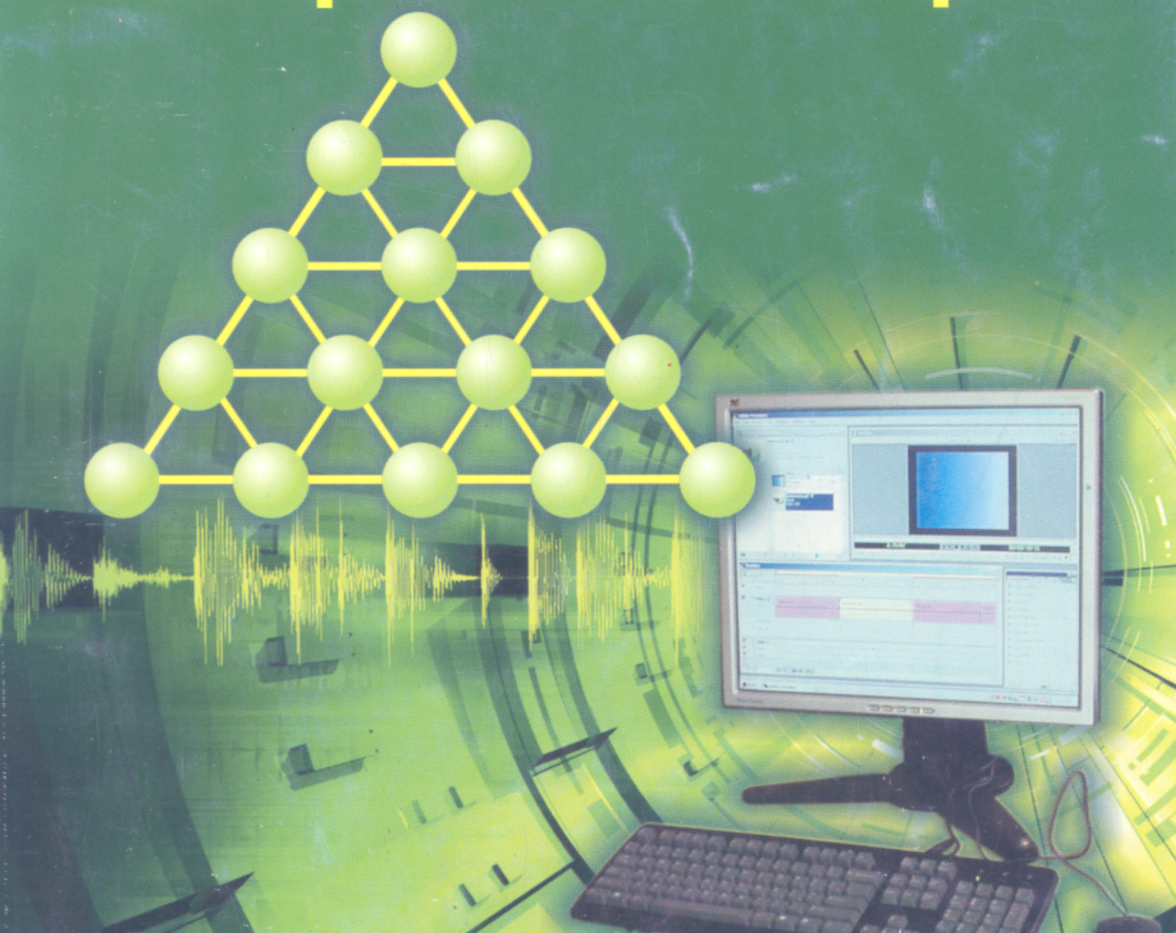
ĐỖ ĐỨC GIÁO

Toán rời rạc

ỨNG DỤNG TRONG TIN HỌC



NHÀ XUẤT BẢN GIÁO DỤC

PGS. TS. ĐỖ ĐỨC GIÁO

TOÁN RỜI RẠC

ỨNG DỤNG TRONG TIN HỌC

NHÀ XUẤT BẢN GIÁO DỤC

113–2008/CXB/59–175/GD Mã số: 7B705Y8–DAI

Lời nói đầu

Để nâng cao chất lượng giảng dạy và học tập môn Toán rời rạc cho sinh viên và học viên ngành Công nghệ thông tin và các ngành khoa học tự nhiên, chúng tôi biên soạn cuốn ***Toán rời rạc ứng dụng trong tin học*** *gồm:*

– Phần I. Kiến thức bổ trợ;

– Phần II. Lôgic và ứng dụng;

– Phần III. Đồ thị và ứng dụng;

– Phần IV. Ngôn ngữ hình thức.

Cuốn ***Toán rời rạc ứng dụng trong tin học*** *trình bày các vấn đề toán học cơ bản nhất, nhưng lại hết sức thiết yếu và cần thiết đối với những ai muốn có được các kiến thức tin học vững chắc. Cuốn sách giúp người học hiểu được lý thuyết thấu đáo, rèn luyện tư duy khoa học, kỹ năng tính toán và khả năng vận dụng toán học vào giải quyết vấn đề, kích thích niềm say mê học tập và từ đó nâng cao kỹ năng thực hành, tư duy sáng tạo khi học các môn học cơ sở và chuyên ngành Công nghệ thông tin tiếp theo. Cuốn sách này cũng rất bổ ích cho việc ôn thi tuyển sinh sau đại học ngành Công nghệ thông tin được tổ chức hàng năm ở Đại học Quốc gia Hà Nội.*

Tác giả chân thành cảm ơn các bạn đồng nghiệp đã động viên tác giả biên soạn cuốn sách này.

Cuốn sách xuất bản lần đầu, nên khó tránh khỏi thiếu sót về hình thức cũng như nội dung. Vì vậy, tác giả mong nhận được sự góp ý của bạn đọc để cuốn sách ngày càng tốt hơn. Mọi góp ý xin gửi về: Công ty Cổ phần Sách Đại học và Dạy nghề: 25 Hàn Thuyên – Hà Nội.

TÁC GIẢ

MỤC LỤC

Phần II. LÔGIC VÀ ỨNG DỤNG

PHẦN I
KIẾN THỨC BỔ TRỢ

Chương 1
CÁC KHÁI NIỆM CƠ BẢN
CỦA THUẬT TOÁN VÀ PHƯƠNG PHÁP ĐỆ QUY

§1. KHÁI NIỆM THUẬT TOÁN

1.1. Thuật toán là gì

Thuật toán là một khái niệm quan trọng của toán học. Nói đến thuật toán là nói đến một dãy các quy tắc, nhằm xác định một dãy các thao tác trên các đối tượng, sao cho sau một số hữu hạn bước thực hiện các thao tác, ta đạt được mục tiêu cần làm.

1.2. Các đặc trưng của thuật toán

– *Tính dừng:* Sau một số hữu hạn bước thuật toán phải dừng.

– *Tính xác định:* Ở mỗi bước, các thao tác phải rõ ràng, không gây nên sự nhập nhằng. Nói rõ hơn, trong cùng một điều kiện hai bộ xử lý cùng thực hiện một bước của thuật toán phải cho những kết quả như nhau.

– *Tính hiệu quả:* Trước hết thuật toán phải đúng đắn, nghĩa là sau khi đưa dữ liệu vào thuật toán hoạt động và đưa ra kết quả mong muốn.

– *Tính phổ dụng:* Thuật toán có thể giải bất kỳ một bài toán nào trong lớp các bài toán. Cụ thể là thuật toán có thể có các đầu vào là các bộ dữ liệu khác nhau trong miền xác định.

– *Yếu tố vào ra:* Đối với một thuật toán luôn có một đối tượng vào (input) và đối tượng ra (output).

1.3. Ngôn ngữ thuật toán

– Ngôn ngữ dùng để miêu tả thuật toán gọi là ngôn ngữ thuật toán.

– Thuật toán thường được mô tả bằng một dãy các lệnh. Bộ xử lý sẽ thực hiện các lệnh đó theo một trật tự nhất định cho đến khi gặp lệnh dừng thì kết thúc.

– Ngôn ngữ thuật toán bao gồm:

+ Ngôn ngữ liệt kê từng bước;

+ Sơ đồ khối;

+ Ngôn ngữ lập trình.

a) Ngôn ngữ liệt kê từng bước bao gồm:

– Thuật toán: Tên thuật toán và chức năng.

– Đầu vào: Các dữ liệu vào với tên, kiểu.

– Đầu ra: Các dữ liệu ra với tên, kiểu.

– Biến phụ (nếu có) gồm tên, kiểu.

– Hành động là các thao tác với các lệnh có nhãn là các số tự nhiên.

Ví dụ 1: Để giải phương trình bậc hai $ax^2 + bx + c = 0$ $(a \neq 0)$, ta có thể mô tả thuật toán bằng ngôn ngữ liệt kê như sau:

Bước 1: Xác định các hệ số a, b, c.

Bước 2: Kiểm tra xem hệ số a có khác 0 hay không? Nếu $a = 0$ quay lại thực hiện bước 1.

Bước 3: Tính biểu thức $\Delta = b^2 - 4ac$.

Bước 4: Nếu $\Delta < 0$ thông báo "phương trình vô nghiệm" và chuyển đến bước 8.

Bước 5: Nếu $\Delta = 0$, tính $x_1 = x_2 = \dfrac{-b}{2a}$ và chuyển sang bước 7.

Bước 6: Tính $x_1 = \dfrac{-b - \sqrt{\Delta}}{2a}$, $x_2 = \dfrac{-b + \sqrt{\Delta}}{2a}$ và chuyển sang bước 7.

Bước 7: Thông báo các nghiệm x_1, x_2.

Bước 8: Kết thúc thuật toán.

b) Sơ đồ khối

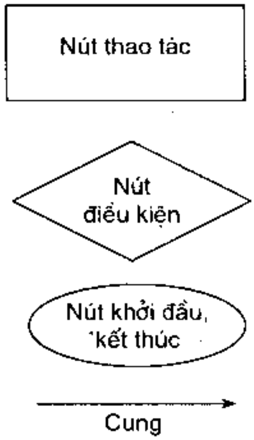
Để mô tả thuật toán bằng sơ đồ khối ta cần dựa vào các nút sau:

– *Nút thao tác:* Biểu diễn bằng hình chữ nhật, trong đó ghi câu lệnh cần thực hiện. Nếu có nhiều câu lệnh liên tiếp cần được thực hiện thì chúng có thể được viết chung trong nút thao tác;

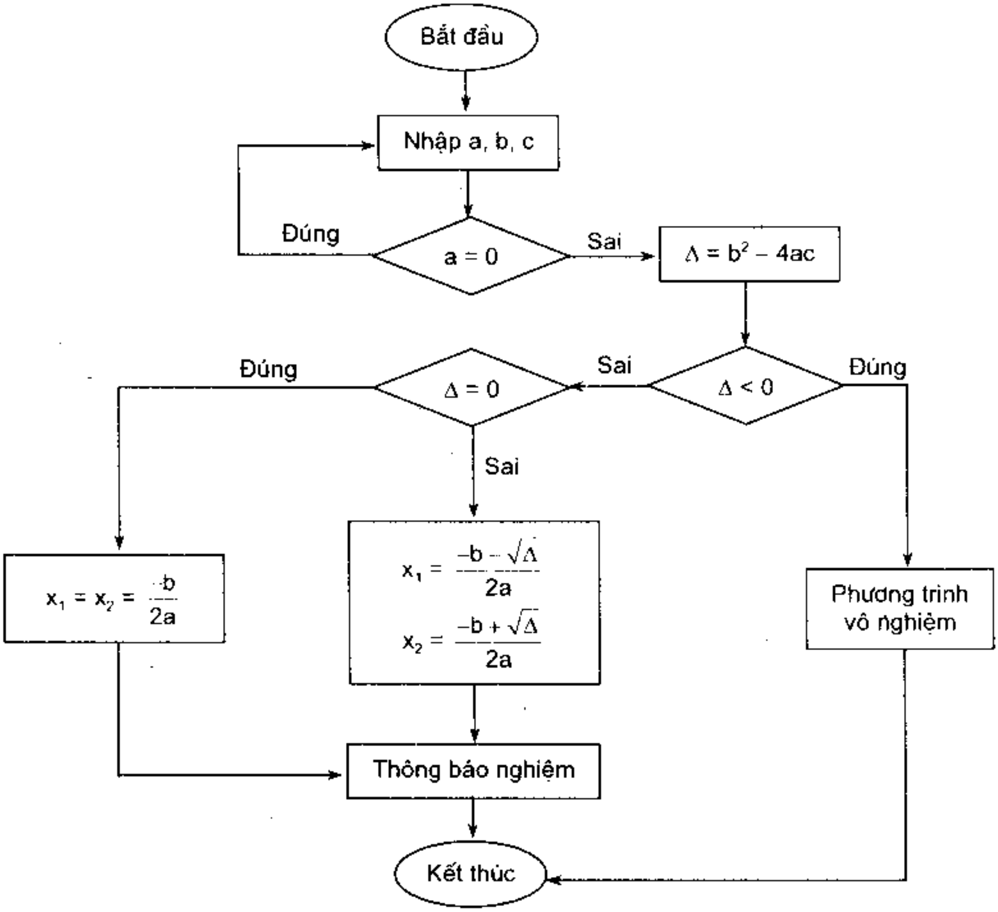
– *Nút điều kiện:* Biểu diễn bằng hình thoi, trong đó ghi điều kiện cần kiểm tra trong quá trình tính toán;

– *Nút khởi đầu, kết thúc:* Biểu diễn bằng hình elip, thể hiện sự bắt đầu hay kết thúc quá trình;

– *Cung:* Biểu diễn bằng đoạn thẳng có hướng, dùng để chỉ đường đi của thuật toán.



Chẳng hạn, giải phương trình bậc hai $ax^2 + bx + c = 0$ ($a \neq 0$) ta mô tả thuật toán bằng phương pháp sơ đồ khối như sau:



c) Ngôn ngữ lập trình

Để giải bài toán bằng máy tính, người ta thường sử dụng một loại ngôn ngữ, gọi là *ngôn ngữ lập trình*, chẳng hạn như ngôn ngữ lập trình Cobol, Algol, Pascal, ... *Chương trình* chính là một dãy hữu hạn các câu lệnh được viết theo một quy tắc nhất định trên một ngôn ngữ lập trình nào đó. Hay nói cách khác; để giải một bài toán trước hết cần có một thuật toán để giải bài toán đó; để máy tính hiểu được thuật toán, người ta sử dụng một ngôn ngữ lập trình cụ thể để diễn đạt thuật toán thông qua ngôn ngữ đó.

Chẳng hạn, giải phương trình bậc hai $ax^2 + bx + c = 0$ $(a \neq 0)$ ta mô tả thuật toán bằng một chương trình Pascal như sau:

```
Program  GIAI_PHUONG_TRINH_BAC_HAI;
uses crt;
var   a, b, c, delta, x1, x2: real;
BEGIN
      clrscr;
      write('Nhap he so :');
      repeat
          write('a = '); readln(a);
          write('b = '); readln(b);
          write('c = '); readln(c);
      until a <> 0;
      delta := sqr(b) – 4*a*c;
      if delta < 0 then
          begin
              write('Phuong trinh vo nghiem');
              halt;
          end
      else
          if delta = 0 then
              begin
                  write('Phuong trinh co nghiem kep x = ', –b/(2*a));
                  exit;
              end
          else
              begin
                  x1:= (–b – sqrt(delta))/(2*a);
                  x2:= (–b + sqrt(delta))/(2*a);
                  writeln('Phuong trinh co hai nghiem phan biet');
                  write('x1 = ', x1, ' x2 = ', x2);
                  exit;
              end
      readln;
END.
```

1.4. Độ phức tạp thuật toán

Khi đề xuất một thuật toán, ngoài việc quan tâm đến tính đúng đắn, thường phải quan tâm đến một số vấn đề như: ưu điểm, nhược điểm, tính phổ dụng, thời gian tính toán, ... Với các thuật toán được sử dụng có tần số cao như các thuật toán sắp xếp, thuật toán tìm kiếm, ... ta đặc biệt quan tâm đến thời gian cần thiết cho việc thực hiện thuật toán đó.

Thông thường với mỗi thuật toán, dữ liệu vào sẽ có kích thước là một số nào đó. Chẳng hạn, khi sắp xếp một dãy số thì kích thước của dữ liệu có thể xem như là số phần tử n của dãy đó. Rõ ràng với n càng lớn thì thời gian cần thiết cho việc sắp xếp sẽ càng lớn và nó là một hàm của đối số n. Ta ký hiệu hàm đó là f(n). Trước khi đưa vào khái niệm độ phức tạp của thuật toán, ta đề cập tới một số khái niệm sau:

a) Khái niệm về độ tăng của hàm

Một trong những khái niệm thường dùng để phân tích độ tăng của một hàm là khái niệm "O" được định nghĩa như sau:

Cho f(x) và g(x) là hai hàm từ tập các số nguyên dương hoặc tập các số thực vào tập các số thực. Ta nói f(x) là O(g(x)) nếu tồn tại hai hằng số C và k sao cho

$$|f(x)| \le C|g(x)| \text{ với mọi } x > k.$$

Cần lưu ý rằng, cặp các hằng số C và k thoả mãn điều kiện trên là không duy nhất, đồng thời nếu f(x) là O(g(x)) mà h(x) là hàm thoả mãn $|g(x)| < |h(x)|$ với $x > k$ thì ta cũng có f(x) là O(h(x)).

Ví dụ 2: Cho $f(x) = a_nx^n + a_{n-1}x^{n-1} + ... + a_1x + a_o$, với a_i là các số thực ($i = 0, 1, ..., n$). Khi đó $f(x) = O(x^n)$.

Thật vậy, với mọi $x > 1$:

$$\begin{aligned}|f(x)| &= |a_nx^n + a_{n-1}x^{n-1} + ... + a_1x + a_o| \\ &\le |a_n|x^n + |a_{n-1}|x^{n-1} + ... + |a_1|x + |a_o| \\ &= x^n\left(|a_n| + \frac{|a_{n-1}|}{x} + ... + \frac{|a_1|}{x^{n-1}} + ... + \frac{|a_o|}{x^n}\right) \\ &\le x^n\left(|a_n| + |a_{n-1}| + ... + |a_1| + |a_o|\right).\end{aligned}$$

Đặt $C = |a_n| + |a_{n-1}| + ... + |a_1| + |a_o|$ và $k = 1$, ta có

$$|f(x)| \le Cx^n \text{ với mọi } x > k, \text{ hay } f(x) \text{ là } O(x^n).$$

Ví dụ 3: Cho $f(n) = 1 + 2 + 3 + ... + n$. Chỉ ra $f(n)$ là $O(n^2)$.

Thật vậy, $f(n) < n + n + n + ... + n = n.n = n^2$, nên $f(n)$ là $O(n^2)$ với $C = k = 1$.

Ví dụ 4: Chỉ ra hàm $n!$ là $O(n^n)$ và $\log n!$ là $O(n\log n)$.

Thật vậy, ta có $n! < n^n$ nên $n!$ là $O(n^n)$ với $C = k = 1$.

Từ $n! < n^n \Rightarrow \log n! < n\log n$, vậy $\log n!$ là $O(n\log n)$ với $C = k = 1$.

Ví dụ 5: Hãy chỉ ra hàm $\log n$ là $O(n)$.

Thật vậy, ta có $n < 2^n$ với mọi $n \geq 1$, nên $\log n < n$ (lôgarit cơ số 2) và vì vậy $\log n$ là $O(n)$ với $C = k = 1$.

b) Độ tăng của tổ hợp các hàm

Từ khái niệm về độ tăng của hàm ta có một số kết quả sau đây:

Ví dụ 6: Nếu $f_1(x)$ là $O(g_1(x))$ và $f_2(x)$ là $O(g_2(x))$ thì

$$(f_1(x) + f_2(x)) \text{ là } O(\max\{|g_1(x)|, |g_2(x)|\}).$$

Thật vậy, theo giả thiết tồn tại C_i và k_i sao cho

$$|f_i(x)| \leq C_i |g_i(x)| \text{ với mọi } x > k_i, (i = 1, 2).$$

Ta lại có

$$\begin{aligned} |f_1(x) + f_2(x)| &\leq |f_1(x)| + |f_2(x)| \\ &\leq C_1|g_1(x)| + C_2|g_2(x)| \\ &\leq C|g(x)| \text{ với mọi } x > k, \end{aligned}$$

ở đây $k = \max\{k_1, k_2\}$; $C = C_1 + C_2$; $g(x) = \max\{|g_1(x)|, |g_2(x)|\}$.

Ví dụ 7: Cho $f_1(x)$ và $f_2(x)$ đều là $O(g(x))$. Khi đó

$f_1(x) + f_2(x)$ là $O(g(x))$ (hệ quả của ví dụ 6).

Ví dụ 8: Nếu $f_1(x)$ là $O(g_1(x))$ và $f_2(x)$ là $O(g_2(x))$ thì

$f_1(x).f_2(x)$ là $O(g_1(x).g_2(x))$.

Thật vậy, theo giả thiết $|f_1(x)| \leq C_1|g_1(x)|$ với mọi $x > k_1$

và $|f_2(x)| \leq C_2|g_2(x)|$ với mọi $x > k_2$.

Xét $|f_1(x).f_2(x)| = |f_1(x)|.|f_2(x)| \leq C_1C_2|g_1(x)|.|g_2(x)|$

$= C_1C_2|g_1(x).g_2(x)|$.

Vậy $f_1(x).f_2(x)$ là $O(g_1(x).g_2(x))$, với $C = C_1C_2$, $k = \max\{k_1, k_2\}$.

Ví dụ 9: Cho $f(n) = n\log(n!) + n^2\log n$, với $n = 1, 2, 3, \ldots$ Khi đó f(n) là $O(n^2\log n)$.

Thật vậy, ta có $n\log(n!)$ là $O(n^2\log n)$, còn $n^2\log n$ là $O(n^2\log n)$. Từ ví dụ 7 ta có f(n) là $O(n^2\log n)$.

Ví dụ 10: Chứng minh rằng, nếu f(x) là O(g(x)) thì $f^n(x)$ là $O(g^n(x))$, ở đây $f^n(x) = \underbrace{|f(x).f(x)\ldots f(x)|}_{\text{n lần}} = \underbrace{|f(x)|.|f(x)|\ldots|f(x)|}_{\text{n lần}} \le C^n |g^n(x)|$

với mọi x > k.

Hay $|f^n(x)| \le C|g^n(x)|$ với mọi x > k, $C := C^n$.

Vậy $f^n(x)$ là $O(g^n(x))$.

Ví dụ 11: Giả sử f(x) là O(g(x)) với f(x), g(x) là các hàm đơn điệu tăng và không giới nội. Chứng minh log(f(x)) là O(log(g(x))).

Thật vậy, theo giả thiết ta có $|f(x)| \le C_1|g(x)|$ với mọi $x > k_1$. Ta có thể giả thiết f(x) > 1, g(x) > 1 với x đủ lớn, hay $f(x) \le Cg(x)$ với x đủ lớn. Lôgarit cơ số 2 bất đẳng thức trên ta có

$$\log(f(x)) \le \log(C_1g(x)) = \log C_1 + \log(g(x)) \le 2\log(g(x)) \text{ với x đủ lớn.}$$

Hay log(f(x)) là O(log(g(x)).

c) Độ phức tạp thuật toán

Ở đây chúng ta chỉ đề cập tới độ phức tạp của thuật toán về thời gian tính toán mà không đề cập tới độ phức tạp về không gian của thuật toán.

Các phép toán được dùng để đo độ phức tạp thời gian của thuật toán là phép so sánh số nguyên; các phép cộng, trừ, nhân, chia các số nguyên; hoặc bất kỳ một phép tính sơ cấp nào khác xuất hiện trong quá trình tính toán.

Cần lưu ý các thuật ngữ dùng trong độ phức tạp thuật toán thường gặp:

– Độ phức tạp O(1) gọi là *độ phức tạp hằng số.*

– Độ phức tạp O(logn) gọi là *độ phức tạp lôgarit.*

– Độ phức tạp O(n) gọi là *độ phức tạp tuyến tính.*

– Độ phức tạp O(nlogn) gọi là *độ phức tạp nlogn.*

– Độ phức tạp $O(n^b)$ gọi là *độ phức tạp đa thức.*

– Độ phức tạp $O(b^n)$ (b > 1) gọi là *độ phức tạp hàm mũ.*

– Độ phức tạp O(n!) gọi là *độ phức tạp giai thừa.*

Để minh hoạ về độ phức tạp của thuật toán ta xét một số ví dụ sau:

Ví dụ 12: Mô tả thuật toán tìm phần tử lớn nhất của dãy hữu hạn các số nguyên và tìm độ phức tạp của thuật toán đó.

Bước 1: Đặt giá trị cực đại tạm thời bằng số nguyên đầu tiên của dãy.

Bước 2: So sánh số nguyên tiếp theo với giá trị tạm thời, nếu nó lớn hơn giá trị cực đại tạm thời thì đặt cực đại tạm thời bằng số nguyên đó.

Bước 3: Lặp lại bước 2 (nếu còn các số nguyên trong dãy).

Bước 4: Dừng khi không còn số nguyên nào trong dãy. Khi đó cực đại tạm thời chính là số nguyên lớn nhất trong dãy.

Mô tả thuật toán tìm phần tử lớn nhất trong dãy hữu hạn:

```
Procedure MAX(a1, a2, ..., an : integer);
    max := a1;
    for i := 2 to n do
        if max < ai then max := ai
    {max là phần tử lớn nhất}
```

Lưu ý: Mỗi số hạng của dãy dùng hai phép so sánh, một để xác định chưa đạt đến cuối dãy, một để xác định có phải nó là giá trị lớn nhất tạm thời hay không. Việc so sánh này được dùng cho mỗi phần tử a_i trong dãy từ phần tử thứ hai trở đi ($i = 2, 3, ..., n$). Sau đó là phép so sánh để ra khỏi vòng lặp, nên số phép so sánh cần dùng tất cả là $2(n - 1) + 1$ đối với thuật toán trên. Vậy thuật toán trên có độ phức tạp thời gian là $O(n)$ (độ phức tạp tuyến tính).

Ví dụ 13: Mô tả thuật toán xác định vị trí của một phần tử trong một bảng liệt kê sắp thứ tự. Bài toán tìm kiếm được phát biểu như sau:

Xác định vị trí của x trong bảng liệt kê các phần tử phân biệt: $a_1, a_2, ..., a_n$.

Dưới đây ta đề cập tới thuật toán tìm kiếm tuyến tính:

Mô tả thuật toán: Trước hết ta so sánh x với a_1, nếu $x = a_1$ thì vị trí là 1, còn trường hợp $x \neq a_1$ ta so sánh tiếp x với a_2. Nếu $x = a_2$ thì vị trí của x là 2, còn trường hợp $x \neq a_2$ thì so sánh tiếp x với a_3. Quá trình trên được tiếp tục cho đến khi nếu $x = a_i$ ($i \leq n$) thì vị trí là i, trường hợp $x \neq a_i$ ($\forall i \leq n$) thì x không có mặt trong bảng liệt kê.

Mô tả thuật toán tìm kiếm tuyến tính:

```
Procedure TÌM_KIẾM_TT (x: integer, a1, a2; ..., an: các số nguyên phân biệt);
    i := 1
    while (i ≤ n) and (x ≠ ai) do
        i := i + 1;
    if i ≤ n then location := i
    else location := 0
    {location là chỉ số của số hạng bằng x (location = 0 nếu không tìm được x)}.
```

Lưu ý: Ở mỗi bước của vòng lặp trong thuật toán trên có hai phép so sánh: một để xem đã tới cuối dãy chưa; một để xem phần tử x có trùng với một phần tử của bảng liệt kê hay không. Cuối cùng là một phép so sánh ngoài vòng lặp.

Nếu $x = a_i$ thì có $2i + 1$ phép so sánh được sử dụng. Số phép so sánh lớn nhất là $2n + 1$ (khi $x \neq a_i$, $i = 1, 2, ..., n$). Sau đó cần một phép so sánh để thoát khỏi vòng lặp. Vậy, khi x không có mặt trong bảng thì tổng số phép so sánh trong thuật toán này là $2n + 2$. Vậy độ phức tạp của thuật toán là $O(n)$.

Xét bài toán trên trong trường hợp các phần tử trong bảng được sắp theo thứ tự tăng dần. Ví dụ, tìm số 19 trong bảng liệt kê theo thứ tự tăng dần: 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 13, 15, 16, 17, 18, 19, 20, 22. Tổng quát: Tìm x trong bảng liệt kê $a_1, a_2, ..., a_n$ với $a_1 < a_2 <... < a_n$.

Để giải bài toán trên ta dùng thuật toán nhị phân, được mô tả như sau: So sánh x với số hạng a_m ở giữa dãy ($m = [(n + 1)/2]$, ở đây $[x]$ là phần nguyên lớn nhất không vượt quá x). Nếu $x > a_m$ thì việc tìm kiếm x trên dãy gồm các số hạng $a_{m+1}, a_{m+2}, ..., a_n$; còn nếu $x \leq a_m$ thì việc tìm kiếm sẽ thực hiện trên dãy gồm các số $a_1, a_2, ..., a_m$. Cả hai trường hợp đều đưa đến bài toán tìm kiếm trên bảng không lớn hơn $[n/2]$ phần tử. Quá trình trên được thực hiện cho tới khi bảng liệt kê chỉ còn một phần tử và so sánh x với chính số hạng này.

Mô tả thuật toán tìm kiếm nhị phân:

```
Procedure TÌM_KIẾM_NHỊ_PHÂN (x : integer, a1, a2, ..., an: integer - tăng dần)
    i := 1 {i là điểm mút trái của khoảng tìm kiếm}
    j := n {j là điểm mút phải của khoảng tìm kiếm}
    while i < j
        begin
            m := [(i + j)/2]
            if x > am then i := m + 1
            else j := m
        end
```

if x = a_i then location := i
else location := 0
{location là chỉ số của số hạng bằng x (location = 0 nếu không tìm thấy x)}.

Lưu ý: Độ phức tạp của thuật toán tìm kiếm nhị phân trên là O(logn) vì nó phải dùng tới 2[logn] + 2 phép toán so sánh. Như vậy, trong trường hợp bảng liệt kê được sắp thứ tự thì thuật toán tìm kiếm nhị phân tốt hơn thuật toán tìm kiếm tuyến tính.

Để kết thúc phần này ta xét thêm thuật toán Euclid.

Trong lý thuyết số người ta đã chứng minh được rằng:

Nếu a = bq + r, trong đó a, b, q, r là các số nguyên thì:

$$\text{ƯCLN}(a, b) = \text{ƯCLN}(b, r) \qquad (*)$$

Giả sử a, b là các số nguyên dương, với $a \geq b$. Đặt $r_o = a$, $r_1 = b$. Bằng cách áp dụng kết quả trên ta có:

$$r_o = r_1q_1 + r_2 \qquad (0 \leq r_2 < r_1)$$

$$r_1 = r_2q_2 + r_3 \qquad (0 \leq r_3 < r_2)$$

$$\dots$$

$$r_{n-2} = r_{n-1}q_{n-1} + r_n \qquad (0 \leq r_n < r_{n-1})$$

$$r_{n-1} = r_nq_n.$$

Số dư xuất hiện trong dãy các phép chia liên tiếp, vì dãy các số $a = r_o > r_1 > r_2 > \dots \geq 0$ không thể chứa quá a số hạng. Từ (*) ta suy ra:

$$\text{ƯCLN}(a, b) = \text{ƯCLN}(r_o, r_1) = \text{ƯCLN}(r_1, r_2) = \dots$$

$$= \text{ƯCLN}(r_{n-2}, r_{n-1}) = \text{ƯCLN}(r_{n-1}, r_n) = \text{ƯCLN}(r_n, 0) = r_n.$$

Vậy, ước chung lớn nhất là số dư khác 0 cuối cùng trong dãy các phép chia.

Ví dụ 14: Tìm ƯCLN(414, 662).

Áp dụng thuật toán Euclid ta được:

$$662 = 414.1 + 248$$

$$414 = 248.2 + 166$$

$$248 = 166.1 + 82$$

$$166 = 82.2 + 2$$

$$82 = 2.41$$

Do đó ƯCLN(414, 662) = 2 (2 là số dư cuối cùng khác không).

Thuật toán Euclid được mô tả như sau:

```
Procedure UCLN(a, b : integer);
    x := a
    y := b
    while y ≠ 0
        begin
            r := x mod y
            x := y
            y := r
    end {UCLN(a, b) là x}
```

Lưu ý: Giá trị ban đầu của x, y tương ứng là a và b. Ở mỗi giai đoạn của thủ tục x được thay bằng y và y được thay bởi x mod y (là số dư r trong phép chia x cho y). Quá trình này được lặp lại nếu $y \neq 0$ và thuật toán sẽ dừng khi $y = 0$. Giá trị của x ở thời điểm này là số dư khác 0 cuối cùng trong thủ tục và chính là ước số chung lớn nhất cần tìm.

§2. PHƯƠNG PHÁP ĐỆ QUY

Đệ quy là một khái niệm tồn tại trong cuộc sống, trong toán học, trong lập trình. Đệ quy cho một phương pháp ngắn gọn và sáng sủa để mô tả các đối tượng cũng như một số quá trình. Như vậy, đệ quy là một phương pháp xác định tập hợp các đối tượng thoả mãn một yêu cầu nào đó. Nó bao gồm các quy tắc, trong đó một số quy tắc dùng để xác định các đối tượng ban đầu, còn quy tắc khác dùng để xác định các đối tượng tiếp theo nhờ các đối tượng ban đầu đã được xác định.

Ví dụ 1: Một dãy số $a_o, a_1, a_2, ..., a_n, ...$ có tính chất: $a_o = 0$, $a_n = 2a_{n-1} + 3$ với mọi $n \geq 1$. Khi đó ta có thể xác định được số hạng a_n tổng quát của dãy số trên như sau:

Trước hết chọn các số α_1, α_2 sao cho

$$(a_n - \alpha_1) = \alpha_2(a_{n-1} - \alpha_1) \quad (1)$$

Từ (1) suy ra $a_n = \alpha_2 a_{n-1} - \alpha_1.\alpha_2 + \alpha_1$.

Rõ ràng các số α_1, α_2 phải thoả mãn hệ phương trình

$$\begin{cases} \alpha_2 = 2 \\ \alpha_1 - \alpha_1.\alpha_2 = 3 \end{cases}$$

Từ hệ trên ta có ngay $\alpha_1 = -3$, $\alpha_2 = 2$. Đặt $q_n = a_n - \alpha_1$ với $n \geq 1$. Suy ra $q_n = \alpha_2.q_{n-1}$ với $n \geq 1$ và $q_1 = 6$, từ đó ta có $q_n = q_1.\alpha_2^{n-1}$.

Vậy $a_n - \alpha_1 = q_1.\alpha_2^{n-1}$, suy ra $a_n = \alpha_1 + q_1.\alpha_2^{n-1} = 6.2^{n-1} - 3$.

Chương trình con đệ quy để tính số hạng tổng quát a_n được minh hoạ như sau:

```
Function fn(n: integer): integer;
    BIGIN
        if n = 0 then fn := 0
        else fn := 2*fn(n-1) + 3;
    END;
```

Ví dụ 2: Một dãy a_o, a_1, a_2, ..., a_n, ... có tính chất: $a_o = 0$, $a_1 = 1$, $a_n = a_{n-1} + a_{n-2}$ với mọi $n \geq 2$, được gọi là dãy Fibonacci. Như vậy, dãy Fibonacci là dãy: 0, 1, 1, 2, 3, 5, 8, 13, 21, ... và với $n \geq 2$ ta có thể tìm được số Fibonacci thứ n nếu biết số Fibonacci thứ $n - 1$ và thứ $n - 2$. Bằng quy nạp toán học ta dễ dàng chứng minh được rằng, với $n \geq 0$

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

Tuy nhiên công thức trên có thể tính trực tiếp như sau:

Trước hết hãy tìm một cặp số α_1, α_2 sao cho:

$$a_n - \alpha_1 a_{n-1} = \alpha_2(a_{n-1} - \alpha_1 a_{n-2})$$

$$\Rightarrow \quad a_n = (\alpha_1 + \alpha_2)a_{n-1} - \alpha_1\alpha_2 a_{n-2}$$

Rõ ràng các cặp số α_1, α_2 thoả mãn hệ phương trình

$$\begin{cases} \alpha_1 + \alpha_2 = 1 \\ \alpha_1\alpha_2 = -1 \end{cases}$$

Khi đó α_1, α_2 sẽ là nghiệm của phương trình $x^2 - x - 1 = 0$, do đó ta có thể chọn $\alpha_1 = \dfrac{1+\sqrt{5}}{2}$, $\alpha_2 = \dfrac{1-\sqrt{5}}{2}$. Đặt $q_n = a_n - \alpha_1 a_{n-1}$, khi đó $q_n = \alpha_2 q_{n-1}$ với $n \geq 2$ và $q_1 = 1$. Ta có $q_n = \alpha_2^{n-1}$.

Vậy $a_n - \alpha_1 a_{n-1} = \alpha_2^{n-1}$, suy ra $a_n = \alpha_1 a_{n-1} + \alpha_2^{n-1}$.

$$a_2 = \alpha_1 a_1 + \alpha_2 \qquad (1)$$
$$a_3 = \alpha_1 a_2 + \alpha_2^2 \qquad (2)$$
$$a_4 = \alpha_1 a_2 + \alpha_2^3 \qquad (3)$$
...
$$a_{n-1} = \alpha_1 a_{n-2} + \alpha_2^{n-2} \qquad (n-2)$$
$$a_n = \alpha_1 a_{n-1} + \alpha_2^{n-1} \qquad (n-1)$$

Khi nhân hai vế của đẳng thức (n – 2) với α_1, đẳng thức (n – 3) với α_1^2, ... đẳng thức (1) với α_1^{n-2} và cộng lại theo từng vế ta có:

$$a_n = \alpha_1^{n-1}a_1 + \alpha_1^{n-2}\alpha_2 + ... + \alpha_1^{n-3}\alpha_2^2 = \frac{\alpha_1^n - \alpha_2^n}{\alpha_1 - \alpha_2}$$

Vậy $a_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$ với $n \geq 2$. Tuy nhiên công thức này cũng đúng với $n \geq 0$.

Chương trình con để tính số hạng tổng quát thứ n có thể được viết như sau:

```
Function fn(n : integer) : integer;
  BEGIN
    if n <= 1 then fn := n
    else fn := fn(n – 1) + fn(n – 2);
  END;
```

Ví dụ 3: Có m phần thưởng được chia cho n học sinh giỏi được xếp hạng từ 1, 2, ..., n. Hỏi rằng có bao nhiêu cách chia phần thưởng trên sao cho các điều kiện sau được thoả mãn:

a) Nếu học sinh A giỏi hơn học sinh B thì số phần thưởng của A sẽ lớn hơn hoặc bằng số phần thưởng của B;

b) Tất cả m phần thưởng đều phải thưởng hết cho học sinh.

Giả sử rằng với $i < j$ và $1 \leq i, j \leq n$ thì học sinh thứ i giỏi hơn học sinh thứ j. Ký hiệu số phần thưởng mà học sinh thứ i nhận được là T_i. Từ giả thiết của bài toán ta suy ra các điều kiện sau cần được thoả mãn:

$$T_1 + T_2 + ... + T_n = m;$$
$$T_1 \geq T_2 \geq ... \geq T_n \geq 0.$$

Ký hiệu CHIA(m, n) là số cách chia m phần thưởng cho n học sinh theo yêu cầu trên.

Nhận xét rằng:

– Khi m = 0, tức là không có phần thưởng nào thì khi đó rõ ràng chỉ có duy nhất một cách chia: mỗi học sinh được 0 phần thưởng. Vậy CHIA(0, n) = 1 đối với mọi n.

– Khi có m phần thưởng và không có học sinh giỏi, đương nhiên sẽ không có cách chia nào. Vậy CHIA(m, 0) = 0 đối với mọi $m \neq 0$.

– Khi có m phần thưởng, n học sinh giỏi và $m < n$, khi đó $n - m$ học sinh phía cuối sẽ không nhận được phần thưởng nào, do đó:

$$\text{CHIA}(m, n) = \text{CHIA}(m, m) \text{ khi } m < n.$$

– Khi có m phần thưởng, n học sinh giỏi và $m > n$. Khi đó ta xét hai trường hợp sau:

+ Khi học sinh cuối cùng không nhận được phần thưởng nào, khi đó số cách chia sẽ là CHIA(m, n – 1);

+ Khi học sinh cuối cùng nhận được 1 phần thưởng, khi đó tất cả các học sinh giỏi sẽ nhận được ít nhất 1 phần thưởng, do đó số cách chia trong trường hợp này sẽ là CHIA(m – n, n).

Tóm lại, khi có m phần thưởng và n học sinh giỏi thì số cách chia sẽ là CHIA(m, n – 1) + CHIA(m – n, n).

Trên cơ sở đó ra có thể viết một chương trình con đệ quy thể hiện thuật toán trên như sau:

```
Function CHIA(m, n : integer) : integer;
    BEGIN
        if m = 0 then CHIA := 1
        else    if n = 0 then CHIA := CHIA(m, m)
                else CHIA := CHIA(m, n – 1) + CHIA(m – n, n)
    END;
```

Thuật toán quay lui

Bài toán: Hãy xây dựng các bộ giá trị gồm có n thành phần $(x_1, x_2, ..., x_n)$ từ một tập hữu hạn cho trước, sao cho các bộ đó thoả mãn một yêu cầu B cho trước nào đó.

Phương pháp:

Xây dựng các phần tử $x_1, x_2, ..., x_n$ theo cách sau đây:

Giả sử đã xác định được k – 1 phần tử đầu tiên của dãy đó là $x_1, x_2, ..., x_{k-1}$. Cần xác định tiếp phần tử x_k, phần tử này được xác định theo cách sau:

Giả sử T_k là tập tất cả các giá trị mà phần tử x_k có thể nhận được. Vì tập T_k là hữu hạn nên có thể đặt n_k là số phần tử của tập T_k. Khi đó có thể liệt kê các phần tử của T_k theo một thứ tự nào đó. Tức là có thể thành lập một ánh xạ 1 – 1 từ tập T_k lên tập $\{1, 2, ..., n_k\}$.

– Nếu k = n, khi đó bộ $(x_1, x_2, ..., x_{k-1}, x_k)$ có đủ n thành phần và thoả mãn yêu cầu đặt ra, do đó bộ này được đưa vào quan hệ kết quả.

Xét $j \in \{1, 2, ..., n_k\}$, ta nói rằng "j chấp nhận được" nếu có thể bổ sung phần tử thứ j trong T_k với tư cách là phần tử x_k vào dãy $x_1, x_2, ..., x_{k-1}, x_k$.

– Nếu k < n ta cần phải thực hiện tiếp quá trình trên, tức là phải bổ sung tiếp phần tử x_{k+1} vào dãy $x_1, x_2, ..., x_{k-1}$ để được dãy $x_1, x_2, ..., x_{k-1}, x_k$.

Như vậy, nét đặc trưng của phương pháp quay lui là ở chỗ có được lời giải ta phải đi từng bước bằng phép thử. Khi một bước lựa chọn chấp nhận cần ghi nhớ thông tin cần thiết và tiến hành bước tiếp theo. Ngược lại, khi không có một lựa chọn nào được chấp nhận thì cần phải thực hiện lại bước trước, xoá bớt ghi nhớ và thử lại với những lựa chọn còn lại.

a) Thủ tục đệ quy cho thuật toán quay lui

```
Procedure THU (k: interger);
   Var j: interger;
   BEGIN
      for j := 1 to nk do
         if <j là chấp nhận được> then
            Begin
               <xác định xk theo j>
               if k = n then <ghi nhận bộ giá trị mới>
               else THU(k + 1);
            End;
   END;
```

Điều đáng quan tâm ở đây là:

1) Làm thế nào để xác định được tập T_k, tức là tập tất cả các khả năng mà phần tử thứ k của dãy $x_1, x_2, ..., x_n$ có thể nhận được.

2) Khi đã có tập T_k để xác định x_k, thấy rằng x_k không những phụ thuộc vào chỉ số j mà còn phụ thuộc vào các $x_1, x_2, ..., x_{k-1}$. Vì vậy cần phải ghi nhớ trạng thái của quá trình xác định dãy bằng một biến nhớ nào đó sau khi xác định được x_k theo j và đồng thời phải trả lại trạng thái cũ sau lời gọi THU(k + 1).

Ngoài thủ tục THU(k) nên có một thủ tục gọi là khởi tạo, nhằm tạo ra một số giá trị cũng như những trạng thái cần thiết ban đầu.

b) Các ví dụ minh hoạ cho thuật toán quay lui

Trong các ví dụ dưới đây bộ giá trị $(x_1, x_2, ..., x_n)$ sẽ được lưu trữ dưới dạng mảng một chiều là x[1], x[2], ..., x[n].

Ví dụ 4: Liệt kê tất cả các hoán vị của n số tự nhiên đầu tiên.

Đặt N = {1, 2, ..., n}. Khi đó một hoán vị của n số tự nhiên đầu tiên sẽ là một bộ x[1], x[2], ..., x[n]. Trong đó x[i] ≠ x[j] với i ≠ j và x[i] ∈ N. Nhận thấy T_1 = N. Giả sử đã xác định được x[1], x[2], ..., x[k−1], khi đó:

$$T_k = N - \{x[1], x[2], ..., x[k-1]\}.$$

Để ghi nhớ tập T_i (i = 1, ..., n), ta cần sử dụng một mảng b[1..n] là các giá trị boolean sao cho b[i] = true khi và chỉ khi i ∈ T_k.

Sau đây là chương trình liệt kê tất cả các hoán vị của n số tự nhiên đầu tiên.

```
Program HOANVI;
uses crt;
type    mang = array[1..20] of interger;
Var     x: mang;
        b: array[1..20] of boolean;
        n, d: interger;
(*-----------------*)
Procedure KHOI_TAO;
Var     i: interger;
BEGIN
        write('so phan tu = "); readln(n);
        for i:= 1 to n do b[i] := true;
        d := 0;
END;
(*----------------------*)
Procedure INKQ;
Var     i: interger;
```

```
BEGIN
          inc(d);
          write ('Hoan vi thu', d:3, ':');
          for i := 1 to n do write (x[i]:4);
          writeln;
END;
(*-----------------*)
Procedure THU(k: interger);
Var       j := interger;
BEGIN
          for j := 1 to n do
              if b[j] then
                  Begin
                      x[k] := j;
                      b[j] := false; {chuyen ve trang thai moi}
                      if k = n then INKQ
                      else THU(k+1);
                      b[j] := true; {tro ve trang thai cu}
                  End;
END;
(*------------------*)
BEGIN {Main}
          clrscr;
          KHOI_TAO;
          THU(1);
          readln;
END. {Hoan vi}
```

Ví dụ 5: Liệt kê tất cả các chuỗi nhị phân có độ dài n.

Nhận thấy rằng, một chuỗi nhị phân độ dài n có dạng x[1], x[2], ..., x[n] với x[i] $\in$ B (B = {0, 1}). Ta có T_1 = B. Giả sử đã xác định được x[1], x[2], ..., x[k−1]. Thấy rằng T_k = B, do đó số phần tử của tập T_k = 2.

Chương trình liệt kê tất cả các chuỗi nhị phân có độ dài n được thể hiện như sau:

```
Program CHUOI_NHI_PHAN;
uses crt;
type      mang = aray[1..20] of 0..1;
```

```
Var     x: mang;
        n, d: integer;
(*----------------------*)
Procedure KHOI_TAO;
BEGIN
        wrete('Do dai cua chuoi:'); readln(n);
        d := 0;
END;
(*---------------------*)
Procedure INKQ;
Var     i: integer;
BEGIN
    inc(d);
    write ('Chuoi thu', d:3, ':');
    for i := 1 to n do write (x[i]:4);
    writeln;
END;
(*---------------------------*)
Procedure THU(k: integer);
Var     J: interger;
BEGIN
    for j := 0 to 1 do
        begin
            x[k] := j;
            if k = n then INKQ
            else THU(k + 1)
        end;
END;
(*---------------------------*)
BEGIN
    clrscr;
    KHOI_TAO;
    THU(1);
    writeln;
    write('Tong so co', d, 'chuoi nhi phan');
    readln;
END.
```

Ví dụ 6: Bài toán đặt các con hậu.

Phát biểu bài toán: Cho bàn cờ hình vuông cỡ n × n. Hãy liệt kê tất cả các cách xếp n con hậu trên bàn cờ sao cho chúng không ăn được nhau. Tức là không có hai con hậu nào cùng dòng, cùng cột, cùng đường chéo nào đó cùng phương với đường chéo chính hoặc đường chéo phụ.

Thuật toán:

Xét một bàn cờ gồm n hàng, n cột được đánh số 1, 2, ..., n. Sử dụng chỉ số k để chỉ hàng và chỉ số j để chỉ cột. Vậy: $1 \le k, j \le n$. Quy ước nếu x[k] = j thì có nghĩa con hậu ở hàng thứ k và cột thứ j.

Nhận thấy $T_1 = N = \{1, 2, ..., n\}$. Giả sử đã xác định được x[1], x[2], ..., x[k−1], cần xác định tiếp x[k] thoả mãn yêu cầu của bài toán. Trước hết ta có một số nhận xét:

– Bàn cờ có hai đường chéo (đường chéo chính và đường chéo phụ), khi đó có 2n đường chéo cùng chiều với đường chéo chính. Nhận thấy rằng, với hai ô bất kỳ trên cùng một đường chéo thì tổng chỉ số hàng với chỉ số cột của các ô đó là như nhau, chẳng hạn cùng bằng một số i nào đó. Khi đó i là chỉ số của đường chéo đó. Rõ ràng $2 < 2n$ và $i = k + j$ nếu đường chéo đó đi qua ô có toạ độ là (k, j).

Trên cơ sở đó, có thể chọn một mảng c[2..2n] các giá trị Boolean để ghi nhận xem đường chéo đó có thể còn đặt con hậu vào đó được không, tức là c[i] = true khi và chỉ khi trên trường chéo đó chưa có con hậu nào.

– Tương tự thấy rằng, cũng có 2n − 1 đường chéo cùng chiều với đường chéo phụ. Nhận thấy, với hai ô bất kỳ trên cùng một đường chéo thì hiệu của chỉ số hàng và chỉ số cột của các ô đó là như nhau, chẳng hạn cùng bằng một số m nào đó. Khi đó ta gọi m là chỉ số của đường chéo đó. Rõ ràng $1 - n \le m \le n - 1$ và $m = k - j$ nếu đường chéo đó đi qua ô có toạ độ là (k, j). Ta cũng chọn một mảng p[2..2n] các giá trị Boolean để ghi nhận xem đường chéo đó có thể còn đặt con hậu vào đó được không, tức là p[m] = true khi và chỉ khi trên đường chéo đó chưa có con hậu nào. Khi đó tập T_k là tập có thể của các chỉ số các cột mà con hậu trên hàng thứ k có thể đặt vào. Như vậy, tập tất cả các giá trị mà x[k] có thể nhận là tập T_k được xác định bởi biểu thức:

$$T_k = \{j : j \in \mathbb{N},\ j \neq x[h] \text{ với } h \in \{1, 2, ..., k-1,$$
$$c[k+j] = \text{true},\ p[k-j] = \text{true}\}$$

Chương trình để giải bài toán:

```
Program TAM_HAU;
uses crt;
type    mang = array[1..20] of integer;
var     x : mang;
        a : array[1..20] of boolean;
        c : array[2..40] of boolean;
        p : array[-19..19] of boolean;
        n, d : integer;
(*--------------------*)
Procedure KHOI_TAO;
var  i, j, m: integer;
BEGIN
     write('Kich co ban co'); readln(n);
     for j := 1 to n do a[j] := true;
     for i := 2*n do c[i] := true;
     for m := 1 – n to n – 1 do p[m] := true;
     d := 0;
END;
(*--------------------*)
Procedure INKQ;
Var  i: integer;
BEGIN
     inc(d);
     write('Phuong an', d:3, ':');
     for i := 1 to n do
        begin
           write(x[i]:4);
           if i mod 10 = 0 then readln;
        end;
     writeln;
END;
(*----------------*)
Procedure THU(k: integer);
Var  j: integer;
BEGIN
     for j := 1 to n do
```

```
            if a[j] and c[j+k] and p[k-j] then
                Begin
                    x[k] := j;
                    a[j] := false;
                    c[j+k] := false;
                    p[k-j] := false;
                    if k = n then INKQ else THU(k+1);
                    a[j] := true;
                    c[k+j] := true;
                    p[k-j] := true;
                End;
END;
(*--------------*)
BEGIN
    clrscr;
    KHOI_TAO;
    THU(1);
    write('Tong so co', d:3, 'phuong an');
    readln;
END.
```

BÀI TẬP

1. Dùng định nghĩa chứng minh rằng:

a) $f(x) = x^4 + 9x^3 + 4x + 7$ là $O(x^4)$;

b) $f(x) = 2^x + 17$ là $O(3^x)$.

