Học viện Công nghệ Bưu chính Viễn thông
Khoa Công nghệ thông tin 1

TOÁN RỜI RẠC 1

TS. Đào Thị Thúy Quỳnh

Vai trò của toán rời rạc trong CNTT

- Là lĩnh vực nghiên cứu cơ bản, đóng vai trò quan trọng trong nhiều lĩnh vực khác của CNTT
  - Trí tuệ nhân tạo
  - Thuật toán
  - Lý thuyết tối ưu
  - ...
- Là môn học cơ sở trong ngành CNTT, nền tảng của nhiều môn học khác
  - Các môn học lập trình
  - Cấu trúc dữ liệu và giải thuật
  - Nhập môn trí tuệ nhân tạo
  - Cơ sở dữ liệu
  - ...

Nội dung môn học (1/5)

- Phần 1: Một số kiến thức cơ bản
  - Lý thuyết tập hợp
  - Logic mệnh đề
  - Logic vị từ
  - Thuật toán và độ phức tạp
  - Bài tập

Nội dung môn học (2/5)

- **Phần 2: Bài toán đếm**
  - Giới thiệu bài toán
  - Các nguyên lý đếm cơ bản
  - Quy về bài toán con
  - Hệ thức truy hồi
  - Phương pháp hàm sinh
  - Bài tập

Nội dung môn học (3/5)

- Phần 3: Bài toán liệt kê
  - Giới thiệu bài toán
  - Phương pháp sinh
  - Phương pháp quay lui
  - Bài tập

Nội dung môn học (4/5)

- Phần 4: Bài toán tối ưu
  - Giới thiệu bài toán
  - Thuật toán duyệt toàn bộ
  - Thuật toán nhánh cận
  - Bài tập

Thông tin môn học

- Giảng viên
  - TS. Đào Thị Thúy Quỳnh, Bộ môn KHMT, Khoa CNTT1
  - Email: thuyquynhtn90@gmail.com
- Tài liệu tham khảo:
  - Nguyễn Duy Phương. Giáo trình Toán rời rạc. Học viện Công nghệ Bưu chính Viễn thông
  - Nguyễn Đức Nghĩa, Nguyễn Tô Thành. Toán rời rạc. Nhà xuất bản giáo dục,2005.
  - Đỗ Đức Giáo. Toán rời rạc. Nhà xuất bản ĐHQG, 2003.
- Đánh giá
  - Chuyên cần (10%)
  - Bài tập (10%)
  - Kiểm tra giữa kỳ (10%)
  - Thi cuối kỳ (70%)

Thiếu điểm thành phần hoặc nghỉ quá 20% số buổi sẽ không được thi hết môn!!!

TOÁN RỜI RẠC 1

Một số kiến thức cơ bản

Nội dung

- Lý thuyết tập hợp
- Logic mệnh đề
- Logic vị từ
- Thuật toán và độ phức tạp
- Bài tập

Một số ký hiệu và tập hợp

- Tập hợp: $A, B, \dots, X, Y, \dots$
- Phần tử của tập hợp: $a, b, \dots, x, y, \dots$
- Phần tử x thuộc (không thuộc) A: $x \in A,\ \ x \notin A$
- Số phần tử của tập hợp A: $|A|$
  - Một tập hợp có n phần tử được gọi là một n-tập
- Tập hợp con: $A \subseteq B$
  - $x \in A \Rightarrow x \in B$
- Tập hợp bằng nhau: nếu $A \subseteq B$ và $B \subseteq A$ thì $A = B$
- Tập rỗng: $\emptyset$
  - Không có phần tử nào
  - Là con của mọi tập hợp

Các phép toán trên tập hợp

- Phần bù của A trong X: $\bar{A} = \{x \in X | x \notin A\}$
- Hợp của hai tập hợp: $A \cup B = \{x | x \in A \text{ hoặc } x \in B\}$
- Giao của hai tập hợp: $A \cap B = \{x | x \in A \text{ và } x \in B\}$
- Hiệu của hai tập hợp: $A \setminus B = \{x | x \in A \text{ và } x \notin B\}$
- Luật kết hợp:

$$(A \cup B) \cup C = A \cup (B \cup C)$$
$$(A \cap B) \cap C = A \cap (B \cap C)$$

- Luật giao hoán: $A \cup B = B \cup A, A \cap B = B \cap A$
- Luật phân bố:

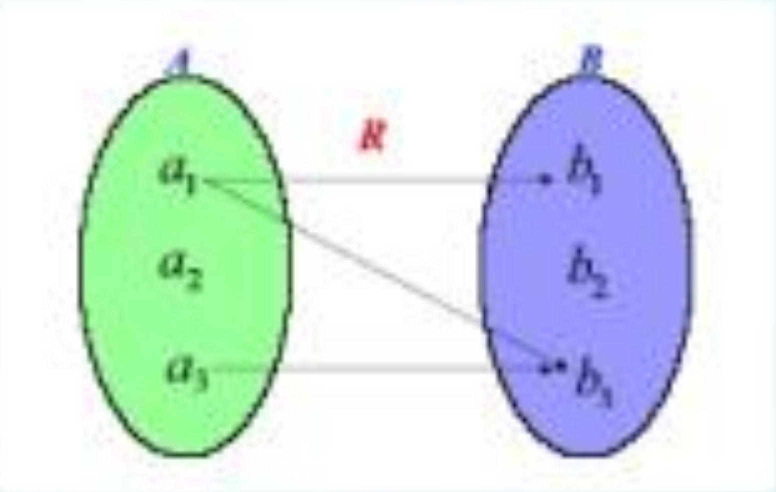
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$
$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

