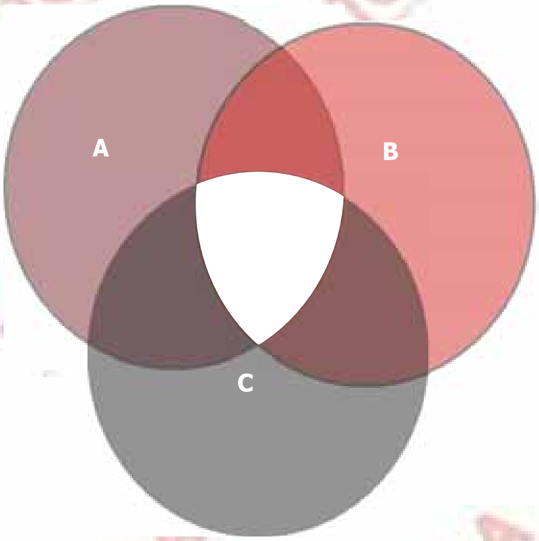
BÀI 1: TẬP HỢP VÀ ĐẠI SỐ MỆNH ĐỀ



Nội dung

- Nhắc lại một số kiến thức về tập hợp
- Các khái niệm cơ bản và những ký hiệu thường dùng
- Các phép toán tập hợp
- Tích Đề-các
- Phân hoạch
- Quan hệ

- Đại số mệnh đề
- Khái niệm về mệnh đề
- Các phép toán mệnh đề
- Vị từ và lượng từ
- Ứng dụng của đại số lôgic

Giới thiệu

Bài này trình bày những vấn đề cơ bản nhất của lý thuyết lôgic cho những người mới bắt đầu, bao gồm những phép toán lôgic và những luật lôgic cơ bản. Cuối bài có đề cập đến một số ứng dụng quan trọng của lý thuyết này.

Mục tiêu

Sau khi học bài này, các bạn có thể:

- Nắm được các khái niệm cơ bản về kiến thức tập hợp, đại số mệnh đề.
- Liệt kê được những ký hiệu thường dùng.
- Sử dụng được các phép toán tập hợp, mệnh đề.
- Liên hệ và sử dụng các kiến thức bài học trong một số ứng dụng quan trọng

Thời lượng học

- 8 tiết

TÌNH HUỐNG DẪN NHẬP

Tình huống

Trong việc giải một bài toán nào đấy, bên cạnh việc xử lý các giá trị số, ta còn gặp các tình huống phải xử lý các giá trị lôgic, chẳng hạn các giá trị của các phép so sánh (bằng nhau, khác nhau, nhỏ hơn, lớn hơn…), vì thế trong các ngôn ngữ lập trình hiện nay, ngoài các phép toán xử lý số, xử lý ký tự, người ta còn xây dựng các phép toán lôgic, nhằm xây dựng các mệnh đề phức hợp làm điều kiện trong các câu lệnh rẽ nhánh hoặc vòng lặp. Trong các câu lệnh điều khiển như rẽ nhánh hay vòng lặp, bao giờ cũng xuất hiện các điều kiện, chúng là những biểu thức lôgic mà giá trị đúng sai của chúng quyết định hoạt động của các lệnh này (vì vậy các biểu thức lôgic còn được gọi là các *biểu thức điều kiện*). Việc hiểu các luật lôgic giúp người lập trình xây dựng được các điều kiện này một cách đúng đắn và có hiệu quả.

Câu hỏi

Các phép toán lôgic, ứng dụng của chúng như thế nào?

Bài này trình bày những vấn đề cơ bản nhất của lý thuyết lôgic cho những người mới bắt đầu, bao gồm những phép toán lôgic và những luật lôgic cơ bản. Cuối bài có đề cập đến một số ứng dụng quan trọng của lý thuyết này. Phần đầu bài giảng dành cho việc nhắc lại một số kiến thức cơ bản của tập hợp (xem như đã biết) nhằm phục vụ cho việc trình bày sau này của giáo trình.

1.1. Nhắc lại một số kiến thức về tập hợp

1.1.1. Các khái niệm cơ bản và những ký hiệu thường dùng

Tập hợp là một trong những khái niệm nguyên thủy không định nghĩa. Có thể xem tập hợp được hình thành từ việc nhóm các đối tượng nào đó với nhau, mà ta gọi chúng là các phần tử của tập hợp.

Ví dụ:

Tập hợp các số nguyên, tập hợp các đường thẳng trên một mặt phẳng, tập hợp các sinh viên của một trường đại học, … Thông thường, các phần tử của một tập hợp được xác định nhờ một tính chất chung nào đấy.

Trong giáo trình này (cũng như nhiều giáo trình toán học khác):

- Tập hợp (nhiều khi gọi ngắn gọn là tập) được ký hiệu bằng các chữ cái lớn A, B, …, X, Y, …
- Những phần tử được ký hiệu bằng các chữ cái nhỏ a, b, …, x, y, …
- Để chỉ x là phần tử thuộc X ta viết $x \in X$, trái lại ta viết $x \notin X$.
- Các quan hệ $A = B$ (A bằng B), $A \neq B$ (A khác B), $A \subseteq B$ (A bao hàm trong B hay A là tập con của B) được ký hiệu và hiểu như thông lệ.

Cách mô tả tập hợp:

Có nhiều cách để mô tả một tập hợp. Đơn giản nhất là liệt kê hết các phần tử của tập hợp khi có thể, mỗi phần tử đúng một lần. Ta sẽ viết các phần tử này trong hai dấu móc, các phần tử phân cách nhau bằng dấu phẩy.

Chẳng hạn tập V gồm tất cả các nguyên âm của bảng chữ cái tiếng Anh có thể được viết như V = {a, e, i, o, u}.

Chú ý: Thứ tự liệt kê không quan trọng. Với cách liệt kê, ta có thể mô tả những tập hợp gồm những phần tử không có liên quan gì đến nhau.

Ví dụ:

A = {a, 2, Fred, while} là một tập gồm 4 phần tử: a là một chữ cái, 2 là một chữ số, Fred là một tên người còn while là một từ khóa trong ngôn ngữ C. Cũng có thể liệt kê một số phần tử đầu tiên, sau đó dùng các dấu chấm chấm (...) và kết thúc bằng phần tử cuối cùng trong trường hợp tập hợp có nhiều phần tử không thể viết hết được.

Chẳng hạn tập A gồm các số tự nhiên từ 1 đến 100 có thể viết A = {1, 2, 3, ..., 100}. Cách này cũng để mô tả một số tập hợp vô hạn (dĩ nhiên kết thúc bằng dấu chấm chấm vì không có phần tử cuối cùng) miễn là cách liệt kê đảm bảo vét cạn các phần tử.

Chẳng hạn tập các số tự nhiên bắt đầu từ 10 trở lên có thể viết {10, 11, 12, ...}.

Để tiện trình bày các tập hợp số trong các ví dụ, giáo trình này cũng dùng các ký hiệu quen thuộc để chỉ các tập hợp số cơ bản trong toán học: N – tập các số tự nhiên (bắt đầu từ 1 – nhiều tác giả xem tập này bắt đầu từ 0, nhưng sự khác nhau này không quan trọng), Z – tập các số nguyên, Q – tập các số hữu tỉ, R – tập các số thực, C – tập các số phức).

Một cách khác để mô tả một tập hợp là chỉ rõ các thuộc tính đặc trưng của các phần tử thuộc tập hợp đó, sao cho từ những thuộc tính này ta có thể xác định được một đối tượng bất kỳ có phải là phần tử của tập hợp đang xét không. Đây cũng là cách mô tả nhiều tập hợp vô hạn mà việc liệt kê các phần tử của chúng là không thể được.

Ví dụ:

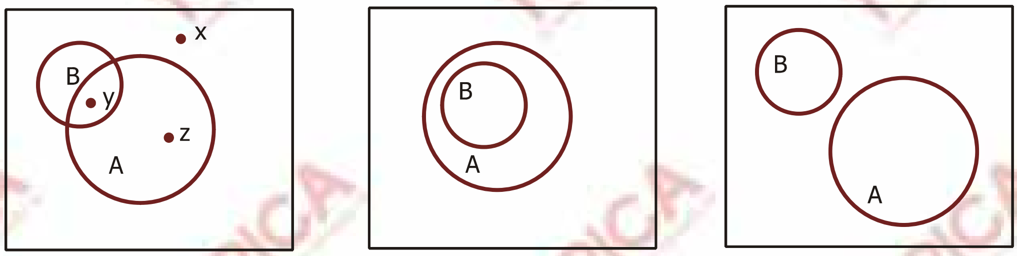
Tập X gồm hai số thực 1 và 2 ngoài cách liệt kê, còn có thể mô tả $X = \{ x \in R \mid x^2 - 3x + 2 = 0 \}$ trong khi tập Y gồm các số thực nằm trong khoảng (0, 1) được mô tả $Y = \{ x \in R \mid 0 < x < 1 \}$mà không thể dùng cách liệt kê được.

Để trực giác, các tập hợp còn được minh họa bằng hình học. Ý tưởng này được nhà toán học người Anh, John Venn đưa ra đầu tiên vào năm 1881. Trong ngữ cảnh được xét, các tập hợp được xem như là các tập con của một tập hợp bao trùm lên tất cả mà người ta gọi là *tập vũ trụ* (hay không gian), ký hiệu U.

Giản đồ Venn:

Với giản đồ Venn, U được biểu diễn bằng một hình chữ nhật, còn các tập hợp được biểu diễn như những vòng tròn nằm trong hình chữ nhật này với ý nghĩa những điểm nằm trong vòng tròn mô tả các phần tử thuộc tập tương ứng.

Giản đồ Venn cho người ta thấy rõ mối quan hệ giữa các phần tử với các tập hợp và giữa các tập hợp với nhau.



Hình trên đây là 3 giản đồ Venn.

Giản đồ thứ nhất: Mô tả hai tập hợp, A là vòng tròn lớn còn B là vòng tròn nhỏ, ba phần tử x, y, z được trình bày như các điểm trên ba vị trí cho thấy x không thuộc A cũng như không thuộc B, y vừa thuộc A vừa thuộc B, còn z thuộc A nhưng không thuộc B.

Giản đồ thứ hai: Biểu diễn B là tập con của A còn giản đồ cuối biểu diễn hai tập A, B không có phần tử chung nào.

Tập vũ trụ U là tập lớn nhất, mọi tập được xét đều là tập con của nó, trái lại, tập nhỏ nhất là tập không có phần tử nào, được gọi là *tập rỗng* và được ký hiệu bởi ϕ. Tập rỗng được coi là tập con của mọi tập hợp.

Nếu A là tập hữu hạn thì ta ký hiệu N(A) là số phần tử của A. Ngoại trừ tập rỗng có số phần tử bằng 0, các tập hữu hạn khác đều có số phần tử là một số tự nhiên nào đấy. Số phần tử của A là một tham số quan trọng trong việc đánh giá độ phức tạp của các thuật toán liên quan đến A

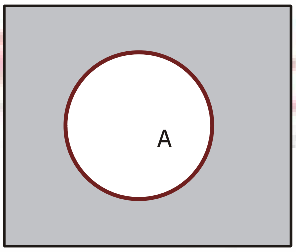
1.1.2. Các phép toán tập hợp

Các phép toán cho trên tập hợp được xây dựng trên các tập con của một tập vũ trụ U nào đấy, kết quả của phép toán cũng là một tập con của tập này. Có ba phép toán cơ bản trên tập hợp:

- **Phép bù:** là phép toán một ngôi. Ta gọi *bù* của A, ký hiệu $\overline{A}$ là tập hợp các phần tử không thuộc A. Nói riêng, bù của U là ϕ, bù của ϕ là U.

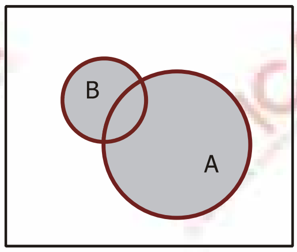
  Ví dụ: U là tập các số nguyên, A là tập các số nguyên chẵn, khi đó phần bù $\overline{A}$ của A là tập hợp các số nguyên lẻ.
- **Phép hợp:** là một phép toán hai ngôi. Giả sử A và B là hai tập hợp, ta gọi *hợp* của A với B, ký hiệu $A \cup B$ là tập hợp gồm các phần tử thuộc ít nhất một trong hai tập hợp.
  Ví dụ: A = {a, b, e}, B = {b, c, d, e}, khi đó $A \cup B$ = {a, b, c, d, e}.
- **Phép giao:** là một phép toán hai ngôi. Giả sử A và B là hai tập hợp, ta gọi *giao* của A với B, ký hiệu $A \cap B$ là tập hợp gồm các phần tử thuộc đồng thời cả hai tập hợp.
  Ví dụ: A = {a, b, e}, B = {b, c, d, e}, khi đó $A \cap B$ = {b, e}.
  Nếu A và B không có phần tử chung nào thì giao của chúng là tập rỗng, khi đó nếu A, B khác rỗng, ta cũng nói A và B là hai tập hợp *rời nhau*.

