

# Mục lục

<b>1</b>	<b>Ma trận</b>	<b>4</b>
1.1	Định thức và hạng ma trận . . . . .	4
1.2	Ma trận nghịch đảo . . . . .	6
<b>2</b>	<b>Discrete logarithm</b>	<b>8</b>
2.1	Các thuật toán tính discrete logarithm . . . . .	8
<b>3</b>	<b>Số học</b>	<b>10</b>
3.1	Phép chia Euclid và thuật toán Euclid . . . . .	10
3.2	Hàm Euler . . . . .	13
3.3	Thặng dư chính phương . . . . .	16
<b>4</b>	<b>Lattice-based crypto</b>	<b>18</b>
4.1	Introduction . . . . .	18
4.2	Thuật toán GGH . . . . .	20
<b>5</b>	<b>Quan hệ hai ngôi</b>	<b>22</b>
5.1	Quan hệ hai ngôi . . . . .	22
<b>6</b>	<b>Bài toán đếm Polya</b>	<b>26</b>
6.1	Bổ đề Burnside . . . . .	26
<b>7</b>	<b>Giải tích</b>	<b>30</b>
7.1	Giới hạn . . . . .	30
7.2	Đạo hàm . . . . .	32
<b>8</b>	<b>Quantum computing</b>	<b>36</b>
8.1	Qubit và toán tử quantum . . . . .	36
<b>9</b>	<b>Lý thuyết xác suất</b>	<b>41</b>

<i>MỤC LỤC</i>	2
9.1 Introduction . . . . .	41
9.2 Biến ngẫu nhiên . . . . .	44
<b>10 Hình học giải tích</b>	<b>49</b>
10.1 Theo dòng lịch sử . . . . .	49
10.2 Phương pháp tọa độ trong mặt phẳng . . . . .	54
10.3 Đạo hàm . . . . .	58
10.4 Tích phân . . . . .	61
<b>11 Hình học affine</b>	<b>65</b>
11.1 Không gian affine . . . . .	65
<b>12 Machine Learning</b>	<b>76</b>
12.1 Các thuật toán cơ sở . . . . .	76
<b>13 Zero Knowledge Proofs</b>	<b>84</b>
13.1 Zero knowledge proof . . . . .	84
<b>A NSUCRYPTO 2021</b>	<b>87</b>
Problem 9. 2021-bit key . . . . .	87
<b>B NSUCRYPTO 2022</b>	<b>89</b>
Lời nói đầu . . . . .	89
Problem 5*. Super dependent S-box . . . . .	89
Problem 11. A long awaited . . . . .	90
<b>C NSUCRYPTO 2023</b>	<b>92</b>
Problem 1. Affine cipher . . . . .	92
Problem 2. Simple ideas for primes . . . . .	93
Problem 3. Mixed hash . . . . .	94
Problem 4*. Column functions . . . . .	95
Problem 5. Primes . . . . .	98
Problem 6**. An aggregated signature . . . . .	98
Problem 7. A unique coding . . . . .	99
Problem 8. Algebraic cryptanalysis . . . . .	103
Problem 9**. Finite-state machines . . . . .	105
Problem 10. Quantum encryption . . . . .	106

<i>MỤC LỤC</i>	<b>3</b>
Problem 11. AntCipher . . . . .	110
<b>D Olympiad</b>	<b>113</b>
D.1 Ôn thi ngày 20/11/2023 . . . . .	113
D.2 RUDN Olympiad 2023 . . . . .	115
<b>E Đường đoản thời</b>	<b>117</b>
<b>F Assembly, stack và heap</b>	<b>124</b>
F.1 Lệnh assembly cơ bản . . . . .	124
F.2 Stack và Heap . . . . .	127

# Chương 1

## Ma trận

Trong các bài viết của về đại số tuyến tính:

- Vector sẽ được ký hiệu bởi chữ thường in đậm (ví dụ  $\mathbf{v}, \mathbf{x}, \dots$ );
- Ma trận sẽ được ký hiệu bởi chữ hoa in đậm (ví dụ  $\mathbf{A}, \mathbf{B}, \dots$ );
- Các đại lượng vô hướng (số) được ký hiệu bởi chữ thường không in đậm (ví dụ  $x_1, N, t, \dots$ ).

### 1.1 Định thức và hạng ma trận

#### Định thức ma trận

##### Định nghĩa 1. Nghịch thế

Cho tập hợp  $A = \{1, 2, \dots, n\}$  và xét hoán vị  $\sigma$  trên  $A$ . Ta gọi hai phần tử  $i$  và  $j$  tạo thành **nghịch thế** (inversion) nếu  $i < j$  và  $\sigma(i) > \sigma(j)$ .

Đặt  $\sigma = \{k_1, k_2, \dots, k_n\}$  là một hoán vị của  $A$ . Ta ký hiệu

$$P\{k_1, k_2, \dots, k_n\}$$

là số lượng nghịch thế của  $\sigma$  và đặt

$$(-1)^{P\{k_1, k_2, \dots, k_n\}} = \text{sign}\{k_1, k_2, \dots, k_n\}.$$

**Ví dụ 1.** Với  $n = 4$ ,  $A = \{1, 2, 3, 4\}$ . Xét hoán vị  $\sigma = \{4, 2, 1, 3\}$ .

Ta nhận thấy các cặp nghịch thế  $(4, 2)$ ,  $(4, 1)$ ,  $(4, 3)$ ,  $(2, 1)$  gồm 4 cặp nghịch thế. Vậy  $P\{4, 2, 1, 3\} = 4$  và  $\text{sign}\{4, 2, 1, 3\} = (-1)^4 = 1$ .

**Định nghĩa 2. Định thức**

Khi đó định thức của ma trận  $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$  được định nghĩa là:

$$\det(\mathbf{A}) = \sum_{(i_1, i_2, \dots, i_n)} a_{1, i_1} \cdot a_{2, i_2} \cdot a_{n, i_n} \cdot \text{sign}\{i_1, i_2, \dots, i_n\} \quad (1.1)$$

với mọi hoán vị  $(i_1, i_2, \dots, i_n)$  của  $(1, 2, \dots, n)$ . Như vậy có  $n!$  phần tử cho tổng trên.

**Ví dụ 2.** Tính định thức ma trận  $\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ .

Xét hoán vị  $\sigma_1 = \{1, 2, 3\}$ . Khi đó  $P\{1, 2, 3\} = 0$ ,  $a_{11} \cdot a_{22} \cdot a_{33} \cdot (-1)^0 = 1 \cdot 5 \cdot 9 \cdot 1 = 45$ .

Xét hoán vị  $\sigma_2 = \{1, 3, 2\}$ . Khi đó  $P\{1, 3, 2\} = 1$ ,  $a_{11} \cdot a_{23} \cdot a_{32} \cdot (-1)^1 = 1 \cdot 6 \cdot 8 \cdot (-1) = -48$ .

Xét hoán vị  $\sigma_3 = \{2, 1, 3\}$ . Khi đó  $P\{2, 1, 3\} = 1$ ,  $a_{12} \cdot a_{21} \cdot a_{33} \cdot (-1)^1 = 2 \cdot 4 \cdot 9 \cdot (-1) = -72$ .

Xét hoán vị  $\sigma_4 = \{2, 3, 1\}$ . Khi đó  $P\{2, 3, 1\} = 2$ ,  $a_{12} \cdot a_{23} \cdot a_{31} \cdot (-1)^2 = 2 \cdot 6 \cdot 7 \cdot 1 = 84$ .

Xét hoán vị  $\sigma_5 = \{3, 1, 2\}$ . Khi đó  $P\{3, 1, 2\} = 2$ ,  $a_{13} \cdot a_{21} \cdot a_{32} \cdot (-1)^2 = 3 \cdot 4 \cdot 8 \cdot 1 = 96$ .

Xét hoán vị  $\sigma_6 = \{3, 2, 1\}$ . Khi đó  $P\{3, 2, 1\} = 3$ ,  $a_{13} \cdot a_{22} \cdot a_{31} \cdot (-1)^3 = 3 \cdot 5 \cdot 7 \cdot (-1) = -105$ .

Như vậy  $\det(\mathbf{A}) = 45 - 48 - 72 + 84 + 96 - 105 = 0$ .

Định thức của ma trận còn được định nghĩa theo **đệ quy** như sau:

Với ma trận  $1 \times 1$  là  $\mathbf{A} = (a_{11})$  thì  $\det(\mathbf{A}) = a_{11}$ .

Với ma trận  $2 \times 2$  là  $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$  thì  $\det(\mathbf{A}) = a_{11}a_{22} - a_{21}a_{12}$ .

Với ma trận  $n \times n$ , gọi  $\mathbf{M}_{ij}$  là ma trận có được từ ma trận  $\mathbf{A}$  khi bỏ đi hàng  $i$  và cột  $j$  của ma trận  $\mathbf{A}$  và ký hiệu  $A_{ij} = (-1)^{i+j} \det(\mathbf{M}_{ij})$ . Khi đó:

**Định lí 1. Định lý Laplace**

Định lý Laplace cho phép ta khai triển định thức của ma trận cấp  $n$  thành tổng các ma trận cấp  $n - 1$ .

Khai triển theo cột  $j$ :

$$\det(\mathbf{A}) = \sum_{i=1}^n a_{ij} A_{ij} = a_{1j} A_{1j} + a_{2j} A_{2j} + \cdots + a_{nj} A_{nj}, \quad j = \overline{1, n}.$$

Khai triển theo hàng  $i$ :

$$\det(\mathbf{A}) = \sum_{j=1}^n a_{ij} A_{ij} = a_{i1} A_{i1} + a_{i2} A_{i2} + \cdots + a_{in} A_{in}, \quad i = \overline{1, n}.$$

**Hạng của ma trận****Định nghĩa 3. Hạng của ma trận**

Cho ma trận  $\mathbf{A}_{m \times n}$ . **Hạng** của ma trận là cấp của ma trận con lớn nhất có định thức khác 0. Nghĩa là một ma trận vuông mà là ma trận con (lấy 1 phần của ma trận gốc) kích thước  $r \times r$  mà có định thức khác 0, thì hạng của ma trận khi đó là  $r$ . Dễ thấy do là ma trận con, và vuông, nên  $r \leq \min(m, n)$ .

**Ví dụ 3.** Ma trận  $\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 1 & 2 & 4 \end{pmatrix}$  có định thức  $\det(\mathbf{A}) = 0$ .

Nhưng ma trận con của  $\mathbf{A}$  là  $\mathbf{B} = \begin{pmatrix} 2 & 3 \\ 2 & 4 \end{pmatrix}$  (lấy dòng 1 và 3, lấy cột 2 và 3) có định thức  $\det(\mathbf{B}) = 2 \neq 0$ , do đó  $r = \text{rank}(\mathbf{A}) = 2$  ( $\text{rank}(\mathbf{A})$  nghĩa là hạng của  $\mathbf{A}$ ).

**1.2 Ma trận nghịch đảo**

Ma trận  $\mathbf{A}^{-1}$  được gọi là **ma trận nghịch đảo** của ma trận vuông  $\mathbf{A}$  nếu  $\mathbf{A}^{-1} \cdot \mathbf{A} = \mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{I}$ . Trong đó  $\mathbf{I}$  là ma trận đơn vị cùng kích thước với  $\mathbf{A}$ .

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} [(A_{ij})_n]^T = \frac{1}{\det(\mathbf{A})} \begin{pmatrix} A_{11} & A_{21} & \cdots & A_{n1} \\ A_{12} & A_{22} & \cdots & A_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ A_{1n} & A_{2n} & \cdots & A_{nn} \end{pmatrix} \quad (1.2)$$

Trong đó,  $A_{ij}$  cũng được định nghĩa tương tự như khi tính định thức bằng khai triển theo dòng hoặc cột. Gọi  $\mathbf{M}_{ij}$  là ma trận có được từ ma trận  $\mathbf{A}$  khi bỏ đi hàng  $i$  và cột  $j$  của ma trận  $\mathbf{A}$  và ký hiệu  $A_{ij} = (-1)^{i+j} \det(\mathbf{M}_{ij})$ .

Như vậy, điều kiện cần và đủ để một ma trận có nghịch đảo là định thức khác 0.

## Chương 2

# Discrete logarithm

### 2.1 Các thuật toán tính discrete logarithm

Thuật toán Baby-Step-Giant-Step (BSGS) giúp tính discrete logarithm trên nhóm cyclic với order là số nguyên tố 1.

---

**Algorithm 1** Thuật toán Baby-Step-Giant-Step

---

**Require:** Nhóm cyclic  $G$  có order  $n$ , generator  $g$  và phần tử  $h \in G$ .

**Ensure:** Số  $x$  duy nhất thuộc  $\{0, 1, \dots, n-1\}$  thỏa  $g^x = h$ .  $m \leftarrow \lfloor \sqrt{n} \rfloor$

```
1: for  $j = 0 \rightarrow m-1$  do
2:   Tính  $g^j$ . Lưu  $(j, g^j)$  vào bảng.
3: end for
4: Tính  $g^{-m}$ .
5:  $\gamma \leftarrow h$ .
6: for  $i = 0 \rightarrow m-1$  do
7:   a) Kiểm tra điều kiện  $\gamma = g^j$  với  $j = 0, 1, \dots, m-1$ .
8:   b) Nếu điều kiện thỏa, trả về  $im + j$ .
9:   c) Nếu không, đặt  $\gamma \leftarrow \gamma \cdot g^{-m}$ .
10: end for
```

---

Khi order của cyclic group là lũy thừa một số nguyên tố thì ta dùng thuật toán Pohlig-Hellman 2.



---

**Algorithm 2** Thuật toán Pohlig-Hellman

---

**Require:** Nhóm cyclic  $G$  có order  $n = p^e$ , generator  $g$  và phần tử  $h \in G$ .

**Ensure:** Số  $x$  duy nhất thuộc  $\{0, 1, \dots, n - 1\}$  thỏa  $g^x = h$ .

- 1: Khởi tạo  $x_0 = 0$ .
  - 2: Tính  $\gamma = g^{p^{e-1}}$ . Theo định lý Lagrange,  $\gamma$  có order là  $p$ .
  - 3: **for**  $k = 0 \rightarrow e - 1$  **do**
  - 4:     a) Tính  $h_k = (g^{-x_k} \cdot h)^{p^{e-1-k}}$ .
  - 5:     b) Sử dụng thuật toán baby-step-giant-step, tìm  $d_k \in \{0, 1, \dots, p - 1\}$  sao cho  $\gamma^{d_k} = h_k$ .
  - 6:     c) Tính  $x_{k+1} = x_k + p^k d_k$ .
  - 7: **end for**
  - 8: Trả về  $x_e$  là kết quả cần tìm.
-

# Chương 3

## Số học

Toán học là vua của các môn  
khoa học, và số học là nữ hoàng

---

Carl Friedrich Gauss

### 3.1 Phép chia Euclid và thuật toán Euclid

#### Phép chia Euclid

Đây là nền tảng, cơ sở của số học. Từ khi biết tới phép chia hai số nguyên, ta có thể tìm *thương* và *số dư*. Nói theo toán học, nếu ta có hai số nguyên dương  $a$  và  $b$  thì tồn tại cặp số  $q, r$  sao cho  $a = qb + r$  với  $0 \leq r < b$ .

Khi đó,  $a$  gọi là số bị chia,  $b$  gọi là số chia,  $q$  là thương (q trong quotient) và  $r$  là số dư (r trong remainder).

Đặc biệt là sự tồn tại của cặp số  $q$  và  $r$  là duy nhất. Thật vậy, nếu ta giả sử tồn tại 2 cặp số  $(q_1, r_1)$  và  $(q_2, r_2)$  đều thỏa đẳng thức trên, nghĩa là

$$a = q_1b + r_1, \quad a = q_2b + r_2$$

Trừ 2 đẳng thức vế theo vế ta có  $(q_1 - q_2)b + (r_1 - r_2) = 0$ . Tương đương  $(r_2 - r_1) = (q_1 - q_2)b$ , mà  $0 \leq r_1, r_2 < b$  nên  $-b < r_2 - r_1 < b$ . Như vậy chỉ có thể xảy ra trường hợp  $r_2 - r_1 = 0$  hay  $r_2 = r_1$ , kéo theo  $q_1 = q_2$ .

#### Thuật toán Euclid

Dựa trên phép chia Euclid, ta có một thuật toán hiệu quả để tìm ước chung lớn nhất giữa hai số  $a$  và  $b$ .

Ký hiệu  $\gcd(a, b)$  là ước chung lớn nhất của  $a$  và  $b$ . Chúng ta thực hiện đệ quy như sau:

$$\gcd(a, b) = \begin{cases} a, & \text{nếu } b = 0 \\ \gcd(b, a \bmod b), & \text{nếu } b \neq 0 \end{cases}$$

Điểm quan trọng ở thuật toán Euclid là thuật toán chắc chắn sẽ dừng sau một số hữu hạn bước, và kết quả sẽ là ước chung lớn nhất của hai số  $a$  và  $b$ .

*Chứng minh.* Đặt  $r_0 = a$  và  $r_1 = b$ . Theo thuật chia Euclid ta có các số  $q_0$  và  $r_2$  sao cho  $r_0 = r_1 q_0 + r_2$  với  $0 \leq r_2 < r_1$ . Thuật toán Euclid hoạt động như sau:

$$\begin{aligned} r_0 &= r_1 q_0 + r_2 \\ r_1 &= r_2 q_1 + r_3 \\ r_2 &= r_3 q_2 + r_4 \\ &\dots = \dots \\ r_i &= r_{i+1} q_i + r_{i+2} \\ &\dots = \dots \\ r_k &= r_{k+1} q_k + 0 \\ r_{k+1} &= 0 \end{aligned}$$

Ta thấy rằng ở mỗi bước,  $r_{i+2}$  luôn nhỏ hơn  $r_{i+1}$ . Do đó cuối cùng sẽ bằng 0, và khi đó ta có ước chung lớn nhất.  $\square$

## Thuật toán Euclid mở rộng

### Định nghĩa 1. Phương trình Diophantos

Cho trước các số nguyên  $a, b$  và  $c$ . Phương trình Diophantos là phương trình có dạng

$$ax + by = c$$

với  $x, y$  là các số nguyên.

**Ví dụ 1.** Giải phương trình  $5x + 3y = 1$ .

Ta có  $y = \frac{1 - 5x}{3} = \frac{1 - 2x - 3x}{3} = \frac{1 - 2x}{3} - x$ . Như vậy nếu  $y \in \mathbb{Z}$  thì  $\frac{1 - 2x}{3} \in \mathbb{Z}$ , nghĩa là  $1 - 2x$  chia hết cho 3. Vậy  $1 - 2x = 3k$  với  $k \in \mathbb{Z}$ .

Tiếp tục,  $1 - 2x = 3k$ , suy ra  $x = \frac{1 - 3k}{2} = \frac{1 - k - 2k}{2} = \frac{1 - k}{2} - k$ . Do  $x$  nguyên nên tương tự  $\frac{1 - k}{2}$  cũng nguyên, hay  $1 - k = 2t$ , tương đương với  $k = 1 - 2t$ .

Thay ngược lại ta có  $x = \frac{1-3k}{2} = \frac{1-3(1-2t)}{2} = -1+3t$ . Tiếp tục thay vào để tìm  $y$  thì  $y = \frac{1-5x}{3} = \frac{1-5(-1+3t)}{3} = 2-5t$ .

Như vậy nghiệm của phương trình là tất cả các nghiệm  $(x, y)$  mà  $x = -1+3t$ ,  $y = 2-5t$  với  $t \in \mathbb{Z}$ .

Ở đây chúng ta đã thực hiện phép chia có dư liên tiếp để tìm nghiệm. Nói cách khác ta đã thực hiện thuật toán Euclid ở bên trên để làm giảm độ phức tạp ở mỗi bước giải. Tổng quát ta có thuật toán Euclid mở rộng để tìm ước chung lớn nhất  $\gcd(a, b)$  của hai số  $a, b$ , và **một** nghiệm của phương trình  $ax + by = \gcd(a, b)$ .

Ở ví dụ trên, ta thấy rằng  $(-1, 2)$  là một nghiệm của phương trình  $5x + 3y = 1$ . Khi đó ta có thể suy ra tất cả nghiệm (họ nghiệm) của phương trình có dạng  $(-1+3t, 2-5t)$  với  $t \in \mathbb{Z}$ .

---

**Algorithm 3** Thuật toán Euclid mở rộng

---

**Require:**  $a, b \in \mathbb{Z}$

**Ensure:**  $\gcd(a, b), x, y$

$r_0 \leftarrow a, r_1 \leftarrow b, r_2 \leftarrow 0$

$x_0 \leftarrow 1, x_1 \leftarrow 0, x_2 \leftarrow 0$

$y_0 \leftarrow 0, y_1 \leftarrow 1, y_2 \leftarrow 0$

**while**  $r_1 \neq 0$  **do**

$q \leftarrow r_0 \text{ div } r_1$

$r_2 \leftarrow r_0 - q * r_1, r_0 \leftarrow r_1, r_1 \leftarrow r_2$

$x_2 \leftarrow x_0 - q * x_1, x_0 \leftarrow x_1, x_1 \leftarrow x_2$

$y_2 \leftarrow y_0 - q * y_1, y_0 \leftarrow y_1, y_1 \leftarrow y_2$

**end while**

**return**  $r_0, x_0, y_0$

---

Ở thuật toán trên,  $r_0, r_1$  và  $r_2$  hoạt động như thuật toán Euclid chuẩn. Ở mỗi bước  $q$  là thương của phép chia hai số nguyên và ta sử dụng  $q$  đó để tính  $x_0$  và  $y_0$  mới. Kết quả cuối cùng  $(r_0, x_0, y_0)$  lần lượt là ước chung lớn nhất, và hai số  $x, y$  thỏa mãn  $ax_0 + yb_0 = r_0$ .

Tại sao chúng ta lại có  $(x_0, x_1) = (1, 0)$  và  $(y_0, y_1) = (0, 1)$ ? Nói cách khác, làm sao biết thuật toán hoạt động đúng?

Mục đích của chúng ta là tìm các số  $(x, y)$  sao cho  $ax + by = \gcd(a, b)$ . Khi đó, dựa trên thuật toán Euclid cơ bản ở trên, ta xây dựng dãy số  $\{x_n\}$  và  $\{y_n\}$  sao cho ở mọi bước thứ  $n$  ta đều có

$$ax_n + by_n = r_n \tag{3.1}$$

Ta có  $r_i = r_{i+1}q_i + r_{i+2}$ . Từ  $q_i$  ở mỗi bước ta tính được

$$x_i = x_{i+1}q_i + x_{i+2}, \quad y_i = y_{i+1}q_i + y_{i+2}$$

Thay vào 3.1 ta được

$$a(x_{i+1}q_i + x_{i+2}) + b(y_{i+1}q_i + y_{i+2}) = r_i \quad (3.2)$$

Tương đương với

$$(ax_{i+1} + by_{i+1})q_i + (ax_{i+2} + by_{i+2}) = r_i$$

Mà  $ax_{i+1} + by_{i+1} = r_{i+1}$  và  $ax_{i+2} + by_{i+2} = r_{i+2}$ . Suy ra  $r_{i+1}q_i + r_{i+2} = r_i$ , đúng với thuật toán Euclid chuẩn ban đầu. Nghĩa là thuật toán hoạt động đúng. Bây giờ ta cần chọn  $(x_0, x_1)$  và  $(y_0, y_1)$  vì chúng ta đã đặt  $r_0 = a$  và  $r_1 = b$ . Ở bước thứ 0,

$$r_0 = a = ax_0 + by_0$$

và ở bước thứ 1,

$$r_1 = b = ax_1 + by_1$$

Để thấy ở bước 0 ta chọn  $(1, 0)$  và ở bước 1 ta chọn  $(0, 1)$  là được.

## 3.2 Hàm Euler

### Định nghĩa 2. Phi hàm Euler

Cho số nguyên dương  $n$ . Số lượng các số dương nhỏ hơn  $n$  và nguyên tố cùng nhau với  $n$  được ký hiệu bởi  $\phi(n)$  và gọi là  $\phi$  hàm Euler.

$$\phi(n) = |\{a : (a, n) = 1\}|$$

Hàm Euler có ý nghĩa quan trọng trong lý thuyết số, công cụ giúp chúng ta giải các vấn đề về số mũ trong modulo.

Sau đây chúng ta xem xét hệ thặng dư đầy đủ và hệ thặng dư thu gọn.

Với số nguyên dương  $n$ , ta định nghĩa

### Định nghĩa 3. Hệ thặng dư đầy đủ

Hệ thặng dư đầy đủ của  $n$  là tập  $\{0, 1, \dots, n-1\}$ .

Nói cách khác, hệ thặng dư đầy đủ của  $n$  là các số dư có thể có khi chia một số bất kì cho  $n$ .

$$\begin{array}{cccc} 1 & m+1 & \cdots & (n-1)m+1 \\ 2 & m+2 & \cdots & (n-1)m+2 \\ \cdots & \cdots & \cdots & \cdots \\ m & m+m & \cdots & (n-1)m+m \end{array}$$

#### Định nghĩa 4. Hệ thặng dư thu gọn

Hệ thặng dư thu gọn của  $n$  là tập các số  $a$  mà  $1 \leq a < n$  và  $(a, n) = 1$ . Số lượng các số  $a$  như vậy là  $\phi(n)$ .

#### Nhận xét 1

1. Hệ thặng dư thu gọn của  $n$  gồm  $\phi(n)$  phần tử là

$$\{a_1, a_2, \dots, a_{\phi(n)}\}$$

2. Nếu  $n$  là số nguyên tố thì  $\phi(n) = n - 1$ .

### Tính chất hàm Euler

#### Nhận xét 2

Với  $(m, n) = 1$  thì  $\phi(mn) = \phi(m)\phi(n)$ .

*Chứng minh.* Ta viết các số từ 1 tới  $mn$  thành bảng như sau

Hàng  $r$  gồm các phần tử dạng  $rm + k$  với  $0 \leq r \leq n - 1$  và  $1 \leq k \leq m$ . Ta thấy rằng nếu  $(rm + k, m) = 1$  thì  $(k, m) = 1$ .

Do đó trên mỗi hàng có  $\phi(m)$  phần tử nguyên tố cùng nhau với  $m$ .

Tiếp theo, trên các hàng vừa tìm được, do  $(m, n) = 1$  nên để  $(rm + k, n) = 1$  thì  $(r, n) = 1$ . Nghĩa là có  $\phi(n)$  hàng như vậy.

Tổng kết lại, ta có  $\phi(m)\phi(n)$  phần tử trong bảng nguyên tố cùng nhau với  $mn$ . Do đó có điều phải chứng minh.  $\square$

Do tính chất này nên hàm Euler là hàm nhân tính.

#### Nhận xét 3

Cho số nguyên dương  $n$ . Khi đó  $\sum_{d|n} \phi(d) = n$ .

*Chứng minh.* Giả sử phân tích thừa số nguyên tố của  $n$  là

$$n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$$

Khi đó mỗi ước  $d$  của  $n$  đều có dạng  $p_1^{f_1} p_2^{f_2} \dots p_k^{f_k}$  với  $0 \leq f_i \leq e_i$  với  $i = 1, 2, \dots, k$ .

Như vậy

$$\sum_{d|n} \phi(d) = \sum_{0 \leq f_i \leq e_i} \phi(p_1^{f_1} p_2^{f_2} \dots p_k^{f_k}) = \phi(p_1^{f_1}) \phi(p_2^{f_2}) \dots \phi(p_k^{f_k})$$

Một dạng biểu thức đơn giản là  $(1+x)(1+y) = 1+x+y+xy$  hay với 3 biến là  $(1+x)(1+y)(1+z) = 1+x+y+z+xy+yz+zx+xyz$ . Tổng quát cho  $k$  biến ở trên thì biểu thức tương đương với

$$\begin{aligned} \sum_{0 \leq f_i \leq e_i} \phi(p_1^{f_1}) \phi(p_2^{f_2}) \dots \phi(p_k^{f_k}) &= (1 + \phi(p_1) + \phi(p_1^2) + \dots + \phi(p_1^{e_1})) \\ &\quad \times (1 + \phi(p_2) + \phi(p_2^2) + \dots + \phi(p_2^{e_2})) \\ &\quad \times \dots \\ &\quad \times (1 + \phi(p_k) + \phi(p_k^2) + \dots + \phi(p_k^{e_k})) \end{aligned}$$

Ở đây ta rút gọn dễ dàng với  $i = 1, 2, \dots, k$ :

$$\begin{aligned} &1 + \phi(p_i) + \phi(p_i^2) + \dots + \phi(p_i^{e_i}) \\ &= 1 + p_i - 1 + p_i^2 - p_i + \dots + p_i^{e_i} - p_i^{e_i-1} \\ &= p_i^{e_i} \end{aligned}$$

Như vậy mỗi tổng  $1 + \phi(p_i) + \dots$  bằng chính  $p_i^{e_i}$ . Nhân chúng lại với nhau ta có lại  $n$ .  $\square$

## Định lý Euler

### Định lí 1. Định lý Euler

Cho số nguyên dương  $n$ . Với mọi số nguyên  $a$  mà  $(a, n) = 1$  thì

$$a^{\phi(n)} \equiv 1 \pmod{n} \quad (3.3)$$

*Chứng minh.* Giả sử  $S = \{a_1, a_2, \dots, a_{\phi(n)}\}$  là hệ thặng dư thu gọn của  $n$ . Ta sẽ chứng minh rằng nếu  $a$  là số sao cho  $(a, n) = 1$  thì tập hợp

$$\{aa_1, aa_2, \dots, aa_{\phi(n)}\}$$

là hoán vị của tập  $S$ .

Thật vậy, giả sử  $aa_i \equiv aa_j \pmod{n}$  với  $1 \leq i, j \leq \phi(n)$  và  $i \neq j$ .

Do  $(a, n) = 1$  nên tồn tại nghịch đảo  $a' \pmod{n}$ , nhân  $a'$  cho 2 vế ta còn  $a_i \equiv a_j \pmod{n}$ .

Nói cách khác, nếu  $a_i \not\equiv a_j \pmod{n}$  thì  $aa_i \not\equiv aa_j \pmod{n}$ . Suy ra tập

$$\{aa_1, aa_2, \dots, aa_{\phi(n)}\}$$

là hoán vị của  $S$ .

Ta nhân tất cả phần tử của  $S$  thì sẽ bằng tích phần tử của tập trên

$$aa_1 \cdot aa_2 \dots aa_{\phi(n)} \equiv a_1 \cdot a_2 \dots a_{\phi(n)} \pmod{n}$$

Đặt  $I = a_1 \cdot a_2 \dots a_{\phi(n)}$  thì phương trình trên tương đương với

$$a^{\phi(n)} I \equiv I \pmod{n}$$

Mà  $(I, n) = 1$  do là tích các số nguyên tố cùng nhau với  $n$  nên rút gọn hai vế ta được

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

Ta có điều phải chứng minh. □

## Định lý Fermat nhỏ

### Định lí 2. Định lý Fermat nhỏ

Cho số nguyên tố  $p$ . Với mọi số nguyên  $a$  thì

$$a^p \equiv a \pmod{p}$$

Khi  $(a, p) = 1$  thì

$$a^{p-1} \equiv 1 \pmod{p}$$

### Nhận xét 4

Khi  $(a, p) = 1$  thì định lý Fermat là hệ quả trực tiếp từ định lý Euler.

## 3.3 Thặng dư chính phương



**Định nghĩa 5. Số chính phương modulo  $p$** 

Xét số dương nguyên tố lẻ  $p$ . Số  $a$  được gọi là **số chính phương modulo  $p$**  nếu  $(a, p) = 1$  và tồn tại số  $x$  sao cho  $x^2 = a \pmod{p}$ .

Nói cách khác phương trình đồng dư  $x^2 \equiv a \pmod{p}$  có nghiệm.

Chúng ta sử dụng kí hiệu Legendre (Legendre symbol) để thể hiện một số  $a$  có phải là số chính phương modulo nguyên tố  $p$  không.

**Định nghĩa 6. Legendre symbol**

Xét  $p$  là số nguyên tố,  $a$  là số nguyên không chia hết cho  $p$ . Khi đó kí hiệu Legendre được định nghĩa là

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & \text{nếu } a \text{ là số chính phương modulo } p. \\ -1, & \text{nếu ngược lại.} \end{cases} \quad (3.4)$$

Một trường hợp tổng quát hơn của kí hiệu Legendre là kí hiệu Jacobi áp dụng cho số nguyên dương bất kì.

**Định nghĩa 7. Jacobi symbol**

Xét  $n$  là số nguyên dương,  $a$  là số nguyên không chia hết cho  $n$ . Khi đó kí hiệu Jacobi được định nghĩa là

$$\left(\frac{a}{n}\right) = \begin{cases} 1, & \text{nếu } a \text{ là số chính phương modulo } n \\ -1, & \text{nếu ngược lại.} \end{cases} \quad (3.5)$$

# Chương 4

## Lattice-based crypto

### 4.1 Introduction

#### Định nghĩa 1. Lattice

Xét các vector thuộc  $\mathbb{R}^n$  độc lập tuyến tính  $\{\mathbf{v}_1, \dots, \mathbf{v}_d\}$ . **Lattice** là tập

$$L = \{a_1\mathbf{v}_1 + \dots + a_d\mathbf{v}_d : a_i \in \mathbb{Z}\} \quad (4.1)$$

Tương tự với định nghĩa không gian vector, một **tập sinh** (hay **basis**) là bất cứ tập hợp các vector độc lập tuyến tính mà sinh ra  $L$ .

Hai tập sinh luôn có cùng số phần tử. Khi đó, số vector trong tập sinh được gọi là **số chiều** (hay **dimension**).

Giả sử  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d\}$  là một cơ sở của  $L$ . Tương tự,  $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_d\}$  là một cơ sở khác của  $L$ .

Ta có thể viết mỗi  $\mathbf{w}_i$  là tổ hợp tuyến tính của các vector  $\mathbf{v}$  như sau

$$\mathbf{w}_1 = a_{11}\mathbf{v}_1 + a_{12}\mathbf{v}_2 + \dots + a_{1d}\mathbf{v}_d$$

$$\mathbf{w}_2 = a_{21}\mathbf{v}_1 + a_{22}\mathbf{v}_2 + \dots + a_{2d}\mathbf{v}_d$$

$$\vdots$$

$$\mathbf{w}_d = a_{d1}\mathbf{v}_1 + a_{d2}\mathbf{v}_2 + \dots + a_{dd}\mathbf{v}_d$$

Khi đó, nếu viết các vector  $\mathbf{w}_i$  thành hàng của ma trận  $\mathbf{W}$  và  $\mathbf{v}_j$  thành hàng của ma trận  $\mathbf{V}$  thì biểu diễn trên tương đương với

$$\mathbf{W} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1d} \\ a_{21} & a_{22} & \cdots & a_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ a_{d1} & a_{d2} & \cdots & a_{dd} \end{pmatrix} \cdot \mathbf{V}$$

Đặt

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1d} \\ a_{21} & a_{22} & \cdots & a_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ a_{d1} & a_{d2} & \cdots & a_{dd} \end{pmatrix}$$

Do  $\mathbf{W}$  và  $\mathbf{V}$  là các cơ sở của  $L$  nên nếu các vector  $\mathbf{w}_i$  có thể biểu diễn qua các vector  $\mathbf{v}_j$  thì ngược lại, các vector  $\mathbf{v}_j$  cũng có thể được biểu diễn qua các vector  $\mathbf{w}_i$ .

Suy ra ma trận  $\mathbf{A}$  là ma trận khả nghịch. Do  $a_{ij} \in \mathbb{Z}$  theo định nghĩa lattice, định thức của  $\mathbf{A} \in \mathbb{Z}$ .

Hơn nữa, vì

$$I = \mathbf{A} \cdot \mathbf{A}^{-1} \Rightarrow 1 = \det(\mathbf{A}) \cdot \det(\mathbf{A}^{-1})$$

nên  $\det(\mathbf{A}) = \pm 1$ .

### Định nghĩa 2. Fundamental domain

Cho lattice  $L$  có số chiều là  $d$  với cơ sở gồm các vector  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d\}$ . Ta gọi **fundamental domain** (hay **fundamental parallelepiped**) của  $L$  ứng với cơ sở trên là tập

$$\mathcal{F}(\mathbf{v}_1, \dots, \mathbf{v}_d) = \{t_1\mathbf{v}_1 + \dots + t_d\mathbf{v}_d : 0 \leq t_i < 1\} \quad (4.2)$$

Trong mặt phẳng  $Oxy$  với hai vector  $\mathbf{u}$  và  $\mathbf{v}$  không cùng phương thì fundamental domain là hình bình hành tạo bởi hai vector đó.

### Nhận xét 1

Gọi  $L \subset \mathbb{R}^n$  là lattice với số chiều là  $n$  và gọi  $\mathcal{F}$  là fundamental domain của  $L$ . Khi đó mọi vector  $\mathbf{w} \in \mathbb{R}^n$  đều có thể viết dưới dạng

$$\mathbf{w} = \mathbf{t} + \mathbf{v}$$

với  $\mathbf{t}$  duy nhất thuộc  $\mathcal{F}$  và  $\mathbf{v}$  duy nhất thuộc  $L$ .

Một cách tương đương, hợp của các fundamental domains

$$\mathcal{F} + \mathbf{v} = \{\mathbf{t} + \mathbf{v} : \mathbf{t} \in \mathcal{F}\}$$

với  $\mathbf{v}$  là các vector trong  $L$ , sẽ cover hết  $\mathbb{R}^n$ .

*Chứng minh.* Để chứng minh nhận xét trên, giả sử  $\{\mathbf{v}_i : 1 \leq i \leq n\}$  là cơ sở của  $L$ . Khi đó các  $\mathbf{v}_i$  độc lập tuyến tính nên cũng là cơ sở của  $\mathbb{R}^n$ .

Ta viết các vector  $\mathbf{w} \in \mathbb{R}^n$  dưới dạng tổ hợp tuyến tính của  $\mathbf{v}_i$  và tách hệ số trước mỗi vector thành phần nguyên và phần lẻ. Phần nguyên cho vector trong  $L$  và phần lẻ cho vector trong  $\mathcal{F}$ .

Để chứng minh tính duy nhất của tổ hợp, xét hai cách biểu diễn khác nhau của  $\mathbf{w}$  và chứng minh hai cách đó là một.  $\square$

### Định lí 1. Bất đẳng thức Hadamard

Cho lattice  $L$ , lấy cơ sở bất kỳ của  $L$  là các vector  $\mathbf{v}_1, \dots, \mathbf{v}_n$  và gọi  $\mathcal{F}$  là fundamental domain cho  $L$ . Khi đó

$$\det L = \text{Vol}(\mathcal{F}) \leq \|\mathbf{v}_1\| \cdot \|\mathbf{v}_2\| \cdots \|\mathbf{v}_n\| \quad (4.3)$$

Cơ sở càng gần với trực giao thì bất đẳng thức Hadamard trên càng trở thành đẳng thức.

## Thuật toán Babai

Thuật toán này giúp tìm một cơ sở "đủ tốt" để giải apprCVP.

## 4.2 Thuật toán GGH

Phần này tham khảo trong [2]

Trong thuật toán GGH, ta chọn số nguyên tố  $q$  làm public parameter.

Sau đó chọn hai số  $f$  và  $g$  làm secret key. Hai số này phải thỏa mãn các điều kiện

$$f < \sqrt{q/2}, \quad \sqrt{q/4} < g < \sqrt{q/2}, \quad \gcd(f, qg) = 1$$

Tính  $h = f^{-1}g \pmod{q}$ . Khi đó public key là  $h$ .

**Encryption.** Để encrypt message  $m$  với số random  $r$  thỏa mãn

$$0 < m < \sqrt{q/4}, \quad 0 < r < \sqrt{q/2}$$

Ta tính  $e = rh + m \pmod{q}$  là ciphertext với  $0 < e < q$ .

**Decryption.** Để decrypt ciphertext  $e$  ta tính

$$a = fe \pmod{q}, \quad b = f^{-1}a \pmod{g}$$

Lưu ý  $f^{-1}$  là nghịch đảo modulo  $g$ . Khi đó  $b \equiv m$  là message ban đầu.

*Chứng minh.* Để chứng minh rằng số  $b$  sau khi tính toán bằng chính xác  $m$  ban đầu ta cần xem xét điều kiện của secret key và public key.

Đầu tiên ta có

$$a \equiv fe \equiv f(rh + m) = f(rf^{-1}g + m) = rg + fm \pmod{q}$$

Từ điều kiện của  $f$ ,  $g$ ,  $r$  và  $m$  ta có

$$rg + fm < \sqrt{\frac{q}{2}} \cdot \sqrt{\frac{q}{2}} + \sqrt{\frac{q}{2}} \cdot \sqrt{\frac{q}{4}} < q$$

Nói cách khác  $rg + fm$  giữ nguyên giá trị trong phép modulo  $q$ , hay  $a \equiv rg + fm$ .