- Luật đối ngẫu: $\overline{A \cup B} = \bar{A} \cap \bar{B}, \overline{A \cap B} = \bar{A} \cup \bar{B}$
- Tích Đề các: $A \times B = \{(a, b) | a \in A, b \in B\}$

Quan hệ

Một *quan hệ hai ngôi* từ tập A đến tập B là tập con của tích Đề các $R \subseteq A \times B$. Chúng ta sẽ viết $a\,R\,b$ thay cho $(a, b) \in R$.

Quan hệ từ A đến chính nó được gọi là quan hệ trên A



$$R = \{ (a_1, b_1), (a_1, b_3), (a_3, b_3) \}$$

Quan hệ

- **Quan hệ**: một quan hệ hai ngôi R trên tập X, $R(X)$, là một tập con của tích Đề các $X \times X$
- Tính chất của quan hệ:
  - **Phản xạ**: mọi phần tử có quan hệ với chính nó
  - **Đối xứng**: a có quan hệ với b kéo theo b có quan hệ với a
  - **Kéo theo**: a có quan hệ với b và b có quan hệ với c kéo theo a có quan hệ với c
- Ví dụ
  - $X = \{1,2,3,4\}$
  - $a, b \in X, a$ có quan hệ R đối với b nếu a chia hết cho b
  - $R(X) = \{(1,1), (2,1), (2,2), (3,1), (3,3), (4,1), (4,2), (4,4)\}$
    - **Phản xạ, kéo theo, nhưng không đối xứng**

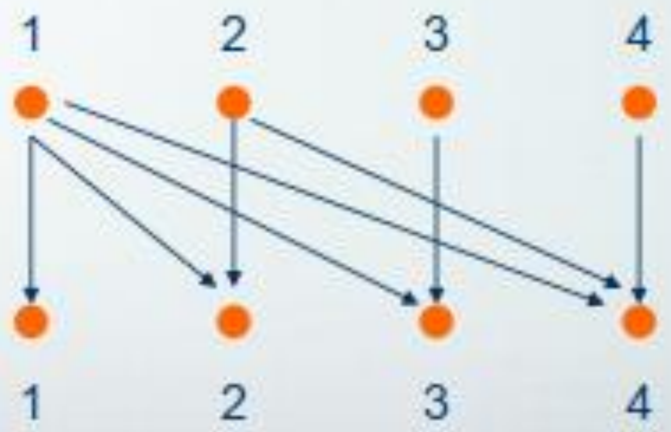
Quan hệ

Ví dụ. Cho $A = \{1, 2, 3, 4\}$, và

$$R = \{(a, b) \mid a \text{ là ước của } b\}$$

Khi đó

$$R = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 4), (3, 3), (4,4)\}$$



Biểu diễn Quan hệ

Cho R là quan hệ từ $A = \{1,2,3,4\}$ đến $B = \{u,v,w\}$:

$$R = \{(1,u),(1,v),(2,w),(3,w),(4,u)\}.$$

Khi đó R có thể biểu diễn như sau

	u	v	w
1	1	1	0
2	0	0	1
3	0	0	1
4	1	0	0

Dòng và cột tiêu đề có thể bỏ qua nếu không gây hiểu nhầm.

Đây là ma trận cấp 4×3 biểu diễn cho quan hệ R

Biểu diễn Quan hệ

Định nghĩa. Cho R là quan hệ từ $A = \{a_1, a_2, \ldots, a_m\}$ đến $B = \{b_1, b_2, \ldots, b_n\}$. Ma trận biểu diễn của R là ma trận cấp $m \times n$ $M_R = [m_{ij}]$ xác định bởi

$$m_{ij} = \begin{cases} 0 & \text{nếu } (a_i, b_j) \notin R \\ 1 & \text{nếu } (a_i, b_j) \in R \end{cases}$$

Ví dụ. Nếu R là quan hệ từ $A = \{1, 2, 3\}$ đến $B = \{1, 2\}$ sao cho $a\,R\,b$ nếu $a > b$. Khi đó ma trận biểu diễn của R là

	1	2
1	**0**	**0**
2	**1**	**0**
3	**1**	**1**

Biểu diễn Quan hệ

$$m_{ij} = \begin{cases} 1 & \text{nếu } (a_i, b_j) \in R \\ 0 & \text{nếu } (a_i, b_j) \notin R \end{cases}$$

Ví dụ. Cho R là quan hệ từ $A = \{a_1, a_2, a_3\}$ đến $B = \{b_1, b_2, b_3, b_4, b_5\}$ được biễu diễn bởi matrận

$$\mathbf{M}_R = \begin{array}{c} \begin{matrix} b_1 & b_2 & b_3 & b_4 & b_5 \end{matrix} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array} \begin{matrix} a_1 \\ a_2 \\ a_3 \end{matrix}$$