Dưới đây là các giản đồ Venn minh họa theo thứ tự kết quả của các phép toán bù, hợp, giao (phần tô đậm):



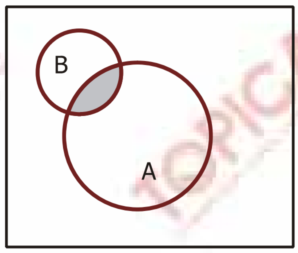
$\overline{A}$

Phép toán bù



$A \cup B$

Phép toán hợp



$A \cap B$

Phép toán giao

Có thể dễ dàng chứng minh trực tiếp bằng định nghĩa các tính chất cơ bản dưới đây của các phép toán tập hợp:

a. $\overline{\overline{A}} = A$ (luật phản xạ)

b. $A \cup (B \cup C) = (A \cup B) \cup C$ (luật kết hợp)

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

Luật kết hợp cho phép viết hợp hay giao của nhiều tập hợp mà không cần đưa dấu ngoặc vào vì vị trí dấu ngoặc đặt ở đâu cũng được.

c. $A \cup B = B \cup A$ (luật giao hoán)

$A \cap B = B \cap A$

Luật giao hoán cho phép khi viết hợp hay giao của nhiều tập hợp, ta không cần quan tâm đến thứ tự của chúng.

d. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ (luật phân bố)

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

e. $A \cup A = A$ (luật lũy đẳng)

$A \cap A = A$

f. $A \cup \phi = A$ (luật đồng nhất)

$A \cap U = A$

g. $A \cup U = U$ (luật hấp thu)

$A \cap \phi = \phi$

h. $A \cup \overline{A} = U$ (luật đầy đủ và phi mâu thuẫn)

$A \cap \overline{A} = \phi$

i. $\overline{A \cup B} = \overline{A} \cap \overline{B}$ (luật De Morgan)

$\overline{A \cap B} = \overline{A} \cup \overline{B}$

Có thể thấy một số tính chất của các phép toán tập hợp giống như các phép tính số học, chẳng hạn tính kết hợp, giao hoán của các phép hợp, giao giống như tính kết hợp, giao hoán của các phép cộng, nhân, tính phản xạ của phép bù giống như tính phản xạ của phép đối hay nghịch đảo. Tuy nhiên một số tính chất của phép toán tập hợp lại khác, chẳng hạn phép hợp, giao có tính phân bố đối với nhau nhưng trong số học chỉ có tính phân bố của phép nhân đối với phép cộng mà không có điều ngược lại, tính lũy đẳng không có đối với phép cộng và nhân các số, ...

Giống như số học, một *biểu thức tập hợp* được xây dựng từ các tập hợp thành phần và các phép toán tập hợp, kết quả trả về cũng là một tập hợp (các tập hợp đều được xét trong không gian các tập con của một tập vũ trụ U nào đấy).

Thứ tự ưu tiên trong phép toán tập hợp:

Trong biểu thức tập hợp, thứ tự ưu tiên các phép toán cũng giống như trong biểu thức số: phép bù (một ngôi) được thực hiện trước các phép hợp, giao (hai ngôi), ngoại trừ hai phép hợp, giao có mức ưu tiên như nhau (trong biểu thức số, phép nhân làm trước phép cộng).

Nếu gặp hai phép toán có cùng mức ưu tiên thì việc thi hành được thực hiện theo thứ tự từ trái sang phải. Để thay đổi thứ tự ngầm định này, cần dùng các cặp ngoặc tròn: trong ngoặc làm trước, ngoài ngoặc làm sau. Các cặp ngoặc tròn có thể lồng nhau với mức độ tùy ý.

Chứng minh đẳng thức:

Để chứng minh đẳng thức giữa hai biểu thức tập hợp, người ta thường dùng hai cách:

- Cách 1: Trực tiếp dùng định nghĩa (chứng minh một phần tử nếu thuộc vế trái thì cũng sẽ thuộc vế phải và ngược lại)
- Cách 2: Dùng các luật cơ bản đã nêu (thực chất các luật này đã được chứng minh) để biến đổi từ vế này sang vế kia.

Nhận xét:

- Cách 1: Rườm rà do phải giải quyết nhiều tình huống, nhất là gặp phải những biểu thức phức tạp nhiều toán hạng, vì thế cách này thường được dùng để chứng minh các đẳng thức đơn giản, có ít phép toán (chẳng hạn dùng để chứng minh các luật cơ bản).
- Cách 2: Ngắn gọn và sáng sủa hơn vì được dựa vào các kết quả trung gian, tuy nhiên lại khó định hướng vì có nhiều cách lựa chọn biến đổi khác nhau, nếu không khéo quá trình biến đổi dễ bị quẩn (quay lại biểu thức đã gặp).

Các phép toán được xây dựng trên không gian các tập con của một tập vũ trụ được gọi là *đại số tập hợp*. Ngoài ba phép toán cơ bản đã nêu, người ta còn xây dựng một số phép toán khác, nhằm nâng cao tính thuận tiện của việc biểu diễn tập hợp. Dưới đây trình bày thêm hai phép toán thường dùng:

- **Phép trừ:** Giả sử A và B là hai tập hợp. Ta gọi *hiệu* của A đối với B, ký hiệu $A - B$ là tập hợp các phần tử thuộc A nhưng không thuộc B.

Ví dụ: A = {a, b, e}, B = {b, c, d, e}, khi đó $A - B = \{a\}$.

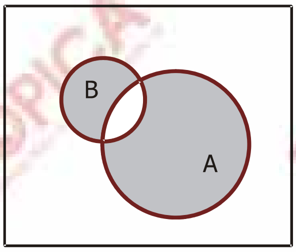
Chú ý: Phép trừ không có tính giao hoán, chẳng hạn, trong ví dụ trên $B - A = \{c, d\}$.

- **Phép cộng:** Giả sử A, B là hai tập hợp. Ta gọi *tổng* của A và B, ký hiệu $A \oplus B$ là tập hợp thuộc A hoặc thuộc B nhưng không được thuộc cả hai.

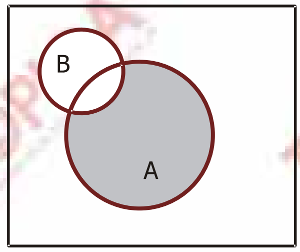
  Ví dụ: A = {a, b, e}, B = {b, c, d, e}, khi đó $A \oplus B = \{a, c, d\}$.

  Chú ý: Phép cộng có tính giao hoán, nghĩa là $A \oplus B = B \oplus A$.

  Dưới đây là các giản đồ Venn minh họa kết quả của các phép toán này (phần tô đậm):



A ⊕ B



A - B

Có thể dễ dàng chứng minh đẳng thức $A \oplus B = (A - B) \cup (B - A)$, vì thế $A \oplus B$ còn được gọi là *hiệu đối xứng* của A và B. Phép cộng vừa định nghĩa là một biến dạng của phép hợp, tương ứng với việc tuyển chọn trong đó chỉ được chọn các phần tử thỏa mãn đúng một trong hai tính chất (gọi là tuyển loại).

Sau này (phần đại số lôgic), ta thấy rằng tất cả các phép toán tập hợp được xây dựng từ các đặc trưng *thuộc/không thuộc* như các phép toán vừa nêu đều được biểu diễn qua ba phép toán cơ bản của tập hợp. Chẳng hạn, hiệu $A - B$ có thể biểu diễn qua phép bù và phép hợp nhờ biểu thức tương đương $A \cap \overline{B}$.

Chú ý:

- Trong các luật đã phát biểu của đại số tập hợp, ngoại trừ luật phản xạ (là luật không chứa các phép hợp và giao), luật nào cũng có hai đẳng thức. Trong đó, đẳng thức này nhận từ đẳng thức kia bằng cách đổi vai trò của các phép toán hợp, giao cho nhau và đổi vai trò của các tập vũ trụ và tập rỗng (nếu có) cho nhau.
- Điều này không phải ngẫu nhiên mà ta có thể chứng minh thành một luật tổng quát: nếu thay cả hai vế của một đẳng thức tập hợp theo các quy tắc đã nêu thì ta cũng được một đẳng thức tập hợp. Quy luật này được gọi là *quy luật đối ngẫu* và hai phép toán hợp, giao cũng được gọi là hai phép toán *đối ngẫu* nhau.

1.1.3. Tích Đề-các

Trong đại số tập hợp, các phép toán được xây dựng trên các tập con của cùng một không gian (tập vũ trụ) và kết quả của phép toán cũng là một tập con của không gian này. Tuy nhiên trong nhiều vấn đề, ta cần phải xây dựng những tập hợp mới từ những tập hợp thành phần trong những không gian khác nhau và những tập mới được xây dựng lại thuộc những không gian mới. Một trong những cách xây dựng như vậy là phép ghép tập hợp mà ta sẽ xét dưới đây.

Bài toán:

Giả sử A và B là hai tập hợp nào đó (có thể thuộc các không gian khác nhau, chẳng hạn A là tập hợp các thí sinh đại học còn B là tập hợp các ngành nghề đào tạo của các trường đại học). Ta xây dựng một tập hợp mới, bằng cách ghép A với B theo nghĩa mỗi phần tử của tập này là một cặp có thứ tự gồm thành phần đầu lấy từ A và thành phần sau lấy từ B.

Tập mới này được gọi là *tích Đề-các* (theo tên của nhà toán học Pháp, René Descartes) của A với B, và được ký hiệu là $A \times B$. Như vậy ta có thể viết:

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

Ví dụ:

A = {a, b, c}, B = {1, 2}, khi đó $A \times B = \{(a,1),(a,2),(b,1),(b,2),(c,1),(c,2)\}$.

Giải thích:

- A mô tả 3 thí sinh (có các tên a, b, c)
- B mô tả 2 ngành nghề (1−xã hội nhân văn, 2 − khoa học kỹ thuật)
- $A \times B$ mô tả tất cả các lựa chọn có thể của các thí sinh này (gồm 6 khả năng).

Chú ý:

- Nếu $A \neq B$ thì $A \times B \neq B \times A$ ngay cả khi A, B có cùng không gian.
- Tích Đề-các được mở rộng một cách tự nhiên cho nhiều tập hợp.

Giả sử A_1, A_2, ..., A_m là m tập nào đấy, ta gọi tích Đề-các (hay ngắn gọn − tích) của các tập này (theo thứ tự đã nêu), ký hiệu $A_1 \times A_2 \times ... \times A_m$, là tập hợp các bộ có thứ tự gồm m thành phần, trong đó thành phần thứ i lấy từ tập A_i (i = 1, 2, ..., m):

$$A_1 \times A_2 \times ... \times A_m = \{(a_1, a_2, ..., a_m) \mid a_i \in A_i, i = 1, 2, ..., m\}$$

Nói riêng, một tập A có thể ghép với chính nó nhiều lần. Tích $A \times A \times ... \times A$ (m lần) được gọi là lũy thừa m của A và được ký hiệu A^m.

Việc ghép các tập hợp cho phép mở rộng nhiều khái niệm trong toán học:

- Từ hình học 1 chiều R, trong đó mỗi điểm tương ứng với một số thực, là hình học biểu diễn các điểm trên một trục, ta mở rộng ra hình học 2 chiều, 3 chiều bằng cách ghép R với chính nó để được R^2 và R^3. Mỗi phần tử của R^2 mô tả một điểm trên mặt phẳng và mỗi phần tử của R^3 mô tả một điểm trong không gian.
- Theo cách đó ta mô tả được tổng quát hình học n chiều mà các điểm là các phần tử thuộc R^n với những khái niệm và kết quả tương tự như hình học thông thường.

Mở rộng tập số thực R:

Cũng như vậy, việc mở rộng tập số thực R thành tập số phức C được thực hiện bằng định nghĩa C = R^2 (mỗi số phức được xem như một cặp số thực), trong đó các phép toán trên số phức được xây dựng sao cho chúng vẫn được bảo toàn trên tập số thực.

Ý nghĩa:

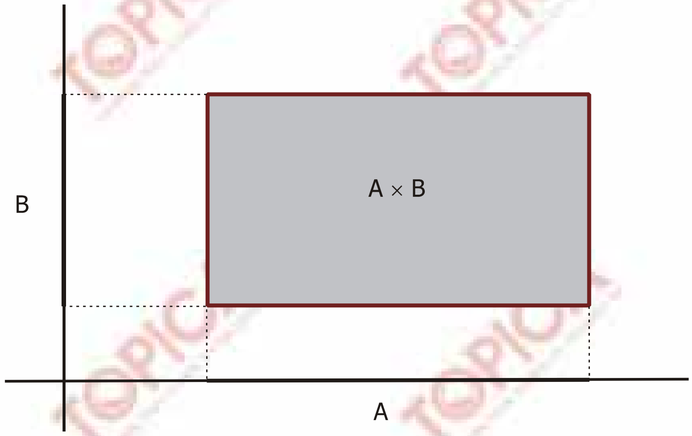
Nhiều vấn đề khó trên số thực được giải quyết dễ dàng và trọn vẹn nhờ số phức (chẳng hạn, xem định lý về tính đầy đủ các nghiệm của một đa thức trên trường số phức và hệ quả của nó là định lý phân tích một đa thức thành tích các nhân tử bậc nhất và bậc hai trên trường số thực).