Suy ra  $b = f^{-1}a = f^{-1}(rg + fm) = m \pmod{g}$  (giá trị  $a$  không thay đổi khi chuyển từ modulo  $q$  sang modulo  $g$ ). Do  $0 < m < \sqrt{q/4}$  và  $\sqrt{q/4} < g < \sqrt{q/2}$  nên  $m < g$ . Nói cách khác  $b$  bằng đúng  $m$  ban đầu.  $\square$

Để tấn công hệ mật mã này ta xây dựng lattice. Để ý rằng  $h = f^{-1}g \pmod{q}$ , hay  $fh + kq = g$  với  $k \in \mathbb{Z}$ .

Ta thấy rằng  $f \cdot (h, 1) + k \cdot (q, 0) = (g, f)$ . Như vậy lattice gồm hai vector  $(h, 1)$  và  $(q, 0)$ . Thuật toán tối giản Gauss sẽ hoạt động trong trường hợp này (số chiều bằng 2).

# Chương 5

## Quan hệ hai ngôi

### 5.1 Quan hệ hai ngôi

#### Định nghĩa 1. Quan hệ hai ngôi

Xét hai tập hợp  $A$  và  $B$ . Ta gọi  $\mathcal{R}$  là một quan hệ hai ngôi trên  $A$  và  $B$  nếu  $\mathcal{R} \subset A \times B$ . Trong đó  $A \times B$  là tích Descartes của hai tập hợp.

Nếu phần tử  $(a, b) \in \mathcal{R}$  với  $a \in A$  và  $b \in B$  thì ta nói  $a$  **có quan hệ với**  $b$  và ký hiệu  $a\mathcal{R}b$ .

Khi  $A \equiv B$  thì ta nói  $\mathcal{R}$  là quan hệ hai ngôi trên  $A$ . Đây cũng là yếu tố quan trọng cho các khái niệm về sau.

**Ví dụ 1.** Xét hai tập hợp  $A = \{1, 2, 3, 4\}$  và  $B = \{a, b, c\}$ . Khi đó tích Descartes

$$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c), \\ (3, a), (3, b), (3, c), (4, a), (4, b), (4, c)\}$$

Giả sử  $\mathcal{R} = \{(1, a), (3, b), (4, c)\}$  thì 1 quan hệ với  $a$  do  $(1, a) \in \mathcal{R}$ , hay  $1Ra$ . Tuy nhiên 1 không có quan hệ với  $b$  do  $(1, b) \notin \mathcal{R}$ .

Sau đây ta định nghĩa các loại quan hệ hai ngôi.

#### Định nghĩa 2

Cho  $\mathcal{R}$  là quan hệ hai ngôi trên tập  $A$ . Ta nói

1.  $\mathcal{R}$  **phản xạ** (hay **reflexive**) nếu với mọi  $x \in A$  thì  $(x, x) \in \mathcal{R}$ ;
2.  $\mathcal{R}$  **đối xứng** (hay **symmetric**) nếu  $(x, y) \in \mathcal{R}$  thì  $(y, x) \in \mathcal{R}$ ;
3.  $\mathcal{R}$  **phản xứng** (hay **antisymmetric**) nếu  $(x, y) \in \mathcal{R}$  thì  $(y, x) \notin \mathcal{R}$ . Nói cách khác nếu  $(x, y) \in \mathcal{R}$  và  $(y, x) \in \mathcal{R}$  thì  $x = y$ ;
4.  $\mathcal{R}$  **bắc cầu** (hay **transitive**) nếu  $(x, y) \in \mathcal{R}$  và  $(y, z) \in \mathcal{R}$  thì  $(x, z) \in \mathcal{R}$ .

## Quan hệ tương đương

Quan hệ tương đương giúp ta chia (phân hoạch) một tập hợp rời rạc thành các tập con mà chỉ cần một phần tử đại diện cho tập con đó là đủ để tính toán.

### Định nghĩa 3. Quan hệ tương đương

Cho  $\mathcal{R}$  là quan hệ trên tập  $A$ . Khi đó  $\mathcal{R}$  được gọi là **quan hệ tương đương** nếu  $\mathcal{R}$  phản xạ, đối xứng và bắc cầu.

Ta có thể ký hiệu  $x\mathcal{R}y$  với  $\mathcal{R}$  là quan hệ tương đương là  $x \sim y$  hoặc  $x\tilde{\mathcal{R}}y$ .

Tiếp theo ta định nghĩa lớp tương đương chứa phần tử  $x$  và tập thương.

### Định nghĩa 4. Lớp tương đương

Cho  $\mathcal{R}$  là quan hệ tương đương trên tập  $A$ . Khi đó với  $x \in A$ , ta định nghĩa lớp tương đương chứa phần tử  $x$  là tập các phần tử của  $A$  có quan hệ với  $x$

$$\bar{x} = \{y \in A, y\mathcal{R}x\}$$

### Định nghĩa 5. Tập thương

Tập hợp các lớp tương đương như trên tạo thành tập thương.

$$A/\mathcal{R} = \{\bar{x}, x \in A\}$$

**Ví dụ 2.** Xét số nguyên dương  $n$ . Với số nguyên  $x$  và  $y$ , ta nói  $x$  có quan hệ với  $y$  nếu  $n|(x - y)$ , hay  $x \equiv y \pmod{n}$ . Ta ký hiệu quan hệ này là  $n\mathbb{Z}$ .

Quan hệ trên là quan hệ tương đương vì

1.  $n|0 = x - x$  với mọi  $x \in \mathbb{Z}$  nên có tính phản xạ;
2.  $n|(x - y) \Rightarrow n|-(x - y) = y - x$  với mọi  $x, y \in \mathbb{Z}$  nên có tính đối xứng;
3.  $n|(x - y)$  và  $n|(y - z)$  thì  $n|(x - y + y - z) = (x - z)$  nên có tính bắc cầu.

Từ đó ta có thể phân tập  $\mathbb{Z}$  thành các lớp tương đương

$$\begin{aligned}\bar{0} &= \{\dots, -2n, -n, 0, n, 2n, \dots\} \\ \bar{1} &= \{\dots, -2n + 1, -n + 1, 1, n + 1, 2n + 1, \dots\} \\ &\vdots \\ \overline{n-1} &= \{\dots, -n - 1, -1, n - 1, 2n - 1, 3n - 1, \dots\}\end{aligned}$$

Tập thương của chúng ta là  $\mathbb{Z}/n\mathbb{Z} = \{\bar{0}, \bar{1}, \dots, \overline{n-1}\}$ .

## Quan hệ thứ tự

### Định nghĩa 6. Quan hệ thứ tự

Cho quan hệ  $\mathcal{R}$  trên tập  $A$ . Ta nói  $\mathcal{R}$  là quan hệ thứ tự nếu  $\mathcal{R}$  phản xạ, phản xứng và bắc cầu.

### Định nghĩa 7

Cho tập hợp  $A$  và quan hệ  $\mathcal{R}$  trên  $A$  là quan hệ thứ tự. Nếu  $x\mathcal{R}y$  thì ta ký hiệu  $x \prec y$ . Khi đó  $(A, \prec)$  được gọi là **tập có thứ tự**.

Tiếp theo là một số định nghĩa quan trọng về tập hợp có thứ tự.

### Định nghĩa 8

Với  $(A, \prec)$  và  $x, y \in A$ ,

1. Nếu  $x \prec y$ , ta nói  $y$  là **trội của  $x$** , hay là  $x$  **được trội bởi  $y$** ;
2.  $y$  là **trội trực tiếp** của  $x$  nếu không tồn tại  $z$  sao cho  $x \prec z$  và  $z \prec y$ .

### Định nghĩa 9

Xét  $(A, \prec)$ .

1.  $x$  và  $y$  thuộc  $A$  được gọi là **so sánh được** nếu  $x \prec y$  hoặc  $y \prec x$ ;
2. nếu với mọi  $x, y \in A$ ,  $x$  và  $y$  so sánh được thì  $(A, \prec)$  được gọi là **quan hệ thứ tự toàn phần**. Ngược lại thì gọi là **quan hệ thứ tự bán phần**.

Để biểu diễn sự so sánh trong một tập hợp, ta sử dụng biểu đồ Hasse.

### Định nghĩa 10

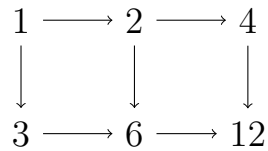
Biểu đồ Hasse của  $(A, \prec)$  với  $A$  là tập hữu hạn bao gồm

1. Tập điểm - mỗi điểm biểu diễn một phần tử của  $A$ ;
2. Tập cung - vẽ một cung từ  $x$  tới  $y$  nếu  $y$  là trội trực tiếp của  $x$ .

**Ví dụ 3.** Xét tập  $U_{12} = \{1, 2, 3, 4, 6, 12\}$  với quan hệ  $x\mathcal{R}y$  được định nghĩa  $x$  là ước của  $y$ .

Theo đó, biểu đồ Hasse của quan hệ trên là hình 5.1.



Hình 5.1. Biểu đồ Hasse của  $U_{12}$ **Định nghĩa 11**

Xét quan hệ thứ tự  $(A, \prec)$ .

1. Phần tử  $M \in A$  được gọi là
  - (a) **Tối đại** nếu  $M \prec x$  thì  $x = M$ ;
  - (b) **Cực đại** (hay **lớn nhất**) nếu với mọi  $x \in A$  thì  $x \prec M$ .
2. Phần tử  $m \in A$  được gọi là
  - (a) **Tối tiểu** nếu  $x \prec m$  thì  $x = m$ ;
  - (b) **Cực tiểu** (hay **nhỏ nhất**) nếu với mọi  $x \in A$  thì  $m \prec x$ .

**Nhận xét 1**

1. Phần tử cực đại nếu có là duy nhất. Tương tự cho cực tiểu;
2. Nếu  $n$  là phần tử tối đại duy nhất thì nó cũng là cực đại. Tương tự cho tối tiểu.

Trong ví dụ  $U_{12}$  ở trên thì 1 là tối tiểu và cũng là cực tiểu, và 12 là tối đại và cũng là cực đại.

# Chương 6

## Bài toán đếm Polya

### 6.1 Bổ đề Burnside

#### Lớp tương đương

Xét nhóm  $G$  và tập hợp  $M$ . Khi đó hai phần tử  $m$  và  $n$  thuộc  $M$  được gọi là **quan hệ với nhau** nếu tồn tại  $g \in G$  mà  $m = gn$ .

#### Nhận xét 1

Quan hệ giữa các phần tử như trên là quan hệ tương đương.

*Chứng minh.* Để chứng minh một quan hệ là tương đương, ta cần chứng minh tính phản xạ, đối xứng và bắc cầu.

Đối với tính phản xạ, mọi vector đều có quan hệ với chính nó qua phần tử đơn vị  $e \in G$ .

Đối với tính đối xứng, nếu  $m$  có quan hệ với  $n$  thì tồn tại  $g \in G$  sao cho  $m = gn$ . Theo tính chất nhóm thì tồn tại phần tử  $g^{-1}$  là nghịch đảo của  $g$  trong  $G$ . Do đó  $g^{-1}m = n$ . Nói cách khác  $n$  cũng có quan hệ với  $m$ . Như vậy ta có tính đối xứng.

Đối với tính bắc cầu, nếu  $m$  có quan hệ với  $n$  thì tồn tại  $g_1 \in G$  sao cho  $m = g_1n$ . Tiếp theo, nếu  $n$  có quan hệ với  $p$  thì tồn tại  $g_2 \in G$  sao cho  $n = g_2p$ . Suy ra  $m = g_1n = g_1(g_2p) = (g_1g_2)p$ . Do  $g_1, g_2 \in G$  nên  $g_1g_2 \in G$ . Như vậy  $m$  cũng có quan hệ với  $p$  nên quan hệ có tính bắc cầu.

Vậy quan hệ được định nghĩa như trên là quan hệ tương đương.  $\square$

#### Tác động nhóm lên vector

Xét nhóm  $G$  và không gian vector  $\mathbb{F}_2^n$ ,  $n \in \mathbb{N}$ . Khi đó hai vector  $\mathbf{x}$  và  $\mathbf{y}$  thuộc  $\mathbb{F}_2^n$  được gọi là **quan hệ với nhau** nếu tồn tại  $g \in G$  mà  $\mathbf{x} = g\mathbf{y}$ .

Ví dụ, xét nhóm hoán vị  $\mathcal{S}_3$ . Giả sử các vector trong  $\mathbb{F}_2^3$  có dạng

$$\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{F}_2^3.$$

Khi đó vector  $(1, 0, 0)$  có quan hệ với  $(0, 0, 1)$  với hoán vị  $(1, 3)(2)$ . Cụ thể là  $(x_1, x_2, x_3) \xrightarrow{(1,3)(2)} (x_3, x_2, x_1)$ .

Tương tự, vector  $(1, 0, 0)$  cũng có quan hệ với  $(0, 1, 0)$  với hoán vị  $(1, 2)(3)$ . Thêm nữa, vector  $(1, 0, 0)$  có quan hệ với chính nó qua hoán vị đồng nhất  $(1)(2)(3)$ .

Trong môn toán rời rạc ta đã biết, nếu một tập có quan hệ tương đương thì ta có thể phân các phần tử của tập đó vào các lớp tương đương rời nhau. Nghĩa là nếu hai phần tử có quan hệ với nhau thì vào cùng một lớp tương đương. Từ phần trên ta đã biết rằng dưới tác động của nhóm, các phần tử trong tập hợp bất kì sẽ phân bổ thành các lớp tương đương.

Câu hỏi đặt ra là, có bao nhiêu lớp tương đương như vậy?

Để giải quyết vấn đề này ta sử dụng bổ đề Burnside.

Nhóm  $\mathcal{S}_3$  có các hoán vị

$$\mathcal{S}_3 = \{(1)(2)(3), (1, 2)(3), (1, 3)(2), (2, 3)(1), (1, 3, 2), (1, 2, 3)\}$$

Lần lượt xét từng hoán vị. Đầu tiên, với  $(1)(2)(3)$  thì các phần tử trong vector đứng yên. Do đó dưới tác động của hoán vị này,  $x_1$  biến thành  $x_1$ ,  $x_2$  biến thành  $x_2$  và  $x_3$  biến thành  $x_3$ . Số cách chọn cho mỗi  $x_i$  là 2 nên theo quy tắc nhân ta có  $2^3 = 8$  cách.

Tiếp theo, với hoán vị  $(1, 2)(3)$  thì  $x_1 \rightarrow x_2$ ,  $x_2 \rightarrow x_1$  và  $x_3 \rightarrow x_3$ . Do đó  $x_1$  và  $x_2$  có cùng giá trị nên có 2 cách chọn,  $x_3$  cũng có 2 cách chọn nên tổng số cách là  $2 \cdot 2 = 4$ . Hoán vị  $(1, 3)(2)$  và  $(2, 3)(1)$  tương tự.

Với hoán vị  $(1, 2, 3)$  thì  $x_1 \rightarrow x_2$ ,  $x_2 \rightarrow x_3$  và  $x_3 \rightarrow x_1$  nên  $x_1 = x_2 = x_3$ , có 2 cách chọn trong trường hợp này. Hoán vị  $(1, 3, 2)$  tương tự.

Như vậy, theo bổ đề Burnside, số lớp tương đương các vector trong  $\mathbb{F}_2^3$  là

$$t(\mathcal{S}_3) = \frac{1}{6}(1 \cdot 2^3 + 3 \cdot 2^2 + 2 \cdot 2) = 4$$

Thật vậy, ta có thể chia các vector thành 4 lớp tương đương là  $\{000\}$ ,  $\{001, 010, 011\}$ ,  $\{011, 101, 110\}$ ,  $\{111\}$ .

Ngoài nhóm  $\mathcal{S}_3$  ra còn các nhóm khác cũng tác động lên các vector. Một số nhóm hay được sử dụng là:

1. Nhóm general linear: gồm các ma trận khả nghịch  $n \times n$  trên  $\mathbb{F}_2$ . Tác động nhóm lúc này là phép nhân ma trận  $\mathbf{A} \in GL(n, 2)$  với vector  $\mathbf{x} \in \mathbb{F}_2^n$ , hay  $\mathbf{A} \cdot \mathbf{x}$ .
2. Nhóm general affine: gồm các ma trận khả nghịch  $n \times n$  trên  $\mathbb{F}_2$  và vector bất kì trong  $\mathbb{F}_2^n$ . Tác động nhóm lúc này là biến đổi affine  $\mathbf{A} \cdot \mathbf{x} + \mathbf{b}$  với  $\mathbf{A} \in GL(n, 2)$  và  $\mathbf{b} \in \mathbb{F}_2^n$ .

## Tác động nhóm lên hàm boolean

Ta tiếp tục xét nhóm  $G$  và không gian vector  $\mathbb{F}_2^n$ ,  $n \in \mathbb{N}$ . Khi đó hai hàm boolean  $n$  biến  $f(x_1, \dots, x_n)$  và  $g(x_1, \dots, x_n)$  được gọi là **quan hệ với nhau** nếu tồn tại  $g \in G$  mà  $g(\mathbf{x}) = f(g\mathbf{x})$  với mọi  $\mathbf{x} \in \mathbb{F}_2^n$ .

Ta cũng xét hoán vị  $\mathcal{S}_3$ . Ta cũng lần lượt xét các phần tử của nhóm.

Đặt  $f_0, f_1, \dots, f_7$  lần lượt là các giá trị hàm  $f$  với các vector  $\mathbf{x} \in \mathbb{F}_2^3$ .

Đầu tiên, với  $(1)(2)(3)$ , ta có bảng chuyển vector như hình 6.1.

$x_1$	$x_2$	$x_3$	$f$	(1)(2)(3) →	$x_1$	$x_2$	$x_3$	$f$
0	0	0	$f_0$		0	0	0	$f_0$
0	0	1	$f_1$		0	0	1	$f_1$
0	1	0	$f_2$		0	1	0	$f_2$
0	1	1	$f_3$		0	1	1	$f_3$
1	0	0	$f_4$		1	0	0	$f_4$
1	0	1	$f_5$		1	0	1	$f_5$
1	1	0	$f_6$		1	1	0	$f_6$
1	1	1	$f_7$		1	1	1	$f_7$

Hình 6.1. Hoán vị  $(1)(2)(3)$

Ta thấy rằng  $f_0 \rightarrow f_0, f_1 \rightarrow f_1, \dots, f_7 \rightarrow f_7$  nên có 8 chu trình. Vậy số lượng cách chọn là  $2^8$ .

Tiếp theo, xét các hoán vị dạng  $(1)(2, 3)$ , ta có bảng chuyển vector như hình 6.2.

$x_1$	$x_2$	$x_3$	$f$	(1)(2, 3) →	$x_1$	$x_3$	$x_2$	$f$
0	0	0	$f_0$		0	0	0	$f_0$
0	0	1	$f_1$		0	1	0	$f_2$
0	1	0	$f_2$		0	0	1	$f_1$
0	1	1	$f_3$		0	1	1	$f_3$
1	0	0	$f_4$		1	0	0	$f_4$
1	0	1	$f_5$		1	1	0	$f_6$
1	1	0	$f_6$		1	0	1	$f_5$
1	1	1	$f_7$		1	1	1	$f_7$

Hình 6.2. Hoán vị  $(1)(2, 3)$

Ta thấy rằng  $f_0 \rightarrow f_0, f_1 \rightarrow f_2 \rightarrow f_1, f_3 \rightarrow f_3, f_4 \rightarrow f_4, f_5 \rightarrow f_6 \rightarrow f_5, f_7 \rightarrow f_7$ . Ở đây có 6 chu trình nên số cách chọn là  $2^6$ .

Tiếp theo ta xét các hoán vị dạng  $(1, 2, 3)$  (hình 6.3).

$x_1$	$x_2$	$x_3$	$f$	$(1, 2, 3)$	$x_2$	$x_3$	$x_1$	$f$
0	0	0	$f_0$		0	0	0	$f_0$
0	0	1	$f_1$		0	1	0	$f_2$
0	1	0	$f_2$		1	0	0	$f_4$
0	1	1	$f_3$		1	1	0	$f_6$
1	0	0	$f_4$		0	0	1	$f_1$
1	0	1	$f_5$		0	1	1	$f_3$
1	1	0	$f_6$		1	0	1	$f_5$
1	1	1	$f_7$		1	1	1	$f_7$

Hình 6.3. Hoán vị  $(1, 2, 3)$

Ta thấy rằng  $f_0 \rightarrow f_0, f_1 \rightarrow f_2 \rightarrow f_4 \rightarrow f_1, f_3 \rightarrow f_6 \rightarrow f_5 \rightarrow f_3, f_7 \rightarrow f_7$  nên ở đây có 4 chu trình. Số cách chọn là  $2^4$ .

Như vậy theo bổ đề Burnside, số lớp hàm bool tương đương dưới tác động của nhóm  $\mathcal{S}_3$  là

$$t(\mathcal{S}_3) = \frac{1}{6}(2^8 + 3 \cdot 2^6 + 2 \cdot 2^4) = 80.$$

# Chương 7

## Giải tích

### 7.1 Giới hạn

#### Giới hạn của dãy số

##### Định nghĩa 1. Giới hạn hữu hạn của dãy số

Cho dãy số  $\{a_n\}$ . Ta nói dãy  $\{a_n\}$  có giới hạn hữu hạn  $L$  nếu với mọi  $\varepsilon > 0$ , tồn tại  $n_0 \in \mathbb{N}$  sao cho với mọi  $n \geq n_0$  thì

$$|a_n - L| < \varepsilon$$

Ký hiệu:  $\lim_{n \rightarrow \infty} a_n = L$ .

**Ví dụ 1.** Xét dãy số cho bởi công thức  $a_n = \frac{1}{n}$ . Ta chứng minh dãy số có giới hạn hữu hạn là 0.

Với mọi  $\varepsilon > 0$  tùy ý, ta cần chứng minh tồn tại số  $n_0 \geq 1$  sao cho với mọi  $n \geq n_0$  thì  $|a_n - 0| < \varepsilon$ .

Nói cách khác  $a_{n_0} < \varepsilon$ , hay tương đương với  $\frac{1}{n_0} < \varepsilon \Leftrightarrow n_0 > \frac{1}{\varepsilon}$ .

Vậy ta chỉ cần chọn  $n_0$  thỏa bất đẳng thức trên (luôn tìm được).

Kết luận:  $\lim_{n \rightarrow \infty} a_n = 0$

##### Định nghĩa 2. Dãy số có giới hạn vô cực

Cho dãy số  $\{a_n\}$ . Ta nói dãy số có giới hạn ở dương vô cực nếu với mọi  $M > 0$ , tồn tại  $n_0 \in \mathbb{N}$  sao cho với mọi  $n \geq n_0$  thì  $a_n > M$ .

Nói cách khác, nếu ta chọn một số  $M$  rất lớn bất kì, thì mọi số hạng của

dãy số kể từ một số hạng nào đó trở đi luôn lớn hơn  $M$ . Định nghĩa về dãy số có giới hạn ở âm vô cực cũng tương tự.

## Giới hạn của hàm số

Để định nghĩa giới hạn của hàm số  $y = f(x)$  khi  $x$  tiến tới  $x_0$  ta có hai loại định nghĩa.

### Định nghĩa 3. Giới hạn hàm số qua giới hạn dãy số

Xét hàm số  $f(x)$ . Ta nói hàm số có giới hạn hữu hạn  $L$  khi  $x$  tiến tới  $x_0$ , nếu với mọi dãy số  $\{x_n\}$  mà  $\lim_{n \rightarrow \infty} x_n = x_0$ , thì  $\lim_{n \rightarrow \infty} f(x_n) = L$ .

Định nghĩa này tuân theo giới hạn của dãy số. Khi đó mọi phần tử của dãy số từ một số hạng nào đó trở đi cho giá trị  $f(x_n)$  tiến về  $L$ .

Định nghĩa của hàm số theo kiểu Cauchy (hay còn được gọi là ngôn ngữ  $\delta - \varepsilon$ ) là kiểu định nghĩa phổ biến được giảng dạy trong nhà trường.

### Định nghĩa 4. Giới hạn hàm số kiểu Cauchy

Xét hàm số  $f(x)$ . Ta nói hàm số có giới hạn hữu hạn  $L$  khi  $x$  tiến tới  $x_0$ , nếu với mọi  $\varepsilon > 0$ , tồn tại  $\delta > 0$  sao cho với mọi  $x$  mà  $|x - x_0| < \delta$  thì  $|f(x) - L| < \varepsilon$ .

Ký hiệu:  $\lim_{x \rightarrow x_0} f(x) = L$

Ta có thể thấy ở đây  $x$  tiến về  $x_0$  (khá giống định nghĩa giới hạn hàm số) và  $f(x)$  tương ứng tiến về  $L$ .

Tương tự ta cũng có giới hạn hàm số ở vô cực.

### Định nghĩa 5. Giới hạn hàm số ở vô cực

Với hàm số  $f(x)$ , ta nói hàm số có giới hạn tại dương vô cực khi  $x$  tiến về  $x_0$  nếu với mọi  $M > 0$ , tồn tại  $\delta > 0$  sao cho với mọi  $x$  mà  $|x - x_0| < \delta$  thì  $f(x) > M$ .

Ký hiệu:  $\lim_{x \rightarrow x_0} f(x) = +\infty$ .

**Định nghĩa 6. Giới hạn một bên**

Ta nói hàm số  $f(x)$  có giới hạn phải  $L$  tại  $x_0$  khi  $x$  tiến về bên phải  $x_0$  nếu với mọi  $\varepsilon > 0$ , tồn tại  $\delta > 0$  sao cho với mọi  $0 < x - x_0 < \delta$  thì  $|f(x) - L| < \varepsilon$ .

Ký hiệu:  $\lim_{x \rightarrow x_0^+} f(x) = L$

Nghĩa là chúng ta chỉ xét giới hạn khi  $x$  tiến tới  $x_0$  từ bên phải  $x > x_0$ . Tương tự cho giới hạn trái.

Lưu ý rằng trong nhiều trường hợp, mặc dù cùng tiến tới  $x_0$  nhưng giới hạn trái và giới hạn phải có thể không bằng nhau.

**Ví dụ 2.** Xét hàm số  $y = \frac{1}{x}$ . Ta thấy hàm số không xác định tại  $x = 0$ , và giới hạn trái và phải khác nhau do

$$\lim_{x \rightarrow 0^+} = +\infty, \quad \lim_{x \rightarrow 0^-} = -\infty$$

**Tính liên tục của hàm số**

Cho hàm số  $f(x)$  xác định trên miền  $D$  và  $x_0$  là một điểm thuộc  $D$ .

**Định nghĩa 7. Hàm số liên tục tại một điểm**

Ta nói hàm số  $f(x)$  liên tục tại  $x_0$  nếu

$$\lim_{x \rightarrow x_0} f(x) = f(x_0)$$

Định nghĩa tương tự cho liên tục trái và liên tục phải (ta lấy giới hạn một bên).

Như vậy, có 3 khả năng hàm số không liên tục tại một điểm.

1. Hàm số không xác định tại  $x_0$
2. Hàm số xác định tại  $x_0$  nhưng giới hạn tại đó không bằng  $f(x_0)$
3. Giới hạn trái và giới hạn phải không bằng nhau

Nếu hàm số không liên tục tại  $x_0$  ta gọi hàm số bị **gián đoạn** tại  $x_0$ .

Nếu hàm số liên tục tại mọi điểm trên khoảng  $(a, b)$  thì ta nói hàm số liên tục trên khoảng đó.

**7.2 Đạo hàm**



**Định nghĩa 8. Đạo hàm**

Cho hàm số  $f(x)$  xác định trên miền  $D$  và  $x_0$  là điểm thuộc  $D$ . Ta nói hàm số  $f(x)$  có đạo hàm tại  $x_0$  (hoặc khả vi tại  $x_0$ ) nếu tồn tại giới hạn hữu hạn

$$\lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}$$

Ký hiệu đạo hàm của  $f$  tại  $x_0$  là  $f'(x_0)$ .

**Ví dụ 3.** Xét hàm số  $f(x) = x^2 + 1$  trên  $\mathbb{R}$ . Tìm đạo hàm tại  $x_0 \in \mathbb{R}$ .

Ta có  $f(x) - f(x_0) = x^2 + 1 - (x_0^2 + 1) = (x - x_0)(x + x_0)$ .

Khi đó  $\frac{f(x) - f(x_0)}{x - x_0} = x + x_0$  nên ta có  $\lim_{x \rightarrow x_0} (x + x_0) = 2x_0$ .

Nếu hàm số khả vi trên mọi điểm thuộc khoảng (đoạn) nào đó thì ta nói hàm số khả vi trên khoảng (đoạn) đó và ký hiệu là  $f'(x)$ .

Với ví dụ trên, ta thấy giới hạn tồn tại với mọi  $x_0 \in \mathbb{R}$  nên ta có thể thay  $x_0$  bởi  $x$  và có  $f'(x) = 2x$  với  $f(x) = x^2 + 1$ .

**Nhận xét 1**

Từ định nghĩa ta thấy rằng nếu  $f(x)$  khả vi tại  $x_0$  thì nó cũng liên tục tại  $x_0$ . Tuy nhiên chiều ngược lại không đúng. Ví dụ với hàm số  $y = |x|$ , hàm số liên tục tại  $x = 0$  nhưng giới hạn (đạo hàm) phải là 1, còn giới hạn (đạo hàm) trái là -1.

Về mặt hình ảnh, khi hàm số khả vi tại một điểm thì đồ thị sẽ "trơn", không gấp khúc tại điểm đó.

**Cực trị**

Đầu tiên chúng ta cần một định lý về tính đơn điệu của hàm số khả vi.

**Định lý 1**

Xét hàm số  $f(x)$  khả vi trên khoảng  $(a, b)$ . Nếu  $f'(x) > 0$  với mọi  $x \in (a, b)$  thì  $f(x)$  đồng biến trên  $(a, b)$ .

Tương tự,  $f'(x) < 0$  với mọi  $x \in (a, b)$  thì  $f(x)$  nghịch biến trên  $(a, b)$ .

**Định nghĩa 9. Cực tiểu của hàm số**

Xét hàm số  $f(x)$  liên tục trên khoảng  $(a, b)$ . Điểm  $(x_0, f(x_0))$  được gọi là **cực tiểu** của hàm số  $f(x)$  nếu tồn tại một lân cận  $U$  chứa  $x_0$  nằm trong khoảng  $(a, b)$  sao cho với mọi  $x \in U$  thì  $f(x) \geq f(x_0)$ .

**Định nghĩa 10. Cực đại của hàm số**

Xét hàm số  $f(x)$  liên tục trên khoảng  $(a, b)$ . Điểm  $(x_0, f(x_0))$  được gọi là **cực đại** của hàm số  $f(x)$  nếu tồn tại một lân cận  $U$  chứa  $x_0$  nằm trong khoảng  $(a, b)$  sao cho với mọi  $x \in U$  thì  $f(x) \leq f(x_0)$ .

Theo định nghĩa cực tiểu thì chỉ cần tồn tại lân cận chứa  $x_0$  mà  $f(x) \geq f(x_0)$  thì điểm đó là cực tiểu. Như vậy một hàm số có thể có nhiều cực tiểu, tương tự cũng có thể có nhiều cực đại.

Lưu ý rằng cực đại và cực tiểu không phải điểm chỉ giá trị lớn nhất hay giá trị nhỏ nhất của hàm số. Nó chỉ lớn nhất hoặc nhỏ nhất trong vùng lân cận đó theo định nghĩa, nên người ta còn gọi là cực trị địa phương.

**Định nghĩa 11. Dãy Cauchy**

Dãy  $(x_n)$  được gọi là dãy Cauchy nếu với mọi  $\varepsilon > 0$ , tồn tại  $N_0 \in \mathbb{N}$  sao cho, với mọi  $m, n > N_0$  thì  $|x_m - x_n| < \varepsilon$ .

**Định lí 2. Tiêu chuẩn Cauchy**

Dãy số  $(x_n)$  có giới hạn hữu hạn khi và chỉ khi nó là dãy Cauchy.

**Định lí 3. Bổ đề Fermat**

Cho  $f$  là một hàm số có đạo hàm trên  $(a, b)$ . Nếu  $x_0 \in (a, b)$  là một điểm cực trị của  $f$  thì ta có  $f'(x_0) = 0$ .

*Chứng minh.* Ta chứng minh trong trường hợp  $x_0$  là điểm cực tiểu. Trường hợp điểm cực đại tương tự.

Hàm  $f$  có đạo hàm trên  $(a, b)$  nên tại điểm  $x_0$  nó có đạo hàm bên trái và đạo hàm bên phải, và hai đạo hàm này bằng nhau.

Ta có  $f'(x_0^+) = \lim_{x \rightarrow x_0^+} \frac{f(x) - f(x_0)}{x - x_0}$ . Vì  $x \rightarrow x_0^+$  nghĩa là  $x > x_0$  ( $x$  tiến tới  $x_0$  từ bên phải), và do  $x_0$  là cực tiểu  $f(x) - f(x_0) \geq 0$  nên phân số dưới dấu giới hạn lớn hơn 0. Suy ra  $f'(x_0^+) \geq 0$ .

Hoàn toàn tương tự ta chứng minh được  $f'(x_0^-) \leq 0$ . Và do  $f'(x_0^+) = f'(x_0^-) = f'(x_0)$  nên  $f'(x_0) = 0$ .

Ta có điều phải chứng minh. □

#### Định lí 4. Định lí Rolle

Xét hàm số  $f$  liên tục trên đoạn  $[a, b]$ , có đạo hàm trên khoảng  $(a, b)$  và  $f(a) = f(b)$ . Khi đó tồn tại  $c$  thuộc  $(a, b)$  sao cho  $f'(c) = 0$ .

#### Định lí 5. Định lí Lagrange

Xét hàm số  $f$  liên tục trên đoạn  $[a, b]$ , có đạo hàm trên khoảng  $(a, b)$ . Khi đó tồn tại  $c$  thuộc  $(a, b)$  sao cho  $f'(c)(b - a) = f(b) - f(a)$ .

#### Định nghĩa 12. Hàm lõm

Hàm số  $f$  liên tục trên khoảng  $\mathbb{I}$  nếu với mọi  $\alpha, \beta$  mà  $\alpha + \beta = 1$  ta đều có

$$f(\alpha x + \beta y) \leq \alpha f(x) + \beta f(y), \quad \forall x, y \in \mathbb{I} \quad (7.1)$$

# Chương 8

## Quantum computing

### 8.1 Qubit và toán tử quantum

Trên máy tính hiện nay, đơn vị xử lý cơ bản là bit (0 hoặc 1). Trong máy tính lượng tử, đơn vị tính toán là qubit (quantum bit).

#### Qubit

Mỗi qubit  $|\psi\rangle$  được biểu diễn dưới dạng tổ hợp tuyến tính của cơ sở gồm  $|0\rangle = (1, 0)$  và  $|1\rangle = (0, 1)$ . Khi đó qubit  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ . Ở đây  $\alpha, \beta \in \mathbb{C}$  (tập số phức).

Tích của  $n$  qubit là các tổ hợp  $|0, 0, \dots, 0\rangle, |0, 0, \dots, 1\rangle, \dots, |1, 1, \dots, 1\rangle$ . Ta cũng ký hiệu  $|0\rangle \otimes |1\rangle = |01\rangle$ .

**Ví dụ 1.** Nếu  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  và  $|\Psi\rangle = \gamma|0\rangle + \delta|1\rangle$  thì

$$|\psi\rangle \otimes |\Psi\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$$

Tiếp theo là **toán tử quantum** và tương ứng với nó là các cổng (gate) trên mạch.

Toán tử quantum tác động lên một qubit hoặc tích của nhiều qubit.

Qubit có dạng  $|\psi\rangle = a|0\rangle + b|1\rangle$ . Ta có thể viết hệ số dưới dạng vector cột  $\begin{pmatrix} a \\ b \end{pmatrix}$ . Khi đó, toán tử quantum sẽ là một ma trận  $2 \times 2$  biến đổi vector trên thành vector mới  $\begin{pmatrix} c \\ d \end{pmatrix}$  tương ứng với qubit  $|\Psi\rangle = c|0\rangle + d|1\rangle$ .

Nói cách khác, đặt toán tử quantum là ma trận  $\mathcal{U} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$  thì ta có

$$|\psi\rangle \rightarrow |\Psi\rangle = \mathcal{U}|\psi\rangle, \quad \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}$$

## Các toán tử quantum thường gặp

1. **Toán tử đồng nhất.** Toán tử đồng nhất identity giữ nguyên qubit đầu vào. Ma trận tương ứng là ma trận đơn vị  $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ .

2. **Toán tử NOT.** Toán tử NOT có ma trận tương ứng là  $\text{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ . Khi đó  $\text{NOT}|\psi\rangle = b|0\rangle + a|1\rangle$  với  $x \in \{0, 1\}$ .

Khi qubit là  $|0\rangle$  hoặc  $|1\rangle$ , tác dụng của toán tử NOT là phép XOR nên ta có  $\text{NOT}|x\rangle = |x \oplus 1\rangle$ .

3. **Toán tử Hadamard.** Đây là một toán tử đặc biệt và được quan tâm nhiều.

Ma trận của toán tử Hadamard là  $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ .

**Ví dụ 2.** Xét qubit  $|\psi\rangle = a|0\rangle + b|1\rangle$ , toán tử Hadamard tương ứng với phép nhân ma trận

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}}(a+b) \\ \frac{1}{\sqrt{2}}(a-b) \end{pmatrix}$$

Ta chuyển cột kết quả về lại dạng tổ hợp tuyến tính thì cổng Hadamard hoạt động trên qubit  $|\psi\rangle = a|0\rangle + b|1\rangle$  cho kết quả là

$$H|\psi\rangle = H(a|0\rangle + b|1\rangle) = \frac{1}{\sqrt{2}}(a+b)|0\rangle + \frac{1}{\sqrt{2}}(a-b)|1\rangle$$

Nếu  $|\psi\rangle \equiv |0\rangle$  thì tương đương với  $a = 1, b = 0$ . Ta có  $H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ .

Nếu  $|\psi\rangle \equiv |1\rangle$  thì tương đương với  $a = 0, b = 1$ . Ta có  $H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$ .

Tổng quát ta nhận thấy, với  $x \in \{0, 1\}$  thì  $H|x\rangle = \frac{|0\rangle + (-1)^x|1\rangle}{\sqrt{2}}$ .

Ta thấy rằng toán tử ngược của toán tử Hadamard là chính nó.

4. **Toán tử control.** Đây là toán tử thường được dùng nhất khi tính toán trên tích của nhiều qubit.

Như đã xem xét ở trên, tích của  $n$  qubit sẽ có  $2^n$  phần tử tương ứng các bộ  $|0, 0, \dots, 0, 0\rangle, |0, 0, \dots, 0, 1\rangle, \dots$ . Do đó toán tử control sẽ là ma trận kích thước  $2^n \times 2^n$ .

Gọi  $\mathcal{U} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$  là toán tử tác động lên một qubit (ví dụ như 3 toán tử

đã đề cập). Xét hai qubit là  $|x\rangle = a|0\rangle + b|1\rangle$  và  $|y\rangle = c|0\rangle + d|1\rangle$ . Từ phía trên

$$|x\rangle \otimes |y\rangle = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

Khi đó toán tử control có dạng ma trận là

$$\mathcal{U} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c_{11} & c_{12} \\ 0 & 0 & c_{21} & c_{22} \end{pmatrix}$$

Hay viết dưới dạng ma trận khối là  $\mathcal{U} = \begin{pmatrix} I & \mathcal{O} \\ \mathcal{O} & \mathcal{U} \end{pmatrix}$ .

Ta cũng viết tích  $|x\rangle \otimes |y\rangle$  dưới dạng vector cột (4 phần tử). Khi đó

$$\mathcal{U}(|x\rangle \otimes |y\rangle) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c_{11} & c_{12} \\ 0 & 0 & c_{21} & c_{22} \end{pmatrix} \cdot \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ c_{11} \cdot bc + c_{12} \cdot bd \\ c_{21} \cdot bc + c_{22} \cdot bd \end{pmatrix}$$

Hai phần tử đầu của vector kết quả không thay đổi, còn phần sau có "một phần" là  $\mathcal{U}|y\rangle$ . Khi viết lại kết quả dưới dạng qubit thì

$$ac|00\rangle + ad|01\rangle + (c_{11} \cdot bc + c_{12} \cdot bd)|10\rangle + (c_{21} \cdot bc + c_{22} \cdot bd)|11\rangle$$

Ta có một số nhận xét sau đây.

Nếu  $|x\rangle \equiv |0\rangle$ , tức là  $a = 1, b = 0$  thì tích trên tương ứng với  $c|00\rangle + d|01\rangle + 0|10\rangle + 0|11\rangle = |0\rangle \otimes (c|0\rangle + d|1\rangle) = |x\rangle \otimes |y\rangle$ .

