HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

----------ᴗ📖ᴗ----------

KHOA CÔNG NGHỆ THÔNG TIN

BÀI GIẢNG

TOÁN RỜI RẠC 1

NGUYỄN DUY PHƯƠNG

Hà Nội 2016

LỜI GIỚI THIỆU

Toán rời rạc là lĩnh vực nghiên cứu và xử lý các đối tượng rời rạc. Toán rời rạc dùng để đếm, quan sát, và xử lý mối quan hệ giữa các đối tượng trong các tập hợp khác nhau. Bản chất tính toán trên máy tính là rời rạc. Chính vì vậy, toán học rời rạc được xem là môn học kinh điển cho sinh viên các ngành Công nghệ thông tin và Điện tử Viễn thông. Tài liệu hướng dẫn môn học toán học rời rạc được xây dựng dựa trên cơ sở kinh nghiệm giảng dạy môn học và kế thừa những nội dung từ giáo trình "Toán học rời rạc ứng dụng trong tin học" của Kenneth Rossen. Tài liệu được trình bày thành hai phần: Lý thuyết tổ hợp (Toán rời rạc 1) và Lý thuyết đồ thị (Toán rời rạc 2).

Phần I trình bày những kiến thức cơ bản về lý thuyết tổ hợp thông qua việc giải quyết bốn bài toán cơ bản đó là: Bài toán đếm, Bài toán tồn tại, Bài toán liệt kê và Bài toán tối ưu. Phần II trình bày những kiến thức cơ bản về Lý thuyết đồ thị: khái niệm, định nghĩa, các thuật toán trên đồ thị, đồ thị Euler, đồ thị Hamilton. Một số bài toán có ứng dụng thực tiễn quan trọng khác của lý thuyết đồ thị cũng được chú trọng giải quyết đó là Bài toán tô màu đồ thị, Bài toán tìm đường đi ngắn nhất và Bài toán luồng cực đại trong mạng.

Trong mỗi phần của tài liệu, chúng tôi cố gắng trình bày ngắn gọn trực tiếp vào bản chất của vấn đề. Các thuật toán được trình bày và cài bằng ngôn ngữ lập trình C++. Mặc dù đã rất cẩn trọng trong quá trình biên soạn, tuy nhiên tài liệu không tránh khỏi những thiếu sót và hạn chế. Chúng tôi rất mong được sự góp ý quí báu của tất cả đọc giả và các bạn đồng nghiệp.

Hà nội, tháng 12 năm 2016

MỤC LỤC

CHƯƠNG 1. LOGIC, TẬP HỢP VÀ ỨNG DỤNG

Nội dung chính của chương này đề cập đến những kiến thức cơ bản về logic mệnh đề, lý thuyết tập hợp và ứng dụng. Nội dung chính của chương bao gồm:

- ✓ Logic mệnh đề và ứng dụng.
- ✓ Logic vị từ và ứng dụng.
- ✓ Lý thuyết tập hợp và ứng dụng.
- ✓ Một số ứng dụng của logic và tập hợp trong tin học.
- ✓ Bài tập chương 1.

Bạn đọc có thể tìm thấy những kiến thức sâu hơn và chi tiết hơn trong các tài liệu [1] và [2] của tài liệu tham khảo.

1.1. Giới thiệu chung

Tổ hợp là một lĩnh vực quan trọng của toán học rời rạc đề cập tới nhiều vấn đề khác nhau của toán học. Lý thuyết Tổ hợp nghiên cứu việc phân bố các phần tử vào các tập hợp. Thông thường các phần tử của tập hợp là hữu hạn và việc phân bố chúng phải thoả mãn những điều kiện nhất định nào đó tuỳ theo yêu cầu của bài toán. Mỗi cách phân bố được coi là một ***"cấu hình của tổ hợp"***. Các cấu hình tổ hợp được xem xét như một lời giải của *bài toán đếm, bài toán liệt kê, bài toán tồn tại hay bài toán tối ưu*.

Bài toán đếm: đây là dạng bài toán nhằm trả lời câu hỏi "có bao nhiêu cấu hình thoả mãn điều kiện đã nêu?". Bài toán đếm được áp dụng có hiệu quả vào những công việc mang tính chất đánh giá như xác suất xảy ra của một sự kiện, thời gian tính toán hay độ phức tạp của một chương trình máy tính.

Bài toán liệt kê: bài toán liệt kê quan tâm đến tất cả các cấu hình có thể có được, vì vậy lời giải của nó được biểu diễn dưới dạng thuật toán **"vét cạn"** tất cả các cấu hình. Bài toán liệt kê thường được làm nền cho nhiều bài toán khác. Hiện nay, một số bài toán tồn tại, bài toán tối ưu, bài toán đếm vẫn chưa có cách nào giải quyết ngoài phương pháp liệt kê. Phương pháp liệt kê càng trở nên quan trọng hơn khi nó được hỗ trợ bởi các hệ thống máy tính.

Bài toán tối ưu: khác với bài toán liệt kê, bài toán tối ưu chỉ quan tâm tới cấu hình ***"tốt nhất"*** theo một nghĩa nào đó. Đây là một bài toán có nhiều ứng dụng thực tiễn được giải quyết bằng lý thuyết tổ hợp.

Bài toán tồn tại: nếu như bài toán đếm thực hiện đếm bao nhiêu cấu hình có thể có, bài toán liệt kê xem xét tất cả các cấu hình có thể có, bài toán tối ưu chỉ ra một cấu hình tốt nhất. Bài toán tồn tại hướng đến giải quyết những vấn đề còn nghi vấn. Điều này có nghĩa là ngay kể cả vấn đề có hay không một cấu hình cũng chưa biết. Những bài toán này thường là những bài toán khó. Do vậy máy tính được xem là công cụ hữu hiệu nhất giải quyết bài toán tồn tại.

1.2. Những kiến thức cơ bản về Logic mệnh đề

Các qui tắc cơ bản của Logic cho ta ý nghĩa chính xác của các mệnh đề. Những qui tắc của logic chính là công cụ cơ sở để chúng ta có thể xây dựng nên các ngôn ngữ lập trình, các bảng mạch máy tính, kiểm chứng tính đúng đắn của chương trình và nhiều ứng dụng quan trọng khác.

1.2.1. Định nghĩa & phép toán

Đối tượng nghiên cứu của logic là các mệnh đề. Một mệnh đề được hiểu là một câu khẳng định hoặc đúng hoặc sai chứ không thể vừa đúng vừa sai.

Ví dụ: Những câu khẳng định sau đây là một mệnh đề:

- *"Hà nội là thủ đô của Việt nam."*
- *1 + 1 = 2*
- *2 + 2 = 3*

Các mệnh đề *" Hà nội là thủ đô của việt nam", " 1 +1 =2 "* là những mệnh đề đúng, mệnh đề *"2 +2 =3"* là sai. Nhưng những câu trong ví dụ sau sẽ không phải là một mệnh đề vì nó những câu đó không cho ta khẳng định đúng cũng chẳng cho ta khẳng định sai.

- *"Bây giờ là mấy giờ ?"*
- *"Hãy suy nghĩ điều này cho kỹ lưỡng"*
- *x +1 =2*
- *x + y = z*

Ta ký hiệu những chữ cái *A, B, C, D, p, q, r, s . . .* là những mệnh đề. Giá trị của một mệnh đề đúng được ký hiệu là *T*, giá trị mệnh đề sai được ký hiệu là *F*. Tập giá trị *T, F* còn được gọi là giá trị chân lý của một mệnh đề.

Định nghĩa 1. Cho *p* là một mệnh đề. Phép phủ định mệnh đề *p* cũng là một mệnh đề (ký hiệu là $\overline{p}$ hoặc ¬p). Mệnh đề $\overline{p}$ có giá trị *F* khi và chỉ khi mệnh đề *p* nhận giá trị *T*, nhận giá trị *F* khi và chỉ khi *p* nhận giá trị *T*.

Định nghĩa 2. Cho *p* và *q* là hai mệnh đề. Phép hội giữa mệnh đề *p* với mệnh đề *q* là một mệnh đề (ký hiệu $p \wedge q$). Mệnh đề $p \wedge q$ có giá trị *T* khi và chỉ khi *p, q* nhận giá trị *T*, có giá trị *F* khi và chỉ khi hoặc *p, q*, hoặc cả hai nhận giá trị *F*.

Định nghĩa 3. Cho *p* và *q* là hai mệnh đề. Phép tuyển giữa mệnh đề *p* với mệnh đề *q* là một mệnh đề (ký hiệu $p \vee p$). Mệnh đề $p \vee p$ có giá trị *T* khi và chỉ khi ít nhất một trong hai mệnh đề *p, q* nhận giá trị *T*, có giá trị *F* khi và chỉ khi cả *p, q* đều nhận giá trị *F*.

Định nghĩa 4. Cho p và q là hai mệnh đề. Phép tuyển loại giữa mệnh p với mệnh đề q (được ký hiệu là $p \oplus q$) là một mệnh đề. Mệnh đề $p \oplus q$ chỉ đúng khi một trong p hoặc q đúng và sai trong các trường hợp khác còn lại.

Định nghĩa 5. Cho p và q là hai mệnh đề. Phép kéo theo giữa mệnh đề p với mệnh đề q (ký hiệu $p \rightarrow q$) là một mệnh đề. Mệnh đề $p \rightarrow q$ nhận giá T khi và chỉ khi p và q nhận giá trị F hoặc p và q cùng nhận giá trị T. Mệnh đề $p \rightarrow q$ nhận giá trị F khi và chỉ khi p nhận giá trị T và q nhận giá trị F.

Định nghĩa 6. Cho p và q là hai mệnh đề. Phép tương đương giữa mệnh đề p với mệnh đề q là một mệnh đề (ký hiệu $p \leftrightarrow q$). Mệnh đề $p \leftrightarrow q$ có giá trị đúng khi p và q có cùng giá trị chân lý và sai trong các trường hợp khác còn lại.

Các phép toán : $\vee$, $\wedge$, $\bar{}$, $\oplus$, $\rightarrow$, $\leftrightarrow$ có thể được định nghĩa thông qua bảng giá trị chân lý sau:

Bảng 1.1: Bảng giá trị chân lý của các phép toán $\vee, \wedge, \neg, \oplus, \rightarrow, \leftrightarrow$

p	q	$\overline{p}$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
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

1.2.2. Sự tương đương giữa các mệnh đề

Một vấn đề hết sức quan trọng trong lập luận toán học là việc thay thế một mệnh đề bằng một mệnh đề khác có cùng giá trị chân lý. Hai mệnh đề có cùng một giá trị chân lý chúng ta có thể hiểu theo cách thông thường là chúng tương đương nhau về ngữ nghĩa. Do vậy, ta sẽ tiếp cận và phân loại các mệnh đề phức hợp thông qua các giá trị chân lý của chúng.

Định nghĩa 7. Một mệnh đề phức hợp luôn luôn đúng với bất kể các giá trị chân lý của các mệnh đề thành phần được gọi là hằng đúng (tautology). Một mệnh đề luôn luôn sai với mọi giá trị chân lý của các mệnh đề thành phần được gọi là mâu thuẫn.

Ví dụ: mệnh đề phức hợp $p \vee \overline{p}$ là hằng đúng, $p \wedge \overline{p}$ là mâu thuẫn vì giá trị chân lý của các mệnh đề trên luôn luôn đúng, hoặc luôn luôn sai như được chỉ ra trong bảng 1.2.

Bảng 1.2. Ví dụ về mệnh đề hằng đúng & mệnh đề mâu thuẫn			
p	$\overline{p}$	p $\vee \overline{p}$	p$\wedge \overline{p}$

T	F	T	F
F	T	T	F

Định nghĩa 8. Hai mệnh đề *p, q* được gọi là tương đương logic với nhau (ký hiệu : $p \Leftrightarrow q$, hoặc $p \equiv q$, hoặc $p=q$) khi và chỉ khi các cột cho giá trị chân lý của chúng giống nhau. Hay mệnh đề $p \leftrightarrow q$ là hằng đúng.

Ví dụ 1. Hai mệnh đề $\overline{(p \vee q)}$và $\overline{p} \wedge \overline{q}$ là tương đương logic vì các cột giá trị chân lý của chúng được thể hiện qua bảng sau:

Bảng 1.3. Bảng giá trị chân lý đối với $\overline{(p \vee q)}$ và $\overline{p} \wedge \overline{q}$						
p	q	$p \wedge q$	$\overline{(p \vee q)}$	$\overline{p}$	$\overline{q}$	$\overline{p} \wedge \overline{q}$
T	T	T	**F**	F	F	**F**
T	F	T	**F**	F	T	**F**
F	T	T	**F**	T	F	**F**
F	F	F	**T**	T	T	**T**

Dùng bảng giá trị chân lý để chứng minh tính tương đương logic giữa hai mệnh đề phức hợp cho ta một phương pháp trực quan dễ hiểu. Tuy nhiên, với những mệnh đề logic phức hợp có *k* mệnh đề thành phần thì cần tới 2^k tổ hợp các bộ giá trị chân lý khác nhau. Do đó, dùng bảng chân lý để chứng minh tính tương đương logic giữa hai mệnh đề phức hợp gặp nhiều khó khăn. Trong trường hợp này ta có thể chứng minh tính tương logic bằng việc thay thế một mệnh đề phức hợp bằng những tương đương logic có trước.

Bằng phương pháp bảng chân lý, dễ dàng chứng minh được sự tương đương của các công thức dưới đây:

$$p \rightarrow q \Leftrightarrow \overline{p} \vee q$$

$$p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$$

$$\overline{\overline{p}} \Leftrightarrow p$$

Bảng 1.4. Bảng các tương đương logic	
TƯƠNG ĐƯƠNG	**TÊN GỌI**
$p \wedge T \Leftrightarrow p$ $p \wedge F \Leftrightarrow p$	Luật đồng nhất
$p \vee T \Leftrightarrow T$	Luật nuốt

$p \wedge F \Leftrightarrow F$	
$p \vee p \Leftrightarrow p$ $p \wedge p \Leftrightarrow p$	Luật luỹ đẳng
$\overline{\overline{p}} \Leftrightarrow p$	Luật phủ định kép
$p \vee q \Leftrightarrow q \vee p$ $p \wedge q \Leftrightarrow q \wedge p$	Luật giao hoán
$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$ $(p \wedge q) \wedge r \Leftrightarrow p \wedge(q \wedge r)$	Luật kết hợp
$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	Luật phân phối
$\overline{(p \wedge q)} \Leftrightarrow \overline{p} \vee \overline{q}$ $\overline{(p \vee q)} \Leftrightarrow \overline{p} \wedge \overline{q}$	Luật De Morgan

Ví dụ: Chứng minh $\overline{p \vee (\overline{p} \vee q)} \Leftrightarrow \overline{p} \wedge \overline{q}$?