Việc ghép các tập hợp còn được dùng nhiều trong khoa học quản lý, đặc biệt trong việc tổ chức dữ liệu cho máy tính để có thể tự động hóa công việc này. Nhiều hệ quản trị cơ sở dữ liệu viết cho máy tính hiện nay như DBase, Foxpro, Access, SQL, ...

đều xây dựng dựa trên cơ sở ghép tập hợp mà ta vẫn gọi là theo mô hình quan hệ. Với mô hình này, mỗi đối tượng được quản lý, xem như được ghép từ một số thuộc tính mà việc quản lý quan tâm.

Để đơn giản, ta ví dụ việc quản lý quan tâm đến 2 thuộc tính mà ta ký hiệu tương ứng A, B là miền giá trị khả dĩ của các thuộc tính này. Khi đó mỗi đối tượng sẽ được xây dựng như là một phần tử của tích $A \times B$ và tập hồ sơ quản lý được lưu trữ như là một tập con của tích $A \times B$.

Hình dưới đây minh họa tập tích $A \times B$ trên mặt phẳng (phần diện tích tô đậm mô tả tích $A \times B$, phần đoạn thẳng tô đậm trên trục hoành mô tả tập A, phần đoạn thẳng tô đậm trên trục tung mô tả tập B)



1.1.4. Phân hoạch

Trong nhiều tình huống, người ta cần chia nhỏ tập đang xét X thành nhiều tập con khác rỗng $A_1, A_2, ..., A_m$ thỏa mãn hai điều kiện sau:

(1) $X = A_1 \cup A_2 \cup ... \cup A_m$

(2) $A_i \cap A_j = \phi \quad (i \neq j)$

Giải thích:

- Điều kiện (1) được gọi là *phủ* X
- Điều kiện (2) được gọi là *rời nhau*.

Họ các tập con của X thỏa mãn các điều kiện đã nêu, được gọi là một *phân hoạch* hay một chia lớp của X, mỗi một tập con của họ này được gọi là một *lớp phân hoạch*.

Ví dụ:

X = {a, b, c, d, e, f}.

Khi đó họ các tập con $\{A_1, A_2, A_3\}$, trong đó A_1 = {a, b}, A_2 = {c}, A_3 = {d, e, f}, là một phân hoạch gồm 3 lớp của X.

Thực tế:

- Trong tổ chức xã hội, nhiều khi việc chia một tập người thành các nhóm cần phải thỏa mãn điều kiện phân hoạch.
- Chẳng hạn như: việc chia một lớp sinh viên thành nhiều tổ để tiện sinh hoạt cần phải đảm bảo các tổ này là một phân hoạch của lớp.

Một tập con thực sự và khác rỗng A của X, một cách tự nhiên, xác định một phân hoạch gồm 2 lớp A và phần bù $\overline{A}$ của nó.

Ví dụ:

Chẳng hạn tập các sinh viên của một lớp (có cả nam và nữ) được phân hoạch thành 2 lớp nam và nữ, tập các số nguyên được phân hoạch thành 2 lớp chẵn và lẻ, ... Kỹ thuật phân hoạch một tập X được dùng cho việc giải nhiều bài toán sau này.

1.1.5. Quan hệ

Khái niệm: Ngoài việc xem xét các tính chất của những phần tử trong tập hợp, người ta còn quan tâm đến những mối *quan hệ* giữa các phần tử này. Một quan hệ giữa hai phần tử (gọi là quan hệ hai ngôi) cần được xác định theo các điều kiện sao cho từ đó ta có thể kiểm tra được hai phần tử bất kỳ của tập hợp có quan hệ đó với nhau hay không.

Ký hiệu:

- **Quan hệ:** Người ta thường ký hiệu một cách hình thức một quan hệ bằng chữ cái R (Relation) với quy ước viết aRb thay cho lời nói "a có quan hệ R với b".
- **Quan hệ phủ định:** Mỗi quan hệ R, xác định một quan hệ phủ định của nó, ký hiệu $\overline{R}$, theo định nghĩa a$\overline{R}$b khi và chỉ khi a không có quan hệ R với b.

Ví dụ:

Trên tập các số nguyên, người ta có thể định nghĩa các quan hệ như sau:

- aRb khi và chỉ khi a < b (R là quan hệ "nhỏ hơn", $\overline{R}$ là quan hệ "không nhỏ hơn").
- aRb khi và chỉ khi a chia hết cho b (R là quan hệ "chia hết", $\overline{R}$ là quan hệ "không chia hết"), ...

Với R là quan hệ "nhỏ hơn" ta có 2R3, 5R7, 5$\overline{R}$2, ...

Với R là quan hệ "chia hết" ta có 4R2, 6R3, 7$\overline{R}$4, 2$\overline{R}$4, ...

Thực tế:

- **Trong hình học:** Ta cũng gặp nhiều quan hệ được xây dựng trên các đối tượng hình học như quan hệ "song song" giữa hai đường thẳng, quan hệ "vuông góc" giữa hai mặt phẳng, quan hệ "đồng dạng" giữa hai tam giác, ...
- **Trong xã hội:** nhiều mối quan hệ được hình thành như quan hệ "quen nhau" giữa hai người, quan hệ "đối tác" giữa hai tập đoàn kinh doanh, ...

Bạn đọc có thể tìm thêm nhiều ví dụ như vậy về các quan hệ trên nhiều lĩnh vực khác nhau.

Việc xác định một quan hệ R trên tập X là việc chỉ rõ điều kiện cần và đủ cho các cặp có thứ tự (a, b); a, b $\in$ X để aRb, đó cũng là điều kiện xác định một tập con nào đó của tích Đề-các X^2. Vì vậy một quan hệ hai ngôi cho trên X có thể đồng nhất với một tập con của tích X^2. Nhiều tác giả đã tiếp cận khái niệm quan hệ theo hướng này.

Trong việc nghiên cứu các quan hệ trên tập X, người ta đặc biệt quan tâm các tính chất dưới đây:

- **Tính phản xạ:** quan hệ R được gọi là có tính *phản xạ* nếu aRa với mọi a thuộc X. Chẳng hạn quan hệ "song song" trên tập các đường thẳng có tính phản xạ, quan hệ "chia hết" trên tập các số tự nhiên có tính phản xạ. Có nhiều quan hệ không có tính chất này như quan hệ "vuông góc" trên tập các đường thẳng, quan hệ "nhỏ hơn" trên tập các số nguyên.

- **Tính đối xứng:** quan hệ R được gọi là có tính *đối xứng* nếu từ aRb ta suy ra bRa với mọi a, b thuộc X. Chẳng hạn quan hệ "song song" trên tập các đường thẳng có tính đối xứng trong khi quan hệ "chia hết" trên tập các số nguyên không có tính đối xứng. Nếu quan hệ R có tính đối xứng thì người ta thường nói "a và b có quan hệ R với nhau" thay cho nói "a có quan hệ R với b". Quan hệ "quen nhau" trên tập hợp người có tính đối xứng vì khi nói " a quen b" đã bao hàm nghĩa "b quen a" (vì thế ta nói "a, b quen nhau"), tuy nhiên quan hệ "biết" không có tính đối xứng vì khi nói "a biết b" không có nghĩa là b cũng biết a. Trong các mối quan hệ xã hội, quan hệ có tính đối xứng còn được gọi là quan hệ *song phương*, còn quan hệ không có tính này được gọi là quan hệ *đơn phương*.
- **Tính bắc cầu:** quan hệ R được gọi là có tính *bắc cầu* nếu từ aRb và bRc ta suy ra aRc với mọi a, b, c thuộc X. Chẳng hạn quan hệ "song song" trên tập các đường thẳng có tính bắc cầu trong khi quan hệ "vuông góc" trên tập này lại không có tính chất đó. Trên tập hợp người, quan hệ "đồng hương" có tính bắc cầu, còn quan hệ "quen nhau" không có tính bắc cầu vì từ giả thiết a quen b, b quen c ta không thể kết luận a cũng quen c.

Một quan hệ có cả 3 tính chất đã nêu được gọi là một *quan hệ tương đương*.

Từ các ví dụ đã nêu ta có thể lấy ra một số quan hệ tương đương: quan hệ "song song" trên tập các đường thẳng, quan hệ "đồng dạng" trên tập các tam giác, quan hệ "đồng hương" trên tập hợp người, ...

Một quan hệ tương đương trên một tập có liên quan mật thiết đến một phân hoạch của tập đó. Điều này được phát biểu trong định lý dưới đây:

Định lý: Một quan hệ tương đương R trên tập X xác định một phân hoạch trên tập đó và ngược lại.

Chứng minh định lý:

Giả sử R là một quan hệ tương đương trên X, ta chia X thành các lớp theo quy tắc hai phần tử của X được xếp vào một lớp nếu chúng có quan hệ R với nhau.

Dựa vào các tính phản xạ, đối xứng, bắc cầu của R, ta chứng minh được rằng các lớp này là một phân hoạch của X. Ngược lại, từ một phân hoạch của X, ta xác định quan hệ R theo định nghĩa aRb khi và chỉ khi a thuộc cùng một lớp phân hoạch với b rồi sau đó thử lại các tính phản xạ, đối xứng, bắc cầu của quan hệ này để khẳng định nó là quan hệ tương đương.

Định lý vừa phát biểu cho thấy rằng, một quan hệ tương đương trên một tập được đồng nhất với một phân hoạch trên tập ấy, vì thế tên của một quan hệ tương đương cũng thường được dùng làm tên của phân hoạch tương ứng (hoặc ngược lại) và các lớp của phân hoạch cũng được gọi là các *lớp tương đương*.

Ví dụ:

Quan hệ "song song" trên tập các đường thẳng sẽ phân hoạch tập này thành các lớp đồng phương, trong đó hai đường thẳng cùng phương khi và chỉ khi chúng thuộc cùng một lớp.

Các khái niệm "đồng dạng" trên tập các tam giác, "đồng dạng" trên tập các ma trận vuông cùng cấp, "đẳng cấu" trên tập các cấu trúc đại số,... đều được xây dựng dựa trên quan hệ tương đương.

Chú ý:

Để xây dựng một phân hoạch từ một quan hệ, theo định lý vừa nêu, quan hệ này nhất thiết phải là quan hệ tương đương, chẳng hạn có thể chia các tổ của một lớp sinh viên theo quan hệ "đồng hương", trong khi không thể chia tổ theo quan hệ "quen nhau".

Bài toán:

Dưới đây trình bày việc xây dựng một quan hệ tương đương trên tập các số nguyên mà phân hoạch tương ứng của nó có rất nhiều ứng dụng:

- Xét tập số nguyên Z và k là một số nguyên lớn hơn 1.
- Khi chia bất kỳ phần tử a nào của Z cho k ta cũng xác định được duy nhất một phần dư là một số nguyên nào đó.
    - Ký hiệu: r(a)
    - Giá trị trong phạm vi từ 0 đến k−1.

Xác định quan hệ R trên Z như sau: aRb khi và chỉ khi r(a) = r(b). Quan hệ này được gọi là *quan hệ đồng dư theo môđun k* và viết a ≡ b (mod k) thay cho cách viết chung aRb, chẳng hạn 7 ≡ 2 (mod 5), −1 ≡ 6 (mod 7), ...

Chứng minh:

Dễ thấy quan hệ này là một quan hệ tương đương, nó phân hoạch tập số nguyên Z thành k lớp theo các giá trị khả dĩ của phần dư trong phép chia cho k:

- Lớp các số nguyên chia hết cho k (dư 0)
- Lớp các số nguyên chia k dư 1, ...
- Lớp các số nguyên chia k dư k−1, mà ta ký hiệu các lớp này tương ứng là $[0]_k$, $[1]_k$, ..., $[k-1]_k$.

Phân hoạch này cũng được gọi là *phân hoạch đồng dư theo môđun k.*

Chẳng hạn, phân hoạch đồng dư theo môđun 4 gồm 4 lớp $[0]_4$, $[1]_4$, $[2]_4$, $[3]_4$.

Số học, trong đó người ta quan tâm đến việc tính toán trên các phần dư, được gọi là *số học môđun*.

Trong số học này, ngoài các phép tính số học thông thường, người ta còn xây dựng thêm phép toán a mod k, trả về kết quả là phần dư của phép chia a cho k. Chẳng hạn 115 mod 7 = 3, −1499 mod 4 = 1, (37+112) mod 6 = 5, ... Việc thực hiện phép toán a mod k còn được gọi là rút gọn a theo môđun k.

Tính chất :

Có thể chứng minh các tính chất sau đây của phép toán này:

(1) (a+b) mod k = ((a mod k)+(b mod k)) mod k

(2) (a.b) mod k = ((a mod k).(b mod k)) mod k

Các tính chất này cho phép khi rút gọn một biểu thức số nguyên (chỉ chứa các phép cộng và nhân các số nguyên) theo môđun k, ta có thể tiến hành rút gọn các toán hạng và những kết quả trung gian trong quá trình tính toán, điều này làm cho các giá trị tham gia trong các phép toán chỉ nằm trong phạm vi từ 0 đến k−1.

Ví dụ:

Tính phần dư của phép chia giá trị a = 12.13 + 15.31.37 cho 7 ta có thể tiến hành lần lượt:

a mod 7 = (12*13 + 15*31*37) mod 7

= (5*6 + 1*3*2) mod 7

= (2 + 6) mod 7

= 1.

