ hoặc } \begin{bmatrix} 10011 \\ 11001 \\ 11100 \end{bmatrix}$$

Ma trận kiểm tra/thử

- Có 2^n từ mã có chiều dài n. Các từ mã này tạo thành một mã hiệu tuyến tính, biểu diễn bằng ma trận sinh (n,n)
- Mã hiệu tuyến tính N từ mã chỉ sử dụng k cơ sở
- Vậy $n - k$ cơ sở còn lại có thể biểu diễn các từ mã trực giao với cá từ mã của mã hiệu (không gian không của mã hiệu), có thể dùng để kiểm tra một từ mã có (không) thuộc mã hiệu ban đầu
- Ma trận H của $n - k$ cơ sở gọi là ma trận thử của mã hiệu.
- Ma trận thử của một mã hiệu (n,k) là ma trận sinh của một mã hiệu khác (n,n-k)

Ma trận kiểm tra/thử *(Tiếp)*

- Mỗi từ mã M trực giao với tất cả các dòng B của ma trận thử

$$\sum_{1}^{n} m_i b_i = 0$$

Từ đó

$$MH^T = 0$$

là điều kiện cần và đủ để một chuỗi n ký hiệu là từ mã

- Ví dụ (5,3):
00000,10011,01010,11001,00101,10110,01111,11100,
Không gian không gồm các từ mã a_1, a_2, a_3, a_4, a_5 thỏa mãn

$$a_1+a_4+a_5 = 0; a_2+a_4 = 0; a_1+a_2+a_5 = 0; a_3+a_5 = 0, a_1+a_3+a_4$$

hay

$$a_1 + x + y = 0; a_2 = a_4 = x; a_3 = a_5 = y$$

Ma trận kiểm tra/thử *(Tiếp)*

Vậy không gian không gồm 4 từ mã

$$00000, 11010, 10101, 01111$$

Ma trận thử sẽ là

$$\begin{bmatrix} 11010 \\ 10101 \end{bmatrix} \begin{bmatrix} 01111 \\ 10101 \end{bmatrix} \begin{bmatrix} 11010 \\ 01111 \end{bmatrix}$$

Có thể kiểm tra được điều kiện trực giao

Dạng chuẩn tắc của mã tuyến tính

- Trong các phép tính cộng giữa các từ mã, vị trí các bít không có vai trò quan trọng. Có thể hoán vị các bít cho nhau
- Khi hoán vị các bit và thực hiện các phép biến đổi hợp lệ với một ma trận sinh, có thể chuyển ma trận sinh về dạng

$$G'' = \left| \begin{array}{l} 10\ldots0p_{11}p_{12}\ldots p_{1n-k} \\ 01\ldots0p_{11}p_{12}\ldots p_{2n-k} \\ \ldots \\ 00\ldots1p_{11}p_{12}\ldots p_{kn-k} \end{array} \right|$$

- Một từ mã bất kỳ sẽ là tổ hợp tuyến tính của các hàng trong ma trận sinh, với các hệ số nhị phân tùy ý $v = (a_1, a_2 \ldots, a_k)$ sẽ có dạng

$$M = vG'' = (a_1, a_2 \ldots, a_k, C_1, C_2, \ldots C_{n-k})$$

trong đó $C_j = \sum_1^k a_i p_{ij}$

Dạng chuẩn tắc của mã tuyến tính *(Tiếp)*

- $v = (a_1, a_2 \ldots, a_k)$ được chọn một cách tùy ý, chính là phần thông tin của một từ mã
- Một từ mã trong mã hóa kênh sẽ gồm phần thông tin, và một phần thông tin điều khiển, được tính bằng một tổ hợp tuyến tính của phần thông tin thực sự
- Ma trận sinh sẽ có dạng (I_k, P_{n-k}), I_k là ma trận đơn vị
- Nguyên tắc phát hiện lỗi: sau khi nhận được từ mã, tính lại phần thông tin điều khiển rồi so sánh với kết quả nhận được
- Nguyên tắc sửa lỗi: xác định các khả năng có thể của từ mã đã truyền đi rồi chọn từ mã thích hợp
- Vấn đề lý thuyết: khối lượng thông tin điều khiển đủ đề phát hiện một số lỗi xác định.
- Lỗi có thể là lỗi đơn hoặc lỗi chùm