Nếu  $|x\rangle \equiv |1\rangle$ , tức là  $a = 0, b = 1$  thì tích trên tương ứng với  $0|00\rangle + 0|01\rangle + (c_{11}c + c_{12}d)|10\rangle + (c_{21}c + c_{22}d)|11\rangle = |1\rangle \otimes ((c_{11}c + c_{12}d)|0\rangle + (c_{21}c + c_{22}d)|1\rangle) = |1\rangle \otimes \mathcal{U}|y\rangle = |x\rangle \otimes \mathcal{U}|y\rangle$ .

Tổng kết lại, với  $x \in \{0, 1\}$  thì

- nếu  $x = 0$  thì  $|x\rangle \otimes |y\rangle \rightarrow |x\rangle \otimes |y\rangle$ .
- nếu  $x = 1$  thì  $|x\rangle \otimes |y\rangle \rightarrow |x\rangle \otimes \mathcal{U}|y\rangle$ .

Tùy vào  $x$  là 0 hay 1 mà toán tử quantum  $\mathcal{U}$  sẽ bị bỏ qua hoặc xem xét. Ở đây qubit  $|x\rangle$  đóng vai trò điều khiển nên đây được gọi là toán tử control.

**5. Toán tử control CNOT (Control NOT).** Toán tử quantum CNOT có ma trận là

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} I & \mathcal{O} \\ \mathcal{O} & \text{NOT} \end{pmatrix}$$

Qubit  $|x\rangle$  với  $x \in \{0, 1\}$  đóng vai trò control cho qubit  $|y\rangle$ . Khi  $x \equiv 0$  thì  $y$  giữ nguyên, hay  $|y \oplus 0\rangle = |y \oplus x\rangle$ . Khi  $x \equiv 1$  thì áp dụng cổng NOT bên trên, khi đó  $y$  biến đổi thành  $y \oplus 1 = y \oplus x$ .

## Một ví dụ về qubit trong NSUCRYPTO 2022

Đồ vui: nếu ta có qubit là  $\alpha|0\rangle + \beta|1\rangle$ , hãy xây dựng mạch logic để biến đổi qubit trên thành  $\alpha|000\rangle + \beta|111\rangle$ .

Giải: sử dụng toán tử Hadamard và NOT. Ta có

$$NOT(\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle + \alpha|1\rangle$$

và

$$\mathcal{H}(\alpha|0\rangle + \beta|1\rangle) = \frac{1}{\sqrt{2}}(\alpha + \beta)|0\rangle + \frac{1}{\sqrt{2}}(\alpha - \beta)|1\rangle$$

Toán tử  $CNOT$  được biểu diễn bởi ma trận

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Giả sử ta có hai qubit là  $|\alpha\rangle = x|0\rangle + y|1\rangle$  và  $|\beta\rangle = z|0\rangle + t|1\rangle$ .

Khi đó  $|\alpha\rangle \otimes |\beta\rangle = xz|00\rangle + xt|01\rangle + yz|10\rangle + yt|11\rangle$ .

Qua toán tử  $CNOT$  ta có

$$\begin{aligned} CNOT(|\alpha\rangle \otimes |\beta\rangle) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} xz \\ xt \\ yz \\ yt \end{pmatrix} = \begin{pmatrix} xz \\ xt \\ yt \\ yz \end{pmatrix} \\ &= xz|00\rangle + xt|01\rangle + yt|10\rangle + yz|11\rangle \end{aligned}$$

Trường hợp  $|\alpha\rangle = |0\rangle$  thì  $x = 1, y = 0$ . Khi đó ta có tương đương

$$CNOT(|0\rangle \otimes |\beta\rangle) = z|00\rangle + t|01\rangle = |0\rangle \otimes (z|0\rangle + t|1\rangle) = |0\rangle \otimes |\beta\rangle.$$

Trường hợp  $|\alpha\rangle = |1\rangle$  thì  $x = 0, y = 1$ . Khi đó ta có tương đương

$$CNOT(|1\rangle \otimes |\beta\rangle) = t|10\rangle + z|11\rangle = |1\rangle \otimes (t|0\rangle + z|1\rangle) = |1\rangle \otimes NOT(|\beta\rangle)$$

Nói cách khác, nếu  $x \in \{0, 1\}$  thì

$$CNOT(|x\rangle \otimes |\beta\rangle) = \begin{cases} |x\rangle \otimes |\beta\rangle, & \text{nếu } x = 0 \\ |x\rangle \otimes NOT(|\beta\rangle), & \text{nếu } x = 1 \end{cases}$$

Do đó các toán tử có ma trận  $\begin{pmatrix} I_n & \mathcal{O} \\ \mathcal{O} & \mathcal{U} \end{pmatrix}$  được gọi là toán tử kiểm soát (controlled).

Một trường hợp riêng nữa là khi  $\beta = 0$  hoặc  $\beta = 1$ . Khi đó, với toán tử *NOT* bên trên ta suy ra

$$CNOT(|x\rangle \otimes |y\rangle) = |x\rangle \otimes |x \oplus y\rangle$$



# Chương 9

## Lý thuyết xác suất

### 9.1 Introduction

#### Định nghĩa xác suất

##### Định nghĩa 1. Định nghĩa cổ điển của xác suất

Định nghĩa thống kê của xác suất nói rằng, giả sử trong một phép thử có  $n$  khả năng có thể xảy ra. Xét một biến cố  $A$  xảy ra khi thực hiện phép thử có  $k$  khả năng xảy ra. Khi đó xác suất của biến cố  $A$  ký hiệu là  $P(A)$  và được tính

$$P(A) = \frac{k}{n}$$

Dễ thấy, do biến cố  $A$  là một trường hợp nhỏ trong tổng thể tất cả trường hợp khi thực hiện phép thử, do đó  $0 \leq k \leq n$ . Nghĩa là

$$0 \leq P(A) \leq 1$$

với mọi biến cố  $A$  bất kì.

**Ví dụ 1.** Xét phép thử tung hai đồng xu. Gọi  $A$  là biến cố hai đồng xu cùng mặt.

Ta ký hiệu  $S$  là đồng xu sấp,  $N$  là đồng xu ngửa. Khi đó các trường hợp có thể xảy ra của phép thử là  $S - S, S - N, N - S, N - N$  (4 trường hợp).

Trong khi đó, các trường hợp có thể xảy ra của biến cố  $A$  là  $S - S, N - N$  (2 trường hợp).

Kết luận:  $P(A) = \frac{2}{4} = \frac{1}{2}$

Chúng ta gọi tập hợp tất cả các trường hợp khi thực hiện phép thử là **không gian mẫu** và ký hiệu là  $\Omega$ . Mỗi phần tử trong không gian mẫu được gọi là **biến cố sơ cấp**. Trong ví dụ trên,  $\Omega = \{S - S, S - N, N - S, N - N\}$ .

Tập hợp các trường hợp có thể xảy ra của biến cố gọi là **không gian biến cố** và ký hiệu là  $\Omega_A$ . Trong ví dụ trên,  $\Omega_A = \{S - S, N - N\}$ .

Như vậy,  $P(A) = \frac{|\Omega_A|}{|\Omega|}$

**Ví dụ 2.** Tung hai con súc sắc cân đối và đồng chất. Tính xác suất tổng số nút của hai con súc sắc bằng 4.

Việc tung mỗi con súc sắc có 6 trường hợp. Do đó  $|\Omega| = 6^2 = 36$

Gọi  $A$  là biến cố tổng số nút của hai con súc sắc bằng 4. Ta có các trường hợp là  $4 = 1 + 3 = 3 + 1 = 2 + 2$  (3 trường hợp).

Như vậy  $|\Omega_A| = 3$  và  $P(A) = \frac{3}{36} = \frac{1}{12}$

### Định nghĩa 2. Biến cố xung khắc

Hai biến cố được gọi là **xung khắc** nếu biến cố này xảy ra thì biến cố kia chắc chắn không xảy ra. Nói cách khác giao của chúng bằng rỗng.

Khi đó, nếu  $A$  và  $B$  là hai biến cố xung khắc,

$$P(A + B) = P(A) + P(B)$$

Ta còn có thể ký hiệu  $P(A + B)$  là  $P(A \cup B)$  (hợp hai biến cố).

### Định nghĩa 3. Biến cố độc lập

Hai biến cố được gọi là **độc lập** nếu việc xảy ra của biến cố này không ảnh hưởng đến việc xảy ra của biến cố kia.

Khi đó, nếu  $A$  và  $B$  là hai biến cố độc lập thì

$$P(AB) = P(A)P(B)$$

## Xác suất có điều kiện

Xét hai tập hợp  $A$  và  $B$ . Số phần tử của phép hợp hai tập hợp trong trường hợp tổng quát được tính như sau:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Tương tự, xác suất của phép cộng xác suất đối với hai biến cố có giao khác rỗng là:

$$P(A + B) = P(A) + P(B) - P(A \cap B)$$

Xét các tập hợp  $A_1, A_2, \dots, A_n$ . Khi đó, số phần tử khi hợp các tập hợp này là:

$$\begin{aligned} |A_1 \cup A_2 \cup \dots \cup A_n| &= |A_1| + |A_2| + \dots + |A_n| - \sum_{i,j} |A_i \cap A_j| \\ &\quad + \sum_{i,j,k} |A_i \cap A_j \cap A_k| + \dots \\ &= \sum_{i=1}^n (-1)^{i+1} \sum_{j_1, j_2, \dots, j_i} |A_{j_1} \cap A_{j_2} \cap \dots \cap A_{j_i}| \end{aligned}$$

Tương tự, ta có phép cộng xác suất:

**Định lí 1. Phép cộng xác suất mở rộng**

$$P(A_1 \cup A_2 \cup \dots \cup A_n) = \sum_{i=1}^n (-1)^{i+1} \sum_{j_1, j_2, \dots, j_i} P(A_{j_1} \cap A_{j_2} \cap \dots \cap A_{j_i})$$

**Định lí 2. Xác suất có điều kiện**

Xét hai biến cố  $A$  và  $B$ . Khi đó xác suất xảy ra của biến cố  $B$  với điều kiện biến cố  $A$  xảy ra là:

$$P(B|A) = \frac{P(AB)}{P(A)} \quad (9.1)$$

Lúc này,  $A$  và  $B$  không độc lập.

Tổng quát, nếu  $n$  biến cố  $A_i, i = 1, \dots, n$  không độc lập thì:

$$\begin{aligned} P(A_1 A_2 \dots A_n) &= P(A_1) \cdot P(A_2|A_1) \cdot P(A_3|A_2 A_1) \dots \\ &\quad P(A_n|A_1 A_2 \dots A_{n-1}) \end{aligned}$$

**Ví dụ 3.** Xét hai câu hỏi trắc nghiệm có 4 lựa chọn. Tính xác suất học sinh trả lời đúng câu thứ hai với điều kiện câu đầu trả lời sai.

**Giải.** Gọi  $A$  là biến cố câu đầu tiên học sinh trả lời sai.  $P(A) = \frac{3}{4}$

Gọi  $B$  là biến cố câu thứ hai học sinh trả lời đúng.  $P(B) = \frac{1}{4}$ .

Do  $A$  và  $B$  là hai biến cố độc lập nên  $P(AB) = P(A)P(B) = \frac{3}{16}$

Như vậy, xác suất học sinh trả lời đúng câu thứ hai với điều kiện câu đầu trả lời đúng là:  $P(B|A) = \frac{P(AB)}{P(A)} = \frac{3/16}{3/4} = \frac{1}{4}$

## Công thức xác suất đầy đủ

### Định nghĩa 4. Hệ biến cố đầy đủ

Xét phép thử có không gian mẫu là  $\Omega$ . Một hệ các biến cố  $A_1, A_2, \dots, A_n$  được gọi là **đầy đủ** nếu chúng thỏa các điều kiện:

- $A_1 \cup A_2 \cup \dots \cup A_n = \Omega$
- $A_i \cap A_j = \emptyset$  với mọi  $i \neq j$

### Định lí 3. Công thức xác suất đầy đủ

Gọi  $A_1, A_2, \dots, A_n$  là một hệ biến cố đầy đủ. Khi đó, với biến cố  $B$  bất kì trong phép thử:

$$P(B) = P(A_1) \cdot P(B|A_1) + \dots P(A_n) \cdot P(B|A_n) \quad (9.2)$$

### Định lí 4. Công thức Bayes

Xét hệ có  $n$  biến cố đầy đủ  $\{A_1, A_2, \dots, A_n\}$ .

Với biến cố  $B$  bất kì thì:

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^n P(A_j)P(B|A_j)}$$

với  $1 \leq i \leq n$ .

## 9.2 Biến ngẫu nhiên

### Biến ngẫu nhiên

Xét phép thử với không gian mẫu  $\Omega$ . Với mỗi biến cố sơ cấp  $\omega \in \Omega$  ta liên kết với một số thực  $\xi(\omega) \in \mathbb{R}$  thì  $\xi$  được gọi là **biến ngẫu nhiên** (BNN).

### Định nghĩa 5. Biến ngẫu nhiên

**Biến ngẫu nhiên**  $\xi$  của một phép thử với không gian mẫu  $\Omega$  là ánh xạ:

$$\xi = \xi(\omega), \quad \omega \in \Omega$$

Giá trị  $\xi(\omega)$  được gọi là một giá trị của biến ngẫu nhiên  $\xi$ .

- Nếu  $\xi(\Omega)$  là một tập hữu hạn  $\{\xi_1, \xi_2, \dots, \xi_n\}$  hay tập vô hạn đếm được thì  $\xi$  được gọi là **biến ngẫu nhiên rời rạc**.
- Nếu  $\xi(\Omega)$  là một khoảng của  $\mathbb{R}$  hay toàn bộ  $\mathbb{R}$  thì  $\xi$  được gọi là **biến ngẫu nhiên liên tục**.

### Định nghĩa 6. Hàm phân phối

**Hàm phân phối** của biến ngẫu nhiên  $\xi$  là hàm số  $F(x)$ , xác định bởi:

$$F(x) = P(\xi \leq x), \quad x \in \mathbb{R} \quad (9.3)$$

Ở đây ta viết gọn  $P(\xi \leq x)$  từ  $P(\{\omega : \xi(\omega) \leq x\})$ . Tập hợp  $\{\omega : \xi(\omega) \leq x\}$  có thể không thuộc một biến cố nào, do đó có thể là tập rỗng (ứng với xác suất là 0).

## Tính chất của hàm phân phối

**Tính chất 1.** Hàm phân phối  $F(x)$  không giảm trên mọi đoạn thẳng.

*Chứng minh.* Đặt  $x_2 > x_1$ . Ta thấy rằng

$$\{\xi \leq x_2\} = \{\xi \leq x_1\} + \{x_1 < \xi \leq x_2\},$$

Do đó nếu ta lấy xác suất thì cũng có

$$P(\xi \leq x_2) = P(\xi \leq x_1) + P(x_1 < \xi \leq x_2)$$

Xác suất luôn không âm, hay  $P(x_1 < \xi \leq x_2) \geq 0$ , suy ra  $P(\xi \leq x_2) \geq P(\xi \leq x_1)$ , hay  $F(x_2) \geq F(x_1)$ .  $\square$

**Tính chất 2.**  $\lim_{x \rightarrow -\infty} F(x) = 0$ .

**Tính chất 3.**  $\lim_{x \rightarrow +\infty} F(x) = 1$ .

**Tính chất 4.** Hàm phân phối  $F(x)$  liên tục phải trên toàn trục số.

Để chứng minh các tính chất 2, 3, 4 chúng ta cần các tiên đề của sự liên tục (continuity axioms) và sẽ không đề cập ở đây.

## Biến ngẫu nhiên rời rạc

Cho BNN rời rạc  $\xi = \xi(\omega)$ ,  $\xi = \{a_1, a_2, \dots, a_n, \dots\}$ . Giả sử  $a_1 < a_2 < \dots < a_n < \dots$  với xác suất tương ứng là  $P(\xi = a_i) = p_i$ ,  $i = 1, 2, \dots$

$$\begin{array}{c|cccc} \xi & a_1 & a_2 & \cdots & a_n & \cdots \\ \hline P & p_1 & p_2 & \cdots & p_n & \cdots \end{array}$$

Ta có thể biểu diễn biến ngẫu nhiên và xác suất tương ứng của nó bằng bảng phân phối xác suất của  $\xi$ .

Rõ ràng rằng  $p_n \geq 0$  với mọi  $n$ . Hơn nữa

$$\sum_{n=1}^{\infty} p_n = 1$$

Không gian mẫu lúc này là hợp của các tập biến ngẫu nhiên rời rạc:

$$\Omega = \{\xi = a_1\} \cup \{\xi = a_2\} \cup \dots$$

Các biến ngẫu nhiên xung khắc nhau (vì  $\xi$  không thể nhận hai giá trị khác nhau cùng lúc), do đó xác suất cả không gian mẫu là

$$1 = P(\Omega) = P(\xi = a_1) + P(\xi = a_2) + \dots = p_1 + p_2 + \dots$$

### Định nghĩa 7. Phân phối nhị thức

Biến ngẫu nhiên  $\xi$  được gọi là có **phân phối nhị thức** với tham số  $p, n$ , với  $p \in (0, 1)$  và  $n$  là số tự nhiên, nếu  $\xi$  nhận các giá trị  $0, 1, \dots, n$  và

$$P(\xi = k) = C_n^k p^k q^{n-k}, \quad k = 0, 1, \dots, n \quad (9.4)$$

Ở đây  $q = 1 - p$ .

**Ví dụ 4.** Một bài kiểm tra có 100 câu hỏi trắc nghiệm bốn đáp án. Xác suất chọn ngẫu nhiên đúng đáp án của mỗi câu hỏi thì bằng nhau và bằng  $\frac{1}{4}$ .

Ở đây xác suất chọn ngẫu nhiên đúng đáp án của một câu hỏi bất kì là  $p = \frac{1}{4}$ , và số lượng câu hỏi là  $n = 100$ .

Gọi  $\xi$  là biến ngẫu nhiên số câu hỏi trả lời đúng. Khi đó  $\xi$  nhận các giá trị  $0, 1, \dots, 100$ .

Do đó bài toán này có phân phối nhị thức và

$$P(\xi = k) = C_{100}^k \left(\frac{1}{4}\right)^k \left(\frac{3}{4}\right)^{100-k}$$

**Định nghĩa 8. Phân phối Poisson**

Biến ngẫu nhiên  $\xi$  được gọi là có **phân phối Poisson** với tham số  $\lambda$ , nếu  $\xi$  nhận các giá trị  $0, 1, \dots, n$  và

$$P(\xi = k) = \frac{\lambda^k \cdot e^{-\lambda}}{k!}, \quad k = 0, 1, \dots, n \quad (9.5)$$

Tham số  $\lambda$  thể hiện số lần trung bình mà một sự kiện xảy ra trong một khoảng thời gian nhất định. Khi đó, nếu một biến ngẫu nhiên có số lần xuất hiện trung bình của một sự kiện trong thời gian  $t$  thì nó có phân phối Poisson với tham số  $\lambda t$ , với  $\lambda$  là số lần trung bình trong một đơn vị thời gian.

**Biến ngẫu nhiên liên tục****Định nghĩa 9. Biến ngẫu nhiên liên tục**

Biến ngẫu nhiên  $\xi$  được gọi là **liên tục**, nếu nó nhận giá trị tại mọi điểm thuộc một đoạn liên tục nào đó trên trục số, và tồn tại một hàm số không âm  $p(x)$  sao cho với mọi đoạn  $[a, b]$  (hữu hạn hoặc vô hạn) ta có

$$P(a \leq \xi \leq b) = \int_a^b p(x) dx \quad (9.6)$$

Hàm  $p(x)$  được gọi là **hàm mật độ** của biến ngẫu nhiên  $\xi$ .

Tương tự biến ngẫu nhiên rời rạc,  $p(x) \geq 0$  với mọi  $x \in \mathbb{R}$  và khi hai cận là vô cực thì biến ngẫu nhiên bao quát toàn bộ không gian mẫu. Nghĩa là

$$\int_{-\infty}^{+\infty} p(x) dx = 1$$

Từ định nghĩa của hàm phân phối  $F(x) = P(\xi \leq x)$  ta có hai tính chất của hàm mật độ:

1.  $F(x) = \int_{-\infty}^x p(x) dx$
2.  $p(x) = F'(x)$

Tính chất thứ nhất là từ định nghĩa hàm phân phối. Tính chất thứ hai suy ra từ việc cận trên của tích phân là hữu hạn.

Hàm mật độ của  $X$  là

$$f(x) = \begin{cases} p_i & \text{khi } x = x_i, \\ 0 & \text{khi } x \neq x_i, \forall i \end{cases}$$

### Nhận xét 1

Ta có các lưu ý sau:

- $p_i \geq 0, \sum p_i = 1, i = 1, 2, \dots$
- $P(a < X \leq b) = \sum_{a < x_i \leq b} p_i$

## Hàm mật độ của biến ngẫu nhiên liên tục

### Định nghĩa 10

Hàm số  $f : \mathbb{R} \mapsto \mathbb{R}$  được gọi là **hàm mật độ** của biến ngẫu nhiên liên tục  $X$  nếu:

$$P(a \leq X \leq b) = \int_a^b f(x) dx, \forall a, b \in \mathbb{R}$$

### Nhận xét 2

Với mọi  $x \in \mathbb{R}$ ,  $f(x) \geq 0$  và  $\int_{-\infty}^{+\infty} f(x) dx = 1$ .

**Ý nghĩa hình học.** Xác suất của biến ngẫu nhiên  $X$  nhận giá trị trong  $[a, b]$  bằng diện tích hình thang cong giới hạn bởi  $x = a$ ,  $x = b$ ,  $y = f(x)$  và  $Ox$ .



# Chương 10

## Hình học giải tích

### 10.1 Theo dòng lịch sử

Hình học xuất hiện từ thời xa xưa, xuất phát từ những nhu cầu thực tế nhất của con người là đo đạc để phân chia đất đai, xây dựng, canh tác, ... Từ đó con người đã có nhận thức rất sớm về quan hệ song song và vuông góc giữa hai đường thẳng.

Một cách hình ảnh (mà thật ra hình học là môn học về hình ảnh) thì hai đường thẳng song song không cắt nhau dù có kéo dài chúng ra vô tận. Các đường thẳng song song luôn có nhiều điều thú vị, cả ở mặt phẳng Euclid lẫn trong không gian. Đầu tiên phải kể đến định lý mang tên triết gia vĩ đại của Hy Lạp: Thales.

#### Thales của Miletus

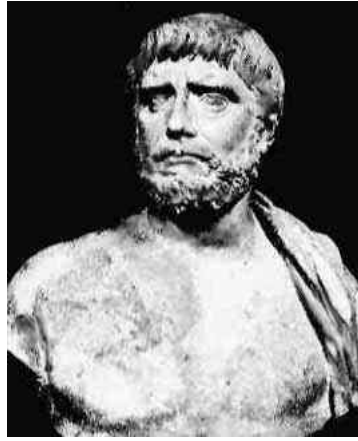
Thales của Miletus được cho rằng sinh vào khoảng năm 624 Trước Công nguyên (TCN) và mất năm 547 TCN tại Miletus (Thổ Nhĩ Kỳ ngày nay)<sup>1</sup>.

Ông được xem là nhà triết học đầu tiên khi không cố gắng giải thích tự nhiên bằng thần thoại hay các thế lực siêu nhiên như trước. Trường phái triết học do ông sáng lập, trường phái Milet, cho rằng mọi vật có nguồn gốc từ nước. Nhà triết học nổi tiếng Aristotle đánh giá rằng Thales là người sáng lập ra *triết học duy vật sơ khai*.

Trong toán học, Thales được biết tới với định lý mang tên ông về các đường song song. Định lý Thales được phát biểu như sau:

---

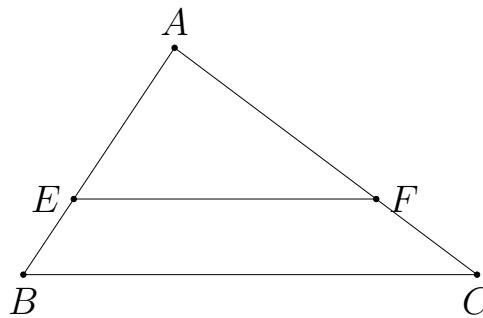
<sup>1</sup><https://mathshistory.st-andrews.ac.uk/Biographies/Thales/>



Hình 10.1. Thales của Miletus

**Định lí 1. Định lý Thales**

Trong một tam giác, đường thẳng song song với một cạnh chắn trên hai cạnh còn lại các đoạn thẳng tương ứng tỉ lệ.



Hình 10.2. Định lý Thales trên mặt phẳng

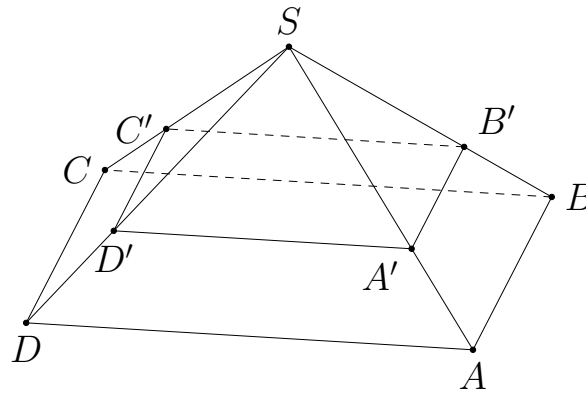
Theo định lý Thales, nếu  $EF$  song song với  $BC$  thì ta có  $\frac{AE}{AB} = \frac{AF}{AC} = \frac{EF}{BC}$  (hình 10.2).

Không dừng lại ở mặt phẳng, khi mở rộng lên không gian định lý Thales cũng cho chúng ta một kết quả quan trọng khi nói tới các mặt phẳng song song nhau.

**Định lí 2. Định lý Thales trong không gian**

Trong khối chóp, mặt phẳng song song mặt đáy chắn các cạnh nối từ đỉnh hình chóp tới các đỉnh của mặt phẳng đáy các đoạn thẳng tương ứng tỉ lệ.

Theo định lý Thales, nếu mặt phẳng  $(ABCD)$  song song với mặt phẳng  $(A'B'C'D')$  thì  $\frac{SA}{SA'} = \frac{SB}{SB'} = \frac{SC}{SC'} = \frac{SD}{SD'}$  (hình 10.3).



Hình 10.3. Định lý Thales trong không gian

## Pythagoras của Samos

Khi nhắc tới vuông góc, chúng ta thường nhớ tới định lý ngày nào được học ở thời học sinh: định lý Pythagoras. Định lý này nói về quan hệ giữa độ dài các cạnh trong một tam giác vuông. Định lý tuy đơn giản nhưng có ý nghĩa rất quan trọng trong đời sống và khoa học của con người suốt chiều dài lịch sử. Đây cũng là tiền đề cho định lý mang tính lịch sử của nhân loại: định lý cuối cùng của Fermat.



Hình 10.4. Pythagoras của Samos

Pythagoras của Samos cũng là nhà triết học Hy Lạp cổ, được cho rằng sinh vào khoảng năm 570 TCN và mất năm 490 TCN<sup>2</sup>. Ông được học tập từ nhà triết học Thales và cũng có nhiều đóng góp cho sự phát triển của toán học, thiên văn học và âm nhạc. Tuy nhiên khác với thầy mình, trường phái triết học của ông cho rằng những con số là nguồn gốc của vạn vật và sử dụng những con số để giải thích những hiện tượng khoa học. Từ đây, các lý thuyết về âm nhạc được ra đời, cụ thể là các mối liên hệ về tần số với sự rung của dây nhạc cụ.

Ông là một trong những người hiếm hoi cho phép cả phụ nữ đi học ở lớp của mình vào thời ấy. Điều đó giúp phổ biến toán học nói riêng và kiến thức

<sup>2</sup><https://mathshistory.st-andrews.ac.uk/Biographies/Pythagoras/>

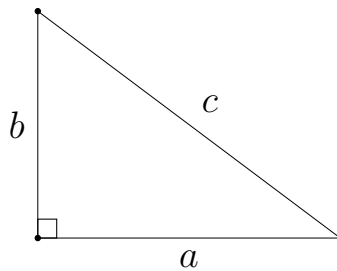
nói chung tới nhiều tầng lớp nhân dân. Tuy nhiên ông cũng có một hội kín rất thú vị. Như đã nói ở trên, trường phái triết học Pythagoras cố gắng giải thích nguồn gốc vạn vật bằng những con số. Điều này đã dẫn họ tới những khám phá động trời vào thời ấy.

Một trong những khám phá đó là về sự tồn tại của số vô tỉ dựa vào định lý mang tên ông. Lịch sử đã chỉ ra rằng trước Pythagoras, người Babylon và Ai Cập đã tìm ra rất nhiều bộ số nguyên  $(a, b, c)$  thỏa mãn  $a^2 + b^2 = c^2$  là độ dài ba cạnh tam giác vuông. Định lý Pythagoras mà ngày nay chúng ta biết được phát biểu rằng:

### Định lí 3. Định lý Pythagoras

Trong một tam giác vuông, bình phương độ dài cạnh huyền bằng tổng bình phương độ dài hai cạnh góc vuông.

Như vậy nếu gọi độ dài cạnh huyền là  $c$ , độ dài hai cạnh góc vuông lần lượt là  $a$  và  $b$  thì  $a^2 + b^2 = c^2$  (hình 10.5).



Hình 10.5. Định lý Pythagoras

Nếu  $a = b = 1$  thì sao? Khi đó bình phương độ dài cạnh huyền  $c^2 = 2$ . Tuy nhiên không thể tìm ra một số hữu tỉ nào để bình phương lên là 2 cả. Phát hiện này là một chấn động đối với thời Pythagoras và ông yêu cầu tất cả thành viên trong hội phải giữ kín bí mật về sự phát hiện này. Tuy nhiên thông tin vẫn lọt ra ngoài và truyền thuyết kể rằng ông đã xử tội chết cho thành viên của hội không tuân thủ.

Pythagoras đã đưa một khái niệm cực kì quan trọng trong toán học, gọi là *chứng minh* (proof). Để chứng minh một mệnh đề là đúng, chúng ta cần các mệnh đề (thường đơn giản hơn) đúng trước đó. Bằng các phép suy luận thích hợp dựa trên các mệnh đề đúng trước đó, chúng ta có thể kết luận rằng mệnh đề cần chứng minh là đúng. Phép chứng minh có thể gọi là *xương song* của toán học, vì nếu không có một phép chứng minh đúng đắn thì một mệnh đề không thể được xác định được là có đúng hay không. Trong trường hợp của Fermat, khi ông đưa ra định lý Fermat nhưng không kèm chứng minh (vì lẽ sách quá chật nên không viết lời giải được) thì chúng ta không thể biết định lý Fermat có đúng hay không (?).

Nếu việc suy luận dựa trên các mệnh đề, hoặc định lý, đã đứng trước đó, thì phải có một lúc nào đó việc này dừng lại. Chúng ta không thể suy ngược tới vô hạn lần được. Do đó chúng ta cần những mệnh đề luôn đúng nhưng tính đúng đắn của nó được kiểm nghiệm trong thực tiễn. Chúng được gọi là *tiên đề* (axiom). Nhân vật tiếp theo được đề cập tới sẽ dẫn chúng ta tới hệ thống tiên đề làm nền tảng cho hình học.

## Euclid của Alexandria

Đúng vậy, Euclid là người đặt nền móng cho hình học với bộ sách nổi tiếng *Elements* của mình. Trong bộ sách này đề cập tới những tiên đề, định lý làm nền tảng cho bộ môn hình học và vẫn còn ý nghĩa cho tới tận ngày nay. Những gì viết trong đó không quá xa lạ với những gì được giảng dạy trong nhà trường.



Hình 10.6. Euclid của Alexandria

Euclid của Alexandria sinh vào khoảng năm 325 TCN và mất vào khoảng năm 265 TCN<sup>3</sup>. Thông tin về ông không có nhiều. Nhưng chỉ mỗi bộ sách *Elements* cũng đủ để người đời sau cho rằng ông là người có ảnh hưởng nhất trong 2000 năm lịch sử phát triển của toán học.

Năm tiên đề cơ bản của hình học được ông phát biểu trong bộ *Elements* được phát biểu như sau:

1. Qua hai điểm bất kì luôn vẽ được một đường thẳng
2. Đường thẳng có thể kéo dài vô hạn về cả hai phía
3. Ta có thể xác định một đường tròn bằng tâm và bán kính của nó
4. Mọi góc vuông đều bằng nhau
5. Nếu một đường thẳng cắt hai đường thẳng khiến tổng hai góc trong cùng phía nhỏ hơn hai vuông thì hai đường thẳng đó chắc chắn sẽ cắt nhau tại một điểm nào đó

Tiên đề số 5 là rắc rối và phức tạp nhất. Nó không thực sự tự nhiên và có nhiều sự vướng mắc. Đây chính là tiên đề cho sự ra đời của hình học phi-Euclid

<sup>3</sup><https://mathshistory.st-andrews.ac.uk/Biographies/Euclid/>

hơn 1500 năm sau.

Bộ *Elements* của Euclid bao gồm 13 quyển. Trong đó đề cập tới rất nhiều vấn đề của hình học, từ những phần tử đơn giản nhất cấu tạo nên hình học là điểm, đoạn thẳng, đường thẳng, tới những hình học lớn hơn như hình chữ nhật, hình tròn, đa giác, mặt phẳng. Thậm chí ông cũng đã có những dấu chân ở hình học không gian như hình chóp, hình cầu, hình nón ([1], [3]).

## 10.2 Phương pháp tọa độ trong mặt phẳng

Cuộc cách mạng trong hình học xảy ra khi nhà toán học lãng tử René Descartes phát minh ra hệ tọa độ và từ đó mọi đối tượng hình học có thể được biểu diễn bởi các phương pháp đại số như phương trình, đẳng thức.

### Danh mục thuật ngữ và ký hiệu

Đầu tiên chúng ta thống nhất các thuật ngữ cũng như ký hiệu được sử dụng kể từ đây.

**Điểm** là đơn vị cơ bản của hình học. Bất kỳ đối tượng hình học nào cũng là một *tập hợp điểm*. Điểm được ký hiệu bởi chữ in hoa, ví dụ như  $A$ ,  $B_1$ ,  $B_2$ .

**Đường thẳng** đi qua hai điểm phân biệt cho trước. Đường thẳng có thể kéo dài vô hạn về hai phía. Đường thẳng được ký hiệu bởi chữ in thường hoặc chữ Hy Lạp trong ngoặc đơn, ví dụ như  $(d)$ ,  $(\Delta)$ .

**Đoạn thẳng** chỉ phần đường thẳng nằm giữa hai điểm.

**Nửa đường thẳng** chỉ phần đường thẳng nằm một phía của một điểm trên đường thẳng và chỉ kéo dài vô hạn về phía đó.

**Vector** là đoạn thẳng có hướng. Với điểm đầu là  $A$  và điểm cuối là  $B$  thì vector từ  $A$  tới  $B$  được ký hiệu là  $\overrightarrow{AB}$ . Để chỉ một vector không cần biết điểm đầu và điểm cuối ta dùng chữ thường in đậm, ví dụ như  $\mathbf{a}$ .

**Góc giữa hai vector**  $\overrightarrow{OA}$  và  $\overrightarrow{OB}$  là góc  $\angle AOB$  và ký hiệu là  $(\overrightarrow{OA}, \overrightarrow{OB})$ .

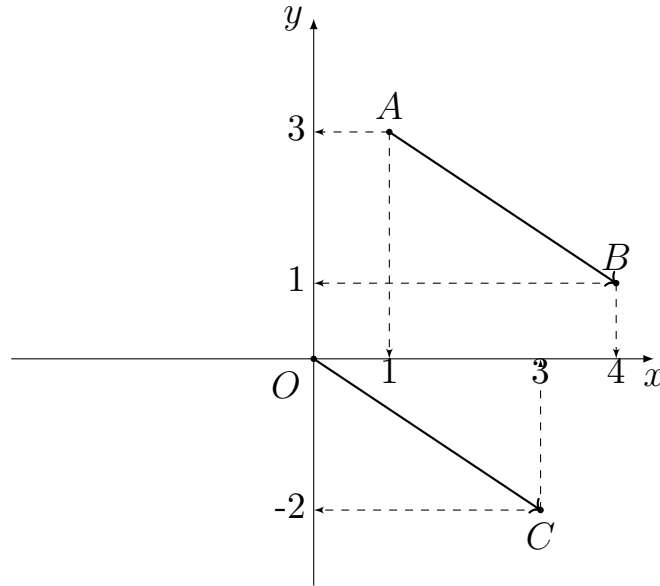
Tương tự đối với vector  $\mathbf{a}$  và  $\mathbf{b}$  thì góc giữa chúng ký hiệu là  $(\mathbf{a}, \mathbf{b})$ .

### Vector trong mặt phẳng

Trong hệ tọa độ  $Oxy$  với tâm  $O$  và hai trục  $Ox$  (trục hoành) và  $Oy$  (trục tung) vuông góc nhau, đặt  $O = (0, 0)$  là tọa độ của tâm  $O$ .

Tiếp theo, mọi điểm trong mặt phẳng Euclid đi liền với cặp số  $(x, y)$  chỉ tọa độ của điểm đó. Ví dụ  $A = (1, 3)$ ,  $B = (4, 1)$ .

Tọa độ của điểm cũng là tọa độ của vector từ  $O$  tới điểm đó. Với hình 10.7 thì  $\overrightarrow{OA} = (1, 3)$  và  $\overrightarrow{OB} = (4, 1)$ . Tọa độ của vector  $\overrightarrow{AB}$  khi đó sẽ là



Hình 10.7. Tọa độ của điểm trong mặt phẳng

$\overrightarrow{AB} = \overrightarrow{OB} - \overrightarrow{OA} = (4, 1) - (1, 3) = (3, -2)$ . Cũng theo hình 10.7 thì ta thấy  $\overrightarrow{AB} = \overrightarrow{OC} = (3, -2)$ .

Như vậy, nếu ta có hai điểm  $A = (x_A, y_A)$  và  $B = (x_B, y_B)$  thì vector  $\overrightarrow{AB}$  là

$$\overrightarrow{AB} = (x_B - x_A, y_B - y_A) \quad (10.1)$$

**Tích vô hướng của hai vector**  $\mathbf{a} = (x_1, y_1)$  và  $\mathbf{b} = (x_2, y_2)$  được định nghĩa là

$$\langle \mathbf{a}, \mathbf{b} \rangle = x_1 x_2 + y_1 y_2 \quad (10.2)$$

Ta cũng có thể ký hiệu tích vô hướng là  $\mathbf{a} \cdot \mathbf{b}$ .

Ta ký hiệu  $\|\mathbf{a}\|$  là độ dài (chuẩn Euclid, Euclid norm) của vector  $\mathbf{a}$ . Trong hệ tọa độ Descartes vuông góc, theo định lý Pythagoras, độ dài của vector là độ dài cạnh huyền tam giác vuông (hình 10.7). Như vậy, độ dài đoạn thẳng  $AB$  với  $A = (x_A, y_A)$  và  $B = (x_B, y_B)$  là

$$AB = \|\overrightarrow{AB}\| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \quad (10.3)$$

Khi đó cosin góc giữa hai vector  $\mathbf{a}$  và  $\mathbf{b}$  là

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \cdot \sqrt{x_2^2 + y_2^2}} \quad (10.4)$$

Nếu góc giữa hai vector bằng 90 độ thì hai vector được gọi là vuông góc nhau. Khi đó tích vô hướng  $\mathbf{a} \cdot \mathbf{b} = 0$ .

## Phương trình đường thẳng trong mặt phẳng

Theo tiên đề Euclid, một đường thẳng được xác định khi biết hai điểm phân biệt thuộc đường thẳng đó. Trong hệ tọa độ, chúng ta có hai cách tìm phương trình đường thẳng.