Chứng minh:

$$
\begin{aligned}
\overline{p \vee (\overline{p} \wedge q)} &\Leftrightarrow \overline{p} \wedge \overline{(\overline{p} \wedge q)} && \\
&\Leftrightarrow \overline{p} \wedge \left[\left(\overline{\overline{p}}\right) \vee \overline{q}\right] && \text{theo luật De Morgan thứ 2} \\
&\Leftrightarrow \overline{p} \wedge \left[p \vee \overline{q}\right] && \text{theo luật De Morgan thứ 2} \\
&\Leftrightarrow (\overline{p} \wedge p) \vee (\overline{p} \wedge \overline{q}) && \text{theo luật phủ định kép} \\
&\Leftrightarrow F \vee (\overline{p} \wedge \overline{q}) && \text{theo luật phân phối} \\
&\Leftrightarrow (\overline{p} \wedge \overline{q}) \vee F && \text{tương đương tiện ích} \\
&\Leftrightarrow (\overline{p} \wedge \overline{q}) && \text{Điều cần chứng minh.}
\end{aligned}
$$

1.2.3. Dạng chuẩn tắc

Các công thức (mệnh đề) tương đương được xem như các biểu diễn khác nhau của cùng một mệnh đề. Để dễ dàng viết các chương trình máy tính thao tác trên các công thức, chúng ta cần chuẩn hóa các công thức, đưa chúng về dạng biểu diễn chuẩn được gọi là ***dạng chuẩn hội***. Một công thức được gọi là ở dạng chuẩn hội nếu nó là hội của các mệnh đề tuyển. Phương pháp để biến đổi một công thức bất kỳ về dạng chuẩn hội bằng cách áp dụng các thủ tục sau:

- ✓ Bỏ các phép kéo theo (→) bằng cách thay (p→q) bởi $\overline{p} \vee q$.
- ✓ Chuyển các phép phủ định ($\overline{\ }$) vào sát các ký hiệu mệnh đề bằng cách áp dụng luật De Morgan và thay $\overline{\overline{p}}$ bởi p.
- ✓ Áp dụng luật phân phối thay các công thức có dạng (p∨(q∧r)) bởi (p∨q)∧(p∨r).

Ví dụ. Ta chuẩn hóa công thức $(p \to q) \vee \overline{(r \vee \overline{s})}$.

Lời giải.

$$\begin{aligned}(p \to q) \vee \overline{(r \vee \overline{s})} &\Leftrightarrow (\overline{p} \vee q) \vee \overline{(r \vee \overline{s})} \\ &\Leftrightarrow (\overline{p} \vee q) \vee (\overline{r} \wedge \overline{\overline{s}}) \\ &\Leftrightarrow (\overline{p} \vee q) \vee (\overline{r} \wedge s) \\ &\Leftrightarrow (\overline{p} \vee q \vee \overline{r}) \wedge (\overline{p} \vee q \vee s)\end{aligned}$$

Như vậy công thức $(p \to q) \vee \overline{(r \vee \overline{s})}$ được đưa về dạng chuẩn hội là $(\overline{p} \vee q \vee \overline{r}) \wedge (\overline{p} \vee q \vee s)$.

1.3. Vị từ và lượng từ

Trong toán học hay trong các chương trình máy tính chúng ta rất hay gặp những khẳng định chưa phải là một mệnh đề. Những khẳng định đó đều có liên quan đến các biến. Chẳng hạn khẳng đinh:

P(x) = "x > 3" không phải là một mệnh đề nhưng tại những giá trị cụ thể của $x=x_0$ nào đó thì $P(x_0)$ lại là một mệnh đề. Hoặc trong những đoạn chương trình gặp câu lệnh:

if (x > 3) then x:= x +1;

thì chương trình sẽ đặt giá trị cụ thể của biến *x* vào *P(x)*, nếu mệnh đề *P(x)* cho giá trị đúng *x* sẽ được tăng lên 1 bởi câu lệnh x*:=x+1, P(x)* có giá trị sai giá trị của *x* được giữ nguyên sau khi thực hiện câu lệnh *if*.

Chúng ta có thể phân tích mỗi khẳng định thành hai phần chủ ngữ và vị ngữ (hay vị từ), trong câu " *x lớn hơn 3*" thì *x* là chủ ngữ, " *lớn hơn 3*" là vị ngữ. Hàm *P(x)* được gọi là hàm mệnh đề. Một hàm mệnh đề có thể có một hoặc nhiều biến. Giá trị chân lý của hàm mệnh đề tại những giá trị cụ thể của biến được xác định như những mệnh đề thông thường.

Ví dụ. Cho *Q(x, y, z)* là hàm mệnh đề xác định câu $x^2 = y^2 + z^2$ hãy xác định giá trị chân lý của các mệnh đề *Q (3, 2, 1), Q (5, 4, 3)*?

Lời giải.

Đặt giá trị cụ thể của *x* , *y* , *z* vào *Q(x,y,z)* ta có :

Q(3,2,1) là mệnh đề "$3^2 = 2^2 + 1^2$" là sai do đó *Q(3,2,1)* là mệnh đề sai. Trong đó, *Q (5, 4, 3)* là mệnh đề " $5^2 = 4^2 + 3^2$" là mệnh đề đúng.

Tổng quát, giả sử *M* là một tập hợp các phần tử nào đó. *M* thường được gọi là trường hay miền xác định của các phần tử thuộc *M*. Khi đó, biểu thức *P(x)* gọi là vị từ xác định trên trường *M* nếu khi thay *x* bởi một phần tử bất kỳ của trường *M* thì *P(x)* sẽ trở thành một mệnh đề trên trường *M*.

Khi tất cả các biến của hàm mệnh đề đều được gán những giá trị cụ thể, thì mệnh đề tạo ra sẽ xác định giá trị chân lý. Tuy nhiên, có một phương pháp quan trọng khác để biến một hàm mệnh đề thành một mệnh đề mà không cần phải kiểm chứng mọi giá trị chân lý của hàm mệnh đề tương ứng với các giá trị của biến thuộc trường đang xét. Phương pháp đó gọi là sự lượng hoá hay lượng từ. Chúng ta xét hai lượng từ quan trọng là lượng từ *với mọi* (ký hiệu :$\forall$), lượng từ *tồn tại* (ký hiệu :$\exists$).

Định nghĩa 1. Lượng từ với mọi của *P(x)* ký hiệu là $\forall x\ P(x)$ là một mệnh đề " *P(x) đúng với mọi phần tử x thuộc trường đang xét*".

Ví dụ. Cho hàm mệnh đề $P(x) = x^2 + x + 41$ là nguyên tố. Xác định giá trị chân lý của mệnh đề $\forall P(x)$ với *x* thuộc không gian bao gồm các số tự nhiên [0..39].

Lời giải. Vì P(x) đúng với mọi giá trị của x $\in$ [0..39] $\Rightarrow \forall$ P(x) là đúng.

Ví dụ : Cho P(x) là hàm mệnh đề " x + 1 > x". Xác định giá trị chân lý của mệnh đề $\forall$ x P(x), trong không gian các số thực.

Lời giải : Vì P(x) đúng với mọi số thực x nên $\forall$x P(x) là đúng.

Định nghĩa 2. Lượng từ tồn tại của hàm mệnh đề P(x) (được ký hiệu là:$\exists$ x P(x)) là một mệnh đề " *Tồn tại một phần tử x trong không gian sao cho P(x) là đúng* ".

Ví dụ: Cho P(x) là hàm mệnh đề "x > 3". Hãy tìm giá trị chân lý của mệnh đề $\exists$ x P(x) trong không gian các số thực.

Lời giải: Vì P(4) là " 4 > 3" đúng nên $\exists$ x P(x) là đúng.

Ví dụ: Cho Q(x) là " x + 1 > x". Hãy tìm giá trị chân lý của mệnh đề $\exists$ x Q(x) trong không gian các số thực.

Lời giải: vì Q(x) sai với mọi x $\in$ R nên mệnh đề $\exists$ x Q(x) là sai.

Bảng 1.5: Giá trị chân lý của lượng từ $\forall$, $\exists$		
$\forall$x P(x)	P(x) đúng với mọi x	Có một giá trị của x để P(x) sai
$\exists$x P(x)	Có một giá trị của x để P(x) đúng	P(x) sai với mọi x

Dịch những câu thông thường thành biểu thức logic: Dịch một câu được phát biểu bằng ngôn ngữ tự nhiên (câu hỏi thông thường) thành một biểu thức logic có vai trò hết sức quan trọng trong xây dựng các ngôn ngữ lập trình, chương trình dịch và xử lý ngôn ngữ tự nhiên. Quá trình dịch một câu từ ngôn ngữ tự nhiên thành một biểu thức sẽ làm mất đi tính tự nhiên của ngôn ngữ vì đa số các ngôn ngữ đều không rõ ràng, nhưng một biểu thức logic lại rất rõ ràng chặt chẽ từ cú pháp thể hiện đến ngữ nghĩa của câu. Điều này dẫn đến phải có một tập hợp các giả thiết hợp lý dựa trên một hàm xác định ngữ nghĩa cuả câu đó. Một khi câu đã được chuyển dịch thành biểu thức logic, chúng ta có thể xác định được giá trị chân lý của biểu thức logic, thao tác trên biểu thức logic, biến đổi tương đương trên biểu thức logic. Chúng ta sẽ minh hoạ việc dịch một câu thông thường thành biểu thức logic thông qua những sau.

Ví dụ dịch câu "*Bạn không được lái xe máy nếu bạn cao dưới 1.5 mét trừ phi bạn trên 18 tuổi*" thành biểu thức logic.

Lời giải.

Ta gọi p là câu : Bạn được lái xe máy.

q là câu : Bạn cao dưới 1.5m.

r là câu : Bạn trên 18 tuổi.

Khi đó: Câu hỏi trên được dịch là: $(q \wedge \overline{r}) \rightarrow \overline{p}$

Ví dụ: Dịch câu " *Tất cả các sinh viên học tin học đều học môn toán học rời rạc*"

Lời giải: Gọi P(x) là câu "x cần học môn toán học rời rạc" và x được xác định trong không gian của các sinh viên học tin học. Khi đó chúng ta có thể phát biểu: $\forall x\, P(x)$.

Ví dụ: Dịch câu "*Có một sinh viên ở lớp này ít nhất đã ở tất cả các phòng của ít nhất một nhà trong ký túc xá*".

Lời giải : Gọi tập sinh viên trong lớp là không gian xác định sinh viên x, tập các nhà trong ký túc xá là không gian xác định căn nhà y, tập các phòng là không gian xác định phòng z. Ta gọi P(z,y) là " *z thuộc y*", Q(x,z) là " *x đã ở z*". Khi đó ta có thể phát biểu :

$\exists x \exists y \forall z \; (P(z,y) \rightarrow Q(x,z))$;

1.4. Một số ứng dụng trên máy tính

Các phép toán bít: Các hệ thống máy tính thường dùng các bit (binary digit) để biểu diễn thông tin. Một bít có hai giá trị chân lý hoặc 0 hoặc 1. Vì giá trị chân lý của một biểu thức logic cũng có hai giá trị hoặc đúng (T) hoặc sai (F). Nếu ta coi giá trị đúng có giá trị 1 và giá trị sai là 0 thì các phép toán với các bít trong máy tính được tương ứng với các liên từ logic.

Một xâu bít (hoặc xâu nhị phân) là dãy không hoặc nhiều bít. Chiều dài của xâu là số các bít trong xâu đó. Ví dụ xâu nhị 101010011 có độ dài là 9. Một số nguyên được biểu diễn như một xâu nhị phân có độ dài 16 bít.

Các phép toán với bít được xây dựng trên các xâu bít có cùng độ dài, bao gồm : AND bít (phép và cấp bít), OR (phép hoặc cấp bít), XOR (phép tuyển loại trừ cấp bít). Ví dụ: cho hai xâu bít 01101 10110 và 11000 11101 hãy tìm xâu AND bít, OR bít, XOR bít.

Bảng 1.5. 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 =1100	B = 8=1000	1000	1101	0101

Thuật toán các phép tính số nguyên: Các thuật toán thực hiện các phép tính với các số nguyên khi dùng khai triển nhị phân là hết sức quan trọng trong bộ xử lý số học của máy tính. Như chúng ta đã biết, thực chất các số nguyên được biểu diễn trong máy tính là các xâu bít nhị phân, do vậy chúng ta có thể sử dụng biểu diễn nhị phân của các số để thực hiện các phép tính.

Giả sử khai triển nhị phân của các số nguyên a và b tương ứng là:

$a = (a_{n-1}a_{n-2} \ldots a_1a_0)_2$, $b = (b_{n-1}b_{n-2} \ldots b_1b_0)_2$. Khai triển của a và b có đúng n bít (chấp nhận những bít 0 ở đầu để làm đặc n bít).

Xét bài toán cộng hai số nguyên viết ở dạng nhị phân. Thủ tục thực hiện việc cộng cũng giống như làm trên giấy thông thường. Phương pháp này tiến hành bằng cách cộng các bít nhị phân tương ứng có nhớ để tính tổng hai số nguyên. Sau đây là mô tả chi tiết cho quá trình cộng hai xâu bít nhị phân.

Để cộng a với b, trước hết ta cộng hai bít phải nhất, nghĩa là:

$a_0 + b_0 = c_0*2 + s_0$; trong đó s_0 là bít phải nhất của số nguyên tổng a + b, c_0 là số cần để nhớ nó có thể bằng 0 hoặc 1. Sau đó ta cộng hai bít tiếp theo và số nhớ:

$a_1 + b_1 + c_0 = c_1*2 + s_1$; s_1 là bít tiếp theo của số a + b, c_1 là số nhớ. Tiếp tục quá trình này bằng cách cộng các bít tương ứng trong khai triển nhị phân và số nhớ, ở giai đoạn cuối cùng : $a_{n-1} + b_{n-1} + c_{n-2} = c_{n-1} * 2 + s_{n-1}$. Bít cuối cùng của tổng là c_{n-1}. Khi đó khai triển nhị phân của tổng a + b là $(s_na_{n-1} \ldots s_1s_0)_2$.