2. Tìm số nguyên nhỏ nhất n sao cho hàm f(x) là $O(x^n)$:

a) $f(x) = 2x^3 + x^2 \log x$;

b) $f(x) = \dfrac{x^4 + x^2 + 1}{x^3 + 1}$.

3. Chứng minh rằng $f(x) = x^2 + 4x + 17$ là $O(x^3)$, nhưng x^3 không là $O(x^2 + 4x + 17)$.

4. Chứng minh nếu $f_1(x)$ và $f_2(x)$ là $O(g(x))$ thì $(f_1 + f_2)(x)$ là $O(g(x))$.

5. Thuật toán thông thường để đánh giá một đa thức

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

tại x = c có thể được mô tả như sau:

```
Procedure DA_THUC (c, a_o, a_1, ..., a_n: real);
    power := 1;
    y := a_o;
    for i := 1 to n do
        begin
            power := power *c
            y := y + a_i * power
        end {y = a_nc^n + a_(n-1)c^(n-1) +... + a_1c + a_o}.
    {*ở đây giá trị cuối cùng của y chính là giá trị của đa thức tại x = c*}
```

a) Đánh giá $3x^2 + x + 1$ tại $x = 2$ bằng cách thực hiện từng bước của thuật toán trên.

b) Có bao nhiêu phép nhân và phép cộng đã được sử dụng để đánh giá đa thức bậc n tại x = c? (không kể các phép cộng được dùng để tăng biến của vòng lặp).

6. Dùng thuật toán Euclid tìm:

a) ƯCLN(12, 18);

b) ƯCLN(1001, 1331).

7. Thuật toán nhân ma trận: *Giả sử $A^{m\times k} = [a_{ij}]$ là ma trận cấp $m \times k$ với phần tử ở hàng i, cột j là a_{ij}. Tương tự $B^{k\times n} = [b_{ij}]$ là ma trận cấp $k \times n$ với phần tử ở hàng i, cột j là b_{ij}.*

Khi đó $A^{m\times k}.B^{k\times n} = C^{m\times n} = [c_{ij}]$ là ma trận cấp $m \times n$, với

$$c_{ij} = \sum_{q=1}^{k} a_{iq}b_{qj}\,.$$

Thuật toán nhân ma trận có thể mô tả như sau:

```
Procedure NHAN_MA_TRAN (A, B: ma trận)
    for i := 1 to m do
        for j := 1 to n do
            begin
                c_ij := 0
                for q := 1 to k do
                    c_ij := c_ij + a_iq b_qj
            end {C = [c_ij] là tích của A và B}.
```

a) Có bao nhiêu phép cộng và phép nhân các số nguyên được dùng trong thuật toán trên để nhân hai ma trận có các phần tử là các số nguyên?

b) Cho A_1, A_2, A_3 tương ứng là các ma trận cấp 30×20, 20×40 và 40×10 với các phần tử đều là số nguyên. Hỏi phải nhân A_1, A_2 và A_3 theo trình tự như thế nào để số các phép nhân là ít nhất?

8. Ma trận lôgic:

Ma trận mà các phần tử là 0 hoặc 1 gọi là ma trận lôgic.

Cho ma trận lôgic $A^{m\times n} = [a_{ij}]$ và $B^{m\times n} = [b_{ij}]$. Ta ký hiệu $A^{m\times n} \vee B^{m\times n}$ là ma trận lôgic hợp giữa $A^{m\times n}$ với $B^{m\times n}$ mà phần tử ở hàng i, cột j là

$$a_{ij} \vee b_{ij} = \begin{cases} 1 \text{ nếu } a_{ij} = 1 \text{ hoặc } b_{ij} = 1 \\ 0 \text{ nếu } a_{ij} = b_{ij} = 0 \end{cases}$$

Tương tự $A^{m\times n} \wedge B^{m\times n}$ là ma trận lôgic giao giữa $A^{m\times n}$ và $B^{m\times n}$ mà phần tử ở hàng i, cột j là: $a_{ij} \wedge b_{ij} = \begin{cases} 1 \text{ nếu } a_{ij} = b_{ij} = 1 \\ 0 \text{ ngược lại} \end{cases}$

Giả sử $A^{m\times n} = [a_{ij}]$ và $B^{n\times k} = [b_{ij}]$ là ma trận lôgic cấp $m \times n$ và $n \times k$ tương ứng. Khi đó tích boole của $A^{m\times k}$ và $B^{k\times n}$ được ký hiệu $A^{m\times k} \otimes B^{k\times n}$ là ma trận cấp $m \times n$ mà phần tử ở hàng i, cột j là c_{ij} được định nghĩa như sau:

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \ldots \vee (a_{ik} \wedge b_{kj})$$

Thuật toán tính tích boole như sau:

```
Procedure TICH_BOOLE (A, B: các ma trận lôgic)
    for i := 1 to m do
        for j := 1 to n do
            begin
                cij := 0
                for q := 1 to k do
                    cij := cij ∨ (aiq ∧ bqj)
            end {C = [cij] là tích boole của A và B}.
```

a) Có bao nhiêu phép toán bit được dùng để tính $A \otimes B$ với A, B là các ma trận lôgic vuông cấp n.

b) Cho A là ma trận lôgic cấp $m \times p$, B là ma trận lôgic cấp $p \times k$ va C là ma trận lôgic cấp $k \times n$.

Chứng minh rằng: $A \otimes (B \otimes C) = (A \otimes B) \otimes C$.

9. Để giải bài toán tìm trong dãy số $a_1, a_2, \ldots, a_n$ một phần tử có giá trị bằng một số cho trước, có thể áp dụng thuật toán sau:

```
input : n và dãy số a1, a2, ..., an;
output : vị trí phần tử có giá trị key hoặc n + 1 nếu không tìm thấy
Function Linear-Search (a, n, key);
    begin
        i := 0;
        repeat
            i := i + 1;
        until (i > n) or (key = ai);
        Linear - Search := i;
    end.
```

Hãy đánh giá thời gian tồi nhất và trung bình của thuật toán trên với đầu vào độ dài n.

10. Để định nghĩa một hàm số xác định trên tập các số nguyên không âm ta xây dựng như sau:

1) Giá trị của hàm tại n = 0;

2) Công thức tính giá trị của hàm đó tại số nguyên n từ các giá trị của nó tại các số nguyên nhỏ hơn n.

Định nghĩa như trên gọi là định nghĩa đệ quy.

Giả sử f được định nghĩa bằng đệ quy như sau:

$$f(0) = 3,\ f(n + 1) = 2f(n) + 3.$$

Hãy tính f(1), f(2), f(3) và f(4).

11. Cho định nghĩa đệ quy của hàm giai thừa F(n) = n! và của hàm

$$F(n) = \sum_{k=1}^{n} a_k .$$

12. Định nghĩa tập hợp bằng đệ quy: Trước tiên đưa ra tập xuất phát, sau đó là quy tắc tạo các phần tử mới từ các phần tử đã biết của tập hợp.

Định nghĩa tập hợp A bằng đệ quy như sau:

$$3 \in A;\ x + y \in A \text{ nếu } x \text{ và } y \text{ thuộc } A.$$

Hãy chỉ ra A là tập các số nguyên chia hết cho 3.

13. Tìm thuật toán đệ quy tính giá trị a^n với a là số thực khác không và n là số nguyên không âm.

14. Tìm thuật toán đệ quy tính ƯCLN của hai số nguyên a, b không âm và a < b.

15. Hãy biểu diễn thuật toán tìm kiếm tuyến tính như một thủ tục đệ quy.

Chương 2
CÁC PHƯƠNG PHÁP ĐẾM

§1. TẬP HỢP VÀ BIỂU DIỄN TẬP HỢP TRÊN MÁY TÍNH

1.1. Các phép toán trên tập hợp

Cho A, B là các tập hợp.

a) Phép hợp (ký hiệu $\cup$)

$$A \cup B = \{x : x \in A \text{ hoặc } x \in B\}$$

b) Phép giao (ký hiệu $\cap$)

$$A \cap B = \{x : x \in A \text{ và } x \in B\}$$

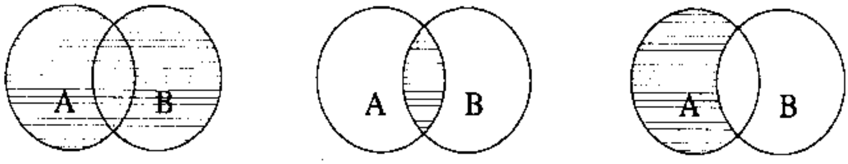
c) Hiệu hai tập hợp (ký hiệu \ hay –)

$$A \setminus B = \{x : x \in A \text{ và } x \notin B\}$$

Đặc biệt khi $A \subseteq B$, phần bù của A ký hiệu là $\overline{A}$ và được định nghĩa như sau:

$\overline{A} = \{x : x \in B \text{ và } x \notin A\}$ gọi là tập vũ trụ.

Sau đây là hình ảnh minh hoạ các phép toán trên:



$A \cup B$ $\quad$ $A \cap B$ $\quad$ $A \setminus B$

Ví dụ 1:

a) Cho: $A = \{1, 2, 3, 4\}$; $B = \{5, 6, 7, 8\}$

$A \cup B = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

b) Cho: $C = \{a, b, c\}$; $D = \{c, d, e\}$

$C \cup D = \{a, b, c, d, e\}$.

Ví dụ 2: A, B, C, D như ví dụ 1.

$A \cap B = \varnothing$ (tập rỗng)

$C \cap D = \{c\}$.

Ví dụ 3: A, B, C, D như ví dụ 1.

$A \setminus B = \{1, 2, 3, 4\} \setminus \{5, 6, 7, 8\} = \{1, 2, 3, 4\} = A$

$C \setminus D = \{a, b, c\} \setminus \{c, d, e\} = \{a, b\}$.

Ví dụ 4: Cho $A = \{1, 2, 3, 4, 5\}$; $B = \{4, 5\}$ $(B \subset A)$

$\overline{B} = \{1, 2, 3\}$.

1.2. Các tính chất của tập hợp

Giả sử $A, B \subseteq X$. Khi đó ta có các tính chất sau:

– *Tính giao hoán:* $A \cup B = B \cup A$; $A \cap B = B \cap A$.

– *Tính kết hợp:* $A \cup (B \cup C) = (A \cup B) \cup C$;

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

– *Tính De Morgan:* $\overline{A \cup B} = \overline{A} \cap \overline{B}$; $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

– *Tính phân bố:* $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$;

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

– *Tính trung hòa:* $A \cup \varnothing = A$; $A \cap X = A$.

– *Tính phần tử bù:* $A \cup \overline{A} = X$; $A \cap \overline{A} = \varnothing$.

– *Tính thống trị:* $A \cup X = X$; $A \cap \varnothing = \varnothing$.

1.3. Lực lượng của tập hợp

Ta ký hiệu *lực lượng* của tập A là $|A|$ và định nghĩa:

$$|A| = \text{số phần tử của } A.$$

Khi đó ta có ba công thức thường gặp khi phải tính số phần tử của một tập hợp:

1) $|A \cup B| = |A| + |B| - |A \cap B|$;

2) $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$;

3) $|A \cup B \cup C \cup D| = |A| + |B| + |C| + |D| - |A \cap B| - |A \cap C| - |A \cap D| - |B \cap C| - |B \cap D| - |C \cap D| + |A \cap B \cap C| + |A \cap B \cap D| + |A \cap C \cap D| + |B \cap C \cap D| - |A \cap B \cap C \cap D|$.

1.4. Tích Đề-các của các tập hợp và lực lượng của nó

- $A \times B = \{(a, b) : a \in A; b \in B\}$ và lực lượng của nó là

$$|A \times B| = |A|.|B|.$$

- $A \times B \times C = \{(a, b, c) : a \in A; b \in B; c \in C\}$ và lực lượng của nó là

$$|A \times B \times C| = |A|.|B|.|C|.$$

- $|A_1 \times A_2 \times ... \times A_n| = \{(a_1, a_2, ..., a_n) \mid a_i \in A_i\ (i = \overline{1, n})\}$ và lực lượng của nó là

$$|A_1 \times A_2 \times ... \times A_n| = |A_1|.|A_2| ... |A_n| = \prod_{i=1}^{n} |A_i|.$$

1.5. Biểu diễn các tập hợp trên máy tính

Có nhiều cách biểu diễn tập hợp trên máy tính. Dưới đây giới thiệu một cách biểu diễn tập hợp trên máy tính bằng cách lưu trữ các phần tử của nó dưới dạng sắp tuỳ ý các phần tử của tập vũ trụ.

Giả sử X là một tập vũ trụ và $A \subseteq X$ (với giả thiết dung lượng bộ nhớ của máy tính không bé hơn lực lượng của X).

Giả sử $|X| = n$, khi đó ta sắp (đánh số) các phần tử của $X = \{a_1, a_2, ..., a_n\}$. Ta có thể biểu diễn tập A trên máy tính bằng một xâu bit có chiều dài n, trong đó bit thứ i là 1 nếu $a_i \in A$, còn bit thứ i là 0 nếu $a_i \notin A$ ($i = \overline{1, n}$).

Ví dụ 5: Cho X = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} (sắp xếp các phần tử của X theo thứ tự tăng dần).

a) Xác định xâu bit của tập $A = \{1, 3, 5, 7, 9\} \subset X$.

b) Xác định xâu bit của tập $B = \{2, 4, 6, 8, 10\} \subset X$.

c) Xác định xâu bit các phần tử không vượt quá 5 trong X, tức là tìm xâu bit của C = {1, 2, 3, 4, 5}.

Giải:

a) Xâu bit của tập A = {1, 3, 5, 7, 9} là 1010101010.

b) Xâu bit của tập B = {2, 4, 6, 8, 10} là 0101010101.

c) Xâu bit của tập C = {1, 2, 3, 4, 5} là 1111100000.

Ví dụ 6: Cho X = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} và

A = {1, 3, 5, 7, 9}.

B = {2, 4, 6, 8, 10} và C = {1, 2, 3, 4, 5} như trong ví dụ 5.

a) Tìm xâu bit của $A \cup B$.

b) Tìm xâu bit của $A \cap B$.

c) Tìm xâu bit của $\overline{A}$ và $\overline{B}$.

Giải:

Xâu bit của A là 1010101010.

Xâu bit của B là 0101010101.

a) Xâu bit của $A \cup B$ là:

$$1010101010 \vee 0101010101 = 1111111111.$$

b) Xâu bit của $A \cap B$ là:

$$1010101010 \wedge 0101010101 = 0000000000.$$

c) Vì xâu bit của $A = \{1, 3, 5, 7, 9\}$ là 1010101010 nên xâu bit của $\overline{A} = \{2, 4, 6, 8, 10\}$ là 0101010101.

Vì xâu bit của $B = \{2, 4, 6, 8, 10\}$ là 0101010101 nên xâu bit của $\overline{B} = A$ là 1010101010.

Chú ý:

– Để nhận được các xâu bit cho hợp của hai tập hợp, ta thực hiện phép tuyển ($\vee$) hai xâu bit đó với nhau.

– Để nhận được xâu bit của giao hai tập hợp, ta thực hiện phép hội ($\wedge$) hai xâu bit đó với nhau.

– Để nhận được xâu bit của phần bù tập hợp A, ta chỉ việc thay 0 bởi 1 và 1 bởi 0 trong xâu bit của A.

Ví dụ 7: Cho $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$A = \{1, 2, 3, 8\}$; $B = \{2, 4, 8, 9\}$; $C = \{6, 7, 8, 9\}$.

a) Tìm xâu bit của $A \cup B$, $A \cup B \cup C$.

b) Tìm xâu bit của $A \cap B$, $A \cap B \cap C$.

c) Tìm xâu bit của $\overline{A}$, $\overline{B}$, $\overline{C}$.

Giải:

Xâu bit của A là 111000010.

Xâu bit của B là 010100011.

Xâu bit của C là 000001111.

a) Xâu bit của $A \cup B$ là 111100011 (hay $A \cup B = \{1, 2, 3, 4, 8, 9\}$).

Xâu bit của $A \cup B \cup C$ là 111101111

(hay $A \cup B \cup C = \{1, 2, 3, 4, 6, 7, 8, 9\}$).

b) Xâu bit của $A \cap B$ là 010000010 (hay $A \cap B = \{2, 8\}$).

Xâu bit của $A \cap B \cap C$ là 000000010 (hay $A \cap B \cap C = \{8\}$).

c) Xâu bit của $\overline{A}$ là 000111101 (hay $\overline{A} = \{4, 5, 6, 7, 9\}$)

Xâu bit của $\overline{B}$ là 101011100 (hay $\overline{B} = \{1, 3, 5, 6, 7\}$)

Xâu bit của $\overline{C}$ là 111110000 (hay $\overline{C} = \{1, 2, 3, 4, 5\}$).

§2. HOÁN VỊ, CHỈNH HỢP VÀ TỔ HỢP

2.1. Hoán vị và chỉnh hợp

Hoán vị của một tập các đối tượng khác nhau là một cách sắp xếp có thứ tự các đối tượng này.

Một cách sắp xếp có thứ tự r phần tử của một tập n phần tử được gọi là *chỉnh hợp chập r* của tập n phần tử. Ký hiệu P(n, r) là số chỉnh hợp chập r của tập n phần tử. Ta có kết quả dưới đây:

Định lý 1: Số chỉnh hợp chập r của tập n phần tử là

$$P(n, r) = n(n-1)(n-2)\dots(n-r+1) \left(= \frac{n!}{(n-r)!}\right)$$

Ví dụ 1: Cho $A = \{1, 2, 3\}$. Có bao nhiêu cách sắp xếp có thứ tự hai phần tử trong tập A?

Giải:

$$P(3, 2) = 3.2 = 6.$$

Đó là các cặp: $\{1, 2\}, \{2, 1\}, \{1, 3\}, \{3, 1\}, \{2, 3\}, \{3, 2\}$.

Ví dụ 2: Có bao nhiêu cách chọn 4 cầu thủ khác nhau trong 10 cầu thủ của đội bóng.

Giải:

$$P(10, 4) = 10(10-1)(10-2)(10-3) = 10.9.8.7 = 5040.$$

Chú ý: Số chỉnh hợp chập n của tập n phần tử là $P(n, n) = n!$

Hay số hoán vị của tập gồm n phần tử là $P(n, n) = n!$.

Ví dụ 3: Giả sử có 8 vận động viên chạy thi tốc độ cự ly 2000m. Người đến đích đầu tiên được trao Huy chương Vàng, người đến đích thứ hai được trao Huy chương Bạc và người đến đích thứ ba được trao Huy chương Đồng. Hỏi có bao nhiêu cách trao Huy chương Vàng, Bạc và Đồng cho 8 vận động viên trên.

Giải: Số cách trao Huy chương Vàng, Bạc và Đồng cho 8 vận động viên chính là chỉnh hợp chập 3 của 8, hay P(8, 3) = 8.7.6 = 336 cách.

Ví dụ 4: Giả sử vận động viên đi xe đạp dự định đi qua 8 thành phố. Vận động viên bắt đầu cuộc hành trình từ một thành phố nào đó và có thể đến thành phố kia theo bất kỳ một thứ tự nào anh ta muốn. Hỏi vận động viên có thể đi qua 8 thành phố này theo bao nhiêu lộ trình khác nhau?

Giải: Số lộ trình có thể giữa các thành phố bằng số hoán vị của 7 thành phố (vì thành phố đầu tiên đã được xác định). 7 thành phố còn lại có thể có thứ tự chọn tuỳ ý. Do đó có P(7, 7) = 7! = 5040 cách để vận động có thể chọn lộ trình khác nhau cho mình.

2.2. Tổ hợp và định lý nhị thức

a) Tổ hợp

Một *tổ hợp* chập r của một tập hợp là một cách chọn không có thứ tự r phần tử của tập đã cho.

Vì vậy, một tổ hợp chập r chính là một tập con r phần tử của tập hợp ban đầu.

Số tổ hợp chập r của tập n phần tử ký hiệu là C(n, r) $(0 \le r \le n)$. Ta có kết quả sau đây:

Định lý 2: Số tổ hợp chập r của n phần tử là

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

Chứng minh:

$$P(n, r) = C(n, r).P(r, r)$$

hay $$C(n, r) = \frac{P(n, r)}{r!} = \frac{n!}{r!(n-r)!}.$$

Hệ quả 1: Cho n, r là các số nguyên không âm sao cho $r \le n$. Khi đó ta có: C(n, r) = C(n, n − r).

Chứng minh: Theo định lý 2 ta có: $C(n, r) = \dfrac{n!}{r!(n-r)!}$.

Vậy

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

Hệ quả 2 (Hằng đẳng thức Pascal):

Cho n, k là các số nguyên dương, với $n \ge k$. Khi đó ta có

$$C(n+1, k) = C(n, k-1) + C(n, k).$$

Hệ quả 3: $\sum_{k=0}^{n} C(n, k) = 2^n$ (n nguyên dương).

Chứng minh: Số các tập con của tập gồm n phần tử là

$$2^n = C(n, 0) + C(n, 1) + \ldots + C(n, n) = \sum_{k=0}^{n} C(n, k).$$

b) Định lý nhị thức

Định lý 3: Cho x, y là hai biến và n là một số nguyên dương. Khi đó ta có

$$(x+y)^n = \sum_{i=0}^{n} C(n, i)x^{n-i}y^i$$

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

Ví dụ 5: Cho A = {a, b, c, d}. Tìm C(4, 2), C(4, 3).

Giải: Ta có: $C(4, 2) = \frac{4!}{2!(4-2)!} = \frac{1.2.3.4}{1.2.1.2} = 6$.

Cụ thể là các cặp: {a, b}, {a, c}, {a, d}, {b, c}, {b, d}, {c, d}.

$$C(4, 3) = \frac{4!}{3!(4-3)!} = 4.$$

Cụ thể là: {a, b, c}, {a, b, d}, {a, c, d}, {b, c, d}.

Ví dụ 6: Có bao nhiêu cách tuyển 5 trong số 10 cầu thủ của một đội quần vợt để đi thi đấu?

Giải: Đó chính là $C(10, 5) = \frac{10!}{5!(10-5)} = 252$.

Ví dụ 7: Cho A = {a, b, c} và B = {a, b, c, d}. Hãy liệt kê tất cả các hoán vị của A và của B.

Giải:

Đối với A có 3! = 6 hoán vị là ; abc, acb, cab, cba, bac, bca.

Đối với B có 4! = 24 hoán vị là: abcd, acbd, adbc, ...

Ví dụ 8: Giả sử A = {1, 2, 3, 4, 5}.

a) Tìm các chỉnh hợp chập 3 của A.

b) Tìm các tổ hợp chập 3 của A.

Giải:

a) P(5, 3) = 5.4.3 = 60.

b) $C(5, 3) = \frac{5!}{3!2!} = \frac{4.5}{2} = 10$.

Ví dụ 9: Tìm giá trị của các đại lượng sau:

a) P(6, 3); c) P(8, 1); e) P(8, 8); i) C(5, 1); n) C(8,4)

b) P(6, 5); d) P(8, 5) f) P(10, 9); j) C(5, 3); m) C(12,6)

Giải:

a) P(6, 3) = 6.5.4 = 120.

b) P(6, 5) = 6.5.4.3.2 = 720.

c) P(8, 1) = 8.

d) P(8, 5) = 8.7.6.5.4 = 6720.

e) P(8, 8) = 6720.3.2.1 = 45720.

f) P(10, 9) = 10.9.8.7.6.5.4.3.2 = 362880.

i) $C(5, 1) = \frac{5!}{4!} = 5$.

j) $C(5, 3) = \frac{5!}{3!2!} = \frac{20}{2} = 10$.

n) $C(8, 4) = \frac{8!}{4!\,4!} = \frac{5.6.7.8}{1.2.3.4} = \frac{5.3.7.2}{3} = 70$.

m) $C(12, 6) = \frac{12!}{6!\,6!} = \frac{7.8.9.10.11.12}{2.3.4.5.6} = 924$.

Ví dụ 10: Tính số chỉnh hợp chập 5 của tập 9 phần tử.

Giải: P(9, 5) = 9.8.7.6.5 = 15120.

Ví dụ 11: Có bao nhiêu khả năng có thể xảy ra đối với các vị trí thứ nhất, thứ hai và thứ ba trong cuộc đua có 12 con ngựa, nếu mọi thứ tự tới đích đều có thể xảy ra?

Giải: Số khả năng cần tìm chính là P(12, 3) = 1320.

Ví dụ 12: Có bao nhiêu cách chọn một tập hợp 5 chữ từ bảng chữ cái tiếng Việt?

Giải: Số cách chọn chính là:

$$C(26,5) = \frac{26!}{5!\,21!} = \frac{22.23.24.25.26}{5!} = \frac{7893600}{120} = 65780.$$

Ví dụ 13: Một tập hợp 100 phần tử có bao nhiêu tập con có nhiều hơn hai phần tử?

Giải:

Theo hệ quả 3 ta có: $\sum_{i=0}^{100} C(n, i) = 2^{100}$, hay

$$2^{100} = C(100, 0) + C(100, 1) + C(100, 2) + C(100, 3) + ... + C(100, 100).$$

Số các tập con của tập 100 phần tử có nhiều hơn hai phần tử là

$$2^{100} - \left(\frac{100!}{0!\,100!} + \frac{100!}{1!\,99!} + \frac{100!}{2!\,98!}\right) = 2^{100} - \left(1 + 100 + \frac{99.100}{2}\right)$$

$$= 2^{100} - (101 + 4950) = 2^{100} - 5051.$$

Ví dụ 14: Có 100 vé đánh số từ 1 đến 100 được bán cho 100 người khác nhau. Người ta sẽ trao 4 giải thưởng kể cả giải độc đắc. Hỏi:

a) Có bao nhiêu cách trao thưởng?

b) Có bao nhiêu cách trao thưởng nếu người giữ vé 47 trúng giải độc đắc?

Giải:

a) Số cách trao thưởng là

$$P(100, 4) = 100.99.98.97 = 94109400.$$

b) Số cách trao thưởng là P(99, 3) = 99.98.97 = 941094.

Ví dụ 15: Một câu lạc bộ có 25 thành viên.

a) Có bao nhiêu cách chọn 4 thành viên vào ủy ban thường trực?

b) Có bao nhiêu cách chọn chủ tịch, phó chủ tịch, thư ký và thủ quỹ?

Giải:

a) Có $C(25, 4) = \dfrac{25!}{4!\,21!} = \dfrac{22.23.24.25}{24} = 22.23.25 = 12650$ cách chọn 4 thành viên vào ban thường trực.

b) Có $P(4, 4).C(25, 4) = 24.12650 = 303600$ cách chọn chủ tịch, phó chủ tịch, thư ký và thủ quỹ.

Ví dụ 16: Giả sử một tổ bộ môn có 10 nam và 15 nữ. Có bao nhiêu cách chọn một hội đồng gồm 6 ủy viên, trong đó số ủy viên nam bằng số ủy viên nữ?

Giải: Có $C(10, 3).C(15, 3) = 120.455 = 54600$ cách chọn.

Ví dụ 17: Tìm khai triển của $(x + y)^5$.

Giải: Ta có

$$(x + y)^5 = C(5, 0)x^5 + C(5, 1)x^4y + C(5, 2)x^3y^2 + C(5, 3)x^2y^3 + C(5, 4)xy^4 + C(5, 5)y^5$$

với $C(5, 0) = 1$, $C(5, 1) = 5$, $C(5, 2) = 10$.

$$(x + y)^5 = x^5 + 5x^4y + 10x^3y^2 + 10x^2y^3 + 5xy^4 + y^5$$

Ví dụ 18: Trong khai triển $(x + y)^{100}$ có bao nhiêu số hạng?

Giải: Có 101 số hạng là $C(100, 0), C(100, 1), \ldots C(100, 100)$.

Ví dụ 19: Tìm hệ số của x^9 trong khai triển của $(2 - x)^{19}$.

Giải:

$$(2 - x)^{19} = C(19, 0)2^{19} + C(19, 1)2^{18}(-x) + C(19, 2)2^{17}(-x)^2 + \ldots + C(19, 9)2^{10}(-x)^9 + \ldots$$

Vậy số hạng của x^9 là: $-C(19, 9)2^{10} = -94595072$.

Ví dụ 20: Tìm hệ số của $x^{101}y^{99}$ trong khai triển của $(2x - 3y)^{200}$.

Giải:

Số hạng thứ 100 trong khai triển $(2x - 39)^{200}$ là

$$C(200, 99)(2x)^{101}(-39)^{99} = 2^{101}.(-3)^{99}C(200, 99)x^{101}y^{99}.$$

Vậy hệ số của $x^{101}y^{99}$ là: $-2^{101}.3^{99}.C(200, 99)$.

§3. CÁC QUY TẮC ĐẾM CƠ BẢN

3.1. Quy tắc cộng

a) Cơ sở của quy tắc cộng là một trong các công thức thường gặp sau đây:

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

hoặc $$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|.$$

b) Quy tắc cộng có thể phát biểu trong hai trường hợp sau:

• Giả sử có hai công việc. Việc thứ nhất có thể làm bằng n_1 cách; việc thứ hai có thể làm bằng n_2 cách và nếu hai việc không thể làm đồng thời thì để làm hai công việc trên ta cần $n_1 + n_2$ cách. Trường hợp này ứng với $A \cap B = \varnothing$ hay $|A \cap B| = 0$.

• Trường hợp giữa hai công việc có k cách làm đồng thời thì để làm hai công việc trên ta có $n_1 + n_2 - k$ cách làm.

Trường hợp này ứng với $A \cap B \neq \varnothing$ và $|A \cap B| = k$.

Chú ý: Quy tắc cộng có thể mở rộng cho n công việc, $n \geq 1$.

Ví dụ 1: Một sinh viên có thể chọn bài thực hành trên máy tính từ 4 danh sách. Danh sách thứ nhất có 23 bài thực hành; danh sách thứ 2 có 19 bài thực hành; danh sách thứ 3 có 15 bài thực hành và danh sách thứ 4 có 20 bài thực hành. Biết các bài thực hành trong các danh sách là khác nhau. Hỏi có bao nhiêu cách chọn bài thực hành trên máy tính?

Giải: Có 23 + 19 + 15 + 20 = 77 cách chọn.

Ví dụ 2: Trong lớp Công nghệ thông tin (CNTT) khoá K46 có 45 sinh viên học tiếng Anh; 30 sinh viên học tiếng Pháp và 10 sinh viên học cả tiếng Anh và pháp.

a) Tính số sinh viên CNTT K46, biết trong lớp không ai không biết một trong hai thứ tiếng trên.

b) Cho biết sĩ số của lớp là 70. Hỏi có bao nhiêu sinh viên không biết ngoại ngữ Anh, Pháp.

Giải:

Đặt: A là số sinh viên học tiếng Anh: $|A| = 45$;

B là số sinh viên học tiếng Pháp: $|B| = 30$.

$A \cap B$ là số sinh viên học tiếng Anh và Pháp: $|A \cap B| = 10$.

a) Theo công thức cơ sở ta có

$$|A \cup B| = |A| + |B| - |A \cap B| = 45 + 30 - 10 = 65.$$

Số sinh viên của K46 ngành CNTT là 65.

b) $|A \cup B| = 65$, đây là số sinh viên học ngoại ngữ Anh, Pháp hoặc cả Anh và Pháp. Số sinh viên không học ngoại ngữ là $70 - 65 = 5$.

3.2. Quy tắc nhân

a) Cơ sở của quy tắc nhân

- $A \times B = \{(a, b) : a \in A \text{ và } b \in B\}$;
  $|A \times B| = |A|.|B|$.
- $A \times B \times C = \{(a, b, c) : a \in A, b \in B, c \in C\}$;
  $|A \times B \times C| = |A|.|B|.|C|$.

b) Quy tắc nhân phát biểu như sau

Nếu một quá trình có thể thực hiện theo hai giai đoạn liên tiếp nhau, sao cho có n_1 cách khác nhau để thực hiện giai đoạn 1 và với mỗi cách lựa chọn trong giai đoạn 1 đều có n_2 cách khác nhau để thực hiện giai đoạn 2. Khi ấy có $n_1.n_2$ cách khác nhau để thực hiện toàn bộ quá trình.

Ví dụ 3: Để chuẩn bị mở đại diện văn phòng ở nước ngoài, giám đốc Công ty X cần chọn một luật sư trong 5 luật sư của công ty và chọn một cố vấn địa ốc trong 3 cố vấn địa ốc của công ty đi làm việc tại văn phòng đại diện ở nước ngoài. Hỏi có bao nhiêu cách chọn hai đại diện theo nguyên tắc trên làm việc ở văn phòng đại diện nước ngoài?

Giải:

A là tập gồm 5 luật sư của công ty.

B là tập gồm 3 cố vấn địa ốc của công ty.

Khi đó một cặp có thứ tự (a, b) với $a \in A$, $b \in B$ là một phương án chọn. Vậy số cách chọn là $|A \times B| = |A|.|B| = 5.3 = 15$ cách chọn hai đại diện (1 luật sư, 1 cố vấn địa ốc) đi làm việc ở văn phòng đại diện nước ngoài.

Chú ý: Quy tắc nhân có thể mở rộng cho quá trình có nhiều hơn hai giai đoạn.

3.3. Một số bài toán đếm kết hợp giữa quy tắc cộng và quy tắc nhân

Ví dụ 4: Để chuẩn bị vào giai đoạn 2, có 150 sinh viên ghi tên học môn Lôgic toán; 120 sinh viên ghi tên học môn Lý thuyết đồ thị và 200 sinh viên ghi tên học môn Văn phạm và ôtômat. Hỏi có bao nhiêu sinh viên ghi tên học một trong ba môn, biết rằng không có sinh viên nào ghi tên học đồng thời 2 môn hoặc cả 3 môn.

Giải:

A := Tập sinh viên học môn Lôgic toán $\Rightarrow |A| = 150$.

B := Tập sinh viên học môn Lý thuyết đồ thị $\Rightarrow |B| = 120$.

C := Tập sinh viên học môn Văn phạm và ôtômat $\Rightarrow |C| = 200$.

Ta có: $A \cap B = \varnothing$, $A \cap C = \varnothing$, $B \cap C = \varnothing$, $A \cap B \cap C = \varnothing$.

Số sinh viên ghi tên học một trong ba môn là:

$$|A \cup B \cup C| = |A| + |B| + |C| = 470.$$

Ví dụ 5: Ghi nhãn cho chiếc ghế trong hội trường bằng một chữ cái và một số nguyên dương không vượt quá 100. Hỏi có bao nhiêu chiếc ghế được ghi nhãn?

Giải:

Thủ tục gán nhãn gồm hai việc: Việc thứ nhất là chọn một chữ cái trong tập A = {26 chữ cái}; việc thứ hai là chọn một chữ số trong các chữ số không vượt quá 100 có thể chọn trong tập $B := \{1, 2, ..., 100\}$. Khi đó nhãn trên ghế là cặp (a, b) với $a \in A$, $b \in B$. Vậy số ghế có thể gán nhãn là:

$$|A \times B| = |A|.|B| = 26.100 = 2600 \text{ chiếc.}$$

Ví dụ 6: Trong một trung tâm máy tính có 50 máy tính. Mỗi máy có 24 cổng. Hỏi có bao nhiêu cổng khác nhau trong trung tâm này.

Giải:

Thủ tục chọn cổng gồm hai việc: Việc chọn máy, sau đó là chọn cổng của chiếc máy này.

Có 50 khả năng chọn máy; mỗi máy có 24 cổng nên số cổng của trung tâm này là: $50 \times 24 = 1200$ cổng.

Ví dụ 7: Có bao nhiêu xâu nhị phân độ dài:

a) Bằng n?

b) Nhỏ hơn hoặc bằng n?

Giải:

a) Xâu nhị phân độ dài n là $a_1a_2...a_n$.

a_i hoặc là 1 hoặc là 0 nên số xâu nhị phân độ dài n là 2^n.

b) Xâu nhị phân có độ dài 1 là a, $a \in \{0, 1\}$ nên có 2 xâu.

Xâu nhị phân có độ dài 2 là a_1a_2 có $2^2 = 4$ xâu.

Xâu nhị phân có độ dài 3 là $a_1a_2a_3$ có $2^3 = 8$ xâu.

...

Xâu nhị phân có độ dài n có 2^n xâu.

Vậy số xâu nhị phân có độ dài $\le$ n là: $2 + 2^2 + 2^3 + ... + 2^n$.

Ví dụ 8: Một dãy XXXYYY độ dài 6. X có thể gán bởi một chữ cái, Y có thể gán một chữ số. Có bao nhiêu dãy được thành lập theo cách trên?

Giải:

Có 26 chữ cái và 10 chữ số. Vậy số dãy được thành lập theo cách trên là: $26^3.10^3 = 17576000$.

Ví dụ 9: Có thể tạo được bao nhiêu hàm số từ tập A có m phần tử vào tập B có n phần tử?

Giải:

Theo định nghĩa một hàm số $f : A \rightarrow B$ là một tương ứng mỗi phần tử $x \in A$ bởi một phần tử nào đó $y \in B$ sao cho $y = f(x)$.

Rõ ràng, với mỗi $x \in A$ ta có n cách chọn $y \in B$ để $y = f(x)$. Mặt khác n có m cách chọn. Vậy số hàm từ tập A vào B là: $\underbrace{n.n...n}_{m} = n^m$.

Ví dụ 10: Trong tương lai số điện thoại cố định gồm 10 chữ số, trong đó 3 chữ số đầu NXX là mã vùng, ba chữ số tiếp theo là mã chi nhánh có dạng NXX, 4 chữ số còn lại là XXXX là mã máy. Biết N có thể nhận chữ số từ 2 đến 9, còn X nhận các số từ 0 đến 9. Hỏi có bao nhiêu số điện thoại khác nhau theo cách trên?

Giải:

Số điện thoại có dạng NXX.NXX.XXXX.

Vậy có tất cả:

$$\underbrace{8.10.10}_{800}.\underbrace{8.10.10}_{800}.10^4 = 800.800.10000 = 6400000000$$ số điện thoại.

Ví dụ 11: Cho tập A, với | A | = n. Dùng quy tắc nhân, hãy chỉ ra số tập con của tập A là 2^n.

Giải:

Giả sử $A = \{a_1, a_2, a_3, ..., a_n\}$.

Xét dãy $a_1, a_2, a_3, ..., a_n$, lấy tập con bất kỳ $B \subseteq A$. Ta có thể biểu diễn trên máy tập B bởi xâu bit độ dài n, trong đó bit thứ i là 1 khi $a_i \in B$; là 0 khi $a_i \notin B$. Mỗi tập con tương ứng với một xâu bit độ dài n. Số tập con của A tương ứng với số xâu bit độ dài n là : $\underbrace{2.2...2}_{n} = 2^n$.

Ví dụ 12: Mỗi người sử dụng máy tính đều có mật khẩu dài từ 6 đến 8 ký tự. Trong đó mỗi ký tự là một chữ hoa hay một chữ số. Mỗi mật khẩu phải chứa ít nhất một chữ số. Hỏi có bao nhiêu mật khẩu?

Giải:

Gọi P là tổng số mật khẩu; P_6, P_7, P_8 là mật khẩu có độ dài 6, 7, 8 tương ứng. Theo quy tắc cộng ta có:

$$P = P_6 + P_7 + P_8.$$

Theo quy tắc nhân:

$$P_6 = (26 + 10)^6 - 26^6 = 1.867.866.560;$$

$$P_7 = (26 + 10)^7 - 26^7 = 70.332.353.920;$$

$$P_8 = (26 + 10)^8 - 26^8 = 2.612.282.842.880.$$

Vậy $P = P_6 + P_7 + P_8 = 2.684.063.360$.

Ví dụ 13: Trong một trường đại học có 18 sinh viên xuất sắc về Toán và 325 sinh viên xuất sắc về CNTT.

a) Có bao nhiêu cách chọn hai đại diện, sao cho một là sinh viên Toán, còn người kia là sinh viên CNTT?

b) Có bao nhiêu cách chọn một đại diện hoặc là sinh viên Toán hoặc là sinh viên CNTT?

Giải:

a) Có 325.18 = 5850 cách.

b) Có 325 + 18 = 343 cách.

Ví dụ 14: Một phiếu trắc nghiệm đa lựa chọn gồm 10 câu hỏi. Mỗi câu hỏi có 4 phương án trả lời.

a) Có bao nhiêu cách điền một phiếu trắc nghiệm nếu mọi câu hỏi đều được trả lời.

b) Có bao nhiêu cách điền một phiếu trắc nghiệm nếu câu hỏi có thể bỏ trống?

Giải:

a) Có 4^{10} cách.

b) Có 5^{10} cách.

Ví dụ 15: Có bao nhiêu người có tên họ viết tắt bằng 3 chữ cái.

Giải:

Tên họ viết tắt bằng 3 chữ cái có dạng XYZ. Vậy có 26^3 người có tên họ viết tắt bằng 3 chữ cái.

Ví dụ 16: Có bao nhiêu xâu nhị phân độ dài 10 có bit đầu tiên và bit cuối cùng là 1?

Giải:

Xâu nhị phân có dạng: $1a_1a_2a_3a_4a_5a_6a_7a_81$, $a_i \in \{0, 1\}$.

Có 2^8 xâu nhị phân như vậy.

Ví dụ 17: Có bao nhiêu xâu nhị phân có độ dài không vượt quá n và chứa toàn số 1, trong đó n là nguyên dương?

Giải:

Các xâu nhị phân có độ dài không vượt quá n gồm các xâu:

$1, 11, 111, \ldots \underbrace{11\ldots1}_{n}$ hay $1, 1^2, 1^3, \ldots, 1^n$. Có n xâu như vậy.

Ví dụ 18: Mỗi sinh viên lớp Toán rời rạc (TRR) hoặc là giỏi Toán, hoặc là giỏi Tin, hoặc giỏi cả hai môn này. Trong lớp có bao nhiêu sinh viên nếu có 38 người giỏi Tin, 23 người giỏi Toán và 7 người giỏi cả hai môn?

Giải:

A := số sinh viên giỏi Tin $\Rightarrow |A| = 38$.

B := số sinh viên giỏi Toán $\Rightarrow |B| = 23$.

$A \cap B$ = số sinh viên giỏi Tin và Toán $\Rightarrow |A \cap B| = 7$.

Ta có: $|A \cup B| = |A| + |B| - |A \cap B| = 38 + 23 - 7 = 54$.

Vậy số sinh viên của lớp là 54.

Ví dụ 19: Giả sử trong Khoa Công nghệ có 1807 sinh viên năm thứ nhất, trong số này có 453 sinh viên chọn môn Tin học; 567 chọn môn Toán và 299 chọn cả hai môn Toán và Tin. Hỏi có bao nhiêu sinh viên không học Toán cũng không học Tin?

Giải:

A := tập các sinh viên học Tin ⇒ $|A| = 453$.

B := tập các sinh viên học Toán ⇒ $|B| = 567$.

$A \cap B$ = tập các sinh viên học cả Tin và Toán ⇒ $|A \cap B| = 299$.

Số sinh viên học Toán hoặc Tin hoặc cả Toán và Tin là

$$|A \cup B| = |A| + |B| - |A \cap B| = 453 + 567 - 299 = 721.$$

Vậy số sinh viên không học Tin lẫn Toán là

$$1807 - 721 = 1086.$$

Ví dụ 20: Biết rằng có 1232 sinh viên học tiếng Tây Ban Nha; 879 sinh viên học tiếng Pháp và 114 sinh viên học tiếng Nga; 103 sinh viên học cả Tây Ban Nha và Pháp; 23 sinh viên học cả Tây Ban Nha và Nga; 14 sinh viên học cả Pháp và Nga. Nếu tất cả 2092 sinh viên đều theo học ít nhất một ngoại ngữ thì có bao nhiêu sinh viên học cả ba thứ tiếng?

Giải:

Gọi: S := tập sinh viên học tiếng Tây Ban Nha ⇒ $|S| = 1232$;

F := tập sinh viên học tiếng Pháp ⇒ $|F| = 879$;

R := tập sinh viên học tiếng Nga ⇒ $|R| = 114$.

Khi đó: $S \cap F$ = tập sinh viên học cả tiếng Tây Ban Nha và Pháp

⇒ $|S \cap F| = 23$;

$S \cap R$ = tập sinh viên học cả tiếng Tây Ban Nha và Nga

⇒ $|S \cap R| = 23$;

$F \cap R$ = tập sinh viên học cả tiếng Pháp và Nga

⇒ $|F \cap R| = 14$;

$S \cap F \cap R$ = tập sinh viên học cả ba thứ tiếng

⇒ $|S \cap F \cap R| = x$.

Ta có:

$$\begin{aligned}|S \cup F \cup R| &= |S| + |F| + |R| - |S \cap F| - |S \cap R| - |F \cap R| \\ &\quad + |S \cap F \cap R| \\ &= 1232 + 879 + 114 - 103 - 23 - 14 + x = 2092\end{aligned}$$

hay $2085 + x = 2092 \Rightarrow x = 7$.

Vậy có 7 sinh viên học cả ba thứ tiếng.

Ví dụ 21: Tập $A \cup B$ có bao nhiêu phần tử nếu A có 12 phần tử, B có 18 phần tử và:

a) $A \cap B = \varnothing$?

b) $|A \cap B| = 1$?

c) $|A \cap B| = 6$?

d) $A \subseteq B$?

Giải:

a) $|A \cup B| = |A| + |B| - |A \cap B| = 12 + 18 = 30$.

b) $|A \cup B| = |A| + |B| - |A \cap B| = 12 + 18 - 1 = 29$.

c) $|A \cup B| = |A| + |B| - |A \cap B| = 12 + 18 - 6 = 24$

d) $|A \cup B| = |A| + |\bar{A}| = 12 + 6 = 18$.

Ví dụ 22: Kết quả của một cuộc điều tra ở Hà Nội cho thấy 96% các gia đình có máy thu hình, 98% có điện thoại và 95% có điện thoại và máy thu hình. Tính tỷ lệ % các gia đình ở Hà Nội không có điện thoại hoặc không có máy thu hình.

Giải:

Tỷ lệ % của các gia đình ở Hà Nội có máy thu hình hoặc điện thoại là:

$$96\% + 98\% - 95\% = 99\%.$$

Vậy, tỷ lệ % không có máy thu hình hoặc không có điện thoại là 1%.

Ví dụ 23: Tìm số phần tử của tập $A \cup B \cup C$ nếu mỗi tập có 100 phần tử và nếu:

a) Các tập hợp là từng cặp rời nhau;

b) Có 50 phần tử chung của mỗi cặp tập hợp và không có phần tử chung của 3 cặp;

c) Có 50 phần tử chung của mỗi cặp tập hợp và 25 phần tử chung của cả 3 cặp tập hợp.

Giải:

a) $|A \cup B \cup C| = |A| + |B| + |C| = 300$.

b) $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C|$
$- |B \cap C| = 300 - 150 = 150$.

c) $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C|$
$- |B \cap C| + |A \cap B \cap C| = 300 - 150 + 25 = 175$.

Ví dụ 24: Giả sử Khoa Công nghệ có tổng số là 2504 sinh viên. Trong đó có 1876 sinh viên học Pascal; 999 sinh viên học Fortran; 345 sinh viên học C. Ngoài ra có 876 sinh viên học cả Pascal và Fortran; 232 sinh viên học Fortran và C; 290 sinh viên học Pascal và C. Nếu có 189 sinh viên học cả ba môn Pascal, Fortran và C thì trong khoa có bao nhiêu sinh viên không học môn nào trong ba môn kể trên?

Giải:

A := tập sinh viên học Pascal $\Rightarrow |A| = 1876$.

B := tập sinh viên học Fortran $\Rightarrow |B| = 999$.

C := tập sinh viên học C $\Rightarrow |C| = 345$.

Ta có: $A \cap B$ là tập sinh viên học cả Pascal và Fortran $\Rightarrow |A \cap B| = 876$;

$B \cap C$ là tập sinh viên học cả Fortran và C $\Rightarrow |B \cap C| = 232$;

$A \cap C$ là tập sinh viên học cả Pascal và C $\Rightarrow |A \cap C| = 290$.

Ngoài ra $A \cap B \cap C$ là tập sinh viên học cả ba môn ngôn ngữ nói trên $\Rightarrow |A \cap B \cap C| = 189$.

Vậy số sinh viên học ít nhất một môn trong cả ba môn là

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

$$= 1876 + 999 + 345 - 876 - 290 - 232 + 189 = 2011.$$

Số sinh viên không học môn nào trong ba môn trên là

$$2504 - 2011 = 493.$$

Ví dụ 25: Sau cuộc phỏng vấn sinh viên tại nhà ăn ĐHQGHN, người ta thấy 64 sinh viên thích ăn cải xanh; 94 sinh viên thích ăn bắp cải; 58 sinh viên thích ăn súp lơ; 26 sinh viên thích ăn cải xanh và bắp cải; 28 sinh viên thích ăn cải xanh và súp lơ; 22 sinh viên thích ăn bắp cải và súp lơ; 11 sinh viên thích ăn cả ba loại rau. Hỏi trong số 270 sinh viên này có bao nhiêu sinh viên không thích ăn cả ba loại rau trên?

Giải:

A_1 := tập sinh viên thích ăn cải xanh $\Rightarrow |A_1| = 64$.

A_2 := tập sinh viên thích ăn bắp cải $\Rightarrow |A_2| = 94$.

A_3 := tập sinh viên thích ăn súp lơ $\Rightarrow |A_3| = 58$.

Ta có: $A_1 \cap A_2 =$ tập sinh viên thích ăn cải xanh và bắp cải

$\Rightarrow |A_1 \cap A_2| = 26$;

$A_1 \cap A_3 =$ tập sinh viên thích ăn cải xanh và súp lơ

$\Rightarrow |A_1 \cap A_3| = 28$;

$A_2 \cap A_3 =$ tập sinh viên thích ăn bắp cải và súp lơ

$\Rightarrow |A_2 \cap A_3| = 22$;

$A_1 \cap A_2 \cap A_3 =$ tập sinh viên thích ăn cả ba loại rau

$\Rightarrow |A_1 \cap A_2 \cap A_3| = 11$.

Số sinh viên thích ăn ít nhất một trong ba loại rau là

$$|A_1 \cup A_2 \cup A_3| = |A_1| + |A_2| + |A_3| - |A_1 \cap A_2|$$

$$- |A_1 \cap A_3| - |A_2 \cap A_3| + |A_1 \cap A_2 \cap A_3|$$

$$= 64 + 94 + 58 - 26 - 28 - 22 + 11 = 151.$$

Vậy số sinh viên không thích ăn cả ba loại rau trên là

$$270 - 151 = 119.$$

§4. NGUYÊN LÝ CHUỒNG CHIM BỒ CÂU

4.1. Nguyên lý chuồng chim bồ câu

Nếu chuồng chim bồ câu có ít cửa hơn số bồ câu thì có ít nhất có hai chim bồ câu đi chung một cửa. Vậy nguyên lý chuồng chim bồ câu được phát biểu dưới dạng tổng quát sau:

Nếu có k + 1 hoặc nhiều hơn đồ vật được đặt vào k hộp, thì tồn tại một hộp chứa hai hoặc nhiều hơn hai đồ vật.

Nguyên lý chuồng chim bồ câu cũng được gọi là nguyên lý Dirichlet trong mục 4.2.

4.2. Nguyên lý Dirichlet

Nếu có N đồ vật, đặt vào trong k hộp thì sẽ tồn tại một hộp chứa ít nhất $\left\lceil \frac{N}{k} \right\rceil$ vật (ở đây: $\left\lceil \frac{N}{k} \right\rceil$ là số nguyên dương bé nhất sao cho $\frac{N}{k} \leq \left\lceil \frac{N}{k} \right\rceil$; $\lceil x \rceil$ được gọi là hàm sàn trên của x).