**Bằng vector pháp tuyến.** Vector pháp tuyến của đường thẳng là vector vuông góc với mọi vector có phương là đường thẳng đó. Giả sử  $\mathbf{v} = (a, b)$  là vector pháp tuyến của đường thẳng đi qua điểm  $M_0 = (x_0, y_0)$ . Khi đó đường thẳng đi qua  $M_0$  nhận  $\mathbf{v}$  làm vector pháp tuyến là tập hợp điểm  $M = (x, y)$  trên mặt phẳng sao cho  $\mathbf{v} \cdot \overrightarrow{M_0M} = 0$ . Điều này tương đương với

$$\mathbf{v} \cdot \overrightarrow{M_0M} = a \cdot (x - x_0) + b \cdot (y - y_0) = 0 \quad (10.5)$$

**Bằng vector chỉ phương.** Vector chỉ phương của đường thẳng là vector có phương song song với đường thẳng đó. Giả sử  $\mathbf{v}' = (a', b')$  là vector chỉ phương của đường thẳng đi qua điểm  $M_0 = (x_0, y_0)$ . Khi đó đường thẳng đi qua  $M_0$  nhận  $\mathbf{v}'$  làm vector chỉ phương là tập hợp điểm  $M = (x, y)$  trên mặt phẳng sao cho  $\mathbf{v}' \parallel \overrightarrow{M_0M}$ . Điều này tương đương với

$$\mathbf{v}' \parallel \overrightarrow{M_0M} \Leftrightarrow \frac{x - x_0}{a'} = \frac{y - y_0}{b'} \quad (10.6)$$

1. Cả hai cách biểu diễn khi khai triển ra đều có dạng  $ax + by + c = 0$  với  $c$  là hằng số. Đây được gọi là dạng tổng quát của phương trình đường thẳng.
2. Cách viết  $\frac{x - x_0}{a'} = \frac{y - y_0}{b'}$  được gọi là dạng chính tắc của phương trình đường thẳng.
3. Dạng chính tắc của phương trình đường thẳng còn có một tác dụng đặc biệt khác

$$\frac{x - x_0}{a'} = \frac{y - y_0}{b'} = t$$

với  $t \in \mathbb{R}$ . Khi đó tọa độ  $M = (x, y)$  có thể được biểu diễn dưới dạng

$$\begin{cases} x = x_0 + a't \\ y = y_0 + b't \end{cases}, \quad t \in \mathbb{R} \quad (10.7)$$

Đây được gọi là phương trình dạng tham số.

Chúng ta chú ý rằng nếu đường thẳng song song với một trong hai trục tọa độ thì vector chỉ phương của nó sẽ cùng phương với vector đơn vị  $(1, 0)$  hoặc  $(0, 1)$ . Do đó không thể viết dưới dạng chính tắc được (không thể chia cho 0) nhưng có thể viết dưới dạng tổng quát hoặc dạng tham số.

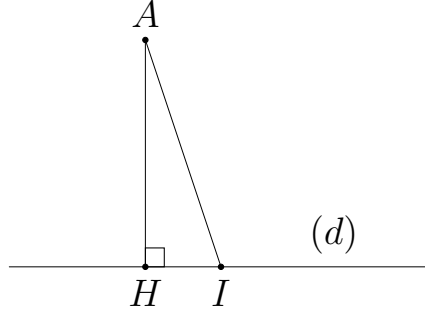
## Khoảng cách giữa điểm và đường thẳng

Nhắc lại một chút kiến thức cơ sở. **Khoảng cách** từ một điểm  $A$  nằm ngoài đường thẳng  $(d)$  là độ dài đoạn thẳng  $AH$  với  $H \in (d)$  sao cho  $AH$  nhỏ nhất



(hình 10.8).

Khi đó  $H$  được gọi là **hình chiếu** của  $A$  lên đường thẳng  $(d)$  và  $AH$  là **khoảng cách** từ  $A$  tới  $(d)$ . Do  $AH$  là đoạn thẳng có độ dài ngắn nhất, điều này xảy ra khi  $AH \perp (d)$ .



Hình 10.8. Hình chiếu và khoảng cách tới đường thẳng

Như vậy, để tìm hình chiếu của điểm  $A$  lên đường thẳng  $(d)$ , ta dựng đường thẳng đi qua điểm  $A$  và vuông góc với  $(d)$ .

Giả sử phương trình đường thẳng  $(d)$  với vector pháp tuyến  $\mathbf{v} = (a, b)$  là  $(d) : ax + by + c = 0$ .

Gọi  $(d')$  là đường thẳng đi qua  $A = (x_0, y_0)$  và vuông góc với  $d$ . Do  $\mathbf{v}$  là vector pháp tuyến của  $(d)$  nên  $\mathbf{v}$  là vector chỉ phương của  $(d')$ . Khi đó phương trình dạng tham số của  $(d')$  là

$$\begin{cases} x = x_0 + at \\ y = y_0 + bt \end{cases}, t \in \mathbb{R}$$

Gọi  $H$  là hình chiếu của  $A$  lên  $(d)$ . Khi đó  $H$  là giao điểm của  $(d)$  và  $(d')$ . Vì  $H \in (d')$  nên tọa độ của  $H$  có dạng  $(x_0 + at, y_0 + bt)$  với  $t$  nào đó thuộc  $\mathbb{R}$ . Chúng ta sẽ đi tìm  $t$  này.

Vì  $H \in (d)$  nên ta thay tọa độ của  $H$  vừa tìm được vào phương trình của  $(d)$  thu được

$$a(x_0 + at) + b(y_0 + bt) + c = 0 \Leftrightarrow t = -\frac{ax_0 + by_0 + c}{a^2 + b^2}$$

Như vậy là ta đã tìm được  $t$  từ đó xác định được tọa độ của  $H$ .

Từ đây ta tính được khoảng cách từ  $A$  tới  $(d)$  hay nói cách khác là độ dài đường  $AH$ . Ta có  $A = (x_0, y_0)$  và  $H = (x_0 + at, y_0 + bt)$  nên  $\overrightarrow{AH} = (at, bt)$ . Suy ra

$$\begin{aligned} AH &= \|\overrightarrow{AH}\| = \sqrt{(at)^2 + (bt)^2} = |t|\sqrt{a^2 + b^2} \\ &= \left| -\frac{ax_0 + by_0 + c}{a^2 + b^2} \right| \cdot \sqrt{a^2 + b^2} = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}} \end{aligned}$$

## 10.3 Đạo hàm

Phép tính vi tích phân đã được con người nghiên cứu từ lâu. Câu chuyện về ai là người phát minh ra phép tính vi tích phân: Newton hay Leibniz, được coi là một trong những vụ tranh cãi đáng xấu hổ nhất lịch sử toán học. Nhưng họ cũng đã để lại một mảnh đất màu mỡ cho toán học về sau.

### Cơ học và sự ra đời của đạo hàm

Trường phái Newton sử dụng đạo hàm như công cụ khảo sát vận tốc từ quãng đường. Ở bậc trung học chúng ta biết rằng *vận tốc trung bình* bằng quãng đường chia thời gian. Tuy nhiên điều đó chỉ đúng cho *chuyển động thẳng đều*. Nếu quãng đường là một hàm số phụ thuộc thời gian (quãng đường là  $s(t)$  với  $t$  là thời gian) thì điều đó không đúng nữa.

Do quãng đường phụ thuộc thời gian nên có thể là vận tốc cũng phụ thuộc thời gian? Hợp lí đấy. Nhưng với mỗi một giá trị thời gian  $t$  cho ta một vị trí  $s(t)$  trên trục số, còn vận tốc thì không thể phụ thuộc một giá trị thời gian được. Rõ ràng vật phải di chuyển một quãng đường từ thời gian  $t_0$  tới  $t_1$  thì mới có vận tốc trên quãng đường đó chứ?

Cách tiếp cận ở đây là, chúng ta cho sự thay đổi thời gian, tức hiệu  $\Delta t = t_1 - t_0$ , rất nhỏ. Khi đó vật đi từ  $s(t_0)$  tới  $s(t_1)$ , vậy là chúng ta có thể tính vận tốc với công thức  $v = \frac{s(t_1) - s(t_0)}{t_1 - t_0}$ . Do  $\Delta t$  rất nhỏ, hay *tiến về 0*, thì vận tốc gần như xảy ra vào đúng một thời điểm. Do đó vận tốc lúc này được gọi là *vận tốc tức thời*. Đó cũng chính là ý nghĩa cơ học và sự ra đời của đạo hàm theo trường phái Newton.

### Định nghĩa đạo hàm

Xét hàm số  $f(x)$  liên tục trên khoảng  $(a, b)$  có chứa điểm  $x_0$ . Đạo hàm của  $f(x)$  tại  $x_0$  được định nghĩa là giới hạn

$$f'(x_0) = \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0} \quad (10.8)$$

Lưu ý rằng nếu giới hạn trên không phải là giới hạn hữu hạn (không tồn tại hoặc tiến tới vô cực) thì hàm số không có đạo hàm tại điểm  $x_0$ .

Ví dụ, để tính đạo hàm của hàm số  $f(x) = x^3 + 2x^2 - 4$  tại  $x_0 = 4$ , ta khai

triển

$$\begin{aligned}
 \frac{f(x) - f(x_0)}{x - x_0} &= \frac{f(x) - f(4)}{x - 4} \\
 &= \frac{x^3 + 2x^2 - 4 - (4^3 + 2 \cdot 4^2 - 4)}{x - 4} \\
 &= \frac{(x^3 - 4^3) + 2(x^2 - 4^2)}{x - 4} \\
 &= \frac{(x - 4)(x^2 + 4x + 16) + 2(x - 4)(x + 4)}{x - 4} \\
 &= x^2 + 4x + 16 + 2(x + 4)
 \end{aligned}$$

Cho  $x$  tiến tới 4 thì ta có đạo hàm tại  $x = 4$

$$\begin{aligned}
 f'(4) &= \lim_{x \rightarrow 4} \frac{f(x) - f(4)}{x - 4} \\
 &= \lim_{x \rightarrow 4} (x^2 + 4x + 16 + 2(x + 4)) \\
 &= 4^2 + 4 \cdot 4 + 16 + 2 \cdot (4 + 4) = 64
 \end{aligned}$$

Trong định nghĩa ở 10.8, nếu ta đặt  $\Delta x = x - x_0$  và  $\Delta y = y - y_0 = f(x) - f(x_0)$ , ta gọi  $\Delta x$  là *số gia* của biến  $x$ , tương tự  $\Delta y$  là *số gia* của biến  $y$ .

Trong định nghĩa,  $x$  tiến tới  $x_0$  tương đương với  $\Delta x$  tiến tới 0. Chuyển về  $x_0$  ta có  $x = x_0 + \Delta x$  và từ đó  $f(x) = f(x_0 + \Delta x)$ . Định nghĩa đạo hàm ở trên có thể được viết lại

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} \quad (10.9)$$

Nếu hàm số có đạo hàm tại mọi điểm trên khoảng  $(a, b)$  thì ta nói hàm số khả vi trên khoảng đó.

Ví dụ đối với hàm số  $f(x) = x^3 + 2x^2 - 4$  như trên. Với mọi  $x_0 \in \mathbb{R}$  ta có

$$\begin{aligned}
 f'(x_0) &= \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0} \\
 &= \lim_{x \rightarrow x_0} \frac{x^3 + 2x^2 - 4 - (x_0^3 + 2x_0^2 - 4)}{x - x_0} \\
 &= \lim_{x \rightarrow x_0} \frac{(x^3 - x_0^3) + 2(x^2 - x_0^2)}{x - x_0} \\
 &= \lim_{x \rightarrow x_0} (x^2 + xx_0 + x_0^2) + 2(x + x_0) \\
 &= x_0^2 + x_0 \cdot x_0 + x_0^2 + 2(x_0 + x_0) = 3x_0^2 + 4x_0
 \end{aligned}$$

Ta thấy rằng giới hạn trên luôn tồn tại với mọi  $x_0 \in \mathbb{R}$  nên thay  $x_0$  thành  $x$  ta có đạo hàm  $f'(x) = 3x^2 + 4x$  của  $f(x)$  trên  $\mathbb{R}$ .

## Vi phân

Trong cách ký hiệu

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x}$$

ta thay  $\Delta y$  thành  $dy$  và  $\Delta x$  thành  $dx$  thì vi phân được định nghĩa là

$$f'(x) = \frac{dy}{dx} \Leftrightarrow dy = f'(x) dx \quad (10.10)$$

Cách ký hiệu vi phân có ý nghĩa là vế trái là vi phân theo biến  $y$  và vế phải là vi phân theo biến  $x$ . Do  $y = f(x)$  nên khi vi phân hai vế sẽ cho ra  $dy = f'(x) dx$  (vế trái là đa thức bậc 1 biến  $y$ ).

Ví dụ phương trình  $y^2 = x^3 + 4x - 7$  thì khi vi phân hai vế ta có

$$(y^2)' dy = (x^3 + 4x - 7) dx \Leftrightarrow 2y dy = (3x^2 + 4) dx$$

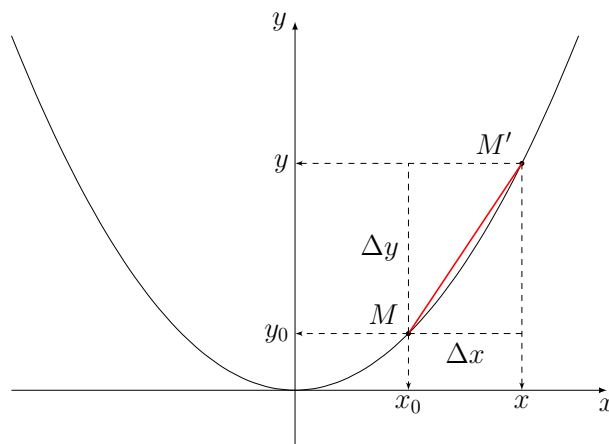
## Ý nghĩa hình học của đạo hàm

Xét hàm số  $y = f(x)$  liên tục trên khoảng  $(a, b)$  chứa điểm  $x_0$ .

Gọi  $M' = (x, y)$  là một điểm thuộc hàm số  $y = f(x)$ . Khi đó đạo hàm của  $f(x)$  tại  $x_0$  là giới hạn

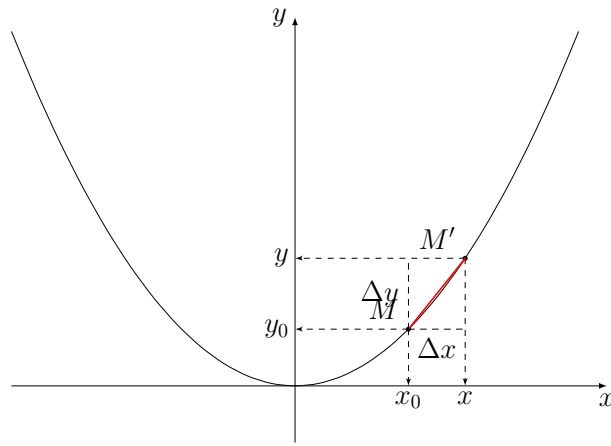
$$\lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0} = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x}$$

Xét hình 10.9, tỉ số  $\Delta y/\Delta x$  là tangent của góc hợp bởi trục hoành  $Ox$  và đường thẳng  $MM'$ .



Hình 10.9. Hệ số góc (trường hợp 1)

Tiếp theo, xét hình 10.10, ta thấy đường thẳng  $MM'$  ngày càng tiến sát lại với đường cong. Như vậy, khi  $\Delta x$  tiến tới 0 thì đường thẳng  $MM'$  cắt đường



Hình 10.10. Hệ số góc (trường hợp 2)

cong tại hai điểm càng sát nhau. Đến khi hai điểm đó trùng nhau, đường thẳng  $MM'$  chỉ đi qua đúng một điểm thuộc đường cong và khi đó  $MM'$  trở thành tiếp tuyến của đường cong tại điểm  $M = (x_0, y_0)$ .

Khi đó  $f'(x_0)$  là tangent của góc hợp bởi  $MM'$  và trục hoành  $Ox$ , hay nói cách khác là *hệ số góc* của đường tiếp tuyến. Thêm nữa  $f'(x_0) = \frac{\Delta y}{\Delta x} = \frac{y - y_0}{x - x_0}$  nên phương trình đường tiếp tuyến đi qua  $M = (x_0, y_0)$  là

$$y = f'(x_0)(x - x_0) + y_0 \quad (10.11)$$

## 10.4 Tích phân

Tích phân là khái niệm quan trọng trong giải tích. Sau đây sẽ trình bày cách tính tích phân theo tổng Riemann.

### Tích phân và phân chia diện tích

Xét phương trình của một đường cong  $y = f(x) > 0$  trên đoạn  $[a, b]$ .

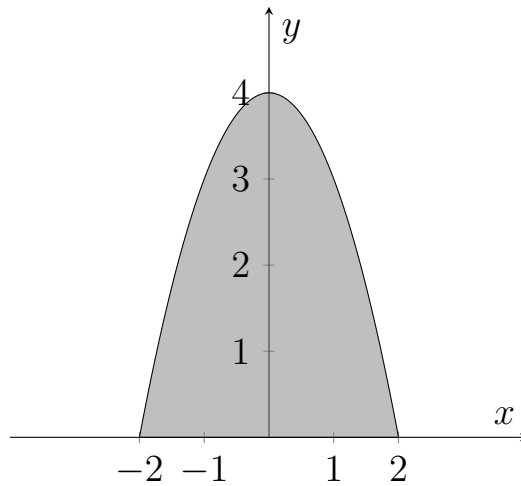
Theo định nghĩa, tích phân từ  $a$  tới  $b$  là diện tích phần hình phẳng giới hạn bởi đường cong  $y = f(x)$ , trục hoành  $Ox$  và hai trục đứng  $x = a$ ,  $x = b$ .

Ở hình 10.11, diện tích phần tô màu xám là tích phân từ -2 tới 2 của hàm số  $f(x) = -x^2 + 4$ .

Chúng ta có thể tính diện tích hình chữ nhật, hình thang, hình vuông. Vậy có cách nào để tính diện tích một hình giới hạn bởi các đường cong bất kì không? Có đấy. Chúng ta sẽ tính xấp xỉ bằng tổng diện tích các hình chữ nhật.

Ví dụ với hàm số  $f(x) = -x^2 + 4$  ở trên, ta chia đoạn  $[a, b]$  thành  $n$  phần bằng nhau

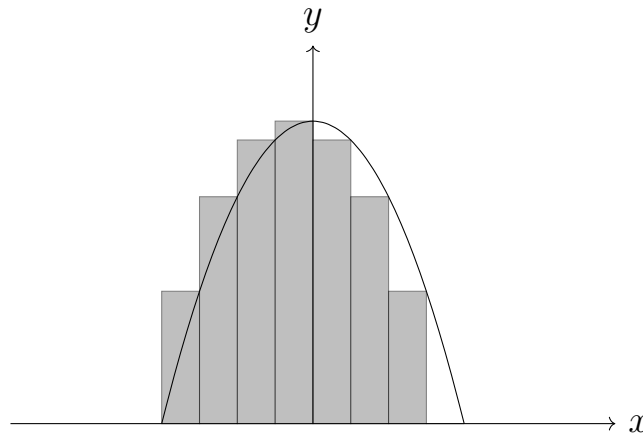
$$a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$$



Hình 10.11. Tích phân từ -2 tới 2 của  $f(x) = -x^2 + 4$

Trong đó  $x_{i+1} - x_i$  cố định và bằng  $\frac{b-a}{n}$ .

Đối với hình 10.12 ta xấp xỉ bằng 7 hình chữ nhật. Đối với hình 10.13 ta xấp xỉ bằng 15 hình chữ nhật. Đối với hình 10.14 ta xấp xỉ bằng 31 hình chữ nhật.

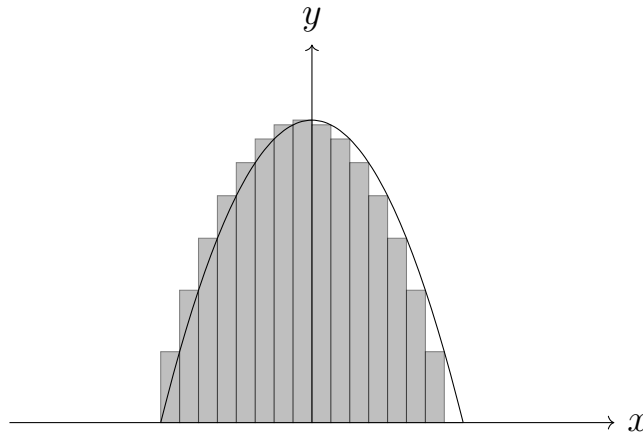


Hình 10.12. Xấp xỉ diện tích bởi 7 hình chữ nhật

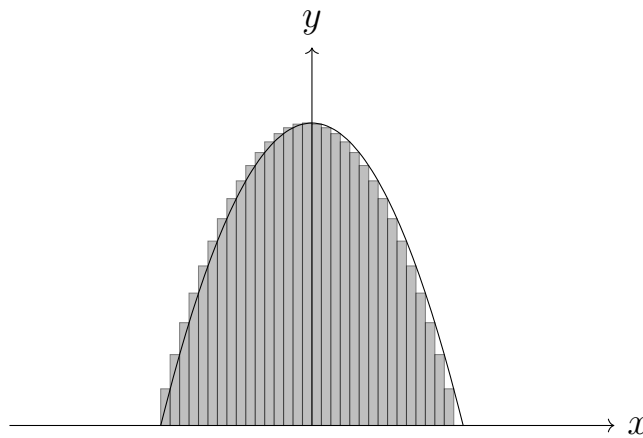
Càng dùng nhiều hình chữ nhật, tổng diện tích của chúng càng gần với diện tích cần tìm, hay là tích phân cần tìm.

Ở ba hình trên, mỗi hình chữ nhật trong đó có chiều rộng bằng nhau là  $\frac{b-a}{n}$  với  $n$  là số đoạn. Chiều dài là  $f(x_i)$  với  $x_i = a + \frac{b-a}{n}i$ ,  $i = 1, 2, \dots, n$  (biên sau).

Cụ thể hơn, hình chữ nhật từ  $x_{i-1}$  tới  $x_i$  sẽ có chiều dài là  $f(x_i)$  và chiều rộng là  $\frac{b-a}{n}$ .



Hình 10.13. Xấp xỉ diện tích bởi 15 hình chữ nhật



Hình 10.14. Xấp xỉ diện tích bởi 31 hình chữ nhật

Khi đó, tổng diện tích của các hình chữ nhật là

$$\sum_{i=1}^n (x_i - x_{i-1})f(x_i) = \sum_{i=1}^n \frac{b-a}{n} f(x_i) \quad (10.12)$$

Khi số lượng hình chữ nhật tăng lên tới vô hạn thì tổng diện tích sẽ tiến tới diện tích chính xác của hình cần tìm, hay nói cách khác là tích phân. Do đó kết quả sẽ là

$$\int_a^b f(x) dx = \lim_{n \rightarrow \infty} \sum_{i=1}^n \frac{b-a}{n} f(x_i), \quad x_i = a + \frac{b-a}{n}i \quad (10.13)$$

**Ví dụ tính tích phân qua tổng Riemann**

Ví dụ, tính tích phân từ -2 tới 2 của hàm số  $f(x) = -x^2 + 4$  ở trên. Ta có  $b = 2$  và  $a = -2$  nên

$$\begin{aligned}\frac{b-a}{n}f(x_i) &= \frac{4}{n} \left( - \left( -2 + \frac{4}{n}i \right)^2 + 4 \right) \\ &= \frac{4}{n} \left( -4 + \frac{16}{n}i - \frac{16}{n^2}i^2 + 4 \right) \\ &= \frac{64}{n} \left( \frac{i}{n} - \frac{i^2}{n^2} \right)\end{aligned}$$

Tính tổng  $i$  từ 1 tới  $n$  ta có  $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ .

Tính tổng  $i^2$  từ 1 tới  $n$  ta có  $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$ .

Suy ra

$$\begin{aligned}\sum_{i=1}^n \frac{64}{n} \left( \frac{i}{n} - \frac{i^2}{n^2} \right) &= \frac{64}{n^2} \sum_{i=1}^n i - \frac{64}{n^3} \sum_{i=1}^n i^2 \\ &= -\frac{64}{n^2} \cdot \frac{n(n+1)}{2} - \frac{64}{n^3} \cdot \frac{n(n+1)(2n+1)}{6}\end{aligned}$$

Khi  $n$  tiến tới vô cực thì biểu thức trên tiến tới  $\frac{64}{2} - \frac{64 \cdot 2}{6} = \frac{32}{3}$ . Đây chính

là giá trị của tích phân  $\int_{-2}^2 (-x^2 + 4) dx$ .



# Chương 11

## Hình học affine

### 11.1 Không gian affine

#### Không gian affine

##### Định nghĩa 1. Không gian affine

Cho  $\mathcal{V}$  là một không gian vector trên trường  $\mathbb{F}$ , và  $\mathcal{A}$  là một tập khác rỗng mà các phần tử của nó gọi là **điểm**. Giả sử có ánh xạ  $\varphi$

$$\begin{aligned}\varphi : \mathcal{A} \times \mathcal{A} &\rightarrow \mathcal{V} \\ (M, N) &\rightarrow \varphi(M, N)\end{aligned}$$

thỏa mãn hai điều kiện sau

1. Với mọi điểm  $M \in \mathcal{A}$  và vector  $\vec{v} \in \mathcal{V}$  có duy nhất một điểm  $N \in \mathcal{A}$  sao cho  $\varphi(M, N) = \vec{v}$ ;
2. Với ba điểm  $M, N, P$  bất kì ta luôn có

$$\varphi(M, N) + \varphi(N, P) = \varphi(M, P).$$

Ta nói  $\mathcal{A}$  là một **không gian affine**.

Tên gọi đầy đủ:  $\mathcal{A}$  là **không gian affine trên trường  $\mathbb{F}$  liên kết với không gian vector  $\mathcal{V}$  bởi ánh xạ liên kết  $\varphi$** .

Khi đó,  $\mathcal{V}$  được gọi là **không gian vector liên kết** với (hay **không gian nền**) của  $\mathcal{A}$  và được ký hiệu là  $\vec{\mathcal{A}}$ .

$\varphi$  được gọi là **ánh xạ liên kết**. Ta ký hiệu  $\varphi(M, N) = \overrightarrow{MN}$  từ đây về sau. Khi đó hai điều kiện trên được viết lại là

1. Với mọi  $M \in \mathcal{A}$ , với mọi  $\vec{v} \in \mathcal{V}$ , tồn tại duy nhất  $N \in \mathcal{A}$  sao cho  $\overrightarrow{MN} = \vec{v}$
2. Với mọi  $M, N, P$  thuộc  $\mathcal{A}$ ,  $\overrightarrow{MN} + \overrightarrow{NP} = \overrightarrow{MP}$

Biểu thức ở điều kiện 2 còn được gọi là **hệ thức Chales**.

Nếu  $\mathbb{F} = \mathbb{R}$  thì ta gọi là không gian affine thực.

Nếu  $\mathbb{F} = \mathbb{C}$  thì ta gọi là không gian affine phức.

Nếu muốn nhấn mạnh trường  $\mathbb{F}$  ta nói là  $\mathbb{F}$ -không gian affine.

Ta ký hiệu một không gian affine là bộ  $(\mathcal{A}, \vec{\mathcal{A}}, \varphi)$ . Ta cũng có thể ghi tắt là  $\mathcal{A}(\mathbb{F})$  hoặc chỉ là  $\mathcal{A}$ .

Nếu  $\vec{\mathcal{A}}$  là không gian vector  $n$  chiều thì ta nói  $\mathcal{A}$  là không gian affine  $n$  chiều và ký hiệu là  $\mathcal{A}^n$ . Như vậy

$$\dim \mathcal{A} = \dim \vec{\mathcal{A}}$$

**Ví dụ 1.** Xét tập hợp các điểm trong không gian  $\mathbb{R}^3$  học ở THPT. Khi đó  $\mathcal{A} = \mathbb{R}^3$  là tập hợp các điểm,  $\vec{\mathcal{A}}$  là tập hợp các vector trong  $\mathbb{R}^3$ . Một vector từ điểm  $A$  tới điểm  $B$  (theo nghĩa hình học) là một đoạn thẳng có hướng nối từ  $A$  tới  $B$ .

**Lưu ý.** Ở THPT chúng ta học rằng tọa độ của một điểm  $M$  cũng chính là tọa độ của vector  $\vec{OM}$ . Tuy nhiên điều đó không phải lúc nào cũng đúng. Ở các phần sau sẽ giải thích lý do tại sao.

### Tính chất của không gian affine

Với mọi  $M, N, Q$  thuộc  $\mathcal{A}$  ta có

1.  $\vec{MN} = \vec{0}$  khi và chỉ khi  $M \equiv N$
2.  $\vec{MN} = -\vec{NM}$
3.  $\vec{MN} = \vec{PQ}$  khi và chỉ khi  $\vec{MP} = \vec{NQ}$
4.  $\vec{MN} = \vec{PN} - \vec{PM}$

*Chứng minh.* Để chứng minh các tính chất trên ta sử dụng hai điều kiện trong định nghĩa không gian affine (đặc biệt là hệ thức Chales).

1. Nếu  $M \equiv N$  thì  $\vec{MM} = \vec{MN} + \vec{NM} = \vec{MM} + \vec{MM}$ . Suy ra  $\vec{MM} = \vec{0}$  hay  $\vec{MN} = \vec{0}$ . Từ đây, nếu  $\vec{MN} = \vec{0}$  thì theo điều kiện 1 trong định nghĩa, tồn tại duy nhất điểm  $N$  thỏa  $\vec{MN} = \vec{0}$ . Điều này tương đương với  $M \equiv N$ .

2. Từ hệ thức Chales ta có

$$\vec{0} = \vec{MM} = \vec{MN} + \vec{NM} \Leftrightarrow \vec{MN} = -\vec{NM}$$

3.  $\vec{MN} = \vec{MP} + \vec{PN}$  và  $\vec{PQ} = \vec{PN} + \vec{NQ}$  nên  $\vec{MP} + \vec{PN} = \vec{PN} + \vec{NQ}$ , hay  $\vec{MP} = \vec{NQ}$ .

4.  $\vec{PM} + \vec{MN} = \vec{PN}$ , chuyển vế  $\vec{PM}$  ta có điều phải chứng minh.  $\square$

## Phẳng

Ở THPT ta có điểm tương đương 0-phẳng, đường thẳng tương đương 1-phẳng, mặt phẳng tương đương 2-phẳng.

Trong mặt phẳng  $Oxy$ , một đường thẳng được xác định khi biết một điểm thuộc nó và một vector chỉ phương  $\vec{v} \neq \vec{0}$ . Khi đó đường thẳng đi qua  $P$  nhận  $\vec{v}$  làm vector chỉ phương là tập hợp các điểm  $M \in \mathbb{R}^2$  sao cho  $\overrightarrow{PM}$  cùng phương  $\vec{v}$ . Nói cách khác

$$d = \{M \in \mathbb{R}^2 : \overrightarrow{PM} = a\vec{v}, a \in \mathbb{R}\}$$

Trong không gian  $Oxyz$ , tương tự một đường thẳng xác định khi biết một điểm thuộc nó và một vector chỉ phương  $\vec{v}$  tương ứng

$$d = \{M \in \mathbb{R}^3 : \overrightarrow{PM} = a\vec{v}, a \in \mathbb{R}\}$$

Một mặt phẳng trong  $\mathbb{R}^3$  xác định khi biết một điểm thuộc nó và một cặp vector chỉ phương  $\vec{u}, \vec{v}$  của nó

$$\alpha = \{M \in \mathbb{R}^3 : \overrightarrow{PM} = a\vec{u} + b\vec{v}, a, b \in \mathbb{R}\}$$

Trong hình học affine ta mở rộng các khái niệm phẳng trên.

### Định nghĩa 2. Phẳng

Cho không gian affine  $(\mathcal{A}, \vec{\mathcal{A}}, \varphi)$ ,  $P$  là một điểm thuộc  $\mathcal{A}$  và  $\vec{\alpha}$  là một không gian vector con của  $\vec{\mathcal{A}}$ . Khi đó tập hợp

$$\alpha = \{M \in \mathcal{A} : \overrightarrow{PM} \in \vec{\alpha}\}$$

được gọi là **phẳng** đi qua  $P$  với (không gian chỉ) phương  $\vec{\alpha}$ .

Nếu  $\dim \vec{\alpha} = m$  thì ta nói  $\alpha$  là một **phẳng**  $m$  **chiều** hay một  $m$ -phẳng và viết  $\dim \alpha = m$ . Như vậy

$$\dim \alpha = \dim \vec{\alpha}$$

Theo cách gọi thông thường, 1-phẳng là đường thẳng, 2-phẳng là mặt phẳng. **Siêu phẳng** là tên gọi của phẳng có đối chiều 1, tức là nếu số chiều của không gian là  $n$  thì số chiều của siêu phẳng là  $n - 1$ .

**Nhận xét 1**

1. Nếu  $\alpha$  là phẳng đi qua  $P$  thì  $P \in \alpha$  và với mọi  $M, N$  thuộc  $\alpha$  ta có  $\overrightarrow{MN} = \overrightarrow{PN} - \overrightarrow{PM}$  cũng thuộc  $\alpha$ ;
2. 0-phẳng là tập chỉ gồm một điểm. Do đó ta có thể xem một điểm là một 0-phẳng;
3. Điểm  $P$  trong định nghĩa không có vai trò quan trọng gì. Mọi điểm  $P$  trong  $\alpha$  đều có ý nghĩa như nhau;
4. Giả sử  $\alpha$  là phẳng đi qua  $P$  với phương  $\vec{\alpha}$ ,  $\beta$  là phẳng đi qua  $Q$  với phương  $\vec{\beta}$ . Khi đó  $\alpha \subset \beta$  khi và chỉ khi  $P \in \beta$  và  $\vec{\alpha} \subset \vec{\beta}$ . Suy ra  $\alpha \equiv \beta$  khi  $P \in \beta$  (hay  $Q \in \alpha$ ) và  $\vec{\alpha} \equiv \vec{\beta}$ ;
5. Nếu  $\alpha$  là phẳng với phương  $\vec{\alpha}$  thì  $\alpha$  được gọi là không gian affine liên kết với  $\vec{\alpha}$  bởi ánh xạ liên kết

$$\varphi_{\alpha \times \alpha} : \alpha \times \alpha \rightarrow \vec{\alpha}$$

Vì vậy ta có thể xem phẳng là không gian affine con.

Để xác định đường thẳng ta chỉ cần biết một vector chỉ phương là đủ. Để xác định mặt phẳng ta chỉ cần biết hai vector không song song của mặt phẳng đó là đủ. Tổng quát, để xác định phương  $\vec{\alpha}$  của  $m$ -phẳng  $\alpha$  ta chỉ cần biết một cơ sở là đủ.

Từ định nghĩa của không gian vector (tập sinh) ta thấy rằng một  $m$ -phẳng chỉ có một không gian chỉ phương duy nhất, nhưng có thể có nhiều cơ sở khác nhau.

## Độc lập affine và phụ thuộc affine

**Định nghĩa 3**

Hệ  $m + 1$  điểm  $\{A_0, A_1, \dots, A_m\}$  ( $m \geq 1$ ) của không gian affine  $\mathcal{A}$  được gọi là **độc lập affine** nếu hệ  $m$  vector

$$\{\overrightarrow{A_0A_1}, \overrightarrow{A_0A_2}, \dots, \overrightarrow{A_0A_m}\}$$

của  $\vec{\mathcal{A}}$  là một hệ vector độc lập tuyến tính.

Hệ điểm không độc lập tuyến tính được gọi là **phụ thuộc affine**.

Chúng ta có một số lưu ý từ định nghĩa.

1. Tập chỉ gồm một điểm  $A_0$  bất kì được quy ước là luôn độc lập;

2. Trong định nghĩa trên điểm  $A_0$  bình đẳng như các điểm khác vì nếu hệ

$$\{\overrightarrow{A_0A_1}, \overrightarrow{A_0A_2}, \dots, \overrightarrow{A_0A_m}\}$$

độc lập affine thì hệ

$$\{\overrightarrow{A_iA_0}, \dots, \overrightarrow{A_iA_{i-1}}, \overrightarrow{A_iA_{i+1}}, \dots, \overrightarrow{A_iA_m}\}$$

cũng độc lập affine;

3. Hệ  $\{A_0, \dots, A_m\}$  phụ thuộc affine thì hệ

$$\{\overrightarrow{A_0A_1}, \dots, \overrightarrow{A_0A_m}\}$$

phụ thuộc affine;

4. Hệ con của một hệ độc lập thì độc lập, nhưng hệ con của một hệ phụ thuộc chưa chắc phụ thuộc.

Ta sẽ chứng minh lưu ý thứ hai.

*Chứng minh.* Ta xét tổ hợp tuyến tính

$$\lambda_1 \overrightarrow{A_0A_1} + \lambda_2 \overrightarrow{A_0A_2} + \dots + \lambda_m \overrightarrow{A_0A_m}$$

Do hệ vector độc lập tuyến tính nên  $\lambda_1 = \dots = \lambda_m = 0$ . Khi đó ta khai triển về trái

$$\begin{aligned} & \lambda_1 \overrightarrow{A_0A_1} + \lambda_2 \overrightarrow{A_0A_2} + \dots + \lambda_m \overrightarrow{A_0A_m} \\ &= \lambda_1 (\overrightarrow{A_iA_1} - \overrightarrow{A_iA_0}) + \lambda_2 (\overrightarrow{A_iA_2} - \overrightarrow{A_iA_0}) + \dots \\ & \quad + \lambda_{i-1} (\overrightarrow{A_iA_{i-1}} - \overrightarrow{A_iA_0}) - \lambda_i \overrightarrow{A_iA_0} + \lambda_{i+1} (\overrightarrow{A_iA_{i+1}} - \overrightarrow{A_iA_0}) \\ & \quad + \lambda_m (\overrightarrow{A_iA_m} - \overrightarrow{A_iA_0}) \\ &= \lambda_1 \overrightarrow{A_iA_1} + \lambda_2 \overrightarrow{A_iA_2} + \dots + \lambda_{i-1} \overrightarrow{A_iA_{i-1}} + \lambda_{i+1} \overrightarrow{A_iA_{i+1}} + \dots \\ & \quad + \lambda_m \overrightarrow{A_iA_m} - (\lambda_1 + \lambda_2 + \dots + \lambda_m) \overrightarrow{A_iA_0} = \vec{0} \end{aligned}$$

Do  $\lambda_1 = \dots = \lambda_m = 0$  nên tổ hợp tuyến tính ứng với các vector  $\overrightarrow{A_iA_j}$  ( $j \neq i$ ) độc lập tuyến tính và ta có điều phải chứng minh.  $\square$

### Định lý 1

Trong không gian affine  $n$  chiều  $\mathcal{A}^n$ , với  $0 < m \leq n + 1$ , luôn tồn tại các hệ  $m$  điểm độc lập. Mọi hệ gồm hơn  $n + 1$  điểm đều phụ thuộc.

## Giao của các phẳng. Bao affine

Cho  $\{\alpha_i : i \in I\}$  là một họ không rỗng các phẳng trong không gian affine  $\mathcal{A}$ .

### Định lý 2

Nếu  $\bigcap_{i \in I} \alpha_i \neq \emptyset$  thì  $\bigcap_{i \in I} \alpha_i$  là một phẳng có phương  $\bigcap_{i \in I} \vec{\alpha}_i$ .

*Chứng minh.* Vì  $\bigcap_{i \in I} \alpha_i \neq \emptyset$  nên tồn tại  $P \in \bigcap_{i \in I} \alpha_i$ , hay  $P \in \alpha_i$  với  $i \in I$ .

Nếu  $M \in \bigcap_{i \in I}$  thì  $M \in \alpha_i$  với  $i \in I$ . Suy ra  $\overrightarrow{PM} \in \alpha_i$ . Do đó

$$\bigcap_{i \in I} \alpha_i = \{M \in \mathcal{A} : \overrightarrow{PM} \in \bigcap_{i \in I} \vec{\alpha}_i\}$$

Điều này nghĩa là  $\bigcap_{i \in I} \alpha_i$  là phẳng đi qua  $P$  với không gian chỉ phương là  $\bigcap_{i \in I} \vec{\alpha}_i$ . □

### Định nghĩa 4. Phẳng giao

Phẳng  $\bigcap_{i \in I} \alpha_i$  trong định lý trên được gọi là **phẳng giao** của các phẳng  $\alpha_i$ .

Từ định nghĩa trên ta thấy rằng  $\bigcap_{i \in I} \alpha_i$  là phẳng lớn nhất (theo quan hệ bao hàm) chứa trong tất cả các phẳng  $\alpha_i, i \in I$ .

### Định nghĩa 5. Bao affine

Cho  $X$  là một tập con khác rỗng của không gian affine  $\mathcal{A}$ . Khi đó giao của mọi phẳng chứa  $X$  trong  $\mathcal{A}$  sẽ là một phẳng, gọi là **bao affine** của  $X$ , ký hiệu là  $\langle X \rangle$ .

Như vậy, bao affine  $\langle X \rangle$ , theo quan hệ bao hàm, của tập  $X$  là phẳng bé nhất chứa  $X$ .

Tương tự phép giao và hợp của hai tập hợp, chúng ta có phép giao các phẳng ở trên và phép tổng của các phẳng sẽ đề cập sau đây.

**Định nghĩa 6. Phẳng tổng**

Cho  $\{\alpha_i : i \in I\}$  là một họ không rỗng các phẳng. Bao affine của tập hợp  $\bigcup_{i \in I} \alpha_i$  được gọi là **phẳng tổng** (hay **tổng**) của các phẳng  $\alpha_i$ , ký hiệu là  $\sum_{i \in I} \alpha_i$ .

Như vậy, phẳng tổng là phẳng bé nhất chứa tất cả các phẳng  $\alpha_i$ ,  $i \in I$ .