Khi đó R gồm các cặp:

$\{(a_1, b_2), (a_2, b_1), (a_2, b_3), (a_2, b_4), (a_3, b_1), (a_3, b_3), (a_3, b_5)\}$

Biểu diễn Quan hệ

- Cho *R* là quan hệ trên tập *A*, khi đó M_R là *ma trận vuông.*

- *R* là *phản xạ* nếu tất cả các phần tử trên *đường chéo* của M_R đều bằng1: $m_{ii} = 1$ với mọi *i*

	u	v	w
u	1	1	0
v	0	1	1
w	0	0	1

Biểu diễn Quan hệ

R* là đối xứng nếu M_R là *đối xứng

$$m_{ij} = m_{ji} \qquad \text{for all } i, j$$

	u	v	w
u	1	0	1
v	0	0	1
w	1	1	0

Quan hệ tương đương

Định nghĩa. Quan hệ R trên tập A được gọi là ***tương đương*** nếu nó có tính chất phản xạ, đối xứng và bắc cầu :

Ví dụ. Quan hệ R trên các chuỗi ký tự xác định bởi aRb nếu a và b có cùng độ dài. Khi đó R là quan hệ tương đương.

Ví dụ. Cho R là quan hệ trên $\mathbf{R}$ sao cho aRb nếu $a-b$ nguyên. Khi đó R là quan hệ tương đương

Một số nguyên lý cơ bản

Nguyên lý cộng

Giả sử để làm công việc A có 2 phương pháp

- Phương pháp 1 có n cách làm
- Phương pháp 2 có m cách làm

Khi đó số cách làm công việc A là n+m

Ví dụ. An có 3 áo tay dài, 5 áo tay ngắn. Để chọn 1 cái áo thì An có mấy cách

Nguyên lý cộng

- Nếu A và B là hai tập rời nhau thì
$$|A \cup B| = |A| + |B|$$
- Nếu $\{A_1, A_2, \ldots, A_k\}$ là một phân hoạch của tập hợp X thì
$$|X| = |A_1| + |A_2| + \cdots + |A_k|$$

- Ví dụ
  - Một đoàn VĐV gồm 2 môn bắn súng và bơi. Nam có 10 người. Số VĐV thi bắn súng là 14. Số nữ VĐV thi bơi bằng số nam VĐV thi bắn súng. Hỏi toàn đoàn có bao nhiêu VĐV?

Một số nguyên lý cơ bản

Nguyên lý nhân

Giả sử để làm công việc A cần thực hiện 2 bước

- Bước 1 có n cách làm
- Bước 2 có m cách làm

Khi đó số cách làm công việc A là n.m

Ví dụ:

Có 3.2 =6 con đường đi từ A đến C

Nguyên lý nhân

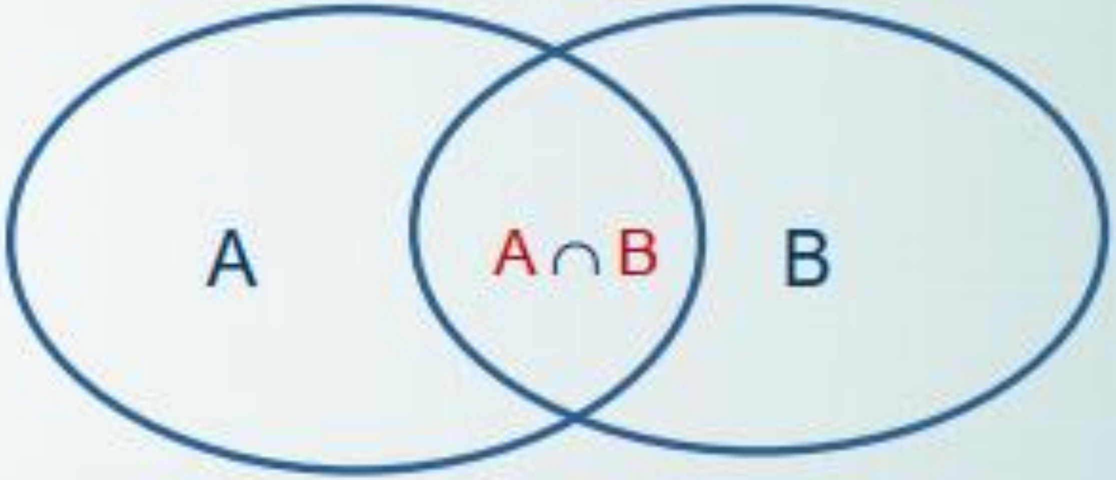
- Nếu mỗi thành phần a_i của bộ có thứ tự k thành phần $(a_1, a_2, \dots, a_k)$ có n_i khả năng chọn, thì số bộ được tạo ra sẽ là tích các khả năng $n_1 n_2 \dots n_k$
- Hệ quả:
  - $|A_1 \times A_2 \times \cdots \times A_k| = |A_1||A_2| \dots |A_k|$
  - $|A^k| = |A|^k$
- Ví dụ
  - Có bao nhiêu số nguyên dương có 5 chữ số được thành lập bởi các chữ số 0,1,2?

Một số nguyên lý cơ bản

Nguyên lý bù trừ.