Chứng minh: Giả sử ngược lại không có hộp nào trong k hộp chứa nhiều hơn $\left[\frac{N}{k}\right] - 1$ đồ vật. Khi đó tổng số các vật được chứa trong các hộp nhiều nhất là $k\left(\left[\frac{N}{k}\right] - 1\right)$.

Nhưng do định nghĩa: $\left[\frac{N}{k}\right] < \frac{N}{k} + 1$, hay $\left[\frac{N}{k}\right] - 1 < \frac{N}{k}$ nên ta có $k\left(\left[\frac{N}{k}\right] - 1\right) < k\left(\left(\frac{N}{k} + 1\right) - 1\right) = N$. Điều này trái với giả thiết.

Chú ý: Theo định nghĩa [x] thì

$$\left[\frac{3}{4}\right] = 1, \left[\frac{4}{3}\right] = 2, \left[-\frac{3}{4}\right] = 0.$$

4.3. Các ví dụ

Ví dụ 1: Trong một nhóm 367 người, bao giờ cũng có ít nhất 2 người cùng ngày sinh.

Giải:

Vì một năm có nhiều nhất 365 ngày, nên N = 367, còn k = 365.

Theo nguyên lý Dirichlet thì $\left[\frac{367}{365}\right] = 2$ nên có ít nhất 2 người trong 367 người có cùng một ngày sinh.

Ví dụ 2: Trong bất kỳ một nhóm 27 từ tiếng Anh nào, ít nhất cũng có 2 từ bắt đầu bằng cùng một chữ cái.

Giải:

N = 27, k = 26. Vậy ít nhất có $\left[\frac{27}{26}\right] = 2$ từ bắt đầu cùng một chữ cái.

Ví dụ 3: Trong 100 người có ít nhất mấy người cùng tháng sinh?

Giải:

Do N = 100, k = 12 nên $\left[\frac{N}{k}\right] = \left[\frac{100}{12}\right] = 9$. Vậy có ít nhất 9 người cùng tháng sinh.

Ví dụ 4: Cần phải có tối thiểu bao nhiêu sinh viên ghi tên vào lớp Toán rời rạc để chắc chắn sẽ có ít nhất 6 sinh viên đạt cùng một điểm thi nếu thang điểm gồm 5 bậc?

Giải:

Cần tìm số sinh viên N tối thiểu để có ít nhất 6 sinh viên đạt cùng một thang điểm $\left\lceil \frac{N}{5} \right\rceil = 6$.

Từ định nghĩa $\left\lceil \frac{N}{5} \right\rceil = 6 < \frac{N}{5} + 1$ hay $5 < \frac{N}{5} \Rightarrow N > 25$. Vậy tối thiểu 26 sinh viên ghi tên vào lớp Toán rời rạc.

Ví dụ 5: Chứng tỏ rằng, trong bất kỳ một tập hợp gồm 6 lớp học nào cũng có ít nhất hai lớp gặp nhau cùng một ngày, biết một tuần học từ thứ 2 đến thứ 6.

Giải:

$N = 6$, $k = 5$ (1 tuần có 5 ngày học).

Vì $\frac{N}{k} = \frac{6}{5} = 1,2$ nên $\left\lceil \frac{N}{k} \right\rceil = 2$. Vậy ít nhất có hai lớp cùng học trong một ngày.

Ví dụ 6: Chứng tỏ rằng, nếu trong một lớp có 30 sinh viên thì ít nhất có 2 sinh viên có tên bắt đầu bằng cùng một chữ cái.

Giải:

$N = 30$, $k = 26$ (bảng ký tự có 26 chữ cái).

Do $\frac{N}{k} = \frac{30}{26} = 1,1$ nên $\left\lceil \frac{N}{k} \right\rceil = 2$. Vậy ít nhất có 2 trong số 30 sinh viên có tên bắt đầu bằng một chữ cái.

Ví dụ 7: Mỗi sinh viên trong lớp K46CA của Khoa Công nghệ đều có quê ở một trong 64 tỉnh thành trong cả nước. Cần phải tuyển bao nhiêu sinh viên để đảm bảo trong lớp K46CA có ít nhất:

a) 2 sinh viên có quê cùng tỉnh.

b) 10 sinh viên có quê cùng tỉnh.

c) 50 sinh viên có quê cùng tỉnh.

Giải:

a) $\left[\dfrac{N}{64}\right] = 2$. Do $\dfrac{N}{64} > 1$ nên $N > 64$.

Vậy cần tuyển vào K46CA ít nhất là 65 sinh viên thì chắc chắn đảm bảo có ít nhất 2 sinh viên cùng một tỉnh.

b) $\left[\dfrac{N}{64}\right] = 10$. Do $\dfrac{N}{64} > 9$ nên $N > 576$. Vậy cần tuyển ít nhất 577 sinh viên.

c) $\left[\dfrac{N}{64}\right] = 50$. Do $\dfrac{N}{64} > 49$ nên $N > 3136$. Vậy cần tuyển ít nhất 3137 sinh viên.

Ví dụ 8: Chỉ ra trong 5 số chọn từ tập 8 số $\{1, 2, 3, 4, 5, 6, 7, 8\}$ bao giờ cũng có một cặp số có tổng bằng 9.

Giải:

Trong tập $\{1, 2, 3, 4, 5, 6, 7, 8\}$ ta có 4 cặp có tổng bằng 9 là: (1, 8), (2, 7), (3, 6), (4, 5).

Do $N = 5$, $k = 4$ nên $\left[\dfrac{N}{k}\right] = \left[\dfrac{5}{4}\right] = 2$. Vậy trong mỗi tập gồm 5 số chọn ra từ tập 8 số ở trên, bao giờ cũng chứa một cặp số có tổng là 9.

Ví dụ 9: Chỉ ra rằng, trong 6 số bất kỳ chọn từ tập 9 số nguyên dương đầu tiên, bao giờ cũng chứa ít nhất một cặp số có tổng bằng 10.

Giải:

Tập gồm 9 số nguyên dương đầu tiên là $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Trong tổng này có 5 cặp số có tổng bằng 10 là: (1, 9), (2, 8), (3, 7), (4, 6), (5, 5).

Do chỉ có 5 hộp, trong khi đó số vật là 6. Theo Dirichlet thì $\left[\dfrac{6}{5}\right] = 2$ nên trong tập 6 số phải chứa ít nhất một cặp số có tổng bằng 10.

Ví dụ 10: Cần phải tung một con xúc xắc bao nhiêu lần để có một mặt xuất hiện ít nhất:

a) 2 lần;

b) 3 lần;

c) n lần ($n \geq 4$).

Giải:

a) Số lần tung xúc xắc là N. Vì con xúc xắc có 6 mặt nên k = 6. Vậy theo Dirichlet thì $\left[\frac{N}{k}\right] = \left[\frac{N}{6}\right] = 2$. Do $\frac{N}{6} > 1$ nên $N > 6 \Rightarrow N = 7$. Vậy cần tung xúc xắc ít nhất 7 lần để có một mặt xuất hiện 2 lần.

b) $\left[\frac{N}{6}\right] = 3$. Do $\frac{N}{6} > 2$ nên $N > 12 \Rightarrow N = 13$. Vậy cần tung xúc xắc ít nhất 13 lần để có một mặt xuất hiện 3 lần.

c) $\left[\frac{N}{6}\right] = n$. Do $\frac{N}{6} > n - 1 \Rightarrow N > 6n - 6 \Rightarrow N = 6n - 5$. Vậy cần tung xúc xắc ít nhất 6n – 5 lần để có một mặt xuất hiện n lần ($n \geq 4$).

Chương 3
QUAN HỆ

§1. QUAN HỆ VÀ BIỂU DIỄN QUAN HỆ

1.1. Quan hệ và ví dụ về quan hệ

Cho hai tập A, B khác rỗng. Một tập con R của tích Đề-các $A \times B$ được gọi là *một quan hệ R* từ tập A vào tập B. Như vậy, cho quan hệ R từ A vào B, tức là cho $R \subseteq A \times B$.

Giả sử $(a, b) \in A \times B$, với $a \in A$, $b \in B$. Khi đó nếu $(a, b) \in R$ thì ta viết aRb và đọc là a có quan hệ R với b. Còn nếu $(a, b) \notin R$ thì ta nói rằng a không có quan hệ R với b và ký hiệu là $a \not R b$.

Ví dụ 1: Cho $A = \{a, b, c, d\}$, $B = \{n, m\}$.

Khi đó tập $R = \{(a, n), (b, n), (c, m)\} \subseteq A \times B$ là một quan hệ từ A vào B và aRn, bRn, cRm; các cặp còn lại trong $A \times B$ không có quan hệ R với nhau.

Ví dụ 2: Cho $A = \{1, 2, 3, 4, 5\}$, $B = \{4, 5, 6, 7\}$.

Giả sử $R = \{(1, 4), (2, 5), (3, 6), (4, 7)\}$. Khi đó R là một quan hệ từ A vào B.

Ví dụ 3: Cho $A = \{1, 2, 3, 4, 5\}$. Với mọi $a, b \in A$ ta định nghĩa aRb khi và chỉ khi $a + b$ là một số lẻ.

Khi đó R là một quan hệ từ A vào A, hay $R \subseteq A \times A$.

$$R = \{(1, 2), (2, 1), (1, 4), (4, 1), (2, 3), (3, 2), (3, 4), (4, 3), (2, 5), (5, 2), (4, 5), (5, 4)\}.$$

1.2. Phương pháp biểu diễn quan hệ

Các phương pháp biểu diễn quan hệ thông thường là: phương pháp định nghĩa, phương pháp ma trận và phương pháp đồ thị.

a) Phương pháp định nghĩa

Phương pháp này được minh hoạ thông qua các ví dụ 1, 2, 3 ở trên. Phương pháp định nghĩa là phương pháp cho R một cách trực tiếp hoặc gián tiếp dựa vào tính chất các phần tử của R.

b) Phương pháp ma trận

Cho hai tập hữu hạn $A = \{a_1, a_2, ..., a_n\}$ và $B = \{b_1, b_2, ..., b_m\}$. Giả sử R là một quan hệ từ A vào B, hay $R \subseteq A \times B$. Khi đó ma trận của quan hệ R là một ma trận cấp $n \times m$, tức là gồm n hàng và m cột mà phần tử δ_{ij} ở hàng i, cột j của ma trận đó được xác định như sau:

$$\delta_{ij} = \begin{cases} 1 \text{ nếu } a_i R b_j \\ 0 \text{ nếu không có } a_i R b_j \end{cases}$$

Ma trận trên được ký hiệu là $M_R^{n\times m} = \left[\delta_{ij}\right]$ $(i = \overline{1,n},\ j = \overline{1,m})$. Do các phần tử của ma trận này là 0 hoặc 1 nên ma trận $M_R^{n\times m}$ còn được gọi là *ma trận lôgic*. Rõ ràng là khi biết quan hệ R, ta tìm được ma trận lôgic biểu diễn nó; và ngược lại, khi biết ma trận lôgic thì ta xác định được quan hệ R ứng với ma trận lôgic đó.

Ví dụ 4: Cho A = {a, b, c, d, e}, B = {x, y, z, t}. Giả sử R là một quan hệ từ A vào B: R = {(a, x), (a, z), (a, t), (b, y), (b, t), (c, x), (c, y), (c, t), (d, x), (d, y), (d, z), (e, y), (e, t)}. Khi đó ma trận biểu diễn quan hệ R sẽ là:

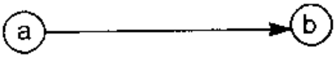
$$M_R^{5\times 4} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Rõ ràng từ ma trận trên ta dễ dàng xác định được quan hệ R.

c) Phương pháp đồ thị

Cho tập hữu hạn khác trống: $A = \{a_1, a_2, ..., a_n\}$.

Khi đó một quan hệ $R \subseteq A \times A$ có thể được biểu diễn bằng một đồ thị có hướng như sau: mỗi một $a \in A$ được xem là một đỉnh của đồ thị, do đó số đỉnh của đồ thị bằng số phần tử của tập A. Với $(a, b) \in R$ thì có một cung đi từ đỉnh a tới đỉnh b. Như vậy, nếu $a, b \in A$ mà $(a, b) \in R$ hay aRb thì từ a đến b có một cung:



Nếu (a, a) ∈ R thì ta có một khuyên tại đỉnh a: 

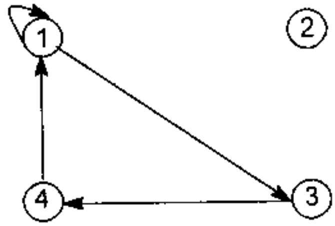
Nếu (a, b) ∉ R thì không có một cung nào đi từ a tới b.

Cũng như đối với cách biểu diễn quan hệ bằng ma trận, với mỗi một quan hệ $R \subseteq A \times A$ ta có thể biểu diễn quan hệ bằng đồ thị có hướng; và ngược lại, ứng với mỗi một đồ thị có hướng mà các đỉnh thuộc tập A có thể xác định được một quan hệ R trên A, sao cho nó nhận đồ thị đó là dạng biểu diễn của nó.

Ví dụ 5: Cho A = {1, 2, 3, 4} và quan hệ $R \subseteq A \times A$, với

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

Khi đó đồ thị biểu diễn quan hệ R sẽ là:



Ngược lại, cũng từ đồ thị ta hoàn toàn có thể xác định lại quan hệ R đã cho.

Ví dụ 6: Cho A = {1, 2, 3, 4, 5, 6} và quan hệ $R \subseteq A \times A$ được xác định như sau: Với mọi a, b ∈ A, aRb khi và chỉ khi hiệu a − b là một số chẵn. Khi đó ta có thể biểu diễn quan hệ theo cả ba phương pháp như sau:

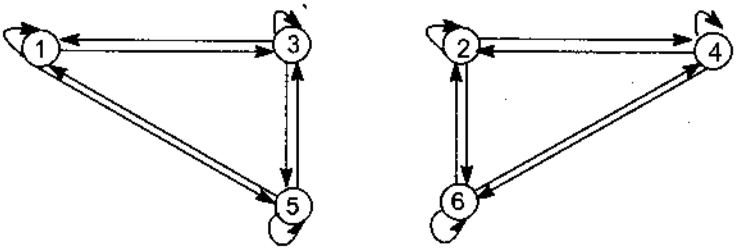
a) Biểu diễn quan hệ R theo định nghĩa:

Quan hệ R chính là tập {(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (1, 3), (3, 1), (1, 5), (5, 1), (2, 4), (4, 2), (2, 6), (6, 2), (3, 5), (5, 3), (4, 6), (6, 4)}.

b) Biểu diễn quan hệ R theo ma trận:

$$M_R^{6\times 6} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

c) Biểu diễn quan hệ R theo đồ thị:



1.3. Tính chất của quan hệ

Cho A là một tập hữu hạn, khác rỗng. Sau đây ta xét một số tính chất của quan hệ $R \subseteq A \times A$.

a) Tính phản xạ

Quan hệ R có tính *phản xạ* nếu $\forall a \in A$ ta luôn có aRa, hay $(a, a) \in R$.

b) Tính đối xứng

Quan hệ R có tính *đối xứng* nếu $\forall a, b \in A$ mà aRb thì bRa, hay nếu $(a, b) \in R$ thì $(b, a) \in R$.

c) Tính phản đối xứng

Quan hệ R có tính *phản đối xứng* nếu $\forall a, b \in A$ mà aRb và bRa thì $a = b$; tức là nếu $(a, b) \in R$ và $(b, a) \in R$ thì $a = b$.

d) Tính bắc cầu

Quan hệ R có tính bắc cầu nếu $\forall a, b, c \in A$ mà aRb và bRc thì có aRc; tức là nếu $(a, b), (b, c) \in R$ thì $(a, c) \in R$.

Nhận xét: Bằng phương pháp biểu diễn quan hệ theo ma trận và theo đồ thị ta dễ dàng nhận biết được một số tính chất của quan hệ.

Chẳng hạn:

– Một quan hệ có tính phản xạ khi và chỉ khi ma trận biểu diễn nó có tất cả các phần tử trên đường chéo chính đều bằng 1.

– Một quan hệ có tính phản xạ khi và chỉ khi đồ thị biểu diễn nó tại mỗi đỉnh đều có khuyên.

– Một quan hệ có tính đối xứng khi và chỉ khi ma trận biểu diễn nó là một ma trận đối xứng qua đường chéo chính.

– Một quan hệ có tính đối xứng khi và chỉ khi đồ thị biểu diễn nó có tính chất: Với a, b là hai đỉnh bất kỳ và nếu có một cung đi từ đỉnh a đến đỉnh b thì cũng có một cung đi từ đỉnh b đến đỉnh a.

Sau đây ta xét một số ví dụ.

Ví dụ 7: Cho $A = \{1, 2, 3, 4\}$. Tập $R \subseteq A \times A$ được định nghĩa như sau: $\forall a, b \in A$ thì aRb khi và chỉ khi tổng $a + b$ là một số lẻ.

a) Biểu diễn R bằng ba phương pháp đã nêu.

b) R có những tính chất gì?

c) Có nhận xét gì về ma trận và đồ thị của R?

Giải:

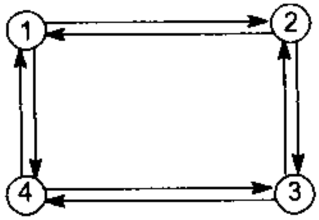
a) – Dạng định nghĩa của R là

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

– Dạng ma trận của R là

$$M_R^{4\times 4} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

– Dạng đồ thị của R là



b) Ta có nhận xét về một số tính chất của quan hệ R như sau:

– Quan hệ R có tính chất đối xứng, vì $a + b = b + a$ hay nếu $(a, b) \in R$ thì $(b, a) \in R$.

– R không phản xạ, vì với $(1, 1) \notin R$.

– R không phản đối xứng, vì $(1, 2) \in R$, $(2, 1) \in R$ nhưng $2 \neq 1$.

– R không bắc cầu, vì $(1, 2) \in R$, $(2, 3) \in R$ nhưng $(1, 3) \notin R$.

c) – Nhận xét tính chất của R từ ma trận của quan hệ R: Trong ma trận trên phần tử trên đường chéo chính bằng 0, nên R là không phản xạ. Do các phần tử đối xứng qua đường chéo chính đều bằng nhau nên R có tính đối xứng.

– Nhận xét tính chất của R từ đồ thị của quan hệ R: Vì trong đồ thị tồn tại đỉnh không có khuyên nên quan hệ R là không phản xạ. Từ đồ thị ta thấy rằng, nếu có một cung đi từ đỉnh i sang đỉnh j thì cũng có cung ngược lại đi từ đỉnh j sang đỉnh i, do đó R có tính chất đối xứng. Tồn tại một cặp cung có hướng là (1, 2) và (2, 3) trong đồ thị, nhưng cung (1, 3) lại không thuộc đồ thị, do đó quan hệ R không có tính bắc cầu.

Ví dụ 8: Cho $A = \{1, 2, 3, 4\}$. Quan hệ $R \subseteq A \times A$ được định nghĩa như sau: $\forall a, b \in A$ thì $aRb \Leftrightarrow a + b = 2k$ (k là nguyên). Chúng ta sẽ chỉ ra quan hệ R thoả mãn ba tính chất: Phản xạ, đối xứng và bắc cầu.

Giải: Thật vậy:

– R có tính chất phản xạ, vì $\forall a \in A$ ta có $a + a = 2a$ là một số chẵn, do đó $(a, a) \in R$.

– R có tính chất đối xứng, vì $\forall a, b \in A$ mà $a + b = 2k$ thì $b + a = 2k$, do đó nếu $(a, b) \in R$ thì $(b, a) \in R$.

– R có tính chất bắc cầu, vì $\forall a, b, c \in A$ nếu $a + b = 2k$, $b + c = 2k'$ thì $a + c = (2k - b) + (2k' - b) = 2(k + k') - 2b = 2k''$ với k, k', k'' là các số nguyên nào đó. Điều đó có nghĩa là nếu $(a, b) \in R$ và $(b, c) \in R$ thì $(a, c) \in R$.

Nhận xét về đồ thị và ma trận của R:

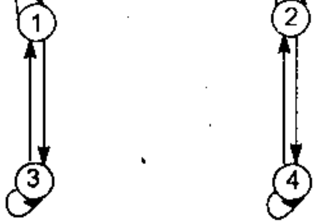
– Ma trận của quan hệ R:

$$M_R^{4\times4} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Trong ma trận của quan hệ R, các phần tử trên đường chéo chính đều bằng 1, do đó quan hệ R có tính phản xạ. Vì các phần tử đối xứng qua đường chéo chính đều bằng nhau suy ra R có tính đối xứng.

– Dạng đồ thị của R:

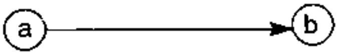
Từ đồ thị của R, mỗi đỉnh đều có khuyên, do đó quan hệ R có tính phản xạ. Từ đỉnh i đến đỉnh j có một cung thì cũng có cung ngược lại từ j đến i nên R có tính đối xứng.



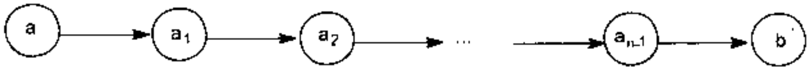
§2. CUNG VÀ ĐƯỜNG TRONG ĐỒ THỊ CỦA QUAN HỆ

2.1. Định nghĩa 1

Giả sử $R \subseteq A \times A$. Nếu $a, b \in A$ mà $(a, b) \in R$ hay aRb thì từ a đến b có một cung hướng đi từ a đến b:



Nếu giữa đỉnh a và đỉnh b có tồn tại một dãy các đỉnh $a = a_0, a_1, a_2, \ldots, a_{n-1}, a_n = b$ sao cho a_iRa_{i+1} $(i = 0, 1, 2, \ldots, n - 1)$ thì ta nói từ a đến b có một *đường đi* trong đồ thị của quan hệ R.



Độ dài của đường đi là số các cung có mặt trong đường đó. Trong đường trên có độ dài n. Nếu đường đi có độ dài bằng 1 thì đường đi đó là một cung, hay cung có thể xem là đường đi có độ dài bằng 1.

2.2. Tính chất

Định lý 1: Cho quan hệ $R \subseteq A \times A$ và R có tính chất bắc cầu. Khi đó nếu trong đồ thị của R có một đường đi độ dài n $(n \geq 1)$ từ đỉnh a đến đỉnh b thì cũng có một cung từ a đến b.

Chứng minh: Quy nạp theo $n \geq 1$.

– Trường hợp $n = 1$: Cung và đường trùng nhau nên định lý là đúng.

– Giả sử định lý đúng với n, tức là từ a đến b có một đường đi có độ dài n thì cũng có một cung từ a sang b.

– Xét đường đi bất kỳ từ a đến b có độ dài $n + 1$ trong đồ thị của R:

Ở đây $aRa_1, a_1Ra_2, \ldots, a_{n-1}Ra_n, a_nRb$. Đường đi từ a đến a_n có độ dài n nên theo giả thiết quy nạp thì từ a đến a_n có một cung, tức là aRa_n. Do R có tính bắc cầu nên từ aRa_n và a_nRb ta có aRb, hay trong đường độ dài $n + 1$ từ a đến b có một cung đi từ a đến b. Định lý được chứng minh.

§3. QUAN HỆ NGƯỢC VÀ QUAN HỆ HỢP THÀNH

3.1. Quan hệ ngược

Định nghĩa 2: Giả sử R là một quan hệ từ tập A vào tập B. Quan hệ *ngược* của quan hệ R được ký hiệu là R^{-1}. Đó là một quan hệ từ tập B vào tập A và được định nghĩa như sau:

$$R^{-1} = \{(b, a) : a \in A, b \in B \text{ và } aRb\}.$$

Ma trận quan hệ ngược: Giả sử $A = \{a_1, a_2, ..., a_n\}$, $B = \{b_1, b_2, ..., b_m\}$. Ma trận quan hệ ngược là ma trận lôgic cấp $m \times n$; tức là ma trận gồm m hàng, n cột mà phần tử hàng j cột i là α_{ji} được xác định như sau:

$$\alpha_{ji} = \begin{cases} 1 \text{ nếu } b_j R^{-1} a_i \\ 0 \text{ trong trường hợp ngược lại} \end{cases}$$

$$= \begin{cases} 1 \text{ nếu } a_i R b_j \\ 0 \text{ trong trường hợp ngược lại} \end{cases}$$

$= \delta_{ij}$ là phần tử ở hàng i, cột j trong ma trận của quan hệ R.

Như vậy, nếu quan hệ R có ma trận là $M_R^{n\times m} = \left[\delta_{ij}\right]$ $(i = \overline{1,n}, j = \overline{1,m})$ thì ma trận của quan hệ ngược $R^{-1} \subseteq B \times A$ mà ta ký hiệu là $M_{R^{-1}}^{m\times n} = [\alpha_{ji}]$ $(j = \overline{1,m}, i = \overline{1,n})$ sẽ nhận được từ $M_R^{n\times m}$ bằng cách đổi hàng thành cột, cột thành hàng.

Ví dụ 1: Cho A = {1, 2, 3, 4}, B = {5, 6, 7}; R là quan hệ từ A vào B được cho bởi tập R = {(1, 5), (1, 7), (2, 6), (3, 6), (3, 5)}. Tìm quan hệ ngược R^{-1} và ma trận của nó.

Theo định nghĩa ta có $R^{-1} = \{(5, 1), (7, 1), (6, 2), (6, 3), (5, 3)\}$

$$M_R^{4\times 3} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}; \quad M_{R^{-1}}^{3\times 4} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Rõ ràng ma trận của quan hệ R^{-1} nhận được từ ma trận của quan hệ R bằng cách đổi hàng thành cột, cột thành hàng.

3.2. Quan hệ hợp thành

Định nghĩa 3: Giả sử R là quan hệ từ tập A vào tập B, S là quan hệ từ tập B vào tập C. Khi đó một quan hệ Q từ A vào C được định nghĩa sau đây gọi là quan hệ *hợp thành* giữa hai quan hệ R và S:

$$Q = \{(a, c) : a \in A, c \in C, \text{ sao cho } \exists b \in B \text{ mà } aRb \text{ và } bSc\}.$$

Ta ký hiệu quan hệ hợp thành giữa R và S là Q = R.S.

Ma trận của quan hệ hợp thành: Giả sử Q là quan hệ hợp thành của hai quan hệ R và S, Q = R.S. Khi đó ma trận của quan hệ hợp thành Q được xác định như sau:

Giả sử có $A = \{a_1, a_2, ..., a_n\}$, $B = \{b_1, b_2, ..., b_m\}$ và $C = \{c_1, c_2, ..., c_r\}$.

Ma trận của quan hệ $R \subseteq A \times B$ là $M_R^{n\times m}$ gồm n hàng, m cột và phần tử hàng i, cột j là:

$$\delta_{ij} = \begin{cases} 1 \text{ nếu } a_iRb_j \\ 0 \text{ trong trường hợp ngược lại} \end{cases}$$

Ma trận của quan hệ $S \subseteq B \times C$ là $M_S^{m\times r}$ gồm m hàng, r cột và phần tử β_{jk} ở hàng j cột k được xác định bởi

$$\beta_{jk} = \begin{cases} 1 \text{ nếu } b_jSc_k \\ 0 \text{ trong trường hợp ngược lại} \end{cases}$$

Ký hiệu $M_Q^{n\times r}$ là ma trận của quan hệ Q, gọi γ_{ik} là phần tử ở hàng i, cột k của ma trận quan hệ hợp thành Q = R.S:

$$\gamma_{ik} = \begin{cases} 1 \text{ nếu } a_iQc_k \\ 0 \text{ trong trường hợp ngược lại} \end{cases}$$

Theo định nghĩa của quan hệ hợp thành thì:

$$\gamma_{ik} = \begin{cases} 1 \text{ nếu } \exists b_j \in B \text{ sao cho } a_iRb_j \text{ và } b_jSc_k \\ 0 \text{ trong trường hợp ngược lại} \end{cases}$$

Từ đó ta thấy $\gamma_{ik} = 1$ khi và chỉ khi $\delta_{ij} = \beta_{jk} = 1$, hay ma trận của quan hệ hợp thành bằng ma trận của quan hệ R nhân với ma trận của quan hệ S:

$$M_Q^{n\times r} = M_R^{n\times m}.M_S^{m\times r}.$$

Ví dụ 2: Cho A = {1, 2, 3, 4, 5}, B = {6, 7, 8, 9} và C = {a, b}.

$R \subseteq A \times B$ được cho bởi:

$$R = \{(1, 6), (1, 8), (2, 7), (2, 9), (3, 9), (4, 6), (4, 8), (5, 7)\}.$$

$S \subseteq B \times C$ được cho bởi:

$$S = \{(6, a), (7, a), (8, b)\}.$$

Xác định quan hệ hợp thành Q = R.S và chỉ ra rằng:

$$M_Q^{5\times2} = M_R^{5\times4}.M_S^{4\times2}.$$

Giải:

Thật vậy, xét quan hệ $Q \subseteq A \times C$ và dựa vào R, S ta có:

$$Q = R.S = \{(1, a), (1, b), (2, a), (4, a), (4, b), (5, a)\};$$

$$M_R^{5\times4} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix};\ M_S^{4\times2} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix};\ M_Q^{5\times2} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

Rõ ràng ta có: $M_Q^{5\times2} = M_R^{5\times4}.M_S^{4\times2}$.

§4. QUAN HỆ TƯƠNG ĐƯƠNG

4.1. Định nghĩa quan hệ tương đương

Giả sử R là một quan hệ từ tập A vào tập A. Quan hệ R được gọi là quan hệ *tương đương* trên A nếu R thoả mãn ba tính chất sau đây:

1) Tính phản xạ: $\forall a \in A$ ta có aRa;

2) Tính đối xứng: $\forall a, b \in A$ nếu aRb thì bRa;

3) Tính bắc cầu: $\forall a, b, c \in A$ nếu aRb và bRc thì aRc.

Ví dụ 1: Cho A = {1, 2, 3, 4, 5, 6, 7, 8, 9}. Trên A ta định nghĩa quan hệ R như sau: $\forall a, b \in A$ thì $aRb \Leftrightarrow a + b = 2k$, với k là một số nguyên dương nào đó.

Ta sẽ chỉ ra R là quan hệ tương đương; tức là cần chỉ ra R thoả mãn các tính chất phản xạ, đối xứng, bắc cầu.

Thật vậy:

– Phản xạ: $\forall a \in A$ ta có $a + a = 2a$ nên aRa.

– Đối xứng: Giả sử $a, b \in A$ và aRb khi đó $a + b = 2k$, do đó $b + a = 2k$ nên bRa.

– Bắc cầu: Giả sử $a, b, c \in A$ và aRb, bRc; tức là

$$a + b = 2k_1 \text{ và } b + c = 2k_2.$$

Xét: $a + c = (2k_1 - b) + (2k_2 - b) = 2(k_1 + k_2) - 2b = 2(k_1 + k_2 - b)$.

Từ đó suy ra $a + c = 2k_3$, với k_3 là một số nguyên dương nào đó, nên aRc. Vậy R là quan hệ tương đương.

4.2. Phân hoạch tương đương và lớp tương đương trên tập hợp

a) Phân hoạch tương đương

Cho tập $A \neq \varnothing$ và hữu hạn phần tử. Ta chia tập A thành n tập con $A_1, A_2, ..., A_n$ sao cho $A = \bigcup_{k=1}^{n} A_k$ và $A_i \cap A_j = \varnothing$ với $i \neq j$. Khi đó ta nói các tập $A_1, A_2, ..., A_n$ là một *phân hoạch tương đương* trên tập A. Ngược lại, ứng với mỗi phân hoạch tương đương $A_1, A_2, ..., A_n$ trên tập A, ta xác định quan hệ tương đương R trên A sinh ra phân hoạch đó như sau: $\forall a, b \in A$, $aRb \Leftrightarrow \exists A_i$ sao cho $a, b \in A_i$ $(i \in \{1, 2, ..., n\})$. Ta sẽ chỉ ra quan hệ R định nghĩa như trên là quan hệ tương đương trên A và R sinh ra phân hoạch đã cho. Thật vậy, ta sẽ chứng tỏ R thoả mãn ba tính chất:

– Phản xạ: Với mỗi $a \in A$ có duy nhất A_i để $a \in A_i$ hay aRa.

– Đối xứng: Nếu $a, b \in A$ mà aRb, suy ra có ít nhất A_i sao cho $a, b \in A_i$ $(i \in \{1, 2, ..., n\})$, từ đó $b, a \in A_i$ nên bRa.

– Bắc cầu: Giả sử $a, b, c \in A$ và aRb, bRc, ta cần chỉ ra aRc. Thật vậy, từ aRb suy ra tồn tại tập A_i để $a, b \in A_i$; từ bRc suy ra tồn tại tập A_j để $b, c \in A_j$. Do tính chất $A_i \cap A_j = \varnothing$ với $i \neq j$ nên suy ra $A_i = A_j$ (vì chúng cùng chứa b). Điều đó chứng tỏ $a, c \in A_i$, hay aRc.

Ví dụ 2: Cho $A = \{2, 3, 4, 5, 6\}$ và phân hoạch của A là $A_1 = \{2, 3\}$, $A_2 = \{4\}$, $A_3 = \{5, 6\}$. Quan hệ tương đương R sinh ra phân hoạch trên là $R = \{(2, 2), (3, 3), (2, 3), (3, 2), (4, 4), (5, 5), (6, 6), (5, 6), (6, 5)\}$. Dễ dàng chỉ ra R là quan hệ tương đương trên A. Để chỉ ra R sinh ra A_1, A_2, A_3 ở trên ta lập ma trận:

$$M_R^{5\times5} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Hàng thứ nhất và thứ hai của ma trận sinh ra $A_1 = \{2, 3\}$.

Hàng thứ ba của ma trận sinh ra $A_2 = \{4\}$.

Hàng thứ tư và thứ năm của ma trận sinh ra $A_3 = \{5, 6\}$.

Cần lưu ý rằng: Với mỗi quan hệ tương đương R trên tập A thì nó sẽ sinh ra một phân hoạch tương đương trên tập A; và ngược lại, với phân hoạch tương đương trên A sẽ sinh ra một quan hệ tương đương trên A.

Ví dụ 3: Cho A = {1, 2, 3, 4, 5, 6, 7}. Trên A ta định nghĩa quan hệ R như sau: $\forall a, b \in A$: $aRb \Leftrightarrow a - b = 3k$ (với k là một số nguyên). Hãy chỉ ra quan hệ R trên là quan hệ tương đương và tìm phân hoạch tương đương trên A do R sinh ra.

Giải: Từ định nghĩa ta có:

$$R = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 7), (1, 4), (4, 1), (1, 7), (7, 1), (2, 5), (5, 2), (3, 6), (6, 3), (4, 7), (7, 4)\}.$$

Trước hết chỉ ra R thoả mãn ba tính chất:

– Phản xạ: Với $\forall a \in A$ rõ ràng $a - a = 0 = 3.0$ nên aRa.

– Đối xứng: Giả sử $a, b \in A$ và aRb, ta có $a - b = 3k$.

Dĩ nhiên $b - a = 3(-k)$, do đó bRa.

– Bắc cầu: Giả sử $a, b, c \in A$; aRb và bRc, theo định nghĩa ta có

$a - b = 3k_1$, $b - c = 3k_2$.

Xét $a - c = (3k_1 + b) - (b - 3k_2) = 3(k_1 + k_2) = 3k_3$.

Vậy aRc.

Để tìm phân hoạch tương đương trên A do R sinh ra ta lập ma trận của R:

$$M_R^{7\times 7} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Các hàng 1, 4 và 7 sinh ra tập $A_1 = \{1, 4, 7\}$.

Các hàng 2 và 5 sinh ra tập $A_2 = \{2, 5\}$.

Các hàng 3 và 6 sinh ra tập $A_3 = \{4, 6\}$.

Khi đó các tập A_1, A_2, A_3 sẽ tạo thành một phân hoạch tương đương trên A.

Định lý 2:

– Nếu $R \subseteq A \times A$ là một quan hệ tương đương trên A ($A \neq \varnothing$) thì R sẽ tạo ra một phân hoạch tương đương trên A.

– Giả sử $A_1, A_2, ..., A_n$ là một phân hoạch tương đương nào đó trên A. Khi đó sẽ có một quan hệ tương đương $S \subseteq A \times A$ sinh ra phân hoạch tương đương trên.

Chứng minh:

Dựa vào định nghĩa quan hệ tương đương và phân hoạch tương đương trên A.

b) Lớp tương đương

Cho $R \subseteq A \times A$ và $a \in A$. Lớp tương đương của a đối với quan hệ R ký hiệu là $[a]_R$ và được định nghĩa như sau:

$$[a]_R = \{b : b \in A \text{ và } (a, b) \in R\}$$

Từ định nghĩa ta có kết quả sau:

Định lý 3: Cho R là một quan hệ tương đương trên A. Ba điều kiện sau là tương đương:

1) aRb;

2) $[a]_R = [b]_R$;

3) $[a]_R \cap [b]_R \neq \varnothing$.

§5. BAO ĐÓNG BẮC CẦU CỦA QUAN HỆ

5.1. Bao đóng bắc cầu của quan hệ

Giả sử $R \subseteq A \times A$. *Bao đóng bắc cầu* của quan hệ R được ký hiệu bởi [R] là quan hệ bắc cầu nhỏ nhất chứa R. Nói cách khác, nếu R là một quan hệ trên A thì [R] là một quan hệ nhỏ nhất chứa R sao cho nếu với mọi cặp $(a, b) \in [R]$ và $(b, c) \in [R]$ thì $(a, c) \in [R]$.

Ví dụ 1: Cho $A = \{a, b, c\}$ và $R = \{(a, b), (b, c)\}$ là một quan hệ trên A. Khi đó dễ dàng thấy rằng, quan hệ $[R] = \{(a, b), (b, c), (a, c)\}$ là bao đóng bắc cầu nhỏ nhất chứa R.

Định lý 4: Giả sử S là quan hệ bắc cầu trên A; R là quan hệ trên A thoả mãn $R \subseteq S$. Khi đó $[R] \subseteq S$.

Chứng minh: Ta chỉ ra rằng, nếu $(a, c) \in [R]$ thì $(a, c) \in S$. Thật vậy, nếu $(a, c) \in R$ thì ta có ngay $(a, c) \in S$.

Nếu $(a, c) \notin R$, điều đó có nghĩa là phải $\exists (a, b) \in R$ và $(b, c) \in R$ để từ đó có $(a, c) \in [R]$. Theo giả thiết, $R \subseteq S$ nên ta cũng có $(a, b) \in S$ và $(b, c) \in S$. Do S là quan hệ bắc cầu, suy ra $(a, c) \in S$.

Định lý đã được chứng minh.

5.2. Xác định bao đóng bắc cầu của quan hệ

Trước hết ta ký hiệu:

$R^1 = \{(a, b) : a, b \in A; aRb\}$: đường độ dài 1 từ a đến b.

$R^2 = \{(a, b) : a, b \in A; \exists c_1 \in A$ sao cho aRc_1 và $c_1Rb\}$: đường độ dài 2 đi từ a đến b.

$R^3 = \{(a, b) : a, b \in A; \exists c_1, c_2 \in A$ sao cho $aRc_1, c_1Rc_2, c_2Rb\}$: đường độ dài 3 đi từ a đến b.

...

$R^n = \{(a, b) : a, b \in A; \exists c_1, c_2, ..., c_{n-1} \in A$ sao cho $aRc_1, c_1Rc_2, ..., c_{n-1}Rb\}$: đường độ dài n đi từ a đến b.

Đặt $R^{\infty} = \bigcup_{n=1}^{\infty} R^n$.

Từ ký hiệu trên ta có: $(a, b) \in R^{\infty}$ nếu tồn tại số nguyên dương k sao cho $(a, b) \in R^k$. Điều đó cũng tương đương với việc tồn tại $c_1, c_2, ..., c_{k-1}$ sao cho

aRc_1, c_1Rc_2, c_2Rc_3, ..., $c_{k-2}Rc_{k-1}$, $c_{k-1}Rb$. Nói cách khác là $(a, b) \in R^\infty$ nếu tồn tại một đường đi độ dài k từ a đến b.

Định lý 5: Giả sử $R \subseteq A \times A$. Khi đó $R^\infty = [R]$.

Điều đó có nghĩa R^∞ là bao đóng bắc cầu nhỏ nhất chứa R.

Chứng minh:

– Trước hết chứng minh rằng R^∞ là bao đóng bắc cầu:

Thật vậy, giả sử $(a, b) \in R^\infty$ và $(b, c) \in R^\infty$, ta sẽ chỉ ra rằng $(a, c) \in R^\infty$. Do $(a, b) \in R^\infty$, suy ra tồn tại số nguyên dương k sao cho $(a, b) \in R^k$. Tương tự do $(b, c) \in R^\infty$, ta có số nguyên dương h sao cho $(b, c) \in R^h$. Khi đó tồn tại $c_1, c_2, ..., c_{k-1} \in A$ sao cho:

$$aRc_1,\ c_1Rc_2,\ c_2Rc_3,\ ...,\ c_{k-2}Rc_{k-1},\ c_{k-1}Rb \qquad (1)$$

Mặt khác cũng tồn tại $s_1, s_2, ..., s_{h-1} \in A$ sao cho:

$$bRs_1,\ s_1Rs_2,\ s_2Rs_3,\ ...,\ s_{h-2}Rs_{h-1},\ s_{h-1}Rc. \qquad (2)$$

Nối đường (1) và (2) tại đỉnh chung b ta có đường độ dài k + h đi từ a đến c, do đó $(a, c) \in R^{k+h}$, hay $(a, c) \in R^\infty$.

– Tiếp theo sẽ chứng minh rằng, với bất kỳ quan hệ bắc cầu S chứa R thì đều có $R^\infty \subseteq S$.

Thật vậy, giả sử có $(a, b) \in R^\infty$. Khi đó tồn tại $c_1, c_2, ..., c_{k-1} \in A$ sao cho: aRc_1, c_1Rc_2, c_2Rc_3, ..., $c_{k-2}Rc_{k-1}$, $c_{k-1}Rb$, tức là từ a đến b có một đường độ dài k. Vì $R \subseteq S$ nên từ aRc_1, c_1Rc_2, c_2Rc_3, ..., $c_{k-2}Rc_{k-1}$, $c_{k-1}Rb$ ta có: aSc_1, c_1Sc_2, c_2Sc_3, ..., $c_{k-2}Sc_{k-1}$, $c_{k-1}Sb$.

Do S là quan hệ bắc cầu nên theo định lý 1, suy ra từ a đến b có một cung, tức là $(a, b) \in S$. Điều đó chứng tỏ $R^\infty \subseteq S$, hay $R^\infty = [R]$ là bao đóng bắc cầu nhỏ nhất chứa R.

Thuật toán tìm bao đóng bắc cầu nhỏ nhất của quan hệ R theo định lý 5 rất khó thực hiện, vì vậy ta cần đưa ra một thuật toán đơn giản hơn sau đây:

Định lý 6: Giả sử tập A có n phần tử và R là một quan hệ từ A vào A. Khi đó $R^\infty = R^1 \cup R^2 \cup R^3 \cup ... \cup R^n$.

Chứng minh:

– Bao hàm thức $R^1 \cup R^2 \cup R^3 \cup ... \cup R^n \subseteq R^\infty$ là hiển nhiên.

– Ta chứng minh bao hàm thức ngược lại:

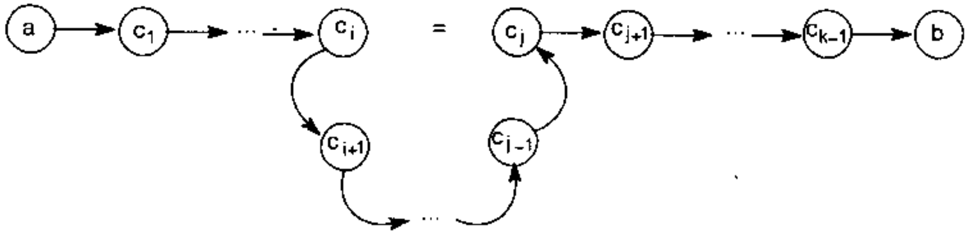
$$R^\infty \subseteq R^1 \cup R^2 \cup R^3 \cup ... \cup R^n.$$

Giả sử $(a, b) \in R^{\infty}$, khi đó tồn tại một số nguyên dương k sao cho $(a, b) \in R^k$. Điều đó có nghĩa là tồn tại $c_1, c_2, ..., c_{k-1} \in A$ sao cho: aRc_1, c_1Rc_2, c_2Rc_3, ..., $c_{k-2}Rc_{k-1}$, $c_{k-1}Rb$. Như vậy, có một đường từ a đến b độ dài k. Có hai khả năng:

1) Khi $k \le n$: Trong trường hợp này R^k là một trong các tập R^i ($i = \overline{1, n}$), do đó $(a, b) \in R^1 \cup R^2 \cup R^3 \cup ... \cup R^n$.

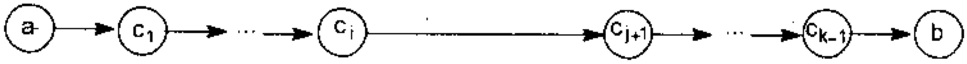
2) Khi $k > n$: Vì từ a đến b có đường với độ dài k, có nghĩa là tồn tại $c_1, c_2, ..., c_{k-1} \in A$ sao cho: aRc_1, c_1Rc_2, c_2Rc_3, ..., $c_{k-1}Rc_{k-1}$, $c_{k-1}Rb$.

Trong đường đó có k + 1 đỉnh, mà k > n nên phải tồn tại ít nhất một cặp đỉnh $\{c_i, c_j\}$ sao cho $c_i = c_j$ và $i < j$. Khi đó đường đi độ dài k từ a đến b có dạng:



Ở đây: aRc_1, c_1Rc_2, c_2Rc_3, ..., $c_{i-1}Rc_i$, ..., $c_{j-1}Rc_j$, ..., $c_{k-1}Rb$

Trong đường trên ta bỏ đi chu trình vòng, tức là bỏ đi các đỉnh c_{i+1}, ..., c_{j-1}, c_j và nối c_i với c_{j+1} bởi một cung ta được đường đi từ a đến b độ dài $k' < k$:



Ở đây: aRc_1, ..., c_iRc_{j+1}, ..., $c_{k-1}Rb$.

Nếu $k' > n$ thì trong đường này vẫn tồn tại các cặp đỉnh bằng nhau, tức là đường đi từ a đến b vẫn còn chu trình. Ta lặp lại quá trình trên để loại bỏ chu trình vòng cho tới khi nào trong đường đi từ a đến b gồm các đỉnh a, b và các đỉnh trung gian khác nhau từng cặp. Đường như thế sẽ có độ dài không lớn hơn n và do đó $(a, b) \in R^1 \cup R^2 \cup R^3 \cup ... \cup R^n$. Định lý đã được chứng minh.

§6. THUẬT TOÁN XÁC ĐỊNH BAO ĐÓNG BẮC CẦU CỦA QUAN HỆ

Theo định lý 6 thì $R^{\infty} = R^1 \cup R^2 \cup ... \cup R^n$ là bao đóng bắc cầu nhỏ nhất của quan hệ R trên tập A gồm n phần tử.

Ta gọi: $M_R = M_{R^1}$ là ma trận của quan hệ R;

M_{R^2} là ma trận của quan hệ R^2;

...

M_{R^n} là ma trận của quan hệ R^n.

Còn $M_{R^{\infty}}$ là ma trận bao đóng bắc cầu nhỏ nhất chứa quan hệ R được xác định trong định lý 6.

Khi đó ta có: $M_{R^{\infty}} = M_R \vee M_{R^2} \vee ... \vee M_{R^n} = M_{[R]}$.

Ví dụ 1: $M_R = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$ là ma trận của $R \subseteq A \times A$ với $|A| = 3$.

Khi đó: $$M_{R^2} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} . \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

và $$M_{R^3} = M_{R^2} . M_R = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} . \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

Vậy $M_{[R]} = M_R \vee M_{R^2} \vee M_{R^3}$

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

Định lý 6 dùng để làm cơ sở cho thuật toán tính ma trận quan hệ của bao đóng bắc cầu nhỏ nhất chứa R trên tập A gồm n phần tử.

Thuật toán tính bao đóng bắc cầu nhỏ nhất chứa quan hệ R trên tập A gồm n phần tử như sau:

```
Procedure BAO_DONG_BAC_CAU (M_R: ma trận cấp n × n của R)
    A := M_R;
    B := A;
    for i := 2 to n do
        Begin
            A := M_R.A;
            B := B ∨ A;
    End; (B là ma trận của [R] = R^1 ∪ R^2 ∪... ∪ R^n)
```

Chú ý: – Độ phức tạp của thuật toán trên là $O(n^4)$.

– Ngoài thuật toán trên, người ta có thể dùng thuật toán Warshall để tính $M_{[R]}$ với độ phức tạp thấp hơn là $O(n^3)$.

BÀI TẬP

1. Cho $N = \{1, 2, 3, 4, ...\}$. Trên N xác định quan hệ R như sau:

$$\forall a, b \in N: aRb \Leftrightarrow \frac{a}{b} = k \ (k = 1, 2, ...).$$

R có những tính chất nào trong các tính chất: phản xạ, đối xứng, phản đối xứng và bắc cầu?

2. Có bao nhiêu quan hệ có tính phản xạ trên một tập hợp có n phần tử.

3. Cho A là tập các sinh viên và B là tập các môn học. Giả sử R_1 là tập tất cả các cặp (a, b), trong đó a là sinh viên còn b là môn học mà a đã học. R_2 gồm tất cả các cặp (a, b), trong đó a là sinh viên cần học môn b để tốt nghiệp. Xác định các quan hệ: $R_1 \cup R_2$, $R_1 \cap R_2$, $R_1 \setminus R_2$, $R_2 \setminus R_1$.

4. Cho $R \subseteq A \times A$. Ta định nghĩa R^n (n = 1, 2, ...) bằng quy nạp như sau:

$$R^1 = R, R^{n+1} = R^n.R.$$

a) Cho $R = \{(1, 1), (2, 1), (3, 2), (4, 3)\}$. Tìm R^n (n = 1, 2, ...).

b) Chứng minh tính chất: Quan hệ R trên tập A là bắc cầu khi và chỉ khi $R^n \subseteq R$.

5. Có bao nhiêu quan hệ khác nhau từ tập có m phần tử vào tập có n phần tử?

6. Cho $R \subseteq A \times B$. Quan hệ ngược của R là

$$R^{-1} = \{(b, a) : b \in B, a \in A \text{ và } (a, b) \in R\}.$$

Còn quan hệ bù của R là $\overline{R} = \{(a, b) : (a, b) \notin R\}$.

a) Cho $R = \{(a, b) : a < b\}$ trên tập các số nguyên. Tìm R^{-1} và $\overline{R}$.

b) Cho R là quan hệ trên tập tất cả các tỉnh, thành phố của Việt Nam và được xác định R = {(a, b) : a là tỉnh giáp với tỉnh b}. Tìm R^{-1} và $\overline{R}$.

7. Cho R là quan hệ trên tập mọi người mà

R = {(a, b) : a là bố hoặc mẹ của b}.

S cũng là một quan hệ trên tập mọi người và

S = {(a, b) : a là anh hoặc chị ruột của b}.

Xác định quan hệ S.R và R.S?

8. Chứng minh rằng, nếu $R \subseteq A \times A$ có tính đối xứng và bắc cầu thì R có tính phản xạ.

9. Chứng minh rằng, quan hệ R trên tập A là phản xạ nếu và chỉ nếu quan hệ R^{-1} là phản xạ.

10. Chứng minh rằng, quan hệ R trên A là phản xạ khi và chỉ khi $\overline{R}$ là không phản xạ.

11. Giả sử R là quan hệ trên tập các xâu chữ cái tiếng Anh sao cho aRb khi và chỉ khi $l(a) = l(b)$, ở đây $l(x)$ là độ dài của xâu x.

R có phải là quan hệ tương đương không?

12. Chứng minh rằng, quan hệ R trên tập các số nguyên được xác định aRb khi và chỉ khi a = b hoặc a = –b là quan hệ tương đương.