Ta có nhận xét sau. Nếu  $X$  là một hệ hữu hạn điểm  $X = \{P_0, P_1, \dots, P_m\}$  thì tổng  $P_0 + P_1 + \dots + P_m$  (ta xem các  $P_i$  là các 0-phẳng) là phẳng có số chiều bé nhất đi qua các điểm này. Hơn nữa

$$\dim(P_0 + P_1 + \dots + P_m) = \text{rank}\{\overrightarrow{P_0P_1}, \overrightarrow{P_0P_2}, \dots, \overrightarrow{P_0P_m}\}$$

Do đó hệ điểm  $\{P_0, P_1, \dots, P_m\}$  độc lập thì  $\dim(P_0 + P_1 + \dots + P_m) = m$ .

*Chứng minh.* Đặt  $I = P_0 + P_1 + \dots + P_m$  là phẳng tổng của hệ điểm

$$\{P_0, P_1, \dots, P_m\}$$

Khi đó  $I$  đi qua các điểm  $P_0, P_1, \dots, P_m$ .

Đặt  $\alpha_i$  là phẳng đi qua  $P_0$  và  $P_i$ ,  $i = 1, 2, \dots, m$ . Khi đó  $\alpha_i$  có phương là  $\overrightarrow{P_0P_i}$ . Tổng  $I$  chính là tổng các phẳng  $\alpha_1 + \alpha_2 + \dots + \alpha_m$ , và  $\overrightarrow{P_0P_1}, \overrightarrow{P_0P_2}, \dots, \overrightarrow{P_0P_m}$  là các vector chỉ phương của nó. Như vậy nếu  $\vec{I}$  là không gian chỉ phương của  $I$  thì nó gồm các vector độc lập tuyến tính  $\overrightarrow{P_0P_{i_1}}, \overrightarrow{P_0P_{i_2}}, \dots, \overrightarrow{P_0P_{i_k}}$ . Khi đó  $\dim I = \dim \vec{I} = k = \text{rank}\{\overrightarrow{P_0P_1}, \overrightarrow{P_0P_2}, \dots, \overrightarrow{P_0P_m}\}$ . Từ đây ta có điều phải chứng minh.  $\square$

**Định lí 3**

Cho  $\alpha$  và  $\beta$  là hai phẳng. Nếu  $\alpha \cap \beta \neq \emptyset$  thì với mọi  $P \in \alpha$  và với mọi  $Q \in \beta$  ta có  $\overrightarrow{PQ} = \vec{\alpha} + \vec{\beta}$ .

Ngược lại nếu có điểm  $P \in \alpha$  và  $Q \in \beta$  sao cho  $\overrightarrow{PQ} = \vec{\alpha} + \vec{\beta}$  thì  $\alpha \cap \beta \neq \emptyset$ .

*Chứng minh.* Giả sử  $\alpha \cap \beta \neq \emptyset$ . Khi đó tồn tại điểm  $M \in \alpha \cap \beta$ , suy ra  $M \in \alpha$  và  $M \in \beta$ . Với mọi  $P \in \alpha$  và với mọi  $Q \in \beta$  thì  $\overrightarrow{PM} \in \vec{\alpha}$  và  $\overrightarrow{MQ} \in \vec{\beta}$ . Từ đó  $\overrightarrow{PQ} = \overrightarrow{PM} + \overrightarrow{MQ} = \vec{\alpha} + \vec{\beta}$ .

Đảo lại, giả sử ta có điểm  $P \in \alpha$  và điểm  $Q \in \beta$  sao cho  $\overrightarrow{PQ} = \vec{\alpha} + \vec{\beta}$ . Khi đó tồn tại hai vector  $\vec{u}$  và  $\vec{v}$  sao cho  $\overrightarrow{PQ} = \vec{u} + \vec{v}$  với  $\vec{u} \in \vec{\alpha}$  và  $\vec{v} \in \vec{\beta}$ . Theo định nghĩa của không gian affine thì với điểm  $P \in \alpha$ , tồn tại duy nhất điểm  $M \in \alpha$  sao cho  $\overrightarrow{PM} = \vec{u}$ . Tương tự với điểm  $Q \in \beta$  tồn tại duy nhất điểm  $N \in \beta$  sao cho  $\overrightarrow{QN} = \vec{v}$ . Suy ra  $\overrightarrow{PQ} = \vec{u} + \vec{v} = \overrightarrow{PM} - \overrightarrow{QN}$ . Chuyển vế  $\overrightarrow{QN}$  ta có  $\overrightarrow{PM} = \vec{u} = \overrightarrow{PQ} + \overrightarrow{QN} = \overrightarrow{PN}$ . Điều này chỉ xảy ra khi  $M \equiv N$ , hay nói cách khác  $M$  và  $N$  thuộc  $\alpha \cap \beta$ . Như vậy  $\alpha \cap \beta \neq \emptyset$ .  $\square$

#### Định lý 4

Giả sử  $\alpha$  và  $\beta$  là hai phẳng với phương lần lượt là  $\vec{\alpha}$  và  $\vec{\beta}$ . Khi đó

1. Nếu  $\alpha \cap \beta \neq \emptyset$  thì

$$\dim(\alpha + \beta) = \dim(\alpha) + \dim(\beta) - \dim(\alpha \cap \beta)$$

2. Nếu  $\alpha \cap \beta = \emptyset$  thì

$$\dim(\alpha + \beta) = \dim(\alpha) + \dim(\beta) - \dim(\vec{\alpha} \cap \vec{\beta}) + 1$$

*Chứng minh.* 1. Nếu  $\alpha \cap \beta \neq \emptyset$  thì theo định lý 11.1 ta có  $\alpha \cap \beta$  là một phẳng có phương  $\vec{\alpha} \cap \vec{\beta}$ . Lấy  $P \in \alpha \cap \beta$  và gọi  $\gamma$  là phẳng đi qua  $P$  với phương  $\vec{\gamma} = \vec{\alpha} + \vec{\beta}$ . Ta có  $\alpha \subset \gamma$  và  $\beta \subset \gamma$ . Ngoài ra nếu có phẳng  $\gamma'$  chứa  $\alpha$  và  $\beta$  thì  $P \in \gamma'$  và phương của  $\gamma'$  phải chứa  $\vec{\alpha}$  và  $\vec{\beta}$ . Nói cách khác  $\gamma \subset \gamma'$ . Vậy  $\gamma$  là phẳng bé nhất chứa  $\alpha$  và  $\beta$ , tức là  $\gamma = \alpha + \beta$ . Do đó

$$\begin{aligned} \dim(\alpha + \beta) &= \dim \gamma = \dim \vec{\gamma} = \dim(\alpha + \beta) \\ &= \dim \vec{\alpha} + \dim \vec{\beta} - \dim(\vec{\alpha} \cap \vec{\beta}) \\ &= \dim \alpha + \dim \beta - \dim(\vec{\alpha} \cap \vec{\beta}) \end{aligned}$$

2. Nếu  $\alpha \cap \beta = \emptyset$  thì theo định lý 11.1, nếu ta lấy  $P \in \alpha$  và  $Q \in \beta$  thì  $\overrightarrow{PQ} \notin \vec{\alpha} + \vec{\beta}$ . Gọi  $\vec{\gamma}$  là không gian con một chiều sinh bởi  $\overrightarrow{PQ}$ , ta có  $(\vec{\alpha} + \vec{\beta}) \cap \vec{\gamma} = \{\vec{0}\}$  (các không gian vector không có vector nào chung ngoài  $\vec{0}$ ). Gọi  $\eta$  là phẳng đi qua  $P$  có không gian chỉ phương là  $\vec{\alpha} + \vec{\beta} + \vec{\gamma}$  thì ta có  $\alpha \subset \eta$  và  $\beta \subset \eta$ . Suy ra  $\alpha + \beta \subset \eta$ .

3.  $\eta'$  là một phẳng chứa  $\alpha$  và  $\beta$  thì  $P \in \eta'$  và phương  $\vec{\eta}'$  của  $\eta'$  phải chứa  $\vec{\alpha}$ ,  $\vec{\beta}$  và  $\vec{\gamma}$ . Từ đó  $\eta \subset \eta'$ . Suy ra  $\eta$  là phẳng bé nhất chứa  $\alpha$  và  $\beta$ , hay  $\eta = \alpha + \beta$ .



Do  $\dim((\vec{\alpha} + \vec{\beta}) \cap \vec{\gamma}) = 0$  nên

$$\begin{aligned}\dim(\alpha + \beta) &= \dim \eta = \dim(\vec{\alpha} + \vec{\beta} + \vec{\gamma}) \\ &= \dim \vec{\alpha} + \dim \vec{\beta} + \dim \vec{\gamma} - \dim(\vec{\alpha} \cap \vec{\beta}) \\ &= \dim \alpha + \dim \beta + 1 - \dim(\vec{\alpha} \cap \vec{\beta})\end{aligned}$$

□

## Vị trí tương đối

### Định nghĩa 7. Cắt nhau, chéo nhau, song song

Hai phẳng  $\alpha$  và  $\beta$  được gọi là **cắt nhau cấp  $r$**  nếu  $\alpha \cap \beta$  là một  $r$ -phẳng. Chúng được gọi là **chéo nhau cấp  $r$**  nếu  $\alpha \cap \beta = \emptyset$  và  $\dim(\vec{\alpha} \cap \vec{\beta}) = r$ . Chúng được gọi là **song song** (với nhau) nếu  $\vec{\alpha} \subset \vec{\beta}$  hoặc  $\vec{\beta} \subset \vec{\alpha}$ .

Theo định lý về dim bên trên, trong  $\mathbb{R}^3$  không tồn tại hai mặt phẳng chéo nhau cấp 0 hoặc 1.

### Định lý 5

Cho hai phẳng song song  $\alpha$  và  $\beta$ . Nếu  $\alpha \cap \beta \neq \emptyset$  thì  $\alpha \subset \beta$  hoặc  $\beta \subset \alpha$ .

*Chứng minh.* Do  $\alpha$  và  $\beta$  có điểm chung nên  $\alpha \cap \beta$  là một phẳng có phương  $\vec{\alpha} \cap \vec{\beta}$ . Theo định nghĩa về sự song song,  $\alpha$  song song  $\beta$  dẫn tới  $\vec{\alpha} \subset \vec{\beta}$  hoặc  $\vec{\beta} \subset \vec{\alpha}$ . Nếu  $\vec{\alpha} \subset \vec{\beta}$  thì  $\vec{\alpha} \cap \vec{\beta} = \vec{\alpha}$ . Suy ra  $\alpha \cap \beta = \alpha$  hay  $\alpha \subset \beta$ . Trường hợp  $\vec{\beta} \subset \vec{\alpha}$  tương tự. □

### Định lý 6

Qua một điểm  $A$  có một và chỉ một  $m$ -phẳng song song với  $m$ -phẳng  $\alpha$  đã cho.

*Chứng minh.* Gọi  $\beta$  là  $m$ -phẳng đi qua  $A$  với phương là  $\vec{\alpha}$ . Khi đó  $\beta$  là phẳng  $m$  chiều song song với  $\alpha$ . Nếu  $\beta'$  cũng là  $m$ -phẳng đi qua  $A$  và song song với  $\alpha$  thì  $\vec{\beta}' = \vec{\beta} = \vec{\alpha}$ . Do  $\beta$  và  $\beta'$  có điểm chung nên theo định lý 11.1 ta có  $\beta \equiv \beta'$  □

**Định lí 7**

Trong không gian affine  $n$  chiều  $\mathcal{A}^n$  cho một siêu phẳng  $\alpha$  và một  $m$ -phẳng  $\beta$  ( $1 \leq m \leq n-1$ ). Khi đó  $\alpha$  và  $\beta$  hoặc song song hoặc cắt nhau theo một  $(m-1)$ -phẳng.

**Mục tiêu và tọa độ affine****Định nghĩa 8. Mục tiêu affine**

Cho  $\mathcal{A}^n$  là một không gian affine  $n$  chiều. Hệ  $\{O, \vec{e}_1, \vec{e}_2, \dots, \vec{e}_n\}$  gồm một điểm  $O \in \mathcal{A}^n$  và một cơ sở  $\{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n\}$  của  $\vec{\mathcal{A}}^n$  được gọi là **mục tiêu affine** (hay **mục tiêu**) của  $\mathcal{A}^n$ .

Điểm  $O$  được gọi là **gốc**, vector  $\vec{e}_i$  được gọi là **vector cơ sở thứ  $i$** ,  $i = 1, 2, \dots, n$ .

Giả sử  $\{O, \vec{e}_i\}$  là một mục tiêu của không gian affine  $\mathcal{A}^n$ . Khi đó với mọi  $M \in \mathcal{A}^n$ , vector  $\vec{OM} \in \vec{\mathcal{A}}^n$  nên ta có biểu diễn tuyến tính của  $\vec{OM}$  qua các cơ sở  $\{\vec{e}_i\}$

$$\vec{OM} = \sum_{i=1}^n x_i \vec{e}_i$$

Nhắc lại đại số tuyến tính, lúc này vector  $\vec{OM}$  có tọa độ  $(x_1, x_2, \dots, x_n)$  đối với cơ sở  $\{\vec{e}_i\}$ ,  $x_i \in \mathbb{F}$ ,  $i = 1, 2, \dots, n$ .

Khi đó bộ  $(x_1, x_2, \dots, x_n)$  được gọi là **tọa độ** của  $M$  trong mục tiêu  $\{O, \vec{e}_i\}$  và  $x_i$  được gọi là **tọa độ thứ  $i$** . Ta ký hiệu tọa độ của  $M$  là  $M(x_i)$  hoặc  $(x_i)$ .

Giả sử  $M$  có tọa độ  $(x_i)$  và  $N$  có tọa độ  $(y_i)$  đối với mục tiêu  $\{\vec{e}_i\}$ . Ta có  $\vec{MN} = \vec{ON} - \vec{OM} = (y_i - x_i)$ . Như vậy  $(y_i - x_i)$  là tọa độ của vector  $\vec{MN}$  trong mục tiêu  $\{O, \vec{e}_i\}$ .

**Nhận xét 2**

1. Giả sử trên  $\mathcal{A}^n$  đã chọn được mục tiêu cố định  $\{O, \vec{e}_i\}$ . Xét ánh xạ

$$\begin{aligned}\varphi : \mathcal{A} &\rightarrow \mathbb{F}^n \\ M &\rightarrow (x_i)\end{aligned}$$

với  $(x_i)$  là tọa độ của  $M$ . Khi đó  $\varphi$  là song ánh và mỗi điểm được đồng nhất với một phần tử của  $\mathbb{F}^n$ . Nói cách khác đối tượng hình học được đồng nhất với đối tượng đại số.

2. Xét mục tiêu affine  $\{O, \vec{e}_i\}$  của  $\mathcal{A}^n$  và gọi  $E_i \in \mathcal{A}$  là các điểm sao cho  $\vec{OE}_i = \vec{e}_i$ . Khi đó hệ điểm  $\{O, E_1, E_2, \dots, E_n\}$  độc lập affine. Ngược lại, một hệ gồm  $n + 1$  điểm  $\{O, E_1, E_2, \dots, E_n\}$  độc lập affine xác định một mục tiêu affine  $\{O, \vec{e}_i\}$  với  $\vec{e}_i = \vec{OE}_i$ . Nếu ta chọn  $O = (0, \dots, 0)$  và  $E_i = (0, \dots, 0, 1, 0, \dots, 0)$  với số 1 ở vị trí  $i$  thì đây được gọi là cơ sở chính tắc.
3. Siêu phẳng đi qua  $n$  điểm độc lập  $O, E_1, E_2, \dots, E_{i-1}, E_{i+1}, \dots, E_n$  được gọi là **siêu phẳng tọa độ thứ  $i$** . Dễ thấy  $M$  thuộc siêu phẳng tọa độ thứ  $i$  khi và chỉ khi  $x_i = 0$  với  $x_i$  là tọa độ thứ  $i$  của  $M$ .

# Chương 12

# Machine Learning

## 12.1 Các thuật toán cơ sở

### Linear Regression

Giả sử ta có  $N$  điểm dữ liệu đầu vào  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  với  $\mathbf{x}_i \in \mathbb{R}^d$ . Ứng với từng điểm dữ liệu đầu vào  $\mathbf{x}_i$  ta có một đầu ra  $y_i$ . Nghĩa là ta có  $N$  cặp dữ liệu  $(\mathbf{x}_i, y_i)$ .

Mục tiêu là xây dựng hàm số  $\hat{y} = f(x_1, x_2, \dots, x_d)$  sao cho tổng sai số của  $y_i$  và  $\hat{y}_i$  là nhỏ nhất, tức là

$$\sum_{i=1}^N \|y_i - \hat{y}_i\|^2 \rightarrow \min$$

Để hàm số đạt giá trị nhỏ nhất (hoặc lớn nhất) ta tìm cực trị của hàm số và khảo sát. Tuy nhiên không phải hàm số nào cũng đạo hàm được. Một cách tiếp cận đơn giản là sử dụng hàm tuyến tính, dễ xây dựng và luôn khả vi. Ta đặt

$$\hat{y} = f(x_1, x_2, \dots, x_d) = w_0 + w_1x_1 + w_2x_2 + \dots + w_dx_d$$

Lúc này, hàm mất mát ở trên có dạng

$$\mathcal{L} = \sum_{i=1}^N \|y_i - (w_0 + w_1x_{i1} + w_2x_{i2} + \dots + w_dx_{id})\|^2$$

Bình phương chuẩn Euclid chính là bình phương của vector. Do đó dưới dấu tổng là các hàm số bình phương. Khi đạo hàm riêng theo  $w_j$  ta có

$$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_{i=1}^N 2x_{ij} \cdot [y_i - (w_0 + w_1x_{i1} + w_2x_{i2} + \dots + w_dx_{id})]$$

với  $1 \leq j \leq d$ .

Với  $j = 0$  có chút khác biệt:

$$\frac{\partial \mathcal{L}}{\partial w_0} = \sum_{i=1}^N 2 \cdot [y_i - (w_0 + w_1 x_{i1} + \dots + w_d x_{id})]$$

Ta cho các đạo hàm riêng  $\frac{\partial \mathcal{L}}{\partial w_j}$  bằng 0 thì được

$$\begin{aligned} \sum_{i=1}^N x_{ij}(w_0 + w_1 x_{i1} + w_2 x_{i2} + \dots + w_d x_{id}) &= \sum_{i=1}^N x_{ij} y_i \\ \Leftrightarrow w_0 \sum_{i=1}^N x_{ij} + w_1 \sum_{i=1}^N x_{ij} x_{i1} + w_2 \sum_{i=1}^N x_{ij} x_{i2} \\ &\quad + \dots + w_d \sum_{i=1}^N x_{ij} x_{id} = \sum_{i=1}^N x_{ij} y_i \end{aligned}$$

Bây giờ chúng ta cần biểu diễn các dấu tổng lại thành dạng đại số (ma trận, vector) vì chúng sẽ được sử dụng để nhân với vector  $\mathbf{w} = (w_0, w_1, \dots, w_d)$ .

$$\text{Ta có } \sum_{i=1}^N x_{ij} = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{Nj} \end{pmatrix}.$$

$$\text{Ta cũng có } \sum_{i=1}^N x_{ij} x_{i1} = \begin{pmatrix} x_{11} & x_{21} & \dots & x_{N1} \end{pmatrix} \cdot \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{Nj} \end{pmatrix}.$$

Cứ tương tự như vậy, ta xếp các dấu sigma thành dạng cột thì tương đương với

$$\begin{pmatrix} * & \sum_{i=1}^N x_{ij} & * \\ * & \sum_{i=1}^N x_{ij} x_{i1} & * \\ \vdots & \vdots & \vdots \\ * & \sum_{i=1}^N x_{ij} x_{id} & * \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_{11} & x_{21} & \dots & x_{N1} \\ \dots & \dots & \ddots & \dots \\ x_{1d} & x_{2d} & \dots & x_{Nd} \end{pmatrix} \cdot \begin{pmatrix} * & x_{1j} & * \\ * & x_{2j} & * \\ \vdots & \vdots & \vdots \\ * & x_{Nj} & * \end{pmatrix}$$

Ghép các cột theo thứ tự  $j$  từ 0 tới  $d$  ta có

$$\begin{aligned} & (w_0 \ w_1 \ \cdots \ w_d) \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_{11} & x_{21} & \cdots & x_{N1} \\ \cdots & \cdots & \ddots & \cdots \\ x_{1d} & x_{2d} & \cdots & x_{Nd} \end{pmatrix} \\ & \quad \times \begin{pmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \cdots & \cdots & \ddots & \cdots \\ 1 & x_{N1} & \cdots & x_{Nd} \end{pmatrix} \\ & = (y_1 \ y_2 \ \cdots \ y_N) \cdot \begin{pmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \cdots & \cdots & \ddots & \cdots \\ 1 & x_{N1} & \cdots & x_{Nd} \end{pmatrix} \end{aligned}$$

Hay nói cách khác, nếu ta đặt  $\mathbf{w} = (w_0, w_1, \dots, w_d)$  là ma trận hàng,  $\mathbf{X}$  là ma trận có các hàng là các input, thì phương trình trên được viết lại là  $\mathbf{w}\mathbf{X}^T\mathbf{X} = \mathbf{y}\mathbf{X}$ .

Nếu đặt  $\mathbf{A} = \mathbf{X}^T\mathbf{X}$  và  $\mathbf{b} = \mathbf{y}\mathbf{X}$  thì đây là hệ phương trình theo các ẩn  $w_0, w_1, \dots, w_d$ . Tuy nhiên không phải lúc nào  $\mathbf{A}$  cũng khả nghịch nên chúng ta sẽ sử dụng một khái niệm gọi là *giả nghịch đảo* để tìm nghiệm cho hệ phương trình.

Ký hiệu  $\mathbf{A}^\dagger$  là giả nghịch đảo của ma trận  $\mathbf{A}$ . Khi đó nghiệm của hệ phương trình là  $\mathbf{w} = \mathbf{b}\mathbf{A}^\dagger$ .

## K-Means clustering

Một công việc thường được quan tâm là phân loại một nhóm các đối tượng thành những nhóm nhỏ hơn theo những tiêu chí nhất định.

Tương tự như phần trước, chúng ta có  $N$  điểm dữ liệu  $\mathbf{x}_i$  thuộc  $\mathbb{R}^d$ . Ta muốn phân cụm các vector này vào những cluster (cụm) sao cho chúng gần nhau nhất (về mặt khoảng cách Euclid).

Giả sử ta muốn phân  $N$  điểm dữ liệu trên vào  $K < N$  cluster. Ta cần tìm các điểm  $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K$  là tâm của các cụm, sao cho tổng khoảng cách từ các điểm  $\mathbf{x}_i$  tới tâm cluster mà nó được phân vào là nhỏ nhất. Nghĩa là ứng với center  $\mathbf{m}_1$  ta cần tìm các điểm  $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \dots, \mathbf{x}_{i_t}$  sao cho  $\sum_{j=1}^t \|\mathbf{x}_{i_j} - \mathbf{m}_1\|^2$  nhỏ nhất. Tương tự cho các tâm khác.

Nhưng câu chuyện phức tạp ở đây là, tâm nằm ở đâu để có thể bao quát các điểm? Tâm được chọn phải có tính tổng quát, và việc phân các điểm vào cluster tương ứng với tâm thực hiện như thế nào?

Một kỹ thuật thường được sử dụng là *one-hot*. Với mỗi điểm dữ liệu  $\mathbf{x}_i$  ta thêm một label  $\mathbf{y}_i = (y_{i1}, \dots, y_{iK})$ . Điểm  $\mathbf{x}_i$  sẽ thuộc cluster  $j$  khi  $y_{ij} = 1$ , không

thuộc thì bằng 0. Như vậy chỉ có đúng một phần tử của  $\mathbf{y}_i$  bằng 1, còn lại bằng 0. Như vậy ràng buộc của  $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{iK})$  là  $y_{ij} \in \{0, 1\}$  và  $\sum_{j=1}^K y_{ij} = 1$ .

Khi đó, ta mong muốn phân các điểm  $\mathbf{x}_i$  vào cluster  $\mathbf{m}_k$  để khoảng cách tới tâm  $\mathbf{m}_k$  là ngắn nhất, hay  $\|\mathbf{x}_i - \mathbf{m}_k\|^2 \rightarrow \min$ . Thêm nữa, với cách ký hiệu  $y_{ij}$  như trên, biểu thức tương đương với

$$\|\mathbf{x}_i - \mathbf{m}_k\|^2 = y_{ik} \|\mathbf{x}_i - \mathbf{m}_k\|^2 = \sum_{j=1}^K y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|^2$$

vì điểm  $\mathbf{x}_i$  sẽ thuộc cluster  $\mathbf{m}_k$  nào đó với  $1 \leq k \leq K$ .

Sai số cho toàn bộ dữ liệu lúc này sẽ là

$$\mathcal{L}(\mathbf{Y}, \mathbf{M}) = \sum_{i=1}^N \sum_{j=1}^K y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|^2$$

Ta cần tối ưu  $\mathbf{Y}$  và  $\mathbf{M}$ . Việc tối ưu hai ma trận cùng lúc là rất khó thậm chí bất khả thi. Do đó chúng ta có một cách tiếp cận khác là luân phiên cố định một bên và tối ưu bên còn lại. Từ đó công việc được chia làm hai bước.

Bước 1. Cố định  $\mathbf{M}$ , tìm  $\mathbf{Y}$ .

Giả sử ta đã biết các center  $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K$ . Lúc này ta cần phân các điểm  $\mathbf{x}_i$  vào cluster gần nó nhất. Dễ thấy rằng center gần nó nhất sẽ có khoảng cách Euclid ngắn nhất. Do đó ta tìm  $j$  sao cho  $\|\mathbf{x}_i - \mathbf{m}_j\|^2$  đạt nhỏ nhất. Không cần thiết phải tính căn bậc hai để giảm độ phức tạp.

Bước 2. Cố định  $\mathbf{Y}$ , tìm  $\mathbf{M}$ .

Khi đã biết  $\mathbf{Y}$  tức là ta đã biết điểm nào được phân vào cluster nào. Khi đó ta cần tìm tâm cho từng cluster. Gọi  $l(\mathbf{m}_j)$  là hàm tổng bình phương khoảng cách các điểm trong cluster tới tâm  $\mathbf{m}_j$ . Nghĩa là

$$l(\mathbf{m}_j) = \sum_{i=1}^N y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|^2$$

Mục tiêu của chúng ta là tối ưu tâm  $\mathbf{m}_j$ . Do đó ta đạo hàm theo vector  $\mathbf{m}_j$  thu được  $\frac{\partial l(\mathbf{m}_j)}{\partial \mathbf{m}_j} = \sum_{i=1}^N 2y_{ij}(\mathbf{x}_i - \mathbf{m}_j)$ . Cho đạo hàm bằng 0 và biến đổi ta có

$$\begin{aligned} 2 \sum_{i=1}^N y_{ij}(\mathbf{x}_i - \mathbf{m}_j) &= 0 \\ \Leftrightarrow \mathbf{m}_j \sum_{i=1}^N y_{ij} &= \sum_{i=1}^N y_{ij} \mathbf{x}_i \\ \Leftrightarrow \mathbf{m}_j &= \frac{\sum_{i=1}^N y_{ij} \mathbf{x}_i}{\sum_{i=1}^N y_{ij}} \end{aligned}$$

Để ý rằng,  $\sum_{i=1}^N y_{ij}$  là số lượng điểm trong cluster, và  $\sum_{i=1}^N y_{ij} \mathbf{x}_i$  là tổng các điểm trong cluster. Như vậy  $\mathbf{m}_j$  là trung bình cộng các điểm trong cluster  $j$ .

---

**Algorithm 4** Thuật toán K-Means clustering
 

---

**Require:** Dữ liệu  $\mathbf{X}$  (có  $N$  điểm dữ liệu) và số cluster  $K$

**Ensure:** Các center  $\mathbf{M}$  và label  $\mathbf{y}$  cho mỗi điểm dữ liệu

1. Chọn  $K$  điểm bất kì làm các cluster ban đầu.
  2. Phân mỗi điểm dữ liệu vào cluster gần nó nhất (cố định  $M$ , tìm  $Y$ ).
  3. Nếu việc phân dữ liệu vào các cluster ở bước 2 không thay đổi so với trước đó thì dừng thuật toán.
  4. Cập nhật center mới cho mỗi cluster bằng cách lấy trung bình cộng các điểm trong cluster (cố định  $Y$ , tìm  $M$ ).
  5. Quay lại bước 2.
- 

## Gradient Descent

Trong nhiều trường hợp chúng ta thường không thể tìm nghiệm của phương trình đạo hàm để từ đó tìm các cực trị địa phương. Một phương pháp hiệu quả là gradient descent.

### Hàm một biến

Giả sử  $x^*$  là local extremum (cực trị địa phương) của hàm số  $f(x)$ . Khi đó chúng ta xây dựng dãy số  $\{x_n\}$  hội tụ về  $x^*$ . Ý tưởng thực hiện là dựa trên nhận xét, nếu  $x_n$  nằm bên phải  $x^*$  thì  $x_{n+1}$  nằm giữa  $x^*$  và  $x_n$ . Ta đã biết nếu  $x^*$  là một điểm cực trị thì  $f'(x) > 0$  với  $x > x^*$  mà  $x_n$  đi từ bên phải sang bên trái (ngược chiều  $Ox$  nên mang dấu âm). Từ đó chúng ta có công thức chung sau

$$x_{n+1} = x_n - \eta f'(x_n)$$

Trong đó  $\eta$  là một số dương nhỏ, gọi là *learning rate* (tốc độ học).

Ta chọn  $x_0$  là một điểm bất kì. Tuy nhiên việc chọn  $x_0$  cũng có thể ảnh hưởng đến tốc độ hội tụ.

Ví dụ với hàm số  $f(x) = x^2 + 5 \sin x$ . Ta có đạo hàm là  $f'(x) = 2x + 5 \cos x$ . Việc giải phương trình đạo hàm bằng 0 là điều không dễ dàng. Do đó gradient descent tỏ ra hiệu quả trong trường hợp này.

Chọn  $\eta = 0.1$  và  $x_0 = 5$ . Sau đó chọn  $\eta = 0.1$  và  $x_0 = -5$ . Ta thấy trường hợp sau tốn ít vòng lặp hơn do  $x_0 = -5$  gần điểm cực trị hơn ( $\approx -1.11$ ).



## Hàm nhiều biến

Lúc này đầu vào của hàm số là một vector  $\mathbf{x}$ . Đặt  $\nabla f(\mathbf{x})$  là đạo hàm của hàm  $f$  theo vector  $\mathbf{x}$ . Tương tự, ta xây dựng dãy vector  $\{\mathbf{x}_n\}$  hội tụ về cực trị  $\mathbf{x}^*$ . Công thức lúc này là

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta \cdot \nabla f(\mathbf{x}_n)$$

Ta đã biết đạo hàm của hàm số theo vector cũng là vector cùng cỡ. Do đó giả sử  $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$  thì đạo hàm của nó là

$$\nabla f(\mathbf{x}) = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)$$

Với ví dụ là bài toán Linear Regression, lúc này hàm mất mát là

$$\mathcal{L} = \frac{1}{2N} \sum_{i=1}^N \|y_i - \mathbf{x}_i \mathbf{w}^T\|^2 = \frac{1}{2N} \|\mathbf{y} - \mathbf{X} \mathbf{w}^T\|^2$$

Đạo hàm của hàm mất mát là

$$\nabla \mathcal{L} = \frac{1}{N} (\mathbf{w} \mathbf{X}^T - \mathbf{y}) \mathbf{X}$$

Lúc này, với vector khởi đầu  $\mathbf{w}_0$  chúng ta xây dựng dãy  $\{\mathbf{w}_n\}$  tới khi nhận được  $\mathbf{w}_n/d < \varepsilon$ , với  $d$  là độ dài vector  $\mathbf{w}$ .

## Perception Learning Algorithm

Một trong những nhiệm vụ quan trọng nhất của ML là phân loại (tiếng Anh - classification).

Perception là thuật toán phân loại cho trường hợp đơn giản nhất khi có hai lớp. Nếu ta có các điểm dữ liệu cho trước trong không gian  $d$  chiều, ta muốn tìm một siêu phẳng ( $(d-1)$ -phẳng) chia các điểm dữ liệu đó thành hai phần. Sau đó khi có một điểm dữ liệu mới ta chỉ cần bỏ nó vào bên này hoặc bên kia của siêu phẳng.

Trong dạng này, mỗi điểm dữ liệu được biểu diễn ở dạng cột của ma trận. Giả sử các điểm dữ liệu là  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ , với  $\mathbf{x}_i \in \mathbb{R}^d$ , thì ma trận dữ liệu là  $\mathbf{X} = (\mathbf{x}_1^T \ \mathbf{x}_2^T \ \dots \ \mathbf{x}_N^T)$ . Ta gọi nhãn tương ứng với  $N$  điểm dữ liệu trên là vector  $\mathbf{y} = (y_1, y_2, \dots, y_N)$  với  $y_i = 1$  nếu  $\mathbf{x}_i$  thuộc class xanh, và  $y_i = -1$  nếu  $\mathbf{x}_i$  thuộc class đỏ.

Một siêu phẳng có phương trình là

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + \dots + w_d x_d = \mathbf{w} \cdot \mathbf{x}^T$$

Một điểm thuộc nửa không gian (tạm gọi là *bên này*) đối với siêu phẳng thì  $f_{\mathbf{w}}(\mathbf{x}) < 0$ , nếu thuộc nửa bên kia thì  $f_{\mathbf{w}}(\mathbf{x}) > 0$ , nếu nằm trên phẳng thì bằng 0.

Gọi  $\text{label}(\mathbf{x})$  là nhãn của điểm  $\mathbf{x}$ . Khi đó điểm  $\mathbf{x}$  thuộc một trong hai bên của phẳng nên  $\text{label}(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x}^T)$  với  $\text{sgn}$  là hàm dấu. Ta quy ước  $\text{sgn}(0) = 1$ .

Khi một điểm bị phân loại sai class thì ta nói điểm đó bị **misclassified**. Ý tưởng của thuật toán là làm giảm thiểu số lượng điểm bị misclassified qua nhiều lần lặp. Đặt

$$J_1(\mathbf{w}) = \sum_{\mathbf{x}_i \in \mathcal{M}} (-y_i \cdot \text{sgn}(\mathbf{w} \cdot \mathbf{x}_i^T))$$

trong đó  $\mathcal{M}$  là tập các điểm bị misclassified (tập này sẽ thay đổi theo  $\mathbf{w}$ ).

Nếu  $\mathbf{x}_i$  bị misclassified thì  $y_i$  và  $\text{sgn}(\mathbf{w} \cdot \mathbf{x}_i^T)$  ngược dấu nhau. Nói cách khác,  $-y_i \cdot \text{sgn}(\mathbf{w} \cdot \mathbf{x}_i^T) = 1$ . Từ đó  $J_1(\mathbf{w})$  là hàm đếm số lượng điểm bị misclassified. Ta thấy rằng  $J_1(\mathbf{w}) \geq 0$  nên ta cần tối ưu để hàm này đạt giá trị nhỏ nhất bằng 0. Khi đó không điểm nào bị misclassified.

Tuy nhiên có một vấn đề. Hàm  $J_1(\mathbf{w})$  là hàm rời rạc (hàm  $\text{sgn}$ ) nên rất khó tối ưu vì không thể tính đạo hàm. Do đó chúng ta cần một cách tiếp cận khác, một hàm mất mát khác tốt hơn.

Nếu ta bỏ đi hàm  $\text{sgn}$  thì có hàm

$$J(\mathbf{w}) = \sum_{\mathbf{x}_i \in \mathcal{M}} (-y_i \cdot \mathbf{w} \cdot \mathbf{x}_i^T)$$

**Nhận xét.** Một điểm bị misclassified nằm càng xa biên giới (siêu phẳng) thì giá trị  $\mathbf{w} \cdot \mathbf{x}_i^T$  càng lớn, tức là hàm  $J$  đi ra xa so với giá trị nhỏ nhất. Hàm  $J$  cũng đạt min ở 0 nên ta cũng có thể dùng hàm này để loại bỏ các điểm bị misclassified.

Lúc này hàm  $J(\mathbf{x})$  khả vi nên ta có thể dùng GD hoặc SGD để tìm nghiệm cho bài toán.

Nếu xét tại một điểm thì

$$J(\mathbf{w}, \mathbf{x}_i, y_i) = -y_i \cdot \mathbf{w} \cdot \mathbf{x}_i^T \Rightarrow \frac{\partial J}{\partial \mathbf{w}} = -y_i \mathbf{x}_i$$

Khi đó quy tắc để cập nhật là  $\mathbf{w} = \mathbf{w} + \eta \cdot y_i \cdot \mathbf{x}_i$  với  $\eta$  là learning rate (thường chọn bằng 1). Nói cách khác ta đang xây dựng dãy  $\{\mathbf{w}_n\}$  hội tụ lại nghiệm bài toán với công thức  $\mathbf{w}_{n+1} = \mathbf{w}_n + \eta \cdot y_i \cdot \mathbf{x}_i$ .

Thuật toán PLA có thể được mô tả như sau:

1. Chọn ngẫu nhiên vector  $\mathbf{w}$  với  $w_i$  xấp xỉ 0.
2. Duyệt ngẫu nhiên qua các  $\mathbf{x}_i$ :
  - Nếu  $\mathbf{x}_i$  được phân lớp đúng, tức  $\text{sgn}(\mathbf{w} \cdot \mathbf{x}_i^T) = y_i$  thì ta không cần làm gì.
  - Nếu  $\mathbf{x}_i$  bị misclassified, ta cập nhật  $\mathbf{w}$  theo công thức  $\mathbf{w} = \mathbf{w} + \eta \cdot y_i \cdot \mathbf{x}_i$ .

3. Kiểm tra xem có bao nhiêu điểm bị misclassified. Nếu không còn điểm nào thì ta dừng thuật toán, ngược lại thì quay lại bước 2.

# Chương 13

## Zero Knowledge Proofs

### 13.1 Zero knowledge proof

ZKP là một protocol mật mã cho phép một bên thuyết phục bên còn lại rằng họ sở hữu những thông tin quan trọng mà không để lộ bất cứ thông tin nào về việc chứng minh.

Khi đó, bên thuyết phục được gọi là **prover**, bên đưa ra thử thách để prover chứng minh bản thân gọi là **challenger** hoặc **verifier**.

Mỗi protocol zero-knowledge phải đảm bảo ba tính chất sau:

- Completeness (tính đầy đủ): nếu mệnh đề đúng thì verifier có thể xác nhận và bị thuyết phục bởi prover;
- Soundness: nếu mệnh đề sai thì prover không thể thuyết phục verifier rằng nó đúng;
- Zero-knowledge: verifier không biết gì về tính đúng sai của mệnh đề.

Lấy một ví dụ đơn giản để xem cách hoạt động của ZKP là protocol QR<sup>1</sup>.

**Ví dụ 1.** Mệnh đề cần kiểm tra:  $x$  là thặng dư chính phương modulo  $n$ .

Public input:  $x$  và  $n$ .

Prover's (Alice) private input: số  $w$  sao cho  $x = w^2 \pmod{n}$ .

Prover  $\rightarrow$  Verifier: Alice chọn ngẫu nhiên số  $u$  từ  $\mathbb{Z}_n^*$  và gửi Bob  $y = u^2 \pmod{n}$ .

Verifier  $\rightarrow$  Prover: Bob chọn  $b \in \{0, 1\}$ .

Prover  $\rightarrow$  Verifier: Nếu  $b = 0$  thì Alice gửi  $u$  cho Bob. Nếu  $b = 1$  thì Alice gửi  $w \cdot u \pmod{n}$ .

Verification: Gọi  $z$  là số được gửi bởi Alice. Bob **chấp nhận** proof nếu  $b = 0$  và  $z^2 = y \pmod{n}$ . Nếu  $b = 1$  thì Bob chấp nhận nếu  $z^2 = xy \pmod{n}$ .

Bob chỉ biết  $x$  và  $n$  trong khi Alice biết căn bậc hai của  $x$  modulo  $n$  (thậm

---

<sup>1</sup><https://www.cs.princeton.edu/courses/archive/fall07/cos433/lec15.pdf>

chỉ factor của  $n = p \cdot q$  trong đó  $p$  và  $q$  là các số nguyên tố).

Ở đây Alice muốn thuyết phục Bob rằng mình biết căn bậc hai của  $x$  (thậm chí factor của  $n$ ).

Nếu Alice thật sự biết căn bậc hai của  $x$  là  $w$ , hay  $x = w^2 \pmod{n}$  thì Alice cần chứng minh cho Bob thấy.

1. Alice chọn số random  $u \in \mathbb{Z}_n^*$  và gửi  $y = u^2 \pmod{n}$  cho Bob.
2. Bob chọn ngẫu nhiên  $b \in \{0, 1\}$  và gửi cho Alice.