2.2.Nguyên lý giải mã tuyến tính

- Bài toán phát hiện và sửa lỗi
  - Nhận được một chuỗi ký hiệu có độ dài n
  - Kiểm tra liệu chuỗi ký hiệu này có là một từ mã hay không
  - Nếu có, xác định vị trí lỗi
  - Công cụ: ma trận sinh, ma trận thử
- với một từ mã M, có $M.H^T = 0$
- Vậy nếu $M.H^T \neq 0$, đã có lỗi xảy ra: bài toán phát hiện lỗi
- $M.H^T$ gọi là syndrom của chuỗi bít, được sử dụng để phát hiện lỗi

Bảng lớp kề và sửa lỗi

- Khi có lỗi xảy ra, một vài bít nào đó bị đổi vị trí. Một chuỗi bít được nhận, sai khác với từ mã ban đầu 1 vài bít, biểu diễn bằng vector sai số bằng hiệu của chuỗi bít thu được và từ mã ban đầu
- Lập bảng lớp kề của các từ mã. Các cột tương ứng với các từ mã. Các hàng tương ứng với các vector lỗi có thể. Mỗi hàng tạo ra một lớp kề của các từ mã tương ứng với các vector lỗi
- Việc xác định các vị trí lỗi chuyển thành việc xác định lớp kề của chuỗi bít nhận được. Dễ thấy nhất là so sánh trong bảng xem chuỗi bít lỗi nằm ở lớp kề nào.
- Giá trị của syndrom thay đổi chỉ phụ thuộc vào vị trí của bít lỗi. Vậy các chuỗi bít trong một hàng có syndrom giống nhau.

Bảng lớp kề và sửa lỗi *(Tiếp)*

- Ngược lại, Khi lập mã hiệu, cần đảm bảo với các vector lỗi có thể, hai vector khác nhau cho hai lớp kề khác nhau, 2 syndrom khác nhau
- Vậy có thể xác định vector lỗi bằng cách tính syndrom của chuỗi bít và xác định lớp kề có syndrom đó.

Ví dụ

- Cho mã hiệu 00000,00111,11001,11110

00000 00111 11001 11110	Ma trận sinh 11001 11110	Ma trận thử 00110 01011 10011

- Bảng lớp kề cho một lỗi và một vài lỗi đúp

	00000	00111	11001	11110	syndrom
00001	00000	00111	11001	11110	011
00010	00010	00101	11011	11100	111
00100	00100	00011	11101	11010	100
01000	01000	01111	10001	10110	010
10000	10000	10111	01001	01110	001
01100	01100	01011	10101	10010	110
11000	11000	11111	00001	00110	101

Các bộ 2 lỗi khác sẽ rơi vào các syndrom đã dùng

Vậy mã hiệu này sửa được 1 lỗi đơn và hai cấu hình lỗi kép

2.3.Các giới hạn lý thuyết của mã tuyến tính

- Đánh giá bằng số ký hiệu tối đa có thể sửa (phát hiện) được k
- Sử dụng trọng số Hamming, quãng cách tối thiểu d giữa các từ mã bằng với trọng số tối thiểu của từ mã
- Để phát hiện lỗi $k < d$. Để sửa lỗi $2k < d$
- Giới hạn trên của d

$$d \leq \frac{n.m^{k-1}(m-1)}{m^k - 1}$$

trong trường hợp nhị phân

$$d \leq \frac{n.2^{k-1}}{2^k - 1}$$

- Số bít cần dùng thêm để sửa một lỗi

$$m^k(n+1) \leq m^n$$

$$r + k + 1 \leq m^r$$

- Nguyên tắc
  - Dùng mã có chiều dài $2^r - 1$, trong đó r bít sử dụng làm thông tin điều khiển
  - Mã trận thử sẽ có kích thước $(2^r - 1)xr$. Ma trận này không có hai cột nào trùng nhau, do đó tổng hai cột bất kỳ luôn luôn khác 0. Vậy tối thiểu 3 cột cộng lại mới có thể có cột 0
  - Điều kiện để $v = (a_1, a_2 \ldots a_n)$ là một từ mã:

$$\sum_{1}^{n} a_i h_{ij} = 0 \forall j$$

tối thiểu 3 hệ số $a_i \neq 0$. Vậy trọng số tối thiểu của mã là 3, quãng cách tối thiểu là 3, mã sửa được 1 lỗi