Cho A và B là hai tập hữu hạn. Khi đó

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



Nhắc lại về lý thuyết tổ hợp

- **Chỉnh hợp:**
  - Chỉnh hợp lặp: Một chỉnh hợp lặp chập k của n phần tử là bộ **có thứ tự** gồm k phần tử lấy từ n phần tử của tập đã cho, mỗi phần tử có thể lấy lặp lại
  - Số lượng chỉnh hợp lặp chập k của n phần tử là:

$$A_n^k = n^k$$

- **Ví dụ:** Từ tập Q={a,b,c} có thể đặt bao nhiêu tên biến có độ dài =4.

Nhắc lại về lý thuyết tổ hợp

- **Chỉnh hợp:**
  - Chỉnh hợp không lặp: Một chỉnh hợp không lặp chập k của n phần tử là bộ **có thứ tự** gồm k phần tử lấy từ n phần tử của tập đã cho, mỗi phần tử không được lấy lặp lại

$$P_n^k = n.(n-1)(n-2)\ldots(n-k+1) = \frac{n!}{(n-k)!}$$

  - **Ví dụ:** Có bao nhiêu số có 4 chữ số khác nhau được chọn từ tập {1,3,4,5,6,7}

Nhắc lại về lý thuyết tổ hợp

- **Hoán vị:** ta gọi các hoán vị của n phần tử là một cách xếp có thứ tự các phần tử đó. Số các hoán vị của tập n phần tử có thể coi là trường hợp riêng của chỉnh hợp không lặp với k=n.

$$P_n^n = n.(n-1)(n-2)\ldots 1 = n!$$

- **Ví dụ:** Có bốn người rủ nhau đi chụp ảnh là Anh, Bắc, Cúc, Dương. Hãy tính có bao nhiêu kiểu ảnh chụp mà tất cả bốn người đứng thành một hàng?

Nhắc lại về lý thuyết tổ hợp

- **Tổ hợp:**
  - Tổ hợp không lặp: Một tổ hợp không lặp chập k của n phần tử là cách chọn **không phân biệt thứ tự** k phần tử từ tập n phần tử, mỗi phần tử không được lấy lặp lại.

$$C_n^k = \frac{n!}{(n-k)!\,k!}$$

  - **Ví dụ:** Có 12 đội bóng tham dự giải chuyên nghiệp quốc gia, các đội thi đấu vòng tròn một lượt. Hỏi có bao nhiêu trận đấu được tổ chức?

Nhắc lại về lý thuyết tổ hợp

- **Tổ hợp:**

  - **Tổ hợp lặp:** Mỗi cách chọn ra k vật từ n loại vật khác nhau (trong đó mỗi loại vật có thể được chọn lại nhiều lần) được gọi là tổ hợp lặp chập k của n

$$R_n^k = C_{n+k-1}^k = \frac{(n+k-1)!}{(n-1)!\,k!}$$

Nội dung

- Lý thuyết tập hợp
- Logic mệnh đề
- Logic vị từ
- Thuật toán và độ phức tạp
- Bài tập

Một số khái niệm của logic mệnh đề

- **Mệnh đề**: là một câu khẳng định hoặc đúng hoặc sai.
- Ví dụ:
  - "Hà Nội là thủ đô của Việt Nam" là một mệnh đề đúng.
  - "(5 < 3)" là một mệnh đề sai, "(5 > 3)" là một mệnh đề đúng.
  - "(a<7)" không phải là mệnh đề vì nó không biết khi nào đúng khi nào sai.
- **Giá trị chân lý của mệnh đề**: mỗi mệnh đề chỉ có một trong 2 giá trị "**đúng**", ký hiệu là "**T**", giá trị "**sai**", ký hiệu là "**F**". Tập { T, F} được gọi là giá trị chân lý của mệnh đề.
- Ký hiệu**:** ta ký hiệu mệnh đề bằng các chữ cái in thường (a, b, p, q, r, s, t). *Mỗi* mệnh đề còn được gọi là một công thức. Từ khái niệm về mệnh đề, giá trị chân lý của mỗi mệnh đề, ta xây dựng nên các mệnh đề phức hợp (được gọi là công thức) thông qua các phép toán trên mệnh đề.

Một số khái niệm về logic mệnh đề

Kiểm tra các khẳng định sau có phải là mệnh đề không?

- Paris là thành phố của Mỹ.
- n là số tự nhiên.
- con nhà ai mà xinh thế!
- 3 là số nguyên tố.
- Toán rời rạc là môn bắt buộc của ngành Tin học.
- Bạn có khỏe không?
- x^2 +1 luôn dương.