Số học môđun được xuất hiện từ rất sớm qua một số bài toán cổ nổi tiếng như định lý phần dư Trung hoa. Hiện nay nhiều kết quả của nó được dùng trong việc phát triển lý thuyết mật mã.

1.2. Đại số mệnh đề

1.2.1. Khái niệm về mệnh đề

Mệnh đề là một phát biểu mà nội dung của nó chỉ có một trong hai giá trị *đúng* hoặc *sai*. Những câu không có nội dung đúng, sai rõ ràng hoặc không phải câu trần thuật, mặc dù được xem là những mệnh đề trong ngôn ngữ thông thường, sẽ không phải là mệnh đề theo định nghĩa này.

Ví dụ: các phát biểu dưới đây là các mệnh đề:

Hà Nội là thủ đô của nước Việt Nam	(có giá trị đúng)
2 + 1 = 4	(có giá trị sai)
Mặt trời quay quanh trái đất	(có giá trị sai)
20 chia hết cho 5	(có giá trị đúng)

Các câu sau đây không phải là mệnh đề theo nghĩa đã nêu:

Hôm nay là thứ mấy?	(không phải câu trần thuật)
x + y = 3	(không rõ đúng, sai)

Các mệnh đề được ký hiệu bởi các chữ cái P, Q, R, ... Có nhiều cách để ký hiệu hai giá trị đúng sai. Thông thường người ta dùng các chữ cái đầu *T* và *F* trong từ tiếng Anh *True* và *False* để chỉ đúng, sai. Tuy nhiên, đơn giản hơn cả và không lệ thuộc vào ý nghĩa ngôn ngữ, ta dùng hai chữ số 1 (đúng), 0 (sai) để ký hiệu hai giá trị này. Các giá trị đúng, sai còn được gọi là các *giá trị lôgic* hay các *giá trị Boole* (theo tên nhà toán học người Anh Geogre Boole, người đầu tiên xây dựng các quy tắc cơ bản của lôgic vào năm 1854 trong cuốn "Các quy luật của tư duy").

1.2.2. Các phép toán mệnh đề

Các *phép toán mệnh đề* được xây dựng trên các toán hạng là mệnh đề và kết quả của phép toán cũng là một mệnh đề. Vì mệnh đề chỉ có các giá trị lôgic nên thực chất các phép toán mệnh đề là các phép toán kết hợp các giá trị lôgic để sinh ra một giá trị lôgic. Vì lẽ đó nên các phép toán mệnh đề còn có tên là các *phép toán lôgic* hay các *phép toán Boole*. Do tính hữu hạn của các giá trị lôgic nên đơn giản nhất là định nghĩa các phép toán này bằng một bảng, trong đó liệt kê tất cả các tình huống. Bảng này được gọi là bảng giá trị (hay bảng chân lý) của phép toán.

Cũng giống như tập hợp, người ta xây dựng ba phép toán lôgic cơ bản.

* **Phép phủ định:** là phép toán một ngôi. Ta gọi *phủ định* của mệnh đề P, ký hiệu $\overline{P}$ hay $\neg P$, là mệnh đề xác định bởi bảng sau:

P	$\overline{P}$
0	1
1	0

Nghĩa là giá trị của $\overline{P}$ là ngược (phủ định) với giá trị của P

- **Phép tuyển:** là phép toán hai ngôi. Ta gọi *tuyển* của P với Q, ký hiệu $P \vee Q$, là mệnh đề xác định bởi bảng sau:

P	Q	$P \vee Q$
0	0	0
0	1	1
1	0	1
1	1	1

Nghĩa là $P \vee Q$ có giá trị đúng khi và chỉ khi có ít nhất một trong hai toán hạng P, Q là đúng.

- **Phép hội:** là phép toán hai ngôi. Ta gọi *hội* của P với Q, ký hiệu $P \wedge Q$, là mệnh đề xác định bởi bảng sau:

P	Q	$P \wedge Q$
0	0	0
0	1	0
1	0	0
1	1	1

Nghĩa là $P \wedge Q$ có giá trị đúng khi và chỉ khi cả hai toán hạng P, Q là đúng.

Dưới đây là các tính chất cơ bản của các phép toán lôgic đã nêu mà việc chứng minh chúng có thể thử lại bằng bảng giá trị:

a. $\overline{\overline{P}} = P$ (luật phản xạ)

b. $P \vee (Q \vee R) = (P \vee Q) \vee R$ (luật kết hợp)

$P \wedge (Q \wedge R) = (P \wedge Q) \wedge R$

Luật kết hợp cho phép viết tuyển hay hội của nhiều mệnh đề mà không cần đưa dấu ngoặc vào vì vị trí dấu ngoặc đặt ở đâu cũng được.

c. $P \vee Q = Q \vee P$ (luật giao hoán)

$P \wedge Q = Q \wedge P$

Luật giao hoán cho phép khi viết tuyển hay hội của nhiều mệnh đề, ta không cần quan tâm đến thứ tự của chúng.

d. $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$ (luật phân bố)

$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$

e. $P \vee P = P$ (luật lũy đẳng)

$P \wedge P = P$

f. $P \vee 0 = P$ (luật đồng nhất)

$P \wedge 1 = P$

g. $P \vee 1 = 1$ (luật hấp thu)

$P \wedge 0 = 0$

h. $P \vee \overline{P} = 1$ (luật đầy đủ và phi mâu thuẫn)

$P \wedge \overline{P} = 0$

i. $\overline{P \vee Q} = \overline{P} \wedge \overline{Q}$ (luật De Morgan)

$\overline{P \wedge Q} = \overline{P} \vee \overline{Q}$

Tập hợp các giá trị lôgic cùng với các phép toán xác định trên nó được gọi là *đại số lôgic*, nó cũng được gọi là *đại số mệnh đề* hay *đại số Boole*.

Ngoài ba phép toán cơ bản đã nêu, người ta cũng xây dựng một số các phép toán lôgic khác, nhằm tăng thêm sự thuận tiện khi cần biểu diễn các phát biểu lôgic thông thường. Dưới đây trình bày một số phép toán như vậy.

- **Phép kéo theo:** là phép toán hai ngôi. Ta gọi P *kéo theo* Q, ký hiệu $P \to Q$, là một mệnh đề xác định bởi bảng sau:

P	Q	$P \to Q$
0	0	1
0	1	1
1	0	0
1	1	1

Nghĩa là $P \to Q$ là đúng trừ khi P đúng Q sai. Phép toán này được định nghĩa để mô tả việc chứng minh "từ P suy ra Q", trong đó ta cần chỉ ra nếu P đúng thì Q cũng đúng mà không quan tâm đến tình huống P sai.

- **Phép tương đương:** là phép toán hai ngôi. Ta gọi P *tương đương* Q, ký hiệu $P \leftrightarrow Q$, là một mệnh đề xác định bởi bảng sau:

P	Q	$P \leftrightarrow Q$
0	0	1
0	1	0
1	0	0
1	1	1

Nghĩa là $P \leftrightarrow Q$ là đúng khi và chỉ khi P và Q có giá trị như nhau (cùng đúng hoặc cùng sai). Có thể thấy ngay rằng $P \leftrightarrow Q$ được xác định giống như $(P \to Q) \wedge (Q \to P)$, nó mô tả việc chứng minh "P, Q là tương đương" giống như việc chứng minh "từ P suy ra Q và ngược lại".

- **Phép cộng:** là phép toán hai ngôi. Ta gọi *tổng* của P với Q, ký hiệu $P \oplus Q$, là một mệnh đề xác định bởi bảng sau:

P	Q	$P \oplus Q$
0	0	0
0	1	1
1	0	1
1	1	0

Phép cộng giống như phép tuyển, ngoại trừ nó cho kết quả 0 khi cả hai toán hạng cùng bằng 1, vì thế nó còn có tên là *phép tuyển loại*, nó cũng là phép cộng theo hệ đếm 2 (không nhớ), vì thế còn được gọi là *phép cộng theo môđun 2*.

Sau này ta sẽ thấy một cách tổng quát rằng mọi phép toán lôgic bất kỳ đều có thể biểu diễn qua 3 phép toán lôgic cơ bản: phủ định, tuyển, hội. Chẳng hạn với 3 phép toán vừa định nghĩa ta có các biểu diễn tương đương sau (có thể kiểm tra bằng phương pháp lập bảng trình bày dưới đây):

$$P \rightarrow Q = \overline{P} \vee Q$$

$$P \leftrightarrow Q = (\overline{P} \vee Q) \wedge (P \vee \overline{Q})$$

$$P \oplus Q = (\overline{P} \wedge Q) \vee (P \wedge \overline{Q})$$

Cũng giống như tập hợp, một *biểu thức lôgic* (còn được gọi là *biểu thức Boole*) được hình thành từ sự kết hợp các giá trị lôgic nhờ các phép toán lôgic để sinh ra một giá trị lôgic mới (vì thế các phép toán lôgic cũng được gọi là các *liên từ lôgic*). Một biểu thức lôgic có thể chứa các hằng lôgic (0, 1), các biến lôgic (là các chữ mà ta có thể thay bằng 0 hoặc 1) và các phép toán lôgic liên kết chúng. Trong biểu thức lôgic, phép toán phủ định (một ngôi) được thức hiện đầu tiên, các phép toán hai ngôi còn lại có thứ tự ưu tiên như nhau và được thực hiện từ trái sang phải. Để thay đổi thứ tự này, người ta cũng dùng các cặp ngoặc tròn theo cách thức giống như trong biểu thức tập hợp: trong ngoặc làm trước, ngoài ngoặc làm sau. Việc đưa các dấu ngoặc vào trong biểu thức (với mức độ lồng nhau tùy ý) có thể làm cho các biểu thức trở nên nặng nề, nhưng nó tránh được các hiểu lầm. Giá trị của biểu thức lôgic nói chung phụ thuộc vào giá trị của các biến lôgic có mặt trong nó, vì thế nó xác định một hàm mà các biến cũng như giá trị hàm đều lấy trên tập $\{0, 1\}$. Một hàm như thế được gọi là *hàm đại số lôgic*.

Hai biểu thức lôgic được gọi là *tương đương* nếu chúng cùng biểu diễn một hàm đại số lôgic. Để biểu thị hai biểu thức lôgic là tương đương, người ta viết dấu đẳng thức (=) giữa hai biểu thức đó (giống như biểu thức số) với ý nghĩa là giá trị của hai biểu thức này là như nhau tại mọi bộ giá trị tương ứng của các biến. Chẳng hạn, các luật lôgic cơ bản có thể xem là những luật tương đương lôgic đơn giản nhất. Một biểu thức tương đương với 1 (nghĩa là luôn đúng) được gọi là hằng đúng, một biểu thức tương đương với 0 (nghĩa là luôn sai) được gọi là hằng sai.

Để chứng minh hai biểu thức lôgic là tương đương, đơn giản nhất là lập bảng giá trị của chúng rồi so sánh. Với cách làm này, ta còn có thể kiểm chứng hai biểu thức lôgic có tương đương hay không. Để tiện so sánh, bảng giá trị của hai biểu thức được ghép thành một bảng có cấu trúc như sau: các cột được chia thành 3 nhóm, nhóm đầu (xếp bên trái) gồm các cột giá trị của các biến lôgic tham gia vào hai biểu thức, nhóm giữa dành cho các cột giá trị trung gian cần thiết cho việc tính giá trị của hai biểu thức, nhóm cuối (xếp bên phải) là hai cột giá trị tương ứng của hai biểu thức. Sau đó ta điền các giá trị lôgic vào bảng, đầu tiên là tất cả các bộ giá trị có thể của các biến, mỗi bộ trên một dòng sao cho việc liệt kê là không trùng lặp và không bỏ sót, tiếp theo là giá trị các cột trung gian được tính từ giá trị các biến và cuối cùng là giá trị các cột biểu thức được tính từ giá trị các cột trung gian. Việc kiểm chứng được tiến hành nhờ so sánh hai cột cuối: hai biểu thức là tương đương khi và chỉ khi giá trị của hai cột này là giống nhau trên từng dòng.

Ví dụ:

Để kiểm chứng sự tương đương giữa $P \to Q$ và $\overline{Q} \to \overline{P}$ ta lập bảng:

P	Q	$\overline{P}$	$\overline{Q}$	$P \to Q$	$\overline{Q} \to \overline{P}$
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	0	0
1	1	0	0	1	1

Giá trị hai cột cuối trong bảng giống nhau trên từng dòng đã khẳng định sự đúng đắn của đẳng thức $P \to Q = \overline{Q} \to \overline{P}$. Bằng bảng giá trị, ta có thể khẳng định hoặc bác bỏ sự tương đương giữa hai biểu thức. Chẳng hạn bạn đọc có thể kiểm chứng lại rằng (xem như bài tập) phép kéo theo không có tính giao hoán cũng như không có tính kết hợp, trong khi các phép toán tương đương, tuyển loại lại có các tính chất này.

Phương pháp lập bảng có ưu điểm là xác định: bất cứ hai biểu thức nào cũng có thể kiểm chứng sự tương đương bằng lập bảng và kết quả là duy nhất. Việc tính các giá trị của bảng chỉ cần dựa vào định nghĩa các phép toán liên quan. Hạn chế của phương pháp này là cồng kềnh do phải tính tất cả các khả năng. Có thể thấy rằng, số bộ giá trị khả dĩ của n biến là 2^n (nghĩa là bảng sẽ có 2^n dòng), còn số cột trung gian của bảng sẽ phát sinh theo độ phức tạp của biểu thức, vì thế cách này thường được dùng cho những biểu thức đơn giản, ít biến, chẳng hạn dùng để chứng minh các luật lôgic cơ bản.