- Xét từ mã v khi chuyển đi bị sai một bít thành $u = v + e$. Tính syndrom

$$uH^T = (v + e)H^T = eH^T$$

chính là hàng của H tương ứng với vị trí của lỗi. Có $2^r - 1$ hàng, mỗi hàng có r ký hiệu Vậy có thể chọn H sao cho r ký hiệu chính là số thứ tự của hàng

Ví dụ

- Mã Hamming (7,4). Độ dài từ mã 7, số ký hiệu điều khiển 3, số ký hiệu thông tin 4. Ma trận thử có các cột là số thứ tự của cột

$$\left[\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}\right]$$

- Để quá trình mã hóa đơn giản chọn các ký hiệu điều khiển ở vị trí 1, 2, 4; Các từ mã sẽ có dạng
  $v = (x, y, a_3, z, a_5, a_6, a_7)$
- Các ký hiệu thử được tính theo $vH^T = 0$

$$z + a_5 + a_6 + a_7 = 0 \Rightarrow z = a_5 + a_6 + a_7$$

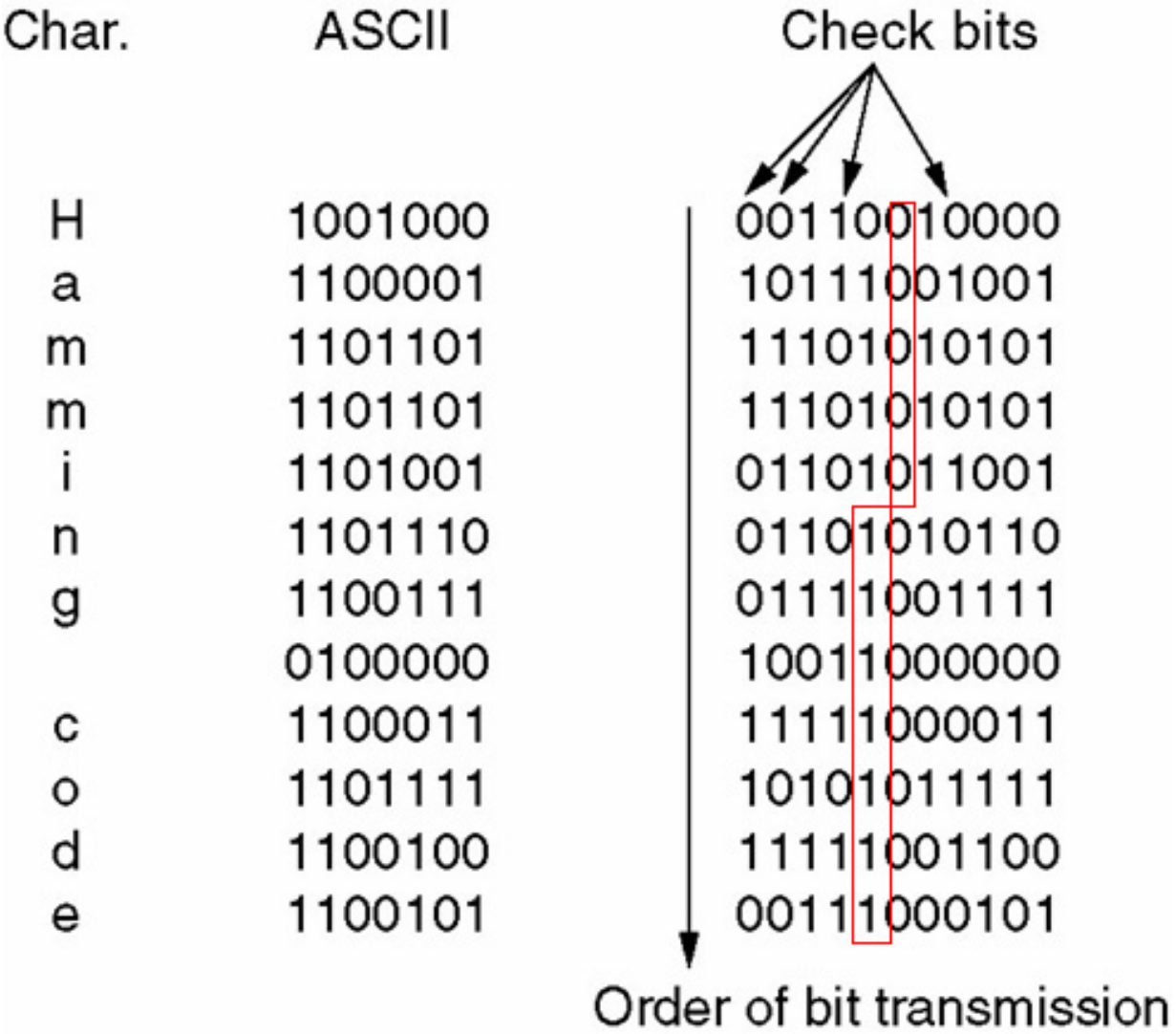
$$y + a_3 + a_6 + a_7 = 0 \Rightarrow y = a_3 + a_6 + a_7$$