3a. Nếu  $b = 0$  thì Alice cần tính căn bậc hai của  $y$  modulo  $n$  và gửi cho Bob. Đó chính là  $u$ .

3b. Nếu  $b = 1$  thì Alice cần tính căn bậc hai của  $xy$  ( $x$  public và  $y$  được gửi trước đó). Ta có  $xy = w^2 u^2 \pmod{n}$  nên Alice cần gửi  $wu$ . Nếu Alice thật sự biết căn bậc hai của  $x$  thì có thể tính được  $wu$ .

Bob có thể kiểm tra số  $z$  được gửi tới có thỏa mãn  $z^2 = y \pmod{n}$  (nếu  $b = 0$ ) hoặc thỏa mãn  $z^2 = xy \pmod{n}$  (nếu  $b = 1$ ) hay không.

Trong ví dụ trên, ta thấy các tính chất của ZKP:

- Completeness: khi  $x$  thực sự là số chính phương modulo  $n$  và Alice có thể đưa số  $w$  sao cho  $x = w^2 \pmod{n}$  thì Bob sẽ chấp nhận với xác suất bằng 1. Điều này khá dễ thấy;
- Soundness: nếu  $x$  không là số chính phương modulo  $n$  thì Bob có thể bác bỏ chứng minh của Alice với xác suất ít nhất  $1/2$  (trong trường hợp  $b = 1$ ). Trong khi đó  $b = 0$  thì vẫn có "cơ may" đúng;
- Zero knowledge: Bob không biết bất cứ thông tin nào liên quan đến Alice nhưng Alice có thể thuyết phục Bob tin rằng mình biết căn bậc hai của  $x$ . Zero knowledge có nghĩa là trong suốt quá trình Bob không biết thêm thông tin gì hơn từ Alice.

# Tài liệu tham khảo

- [1] Euclid. *Euclid's Elements of Geometry*. Revised and corrected. Richard Fitzpatrick. ISBN: 978-0-6151-7984-1.
- [2] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An Introduction to Mathematical Cryptography*. 2014. DOI: 10.1007/978-1-4939-1711-2.
- [3] John Casey, Euclid. *The First Six Books of the Elements of Euclid*. 2007.

# Phụ lục A

## NSUCRYPTO 2021

### Problem 9. 2021-bit key

#### Đề bài

Một generator dùng pseudo-random để sinh ra một dãy bit (0 hoặc 1) từng bước một. Để bắt đầu generator, một người phải trả 1 *nsucoin* và generator sẽ sinh ngẫu nhiên một bit (dãy bit độ dài bằng 1). Khi đó, với một dãy  $S$  được sinh có độ dài  $l$ ,  $l \geq 1$ , một trong các động tác sau được thực hiện:

1. Một dãy ngẫu nhiên độ dài 4 được thêm vào  $S$ , khi đó dãy  $S'$  có độ dài  $l + 4$ . Việc này tốn 2 *nsucoin*.
2. Một dãy ngẫu nhiên độ dài  $2l$  được thêm vào  $S$ , khi đó dãy  $S'$  có độ dài  $3l$ . Việc này tốn 5 *nsucoin*.

Bob cần sinh độ dài chính xác 2021 bit. Số lượng *nsucoin* nhỏ nhất để thực hiện việc này là bao nhiêu?

#### Giải

Dễ thấy rằng khi  $l > 2$  thì sử dụng động tác thứ hai khiến độ dài dãy tăng lên rất nhanh như lại tốn thêm coin.

Như vậy khi  $l > 6$  mình sẽ chứng minh rằng động tác thứ hai hiệu quả hơn.

*Chứng minh.* Để ý rằng nếu mình sử dụng động tác thứ nhất ba lần liên tiếp, khi đó từ dãy có độ dài  $l$  mình thu được dãy mới độ dài  $l + 4 + 4 + 4 = l + 12$  và việc này tốn 6 *nsucoin*.

Trong khi đó, nếu mình dùng động tác thứ hai một lần thì từ dãy độ dài  $l$  mình thu được dãy độ dài  $3l$  và tốn 5 *nsucoin*.

Mình muốn  $3l > l + 12$  vì mình đã có  $3l$  tương ứng 5 *nsucoin* và  $l + 12$  tương ứng 6 *nsucoin*. Như vậy  $l > 6$ .

Chiến thuật lúc này là mình sẽ dùng động tác thứ hai để triple độ dài bất

cứ lúc nào có thể. Nói cách khác là khi đi ngược từ 2021 về 0 thì khi nào số chia hết cho 3, mình sẽ dùng động tác sau, không thì dùng động tác đầu.

Quá trình sẽ diễn ra theo bảng A.1.

$2021 - 4 = 2017$	2 <i>nsucoin</i>
$2017 - 4 = 2013$	2 <i>nsucoin</i>
$2013/3 = 671$	5 <i>nsucoin</i>
$671 - 4 = 667$	2 <i>nsucoin</i>
$667 - 4 = 663$	2 <i>nsucoin</i>
$663/3 = 221$	5 <i>nsucoin</i>
$221 - 4 = 217$	2 <i>nsucoin</i>
$217 - 4 = 213$	2 <i>nsucoin</i>
$213/3 = 71$	5 <i>nsucoin</i>
$71 - 4 = 67$	2 <i>nsucoin</i>
$67 - 4 = 63$	2 <i>nsucoin</i>
$63/3 = 21$	5 <i>nsucoin</i>

Bảng A.1: Tính nsucoin

Từ 21 trở đi không thể áp dụng quy tắc trên nữa vì theo chứng minh trên chiến thuật chỉ hiệu quả với  $l > 6$ .

Mình khai triển  $21 = 1 + 4 + 4 + 4 + 4 + 4$ , thực hiện 4 phép trừ (động tác đầu) tốn  $5 \cdot 2 = 10$  *nsucoin* và trừ 1 tốn 1 *nsucoin*.

Như vậy tổng số *nsucoin* nhỏ nhất cần là 47. □



# Phụ lục B

## NSUCRYPTO 2022

### Lời nói đầu

Thời kì đen tối ...

Đề thi năm 2022 khá lạ và phức tạp. Bảng điểm round 2 dành cho University student cũng ảo ma không kém.

### Problem 5\*. Super dependent S-box

#### Đề bài

Harry muốn tìm một super dependent S-box cho mã hóa mới. Anh ấy dùng một hoán vị liên kết chặt chẽ với mọi biến của nó và muốn ước lượng số các hoán vị như vậy.

Một hàm boolean vectorial  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))$ , với  $x \in \mathbb{F}_2^n$ , là một **hoán vị** trên  $\mathbb{F}_2^n$  nếu nó là một ánh xạ one-to-one từ  $\mathbb{F}_2^n$  tới  $\mathbb{F}_2^n$ .

Các hàm tọa độ  $f_k(x)$  (là các hàm boolean từ  $\mathbb{F}_2^n$  tới  $\mathbb{F}_2$ ) được gọi là *essential depend* trên các biến  $x_j$  nếu tồn tại các giá trị  $b_1, b_2, \dots, b_{j-1}, b_{j+1}, \dots, b_n \in \mathbb{F}_2$  sao cho

$$f_k(b_1, b_2, \dots, b_{j-1}, 0, b_{j+1}, \dots, b_n) \neq f_k(b_1, b_2, \dots, b_{j-1}, 1, b_{j+1}, \dots, b_n)$$

Nói cách khác, essential depend trên biến  $x_j$  nghĩa là trong dạng biểu diễn ANF (đa thức Zhegalkin) của hàm  $f$  có sự có mặt của biến  $x_j$ .

**Ví dụ 1.** Xét  $n = 3$ . Khi đó hàm boolean  $f(x_1, x_2, x_3) = x_1x_2 \oplus x_3$  essential depend trên cả ba biến, nhưng hàm  $g(x_1, x_2, x_3) = x_1x_2 \oplus x_2 \oplus 1$  chỉ essential depend trên  $x_1$  và  $x_2$ .

**Câu hỏi.** Tìm số lượng hoán vị trên  $\mathbb{F}_2^n$  mà các hàm tọa độ của nó đều essential depend trên cả  $n$  biến.

**Q1.** Tìm đáp án cho  $n = 2, 3$ .

**Q2.** Tìm đáp án cho  $n$  (special prize).

## Giải

Q1 có thể được giải với SageMath. Tuy nhiên năm đó mình không dùng SageMath mà dùng Python thuần nên đáp án sai mất 😞. Đáp án cho Q1 với  $n = 2$  là 0 (liệt kê tất cả hàm ra) và đáp án cho  $n = 3$  là 24576.

Ở Q2 các team giải ra chứng minh được rằng, đáp án sẽ là một số chia hết cho  $2^n \cdot n!$ .

## Problem 11. A long awaited event

### Đề bài

Bob nhận được một thông điệp

L78V8LC7GBEYEE

về một sự kiện quan trọng từ Alice.

Alice sử dụng một bảng chữ cái 37 ký tự gồm chữ cái từ A tới Z, số từ 0 tới 9 và dấu space. Các ký tự được encode như sau:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
S	T	U	V	W	X	Y	Z	0	1	2	3	4	5	6	7	8	9
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
SPACE																	
36																	

Để mã hóa Alice sử dụng hàm  $f$  sao cho  $f(x) = ax^2 + bx + c \pmod{37}$  với số  $a, b, c$  nào đó và hàm  $f$  thỏa tính chất

$$f(x - y) - 2f(x)f(y) + f(1 + xy) = 1 \pmod{37} \text{ với mọi số nguyên } x, y$$

**Q.** Hãy giải mã thông điệp Bob nhận được.

## Giải

Sử dụng hàm  $f$ , ta có:

1. Nếu cho  $y = 0$  thì  $f(x) - 2f(x) \cdot f(0) + f(1) = 1 \pmod{37}$  với mọi  $x \in \mathbb{Z}_{37}$ .

Điều này tương đương với  $(1 - 2f(0)) \cdot f(x) = 1 - f(1) \pmod{37}$  với mọi  $x \in \mathbb{Z}_{37}$ .

Đẳng thức trên đúng với mọi  $x \in \mathbb{Z}_{37}$  khi và chỉ khi  $1 - 2f(0) = 1 - f(1) = 0$ . Suy ra  $f(0) = 19$  và  $f(1) = 1$ .

2. Cho  $x = y = 1$  thì  $f(2) = 1 - f(0) - 2 \cdot (f(1))^2 = 21 \pmod{37}$ .

Với các cặp giá trị  $(x, f(x))$  là  $(0, 19)$ ,  $(1, 1)$  và  $(2, 21)$  thì ta tìm lại được (bằng nội suy hoặc phép thế) đa thức  $f(x) = 19x^2 + 19 \pmod{37}$ .

Khi đó, với mỗi giá trị  $f(x)$  ta có hai giá trị  $x$  thỏa mãn đẳng thức. Truy ngược ra đáp án khả thi cho ta thông điệp ban đầu.

Ciphertext		L	7	8	V	8	L	C	7	G	B	E	Y	E	E
Plain	TH 1	N	S	R	C	R	N	P	S	O	B	J	L	J	J
		13	18	17	2	17	13	15	18	14	1	9	11	9	9
	TH 2	Y	T	U	9	U	Y	W	T	X	SPC	2	0	2	2
		24	19	20	35	20	24	22	19	23	36	28	26	28	28

Bảng B.1: Giải mã

Theo bảng B.1 thì thông điệp là "NSUCRYPTO 2022".

# Phụ lục C

## NSUCRYPTO 2023

### Problem 1. Affine cipher

Đây là bài 1 của round 2 và được giải bởi bạn Chương.

#### Đề bài

Ta xét bảng chữ cái A, ..., Z,  $\alpha$ ,  $\beta$ ,  $\gamma$  có 29 chữ cái. Ta đánh số A, ..., Z từ 0 tới 25, và  $\alpha$ ,  $\beta$ ,  $\gamma$  là 26, 27, 28.

Ta sử dụng cryptosystem mã hóa từng khối 2 ký tự, gọi là bigram. Với  $x$  và  $y$  là hai ký tự của bigram, thì plaintext sẽ là  $P = 29x + y$ .

Mã hóa sử dụng biến đổi affine (giống hệ mã affine) là  $C = aP + b \pmod{841}$ .

Khi phân tích một đoạn văn bản dài, người ta phát hiện ra rằng các bigram sau xuất hiện nhiều nhất " $\beta\gamma$ ", "UM" và "LC". Đồng thời, trong tiếng Anh thì các bigram "TH", "HE" và "IN" cũng xuất hiện nhiều nhất.

**Q.** Có thể giải mã "KEUDCR" mà không cần khóa hay không? Còn key thì sao?

#### Giải

Theo thống kê các bigram xuất hiện nhiều nhất trong ciphertext và trong plaintext sẽ khớp nhau. Do đó có thể thấy "TH" mã hóa thành " $\beta\gamma$ " và "HE" mã hóa thành "LC". Như vậy ta có hệ phương trình

$$812 = a \cdot 558 + b \pmod{841}$$

$$321 = a \cdot 207 + b \pmod{841}$$

Giải hệ ta có  $a = 15$ ,  $b = 10$ . Đây là key.

Từ đây chúng ta có thể giải mã thành **CRYPTO** là plaintext ban đầu. Bài này ăn trọn 4/4 điểm.

## Problem 2. Simple ideas for primes

### Đề bài

Chúng ta xem xét một số dãy số bao gồm các số nguyên tố.

- *Số Fermat*,  $F_k = 2^{2^k} + 1$ , với  $k$  bắt đầu từ 0. Ta có các số  $F_0, F_1, F_2, F_3, F_4$  là các số nguyên tố, còn  $F_5$  thì không phải.
- *Số Mersenne*,  $M_k = 2^k - 1$ . Ta có  $M_2, M_3, M_5, M_7$  là các số nguyên tố, trong khi  $M_{11}$  là hợp số. Các số nguyên tố Mersenne là các số dạng  $2^k - 1$  với  $k$  là số nguyên tố.
- Dãy số 31, 331, 3331, 33331, 333331, 3333331, 33333331 là các số nguyên tố được xây dựng theo quy tắc trên, nhưng số 333333331 là hợp số chia hết cho 17.

Ta nói dãy Fermat trên có *sequence primality parameter* là 5, dãy Mersenne bằng 4, dãy cuối cùng bằng 7.

**Q.** Xây dựng một dãy bao gồm các số nguyên tố như vậy. Điều kiện quan trọng ở đây là các số hạng được xác định bởi chỉ số của dãy, không phụ thuộc vào các số trước nó.

### Giải

Bắt đầu với dãy Euler

$$f(n) = n^2 + n + 41$$

Đây là dãy các số nguyên tố với  $n = 0, 1, \dots, 39$  và  $f(40)$  là hợp số. Như vậy đây là dãy nguyên tố độ dài 40.

Và tất nhiên, dãy "ai cũng biết" thì chỉ được 2 điểm thôi 🙄.

Sau khi tham khảo những thí sinh khác thì có một số cách xây dựng nhằm cải tiến điều này, tham khảo từ <sup>1</sup>.

Nếu ta chuyển dãy trên thành

$$g(n) = f(n - 40) = (n - 40)^2 + (n - 40) + 41 = n^2 - 79n + 1601$$

thì thu được dãy số nguyên tố với độ dài 80. Các nhà toán học thế kỷ 20 đã chứng minh được rằng, nếu  $p(x)$  là một đa thức sinh ra dãy số nguyên tố với  $0 \leq x \leq n$  thì đa thức  $p(n - x)$  cũng vậy.

Trong bảng, dãy nguyên tố có độ dài lớn nhất là 56 được biểu diễn bởi đa thức

$$\frac{1}{4} (n^5 - 133n^4 + 6729n^3 - 158379n^2 + 1720294n - 6823316)$$

<sup>1</sup><https://mathworld.wolfram.com/Prime-GeneratingPolynomial.html>

Dựa theo lời giải của team Himanshu Sheoran, Yo Iida và Pranshu Kumar chúng ta có thể sinh một dãy có độ dài bất kì.

Lời giải dựa trên bài blog của A. W. Walker<sup>2</sup>. Bài blog này phân tích về một bài báo năm 1977 bởi Chang và Lih với tiêu đề *Polynomial Representation of Primes* nhưng hiện tại không có bản online.

Bài báo này đưa ra một phương pháp xây dựng đa thức  $F(n)$  bậc  $n$  mà với mọi  $x \in [0, n]$  thì  $F(n)$  đều là số nguyên tố.

Bài báo có thể được tóm gọn như sau. Xét đa thức

$$F(x) = 1 + \left| \sum_{n=0}^M \frac{a_n}{x - n} \prod_{j=0}^M (x - j) \right|$$

sẽ sinh ra các số nguyên tố với mọi  $x \in [0, M]$  nếu và chỉ nếu  $a_n$  phân biệt và  $(a_n \cdot M! + 1)$  là các số nguyên tố. Như vậy ta có một thuật toán đơn giản để bruteforce các đa thức trên.

---

**Algorithm 5** Thuật toán sinh dãy nguyên tố độ dài  $M$

---

**Require:** Độ dài dãy nguyên tố  $M$

**Ensure:** Đa thức  $F(x)$  cho kết quả là số nguyên tố với mọi  $x \in [0, M]$

coeffs = [ ] chứa các số hạng  $a_n$

$an \leftarrow 1$

**while** Chưa đủ  $M + 1$  số hạng trong coeffs **do**

**if**  $(an \cdot M! + 1)$  là số nguyên tố **then**

        kết nạp  $an$  vào dãy coeffs

**end if**

$an \leftarrow an + 1$

**end while**

---

## Problem 3. Mixed hashes

### Đề bài

Alice và Bob trao đổi các thông điệp mã hóa. Họ dùng thuật toán mã hóa khối PRESENT với key 80-bit và ECB mode. Ở đây, thông tin được lưu dạng ảnh .ppm.

Header của file .ppm gồm 3 dòng theo dạng **P6\nX\nY\n255**. Trong đó X và Y là kích thước của ảnh theo chiều ngang và dọc.

Để đảm bảo an toàn, header sẽ được loại bỏ trước khi encrypt. Để có thể khôi phục header, hash của header sẽ được gửi đi thay vì header. Khi đó 3 phần

---

<sup>2</sup><https://awwalker.com/2017/02/27/prime-generating-polynomials/>

của header sẽ được ngăn cách bởi dấu cách (space) thay vì newline như trên. Nghĩa là "P6 X Y 255".

Bob chuẩn bị 8 ảnh (trong file đính kèm) mà không có header. Bob encrypt 8 ảnh đó với cùng một key theo thuật toán PRESENT và ECB mode. Bob cũng gửi hash của 8 headers đi kèm. Tuy nhiên các hash đã bị trộn lẫn với nhau. Liệu chúng ta có thể khôi phục thông điệp mà Bob muốn gửi Alice?

## Giải

Bài này là bài 3 ở round 1 và round 2. Trong thời gian 2 round mình đều giải ra (round 2 chi tiết hơn và trình bày đẹp hơn :v).

Đề cho một file mẫu là mikky.ppm. Khi phân tích file này mình thấy rằng, nếu gọi  $w$  và  $h$  là độ rộng và độ cao của ảnh (lấy từ header) thì độ dài file không có header là  $3 \cdot w \cdot h$ .

Sau khi encrypt bằng thuật toán mã hóa khối với ECB mode, độ dài sẽ là  $3 \cdot w \cdot h + pd$ , trong đó  $pd$  là padding. Theo thuật toán **PRESENT** thì  $0 \leq pd \leq 8$ .

Với dự đoán rằng  $w \approx h$ , mình lấy căn bậc hai của độ dài các filel đề cho, và đưa ra dự đoán  $w, h \in [400, 600]$ . Nếu sai thì mình tăng độ rộng khoảng này thôi.

Tiếp theo, bruteforce  $w$  và  $h$  trong khoảng này, cho tới khi hash "P6 x y 255" xuất hiện trong số các hash trên, và

$$0 \leq \text{len}(\text{ciphertext}) - 3 * w * h \leq 8$$

thì mình lấy  $w$  và  $h$  này. Thế là mình có header.

Do cả 8 file được encrypt bởi cùng một key **PRESENT**, và key có 80 bit tương ứng 10 bytes, hay 10 ký tự, nhìn đề mình nhận thấy có chuỗi **P6 X Y 255** là hợp lý. Như vậy key cho PRESENT là chuỗi **P6 X Y 255**.

Cuối cùng, mình giải mã lần lượt từng file với key trên, ghép header tương ứng vào, như vậy là mình giải mã được tất cả file rồi.

Vậy thông điệp gốc là "♡Loveyou". Bài này được 5/6 điểm vì không nộp code tính toán header, mất điểm vì chủ quan.

## Problem 4\*. Column functions

### Đề bài

Alice muốn xây dựng mã đối xứng mạnh bằng việc giải một số bài toán khó.

Xét  $2^n$  hàm vectorial one-to-one đôi một khác nhau,  $G_i : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ , với  $i = 1, \dots, 2^n$ . Sử dụng các hàm này, chúng ta xây dựng một ma trận binary đặc

biệt và xác định một số tính chất của nó.

Với  $n = 2^m$ ,  $m \geq 5$ , ta định nghĩa ma trận  $M$  kích thước  $2^n \times n2^n$  theo quy tắc sau. Hàm thứ  $i$ ,  $i = 1, \dots, 2^n$ , là ghép của các giá trị  $G_i(0, 0, \dots, 0, 0)$ ,  $G_i(0, 0, \dots, 0, 1)$ , ...,  $G_i(1, 1, \dots, 1, 1)$ . Các cột của  $M$  có thể được xem như các vector của  $n2^n$  hàm boolean, mỗi hàm  $n$  biến. Ta gọi chúng là *column functions*.

Chứng minh hoặc phản bác giả thuyết sau cho ít nhất một giá trị  $m \geq 5$ : với mọi cách xây dựng ma trận như trên, tồn tại  $2^{n/2}$  columns functions  $f_1, \dots, f_{2^{n/2}}$  sao cho tồn tại một hàm boolean nonzero  $f : \mathbb{F}_2^{2^{n/2}} \rightarrow \mathbb{F}_2$  thỏa mãn các điều kiện sau:

- với mọi  $\mathbf{x} \in \mathbb{F}_2^n$

$$f(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{2^{n/2}}(\mathbf{x})) = 0;$$

- với mọi  $\mathbf{y} \in \mathbb{F}_2^{2^{n/2}}$ , giá trị  $f(\mathbf{y})$  có thể tính với không quá  $2^{n/2}$  phép cộng và phép nhân modulo 2.

**Ví dụ 1.** Với  $m = 1$  thì  $n = 2$  và ta xây dựng ma trận  $4 \times 8$ . Xét các hàm boolean vectorial one-to-one  $G_1, G_2, G_3, G_4$  từ  $\mathbb{F}_2^2$  tới  $\mathbb{F}_2^2$  xác định bởi các giá trị  $(0, 1, 2, 3)$ ,  $(0, 2, 1, 3)$ ,  $(0, 3, 1, 2)$  và  $(3, 2, 1, 0)$ . Khi đó ma trận là

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Ta cần tìm  $2^{n/2} = 2$  column functions. Gọi  $f_1$  và  $f_2$  là hàm bool ứng với cột đầu và cột thứ hai của ma trận, và  $f(x_1, x_2) = x_1 \oplus x_2$ . Khi đó  $f(f_1(\mathbf{x}), f_2(\mathbf{x})) \equiv 0$  vì  $f_1(\mathbf{x}) = f_2(\mathbf{x}) = 0$  với mọi  $\mathbf{x} \in \mathbb{F}_2^n$ .

Ta cũng thấy rằng có thể chọn  $f_1$  và  $f_2$  là cột 5 và 6 của ma trận. Khi đó, đặt  $f(x_1, x_2) = x_1 x_2$  thì  $f(f_1(\mathbf{x}), f_2(\mathbf{x})) \equiv 0$  vì  $f_1(\mathbf{x}) \neq f_2(\mathbf{x})$  với mọi  $\mathbf{x} \in \mathbb{F}_2^n$ .

Trong cả hai trường hợp ta chỉ cần đúng một phép tính. Lưu ý rằng hàm  $f$  thỏa  $f_1$  và  $f_2$  được gọi là *algebraically dependent*.

## Giải

Cách giải của mình không đúng hoàn toàn nên chỉ được 1/8. Dưới đây trình bày cách giải của đội Robin Jadoul, Jack Pope và Esrever Yu được 8/8 điểm.

Giả thuyết trong đề bài đúng với  $m$  lớn. Ý tưởng chính là chúng ta cố định các cột sẽ chọn, có một số lượng lũy thừa các hàm  $f$  (rất rất lớn) nhưng chỉ có số lượng đa thức các hàng (ít lớn hơn 🌀), do đó chúng ta có thể chọn hàm  $f$  triệt tiêu tất cả hàng.



**Định lý 1**

Cho ma trận binary  $M$  kích thước  $2^n \times n2^n$ . Với  $n + 1$  column functions bất kì, tồn tại một hàm nonzero  $f : \mathbb{F}_2^{n+1} \rightarrow \mathbb{F}_2$  triệt tiêu trên các column functions và sử dụng nhiều nhất  $2n + 1$  toán tử cộng và nhân.

*Chứng minh.* Gọi  $f_1, f_2, \dots, f_{n+1}$  là các column functions. Đặt

$$S = \{(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{n+1}(\mathbf{x})) \in \mathbb{F}_2^{n+1} : \mathbf{x} \in \mathbb{F}_2^n\}$$

là tập hợp các bộ giá trị từ các column functions. Vì  $|S| < |\mathbb{F}_2^{n+1}|$ , ta có thể chọn vector  $\mathbf{z} = (z_1, \dots, z_{n+1}) \in \mathbb{F}_2^{n+1} \setminus S$ . Từ đây ta định nghĩa hàm  $f$  là

$$f(x_1, \dots, x_{n+1}) = (x_1 \oplus (z_1 \oplus 1)) \cdot (x_2 \oplus (z_2 \oplus 1)) \cdots (x_{n+1} \oplus (z_{n+1} \oplus 1))$$

Lúc này,  $f$  cho output là 1 chỉ khi  $\mathbf{z}$  là input (?), do đó với mọi  $\mathbf{y} \in S$  thì  $f(\mathbf{y}) = 0$ .

Đối với số lượng phép tính, với mọi  $i$ ,  $z_i \oplus 1$  là hằng số nên ta không xét đến khi tính số lượng phép tính cho hàm  $f$ . Dựa trên vector  $\mathbf{z}$ , có tối đa  $n + 1$  phép cộng (tương ứng  $x_i \oplus (z_i \oplus 1)$ ) và  $n$  phép nhân  $n + 1$  hạng tử với nhau. Vì vậy số phép tính tối đa là  $2n + 1$ .  $\square$

**Hệ quả 1.** Giả thuyết trên đúng với mọi  $m \geq 4$ .

*Chứng minh.* Theo định lý trên ta có  $f^3$ . Bây giờ ta chỉ cần chặn lại số lượng cột mà ta cần và số lượng phép tính.

Với số lượng cột, ta cần  $n + 1 \leq 2^{n/2}$ . Với số lượng phép tính, ta cần  $2n + 1 \leq 2^{n/2}$ . Kết hợp cả hai ta có  $n \geq 9$ , tương đương với  $m \geq 4$ .  $\square$

## Bình luận

Về tổng thể cách giải của team kia hợp lý trừ phần mình đánh dấu, chưa hiểu lắm khi  $\mathbf{z}$  là input nghĩa là sao.

Thêm nữa, chưa có cơ sở để nghĩ ra bài giải này. Có vẻ như Esrever đã rất căng não để giải ra 🤔.

**UPDATE.** Ý nghĩa của phần đánh dấu là: trong phép tính

$$f(x_1, \dots, x_{n+1}) = (x_1 \oplus (z_1 \oplus 1)) \cdot (x_2 \oplus (z_2 \oplus 1)) \cdots (x_{n+1} \cdot (z_{n+1} \oplus 1))$$

Kết quả phép tính ra 1 khi và chỉ khi tất cả hạng tử bằng 1. Nói cách khác  $x_i \oplus (z_i \oplus 1) = 1$  với mọi  $1 \leq i \leq n + 1$ .

<sup>3</sup>Bài toán yêu cầu hàm  $f$  lấy  $2^{n/2}$  input. Tuy nhiên chúng ta có thể thêm các dummy input miễn là số lượng cột không lớn hơn  $n + 1$  (mục tiêu ban đầu)

Điều này tương đương với  $x_i = z_i$  với mọi  $1 \leq i \leq n+1$ . Nhưng điều này vô lý vì khi đó  $\mathbf{x} \equiv \mathbf{z}$  mà hai vector này nằm ở hai tập rời nhau.

Do đó kết quả hàm  $f$  luôn là 0. Ta có điều phải chứng minh.

## Problem 5. Primes

Đây là bài 5 của round 2 và được giải bởi bạn Uyên.

### Đề bài

Marcus chọn hai số nguyên tố lớn  $p$  và  $q$  rồi tính  $n = p \cdot q$  và  $m = p + q$ . Sau đó số  $n \cdot m$  được sử dụng trong cryptosystem.

Khi test Marcus thấy các số  $p$  và  $q$  cho tích  $n \cdot m$  kết thúc bởi 2023. Điều đó khả thi không?

### Giải

Do  $p$  và  $q$  là các số nguyên tố lớn nên chúng lẻ. Suy ra  $m = p + q$  là số chẵn, nên tích  $n \cdot m$  cũng là số chẵn.

Số chẵn kết thúc bởi 0, 2, 4, 6, 8 nên không thể kết thúc bởi 3. Do đó điều này không thể xảy ra.

## Problem 6\*\*. An aggregated signature

Bài này không biết làm 🤔.

### Đề bài

Trong một tổ chức quốc tế lớn, gọi là **NSUCRYPTO association**, mọi người quyết định tổ chức một tờ báo thông tin (news journal) trong lĩnh vực mật mã. Tổ chức muốn rằng, tin tức chỉ được công bố khi đã được kiểm duyệt bởi một nhóm lớn các nhà mật mã. Vì vậy, 10 000 chuyên gia mật mã đã được mời tới làm biên tập cho tờ báo.

Quy định công bố như sau. Tin tức được công bố khi nó được ký bởi tất cả các thành viên biên tập. Tuy nhiên các nhà mật mã không rảnh để thu thập 10 000 chữ ký cá nhân (gặp mình thì mình cũng không muốn 😊). Do đó mọi người muốn một chữ ký postquantum dùng chung mà không thể chia nhỏ ra các chữ ký cá nhân.

Yêu cầu là cần xây dựng mô hình chữ ký thỏa mãn các yêu cầu sau:

- kích thước chữ ký không quá lớn, có thể hơn vài kilobytes;
- kích thước public key (để kiểm tra chữ ký) là nhỏ. Kích thước của key nên là cố định (hoặc gần cố định) kể cả khi số lượng chuyên gia tăng lên, ví dụ 20 000;
- việc kiểm tra chữ ký tốn không quá 2 phút;
- chữ ký có thể chống lại các tấn công trên máy tính lượng tử.

## Problem 7. A unique coding

Bài này khi nhìn đề thì "có vẻ" câu hỏi Q2 là trường hợp nhỏ hơn của Q1. Mình giải Q2 (không chắc đúng hoàn toàn) nên lời giải sau đây áp dụng cho cả Q1 và Q2.

### Đề bài

Xét binary error-correcting code  $\mathcal{C}$  với độ dài  $n$ . Code này chỉ đơn giản là tập con của  $\mathbb{F}_2^n$  thôi và ta truyền một phần tử của code này qua các kênh truyền.

Khi đi qua các kênh truyền các bit có thể bị sai, dẫn tới bị đảo bit. Khi nhận được vector  $\mathbf{y} \in \mathbb{F}_2^n$ , ta sẽ decode thành một phần tử thuộc  $\mathcal{C}$  mà có khoảng cách gần  $\mathbf{y}$  nhất, nói cách khác là Hamming weight gần nhất.

Xét cơ chế decode maximal-likelihood. Giả sử ta nhận được  $\mathbf{y} \in \mathbb{F}_2^n$ , ta muốn xét các trường hợp lỗi xảy ra ít nhất, gọi là  $d_{\mathbf{y}}$ , nghĩa là

$$d_{\mathbf{y}} = \min_{\mathbf{x} \in \mathcal{C}} wt(\mathbf{x}, \mathbf{y})$$

Tiếp theo, đặt  $\mathcal{D}(\mathbf{y}) = \{\mathbf{x} \in \mathcal{C} : wt(\mathbf{x}, \mathbf{y}) = d_{\mathbf{y}}\}$ . Cuối cùng ta decode  $\mathbf{y}$  thành phần tử  $\mathbf{x}$  nào đó trong  $\mathcal{D}(\mathbf{y})$ .

Chúng ta quan tâm tới các trường hợp code  $\mathcal{C}$  khiến  $|\mathcal{D}(\mathbf{y})| = 1$  với mọi  $\mathbf{y} \in \mathbb{F}_2^n$ . Nói cách khác khi nhận được  $\mathbf{y}$  bất kì của  $\mathbb{F}_2^n$  ta có thể decode thành một dạng duy nhất.

**Q1.** Code  $\mathcal{C}$  như nào thì thỏa mãn tính chất này?

**Q2.** Code  $\mathcal{C}$  như nào thỏa mãn tính chất này và là không gian tuyến tính con của  $\mathbb{F}_2^n$ ?

### Giải

Đầu tiên mình có nhận xét (khá rõ ràng) sau đây:

**Nhận xét 1**

Với mọi  $n$ , code  $\mathcal{C} \equiv \mathbb{F}_2^n$  thỏa mãn tính chất trên.

Chúng ta có thể thấy rằng với mọi  $\mathbf{y} \in \mathbb{F}_2^n$  nhận được thì sẽ decode thành chính nó trong  $\mathcal{C}$ .

Tiếp theo, ta xét nửa trên của  $\mathbb{F}_2^n$ . Trong hệ thập phân thì đó là các số từ 0 tới  $2^{n-1} - 1$  và có dạng

$$\mathbf{x} = (0, x_1, x_2, \dots, x_{n-1})$$

Nói cách khác, code  $\mathcal{C}$  là tập hợp

$$\mathcal{C} = \{\mathbf{x} = (0, x_1, x_2, \dots, x_{n-1}) : x_i \in \mathbb{F}_2\}$$

Code  $\mathcal{C}$  này thỏa mãn tính chất trên và mình sẽ chứng minh ngay sau đây.

*Chứng minh.* Giả sử ta nhận được  $\mathbf{y} \in \mathbb{F}_2^n$ . Ta có hai trường hợp:

- nếu  $\mathbf{y} = (0, y_1, y_2, \dots, y_{n-1})$ , hay nói cách khác biểu diễn thập phân của  $\mathbf{y}$  là từ 0 tới  $2^{n-1} - 1$ , thì  $\mathbf{y}$  được decode thành chính nó trong  $\mathcal{C}$ . Khi này  $d_{\mathbf{y}} = 0$  nhỏ nhất và không có vector nào khác cho Hamming weight bằng 0 trừ chính nó.
- nếu  $\mathbf{y} = (1, y_1, y_2, \dots, y_{n-1})$ , hay nói cách khác biểu diễn thập phân của  $\mathbf{y}$  là từ  $2^{n-1}$  tới  $2^n - 1$ , thì  $\mathbf{y}$  được decode thành  $\mathbf{x} = (0, y_1, y_2, \dots, y_{n-1})$  trong  $\mathcal{C}$ . Khi này  $d_{\mathbf{y}} = 1$  nhỏ nhất vì khác mỗi bit đầu tiên và cũng không có vector nào khác cho Hamming weight bằng 1.

□

Tiếp theo, mình viết các vector trong  $\mathcal{C}$  thành các hàng của 1 ma trận  $2^{n-1} \times n$ . Gọi  $A$  là ma trận hoán vị các cột của ma trận  $2^{n-1} \times n$  đó. Khi đó  $A$  là ma trận có tính chất: trên mỗi hàng và trên mỗi cột có đúng một phần tử (bằng 1) và ma trận  $A$  khả nghịch. Ví dụ, với  $n = 4$ , ma trận để hoán vị cột 2

với cột 4 là 
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Khi đó, nếu mình nhân ma trận  $2^{n-1} \times n$  của code  $\mathcal{C}$  với bất kì ma trận  $A$  nào như vậy thì code  $\mathcal{C}'$  nhận được cũng thỏa mãn tính chất trên.

**Ví dụ 2.** Với  $n = 4$  thì code  $\mathcal{C}$  gồm các vector

$$\mathcal{C} = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111\}$$

Với ma trận  $A$  hoán vị cột 2 và 4 như trên ta có

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} = \mathcal{C}'$$

Bây giờ mình sẽ chứng minh rằng với mọi ma trận  $A$  hoán vị các cột như vậy thì code  $\mathcal{C}'$  cũng thỏa mãn tính chất.

*Chứng minh.* Đặt

$$\mathcal{C} = \{(0, x_1, x_2, \dots, x_{n-1}), x_i \in \mathbb{F}_2\}$$

Gọi  $A$  là ma trận hoán vị cột kích thước  $n \times n$ . Khi đó ánh xạ

$$A : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n, \quad \mathbf{y} \rightarrow \mathbf{y} \cdot A$$

là song ánh do  $A$  là ma trận khả nghịch. Khi đó xét code

$$\mathcal{C}' = \{\mathbf{x} \cdot A : \mathbf{x} \in \mathcal{C}\}$$

Mình vẫn có hai trường hợp.

*Trường hợp 1.* Với  $\mathbf{y} = (0, y_1, y_2, \dots, y_{n-1}) \in \mathbb{F}_2^n$  từ 0 tới  $2^{n-1} - 1$  như trên. Xét  $\mathbf{y}' = \mathbf{y} \cdot A$ .

Khi đó, với  $\mathbf{x} = (0, y_1, y_2, \dots, y_{n-1}) \in \mathcal{C}$ , ta có  $\mathbf{x}' = \mathbf{x} \cdot A \in \mathcal{C}'$ . Từ đây suy ra

$$wt(\mathbf{x}' \oplus \mathbf{y}') = wt((\mathbf{x} \cdot A) \oplus (\mathbf{y} \cdot A)) = wt((\mathbf{x} \oplus \mathbf{y}) \cdot A) = wt(\mathbf{0} \cdot A) = 0$$

Ở đây  $\mathbf{0} = (0, 0, \dots, 0) \in \mathbb{F}_2^n$ .

Nói cách khác  $d_{\mathbf{y}'} = 0$  và có duy nhất một vector  $\mathbf{x}'$  được định nghĩa như trên thỏa mãn. Do đó  $|\mathcal{D}(\mathbf{y}')| = 1$ .

*Trường hợp 2.* Với  $\mathbf{y} = (1, y_1, y_2, \dots, y_{n-1}) \in \mathbb{F}_2^n$  từ  $2^{n-1}$  tới  $2^n - 1$  như trên. Ta cũng xét  $\mathbf{y}' = \mathbf{y} \cdot A$ .

Khi đó, với  $\mathbf{x} = (0, y_1, y_2, \dots, y_{n-1}) \in \mathcal{C}$ , ta cũng có  $\mathbf{x}' = \mathbf{x} \cdot A \in \mathcal{C}'$ . Từ đây ta có

$$wt(\mathbf{x}' \oplus \mathbf{y}') = wt((\mathbf{x} \cdot A) \oplus (\mathbf{y} \cdot A)) = wt((\mathbf{x} \oplus \mathbf{y}) \cdot A) = wt((1, 0, 0, \dots, 0) \cdot A) = 1$$

Ở phép nhân vector  $(1, 0, \dots, 0)$  với ma trận  $A$ , vì ma trận  $A$  chỉ có duy nhất một cột có dạng  $(1, 0, \dots, 0)^T$  nên kết quả phép nhân là một vector có đúng một phần tử 1, còn lại là 0.

Nói cách khác  $d_{\mathbf{y}'} = 1$  và có duy nhất một vector  $\mathbf{x}'$  được định nghĩa như trên thỏa mãn. Do đó  $|\mathcal{D}(\mathbf{y}')| = 1$ .

Như vậy ta đã chứng minh xong.  $\square$

Hoàn toàn tương tự, khi code  $\mathcal{C}$  là các vector bắt đầu với hai số 0 thì ta lần lượt xét  $\mathbf{y}$  trong các khoảng  $[0, 2^{n-2} - 1]$ ,  $[2^{n-2}, 2^{n-1} - 1]$ ,  $[2^{n-1}, 2^{n-1} + 2^{n-2} - 1]$ ,  $[2^{n-1} + 2^{n-2} - 1, 2^n - 1]$ . Nghĩa là

$$\mathcal{C} = \{\mathbf{x} = (0, 0, x_1, x_2, \dots, x_{n-2} : x_i \in \mathbb{F}_2^n)\}$$

Khi đó ta xét các vector  $\mathbf{y}$  có dạng:

- $\mathbf{y} = (0, 0, y_1, y_2, \dots, y_{n-2})$
- $\mathbf{y} = (0, 1, y_1, y_2, \dots, y_{n-2})$
- $\mathbf{y} = (1, 0, y_1, y_2, \dots, y_{n-2})$
- $\mathbf{y} = (1, 1, y_1, y_2, \dots, y_{n-2})$

Theo quy nạp thì code  $\mathcal{C}$  bắt đầu với  $i$  số 0 đều đúng,  $0 \leq i \leq n$ . Nghĩa là

$$\mathcal{C} = \{\mathbf{x} = (0, \dots, 0, x_1, x_2, \dots, x_{n-i}) : x_i \in \mathbb{F}_2\}$$

Sau đó chúng ta lại áp dụng phép nhân với ma trận hoán vị cột  $A$  như bên trên thì các code  $\mathcal{C}'$  cũng thỏa mãn.