Ví dụ: cộng $a =(1110)_2$, $b = (1011)_2$

Lời giải:

Trước hết lấy:

$$a_0 + b_0 = 0 + 1 = 0 * 2 + 1 \Rightarrow c_0=0, s_0 = 1$$

Tiếp tục:

$$a_1 + b_1 + c_0 = 1 + 1 + 0 = 1 * 2 + 0 \Rightarrow c_1=1, s_1 = 0$$

$$a_2 + b_2 + c_1 = 1 + 0 + 1 = 1 * 2 + 0 \Rightarrow c_2=1, s_2 = 0$$

$$a_3 + b_3 + c_2 = 1 + 1 + 1 = 1 * 2 + 1 \Rightarrow c_3=1, s_3 = 1$$

Cuối cùng:

$s_4 = c_3 = 1 \Rightarrow a + b = (11001)_2$

Thuật toán cộng:

```
void  Cong(a , b: positive integer)
{
        /*a = (a_{n-1}a_{n-2} . . .a_1a_0)_2 , b = (b_{n-1}b_{n-2} . . .b_1b_0)_2 */
        c=0;
        for (j=0 ; j≤n-1; j++) {
                d= [( a_j + b_j + c)/ 2];
                s_j = a_j + b_j + c – 2d;
                c = d;
        }
        s_n = c;
         khai triển nhị phân của tổng là (s_na_{n-1} . . .s_1s_0)_2;
}
```

Thuật toán nhân: Để nhân hai số nguyên n bít a, b ta bắt đầu từ việc phân tích:

$a = (a_{n-1}a_{n-2}. . .a_1a_0)$, $b = (b_{n-1}b_{n-2}. . .b_1b_0)$

Ta có thể tính a.b từ phương trình trên. Trước hết, ta nhận thấy $ab_j = a$ nếu $b_j=1$, $ab_j=0$ nếu $b_j=0$. Mỗi lần tính ta nhân với 2^j hay dịch chuyển sang trái j bít 0 bằng cách thêm j bít 0 vào bên trái kết quả nhận được. Cuối cùng, cộng n số nguyên $ab_j 2^j$ (j=0..n-1) ta nhận được a.b. Ví dụ sau đây sẽ minh hoạ cho thuật toán nhân:

Ví dụ: Tìm tích của $a = (110)_2$, $b= (101)_2$

Giải: Ta nhận thấy

$ab_0 2^0 = (110)_2*1*2^0 = (110)_2$

$ab_1 2^1 = (110)_2*0*2^1 = (0000)_2$

$ab_2 2^2 = (110)_2*1*2^2 = (11000)_2$

Sử dụng thuật toán tính tổng hai số nguyên a, b có biểu diễn n bít ta nhận được(ta có thể thêm số 0 vào đầu mỗi toán hạng):

$(0\,110)_2 + (0000)_2 = (0110)_2$;

$(0\,0110)_2 + (11000)_2 = (11110)_2 = ab.$

Thuật toán nhân hai số nguyên n bít có thể được mô phỏng như sau:

```
void Nhan( a, b : Positive integer){
        /* khai triển nhị phân tương ứng của a = (a_{n-1}a_{n-2}. . .a_1a_0),
                b = (b_{n-1}b_{n-2}. . .b_1b_0) */
        for (j=0; j≤n-1; j++) {
                if ( ( b_j==1)
                        c_j = a * 2^j; /* a được dịch trái j bít 0 */
                else    c_j =0;
        }
```

/c_0, c_1.., c_{n-1} là những tích riêng của $ab_j\ 2^j$(j=0..n-1 */*

p=0;

for (j=0 ; j≤ n-1; j++)

p= p + c_j;

/ p là giá trị của tích ab */*

}

1.5. Những kiến thức cơ bản về lý thuyết tập hợp

1.5.1. Khái niệm & định nghĩa

Các tập hợp dùng để nhóm các đối tượng lại với nhau. Thông thường, các đối tượng trong tập hợp có các tính chất tương tự nhau. Ví dụ, tất cả sinh viên mới nhập trường tạo nên một tập hợp, tất cả sinh viên thuộc khoa Công nghệ thông tin là một tập hợp, các số tự nhiên, các số thực cũng tạo nên các tập hợp. Chú ý rằng, thuật ngữ đối tượng được dùng ở đây không chỉ rõ cụ thể một đối tượng nào, sự mô tả một tập hợp nào đó hoàn toàn mang tính trực giác về các đối tượng.

Định nghĩa 1. Tập các đối tượng trong một tập hợp được gọi là các phần tử của tập hợp. Các tập hợp thường được ký hiệu bởi những chữ cái in hoa đậm như A, B, X, Y . . ., các phần tử thuộc tập hợp hay được ký hiệu bởi các chữ cái in thường như a, b, c, u, v . . . Để chỉ a là phần tử của tập hợp A ta viết $a \in A$, trái lại nếu a không thuộc A ta viết $a \notin A$.

Tập hợp không chứa bất kỳ một phần tử nào được gọi là tập rỗng (ký hiệu là ϕ hoặc { }). Tập hợp A được gọi là bằng tập hợp B khi và chỉ khi chúng có cùng chung các phần tử và được ký hiệu là A=B. Ví dụ tập A={ 1, 3, 5 } sẽ bằng tập B = { 3, 5, 1 }.

Định nghĩa 2. Tập A được gọi là một tập con của tập hợp B và ký hiệu là $A \subseteq B$ khi và chỉ khi mỗi phần tử của A là một phần tử của B. Hay $A \subseteq B$ khi và chỉ khi lượng từ

$\forall x (x \in A \rightarrow x \in B)$ cho ta giá trị đúng.

Từ định nghĩa trên chúng ta rút ra một số hệ quả sau:

- Tập rỗng ϕ là tập con của mọi tập hợp.
- Mọi tập hợp là tập con của chính nó.
- Nếu $A \subseteq B$ và $B \subseteq A$ thì A=B hay mệnh đề :

  $x (x \in A \rightarrow x \in B) \vee \forall x (x \in B \rightarrow x \in A)$ cho ta giá trị đúng.
- Nếu $A \subseteq B$ và $A \neq B$ thì ta nói A là tập con thực sự của B và ký hiệu là $A \subset B$.

Định nghĩa 3. Cho S là một tập hợp. Nếu S có chính xác n phần tử phân biệt trong S, với n là số nguyên không âm thì ta nói S là một tập hữu hạn và n được gọi là bản số của S. Bản số của S được ký hiệu là |S | hay N(S).

Định nghĩa 4. Cho tập hợp S. Tập luỹ thừa của S ký hiệu là P(S) là tập tất cả các tập con của S.

Ví dụ S = { 0, 1, 2 } $\Rightarrow$ P(S) ={ ϕ, {0}, {1}, {2}, {0,1}, {0, 2}, {1, 2} {0, 1, 2}}.

Định nghĩa 5. Dãy sắp thứ tự (a_1, a_2,.., a_n) là một tập hợp sắp thứ tự có a_1 là phần tử thứ nhất, a_2 là phần tử thứ 2, .., a_n là phần tử thứ n. Chúng ta nói hai dãy sắp thứ tự là bằng nhau khi và chỉ khi các phần tử tương ứng của chúng là bằng nhau. Nói cách khác (a_1, a_2,.., a_n) bằng (b_1, b_2,.., b_n) khi và chỉ khi $a_i = b_i$ với mọi i =1, 2, ..n.

Định nghĩa 6. Cho A và B là hai tập hợp. Tích đề các của A và B được ký hiệu là A$\times$B, là tập hợp của tất cả các cặp (a,b) với a$\in$A, b $\in$B. Hay có thể biểu diễn bằng biểu thức:

$$A \times B = \{ (a, b) \mid a\in A, b \in B \}$$

Định nghĩa 7. Tích đề các của các tập A_1, A_2, . ., A_n được ký hiệu là $A_1\times A_2\times..\times A_n$ là tập hợp của dãy sắp thứ tự (a_1, a_2,.., a_n) trong đó $a_i\in A_i$ với i = 1, 2,..n. Nói cách khác:

$$A_1\times A_2\times..\times A_n = \{ (a_1, a_2,.., a_n) \mid a_i\in A_i \text{ với } i = 1, 2,..n \}$$

1.5.2. Các phép toán trên tập hợp

Các tập hợp có thể được tổ hợp với nhau theo nhiều cách khác nhau thông qua các phép toán trên tập hợp. Các phép toán trên tập hợp bao gồm: Phép hợp (Union), phép giao (Intersection), phép trừ (Minus).

Định nghĩa 1. Cho A và B là hai tập hợp. Hợp của A và B được ký hiệu là A$\cup$B, là tập chứa tất cả các phần tử hoặc thuộc tập hợp A hoặc thuộc tập hợp B. Nói cách khác:

$$A\cup B = \{ x \mid x \in A \vee x\in B \}$$

Định nghĩa 2. Cho A và B là hai tập hợp. Giao của A và B được ký hiệu là A$\cap$B, là tập chứa tất cả các phần tử thuộc A và thuộc B. Nói cách khác:

$$A\cup B = \{ x \mid x \in A \wedge x\in B \}$$

Định nghĩa 3. Hai tập hợp A và B được gọi là rời nhau nếu giao của chúng là tập rỗng (A$\cap$B = ϕ).

Định nghĩa 4. Cho A và B là hai tập hợp. Hiệu của A và B là tập hợp đuợc ký hiệu là A-B, có các phần tử thuộc tập hợp A nhưng không thuộc tập hợp B. Hiệu của A và B còn được gọi là phần bù của B đối với A. Nói cách khác:

$$A – B = \{ x \mid x\in A \wedge x \notin B \}$$

Định nghĩa 5. Cho tập hợp A. Ta gọi $\overline{A}$ là phần bù của A là một tập hợp bao gồm những phần tử không thuộc A.

$$\overline{A} = \{x \mid x \notin A\}$$

Định nghĩa 6. Cho các tập hợp $A_1, A_2, .., A_n$. Hợp của các tập hợp là tập hợp chứa tất cả các phần tử thuộc ít nhất một trong số các tập hợp A_i (i=1, 2, . ., n). Ký hiệu:

$$\bigcup_{i=1}^{n} A_i = A_1 \cup A_2 \cup \cdots \cup A_n$$

Định nghĩa 7: Cho các tập hợp $A_1, A_2, .., A_n$. Giao của các tập hợp là tập hợp chứa các phần tử thuộc tất cả n tập hợp A_i (i=1, 2, . ., n).

$$\bigcup_{i=1}^{n} A_i = A_1 \cap A_2 \cap .. A_n$$

1.5.3. Các hằng đẳng thức trên tập hợp

Mỗi tập con của tập hợp tương ứng với một tính chất xác định trên tập hợp đã cho được gọi là mệnh đề. Với tương ứng này, các phép toán trên tập hợp được chuyển sang các phép toán của logic mệnh đề:

- Phủ định của A, ký hiệu $\overline{A}$ (hay NOT A) tương ứng với phần bù $\overline{A}$.
- Tuyển của A và B, ký hiệu $A \vee B$ (hay A or B) tương ứng với $A \cup B$.
- Hội của A và B, ký hiệu $A \wedge B$ (hay A and B) tương ứng với $A \cap B$.

Các mệnh đề cùng với các phép toán trên nó lập thành một đại số mệnh đề (hay đại số logic). Như thế, đại số tập hợp và đại số logic là hai đại số đẳng cấu với nhau (những mệnh đề phát biểu trên đại số logic tương đương với mệnh đề phát biểu trên đại số tập hợp). Với những trường hợp cụ thể, tuỳ theo tình huống, một bài toán có thể được phát biểu bằng ngôn ngữ của đại số logic hay ngôn ngữ của đại số tập hợp. Bảng 1.6 thể hiện một số hằng đẳng thức của đại số tập hợp.

Bảng 1.6. Một số hằng đẳng thức trên tập hợp

HẰNG ĐẲNG THỨC	TÊN GỌI
$A \cup \phi = A$ $A \cap U = A$ (U là tập vũ trụ)	Luật đồng nhất
$A \cup U = U$ $A \cap \phi = A$	Luật nuốt
$A \cap A = A$ $A \cup A = A$	Luật luỹ đẳng
$\overline{\overline{A}} = A$	Luật bù

$A \cap B = B \cap A$ $A \cup B = B \cup A$	Luật giao hoán
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Luật kết hợp
$A \cup (B \cap C) = (A \cup B) \cup (A \cap C)$ $A \cap (B \cup C) = (A \cup B) \cap (A \cup C)$	Luật phân phối
$\overline{A \cup B} = \overline{A} \cap \overline{B}$ $\overline{A \cap B} = \overline{A} \cup \overline{B}$	Luật De Morgan

1.6. Biểu diễn tập hợp trên máy tính

Có nhiều cách khác nhau để biểu diễn tập hợp trên máy tính, phương pháp phổ biến là lưu trữ các phần tử của tập hợp không sắp thứ tự. Với việc lưu trữ bằng phương pháp này, ngoài những lãng phí bộ nhớ không cần thiết, thì quá trình tính hợp, giao, hiệu các tập hợp gặp nhiều khó khăn và mất nhiều thời gian vì mỗi phép tính đòi hỏi nhiều thao tác tìm kiếm trên các phần tử. Một phương pháp lưu trữ các phần tử bằng cách biểu diễn có thứ tự của các phần tử của một tập vũ trụ tỏ ra hiệu quả hơn rất nhiều trong quá trình tính toán.

Giả sử tập vũ trụ U là hữu hạn gồm n phần tử(hữu hạn được hiểu theo nghĩa các phần tử của U lưu trữ được trong bộ nhớ máy tính). Giả sử ta muốn biểu diễn tập hợp $A \subseteq U$. Trước hết ta chọn một thứ tự tuỳ ý nào đó đối với các phần tử của tập vũ trụ U, giả sử ta được bộ có thứ tự $a_1, a_2, \ldots, a_n$. Sau đó xây dựng một xâu bít nhị phân có độ dài n, sao cho nếu bít thứ i có giá trị 1 thì phần tử $a_i \in A$, nếu $a_i = 0$ thì $a_i \notin A$ (i=1,2..,n). Ví dụ sau sẽ minh hoạ kỹ thuật biểu diễn tập hợp bằng xâu bít nhị phân.

Ví dụ. Giả sử U = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }. Hãy biểu diễn tập hợp $A \subseteq U$ là

1- Tập các số nguyên lẻ $A \subseteq U$.

2- Tập các số nguyên chẵn $B \subseteq U$.

3- Tập các số nguyên nhỏ hơn 5 $C \subseteq U$.

4- Tìm $A \cup B$

5- Tìm $A \cap C$ . . .

Lời giải. Trước hết ta coi thứ tự các phần tử được sắp xếp theo thứ tự tăng dần tức $a_i = i$ (i=1,2,..,10). Khi đó :

1. Xâu bít biểu diễn các số lẻ trong U ({1, 3, 5, 7, 9 }) là xâu có độ dài n = 10 trong đó các bít ở vị trí thứ 1, 3, 5, 7, 9 có giá trị là 1, các bít còn lại có giá trị là 0. Từ đó ta có xâu bít biểu diễn tập hợp A là: 1 0 1 0 1 0 1 0 1 0.

2. Xâu bít biểu diễn các số chẵn trong U ({2, 4, 6, 8, 10 }) là xâu có độ dài n = 10 trong đó các bít ở vị trí thứ 2, 4, 6, 8, 10 có giá trị là 1, các bít còn lại có giá trị là 0. Từ đó ta có xâu bít biểu diễn tập hợp B là: 0 1 0 1 0 1 0 1 0 1.

3. Xâu bít biểu diễn các số nhỏ hơn 5 trong U ({1, 2, 3, 4 }) là xâu có độ dài n = 10 trong đó các bít ở vị trí thứ 1, 2, 3, 4 có giá trị là 1, các bít còn lại có giá trị là 0. Từ đó ta có xâu bít biểu diễn tập hợp C là: 1 1 1 1 0 0 0 0 0 0.

4. Xâu bít biểu diễn tập hợp $A \cup B$ là : (1 0 1 0 1 0 1 0 1 0 $\vee$ 0 1 0 1 0 1 0 1 0 1) là xâu 1 1 1 1 1 1 1 1 1 1. Như vậy, $A \cup B = U$.