13. Cho m là một số nguyên dương lớn hơn 1. Chứng minh rằng,

R = {(a, b) : a ≡ b (mod m)} là quan hệ tương đương.

14. Giả sử R là một quan hệ tương đương trên tập A. Khi đó ta nói lớp tương đương của phần tử a đối với R là $[a]_R$ (để cho gọn ta ký hiệu là [a]) được định nghĩa như sau:

$$[a]_R = [a] = \{s : (a, s) \in R\}.$$

Nếu $b \in [a]_R$ thì người ta nói b là đại diện của lớp tương đương đó.

a) Xác định lớp tương đương của một số nguyên đối với quan hệ cho trong bài 12.

b) Xác định các lớp tương đương của 0 và 1 đối với quan hệ đồng dư theo mod 4 (môđun 4).

15. Chứng minh rằng, nếu R là một quan hệ tương đương trên A thì các mệnh đề sau đây là tương đương:

1) aRb;

2) $[a] = [b]$;

3) $[a] \cap [b] \neq \varnothing$.

16. Các quan hệ nào trong số các quan hệ trên tập {0, 1, 2, 3} cho dưới đây là quan hệ tương đương? Xác định các tính chất của một quan hệ tương đương mà các quan hệ khác không có:

a) {(0, 0), (1, 1), (2, 2), (3, 3)};

b) {(0, 0), (0, 2), (2, 0), (2, 2), (2, 3), (3, 2), (3, 3)};

c) {(0, 0), (1, 1), (1, 2), (2, 1), (2, 2), (3, 3)};

d) {(0, 0), (1, 1), (1, 3), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)};

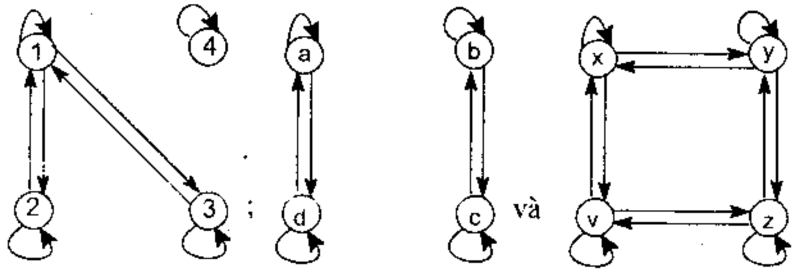
e) {(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 2), (3, 3)}.

17. Giả sử A là một tập không rỗng và f là một hàm có A là miền xác định. Trên A ta định nghĩa quan hệ R như sau: $\forall x, y \in A$: $xRy \Leftrightarrow f(x) = f(y)$.

a) Chứng minh R là quan hệ tương đương.

b) Xác định các lớp tương đương của R.

18. Cho các đồ thị:



Hãy xác định quan hệ được biểu diễn bởi các đồ thị trên là quan hệ tương đương.

19. Xác định xem các quan hệ được biểu diễn bởi các ma trận lôgic dưới đây có phải là quan hệ tương đương hay không?

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \text{ và } \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

20. Cho tập $\Sigma = \{0, 1\}$. Một dãy hữu hạn các ký tự trong tập Σ được sắp xếp đứng liền nhau, ta gọi là *một xâu*. Ví dụ: x = 11010 là một xâu trên Σ.

Tập tất cả các xâu (kể cả xâu rỗng – xâu không có ký tự nào) được thành lập từ Σ ta ký hiệu là Σ^*. Trên Σ^* ta định nghĩa quan hệ R như sau:

$$\forall x, y \in \Sigma^*: xRy \Leftrightarrow \text{số 1 trong các xâu } x, y \text{ là như nhau.}$$

Chứng minh rằng R là quan hệ tương đương.

21. Xác định lớp tương đương của xâu x = 011 đối với quan hệ tương đương trong bài 20.

22. Giả sử R_1 và R_2 là hai quan hệ tương đương trên tập A. Xác định xem các quan hệ $R_1 \cup R_2$, $R_1 \cap R_2$ có nhất thiết là quan hệ tương đương hay không?

23. Xác định số các quan hệ tương đương khác nhau trên tập ba phần tử bằng cách liệt kê ra các quan hệ đó.

24. Cho A = {1, 2, 3, 4, 5}. Trên A xác định quan hệ R như sau:

$$\forall a, b \in A: aRb \Leftrightarrow a + b = 2k \ (k = 1, 2, \ldots).$$

a) Biểu diễn R bằng các phương pháp liệt kê, ma trận và đồ thị có hướng.

b) Chứng minh R là quan hệ tương đương trên A.

c) Tìm phân hoạch tương đương trên A do R sinh ra.

d) Cho $A_1 = \{1\}$, $A_2 = \{2, 3\}$, $A_3 = \{4, 5\}$.

Tìm quan hệ tương đương S trên A mà S sinh ra phân hoạch A_1, A_2, A_3 ở trên.

e) Chứng minh $F = R \cup S$ là một quan hệ trên A mà $M_F = R_R \vee M_S$.

f) Cho Q = {(1, 3), (3, 5)} là một quan hệ trên A. Dùng thuật toán tìm bao đóng nhỏ nhất của một quan hệ để tìm [Q].

25. Cho A = {1, 2, 3, 4, 5, 6}. Trên A ta xác định quan hệ R như sau:

$$\forall a, b \in A: aRb \Leftrightarrow a - b = 3k \ (k = 0, \pm 1, \pm 2, \ldots).$$

a) Biểu diễn R bằng các phương pháp liệt kê, ma trận và đồ thị có hướng.

b) Chứng minh R là quan hệ tương đương trên A.

c) Tìm phân hoạch tương đương trên A do R sinh ra.

d) Tìm quan hệ tương đương S trên A sinh ra các tập

$$A_1 = \{1\}, A_2 = \{2, 3\}, A_3 = \{4, 5, 6\}.$$

e) Chứng minh $R^{-1} = R$ và $[R] = R$.

f) Tìm [Q] với $Q \subseteq A \times A$ và Q = {(1, 2), (2, 3), (3, 4)}.

PHẦN II
LÔGIC VÀ ỨNG DỤNG

Chương 4
LÔGIC MỆNH ĐỀ

§1. CÁC PHÉP TOÁN VÀ CÔNG THỨC

Trong chương này chúng ta chỉ đề cập tới các mệnh đề hoặc đúng, hoặc sai.

Ví dụ 1: "Hà Nội là thủ đô của Việt Nam", "1 > 3" là những mệnh đề hoặc đúng, hoặc sai. Chúng ta không nghiên cứu mệnh đề "Đề nghị các em trật tự" là những mệnh đề không đủ nghĩa để khẳng định mệnh đề đó là đúng hay sai.

Chính vì ta chỉ quan tâm tới mệnh đề hoặc đúng, hoặc sai mà không quan tâm tới nội dung cũng như cấu trúc của mệnh đề. Vì vậy, hai mệnh đề là tương đương nhau nếu nó hoặc đúng, hoặc sai như nhau.

Ví dụ:

– Hai mệnh đề: "Bác Hồ sinh năm 1890" và "Bác Hồ mất năm 1969, lúc Người 79 tuổi" là hai mệnh đề tương đương nhau.

– Hai mệnh đề: "3 là số nguyên tố" và "Chu vi đường tròn có tâm là gốc toạ độ, bán kính bằng 1 là số hữu tỷ" là không tương đương nhau.

Các mệnh đề chúng ta nghiên cứu trong phần này chỉ nhận một trong hai giá trị đúng, hoặc sai nên ta có thể định nghĩa các phép toán *Tuyển* ($\vee$), *Hội* ($\wedge$), *Kéo theo* ($\rightarrow$), *Phủ định* ($-$) trên tập các mệnh đề hoặc đúng, hoặc sai. Sự nghiên cứu các mệnh đề gọi là *đại số mệnh đề*.

Dưới đây để chỉ mệnh đề A nhận giá trị đúng tương ứng với ký hiệu T (*True*), còn nhận giá trị sai tương ứng với ký hiệu F (*False*).

1.1. Định nghĩa các phép toán trong đại số mệnh đề

Ký hiệu X, Y, Z (đôi khi cả chỉ số) là các mệnh đề hoặc đúng, hoặc sai (sau này còn gọi là mệnh đề sơ cấp hay biến mệnh đề).

a) Phép tuyển

Mệnh đề X tuyển với mệnh đề Y (ký hiệu $X \vee Y$) là một mệnh đề được định nghĩa như sau: $X \vee Y$ nhận giá trị T khi và chỉ khi ít nhất một trong hai mệnh đề X, Y nhận giá trị T. Mệnh đề $X \vee Y$ nhận giá trị F khi và chỉ khi cả X, Y đều nhận giá trị F.

b) Phép hội

Mệnh đề X hội mệnh đề Y (ký hiệu $X \wedge Y$) là một mệnh đề mà nó nhận giá trị T khi và chỉ khi cả hai mệnh đề nhận giá trị T. Mệnh đề $X \wedge Y$ nhận giá trị F khi và chỉ khi ít nhất một mệnh đề nhận giá trị F.

c) Phép phủ định

Phủ định của mệnh đề X (ký hiệu $\overline{X}$) nhận giá trị F khi và chỉ khi mệnh đề X nhận giá trị T. Mệnh đề $\overline{X}$ nhận giá trị T khi và chỉ khi X nhận giá trị F.

d) Phép kéo theo

Mệnh đề X kéo theo (suy ra) mệnh đề Y (ký hiệu $X \rightarrow Y$) nhận giá trị T khi và chỉ khi X nhận giá trị F hoặc X và Y cùng nhận giá trị T. Mệnh đề $X \rightarrow Y$ nhận giá trị F khi và chỉ khi X nhận giá trị T và Y nhận giá trị F.

Các phép toán ($\vee$, $\wedge$, —, $\rightarrow$) có thể định nghĩa thông qua bảng giá trị sau:

X	Y	$X \vee Y$	$X \wedge Y$	$\overline{X}$	$X \rightarrow Y$
T	T	T	T	F	T
T	F	T	F	F	F
F	T	T	F	T	T
F	F	F	F	T	T

Chú ý: Đôi khi giá trị đúng (T) ký hiệu là 1, giá trị sai (F) ký hiệu là 0.

1.2. Định nghĩa công thức trong lôgic mệnh đề

Định nghĩa 1:

a) Mỗi mệnh đề sơ cấp, ký hiệu là X, Y, Z (đôi khi cả số nữa) là một công thức.

b) Nếu A, B là hai công thức thì $(A \vee B)$, $(A \wedge B)$, $(\overline{A})$, $(A \rightarrow B)$ cũng là những công thức.

Lưu ý: Mỗi công thức chỉ chứa các ký hiệu X, Y, Z, các phép toán $\vee$, $\wedge$, $\overline{}$, $\rightarrow$ và các ký hiệu mở ngoặc "(", đóng ngoặc ")". Bản thân $A \vee B$, $A \wedge B$, $\overline{A}$, $A \rightarrow B$ không phải là những công thức theo định nghĩa, nhưng đôi khi để cho gọn ta viết $A \vee B$ thay cho $(A \vee B)$, ...

1.3. Công thức đồng nhất bằng nhau và công thức đồng nhất đúng

Định nghĩa 2: Hai công thức A và B được gọi là *đồng nhất bằng nhau* (ký hiệu $A \equiv B$) khi và chỉ khi chúng cùng nhận giá trị đúng, sai như nhau đối với mọi bộ giá trị đúng, sai của các mệnh đề sơ cấp có trong hai công thức A và B.

Định nghĩa 3: Công thức A được gọi là *đồng nhất đúng* (hằng đúng, ký hiệu là $\models A$ hoặc $A = 1$) khi và chỉ khi A luôn nhận giá trị đúng với mọi bộ giá trị đúng, sai có thể của các mệnh đề cơ sấp trong A.

Định nghĩa 4: Công thức A được gọi là *đồng nhất sai* (hằng sai, ký hiệu là $A \equiv \mathbf{0}$) khi và chỉ khi A luôn nhận giá trị sai với mọi bộ giá trị đúng, sai có thể của các mệnh đề sơ cấp trong A.

1.4. Bảng công thức đồng nhất bằng nhau

1) $\overline{\overline{A}} \equiv A$
2) $A \vee B \equiv B \vee A$
3) $A \wedge B \equiv B \wedge A$
4) $(A \vee B) \vee C \equiv A \vee (B \vee C) \equiv A \vee B \vee C$
5) $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C) \equiv A \wedge B \wedge C$
6) $\overline{A \vee B} \equiv \overline{A} \wedge \overline{B}$
7) $\overline{A \wedge B} \equiv \overline{A} \vee \overline{B}$
8) $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
9) $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
10) $A \rightarrow B \equiv \overline{A} \vee B$
11) $A \vee A \equiv A$

12) $A \wedge A \equiv A$

13) $A \wedge \overline{A} \equiv \mathbf{0}$

14) $A \vee \overline{A} \equiv \mathbf{1}$

15) $A \wedge \mathbf{1} \equiv A$

16) $A \vee \mathbf{0} \equiv A$

17) $A \vee \mathbf{1} \equiv \mathbf{1}$

18) $A \wedge \mathbf{0} \equiv \mathbf{0}$

19) $A \vee (A \wedge B) \equiv A$

20) $A \wedge (A \vee B) \equiv A$

Chú ý: 1) Các công thức từ 1 đến 20 dùng để tìm dạng chuẩn tắc hội và dạng chuẩn tắc tuyển của một công thức cho trước.

2) Ký hiệu **1** chỉ công thức hằng đúng còn **0** chỉ công thức hằng sai.

1.5. Bảng công thức hằng đúng

1) $\models A \to (B \to A)$

2) $\models (A \to (B \to C)) \to ((A \to B) \to (A \to C))$

3) $\models (A \wedge B) \to A$

4) $\models (A \wedge B) \to B$

5) $\models (A \to B) \to ((A \to C) \to (A \to (B \wedge C)))$

6) $\models A \to (A \vee B)$

7) $\models B \to (A \vee B)$

8) $\models (A \to C) \to ((B \to C) \to ((A \vee B) \to C))$

9) $\models (A \to B) \to (\overline{B} \to \overline{A})$

10) $\models A \to \overline{\overline{A}}$

11) $\models \overline{\overline{A}} \to A$

1.6. Luật đối ngẫu

Giả sử A là một công thức chỉ chứa các phép toán $\vee$, $\wedge$, — mà không chứa phép toán $\to$. Trong A đổi chỗ vai trò hai phép toán $\wedge$ và $\vee$ cho nhau ta được công thức A*, ta gọi A* là *công thức đối ngẫu* của công thức A.

Ví dụ 2: Đối ngẫu của công thức $A \equiv (X \vee Y) \wedge \overline{Z}$ là công thức

$$A^* \equiv (X \wedge Y) \vee \overline{Z}.$$

Định lý 1: Giả sử $A(X_1, X_2, ..., X_n)$ là một công thức, trong đó X_i $(i = 1, ..., n)$ là các mệnh đề sơ cấp trong A. Khi đó ta luôn có:

$$A^* \equiv \overline{A}(\overline{X}_1, \overline{X}_2, ..., \overline{X}_n).$$

Chứng minh: Dựa vào định nghĩa đệ quy của công thức A. Để cho gọn ta viết $\overline{A}(\overline{X})$ thay cho $\overline{A}(\overline{X}_1, \overline{X}_2, ..., \overline{X}_n)$.

– Nếu $A \equiv X$, ở đây X là mệnh đề sơ cấp thì hiển nhiên

$$\overline{A}(\overline{X}) \equiv \overline{\overline{X}} \equiv X \text{ hay } A^* \equiv \overline{A}(\overline{X}).$$

– Giả sử công thức đã được chứng minh cho các công thức A và B, tức là $A^* \equiv \overline{A}(\overline{X})$ và $B^* \equiv \overline{B}(\overline{X})$. Ta chứng minh định lý cũng đúng cho các công thức $(A \wedge B)$, $(A \vee B)$ và $(\overline{A})$.

+ Xét công thức $(A \wedge B)$:

$$(A \wedge B)^* \equiv A^* \vee B^* \equiv \overline{A}(\overline{X}) \vee \overline{B}(\overline{X}) \equiv (\overline{A \wedge B})(\overline{X}).$$

+ Xét công thức $(A \vee B)$:

$$(A \vee B)^* \equiv A^* \wedge B^* \equiv \overline{A}(\overline{X}) \wedge \overline{B}(\overline{X}) \equiv (\overline{A \vee B})(\overline{X}).$$

+ Xét công thức $(\overline{A})$:

$$(\overline{A})^* \equiv \overline{A^*} \equiv \overline{\overline{A}}(\overline{X}).$$

Định lý được chứng minh.

Ví dụ 3: Cho $A \equiv (X \vee Y) \wedge \overline{Z}$. Theo định lý thì

$$\overline{A}(\overline{X}) \equiv \overline{(\overline{X} \vee \overline{Y}) \wedge \overline{\overline{Z}}} \equiv \overline{(\overline{X} \vee \overline{Y})} \vee \overline{\overline{\overline{Z}}} \equiv (X \wedge Y) \vee \overline{Z} \equiv A^*.$$

Cần chú ý rằng, nếu $A \equiv B$ thì $A^* \equiv B^*$.

1.7. Luật thay thế

Nếu A là một công thức chứa mệnh đề sơ cấp X thì khi thay X bởi một công thức E nào đó ta được công thức mới ký hiệu là B. Với các ký hiệu đó ta có:

Định lý 2: Nếu $\models A$ thì $\models B$.

Chứng minh suy ra từ định nghĩa công thức đồng nhất đúng.

1.8. Luật kết luận

Định lý 3: Nếu $\models A$ và $\models A \to B$ thì $\models B$.

Chứng minh bằng phương pháp phản chứng.

§2. ĐIỀU KIỆN ĐỒNG NHẤT ĐÚNG (hằng đúng) ĐIỀU KIỆN ĐỒNG NHẤT SAI (hằng sai)

2.1. Tuyển và hội sơ cấp

Bài toán: Cho công thức A bất kỳ. Có hay không một thuật toán mà sau hữu hạn các bước làm việc ta có thể biết công thức A có phải là công thức đồng nhất đúng hay không?

Đối với bài toán trên luôn có một thuật toán để biết công thức A có phải là công thức đồng nhất đúng hay không, đó là thuật toán lập bảng.

	X_1	X_2	...	X_n	$A(X_1, X_2, ..., X_n)$
	F	F	...	F	?
2n hàng	F	F	...	T	?
			...		....
	T	T	...	T	?

Phương pháp lập bảng sẽ gặp khó khăn khi số các biến trong A quá lớn. Để khắc phục nhược điểm trên ta cần xây dựng thuật toán khác tốt hơn, đó là thuật toán tìm dạng chuẩn tắc tuyển và dạng chuẩn tắc hội của công thức A.

Trước hết, ta đưa vào các khái niệm sau:

Định nghĩa 5:

1) Tuyển các mệnh đề sơ cấp và phủ định của nó gọi là *tuyển sơ cấp* (TSC).

2) Hội các mệnh đề sơ cấp và phủ định của nó gọi là *hội sơ cấp* (HSC).

Định lý 4:

1) Điều kiện cần và đủ để một TSC đồng nhất đúng là trong tuyển đó có chứa một mệnh đề sơ cấp đồng thời với phủ định của nó.

2) Điều kiện cần và đủ để một HSC đồng nhất sai là trong hội đó có chứa một mệnh đề sơ cấp đồng thời với phủ định của nó.

Chứng minh:

1) *Điều kiện cần:* Giả sử TSC là đồng nhất đúng. Ta chỉ ra trong TSC có chứa một mệnh đề đồng thời với phủ định của nó. Giả sử ngược lại, trong TSC không chứa một mệnh đề sơ cấp nào đồng thời với phủ định của nó. Trong TSC thay mệnh đề sơ cấp không có dấu phủ định bằng giá trị F, còn mệnh đề sơ cấp có dấu phủ định thì thay bằng T thì TSC nhận giá trị F. Điều này mâu thuẫn với giả thiết. Chứng tỏ TSC phải chứa mệnh đề sơ cấp cùng với phủ định của nó.

Điều kiện đủ: Giả sử trong TSC có chứa một mệnh đề sơ cấp đồng thời với phủ định của nó, chẳng hạn TSC $\equiv X \vee \overline{X} \vee Y \vee ... \vee \overline{Z}$. Vì $X \vee \overline{X}$ là công thức đồng nhất đúng nên TSC phải là công thức đồng nhất đúng.

2) Chứng minh tương tự.

2.2. Dạng chuẩn tắc tuyển và chuẩn tắc hội

Định nghĩa 6:

1) Giả sử A là một công thức. Nếu $A' \equiv A$ mà A' là tuyển của các HSC, tức là $A' \equiv (HSC)_1 \vee (HSC)_2 \vee ... \vee (HSC)_n$ thì A' được gọi là *dạng chuẩn tắc tuyển* (DCTT) của A.

2) Nếu $A'' \equiv A$ mà A'' là hội của các TSC, tức là

$$A'' \equiv (TSC)_1 \vee (TSC)_2 \wedge ... \wedge (TSC)_n$$

ta nói A'' là *dạng chuẩn tắc hội* (DCTH) của A.

Ví dụ 1: $A \equiv X \wedge (X \to Y)$.

$A' \equiv (X \wedge \overline{X}) \vee (X \wedge Y)$ là DCTT của A.

$A'' \equiv X \wedge (\overline{X} \vee \overline{Y})$ là DCTH của A.

Dĩ nhiên $A \equiv A' \equiv A''$.

Định lý 5: Mọi công thức trong đại số mệnh đề đều có DCTT và DCTH.

Chứng minh: Giả sử A là một công thức bất kỳ trong đại số mệnh đề. Nếu trong A có chứa phép toán $\to$ thì có thể khử phép $\to$ bằng cách áp dụng công thức đồng nhất bằng nhau $X \to Y \equiv \overline{X} \vee Y$. Vì vậy ta có thể giả thiết A chỉ chứa phép toán $\vee$, $\wedge$ và $\overline{}$. Nếu phép phủ định chưa trực tiếp đối với các mệnh đề sơ cấp trong A thì ta sử dụng các công thức đồng nhất bằng nhau $\overline{X \vee Y} \equiv \overline{X} \wedge \overline{Y}$ hay $\overline{X \wedge Y} \equiv \overline{X} \vee \overline{Y}$. Tiếp theo nhờ các công thức đồng nhất bằng nhau:

$$X \vee (Y \wedge Z) \equiv (X \vee Y) \wedge (X \vee Z)$$

$$X \wedge (Y \vee Z) \equiv (X \wedge Y) \vee (X \wedge Z)$$

để đưa A về DCTH và DCTT. Định lý được chứng minh.

Ví dụ 2: Cho $A \equiv Z \rightarrow (Y \rightarrow X)$, tìm DCTH và DCTT của A.

Ta có $A \equiv X \rightarrow (Y \rightarrow X) \equiv \overline{X} \vee \overline{Y} \vee X \equiv A'$ là DCTH với các TSC $\equiv \overline{X} \vee \overline{Y} \vee X$ và cũng là DCTT với các HSC là $\overline{X}$, $\overline{Y}$ và X.

Định lý 6:

1) Điều kiện cần và đủ để công thức A đồng nhất đúng là trong DCTH của A mỗi TSC chứa một mệnh đề sơ cấp đồng thời với phủ định của nó.

2) Điều kiện cần và đủ để công thức A đồng nhất sai là trong DCTT của A mỗi HSC chứa một mệnh đề sơ cấp đồng thời với phủ định của nó.

Chứng minh:

1) *Điều kiện cần:* Giả sử A là công thức đồng nhất đúng.

Theo định lý 5 thì với A có tồn tại DCTH:

$$A' \equiv (TSC)_1 \wedge (TSC)_2 \wedge \ldots \wedge (TCS)_n \equiv A.$$

Vì A đồng nhất đúng nên $(TSC)_i$ ($i = 1, \ldots, n$) là đồng nhất đúng. Theo định lý 4 thì trong mỗi $(TSC)_i$ ($i = 1, \ldots, n$) có chứa một mệnh đề sơ cấp đồng thời với phủ định của nó.

Điều kiện đủ: Giả sử $A' \equiv (TSC)_1 \wedge (TSC)_2 \wedge \ldots \wedge (TSC)_n$ là DCTH của A, trong đó mỗi $(TSC)_i$ ($i = 1, \ldots, n$) có chứa một mệnh đề sơ cấp đồng thời với phủ định của nó. Theo định lý 4 thì mỗi $(TSC)_i$ ($i = 1, \ldots, n$) là đồng nhất đúng và do đó A là công thức đồng nhất đúng.

2) Chứng minh tương tự.

2.3. Thuật toán nhận biết hằng đúng, hằng sai và thực hiện được của công thức trong lôgic mệnh đề

Bài toán:	input:	A là công thức bất kỳ.
	output:	DCTH, DCTT của A;
		A hằng đúng, hằng sai?

Thuật toán:

Bước 1: Khử phép toán kéo theo (→) trong A bằng cách áp dụng công thức $X \rightarrow Y \equiv \overline{X} \vee Y$ ta được công thức A_1 ($\equiv$ A). Chuyển sang bước 2.

Bước 2: Đưa phép toán phủ định (–) trong A_1 về trực tiếp liên quan tới từng biến mệnh đề sơ cấp có mặt trong A_1 bằng cách áp dụng công thức $\overline{X \vee Y} \equiv \overline{X} \wedge \overline{Y}$ và $\overline{X \wedge Y} \equiv \overline{X} \vee \overline{Y}$ ta được công thức A_2 ($\equiv A_1$). Chuyển sang bước 3.

Bước 3: Đưa A_2 về DCTH bằng cách áp dụng công thức

$$X \vee (Y \wedge Z) \equiv (X \vee Y) \wedge (X \vee Z)$$

ta được công thức A_3 ($\equiv A_2$), ở đây $A_3 \equiv (TSC_1) \wedge (TSC_2) \wedge \ldots \wedge (TSC_n)$ ($n \geq 1$).

– Nếu trong mỗi TSC có chứa một biến mệnh đề sơ cấp đồng thời với phủ định của nó thì A_3 hằng đúng hay A hằng đúng.

– Ngược lại, nếu tồn tại một TSC không chứa biến mệnh đề sơ cấp nào đồng thời với phủ định của nó thì A không hằng đúng. Chuyển sang bước 4.

Bước 4: Đưa A_2 về DCTT bằng cách áp dụng công thức

$$X \wedge (Y \vee Z) \equiv (X \wedge Y) \vee (X \wedge Z)$$

ta được công thức A'_3 ($\equiv A_2$), ở đây $A'_3 \equiv (HSC_1) \vee (HSC_2) \vee \ldots \vee (HSC_n)$ ($n \geq 1$).

– Nếu trong mỗi HSC có chứa một biến mệnh đề sơ cấp đồng thời với phủ định của nó thì A'_3 hằng sai hay A hằng sai.

– Ngược lại, A không hằng sai và A là công thức thực hiện được.

Chú ý: Công thức A thực hiện được khi và chỉ khi trong A có tồn tại một bộ giá trị đúng, sai của các mệnh đề sơ cấp trong A sao cho với bộ giá trị đúng, sai đó thì A nhận giá trị đúng.

§3. CÁC QUY TẮC SUY DIỄN TRONG LÔGIC MỆNH ĐỀ

Một hệ thống toán học bao gồm các tiên đề, các định nghĩa và các khái niệm không được định nghĩa. Các tiên đề được giả định là đúng. Các định nghĩa được sử dụng để xây dựng khái niệm mới trên cơ sở những khái niệm đã có. Trong một hệ toán học, ta có thể suy ra được các định lý. Một định lý là một khẳng định được chứng minh là đúng. Một lập luận chỉ ra được tính đúng đắn của một mệnh đề phát biểu trong định lý được gọi là một chứng minh. Lôgic là một công cụ phục vụ cho việc phân tích các chứng minh. Thông thường một chứng minh sẽ bao gồm nhiều bước suy luận mà ở mỗi bước ta đi đến (hay suy ra) một sự khẳng định mới từ những khẳng định đã biết. Trong chứng minh toán học, ta thường gặp dạng suy diễn sau đây: Nếu A_1 và A_2 và ... và A_n thì B. Dạng suy luận này là hợp lý (chấp nhận là đúng) khi công thức $(A_1 \wedge A_2 \wedge ... \wedge A_n) \to B$ là hằng đúng. Ở đây $A_1 \wedge A_2 \wedge ... \wedge A_n$ là giả thiết (hay tiên đề), còn B là kết luận (hay B là hệ quả lôgic của giả thiết). Các tiên đề đã đúng thì hệ quả lôgic của nó phải đúng.

Mô hình suy diễn của công thức $(A_1 \wedge A_2 \wedge ... \wedge A_n) \to B$ viết dưới dạng

$$\begin{array}{l} A_1 \\ A_2 \\ \vdots \\ \underline{A_n} \\ \therefore B \end{array}$$

(đọc là: Nếu A_1 đúng, A_2 đúng, ..., A_n đúng thì B đúng).

3.1. Các quy tắc suy diễn

1. Quy tắc suy diễn rút gọn (Rg)

Cơ sở của quy tắc suy diễn rút gọn là hằng đúng: $(A \wedge B) \to A$.

Mô hình suy diễn của công thức này viết dưới dạng:

$$\begin{array}{c} A \\ \underline{B} \\ \therefore A \end{array}$$

Ký hiệu trên vạch là *giả thiết*, ký hiệu dưới vạch là *kết luận*, còn ký hiệu $\therefore$ có nghĩa là *thì*. Đọc: "Nếu A đúng và B đúng thì A đúng".

2. Quy tắc suy diễn cộng (Cg)

Cơ sở của quy tắc suy diễn là hằng đúng: $A \to (A \vee B)$.

Mô hình suy diễn của công thức này là:

$$\begin{array}{c} \underline{A} \\ \therefore A \vee B \end{array}$$

3. Quy tắc suy diễn khẳng định (Kđ)

Cơ sở của quy tắc suy diễn khẳng định là hằng đúng: $(A \wedge (A \to B)) \to B$.

Mô hình suy diễn của công thức này là:

$$\begin{array}{l} A \\ A \to B \\ \hline \therefore B \end{array}$$

4. Quy tắc suy diễn phủ định (Pđ)

Cơ sở của quy tắc suy diễn này là hằng đúng: $\left((A \to B) \wedge \overline{B}\right) \to \overline{A}$.

Mô hình suy diễn của công thức này là:

$$\begin{array}{l} A \to B \\ \overline{B} \\ \hline \therefore \ \overline{A} \end{array}$$

5. Quy tắc suy diễn tam đoạn luận (Tđl)

Cơ sở của quy tắc suy diễn tam đoạn luận là hằng đúng:

$$((A \to B) \wedge (B \to C)) \to (A \to C).$$

Mô hình suy diễn của công thức này là:

$$\begin{array}{l} A \to B \\ B \to C \\ \hline \therefore \ A \to C \end{array}$$

6. Quy tắc suy diễn tam đoạn luận rời (Tđlr)

Cơ sở của quy tắc tam đoạn luận rời là hằng đúng: $\left((A \vee B) \wedge \overline{A}\right) \to B$.

Mô hình suy diễn của công thức này là:

$$\begin{array}{l} A \vee B \\ \overline{A} \\ \hline \therefore \ B \end{array}$$

7. Quy tắc suy diễn mâu thuẫn (Mt)

Cơ sở của quy tắc suy diễn mâu thuẫn là hằng đúng:

$$(A_1 \wedge A_2 \wedge ... \wedge A_n) \to B \equiv (A_1 \wedge A_2 \wedge ... \wedge A_n \wedge \overline{B}) \to \mathbf{0}$$

Mô hình suy diễn của công thức này là:

$$\begin{array}{l} A_1 \\ A_2 \\ \vdots \\ \underline{A_n} \\ \therefore B \end{array} \equiv \begin{array}{l} A_1 \\ A_2 \\ \vdots \\ A_n \\ \underline{\overline{B}} \\ \therefore \mathbf{0} \end{array}$$

Công thức này có nghĩa là: Nếu ta thêm vào giả thiết ban đầu giả thiết phụ $\overline{B}$ mà dẫn tới mâu thuẫn thì B là hệ quả lôgic của các giả thiết ban đầu.

8. Quy tắc suy diễn theo trường hợp (Th)

Cơ sở của quy tắc suy diễn này là hằng đúng:

$$((A \to C) \wedge (B \to C)) \to ((A \vee B) \to C).$$

Mô hình suy diễn của công thức này là:

$$\begin{array}{c} A \to C \\ \underline{B \to C} \\ \therefore (A \vee B) \to C \end{array}$$

Công thức này có nghĩa là: Nếu một giả thiết có thể tách làm hai trường hợp A đúng hay B đúng, và ta đã chứng minh được riêng rẽ cho từng trường hợp là kết luận C đúng thì C cũng đúng trong cả hai trường hợp.

3.2. Ví dụ minh hoạ việc áp dụng các quy tắc suy diễn

Ví dụ 1: Dùng quy tắc suy diễn, chỉ ra công thức sau là hằng đúng:

$$(A \wedge B) \to (A \wedge B) \qquad (*)$$

Giải:

Công thức (*) có mô hình suy diễn là:

$$\frac{A \wedge B}{\therefore A \vee B} \equiv \frac{\begin{cases} A \wedge B \\ A \wedge B \to A \end{cases}}{\therefore A \vee B}(\text{Kd}) \equiv \frac{A}{\therefore A \vee B} \equiv \mathbf{1}$$

Vậy (*) là hằng đúng.

Ví dụ 2: Chứng minh công thức sau là hằng đúng:

$$(X_1 \vee X_2) \wedge \left((X_1 \vee X_2) \to \overline{X_3 \wedge X_4}\right) \to \left(\overline{X_3 \wedge X_4}\right) \qquad (*)$$

Giải: Mô hình suy diễn của (*) là:

$$\frac{\begin{cases}(X_1 \vee X_2) \\ (X_1 \vee X_2) \to \left(\overline{X_3 \wedge X_4}\right)\end{cases}}{\therefore\ \overline{X_3 \wedge X_4}}(\text{Kđ}) \equiv \frac{\overline{X_3 \wedge X_4}}{\therefore\ \overline{X_3 \wedge X_4}} \equiv 1$$

Vậy (*) hằng đúng.

Ví dụ 3: Chỉ ra suy luận dưới đây là đúng:

"Bình đi chơi thì Bình không học toán rời rạc. Bình không học toán rời rạc thì Bình thi trượt toán rời rạc. Mà Bình thích đi chơi. Vậy Bình thi trượt toán rời rạc."

Giải:

Đặt X_1: Bình đi chơi;

X_2: Bình không học toán rời rạc;

X_3: Bình thi trượt toán rời rạc.

Đoạn văn trên có thể viết dưới dạng mô hình suy diễn sau đây:

$$\frac{\begin{cases}X_1 \to X_2 \\ X_2 \to X_3\end{cases} \\ X_1}{\therefore\ X_3}(\text{Tđl}) \equiv \frac{\begin{cases}X_1 \to X_3 \\ X_1\end{cases}}{\therefore\ X_3} \equiv \frac{X_3}{\therefore\ X_3} \equiv 1$$

Vậy suy luận của đoạn văn trên là đúng.

Ví dụ 4: "Nếu được thưởng cuối năm An sẽ đi Đà Lạt. Nếu đi Đà Lạt thì An sẽ thăm Thiền Viện. Mà An không thăm Thiền Viện. Vậy An không được thưởng cuối năm."

Suy luận của đoạn văn trên có đúng không?

Giải:

Đặt X_1: An được thưởng cuối năm;

X_2: An sẽ đi Đà Lạt;

X_3: An sẽ thăm Thiền Viện.

Đoạn văn trên tương đương với mô hình suy diễn dưới đây:

$$\begin{array}{l} X_1 \to X_2 \\ \begin{cases} X_2 \to X_3 \\ \overline{X_3} \end{cases} \\ \hline \therefore\ \overline{X_1} \end{array} (Pđ) \equiv \begin{array}{l} X_1 \to X_2 \\ \overline{X_2} \\ \hline \therefore\ \overline{X_1} \end{array} (Pđ) \equiv \frac{\overline{X_1}}{\therefore\ \overline{X_1}} \equiv 1$$

Suy luận của đoạn văn trên là đúng.

Ví dụ 5: Chỉ ra

$$\left((A \to B) \wedge (B \to C) \wedge (D \vee \overline{C}) \wedge (\overline{D} \vee E) \wedge \overline{E}\right) \to \overline{A} \text{ hằng đúng.}$$

Giải: Thay $D \vee \overline{C} \equiv C \to D$ và $\overline{D} \vee E \equiv D \to E$.

Công thức trên có mô hình suy diễn là:

$$\begin{array}{l} \begin{cases} A \to B \\ B \to C \\ C \to D \\ D \to E \end{cases} \\ \overline{E} \\ \hline \therefore\ \overline{A} \end{array} (Tđl) \equiv \begin{array}{l} A \to E \\ \overline{E} \\ \hline \therefore\ \overline{A} \end{array} (Pđ) \equiv \frac{\overline{A}}{\therefore\ \overline{A}} \equiv 1$$

Vậy công thức trên là hằng đúng.

Ví dụ 6: Suy luận dưới đây có đúng không?

$$\begin{array}{l} A \to B \\ B \to C \\ C \to D \\ A \\ \hline \therefore\ D \end{array} (Tđl) \equiv \begin{array}{l} A \to D \\ A \\ \hline \therefore\ D \end{array} (Kđ) \equiv \frac{D}{\therefore\ D} \equiv 1.$$

Suy luận đúng.

Ví dụ 7: Chỉ ra công thức dưới đây là hằng đúng:

$$\left((A \to B) \wedge (\overline{A} \to C) \wedge (C \to D)\right) \to \left(\overline{B} \to D\right).$$

Giải: Vì $\overline{\overline{B} \to D} \equiv \overline{B} \wedge \overline{D}$ nên áp dụng quy tắc mâu thuẫn ta suy luận như sau:

$$
\begin{array}{l}
A \to B \\
\overline{A} \to C \\
C \to D \\
\hline
\therefore \overline{B} \to D
\end{array}
\equiv
\begin{array}{l}
A \to B \\
\overline{A} \to C \\
C \to D \\
\overline{B} \\
\overline{D} \\
\hline
\therefore \mathbf{0}
\end{array}
\equiv
\begin{array}{l}
\begin{cases} A \to B \\ \overline{B} \end{cases} \\
\overline{A} \to C \\
\begin{cases} C \to D \\ \overline{D} \end{cases} \\
\hline
\therefore \mathbf{0}
\end{array}
\equiv
\begin{array}{l}
\begin{cases} \overline{A} \\ \overline{A} \to C \end{cases} \\
\overline{C} \\
\hline
\therefore \mathbf{0}
\end{array}
(\text{Kđ}) \equiv
$$

$$
\equiv
\begin{array}{c}
\overline{C} \\
C \\
\hline
\therefore \mathbf{0}
\end{array}
\equiv
\begin{array}{c}
C \wedge \overline{C} \\
\hline
\therefore \mathbf{0}
\end{array}
\equiv
\begin{array}{c}
\mathbf{0} \\
\hline
\therefore \mathbf{0}
\end{array}
\equiv \mathbf{1}
$$

Vậy công thức trên là hằng đúng.

Ví dụ 8: Chỉ ra suy luận dưới đây là đúng:

$$
\begin{array}{l}
X \to Y \\
\overline{X} \to Z \\
Z \to Z_1 \\
\hline
\therefore \overline{Y} \to Z_1
\end{array}
\qquad (*)
$$

Giải:

Dùng quy tắc mâu thuẫn ta có (*) tương đương với

$$
\begin{array}{l}
X \to Y \\
\begin{cases} \overline{X} \to Z \\ Z \to Z_1 \end{cases} \\
\overline{Y} \wedge \overline{Z_1} \\
\hline
\therefore \mathbf{0}
\end{array}
(\text{Tđl}) \equiv
\begin{array}{l}
\begin{cases} X \to Y \\ \overline{Y} \end{cases} \\
\begin{cases} \overline{X} \to Z_1 \\ \overline{Z_1} \end{cases} \\
\hline
\therefore \mathbf{0}
\end{array}
(\text{Pđ}) \equiv
\begin{array}{c}
\overline{X} \\
X \\
\hline
\therefore \mathbf{0}
\end{array}
\equiv
\begin{array}{c}
\overline{X} \wedge X \\
\hline
\therefore \mathbf{0}
\end{array}
\equiv
\begin{array}{c}
\mathbf{0} \\
\hline
\therefore \mathbf{0}
\end{array}
\equiv \mathbf{1}.
$$

Suy luận trên là đúng.

Ví dụ 9: "Nếu An được lên chức và làm việc nhiều thì An được tăng lương. An được tăng lương thì An mua xe máy. Mà An không mua xe máy. Vậy An không được lên chức hay An không làm việc nhiều."

Suy luận trên có đúng không?

Giải:

Đặt X_1: An được lên chức;

X_2: An làm việc nhiều;

X_3: An được tăng lương;

X_4: An mua xe máy.

Suy luận trên tương đương với:

$$\frac{\begin{array}{l}(X_1 \wedge X_2) \to X_3\\ X_3 \to X_4\\ \overline{X}_4\end{array}}{\therefore\ \overline{X}_1 \vee \overline{X}_2}(\text{Mt}) \equiv \frac{\begin{array}{l}(X_1 \wedge X_2) \to X_3\\ \begin{cases}X_3 \to X_4\\ \overline{X}_4\end{cases}\\ X_1 \wedge X_2\end{array}}{\therefore\ \mathbf{0}}(\text{Pđ}) \equiv$$

$$\equiv \frac{\begin{array}{l}\begin{cases}(X_1 \wedge X_2) \to X_3\\ \overline{X}_3\end{cases}\\ X_1 \wedge X_2\end{array}}{\therefore\ \mathbf{0}}(\text{Pđ}) \equiv \frac{\begin{array}{l}\overline{X_1 \wedge X_2}\\ X_1 \wedge X_2\end{array}}{\therefore\ \mathbf{0}} \equiv \frac{\mathbf{0}}{\therefore\ \mathbf{0}} \equiv \mathbf{1}$$

Vậy suy luận trên đúng.

Ví dụ 10: Suy luận dưới đây có đúng không?

"Nếu muốn đi họp sáng thứ ba thì An phải dậy sớm. Nếu An đi nghe nhạc tối thứ hai thì An sẽ về muộn. Nếu An về muộn và thức dậy sớm thì An phải đi họp sáng thứ ba và chỉ được ngủ dưới 7 giờ trong ngày. Nhưng An không thể đi họp nếu chỉ ngủ dưới 7 giờ. Vậy hoặc An không đi nghe nhạc tối thứ hai hoặc An phải bỏ họp sáng thứ ba."

Giải:

Đặt X_1: An muốn đi họp sáng thứ ba;

X_2: An phải dậy sớm;

X_3: An đi nghe nhạc tối thứ hai;

X_4: An sẽ về muộn;

X_5: An ngủ dưới 7 giờ trong một ngày.

Khi đó suy luận trên tương đương với

$$\frac{\begin{array}{l}X_1 \to X_2\\ X_3 \to X_4\\ (X_4 \wedge X_2) \to (X_1 \wedge X_5)\\ X_5 \to \overline{X}_1\end{array}}{\therefore\ \overline{X}_3 \vee \overline{X}_1}(\text{Mt}) \equiv \frac{\begin{array}{l}X_1 \to X_2\\ X_3 \to X_4\\ (X_4 \wedge X_2) \to (X_1 \wedge X_5)\\ X_5 \to \overline{X}_1\\ \overline{\overline{X}_3 \vee \overline{X}_1}\end{array}}{\mathbf{0}} \equiv$$

$$\equiv \frac{\begin{array}{l}\begin{cases}X_1 \to X_2\\ X_1\end{cases}\\ \begin{cases}X_3 \to X_4\\ X_3\end{cases}\\ (X_4 \wedge X_2) \to (X_1 \wedge X_5)\\ X_5 \to \overline{X}_1\end{array}}{\therefore\ 0}\ (\text{Kđ}) \equiv \frac{\begin{array}{l}\begin{cases}X_4 \wedge X_2\\ (X_4 \wedge X_2) \to (X_1 \wedge X_5)\end{cases}\\ \begin{cases}X_5 \to \overline{X}_1\\ X_1 \equiv \overline{\overline{X}}_1\end{cases}\end{array}}{\therefore\ 0}\ (\text{Kđ và Pđ})$$

$$\equiv \frac{\begin{array}{l}X_1 \wedge X_5\\ \overline{X}_5\end{array}}{\therefore\ 0} \equiv \frac{X_5 \wedge \overline{X}_5}{\therefore\ 0} \equiv \frac{0}{\therefore\ 0} \equiv 1$$

Suy luận trên đúng.

Ví dụ 11: "Nếu An đi làm về muộn thì vợ An sẽ rất giận dữ. Nếu Bình thường xuyên vắng nhà thì vợ Bình cũng sẽ rất giận dữ. Nếu vợ Bình hoặc vợ An giận dữ thì cô Hà bạn họ nhận được than phiền, mà cô Hà không hề nhận được lời than phiền. Vậy An đi làm về sớm và Bình rất ít khi vắng nhà."

Dùng quy tắc suy diễn để chỉ ra suy luận trên là đúng.

Giải:

Đặt X_1: An đi làm về muộn;

X_2: Vợ An sẽ rất giận dữ;

X_3: Bình thường xuyên vắng nhà;

X_4: Vợ Bình cũng rất giận dữ;

X_5: Cô Hà bạn họ nhận được lời than phiền.

Suy luận trên tương đương với

$$\frac{\begin{array}{l}X_1 \to X_2\\ X_3 \to X_4\\ \begin{cases}(X_2 \vee X_4) \to X_5\\ \overline{X}_5\end{cases}\end{array}}{\therefore\ \overline{X}_1 \wedge \overline{X}_3}\ (\text{Pđ}) \equiv \frac{\begin{array}{l}X_1 \to X_2\\ \begin{cases}X_3 \to X_4\\ \overline{X_2 \vee X_4}\end{cases}\end{array}}{\therefore\ \overline{X}_1 \wedge \overline{X}_3} \equiv$$

$$\equiv \frac{\begin{cases} X_1 \to X_2 \\ \overline{X}_2 \end{cases} \quad \begin{cases} X_3 \to X_4 \\ \overline{X}_4 \end{cases}}{\therefore \overline{X}_1 \wedge \overline{X}_3} (\text{Pd}) \equiv \frac{\overline{X}_1 \wedge \overline{X}_3}{\therefore \overline{X}_1 \wedge \overline{X}_3} \equiv \mathbf{1}$$

Vậy suy luận trên là đúng.

Ví dụ 12: Chứng minh

$$\left((X \to Y) \wedge (Y \to Z) \wedge (Z_1 \vee \overline{Z}) \wedge (\overline{Z}_1 \vee Z_2) \wedge \overline{Z}_2\right) \to \overline{X} \equiv \mathbf{1}$$

Giải: Công thức trên tương đương với

$$\frac{\begin{matrix} X \to Y \\ Y \to Z \\ Z_1 \vee \overline{Z} \\ \overline{Z}_1 \vee Z_2 \\ \overline{Z}_2 \end{matrix}}{\therefore \overline{X}} \equiv \frac{\begin{cases} X \to Y \\ Y \to Z \\ Z \to Z_1 \\ Z_1 \to Z_2 \end{cases} \\ \overline{Z}_2}{\therefore \overline{X}} (\text{Tđl, Pd}) \equiv \frac{\begin{matrix} X \to Z_2 \\ \overline{Z}_2 \end{matrix}}{\therefore \overline{X}} (\text{Pd}) \equiv \frac{\overline{X}}{\therefore \overline{X}} \equiv \mathbf{1}$$

Vậy công thức trên là hằng đúng.

Ví dụ 13: Sử dụng quy tắc suy diễn mâu thuẫn, chỉ ra suy luận dưới đây là đúng:

$$\frac{\begin{matrix} X \to Y \\ \overline{X} \to Z \\ Z \to Z_1 \end{matrix}}{\therefore \overline{Y} \to Z_1} \equiv \frac{\begin{matrix} X \to Y \\ \overline{X} \to Z \\ Z \to Z_1 \\ \overline{\overline{Y} \to Z_1} \end{matrix}}{\therefore \mathbf{0}} \equiv \frac{\begin{cases} X \to Y \\ \overline{Y} \end{cases} \\ \overline{X} \to Z \\ \begin{cases} Z \to Z_1 \\ \overline{Z}_1 \end{cases}}{\therefore \mathbf{0}} (\text{Pd}) \equiv \frac{\begin{cases} \overline{X} \to Z \\ \overline{X} \end{cases} \\ \overline{Z}}{\therefore \mathbf{0}} (\text{Kđ})$$

$$\equiv \frac{Z \wedge \overline{Z}}{\therefore \mathbf{0}} \equiv \frac{\mathbf{0}}{\therefore \mathbf{0}} \equiv \mathbf{1}.$$

Ví dụ 14: Nếu nghệ sĩ nhân dân (NSND) X không trình diễn hay số vé bán ra ít hơn 50 vé thì đêm biểu diễn ở Công viên Hồ Tây bị huỷ và ông bầu rất buồn. Nếu đêm biểu diễn huỷ bỏ thì phải trả tiền vé lại cho người xem. Tiền vé đã không trả lại cho người xem.

Vậy NSND X đã trình diễn. Suy luận trên có đúng không?

Giải:

Đặt X_1: NSND X đã trình diễn;

X_2: Số vé bán ra ít hơn 50 vé;

X_3: Đêm biểu diễn ở Công viên Hồ Tây bị huỷ;

X_4: Ông bầu rất buồn;

X_5: Trả tiền vé lại cho người xem.

Suy luận trên tương đương với

$$\frac{\begin{array}{l}(\overline{X_1}\vee X_2)\to(X_3\wedge X_4)\\ \begin{cases}X_3\to X_5\\ \overline{X_5}\end{cases}\end{array}}{\therefore\ X_1}\text{(Pđ)} \equiv \frac{\begin{array}{l}(\overline{X_1}\vee X_2)\to(X_3\wedge X_4)\\ \overline{X_3}\end{array}}{\therefore\ X_1}\text{(Rg)}$$

$$\frac{\begin{array}{l}\begin{cases}(\overline{X_1}\vee X_2)\to(X_3\wedge X_4)\\ (X_3\wedge X_4)\to X_3\end{cases}\\ \overline{X_3}\end{array}}{\therefore\ X_1}\text{(Tđl)} \equiv \frac{\begin{cases}(\overline{X_1}\vee X_2)\to X_3\\ \overline{X_3}\end{cases}}{\therefore\ X_1}\text{(Pđ)}$$

$$\equiv \frac{\overline{\overline{X_1}\vee X_2}}{\therefore\ X_1} \equiv \frac{X_1\wedge\overline{X_2}}{\therefore\ X_1} \equiv \frac{\begin{cases}(X_1\wedge\overline{X_2})\to X_1\\ (X_1\wedge\overline{X_2})\end{cases}}{\therefore\ X_1}\text{(Kđ)} \equiv \frac{X_1}{\therefore\ X_1} \equiv \mathbf{1}$$

Vậy suy luận trên là đúng.

Ví dụ 15: Chỉ ra công thức dưới đây là hằng đúng:

$$\big(X_1\wedge(X_1\to X_2)\wedge(X_3\wedge X_4)\wedge(X_4\to\overline{X_2})\big)\to(X_3\vee X_5)$$

Giải:

$$\frac{\begin{array}{l}\begin{cases}X_1\\ X_1\to X_2\end{cases}\\ X_3\vee X_4\\ X_4\to\overline{X_2}\end{array}}{\therefore\ X_3\vee X_5}\text{(Kđ)} \equiv \frac{\begin{array}{l}X_3\vee X_4\\ \begin{cases}X_4\to\overline{X_2}\\ X_2\end{cases}\end{array}}{\therefore\ X_3\vee X_5}\text{(Pđ)} \equiv \frac{\begin{array}{l}X_3\vee X_4\\ \overline{X_4}\end{array}}{\therefore\ X_3\vee X_5}\text{(Tdlr)} \equiv$$

$$\equiv \frac{X_3}{\therefore X_3 \vee X_5}(\text{Cg}) \equiv \mathbf{1}$$

Vậy suy luận là đúng.