Một phương pháp khác để chứng minh sự tương đương giữa hai biểu thức là dùng các luật tương đương cơ bản để đưa biểu thức này về biểu thức kia (giống như làm trên các biểu thức số). Chẳng hạn để chứng minh đẳng thức:

$$(P \wedge Q \wedge R) \vee (P \wedge Q \wedge \overline{R}) \vee (\overline{P} \wedge Q \wedge \overline{R}) = Q \wedge (P \vee \overline{R})$$

ta có thể biến đổi từ vế trái sang vế phải như sau:

$$(P \wedge Q \wedge R) \vee (P \wedge Q \wedge \overline{R}) \vee (\overline{P} \wedge Q \wedge \overline{R}) =$$

$$= (P \wedge Q \wedge R) \vee (P \wedge Q \wedge \overline{R}) \vee (P \wedge Q \wedge \overline{R}) \vee (\overline{P} \wedge Q \wedge \overline{R}) =$$

$$= ((P \wedge Q) \wedge (R \vee \overline{R})) \vee ((P \vee \overline{P}) \wedge (Q \wedge \overline{R})) =$$

$$= ((P \wedge Q) \wedge 1) \vee (1 \wedge (Q \wedge \overline{R})) =$$

$$= (Q \wedge P) \vee (Q \wedge \overline{R}) =$$

$$= Q \wedge (P \vee \overline{R}).$$

Ưu điểm của phương pháp này là ngắn gọn vì không phải liệt kê tất cả. Trong ví dụ vừa nêu, nếu lập bảng, số dòng sẽ là 8 và số cột sẽ là 11 (bạn đọc thử liệt kê). Tuy nhiên phương pháp này là không xác định: tại mỗi bước, có nhiều phương án biến đổi khác nhau, người thực hiện phải nhớ nhiều các luật biến đổi trung gian và phải có nhiều kinh nghiệm trong biến đổi, nếu không khéo, việc biến đổi có thể không dẫn đến kết quả. Ngoài ra, phương pháp này chỉ có thể dùng khẳng định sự tương đương chứ không thể dùng để phủ định nó.

Một ứng dụng khác của biến đổi tương đương là rút gọn một biểu thức lôgic: từ một biểu thức đã cho, ta cần tìm một biểu thức tương đương với nó nhưng "ngắn gọn" hơn.

Đây là một bài toán khó vì độ bất định của nó rất lớn nhưng kết quả của nó lại có vai trò quan trọng trong kỹ thuật tổng hợp các mạng lôgic mà ta sẽ xét trong phần ứng dụng sau này.

Ví dụ:

Để rút gọn biểu thức $(P \wedge Q) \vee (\overline{P} \wedge Q) \vee (\overline{Q} \wedge R)$, ta có thể biến đổi như sau:

$$\begin{aligned}
&(P \wedge Q) \vee (\overline{P} \wedge Q) \vee (\overline{Q} \wedge R) \\
&= ((P \vee \overline{P}) \wedge Q) \vee (\overline{Q} \wedge R) \\
&= Q \vee (\overline{Q} \wedge R) \\
&= (Q \vee \overline{Q}) \wedge (Q \vee R) \\
&= Q \vee R.
\end{aligned}$$

Việc dừng lại ở biểu thức cuối cùng cho ta một dạng "ngắn gọn" hơn, Tuy nhiên, về lý thuyết, không thể khẳng định rằng, khi dừng lại thì không thể biểu diễn "ngắn gọn" hơn được nữa (mặc dù trong ví dụ này là đúng). Hiện nay, kết quả giải bài toán này một cách tổng quát, còn rất hạn chế vì việc xem xét tất cả các khả năng (dù hữu hạn) là không khả thi. Thông thường, việc biến đổi dựa vào kinh nghiệm và dạng rút gọn chỉ là tương đối.

Hai luật biến đổi tương đương dưới đây (được thêm vào danh sách các biến đổi cơ bản) thường được dùng cho việc rút gọn:

- **Luật nuốt:** $P \vee (P \wedge Q) = P$

  $P \wedge (P \vee Q) = P$

- **Luật dán:** $P \vee (\overline{P} \wedge Q) = P \vee Q$

  $P \wedge (\overline{P} \vee Q) = P \wedge Q$

(Chứng minh được quy luật này xem như bài tập).

Sự giống nhau giữa đại số tập hợp và đại số lôgic là không phải ngẫu nhiên. Việc xác định một tập hợp dựa vào đặc trưng "thuộc/không thuộc" vào tập đó của các phần tử trong không gian đang xét, nghĩa là khi xác định một phép toán trên các toán hạng tập hợp, ta có thể dùng bảng liệt kê tất cả các tình huống "thuộc/không thuộc" của kết quả từ các tình huống "thuộc/không thuộc" của các toán hạng (bảng này gọi là bảng thuộc). Chẳng hạn phép hợp hai tập hợp A, B được xác định bởi bảng thuộc như sau:

A	B	$A \cup B$
không thuộc	không thuộc	không thuộc
không thuộc	thuộc	thuộc
thuộc	không thuộc	thuộc
thuộc	thuộc	thuộc

Bảng trên chẳng khác gì bảng giá trị của phép tuyển hai mệnh đề A, B với tương ứng 1-thuộc, 0-không thuộc. Dễ dàng thử lại tương ứng này đồng nhất các phép toán cơ bản còn lại của đại số tập hợp và đại số lôgic: giao-hội, bù-phủ định. Từ đó có thể thấy rằng đại số tập hợp và đại số lôgic là hai đại số tương đương: những luật áp dụng cho đại số tập hợp cũng được áp dụng cho đại số lôgic và ngược lại, chẳng hạn luật đối ngẫu trong tập hợp được chuyển một cách tự nhiên sang luật đối ngẫu của lôgic.

Nhờ các phép toán lôgic người ta có thể diễn đạt các phát biểu bằng lời một cách hình thức, tránh tình trạng khó hiểu, mập mờ, nhờ vậy đảm bảo tính đúng đắn trong suy diễn.

Ví dụ:

Một luật giao thông được phát biểu như sau: "bạn đang đi xe máy sẽ bị phạt tiền nếu phạm phải một trong các điều sau đây: không có bằng lái, không đội mũ bảo hiểm, vượt đèn đỏ" có thể diễn đạt bằng biểu thức lôgic $(P \wedge (Q \vee R \vee S)) \rightarrow T$.

Trong đó:

- P là mệnh đề "bạn đang đi xe máy"
- Q là mệnh đề "bạn không có bằng lái"
- R là mệnh đề "bạn không đội mũ bảo hiểm"
- S là mệnh đề "bạn vượt đèn đỏ"
- T là mệnh đề "bạn bị phạt tiền".

1.2.3. Vị từ và lượng từ

Một số câu mà giá trị đúng, sai của nó tùy thuộc vào một số ngữ cảnh sao cho với một giá trị ngữ cảnh cụ thể, nó là một mệnh đề.

Ví dụ:

Phát biểu P(x) có nội dung "x là một số nguyên lớn hơn 5" sẽ là các mệnh đề với từng giá trị x cụ thể trên tập các số nguyên.

Ta có:

- Với x=7: P(7) là một mệnh đề đúng
- Với x=3: P(3) là một mệnh đề sai.

Nhận xét: Một phát biểu kiểu như vậy được gọi là một *vị từ*, nó xác định một *hàm mệnh đề*, nhận các giá trị lôgic {0, 1} với các biến nhận giá trị trên một tập nào đó mà người ta gọi là *không gian* của biến vị từ. Trong ví dụ trên P(x) là một vị từ với biến x thuộc không gian các số nguyên.

Phép tính trên mệnh đề:

Giống như hàm số, các phép tính trên vị từ (có cùng không gian biến) được định nghĩa theo các phép tính trên mệnh đề với những tên gọi tương ứng:

- **Phủ định của vị từ P:** ký hiệu $\overline{P}$ được định nghĩa $\overline{P}(x) = \overline{P(x)}$.
- **Tuyển của vị từ P và Q**: ký hiệu $P \vee Q$ được định nghĩa $(P \vee Q)(x) = P(x) \vee Q(x)$.
- **Hội của vị từ P và Q**: ký hiệu $P \wedge Q$ được định nghĩa $(P \wedge Q)(x) = P(x) \wedge Q(x)$.

Như thế, vị từ là một sự mở rộng tự nhiên của mệnh đề. Nhiều khi ta vẫn gọi vị từ là một mệnh đề mà không sợ gây ra nhầm lẫn giống như trong số học ta vẫn gọi 5, x+3, ... là các giá trị nói chung mà không cần nói rõ 5 là giá trị số hay x + 3 là giá trị hàm.

Một vị từ có thể phụ thuộc nhiều hơn một biến, chẳng hạn phát biểu P(x, y) = "x, y là các số nguyên có tổng chia hết cho 5" là một vị từ có hai biến x, y thuộc không gian số nguyên. Với một bộ giá trị cụ thể (x, y), P(x, y) sẽ cho một giá trị đúng, sai xác định: P(3, 5) = 0, P(6, 4) = 1, ...

Ví dụ:

Giả sử x là số nguyên, y là tập con của tập các số nguyên, khi đó phát biểu P(x, y) = "x là phần tử thuộc y" là một vị từ hai biến x, y, trong đó không gian của x là các số nguyên, không gian của y là các tập con của tập các số nguyên, P(2, {1, 2, 3}) = 1, P(1, {2, 3}) = 0, ... Ví dụ này cũng cho thấy các biến của một vị từ có thể thuộc về các không gian khác nhau.

Ngoài việc cho các biến các giá trị cụ thể để vị từ trở thành mệnh đề, người ta còn có cách *lượng hóa* các biến này để biến vị từ thành mệnh đề. Có hai loại lượng hóa, mà ta gọi là *lượng từ*, thường được dùng trong các phát biểu toán học, đó là *lượng từ với mọi* và *lượng từ tồn tại*.

Giả sử P(x) là một vị từ nào đấy với x thuộc không gian X, khi đó phát biểu "với mọi x thuộc X, P(x) có giá trị đúng" được ký hiệu là $\forall x\, P(x)$. Khi đó $\forall x\, P(x)$ trở thành một mệnh đề, nó có giá trị đúng khi mọi x thuộc X đều có P(x) đúng và có giá trị sai khi ít nhất có một x thuộc X mà P(x) sai. Lượng từ với ký hiệu $\forall$ được gọi là *lượng từ với mọi* hay *lượng từ phổ dụng*. Một cách đối ngẫu, phát biểu "tồn tại x thuộc X, P(x) có giá trị đúng" được ký hiệu là $\exists\, x\ P(x)$ và $\exists\, x\ P(x)$ trở thành một mệnh đề có giá trị đúng khi có một x nào đó thuộc X làm cho P(x) đúng và có giá trị sai khi mọi x thuộc X đều có P(x) sai. Lượng từ với ký hiệu $\exists$ được gọi là *lượng từ tồn tại*.

Ví dụ:

P(x) là vị từ "x < x + 1" trên không gian các số thực, khi đó lượng từ $\forall x\, P(x)$ có giá trị đúng còn lượng từ $\exists\, x\ \overline{P(x)}$ có giá trị sai.

Từ định nghĩa của các lượng từ, dễ thấy các tương đương dưới đây:

$$\overline{\forall x\, P(x)} = \exists x\, \overline{P(x)}$$
$$\overline{\exists x\, P(x)} = \forall x\, \overline{P(x)}$$

Các tương đương này được gọi là các *luật phủ định* của lượng từ.

Đối với những vị từ phụ thuộc nhiều hơn một biến, khi lượng hóa một biến nào đó, vị từ đang xét sẽ trở nên không phụ thuộc vào biến đó. Vì thế, với những vị từ này, để trở thành mệnh đề, ta phải lượng hóa tất cả các biến của chúng, điều này dẫn đến nhiều tổ hợp khác nhau của các lượng từ. Chẳng hạn với P(x, y) là một vị từ hai biến, ta có thể có các lượng từ sau: $\forall x \forall y\, P(x,y)$, $\forall y \forall x\, P(x,y)$, $\exists x \forall y\, P(x,y)$, $\forall y \exists\, x\, P(x,y)$, ... Trong những trường hợp này, không phải lúc nào chúng cũng tương đương. Chẳng hạn có thể khẳng định luật giao hoán đối với các lượng từ cùng loại:

$$\forall x \forall y\, P(x,y) = \forall y \forall x\, P(x,y)$$
$$\exists x \exists y\, P(x,y) = \exists y \exists x\, P(x,y)$$

trong khi không thể áp dụng luật này với những lượng từ khác loại.

Ví dụ:

- Xét vị từ P(x, y) = "x > y" với x, y thuộc không gian các số nguyên. Khi đó lượng từ $\exists\, x \forall y\, P(x,y)$ có giá trị sai (vì không có một số nguyên nào lớn hơn mọi số nguyên), trong khi lượng từ $\forall y \exists\, x\, P(x,y)$ có giá trị đúng (với mọi số nguyên y, đều tìm được số nguyên x, chẳng hạn x = y + 1, để x > y).