Các phép toán của logic mệnh đề (1/2)

Cho p và q là hai mệnh đề

- Phép phủ định
  - $\neg p$ là ký hiệu mệnh đề, đọc là "Không phải p"
  - Mệnh đề cho giá trị đúng nếu p sai và cho giá trị sai nếu p đúng
- Phép hội
  - $p \wedge q$ là ký hiệu mệnh đề, đọc là "p và q"
  - Mệnh đề có giá trị đúng khi cả p và q có giá trị đúng, có giá trị sai trong các trường hợp khác còn lại
- Phép tuyển
  - $p \vee q$ là ký hiệu mệnh đề, đọc là "p hoặc q"
  - Mệnh đề có giá trị sai khi cả p và q có giá trị sai, có giá trị đúng trong các trường hợp khác còn lại

Các phép toán của logic mệnh đề (2/2)

- Phép tuyển loại
  - $p \oplus q$ là ký hiệu mệnh đề đọc là "hoặc p hoặc q"
  - Mệnh đề có giá trị đúng khi một trong p hoặc q có giá trị đúng, có giá trị sai trong các trường hợp khác còn lại
- Phép kéo theo
  - $p \Rightarrow q$ là ký hiệu mệnh đề đọc là "p kéo theo q"
  - Mệnh đề có giá trị sai khi p đúng và q sai, có giá trị đúng trong các trường hợp khác còn lại
- Phép tương đương
  - $p \Leftrightarrow q$ là ký hiệu mệnh đề đọc là "p tương đương q"
  - Mệnh đề có giá trị đúng khi p và q có cùng giá trị chân lý, có giá trị sai trong các trường hợp khác còn lại

Bảng giá trị chân lý

Bảng giá trị chân lý các phép toán mệnh đề							
p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \Rightarrow q$	$p \Leftrightarrow q$
T	T	F	T	T	F	T	T
T	F	F	F	T	T	F	F
F	T	T	F	T	T	T	F
F	F	T	F	F	F	T	T

Các phép toán cấp bít ứng dụng trong ngôn ngữ LT				
Giá trị của A	Giá trị của B	A **and** B	A **or** B	A **xor** B
A = 13 =1101	B = 8=1000	1000	1101	0101

Một số khái niệm

- Thỏa được
  - Một mệnh đề là thỏa được nếu nó đúng với một bộ giá trị chân lý nào đó của các mệnh đề thành phần
- Không thỏa được
  - Một mệnh đề là không thỏa được nếu nó sai với mọi bộ giá trị chân lý của các mệnh đề thành phần
- Vững chắc
  - Một mệnh đề là vững chắc nếu nó đúng với mọi bộ giá trị chân lý của các mệnh đề thành phần

Các mệnh đề tương đương logic (1/2)

- Hai mệnh đề a và b được gọi là tương đương logic nếu chúng có cùng giá trị chân lý với mọi bộ giá trị chân lý của các mệnh đề thành phần
  - Ký hiệu: $a \equiv b$

- Một số mệnh đề tương đương cơ bản
  - $a \vee F \equiv a$
  - $a \wedge F \equiv F$
  - $a \vee T \equiv T$
  - $a \wedge T \equiv a$
  - $a \Rightarrow b \equiv \neg a \vee b$
  - $a \Leftrightarrow b \equiv (a \Rightarrow b) \wedge (b \Rightarrow a)$
  - $\neg(\neg a) \equiv a$

Các mệnh đề tương đương logic (2/2)

- Luật giao hoán
  - $a \vee b \equiv b \vee a$
  - $a \wedge b \equiv b \wedge a$
- Luật kết hợp
  - $(a \vee b) \vee c \equiv a \vee (b \vee c)$
  - $(a \wedge b) \wedge c \equiv a \wedge (b \wedge c)$
- Luật phân phối
  - $a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c)$
  - $a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$
- Luật De Morgan
  - $\neg(a \vee b) \equiv \neg a \wedge \neg b$
  - $\neg(a \wedge b) \equiv \neg a \vee \neg b$

Dạng chuẩn tắc hội (1/2)

- Một câu (mệnh đề) tuyển là tuyển của các mệnh đề nguyên thủy
  - Câu tuyển có dạng $p_1 \vee p_2 \vee \ldots \vee p_n$ trong đó p_i là các mệnh đề nguyên thủy
- Một công thức ở dạng chuẩn tắc hội nếu nó là hội của các câu tuyển
  - $(a \vee e \vee f \vee g) \wedge (b \vee c \vee d)$

Dạng chuẩn tắc hội (2/2)

- Ta có thể biến đổi một công thức bất kỳ về dạng chuẩn tắc hội bằng cách biến đổi theo nguyên tắc sau:
  - Khử các phép tương đương: $a \Leftrightarrow b \equiv (a \Rightarrow b) \land (b \Rightarrow a)$
  - Khử các phép kéo theo: $a \Rightarrow b \equiv \neg a \lor b$
  - Chuyển các phép phủ định vào sát các ký hiệu mệnh đề bằng cách áp dụng luật De Morgan
  - Khử phủ định kép: $\neg(\neg a) \equiv a$
  - Áp dụng luật phân phối: $a \lor (b \land c) \equiv (a \lor b) \land (a \lor c)$

Bài tập 1

- Sử dụng phương pháp bảng chân lý chứng minh sự tương đương logic giữa các mệnh đề
  - Các mệnh đề tương đương cơ bản
  - Các luật