Ví dụ 16: Suy luận dưới đây có đúng không?

$$\begin{array}{l} (\overline{X}_1 \vee X_2) \to X_3 \\ X_3 \to (X_4 \vee X_5) \\ \overline{X}_4 \wedge \overline{X}_6 \\ \overline{X}_6 \to \overline{X}_5 \\ \hline \qquad \therefore X_1 \end{array} \qquad (*)$$

Giải:

$$\frac{\begin{array}{l} \begin{cases} (\overline{X}_1 \vee X_2) \to X_3 \\ X_3 \to (X_4 \vee X_5) \end{cases} \\ \overline{X}_4 \wedge \overline{X}_6 \\ \overline{X}_6 \to \overline{X}_5 \end{array}}{\therefore X_1} \equiv (\text{Tđl}) \frac{\begin{array}{l} (\overline{X}_1 \vee X_2) \to (X_4 \vee X_5) \\ \overline{X}_4 \\ \begin{cases} \overline{X}_6 \\ \overline{X}_6 \to \overline{X}_5 \end{cases} \end{array}}{\therefore X_1}(\text{Kđ})$$

$$\frac{\begin{cases} (\overline{X}_1 \vee X_2) \to (X_4 \vee X_5) \\ \overline{X}_4 \wedge \overline{X}_5 \equiv \overline{X_4 \vee X_5} \end{cases}}{\therefore X_1}(\text{Pđ}) \equiv \frac{\overline{\overline{X}_1 \vee X_2}}{\therefore X_1} \equiv \frac{X_1 \wedge \overline{X}_2}{\therefore X_1} \equiv \mathbf{1}.$$

Suy luận (*) là đúng.

Ví dụ 17: Kiểm tra các suy luận dưới đây:

a) $$\frac{\begin{cases} X \to Y \\ \overline{Y} \\ \overline{Z} \end{cases}}{\therefore \overline{X \vee Z}}(\text{Pđ}) \equiv \frac{\begin{array}{c}\overline{X} \\ \overline{Z}\end{array}}{\therefore \overline{X \vee Z}} \equiv \frac{\overline{X} \wedge \overline{Z}}{\therefore \overline{X \vee Z}} \equiv \frac{\overline{X \vee Z}}{\therefore \overline{X \vee Z}} \equiv \mathbf{1}$$

Suy luận là đúng.

b) $$\frac{\begin{array}{l} X \to Y \\ \begin{cases} Z \to \overline{Y} \\ Z \end{cases} \end{array}}{\therefore \overline{X}}(\text{Kđ}) \equiv \frac{\begin{cases} X \to Y \\ \overline{Y} \end{cases}}{\therefore \overline{X}}(\text{Pđ}) \equiv \frac{\overline{X}}{\therefore \overline{X}} \equiv \mathbf{1}.$$

Suy luận là đúng.

c)

$$\frac{\begin{array}{l} X \to (Y \to Z) \\ \overline{Y} \to \overline{X} \\ X \end{array}}{\therefore\ Z}(\text{Kđ}) \equiv \frac{\begin{array}{l} Y \to Z \\ \overline{Y} \to \overline{X} \end{array}}{\therefore\ Z} \equiv \frac{\begin{array}{l} Y \to Z \\ \begin{cases} \overline{Y} \to \overline{X} \\ X \equiv \overline{\overline{X}} \end{cases} \end{array}}{\therefore\ Z}(\text{Pđ})$$

$$\equiv \frac{\begin{cases} Y \to Z \\ Y \end{cases}}{\therefore\ Z}(\text{Kđ}) \equiv \frac{Z}{\therefore\ Z} \equiv \mathbf{1}$$

Suy luận là đúng.

d)

$$\frac{\begin{array}{l} X_1 \to (X_2 \to X_3) \\ X_1 \vee X_4 \\ X_5 \to X_2 \\ \overline{X}_4 \end{array}}{\therefore\ \overline{X}_3 \to \overline{X}_5} \equiv \frac{\begin{array}{l} X_1 \to (X_2 \to X_3) \\ \begin{cases} X_1 \vee X_4 \\ \overline{X}_4 \end{cases} \\ X_5 \to X_2 \end{array}}{\therefore\ \overline{X}_3 \to \overline{X}_5}(\text{Tđlr})$$

$$\equiv \frac{\begin{cases} X_2 \to X_3 \\ X_5 \to X_2 \end{cases}}{\therefore\ \overline{X}_3 \to \overline{X}_5}(\text{Tđl}) \equiv \frac{X_5 \to X_3}{\therefore\ \overline{X}_3 \to \overline{X}_5}$$

$$\equiv \frac{X_3 \vee \overline{X}_5}{\therefore\ \overline{X}_3 \to \overline{X}_5} \equiv \frac{\overline{X}_3 \to \overline{X}_5}{\therefore\ \overline{X}_3 \to \overline{X}_5} \equiv \mathbf{1}$$

Suy luận là đúng.

e)

$$\frac{\begin{array}{l} X \vee Y \\ \begin{cases} \overline{X} \vee Z \\ \overline{Z} \end{cases} \end{array}}{\therefore\ Y}(\text{Tđlr}) \equiv \frac{\begin{cases} X \vee Y \\ \overline{X} \end{cases}}{\therefore\ Y}(\text{Tđlr}) \equiv \frac{Y}{\therefore\ Y} \equiv \mathbf{1}$$

Suy luận là đúng.

f)

$$\dfrac{\begin{array}{l} X_1 \wedge X_2 \\ X_1 \to (X_3 \wedge X_2) \\ X_3 \to (X_4 \vee X_5) \\ \overline{X}_4 \end{array}}{\therefore\ X_5} \equiv \dfrac{\begin{array}{l} \begin{cases} (X_1 \wedge X_2) \to X_1 \\ (X_1 \wedge X_2) \end{cases} \\ X_1 \to (X_3 \wedge X_2) \\ X_3 \to (X_4 \vee X_5) \\ \overline{X}_4 \end{array}}{\therefore\ X_5} (\text{Kđ})$$

$$\equiv \dfrac{\begin{array}{l} \begin{cases} X_1 \\ X_1 \to (X_3 \wedge X_2) \end{cases} \\ X_3 \to (X_4 \vee X_5) \\ \overline{X}_4 \end{array}}{\therefore\ X_5} (\text{Kđ}) \equiv \dfrac{\begin{array}{l} X_3 \wedge X_2 \\ X_3 \to (X_4 \vee X_5) \\ \overline{X}_4 \end{array}}{\therefore\ X_5}$$

$$\equiv \dfrac{\begin{array}{l} \begin{cases} (X_3 \wedge X_2) \to X_3 \\ (X_3 \wedge X_2) \end{cases} \\ X_3 \to (X_4 \vee X_5) \\ \overline{X}_4 \end{array}}{\therefore\ X_5} (\text{Kđ}) \equiv \dfrac{\begin{array}{l} \begin{cases} X_3 \\ X_3 \to (X_4 \vee X_5) \end{cases} \\ \overline{X}_4 \end{array}}{\therefore\ X_5} (\text{Kđ})$$

$$\equiv \dfrac{\begin{cases} X_4 \vee X_5 \\ \overline{X}_4 \end{cases}}{\therefore\ X_5} \equiv (\text{Tdlr}) \dfrac{X_5}{\therefore\ X_5} \equiv \mathbf{1}$$

Suy luận là đúng.

Ví dụ 18: Chứng minh công thức sau hằng đúng

$\left((X_1 \to X_2) \wedge (\overline{X}_3 \vee X_4) \wedge (X_1 \vee X_3)\right) \to (\overline{X}_2 \to X_4)$ bằng cách:

a) Lập bảng.

b) Đưa công thức về DCTH.

c) Dùng quy tắc suy diễn.

Giải:

c) Dùng quy tắc suy biến

Công thức trên tương đương với:

$$
\frac{\begin{array}{l} X_1 \to X_2 \\ \overline{X}_3 \vee X_4 \\ X_1 \vee X_3 \end{array}}{\therefore\ \overline{X}_2 \to X_4}
\equiv
\frac{\begin{array}{l} X_1 \to X_2 \\ \overline{X}_3 \vee X_4 \\ X_1 \vee X_3 \\ \overline{\overline{X}_2 \to X_4} \end{array}}{\therefore\ \mathbf{0}}
\equiv
\frac{\begin{array}{l} X_1 \to X_2 \\ \overline{X}_3 \vee X_4 \\ X_1 \vee X_3 \\ \overline{X}_2 \wedge \overline{X}_4 \end{array}}{\therefore\ \mathbf{0}}
\equiv
\frac{\begin{array}{l} \begin{cases} X_1 \to X_2 \\ \overline{X}_2 \end{cases} \\ \begin{cases} \overline{X}_3 \vee X_4 \\ \overline{X}_4 \end{cases} \\ X_1 \vee X_3 \end{array}}{\therefore\ \mathbf{0}} (\text{Pd}, \text{Tdlr})
$$

$$
\equiv
\frac{\begin{array}{l} \overline{X}_1 \\ \begin{cases} \overline{X}_3 \\ X_1 \vee X_3 \end{cases} \end{array}}{\therefore\ \mathbf{0}}
\equiv
\frac{\overline{X}_1 \wedge X_1}{\therefore\ \mathbf{0}} \equiv \mathbf{1}
$$

Bạn đọc tự chứng minh câu a) và b).

Ví dụ 19: Tìm phản ví dụ cho suy luận dưới đây:

$$
\text{a)}\ \frac{\begin{array}{l} X \equiv Y \\ Y \to Z_1 \\ Z_1 \vee \overline{Z}_2 \\ \overline{Z}_2 \to Y \end{array}}{\therefore\ Z_2}
\qquad\qquad
\text{b)}\ \frac{\begin{array}{l} X_1 \\ X_1 \to X_2 \\ X_1 \to (X_3 \vee \overline{X}_2) \\ \overline{X}_3 \vee \overline{X}_4 \end{array}}{\therefore\ X_4}
$$

Giải:

a) Chọn $X \equiv Y = 1$ ta suy ra $Z_1 = 1$ và $Z_2 = 0$. Như vậy giả thiết đúng, kết luận sai. Hay suy luận trên không đúng.

b) Suy luận này sai vì chọn $X_1 = X_2 = X_3 = 1$ và $X_4 = 0$.

Khi đó giả thiết đúng nhưng kết luận sai.

Ví dụ 20: Chỉ ra suy luận dưới đây là sai:

"Ông Minh đã khẳng định rằng, nếu không được tăng lương thì ông ta sẽ xin nghỉ việc. Mặt khác, nếu ông Minh nghỉ việc mà vợ ông ta bị mất việc thì phải bán xe máy. Biết rằng, nếu vợ ông Minh hay đi làm muộn thì sẽ mất việc và cuối cùng ông Minh đã được tăng lương. Vậy nếu ông Minh không bán xe thì vợ ông ta không đi làm muộn."

Giải:

Đặt X_1: Ông Minh được tăng lương;

X_2: Ông Minh xin nghỉ việc;

X_3: Vợ ông Minh bị mất việc;

X_4: Ông Minh phải bán xe;

X_5: Vợ ông Minh đi làm muộn.

Đoạn văn trên tương đương với mô hình suy diễn dưới đây:

$$\begin{array}{l} \overline{X}_1 \to X_2 \\ (X_2 \wedge X_3) \to X_4 \\ X_5 \to X_3 \\ \underline{X_1 \qquad\qquad\qquad} \\ \therefore\ \overline{X}_4 \to \overline{X}_5 \end{array}$$

Chọn $X_4 = 0$ và $X_5 = 1$ thì kết luận sai.

Từ $X_4 = 0$ và $X_5 = 1$ ta có $X_3 = 1$, $X_2 = 0$ và $X_1 = 1$. Với giá trị này thì giả thiết đúng. Vậy giả thiết đúng mà kết luận sai thì mô hình trên là sai. Hay đoạn văn trên không đúng.

Chương 5
LÔGIC VỊ TỪ

§1. ĐỊNH NGHĨA VỊ TỪ

Giả sử $\mathcal{M}$ là một tập hợp gồm các phần tử nào đấy. $\mathcal{M}$ thường gọi là trường, các phần tử của $\mathcal{M}$ thường ký hiệu là các chữ a, b, c, ...

Trên trường $\mathcal{M}$ ta thành lập các mệnh đề, mệnh đề gồm một phần tử sẽ nói lên tính chất của phần tử; mệnh đề gồm hai phần tử tham gia sẽ nói lên quan hệ giữa các phần tử.

Chẳng hạn, lấy trường $\mathcal{M} = \{1, 2, 3, ...\}$, trên trường này ta thành lập các mệnh đề sau đây: "3 là số nguyên tố", "1 > 2", ...

Mệnh đề thứ nhất đúng, nói lên tính chất của số 3.

Mệnh đề thứ hai sai, nói lên quan hệ giữa hai số 1 và 2.

Tóm lại, đối với các phần tử của trường $\mathcal{M}$ ta có thể thiết lập được các mệnh đề.

Các mệnh đề ở đây chỉ quan tâm tới sự đúng, sai mà không quan tâm tới nội dung của mệnh đề; hay nó thoả mãn luật bài chung, tức là hoặc đúng, hoặc sai, cũng như trong đại số mệnh đề ta không xét mệnh đề không đúng mà cũng không sai.

Định nghĩa 1: Biểu thức P(x) gọi là vị từ xác định trên trường $\mathcal{M}$ nếu khi thay x bởi phần tử bất kỳ của trường $\mathcal{M}$ thì P(x) sẽ trở thành mệnh đề xác định trên trường $\mathcal{M}$.

Chẳng hạn: $\mathcal{M} = \{1, 2, ...\}$.

Khi đó biểu thức "x là số nguyên tố" là một vị từ xác định trên trường $\mathcal{M}$, bởi vì khi thay x bởi 3 ta được mệnh đề là "3 là số nguyên tố".

Như vậy P(x) có thể xem như một ánh xạ (hay một hàm) xác định trên trường $\mathcal{M}$ và nhận giá trị trong tập hợp {đúng, sai}, loại vị từ nói trên gọi là vị từ một ngôi. Vị từ một ngôi P(x) chỉ nói lên tính chất của một phần tử thuộc $\mathcal{M}$.

Để nói lên quan hệ giữa các phần tử của trường $\mathcal{M}$, người ta cần đưa vào vị từ nhiều ngôi.

Định nghĩa 2: Biểu thức $P(x_1, x_2, ..., x_n)$ gọi là vị từ xác định trên tập $\mathcal{M}^n = \mathcal{M}_1 \times \mathcal{M}_2 \times ... \times \mathcal{M}_n$ nếu ta thay x_i $(i = 1, ..., n)$ bởi phần tử bất kỳ của trường $\mathcal{M}_i$ $(i = 1, ..., n)$ thì ta được các mệnh đề xác định trên $\mathcal{M}^n = \mathcal{M}_1 \times \mathcal{M}_2 \times ... \times \mathcal{M}_n$. $P(x_1, x_2, ..., x_n)$ gọi là vị từ n ngôi.

Ta có thể nói vị từ P trên trường $\mathcal{M}$ không phải là một mệnh đề hoặc đúng, hoặc sai. Nhưng khi thay biến của nó bởi phần tử trên trường $\mathcal{M}$ thì nó trở thành mệnh đề hoặc đúng, hoặc sai. Mệnh đề đó còn gọi là vị từ cụ thể trên trường $\mathcal{M}$.

$\mathcal{M}^n = \mathcal{M}_1 \times \mathcal{M}_2 \times ... \times \mathcal{M}_n$ là tập tích Đề-các, còn $x_1, x_2, ..., x_n$ trong vị từ n ngôi ta gọi là các biến tử. Các biến tử $x_1, x_2, ..., x_n$ sẽ nhận các phần tử $a, b, c, ... \in \mathcal{M}$ ta gọi là các hằng tử.

Các vị từ thường ký hiệu bởi các chữ P, F, Q, R (đôi khi cả chỉ số) và ta cũng gọi chúng là các biến vị từ.

Ta ký hiệu các chữ X, Y, Z (đôi khi cả chỉ số) là các mệnh đề sơ cấp và gọi là các biến mệnh đề (xem trong Lôgic mệnh đề).

Định nghĩa 3 (Định nghĩa công thức)

1) Mỗi một biến mệnh đề hoặc biến vị từ gọi là một công thức.

2) Nếu A, B là các công thức thì $(A \wedge B)$, $(A \vee B)$, $A \to B)$, $(\overline{A})$ cũng là những công thức.

3) Nếu A là công thức thì biểu thức: $(\forall x)$ A hoặc $(\exists x)$ A cũng là công thức.

Từ định nghĩa công thức ta thấy lôgic vị từ gồm các phép toán sau: $\wedge$ (hội), $\vee$ (tuyển), $\to$ (kéo theo), — (phủ định), ký hiệu $\forall$ (với mọi) và ký hiệu $\exists$ (tồn tại}. Các phép toán $\wedge$, $\vee$, $\to$, — định nghĩa như trong lôgic mệnh đề. Hai ký hiệu $\forall$, $\exists$ gọi là các lượng từ định nghĩa như sau:

Giả sử A là một công thức xác định trên trường $\mathcal{M}$. Khi đó biểu thức $(\forall x)A$ là một mệnh đề. Mệnh đề này đúng khi A đúng với mọi phần tử của trường $\mathcal{M}$ và sai trong trường hợp ngược lại. Nó không phụ thuộc vào x và ta diễn đạt bằng lời như sau: "Đối với mọi x, A đúng". Ký hiệu $\forall$ gọi là lượng từ toàn thể.

Tương tự ta định nghĩa ký hiệu $\exists$ như sau:

$(\exists x)A$ là một mệnh đề. Mệnh đề này đúng nếu có một phần tử của trường $\mathcal{M}$ mà A đúng và sai trong trường hợp ngược lại. Nó phụ thuộc vào x và có diễn đạt bằng lời như sau: "Tồn tại x, A đúng". Ký hiệu $\exists$ gọi là lượng từ tồn tại.

Chú ý 1: Trong các công thức $(\forall x)$ A và $(\exists x)$ A thì công thức A gọi là miền tác dụng của các lượng từ $(\forall x)$ và $(\exists x)$.

Chú ý 2: Nếu trong các công thức A của công thức $(\forall x)$ A hoặc $(\exists x)$ A chứa các biến x và có thể thêm một số biến khác không nằm dưới dấu $\forall$, $\exists$ thì biến x gọi là biến ràng buộc, các biến khác gọi là các biến tự do.

Ví dụ 1: $(\forall x)\ A(x) \to F(x)$ là một công thức, đây là một vị từ mà miền tác dụng của lượng từ là toàn thể A, x có mặt trong A gọi là biến ràng buộc. Còn x trong F gọi là biến độc lập (vì không bị ràng buộc bởi $\forall$).

Ví dụ 2: $(\exists x)\ A(x) \vee F(x)$ là một công thức, A(x) là miền tác dụng của $\exists$, biến x trong A gọi là biến ràng buộc, còn biến x trong F gọi là biến độc lập.

Ví dụ 3: $A(x_1 : x_2) \to (\forall x_1)\ B(x_1)$ là một công thức có miền tác dụng là $B(x_1)$, x_1 trong B là biến ràng buộc, x_1 trong A là biến tự do, x_2 trong A cũng là biến tự do.

Như vậy, qua ba ví dụ trên ta nhận thấy một biến trong cùng một công thức lôgic vị từ có thể vừa là tự do vừa là ràng buộc.

§2. KHÁI NIỆM CÔNG THỨC ĐỒNG NHẤT BẰNG NHAU, ĐỒNG NHẤT ĐÚNG VÀ ĐỒNG NHẤT SAI

Định nghĩa 4: Công thức A đồng nhất bằng công thức B (ký hiệu $A \equiv B$) trên trường $\mathcal{M}$ khi và chỉ khi A và B cùng nhận giá trị (đúng, sai) như nhau đối với mọi bộ giá trị (đúng, sai) của các biến mệnh đề và các vị từ cụ thể trên trường $\mathcal{M}$ có mặt trong A và B.

Định nghĩa 5: Công thức A gọi là đồng nhất đúng (hằng đúng) (ký hiệu $\models A$ hoặc $A \equiv \mathbf{1}$) trên trường $\mathcal{M}$ khi và chỉ khi A luôn nhận giá trị đúng với mọi bộ giá trị đúng, sai của các biến mệnh đề và các vị từ cụ thể trên trường $\mathcal{M}$ có mặt trong A.

Định nghĩa 6: Công thức A gọi là đồng nhất sai (hằng sai) (ký hiệu $A \equiv \mathbf{0}$) trên trường $\mathcal{M}$ khi và chỉ khi A luôn nhận giá trị sai với mọi bộ giá trị đúng, sai của các biến mệnh đề và các vị từ cụ thể trên trường $\mathcal{M}$ có mặt trong A.

Chú ý: Từ các định nghĩa ta thấy lớp các lôgic vị từ rộng hơn lớp các lôgic mệnh đề. Do đó nếu A là công thức đồng nhất đúng trong lôgic mệnh đề thì nó cũng là công thức đồng nhất đúng trong lôgic vị từ. Mọi đồng nhất đúng trong lôgic mệnh đề cũng là đồng nhất đúng trong lôgic vị từ.

Chẳng hạn, trong đại số mệnh đề ta có đồng nhất thức

$$A \to B \equiv \overline{A} \vee B \tag{1}$$

thì trong lôgic vị từ ta cũng có đẳng thức trên.

Chính vì vậy, trong lôgic vị từ ta dùng đẳng thức trên để khử phép → để đưa một công thức phức tạp về công thức đơn giản hơn chỉ chứa các phép toán ∧, ∨, —. Tiếp tục dùng các công thức trong đại số mệnh đề đồng thời cũng đúng trong lôgic vị từ:

$$\overline{A \wedge B} \equiv \overline{A} \vee \overline{B} \tag{2}$$

$$\overline{A \vee B} \equiv \overline{A} \wedge \overline{B} \tag{3}$$

để đưa dấu — chỉ liên quan tới các công thức sơ cấp.

Ngoài ra trong lôgic vị từ ta còn có các đồng nhất thức sau:

$$\overline{(\forall x)A} \equiv (\exists x)\overline{A} \tag{4}$$

$$\overline{(\exists x)A} \equiv (\forall x)\overline{A} \tag{5}$$

Như vậy, với mỗi công thức ta đưa về công thức mới đồng nhất bằng nó mà chỉ chứa các phép toán ∧, ∨, — (do 1); lại áp dụng (2) đến (5) ta đưa dấu — chỉ liên quan tới các công thức sơ cấp.

Ví dụ 4: Xét công thức

$$\overline{(\exists x)(A(x) \to (\forall y)B(y))} \equiv \overline{(\exists x)(\overline{A}(x) \vee (\forall y)B(y))}$$

$$\equiv (\forall x)(\overline{\overline{A}}(x) \wedge (\exists y)\overline{B}(y)) \equiv (\forall x)(A(x) \wedge (\exists y)\overline{B}(y))$$

Công thức cuối cùng chứa dấu — chỉ liên quan tới B(y).

Định nghĩa 7: Một công thức mà trong đó chỉ chứa các phép toán ∧, ∨, —; dấu — chỉ liên quan trực tiếp tới từng biến mệnh đề và biến vị từ được gọi là công thức rút gọn.

Vậy, từ lý luận trên ta thấy trong lôgic vị từ mọi công thức đều có công thức rút gọn đồng nhất bằng nó. Chẳng hạn công thức

$$(\forall x)(A(x) \wedge (\exists y)\overline{B}(y))$$

là công thức rút gọn của công thức:

$$\overline{(\exists x)(A(x) \to (\forall y)B(y))}.$$

§3. Ý NGHĨA CÁC VỊ TỪ THEO LÝ THUYẾT TẬP HỢP

3.1. Vị từ một ngôi

Giả sử $\mathcal{M} \neq \varnothing$ là một trường nào đó, P(x) là vị từ xác định trên trường $\mathcal{M}$. Ta ký hiệu tập $E_P = \{x \in \mathcal{M} : P(x) \text{ đúng}\}$, như vậy ứng với lượng từ P(x) xác định trên trường $\mathcal{M}$ ta có tập $E_P \subset \mathcal{M}$.

Điều ngược lại cũng đúng, tức là ứng với một tập hợp con $E \subset \mathcal{M}$ sẽ tồn tại một vị từ P(x) xác định trên $\mathcal{M}$ sao cho $E = E_P$. Thật vậy, với $E \subset \mathcal{M}$, ta xây dựng vị từ P(x) là vị từ "x thuộc E". Khi đó ta thấy vị từ P(x) đúng khi $x \in E$ và sai khi $x \notin E$, rõ ràng $E_P = E$.

Như vậy, giữa các tập hợp con của trường $\mathcal{M}$ với các vị từ một biến xác định trên $\mathcal{M}$ có một tương ứng nào đó.

Giả sử $P_1(x)$, $P_2(x)$ là hai vị từ xác định trên trường $\mathcal{M}$ và giả sử

$$E_{P_1} = \{x \in \mathcal{M} : P_1(x) \text{ đúng}\},\ E_{P_2} = \{x \in \mathcal{M} : P_2(x) \text{ đúng}\}$$

a) Xét vị từ $P(x) = P_1(x) \vee P_2(x)$ và gọi $E_P = \{x \in \mathcal{M} : P(x) \text{ đúng}\}$.

Ta chứng minh rằng $E_P = E_{P_1} \cup E_{P_2}$.

Giả sử $x \in E_P$, khi đó P(x) đúng; hay với x đó thì hoặc $P_1(x)$ đúng, hoặc $P_2(x)$ đúng, hoặc cả hai $P_1(x)$ và $P_2(x)$ đều đúng. Trong mọi trường hợp ta đều có $x \in E_{P_1} \cup E_{P_2}$. Vậy ta có bao hàm thức $E_P \subseteq E_{P_1} \cup E_{P_2}$.

Bao hàm thức ngược lại $E_{P_1} \cup E_{P_2} \subseteq E_P$ là hiển nhiên.

Từ hai bao hàm thức trên ta có $E_P = E_{P_1} \cup E_{P_2}$.

b) Xét vị từ $P(x) = P_1(x) \wedge P_2(x)$.

Ta chứng minh $E_P = E_{P_1} \cap E_{P_2}$. Giả sử $x \in E_P$, khi đó P(x) đúng nên $P_1(x)$ và $P_2(x)$ cũng đúng. Từ đó $x \in E_{P_1}$ và $x \in E_{P_2}$ nên ta có $x \in E_{P_1} \cap E_{P_2}$. Vậy $E_P \subseteq E_{P_1} \cap E_{P_2}$.

Giả sử $x \in E_{P_1} \cap E_{P_2}$, khi đó $x \in E_{P_1}$ và $x \in E_{P_2}$. Vậy $P_1(x)$, $P_2(x)$ đúng tức là P(x) đúng và $x \in E_P$.

Vậy $E_P = E_{P_1} \cap E_{P_2}$.

c) Xét vị từ $\overline{P}(x)$.

Giả sử $E_P = \{x \in \mathcal{M} : P(x) \text{ đúng}\}$. Khi đó $E_{\overline{P}(x)} = \mathcal{M} \setminus E_P$.

Thật vậy, $x \in E_{\overline{P}(x)} \Leftrightarrow \overline{P}(x)$ đúng $\Leftrightarrow P(x)$ sai

$\Leftrightarrow x \in \mathcal{M} \setminus E_P$.

3.2. Mở rộng cho vị từ n ngôi

Giả sử có $P(x_1, x_2, ..., x_n)$, $x_i \in \mathcal{M}_i$ là vị từ n ngôi xác định trên trường $\mathcal{M}^n = \mathcal{M}_1 \times \mathcal{M}_2 \times ... \times \mathcal{M}_n$.

Ta định nghĩa $E_P = \{(x_1, x_2, ..., x_n) \in \mathcal{M}^n : P(x_1, x_2, ..., x_n) \text{ đúng}\}$, ở đây ta chỉ xét vị từ hai ngôi $P(x, y)$ trên trường $\mathcal{M}_1 \times \mathcal{M}_2$ là tập hợp các điểm trên mặt phẳng mà hoành độ $x \in \mathcal{M}_1$, tung độ $y \in \mathcal{M}_2$ ($\mathcal{M}_1, \mathcal{M}_2$ chính là trục hoành và trục tung tương ứng).

Giả sử $S \subset \mathcal{M}_1 \times \mathcal{M}_2$, ta ký hiệu hình chiếu của S trên trục hoành là $hc_1S = \{x \in \mathcal{M}_1 : \exists y \text{ để } (x, y) \in S\}$; tương tự $hc_2S = \{y \in \mathcal{M}_2 : \exists x \text{ để } (x, y) \in S\}$ là hình chiếu của S trên trục tung.

a) Xét lượng từ: $(\exists x)\, P(x, y) =: P_1(y)$

Theo định nghĩa hình chiếu trên thì: $E_{P_1} = hc_2E_P$, trong đó $E_{P_1} = \{y \in \mathcal{M}_2 : P_1(y) \text{ đúng}\}$ và $E_P = \{(x, y) \in \mathcal{M}_2 : P(x, y) \text{ đúng}\}$.

Thật vậy, $y \in E_{P_1} \Leftrightarrow P_1(y)$ đúng $\Leftrightarrow \exists x$ để $P(x, y)$ đúng

$\Leftrightarrow \exists x$ để $(x, y) \in E_P \Leftrightarrow y \in hc_2E_P$, hay $E_{P_1} = hc_2E_P$.

b) Xét lượng từ $(\forall x)\, P(x, y) =: P_2(y)$

Để có $E_{P_2}(y)$ ta biến đổi như sau:

$$P_2(y) = \overline{(\exists x)\,\overline{P}(x, y)}$$

Đặt $(\exists x)\, \overline{P}(x,y) = Q(y)$. Khi đó $P_2(y) = \overline{Q(y)}$, vì $E_{\overline{Q}} = \mathcal{M}^2 \setminus E_Q$ nên $E_{P_2(y)} = \mathcal{M}^2 \setminus E_Q$.

Nhưng $E_{Q(y)} = hc_2\, E_{\overline{P}} = hc_2(\mathcal{M}^2 \setminus E_P)$ nên

$$E_{P_2(y)} = \mathcal{M}^2 \setminus hc_2(\mathcal{M}^2 \setminus E_P).$$

§4. DẠNG CHUẨN TẮC HỘI VÀ DẠNG CHUẨN TẮC TUYỂN CỦA CÔNG THỨC

Như chúng ta đã biết, một công thức trong lôgic vị từ đều có dạng công thức rút gọn đồng nhất bằng nó. Nói cách khác, nếu A là một công thức thì bao giờ cũng tồn tại công thức B chỉ chứa các phép toán $\wedge$, $\vee$, —; trong đó phép — chỉ liên quan tới công thức sơ cấp trong B mà $A \equiv B$.

Định nghĩa 8: Trong công thức B nếu mọi ký hiệu $\forall$, $\exists$ đều đứng trước các phép toán lôgic khác thì B được gọi là công thức chuẩn tắc. Nếu B là công thức chuẩn tắc mà $A \equiv B$ thì ta nói B là dạng chuẩn tắc của A.

Ví dụ 1: Công thức $(\forall x_1)(\forall x_2) \ldots (\forall x_i) \ldots (\forall x_n) A(x_1, x_2, \ldots, x_n)$, trong đó $A(x_1, x_2, \ldots, x_n)$ không chứa ký hiệu $\rightarrow$, $\forall$, $\exists$; còn ký hiệu — chỉ liên quan tới công thức sơ cấp thì công thức trên gọi là công thức chuẩn tắc.

Định lý 1: Trong lôgic vị từ, mọi công thức đều có dạng chuẩn tắc.

Chứng minh: Trước tiên ta chứng minh bốn đồng nhất thức sau đây, trong đó A(x) là công thức, H là mệnh đề:

1) $(\forall x) A(x) \vee H \equiv (\forall x)(A(x) \vee H)$

2) $(\forall x) A(x) \wedge H \equiv (\forall x)(A(x) \wedge H)$

3) $(\exists x) A(x) \vee H \equiv (\exists x)(A(x) \vee H)$

4) $(\exists x) A(x) \wedge H \equiv (\exists x)(A(x) \wedge H)$

Thật vậy ta có: $(\forall x) A(x) \vee H$ đúng $\Leftrightarrow \forall x\, A(x)$ đúng hoặc H đúng

$\Leftrightarrow (\forall x)(A(x) \vee H)$ đúng.

Vậy đồng nhất thức (1) là đúng.

2) $(\forall x) A(x) \wedge H$ đúng $\Leftrightarrow (\forall x) A(x)$ đúng và H đúng

$\Leftrightarrow A(x)$ đúng $\forall x$ và H đúng

$\Leftrightarrow A(x) \wedge H$ đúng $\forall x$

$\Leftrightarrow (\forall x)(A(x) \wedge H)$ đúng.

Vậy $(\forall x) A(x) \wedge H \equiv (\forall x)(A(x) \wedge H)$.

Dĩ nhiên bốn đồng nhất thức trên cũng đúng khi các lượng từ đứng trước H và cách chứng minh cũng tương tự.

Bây giờ ta chứng minh định lý trên bằng quy nạp theo định nghĩa công thức. Giả sử A là công thức trong lôgic vị từ.

1) Nếu A là công thức sơ cấp, tức A là biến mệnh đề hay biến vị từ thì trường hợp này định lý đúng, bởi vì chính các biến vị từ hoặc các biến mệnh đề là các công thức chuẩn tắc.

2) Giả sử A_1, A_2 có dạng chuẩn tắc tương ứng là:

$$A_1^* = (\forall x_1) \dots (\forall x_i) (\exists x_{i+1}) \dots (\exists x_n) A'(x_1, \dots, x_n) \equiv A_1.$$

$$A_2^* = (\exists y_1) \dots (\forall y_i) \dots (\forall y_m) A''(y_1, \dots, y_n) \equiv A_2.$$

Ta chứng minh $A_1 \vee A_2$ cũng có dạng chuẩn tắc. Áp dụng (1) và (3) ta có:

$$\begin{aligned} A_1 \vee A_2 = \quad & (\forall x_1) \dots (\forall x_i) (\exists x_{i+1}) \dots (\exists x_n) A'(x_1, \dots, x_n) \\ & \qquad \vee (\exists y_1) \dots (\forall y_i) \dots (\forall y_m) A''(y_1, \dots, y_n) \\ \equiv \quad & (\forall x_1) \dots (\forall x_i) (\exists x_{i+1}) \dots (\exists x_n) \big(A'(x_1, \dots, x_n) \vee \\ & \qquad (\exists y_1) \dots (\forall y_i) \dots (\forall y_m) A''(y_1, \dots, y_n)\big) \\ \equiv \quad & (\forall x_1) \dots (\forall x_i) (\exists x_{i+1}) \dots (\exists x_n) (\exists y_1) \dots (\forall y_m) \\ & \qquad \big(A'(x_1, \dots, x_n) \vee A''(y_1, \dots, y_n)\big). \end{aligned}$$

Rõ ràng $(\forall x_i) \dots (\exists x_n) (\exists y_1) \dots (\forall y_m) (A' \vee A'')$ là công thức chuẩn tắc, hay $A_1 \vee A_2$ có dạng chuẩn tắc.

Áp dụng (2) và (4) ta tìm được dạng chuẩn tắc của $A_1 \wedge A_2$.

Giả sử A_1 có dạng chuẩn tắc A_1^* như trên, ta chỉ ra $\overline{A_1}$ cũng có dạng chuẩn tắc.

Thật vậy, $\overline{A_1} \equiv \overline{A_1^*} = \overline{(\forall x_1) \dots (\forall x_i)(\exists x_{i+1}) \dots (\exists x_n) A'(x_1 \dots x_n)}$

$$= (\exists x_1) \dots (\exists x_n) (\forall x_{i+1}) \dots (\forall x_n) \overline{A'}(x_1, \dots, x_2)$$

Nếu A' chưa phải là công thức sơ cấp thì tiếp tục đưa dấu — chỉ liên quan tới công thức sơ cấp.

Giả sử $\overline{A'}(x_1, \dots, x_2) \equiv B'(x_1, \dots, x_2)$, trong đó B' chứa $\wedge$, $\vee$, —, với — chỉ liên quan tới công thức sơ cấp. Khi đó $(\exists x_1) \dots (\exists x_i) (\forall x_{i+1}) \dots (\forall x_n)$ $B'(x_1, \dots, x_2)$ chính là công thức chuẩn tắc và là dạng chuẩn tắc của $\overline{A_1}$.

Vì $A_1 \to A_2 \equiv \overline{A_1} \vee A_2$ nên từ trên ta thấy $A_1 \to A_2$ cũng có dạng chuẩn tắc.

3) Giả sử A có dạng chuẩn tắc là A^*. Khi đó $(\forall x)$ A và $(\exists x)$ A cũng có dạng chuẩn tắc, cụ thể là $(\forall x) A^*$ và $(\exists x) A^*$ là dạng chuẩn tắc của $(\forall x)$ A và $(\exists x)$ A tương ứng.

Tóm lại định lý được chứng minh.

Định nghĩa 9: Nếu trong dạng chuẩn tắc B của công thức A mà phần công thức đứng sau các lượng từ có DCTH (DCTT) thì B được gọi là DCTH (DCTT) của A.

Định lý 2: Trong lôgic vị từ một biến, mọi công thức đều có DCTH và DCTT.

Chứng minh: Dựa vào định lý 1 ta xây dựng được dạng chuẩn tắc B của A và dùng bảng các công thức đồng nhất bằng nhau trong vị từ cấp 1 dưới đây sẽ đưa B về DCTH, DCTT của A.

4.1. Bảng các công thức đồng nhất bằng nhau trong lôgic vị từ cấp 1

1) $\overline{\overline{A}} \equiv A$

2) $A \vee B \equiv B \vee A$

3) $A \wedge B \equiv B \wedge A$

4) $(A \vee B) \vee C \equiv A \vee (B \vee C) \equiv A \vee B \vee C$

5) $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C) \equiv A \wedge B \wedge C$

6) $\overline{A \vee B} \equiv \overline{A} \wedge \overline{B}$

7) $\overline{A \wedge B} \equiv \overline{A} \vee \overline{B}$

8) $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

9) $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$

10) $A \to B \equiv \overline{A} \vee B$

11) $A \wedge A \equiv A$

12) $A \vee A \equiv A$

13) $A \wedge \mathbf{1} \equiv A$

14) $A \vee \overline{A} \equiv \mathbf{1}$

15) $A \wedge \overline{A} \equiv \mathbf{0}$

16) $A \wedge \mathbf{0} \equiv \mathbf{0}$

17) $A \vee \mathbf{1} \equiv \mathbf{1}$

18) $A \vee (A \wedge B) \equiv A$

19) $A \wedge (A \vee B) \equiv A$

20) $\overline{(\forall x)A} \equiv (\exists x)\overline{A}$

21) $\overline{(\exists x)A} \equiv (\forall x)\overline{A}$

22) $(\forall x)\ A \vee H \equiv (\forall x)\ (A \vee H)$

23) $(\forall x)\ A \wedge H \equiv (\forall x)\ (A \wedge H)$

24) $(\exists x)\ A \vee H \equiv (\exists x)\ (A \vee H)$

25) $(\exists x)\ A \wedge H \equiv (\exists x)\ (A \wedge H)$

26) $(\forall x)\ P(x) \wedge (\forall x)\ Q(x) \equiv (\forall x)\ (P(x) \wedge Q(x))$

27) $(\exists x)\ P(x) \vee (\exists x)\ Q(x) \equiv (\exists x)\ (P(x) \vee Q(x))$

28) $(\forall x)\ P(x) \vee (\forall x)\ Q(x) \equiv (\forall x)\ (\forall y)\ (P(x) \vee Q(y))$

29) $(\forall x)\ P(x) \wedge (\exists x)\ Q(x) \equiv (\forall x)\ (\exists y)\ (P(x) \wedge Q(y))$

30) $(\exists x)\ P(x) \wedge (\forall x)\ Q(x) \equiv (\exists x)\ (\forall y)\ (P(x) \wedge Q(y))$

31) $(\exists x)\ P(x) \wedge (\exists x)\ Q(x) \equiv (\exists x)\ (\exists y)\ (P(x) \wedge Q(y))$

32) $(\exists x)\ P(x) \vee (\forall x)\ Q(x) \equiv (\exists x)\ (\forall y)\ (P(x) \vee Q(y))$

33) $(\forall x)\ P(x) \vee (\exists x)\ Q(x) \equiv (\forall x)\ (\exists y)\ (P(x) \vee Q(y))$

Chú ý:

1) H trong các công thức 22, 23, 24 và 25 là công thức trong lôgic mệnh đề.

2) Các công thức trên dùng để tìm DCTH và DCTT của công thức trong lôgic vị từ theo thuật toán trình bày trong mục 4.2 dưới đây.

4.2. Thuật toán tìm DCTH và DCTT của công thức A trong lôgic vị từ cấp 1

Bài toán:

Input: A công thức bất kỳ trong lôgic vị từ cấp 1.

Output: DCTH, DCTT của A.

Thuật toán:

Bước 1: Khử tất cả các phép kéo theo ($\rightarrow$) trong A bằng cách dùng công thức $X \rightarrow Y \equiv \overline{X} \vee Y$ ta được công thức $A_1 \equiv A$.

Bước 2: Đưa phép toán phủ định trong A_1 về trực tiếp liên quan tới từng biến mệnh đề và biến vị từ có trong A_1 bằng cách áp dụng các công thức

$$\overline{X \vee Y} \equiv \overline{X} \wedge \overline{Y},\ \ \overline{X \wedge Y} \equiv \overline{X} \vee \overline{Y},\ \overline{(\forall x)B} \equiv (\exists x)\overline{B}$$

và $\overline{(\exists x)B} \equiv (\forall x)\overline{B}$ ta được công thức $A_2 \equiv A_1 \equiv A$.

Bước 3. Đưa các ký hiệu lượng từ $\forall$, $\exists$ trong A_2 lên trước mọi phép toán lôgic khác bằng cách áp dụng các công thức:

$$(\forall x)\ A \vee H \equiv (\forall x)\ (A \vee H)$$
$$(\forall x)\ A \wedge H \equiv (\forall x)\ (A \wedge H)$$
$$(\exists x)\ A \vee H \equiv (\exists x)\ (A \vee H)$$
$$(\exists x)\ A \wedge H \equiv (\exists x)\ (A \wedge H)$$
$$(\forall x)\ P(x) \wedge (\forall x)\ Q(x) \equiv (\forall x)\ (P(x) \wedge Q(x))$$
$$(\exists x)\ P(x) \vee (\exists x)\ Q(x) \equiv (\exists x)\ (P(x) \vee Q(x))$$
$$(\forall x)\ P(x) \vee (\forall x)\ Q(x) \equiv (\forall x)\ (\forall y)\ (P(x) \vee Q(y))$$
$$(\exists x)\ P(x) \wedge (\forall x)\ Q(x) \equiv (\exists x)\ (\forall y)\ (P(x) \wedge Q(y))$$
$$(\forall x)\ P(x) \wedge (\exists x)\ Q(x) \equiv (\forall x)\ (\exists y)\ (P(x) \wedge Q(y))$$
$$(\exists x)\ P(x) \vee (\forall x)\ Q(x) \equiv (\exists x)\ (\forall y)\ (P(x) \vee Q(y))$$
$$(\forall x)\ P(x) \vee (\exists x)\ Q(x) \equiv (\forall x)\ (\exists y)\ (P(x) \vee Q(y))$$

ta được công thức $A_3 \equiv A_2 \equiv A_1 \equiv A$.

Trong A_3 phần công thức đứng sau các lượng từ $\forall$, $\exists$ ta ký hiệu là A_0. Khi đó ta viết $A_3 \equiv (\forall, \exists)\ A_0$ (A_0 chính là B của A trong định nghĩa 10).

Bước 4: a) Trong A_0 nếu ta áp dụng công thức

$$X \vee (Y \wedge Z) \equiv (X \vee Y) \wedge (X \vee Z)$$

ta được $A^* \equiv A_0$ mà A^* là DCTH của A_0, hay $A \equiv (\forall, \exists)\ A^*$ là DCTH của A.

b) Trong A_0 nếu ta áp dụng công thức

$$X \wedge (Y \vee Z) \equiv (X \wedge Y) \vee (X \wedge Z)$$

ta được $A^{**} \equiv A_0$ mà A^{**} là DCTT của A_0, hay $A \equiv (\forall, \exists)\ A^{**}$ là DCTT của A.

Ví dụ 2: Cho

$$A \equiv ((\exists x)\ (P(x) \vee (\exists x)\ Q(x)) \rightarrow ((\forall x)\ R(x) \vee ((X \rightarrow Y) \rightarrow (Z \rightarrow (X \rightarrow Y)))).$$

Tìm DCTH và DCTT của A.

Giải:

Theo các bước trong thuật toán ta có:

Bước 1:

$$A \equiv \overline{(\exists x)\, P(x) \vee (\exists x)\, Q(x)} \vee (\forall x)\, R(x) \vee \overline{\overline{X} \vee Y} \vee \overline{Z} \vee \overline{X} \vee Y$$

Bước 2:

$$A \equiv \overline{(\exists x)\, P(x)} \wedge \overline{(\exists x)\, Q(x)} \vee (\forall x)\, R(x) \vee (X \wedge \overline{Y}) \vee \overline{Z} \vee \overline{X} \vee Y$$

$$\equiv (\forall x)\, \overline{P}(x) \wedge (\forall x)\, \overline{Q}(x) \vee (\forall x)\, R(x) \vee (X \wedge \overline{Y}) \vee \overline{Z} \vee \overline{X} \vee Y$$

Bước 3:

$A \equiv (\forall x)(\overline{P}(x) \wedge \overline{Q}(x)) \vee (\forall x)(R(x) \vee (X \wedge \overline{Y}) \vee \overline{Z} \vee \overline{X} \vee Y)$

$\equiv (\forall x)(\forall y)(\overline{P}(x) \wedge \overline{Q}(x) \vee R(y) \vee X \wedge \overline{Y} \vee \overline{Z} \vee \overline{X} \vee Y)$

Bước 4: $A_0 \equiv \overline{P}(x) \wedge \overline{Q}(x) \vee R(y) \vee X \wedge \overline{Y} \vee \overline{Z} \vee \overline{X} \vee Y$

a) $A_0 \equiv A^*$ với $A^* \equiv \overline{P}(x) \wedge \overline{Q}(x) \vee R(y) \vee \overline{Z} \vee \overline{X} \vee Y \vee X) \wedge$

$(\overline{P}(x) \wedge \overline{Q}(x) \vee R(y) \vee \overline{Z} \vee \overline{X} \vee Y \vee \overline{Y}).$

$A \equiv (\forall x)(\forall y)((\overline{P}(x) \wedge \overline{Q}(x) \vee R(y) \vee \overline{Z} \vee \overline{X} \vee Y \vee X) \wedge$

$(\overline{P}(x) \wedge \overline{Q}(x) \vee R(y) \vee \overline{Z} \vee \overline{X} \vee Y \vee \overline{Y})$

$\equiv (\forall x)(\forall y)((\overline{P}(x) \vee R(y) \vee \overline{Z} \vee \overline{X} \vee Y \vee X) \wedge (\overline{Q}(x) \vee R(y) \vee$

$\overline{Z} \vee \overline{X} \vee Y \vee X) \wedge (\overline{P}(x) \vee R(y) \vee \overline{Z} \vee \overline{X} \vee Y \vee \overline{Y})$

$\wedge (\overline{Q}(x) \vee R(y) \vee \overline{Z} \vee \overline{X} \vee Y \vee \overline{Y})$

là DCTH của A.

b) $A \equiv (\forall x)(\forall y)((\overline{P}(x) \wedge \overline{Q}(x) \vee R(y) \vee (X \wedge \overline{Y}) \vee \overline{Z} \vee \overline{X} \vee Y \vee \overline{Y})$ là chuẩn tắc tuyển của A.

Chú ý: $A \equiv \mathbf{1}$ vì $A_0 \equiv \mathbf{1}$ nên $(\forall x)(\forall y)\, A_0 \equiv \mathbf{1}$.

§5. VẤN ĐỀ VỀ TÍNH GIẢI ĐƯỢC

Trong đại số mệnh đề thì bài toán về tính giải được hoàn toàn được giải quyết. Nghĩa là nếu A là công thức trong đại số mệnh đề thì ta có một phương pháp để biết A có phải là đồng nhất đúng hay không.

Ta hãy đặt bài toán này đối với lôgic vị từ, trong đó ta chỉ giới hạn các công thức không chứa các phần tử và các vị từ cụ thể.

Cũng như trong đại số mệnh đề ta đưa vào định nghĩa:

Định nghĩa 10: Công thức A trong lôgic vị từ gọi là thực hiện được trên trường $\mathcal{M}$ nếu nó đúng với những vị từ nào đó trên trường $\mathcal{M}$.

Định nghĩa 11: Công thức A gọi là đồng nhất đúng (gọi tắt là đúng) nếu nó đúng với mọi trường và đối với mọi vị từ.

Định nghĩa 12: Công thức A gọi là sai hay gọi là không thực hiện được nếu nó không đúng đối với mọi trường và đối với mọi vị từ.

Từ định nghĩa ta suy ra rằng: Nếu A là đồng nhất đúng thì $\overline{A}$ là sai và ngược lại.

Bài toán giải được đặt ra cho lôgic vị từ như sau:

– Hãy tìm một phương pháp để biết một công thức A là đồng nhất đúng hay không?

– Nếu có một phương pháp nào đó để được công thức A đồng nhất đúng, và do đó cũng biết được $\overline{A}$ có là sai hay không?

Giả sử $\overline{A}$ sai, khi đó A là đồng nhất đúng. Vậy A thực hiện được, còn $\overline{A}$ là không thực hiện được.

Nếu $\overline{A}$ là không đồng nhất đúng thì A không sai, vậy A thực hiện được.

Người ta đã chứng minh được rằng, bài toán về tính giải được đối với lôgic vị từ là không giải quyết được. Nghĩa là đối với công thức A bất kỳ trong lôgic vị từ không có một thuật toán nào để biết A có đồng nhất đúng hay không.

Tuy nhiên, nếu ta xét một tập hợp con các công thức của lôgic vị từ thì bài toán trên được giải quyết. Chẳng hạn, tập hợp các công thức trong đại số mệnh đề là tập hợp con các công thức của lôgic vị từ thì bài toán về tính giải được lại được giải quyết.

Ngoài ta, nếu xét vị từ một biến thì vấn đề công thức đồng nhất đúng được giải quyết.

Định nghĩa 13: Công thức A trong lôgic vị từ gọi là thoả mãn trên trường $\mathcal{M}$ nếu A đúng với những vị từ cụ thể nào đó trên $\mathcal{M}$.

Định lý 3: Nếu công thức A (trong lôgic vị từ) chứa n vị từ một biến thoả mãn trên trường $\mathcal{M}$ thì nó cũng sẽ thoả mãn trên mọi trường có số phần tử không lớn hơn 2^n.

Chứng minh: Giả sử công thức A thoả mãn trên trường $\mathcal{M}$ và n là số vị từ một biến có trong A. Theo định lý trên ta có thể giả thiết A là công thức chuẩn tắc. Trong A không có biến tử tự do (vì nếu A có một biến tử tự do x nào đấy thì việc xét tính thoả mãn của vị từ A(x) tương đương với việc xét tính thoả mãn của vị từ $(\forall x)\ A(x)$ mà trong $(\forall x)\ A(x)$ không còn biến tử tự do). Với các giả thiết đó ta sẽ giả sử các vị từ trong A là $A_1, \ldots, A_n$ là vị từ một biến, khi đó công thức A viết dưới dạng:

$$A = (\sigma x_1)\ (\sigma x_2) \ldots (\sigma x_p)\ B(A_1, \ldots, A_n, x_1, \ldots, x_p),$$

trong đó: σx_i hoặc là lượng từ $(\forall x_i)$ hoặc là $(\exists x_i)$; B là công thức không chứa lượng từ (do A chuẩn tắc); $x_1, \ldots, x_p$ là các biến ràng buộc.

Thực ra có thể viết:

$$B(A_1(x_{i_1})), ..., B(A_n(x_{i_n})), \text{ trong đó } x_{i_j} \in \{x_1, ..., x_p\}.$$

Tl co giả thiết của định lý, A thoả mãn trên trường $\mathcal{M}$, do đó có các vị từ cụ thể nào đấy mà ta ký hiệu là $A_1^0, ..., A_n^0$ trên $\mathcal{M}$ sao cho $A(A_1^0, ..., A_n^0)$ cũng sẽ đúng trên mọi trường $\mathcal{M}'$ mà $|\mathcal{M}'| \leq 2^n$.