Nhờ các lượng từ, việc diễn đạt các câu thường dùng qua đại số lôgic được mở rộng hơn, đặc biệt các phát biểu định nghĩa, định lý, ... trong toán học, nhờ vậy, có thể thực hiện các suy diễn trên chúng một cách đúng đắn.

Ví dụ:

- Câu nói "bất kỳ người nào cũng có ít nhất một người bạn tốt" có thể diễn đạt bằng lượng từ $\forall x \exists y P(x, y)$ trong đó P(x, y) là vị từ "y là bạn tốt của x".
- Câu nói "bất cứ một phụ nữ nào đã sinh đẻ đều là mẹ của một người nào đấy" có thể diễn đạt bằng mệnh đề $\forall x (F(x) \wedge P(x)) \rightarrow \exists y M(x, y)$, trong đó F(x) là vị từ "x là phụ nữ", P(x) là vị từ "x đã sinh đẻ", còn M(x, y) là vị từ "x là mẹ của y".
- Trong toán học, hàm số f(x) liên tục tại điểm x = a được định nghĩa như sau "với mọi số dương ε, đều tìm được số dương δ sao cho khi $|x - a| < \delta$ ta có $|f(x) - f(a)| < \varepsilon$". Định nghĩa này có thể diễn đạt bằng mệnh đề

  $$\forall \varepsilon \exists \delta \forall x ((|x - a| < \delta) \rightarrow (|f(x) - f(a)| < \varepsilon)).$$

  Phủ định của mệnh đề này là (nhận được từ luật phủ định)

  $$\exists \varepsilon \forall \delta \exists x ((|x - a| < \delta) \wedge (|f(x) - f(a)| \geq \varepsilon)),$$

  kết quả này cho phép định hướng việc chứng minh hàm f(x) không liên tục tại a.

1.2.4. Ứng dụng của đại số lôgic

1.2.4.1. Kỹ thuật chứng minh

Lý thuyết lôgic xây dựng một cơ sở toán học cho việc lập luận, suy diễn. Nhờ nó ta có thể kiểm chứng tính đúng đắn của các lập luận trong suy diễn hàng ngày cũng như trong toán học.

Dưới đây ta xem xét lại một số phương pháp chứng minh định lý trong toán học và kiểm chứng tính đúng đắn của các phương pháp này dựa trên các kết quả của lý thuyết lôgic.

Một định lý thường được phát biểu dưới dạng "Nếu có P thì có Q" hoặc "từ P suy ra Q", trong đó P là giả thiết còn Q là kết luận. Để chứng minh phát biểu này là đúng, ta xuất phát từ giá trị đúng của P và cố gắng chứng tỏ khi đó Q cũng có giá trị đúng. Theo định nghĩa phép toán kéo theo, điều này là tương đương với việc chứng minh biểu thức lôgic $P \rightarrow Q$ là hằng đúng. Những định lý dạng này thường có hai cách chứng minh.

- **Chứng minh trực tiếp:** Xuất phát từ P đúng ta dẫn về một hệ quả H dễ thấy hơn, nghĩa là chứng minh $P \rightarrow H$ là hằng đúng, sau đó nếu ta chứng minh được H dẫn về Q, tức là $H \rightarrow Q$ là hằng đúng, thì định lý được chứng minh. Cơ sở của phép suy luận này dựa vào biểu thức $((P \rightarrow H) \wedge (H \rightarrow Q)) \rightarrow (P \rightarrow Q)$ là hằng đúng, được gọi là *luật bắc cầu* hay *tam đoạn luận*, mà việc chứng minh nó được thực hiện dễ dàng trong lôgic (chẳng hạn, bằng cách lập bảng giá trị).

  Ví dụ:

  Khẳng định "nếu n là số nguyên lẻ thì n^2 cũng là số nguyên lẻ" được chứng minh như sau: "n là số nguyên lẻ" $\Rightarrow$ "n được viết dưới dạng 2m + 1 với m nguyên nào đó" $\Rightarrow$ "$n^2 = (2m + 1)^2 = 4m^2 + 4m + 1 = 2(2m^2 + 2m) + 1$" $\Rightarrow$ "n^2 là số nguyên lẻ" (ký hiệu $\Rightarrow$ có nghĩa là "dẫn về" hay "suy ra").

Chú ý:

Giả thiết P có thể thay bằng tương đương $1 \wedge P$, nghĩa là ta có thể thêm vào giả thiết bất cứ điều gì đã được khẳng định là đúng, điều này có nghĩa khi chứng minh định lý, càng có nhiều kiến thức càng dễ thực hiện. Nhiều định lý không có phần giả thiết P mà chỉ có phần kết luận Q, tuy nhiên có thể xem đây là trường hợp riêng vì Q tương đương với $1 \to Q$.

- **Chứng minh gián tiếp:** Khi việc chứng minh trực tiếp từ P suy ra Q gặp nhiều khó khăn, người ta thường dùng *phương pháp phản chứng*: xuất phát đồng thời có P (P đúng) và không có Q (Q sai), ta dẫn được về một điều sai, khi đó định lý được chứng minh. Cơ sở của phép chứng minh này là sự tương đương giữa biểu thức $(P \wedge \overline{Q}) \to 0$ và biểu thức $P \to Q$, được gọi là *luật phản chứng*.

Ví dụ:

Khẳng định "nếu n^2 là số nguyên lẻ thì n cũng là số nguyên lẻ" (mệnh đề đảo của mệnh đề trong ví dụ trước) được chứng minh bằng phản chứng như sau:

Giả sử n^2 là số nguyên lẻ đồng thời n là số nguyên chẵn, khi đó ta có các suy diễn "n là số nguyên chẵn" $\Rightarrow$ "n được viết dưới dạng 2m với m nguyên nào đó" $\Rightarrow$ "$n^2 = (2m)^2 = 2(2m^2)$" $\Rightarrow$ "n^2 là số nguyên chẵn". Việc đồng thời có hai mệnh đề phủ định nhau "n^2 là số nguyên lẻ" và "n^2 là số nguyên chẵn" là một điều sai.

Một số phát biểu định lý có dạng "nếu P thì Q và ngược lại", hay "có P khi và chỉ khi có Q", hay "cần và đủ để có P là có Q", được chứng minh trên cơ sở $P \leftrightarrow Q$ là tương đương với $(P \to Q) \wedge (Q \to P)$. Chẳng hạn, chứng minh trong hai ví dụ trên khẳng định "bình phương của một số nguyên là lẻ khi và chỉ khi số nguyên đó là lẻ" (hay "bình phương của một số nguyên là chẵn khi và chỉ khi số nguyên đó là chẵn").

Một số định lý phát biểu sự tương đương của nhiều mệnh đề P_1, P_2, ..., P_k. Khi đó để chứng minh, người ta dùng kỹ thuật "chứng minh theo vòng tròn": từ P_1 dẫn về P_2, từ P_2 dẫn về P_3, ..., từ P_{k-1} dẫn về P_k, và cuối cùng, từ P_k dẫn về P_1. Cơ sở của phép chứng minh này dựa vào luật bắc cầu nhiều lần.

Nhiều định lý được phát biểu như là các mệnh đề có chứa lượng từ. Dưới đây, ta xét một vài cách chứng minh thường gặp.

Giả sử phải chứng minh mệnh đề $\forall x\, P(x)$ có giá trị đúng. Khi đó ta phải chỉ ra rằng P(x) đúng với mọi x thuộc không gian của nó. Nếu không gian này là hữu hạn gồm các giá trị x_1, x_2, ..., x_k thì việc chứng minh tương đương với việc chỉ ra đồng thời các mệnh đề $P(x_1)$, $P(x_2)$, ..., $P(x_k)$ có giá trị đúng. Nếu không gian của x là vô hạn mà không có một lập luận chung nào chứng tỏ P(x) đúng cho mọi x thì người ta cố gắng tìm một phân hoạch gồm hữu hạn các lớp của không gian này sao cho có thể lần lượt chứng tỏ $\forall x\, P(x)$ là đúng trong mỗi lớp được xét. Kỹ thuật chứng minh này được gọi là phương pháp *chứng minh theo trường hợp*. Trên tập các số nguyên, phân hoạch đồng dư theo môđun thường hay được dùng với mục đích như vậy.

Ví dụ:

Chứng minh "với mọi số nguyên n ta đều có $n^5 - n$ chia hết cho 5" được tiến hành như sau:

Đầu tiên, ta có phân tích $n^5 - n = n(n-1)(n+1)(n^2+1)$, sau đó xét phân hoạch tập số nguyên thành các lớp đồng dư theo môđun 5. Phân hoạch này gồm 5 lớp, ký hiệu K_i là lớp các số nguyên chia cho 5 dư i (i = 0, 1, 2, 3, 4). Nếu n thuộc K_0 thì n chia hết cho 5, nếu n thuộc K_1 thì n − 1 chia hết cho 5, nếu n thuộc K_2 hoặc thuộc K_3 thì $n^2 + 1$ chia hết cho 5, nếu n thuộc K_4 thì n + 1 chia hết cho 5. Như vậy mệnh đề là đúng trên mọi lớp của phân hoạch đang xét, nghĩa là nó đúng trên toàn bộ không gian các số nguyên.

Chú ý:

Dù có rất nhiều các giá trị x (thậm chí vô hạn) để P(x) đúng cũng chưa thể khẳng định mệnh đề $\forall x\, P(x)$ là đúng, chừng nào các giá trị này chưa vét cạn không gian của x. Ví dụ mệnh đề "$x^2 - x + 41$ là một số nguyên tố" là đúng với mọi số tự nhiên x = 1, 2, ..., 40 nhưng là sai với giá trị x = 41.

Trong nhiều trường hợp mệnh đề $\forall x\, P(x)$ được phát biểu trên không gian các số tự nhiên. Dựa vào đặc điểm "mọi tập con khác rỗng của tập các số tự nhiên đều có phần tử nhỏ nhất" (*tính được sắp tốt*), người ta chứng minh được luật quy nạp sau đây:

$$(P(1) \wedge \forall n(P(n) \to P(n+1))) \to \forall n P(n) \text{ là hằng đúng.}$$

Từ đó nhận được cách chứng minh mệnh đề $\forall x\, P(x)$ trên không gian các số tự nhiên là đúng bằng *phương pháp quy nạp* như sau:

Đầu tiên chứng tỏ P(1) đúng (*bước cơ sở*).

Sau đó chứng minh $P(n) \to P(n+1)$ đúng với mọi n tự nhiên (*bước quy nạp*).

Mệnh đề P(n) trong phép kéo theo $P(n) \to P(n+1)$ được gọi là *giả thiết quy nạp*.

Ví dụ:

Chứng minh đẳng thức $1 + 3 + ... + 2n - 1 = n^2$ với mọi giá trị tự nhiên n.

Ta có $1 = 1^2$ (bước cơ sở).

Giả sử $1 + 3 + ... + (2n - 1) = n^2$ (giả thiết quy nạp), khi đó ta nhận được:

$1 + 3 + ... + (2n - 1) + (2n + 1) = n^2 + 2n + 1 = (n + 1)^2$ (bước quy nạp). Đẳng thức được chứng minh.

Bây giờ ta xét các cách chứng minh mệnh đề $\exists x\, P(x)$ là đúng. Có hai cách chứng minh, cách thứ nhất, xác định cụ thể giá trị x (bằng công thức hoặc thuật toán) sao cho P(x) là đúng, được gọi là cách *chứng minh kiến thiết*, cách thứ hai, dùng phản chứng, chứng minh rằng phủ định của $\exists x\, P(x)$ dẫn đến sai, được gọi là cách *chứng minh không kiến thiết*.

Ví dụ:

Mệnh đề "với mọi số tự nhiên n, tồn tại n số tự nhiên liên tiếp là hợp số" được chứng minh kiến thiết bằng cách chỉ rõ giá trị của n hợp số này như sau:

Gọi x là số tự nhiên xác định bởi công thức x = (n + 1)! + 1, khi đó n số tự nhiên liên tiếp x + i (i = 1, 2, ..., n) đều có ước tương ứng là i + 1 và như vậy chúng là các hợp số.

Việc chỉ ra x một cách cụ thể không phải là dễ, nhiều tình huống ta chỉ có thể khẳng định sự tồn tại của x mà không chỉ rõ được cách xây dựng x.

Ví dụ:

Định lý tồn tại số nguyên tố dưới đây được chứng minh bằng phản chứng (không kiến thiết): "với mọi số tự nhiên n, đều tìm được một số nguyên tố lớn hơn n" (tính chất vô hạn của số nguyên tố).

Chứng minh.

Giả sử trái lại, tất cả tất cả các số nguyên tố đều nhỏ hơn (hay bằng) n, từ đó nhận được chỉ có một số hữu hạn tất cả các số nguyên tố 2, 3, ..., p (p là số nguyên tố lớn nhất). Xét số tự nhiên m = 2.3...p + 1. Hiển nhiên m > p nên nó là hợp số, vì thế nó phải chia hết cho một số nguyên tố nào đó, tuy nhiên, theo công thức của m, khi chia m cho bất cứ số nguyên tố nào cũng đều dư 1. Mâu thuẫn này chứng minh định lý.