5. Tương tự như vậy với $A \cap C \Leftrightarrow$ (1 0 1 0 1 0 1 0 1 0 $\wedge$ 1 1 1 1 0 0 0 0 0 0) là xâu 1 0 1 0 0 0 0 0 0 0. Như vậy $A \cap C = \{ 1, 3 \}$

1.7. Những nội dung cần ghi nhớ

Cần hiểu và nắm vững được những nội dung sau:

- ✓ Các phép toán hội, tuyển, tuyển loại, suy ra, kéo theo của logic mệnh đề.
- ✓ Các phương pháp chứng minh định lý dùng bảng chân lý và các tương đương locgic.
- ✓ Phương pháp biểu diễn các câu hỏi thông thường bằng logic vị từ.
- ✓ Định nghĩa và các phép toán trên tập hợp.
- ✓ Phương pháp biểu diễn tập hợp trên máy tính

BÀI TẬP CHƯƠNG 1

1. Lập bảng giá trị chân lý cho các mệnh đề phức hợp sau:

   a) $(p \to q) \leftrightarrow (\neg q \to \neg p)$ b) $(p \to q) \to (q \to p)$

   c) $(p \leftrightarrow q) \vee (p \oplus \overline{q})$ d) $(p \oplus q) \to (p \oplus \overline{q})$

   e) $(p \leftrightarrow q) \vee (p \oplus \overline{q})$ f) $(\overline{p} \leftrightarrow \overline{q}) \leftrightarrow (p \leftrightarrow q)$

   g) $(p \vee q) \wedge \overline{r}$ h) $(p \wedge q) \vee \overline{r}$

   i) $(p \leftrightarrow q) \vee (\overline{q} \leftrightarrow r)$ j) $(\overline{p} \leftrightarrow \overline{q}) \leftrightarrow (q \leftrightarrow r)$

2- Dùng bảng chân lý chứng minh:

   a) Luật giao hoán

   $p \vee q \Leftrightarrow q \vee p$

p ∧ q ⇔ q ∧ p

b) Luật kết hợp

(p ∨ q) ∨ r ⇔ p ∨ (q ∨ r)

(p ∧ q) ∧ r ⇔ p ∧(q ∧ r)

c) Luật phân phối

p ∧ (q ∨ r) ⇔ (p ∧ q) ∨ (p ∧ r)

3- Chứng minh các công thức sau đây là đồng nhất đúng bằng cách lập bảng giá trị chân lý:

a) (X→(Y→Z)) →((X →Y)→(X→Z));

b) (X→Y)→((X→Z)→(X→(Y∧Z)));

c) (X→Z) →((Y→Z)→((X∨Y)→Z)).

4. Chứng minh các công thức sau đây là tương đương logic

$$a)\quad X \vee \quad (Y_1 \wedge Y_2 \wedge ... \wedge Y_n \Leftrightarrow (X \vee Y_1) \wedge (X \vee Y_2) \wedge ... \wedge (X \vee Y_n)$$

$$b)\quad X \wedge \quad (Y_1 \vee Y_2 \vee ... \vee Y_n \Leftrightarrow (X \wedge Y_1) \vee (X \wedge Y_2) \vee ... \vee (X \wedge Y_n)$$

$$c)\quad \overline{(X_1 \vee X_2 \vee \cdots \vee X_n} \Leftrightarrow \quad \overline{X_1} \wedge \overline{X_1} \wedge \cdots \wedge \overline{X_n}$$

$$d)\quad \overline{X_1 \wedge X_2 \wedge \cdots \wedge X_n} \Leftrightarrow \quad \overline{X_1} \vee \overline{X_2} \vee \cdots \vee \overline{X_n}$$

5. Cho A, B, C là các tập hợp. Chứng minh rằng:

$$a)\quad \overline{A \cap B \cap C} = \overline{A} \cup \overline{B} \cup \overline{C}$$

$$b)\quad (A \cap B \cap C) \subseteq (A \cap B)$$

$$c)\quad (A - B) - C \subseteq (A - C)$$

$$d)\quad (A - C) \cap (C - B) = \Phi$$

$$e)\quad (B - A) \cup (C - A) = (B \cup C) - A$$

$$f)\quad A - B = A \cap \overline{B}$$

$$g)\quad (A \cap B) \cup (A \cap \overline{B} = A$$

CHƯƠNG 2. BÀI TOÁN ĐẾM

Đếm các đối tượng có những tính chất nào đó là một bài toán quan trọng của lý thuyết tổ hợp. Giải quyết tốt bài toán đếm giúp ta giải nhiều bài toán khác nhau trong đánh giá độ phức tạp tính toán của các thuật toán và tìm xác suất rời rạc các biến cố. Phương pháp chung để giải bài toán đếm được dựa trên các nguyên lý đếm cơ bản (nguyên lý cộng, nguyên lý nhân). Một số bài toán đếm phức tạp hơn được giải bằng phương pháp qui về các bài toán con, xây dựng công thức truy hồi hoặc phương pháp hàm sinh. Nội dung chính được đề cập trong chương này bao gồm:

- ✓ Các nguyên lý đếm cơ bản
- ✓ Nguyên lý bù trừ
- ✓ Hoán vị và tổ hợp
- ✓ Hệ thức truy hồi
- ✓ Qui về các bài toán con
- ✓ Giới thiệu bài toán tồn tại

Bạn đọc có thể tìm hiểu nhiều kỹ thuật đếm cao cấp hơn trong tài liệu [1], [2] trong phần tham khảo của tài liệu này.

2.1. Những nguyên lý đếm cơ bản

2.1.1. Nguyên lý cộng

Giả sử có hai công việc. Việc thứ nhất có thể tiến hành bằng n_1 cách, việc thứ hai có thể tiến hành bằng n_2 cách và nếu hai việc này không thể tiến hành đồng thời. Khi đó sẽ có $n_1 + n_2$ cách để giải giải quyết một trong hai việc trên.

Chúng ta có thể mở rộng qui tắc cộng cho trường hợp nhiều hơn hai công việc. Giả sử các việc $T_1, T_2,.., T_m$ có thể làm tương ứng bằng $n_1, n_2, .., n_m$ cách và giả sử không có hai việc T_i, T_j nào làm việc đồng thời ($i,j = 1, 2, .., m$; $i \neq j$). Khi đó, có $n_1 + n_2 +.. +n_m$ cách thực hiện một trong các công việc $T_1, T_2, . ., T_m$.

Qui tắc cộng được phát biểu dưới dạng của ngôn ngữ tập hợp như sau:

- Nếu A và B là hai tập rời nhau ($A \cap B = \phi$) thì : $N(A\cup B) = N(A) + N(B)$.
- Nếu $A_1, A_2, .., A_n$ là những tập hợp rời nhau thì:

$$N(A_1 \cup A_2 \cup . . \cup A_n) = N(A_1) + N(A_2) +..+ N(A_n).$$

Ví dụ 1. Giả sử cần chọn hoặc một cán bộ hoặc một sinh viên tham gia một hội đồng của một trường đại học. Hỏi có bao nhiêu cách chọn vị đại biểu này nếu như có 37 cán bộ và 63 sinh viên.

Lời giải: Gọi việc thứ nhất là chọn một cán bộ từ tập cán bộ ta có 37 cách. Gọi việc thứ hai là chọn một sinh viên từ tập sinh viên ta có 63 cách. Vì tập cán bộ và tập sinh viên là rời nhau, theo nguyên lý cộng ta có tổng số cách chọn vị đại biểu này là 37 + 63 = 100 cách chọn.

Ví dụ 2. một đoàn vận động viên gồm môn bắn súng và bơi được cử đi thi đấu ở nước ngoài. Số vận động viên nam là 10 người. Số vận động viên thi bắn súng kể cả nam và nữ là 14 người. Số nữ vận động viên thi bơi bằng số vận động viên nam thi bắn súng. Hỏi đoàn có bao nhiêu người.

Lời giải. Chia đoàn thành hai tập, tập các vận động viên nam và tập các vận động viên nữ. Ta nhận thấy tập nữ lại được chia thành hai: thi bắn súng và thi bơi. Thay số nữ thi bơi bằng số nam thi bắn súng , ta được số nữ bằng tổng số vận động viên thi bắn súng. Từ đó theo nguyên lý cộng toàn đoàn có 14 + 10 = 24 người.

Ví dụ 3. giá trị của biến k sẽ bằng bao nhiêu sau khi thực hiện đoạn chương trình sau :

```
k := 0
for i1:= 1to n1
        k:=k+1
for i2:= 1to n2
        k:=k+1
. . . . . . . . . .
for im:= 1 to nm
        k:=k+1
```

Lời giải. coi mỗi vòng for là một công việc, do đó ta có m công việc T_1, T_2, . ., T_m. Trong đó T_i thực hiện bởi n_i cách (i= 1, 2, .., m). Vì các vòng for không lồng nhau hay các công việc không thực hiện đồng thời nên theo nguyên lý cộng tổng tất cả các cách để hoàn thành T_1, T_2,.., T_m là $k= n_1 + n_2 +.. + n_m$.

2.1.2. Nguyên lý nhân

Giả sử một nhiệm vụ nào đó được tách ra hai công việc. Việc thứ nhất được thực hiện bằng n_1 cách, việc thứ hai được thực hiện bằng n_2 cách sau khi việc thứ nhất đã được làm, khi đó sẽ có $n_1.n_2$ cách thực hiện nhiệm vụ này. Nguyên lý nhân có thể được phát biểu tổng quát bằng ngôn ngữ tập hợp như sau:

Nếu A_1, A_2, .., A_m là những tập hợp hữu hạn, khi đó số phần tử của tích đề các các tập này bằng tích số các phần tử của mỗi tập thành phần. Hay đẳng thức:

$$N (A_1\times A_2\times .. A_m) = N (A_1) N (A_2) . . . N (A_m).$$

Nếu $A_1 = A_2 =. . A_m$ thì $N(A^k) = N(A)^k$

Ví dụ 1. giá trị của k sẽ bằng bao nhiêu sau khi ta thực hiện đoạn chương trình sau:

k:=0

for i_1 = 1 to n_1

for i_2 = 1 to n_2

………

. . ……

for i_n =1 to n_m

k:=k +1

Lời giải. Giá trị khởi tạo k=0. Mỗi vòng lặp kồng nhau đi qua giá trị của k được tăng lên 1 đơn vị. Gọi T_i là việc thi hành vòng lặp thứ i. Khi đó, số lần vòng lặp là số cách thực hiện công việc. Số cách thực hiện công việc T_j là n_j (j=1,2, . ., n). Theo qui tắc nhân ta vòng lặp kép được duyệt qua $n_1 + n_2 + .. + n_m$ lần và chính là giá trị của k.

Ví dụ 2. Người ta có thể ghi nhãn cho những chiếc ghế của một giảng đường bằng một chữ cái và sau đó là một số nguyên nhỏ hơn 100. Bằng cách như vậy hỏi có nhiều nhất bao nhiêu chiếc ghế có thể ghi nhãn khác nhau.

Lời giải: có nhiều nhất là 26 x 100 = 2600 ghế được ghi nhãn. Vì kí tự gán nhãn đầu tiên là một chữ cái vậy có 26 cách chọn các chữ cái khác nhau để ghi kí tự đầu tiên, tiếp theo sau là một số nguyên dương nhỏ hơn 100 do vậy có 100 cách chọn các số nguyên để gán tiếp sau của một nhãn. Theo qui tắc nhân ta nhận được 26 x 100 = 2600 nhãn khác nhau.

Ví dụ 3. Có bao nhiêu xâu nhị phân có độ dài 7?

Lời giải. một xâu nhị phân có độ dài 7 gồm 7 bít, mỗi bít có hai cách chọn (hoặc giá trị 0 hoặc giá trị 1), theo qui tắc nhân ta có $2.2.2.2.2.2.2 = 2^7 = 128$ xâu bít nị phân độ dài 7.

Ví dụ 4. Có bao nhiêu hàm đơn ánh xác định từ một tập A có m phần tử nhận giá trị trên tập B có n phần tử?

Lời giải. Trước tiên ta nhận thấy, nếu m >n thì tồn tại ít nhất hai phần tử khác nhau của A cùng nhận một giá trị trên B, như vậy với m>n thì số các hàm đơn ánh từ $A \rightarrow B$ là 0. Nếu m<=n, khi đó phần tử đầu tiên của A có n cách chọn, phần tử thứ hai có n-1 cách chọn, . ., phần tử thứ k có n-k+1 cách chọn. Theo qui tắc nhân ta có n(n-1) (n-2) . . .(n-m+1) hàm đơn ánh từ tập A sang tập B.

Ví dụ 5. Dạng của số điện thoại ở Bắc Mỹ được qui định như sau: số điện thoại gồm 10 chữ số được tách ra thành một nhóm mã vùng gồm 3 chữ số, nhóm mã chi nhánh gồm 3 chữ số và nhóm mã máy gồm 4 chữ số. Vì những nguyên nhân kỹ thuật nên có một số hạn chế đối với một số con số. Ta giả sử, X biểu thị một số có thể nhận các giá trị từ 0..9, N là số có thể nhận các chữ số từ 2..9, Y là các số có thể nhận các chữ số 0 hoặc 1. Hỏi theo hai dự án đánh số NYX NNX XXXX và NXX NXX XXXX có bao nhiêu số điện thoại được đánh số khác nhau ở Bắc Mỹ?

Lời giải: đánh số theo dự án NYX NNX XXXX được nhiều nhất là :

$8 \times 2 \times 10 \times 8 \times 8 \times 10 \times 10 \times 10 \times 10 \times 10 \times 10 = 2 \times 8^3 \times 10^6 = 1\ 024.\ 10^6$

đánh số theo dự án NXX NXX XXXX được nhiều nhất là :

$8 \times 10 \times 10 \times 8 \times 10 \times 10 \times 10 \times 10 \times 10 \times 10 \times 10 = 8^2 \times 10^8 = 64.\ 10^8$

Ví dụ 6. Dùng qui tắc nhân hãy chỉ ra rằng số tập con của một tập S hữu hạn là $2^{N(S)}$.

Lời giải. ta liệt kê các phần tử của tập S là $s_1, s_2, .., s_{N(S)}$. Xây dựng một xâu bít nhị phân dài N(S) bít, trong đó nếu bít thứ i có giá trị 0 thì phần tử $s_i \notin S$, nếu bít thứ i có giá trị 1 thì phần tử $s_i \in S$ (i=1, 2, .., N(S)). Như vậy, theo nguyên lý nhân, số tập con của tập hợp S chính là số xâu bít nhị phân có độ dài N(S). Theo ví dụ 3, chúng ta có $2^{N(S)}$ xâu bít nhị phân độ dài N(S).

2.2. Nguyên lý bù trừ

Trong một số bài toán đếm phức tạp hơn. Nếu không có giả thiết gì về sự rời nhau giữa hai tập A và B thì $N(A \cup B) = N(A) + N(B) - N(A \cap B)$.

Ví dụ 1. lớp toán học rời rạc có 25 sinh viên giỏi tin học, 13 sinh viên giỏi toán và 8 sinh viên giỏi cả toán và tin học. Hỏi lớp có bao nhiêu sinh viên nếu mỗi sinh viên hoặc giỏi toán hoặc học giỏi tin học hoặc giỏi cả hai môn?

Lời giải. Gọi A tập là tập các sinh viên giỏi Tin học, B là tập các sinh viên giỏi toán. Khi đó $A \cap B$ là tập sinh viên giỏi cả toán học và tin học. Vì mỗi sinh viên trong lớp hoặc giỏi toán, hoặc giỏi tin học hoặc giỏi cả hai nên ta có tổng số sinh viên trong lớp là $N(A \cup B)$. Do vậy ta có:

$N(A \cup B) = N(A) + N(B) - N(A \cap B) = 25 + 13 - 8 = 30.$

Ví dụ 2. Có bao nhiêu số nguyên không lớn hơn 1000 chia hết cho 7 hoặc 11.