Vấn đề ở đây là, những code  $\mathcal{C}$  như vậy là không gian vector sinh bởi  $i$  vector ( $0 \leq i \leq n$ ) trong các vector sau:

$$\begin{aligned} \mathbf{v}_1 &= (1, 0, 0, \dots, 0, 0) \\ \mathbf{v}_2 &= (0, 1, 0, \dots, 0, 0) \\ \mathbf{v}_3 &= (0, 0, 1, \dots, 0, 0) \\ &\dots = \dots \\ \mathbf{v}_n &= (0, 0, 0, \dots, 0, 1) \end{aligned}$$

Số cách chọn  $i$  vector từ  $n$  vector là

$$C_n^0 + C_n^1 + \dots + C_n^n = 2^n$$

cách. Nói cách khác có  $2^n$  code  $\mathcal{C}$  thỏa tính chất đề bài và là không gian tuyến tính của  $\mathbb{F}_2^n$ .

**Ví dụ 3.** Với  $n = 3$  thì các code sau thỏa mãn tính chất

$$\begin{aligned}\mathcal{C}_1 &= \{000\}, \\ \mathcal{C}_2 &= \{000, 001\}, \\ \mathcal{C}_3 &= \{000, 010\}, \\ \mathcal{C}_4 &= \{000, 100\}, \\ \mathcal{C}_5 &= \{000, 001, 010, 011\}, \\ \mathcal{C}_6 &= \{000, 001, 100, 101\}, \\ \mathcal{C}_7 &= \{000, 010, 100, 110\}, \\ \mathcal{C}_8 &= \{000, 001, 010, 011, 100, 101, 110, 111\}\end{aligned}$$

## Bình luận

Đối với Q1 có thể thấy rằng bất cứ code nào chỉ chứa đúng một vector sẽ thỏa mãn điều kiện. Lý do là vì bất cứ  $\mathbf{y}$  nào được gửi tới cũng sẽ decode ra vector đó.

Bài này mình được 6/12 điểm vì đưa ra cách xây dựng tốt, trình bày đẹp.

## Problem 8. Algebraic cryptanalysis

Bài này là bài 7 ở round 1 và là bài 8 ở round 2. Bài này mình giải khá qua loa ở round 1 và được giải đầy đủ, rõ ràng hơn bởi người đồng đội vip pro Chương ở round 2.

## Đề bài

Bob muốn xây dựng stream cipher **BOB-0.1**.

Bob sử dụng một binary key độ dài 8 là  $K = (k_1, \dots, k_8)$ . Sau đó anh ấy sinh ra dãy nhị phân  $\beta$  theo quy tắc:

- $\beta_n = k_n$  khi  $n = 1, 2, \dots, 8$
- $\beta_n = \beta_{n-1} \oplus \beta_{n-8}$  khi  $n \geq 9$

Sau đó Bob sinh dãy nhị phân  $\gamma$  dùng trong phép XOR với plaintext. Dãy  $\gamma$  được tạo bởi quy tắc  $\gamma_n = \beta_n \cdot \beta_{n+2} \oplus \beta_{n+7}$  với  $n \geq 1$ .

Alice chặn được 8 bit của  $\gamma$  sau khi để lỡ 1200 bit. Các bit đó là ‘00100001’. Liệu Alice có thể tìm lại được key  $K$  ban đầu không?

## Giải

Độ dài  $K$  là 8 bit, nếu chúng ta brutefore  $K = (k_1, \dots, k_8)$  rồi sinh ra 1208 bit  $\gamma$  theo quy tắc trên và so sánh xem  $\gamma_{1201}, \dots, \gamma_{1208}$  nào khớp với 8 bit trên

thì ta có thể biết được  $K$  ban đầu là gì.

Và, bất ngờ chưa, có tới hai trường hợp  $K$  thỏa mãn :v :v

Bây giờ thì chúng ta cần xem xem tại sao lại có hai trường hợp thỏa mãn.

Cùng nhau khai triển  $\beta_{n+1}, \dots, \beta_{n+8}$  theo  $(\beta_{n-7}, \dots, \beta_n)$  nào.

- $\beta_{n+1} = \beta_n \oplus \beta_{n-7}$
- $\beta_{n+2} = \beta_{n+1} \oplus \beta_{n-6} = \beta_n \oplus \beta_{n-7} \oplus \beta_{n-6}$
- $\beta_{n+3} = \beta_{n+2} \oplus \beta_{n-5} = \beta_n \oplus \beta_{n-7} \oplus \beta_{n-6} \oplus \beta_{n-5}$
- $\beta_{n+4} = \beta_{n+3} \oplus \beta_{n-4} = \beta_n \oplus \beta_{n-7} \oplus \beta_{n-6} \oplus \beta_{n-5} \oplus \beta_{n-4}$
- $\beta_{n+5} = \beta_{n+4} \oplus \beta_{n-3} = \beta_n \oplus \beta_{n-7} \oplus \beta_{n-6} \oplus \beta_{n-5} \oplus \beta_{n-4} \oplus \beta_{n-3}$
- $\beta_{n+6} = \beta_{n+5} \oplus \beta_{n-2} = \beta_n \oplus \beta_{n-7} \oplus \beta_{n-6} \oplus \beta_{n-5} \oplus \beta_{n-4} \oplus \beta_{n-3} \oplus \beta_{n-2}$
- $\beta_{n+7} = \beta_{n+6} \oplus \beta_{n-1} = \beta_n \oplus \beta_{n-7} \oplus \beta_{n-6} \oplus \beta_{n-5} \oplus \beta_{n-4} \oplus \beta_{n-3} \oplus \beta_{n-2} \oplus \beta_{n-1}$
- $\beta_{n+8} = \beta_{n+7} \oplus \beta_n = \beta_{n-7} \oplus \beta_{n-6} \oplus \beta_{n-5} \oplus \beta_{n-4} \oplus \beta_{n-3} \oplus \beta_{n-2} \oplus \beta_{n-1}$

Nếu viết ở dạng phép nhân ma trận modulo 2 ta có

$$\begin{pmatrix} \beta_{n+1} \\ \beta_{n+2} \\ \beta_{n+3} \\ \beta_{n+4} \\ \beta_{n+5} \\ \beta_{n+6} \\ \beta_{n+7} \\ \beta_{n+8} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} \beta_{n-7} \\ \beta_{n-6} \\ \beta_{n-5} \\ \beta_{n-4} \\ \beta_{n-3} \\ \beta_{n-2} \\ \beta_{n-1} \\ \beta_n \end{pmatrix}$$

Ma trận to to kia là ma trận khả nghịch. Do đó, nếu chúng ta có các  $\beta_{n+1}, \dots, \beta_{n+8}$  thì chúng ta có thể tìm ngược lại  $\beta_{n-7}, \dots, \beta_n$ . Tiếp tục quá trình này cuối cùng ta có thể tìm lại  $(\beta_1, \dots, \beta_8) = K$ .

Tiếp theo, cũng tương tự, chúng ta biểu diễn dãy  $\gamma$  theo  $\beta$ ..

- $\gamma_{n+1} = \beta_{n+1} \cdot \beta_{n+3} \oplus \beta_{n+8}$
- $\gamma_{n+2} = \beta_{n+2} \cdot \beta_{n+4} \oplus \beta_{n+1} \oplus \beta_{n+8}$
- $\gamma_{n+3} = \beta_{n+3} \cdot \beta_{n+5} \oplus \beta_{n+1} \oplus \beta_{n+2} \oplus \beta_{n+8}$
- $\gamma_{n+4} = \beta_{n+4} \cdot \beta_{n+6} \oplus \beta_{n+1} \oplus \beta_{n+2} \oplus \beta_{n+3} \oplus \beta_{n+8}$
- $\gamma_{n+5} = \beta_{n+5} \cdot \beta_{n+7} \oplus \beta_{n+1} \oplus \beta_{n+2} \oplus \beta_{n+3} \oplus \beta_{n+4} \oplus \beta_{n+8}$
- $\gamma_{n+6} = \beta_{n+6} \cdot \beta_{n+8} \oplus \beta_{n+1} \oplus \beta_{n+2} \oplus \beta_{n+3} \oplus \beta_{n+4} \oplus \beta_{n+5} \oplus \beta_{n+8}$
- $\gamma_{n+7} = \beta_{n+7} \cdot \beta_{n+1} \oplus \beta_{n+7} \cdot \beta_{n+8} \oplus \beta_{n+1} \oplus \beta_{n+2} \oplus \beta_{n+3} \oplus \beta_{n+4} \oplus \beta_{n+5} \oplus \beta_{n+6} \oplus \beta_{n+8}$
- $\gamma_{n+8} = \beta_{n+8} \cdot \beta_{n+1} \oplus \beta_{n+8} \cdot \beta_{n+2} \oplus \beta_{n+1} \oplus \beta_{n+2} \oplus \beta_{n+3} \oplus \beta_{n+4} \oplus \beta_{n+5} \oplus \beta_{n+6} \oplus \beta_{n+7}$

*Trường hợp 1.*  $\beta_{n+1} = 0$ . Khi đó từ  $\gamma_{1201}$  tới  $\gamma_{1208}$  tương đương với hệ phương



trình

$$\begin{aligned}
0 &= \beta_{n+8} \\
0 &= \beta_{n+2} \cdot \beta_{n+4} \\
1 &= \beta_{n+3} \cdot \beta_{n+5} \oplus \beta_{n+2} \\
0 &= \beta_{n+4} \cdot \beta_{n+6} \oplus \beta_{n+2} \oplus \beta_{n+3} \\
0 &= \beta_{n+5} \cdot \beta_{n+7} \oplus \beta_{n+2} \oplus \beta_{n+3} \cdot \beta_{n+4} \\
0 &= \beta_{n+2} \oplus \beta_{n+3} \oplus \beta_{n+4} \oplus \beta_{n+5} \\
0 &= \beta_{n+2} \oplus \beta_{n+3} \oplus \beta_{n+4} \oplus \beta_{n+5} \oplus \beta_{n+6} \\
1 &= \beta_{n+2} \oplus \beta_{n+3} \oplus \beta_{n+4} \oplus \beta_{n+5} \oplus \beta_{n+6} \oplus \beta_{n+7}
\end{aligned}$$

Hệ phương trình trên có nghiệm duy nhất  $(\beta_i) = (0, 1, 1, 0, 0, 0, 1, 0)$ .

*Trường hợp 2.*  $\beta_{n+1} = 1$ . Tương tự hệ phương trình cũng có nghiệm duy nhất  $(\beta_i) = (1, 1, 1, 0, 0, 0, 0, 1)$ .

Như vậy ta có hai nghiệm thỏa mãn chuỗi 8 bit  $\gamma_{1201}, \dots, \gamma_{1208}$ . Do không có điều kiện nào thêm, ta không thể xác định đâu là khóa trong hai trường hợp trên.

Bài này bạn Chương được 4/4 điểm. Good job nigga 😄.

## Problem 9\*\*. Finite-state machines

Bài này không biết làm!!!

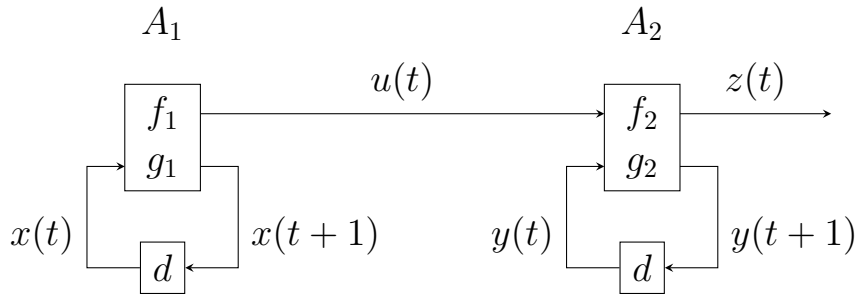
### Đề bài

Alice muốn tạo một generator để sinh một dãy số với độ dài chu kỳ lớn nhất có thể. Vì cô ấy biết về finite-state machine, generator  $G$  sẽ được xây dựng bởi hai machine  $A_1$  và  $A_2$  sao cho:

- $A_1 = (\mathbb{F}_2^n, \mathbb{F}_2, g_1, f_1)$  với hàm state-transition (hàm chuyển trạng thái)  $g_1 : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$  và hàm output  $f_1 : \mathbb{F}_2^n \rightarrow \mathbb{F}_2, n \geq 1$ ;
- $A_2 = (\mathbb{F}_2, \mathbb{F}_2^m, \mathbb{F}_2, g_2, f_2)$  với hàm state-transition  $g_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$  và hàm output  $f_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \rightarrow \mathbb{F}_2, m \geq 1$ .

Với mỗi  $t = 1, 2, \dots$ , đặt

1.  $x(t)$  và  $y(t)$  là trạng thái của  $A_1$  và  $A_2$ ,  $x(1)$  và  $y(1)$  là các giá trị khởi tạo;
2.  $x(t+1) = g_1(t)$  là trạng thái tiếp theo của  $A_1$  và  $u(t) = f_1(x(t))$  là output bit của  $A_1$ ;
3.  $y(t+1) = g_2(u(t), y(t))$  là trạng thái tiếp theo của  $A_2$  và  $z(t) = f_2(u(t), y(t))$  là output bit của  $A_2$ .



Dãy  $z(1), z(2), z(3), \dots$  là output của generator  $G$ . Dễ thấy rằng dãy sinh bởi  $G$  có chu kỳ nhỏ nhất không vượt quá  $2^{n+m}$ .

Theo thí nghiệm Alice thấy rằng, dãy output của  $G$  sẽ có chu kỳ nhỏ nhất nhỏ hơn  $2^{n+m}$  nếu Hamming weight của  $f_1$  là chẵn. Hãy chứng minh hoặc phủ định nhận xét của Alice.

## Giải

Note theo gợi ý của thầy Kolomeec, chưa hiểu hết.

Dễ thấy rằng nếu  $z(t)$  có chu kỳ tối đa thì các máy trạng thái  $A_1$  và  $A_2$  phải có chu kỳ tối đa lần lượt là  $2^n$  và  $2^m$ .

Các giá trị của  $g_2$  có thể chia ra hai phần là  $g_2(0, \mathbf{x})$  và  $g_2(1, \mathbf{x})$  với  $\mathbf{x} \in \mathbb{F}_2^m$ . Trong đó 0 và 1 xác định từ  $f_1$ .

Nếu  $g_2$  có chu kỳ cực đại thì trọng số phải là  $2^m$ , nói cách khác là hàm cân bằng, suy ra  $g_2(0, \mathbf{x})$  và  $g_2(1, \mathbf{x})$  đều có số chẵn phần tử.

Điều này có nghĩa là  $g_2$  sẽ sinh ra dãy có chu kỳ tạo thành hoán vị chẵn.

Tuy nhiên nếu  $g_2$  có chu kỳ cực đại thì chu kỳ đó phải tạo thành hoán vị lẻ vì khi đó hoán vị

$$1 \rightarrow 2 \rightarrow \dots \rightarrow t \rightarrow 1 = (1, 2)(1, 3) \dots (1, t)$$

có  $t - 1$  tranposition. Nói cách khác đây là hoán vị lẻ vì  $t = 2^m$  chẵn.

Như vậy để chu trình đạt chu kỳ tối đa thì nó phải là hoán vị lẻ, mâu thuẫn với phân tích ở trên  $g_2$  sẽ sinh ra dãy là hoán vị chẵn.

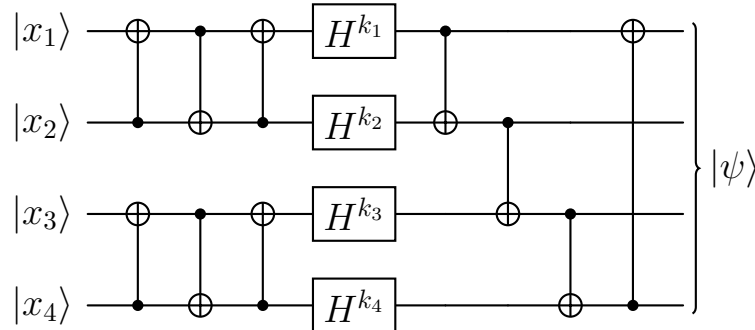
Trong cách giải này có hai chỗ mình không biết thầy lấy đâu ra: tại sao  $g_2$  cân bằng thì cả  $g_2(0, \mathbf{x})$  và  $g_2(1, \mathbf{x})$  đều phải có số chẵn phần tử mà không phải là đều có số lẻ? Liên hệ giữa trọng số hàm bool và hoán vị chẵn/lẻ là gì?

## Problem 10. Quantum encryption

Đây là bài 8 của round 1 và bài 10 của round 2. Bài này sai gần bước cuối mới cay 🤔.

## Đề bài

Bob tạo một thuật toán mã hóa encrypt 4 bit  $(x_1, x_2, x_3, x_4)$  bằng key cũng 4 bit  $(k_1, k_2, k_3, k_4)$  với mạch sau:



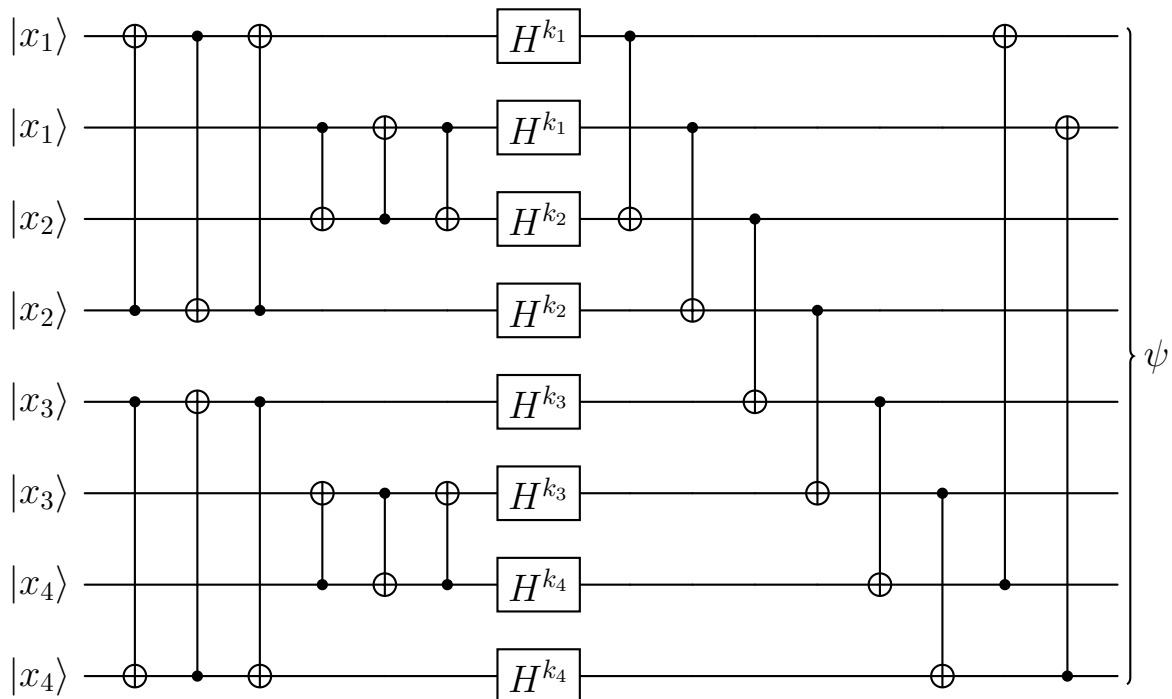
Plaintext 4 bit  $(x_1, x_2, x_3, x_4)$  được biểu diễn ở dạng 4-qubit "plainstate"  $|x_1, x_2, x_3, x_4\rangle$ . Quantum state này là input cho mạch ở dạng qubit đơn đi qua các cổng.

Ở đây hai loại cổng được sử dụng là CNOT và Hadamard.

Ký hiệu  $H^b$  với  $b \in \{0, 1\}$  có nghĩa là, nếu  $b = 0$  thì cổng đồng nhất  $I$  được sử dụng (không thay đổi), còn nếu  $b = 1$  thì cổng Hadamard sẽ được sử dụng.

Kết quả sau khi qua mạch là "cipherstate"  $|\psi\rangle$ .

Bob có nhiệm vụ tăng số qubit đầu vào lên nhằm giảm các sai số khi tính toán và truyền dữ liệu trên kênh quantum. Do đó Bob biến đổi thành mạch sau:

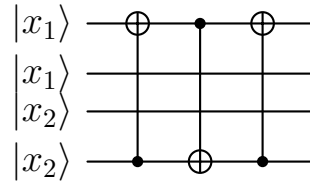


Alice nói rằng cô ấy có thể tìm lại được key nếu biết  $N$  amplitude của kết

quả  $|\psi\rangle$ . Do có 8 qubits ở kết quả nên số lượng amplitude tối đa là  $2^8 = 256$ , nói cách khác  $N \leq 256$ . Vậy Alice cần ít nhất bao nhiêu amplitude là đủ để tìm lại key?

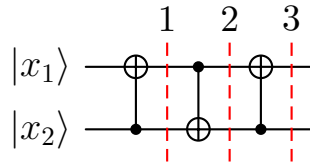
## Giải

Đầu tiên xét 4 dây trên, 4 dây dưới tương tự.

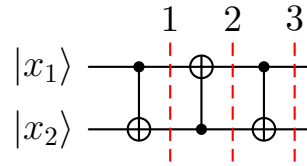


Chúng ta xét dây 1 và 4 của mạch (tương tự cho dây 2 và 3). Áp dụng cổng CNOT liên tiếp 3 lần ta có

$$|x_1\rangle \otimes |x_2\rangle \rightarrow |x_1 \oplus x_2\rangle \otimes |x_2\rangle \rightarrow |x_1 \oplus x_2\rangle \otimes |x_1\rangle \rightarrow |x_2\rangle \otimes |x_1\rangle$$



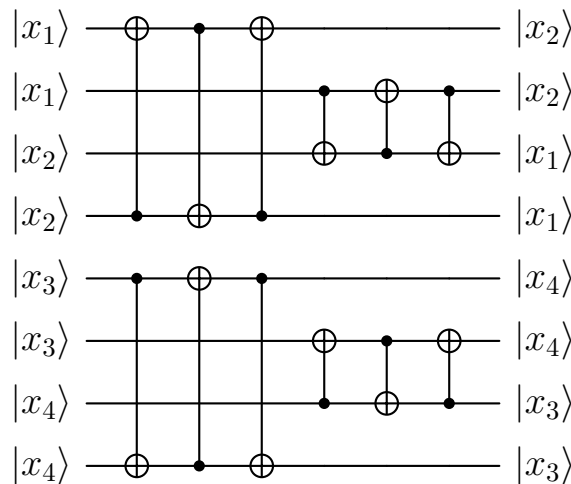
(a) Dây 1 và dây 4



(b) Dây 2 và dây 3

Nói cách khác là đảo bit 🤪.

Tương tự cho các cặp dây (5, 8) và (6, 7). Do đó khi tới trước các cổng Hadamard thì thứ tự các qubit từ trên xuống dưới là hình C.2.

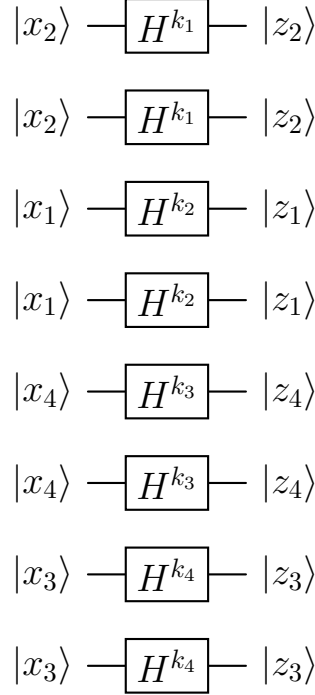


Hình C.2. Qubits trước Hadamard

Mạch ở đây 1 và 2 đều có dạng  $|x_2\rangle$  đi qua  $H^{k_1}$  nên sau khi qua cổng mình đặt  $|z_2\rangle = H^{k_1}|x_2\rangle$ .

Tương tự,  $|z_1\rangle = H^{k_2}|x_1\rangle$  cho dây 3 và 4,  $|z_4\rangle = H^{k_3}|x_4\rangle$  cho dây 5 và 6,  $|z_3\rangle = H^{k_4}|x_3\rangle$  cho dây 7 và 8.

Mạch sau khi đi qua Hadamard có dạng C.3



Hình C.3. Qubits sau Hadamard

Ở đây chúng ta có một lưu ý nhỏ có thể giúp ích trong việc giới hạn số lượng amplitude theo đề bài. Nếu  $k_1 = 0$  thì  $|z_2\rangle = |x_2\rangle$ . Nếu  $k_1 = 1$  thì  $|z_2\rangle = \frac{|0\rangle + (-1)^{x_2}|1\rangle}{\sqrt{2}}$ . Như vậy, hệ số trước  $|0\rangle$  của  $|z_2\rangle$  có thể là  $0, 1, \frac{1}{\sqrt{2}}$  đều không âm.

Bây giờ chúng ta quay lại toán tử CNOT. Ma trận tương ứng của toán tử

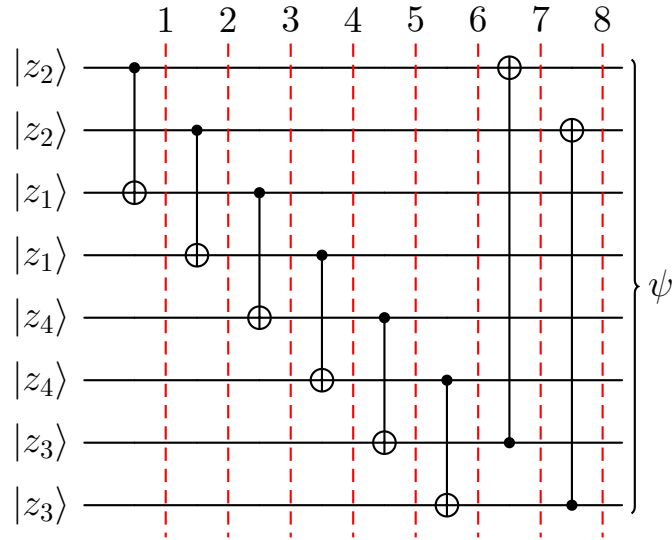
CNOT là  $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ . Kết quả sau khi thực hiện toán tử CNOT là hệ số

trước  $|00\rangle$  và  $|01\rangle$  giữ nguyên, còn hệ số trước  $|10\rangle$  và  $|11\rangle$  đổi chỗ cho nhau.

Đối với 3 qubit, mình **dự đoán** tương tự.

Ở cổng CNOT đầu tiên, dây 1 control dây 3. Nếu mình chỉ xét 3 dây đầu thì tích các qubit gồm  $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$ .

Áp dụng "chiến thuật" tương tự, mình chỉ quan tâm vị trí 1 và 3. Nghĩa là hệ số của  $|0x0\rangle$  và  $|0x1\rangle$  giữ nguyên, còn hệ số trước  $|1x0\rangle$  và  $|1x1\rangle$  đổi chỗ cho nhau, với  $x \in \{0, 1\}$ . Nói cách khác, 8 hệ số trước amplitude chỉ thay đổi vị trí chứ không nhiều hơn hay ít đi, hay tập hợp hệ số giữ nguyên.



Như vậy, giả sử  $|z_2\rangle = a|0\rangle + b|1\rangle$ ,  $|z_1\rangle = c|0\rangle + d|1\rangle$ ,  $|z_4\rangle = e|0\rangle + f|1\rangle$ ,  $|z_3\rangle = g|0\rangle + h|1\rangle$ . Khi đó kết quả cipherstate là

$$|\psi\rangle = |z_2\rangle \otimes |z_2\rangle \otimes |z_1\rangle \otimes |z_1\rangle \otimes |z_4\rangle \otimes |z_4\rangle \otimes |z_3\rangle \otimes |z_3\rangle$$

Xét  $|z_2\rangle \otimes |z_2\rangle = a^2|00\rangle + ab|01\rangle + ab|10\rangle + b^2|11\rangle$ . Ở đây có 3 hệ số khác nhau là  $(a^2, ab, b^2)$ . Với lưu ý bên trên  $a \geq 0$  nên từ  $a^2$  tính được  $a$ . Từ  $a$ , ta cần thêm  $ab$  để xác định  $b$ .

Như vậy mình cần  $2^4 = 16$  hệ số để tìm lại các key ban đầu.

## Bình luận

Thế éo nào mình lại nhầm khúc cuối mà lấy cả  $a^2$ ,  $ab$  và  $b^2$  nên kết quả ra  $3^4 = 81$ . Tất nhiên là **SAI BÉT** nên chỉ được 2/8 🤡.

## Problem 11. AntCipher

Bài này là bài số 2 ở round 1 và là bài số 11 ở round 2. Lúc thi round 1 mình không biết giải, còn ở round 2 thì mình đã giải theo cách như sau.

## Đề bài

Đặt

$$\begin{aligned} f = & (x_1 \vee x_2 \vee x_9) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_9) \wedge (\neg x_1 \vee x_2 \neg x_9) \wedge (x_1 \vee \neg x_2 \vee x_9) \wedge \\ & (x_1 \vee x_2 \vee x_3) \wedge (\neg x_9 \vee \neg x_{10} \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_9 \vee x_{10} \vee \neg x_4) \wedge \\ & (\neg x_1 \vee x_2 \vee x_5) \wedge (x_9 \vee \neg x_{10} \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_6) \wedge (x_9 \vee x_{10} \vee \neg x_6) \wedge \\ & (x_1 \vee x_2 \vee x_3 \vee x_4 \vee \neg x_7) \wedge (x_2 \vee x_3 \vee x_4 \vee \neg x_7 \vee \neg x_8) \end{aligned}$$

Hàm  $f$  gồm 10 biến được viết dưới dạng CNF (conjunctive normal form). Thuật toán mã hóa dựa trên hàm  $f$  biến đổi hai bit plaintext  $(x_1, x_2)$  thành hai bit ciphertext  $(x_9, x_{10})$  khi giá trị hàm  $f = True$ . Hàm  $f$  này có 10 biến  $x_1, x_2, \dots, x_{10}$  và 46 literals, là các hạng tử trong biểu diễn CNF của hàm. Ví dụ với dấu ngoặc thứ hai có 3 literals là  $\neg x_1$ ,  $\neg x_2$  và  $\neg x_9$ .

**Q.** Vì các giới hạn tính toán nên chúng ta chỉ có thể sử dụng tối đa 16 biến với 20 literals. Nhắc lại rằng hàm  $f$  ở trên có 10 biến và 46 literals. Hãy tìm cách biểu diễn tương đương của thuật toán mã hóa trên với giới hạn đã cho.

## Giải

Khi mình code hàm để tính giá trị hàm  $f$  và xem xét những vector

$$\mathbf{x} = (x_1, \dots, x_{10})$$

mà  $f = True$ , mình nhận thấy rằng:

- nếu  $(x_1, x_2) = (0, 0)$  thì  $(x_9, x_{10}) = (1, 0)$
- nếu  $(x_1, x_2) = (0, 1)$  thì  $(x_9, x_{10}) = (1, 1)$
- nếu  $(x_1, x_2) = (1, 0)$  thì  $(x_9, x_{10}) = (0, 0)$
- nếu  $(x_1, x_2) = (1, 1)$  thì  $(x_9, x_{10}) = (0, 1)$

Mình nhận ra rằng các biến  $x_3, x_4, \dots, x_7, x_8$  hoàn toàn không tác động lên việc mã hóa từ  $(x_1, x_2)$  thành  $(x_9, x_{10})$  (ít nhất là ở những chỗ  $f = True$  :v).

Như vậy bài toán được rút gọn thành hàm boolean 4 biến  $x_1, x_2, x_9$  và  $x_{10}$ . Ở đó  $f(0010) = f(0111) = f(1000) = f(1101) = 1$ . Các vector còn lại thì  $f = 0$ . Ở dưới là bảng chân trị C.1.

Từ bảng chân trị trên, sử dụng phương pháp bìa Karnaugh mình rút gọn được thành

$$\begin{aligned} f(x_1, x_2, x_9, x_{10}) = & (\neg x_1 \vee \neg x_9) \wedge (x_1 \vee x_9) \wedge \\ & (\neg x_1 \vee \neg x_2 \vee x_{10}) \wedge (x_1 \vee x_2 \vee \neg x_{10}) \wedge \\ & (\neg x_1 \vee x_2 \vee \neg x_{10}) \wedge (x_1 \vee \neg x_2 \vee x_{10}) \end{aligned}$$

CNF này có 4 biến và 16 literals, thỏa mãn yêu cầu đề bài và ăn trọn 6/6 điểm 😊.

$x_1$	$x_2$	$x_9$	$x_{10}$	$f$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Bảng C.1: Bảng chân trị hàm  $f$



# Phụ lục D

## Olympiad

### D.1 Ôn thi ngày 20/11/2023

#### Toán tử tuyến tính

Toán tử tuyến tính là một ánh xạ

$$A : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Nếu  $A$  là một ma trận cỡ  $m \times n$  thì đây là một ánh xạ tuyến tính với phép nhân ma trận với vector  $A \cdot \mathbf{x} = \mathbf{y}$ .

Ở đây  $\mathbf{x} \in \mathbb{R}^n$  và  $\mathbf{y} \in \mathbb{R}^m$ .

#### Định nghĩa 1. Hạt nhân

**Hạt nhân** của ánh xạ tuyến tính  $A$  là tập hợp nghiệm của hệ thuần nhất và được ký hiệu là  $\ker(A)$ . Nói cách khác

$$\ker(A) = \{\mathbf{x} \in \mathbb{R}^n : A \cdot \mathbf{x} = \mathbf{0}\} \quad (\text{D.1})$$

#### Định nghĩa 2. Ảnh

**Ảnh** của ánh xạ tuyến tính  $A$  là tập hợp tất cả giá trị có thể của phép nhân ma trận và được ký hiệu là  $\text{im}(A)$ . Nói cách khác

$$\text{im}(A) = \{A \cdot \mathbf{x} : \mathbf{x} \in \mathbb{R}^n\} \quad (\text{D.2})$$

Tính chất đối với ánh xạ  $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$  là  $\dim(\ker A) + \dim(\text{im } A) = n$ .

#### Trị riêng và vector riêng

### Định nghĩa 3. Trị riêng, vector riêng

Xét hệ phương trình tuyến tính thuần nhất biểu diễn bởi phép nhân

$$A \cdot \mathbf{x} = \lambda \cdot \mathbf{x}$$

Giá trị  $\lambda$  khiến phương trình có nghiệm không tầm thường được gọi là **trị riêng** (eigenvalue) của ánh xạ tuyến tính.

Vector  $\mathbf{x}$  là cơ sở của không gian vector nghiệm khi đó được gọi là **vector riêng** (eigenvector) ứng với trị riêng  $\lambda$ .

Lưu ý rằng có thể có nhiều vector riêng tương ứng với một trị riêng.

Để tìm trị riêng ta giải phương trình đặc trưng  $\det(A - \lambda I) = 0$  và tìm tất cả nghiệm thực  $\lambda$  của phương trình.

Sau đó ta thế từng  $\lambda$  vào hệ  $A\mathbf{x} = \lambda\mathbf{x}$  và tìm cơ sở của không gian nghiệm. Các vector trong cơ sở là vector riêng tương ứng với  $\lambda$  đó.

Một số tính chất của trị riêng và vector riêng (giả sử rằng đối với ma trận  $A$  cỡ  $n \times n$  thì phương trình đặc trưng có đầy đủ  $n$  nghiệm thực).

1.  $\text{tr } A = \lambda_1 + \lambda_2 + \dots + \lambda_n$
2.  $\det A = \lambda_1 \cdot \lambda_2 \cdots \lambda_n$

Tính chất liên quan đến rank và trace:

1.  $\text{tr}(AB) = \text{tr}(BA)$
2.  $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$

## Bài tập

**Bài 1.** Cho vector cột  $\mathbf{v} \in \mathbb{R}^n$ . Đặt  $A = \mathbf{v} \cdot \mathbf{v}^T$ . Tìm  $\text{spa } A$ .

Các cột của  $A$  có dạng  $\mathbf{v} \cdot v_1, \mathbf{v} \cdot v_2, \dots, \mathbf{v} \cdot v_n$ . Như vậy các cột đều tỉ lệ với cột đầu nên  $\text{rank } A = 1$ .

Suy ra  $\dim \ker A = n - 1$  và do đó  $\lambda = 0$  là nghiệm bậc  $n - 1$  trong phương trình đặc trưng.

Như vậy phương trình đặc trưng còn một nghiệm  $\lambda \neq 0$ .

Do  $(\mathbf{v} \cdot \mathbf{v}^T)\mathbf{x} = \lambda\mathbf{x} \Leftrightarrow \mathbf{v}(\mathbf{v}^T \cdot \mathbf{x}) = \lambda\mathbf{x}$ .

Đặt  $\mathbf{v}^T \cdot \mathbf{x} = \alpha$  thì  $\alpha\mathbf{v} = \lambda\mathbf{x}$ . Suy ra  $\mathbf{x} = \mathbf{v}$  và do đó  $\alpha = \lambda = \|\mathbf{v}\|^2$ .

Vậy  $\text{spa } A = \{\|\mathbf{v}\|^2, 0, 0, \dots, 0\}$ .

**Bài 3.** Cho ma trận  $A_{3 \times 3}$ . Biết rằng  $\text{tr } A = \text{tr } A^{-1} = 0$  và  $\det A = 1$ . Chứng minh rằng  $A^3 = I$ .

Phương trình đặc trưng có dạng  $P_3(\lambda) = -\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0$ .

Theo tính chất trên thì  $a_2 = \sum \lambda = \text{tr } A = 0$ .

Do  $\lambda$  là trị riêng nên  $A\mathbf{x} = \lambda\mathbf{x}$ . Do  $A$  khả nghịch nên  $\frac{1}{\lambda}\mathbf{x} = A^{-1}\mathbf{x}$ .

Nghĩa là  $\frac{1}{\lambda}$  là trị riêng của ma trận  $A^{-1}$ . Suy ra  $\frac{1}{\lambda_1} + \frac{1}{\lambda_2} + \frac{1}{\lambda_3} = \text{tr } A^{-1} = 0$ .

Từ đó suy ra  $\lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1 = 0$ .

Cuối cùng  $\det A = \lambda_1 \cdot \lambda_2 \cdot \lambda_3 = 1$ .

Vậy phương trình đặc trưng là  $P_3(\lambda) = -\lambda^3 + 1$ . Theo định lý Cayley-Hamilton thì  $P_3(A) = -A^3 + I = 0$ , hay  $A^3 = I$ .

**Bài 4.** Cho ma trận  $A_{n \times n}$ ,  $A_{ij} \geq 0$ . Giả sử ma trận có đủ  $n$  trị riêng thực. Chứng minh rằng  $\lambda_1^k + \lambda_2^k + \dots + \lambda_n^k \geq 0$  với mọi  $k \in \mathbb{N}$ .

Ta thấy rằng với  $k = 1$  thì  $\lambda_1 + \dots + \lambda_n = \text{tr}(A) \geq 0$ .

Vì  $\lambda_i$  là thỏa phương trình  $A\mathbf{x} = \lambda_i\mathbf{x}$  nên nhân hai vế cho  $A$  ta có  $A \cdot A\mathbf{x} = A \cdot \lambda_i\mathbf{x}$ . Tương đương với  $A^2\mathbf{x} = \lambda_i(A\mathbf{x}) = \lambda_i^2\mathbf{x}$ .

Nói cách khác,  $\lambda_i^2$  là trị riêng của ma trận  $A^2$ . Thực hiện tương tự ta có  $\lambda_i^k$  là trị riêng của ma trận  $A^k$ .

Do đó  $\lambda_1^k + \dots + \lambda_n^k = \text{tr}(A^k) \geq 0$ .

**Bài 5.** Cho ma trận  $A$  khả nghịch.  $X$  là ma trận sao cho  $AX + XA = 0$ . Chứng minh rằng  $\text{tr } X = 0$ .

Nhân bên trái hai vế cho  $A^{-1}$  ta có  $X + A^{-1}XA = 0$ . Ta biết rằng  $A^{-1}XA$  là ma trận tương đương ma trận  $X$  nên  $\text{tr}(A^{-1}XA) = \text{tr } X$ .