Muốn vậy ta chia $\mathcal{M}$ thành các lớp sau đây:

Ký hiệu $(\alpha_1, ..., \alpha_n)$ là một bộ mà $\alpha_i \in \{0, 1\}$ và ký hiệu

$$M_{x_1,...,x_n} = \{x \in \mathcal{M} : A_i^0(x) = \alpha_i, \forall i = 1, ..., n\}$$

Như vậy, ứng với mỗi bộ $(\alpha_1, ..., \alpha_n)$ khác nhau ta có lớp $M_{x_1,...,x_n}$ khác nhau.

Tính chất: Với $(\alpha_1, ..., \alpha_n) \neq (\alpha'_1, ..., \alpha'_n)$ thì

$$M_{\alpha_1,...,\alpha_n} \cap M_{\alpha'_1,...,\alpha'_n} = \varnothing$$

Thật vậy, giả sử ngược lại chúng có phần tử chung là $x \in \mathcal{M}$ sao cho: $A_i^0(x) = \alpha_i$ và $A_i^0(x) = \alpha'_i$ mà $\alpha_i \neq \alpha'_i$ (trái với tính đơn trị của vị từ A_i^0).

Dĩ nhiên mỗi phần tử thuộc $\mathcal{M}$ thì thuộc vào một lớp nào đó. Với cách chia trên ta được một phân hoạch đối với $\mathcal{M}$. Số các lớp của phân hoạch này không thể vượt quá số bộ $(\alpha_1, ..., \alpha_n)$ khác nhau, nghĩa là số lớp của phân hoạch này $\leq 2^n$. Ta gọi số lớp của phân hoạch là q ($\leq 2^n$), mỗi lớp lấy ra một phần tử ký hiệu lần lượt là $a_1, a_2, ..., a_q$. Ta ký hiệu tập hợp các phần tử này là $\mathcal{M}'$ ($\mathcal{M}' \subseteq \mathcal{M}$).

Ta chứng minh A thoả mãn trên trường $\mathcal{M}'$.

Cụ thể ta chứng minh công thức $A(A_1^0,..., A_n^0)$ đúng trên $\mathcal{M}'$. Thật vậy, với mỗi phần tử $x \in \mathcal{M}$, ký hiệu $\varphi(x)$ là phần tử trong $\mathcal{M}'$, tức là $\varphi(x) = \alpha_i$ nào đó thuộc $\mathcal{M}'$ sao cho x, α_i thuộc cùng một lớp của phân hoạch. Theo định nghĩa, điều này có nghĩa là: $A_i^0(\varphi(x)) \equiv A_i^0(x)$ (i = 1, ..., n), tức là nếu trong công thức A ta thay $x = \varphi(x)$ thì ta được công thức đồng nhất với A.

Giả thiết rằng R(x, y, ..., u) là vị từ bất kỳ xác định trên $\mathcal{M}$. Ký hiệu $(\overline{\forall}(x))R(x, y, ..., u)$ là vị từ chỉ phụ thuộc vào y, ..., u và nhận giá trị đúng khi và chỉ khi R(x, y, ..., u) đúng với mọi $x \in \mathcal{M}'$, (x là a_i nào đấy).

Hoàn toàn tương tự ta định nghĩa vị từ $(\widehat{\exists(x)})R(x, y, ..., u)$ đúng $\Leftrightarrow$ tồn tại $x \in \mathcal{M}'$ để $R(x, y, ..., u)$ đúng. Các ký hiệu $(\widehat{\forall(x)})$, $(\widehat{\exists(x)})$ gọi là các lượng từ hạn chế. Bây giờ ta chứng minh công thức:

$$(\widehat{\forall(x)})R(x, y, ..., u) \equiv (\forall(x))\, R(\varphi(x), y, ..., u).$$

Ta chỉ cần chứng minh vế phải đúng khi vế trái đúng và ngược lại.

– Giả sử vế trái đúng, tức là $R(x, y, ..., u)$ đúng với mọi $x \in \mathcal{M}'$. Xét $\forall(x)\ R(\varphi(x), y, ..., u)$ đúng (lưu ý: với mọi $x \in \mathcal{M}'$ thì $\varphi(x) \in \mathcal{M}'$ mà $R(x, y, ..., u)$ đúng với mọi $x \in \mathcal{M}'$ thì $R(\varphi(x), y, ..., u)$ đúng với mọi x hay $(\forall(x))\ R(\varphi(x), y, ..., u)$ đúng.

– Giả sử vế phải đúng, tức là $(\forall(x))\ R(\varphi(x), y, ..., u)$ đúng với mọi $x \in \mathcal{M}'$, theo định nghĩa thì $(\widehat{\forall(x)})R(x, y, ..., u)$ đúng.

Hoàn toàn tương tự ta chứng minh được công thức:

$$(\widehat{\exists(x)})R(x, y, ..., u) \equiv (\exists(x))\, R(\varphi(x), y, ..., u)$$

Dựa vào hai công thức trên xét công thức:

$$A = (\sigma x_1)\,(\sigma x_2)\, ...\, (\sigma x_p)\, B(A_1^0, ..., A_n^0, x_1, ..., x_p)$$

Ta chứng minh A đúng trên $\mathcal{M}'$.

Thật vậy, Xét $B(A_1^0, ..., A_n^0, x_1, ..., x_p) \equiv B(A_1^0, ..., A_n^0, \varphi(x_1), ..., \varphi(x_p))$

Do nhận xét ở trên, ta có:

$$(\sigma x_p)\, B(A_1^0, ..., A_n^0, x_1, ..., x_p) \equiv (\sigma x_p)\, B(A_1^0, ..., A_n^0, \varphi(x_1), ..., \varphi(x_p))$$
$$\equiv (\widehat{\sigma x_p})B(A_1^0, ..., A_n^0, \varphi(x_1), ..., \varphi(x_p))$$

Tương tự như vậy đối với $\sigma x_1, ..., \sigma x_{p-1}$ ta có:

$$(\sigma x_1)\,(\sigma x_2)\, ...\, (\sigma x_p)\, B(A_1^0, ..., A_n^0, x_1, ..., x_p)$$
$$\equiv (\widehat{\sigma x_1})\, ...\, (\widehat{\sigma x_p})\ B(A_1^0, ..., A_n^0, \varphi(x_1), ..., \varphi(x_p))$$

Công thức vế phải cũng là công thức vế trái nhưng xét trên trường $\mathcal{M}'$. Tóm lại, nếu A đúng trên $\mathcal{M}$ thì nó cũng đúng trên trường $\mathcal{M}'$ với $|\mathcal{M}'| \le 2^n$. Định lý được chứng minh.

Hệ quả 1: Nếu công thức A chỉ chứa các vị từ một biến và luôn luôn đúng với mọi trường có không quá 2^n phần tử (n là số vị từ một biến có mặt trong A) thì A là công thức đồng nhất đúng, nghĩa là A đúng với mọi trường bất kỳ.

Chứng minh: Giả sử ngược lại, A không đồng nhất đúng, nghĩa là có một trường $\mathcal{M}$ nào đó sao cho $\overline{A}$ thoả mãn trên $\mathcal{M}$, theo định lý trên tồn tại trường $\mathcal{M}'$ với $|\mathcal{M}'| \leq 2^n$ sao cho $\overline{A}$ thoả mãn trên $\mathcal{M}'$. Chứng tỏ A không phải là đồng nhất đúng trên $\mathcal{M}'$ (trái với giả thiết). Vậy A đúng trên mọi trường có số phần tử không quá 2^n.

Hệ quả 2: Với mọi công thức A chỉ chứa vị từ một biến đều có thuật toán để giải quyết vấn đề A là đồng nhất đúng hay không.

Chứng minh: Lưu ý rằng, theo định nghĩa thì việc đồng nhất đúng hay không trên trường $\mathcal{M}$ chỉ phụ thuộc vào lực lượng của $\mathcal{M}$. Nói cách khác, nếu $\mathcal{M}$ và $\mathcal{M}'$ có lực lượng như nhau thì A đúng trên $\mathcal{M}$ khi và chỉ khi A đúng trên $\mathcal{M}'$.

Thật vậy, để chứng minh điều đó ta chỉ cần chứng minh rằng: Nếu $|\mathcal{M}| = |\mathcal{M}'|$ thì A sai trên trường $\mathcal{M}$ với vị từ cụ thể nào đó khi và chỉ khi A sai trên trường $\mathcal{M}'$ với vị từ cụ thể, chẳng hạn là $\{A_i^0\}$.

Vì vậy, đối với $A_i^0(x)$ trên $\mathcal{M}$ ta có thể lập được vị từ $B_i^0(x')$ trên trường $\mathcal{M}'$ bằng cách tương ứng $B_i^0(x') \equiv A_i^0(x)$ khi và chỉ khi $x \leftrightarrow x'$. Từ đó suy ra nếu A sai trên $\mathcal{M}$ ứng với $A_i^0(x)$ khi và chỉ khi A sai trên $\mathcal{M}'$ ứng với $B_i^0(x')$.

Tóm lại, tính đồng nhất đúng của công thức chỉ phụ thuộc vào lực lượng của trường mà không phụ thuộc vào các phần tử cụ thể của trường.

Thuật toán:

Giả sử A là công thức chỉ chứa vị từ một biến.

Bước 1: Thử A với tất cả các trường có lực lượng $\leq 2^n$ (n là số vị từ một biến có mặt trong A).

Bước 2: Nếu nó đồng nhất đúng với mọi trường có lực lượng $\leq 2^n$ thì A là đồng nhất đúng. Nếu không thì A không đồng nhất đúng.

Thuật toán này qua một số hữu hạn bước sẽ dừng lại vì số các trường có lực lượng $\leq 2^n$ khác nhau là hữu hạn, còn những trường có lực lượng giống nhau thì không cần xét đến.

§6. NGUYÊN LÝ QUY NẠP

Để chứng minh mệnh đề P(n) đúng với mọi số nguyên dương $n \geq n_0$ ta thực hiện các bước sau:

Bước 1: Chỉ ra $P(n_0)$ đúng.

Bước 2: Chứng minh phép kéo theo $P(n) \rightarrow P(n + 1)$ đúng với mọi $n \geq n_0$, trong đó P(n) là giả thiết quy nạp.

Khi hai bước trên hoàn thành ta kết luận mệnh đề P(n) đúng với mọi $n \geq n_0$.

Cơ sở của nguyên lý quy nạp trên là công thức hằng đúng sau đây:

$$(P(n_0) \wedge ((\forall n > n_0)\ (P(n) \rightarrow P(n+1)))) \rightarrow ((\forall n \geq n_0)\ P(n))$$

Ví dụ 1: Xét vị từ P(n) :

$$0 + 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$$

Giải:

P(0) đúng, vì $0 = \dfrac{0 \times 1}{2}$.

Để chứng minh $(\forall n \geq 0)\ (P(n) \rightarrow P(n + 1)$ đúng ta dùng phép tổng quát hoá phổ dụng, thay n = k và chứng minh $P(k) \rightarrow P(k + 1)$ đúng.

Giả sử P(k) đúng, tức là:

$$0 + 1 + 2 + \ldots + k = \frac{k(k+1)}{2}.$$

Khi đó

$$0 + 1 + 2 + \ldots + k + k + 1 = \frac{k(k+1)}{2} + k + 1 = \frac{(k+1)(k+2)}{2},$$

nghĩa là P(k + 1) cũng đúng.

Từ hai bước trên ta suy ra P(n) đúng với mọi $n \geq 0$.

Ví dụ 2: Chứng minh rằng, số các tập con của tập A_n phần tử là 2^n.

Giải:

Đặt P(n): Tập có n phần tử có 2^n tập con;

P(0): Tập 0 phần tử có $2^0 = 1$ tập con (đúng).

Giả sử P(n) đúng, ta chỉ ra P(n + 1) cũng đúng. Giả sử tập n + 1 phần tử là B, ở đây $B = A \cup \{a\}$ ($a \notin A$). Lấy tập con A' bất kỳ của A, ta được hai tập con của B là A' và $A' \cup \{a\}$. Như vậy số tập con của B là $2.2^n = 2^{n+1}$. Vậy P(n + 1) đúng.

Kết luận P(n) đúng với mọi $n \geq 0$.

Ví dụ 3: Chứng minh rằng:

a) $\overline{\bigcup_{i=1}^{n} A_i} = \bigcap_{i=1}^{n} \overline{A_i}$

b) $\overline{\bigcap_{i=1}^{n} A_i} = \bigcup_{i=1}^{n} \overline{A_i}$

Ở đây A_i ($i = \overline{1,n}$) là các tập con của tập X.

(Bạn đọc tự chứng minh xem như một bài tập).

Ví dụ 4: Xét đoạn chương trình viết bằng ngôn ngữ Pascal:

```
While n <> 0 do
        Begin
                x := x + y;
                n := n – 1
        End;
Kết quả := x;
```

Chứng minh rằng, khi gọi đoạn chương trình trên với các biến x, y lấy giá trị thực và biến n lấy giá trị tự nhiên thì khi ra khỏi đoạn chương trình, biến kết quả được gán giá trị x + ny.

Giải:

Gọi P(n) là vị từ "Với bất kỳ số thực x, y, nếu khi con trỏ điều khiển chương trình trở về dòng lệch While với các giá trị của các biến x, y, n thì khi ra khỏi chương trình biến x có giá trị x + ny".

P(0): "Khi trở về dòng lặp While với các giá trị biến x, y, 0 thì vòng lặp không được thực hiện tiếp và ra khỏi chương trình với giá trị biến x là x = x + 0.y.

Giả sử P(k) đúng, ta chỉ ra P(k + 1) cũng đúng. Gọi x, y là các số thực tuỳ ý và giả sử ta trở về dòng lệch While với giá trị của các biến x, y, n = k + 1. Do k ≥ 0 nên k + 1 > 0. Vậy vòng lặp While được thực hiện thêm một lần nữa. Do P(k) đúng nên khi ra khỏi đoạn chương trình, biến x có giá trị là x := (x + y) + ky = x + (k + 1)y, tức là P(k + 1) đúng. Hay (∀n ≥ 0) P(n) đúng.

Ví dụ 5: Chứng minh với mọi n (n ∈ ℕ) ta có:

$$1^2 + 2^2 + ... + n^2 = \frac{n.(n+1).(2n+1)}{6} \qquad (1)$$

Giải:

– Với n = 1 thì hai vế bằng nhau và bằng 1.

– Giả sử đẳng thức (1) đúng với mọi n ≤ k, có nghĩa là với mọi n ≤ k ta có:

$$1^2 + 2^2 + ... + n^2 = \frac{n.(n+1).(2n+1)}{6}.$$

– Ta phải chứng minh (1) đúng với n = k + 1, tức là phải chứng minh

$$1^2 + 2^2 + ... + k^2 + (k + 1)^2 = \frac{(k+1).(k+2).(2k+3)}{6}$$

Thật vậy, với n = k theo giả thiết quy nạp ta có:

$$1^2 + 2^2 + ... + k^2 = \frac{k.(k+1).(2k+1)}{6}$$

Do đó:

$$1^2 + 2^2 + ... + k^2 + (k+1)^2 = \frac{k.(k+1).(2k+1)}{6} + (k+1)^2$$

$$= \frac{(k+1).(k+2).(2k+3)}{6}$$

Vậy đẳng thức đúng với n = k + 1, do đó (1) được chứng minh.

Ví dụ 6: Chứng minh đẳng thức sau đúng với mọi n ≥ 1.

$$1.1! + 2.2! + ... + n.n! = (n + 1)! - 1 \qquad (1)$$

Giải:

– Với n = 1, khi đó hai vế của (1) bằng nhau nên (1) đúng.

– Giả sử (1) đúng với n = k, nghĩa là với mọi n ≤ k đẳng thức sau luôn đúng:

$$1.1! + 2.2! + ... + n.n! = (n + 1)! - 1.$$

– Ta phải chứng minh đẳng thức đúng với $n = k + 1$, tức là cần chứng minh đẳng thức:

$$1.1! + 2.2! + ... + k.k! + (k + 1).(k + 1)! = (k + 2)! - 1.$$

Thật vậy, với $n = k$ theo giả thiết quy nạp ta có

$$1.1! + 2.2! + ... + k.k! = (k + 1)! - 1.$$

Do đó

$$1.1! + 2.2! + ... + k.k! + (k + 1).(k+1)! = (k+1)! - 1 + (k + 1).(k+1)!$$
$$= (k + 2)! - 1.$$

Vậy đẳng thức (1) đúng với $n = k + 1$ và do đó (1) được chứng minh.

Ví dụ 7: Chứng minh đẳng thức sau đúng với mọi $n > 1$, $a > -1$ và $a \neq 0$:

$$(1 + a)^n > 1 + na \qquad (1)$$

Giải:

– Với $n = 2$, khi đó vế trái của (1) lớn hơn vế phải của (1). Vậy (1) đúng.

– Giả sử (1) đúng với mọi $n \leq k$ tức là với $k > 1$ và $n \leq k$ thì ta có:

$$(1 + a)^n > 1 + na \qquad (2)$$

– Tiếp theo cần chứng minh bất đẳng thức đúng với $n = k + 1$, tức là cần phải chứng minh

$$(1 + a)^{k+1} > 1 + (k + 1)a \qquad (3)$$

Thật vậy, vì $(1 + a) > 0$ và từ (2) ta có:

$$(1 + a)^{k+1} = (1 + a)^k.(1 + a)$$

Vậy $$(1 + a)^{k+1} > (1 + ka)(1 + a) \qquad (4)$$

Ta biến đổi

$$(1 + ka)(1 + a) = 1 + ka + a + k.a^2 > 1 + (1 + k)a \text{ (vì } ka^2 > 0) \qquad (5)$$

Từ (4) và (5) suy ra (3) đúng và do đó bất đẳng thức đã được chứng minh.

Ví dụ 8: Chứng minh rằng tích của ba số tự nhiên liên tiếp luôn chia hết cho 6.

Giải:

Thật vậy, tích của ba số nguyên liên tiếp sẽ là một số có dạng $n(n + 1)(n + 2)$. Vậy ta cần chỉ ra $n(n + 1)(n + 2) \vdots 6$.

– Với $n = 1$, khi đó $n(n + 1)(n + 2) = 6 \vdots 6$.

– Giả sử với mọi $n \le k$, ta đều có

$$n(n+1)(n+2) \vdots 6 \qquad (1)$$

– Tiếp theo cần chứng minh tính chất đó đúng với $n = k + 1$, tức là cần phải chứng minh:

$$(k+1)(k+2)(k+3) \vdots 6 \qquad (2)$$

Do giả thiết quy nạp (1) và $n = k$, ta có:

$$k(k+1)(k+2) \vdots 6 \qquad (3)$$

Thấy rằng $(k + 1)(k + 2)(k + 3) = k(k + 1)(k + 2) + 3(k + 1)(k + 2)$. Từ (3) ta thấy rằng khẳng định (2) sẽ là đúng đắn nếu

$$3(k+1)(k+2) \vdots 6 \qquad (4)$$

Do $(k + 1)$ và $(k + 2)$ là hai số tự nhiên liên tiếp nên tích của hai số đó sẽ phải là số chẵn nên $(k + 1)(k + 2) \vdots 2$ suy ra $3(k + 1)(k + 2) \vdots 6$. Vậy (4) đúng.

Qua các ví dụ trên ta thấy rằng, bằng phương pháp chứng minh quy nạp toán học ta có thể chứng minh một số mệnh đề có liên quan đến dãy số tự nhiên nếu dự đoán được quy luật của nó phụ thuộc vào các số cụ thể như thế nào. Tuy nhiên, trong một số bài toán ta không biết được điều cần chứng minh là gì. Ví dụ, hãy tính tổng của n số tự nhiên lẻ đầu tiên. Rõ ràng là đối với một số bài toán không khó lắm trong việc tìm ra quy luật, chẳng hạn như bài toán vừa phát biểu ở trên ta vẫn có thể ứng dụng được phương pháp chứng minh quy nạp toán học. Tất nhiên là trước hết bằng sự quan sát có tính thống kê ta có thể dự đoán được quy luật cho bài toán, sau đó hãy dùng phương pháp quy nạp toán học để chứng minh. Chúng ta hãy xét một số ví dụ:

Ví dụ 9: Tính tổng của n số tự nhiên lẻ đầu tiên

$$T = 1 + 3 + 5 + ... + (2n - 1).$$

Giải:

Nhận thấy rằng:

Khi $n = 1$, $T = 1$

Khi $n = 2$, $T = 1 + 3 = 4 = 2^2$

Khi $n = 3$, $T = 1 + 3 + 5 = 3^2$

Khi $n = 4$, $T = 1 + 3 + 5 + 7 = 4^2$

...

Quan sát sự thống kê đó ta có thể dự đoán rằng với n là một số tự nhiên bất kỳ thì

$$T = 1 + 3 + ... + (2n - 1) = n^2 \quad (1)$$

Tiếp theo, bằng chứng minh quy nạp ta có thể dễ dàng chứng minh (1) là đúng đắn.

Ví dụ 10: Tính tổng $T = \frac{1}{1.2} + \frac{1}{2.3} + ... + \frac{1}{n(n+1)}$

Giải:

Khi $n = 1$, $T = \frac{1}{1.2} = \frac{1}{2}$.

Khi $n = 2$, $T = \frac{1}{1.2} + \frac{1}{2.3} = \frac{2}{3}$

...

Vậy liệu chăng tổng T sẽ thoả mãn đẳng thức

$$T = \frac{1}{1.2} + \frac{1}{2.3} + ... + \frac{1}{n(n+1)} = \frac{n}{n+1}$$

Bằng phương pháp quy nạp toán học ta thấy rằng đẳng thức trên là đúng đắn. Tuy nhiên phương pháp quy nạp toán học không phải là phương pháp duy nhất để giải quyết những bài toán như trên. Chẳng hạn để tính tổng

$$T = \frac{1}{1.2} + \frac{1}{2.3} + ... + \frac{1}{n(n+1)}$$

Ngoài phương pháp đã nêu trên ta có thể tính T bằng phân tích như sau:

Nhận thấy rằng, với mọi số tự nhiên $k \geq 1$ ta có

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

Do đó với: $k = 1$, $\frac{1}{1.2} = \frac{1}{1} - \frac{1}{2}$

$k = 2$, $\frac{1}{2.3} = \frac{1}{2} - \frac{1}{3}$

...

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

Cộng theo từng vế ta có ngay

$$T = \frac{1}{1} - \frac{1}{n+1} = \frac{n}{n+1}$$

§7. QUY TẮC SUY DIỄN TRONG LÔGIC VỊ TỪ CẤP 1

7.1. Các lượng từ và các mệnh đề có lượng từ

Cho P(x) là một vị từ một biến trên trường $\mathcal{M}$ nào đó, chẳng hạn $\mathcal{M} = \{1, 2, ..., n\}$, khi đó công thức $(\forall x)$ P(x) và $(\exists x)$ P(x) là các mệnh đề (có lượng từ) hoặc đúng, hoặc sai trên trường $\mathcal{M}$ và theo định nghĩa thì

$$(\forall x)\ P(x) \equiv P(1) \wedge P(2) \wedge ... \wedge P(n) \text{ và } (\exists x)\ P(x) \equiv P(1) \vee P(x) \vee ... \vee P(n).$$

Tuy nhiên, nếu ta xét vị từ hai biến P(x, y) trên trường $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$ ($x \in \mathcal{M}_1$ và $y \in \mathcal{M}_2$) thì công thức: $(\forall x)$ P(x, y) và $(\exists x)$ P(x, y) không phải là các mệnh đề có lượng từ nữa mà là các vị từ theo biến y. x là biến bị ràng buộc bởi các lượng từ, còn y là biến tự do. Tuy nhiên, công thức $(\forall x)(\exists y)$ P(x, y) là mệnh đề có giá trị hoàn toàn được xác định mà không phụ thuộc vào x, y.

7.2. Một số quy tắc suy diễn trong lôgic vị từ

- ***Quy tắc thay đổi thứ tự lượng từ hoá hai biến***

$$(\forall x)(\forall y)\ P(x, y) \equiv (\forall y)(\forall x)\ P(x, y)$$

$$(\exists x)(\exists y)\ P(x, y) \equiv (\exists y)(\exists x)\ P(x, y)$$

- ***Quy tắc đặc biệt hoá phổ dụng***

Giả sử một mệnh đề có lượng từ, trong đó biến x với miền xác định $\mathcal{M}$ bị ràng buộc bởi lượng từ phổ dụng ($\forall$) và mệnh đề là đúng. Khi đó nếu thay x bởi $a \in \mathcal{M}$ thì ta được mệnh đề đúng.

- ***Quy tắc tổng quát hoá phổ dụng***

Cho mệnh đề lượng từ hoá $(\forall x)$ P(x) trên trường $\mathcal{M}$. Nếu ta lấy x = a là phần tử bất kỳ trong $\mathcal{M}$ mà P(a) đúng thì mệnh đề lượng từ hoá $(\forall x)$ P(x) là mệnh đề đúng.

Bổ đề 1: Cho P(x), Q(x) là các vị từ theo biến x trên trường $\mathcal{M}$ nào đó, a là một phần tử cố định tuỳ ý thuộc $\mathcal{M}$. Khi đó ta có mô hình suy diễn:

$$\begin{array}{l} (\forall x)(P(x) \rightarrow Q(x)) \\ \underline{P(a) \qquad\qquad\qquad} \\ \qquad \therefore\ Q(a) \end{array}$$

Bổ đề 2: Cho P(x), Q(x) và R(x) là các vị từ theo biến x trên trường $\mathcal{M}$. Ta có mô hình suy diễn:

$$\begin{array}{l} (\forall x)\,(P(x) \to Q(x)) \\ \underline{(\forall x)\,(Q(x) \to R(x))} \\ \therefore\ (\forall x)\,(P(x) \to R(x)) \end{array}$$

Bổ đề 3: $\dfrac{(\forall x)\,P(x)}{\therefore\ P(a)} \equiv \dfrac{(\forall x)P(x)}{\therefore\ (\exists x)\,P(x)}$ ($a \in \mathcal{M}$ là phần tử cho trước bất kỳ).

Bổ đề 4: $P(x)$ là vị từ xác định trên $X = X_1 \cup X_2 \cup \ldots \cup X_n$

Khi đó:

$$\frac{(\forall x \in X_1)P(x),\ (\forall x \in X_2)P(x),\ \ldots,\ (\forall x \in X_n)P(x)}{\therefore\ (\forall x \in X)P(x)}$$

7.3. Một số ví dụ áp dụng

Ví dụ 1: "Mọi sinh viên CNTT đều học môn Toán rời rạc. Minh là sinh viên CNTT. Vậy Minh học môn Toán rời rạc."

Chỉ ra suy luận trên là đúng.

Giải: Đặt $P(x)$ = "x là sinh viên Tin học";

$Q(x)$ = "x học môn Toán rời rạc"

là hai vị từ một biến trên trường sinh viên $\mathcal{M}$.

Đoạn văn trên tương đương với công thức trong lôgic vị từ cấp 1: $((\forall x)(P(x) \to Q(x) \wedge P(x)) \to Q(a)$ trên trường sinh viên $\mathcal{M}$, ở đây a chỉ sinh viên Minh trong trường $\mathcal{M}$.

Mô hình suy diễn của công thức trên có dạng:

$$\begin{array}{l} (\forall x)(P(x) \to Q(x)) \\ \underline{P(a)\qquad\qquad\qquad} \\ \qquad\therefore\ Q(a) \end{array} \qquad (*)$$

Theo bổ đề 1 thì mô hình (*) là đúng, hay suy luận trên là đúng.

Chú ý: (*) tương đương với $\dfrac{\begin{array}{c}P(a) \to Q(a)\\ P(a)\end{array}}{\therefore\ Q(a)}$ trong lôgic mệnh đề và mô hình suy diễn này đúng theo quy tắc suy diễn khẳng định.

Ví dụ 2: Áp dụng mô hình suy diễn trong bổ đề 2 để chỉ ra cách giải phương trình $4x - 5 = 15$ dưới đây là đúng:

Giải:

Giả sử $4x - 5 = 15$, khi đó $4x = 20$. Giả sử $4x = 20$, khi đó $x = 5$. Như vậy $4x - 5 = 15$ thì $x = 5$.

Đặt $P(x) = "4x - 5 = 15"$

$Q(x) = "4x - 20"$

$R(x) = "x = 5"$

Lý luận trên có thể diễn đạt dưới dạng mô hình sau:

$$\frac{\begin{array}{l}(\forall x)\,(P(x) \to Q(x))\\(\forall x)\,(Q(x) \to R(x))\end{array}}{\therefore\ (\forall x)\,(P(x) \to R(x))} \text{ trên trường } \mathscr{M} \text{ các số thực.}$$

Theo bổ đề 2 ta có cách giải phương trình $4x - 5 = 15$ là đúng.

Ví dụ 3: Mô hình suy diễn dưới đây trên trường $\mathscr{M}$ có đúng không?

$$\frac{\begin{array}{l}(\forall x)\,(P(x) \to (Q(x) \wedge R(x)))\\(\forall x)\,(P(x) \wedge F(x))\end{array}}{\therefore\ (\forall x)\,(R(x) \wedge F(x))} \qquad (1)$$

Giải: Lấy a là phần tử cố định bất kỳ trong $\mathscr{M}$, thay $x = a$ trong (1) ta được mô hình suy diễn trong lôgic mệnh đề là

$$\frac{\begin{array}{l}P(a) \to (Q(a) \wedge R(a))\\P(a) \wedge F(a)\end{array}}{\therefore\ R(a) \wedge F(a)} \equiv \frac{\begin{array}{l}\begin{cases}P(a) \to (Q(a) \wedge R(a))\\P(a)\end{cases}\\F(a)\end{array}}{\therefore\ R(a) \wedge F(a)} (\text{Kđ})$$

$$\equiv \frac{\begin{array}{l}Q(a) \wedge R(a)\\F(a)\end{array}}{\therefore\ R(a) \wedge F(a)} \equiv \frac{Q(a) \wedge R(a) \wedge F(a)}{\therefore\ R(a) \wedge F(a)} (\text{Rg}) \equiv \mathbf{1}.$$

Vậy (1) là đúng.

Ví dụ 4: Chỉ ra mô hình suy diễn dưới đây là đúng trên trường $\mathscr{M}$.

$$\frac{\begin{array}{l}(\forall x)\,(P(x) \vee Q(x))\\(\exists x)\,\overline{P}(x)\\(\forall x)\,(\overline{Q}(x) \vee R(x))\\(\forall x)\,(F(x) \to \overline{R}(x))\end{array}}{\therefore\ (\exists x)\,\overline{F}(x)} \qquad (1)$$

Giải: Áp dụng quy tắc tổng quát hoá phổ dụng. Lấy $a \in \mathcal{M}$ bất kỳ sao cho $\overline{P}(a)$ đúng, thay x = a vào (1) ta được mô hình suy diễn sau:

$$\begin{array}{l} \begin{cases} P(a) \vee Q(a) \\ \overline{P}(a) \end{cases} \\ \overline{Q}(a) \vee R(a) \\ F(a) \to \overline{R}(a) \\ \hline \therefore \overline{F}(a) \end{array} \equiv \begin{array}{l} \begin{cases} Q(a) \\ \overline{Q}(a) \vee R(a) \end{cases} \\ F(a) \to \overline{R}(a) \\ \hline \therefore \overline{F}(a) \end{array} \equiv \begin{array}{l} \begin{cases} R(a) \\ F(a) \to \overline{R}(a) \end{cases} \\ \hline \therefore \overline{F}(a) \end{array} \equiv \begin{array}{c} \overline{F}(a) \\ \hline \therefore \overline{F}(a) \end{array} \equiv \mathbf{1}.$$

Vậy (1) đúng.

Ví dụ 5: Chỉ ra mô hình suy diễn dưới đây là đúng trên trường $\mathcal{M}$.

$$\begin{array}{l} (\forall x)\,(P(x) \to Q(x)) \\ (\forall x)\,(R(x) \to F(x)) \\ (\forall x)\,((F(x) \wedge Q(x)) \to (P(x) \wedge H(x))) \\ (\forall x)\,(H(x) \to \overline{P}(x)) \\ \hline \qquad \therefore (\forall x)\,(\overline{R}(x) \vee \overline{P}(x)) \end{array} \qquad (1)$$

Giải: Với mọi $a \in \mathcal{M}$, thay x = a trong (1) ta được mô hình suy diễn có dạng:

$$\begin{array}{l} P(a) \to Q(a) \\ R(a) \to F(a) \\ (F(a) \wedge Q(a)) \to (P(a) \wedge H(a)) \\ H(a) \to \overline{P}(a) \\ \hline \qquad \therefore \overline{R}(a) \vee \overline{P}(a) \end{array} \text{(Mt)} \qquad (2)$$

$$\equiv \begin{array}{l} \begin{cases} P(a) \to Q(a) \\ P(a) \end{cases} \\ \begin{cases} R(a) \to F(a) \\ R(a) \end{cases} \\ (F(a) \wedge Q(a)) \to (P(a) \wedge H(a)) \\ H(a) \to \overline{P}(a) \\ \hline \qquad \therefore \mathbf{0} \end{array}$$

$$\equiv \frac{\begin{cases} F(a) \wedge Q(a) \\ (F(a) \wedge Q(a)) \to (P(a) \wedge H(a)) \end{cases} \\ H(a) \to \overline{P}(a)}{\therefore 0} \equiv \frac{P(a) \\ \begin{cases} H(a) \\ H(a) \to \overline{P}(a) \end{cases}}{\therefore 0}$$

$$\equiv \frac{P(a) \\ \overline{P}(a)}{\therefore 0} \equiv \frac{0}{\therefore 0} \equiv 1$$

Vậy mô hình (1) đúng theo quy tắc tổng quát hoá phổ dụng.

Các ví dụ dưới đây dành cho bạn đọc:

Ví dụ 6: Mô hình suy diễn dưới đây có đúng không trên trường $\mathcal{M}$?

$$\frac{\begin{array}{l} (\forall x)\ (\overline{P}(x) \vee Q(x)) \\ (\forall x)\ (R(x) \to F(x)) \\ (\forall x)\ (\overline{Q(x) \vee F(x)} \vee H(x)) \\ (\forall x)\ \overline{H}(x) \end{array}}{\therefore (\forall x)\ (\overline{P}(x) \wedge \overline{F}(x))}$$

Ví dụ 7: Cho công thức trong lôgic vị từ cấp 1:

$$A \equiv (\forall x)((P_1(x) \to P_2(x)) \to P_3(x)) \wedge (P_4(x) \vee \overline{P}_3(x))$$
$$\wedge (P_1(x) \vee P_5(x) \wedge \overline{P}_5(x)) \to (\forall x)\overline{P}_1(x)$$

Dùng hai phương pháp: Tìm DCTH và mô hình suy diễn để kiểm tra xem công thức A là hằng đúng hay không?

Ví dụ 8: Cùng yêu cầu như ví dụ 7 với công thức:

$$A \equiv (\forall x)(P_1(x) \wedge (P_1(x) \to P_2(x)) \wedge (P_3(x) \vee P_4(x)) \wedge \overline{(P_4(x) \wedge P_2(x)))}$$
$$\to (\forall x)(P_3(x) \vee P_5(x))$$

Ví dụ 9: Chỉ ra mô hình suy diễn dưới đây là đúng trên trường $\mathcal{M}$.

$$\frac{\begin{array}{l} (\forall x)\ ((\overline{P}_1(x) \vee P_2(x)) \to P_3(x)) \\ (\forall x)\ (\overline{P}_3(x) \vee P_4(x) \vee P_5(x)) \\ (\forall x)\ ((\overline{P}_4(x) \wedge \overline{P}_6(x) \wedge (\overline{P}_6(x) \to \overline{P}_5(x))) \\ (\forall x)\ \overline{H}(x) \end{array}}{\therefore\ (\forall x) P_1(x)}$$

BÀI TẬP

Trong chương này chúng tôi chỉ đưa ra các bài tập về lôgic mệnh đề và lôgic vị từ.

1. Tìm phủ định của mệnh đề: "Hôm nay là thứ 6".

2. Tìm phủ định của mệnh đề: "Không có ô nhiễm ở Hà Nội".

3. Tìm phủ định của mệnh đề: "2 + 1 = 3".

4. Tìm phủ định của mệnh đề: "Mùa hè ở Hà Nội nóng và nắng".

5. Tìm hội của mệnh đề X và Y, ở đây X là mệnh đề: "Hôm nay là thứ 6"; Y là mệnh đề: "Hôm nay trời mưa".

6. Tìm hội của các mệnh đề: "Tôi đã mua vé sổ số tuần này" và "Tôi đã trúng giải độc đắc 500 triệu đồng vào hôm thứ 6".

7. Lập tuyển của hai mệnh đề: "Hôm nay trời mưa" và "Hôm nay là thứ 6".

8. Lập tuyển của hai mệnh đề cho trong bài 6.

9. Xác định giá trị chân lý của phép kéo theo: "Nếu hôm nay trời nắng, chúng tôi sẽ đi tắm biển".

10. Xác định giá trị chân lý của mệnh đề phép kéo theo: "Nếu hôm nay là chủ nhật thì 2 + 3 = 5".

11. Trong ngôn ngữ lập trình thường chứa câu lệnh: *Nếu p thì S (if p then S)*, trong đó p là một mệnh đề, S là một đoạn chương trình (gồm một hoặc nhiều lệnh phải thực hiện). Trong cấu trúc của câu lệnh trên thì S sẽ được thực hiện nếu p đúng và S sẽ không thực hiện nếu p sai.

Ví dụ: Xác định giá trị của biến x sau câu lệnh if 2 + 2 = 4 then x := x + 1. Nếu trước câu lệnh đó x = 0. Vì 2 + 2 = 4 là đúng nên ta thực hiện phép gán x cho x + 1. Vì thế x có giá trị là 0 + 1 = 1.

Xác định giá trị của x sau câu lệnh: if 2 + 3 = 6 then x := x + 1 với giả thiết trước câu lệnh đó x = 1.

12. Cho mệnh đề $X \to Y$; mệnh đề đảo của mệnh đề này là $Y \to X$; mệnh đề phản đảo của $X \to Y$ là $\overline{Y} \to \overline{X}$. Tìm các mệnh đề đảo và phản đảo của các mệnh đề kéo theo sau:

a) "Nếu hôm nay là thứ 5 thì tôi có cuộc đi tham quan Đền Hùng".

b) "Nếu hôm nay trời mưa, thì tôi sẽ nghỉ ở nhà".

13. Cho hai mệnh đề X và Y. Mệnh đề tuyển loại của X và Y ký hiệu là $X \oplus Y$ là một mệnh đề chỉ đúng khi một trong hai mệnh đề X, Y đúng và sai trong các trường hợp còn lại. Bảng giá trị chân lý của $X \oplus Y$ là

X	Y	$X \oplus Y$
1	1	0
1	0	1
0	1	1
0	0	0

a) Lập mệnh đề tuyển loại của hai mệnh đề: "2 + 3 = 5" và "4 + 3 = 7".

b) Lập mệnh đề tuyển loại của hai mệnh đề: "2 + 2 = 5" và "4 + 3 = 8".

c) Lập mệnh đề tuyển loại của hai mệnh đề: "2 > 2" và "2 > 3".

14. Dịch câu sau đây ra biểu thức lôgic bằng cách dùng các biến mệnh đề X, Y, Z để diễn đạt các bộ phận của câu nói rồi liên kết chúng với nhau bởi các phép toán lôgic: "Bạn không được lái xe máy nếu bạn cao dưới một mét, trừ khi bạn trên 18 tuổi".

15. Cho X và Y là hai mệnh đề:

X: "Bạn lái xe với tốc độ trên 65 km/h";

Y: "Bạn bị phạt vì vượt quá tốc độ cho phép".

Hãy viết các mệnh đề sau bằng cách dùng X, Y và các phép toán lôgic:

a) "Bạn không lái xe với tốc độ trên 65 km/h".

b) "Bạn lái xe với tốc độ trên 65 km/h, nhưng bạn không bị phạt vì quá tốc độ cho phép".

c) "Bạn sẽ bị phạt vì quá tốc độ cho phép nếu bạn lái xe với tốc độ trên 65 km/h".

d) "Nếu bạn không lái xe với tốc độ trên 65 km/h thì bạn sẽ không bị phạt vì vượt quá tốc độ cho phép".

e) "Lái xe với tốc độ trên 65 km/h là đủ để bị phạt vì vượt quá tốc độ cho phép".

f) "Bạn bị phạt vì vượt quá tốc độ cho phép, nhưng bạn không lai xe với tốc độ trên 65 km/h".

g) "Mỗi lần bị phạt vì vượt quá tốc độ cho phép là bạn đã lái xe với tốc độ trên 65 km/h".

16. Ta dùng các ký hiệu OR, AND và XOR thay cho các phép toán lôgic $\vee$, $\wedge$ và $\oplus$ như thường được dùng trong ngôn ngữ lập trình. Một xâu bit là một dãy các số 0 và 1. Còn thông tin thì được biểu diễn bằng các xâu bit.

Ta có thể thực hiện các phép toán OR bit, AND bit và XOR bit đối với các xâu bit có cùng độ dài, chẳng hạn

$$\begin{array}{ccl} 01101 & 00101 & \\ 11010 & 01101 & \\ \hline 11111 & 01101 & \text{(OR bit)} \end{array}$$

$$\begin{array}{ccl} 01101 & 00101 & \\ 11010 & 01101 & \\ \hline 01000 & 00101 & \text{(AND bit)} \end{array}$$

$$\begin{array}{ccl} 01101 & 00101 & \\ 11010 & 01101 & \\ \hline 10111 & 01000 & \text{(XOR bit)} \end{array}$$

Tìm các OR bit, AND bit và XOR bit của các cặp xâu bit sau:

a) 10 11110; 01 00001;

b) 111 00010; 001 11101;

c) 11010 00011; 00100 11111.

17. Xác định các biểu thức sau:

a) $11000 \wedge (01011 \vee 11011)$;

b) $(01111 \wedge 10101) \vee 01000$;

c) $(01101 \oplus 01010) \oplus 01101$.

18. Xác định giá trị của x sau mỗi khi gặp câu lệnh dưới đây trong một chương trình máy tính, nếu trước khi tới câu lệnh đó x = 1:

a) if 1 + 2 = 3 then x := x + 1;

b) if (1 + 1 = 3) OR (2 + 2 = 3) then x := x + 1;

c) if (2 + 3 = 5) AND (3 + 4 = 7) then x := x + 1;

d) if (1 + 1 = 2) XOR (1 + 2 = 3) then x := x + 1;

e) if x < 2 then x := x + 1.

19. Lập bảng giá trị chân lý của các công thức sau:

a) $(X \oplus Y) \vee (X \oplus \overline{Y})$;

b) $X \rightarrow ((X \wedge Y) \vee Z)$;

c) $(X \rightarrow Y) \wedge (\overline{X} \rightarrow Z)$;

d) $X \vee (Y \oplus Z) \wedge (\overline{Y} \oplus \overline{Z})$.

20. Chứng minh các công thức sau đây là tương đương lôgic (đồng nhất bằng nhau) bằng cách quy nạp:

a) $X \vee (Y_1 \wedge Y_2 \wedge ... \wedge Y_n)$ và $(X \vee Y_1) \wedge (X \vee Y_2) \wedge ... \wedge (X \vee Y_n)$;

b) $X \wedge (Y_1 \vee Y_2 \vee ... \vee Y_n)$ và $(X \wedge Y_1) \vee (X \wedge Y_2) \vee ... \vee (X \wedge Y_n)$;

c) $\overline{X_1 \vee X_2 \vee ... \vee X_n}$ và $\overline{X}_1 \wedge \overline{X}_2 \wedge ... \wedge \overline{X}_n$.

d) $\overline{X_1 \wedge X_2 \wedge ... \wedge X_n}$ và $\overline{X}_1 \vee \overline{X}_2 \vee ... \vee \overline{X}_n$.

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

a) $(X \rightarrow (Y \rightarrow Z)) \rightarrow ((X \rightarrow Y) \rightarrow (X \rightarrow Z))$;

b) $(X \rightarrow Y) \rightarrow ((X \rightarrow Z) \rightarrow (X \rightarrow (Y \wedge Z)))$;

c) $(X \rightarrow Z) \rightarrow ((Y \rightarrow Z) \rightarrow ((X \vee Y) \rightarrow Z))$.

22. Tìm DCTH của các công thức a, b và c trong bài 21. Từ đó suy ra sự đồng nhất đúng của các công thức đó mà không cần lập bảng giá trị chân lý.

23. Cho công thức $\overline{X \rightarrow (Y \rightarrow X)}$. Tìm DCTH và DCTT của công thức trên. Vì sao công thức trên là đồng nhất sai?

24. Tìm DCTH và DCTT của các công thức a, b và c trong bài 21 bằng cách thực hiện các bước biến đổi tương đương lôgic sau:

Bước 1: Khử phép toán kéo theo ($\rightarrow$);

Bước 2: Đưa phép toán phủ định về liên quan trực tiếp với từng mệnh đề X, Y và Z;

Bước 3: Biến đổi công thức nhận được ở bước 2 về DCTH và DCTT. Từ DCTH của mỗi công thức a, b và c suy ra công thức a, b và c là đồng nhất đúng mà không cần dựa vào bảng giá trị chân lý của nó.

25. Lôgic mờ được sử dụng nhiều trong lĩnh vực trí tuệ nhân tạo. Trong lôgic mờ giá trị chân lý của một mệnh đề là một số nằm giữa 0 và 1. Một mệnh đề với giá trị 0 là mệnh đề sai, còn mệnh đề với giá trị 1 là mệnh đề đúng. Còn giá trị mệnh đề nằm giữa 0 và 1 nói lên mức độ thay đổi chân lý của mệnh đề.

a) Phủ định giá trị chân lý của một mệnh đề trong lôgic mờ là hiệu giữa 1 và giá trị chân lý của mệnh đề đó.

Hãy xác định chân lý của các mệnh đề: "A không hạnh phúc" và "B không hạnh phúc" nếu mệnh đề "A hạnh phúc" có giá trị chân lý là 0,8 và "B hạnh phúc" có giá trị chân lý là 0,4.

b) Giá trị chân lý của tuyển hai mệnh đề trong lôgic mờ là giá trị chân lý lớn nhất của hai mệnh đề đó.

Hãy xác định chân lý của các mệnh đề sau:

– "A hạnh phúc hoặc B hạnh phúc";

– "A không hạnh phúc hoặc B không hạnh phúc".

Cho biết giá trị chân lý của mệnh đề "A hạnh phúc" là 0,9 còn của "B hạnh phúc" là 0,8.

c) Giá trị chân lý của hội hai mệnh đề là giá trị chân lý nhỏ nhất của hai mệnh đề đó.

Hãy xác định giá trị chân lý của hai mệnh đề sau:

– "A và B đều hạnh phúc";

– "A và B đều không hạnh phúc".

Biết rằng, giá trị chân lý của "A hạnh phúc" là 0,8 và của "B hạnh phúc" là 0,6.

26. Chứng minh rằng:

a) Công thức $\overline{X \vee (\overline{X} \wedge Y)}$ và $(\overline{X} \wedge \overline{Y})$ là tương đương lôgic;

b) Công thức $(X \wedge Y) \rightarrow (X \vee Y)$ là hằng đúng (hay đồng nhất đúng).

27. Tìm công thức đối ngẫu của các công thức sau:

a) $X \wedge \overline{Y} \wedge \overline{Z}$;

b) $(X \wedge Y \wedge Z) \vee X$;

c) $(X \vee F) \wedge (Y \vee T)$

(với F chỉ giá trị sai, còn T chỉ giá trị đúng).

28. Chứng minh đối ngẫu của đối ngẫu một công thức là chính nó.

29. Một tập hợp các phép toán lôgic được gọi là đầy đủ nếu mọi công thức đều tương đương lôgic với một công thức chỉ chứa các phép toán lôgic đó.

a) Chứng minh rằng $\vee$, $\wedge$ và — tạo thành một tập hợp đầy đủ của phép toán lôgic.

b) Chứng minh rằng $\vee$ và — ($\wedge$ và —) tạo nên một tập hợp đầy đủ của các phép toán lôgic.

30. Ta đưa vào phép toán lôgic mới NAND và NOR được định nghĩa như sau:

X	Y	X NAND Y	X NOR Y
1	1	0	0
1	0	1	0
0	1	1	0
0	0	1	1

Để cho gọn ta viết: $X \mid Y$ thay cho X NAND Y;

$X \downarrow Y$ thay cho X NOR Y.

a) Chứng minh $X \mid Y$ tương đương lôgic với $\overline{X \wedge Y}$.

b) Chứng minh $X \downarrow Y$ tương đương lôgic với $\overline{X \vee Y}$.

c) Chứng minh $X \downarrow X$ tương đương lôgic với $\overline{X}$.

d) Chứng minh $(X \downarrow Y) \downarrow (X \downarrow Y)$ tương đương lôgic với $X \vee Y$.

31. Chứng minh $\{\downarrow\}$ là tập hợp đầy đủ của các phép toán lôgic.

32. Tìm mệnh đề tương đương với $X \rightarrow Y$ bằng cách chỉ dùng phép toán $\downarrow$.

33. Chứng minh $X \mid Y$ và $Y \mid X$ là tương đương.

34. Chứng minh $X \mid (Y \mid Z)$ và $(X \mid Y) \mid Z$ là không tương đương (do đó phép toán $\mid$ không có tính kết hợp).

35. Xác định giá trị chân lý của vị từ một ngôi P(x) là "$x > 3$" tại $x = 2$ và $x = 4$.

36. Cho vị từ hai ngôi Q(x, y) là "$x = y + 3$". Xác định giá trị chân lý của mệnh đề Q(1, 2), Q(3, 0).

37. Cho vị từ ba ngôi R(x, y, z) là "$x + y = z$". Xác định giá trị chân lý của các mệnh đề R(1, 2, 3) < R(0, 0, 1).

38. Cho vị từ P(x) là câu: "Từ x chứa chữ cái a". Xác định giá trị chân lý của các mệnh đề:

a) P(Cam);

b) P(Quýt);

c) P(đúng);

d) P(sai).

39. Cho vị từ Q(x, y) là câu "x là thủ phủ của y". Xác định giá trị chân lý của các mệnh đề sau:

a) Q(Hà Nội, Việt Nam);

b) Q(Băng Cốc, Lào).

40. Cho câu lệnh: if $x > 0$ then $x := x + 1$.

Khi gặp câu lệnh này trong chương trình, giá trị của biến x ở điểm đó trong quá trình thực hiện chương trình sẽ được đặt vào vị từ một ngôi P(x), tức là câu "$x > 0$". Nếu P(x) đúng với giá trị này của x thì thực hiện lệnh gán $x := x + 1$, tức là giá trị của x tăng lên 1; ngược lại nếu P(x) sai tại giá trị này của x thì không thực hiện lệnh gán $x := x + 1$, tức giá trị x không thay đổi.

Cho biết giá trị của x sau khi lệnh if P(x) then $x := 1$ được thực hiện, biết rằng P(x) là câu "$x > 1$" và giá trị của x khi tới câu lệnh này là:

a) $x = 0$;

b) $x = 1$;

c) $x = 2$.

41. Diễn đạt câu: "Tất cả sinh viên ở lớp này đều đã học giải tích" như một lượng từ toàn thể.

42. Cho P(x) là câu: "$x + 1 > x$". Xác định giá trị chân lý của $(\forall x)$ P(x) trên trường các số thực.

43. Cho P(x) là câu "$x < 2$". Xác định các giá trị chân lý của $(\forall x)$ P(x) trên trường số thực.

44. Xác định giá trị chân lý của $(\forall x)$ P(x) với P(x) là câu: "$x \leq 10$" trên trường các số tự nhiên không vượt quá 4.

45. Cho P(x) là câu: "$x > 3$". Tìm giá trị chân lý của $(\exists x)$ P(x) trên trường các số thực.

46. Cho P(x) là câu: "$x = x + 1$". Tìm giá trị chân lý của $(\exists x)$ P(x) trên trường các số thực.

47. Cho P(x, y) với câu: "$x + y = y + x$" trên trường số thực. Xác định giá trị chân lý của $(\forall x)$ $(\forall y)$ P(x, y).