Lời giải. Gọi A là tập các số nguyên không lớn hơn 1000 chia hết cho 7, B là tập các số nguyên không lớn hơn 1000 chia hết cho 11. Khi đó tập số nguyên không lớn hơn 1000 hoặc chia hết cho 7 hoặc chia hết cho 11 là $N(A \cup B)$. Theo công thức 1 ta có:

$$N(A \cup B) = N(A) + N(B) - N(A \cap B) = \lfloor 1000/7 \rfloor + \lfloor 1000/11 \rfloor - \lfloor 1000/7.11 \rfloor$$
$$= 142 + 90 - 12 = 220.$$

Trước khi đưa ra công thức tổng quát cho n tập hợp hữu hạn. Chúng ta đưa ra công thức tính số phần tử của hợp 3 tập A, B, C.

Ta nhận thấy $N(A) + N(B) + N(C)$ đếm một lần những phần tử chỉ thuộc một trong ba tập hợp. Như vậy, số phần tử của $A\cap B$, $A\cap C$, $B\cap C$ được đếm hai lần và bằng $N(A\cap B)$, $N(A\cap C)$, $N(B\cap C)$, được đếm ba lần là những phần tử thuộc $A\cap B\cap C$. Như vậy, biểu thức:

$$N(A\cup B\cup C) - N(A\cap B)- N(A\cap C) - N(B\cap C)$$

chỉ đếm các phần tử chỉ thuộc một trong ba tập hợp và loại bỏ đi những phần tử được đếm hai lần. Như vậy, số phần tử được đếm ba lần chưa được đếm, nên ta phải cộng thêm với giao của cả ba tập hợp. Từ đó ta có công thức đối với 3 tập không rời nhau:

$$N(A\cup B\cup C)= N(A)+N(B)+N(C)-N(A\cap B)-N(A\cap C)-N(B\cap C)+N(A\cap B\cap C)$$

Định lý. Nguyên lý bù trừ. Giả sử $A_1, A_2, . ., A_m$ là những tập hữu hạn. Khi đó

$$N(A_1\cup A_2 \cup. . .\cup A_m) = N_1- N_2 + . . +(-1)^{m-1}N_m,$$

trong đó N_k là tổng phần tử của tất cả các giao của k tập lấy từ m tập đã cho. (nói riêng $N_1=N(A_1) + N(A_2) + . .+ N(A_m)$, $N_m = N(A_1 \cap A_2 \cap . . .\cap A_m)$. Nói cách khác:

$$N(A_1 \cup A_2 \cup ...A_n) = \sum_{1\le i\le n} N(A_i) - \sum_{1\le i,j\prec n} N(A_i \cap A_j) + \sum_{1\le i<j<k\le n} N(A_i \cap A_j \cap A_k$$
$$-... + (-1)^{n+1} N(A_1 \cap A_2 \cap .. \cap A_n)$$

Định lý được chứng minh bằng cách chỉ ra mỗi phần tử của hợp n tập hợp được đếm đúng một lần. Bạn đọc có thể tham khảo cách chứng minh trong tài liệu [1].

Ví dụ 3. Tìm công thức tính số phần tử của 4 tập hợp.

Lời giải. Từ nguyên lý bù trừ ta có

$$N(A_1\cup A_2\cup A_3\cup A_4) = N(A_1) + N(A_2) + N(A_3) + N(A_4) - N(A_1\cap A_2) - N(A_1\cap A_3) - N(A_1\cap A_4) - N(A_2\cap A_3) - N(A_2\cap A_4) - N(A_3\cap A_4) + N(A_1\cap A_2\cap A_3) + N(A_1\cap A_2\cap A_4) + N(A_1\cap A_3\cap A_4 + N(A_2\cap A_3\cap A_4) - N(A_1\cap A_2\cap A_3\cap A_4).$$

Ví dụ 4. Hỏi trong tập X = { 1, 2, . ., 10000} có bao nhiêu số không chia hết cho bất cứ số nào trong các số 3, 4, 7?

Lời giải. Gọi A là tập các số nhỏ hơn 10000 chia hết cho 3, B là tập các số nhỏ hơn 10000 chia hết cho 4, C là tập các số nhỏ hơn 10000 chia hết cho 7. Theo nguyên lý bù trừ ta có:

$$N(A\cup B\cup C)= N(A)+N(B) + N(C) - N(A\cap B - N(A\cap C) - N(B\cap C) + N(A\cap B\cap C)$$

trong đó :

$N(A) + N(B) + N (C) = [10 000/3] + [10 000/4] + [10 000/7]$

$= 3333 + 2500 + 1428 = 7261$

$N(A\cap B) = N(A) + N(B) - N(A\cap B) = 3333 + 2500 - [10000/3x4] = 833$

N(A∩C) = N(A) + N(C) – N(A∩C) = 3333 + 1428 – [10000/3x7] = 476

N(B∩C) = N(B) + N(C) – N(B∩C) = 2500 + 1428 – [10000/4x7] = 357

N(A∩B) + N(A∩C) + N(B∩C) = 833 + 476 + 357 = 1666

N(A∩B∩C) = [10000/3x4x7] = 119.

=>Số các số nhỏ hơn 10000 cần đếm là :

1000 - N(A∪B∪C) = 7261 – 1666 + 119 = 4286.

Ví dụ 5. Có bao nhiêu xâu nhị phân độ dài 10 bắt đầu bởi 00 hoặc kết thúc bởi 11.

Lời giải. Gọi A là số xâu nhị phân độ dài 10 bắt đầu bởi 00 , B là số xâu nhị phân độ dài 10 kết thúc bởi 11. Dễ ràng nhận thấy, N(A) = N(B) = 256, N(A∩B) = 2^6 = 64. Theo nguyên lý bù trừ ta có:

N(A∪B) = N(A) + N(B) – N(A∩B)

= 256 + 256 – 64 = 448.

Ví dụ 6. Bài toán bỏ thư. Có n lá thư và n phong bì ghi sẵn địa chỉ. Bỏ ngẫu nhiên các lá thư vào các phong bì. Hỏi xác suất để xảy ra không một là thư nào bỏ đúng địa chỉ là bao nhiêu?

Lời giải: Có tất cả n! cách bỏ thư. Vấn đề đặt ra là đếm số cách bỏ thư sao cho không lá thư nào đúng địa chỉ. Gọi X là tập hợp tất cả các cách bỏ thư và A_k là tính chất lá thư k bỏ đúng địa chỉ. Khi đó theo nguyên lý bù trừ ta có:

$$\overline{N} = N - N_1 + N_2 - ... + (-1)^n N_n$$

Trong đó $\overline{N}$ là số cần tìm, N = n!, N_k là số tất cả các cách bỏ thư sao cho có k lá thư đúng địa chỉ. Nhận xét rằng, N_k là mọi cách lấy k lá thư từ n lá, với mỗi cách lấy k lá thư, có (n-k)! cách bỏ để k lá thư này đúng địa chỉ, từ đó ta nhận được.

$$N_k = C(n,k)(n-k)! = \frac{n!}{k!} \text{ và } \overline{N} = n!(1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + \frac{(-1)^n}{n!}$$

Từ đó ta có xác xuất cần tìm là:

$$1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + \frac{(-1)^n}{n!} = e^{-1}$$

Số được tính như trên được gọi là số mất thứ tự và được ký hiệu là D_n. Dưới đây là một vài giá trị của D_n, sự tăng nhanh của D_n một lần nữa cho ta thấy rõ sự bùng nổ tổ hợp.

N	2	3	4	5	6	7	8	9	10	11
D_n	1	2	9	44	265	1845	14833	133496	1334961	4890741

2.3. Đếm các hoán vị và tổ hợp

2.3.1. Chỉnh hợp lặp

Định nghĩa 1. Một chỉnh hợp lặp chập k của n phần tử là bộ có thứ tự gồm k thành phần lấy từ n phần tử của tập đã cho.

Như vậy, một chỉnh hợp lặp chập k của n phần tử có thể xem là phần tử của tích đề các A^k với A là tập đã cho. Theo nguyên lý nhân, số các tất cả các chỉnh hợp lặp chập k của n sẽ là n^k.

Ví dụ 1. Tính số hàm từ tập có k phần tử vào tập có n phần tử?

Lời giải: Biểu diễn mỗi hàm bằng một bộ k thành phần, trong đó thành phần thứ i là ảnh của phần tử thứ i (1<=i<=k). Mỗi thành phần được lấy ra từ một trong n giá trị. Từ đó suy ra số hàm là số bộ k thành phần lấy từ n thành phần bằng n^k.

Ví dụ 2. Từ bảng chữ cái tiếng Anh có thể tạo ra được bao nhiêu xâu có độ dài n?

Lời giải : Bảng chữ cái tiếng Anh gồm 26 kí tự ['A'..'Z'], số các xâu có độ dài n được chọn từ 26 chữ cái chính là chỉnh hợp lặp n của 26 phần tử và bằng 26^n.

Ví dụ 3. Tính xác xuất lấy ra liên tiếp được 3 quả bóng đỏ ra khỏi bình kín chứa 5 quả đỏ, 7 quả xanh nếu sau mỗi lần lấy một quả bóng ra lại bỏ nó trở lại bình?

Lời giải: Số kết cục có lợi để ta lấy ra liên tiếp 3 quả bóng đỏ là 5^3 vì có 5 quả đỏ ta phải lấy 3 quả (chú ý vì có hoàn lại). Toàn bộ kết cục có thể để lấy ra ba quả bóng bất kỳ trong 12 quả bóng là 12^3. Như vậy, xác suất để có thể lấy ra 3 quả bóng đỏ liên tiếp là $5^3/12^3$.

2.3.2. Chỉnh hợp không lặp

Định nghĩa 2. Chỉnh hợp không lặp chập k của n phần tử là bộ có thứ tự gồm k thành phần lấy ra từ n phần tử đã cho. Các phần tử không được lặp lại.

Để xây dựng một chỉnh hợp không lặp, ta xây dựng từ thành phần đầu tiên. Thành phần này có n khả năng chọn. Mỗi thành phần tiếp theo những khả năng chọn giảm đi 1 (vì không được lấy lặp lại). Tới thành phần thứ k có n-k + 1 khả năng chọn. Theo nguyên lý nhân ta có số chỉnh hợp không lặp chập k của tập hợp n phần tử ký hiệu là P(n, k) được tính theo công thức:

$$P(n,k) = n(n-1)..(n-k+1) = \frac{n!}{(n-k)!}$$

Ví dụ 1. Tìm số hàm đơn ánh có thể xây dựng được từ tập k phần tử sang tập n phần tử?

Lời giải: Số hàm đơn ánh từ tập k phần tử sang tập n phần tử chính là P(n,k).

Ví dụ 2. Giả sử có tám vận động viên chạy thi. Người về nhất sẽ được nhận huân chương vàng, người về nhì nhận huân chương bạc, người về ba nhận huy chương đồng. Hỏi có bao nhiêu cách trao huy chương nếu tất cả các kết cục đều có thể xảy ra?

Lời giải. Số cách trao huy chương chính là số chỉnh hợp chập 3 của tập hợp 8 phần tử. Vì thế có P(8,3) = 8.7.6 = 336 cách trao huy chương.

Ví dụ 3. Có bao nhiêu cách chọn 4 cầu thủ khác nhau trong đội bóng gồm 10 cầu thủ để tham gia các trận đấu đơn.

Lời giải. Có P(10,4) = 10.9.8.7 = 5040 cách chọn.

2.3.3. Hoán vị

Định nghĩa 3. 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.

Ta cũng có thể đồng nhất một hoán vị với một song ánh từ tập n phần tử lên chính nó. Như vậy, số hoán vị của tập gồm n phần tử là P(n, n) = n!.

Ví dụ 1. Có 6 người xếp thành hàng để chụp ảnh. Hỏi có thể bố trí chụp được bao nhiêu kiểu khác nhau?

Lời giải. Mỗi kiểu ảnh là một hoán vị của 6 người. Do đó có 6! = 720 kiểu ảnh khác nhau có thể chụp.

Ví dụ 2. Cần bố trí thực hiện n chương trình trên một máy tính. Hỏi có bao nhiêu cách bố trí khác nhau?

Lời giải. Số chương trình được đánh số từ 1, 2, . ., n. Như vậy, số chương trình cần thực hiện trên một máy tính là số hoán vị của 1, 2, .., n.

Ví dụ 3. Một thương nhân đi bán hàng tại tám thành phố. Chị ta có thể bắt đầu hành trình của mình tại một thành phố nào đó nhưng phải qua 7 thành phố kia theo bất kỳ thứ tự nào mà chị ta muốn. Hỏi có bao nhiêu lộ trình khác nhau mà chị ta có thể đi?

Lời giải. Vì thành phố xuất phát đã được xác định. Do vậy thương nhân có thể chọn tuỳ ý 7 thành phố còn lại để hành trình. Như vậy, tất cả số hành trình của thương nhân có thể đi qua là 7! = 5040 cách.

2.3.4. Tổ hợp

Định nghĩa 4. Một tổ hợp chập k của n phần tử là một bộ không kể thứ tự gồm k thành phần khác nhau lấy từ n phần tử đã cho. Nói cách khác, ta có thể coi một tổ hợp chập k của n phần tử là một tập con k phần tử lấy trong n phần tử. Số tổ hợp chập k của n phần tử ký hiệu là C(n,k).

Ta có thể tính được trực tiếp số các tổ hợp chập k của tập n phần tử thông qua chỉnh hợp không lặp của k phần tử. Xét tập hợp tất cả các chỉnh hợp không lặp chập k của n phần tử. Sắp xếp chúng thành những lớp sao cho hai chỉnh hợp thuộc cùng một lớp chỉ khác nhau về thứ tự. Rõ ràng mỗi lớp như vậy là một tổ hợp chập k của n phần tử(P(n,k)). Số chỉnh hợp trong mỗi lớp đều bằng nhau và bằng k! (số hoán vị k phần tử: P(k,k)). Số các lớp bằng số tổ hợp chập k của n (P(n,k)). Từ đó ta có:

$$P(n,k) = C(n,k).P(k,k) \Rightarrow C(n,k) = \frac{P(n,k)}{k!} = \frac{n!}{k!(n-k)!}$$

Ví dụ 1. Cho S = { a, b, c, d } tìm C(4,2)?

Lời giải. Rõ ràng C(4,2) = 6 tương ứng với 6 tập con {a, b}, {a, c}, {a, d}, {b,c}, {b, d} {c,d}.

Ví dụ 2. Có n đội bóng thi đấu vòng tròn. Hỏi phải tổ chức bao nhiêu trận đấu.

Lời giải: Cứ hai đội bóng thì có một trận. Từ đó suy ra số trận đấu sẽ bằng số cách chọn 2 trong n đội, nghĩa là bằng C(n, 2) = n! / 2!(n-2)! = n(n-1)/2 trận đấu.

Ví dụ 3. Chứng minh

a) C(n,k) = C(n, n-k)

b) C(n, 0) = C(n,n)= 1

c) C(n,k) = C(n-1,k-1) + C(n-1,k)

Lời giải

a. C(n,n-k) = n!/(n-k)! (n-n+k)! = n!/k!(n-k)! = C(n,k).