Chú ý:

Lượng từ "với mọi" và lượng từ "tồn tại" là phủ định của nhau, vì thế chúng có thể chuyển đổi phát biểu cho nhau. Chẳng hạn phát biểu $\exists x\, P(x)$ là đúng (sai) có thể thay bằng phát biểu $\forall x\, \overline{P(x)}$ là sai (đúng), ... Giá trị x để P(x) đúng được gọi là ví dụ, còn giá trị x để P(x) sai được gọi là phản ví dụ. Nhiều giả thuyết dạng $\forall x\, P(x)$ tồn tại trong một thời gian dài sau đó kết thúc với kết quả phủ định bằng một phản thí dụ.

Lịch sử toán học cho thấy việc xác định giá trị đúng hay sai của các mệnh đề dạng $\exists x\, P(x)$ hay $\forall x\, P(x)$ là không đơn giản. Chẳng hạn định lý nổi tiếng của Fermat dưới đây: "với mọi số nguyên dương m lớn hơn 2, không tồn tại các số nguyên x, y, z thỏa mãn phương trình $z^m = x^m + y^m$" mãi đến gần đây mới có chứng minh (mà việc chứng minh rất phức tạp) còn phỏng đoán của Goldbach "mọi số nguyên chẵn lớn hơn 2 đều được viết dưới dạng tổng của hai số nguyên tố" cho đến nay vẫn chưa được khẳng định, mặc dù nó được khẳng định là đúng với tất cả những số nguyên được thử. Về những ứng dụng của lý thuyết lôgic trong chứng minh các bạn có thể tham khảo [3].

1.2.4.2. Kỹ thuật lập trình

Trong việc giải một bài toán nào đấy, bên cạnh việc xử lý các giá trị số, ta còn gặp các tình huống phải xử lý các giá trị lôgic, chẳng hạn các giá trị của các phép so sánh (bằng nhau, khác nhau, nhỏ hơn, lớn hơn), vì thế trong các ngôn ngữ lập trình hiện nay, ngoài các phép toán xử lý số, xử lý ký tự, người ta còn xây dựng các phép toán lôgic, nhằm xây dựng các mệnh đề phức hợp làm điều kiện trong các câu lệnh rẽ nhánh hoặc vòng lặp. Trong các câu lệnh điều khiển như rẽ nhánh hay vòng lặp, bao giờ cũng xuất hiện các điều kiện, chúng là những biểu thức lôgic mà giá trị đúng sai của chúng quyết định hoạt động của các lệnh này (vì vậy các biểu thức lôgic còn được gọi là các *biểu thức điều kiện*). Việc hiểu các luật lôgic giúp người lập trình xây dựng được các điều kiện này một cách đúng đắn và có hiệu quả.

Ví dụ:

Ngôn ngữ Pascal xây dựng riêng kiểu lôgic với tên kiểu là Boolean, giá trị đúng được đặt tên là TRUE và giá trị sai được đặt tên là FALSE. Các phép toán lôgic cơ bản đều được xây dựng và được đặt tên theo các từ khóa tiếng Anh: NOT (phủ định), OR (tuyển), AND (hội), XOR (tuyển loại). Trong ngôn ngữ C, kiểu lôgic không được xây dựng riêng mà được dùng như những giá trị số: 0 (sai), khác 0 (đúng), các phép toán lôgic được ký hiệu ! (phủ định), || (tuyển), && (hội).

Trong biểu thức điều kiện, thứ tự thực hiện các phép toán là quan trọng. Các ngôn ngữ lập trình đều tuân thủ thứ tự phép toán một ngôi (phủ định) làm trước, các phép toán hai ngôi (tuyển, hội) làm sau, tuy nhiên đối với các phép toán hai ngôi, thứ tự thực hiện có thể khác nhau đôi chút trên một vài ngôn ngữ. Để tránh nhầm lẫn tốt hơn cả là dùng các cặp ngoặc tròn để quy định thứ tự này một cách chủ động, mặc dù cách viết này gây nên một chút rườm rà.

Các luật thay thế tương đương được áp dụng cho các biểu thức điều kiện, nhằm các điều kiện này được viết gọn gàng và sáng sủa hơn.

Ví dụ : Điều kiện NOT ((x >= 3) OR (x <= 1)) có thể thay thế bằng điều kiện tương đương (x > 1) AND (x < 3). Một số trường hợp lệnh IF lồng nhau có thể thay bằng một lệnh IF.

Chẳng hạn lệnh: IF (a > 1) THEN
IF (b < 5) THEN ...

có thể thay bằng IF (a > 1) AND (b < 5) THEN ...

Việc kiểm tra giá trị của các biểu thức điều kiện trong các câu lệnh điều khiển là một trong những thao tác không thể thiếu khi kiểm chứng chương trình. Nhờ kiểm tra, ta có thể phát hiện những sai lầm trong những câu lệnh IF, WHILE hoặc bỏ bớt chúng. Ví dụ trong câu lệnh: IF P(x) THEN ... (không có ELSE)

nếu điều kiện P(x) là luôn đúng với mọi x thì lệnh này được thay bằng thân của nó (không còn IF), còn P(x) luôn sai với mọi x thì có thể bỏ câu lệnh này đi. Cũng vậy, trong vòng lặp WHILE, nếu giá trị điều kiện của nó là sai trước khi vào vòng lặp thì có thể bỏ vòng lặp này còn nếu giá trị điều kiện là đúng và không thay đổi trong mọi lần lặp thì vòng lặp trở nên không kết thúc (trong trường hợp này cần phải sửa lại chương trình).

Các kỹ thuật rút gọn biểu thức lôgic là có ích khi ta cần xây dựng các điều kiện sao cho hiệu quả. Chẳng hạn một điều kiện có nội dung:

(x < 1) AND (y > 5) AND ((x > 0) OR (y < 6) OR (z > 0))

trong đó x, y, z là các biến nguyên nào đó, có thể thay bằng điều kiện tương đương ngắn gọn hơn nhờ các luật biến đổi trong lôgic:

(x < 1) AND (y > 5) AND (z > 0).

Đối với các toán hạng là số nguyên, ngoài các phép toán số học, các ngôn ngữ lập trình còn xây dựng các toán tử bit, tác động trên từng bit 0, 1 trong biểu diễn các số nguyên theo các quy tắc của phép toán lôgic. Chẳng hạn, ngôn ngữ Pascal xây dựng các toán tử bit cũng với tên gọi NOT, AND, OR, XOR nhưng làm việc trên các toán hạng nguyên.

Ví dụ:

Với các số nguyên không dấu 8 bit (1 byte), ta có:

NOT 19 = NOT (00010011) = 11101100 = 236
19 AND 21 = (00010011) AND (00010101) = 00010001 = 17
19 OR 21 = (00010011) OR (00010101) = 00010111 = 23
19 XOR 21 = (00010011) XOR (00010101) = 00000110 = 6

Trong ngôn ngữ C, do kiểu lôgic được dùng như kiểu nguyên nên các toán tử bit được ký hiệu khác với toán tử lôgic: ~ (phủ định), | (tuyển), & (hội), ^ (tuyển loại).

Các phép toán bit được dùng trong việc xử lý các dữ liệu phức hợp như tập hợp, ảnh, ...

1.2.4.3. Kỹ thuật tổng hợp mạch lôgic

Một trong những ứng dụng quan trọng của đại số lôgic trong kỹ thuật là tổng hợp các mạch lôgic. Nhờ các kết quả biểu diễn các hàm lôgic, ta có phương pháp tổng hợp một mạch lôgic bất kỳ, ngược lại, những vấn đề nảy sinh trong quá trình tổng hợp mạch lôgic lại đặt ra cho lý thuyết lôgic những bài toán cần giải quyết.

Như đã trình bày trong phần các phép toán lôgic, một biểu thức lôgic có chứa các biến lôgic x, y, z, ... sẽ nhận một giá trị lôgic tùy thuộc vào giá trị của các biến này, nói khác đi nó xác định một hàm có các biến thuộc tập {0, 1} và giá trị hàm cũng thuộc tập {0, 1}. Một hàm như vậy được gọi là một hàm đại *số lôgic*.

Bây giờ ta đặt vấn đề ngược lại, cho trước một hàm đại số lôgic bất kỳ, liệu có tồn tại một biểu thức nào thực hiện hàm đó hay không (nghĩa là bảng giá trị của biểu thức trùng với bảng giá trị của hàm)?

Câu trả lời là khẳng định, hơn nữa, còn khẳng định rằng trong biểu thức chỉ cần chứa ba phép toán cơ bản của lôgic là phủ định, hội, tuyển.

Định lý. Mọi hàm đại số lôgic đều được thực hiện bằng một biểu thức lôgic trong đó chỉ chứa các phép toán phủ định, hội, tuyển.

Chứng minh:

Ta sẽ chứng minh định lý này bằng cách chỉ ra cách xác định một biểu thức lôgic thỏa mãn các yêu cầu từ bảng giá trị của hàm đã cho.

Để bớt các dấu ngoặc và viết như một biểu thức số, ta quy ước thêm rằng trong biểu thức lôgic, phép hội được thực hiện trước mọi phép toán hai ngôi khác (chẳng hạn phép tuyển) và viết xy thay cho $x \wedge y$.

Ví dụ: Biểu thức $(x \wedge \overline{y}) \vee (\overline{x} \wedge y)$ được viết lại một cách gọn hơn là $x\overline{y} \vee \overline{x}y$.

Bây giờ giả sử có hàm đại số lôgic xác định bằng một bảng giá trị nào đó. Ví dụ hàm f(x, y, z) xác định bởi bảng sau:

	x	y	z	f(x, y, z)	
	0	0	0	0	
$\overline{x}\,\overline{y}z$	0	0	1	1	*
	0	1	0	0	
$\overline{x}yz$	0	1	1	1	*
	1	0	0	0	
	1	0	1	0	
$xy\overline{z}$	1	1	0	1	*
xyz	1	1	1	1	*

Đầu tiên ta đánh dấu những dòng của bảng mà hàm có giá trị 1 (trong bảng trên là các dòng có dấu *). Với mỗi dòng này, ta lập hội gồm tất cả các biến, trong đó mỗi biến hoặc để nguyên nếu giá trị tương ứng của nó bằng 1 hoặc có dấu phủ định nếu giá trị tương ứng của nó bằng 0 (trong bảng trên, các hội này là $\overline{x}\,\overline{y}z$, $\overline{x}yz$, $xy\overline{z}$, xyz được ghi bên trái bảng), cuối cùng biểu thức cần tìm là tuyển của tất cả các hội này. Trong ví dụ trên, ta nhận được:

$$f(x, y, z) = \overline{x}\,\overline{y}z \vee \overline{x}yz \vee xy\overline{z} \vee xyz$$

Biểu thức lôgic nhận được từ cách làm trên, được gọi là *dạng tuyển chuẩn tắc đầy đủ* của hàm đại số lôgic đã cho.

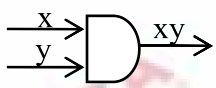
Để chứng minh biểu thức này thực hiện hàm đã cho, ta cần chứng tỏ tại những bộ giá trị của các biến làm cho hàm bằng 1 thì tại đó biểu thức cũng bằng 1 và ngược lại. Thật vậy tại những dòng của bảng mà hàm có giá trị 1, hội tương ứng được xây dựng theo cách làm đã nêu, cũng bằng 1, từ đó kéo theo biểu thức (gồm tuyển của các hội này) cũng bằng 1. Ngược lại, tại những bộ giá trị của biến làm cho biểu thức bằng 1, sẽ có ít nhất một hội trong biểu thức này bằng 1, nghĩa là tại dòng tương ứng của hội này trong bảng, hàm có giá trị 1.

Từ kết quả vừa chứng minh, trong kỹ thuật, người ta thiết kế 3 mạch cơ bản, thực hiện các phép phủ định, hội, tuyển và từ các mạch cơ bản này ta có thể tổng hợp một hàm đại số lôgic bất kỳ. Một thiết bị xử lý thông tin có thể được mô phỏng như một mạch lôgic thực hiện một họ hàm đại số lôgic nào đó. Như vậy từ các mạch cơ bản, ta có thể tổng hợp một mạch lôgic bất kỳ.

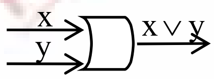
Các mạch lôgic cơ bản thường được gọi là các *cổng lôgic* (gọi tắt là *cổng*). Có 3 loại cổng lôgic được định nghĩa như sau:

Cổng phủ định (cổng NOT): là cổng có một đầu vào và một đầu ra thực hiện hàm phủ định $f(x)=\overline{x}$, được biểu diễn bằng hình vẽ:

Cổng hội (cổng AND): là cổng có hai đầu vào và một đầu ra thực hiện hàm hội $f(x,y)=xy$, được biểu diễn bằng hình vẽ:



Cổng tuyển (cổng OR): là cổng có hai đầu vào và một đầu ra thực hiện hàm tuyển $f(x,y)=x\vee y$, được biểu diễn bằng hình vẽ:



Ta có thể ghép các cổng hội (tuyển) hai đầu vào để được các cổng (hội) tuyển nhiều đầu vào cho phép thực hiện các hàm hội (tuyển) của nhiều biến, vì thế để đơn giản cách trình bày, ta có thể coi các cổng hội, tuyển là các cổng có nhiều đầu vào.