48. Cho Q(x, y) là câu: "$x + y = 0$" trên trường số thực. Xác định giá trị chân lý của $(\exists y)$ $(\forall x)$ Q(x, y) và $(\forall x)$ $(\exists y)$ Q(x, y).

49. Cho Q(x, y, z) là câu: "x + y = z" trên trường số thực. Xác định giá trị chân lý của $(\forall x)(\forall y)(\exists z)\, Q(x, y, z)$ và $(\exists z)(\forall x)(\forall y)\, Q(x, y, z)$.

50. Biểu diễn câu: "Mọi người đều có chính xác một người bạn rất tốt" thành một biểu thức lôgic.

51. Biểu diễn câu: "Nếu một phụ nữ nào đó đã sinh đẻ, thì người đó là mẹ của một người nào đó" thành một biểu thức lôgic.

52. Cho P(x) là câu: "x học ở lớp hơn 5 giờ mỗi ngày trong tuần" trên trường các sinh viên. Hãy diễn đạt các lượng từ sau thành câu thông thường:

a) $(\exists x)\, P(x)$;

b) $(\forall x)\, P(x)$;

c) $(\exists x)\, \overline{P}(x)$;

d) $(\forall x)\, \overline{P}(x)$.

53. Cho P(x, y) là câu: "x đã học môn y" với trường của x là tập hợp tất cả sinh viên trong lớp, còn trường của y là tập tất cả các môn Tin học của khoa đã quy định. Hãy diễn đạt các lượng từ sau thành câu thông thường:

a) $(\exists x)(\exists y)\, P(x, y)$;

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

54. F(x, y) là câu: "x yêu y", với x và y cho trên trường tập hợp mọi người trên thế giới.

Hãy dùng các lượng từ để diễn đạt các câu sau:

a) "Mọi người đều yêu Jerry";

b) "Mọi người đều yêu một người nào đó";

c) "Có một người mà tất cả mọi người đều yêu";

d) "Không có ai yêu tất cả mọi người";

e) "Có một người không ai yêu".

55. Cho P(x), Q(x) và F(x) là các câu tương ứng sau: "x là giáo sư"; "x là kẻ ngu ngốc" và "x là kẻ vô tích sự". Bằng cách dùng các lượng từ, các liên từ lôgic cùng với P(x), Q(x), F(x) để diễn đạt các câu sau trên trường tất cả mọi người:

a) "Không có giáo sư nào là kẻ ngu ngốc";

b) "Mọi kẻ ngu ngốc đều là vô tích sự";

c) "Không có giáo sư nào là vô tích sự".

56. Chứng tỏ rằng công thức $\overline{(\exists x)(\forall y)P(x,y)}$ có cùng giá trị chân lý với công thức $(\forall x)(\exists y)\overline{P}(x,y)$.

57. Chứng tỏ rằng $(\forall x)P(x) \vee (\forall x)Q(x)$ và $(\forall x)(P(x) \vee Q(x))$ là không tương đương lôgic; còn $(\forall x)P(x) \wedge A$ và $(\forall x)(P(x) \wedge A)$ là tương đương lôgic, với A là mênh đề không có phép toán lượng từ.

58. Chứng tỏ rằng $(\exists x)P(x) \wedge (\exists x)Q(x)$ và $(\exists x)(P(x) \wedge Q(x))$ là không tương đương lôgic.

59. Xác định giá trị chân lý của các công thức:

a) $(\exists x!)P(x) \rightarrow (\exists x)P(x)$;

b) $(\forall x)P(x) \rightarrow (\exists x!)P(x)$;

c) $(\exists x!)\overline{P}(x) \rightarrow \overline{(\forall x)P(x)}$.

Ở đây $(\exists x!)P(x)$ là ký hiệu mệnh đề: "Tồn tại duy nhất một x sao cho P(x) là đúng" trên trường các số nguyên.

60. Cho công thức lôgic vị từ

$$A \equiv ((\forall x)P(x) \wedge (\forall x)Q(x) \vee (\exists x)\overline{P}(x) \vee (\exists x)(Q(x) \wedge \overline{P}(x)))$$
$$\rightarrow (X \rightarrow (Y \wedge Z) \rightarrow X)$$

Hãy thực hiện các bước biến đổi tương đương lôgic sau:

Bước 1: Khử phép toán →;

Bước 2: Đưa dấu phủ định (—) về dạng chỉ liên quan trực tiếp tới các vị từ P(x), Q(x) và F(x).

Bước 3: Đưa các dấu lượng từ ∀, ∃ về trước công thức;

Bước 4: Đưa phần công thức đứng sau các lượng từ ∃, ∀ nhận được trong bước 3 về dạng tuyển của các HSC (hội của các TSC) mà trong HSC (TSC) gồm các vị từ P(x), Q(x) và các biến mệnh đề X, Y, Z cùng phủ định của nó.

61. Cho công thức lôgic vị từ

$$A \equiv ((\forall x)P(x) \vee (\forall x)Q(x) \vee (\exists x)P(x) \vee (\exists x)(Q(x) \wedge \overline{P}(x)))$$
$$\rightarrow (X \rightarrow (Y \rightarrow X))$$

Tìm DCTT và DCTH của công thức A.

62. Cho công thức lôgic vị từ

$$A \equiv \overline{(\forall x)P(x) \wedge (\forall x)Q(x)}$$
$$\rightarrow ((\exists x)F(x) \vee (\exists x)(\overline{Q}(x) \vee P(x)) \vee (X \rightarrow (X \rightarrow X)))$$

Hãy thực hiện các phép biến đổi tương đương đối với công thức A như sau:

Bước 1: Khử phép toán →;

Bước 2: Đưa dấu phủ định về trực tiếp tới các vị từ P(x), Q(x), F(x) và các biến mệnh đề X;

Bước 3: Đưa các lượng từ ∀, ∃ về trước công thức;

Bước 4: Phần công thức đứng sau các lượng từ ∀, ∃ hãy biến đổi về DCTH và DCTT.

63. Cho công thức lôgic vị từ

$$A \equiv ((\forall x)P(x) \wedge (\forall x)Q(x) \vee (X \rightarrow (Y \rightarrow Z)))$$
$$\rightarrow (((\exists x)R(x) \vee (\exists x)F(x)) \rightarrow (\forall x)(Q(x) \vee F(x) \vee R(x)))$$

Tìm công thức tương đương lôgic của A là A', trong đó A' không còn phép kéo theo và dấu phủ định chỉ liên quan trực tiếp tới các vị từ P(x), Q(x), R(x), F(x) và các biến mệnh đề X, Y, Z.

64. Các số điều hoà H_k (k = 1, 2, 3, ...) được định nghĩa như sau:

$$H_k = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{k}$$

Dùng phương pháp quy nạp toán học, chứng minh rằng:

$$H_{2^n} \geq 1 + \frac{n}{2} \text{ (n là số nguyên không âm).}$$

65. Dùng phương pháp quy nạp toán học chỉ ra rằng: Nếu A có n phần tử thì số các tập con của A là 2^n ($n \geq 0$).

66. Dùng phương pháp quy nạp toán học chứng minh công thức De Morgan:

$$\overline{\bigcap_{i=1}^{n} A_i} = \bigcup_{i=1}^{n} \overline{A_i} \text{ và } \overline{\bigcup_{i=1}^{n} A_i} = \bigcap_{i=1}^{n} \overline{A_i}$$

67. Giả sử n là một số nguyên dương. Một bàn cờ hình vuông có mỗi chiều bằng 2^n đơn vị và bị bỏ đi một ô vuông. Chỉ ra rằng có thể lát bàn cờ đó bằng các miếng hình chữ L (gồm ba hình vuông đơn vị) như hình vẽ

68. Chứng minh các đẳng thức sau đây:

a) $1.2 + 2.3 + ... + n(n + 1) = \dfrac{n(n+1)(n+2)}{3}$ với mọi $n \geq 1$.

b) $1.2.3 + 2.3.4 + ... + n(n + 1)(n + 2) = \dfrac{n(n+1)(n+2)(n+3)}{4}$ với mọi $n \geq 1$.

69. Chứng minh đẳng thức

$$\sum_{i=1}^{n} i^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30} \text{ với mọi } n \geq 1.$$

Chương 6
HỆ TOÁN MỆNH ĐỀ

Trong phần đại số mệnh đề, ta đã nghiên cứu các mệnh đề hoặc đúng, hoặc sai, mà không xét đến nội dung cũng như cấu trúc của mệnh đề. Trong thực tế có nhiều tập hợp, chẳng hạn tập các hàm đại số lôgic cũng có những tính chất như trong đại số mệnh đề. Vì vậy, ta nghiên cứu mệnh đề một cách tổng quát hơn. Một trong các phương pháp đó là phương pháp tiên đề hoá.

Chúng ta sẽ thấy phương pháp tiên đề hoá còn được sử dụng trong nhiều lĩnh vực khác, đặc biệt người ta sử dụng phương pháp tiên đề hoá để phân loại, tìm kiếm thông tin và lập bảng mã cho các thông tin đã được phân loại dưới dạng cây nhị nguyên.

Các môn lý thuyết được nghiên cứu và xây dựng theo trình tự sau đây:

1) Đưa ra các đối tượng nghiên cứu.

2) Xây dựng hệ tiên đề.

3) Xây dựng các định lý.

Chẳng hạn, môn hình học Ơclit:

– Đối tượng nghiên cứu là điểm, đường thẳng, mặt phẳng, ...

– Xây dựng các tiên đề Ơclit.

– Các kết quả của môn hình học Ơclit.

Có nhiều cách xây dựng hệ hình thức, tuỳ theo cách chọn các đối tượng nghiên cứu. Nhưng hệ hình thức chỉ có nghĩa khi nó phù hợp với thực tế. Muốn vậy hệ hình thức phải thoả mãn ba điều kiện sau:

1) Hệ hình thức là phi mâu thuẫn.

2) Hệ hình thức là đầy đủ.

3) Hệ hình thức là độc lập.

Khái niệm về phi mâu thuẫn, đầy đủ, độc lập sẽ được định nghĩa chính xác ở phần sau.

Hệ hình thức mà ta nghiên cứu dưới đây là hệ toán mệnh đề.

• ***Mô hình hệ toán mệnh đề***

1) Chọn ra một tập hợp đếm được các ký hiệu mà ta sẽ gọi là các ký hiệu của hệ toán. Dãy hữu hạn các ký hiệu đó gọi là mệnh đề.

2) Chọn ra tập hợp con các mệnh đề mà ta gọi là công thức của hệ toán.

3) Trong các công thức của hệ toán ta chọn ra một tập hợp con các công thức và gọi là các tiên đề của hệ toán.

4) Xây dựng quy tắc dẫn xuất.

Việc xây dựng hệ toán mệnh đề được thực hiện theo bốn bước trên.

§1. HỆ TOÁN MỆNH ĐỀ

1.1. Xây dựng hệ toán mệnh đề

Ta xây dựng hệ toán mệnh đề theo trình tự mô tả ở trên.

1.1.1. Các ký hiệu dùng trong hệ toán mệnh đề

1) Các chữ X, Y, Z (đôi khi cả số) dùng chỉ các mệnh đề sơ cấp.

2) Các ký hiệu $\vee$, $\wedge$, —, $\rightarrow$ dùng chỉ các phép toán hội, tuyển, phủ định, kéo theo tương ứng.

3) Ký hiệu "(", ")" gọi là dấu mở ngoặc và dấu đóng ngoặc.

4) Các chữ A, B, C, D, K (đôi khi cả chỉ số) dùng để chỉ các công thức của hệ toán.

Hệ toán mệnh đề chỉ gồm những ký hiệu trên, không còn những ký hiệu nào khác.

1.1.2. Định nghĩa công thức của hệ toán

Công thức trong hệ toán được định nghĩa đệ quy như sau:

1) Mỗi mệnh đề sơ cấp là một công thức.

2) Nếu A, B là hai công thức thì dãy các ký hiệu $(A \wedge B)$, $(A \vee B)$, $(\overline{A})$, $(A \rightarrow B)$ cũng là các công thức. Để cho gọn đôi khi ta viết $A \vee B$ thay cho $(A \vee B)$.

1.1.3. Các tiên đề của hệ toán mệnh đề

Ta chọn nhóm các công thức sau làm tiên đề của hệ toán mệnh đề:

$$(1) \quad \begin{cases} \text{a) } (A \rightarrow (B \rightarrow A)) \\ \text{b) } ((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))) \end{cases}$$

$$(2)\quad \begin{cases} a)\ ((A \wedge B) \to A) \\ b)\ ((A \wedge B) \to B) \\ c)\ ((A \to B) \to ((A \to C) \to (A \to (B \wedge C)))) \end{cases}$$

$$(3)\quad \begin{cases} a)\ (A \to (A \vee B)) \\ b)\ (B \to (A \vee B)) \\ c)\ ((A \to C) \to ((B \to C) \to ((A \vee B) \to C))) \end{cases}$$

$$(4)\quad \begin{cases} a)\ ((A \to B) \to (\overline{B} \to \overline{A})) \\ b)\ (A \to \overline{\overline{A}}) \\ c)\ (\overline{\overline{A}} \to A) \end{cases}$$

Để cho gọn ta ký hiệu i(x) là tiên đề x trong nhóm i ($i \in \{1, 2, 3, 4\}$, $x \in \{a, b, c\}$).

1.1.4. Quy tắc dẫn xuất

Nếu A và $A \to B$ thì B.

1.2. Các định nghĩa trong hệ toán mệnh đề

1.2.1. Định nghĩa định lý

a) Mỗi tiên đề là một định lý.

b) Nếu A và $A \to B$ là các định lý thì B cũng là một định lý.

A là một định lý ta ký hiệu là $\vdash A$.

1.2.2. Định nghĩa dẫn xuất

Một dẫn xuất trong hệ toán mệnh đề là dãy các công thức $A_1, A_2, ..., A_n$, trong đó A_i ($i = \overline{1,n}$) hoặc là tiên đề, hoặc trực tiếp suy ra từ các công thức đứng trước nó theo quy tắc dẫn xuất.

Từ định nghĩa dẫn xuất đôi khi ta còn định nghĩa định lý như sau:

Công thức A trong hệ toán mệnh đề được gọi là một định lý nếu tồn tại một dẫn xuất trong hệ toán mệnh đề mà công thức cuối cùng trong dẫn xuất đó là A. Nói cách khác, nếu A là một định lý thì tồn tại dãy các công thức $A_1, A_2, ..., A_n \equiv A$, trong đó A_i ($i = \overline{1,n}$) hoặc là tiên đề, hoặc suy ra từ công thức đứng trước nó theo quy tắc dẫn xuất. Người ta thường gọi dãy A_1, A_2, ..., A_n là một chứng minh của A.

1.2.3. Định nghĩa dẫn được

Công thức A gọi là dẫn được từ các công thức $A_1, A_2, ..., A_n$ (ký hiệu $A_1, A_2, ..., A_n \vdash A$) nếu tồn tại dãy các công thức $B_1, B_2, ..., B_m \equiv A$, trong đó B_i $(i = 1, 2, ..., m)$ hoặc A_j $(j = 1, ..., n)$ suy ra từ công thức đứng trước nó theo quy tắc dẫn xuất.

Dãy $B_1, B_2, ..., B_m$ được gọi là một dẫn xuất của A.

Chú ý: Từ các định nghĩa trên ta suy ra:

a) Mỗi tiên đề là một định lý.

b) Nếu A, B là các công thức mà $B \subseteq A$ thì $B \vdash C$ khi và chỉ khi $A \vdash C$.

c) Bản thân A dẫn được từ chính nó, hay $A \vdash A$.

1.3. Một số ví dụ về định lý

Ví dụ 1: Chứng minh công thức $(A \to B) \to (A \to A)$ là một định lý, hay $\vdash (A \to B) \to (A \to A)$.

Giải:

Thật vậy, theo 1(b) ta có:

$$\vdash (A \to (B \to A)) \to ((A \to B) \to (A \to A)) \quad (1)$$

Theo 1(a) ta có:

$$\vdash A \to (B \to A) \quad (2)$$

Từ (1), (2) theo quy tắc dẫn xuất ta có điều cần chứng minh là:

$$\vdash (A \to B) \to (A \to A).$$

Ví dụ 2: Chứng minh $\vdash (A \wedge B) \to (B \wedge A)$.

Giải:

Thật vậy, theo 2(c) ta có:

$$\vdash ((A \wedge B) \to B) \to (((A \wedge B) \to A) \to ((A \wedge B) \to (B \wedge A))) \quad (1)$$

Theo 2(b) ta có:

$$\vdash (A \wedge B) \to B \quad (2)$$

Từ (1), (2) theo quy tắc dẫn xuất thì

$$\vdash ((A \wedge B) \to A) \to ((A \wedge B) \to (B \wedge A)) \quad (3)$$

Từ 2(a) ta có:

$$\vdash (A \wedge B) \to A \tag{4}$$

Từ (3) và (4) theo quy tắc dẫn xuất ta có:

$$\vdash (A \wedge B) \to (B \wedge A).$$

Ví dụ 3: Chứng minh $\vdash \overline{\overline{\overline{A}}} \to \overline{A}$.

Giải:

Thật vậy, do 4(a)

$$\vdash (A \to \overline{\overline{A}}) \to (\overline{\overline{\overline{A}}} \to \overline{A}) \tag{1}$$

Do 4(b)

$$\vdash (A \to \overline{\overline{A}}) \tag{2}$$

Từ (1), (2) theo quy tắc dẫn xuất ta có

$$\vdash \overline{\overline{\overline{A}}} \to \overline{A}.$$

§2. CÁC TÍNH CHẤT CỦA HỆ TOÁN MỆNH ĐỀ

Định lý 1:

Nếu $A_1, A_2, ..., A_m \vdash B_1$ (1)

$A_1, A_2, ..., A_m \vdash B_2$ (2)

...

$A_1, A_2, ..., A_m \vdash B_p$ (p)

và $B_1, B_2, ..., B_p \vdash A$ thì $A_1, A_2, ..., A_m \vdash A$.

Chứng minh: Theo định nghĩa dẫn được thì từ (1), (2), ..., (p) có tồn tại các dãy công thức:

$$C_1^{(1)}, C_2^{(1)}, ..., C_{n_1}^{(1)} \equiv B_1.$$

$$C_1^{(2)}, C_2^{(2)}, ..., C_{n_2}^{(2)} \equiv B_2.$$

...

$$C_1^{(p)}, C_2^{(p)}, ..., C_{n_p}^{(p)} \equiv B_p.$$

trong đó $C_i^{(k)}$ hoặc là tiên đề, hoặc là A_j ($j = 1, 2, ..., m$), hoặc suy từ các công thức đứng trước nó theo quy tắc dẫn xuất. Mặt khác, theo giả thiết $B_1, B_2, ..., B_p \vdash A$ nên tồn tại dãy các công thức $C_1^{(p+1)}, C_2^{(p+1)}, ..., C_{n_{p+1}}^{(p+1)} \equiv A$, ở đây $C_i^{(p+1)}$ hoặc là tiên đề, hoặc là B_j ($j = 1, 2, ..., p$), hoặc suy từ công thức đứng trước nó theo quy tắc dẫn xuất.

Ghép p + 1 các dãy trên ta được dãy:

$$C_1^{(1)}, C_2^{(1)}, ..., C_{n_1}^{(1)} \equiv B_1, C_1^{(2)}, C_2^{(2)}, ..., C_{n_2}^{(2)} \equiv B_2,$$

...

$$C_1^{(p)}, C_2^{(p)}, ..., C_{n_p}^{(p)} \equiv B_p, C_1^{(p+1)}, C_2^{(p+1)}, ..., C_{n_{p+1}}^{(p+1)} \equiv A$$

là một dẫn xuất của A từ $A_1, A_2, ..., A_m$ hay $A_1, A_2, ..., A_m \vdash A$. Định lý được chứng minh.

Định lý 2: Nếu $A_1, A_2, ..., A_{n-1} \vdash A_n \to B$ thì $A_1, A_2, ..., A_{n-1}, A_n \vdash B$.

Chứng minh: Từ giả thiết có tồn tại các công thức $B_1, B_2, ..., B_m \equiv A_n \to B$, trong đó B_i ($i = 1, 2, ..., m$) hoặc là tiên đề, hoặc là A_i ($i = 1, 2, ..., n - 1$), hoặc suy từ các công thức đứng trước nó theo quy tắc dẫn xuất.

Xét dãy $B_1, B_2, ..., B_m, A_n, B$ (1)

Ta chỉ ra (1) là một dẫn xuất của B từ $A_1, A_2, ..., A_n$. Thật vậy, $B_1, B_2, ..., b_m, A_n$ hoặc là tiên đề, hoặc là A_i ($i = 1, 2, ..., n$), hoặc suy từ các công thức đứng trước nó theo quy tắc dẫn xuất.

Vậy chỉ cần xét thêm B, Ta có

$$B_m, A_n \vdash B_m$$
$$B_m, A_n \vdash A_n$$

(do chú ý trong mục 1.2.3).

Vì $B_m \equiv A_n \to B$ nên ta có $B_m, A_n \vdash A_n \to B$ và $B_m, A_n \vdash A_n$. Từ đó theo quy tắc dẫn xuất ta có $B_m, A_n \vdash B$ hay B suy ra từ công thức đứng trước nó theo quy tắc dẫn xuất. Tóm lại dãy (1) chính là một dẫn xuất của B từ $A_1, A_2, ..., A_n$ hay $A_1, A_2, ..., A_n \vdash B$.

Định lý được chứng minh.

Định lý 3: Nếu $\vdash D$ thì $\vdash A \to D$, ở đây A là công thức bất kỳ.

Chứng minh: Sử dụng tiên đề 1(a), quy tắc dẫn xuất và giả thiết.

Định lý 4: Nếu A là một công thức bất kỳ thì ta luôn có $\vdash A \to A$.

Chứng minh: Ta lần lượt có:

$\vdash (A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$ (1) (do 1(b))

$\vdash A \to ((A \to A) \to A))$ (2) (do 1(a)).

Từ (1), (2) theo quy tắc dẫn xuất ta có:

$$\vdash (A \to (A \to A)) \to (A \to A) \qquad (3)$$

Mặt khác theo 1(a) thì:

$$\vdash (A \to (A \to A)) \qquad (4)$$

Từ (3), (4) theo quy tắc dẫn xuất ta có công thức cần chứng minh là $\vdash A \to A$. Định lý được chứng minh.

Định lý 5: (Định lý suy diễn)

Nếu $A_1, A_2, ..., A_n \vdash B$ thì $A_1, A_2, ..., A_{n-1} \vdash A_n \to B$ ($n \geq 1$).

Chứng minh: Quy nạp theo định nghĩa dẫn được của B từ $A_1, A_2, ..., A_n$.

– Nếu B là một tiên đề thì $\vdash B$ (chú ý trong mục 1.2.3).

Theo định lý 3 thì $\vdash A_n \to B$. Theo định lý 2 và 3 ta có

$$A_1, A_2, ..., A_{n-1} \vdash A_n \to B.$$

– Nếu B là một trong các công thức A_i ($i = 1, 2, ..., n-1$) thì ta cũng có:

$A_1, A_2, ..., A_{n-1} \vdash B$ (chú ý trong mục 1.2.3) (1)

Theo 1(a) thì $\vdash B \to (A_n \to B)$ hay

$$A_1, A_2, ..., A_{n-1} \vdash B \to (A_n \to B) \qquad (2)$$

Từ (1), (2) và quy tắc dẫn xuất ta có:

$$A_1, A_2, ..., A_{n-1} \vdash A_n \to B.$$

Trường hợp $B \equiv A_n$ thì do định lý 4 ta có: $\vdash A_n \to B$ vì vậy ta cũng có $A_1, A_2, ..., A_{n-1} \vdash A_n \to B$.

– Giả sử B' và B' → B có tính chất sau:

$$A_1, A_2, ..., A_{n-1} \vdash A_n \to B' \qquad (3)$$

và $$A_1, A_2, ..., A_{n-1} \vdash A_n \to (B' \to B) \qquad (4)$$

Ta chứng minh: $A_1, A_2, ..., A_{n-1} \vdash A_n \to B$.

Thật vậy theo 1(b) ta có:

$$A_1, A_2, ..., A_{n-1} \vdash (A_n \to (B' \to B)) \to ((A_n \to B') \to (A_n \to B)) \qquad (5)$$

Từ (4), (5) theo quy tắc dẫn xuất thì:

$$A_1, A_2, ..., A_{n-1} \vdash (A_n \to B) \to (A_n \to B) \qquad (6)$$

Từ (3), (6) theo quy tắc dẫn xuất thì $A_1, A_2, ..., A_{n-1} \vdash A_n \to B$. Định lý được chứng minh.

Hệ quả: Nếu $A_1, A_2, ..., A_n \vdash B$ thì $\vdash A_1 \to (A_2 \to ... (A_n \to B)...)$.

Áp dụng định lý suy diễn n lần.

Định lý 6: (Quy tắc bắc cầu)

Nếu $\vdash A \to B$ và $\vdash B \to C$ thì $\vdash A \to C$.

Chứng minh: Từ $\vdash A \to B$ theo định lý 2 ta có $A \vdash B$.

Tương tự từ $\vdash B \to C$ ta cũng có $B \vdash C$.

Theo định lý 1 thì $A \vdash C$ và áp dụng định lý suy diễn ta có: $\vdash A \to C$.

Định lý 7: $\vdash A$ và $\vdash B$ khi và chỉ khi $\vdash A \wedge B$.

Chứng minh:

Điều kiện cần: Giả sử $\vdash A$ và $\vdash B$. Lấy công thức D sao cho $\vdash D$. Theo 2(c) thì

$$\vdash (D \to A) \to ((D \to B) \to (D \to (A \wedge B))) \qquad (1)$$

nhưng do $\vdash A$ nên theo định lý 3 thì

$$\vdash D \to A \qquad (2)$$

Từ (1), (2) theo quy tắc dẫn xuất ta có:

$$\vdash (D \to B) \to (D \to (A \wedge B)) \qquad (3)$$

Vì $\vdash B$ nên $\vdash D \to B$ (do định lý 3) (4)

Từ (3) và (4) ta lại có:

$$\vdash D \to (A \wedge B) \qquad (5)$$

Mặt khác: $\vdash D$ (6)

Từ (5) và (6) ta có: $\vdash A \wedge B$.

Điều kiện đủ: Giả sử $\vdash A \wedge B$, theo 2(a) thì

$$\vdash (A \wedge B) \to A \qquad (7)$$

và theo 2(b) ta có

$$\vdash (A \wedge B) \to B \qquad (8)$$

Từ giả thiết và (7), (8) ta có $\vdash A$ và $\vdash B$.

Định lý được chứng minh.

Định lý 8: (Hoán vị giả thiết)

Nếu $\vdash A \to (B \to C)$ thì $\vdash B \to (A \to C)$.

Chứng minh: Từ $\vdash A \to (B \to C)$ theo định lý 2 thì $A \vdash B \to C$ hay $A, B \vdash C$. Ta cũng có thể viết $B, A \vdash C$. Áp dụng định lý suy diễn 2 lần ta có:

$$\vdash B \to (A \to C).$$

Định nghĩa 1:

1) K gọi là công thức sai trong hệ toán mệnh đề nếu $\vdash \overline{K}$.

2) D gọi là công thức đúng trong hệ toán mệnh đề nếu $\vdash D$.

Định lý 9: Nếu K là công thức sai và A là công thức bất kỳ thì $\vdash K \to A$.

Chứng minh: Theo 4(a) thì

$$\vdash (\overline{A} \to \overline{K}) \to (\overline{\overline{K}} \to \overline{\overline{A}}) \tag{1}$$

Vì $\vdash \overline{K}$ nên áp dụng định lý 3 ta có:

$$\vdash \overline{A} \to \overline{K} \tag{2}$$

Từ (1), (2) theo quy tắc dẫn xuất thì

$$\vdash \overline{\overline{K}} \to \overline{\overline{A}} \tag{3}$$

Theo 4(b) thì

$$\vdash K \to \overline{\overline{K}} \tag{4}$$

Từ (3), (4) theo quy tắc bắc cầu ta có:

$$\vdash K \to \overline{\overline{A}} \tag{5}$$

Theo 4(c) thì

$$\vdash \overline{\overline{A}} \to A \tag{6}$$

Từ (5), (6) ta có $\vdash K \to A$. Định lý được chứng minh.

Định lý 10: (Quy tắc ghép giả thiết)

Nếu $\vdash A \to (B \to C)$ thì $\vdash (A \wedge B) \to C$.

Chứng minh: Theo 2(a) ta có: $\vdash (A \wedge B) \to A$. Áp dụng định lý 2 ta có:

$$A \wedge B \vdash A \tag{1}$$

Theo 2(b) ta có: $\vdash (A \wedge B) \to B$. Áp dụng định lý 2 ta có:

$$A \wedge B \vdash B \tag{2}$$

Từ giả thiết $\vdash A \to (B \to C)$. Áp dụng định lý 2 ta có:

$$A, B \vdash C \qquad (3)$$

Từ (1), (2), (3) và do định lý 1 ta có:

$A \wedge B \vdash C$, hay $\vdash (A \wedge B) \to C$ (suy từ định lý suy diễn).

Định lý 11: (Quy tắc tách giả thiết)

Nếu $\vdash (A \wedge B) \to C$ thì $\vdash A \to (B \to C)$.

Chứng minh: Vì $\vdash A \to (B \to (A \wedge B))$ nên áp dụng hai lần định lý 2 ta có:

$$A, B \vdash A \wedge B \qquad (1)$$

Áp dụng định lý 2 cho giả thiết ta có:

$$A \wedge B \vdash C \qquad (2)$$

Từ (1), (2) và theo định lý 1 ta có:

$A, B \vdash C$ hay $\vdash A \to (B \to C)$ (do định lý suy diễn).

Định lý 12: Với mọi công thức A và B ta luôn có

$$\vdash (A \wedge \overline{A}) \to B$$

Chứng minh: Suy từ định lý 9, vì $A \wedge \overline{A}$ là một công thức sai.

Định lý 13: Nếu A_1, A_2 là hai công thức sao cho $\vdash A_1 \to A_2$ thì ta có các trường hợp sau:

a) $\vdash (A_1 \wedge A) \to (A_2 \wedge A)$

b) $\vdash (A \wedge A_1) \to (A \wedge A_2)$

c) $\vdash (A_1 \vee A) \to (A_2 \vee A)$

d) $\vdash (A \vee A_1) \to (A \vee A_2)$

e) $\vdash (A_2 \to A) \to (A_1 \to A)$

f) $\vdash (A \to A_1) \to (A \to A_2)$

Chứng minh: Trước hết ta chứng minh hai bổ đề phụ sau đây:

Bổ đề 1: $\vdash A \to (B \wedge C)$ khi và chỉ khi $\vdash A \to B$ và $\vdash A \to C$.

Chứng minh:

Điều kiện cần: Giả sử $\vdash A \to (B \wedge C)$ (1)

Theo 2(a) thì $\vdash (B \wedge C) \to B$ (2)

Từ (1), (2) áp dụng quy tắc bắc cầu ta có: $\vdash A \to B$.

Theo 2(b) thì $\vdash (B \wedge C) \to C$ (3)

Từ (1), (3) theo quy tắc bắc cầu ta có: $\vdash A \to C$ (4)

Điều kiện đủ: Giả sử $\vdash A \to B$ và $\vdash A \to C$. Theo 2(c) thì

$$\vdash (A \to B) \to ((A \to C) \to (A \to (B \wedge C))) \quad (5)$$

Từ giả thiết và (5) theo quy tắc dẫn xuất ta có: $\vdash A \to (B \wedge C)$.

Bổ đề được chứng minh.

Bổ đề 2: $\vdash (A \vee B) \to C$ khi và chỉ khi $\vdash A \to C$ và $\vdash B \to C$.

Chứng minh:

Điều kiện cần: Giả sử $\vdash (A \vee B) \to C$ (1)

Theo 3(a) thì $\vdash A \to (A \vee B)$ (2)

Theo 3(b) thì $\vdash B \to (A \vee B)$ (3)

Từ (1), (3) và theo quy tắc bắc cầu ta có: $\vdash B \to C$.

Từ (1), (2) và theo quy tắc bắc cầu ta có: $\vdash A \to C$.

Điều kiện đủ: Giả sử $\vdash A \to C$ và $\vdash B \to C$. Theo 3(c) thì

$$\vdash (A \to C) \to ((B \to C) \to ((A \vee B) \to C)) \quad (4)$$

Từ giả thiết và (4), theo quy tắc dẫn xuất ta có:

$$\vdash (A \vee B) \to C.$$

Bổ đề được chứng minh.

Bây giờ ta chứng minh định lý.

a) Chứng minh: $\vdash (A_1 \wedge A) \to (A_2 \wedge A)$.

Để chứng minh a) theo bổ đề 1 cần chứng minh:

$$\vdash (A_1 \wedge A) \to A_2 \quad (1)$$

$$\vdash (A_1 \wedge A) \to A \quad (2)$$

Ở đây (2) là hiển nhiên theo 2(b). Ta chỉ cần chứng minh (1).

Theo 2(a) thì $\vdash (A_1 \wedge A) \to A_1$ (3)

và theo giả thiết ta có: $\vdash A_1 \to A_2$ (4)

Từ (3), (4) theo quy tắc bắc cầu thì $\vdash (A_1 \wedge A) \to A_2$.

Từ (1), (2) áp dụng bổ đề 1 ta có:

$$\vdash (A_1 \wedge A) \to (A_2 \wedge A).$$

b) Cũng theo bổ đề 1 để chứng minh b) ta cần chứng minh:

$$\vdash (A \wedge A_1) \to A \quad (5)$$

$$\vdash (A \wedge A_1) \to A_2 \quad (6)$$

Ở đây (5) là hiển nhiên theo 2(a). Ta chỉ cần chứng minh (6).

Theo 2(b) thì $\vdash (A \wedge A_1) \to A_1$ (7)

và theo giả thiết $\vdash A_1 \to A_2$ (8)

Từ (7) và (8) ta có: $\vdash (A \wedge A_1) \to A_2$.

Từ (5), (6) theo bổ đề 1 thì:

$$\vdash (A \wedge A_1) \to (A \wedge A_2)$$

c) Chứng minh: $\vdash (A_1 \vee A) \to (A_2 \vee A)$

Theo bổ đề 2 thì để chứng minh c) ta cần chứng minh:

$$\vdash A_1 \to (A_2 \vee A)$$

$$\vdash A \to (A_2 \vee A)$$

Nhưng $\vdash A \to (A_2 \vee A)$ suy từ 3(b).

Ta chứng minh $\vdash A_1 \to (A_2 \vee A)$ như sau:

Theo 3(a) thì $\vdash A_2 \to (A_2 \vee A)$ và theo giả thiết thì $\vdash A_1 \to A_2$. Từ đó ta có $\vdash A_1 \to (A_2 \vee A)$ theo quy tắc bắc cầu.

d) Chứng minh: $\vdash (A \vee A_1) \to (A \vee A_2)$

Theo bổ đề 2 để chứng minh d) cần chứng minh hai biểu thức sau:

$$\vdash A \to (A \vee A_2)$$

$$\vdash A_1 \to (A \vee A_2)$$

Biểu thức $\vdash A \to (A \vee A_2)$ hiển nhiên do 3(a).

Ta chứng minh $\vdash A_1 \to (A \vee A_2)$.

Theo 3(b) thì $\vdash A_2 \to (A \vee A_2)$ và theo giả thiết $\vdash A_1 \to A_2$ ta có:

$$\vdash A_1 \to (A \vee A_2).$$

e) Chứng minh: $\vdash (A_2 \to A) \to (A_1 \to A)$

Theo câu b) thì $\vdash (A \wedge A_1) \to (A \wedge A_2)$.

Thay A bởi $A_2 \to A$ ta có:

$$\vdash (A_2 \to A) \wedge A_1 \to (A_2 \to A) \wedge A_2 \quad (7)$$

Theo định lý 4 ta có:

$$\vdash (A_2 \to A) \to (A_2 \to A)$$

và dùng quy tắc ghép giả thiết ta được:

$$\vdash (A_2 \to A) \wedge A_2 \to A \quad (8)$$

Từ (7), (8) theo quy tắc bắc cầu thì $\vdash (A_2 \to A) \wedge A_1 \to A$.

Lại tách giả thiết ta có: $\vdash (A_2 \to A) \to (A_1 \to A)$.

f) Chứng minh: $\vdash (A \to A_1) \to (A \to A_2)$

Theo định lý 4 thì $\vdash (A \to A_1) \to (A \to A_1)$. Ghép giả thiết ta được

$$\vdash (A \to A_1) \wedge A \to A_1 \tag{9}$$

Theo giả thiết thì $\vdash A_1 \to A_2$ và kết hợp với (9) ta có:

$$\vdash (A \to A_1) \wedge A \to A_2$$

Tách giả thiết ta được công thức cần chứng minh là:

$$\vdash (A \to A_1) \to (A \to A_2).$$

Tóm lại, định lý được chứng minh.

§3. ĐỊNH LÝ TƯƠNG ĐƯƠNG

Ta ký hiệu $A \sim B$ qua $(A \to B) \wedge (B \to A)$. Theo định lý 7 thì

$$\vdash (A \to B) \wedge (B \to A) \Leftrightarrow \vdash A \to B \text{ và } \vdash B \to A.$$

Do đó ta có định nghĩa sau đây:

Định nghĩa 2: (Định nghĩa tương đương)

Nếu $\vdash (A \to B) \wedge (B \to A)$ thì ta nói A và B là tương đương nhau (ký hiệu là $A \sim B$).

Cũng như vậy, nếu $\vdash A \to B$ và $\vdash B \to A$ thì ta nói A và B là tương đương nhau.

Chú ý: Định nghĩa tương đương trên thoả mãn ba tính chất:

1) Phản xạ: Với mọi công thức A ta luôn có $A \sim A$.

2) Đối xứng: Nếu $A \sim B$ thì $B \sim A$.

3) Bắc cầu: Nếu $A \sim B$ và $B \sim C$ thì $A \sim C$.

Giả sử A là một công thức bất kỳ, X là một mệnh đề sơ cấp. Thay X bởi B trong A, kết quả là một công thức mà ta ký hiệu là $\int_X^B A$ và được định nghĩa như sau:

Định nghĩa 3: (Định nghĩa ký hiệu $\int_X^B A$)

Quy nạp theo định nghĩa của công thức A.

1) Nếu A là một mệnh đề sơ cấp thì có hai khả năng sau đây:

- $A \equiv X$: Trong trường hợp này ta định nghĩa $\int_X^B A = B$.
- Nếu $A \not\equiv X$: Trường hợp này ta đặt $\int_X^B A = A$.

2) Giả sử đối với hai công thức A_1, A_2 đã được định nghĩa phép thế $\int_X^B A_1$ và $\int_X^B A_2$ tương ứng. Ta định nghĩa phép thế cho các công thức: $(A_1 \wedge A_2)$, $(A_1 \vee A_2)$, $(A_1 \to A_2)$ và $(\overline{A_1})$ như sau:

- $\int_X^B (A_1 \wedge A_2) = \int_X^B A_1 \wedge \int_X^B A_2$
- $\int_X^B (A_1 \vee A_2) = \int_X^B A_1 \vee \int_X^B A_2$
- $\int_X^B (A_1 \to A_2) = \int_X^B A_1 \to \int_X^B A_2$
- $\int_X^B (\overline{A_1}) = \overline{\int_X^B A_1}$

Định lý 14: (Định lý về tính tương đương)

Nếu X là một mệnh đề sơ cấp; A, B_1, B_2 là các công thức bất kỳ thì $\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} A \sim \int_X^{B_2} A \right)$.

Chứng minh: Quy nạp theo định nghĩa công thức A.

1) Nếu A là mệnh đề sơ cấp, thì có hai khả năng sau đây:

• $A \equiv X$, khi đó $\int_X^{B_1} A = B_1$ và $\int_X^{B_2} A = B_2$.

Theo định lý 4 ta có:

$$\vdash (B_1 \sim B_2) \to (B_1 \sim B_2) \text{ hay } \vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} A \sim \int_X^{B_2} A \right)$$

• $A \neq X$, khi đó $\int_X^{B_1} A = A$ và $\int_X^{B_2} A = A$.

Theo định lý 4 thì $\vdash A \to A$ hay $\vdash \left(\int_X^{B_1} A \sim \int_X^{B_2} A \right)$.

Áp dụng định lý 3 ta được $\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} A \sim \int_X^{B_2} A \right)$.

2) Giả sử ta chứng minh được cho hai công thức A_1 và A_2, tức là ta đã chứng minh được:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} A_1 \sim \int_X^{B_2} A_1 \right)$$

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} A_2 \sim \int_X^{B_2} A_2 \right)$$

Ta cần chứng minh các trường hợp sau:

a) $$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} (A_1 \wedge A_2) \sim \int_X^{B_2} (A_1 \wedge A_2) \right)$$

b) $$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} (A_1 \vee A_2) \sim \int_X^{B_2} (A_1 \vee A_2) \right)$$

c) $$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} (A_1 \to A_2) \sim \int_X^{B_2} (A_1 \to A_2) \right)$$

d) $\vdash (B_1 \sim B_2) \rightarrow \left(\int_X^{B_1} \overline{A}_1 \sim \int_X^{B_2} \overline{A}_1 \right)$.

Chứng minh a) Để chứng minh a) ta cần sử dụng bổ đề 1:

$$\vdash A \rightarrow (B \wedge C) \Leftrightarrow \vdash A \rightarrow B \text{ và } \vdash B \rightarrow A$$

Theo giả thiết quy nạp:

$$\vdash (B_1 \sim B_2) \rightarrow \left(\int_X^{B_1} A_1 \sim \int_X^{B_2} A_1 \right)$$

Áp dụng bổ đề 1 ta có:

$$\vdash (B_1 \sim B_2) \rightarrow \left(\int_X^{B_1} A_1 \rightarrow \int_X^{B_2} A_1 \right)$$

Áp dụng định lý 2 ta có:

$$B_1 \sim B_2 \vdash \int_X^{B_1} A_1 \rightarrow \int_X^{B_2} A_1 \qquad (1)$$

Theo 2(a) thì $\vdash \int_X^{B_1} (A_1 \wedge A_2) \rightarrow \int_X^{B_1} A_1$ hay

$$B_1 \sim B_2 \vdash \int_X^{B_1} (A_1 \wedge A_2) \rightarrow \int_X^{B_1} A_1 \qquad (2)$$

Từ (1), (2) và quy tắc bắc cầu ta có:

$$B_1 \sim B_2 \vdash \int_X^{B_1} (A_1 \wedge A_2) \rightarrow \int_X^{B_2} A_1 \qquad (3)$$

Theo 2(b) ta có: $\vdash \int_X^{B_1} (A_1 \wedge A_2) \rightarrow \int_X^{B_1} A_2$ hay

$$B_1 \sim B_2 \vdash \int_X^{B_1} (A_1 \wedge A_2) \rightarrow \int_X^{B_1} A_2 \qquad (4)$$

Do bổ đề 1 thì :

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} A_2 \sim \int_X^{B_2} A_2 \right)$$

Do bổ đề 1 thì:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} A_2 \to \int_X^{B_2} A_2 \right).$$

Theo định lý 2 ta có:

$$B_1 \sim B_2 \vdash \int_X^{B_1} A_2 \to \int_X^{B_2} A_2 \qquad (5)$$

Từ (4), (5) và quy tắc bắc cầu ta có:

$$(B_1 \sim B_2) \vdash \int_X^{B_1} (A_1 \wedge A_2) \to \int_X^{B_2} A_2 \qquad (6)$$

Từ (3), (6) theo bổ đề 1 ta có:

$$B_1 \sim B_2 \vdash \int_X^{B_1} (A_1 \wedge A_2) \to \int_X^{B_2} (A_1 \wedge A_2) \qquad (*)$$

Hoàn toàn tương tự đổi vai trò giữa B_1 và B_2 ta có:

$$B_1 \sim B_2 \vdash \int_X^{B_2} (A_1 \wedge A_2) \to \int_X^{B_1} (A_1 \wedge A_2) \qquad (**)$$

Theo định lý 7 từ (*) và (**) ta có:

$$B_1 \sim B_2 \vdash \left(\int_X^{B_1} (A_1 \wedge A_2) \to \int_X^{B_2} (A_1 \wedge A_2) \right) \wedge \left(\int_X^{B_2} (A_1 \wedge A_2) \to \int_X^{B_1} (A_1 \wedge A_2) \right)$$

hay $B_1 \sim B_2 \vdash \int_X^{B_1} (A_1 \wedge A_2) \sim \int_X^{B_2} (A_1 \wedge A_2)$.

Đến đây áp dụng định lý suy diễn ta có điều cần chứng minh là:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} (A_1 \wedge A_2) \sim \int_X^{B_2} (A_1 \wedge A_2) \right).$$

Chứng minh b) $\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} (A_1 \vee A_2) \sim \int_X^{B_2} (A_1 \vee A_2) \right)$.

Trong phần a) ta đã chỉ ra công thức:

$$B_1 \sim B_2 \vdash \int_X^{B_1} A_1 \to \int_X^{B_2} A_1 \qquad (1)$$

Theo 3(a) thì

$$B_1 \sim B_2 \vdash \int_X^{B_2} A_1 \to \int_X^{B_2} (A_1 \vee A_2) \qquad (2)$$

Từ (1), (2) và quy tắc bắc cầu ta có:

$$B_1 \sim B_2 \vdash \int_X^{B_1} A_1 \to \int_X^{B_2} (A_1 \vee A_2) \qquad (*)$$

Chứng minh tương tự như (*) ta được:

$$B_1 \sim B_2 \vdash \int_X^{B_1} A_2 \to \int_X^{B_2} (A_1 \vee A_2) \qquad (**)$$

Từ (*) và (**) theo bổ đề 2 ta có:

$$B_1 \sim B_2 \vdash \int_X^{B_1} (A_1 \vee A_2) \to \int_X^{B_2} (A_1 \vee A_2) \qquad (3)$$

Đổi vai trò giữa B_1 và B_2 ta có:

$$B_1 \sim B_2 \vdash \int_X^{B_2} (A_1 \vee A_2) \to \int_X^{B_1} (A_1 \vee A_2) \qquad (4)$$

Từ (3) và (4) theo định lý 7 ta có:

$$B_1 \sim B_2 \vdash \int_X^{B_1} (A_1 \vee A_2) \sim \int_X^{B_2} (A_1 \vee A_2),$$

Áp dụng định lý suy diễn ta được công thức cần phải chứng minh là:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} (A_1 \vee A_2) \sim \int_X^{B_2} (A_1 \vee A_2) \right)$$

Chứng minh c) $\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} (A_1 \to A_2) \sim \int_X^{B_2} (A_1 \to A_2) \right)$.

Theo giả thiết quy nạp ta có:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} A_1 \sim \int_X^{B_2} A_1 \right)$$

hay
$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} A_1 \to \int_X^{B_2} A_1 \right) \wedge \left(\int_X^{B_2} A_1 \to \int_X^{B_1} A_1 \right) \tag{1}$$

Theo bổ đề 1 từ (1) ta có:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_2} A_1 \to \int_X^{B_1} A_1 \right).$$

Áp dụng hai lần định lý 2 ta được:

$$B_1 \sim B_2,\ \int_X^{B_2} A_1 \vdash \int_X^{B_1} A_1 \tag{2}$$

Mặt khác theo định lý 4 ta có:

$$\vdash \int_X^{B_1} (A_1 \to A_2) \to \int_X^{B_1} (A_1 \to A_2)$$

hay
$$\vdash \left(\int_X^{B_1} A_1 \to \int_X^{B_1} A_2 \right) \to \left(\int_X^{B_1} A_1 \to \int_X^{B_1} A_2 \right).$$

Hoán vị giả thiết ta có:

$$\vdash \int_X^{B_1} A_1 \to \left(\int_X^{B_1} (A_1 \to A_2) \to \int_X^{B_1} A_2 \right)$$

hay ta có thể viết

$$B_1 \sim B_2,\ \int_X^{B_2} A_1 \vdash \int_X^{B_1} A_1 \to \left(\int_X^{B_1} (A_1 \to A_2) \to \int_X^{B_1} A_2 \right) \tag{3}$$

Từ (2), (3) theo quy tắc dẫn xuất ta có:

$$B_1 \sim B_2,\ \int_X^{B_2} A_1 \vdash \int_X^{B_1} (A_1 \to A_2) \to \int_X^{B_1} A_2$$

Theo định lý 2 ta có:

$$B_1 \sim B_2,\ \int_X^{B_2} A_1,\ \int_X^{B_1} (A_1 \to A_2) \vdash \int_X^{B_1} A_2 \tag{4}$$

Theo giả thiết quy nạp với A_2 thì:

$$\vdash (B_1) \sim (B_2) \to \left(\int_X^{B_1} A_2 \sim \int_X^{B_2} A_2 \right)$$

Theo bổ đề 1 ta có:

$$\vdash (B_1) \sim (B_2) \to \left(\int_X^{B_1} A_2 \to \int_X^{B_2} A_2 \right)$$

Áp dụng hai lần định lý 2 ta thu được:

$$B_1 \sim B_2,\ \int_X^{B_1} A_2 \vdash \int_X^{B_2} A_2 \tag{5}$$

$$B_1 \sim B_2,\ \int_X^{B_2} A_1,\ \int_X^{B_1} (A_1 \to A_2) \vdash B_1 \sim B_2 \tag{6}$$

Từ (4), (5) và (6) theo định lý 1 ta có:

$$B_1 \sim B_2,\ \int_X^{B_2} A_1,\ \int_X^{B_1} (A_1 \to A_2) \vdash \int_X^{B_2} A_2$$

hay
$$B_1 \sim B_2,\ \int_X^{B_1} (A_1 \to A_2),\ \int_X^{B_2} A_1 \vdash \int_X^{B_2} A_2$$

Đến đây áp dụng định lý suy diễn ta được:

$$B_1 \sim B_2 \vdash \int_X^{B_1} (A_1 \to A_2) \to \int_X^{B_2} (A_1 \to A_2) \tag{7}$$

Hoàn toàn tương tự đổi vai trò giữa B_1 và B_2 ta có:

$$B_1 \sim B_2 \vdash \int_X^{B_2}(A_1 \to A_2) \to \int_X^{B_1}(A_1 \to A_2) \tag{8}$$

Từ (7) và (8) theo định lý 7 ta thu được:

$$B_1 \sim B_2 \vdash \int_X^{B_1}(A_1 \to A_2) \sim \int_X^{B_2}(A_1 \to A_2)$$

Áp dụng định lý suy diễn ta có công thức cần chứng minh là:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1}(A_1 \to A_2) \sim \int_X^{B_2}(A_1 \to A_2)\right)$$

Chứng minh d) $\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1}\overline{A_1} \sim \int_X^{B_2}\overline{A_1}\right)$.

Theo giả thiết quy nạp đối với A_1 ta có:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1}A_1 \sim \int_X^{B_2}A_1\right) \tag{1}$$

Theo bổ đề 1, từ (1) ta có:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1}A_1 \to \int_X^{B_2}A_1\right) \tag{2}$$

Theo 4(a):

$$\vdash \left(\int_X^{B_1}A_1 \to \int_X^{B_2}A_1\right) \to \left(\int_X^{B_2}\overline{A_1} \to \int_X^{B_1}\overline{A_1}\right) \tag{3}$$

Từ (2), (3) và quy tắc bắc cầu ta có:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_2}\overline{A_1} \to \int_X^{B_1}\overline{A_1}\right) \tag{*}$$

Hoàn toàn tương tự ta có :

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1}\overline{A_1} \to \int_X^{B_2}\overline{A_1}\right) \tag{**}$$

Từ (*), (**) và bổ đề 1 ta có công thức cần chứng minh là:

$$\vdash (B_1 \sim B_2) \to \left(\int_X^{B_1} \overline{A}_1 \sim \int_X^{B_2} \overline{A}_1 \right)$$

Định lý được chứng minh.

Dưới đây ta luôn ký hiệu công thức đồng nhất đúng là D, công thức đồng nhất sai là K, tức là ta luôn có $\vdash D$ và $\vdash \overline{K}$.

Định lý 15: $\vdash (A \sim D) \sim A$.