Hoặc C(n, k) = n!/k!(n-k)! = n!/ (n-k)! (n-(n-k))! = C(n, n-k);

b) Chú ý 0!=1 => b hiển nhiên đúng

c) C(n,k) = C(n-1,k-1) + C(n-1,k)

$$C(n-1,k-1) + C(n-1,k) = \frac{(n-1)!}{(k-1)!(n-1-k+1)!} + \frac{(n-1)!}{k!(n-k-1)!}$$

$$= \frac{(n-1)!}{(k-1)!(n-k-1)!}\left(\frac{1}{n-k} + \frac{1}{k}\right) = \frac{(n-1)!.n}{(k-1)!k(n-k-1)!(n-k)}$$

$$= \frac{n!}{k!(n-k)!} = C(n.k)$$

Từ những tính chất trên, ta có thể tính tất cả các hệ số tổ hợp chỉ bằng phép cộng. Các hệ số này được tính và viết lần lượt theo dòng, trên mỗi dòng ta tính và thực hiện theo cột. Bảng có dạng tam giác chính là tam giác Pascal.

Các hệ số tổ hợp có liên quan chặt chẽ tới việc khai triển luỹ thừa của một nhị thức. Thực vậy, trong tích

$(x+y)^n = (x+y)(x+y)...(x+y)$ hệ số của $x^k y^{k-n}$ sẽ là số cách chọn k phần tử (x+y) mà từ đó lấy ra x và đồng thời (n-k) nhân tử còn lại lấy ra y, nghĩa là:

$$(x+y)^n = C(n,0)x^n + C(n,1)x^{n-1}y + ... + C(n,n-1)xy^{n-1} + C(n,0)y^n = \sum_{k=0}^{n} C(n,k)x^{n-k}y^k$$

Công thức trên còn được gọi là khai triển nhị thức Newton, các hệ số tổ hợp còn được gọi là hệ số nhị thức. Chẳng hạn luỹ thừa bậc 8 của nhị thức $(x+y)^8$ được khai triển như sau:

$$(x+y)^8 = x^8 + 8x^7y + 28x^6y^2 + 56x^5y^3 + 70x^4y^4 + 56x^3y^5 + 28x^2y^6 + 8xy^7 + y^8$$

Trong trường hợp y=1, tức khai triển $(x+1)^n$ ta có:

$$(x+1)^n = C(n,0)x^n + C(n,1)x^{n-1} + ... + C(n,n-1)x + C(n,n)$$

Hoặc đẳng thức sau sẽ được rút ra từ khai triển nhị thức Newton:

$$2^n = (1+1)^n = C(n,0) + C(n,1) + ... + C(n,n-1) + C(n,n)$$

Có thể nói rất nhiều đẳng thức về hệ số tổ hợp sẽ được suy ra. Như tính các tập lẻ, đạo hàm.

2.3.5. Tổ hợp lặp

Từ Định nghĩa tổ hợp, ta có thể tính toán được số tổ hợp lặp chập k từ tập n phần tử bằng *C(n + k -1, k*).

Ví dụ 1. Phương trình $x_1 + x_2 + x_3 = 11$ có bao nhiêu nghiệm nguyên không âm?

Lời giải. Mỗi nghiệm nguyên không âm của phương trình ứng với một cách chọn 11 phần tử từ một tập có 3 loại, sao cho có x_1 phần tử loại 1 được chọn, x_2 phần tử loại 2 được chọn, x_3 phần tử loại 3 được chọn. Số này chính bằng số tổ hợp lặp chập 11 từ tập có 3 phần tử. Vì vậy, số nghiệm nguyên không âm của phương trình là:

C(3 + 11 -1, 11) = C (13, 11) = (13.12) / 2 = 78.

Ví dụ 2. Phương trình $x_1 + x_2 + x_3 = 11$ có bao nhiêu nghiệm nguyên không âm thỏa mãn $x_1 \geq 1$, $x_2 \geq 2$, $x_3 \geq 3$?

Lời giải. Mỗi nghiệm nguyên không âm của phương trình ứng với một cách chọn 11 phần tử từ một tập có 3 loại, sao cho có x_1 phần tử loại 1 được chọn, x_2 phần tử loại 2 được chọn, x_3 phần tử loại 3 được chọn. Trong đó, có ít nhất một phần tử loại 1, hai phần tử loại 2 và ba phần tử loại 3. Vì thế ta chọn một phần tử loại 1, hai phần tử loại 2, ba phần tử loại 3 sau đó chọn thêm 5 phần tử nữa. Số này chính bằng số tổ hợp lặp chập 5 từ tập có 3 phần tử. Vì vậy, số nghiệm nguyên không âm của phương trình là:

C(3 + 5 -1, 5) = C (7, 5) = (7.6) / 2 = 21.

Ví dụ 3. Phương trình $x_1 + x_2 + x_3 = 11$ có bao nhiêu nghiệm nguyên không âm thỏa mãn $1 \leq x_1 \leq 3$?

Lời giải. Ta nhận thấy, số nghiệm nguyên không âm của phương trình thỏa mãn $1\leq x_1\leq 3$ chính bằng số nghiệm nguyên không âm của phương trình thỏa mãn $x_1\geq 1$ sau đó trừ bớt đi những nghiệm $x_1\geq 4$. Từ Ví dụ 1 ta tính được số nghiệm nguyên không âm của phương trình thỏa mãn $1\leq x_1\leq 3$ là

$$C(3 + 10 - 1, 10) - C(3 + 7 - 1, 7) = C(12, 10) - C(9, 7).$$

2.4. Hệ thức truy hồi

Thông thường người ta thường quan tâm tới những bài toán đếm trong đó kết quả đếm phụ thuộc vào một tham số đầu vào (mà ta ký hiệu là n), chẳng hạn như các số mất thứ tự D_n . Việc biểu diễn kết quả này như một hàm của n bằng một số hữu hạn các phép toán không phải là đơn giản. Trong nhiều trường hợp, việc tìm ra một công thức trực tiếp giữa kết quả đếm và n là hết sức khó khăn và nhiều khi không giải quyết được, trong khi đó công thức liên hệ giữa kết quả đếm ứng với giá trị n với các kết quả bé hơn n lại đơn giản và dễ tìm. Thông qua công thức này và một vài giá trị ban đầu, ta có thể tính mọi giá trị còn lại khác. Công thức đó gọi là công thức truy hồi hay công thức đệ qui. Đặc biệt, công thức truy hồi rất thích hợp với lập trình trên máy tính. Nó cũng cho phép giảm đáng kể độ phức tạp cũng như gia tăng độ ổn định của quá trình tính toán.

2.4.1. Định nghĩa và ví dụ

Định nghĩa 1. Hệ thức truy hồi đối với dãy số $\{a_n\}$ là công thức biểu diễn a_n qua một hay nhiều số hạng đi trước của dãy, cụ thể là $a_1, a_2, .., a_{n-1}$ với mọi $n\geq n0$ nguyên dương. Dãy số được gọi là lời giải hay nghiệm của hệ thức truy hồi nếu các số hạng của nó thoả mãn hệ thức truy hồi.

Ví dụ 1. Lãi kép. Giả sử một người gửi 10000 đô la vào tài khoản của mình tại một ngân hàng với lãi xuất kép 11% mỗi năm. Hỏi sau 30 năm anh ta có bao nhiêu tiền trong tài khoản của mình?

Lời giải. Gọi P_n là tổng số tiền có trong tài khoản sau n năm. Vì số tiền có trong tài khoản sau n năm bằng số tiền có được trong n-1 năm cộng với lãi xuất năm thứ n. Nên dãy $\{P_n\}$ thoả mãn hệ thức truy hồi :

$$P_n = P_{n-1} + 0.11_{Pn-1} = 1.11P_{n-1}$$

Chúng ta có thể dùng phương pháp lặp để tìm công thức trên cho P_n. Dễ nhận thấy rằng:

$P_0 = 10000$

$P_1 = 1.11P_0$

$P_2 = 1.11P_1 = (1.11)^2P_0$

. . . .

$P_n = 1.11P_{n-1} = (1.11)^{n-1}P_0$

Ta có thể chứng minh tính đúng đắn của công thức truy hồi bằng qui nạp.

Thay P_0= 10000, và n = 30 ta được:

$P_{30} = (1.11)^{30}10000 = 228922,97$ $

Ví dụ 2. Họ nhà thỏ và số Fibonaci. Một cặp thỏ sinh đôi (một con đực và một con cái) được thả lên một hòn đảo. Giả sử rằng cặp thỏ sẽ chưa sinh sản được trước khi đầy hai tháng tuổi. Từ khi chúng đầy hai tháng tuổi, mỗi tháng chúng sinh thêm được một cặp thỏ. Tìm công thức truy hồi tính số cặp thỏ trên đảo sau n tháng với giả sử các cặp thỏ là trường thọ.

Số tháng	Số cặp sinh sản	Số cặp thỏ con	Tổng số cặp thỏ
1	0	1	1
2	0	1	1
3	1	1	2
4	1	2	3
5	2	3	5
6	3	5	8
……..	……..	……..	……..

Lời giải: Giả sử f_n là số cặp thỏ sau n tháng. Ta sẽ chỉ ra rằng $f_1, f_2, . ., f_n$ (n=1, 2, . ., n) là các số của dãy fibonaci.

Cuối tháng thứ nhất số cặp thỏ trên đảo là $f_1 = 1$. Vì tháng thứ hai cặp thỏ vẫn chưa đến tuổi sinh sản được nên trong tháng thứ hai $f_2 =1$. Vì mỗi cặp thỏ chỉ được sinh sản sau ít nhất hai tháng tuổi, nên ta tìm số cặp thỏ sau tháng thứ n bằng cách cộng số cặp thỏ sau tháng n-2 và tháng n-1 hay $f_n = f_{n-1} + f_{n-2}$. Do vậy, dãy $\{ f_n\}$ thoả mãn hệ thức truy hồi

$f_n = f_{n-1} + f_{n-2}$ với n>=3 và $f_1 = 1$, $f_2 = 1$.

Ví dụ 3. Tính số mất thứ tự D_n.

Lời giải: Đánh số thư và phong bì thư từ 1 đến n (thư i gửi đúng địa chỉ nếu bỏ vào phong bì i). Một cách bỏ thư đuợc đồng nhất với hoán vị $(a_1, a_2, . ., a_n)$ của $\{ 1, 2, . ., n \}$. Một mất thứ tự được định nghĩa là là một hoán vị $(a_1, a_2, . ., a_n)$ sao cho $a_i \neq i$ với mọi i. Thành phần a_1 có thể chấp nhận mọi giá trị ngoài 1. Với mỗi giá trị k ($k\neq1$) của a_1, xét hai trường hợp:

1. $a_k =1$, khi đó các thành phần còn lại được xác định như một mất thứ tự của n-2 phần tử, tức là số mất thứ tự loại này bằng D_{n-2}.
2. $a_k\neq1$, khi đó các thành phần từ 2 đến n được xác định như một mất thứ tự của n-1 phần tử còn lại, tức là số mất thứ tự này thuộc loại D_{n-1}.

Từ đó ta nhận được công thức

$D_n = (n\text{-}1)\ (D_{n-1} + D_{n-2})$, n>=3 với $D_1 = 0$, $D_2 = 1$.

Mọi giá trị còn lại được tính đơn giản nhờ luật kế thừa.

$D_3 = (3 - 1)\ (0 + 1) \qquad = 2$

$D_4 = (4 - 1)(1 + 2) \qquad = 9$

$D_5 = (5 - 1)(9 + 2) \qquad = 44$

$D_6 = (6 - 1)(9 + 44) \qquad = 265$

$D_7 = (7 - 1)(44 + 265) \qquad = 1854$

$D_8 = (8 - 1)(265 + 1854) \qquad = 14833$

. . . .

Để công thức đúng với n = 2, ta coi $D_0 = 1$

Có thể nhận được số mất thứ tự thông qua công thức truy hồi trên vì:

$D_n = (n-1)(D_{n-1} + D_{n-2}) \Rightarrow D_n - nD_{n-1} = -(D_{n-1} - (n-1)D_{n-2})$

Đặt $V_n = D_n - nD_{n-1}$ ta có:

$D_n - nD_{n-1} = V_n = -V_{n-1} = \cdots = (-1)^{n-1}V_1 = (-1)^n$. Hay ta có thể viết:

$\frac{D_n}{n!} - \frac{D_{n-1}}{(n-1)!} = \frac{(-1)^n}{n!}$. Cộng các hệ thức trên với n = 1, 2, .., n ta được:

$\frac{D_n}{n!} = 1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + \frac{(-1)^n}{n!}$. Từ đó thu lại được công thức cũ:

$D_n = n!(1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + \frac{(-1)^n}{n!})$

Ví dụ 3. Tính hệ số tổ hợp C(n,k).

Lời giải. Chọn phần tử cố định a trong n phần tử đang xét. Chia số cách chọn tập con k phần tử này thành hai lớp (lớp chứa a và lớp không chứa a). Nếu a được chọn thì ta cần bổ xung k-1 phần tử từ n-1 phần tử còn lại, từ đó lớp chứa a gồm C(n-1, k-1) cách. Nếu a không được chọn, thì ta phải chọn k phần tử từ n-1 phần tử còn lại, từ đó lớp không chứa a gồm C(n-1, k) cách. Theo nguyên lý cộng ta được công thức truy hồi:

C(n, k) = C(n-1, k-1) + C(n-1,k) với các giá trị biên được suy ra trực tiếp:

C(n,0) = C(n,n) = 1.

Phương pháp này được gọi là phương pháp khử. Không phải lúc nào cũng dễ dàng khử được công thức truy hồi để đưa về công thức trực tiếp. Tuy nhiên, trong một số trường hợp đặc biệt ta có thể đưa ra phương pháp tổng quát để giải công thức truy hồi.

2.4.2. Giải công thức truy hồi tuyến tính thuần nhất với hệ số hằng số

Định nghĩa 1. Một hệ thức truy hồi tuyến tính thuần nhất bậc k với hệ số hằng số là hệ thức truy hồi có dạng:

$$a_n = c_1 a_{n-1} + c_1 a_{n-2} + \cdots + c_k a_{n-k} \qquad (1)$$

trong đó $c_1, c_2, \ldots, c_k$ là các số thực và $c_k \neq 0$

Ta cần tìm công thức trực tiếp cho số hạng a_n của dãy số $\{a_n\}$ thoả mãn công thức (1). Theo nguyên lý thứ hai của qui nạp toán học thì dãy số thoả mãn định nghĩa trên được xác định duy nhất nếu như nó thoả mãn k điều kiện đầu:

$a_0 = C_0$, $a_1 = C_1$, . ., $a_{k-1} = C_{k-1}$, trong đó C_1, C_2, . ., C_{k-1} là các hằng số.

Ví dụ 1. Hệ thức truy hồi $P_n=(1.11)P_{n-1}$ là hệ thức truy hồi tuyến tính thuần nhất bậc 1. Hệ thức truy hồi $f_n = f_{n-1} + f_{n-2}$ là hệ thức truy hồi tuyến tính thuần nhất bậc 2. Hệ thức truy hồi $a_n = a_{n-5}$ là hệ thức truy hồi tuyến tính thuần nhất bậc 5. Hệ thức truy hồi $B_n=nB_{n-1}$ không phải là hệ thức truy hồi tuyến tính thuần nhất vì nó không có hệ số hằng số.