- Sử dụng các mệnh đề tương đương cơ bản và các luật (giao hoán, kết hợp, phân phối, De Morgan) chứng minh sự tương đương logic giữa các mệnh đề

$$\neg(p \vee (\neg p \wedge q)) \equiv \neg p \wedge \neg q$$

Bài tập 2

- Chứng minh các mệnh đề sau là vững chắc

a) $(p \wedge q) \Rightarrow p$

b) $p \Rightarrow (p \vee q)$

c) $\neg p \Rightarrow (p \Rightarrow q)$

d) $(p \wedge q) \Rightarrow (p \Rightarrow q)$

e) $\neg(p \Rightarrow q) \Rightarrow p$

f) $\neg(p \Rightarrow q) \Rightarrow \neg q$

g) $\neg p \wedge (p \vee q) \Rightarrow q$

h) $(p \Rightarrow q) \wedge (q \Rightarrow r) \Rightarrow (p \Rightarrow r)$

i) $(p \wedge (p \Rightarrow q)) \Rightarrow q$

j) $((p \vee q) \wedge (p \Rightarrow r) \wedge (q \Rightarrow r)) \Rightarrow r$

Bài tập 3

- Chứng minh các tương đương logic sau

1) $(p \Leftrightarrow q) \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$

2) $\neg p \Leftrightarrow q \equiv p \Leftrightarrow \neg q$

3) $\neg(p \Leftrightarrow q) \equiv \neg p \Leftrightarrow q$

Bài tập 4

- Chuẩn hóa về dạng chuẩn tắc hội

$$(p \Rightarrow q) \vee \neg(r \vee \neg s)$$

Nội dung

- Lý thuyết tập hợp
- Logic mệnh đề
- Logic vị từ
- Thuật toán và độ phức tạp
- Bài tập

Cú pháp của logic vị từ (1/4)

- Các ký hiệu
  - Các ký hiệu hằng: a, b, c, An, Ba, John, ...
  - Các ký hiệu biến: x, y, z, u, v, w, ...
  - Các ký hiệu vị từ: P, Q, R, S, Like, Friend, ...
    - Mỗi vị từ là vị từ của n biến ($n \geq 0$)
    - Vị từ không biến là các ký hiệu mệnh đề
  - Các ký hiệu hàm: f, g, cos, sin, mother, husband, ...
    - Mỗi hàm là hàm của n biến ($n \geq 0$)
  - Các ký hiệu kết nối logic: $\land$ (hội), $\lor$ (tuyển), $\neg$ (phủ định), $\Rightarrow$ (kéo theo), $\Leftrightarrow$ (kéo theo nhau)
  - Các ký hiệu lượng tử: $\forall$ (mọi) , $\exists$ (tồn tại)
  - Các ký hiệu ngăn cách: dấu phẩy, mở ngoặc, đóng ngoặc

Cú pháp của logic vị từ (2/4)

- Các hạng thức (term)
  - Là các biểu thức mô tả đối tượng, được xác định đệ quy như sau
    - Các ký hiệu hằng và các ký hiệu biến là hạng thức
    - Nếu $t_1, t_2, \ldots, t_n$ là n hạng thức, và f là một ký hiệu hàm n biến thì $f(t_1, t_2, \ldots, t_n)$ hạng thức
  - Một hạng thức không chứa biến được gọi là một hạng thức cụ thể (ground term)
  - Hai hạng thức bằng nhau nếu cùng tương ứng với một đối tượng
    - Father(John) = Mike
- Công thức nguyên tử (câu đơn)
  - Biểu diễn tính chất của đối tượng, hoặc quan hệ giữa các đối tượng, được xác định đệ quy như sau
    - Các ký hiệu vị từ không biến (mệnh đề) là công thức nguyên tử
    - Nếu $t_1, t_2, \ldots, t_n$ là n hạng thức, và P là vị từ của n biến thì $P(t_1, t_2, \ldots, t_n)$ công thức nguyên tử

Cú pháp của logic vị từ (3/4)

- **Công thức**
  - Được xây dựng từ công thức nguyên tử, sử dụng các kết nối logic và các lượng tử, theo đệ quy như sau
    - Các công thức nguyên tử là công thức
    - Nếu G và H là các công thức, thì các biểu thức sau là công thức
      - $(G \wedge H)$, $(G \vee H)$, $(\neg G)$, $(G \Rightarrow H)$, $(G \Leftrightarrow H)$
    - Nếu G là công thức và x là biến thì các biểu thức sau là công thức
      - $(\forall x G)$, $(\exists x G)$
- **Một số quy ước**
  - Các công thức không phải công thức nguyên tử gọi là công *thức phức hợp* (câu phức hợp)
  - Công thức không chứa biến gọi là công *thức cụ thể*
  - Khi viết công thức ta bỏ đi các dấu ngoặc không cần thiết

Cú pháp của logic vị từ (4/4)

- Lượng tử phổ dụng (∀)
  - Mô tả tính chất của cả một lớp các đối tượng, mà không cần liệt kê các đối tượng ra
  - $\forall x(Elephant(x) \Rightarrow Color(x, Gray))$
- Lượng tử tồn tại (∃)
  - Cho phép tạo ra câu nói đến một đối tượng nào đó trong một lớp đối tượng, có tính chất hoặc thỏa mãn một quan hệ nào đó
  - $\exists x(Student(x) \wedge Inside(x, P301))$
- Literal
  - Là công thức nguyên tử hoặc phủ định của công thức nguyên tử
  - $Play(x, Football)$, $\neg Like(Lan, Rose)$
- Câu tuyển
  - Là tuyển của các literal
  - $Male(x) \vee \neg Like(x, Football)$