$$x + a_3 + a_5 + a_7 = 0 \Rightarrow x = a_3 + a_5 + a_7$$

- Từ công thức trên có thể lập ra bảng các từ mã
- Phát hiện lỗi đơn giản: giá trị của 3 bít điều khiển là vị trí lỗi, nếu là 0,1,2, 4 không có lỗi

Mã Hamming sửa lỗi chùm

- Lỗi chùm: một chuỗi bít liên tiếp bị lỗi
- Giải quyết bằng mã hamming?



3. Mã vòng (CRC)

1. Khái niệm cơ bản
2. Mã tuyến tính
3. Mã vòng (CRC)
   - Khái niệm
   - Tính chất
   - Mã hóa
   - Giải mã
4. Mã chập

3.1.Khái niệm

- Là mã tuyến tính, thường dùng để phát hiện lỗi
- Tất cả các hoán vị của một từ mã là một từ mã $a_1, a_2, \ldots a_n$ là từ mã thì $a_n, a_1, a_2 \ldots a_{n-1}$ cũng là từ mã
- Dựa vào tính chất vòng, sẽ biểu diễn mã vòng bằng một từ mã+một mã gốc.
- Biểu diễn mã vòng: đa thức

$$a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1x + a_0$$

- Tuyến tính định nghĩa trên phép cộng và phép nhân đa thức
  - Phép cộng đa thức: phép cộng đa thức thông thường định nghĩa trên trường hữu hạn các ký hiệu (modulo m)
  - Phép nhân đa thức: nhân 2 đa thức có bậc tối đa n-1, được một đa thức có bậc tối đa $2n-2$. Chia đa thức này cho $x^n - 1$ rồi lấy phần dư làm kết quả

- Định nghĩa phép cộng và nhân như trên để đảm bảo tính đóng của hai phép tính trong tập hợp các từ mã có độ dài n, tạo thành từ một bảng chữ cái có m ký hiệu

- Xét từ mã $a_1, a_2, \ldots a_n$ biểu diễn bằng đa thức

$$a_1 x^{n-1} + a_2 x^{n-2} + \ldots + a_n$$

Nhân đa thức với đa thức x được

$$a_1 x^n + a_2 x^{n-1} + \ldots + a_n x$$

Chia cho $x^n - 1$

$$a_1(x^n - 1) + a_1 + a_2 x^{n-1} + \ldots + a_n x$$

$$a_2 x^{n-1} + \ldots + a_n x + a_1$$

chính là đa thức của từ mã $a_2, \ldots a_n, a_1$

- Vậy phép nhân với x là phép dịch từ mã một ký hiệu

3.1.Khái niệm *(Tiếp)*

- Lần lượt nhân từ mã với $x^2, x^3, \ldots, x^{k-1}$ có k từ mã
- Theo tính chất tuyến tính, có thể tổ hợp các từ mã này tạo ra mã hiệu (n,k) có 2^k từ mã, tuyến tính. Ngược lại?
- Tất cả các từ mã của một mã vòng đều chia hết cho một đa thức

3.2.Tính chất

1. Nếu M(x) là một từ mã, P(x) là một đa thức bậc tối đa $n-1$, thì $M(x)P(x)$ cũng là một từ mã
2. Nếu một mã vòng (n,k), G(x) là một từ mã khác không biểu diễn bởi đa thức có bậc nhỏ nhất thì:
   - Bậc của G là n-k
   - Tất cả các từ mã là kết quả của phép nhân G(x) với một đa thức bậc k
   - Tất cả các từ mã bậc n-k là G(x) nhân với một hằng số
   - Ký hiệu đầu tiên của G khác không
   - Đa thức G(x) gọi là đa thức sinh của mã hiệu