Phương pháp cơ bản để giải hệ thưc truy hồi tuyến tính thuần nhất là tìm nghiệm dưới dạng $a_n = r^n$, trong đó r là hằng số. Cũng cần chú ý rằng $a_n = r^n$ là nghiệm của hệ thức truy hồi $a_n = c_1a_{n-1} + c_2\, a_{n-2} + \ldots + c_ka_{n-k}$ nếu và chỉ nếu

$$a_n = c_1r_{n-1} + c_2r_{n-2} + \ldots + c_kr_{n-k} .$$

Chia cả hai vế cho rn-k ta nhận được

$$r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_{k-1}r - c_k = 0 \qquad (2)$$

Vậy dãy $\{a_n\}$ với $a_n=r^n$ là nghiệm nếu và chỉ nếu r là nghiệm của (2). Phương trình 2 còn được gọi là phương trình đặc trưng của hệ thức truy hồi, nghiệm của nó là nghiệm đặc trưng của hệ thức truy hồi. Nghiệm của phương trình đặc trưng dùng để biểu diễn công thức tất cả các nghiệm của hệ thức truy hồi.

Chúng ta sẽ trình bày các kết quả với hệ thức truy hồi tuyến tính thuần nhất bậc hai. Sau đó ta sẽ nêu ra những kết quả tương tự cho trường hợp tổng quát khi bậc lớn hơn hai.

Định lý 1. Cho c_1, c_2 là các hằng số thực. Giả sử $r^2 - c_1r + c_2 = 0$ có hai nghiệm phân biệt r_1, r_2. Khi đó dãy $\{a_n\}$ là nghiệm của hệ thức truy hồi $a_n = c_1a_{n-1} + c_2a_{n-2}$ khi và chỉ khi $a_n = \alpha_1 r_1^{\,n} + \alpha_2 r^n{}_2$ với $n = 1, 2, \ldots$, α_1, α_2 là các hằng số.

Chứng minh: Để chứng minh định lý này ta cần thực hiện hai việc. Đầu tiên ta cần chỉ ra rằng nếu r_1, r_2 là hai nghiệm của phương trình đặc trưng và α_1, α_2 là hai hằng số thì dãy $\{a_n\}$ với $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ là nghiệm của hệ thức truy hồi. Ngược lại, cần phải chứng minh rằng nếu $\{a_n\}$ là nghiệm thì $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ với α_1, α_2 là các hằng số nào đó.

($\Rightarrow$): Giả sử r_1 và r_2 là hai nghiệm phân biệt của $r^2 - c_1r + c_2=0$, khi đó $r_1^2 = c_1^2 r_1 + c_2;\quad r_2^2 = c_1 r_2 + c_2$ đồng thời ta thực hiện dãy các phép biến đổi sau:

$$\begin{aligned} c_1 a_{n-1} + c_2 a_{n-2} &= c_1(\alpha_1 r_1^{n-1} + \alpha_2 r_2^{n-2}) + c_2(\alpha_1 r_1^{n-2} + \alpha_2 r_2^{n-2}) \\ &= \alpha_1 r_1^{n-1}(c_1 r_1 + c_2) + \alpha_2 r_2^{n-2}(c_1 r_2 + c_2) \\ &= \alpha_1 r_1^{n-2} r_1^2 + \alpha_2 r_2^{n-2} r_2^2 \\ &= \alpha_1 r_1^n + \alpha_2 r_2^n = a_n \end{aligned}$$

Điều này chứng tỏ dãy $\{a_n\}$ với $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ là nghiệm của hệ thức truy hồi đã cho.

($\Leftarrow$):Để chứng minh ngược lại, ta giả sử dãy $\{a_n\}$ là một nghiệm bất kỳ của hệ thức truy hồi. Ta chọn α_1, α_2 sao cho dãy $\{a_n\}$ với $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ thoả mãn các điều kiện đầu $a_0 = C_0$, $a_1 = C_1$. Thực vậy,

$$\begin{aligned} a_0 &= C_0 = \alpha_1 + \alpha_2 \\ a_1 &= C_1 = \alpha_1 r_1 + \alpha_2 r_2 \end{aligned}$$

Từ phương trình đầu ta có $\alpha_2 = C_0 - \alpha_1$ thế vào phương trình thứ hai ta có:

$C_1 = \alpha_1 r_1 + (C_0 - \alpha_1) r_2 = \alpha_1(r_1 - r_2) + C_0 - r_2$; Từ đây suy ra:

$$\alpha_1 = \frac{(C_1 - C_0 r_2)}{r_1 - r_2}; \alpha_2 = C_0 - \alpha_1 = C_0 - \frac{(C_1 - C_0 r_2)}{r_1 - r_2} = \frac{(C_0 r_1 - C_1)}{r_1 - r_2}.$$

Như vậy, khi chọn những giá trị trên cho α_1, α_2 dãy $\{a_n\}$ với $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ thoả mãn các điều kiện đầu. Vì hệ thức truy hồi và các điều kiện đầu được xác định duy nhất nên $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$. Định lý được chứng minh.

Ví dụ 1. Tìm nghiệm của hệ thức truy hồi $a_n = a_{n-1} + 2a_{n-2}$ với $a_0 = 2$, $a_1 = 7$.

Lời giải. Phương trình đặc trưng của hệ thức truy hồi có dạng $r^2 - r - 2 = 0$. Nghiệm của nó là $r=2$ và $r = -1$. Theo định lý 1, dãy $\{a_n\}$ là nghiệm của hệ thức truy hồi nếu và chỉ nếu :

$a_n = \alpha_1 2^n + \alpha_2(-1)^n$ với α_1, α_2 là các hằng số nào đó. Từ các điều kiện đầu suy ra:

$a_0 = 2 = \alpha_1 + \alpha_2$

$a_1 = 7 = \alpha_1 2 + \alpha_2(-1)$

Giải ra ta được $\alpha_1=3$, $\alpha_2=-1$. Vậy nghiệm của biểu thức truy hồi với điều kiện đầu là dãy $\{a_n\}$ với $a_n = 3.2^n - (-1)^n$.

Ví dụ 2. Tìm công thức hiển của các số fibonaci.

Giải: Các số fibonaci thoả mãn hệ thức $f_n = f_{n-1} + f_{n-2}$ và các điều kiện đầu $f_0 = 0$, $f_1=1$. Các nghiệm của phương trình đặc trưng là:

$r_1 = \left(\frac{1+\sqrt{5}}{2}\right); \quad r_2 = \left(\frac{1-\sqrt{5}}{2}\right)$ theo định lý 1 ta suy ra số fibonaci được cho bởi công thức sau:

$f_n = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right)^n$ với α_1, α_2 là hai hằng số. Các điều kiện đầu $f_0=0$, $f_1=1$ được dùng để xác định các hằng số α_1, α_2.

$$f_0 = \alpha_1 + \alpha_2 = 0$$

$$f_1 = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right) = 1$$

Từ hai phương trình này ta suy ra $\alpha_1 = \frac{1}{\sqrt{5}}; \alpha_2 = -\frac{1}{\sqrt{5}}$ do đó các số fibonaci được cho bằng công thức dạng hiển như sau:

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

Định lý 1 không dùng được trong trường hợp nghiệm của phương trình đặc trưng là nghiệm bội. Khi phương trình đặc trưng có nghiệm bội ta sử dụng định lý sau.

Định lý 2. Cho c_1, c_2 là các hằng số thực, $c_2 \neq 0$. Giả sử $r^2 - c_1r - c_2 = 0$ chỉ có một nghiệm r_0. Dãy $\{a_n\}$ là nghiệm của hệ thức truy hồi $a_n = c_1a_{n-1} + c_2a_{n-2}$ khi và chỉ khi $a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$ với $n = 1, 2, ..$ trong đó α_1, α_2 là những hằng số.

Ví dụ 3. Tìm nghiệm của công thức truy hồi $a_n = 6a_{n-1} - 9a_{n-2}$ với các điều kiện đầu $a_0=1$, $a_1 = 6$.

Giải: Phương trình đặc trưng $r^2 - 6r - 9 = 0$ có nghiệm kép r=3. Do đó nghiệm của hệ thức truy hồi có dạng:

$a_n = \alpha_1 3^n + \alpha_2 n3^n$ với $\alpha1$, $\alpha2$ là các hằng số nào đó. Từ các điều kiện đầu ta suy ra:

$a_0 = 1 = \alpha_1$

$a_1 = 6 = \alpha_1 3 + \alpha_2 3 \Rightarrow \alpha_1 = 1, \alpha_2 = 1$ vậy nghiệm của hệ thức truy hồi và các điều kiện đầu đã cho là:

$a_n = 3^n + n3^n$

Định lý 1, Định lý 2 không dùng được trong trường hợp nghiệm của phương trình đặc trưng là nghiệm phức. Khi phương trình đặc trưng có nghiệm phức ta sử dụng định lý sau.

Định lý 3. Cho c_1, c_2 là các hằng số thực, $c_2 \neq 0$. Giả sử $r^2 - c_1r - c_2 = 0$ có hai nghiệm phức liên hợp

$$\begin{cases} r_1 = r(\cos(\theta) + i.\sin(\theta)) \\ r_2 = r(\cos(\theta) - i.\sin(\theta)) \end{cases}$$

Khi đó, dãy $\{a_n\}$ là nghiệm của hệ thức truy hồi $a_n = c_1a_{n-1} + c_2a_{n-2}$ khi và chỉ khi $a_n = r^n(\alpha_1\cos(n\theta) + \alpha_2\sin(n\theta))$ với $n = 1, 2, \ldots$ trong đó α_1, α_2 là những hằng số.

Ví dụ. Tìm nghiệm của hệ thức truy hồi sau $a_n = 2a_{n-1} - 4a_{n-2}$, với $a_0 = 4$, $a_1 = 4$?

Lời giải.

Bước 1. Tìm nghiệm của phương trình đặc trưng:

$$r^2 + 2r + 4 = 0 \Leftrightarrow \begin{cases} r_1 = 2(\cos(\pi/3) + i\sin(\pi/3)) \\ r_2 = 2(\cos(\pi/3) - i\sin(\pi/3)) \end{cases}$$

Bước 2. Xây dựng công thức tổng quát cho a_n:

$$a_n = r^n(c_1\cos(n\theta) + c_2\sin(n\theta)) = 2^n(c_1\cos(n\pi/3) + c_2\sin(n\pi/3))$$

Bước 3. Tìm các hằng số thông qua điều kiện đầu:

$$\begin{cases} 2\left(\frac{1}{2}c_1 + \frac{\sqrt{3}}{2}c_2\right) = 4 \\ 4\left(-\frac{1}{2}c_1 + \frac{\sqrt{3}}{2}c_2\right) = 4 \end{cases} \Leftrightarrow \alpha_1 = 1; \alpha_2 = \sqrt{3}$$

Bước 4. Hoàn chỉnh công thức cho a_n.

$$a_n = 2^n(\cos(n\pi/3) + \sqrt{3}\sin(n\pi/3))$$

Bây giờ ta phát biểu kết quả tổng quát về nghiệm các hệ thức truy hồi tuyến tính thuần nhất với các hệ số hằng số.

Định lý 4. Cho $c_1, c_2, \ldots, c_k$ là các số thực. Giả sử phương trình đặc trưng

$r^k - c_1r^{k-1} - \ldots - c_k = 0$ có k nghiệm phân biệt $r_1, r_2, \ldots, r_k$. Khi đó dãy $\{a_n\}$ là nghiệm của hệ thức truy hồi $a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k}$ khi và chỉ khi $a_n = \alpha_1r_1^n + \alpha_2r_2^n + \cdots + \alpha_kr_k^n$ với $n=0,1,2,\ldots$, trong đó $\alpha_1, \alpha_2, \ldots, \alpha_k$ là các hằng số.

Ví dụ 4. Tìm nghiệm của hệ thức truy hồi $a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$ với điều kiện đầu $a_0=2$, $a_1=5$, $a_2=15$.

Giải: Đa thức đặc trưng của hệ thức truy hồi là :

$r^3 - 6r^2 + 11r - 6$ có các nghiệm là $r_1=1$, $r_2 = 2$, $r_3 = 3$. Do vậy nghiệm của hệ thức truy hồi có dạng: $a_n = \alpha_1 1^n + \alpha_2 2^n + \alpha_3 3^n$.

Để tìm các hằng số α_1, α_2, α_3 ta dựa vào những điều kiện ban đầu:

$a_0 = 2 = \alpha_1 + \alpha_2 + \alpha_3$

$a_1 = 5 = \alpha_1 + \alpha_2 2 + \alpha_3 3$

$a_2 = 15 = \alpha_1 + \alpha_2 4 + \alpha_3 9$

Giải: hệ phương trình này ta nhận được $\alpha_1 = 1$, $\alpha_2 = -1$, $\alpha_3 = 2$. Vì vậy nghiệm duy nhất của hệ thức truy hồi này và các điều đầu đã cho là dãy $\{a_n\}$ với:

$$a_n = 1 - 2^n + 2.3^n$$

2.5. Qui về các bài toán đơn giản

Một trong những phương pháp giải quyết bài toán đếm phức tạp là qui bài toán đang xét về những bài toán nhỏ hơn. Sự phân chia này được thực hiện một cách liên tiếp cho tới khi nhận được lời giải của bài toán nhỏ một cách dễ dàng. Tuy nhiên điều này không phải lúc nào cũng thực hiện được vì nó đòi hỏi một sự phân tích sâu sắc cấu hình cần đếm.

Giả sử rằng có một thuật toán phân chia bài toán cỡ n thành a bài toán nhỏ, trong đó mỗi bài toán nhỏ có cỡ n/b(để đơn giản ta giả sử n chia hết cho b); trong thực tế các bài toán nhỏ thường có cỡ là số nguyên gần nhất với n/b. Giả sử tổng các phép toán thêm vào khi thực hiện phân chia bài toán cỡ n thành các bài toán cỡ nhỏ hơn là g(n). Khi đó nếu f(n) là số các phép toán cần thiết để giải bài toán đã cho thì f thoả mãn hệ thức truy hồi sau:

$f(n) = af\left(\frac{n}{b}\right) + g(n)$; hệ thức này có tên là hệ thức chia để trị.

Ví dụ 1. Xét thuật toán nhân hai số nguyên kích cỡ 2n bít. Kỹ thuật này gọi là thuật toán nhân nhanh có dùng kỹ thuật chia để trị.

Giải: Giả sử a và b là các số nguyên có biểu diễn nhị phân là 2n bít (có thể thêm các bít 0 vào đầu để chúng có thể dài bằng nhau) .

$a = (a_{2n-1}a_{2n-2}\cdots a_1a_0)_2$ và $b = (b_{2n-1}b_{2n-2}\cdots b_1b_0)_2$

Giả sử $a = 2^nA_1 + A_0$, $b = 2^nB_1 + B_0$

trong đó

$A_1 = (a_{2n-1}a_{2n-2}\cdots a_{n-1}a_n)_2$; $A_0 = (a_{n-1}a_{n-2}\cdots a_1a_0)_2$

$B_1 = (b_{2n-1}b_{2n-2}\cdots b_{n-1}b_n)_2$; $B_0 = (b_{n-1}b_{n-2}\cdots b_1b_0)_2$

Thuật toán nhân nhanh được dựa trên đẳng thức:

$$ab = (2^{2n} + 2^n)A_1B_1 + 2^n(A_1 - A_0)(B_0 - B_1) + (2^n + 1)A_0B_0$$

Điều này chỉ ra rằng phép nhân hai số nguyên 2n bít có thể thực hiện bằng cách dùng 3 phép nhân các số nguyên n bít và các phép cộng, trừ dịch chuyển. Như vậy, nếu f(n) là tổng các phép toán nhị phân cần thiết để nhân hai số n bít thì

$$f(2n) = 3f(n) + Cn$$

Ba phép nhân các số nhị phân n bít cần *3f(n)* phép toán nhị phân. Mỗi một phép toán cộng, trừ, dịch chuyển dùng một hằng số nhân với n lần chính là *Cn*.