Ngữ nghĩa của logic vị từ (1/3)

- Minh họa
  - Là một cách gán cho các biến đối tượng một đối tượng cụ thể, gán cho các ký hiệu hàm một hàm cụ thể, và các ký hiệu vị từ một vị từ cụ thể
  - Ý nghĩa của công thức trong một thế giới hiện thực nào đó
- Ngữ nghĩa của câu đơn
  - Trong một minh họa, mỗi câu đơn sẽ chỉ định một sự kiện cụ thể, có thể đúng (True) hoặc sai (False)
    - $Student(Lan)$
- Ngữ nghĩa của câu phức
  - Được xác định dựa trên ngữ nghĩa của các câu đơn và các kết nối logic
    - $Student(Lan) \wedge Like(An, Rose)$
    - $Like(An, Rose) \vee \neg Like(An, Tulip)$

Ngữ nghĩa của logic vị từ (2/3)

- Ngữ nghĩa của câu chứa lượng từ
  - Công thức $\forall x G$ là đúng nếu và chỉ nếu mọi công thức nhận được từ G bằng cách thay x bởi một đối tượng trong miền đối tượng đều đúng
    - Ví dụ: Miền đối tượng {An, Ba, Lan}, ngữ nghĩa của câu $\forall x Student(x)$ được xác định là ngữ nghĩa của câu
    - $Student(An) \land Student(Ba) \land Student(Lan)$
  - Công thức $\exists x G$ là đúng nếu và chỉ nếu một trong các công thức nhận được từ G bằng cách thay x bởi một đối tượng trong miền đối tượng đều đúng
    - Ví dụ: ngữ nghĩa của câu $\exists x Student(x)$ được xác định là ngữ nghĩa của câu
    - $Student(An) \lor Student(Ba) \lor Student(Lan)$
- Các khái niệm công thức thỏa được, không thỏa được, vững chắc, mô hình, tương tự logic mệnh đề

Ngữ nghĩa của logic vị từ (3/3)

- Các lượng tử lồng nhau
  - Có thể sử dụng đồng thời nhiều lượng tử trong câu phức hợp

$$\forall x \forall y Sibling(x, y) \Rightarrow Relationship(x, y)$$
$$\forall x \exists y Love(x, y)$$

  - Nhiều lượng tử cùng loại có thể được viết gọn bằng một ký hiệu lượng tử

$$\forall x, y Sibling(x, y) \Rightarrow Relationship(x, y)$$

  - Không được phép thay đổi các lượng tử khác loại trong câu

$\forall x \exists y Love(x, y)$ Mọi người đều có ai đó yêu

$\exists y \forall x Love(x, y)$ Có ai đó mà tất cả mọi người đều yêu

Bài tập

- Chuyển các câu sau sang logic vị từ cấp 1
  1. An không cao
  2. An không cao nhưng bố An cao
  3. An và Ba là anh em
  4. Tất cả nhà nông đều thích mặt trời
  5. Mọi cây nấm đỏ đều không có độc
  6. Có học sinh thích hoa hồng

Nội dung

- Lý thuyết tập hợp
- Logic mệnh đề
- Logic vị từ
- Thuật toán và độ phức tạp
- Bài tập

Khái niệm thuật toán

- Thuật toán hoặc giải thuật (algorithm)
  - Là một thủ tục giải quyết một vấn đề nào đó trong một số hữu hạn bước
  - Là một tập hữu hạn các chỉ thị được định nghĩa rõ ràng để giải quyết một vấn đề nào đó
  - Là một tập các quy tắc định nghĩa chính xác một dãy các hành động

- Mô tả thuật toán
  - Sử dụng ngôn ngữ tự nhiên
  - Sử dụng dạng giả mã (Pseudo-code)
  - Sử dụng ngôn ngữ lập trình

Ví dụ thuật toán (1/2)

- **Thuật toán**: Tìm số nguyên lớn nhất trong danh sách n số nguyên (chưa sắp xếp)

- Mô tả thuật toán sử dụng ngôn ngữ tự nhiên
  - Nếu không có số nào trong danh sách thì không có số lớn nhất
  - Giải sử số đầu tiên là số lớn nhất trong danh sách
  - Với mỗi số còn lại trong danh sách, nếu số này lớn hơn số lớn nhất hiện tại thì coi số này là số lớn nhất trong danh sách
  - Khi tất cả các số trong danh sách đều đã được xem xét, số lớn nhất hiện tại là số lớn nhất trong danh sách

Dài dòng, ít được sử dụng

Ví dụ thuật toán (2/2)

- **Thuật toán**: Tìm số nguyên lớn nhất trong danh sách n số nguyên (chưa sắp xếp)

- Mô tả thuật toán sử dụng dạng giả mã

Algorithm LargestNumber
Input: A list of numbers L.
Output: The largest number in the list L.

```
largest ← null
for each item in L do
    if item > largest then
        largest ← item
return largest
```

Dễ hiểu, hay được sử dụng,
không phụ thuộc vào ngôn ngữ lập trình

Độ phức tạp thuật toán

- Hầu hết các thuật toán được thiết kế làm việc với kích thước dữ liệu đầu vào tùy ý
  - Ví dụ thuật toán ở trang trước, kích thước dữ liệu đầu vào là số phần tử trong danh sách (n)