3. Nếu C là một mã vòng với đa thức sinh $G(x) = a_1 + a_2x + \ldots + a_{n-k+1}X^{n-k}$ thì ma trận $k \times n$ sau là ma trận sinh của mã hiệu

$$\begin{pmatrix} a_1 & a_2 & \ldots & a_{n-k+1} & 0 & 0 & 0 & \ldots & 0 \\ 0 & a_1 & a_2 & \ldots & a_{n-k+1} & 0 & 0 & \ldots & 0 \\ 0 & 0 & a_1 & a_2 & \ldots & a_{n-k+1} & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 0 & a_1 & a_2 & \ldots & a_{n-k+1} \end{pmatrix}$$

4. Điều kiện cần và đủ để một đa thức $G(x)$ có thể sinh ra một mã hiệu là tồn tại một đa thức bậc k $H(x)$ sao cho

$$G(X)H(X) = 0$$

Có nghĩa là $G(X)$ phải là thừa số của $D^n - 1$. H(X) còn gọi là đa thức kiểm tra của mã hiệu.

5. Điều kiện cần và đủ để một tổ hợp mã P(x) là từ mã

$$P(X)H(X) = 0$$

- Theo phương pháp mã tuyến tính nếu các ký hiệu mang thông tin tạo thành tổ hợp s(k ký hiệu), từ mã tương ứng sẽ được tạo ra bằng cách $M = sG \Leftrightarrow M(x) = sG(x)$
- Sử dụng đại số đa thức
  - Nếu s(x) là thông tin cần chuyển, chọn tổ hợp mã M(x) sao cho
    $$M(x) = x^{n-k}s(x) - r(x)$$
    Trong đó r(x) là số dư của phép chia $x^{n-k}s(x)$ cho đa thức sinh p(x)
    $$x^{n-k}s(x) = p(x)q(x) + r(x)$$
  - M(x) có đúng n ký hiệu
  - k ký hiệu đầu là của s(x), vì r(x) có bậc tối đa n-k
  - M(x) là từ mã vì
    $M(x) = x^{n-k}s(x) - r(x) = p(x)q(x) + r(x) - r(x) = p(x)q(x)$

3.4.Giải mã

- Nếu không có lỗi
- Phát hiện lỗi: căn cứ vào số dư của đa thức thu được: đa thức syndrom
- Sửa lỗi (nhị phân):
  - lập ra các lớp kề
  - tính syndrom cho mỗi lớp kề
  - Căn cứ vào syndrom thu được để phát hiện vị trí lỗi
- Các đa thức hay sử dụng
  - Hamming
  - Golay
  - Parity Check
  - Sửa lỗi / phát hiện và truyền lại