Ví dụ 2. Bài toán xếp khách của Lucas. Có một bàn tròn, xung quanh có 2n ghế. Cần sắp chỗ cho n cặp vợ chồng sao cho các ông ngồi sen kẽ các bà và không có hai cặp vợ chồng nào ngồi cạnh nhau. Hỏi có tất cả bao nhiêu cách xếp ?

Lời giải. Gọi số phải tìm là M_n. Xếp cho các bà trước(cứ xếp một ghế thì một ghế để trống dành cho các ông), số cách xếp cho các bà là 2n! cách. Gọi số cách xếp cho các ông ứng với một cách xếp các bà là U_n ta được số cách xếp là

$M_n = 2n! \times U_n$ .Vấn đề còn lại là tính số U_n.

Đánh số các bà (đã xếp) từ 1 đến n, đánh số các ông tương ứng với các bà (ông i là chồng bà i), sau đó đánh số các ghế trống theo nguyên tắc: ghế số i nằm giữa bà i và bà i+1 (các phép cộng được hiểu lấy modul n nghĩa là n +1 = 1). Mỗi cách xếp các ông được biểu diễn bằng một phép thế φ trên tập {1, 2, . ., n } với qui ước φ(i) = j có nghĩa là ghế i được xếp cho ông j. Theo giả thiết φ phải thoả mãn:

$\varphi(i) \neq i$ và $\varphi(i) \neq i+1$ (*)

Như vậy, U_n là số tất cả các phép thế φ thoả mãn điều kiện (*). Trong toán học gọi U_n là số phân bố.

Xét tập hợp tất cả các phép thế φ của { 1, 2, . ., n }. Trên tập này ta gọi P_i là tính chất φ(i) = i, Q_i là tính chất φ(i) = i+1. Đặt $P_{n+i} = Q_i$, theo nguyên lý bù trừ tương ứng với 2n tính chất P_i ta có:

$U_n = \overline{N} = n! - N_1 + N_2 + ...$ trong đó N_k là tổng số tất cả các phép thế thoả mãn k tính chất lấy từ 2n tính chất đang xét. Cần chú ý rằng, không thể xảy ra đồng thời thoả mãn P_i và Q_i. Do đó trong các phép lấy ra k tính chất từ 2n tính chất đang xét cần thêm vào điều kiện: P_i và Q_i hoặc P_{i+1} và Q_i không được đồng thời có mặt. Gọi số các cách này là g(2n, k) (nói riêng g(2n,k)=0 khi k>n). Với mỗi cách lấy ra k tính chất như vậy (k<=n) ta có (n-k)! phép thế thảo mãn chúng. Từ đó ta nhận được N_k = g(2n, k) (n-k)! và

$$U_n = n! - g(2n,1)(n-1)! + g(2n-2)(n-2)! - \cdots + (-1)^n g(2n,n)$$

Bây giờ chúng ta phải tính các hệ số g(2n,k), k = 1, 2, . ., n.

Xếp 2n tính chất đang xét trên còng tròn theo thứ tự P_1, Q_1, P_2, Q_2, . . , P_n, Q_n, ta thấy rằng g(2n,k) chính là số cách lấy k phần tử trong 2n phần tử xếp thành vòng tròn sao cho không có hai phần tử nào kề nhau cùng được lấy ra. Để tính g(2n,k) ta giải hai bài toán con sau:

Bài toán 1. Có bao nhiêu cách lấy ra k phần tử trong n phần tử xếp trên đường thẳng sao cho không có hai phần tử nào kề nhau cùng được lấy ra.

Lời Giải. Khi lấy k phần tử, ta còn n-k phần tử. Giữa n-k phần tử còn lại có n-k+1 khoảng trống (kể cả hai đầu). Mỗi cách lấy ra k khoảng từ các khoảng này sẽ tương ứng với một cách chọn k phần tử thoả mãn yêu cầu đã nêu. Vậy số cách chọn cần tìm là C(n-k+1, k).

Bài toán 2. Giống như bài toán 1 nhưng n phần tử xếp trên vòng tròn.

Lời giải. Cố định phần tử a được chọn chia các cách lấy thành 2 lớp

1- Các cách mà a được chọn khi đó 2 phần tử kề a sẽ không được chọn và phải lấy k-1 phần tử từ n-3 phần tử còn lại. Các phần tử này xem như kết quả của bài toán 1. Theo bài toán 1, số cách thuộc lớp kiểu này là C(n-k-1, k-1).

2- Các cách mà a không được chọn, khi đó bỏ a đi và bài toán trở về bài toán 1 chọn k phần tử từ n-1 phần tử xếp trên đường thẳng. Theo bài toán 1 số cách xếp kiểu này là C(n-k,k).

Vậy theo nguyên lý cộng số cách cần tìm là :

$$C(n-k-1,k-1)+C(n-k,k)=\frac{n}{n-k}C(n-k,k)$$

Từ kết quả của hai bài toán trên ta nhận được:

$g(2n,k)=\dfrac{2n}{2n-k}C(2n-k,k)$ và số phân bố U_n được tính bằng

$$U_n=n!-\frac{2n}{2n-1}C(2n-1,1)(n-1)!+\frac{2n}{2n-2}C(2n-2,2)(n-2)!-\cdots+(-1)^n\frac{2n}{n}C(2n,n)$$

Dưới đây là một số giá trị của U_n, một lần nữa chúng ta lại được quan sát hiện tượng bùng nổ tổ hợp.

n	2	3	4	5	6	7	8	9	10
U_n	0	1	2	13	80	579	4783	43387	439792

2.6. Phương pháp liệt kê

Việc tìm một công thức cho kết quả đếm ngay cả trong trường hợp công thức truy hồi không phải dễ dàng và lúc nào cũng thực hiện được. Cho đến nay còn nhiều bài toán đếm chưa có lời giải dưới dạng một công thức. Đối với những bài toán như vậy, người ta chỉ còn cách chỉ ra một phương pháp liệt kê, theo đó có thể đi qua được tất cả các cấu hình cần đếm. Rõ ràng bản thân phương pháp liệt kê không chỉ ra được một kết quả cụ thể nào nhưng qua đó người ta có thể lập trình cho máy tính điện tử đếm hộ.

Để minh hoạ cho phương pháp liệt kê, ta xét một cấu hình tổ hợp nổi tiếng đó là các hình chữ nhật la tinh.

Giả sử S là tập gồm n phần tử. Không mất tính tổng quát ta giả sử S = {1, 2, . ., n} Một hình chữ nhật la tinh trên S là một bảng gồm p dòng, q cột sao cho mỗi dòng của nó là một chỉnh hợp không lặp chập q của S và mỗi cột của nó là một chỉnh hợp không lặp chập p của S.

Theo định nghĩa ta có p≤n, q≤n. Đặc biệt trong trường hợp q = n, mỗi dòng của hình chữ nhật la tinh là một hoán vị của S, sao cho không có cột nào chứa hai phần tử lặp lại. Hình chữ nhật la tinh dạng này được gọi là chuẩn nếu dòng đầu của nó là hoán vị 1, 2,. ., n.

Ví dụ. Hình la tinh chuẩn trên tập S = {1, 2, 3, 4, 5, 6, 7 }

1	2	3	4	5	6	7
2	3	4	5	6	7	1
3	4	5	6	7	1	2

Gọi L(p,n) là số hình chữ nhật la tinh p x n, còn K(p,n) là số hình chữ nhật la tinh chuẩn p x n ta có:

$$L(p,n) = n!\, K(p,n)$$

Dễ dàng nhận thấy rằng, số mất D_n là số hình la tinh chuẩn 2 x n, số phân bố U_n là số hình chữ nhật la tinh chuẩn 3 x n với hai dòng đầu là:

1	2	. . .	n-1	n
2	3	. . .	n	1

Riodan J(1946) đã chứng minh công thức:

$K(3,n) = \sum_{k=0}^{m} C(n,k) D_{n-k} D_k U_{n-2k}$ trong đó m= [n/2], U_0 = 1.

Bài toán đếm với số dòng nhiều hơn đến nay vẫn chưa được giải quyết. Người ta mới chỉ đưa ra được một vài dạng tiệm cận của L(p,n).

Nếu p=q=n, thì hình chữ nhật la tinh được gọi là hình vuông la tinh. Một hình vuông la tinh cấp n được gọi là chuẩn nếu có dòng đầu và cột đầu là hoán vị 1, 2,. .n. Thí dụ một hình vuông la tinh chuẩn cấp 7.

1	2	3	4	5	6	7
2	3	4	5	6	7	1
3	4	5	6	7	1	2
4	5	6	7	1	2	3
5	6	7	1	2	3	4
6	7	1	2	3	4	5

7 1 2 3 4 5 6

Gọi l_n là số các hình vuông như thế ta có $L(n,n) = n!(n-1)!l_n$

Việc tìm một công thức cho l_n đến nay vẫn bỏ ngỏ. Tuy nhiên ta có thể nhờ máy tính liệt kê tất cả các hình vuông chuẩn cấp n. Dưới đây là một vài giá trị tính được.

N	1	2	3	4	5	6	7
l_n	1	1	1	4	56	9408	16942080

BÀI TẬP CHƯƠNG 2

1. Xâu thuận nghịch độc là một xâu khi viết theo thứ tự ngược lại cũng bằng chính nó. Hãy đếm số xâu nhị phân có độ dài n là thuận nghịch độc.
2. Cô dâu và chú rể mời bốn bạn đứng thành một hàng để chụp ảnh. Hỏi có bao nhiêu cách xếp hàng nếu:
   a) Cô dâu đứng cạnh chú rể
   b) Cô dâu không đứng cạnh chú rể
   c) Cô dâu đứng ở phía bên phải chú rể
3. Có bao nhiêu xâu nhị phân độ dài 10 có năm số 0 liền nhau hoặc năm số 1 liến nhau.
4. Có bao nhiêu xâu nhị phân độ dài bằng 8 có 3 số 0 liền nhau hoặc 4 số 1 liền nhau.
5. Mỗi sinh viên lớp toán học rời rạc hoặc giỏi toán hoặc giỏi tin học hoặc giỏi cả hai môn này. Trong lớp có bao nhiêu sinh viên nếu 38 người giỏi tin (kể cả người giỏi cả hai môn), 23 người giỏi toán (kể cả người giỏi cả hai môn), và 7 người giỏi cả hai môn.
6. Chứng tỏ rằng, trong n+1 số nguyên dương không vượt quá 2n tồn tại ít nhất một số chia hết cho một số khác.
7. Chứng minh rằng, trong dãy gồm $n^2 + 1$ số thực phân biệt đều có một dãy con dài n+1 hoặc thực sự tăng, hoặc thực sự giảm.
8. Giả sử trong một nhóm 6 người mỗi cặp hai hoặc là bạn, hoặc là thù. Chứng tỏ rằng trong nhóm có ba người là bạn của nhau hoặc là kẻ thù của nhau.
9. Hãy chỉ ra rằng, trong 102 người có chiều cao khác nhau đứng thành một hàng có thể tìm được 11 người có chiều cao tăng dần hoặc giảm dần mà không cần thay đổi thứ tự của họ trong hàng.
10. Một đô vật tay tham gia thi đấu giành chức vô địch trong 75 giờ. Mỗi giờ anh ta thi đấu ít nhất một trận, nhưng toàn bộ anh ta không thi đấu quá 125 trận. Chứng tỏ rằng, có những giờ liên tiếp anh ta thi đấu 24 trận.
11. Một nhân viên bắt đầu làm việc tại công ty từ năm 1987 với mức lương khởi điểm là 50000 đô la. Hàng năm anh ta được nhận thêm 1000 đô la và 5% lương của năm trước.
    a) Hãy thiết lập hệ thức truy hồi tính lương của nhân viên đó n năm sau năm 1987.
    b) Lương vào năm 1995 của anh ta là bao nhiêu?
    c) Hãy tìm công thức tường minh tính lương của nhân viên này n năm sau năm 1987.

12. Tìm hệ thức truy hồi cho số hoán vị của tập n phần tử. Dùng hệ thức truy hồi đó tính hoán vị của tập n phần tử.

13. Một máy bán tem tự động chỉ nhận các đồng xu một đô la và các loại tờ tiền 1 đô la và 5 đô la.

    a) Hãy tìm hệ thức truy hồi tính số cách đặt n đô la vào trong máy bán hàng, trong đó thứ tự các đồng xu, các tờ tiền là quan trọng.

    b) Tìm các điều kiện đầu.

    c) Có bao nhiêu cách đặt 10 đô la vào máy để mua một bộ tem.

14. Giải các hệ thức truy hồi với các điều đầu sau:

    a) $a_n = a_{n-1} + 6a_{n-2}$ với $n \geq 2$, $a_0 = 3$, $a_1 = 6$.

    b) $a_n = 7a_{n-1} - 6a_{n-2}$ với $n \geq 2$, $a_0 = 2$, $a_1 = 1$.

    c) $a_n = 6a_{n-1} - 8a_{n-2}$ với $n \geq 2$, $a_0 = 4$, $a_1 = 10$.

    d) $a_n = 2a_{n-1} - a_{n-2}$ với $n \geq 2$, $a_0 = 4$, $a_1 = 1$.

    e) $a_n = a_{n-2}$ với $n \geq 2$, $a_0 = 5$, $a_1 = -1$.

    f) $a_n = -6a_{n-1} - 9a_{n-2}$ với $n \geq 2$, $a_0 = 3$, $a_1 = -3$.

    g) $a_{n+2} = -4a_{n+1} + 5a_n$ với $n \geq 0$, $a_0 = 2$, $a_1 = 8$.

15. Tìm các nghiệm đặc trưng của hệ thức truy hồi tuyến tính thuần nhất

    a) $a_n = 2a_{n-1} - 2a_{n-2}$

    b) Tìm nghiệm thoả mãn hệ thức truy hồi trên và các điều kiện đầu $a_0 =1$, $a_1 =2$.

16. a) Tìm nghiệm đặc trưng của hệ thức truy hồi tuyến tính thuần nhất a_n = an-4

    b) Tìm nghiệm thoả mãn hệ thức truy hồi trên và các điều kiện đầu a_0=1, a_1=0, a_2=-1, a_3=1.

17. Một báo cáo về thị trường máy tính cá nhân cho biết có 65000 người sẽ mua modem cho máy tính của họ trong năm tới, 1 250 000 người sẽ mua ít nhất một sản phẩm phần mềm. Nếu báo cáo này nói rằng 1.450.000 người sẽ mua hoặc là modem hoặc là ít nhất một sản phẩm phần mềm thì sẽ có bao nhiêu người sẽ mua cả modem và mua ít nhất một sản phẩm phần mềm.