Suy ra  $\text{tr } X + \text{tr } X = \text{tr } 0 = 0$ . Từ đây có  $\text{tr } X = 0$ .

## D.2 RUDN Olympiad 2023

Lần đầu tiên mình được tham dự thi toán đồng đội theo hình thức MathBoy (trận chiến toán).

Trong cách thi này, mỗi đội có 3 vị trí: người thuyết trình (докладчик), người phản biện (оппонент) và người giám sát (наблюдатель).

Ở mỗi vòng sẽ có 3 đội thi với nhau. Mỗi đội sẽ có 1 vị trí tương ứng với 3 vị trí trên. Sau đây là ví dụ

	Đội 1	Đội 2	Đội 3
Vòng 1	O	Д	H
Vòng 2	H	O	Д
Vòng 3	Д	H	O

Ở mỗi vòng, đội đóng vai trò người thuyết trình lên bảng ghi bài giải trong thời gian cho phép và thuyết trình về bài giải của đội mình. Đội phản biện có nhiệm vụ phản biện bài thuyết trình đó. Đội giám sát, dựa trên bài thuyết trình cũng như phản biện mà ghi chép lại các lỗi, chỗ khó hiểu, ... và trình lên cho giám khảo.

Ngoài ra, đội thuyết trình trước đó phải trình bài giải viết tay cho giám khảo chấm trước khi lên thuyết trình.

Ở đây có rất nhiều câu chuyện hack não đã xảy ra. Lúc mình thi vòng 1, câu hỏi quá khó nên đội thuyết trình chỉ viết được một ít. Đồng nghĩa việc đội phản biện cũng như đội giám sát ... thất nghiệp, không có gì để nói.

Đối với vòng 2, trận chiến cân bằng hơn, đội mình làm việc giám sát. Dựa trên bài giải của đội thuyết trình, chúng mình thấy những trường hợp chưa được xét tới và có thể bị sai, do đó cả ba đội đều có điểm (đội thuyết trình có nhiều điểm nhất vì các bạn giải hơn 1 nửa rồi).

Đối với vòng 3, đội mình thuyết trình. Đội mình clear bài đó nên giành điểm tuyệt đối cho phần thuyết trình. Tuy nhiên các bạn phản biện cũng không vừa, vẫn cố gắng bắt một số lỗi do trình bày quá cô đọng. Kết quả là đội mình (thuyết trình) full điểm cho vòng 3, đội phản biện được 3 điểm.

# Phụ lục E

## Đường đoản thời

### Lời nói đầu

Động lực để tác giả viết bài này là sau khi đọc về sự ra đời phép tính vi tích phân cùng vụ tranh cãi đáng xấu hổ trong lịch sử toán học giữa Newton và Leibniz, cùng với bài toán của Johann Bernoulli.

Bài nghiên cứu này được tham khảo nhiều nguồn (từ Miguel A. Lerma<sup>1</sup> và Lê Quang Ánh<sup>2</sup>) và là tài liệu học tập cá nhân. Tác giả hy vọng rằng bài nghiên cứu nhỏ này sẽ giúp ích được cho các bạn học sinh, sinh viên đam mê toán và vật lý (mặc dù tác giả không phải dân lý hihi).

### Bối cảnh lịch sử

Thế kỷ 17 đã chứng kiến một drama có thể gọi là đáng xấu hổ nhất lịch sử toán học. Hai nhà toán học có ảnh hưởng rất lớn lại vướng vào một vụ kiện tụng và tranh cãi khó coi để xem ai là người phát minh ra phép tính vi tích phân. Vâng, chúng ta đang nói đến Newton và Leibniz. Vào thời điểm đó có một nhà toán học xuất sắc thuộc một dòng họ cũng gồm rất nhiều nhân vật xuất sắc đã đưa ra một bài toán đố cho các nhà toán học trên thế giới. Bài toán đó đã chứng minh được ưu thế vượt trội trong phương pháp vi tích phân của Leibniz.

Nhà toán học xuất sắc đó là Johann Bernoulli, thuộc dòng họ Bernoulli nổi tiếng. Bài toán đó được phát biểu như sau:

*Cho hai điểm  $A$  và  $B$  nằm trong mặt phẳng thẳng đứng  $P$  ( $A$  cao hơn  $B$ ). Hãy xác định đường nối hai điểm  $A$  và  $B$  và nằm trong mặt phẳng  $P$  sao cho một điểm chỉ chịu trọng lực chạy từ  $A$  đến  $B$  trong thời gian ngắn nhất.*

---

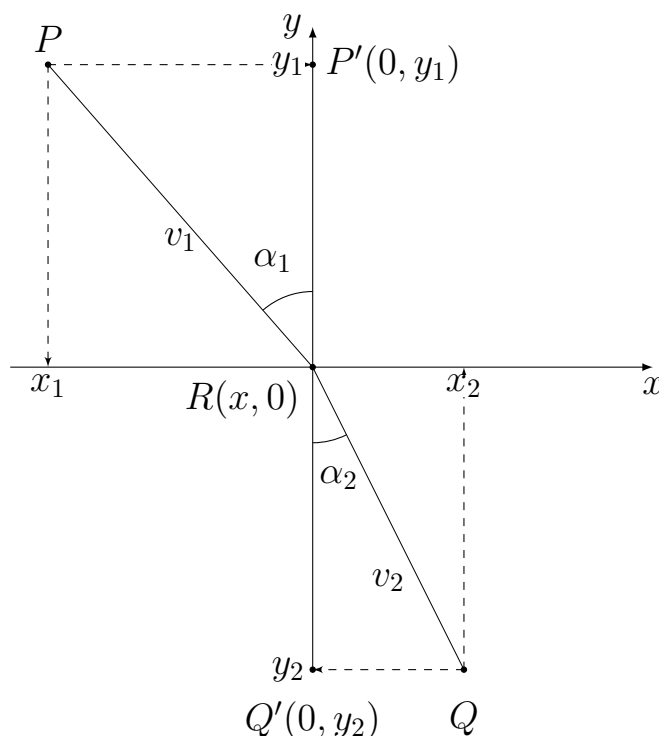
<sup>1</sup>A simple derivation of the equation for the brachistochrone curve, <https://sites.math.northwestern.edu/~mlerma/papers-and-preprints/brachistochrone.pdf>

<sup>2</sup>Gia đình Bernoulli: một dòng họ Toán học, trang 7, <https://rosetta.vn/lequanganh/gia-dinh-bernoulli-mot-dong-ho-toan-hoc/>

Chúng ta đã biết rằng đường đi ngắn nhất giữa hai điểm là đoạn thẳng nối hai điểm đó. Tuy nhiên trong bài toán của Johann Bernoulli thì đại lượng ngắn nhất cần tìm không phải khoảng cách giữa hai điểm mà là thời gian di chuyển giữa hai điểm. Mục tiêu cần làm ở bài toán này là xác định đường đi (hay quỹ đạo) thời gian ngắn nhất đó. Do đó bài toán này được gọi là bài toán *đường đoản thời* (brachistochrone curve).

Để giải bài toán này chúng ta cần một định luật cũng về thời gian ngắn nhất. Đó là nguyên lý thời gian ngắn nhất của Fermat và một hệ quả của nó là định luật Snell-Descartes.

## Định luật Snell-Descartes



Hình E.1. Định luật Snell-Descartes

Nguyên lý thời gian ngắn nhất của Fermat phát biểu rằng

*Khi ánh sáng truyền từ môi trường này sang môi trường khác thì nó luôn truyền đi theo đường nhanh nhất.*

Hệ quả của nguyên lý của Fermat là định luật Snell-Descartes mà chúng ta thường thấy ở chương trình vật lý ở phổ thông dưới dạng định luật khúc xạ ánh sáng

$$\frac{\sin \alpha_1}{\sin \alpha_2} = \frac{v_1}{v_2} \quad (\text{E.1})$$

với  $\alpha_1$  và  $\alpha_2$  lần lượt là góc hợp bởi tia vào và tia ra với pháp tuyến tại điểm tới,  $v_1$  và  $v_2$  là vận tốc truyền trong môi trường ở nửa trên và nửa dưới  $Ox$  (xem

hình E.1).

Để chứng minh định luật trên, ta thấy rằng  $v_1$  là vận tốc khi di chuyển từ điểm  $P$  tới điểm  $R$  nên thời gian  $t_1$  đi từ điểm  $P$  tới  $R$  là

$$t_1 = \frac{\|\overrightarrow{PR}\|}{v_1} = \frac{\sqrt{(x - x_1)^2 + y_1^2}}{v_1} \quad (\text{E.2})$$

Lưu ý rằng tia sáng không truyền tới gốc tọa độ  $O(0,0)$  mà truyền tới một điểm  $R(x,0)$  là vì điểm bắt đầu là  $P(x_1, y_1)$  và ánh sáng truyền đi theo đường nhanh nhất (theo nguyên lý Fermat) nên không có gì đảm bảo rằng nó sẽ truyền tới  $O(0,0)$ .

Tương tự, thời gian  $t_2$  đi từ điểm  $R$  tới  $Q$  là

$$t_2 = \frac{\|\overrightarrow{RQ}\|}{v_2} = \frac{\sqrt{(x - x_2)^2 + y_2^2}}{v_2} \quad (\text{E.3})$$

Kết hợp hai phương trình của  $t_1$  và  $t_2$  lại thì tổng thời gian di chuyển từ  $P$  tới  $Q$  biểu diễn theo  $x$  là

$$T(x) = t_1 + t_2 = \frac{\sqrt{(x - x_1)^2 + y_1^2}}{v_1} + \frac{\sqrt{(x - x_2)^2 + y_2^2}}{v_2} \quad (\text{E.4})$$

Đạo hàm theo  $x$  ta có

$$T'(x) = \frac{x - x_1}{v_1 \sqrt{(x - x_1)^2 + y_1^2}} + \frac{x - x_2}{v_2 \sqrt{(x - x_2)^2 + y_2^2}} \quad (\text{E.5})$$

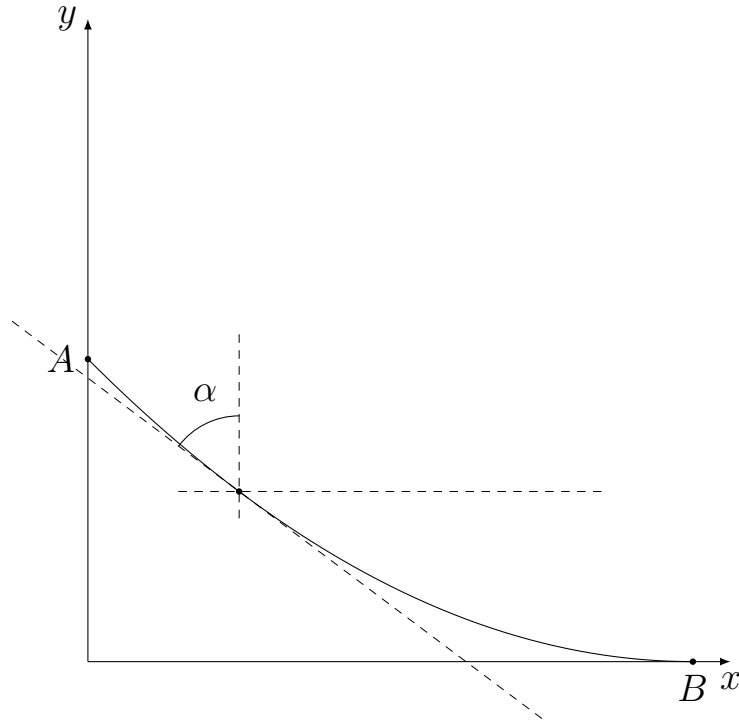
Để ý rằng  $x > x_1$  nên  $x - x_1 = \|\overrightarrow{PP'}\|$ . Tương tự  $x_2 - x = \|\overrightarrow{QQ'}\|$ . Để tìm cực trị ta cho đạo hàm bằng 0 rồi tính đạo hàm cấp 2. Ta có

$$T'(x) = 0 \Leftrightarrow \frac{\|\overrightarrow{PP'}\|}{v_1 \|\overrightarrow{PR}\|} - \frac{\|\overrightarrow{QQ'}\|}{v_2 \|\overrightarrow{RQ}\|} = 0 \Leftrightarrow \frac{\sin \alpha_1}{v_1} - \frac{\sin \alpha_2}{v_2} = 0$$

Như vậy  $\frac{\sin \alpha_1}{v_1} = \frac{\sin \alpha_2}{v_2}$ . Đạo hàm cấp 2 tương ứng là

$$T''(x) = \frac{y_1^2}{v_1((x - x_1)^2 + y_1^2)} + \frac{y_2^2}{v_2((x - x_2)^2 + y_2^2)} > 0$$

Do đó  $x$  thỏa  $T'(x) = 0$  ở trên là cực tiểu và định luật Snell-Descartes được chứng minh.



Hình E.2. Đường cycloid

## Đường cong Cycloid

Đáp án cho bài toán mà Johann Bernoulli đặt ra là đường cong Cycloid. Sau đây sẽ trình bày cách giải bài toán của Johann Bernoulli.

Phương của vận tốc tức thời tại một điểm khi một vật đi theo một quỹ đạo đường cong là tiếp tuyến với đường cong tại điểm đó. Khi đó góc  $\alpha$  trong định luật Snell-Descartes sẽ có liên hệ với hệ số góc của tiếp tuyến với đường cong. Nói rõ hơn, góc hợp bởi tiếp tuyến và trục  $Ox$  là  $\alpha + \frac{\pi}{2}$  và hệ số góc của tiếp tuyến là  $\tan\left(\alpha + \frac{\pi}{2}\right) = \frac{dy}{dx}$  (hình E.2).

Ta có  $\tan\left(\alpha + \frac{\pi}{2}\right) = -\cot \alpha$  và  $1 + \cot^2 \alpha = \frac{1}{\sin^2 \alpha}$  nên

$$\frac{1}{\sin^2 \alpha} = 1 + \cot^2 \alpha = 1 + \left(\frac{dy}{dx}\right)^2 \quad (\text{E.6})$$

Giả sử tọa độ của  $A$  là  $(x_0, y_0)$ . Khi một điểm di chuyển từ  $A$  tới  $B$ , gọi  $(x, y)$  là tọa độ của điểm đó trên đường cong. Theo định luật bảo toàn cơ năng thì

$$mgy_0 = \frac{1}{2}mv^2 + mgy$$

với  $v$  là vận tốc tức thời tại điểm  $(x, y)$  và  $mgy$  là thế năng tại điểm đó. Như vậy ta có

$$v^2 = 2g(-y + y_0) \quad (\text{E.7})$$



Theo định luật Snell-Descartes thì  $\frac{v}{\sin \alpha}$  là một hằng số khi nằm trong cùng môi trường. Do đó tồn tại số  $r$  cố định sao cho  $\frac{v^2}{\sin^2 \alpha} = 4gr$ . Từ hai biểu thức của  $v^2$  và  $\frac{1}{\sin^2 \alpha}$  ở trên ta có

$$\frac{v^2}{\sin^2 \alpha} = 2g(-y + y_0) \left( 1 + \left( \frac{dy}{dx} \right)^2 \right) = 4gr \quad (\text{E.8})$$

Suy ra

$$\left( \frac{dy}{dx} \right)^2 = \frac{2r}{y_0 - y} - 1 \quad (\text{E.9})$$

Tới đây ta thấy rằng bậc của  $dy$  và  $dx$  ở vế trái là giống nhau, trong khi vế phải chỉ có  $y$  mà không có  $x$ . Do đó "bắt chước" cách đổi biến của đường tròn, đặt

$$\begin{cases} x = a\theta + b \cos \theta \\ y = c + d \sin \theta \end{cases}$$

với  $a, b, c, d$  là các số thực cần tìm,  $\theta$  là góc hợp bởi  $Oy$  và đoạn thẳng nối tâm  $O$  và điểm trên đường cong (theo góc  $\alpha$ ). Lưu ý rằng khi thay  $\theta = 0$  và  $\theta = \pi/2$  vào hai phương trình trên ta phải thu được hai điểm trên hai trục tọa độ.

Lấy vi phân hai phương trình trên ta có

$$\begin{cases} dx = a - b \sin \theta d\theta \\ dy = d \cos \theta d\theta \end{cases}$$

Thay vào phương trình E.9 ta được

$$\frac{d^2 \cos^2 \theta}{(a - b \sin \theta)^2} = \frac{2r}{y_0 - c - d \sin \theta} - 1 = \frac{2r - y_0 + c + d \sin \theta}{y_0 - c - d \sin \theta}$$

Do  $\cos^2 \theta = 1 - \sin^2 \theta = (1 - \sin \theta)(1 + \sin \theta)$  nên ta muốn chọn  $a$  và  $b$  có thể rút gọn được cho tử số.

**Trường hợp 1.**  $a = b$ , ta thu được

$$\frac{d^2(1 + \sin \theta)}{a^2(1 - \sin \theta)} = \frac{(2r - y_0 + c) + d \sin \theta}{(y_0 - c) - d \sin \theta}$$

Ta sẽ muốn đồng nhất hệ số tự do và hệ số trước  $\sin \theta$  để dễ tính toán sau này. Do đó một cách chọn đơn giản là  $2r - y_0 + c = d$  và  $y_0 - c = d$ . Suy ra  $r = d$ . Thu gọn phương trình ta được

$$\frac{d^2(1 + \sin \theta)}{a^2(1 - \sin \theta)} = \frac{1 + \sin \theta}{1 - \sin \theta}$$

Như vậy  $a^2 = d^2$  nên  $a = d$  hoặc  $a = -d$ . Ta xét trường hợp  $a = d$ , trường hợp  $a = -d$  cũng cho kết quả tương tự (không thỏa mãn).

Ta có  $a = b = d = r$  và  $c = y_0 - d = y_0 - r$ . Phương trình đường cong trong tọa độ cực sẽ là

$$\begin{cases} x = r(\theta + \cos \theta) \\ y = (y_0 - r) + r \sin \theta \end{cases}$$

Với  $\theta = 0$  thì  $(x, y) = (r, y_0 - r)$ . Với  $\theta = \pi/2$  thì  $(x, y) = (\pi r/2, y_0)$ .

Tới đây chúng ta có thể thêm bớt một hằng số để "kéo" các tọa độ về trục.

Ta đưa tọa độ khi  $\theta = 0$  về  $Oy$  thì  $x' = x - r$ . Tương tự tọa độ khi  $\theta = \pi/2$  sẽ về  $Ox$  nên  $y' = y - y_0$ . Như vậy tọa độ (mới) cho hai trường hợp  $\theta$  là  $(0, -r)$  và  $(\pi r/2 - 1, 0)$  nhưng vì  $r$  là số dương (bán kính) nên  $(0, -r)$  nằm dưới trục  $Ox$ , không phù hợp với hình vẽ.

**Trường hợp 2.**  $a = -b$ , ta thu được

$$\frac{d^2(1 - \sin \theta)}{a^2(1 + \sin \theta)} = \frac{(2r - y_0 + c) + d \sin \theta}{(y_0 - c) - d \sin \theta}$$

Tương tự, để đồng nhất và rút gọn hệ số cho hợp với bên vế trái ta chọn  $2r - y_0 + c = -d$  và  $y_0 - c = -d$ . Suy ra  $d = -r$ . Thu gọn phương trình ta được

$$\frac{d^2(1 - \sin \theta)}{a^2(1 + \sin \theta)} = \frac{1 - \sin \theta}{1 + \sin \theta}$$

Như vậy  $a^2 = d^2$  nên  $a = d$  hoặc  $a = -d$ . Ta xét trường hợp  $a = -d$ .

Khi đó  $a = -b = -d = r$  và  $c = y_0 + d = y_0 - r$ . Phương trình đường cong trong tọa độ cực sẽ là

$$\begin{cases} x = r(\theta - \cos \theta) \\ y = (y_0 - r) - r \sin \theta \end{cases}$$

Với  $\theta = 0$  thì  $(x, y) = (-r, y_0)$ . Với  $\theta = \pi/2$  thì  $(x, y) = (r(\pi/2 - 1), y_0 - 2r)$ .

Tới đây ta cũng thêm bớt một hằng số vào hoành độ và tung độ để "kéo" các tọa độ về trục.

Ta đưa tọa độ khi  $\theta = 0$  về  $Oy$  thì  $x' = x + r$ . Tương tự ta đưa tọa độ khi  $\theta = \pi/2$  về  $Ox$  thì  $y' = y - y_0 + 2r$ . Khi đó tọa độ (mới) là  $(0, 2r)$  và  $(\pi r/2, 0)$ . Điều này phù hợp với yêu cầu bài toán và tương đương với phương trình trong tọa độ cực

$$\begin{cases} x = r(\theta - \cos \theta) + r = r(1 + \theta - \cos \theta) \\ y = (y_0 - r) - r \sin \theta - (y_0 - 2r) = r(1 - \sin \theta) \end{cases}, 0 \leq \theta \leq \frac{\pi}{2} \quad (\text{E.10})$$

Đây chính là kết quả cần tìm. Thêm nữa vị trí ban đầu của vật là  $(0, y_0)$  và tọa độ theo phương trình là  $(0, 2r)$  nên suy ra  $y_0 = 2r$ .

## Phương trình phụ thuộc thời gian

Trong phương trình đường cong có sự tham gia của bán kính  $r$  cố định và góc quét  $\theta$ . Chúng ta cần mối liên hệ giữa các phương trình theo thời gian.

Nhắc lại, vận tốc tức thời tại một điểm có phương trùng với tiếp tuyến với đường cong tại điểm đó. Do đó  $v = \frac{\sqrt{(dy)^2 + (dx)^2}}{dt}$  xác định vận tốc tức thời với quãng đường là  $(dy)^2 + (dx)^2$  là bình phương khoảng cách trong mặt phẳng. Từ đây suy ra

$$\begin{aligned} v^2 &= \left(\frac{dy}{dt}\right)^2 + \left(\frac{dx}{dt}\right)^2 = r^2 \cos^2 \theta \left(\frac{d\theta}{dt}\right)^2 + r^2 (1 + \sin \theta)^2 \left(\frac{d\theta}{dt}\right)^2 \\ &= 2r^2 (1 + \sin \theta) \left(\frac{d\theta}{dt}\right)^2 \end{aligned}$$

Từ bên trên và  $y_0 = 2r$  ta có

$$v^2 = 2g(y_0 - y) = 2g(y_0 - r + r \sin \theta) = 2gr(1 + \sin \theta)$$

Suy ra

$$2r^2 (1 + \sin \theta) \left(\frac{d\theta}{dt}\right)^2 = 2gr(1 + \sin \theta)$$

Hay

$$\left(\frac{d\theta}{dt}\right)^2 = \frac{g}{r} \Rightarrow \frac{d\theta}{dt} = \sqrt{\frac{g}{r}} = \text{const} \quad (\text{E.11})$$

Như vậy  $\theta = \sqrt{\frac{g}{r}}t = \omega t$ . Ở đây  $t$  là thời gian tính từ lúc bắt đầu thả vật từ điểm  $A$ . Cuối cùng phương trình phụ thuộc thời gian của đường cong Cycloid là

$$\begin{cases} x = r(1 + \omega t - \cos \omega t) \\ y = r(1 - \sin \omega t) \end{cases} \quad (\text{E.12})$$

Trong đó,  $r$  là bán kính cố định (bằng nửa độ cao ban đầu  $y_0$  của vật),  $\omega = \frac{g}{r}$  là tần số góc,  $y_0$  là độ cao ban đầu của vật (tung độ điểm  $A$ ).

# Phụ lục F

## Assembly, stack và heap

### F.1 Lệnh assembly cơ bản

Các lệnh hợp ngữ xử lý trên các toán hạng. Địa chỉ của toán hạng cho biết vị trí của dữ liệu được xử lý trên bộ nhớ. Một số lệnh không yêu cầu toán hạng, một số lệnh khác lại yêu cầu 1, 2 hoặc 3 toán hạng.

Về cơ bản có 3 chế độ lập địa chỉ là:

- Địa chỉ tức thời (Immediate addressing);
- Địa chỉ thanh ghi (Register addressing);
- Địa chỉ vùng nhớ (Memory addressing).

Khi lệnh có 2 toán hạng, thường thì toán hạng đầu tiên là toán hạng đích, lưu giá trị trên thanh ghi hoặc ở 1 vị trí nào đó trên vùng nhớ, còn toán hạng thứ 2 là nguồn. Nguồn thường là các hằng số (chứa trong địa chỉ tức thời) hoặc địa chỉ (trong thanh ghi hoặc bộ nhớ). Giá trị của nguồn không thay đổi sau khi tính toán.

### Nhóm lệnh truyền dữ liệu

- **mov [đích], [nguồn]**: truyền dữ liệu (word hoặc block) từ nguồn (thanh ghi hay bộ nhớ trong) tới đích;
- **load [đích], [nguồn]**: đọc dữ liệu từ bộ nhớ nguồn vào thanh ghi đích;
- **store [đích], [nguồn]**: lưu dữ liệu từ thanh ghi nguồn vào vùng nhớ ở địa chỉ đích;
- **push [nguồn]**: lưu dữ liệu từ thanh ghi vào stack (thường thấy khi gọi hàm);
- **pop**: đưa dữ liệu từ stack vào thanh ghi.

Một số lệnh và ý nghĩa:

Đích	Nguồn	Ví dụ	Ý nghĩa
Bộ nhớ	Thanh ghi	mov 100h, ax	Chuyển nội dung trong thanh ghi ax vào vị trí nhớ 100h
Thanh ghi	Bộ nhớ	mov eax, mem1	Chuyển nội dung trong vị trí nhớ mem1 vào thanh ghi eax
Thanh ghi	Thanh ghi	mov eax, ebx	Chuyển nội dung trong thanh ghi ebx vào thanh ghi eax
Thanh ghi	Hằng số	mov eax, 13h	chuyển giá trị hằng 13 ở hệ hexa vào thanh ghi eax

## Nhóm lệnh tính toán số học

Các lệnh số học bao gồm phép tính cộng, trừ, nhân, chia và đảo dấu toán hạng. Trong đó toán hạng nguồn và đích có thể là các địa chỉ khác nhau nhưng phải chứa dữ liệu có cùng độ dài (ax thì tương ứng bx còn ebx không hợp lệ,.....).

**Ví dụ 1.** Ví dụ các lệnh cộng, trừ.

```
add ax, bx // ax ← ax + bx
sub cx, 15h // cx ← cx - [15h]
add ax, 15h // ax ← ax + [15h]
```

Các lệnh số học cơ bản:

- **add**: cộng;
- **addb**: cộng số có dấu chấm động, chính xác kép;
- **sub**: trừ;
- **subd**: trừ số có dấu chấm động, chính xác kép;
- **mul**: nhân;
- **div**: chia;
- **neg**: đảo dấu toán hạng;
- **inc**: tăng lên 1;
- **dec**: giảm đi 1.

**Ví dụ 2.** Để cộng 2 số 5h và 3h sau đó lưu thanh ghi cx ta có thể làm như sau:

```
mov ax, 5h // ax ← 5h
mov bx, 3h // bx ← 3h
add ax, bx // ax ← ax + bx
mov cx, ax // cx ← ax
```

## Nhóm lệnh logic

Nhóm này gồm các phép tính logic not, and, or, xor và test.

- **not**: đảo tất cả bit trong toán hạng (0 thành 1 và 1 thành 0);
- **and/or/xor**: thực hiện phép and/or/xor trên từng cặp bit trong toán hạng nguồn và đích;
- **test**: giống and nhưng không làm thay đổi giá trị toán hạng đầu tiên (toán hạng đích).

## Nhóm các lệnh điều kiện và lệnh nhảy

Nhóm lệnh điều kiện có chức năng kiểm tra điều kiện để quyết định thực hiện lệnh kế tiếp là lệnh nào, thông thường sẽ đi kèm lệnh nhảy (nhảy tới lệnh kế tiếp khi thỏa hoặc không thỏa điều kiện).

Có hai loại lệnh nhảy:

- **Nhảy không điều kiện**: thực hiện bởi lệnh **jmp**;
- **Nhảy có điều kiện**: thực hiện bởi lệnh **j< điều kiện >** phụ thuộc vào điều kiện được xét.

Lệnh **cmp**, là câu lệnh để thực hiện việc so sánh (compare) điều kiện.

Cú pháp là **cmp [đích] [nguồn]**, tương tự như các lệnh tính toán số học, nguồn có thể là hằng, thanh ghi, vùng nhớ, còn đích thì là thanh ghi hoặc vùng nhớ.

Thông thường, việc so sánh là để quyết định việc rẽ nhánh và nhảy đến một vị trí nào đó.

1. **Lệnh nhảy không điều kiện**. Lệnh **jmp** cho phép nhảy tới một nhãn với cú pháp là **jmp lable**

**Ví dụ 3.** Dòng lệnh cuối là chỉ thị quay lại vòng lặp mô tả bởi nhãn LE003.

```
mov ax, 8h
mov bx, 9h
mov cx, 0ffh
LE003:
    add ax, 01
    sub bx, 01
    add cx, ax
    jmp LE003
```

2. **Lệnh nhảy có điều kiện**. Khi một điều kiện nhảy thỏa mãn, luồng điều khiển sẽ được đưa đến câu lệnh cần thực thi. Có nhiều loại điều kiện nhảy, ví dụ:

- **je/jz**: jump equal/jump zero, nhảy khi bằng nhau/nhảy khi kết quả so sánh bằng 0;
- **jg/jnle**: jump greater/jump not less or equal, nhảy khi lớn hơn/nhảy khi không nhỏ hơn hoặc bằng.

## F.2 Stack và Heap

Phần này sẽ sử dụng ngôn ngữ C/C++ và 1 số câu lệnh Assembly để giải thích nguyên lý của stack và heap.

Để đơn giản thì dưới đây bắt đầu với kiến trúc x86 (thanh ghi độ dài 32 bit và có tiền tố là e như eax, ebx, ...).

### Stack

Stack là cấu trúc dữ liệu vào sau ra trước. Khi hàm thực thi, nó sẽ tạo ra vùng nhớ stack cho bản thân nó (kể cả hàm main). Vùng nhớ này chứa các tham số truyền vào (trừ tham chiếu) và các biến cục bộ. Ta xét chương trình sau viết trên C:

```
#include <stdio.h>
int m = 100;
double n = 8;
int square(int x)
{
    return x * x;
}
int cube(int x)
{
    return square(x) * x;
}
int main()
{
    int a = 4, b = 8;
    int c = square(a);
    int d = cube(b);
    return 0;
}
```

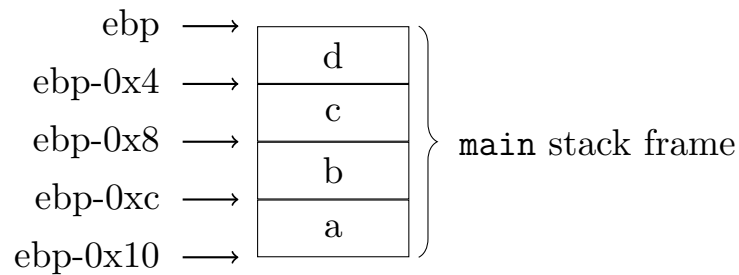
Chương trình được biên dịch thành executable file 32 bit bởi GCC phiên bản 11.4.0 bằng lệnh Linux:

```
gcc -m32 main.c -o main
```

Bước 1. Biến toàn cục. Đầu tiên, biến toàn cục m và n sẽ được lưu trong Data Segment.

Bước 2. Hàm main. Sau đó là Stack Segment. Các chương trình C/C++ sẽ được thực thi bắt đầu từ hàm main. Khi hàm main chạy, nó tạo ra vùng nhớ cho nó, đẩy các biến a, b, c, d vào stack ngược thứ tự khai báo (hình F.1).

Hình F.1 có thể được giải thích khi dùng objdump ra đoạn assembly dưới

Hình F.1. Hàm `main`

đây.

```

000011c0 <main>:
    push    ebp
    mov     ebp, esp
    sub     esp, 0x10
    call    1201 <__x86.get_pc_thunk.ax>
    add     eax, 0x2e11
    mov     DWORD PTR [ebp-0x10], 0x4
    mov     DWORD PTR [ebp-0xc], 0x8
    push    DWORD PTR [ebp-0x10]
    call    118d <square>
    add     esp, 0x4
    mov     DWORD PTR [ebp-0x8], eax
    push    DWORD PTR [ebp-0xc]
    call    11a2 <cube>
    add     esp, 0x4
    mov     DWORD PTR [ebp-0x4], eax
    mov     eax, 0x0
    leave
    ret

```

Biến `a` nằm ở vị trí `0x10` và biến `b` nằm ở vị trí `0xc`. Vậy thì `a` nằm thấp hơn `b`. Điều này nghĩa là stack đi từ trên xuống.

Sau đó, giá trị 4 được chép qua biến `a`, giá trị 8 được chép qua biến `b`. Kế tiếp, trong hàm `main` gọi hàm `square`. Khi đó các tham số cần thiết cho hàm `square` sẽ được nạp vào stack (ở đây là biến `a` ở `ebp-0x10`) và hàm `square` sẽ được nạp vào stack F.2.

Bước 3. Gọi hàm `square`. Hàm `square` có một tham số truyền vào là `a` ở địa chỉ `ebp+0x8` (`ebp` mới của hàm `square`, không phải của `main`).

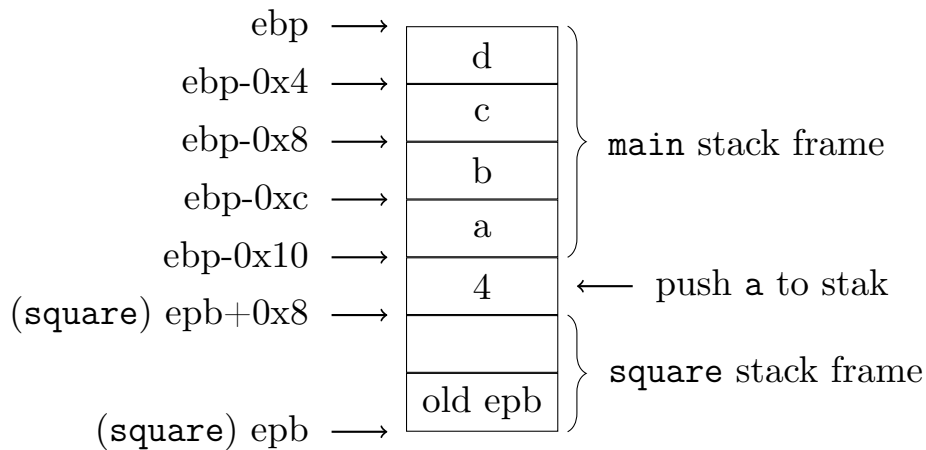
Ở đây, chương trình đưa giá trị 4 (ở `ebp+0x8`) vào thanh ghi `eax` và bình phương (lệnh `imult`). Kết quả trả về của hàm được gán trong thanh ghi `eax`.

```

0000118d <square>:
    push    ebp

```



Hình F.2. Gọi hàm `square`

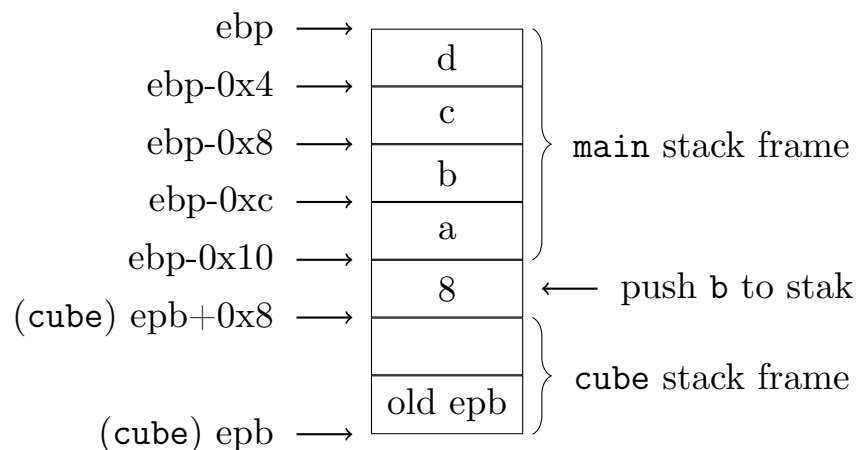
```

mov     ebp, esp
call    1201 <__x86.get_pc_thunk.ax>
add     eax, 0x2e47
mov     eax, DWORD PTR [ebp+0x8]
imul    eax, eax
pop     ebp
ret

```

Sau khi hàm `square` chạy xong, nó gán giá trị 16 vào biến `c` (từ thanh ghi `eax` vào địa chỉ `ebp-0x8`), lúc này trong stack trở lại như lúc chưa có hàm `square`.

Bước 4. Gọi hàm `cube`. Khi hàm `main` gọi hàm `cube`, nó thực hiện giống như cho hàm `square` ở trên, tức là tạo stack frame cho `cube` (hình F.3).

Hình F.3. Hàm `main` gọi hàm `cube`

```

000011a2 <cube>:
        push    ebp
        mov     ebp, esp
        call    1201 <__x86.get_pc_thunk.ax>

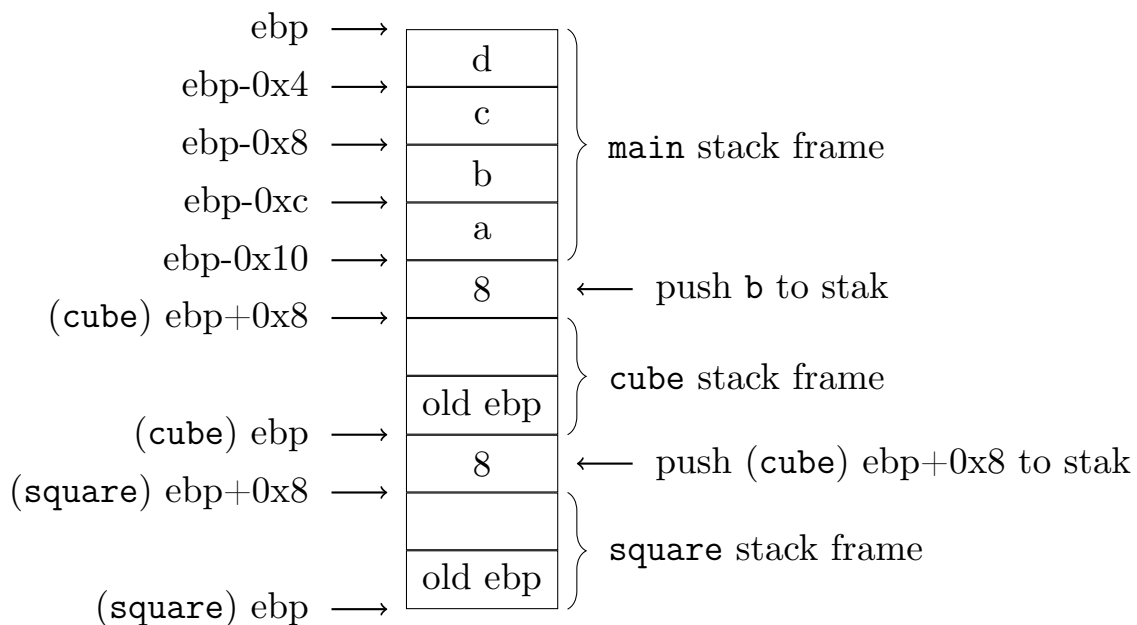
```

```

add    eax,0x2e32
push   DWORD PTR [ebp+0x8]
call   118d <square>
add    esp,0x4
imul   eax,DWORD PTR [ebp+0x8]
leave
ret
    
```

Tiếp theo, bên trong hàm `cube` lại gọi hàm `square` nên hàm `cube` sau khi đẩy các tham số cho hàm `square` vào stack thì gọi hàm `square` (hình F.4).

Sau khi hàm `square` thực hiện tính toán, giá trị trả về của nó được lưu trong `eax`. Sau đó hàm `cube` lấy kết quả ở `eax` (giá trị trả về của `square`) nhân với `ebp+0x8` (tham số, ở đây là 8). Kết quả cuối cùng vẫn được lưu trữ ở `eax` để hàm `main` sử dụng.



Hình F.4. Hàm `cube` gọi hàm `square`

Cứ mỗi lần gọi hàm, hàm được gọi sẽ được nạp vào stack, và khi số lượng hàm quá lớn (có thể do đệ quy quá nhiều) sẽ gây tràn stack và gây ra lỗi stack overflow (tràn stack).

Do tất cả biến cục bộ được lưu trữ trong stack, nên trong mỗi hàm chỉ có thể có một số lượng biến nhất định. Vì vậy độ dài mảng cấp phát được thường là khá ít. Từ đó, một loại vùng nhớ được sử dụng để khắc phục nhược điểm này là heap.