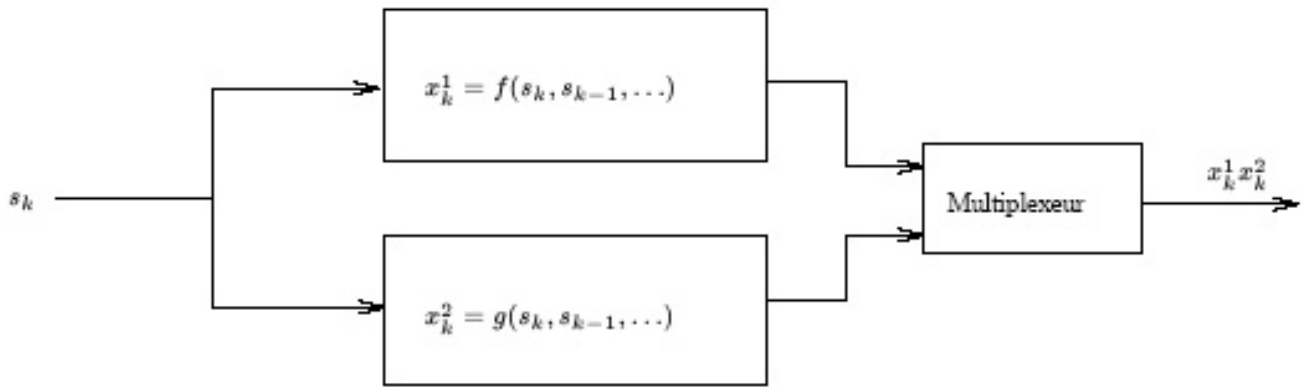
4. Mã chập

1. Khái niệm cơ bản
2. Mã tuyến tính
3. Mã vòng (CRC)
4. Mã chập
   - Khái niệm
   - Mã hóa
   - Biểu diễn
   - Giải mã

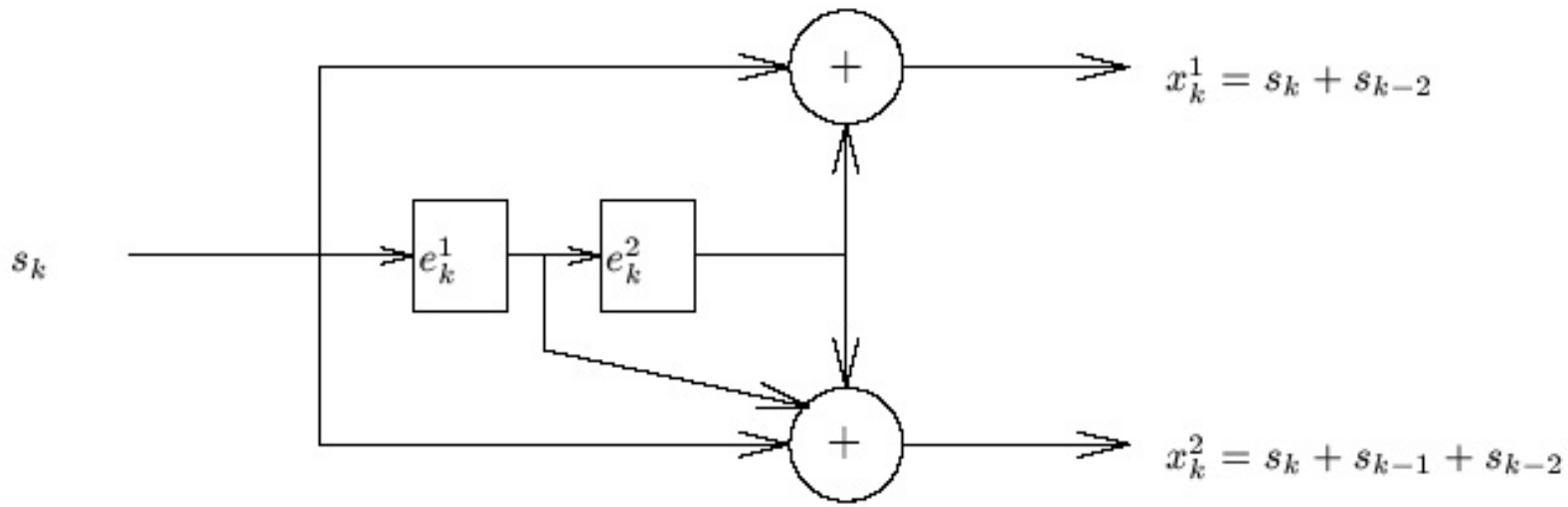
4.1.Khái niệm

- Trường hợp sử dụng
  - Nếu kênh có hệ số nhiễu nhỏ: Sử dụng nhiều ký hiệu, nhiều mức tín hiệu, xử lý tức khắc từng ký hiệu
  - Nếu kênh có hệ số nhiễu lớn: dùng càng ít ký hiệu càng tốt, xử lí một khối các ký hiệu nhận được, dùng tất cả các thông tin của đầu ra kênh tin, sử dụng các tiêu chuẩn thống kê
- Phép toán chập: Nhiều ký hiệu của nguồn đầu vào được đưa tuần tự vào các bộ biến đổi. Kết quả của các phép biến đổi được tổng hợp lại thành đầu ra
- Mã chập biến đổi các ký hiệu nguồn thành các ký hiệu đầu ra sử dụng một bộ nhớ
- Khác với các phương pháp mã hóa đã học, mã chập mã hóa một số lượng tùy ý các ký hiệu cùng một lúc