- Độ phức tạp thời gian (time complexity)
  - Xác định lượng thời gian cần thiết để thực hiện giải thuật
    - Được tính là số phép toán cơ bản thực hiện giải thuật
- Độ phức tạp không gian (space complexity)
  - Xác định lượng bộ nhớ cần thiết để thực hiện giải thuật
    - Lượng bộ nhớ lớn nhất cần thiết để lưu các đối tượng của thuật toán tại một thời điểm thực hiện thuật toán

Thường được biểu diễn như là một hàm của kích thước dữ liệu đầu vào

Độ phức tạp thuật toán

- **Quy tắc bỏ hằng số**
  - Nếu đoạn chương trình P có thời gian thực hiện T(n) = O(c1.f(n)) với c1 là một hằng số dương thì có thể coi đoạn chương trình đó có độ phức tạp tính toán là O(f(n)).
- **Quy tắc cộng – lấy max**
  - Nếu T1(n) và T2(n) là thời gian thực hiện của hai đoạn chương trình P1 và P2; và T1(n)=O(f(n)), T2(n)=O(g(n)) thì thời gian thực hiện của đoạn hai chương trình đó nối tiếp nhau là T(n)=O(max(f(n),g(n)))
  - **Ví dụ:**

Lệnh gán x:=15 tốn một hằng thời gian hay O(1), Lệnh đọc dữ liệu readln(x) tốn một hằng thời gian hay O(1).

Vậy thời gian thực hiện cả hai lệnh trên nối tiếp nhau là O(max(1,1))=O(1)

Độ phức tạp thuật toán

- **Quy tắc nhân**
- Nếu T1(n) và T2(n) là thời gian thực hiện của hai đoạn chương trình P1 và P2 và T1(n) = O(f(n)), T2(n) = O(g(n)) thì thời gian thực hiện của đoạn hai đoạn chương trình đó lồng nhau là T(n) = O(f(n).g(n))
- **Quy tắc tổng quát để phân tích một chương trình**
  - Thời gian thực hiện của mỗi lệnh gán, READ, WRITE là O(1).
  - Thời gian thực hiện của một chuỗi tuần tự các lệnh được xác định bằng qui tắc cộng. Như vậy thời gian này là thời gian thi hành một lệnh nào đó lâu nhất trong chuỗi lệnh.
  - Thời gian thực hiện cấu trúc IF là thời gian lớn nhất thực hiện lệnh sau THEN hoặc sau ELSE và thời gian kiểm tra điều kiện.
  - Thời gian kiểm tra điều kiện là O(1).
  - Thời gian thực hiện vòng lặp là tổng (trên tất cả các lần lặp) thời gian thực hiện thân vòng lặp. Nếu thời gian thực hiện thân vòng lặp không đổi thì thời gian thực hiện vòng lặp là tích của số lần lặp với thời gian thực hiện thân vòng lặp.

Độ phức tạp thuật toán

- Ví dụ 1: Xác định độ phức tạp của chương trình sau

```
VAR i,j,n : longint
        s1,s2: longint
Begin
{1}    readln(n);
{2}    s1:=0;
{3}    for i:=1 to n do
{4}    s1:=s1+i
{5}    s2:=0;
{6}    for j:=1 to n do
{7}    s2:=s2+j*j;
{8}    write(s1);
{9}    write(s2);
End.
```

Độ phức tạp thuật toán

- Ví dụ 2: Xác định độ phức tạp của chương trình sau

```
VAR i,j,n: longint;
      c: longint;
Begin
{1} readln(n);
{2} c:=0;
{3} for i:=1 to 2*n do
{4} c:=c+1
{5} for i:=1 to n do
{6}     for j:=1 to n do
{7}     c:=c+1;
{8} writeln(c);
End.
```

Ví dụ về độ phức tạp thuật toán

- **Thuật toán**: Tìm số nguyên lớn nhất trong danh sách n số nguyên

Algorithm LargestNumber
Input: A list of numbers L.
Output: The largest number in the list L.

largest ← null
for each *item* **in** L **do**
    if *item* > *largest* **then**
        largest ← *item*
return *largest*

Độ phức tạp thời gian và không gian đều là O(n)

Bài tập 1

- Tìm độ phức tạp tính toán của thuật toán sắp xếp nổi bọt (bubble sort)

```
function bubble_sort(List L, number n) // n chiều dài L
    for i from n downto 2
        for j from 1 to (i - 1)
            if L[j] > L[j + 1] then //nếu chúng không đúng thứ tự
                swap(L[j], L[j + 1]) //đổi chỗ chúng cho nhau
            end if
        end for
    end for
end function
```

Bài tập 2

- Tìm độ phức tạp tính toán của thuật toán tìm kiếm nhị phân trên danh sách đã sắp xếp

```
function binary_search(A, x, L, R)
    if L > R then
        return Fail
    else
        i ← ⌊(L + R)/2⌋
        if A[i] == x then
            return i
        else if A[i] > x then
            return binary_search(A, x, L, i − 1)
        else
            return binary_search(A, x, i + 1, R)
        end if
    end if
end function
```