Việc tổng hợp một hàm lôgic nhờ các cổng cơ bản đã nêu có thể tiến hành theo các bước sau:

- Đầu tiên, căn cứ vào yêu cầu vào/ra của hàm, lập bảng giá trị của nó
- Sau đó tìm dạng tuyển chuẩn tắc đầy đủ của hàm đã cho theo thuật toán đã nêu
- Cuối cùng lắp ghép các cổng cơ bản theo biểu thức tìm được

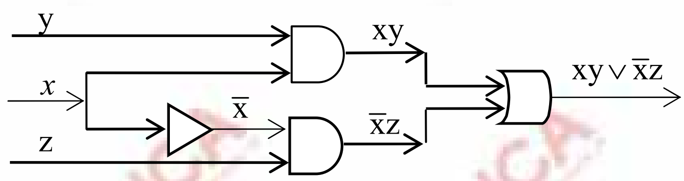
Chú ý: Có nhiều biểu thức lôgic cùng thực hiện một hàm (các biểu thức tương đương), vì thế người ta cố gắng tìm một biểu thức thực hiện hàm đã cho càng gọn càng tốt để tiết kiệm chi phí cho mạch. Việc này thường được thực hiện bằng cách rút gọn dạng tuyển chuẩn tắc đầy đủ của hàm. Ví dụ đối với hàm đã cho ở bảng trên, ta tìm được dạng chuẩn tắc đầy đủ của nó là:

$$f(x,y,z)=\overline{x}\,\overline{y}z\vee\overline{x}yz\vee xy\overline{z}\vee xyz$$

Sau khi rút gọn (bạn đọc tự tìm hiểu việc rút gọn trong từng bước), ta được biểu thức thực hiện hàm đã cho một cách gọn hơn:

$$f(x,y,z) = xy \vee \overline{x}z$$

Mạch lôgic thực hiện hàm này được trình bày trong hình vẽ dưới đây:



Ví dụ:

Trong mạng điện dân dụng: xây dựng một mạch lôgic thực hiện hệ thống đèn cầu thang. Đèn cầu thang khác với những đèn thông thường là nó được điều khiển bởi hai công tắc: khi thay đổi trạng thái của một trong hai công tắc đèn phải thay đổi trạng thái (từ bật sang tắt hoặc ngược lại).

Có thể xem trạng thái của đèn là một hàm đại số lôgic của hai biến x, y, trong đó giá trị của hàm mô tả hai trạng thái tắt (0), bật (1) của đèn và giá trị của các biến mô tả hai trạng thái (ký hiệu 0, 1) của các công tắc.

Giả sử trạng thái ban đầu của các công tắc là x = 0, y = 0 và đèn đang tắt.

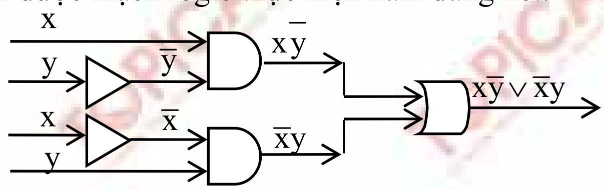
Khi đó hoạt động của đèn được cho bởi hàm đại số lôgic $f(x, y)$ xác định bởi bảng giá trị sau:

x	y	f(x, y)
0	0	0
1	0	1
1	1	0
0	1	1

Dễ thấy dạng tuyển chuẩn tắc đầy đủ của hàm là:

$$f(x,y) = x\overline{y} \vee \overline{x}y$$

Từ đó nhận được mạch lôgic thực hiện hàm đang xét:



Việc tổng hợp các mạch lôgic đã đặt cho đại số lôgic một số bài toán lý thuyết như việc tìm dạng biểu diễn tối thiểu của hàm đại số lôgic, việc mở rộng các hệ hàm đầy đủ, ... Đây là những bài toán khó mà con người đang nỗ lực tìm kiếm lời giải, và chính điều này là một trong những động lực thúc đẩy sự phát triển của lý thuyết lôgic (về phần này, bạn đọc có thể tham khảo [1]).

TÓM LƯỢC CUỐI BÀI

Qua bài học, các bạn đã nắm được những vấn đề cơ bản nhất của lý thuyết lôgic bao gồm những phép toán lôgic và những luật lôgic cơ bản. Ngoài ra, phần cuối bài có đề cập đến một số ứng dụng quan trọng của lý thuyết này.

Các bạn cần ghi nhớ các vấn đề sau:

- Khái niệm cơ bản về tập hợp và những ký hiệu thường dùng
- Các phép toán tập hợp
- Tích Đề-các; Phân hoạch và Quan hệ
- Khái niệm và các phép toán về mệnh đề
- Vị từ và lượng từ
- Ứng dụng của đại số lôgic

BÀI TẬP

1. Cho tập vũ trụ U = {1, 2, ..., 9} và các tập con của nó: A = {2, 4, 5, 6, 7}, B = {1, 2, 4}
Hãy xác định các tập $A \cup B, A \cap B, \overline{A}, A \oplus B, A - B, A \times B$

2. Chứng minh phép cộng ($\oplus$) tập hợp có các tính chất giao hoán, kết hợp, trong khi phép trừ ($-$) tập hợp không có các tính chất này.

3. Rút gọn các biểu thức tập hợp sau:

a. $((A \cup B) \cap (A \cup \overline{B})) \cup B$

b. $(\overline{A \cup B}) \cup ((\overline{A} \cap B) \cup \overline{B})$

4. Xem các quan hệ R cho trên tập số nguyên dưới đây, quan hệ nào là quan hệ tương đương và trong trường hợp R là tương đương hãy chỉ rõ phân hoạch tương ứng:

a. xRy khi và chỉ khi x + y là chẵn

b. xRy khi và chỉ khi x − y là lẻ

c. xRy khi và chỉ khi x − y chia hết cho 5

d. xRy khi và chỉ khi |x| = |y|

e. xRy khi và chỉ khi $x^2 + y^2$ là chẵn

5. Chứng minh các đẳng thức lôgic sau:

a. $(P \wedge Q \wedge R) \vee (P \wedge Q \wedge \overline{R}) \vee (\overline{P} \wedge Q \wedge \overline{R}) \;=\; Q \wedge (P \vee \overline{R})$

b. $(P \vee Q) \to R \;=\; \overline{R} \to (\overline{P} \wedge \overline{Q})$

c. $(P \to Q) \to Q \;=\; P \vee Q$

d. $\overline{P \wedge \overline{Q}} \to (\overline{Q} \to P) \;=\; \overline{P \to Q} \vee P \vee Q$

6. Chứng minh các bất đẳng thức lôgic sau:

a. $P \to Q \;\neq\; Q \to P$

b. $(P \to Q) \to R \;\neq\; P \to (Q \to R)$

7. Chứng minh các biểu thức lôgic sau là hằng đúng:

a. $((P \to Q) \wedge P) \to Q$

b. $((P \to Q) \wedge \overline{Q}) \to \overline{P}$

c. $((P \to (Q \to R)) \leftrightarrow ((P \wedge Q) \to R)$

d. $((P \to R) \wedge (Q \to R)) \leftrightarrow ((P \vee Q) \to R)$

8. Rút gọn các biểu thức lôgic sau:

a. $(P \wedge Q) \vee (\overline{P} \wedge Q) \vee (\overline{Q} \wedge R)$

b. $(P \wedge Q \wedge R) \vee (P \wedge \overline{Q} \wedge R) \vee (\overline{P} \wedge Q \wedge \overline{R})$

c. $(P \vee Q \vee \overline{R}) \wedge (\overline{P} \vee Q \vee R) \wedge (P \vee \overline{Q} \vee \overline{R})$

9. Cho các phát biểu sau:

P = "Bạn đã làm hết bài tập trong cuốn sách này"

Q = "Bạn là người đã nắm vững môn Toán rời rạc"

R = "Bạn có thể tham gia chương trình Cao học ngành Công nghệ thông tin"

Xem các phát biểu trên như các biến mệnh đề, hãy tìm một biểu thức lôgic cho các phát biểu sau:

a. "Bạn là người đã nắm vững môn Toán rời rạc nên bạn có thể tham gia chương trình Cao học ngành Công nghệ thông tin"

b. "Bạn đã nắm vững môn Toán rời rạc vì bạn đã làm hết bài tập trong cuốn sách này"

c. "Bạn có thể tham gia chương trình Cao học ngành Công nghệ thông tin dù bạn chưa làm hết các bài tập trong cuốn sách này"

d. "Bạn không thể tham gia chương trình Cao học ngành Công nghệ thông tin vì bạn chưa nắm vững môn Toán rời rạc"

10. Cho P(x, y) là vị từ "x là phần tử của y", trong đó x là phần tử thuộc tập X có ít nhất hai phần tử và y thuộc tập hợp các tập con khác rỗng của X. Hãy xác định giá trị của các mệnh đề dưới đây:

a. $\forall x \exists y P(x, y)$

b. $\forall y \exists x P(x, y)$

c. $\exists x \forall y P(x, y)$

d. $\exists y \forall x P(x, y)$

11. Người ta định nghĩa hàm số một biến thực f(x) là bị chặn trong quá trình $x \to \infty$ như sau: $\exists A \exists N \forall x ((|x| > N) \to (|f(x)| \le A))$. Hãy tìm phát biểu định nghĩa hàm f(x) là không bị chặn trong quá trình đang xét.

12. Giả sử P(n) là vị từ "n là một số nguyên tố lớn hơn 2", n là số tự nhiên. Xác định giá trị của các mệnh đề:

a. $\exists n(P(n) \wedge P(n+2) \wedge P(n+4))$

b. $\forall n(P(n) \to P(n+2))$

13. Chứng minh hoặc bác bỏ các phán đoán sau:

a. "2^n +1 là một số nguyên tố với mọi số nguyên không âm"

b. "max (x, y) + min (x, y) = x + y với mọi số thực x, y"

c. "tổng của một số hữu tỉ với một số vô tỉ là một số vô tỉ"

14. Chứng minh các suy diễn sau:

a. Từ các giả thiết $P, P \to Q, R \to \overline{Q}, S \vee R$ suy ra kết luận S

b. Từ các giả thiết $P \wedge Q, P \to (Q \wedge R), R \to (S \vee T), \overline{S}$ suy ra kết luận T

c. Từ các giả thiết $\overline{S}, T, P \to (Q \to R), P \vee S, T \to Q$ suy ra kết luận R

15. Chứng minh bằng quy nạp các khẳng định sau:

a. "$n^3 + 2n$ chia hết cho 3 với mọi số tự nhiên n"

b. "$(x + y)^n \ge x^n + y^n$ với mọi số thực dương x, y và mọi số tự nhiên n"

c. "mọi bưu phí có giá trị là một số nguyên lớn hơn 7 xu đều có thể tạo bởi hai loại tem 3 xu và 5 xu"

16. Cho các vị từ trong không gian các số nguyên P(x) = "x ≤ 5", Q(x) = "x+3 là số chẵn", R(x) = "x > 0"

a. Tìm các giá trị x để $(P \wedge Q \wedge R)(x)$ có giá trị đúng.

b. Tìm x nhỏ nhất để $(P \to (\overline{Q} \wedge R))(x)$ có giá trị đúng.

17. Đối với mỗi hàm đại số lôgic cho dưới đây, hãy làm các bước sau:

- Biểu diễn hàm chỉ qua các phép toán tuyển, hội, phủ định (càng gọn càng tốt)
- Xây dựng mạch lôgic gồm các cổng NOT, AND, OR thực hiện hàm

a. $f(x,y,z) = (x \vee y) \to (y \vee \overline{z})$

b. $f(x,y,z) = x\overline{y} \to z$

c. $f(x,y,z) = (x \oplus y) \vee (x \oplus z)$

d. $f(x,y,z) = x \to (xy \vee z)$

e. $f(x,y,z) = x \vee (y \oplus z)(\overline{x} \oplus \overline{y})$

f. $f(x,y,z) = (x \to y)(\overline{x} \to \overline{y})$

18. Cho hàm đại số lôgic f(x, y, z) lấy giá trị 1 khi và chỉ khi có ít nhất hai biến lấy giá trị 1 (mô tả kết quả bỏ phiếu theo đa số với 3 phiếu bầu). Hãy làm các bước sau:

- lập bảng giá trị của f
- tìm dạng chuẩn tắc đầy đủ của f và rút gọn nếu có thể
- xây dựng mạch lôgic gồm các cổng NOT, AND, OR thực hiện f

19. Yêu cầu như bài 18 đối với hàm f(x, y, z) mô tả trạng thái của một đèn được điều khiển bằng 3 công tắc (đèn thay đổi trạng thái khi và chỉ khi có một công tắc thay đổi trạng thái − xem ví dụ đèn cầu thang trong bài giảng).

CÂU HỎI THƯỜNG GẶP

1. Hai biểu thức lôgic tương đương khi nào?

2. Ưu điểm và hạn chế của phương pháp lập bảng trong việc chứng minh 2 biểu thức lôgic tương tương ?

3. Để chứng minh 2 biểu thức lôgic tương được, ngoài phương pháp lập bảng còn phương pháp nào khác không ?