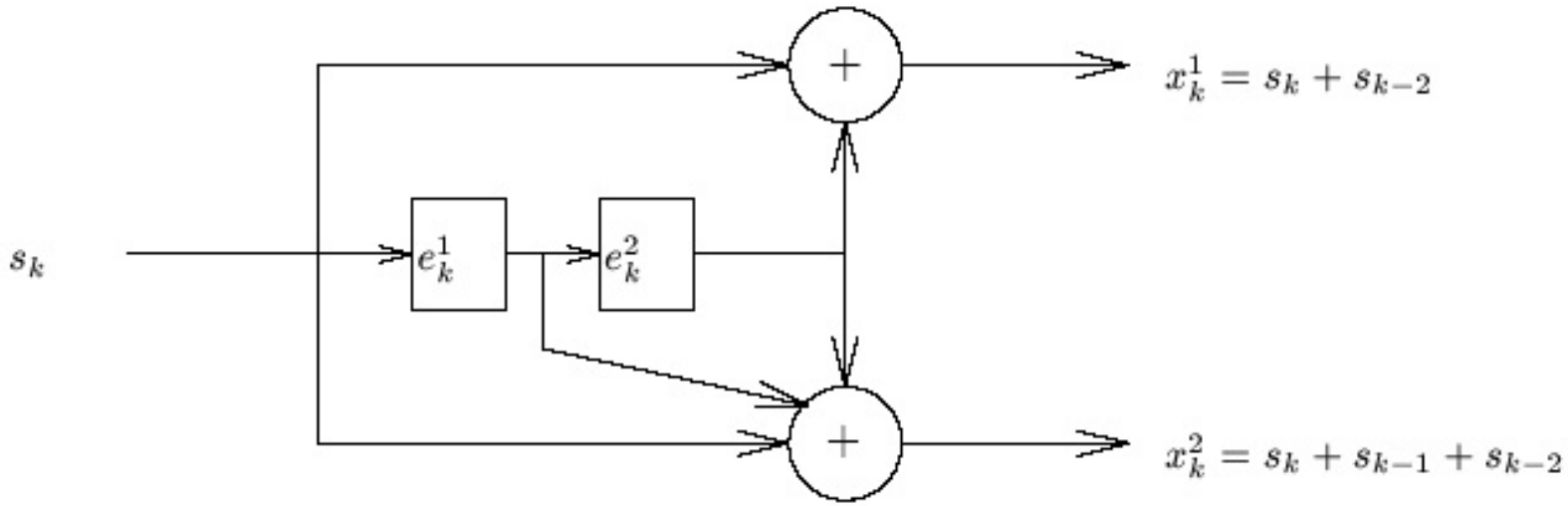
- Tốc độ lập tin đầu ra của mã chập nhỏ hơn tốc độ lập tin đầu vào: $1/R$



- Nguyên tắc
  - Sử dụng một thanh ghi dịch để lưu trữ các ký hiệu đầu vào
  - Sử dụng các mạch logic để tính toán các ký hiệu đầu ra
  - Sử dụng bộ dồn kênh để xếp các ký hiệu đầu ra vào một chuỗi tuần tự



Bằng đa thức



- Lấy D là biến của các đa thức, phép nhân đa thức với D biểu diễn phép dịch sang phải một ô, một ký hiệu, thì

$$s(D) = s_0 + s_1 D + s_2 D^2 + \ldots + s_k D^k + \ldots$$

$$x^i(D) = x_0^i + x_1^i D + x_2^i D^2 + \ldots + x_k^i D^k + \ldots$$

Bằng đa thức *(Tiếp)*

- Biểu diễn theo đặc trưng xung của từng modul: đầu ra $h^i(D)$ tương ứng với đầu vào $10000\ldots000\ldots$

$$x_i(D) = h^i(D)s()D$$

- Trong ví dụ trên

$$h^1(D) = 1 + D^2, h^2(D) = 1 + D + D^2$$

Các trạng thái của bộ mã hóa

$$e^1(D) = Ds(D)$$

$$e^2(D) = De^1(D) = D^2s(Ds)$$

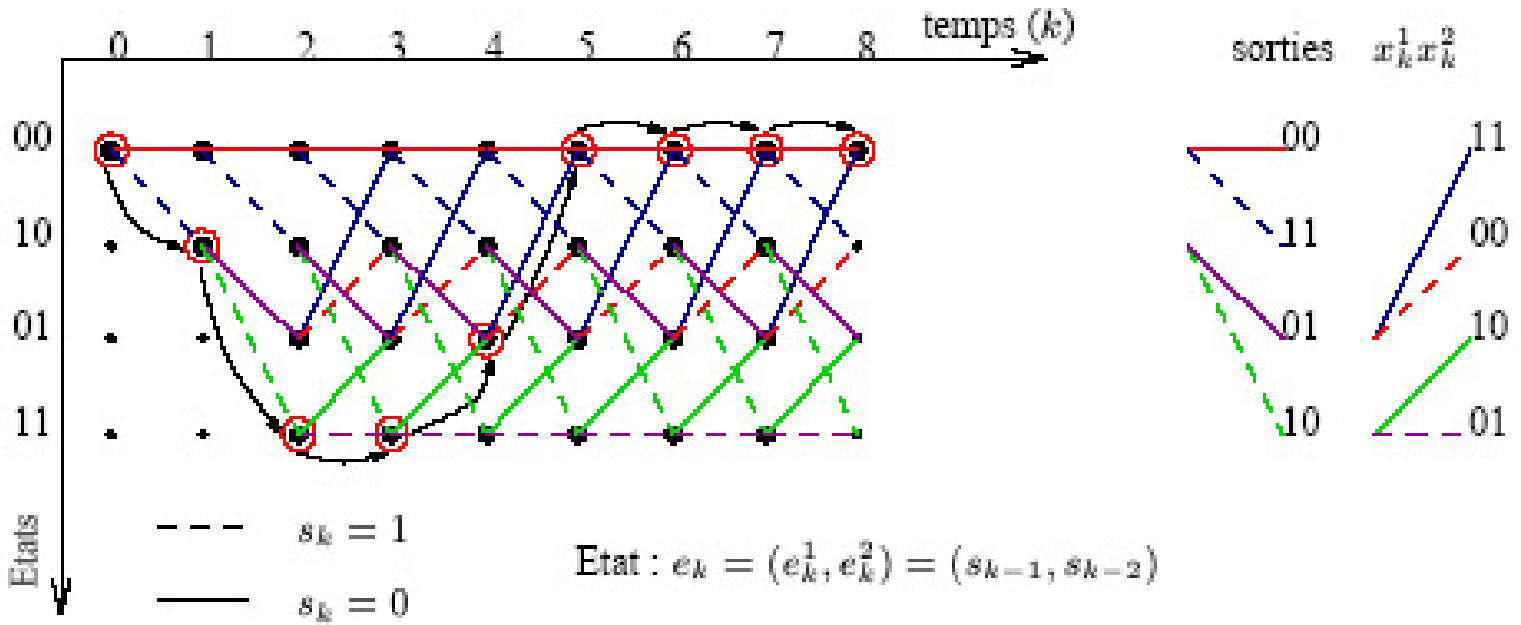
$$x^1(D) = (1 + D^2)s(D)$$

$$x^2(D) = (1 + D + D^2)s(D)$$

- Chú ý: giới hạn của mã chập: số trạng thái là hàm mũ của số ô nhớ

Biểu diễn bằng đồ thị (Trellis)

- Thiết bị mã hóa là một thiết bị có nhớ: Biểu diễn đồ thị thích hợp
- Trục tung: các trạng thái có thể
- Trục hoành: các mốc thời gian
- Các liên kết giữa các điểm: chuyển đổi trạng thái của hệ thống tương ứng với các ký hiệu và các trạng thái tương ứng
- Các thông tin bổ sung: các ký hiệu đầu ra



- Xét nguồn không nhớ và kênh không nhớ. Giả sử dữ liệu truyền đi là X^N, dữ liệu nhận được là Y^N. Cần tìm một chuỗi $\widehat{X}^N$ sao cho xác suất lỗi tối thiểu ($P(X^N \neq \widehat{X}^N)$ tối thiểu), tức là

$$P(\widehat{X}^N|Y^N) \geq P(X^N|Y^N) \forall X^N$$

, tức là

$$P(Y^N|X^N)P(X^N)$$

cực đại Với điều kiện nguồn không nhớ, $P(X^N) = const$, $P(Y^N|X^N) = \prod_1^N P(Y_i^N|X_i^N)$ Vậy cần tìm giá trị tối thiểu của $\sum_1^N -\log P(Y_i^N|X_i^N)$

- Bài toán giải mã tối ưu chuyển về bài toán tìm đường đi ngắn nhất trong Trellis, với các trọng số của đường đi là $-\log P(Y_i^N|X_i^N)$

Thuật toán Viterby

- Bài toán tìm đường ngắn nhất trong đồ thị
  - Độ phức tạp NP
  - chỉ có các lời giải gần đúng
  - Đặc biệt: Trellis. VD 1000 ký hiệu
- Dựa trên cơ sở khẳng định
  Trong một treliss, nếu E^{k+1} là đường đi tối ưu thì E^k là đường đi tối ưu
- Giải thuật Viterby giảm độ phức tạp xuống còn tuyến tính