Chứng minh: Theo định lý 7 để chứng minh định lý 15 ta cần chứng minh hai công thức:

a) $\vdash (A \sim D) \to A$.

b) $\vdash A \to (D \sim A)$.

Chứng minh a) Tức là chứng minh

$$\vdash (A \to D) \land (D \to A) \to A$$

Theo 2(b) ta có:

$$\vdash (A \sim D) \to (D \to A) \qquad (1)$$

Vì $\vdash D$ nên theo định lý 3 ta có:

$$\vdash (D \to A) \to D \text{ hay } D \to A \vdash D \qquad (2)$$

Mặt khác theo định lý 4 ta có:

$$\vdash (D \to A) \to (D \to A) \qquad (3)$$

Từ (3) ta có

$$D \to A \vdash D \to A \qquad (4)$$

Từ (2), (4) và quy tắc dẫn xuất ta có:

$$D \to A \vdash A$$

Áp dụng định lý suy diễn ta được:

$$\vdash (D \to A) \to A \qquad (5)$$

Từ (1), (5) theo quy tắc bắc cầu ta được:

$$\vdash (A \sim D) \to A$$

Chứng minh b) Do $\vdash D$ nên theo định lý 3 ta có:

$$\vdash A \to D \qquad (1)$$

Áp dụng định lý 3 cho (1) ta có:

$$\vdash A \to (A \to D) \qquad (2)$$

Theo 1(a):

$$\vdash A \to (D \to A) \tag{3}$$

Từ (2), (3) và bổ đề 1 ta có:

$$\vdash A \to (D \sim A)$$

Định lý được chứng minh.

Định lý 16: $\vdash (A \sim K) \sim \overline{A}$.

Chứng minh: Theo định lý 7 thì cần chứng minh hai công thức sau:

a) $\vdash (A \sim K) \to \overline{A}$.

b) $\vdash \overline{A} \to (A \sim K)$.

Chứng minh a) Theo 2(a) thì

$$\vdash (A \sim K) \to (A \to K) \tag{1}$$

Theo 4(a):

$$\vdash (A \to K) \to (\overline{K} \to \overline{A}) \tag{2}$$

Từ (1), (2) theo quy tắc bắc cầu ta có:

$$\vdash (A \sim K) \to (\overline{K} \to \overline{A}) \tag{3}$$

Hoán vị giả thiết trong (3) ta có:

$$\vdash \overline{K} \to ((A \sim K) \to \overline{A}) \tag{4}$$

Vì $\vdash \overline{K}$ (5) nên từ (4) và (5) theo quy tắc dẫn xuất ta có:

$$\vdash (A \sim K) \to \overline{A}.$$

Chứng minh b) Theo định lý 12 ta có:

$$\vdash A \wedge \overline{A} \to K$$

Tách giả thiết ta được:

$$\vdash A \to (\overline{A} \to K)$$

Hoán vị giả thiết ta có:

$$\vdash \overline{A} \to (A \to K) \tag{*}$$

Theo định lý 9 ta có:

$$\vdash K \to A$$

Áp dụng định lý 3 ta nhận được:

$$\vdash \overline{A} \to (A \sim K)$$

Định lý được chứng minh.

Định lý 17:

$$\vdash \left(\int_{X}^{D} A \wedge \int_{X}^{K} A \right) \rightarrow (((X \sim D) \rightarrow A) \wedge ((X \sim K) \rightarrow A))).$$

Chứng minh:

Theo định lý tương đương ta có:

$$\vdash (B_1 \sim B_2) \rightarrow \left(\int_{X}^{B_1} A \sim \int_{X}^{B_2} A \right)$$

Trong công thức này ta thay B_1 bởi X và B_2 bởi B ta được:

$$\vdash (X \sim B) \rightarrow \left(\int_{X}^{X} A \sim \int_{X}^{B} A \right)$$

Theo bổ đề 1 thì từ công thức trên ta có:

$$\vdash (X \sim B) \rightarrow \left(\int_{X}^{B} A \sim \int_{X}^{X} A \right)$$

hay $\vdash (X \sim B) \rightarrow \left(\int_{X}^{B} A \rightarrow A \right)$

Hoán vị giả thiết ta có:

$$\vdash \int_{X}^{B} A \rightarrow ((X \sim B) \rightarrow A) \qquad (1)$$

Trong công thức (1) thay B bởi D và K ta được:

$$\vdash \int_{X}^{D} A \rightarrow ((X \sim D) \rightarrow A) \qquad (2)$$

$$\vdash \int_{X}^{K} A \rightarrow ((X \sim K) \rightarrow A) \qquad (3)$$

Mặt khác, do 2(a) thì

$$\vdash \left(\int_{X}^{D} A \wedge \int_{X}^{K} A \right) \rightarrow \int_{X}^{D} A \qquad (4)$$

Từ (2) và (4) theo quy tắc bắc cầu ta nhận được:

$$\vdash\left(\int_X^D A \wedge \int_X^K A\right) \to ((X \sim D) \to A) \tag{5}$$

Tương tự từ 2(b) có:

$$\vdash\left(\int_X^D A \wedge \int_X^K A\right) \to \int_X^K A \tag{6}$$

Công thức $\vdash\left(\int_X^D A \wedge \int_X^K A\right) \to ((X \sim K) \to A)$ (7)

nhận được từ (3), (6) và quy tắc bắc cầu.

Từ (5) và (7) áp dụng bổ đề 1 ta có công thức cần chứng minh:

$$\vdash\left(\int_X^D A \wedge \int_X^K A\right) \to (((X \sim D) \to A) \wedge ((X \sim K) \to A))\,.$$

Định lý 18:

$$\vdash (((X \sim D) \to A) \wedge ((X \sim K) \to A)) \to (((X \sim D) \to A) \vee ((X \sim K) \to A))$$

Chứng minh: Theo 3(c) ta có:

$$\vdash (A \to C) \to ((B \to C) \to ((A \vee B) \to C))$$

Ghép giả thiết trong công thức này ta được:

$$\vdash ((A \to C) \wedge (B \to C)) \to ((A \vee B) \to C) \tag{1}$$

Trong (1) thay A bởi $X \sim D$, thay B bởi $X \sim K$, C thay bởi A ta được công thức cần chứng minh.

Định lý 19: Với mọi công thức A ta luôn có $\vdash A \vee \overline{A}$.

Chứng minh: Theo 3(a) ta có:

$$\vdash A \to (A \vee \overline{A}) \tag{1}$$

Theo 4(a) ta có:

$$\vdash (A \to (A \vee \overline{A})) \to (\overline{A \vee \overline{A}} \to \overline{A}) \tag{2}$$

Từ (1), (2) theo quy tắc dẫn xuất ta có:

$$\vdash \overline{A \vee \overline{A}} \to \overline{A} \tag{3}$$

Theo 3(b):

$$\vdash \overline{A} \rightarrow (A \vee \overline{A}) \qquad (4)$$

và theo 4(a) ta có:

$$\vdash (\overline{A} \rightarrow (A \vee \overline{A})) \rightarrow (\overline{A \vee \overline{A}} \rightarrow \overline{\overline{A}}) \qquad (5)$$

Từ (4) và (5) theo quy tắc dẫn xuất ta có:

$$\vdash (\overline{A \vee \overline{A}} \rightarrow \overline{\overline{A}}) \qquad (6)$$

Theo 4(c) thì

$$\vdash \overline{\overline{A}} \rightarrow A \qquad (7)$$

Từ (6), (7) ta có:

$$\vdash \overline{A \vee \overline{A}} \rightarrow A \qquad (8)$$

Từ (3), (8) theo bổ đề 1 ta có:

$$\vdash \overline{(A \vee \overline{A})} \rightarrow A \wedge \overline{A} \qquad (9)$$

Theo định lý 12 thì

$$\vdash (A \wedge \overline{A}) \rightarrow K \qquad (10)$$

Từ (9), (10) theo quy tắc bắc cầu ta có:

$$\vdash \overline{(A \vee \overline{A})} \rightarrow K \qquad (11)$$

Theo 4(a) thì

$$\vdash (\overline{A \vee \overline{A}} \rightarrow K) \rightarrow (\overline{K} \rightarrow \overline{\overline{A \vee \overline{A}}}) \qquad (12)$$

Từ (11), (12) theo quy tắc dẫn xuất ta được:

$$\vdash \overline{K} \rightarrow \overline{\overline{A \vee \overline{A}}} \qquad (13)$$

Theo 4(c) ta có:

$$\vdash \overline{\overline{A \vee \overline{A}}} \rightarrow (A \vee \overline{A}) \qquad (14)$$

Từ (12), (14) theo quy tắc bắc cầu ta có:

$$\vdash \overline{K} \rightarrow (A \vee \overline{A}) \qquad (15)$$

Theo định nghĩa công thức sai thì $\vdash \overline{K}$ (16)

Từ (15), (16) dùng quy tắc dẫn xuất ta có: $\vdash A \vee \overline{A}$.

Định lý được chứng minh.

Định lý 20: $\vdash (X \sim D) \vee (X \sim K)$.

Chứng minh: Theo định lý 15 thì

$$\vdash (X \sim D) \sim X \qquad (1)$$

Theo định lý 16 thì

$$\vdash (X \sim K) \sim \overline{X} \qquad (2)$$

Từ (1) theo định lý 7 ta có:

$$\vdash X \rightarrow (X \sim D) \qquad (3)$$

Mặt khác theo 3(a) thì

$$\vdash (X \sim D) \rightarrow ((X \sim D) \vee (X \sim K)) \qquad (4)$$

Từ (3), (4) theo quy tắc bắc cầu thì

$$\vdash X \rightarrow ((X \sim D) \vee (X \sim K)) \qquad (5)$$

Từ (2) theo định lý 7 ta có

$$\vdash \overline{X} \rightarrow (X \sim K) \qquad (6)$$

Theo 3(b) ta có

$$\vdash (X \sim K) \rightarrow ((X \sim D) \vee (X \sim K)) \qquad (7)$$

Từ (6) và (7) theo quy tắc bắc cầu ta có

$$\vdash \overline{X} \rightarrow ((X \sim D) \vee (X \sim K)) \qquad (8)$$

Từ (5), (8) theo bổ đề 2 ta được

$$\vdash (X \vee \overline{X}) \rightarrow ((X \sim D) \vee (X \sim K)) \qquad (9)$$

Theo định lý 19 thì

$$\vdash X \vee \overline{X} \qquad (10)$$

Từ (9), (10) theo quy tắc dẫn xuất ta có:

$$\vdash (X \sim D) \vee (X \sim K)$$

Định lý được chứng minh.

Định lý 21: $\vdash \left(\int_X^D A \wedge \int_X^K A\right) \rightarrow A$.

Chứng minh: Theo định lý 15 ta có:

$$\vdash \left(\int_X^D A \wedge \int_X^K A\right) \rightarrow (((X \sim D) \rightarrow A) \wedge ((X \sim K) \rightarrow A))) \qquad (1)$$

Theo định lý 18 ta có:

$$\vdash ((X \sim D) \to A) \wedge ((X \sim K) \to A)) \to (((X \sim D) \vee (X \sim K)) \to A) \quad (2)$$

Từ (1), (2) theo quy tắc bắc cầu ta có:

$$\vdash \left(\int_X^D A \wedge \int_X^K A \right) \to (((X \sim D) \vee (X \sim K)) \to A) \quad (3)$$

Hoán vị giả thiết trong (3) ta được:

$$\vdash ((X \sim D) \vee (X \sim K)) \to \left(\left(\int_X^D A \wedge \int_X^K A \right) \to A \right) \quad (4)$$

Theo định lý 20 thì

$$\vdash (X \sim D) \vee (X \sim K) \quad (5)$$

Từ (4) và (5) theo quy tắc dẫn xuất ta có:

$$\vdash \left(\int_X^D A \wedge \int_X^K A \right) \to A$$

Định lý được chứng minh.

Định nghĩa 4: Giả sử A là một công thức bất kỳ; $X_1, X_2, ..., X_n$ là các mệnh đề sơ cấp. Ta định nghĩa bằng quy nạp ký hiệu $\prod\limits_{X_1X_2...X_n} A$ như sau:

- $n = 1$: Ta định nghĩa $\prod\limits_{X_1} A := \int\limits_{X_1}^{D} A \wedge \int\limits_{X_1}^{K} A$.

- Giả sử ta đã định nghĩa cho n, tức là ta định nghĩa $\prod\limits_{X_1X_2...X_n} A$, bây giờ ta định nghĩa cho n + 1 như sau:

$$\prod_{X_1...X_{n+1}} := \int_{X_{n+1}}^{D} \left(\prod_{X_1...X_n} A \right) \wedge \int_{X_{n+1}}^{K} \left(\prod_{X_1...X_n} A \right)$$

Ta chứng minh định lý sau đây:

Định lý 22: $\vdash \left(\prod\limits_{X_1...X_n} A \right) \to A$.

Chứng minh: Quy nạp theo $n \geq 1$.

• n = 1: Ta cần chứng minh

$$\vdash \left(\prod_{X_1} A \right) \rightarrow A \text{ hay } \vdash \left(\int_{X_1}^{D} A \wedge \int_{X_1}^{K} A \right) \rightarrow A$$

Công thức này là đúng do định lý 21.

• Giả sử định lý đúng với n, tức là đã có:

$$\vdash \left(\prod_{X_1 \ldots X_n} A \right) \rightarrow A \tag{1}$$

Ta chứng minh $\vdash \left(\prod_{X_1 \ldots X_{n+1}} A \right) \rightarrow A$

Thật vậy ta có:

$$\prod_{X_1 \ldots X_{n+1}} A := \int_{X_{n+1}}^{D} \left(\prod_{X_1 \ldots X_n} A \right) \wedge \int_{X_{n+1}}^{K} \left(\prod_{X_1 \ldots X_n} A \right)$$

Theo định lý 21 thì

$$\vdash \int_{X_{n+1}}^{D} \left(\prod_{X_1 \ldots X_n} A \right) \wedge \left(\int_{X_{n+1}}^{K} \left(\prod_{X_1 \ldots X_n} A \right) \rightarrow \prod_{X_1 \ldots X_n} A \right) \tag{2}$$

Từ (1), (2) theo quy tắc bắc cầu ta có:

$$\vdash \int_{X_{n+1}}^{D} \left(\prod_{X_1 \ldots X_n} A \right) \wedge \left(\int_{X_{n+1}}^{K} \left(\prod_{X_1 \ldots X_n} A \right) \rightarrow A \right) \tag{3}$$

hay $\vdash \left(\prod_{X_1 \ldots A_{n+1}} A \right) \rightarrow A.$

Định lý được chứng minh.

Định lý 23: Nếu A là một định lý trong hệ toán mệnh đề thì A là công thức đồng nhất đúng trong đại số mệnh đề.

Chứng minh: Giả sử A là định lý trong hệ toán mệnh đề, hay $\vdash A$. Ta chứng minh A là công thức đồng nhất đúng trong đại số mệnh đề, hay $\models A$.

Ta chứng minh $\models A$ theo định nghĩa, định lý.

• A là một tiên đề.

Trường hợp này là hiển nhiên vì mỗi tiên đề là một công thức đồng nhất đúng.

• Nếu $\vdash A$ và $\vdash A \rightarrow B$ thì $\vdash B$.

Ta chỉ ra nếu $\models A$ và $\models A \rightarrow B$ thì $\models B$. Giả sử ngược lại, B không là công thức đồng nhất đúng, tức là có tồn tại bộ giá trị đúng, sai sao cho với bộ giá trị đó thì B nhận giá trị sai. Vì A là đồng nhất đúng nên với bộ giá trị đó A nhận giá trị đúng. Điều này chứng tỏ $A \rightarrow B$ nhận giá trị sai trên bộ giá trị đúng, sai đó. Điều này mâu thuẫn với giả thiết $A \rightarrow B$ là công thức đồng nhất đúng. Định lý được chứng minh.

§4. QUAN HỆ GIỮA LÔGIC MỆNH ĐỀ VÀ HỆ TOÁN MỆNH ĐỀ

Giả sử A là một công thức trong hệ toán mệnh đề. A có thể là một công thức trong đại số mệnh đề nếu ta xem các biến sơ cấp trong hệ toán mệnh đề là các biến sơ cấp trong đại số mệnh đề, nghĩa là chúng nhận giá trị hoặc đúng, hoặc sai. Ngoài ra các phép toán $\wedge$, $\vee$, $\overline{}$, $\rightarrow$ cũng định nghĩa như trong đại số mệnh đề. Khi đó mỗi công thức trong hệ toán mệnh đề sẽ nhận giá trị đúng hoặc sai theo cách tính của đại số mệnh đề.

Trong công thức A của hệ toán mệnh đề tiến hành phép thế mỗi biến mệnh đề bởi công thức D và K. Ta xem A như là công thức của đại số mệnh đề và thế các biến mệnh đề bởi giá trị đúng hoặc sai tuỳ thuộc vào biến mệnh đề đó trong hệ toán mệnh đề nhận công thức D hay công thức K. Nói cách khác, nếu biến mệnh đề trong A nhận công thức D thì trong đại số mệnh đề ta thay biến đó bởi giá trị đúng, còn nếu biến mệnh đề nhận công thức K thì trong đại số mệnh đề biến đó được thay bởi giá trị sai. Khi đó công thức A trong đại số mệnh đề nhận một trong hai giá trị đúng hoặc sai.

Ta đã định nghĩa ký hiệu $\int_X^B A$ (thay X bởi B trong A).

Bây giờ ta định nghĩa ký hiệu mở rộng $\int_{X_1 \ldots X_n}^{G_1 \ldots G_n} A$.

Định nghĩa 5: Giả sử A là một công thức; $A_1, X_2, ..., X_n$ là các mệnh đề sơ cấp; $B_1, B_2, ..., B_n$ là các công thức nào đó.

Lấy các công thức $Y_1, Y_2, ..., Y_n$ sao cho các công thức này không nằm trong các công thức $B_1, B_2, ..., B_n$.

Ta định nghĩa ký hiệu mở rộng trên như sau:

$$\int_{X_1...X_n}^{Y_1...Y_n} A := \int_{X_n}^{Y_n}\left(\int_{X_{n-1}}^{Y_{n-1}}\left(...\left(\int_{X_1}^{Y_1} A\right)...\right)\right)$$

Tiếp theo ta định nghĩa:

$$\int_{Y_1...Y_n}^{B_1...B_n} A := \int_{Y_n}^{B_n}\left(\int_{Y_{n-1}}^{B_{n-1}}\left(...\left(\int_{Y_1}^{B_1} A\right)...\right)\right)$$

Cuối cùng ta định nghĩa:

$$\int_{Y_1...Y_n}^{B_1...B_n} A := \int_{Y_1...Y_n}^{B_1...B_n}\left(\int_{X_1...X_n}^{Y_1...Y_n} A\right)$$

Ta có định lý sau về quan hệ giữa hệ toán mệnh đề và đại số mệnh đề.

Định lý 24: Giả sử A là một công thức; $\delta_1, \delta_2, ..., \delta_n$ là bộ giá trị T, F, ở đây $\delta_i \in \{T, F\}$ $(i = \overline{1,n})$. Lấy các công thức $G_1, G_2, ..., G_n$ sao cho

$$G_i \equiv \begin{cases} D \text{ nếu } \delta_i = T \\ K \text{ nếu } \delta_i = F \end{cases}$$

Giả sử $X_1, X_2, ..., X_n$ là các mệnh đề sơ cấp trong A, cho các mệnh đề sơ cấp này nhận bộ $\delta_1, \delta_2, ..., \delta_n$ tương ứng.

Khi đó:

- Nếu A (trong đại số mệnh đề) nhận giá trị T thì trong hệ toán mệnh đề ta có

$$\vdash \left(\int_{X_1...X_n}^{G_1...G_n} A\right) \sim D.$$

- Nếu A (trong đại số mệnh đề) nhận giá trị F thì trong hệ toán mệnh đề ta có

$$\vdash \left(\int_{X_1...X_n}^{G_1...G_n} A\right) \sim K.$$

Chứng minh: Quy nạp theo định nghĩa công thức.

1) A là công thức sơ cấp, khi đó $A \equiv X_i$ ($i = \overline{1, n}$) ta có

$$\vdash \int_{X_1 \ldots X_n}^{G_1 \ldots G_n} A = G_i$$

Khi thay X_i bởi δ_i mà $\delta_i = T$ thì $G_i \equiv D$ hay $\vdash \int_{X_1 \ldots X_n}^{G_1 \ldots G_n} A = D$.

Vì $\vdash D \sim D$ nên $\vdash \left(\int_{X_1 \ldots X_n}^{G_1 \ldots G_n} A \right) \sim D$, còn nếu $\delta_i = F$ thì $G_i \equiv K$.

Vì $\vdash K \sim K$ nên $\vdash \left(\int_{X_1 \ldots X_n}^{G_1 \ldots G_n} A \right) \sim K$. Vậy trường hợp này là đúng.

2) Giả sử định lý đã đúng với các công thức A_1, A_2. Ta chứng minh nó cũng đúng với các công thức $(\overline{A_1})$, $(A_1 \vee A_2)$, $(A_1 \wedge A_2)$, $(A_1 \rightarrow A_2)$.

Để cho gọn ta ký hiệu X thay cho $(X_1, X_2, \ldots, X_n)$, còn G thay cho $(G_1, G_2, \ldots, G_n)$.

a) Xét $(\overline{A_1})$.

Ta chứng minh khi X_i nhận giá trị δ_i và nếu $\overline{A_1}$ nhận giá trị T thì

$$\vdash \int_X^G \overline{A_1} \sim D$$

Còn nếu $\overline{A_1}$ nhận giá trị F thì $\vdash \int_X^G \overline{A_1} \sim K$.

Xét các trường hợp sau:

Trường hợp 1: Khi $\overline{A_1}$ nhận giá trị T thì A_1 nhận giá trị F.

Theo giả thiết quy nạp với A_1 thì

$$\vdash \int_X^G A_1 \sim K \tag{1}$$

Theo định lý 7 thì từ (1) ta có:

$$\vdash \int_X^G A_l \sim K \tag{2}$$

Theo 4(a) thì

$$\vdash \left(\int_X^G A_l \to \int_X^G \overline{A_l} \right) \tag{3}$$

Từ (2), (3) theo quy tắc dẫn xuất ta có:

$$\vdash \overline{K} \to \int_X^G \overline{A_l} \tag{4}$$

Mặt khác từ (1) theo định lý 7 thì

$$\vdash K \to \int_X^G A_l \tag{5}$$

Lại theo tiên đề 4(a) ta có:

$$\vdash \left(K \to \int_X^G A_l \right) \to \left(\int_X^G \overline{A_l} \to \overline{K} \right) \tag{6}$$

Từ (5), (6) theo quy tắc dẫn xuất ta có:

$$\vdash \int_X^G \overline{A_l} \to \overline{K} \tag{7}$$

Từ (4), (7) và định lý 7 ta có:

$$\vdash \int_X^G \overline{A_l} \sim \overline{K} \tag{8}$$

Vì $\vdash D$ và $\vdash \overline{K}$ nên $\vdash \overline{K} \sim D$ (9)

Từ (8), (9) ta có:

$$\vdash \int_X^G \overline{A_l} \sim D \text{ (do quan hệ bắc cầu).}$$

Trường hợp 2: Khi $\overline{A_l}$ nhận giá trị F thì A_l nhận giá trị T.

Theo giả thiết quy nạp thì:

$$\vdash \int_X^G A_1 \sim D \tag{10}$$

Từ (10) theo định lý 7 ta có:

$$\vdash \int_X^G A_1 \to D \tag{11}$$

và $$\vdash D \to \int_X^G A_1 \tag{12}$$

Theo 4(a) thì

$$\vdash \left(\int_X^G A_1 \to D\right) \to \left(\overline{D} \to \int_X^G \overline{A_1}\right) \tag{13}$$

và $$\vdash \left(D \to \int_X^G A_1\right) \to \left(\int_X^G \overline{A_1} \to \overline{D}\right) \tag{14}$$

Từ (11), (13) theo quy tắc dẫn xuất ta có:

$$\vdash \overline{D} \to \int_X^G \overline{A_1} \tag{15}$$

Từ (12) và (14) theo quy tắc dẫn xuất ta có:

$$\vdash \int_X^G \overline{A_1} \to \overline{D} \tag{16}$$

Theo định lý 7 thì từ (15), (16) ta có:

$$\vdash \int_X^G \overline{A_1} \sim \overline{D} \tag{17}$$

Mặt khác $\vdash \overline{D} \sim K$ (18), nên từ (17), (18) ta suy ra

$$\vdash \int_X^G \overline{A_1} \sim K.$$

b) Xét $A_1 \vee A_2$.

Trường hợp 1: Giả sử $A_1 \vee A_2$ nhận giá trị T thì ít nhất một trong hai công thức nhận giá trị T. Giả sử A_1 nhận giá trị T.

Theo giả thiết quy nạp thì

$$\vdash \int_X^G A_1 \sim D \tag{1}$$

Ta cần chỉ ra $\vdash \int_X^G (A_1 \vee A_2) \sim D$.

Từ (1) theo định lý 7 ta có:

$$\vdash D \to \int_X^G A_1 \tag{2}$$

Theo 3(a) thì

$$\vdash \int_X^G A_1 \to \int_X^G (A_1 \vee A_2) \tag{3}$$

Từ (2), (3) theo quy tắc bắc cầu ta có:

$$\vdash D \to \int_X^G (A_1 \vee A_2) \tag{4}$$

Vì $\vdash D$ nên theo định lý 3 ta có:

$$\vdash \int_X^G (A_1 \vee A_2) \to D \tag{5}$$

Từ (4), (5) theo định lý 7 ta có:

$$\vdash \int_X^G (A_1 \vee A_2) \sim D.$$

Trường hợp 2: Giả sử $A_1 \vee A_2$ nhận giá trị F, điều này chứng tỏ cả A_1 và A_2 đều nhận giá trị F.

Theo giả thiết quy nạp thì

$$\vdash \int_X^G A_1 \sim K \tag{6}$$

và $$\vdash \int_X^G A_2 \sim K \tag{7}$$

Từ (6), (7) theo định lý 7 ta có:

$$\vdash \int_X^G A_1 \to K \qquad (8)$$

và $$\vdash \int_X^G A_2 \to K \qquad (9)$$

Từ (8), (9) theo bổ đề 2 thì

$$\vdash \int_X^G (A_1 \vee A_2) \to K \qquad (10)$$

Vì $\vdash \overline{K}$ nên theo định lý 9 ta có:

$$\vdash K \to \int_X^G (A_1 \vee A_2) \qquad (11)$$

Từ (10) và (11) áp dụng định lý 7 ta có công thức cần chứng minh là:

$$\vdash \int_X^G (A_1 \vee A_2) \sim K.$$

c) Xét $A_1 \wedge A_2$.

Trường hợp 1: Giả sử $A_1 \wedge A_2$ nhận giá trị T, khi đó A_1 và A_2 cùng nhận giá trị T. Theo giả thiết quy nạp thì:

$$\vdash \int_X^G A_1 \sim D \qquad (1)$$

và $$\vdash \int_X^G A_2 \sim D \qquad (2)$$

Từ (1), (2) theo định lý 7 ta có:

$$\vdash D \to \int_X^G A_1 \qquad (3)$$

và $$\vdash D \to \int_X^G A_2 \qquad (4)$$

Từ (3), (4) theo bổ đề 1 ta có:

$$\vdash D \to \int_X^G (A_1 \wedge A_2) \tag{5}$$

Vì $\vdash D$ nên theo định lý 3 ta có:

$$\vdash \int_X^G (A_1 \wedge A_2) \to D \tag{6}$$

Từ (5), (6) theo định lý 7 ta có:

$$\vdash \int_X^G (A_1 \wedge A_2) \sim D.$$

Trường hợp 2: Giả sử $A_1 \wedge A_2$ nhận giá trị F, khi đó ít nhất một trong hai công thức nhận giá trị F. Chẳng hạn A_1 nhận giá trị F.

Theo giả thiết quy nạp thì

$$\vdash \int_X^G A_1 \sim K \tag{7}$$

Từ (7) do định lý 7 ta có:

$$\vdash \int_X^G A_1 \to K \tag{8}$$

Theo 2(a) thì

$$\vdash \int_X^G (A_1 \wedge A_2) \to \int_X^G A_1 \tag{9}$$

Từ (8), (9) theo quy tắc bắc cầu ta có:

$$\vdash \int_X^G (A_1 \wedge A_2) \to K \tag{10}$$

Do $\vdash \overline{K}$ nên theo định lý 9 ta có:

$$\vdash K \to \int_X^G (A_1 \wedge A_2) \tag{11}$$

Từ (10), (11) ta nhận được

$$\vdash \int_X^G (A_1 \wedge A_2) \sim K \text{ (do định lý 7).}$$

d) Xét $A_1 \to A_2$.

Trường hợp 1: $A_1 \to A_2$ nhận giá trị T. Có hai khả năng xảy ra:

d1) A_1 và A_2 nhận giá trị T.

Trường hợp này theo giả thiết quy nạp ta có:

$$\vdash \int_X^G A_1 \sim D \tag{1}$$

và
$$\vdash \int_X^G A_2 \sim D \tag{2}$$

Từ (1), (2) theo định lý 7 ta có:

$$\vdash \int_X^G A_1 \to D \tag{3}$$

và
$$\vdash D \to \int_X^G A_2 \tag{4}$$

Từ (3), (4) theo quy tắc bắc cầu ta có:

$$\vdash \int_X^G A_1 \to \int_X^G A_2 \text{ hay } \vdash \int_X^G (A_1 \to A_2) \tag{5}$$

Vì $\vdash D$ nên theo định lý 3 ta có:

$$\vdash \int_X^G (A_1 \to A_2) \to D \tag{6}$$

Từ (5) theo định lý 3 thì

$$\vdash D \to \int_X^G (A_1 \to A_2) \tag{7}$$

Từ (6) và (7) theo định lý 7 ta có:

$$\vdash \int_X^G (A_1 \to A_2) \sim D.$$

d2) A_1 nhận giá trị F.

Theo giả thiết quy nạp thì

$$\vdash \int_X^G A_1 \sim K \tag{8}$$

Từ (8) theo định lý 7 thì

$$\vdash \int_X^G A_1 \to K \tag{9}$$

Vì $\vdash \overline{K}$ và do định lý 9 ta có:

$$\vdash K \to \int_X^G A_2 \tag{10}$$

Từ (9), (10) theo quy tắc bắc cầu ta có

$$\vdash \int_X^G A_1 \to \int_X^G A_2 \text{ hay } \vdash \int_X^G (A_1 \to A_2) \tag{11}$$

Theo định lý 3 từ (11) ta có:

$$\vdash D \to \int_X^G (A_1 \to A_2) \tag{12}$$

Do $\vdash D$ ta có:

$$\vdash \int_X^G (A_1 \to A_2) \to D \tag{13}$$

Từ (12) và (13) áp dụng định lý 7 ta thu được:

$$\vdash \int_X^G (A_1 \to A_2) \sim D.$$

Trường hợp 2: $A_1 \to A_2$ nhận giá trị F, khi đó ta có A_1 nhận giá trị T còn A_2 nhận giá trị F. Theo giả thiết quy nạp ta có:

$$\vdash \int_X^G A_1 \sim D \tag{14}$$

và $$\vdash \int_X^G A_2 \sim K \tag{15}$$

Theo định lý 7 thì

$$\vdash D \to \int_X^G A_1 \tag{16}$$

và $$\vdash \int_X^G A_2 \to K \tag{17}$$

Áp dụng định lý 4 ta có:

$$\vdash \int_X^G (A_1 \to A_2) \to \int_X^G (A_1 \to A_2) \text{ hay } \vdash \left(\int_X^G A_1 \to \int_X^G A_2\right) \to \left(\int_X^G A_1 \to \int_X^G A_2\right) \tag{18}$$

Hoán vị giả thiết trong (18) ta có:

$$\vdash \int_X^G A_1 \to \left(\left(\int_X^G A_1 \to \int_X^G A_2\right) \to \int_X^G A_2\right) \tag{19}$$

Từ (16) do $\vdash D$ nên ta có:

$$\vdash \int_X^G A_1 \tag{20}$$

Từ (19), (20) áp dụng quy tắc dẫn xuất ta có:

$$\vdash \left(\int_X^G A_1 \to \int_X^G A_2\right) \to \int_X^G A_2 \text{ hay } \vdash \int_X^G (A_1 \to A_2) \to \int_X^G A_2 \tag{21}$$

Từ (17) và (21) áp dụng quy tắc bắc cầu ta có:

$$\vdash \int_X^G (A_1 \to A_2) \to K \tag{22}$$

Mặt khác do $\vdash \overline{K}$ và định lý 9 ta có:

$$\vdash K \to \int_X^G (A_1 \to A_2) \tag{23}$$

Từ (22) và (23) áp dụng định lý 7 ta có:

$$\vdash \int_X^G (A_1 \to A_2) \sim K$$

Định lý được chứng minh.

§5. TÍNH PHI MÂU THUẪN, TÍNH ĐẦY ĐỦ, TÍNH ĐỘC LẬP CỦA HỆ TOÁN MỆNH ĐỀ

5.1. Tính phi mâu thuẫn của hệ toán mệnh đề

Định nghĩa 6: Hệ toán mệnh đề là phi mâu thuẫn nếu không tồn tại một công thức A nào đó sao cho $\vdash A$ và $\vdash \overline{A}$.

Định lý 25: Hệ toán mệnh đề là phi mâu thuẫn.

Chứng minh: Giả sử ngược lại, hệ toán mệnh đề là mâu thuẫn, tức là có tồn tại công thức A mà $\vdash A$ và $\vdash \overline{A}$. Theo định lý 23 thì từ $\vdash A$ và $\vdash \overline{A}$ suy ra $\models A$ và $\models \overline{A}$. Điều này không thể có trong đại số mệnh đề.

5.2. Tính đầy đủ của hệ toán mệnh đề

Định nghĩa 7: Hệ toán mệnh đề là đầy đủ nếu mọi công thức đồng nhất đúng trong đại số mệnh đề là định lý trong hệ toán mệnh đề.

Định lý 26: Hệ toán mệnh đề là đầy đủ.

Chứng minh: Ta cần chứng minh: Nếu $\models A$ thì $\vdash A$.

Giả sử $\models A$ và A chứa các mệnh đề sơ cấp $X_1, X_2, ..., X_n$. Ta lấy D và K là các công thức đúng và sai sao cho D, K không chứa các $X_1, X_2, ..., X_n$.

Theo định nghĩa ký hiệu $\prod\limits_{X_1...X_n} A$ có thể viết

$$\prod_{X_1...X_n} A = \wedge \int_{X_1...X_n}^{G_1...G_n} A$$

trong đó G_i hoặc là D hoặc là K ($i = \overline{1,n}$).

Xét $\int\limits_{X_1...X_n}^{G_1...G_n} A$ với bộ $(\delta_1, \delta_2, ..., \delta_n)$, ở đây $\delta_i = T$ nếu G_i được thay bởi D, còn nếu $\delta_i = F$ thì G_i được thay bởi K.

Giả sử $X_1, X_2, ..., X_n$ nhận bộ giá trị $(\delta_1, \delta_2, ..., \delta_n)$ tương ứng. Vì $\models A$ nên đối với bộ giá trị đó thì A nhận giá trị T. Theo định lý 24 ta có:

$$\vdash \int_{X_1...X_n}^{G_1...G_n} A \sim D \tag{1}$$

Từ (1) theo định lý 7 ta có:

$$\vdash D \rightarrow \int_{X_1...X_n}^{G_1...G_n} A \tag{2}$$

Vì $\vdash D$ nên từ (2) ta có:

$$\vdash \int_{X_1...X_n}^{G_1...G_n} A \tag{3}$$

Từ (3) áp dụng định lý 7 ta lại có:

$$\vdash \wedge \int_{X_1...X_n}^{G_1...G_n} A \quad \text{hay} \vdash \prod_{X_1...X_n} A \tag{4}$$

Mặt khác theo định lý 22 thì

$$\vdash \prod_{X_1...X_n} A \rightarrow A \tag{5}$$

Từ (4) và (5) theo quy tắc dẫn xuất ta có $\vdash A$. Định lý được chứng minh.

5.3. Tính độc lập của hệ toán mệnh đề

Ta ký hiệu các tiên đề của hệ toán mệnh đề qua

$$X = \{1(a), 1(b), 2(a), 2(b), 2(c), 3(a), 3(b), 3(c), 4(a), 4(b), 4(c)\}$$

Định nghĩa 8:

– Tiên đề $i(x) \in X$ ($i \in \{1, 2, 3, 4\}$, $x \in \{a, b, c\}$) được gọi là độc lập với các tiên đề còn lại ($X \setminus \{i(x)\}$) nếu như $i(x)$ không dẫn được từ các tiên đề $X \setminus \{i(x)\}$ theo các quy tắc dẫn xuất.

– Hệ toán mệnh đề gọi là độc lập nếu mọi tiên đề của X là độc lập với các tiên đề còn lại.

Định lý 27: Hệ toán mệnh đề là độc lập.

Chứng minh: Theo định nghĩa trên, để chứng minh định lý 27 ta cần tìm một tính chất $\mathcal{G}$ nào đấy thoả mãn điều kiện sau đây:

– $\mathcal{A} \setminus A$ có tính chất $\mathcal{G}$.

– Nếu công thức B dẫn được từ các tiên đề $\mathcal{A} \setminus A$ thì B có tính chất $\mathcal{G}$.

– Tiên đề A không có tính chất $\mathcal{G}$.

Ở đây: $\mathcal{A}$ là hệ các tiên đề.

Nếu chứng minh được như vậy, thì theo định nghĩa ta có A là tiên đề độc lập với các tiên đề còn lại. Nếu $\mathcal{A}$ gồm các tiên đề độc lập thì hệ tiên đề $\mathcal{A}$ là độc lập.

Để chứng minh định lý trên, ta chứng minh mọi tiên đề của hệ toán mệnh đề là độc lập với các tiên đề còn lại. Cụ thể, hệ tiên đề của ta gồm 4 nhóm: (1), (2), (3) và (4). Trước tiên ta chứng minh tính độc lập cho nhóm (2), (3) và (4) trước.

Lấy các ký hiệu α, β, γ, ... là những giá trị mà công thức có thể nhận được, khi đó ta sẽ định nghĩa các phép toán $\wedge$, $\vee$, $\rightarrow$, $\overline{}$ trên các ký hiệu đó.

Tính chất $\mathcal{G}$ được chọn là: "Công thức A luôn luôn nhận giá trị α".

Việc định nghĩa các phép toán trên các ký hiệu α, β, γ, ... trên là tuỳ ý sao cho tính chất $\mathcal{G}$ thoả mãn các tiên đề $\mathcal{A} \setminus \{A\}$ (trừ tiên đề A).

Thực ra để chứng minh tính độc lập của các nhóm tiên đề (2), (3), (4) ta chỉ cần hai ký hiệu α, β; đặt $\alpha = T$, $\beta = F$.

a) Chứng minh tính độc lập của tiên đề nhóm (2)

Chú ý rằng, các tiên đề trong nhóm (2) chỉ chứa các phép toán $\wedge$, $\rightarrow$; còn các tiên đề trong nhóm (1), (3), (4) chứa các phép toán $\vee$, $\rightarrow$ và $\overline{}$. Vì vậy ta định nghĩa các phép toán $\vee$, $\rightarrow$, $\overline{}$ như trong đại số mệnh đề, còn tính chất $\mathcal{G}$ ta chọn như sau: "Công thức A luôn nhận giá trị α, tức là A luôn luôn nhận giá trị đúng".

Với tính chất $\mathcal{G}$ như trên và với cách định nghĩa các phép toán $\vee$, $\rightarrow$, $\overline{}$ như trong đại số mệnh đề, rõ ràng các tiên đề trong các nhóm (1), (3), (4) đều có tính chất $\mathcal{G}$ bởi vì chúng đều là những công thức đồng nhất đúng. Bây giờ ta chỉ cần xét nhóm tiên đề (2).

* *Tính độc lập của tiên đề 2(a):*

Đối với 2(a) ta định nghĩa phép toán $\wedge$ như sau:

$$A \wedge B = B$$

Với cách định nghĩa trên tiên đề 2(a) không có tính chất $\mathcal{G}$, bởi vì $A \wedge B \rightarrow A \equiv B \rightarrow A \neq \alpha$ nếu ta chọn B đúng, A sai. Còn mọi tiên đề khác đều có tính chất $\mathcal{G}$.

Thật vậy, các tiên đề trong nhóm (1), (3) và (4) có tính chất $\mathcal{G}$ là hiển nhiên. Xét tiên đề 2(b): $A \wedge B \rightarrow B \equiv B \rightarrow B \equiv \alpha$ (có tính chất $\mathcal{G}$).

Xét tiên đề 2(c): $(A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C)))$

$$\equiv (A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow C)) \equiv (A \rightarrow B) \rightarrow \alpha \equiv \alpha$$

Vậy 2(c) có tính chất $\mathcal{G}$.

Để kết thúc chứng minh tính độc lập của tiên đề 2(a), ta cần chứng minh thêm rằng: Nếu A có tính chất $\mathcal{G}$, A → B có tính chất $\mathcal{G}$ thì B có tính chất $\mathcal{G}$.

Vì A, A → B có tính chất $\mathcal{G}$, nghĩa là A = T, A → B = T. Khi đó B = T, bởi vì nếu không thì sẽ tồn tại một khả năng nào đó để B = F, với khả năng đó A đúng mà phép → định nghĩa như trong đại số mệnh đề nên A → B sai. Điều này vô lý bởi vì $A \to B \equiv \alpha$.

Tóm lại ta đã chứng minh được:

– Các tiên đề 2(b), 2(c) và các tiên đề trong các nhóm (3), (4), (1) đều có tính chất $\mathcal{G}$.

– Nếu B dẫn được từ các tiên đề còn lại thì B có tính chất $\mathcal{G}$.

– Tiên đề 2(a) không có tính chất $\mathcal{G}$.

Vậy 2(a) là tiên đề độc lập với các tiên đề còn lại.

* *Chứng minh tính độc lập của 2(b):*

Đối với 2(b) ta định nghĩa phép toán ∧ như sau:

$$A \wedge B = A$$

Với cách định nghĩa trên ta thấy mọi tiên đề không thuộc nhóm (2) đều có tính chất $\mathcal{G}$, bởi vì các tiên đề này chỉ chứa các phép toán ∨, →, — mà ta định nghĩa như trong đại số mệnh đề, nên chúng đều là các công thức đồng nhất đúng.

Xét tiên đề 2(a): $A \wedge B \to A \equiv A \to A \equiv \alpha$ hay tiên đề 2(a) có tính chất $\mathcal{G}$.

Xét tiên đề 2(c): $(A \to B) \to ((A \to C) \to (A \to (B \wedge C)))$

$$\equiv (A \to B) \to ((A \to C) \to (A \to B)) \equiv \alpha$$

Vậy 2(c) cũng có tính chất $\mathcal{G}$.

Tóm lại, mọi tiên đề trong hệ toán mệnh đề (trừ tiên đề 2(b)) đều có tính chất $\mathcal{G}$.

Tiên đề 2(b) không có tính chất $\mathcal{G}$.

Thật vậy: $A \wedge B \to B \equiv A \to B \equiv \beta$ nếu ta chọn $A = \alpha$, $B = \beta$.

Mọi công thức dẫn được từ các tiên đề (trừ tiên đề 2(b)) đều có tính chất $\mathcal{G}$. Chứng minh tương tự như đối với tiên đề 2(a). Vậy tiên đề 2(b) là độc lập đối với các tiên đề còn lại.

* *Chứng minh tính độc lập của tiên đề 2(c):*

Đối với 2(c) ta định nghĩa phép ∧ như sau:

$$A \wedge B = \beta$$

Với cách định nghĩa trên ta thấy mọi tiên đề không thuộc nhóm (2) đều có tính chất $\mathcal{G}$.

Xét tiên đề 2(a): $A \wedge B \rightarrow A \equiv \beta \rightarrow A \equiv \alpha$. Vậy 2(a) có tính chất $\mathcal{G}$.

Xét tiên đề 2(b): $A \wedge B \rightarrow B \equiv \beta \rightarrow B \equiv \alpha$. Vậy 2(b) có tính chất $\mathcal{G}$.

Nói tóm lại, mọi tiên đề trong hệ toán mệnh đề (trừ tiên đề 2(c)) đều có tính chất $\mathcal{G}$.

Tiên đề 2(c) không có tính chất $\mathcal{G}$.

Thật vậy: $(A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C)))$

$$\equiv (A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow \beta)) \equiv \beta$$

nếu ta chọn $A = \alpha$, $B = \alpha$, $C = \alpha$.

Mọi công thức dẫn được từ các tiên đề (trừ tiên đề 2(c)) đều có tính chất $\mathcal{G}$ (chứng minh như các phần trên). Vậy tiên đề 2(c) độc lập đối với các tiên đề còn lại.

b) Chứng minh tính độc lập của các tiên đề nhóm (3)

Đối với nhóm (3), ta định nghĩa các phép toán $\wedge$, $\rightarrow$, — như trong đại số mệnh đề.

** Chứng minh tính độc lập của tiên đề 3(a):*

Ta định nghĩa phép toán $\vee$ như sau:

$$A \vee B = B$$

Với các định nghĩa trên, mọi tiên đề (trừ tiên đề 3(a)) đều có tính chất $\mathcal{G}$. Thật vậy, các tiên đề không thuộc nhóm (3) có tính chất $\mathcal{G}$.

Xét tiên đề 3(b): $B \rightarrow (A \vee B) \equiv B \rightarrow B \equiv \alpha$. Vậy 3(b) có tính chất $\mathcal{G}$.

Xét tiên đề 3(c): $(A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow ((A \vee B) \rightarrow C))$

$$\equiv (A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C)) \equiv \alpha$$

hay 3(c) có tính chất $\mathcal{G}$.

Tiên đề 3(a) không có tính chất $\mathcal{G}$.

Thật vậy, xét tiên đề 3(a): $A \rightarrow (A \vee B) \equiv A \rightarrow B \equiv \beta$ nếu ta chọn $A = \alpha$, $B = \beta$.

Mọi công thức dẫn được từ các tiên đề (trừ tiên đề 3(a)) đều có tính chất $\mathcal{G}$. Vậy tiên đề 3(a) là độc lập đối với các tiên đề còn lại.

** Tính độc lập của tiên đề 3(b):*

Ta định nghĩa phép toán $\vee$ như sau:

$$A \vee B = A.$$

Khi đó mọi tiên đề (trừ tiên đề 3(b)) đều có tính chất $\mathcal{G}$.

Thật vậy các tiên đề không thuộc nhóm (3) đều có tính chất $\mathcal{G}$ là rõ ràng.

Xét tiên đề 3(a): $A \to (A \vee B) \equiv A \to A \equiv \alpha$. Vậy 3(a) có tính chất $\mathcal{G}$.

Xét tiên đề 3(c): $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$

$$\equiv (A \to C) \to ((B \to C) \to (A \to C)) \equiv \alpha$$

hay 3(c) có tính chất $\mathcal{G}$.

Tiên đề 3(b) không có tính chất $\mathcal{G}$. Thật vậy, xét tiên đề 3(b):

$$B \to (A \vee B) \equiv B \to A \equiv \beta \text{ nếu ta chọn } B = \alpha,\ A = \beta.$$

Mọi tiên đề dẫn được từ các tiên đề (trừ tiên đề 3(b)) đều có tính chất $\mathcal{G}$. Tóm lại tiên đề 3(b) là độc lập đối với các tiên đề còn lại.

** Tính độc lập của tiên đề 3(c):*

Ta định nghĩa phép toán $\vee$ như sau:

$$A \vee B = \alpha$$

Khi đó mọi tiên đề (trừ tiên đề 3(c)) đều có tính chất $\mathcal{G}$.

Thật vậy, mọi tiên đề không thuộc nhóm (3) đều có tính chất $\mathcal{G}$.

Xét tiên đề 3(a): $A \to (A \vee B) \equiv A \vee \alpha \equiv \alpha$. Vậy 3(a) có tính chất $\mathcal{G}$.

Xét tiên đề 3(b): $B \to (A \vee B) \equiv B \vee \alpha \equiv \alpha$. Vậy 3(b) có tính chất $\mathcal{G}$.

Tiên đề 3(c) không có tính chất $\mathcal{G}$.

Thật vậy: $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$

$$\equiv (A \to C) \to ((B \to C) \to (\alpha \to C)) \equiv \beta$$

nếu ta chọn $c = \beta$, $A = B = \beta$.

Vậy mọi công thức dẫn được từ các tiên đề (trừ tiên đề 3(c)) đều có tính chất $\mathcal{G}$. Tóm lại tiên đề 3(c) là độc lập đối với các tiên đề còn lại.

c) Chứng minh tính độc lập của các tiên đề nhóm (4)

Các tiên đề trong nhóm (4) chỉ chứa các phép toán $\to$, $\overline{}$, riêng phép toán $\overline{}$ thì các tiên đề khác không chứa.

Vậy ta định nghĩa các phép toán $\wedge$, $\to$, $\overline{}$ như trong đại số mệnh đề.

** Tính độc lập của tiên đề 4(a):*

Đối với tiên đề 4(a) ta định nghĩa phép toán $\overline{}$ như sau:

$$\overline{A} = A.$$

Với định nghĩa trên thì mọi tiên đề (trừ tiên đề 4(a)) đều có tính chất $\mathcal{G}$.

Xét tiên đề 4(b): $A \to \overline{\overline{A}} \equiv A \to \overline{A} \equiv A \to A \equiv \alpha$. Vậy 4(b) có tính chất $\mathcal{G}$.

Xét tiên đề 4(c): $\overline{\overline{A}} \to A \equiv \overline{A} \to A \equiv A \to A \equiv \alpha$. Vậy 4(c) có tính chất $\mathcal{G}$.

Tiên đề 4(a) không có tính chất $\mathcal{G}$.

Thật vậy, $(A \to B) \to (\overline{B} \to \overline{A}) \equiv (A \to B) \to (B \to A) \equiv \beta$ nếu ta chọn $A = \beta$, $B = \alpha$.

Mọi công thức dẫn được từ các tiên đề (trừ tiên đề 4(a)) đều có tính chất $\mathcal{G}$. Vậy tiên đề 4(a) là độc lập đối với các tiên đề còn lại.

* *Tính độc lập của tiên đề 4(b):*

Trước tiên đối với tiên đề này ta định nghĩa phép toán — như sau:

$$\overline{A} = \beta.$$

Rõ ràng mọi tiên đề không thuộc nhóm (4) đều có tính chất $\mathcal{G}$.

Xét tiên đề 4(a): $(A \to B) \to (\overline{B} \to \overline{A}) \equiv (A \to B) \to (\beta \to \beta)$

$$\equiv (A \to B) \to \alpha \equiv \alpha.$$

Vậy tiên đề 4(a) có tính chất $\mathcal{G}$.

Xét tiên đề 4(c): $\overline{\overline{A}} \to A \equiv \beta \to A \equiv \alpha$. Vậy 4(c) có tính chất $\mathcal{G}$.

Riêng tiên đề 4(b) không có tính chất $\mathcal{G}$.

Thật vậy, $A \to \overline{\overline{A}} \equiv A \to \beta \equiv \beta$ nếu ta chọn $A = \alpha$.

Mọi công thức dẫn được từ các tiên đề (trừ tiên đề 4(b)) đều có tính chất $\mathcal{G}$. Vậy tiên đề 4(b) là độc lập đối với các tiên đề còn lại.

Tính độc lập của tiên đề 4(c):

Ta định nghĩa phép toán — như sau:

$$\overline{A} = \alpha.$$

Khi đó mọi tiên đề trừ tiên đề 4(c) đều có tính chất $\mathcal{G}$.

Thật vậy, các tiên đề không thuộc nhóm (4) đều có tính chất $\mathcal{G}$.

Xét tiên đề 4(a):

$$(A \to B) \to (\overline{B} \to \overline{A}) \equiv (A \to B) \to (\alpha \to \alpha) \equiv (A \to B) \to \alpha \equiv \alpha$$

Xét tiên đề 4(b): $A \to \overline{A} \equiv A \to \alpha \equiv \alpha$.

Vậy hai tiên đề 4(a) và 4(b) có tính chất $\mathcal{G}$. Riêng 4(c) không có tính chất $\mathcal{G}$.

Thật vậy: $\overline{\overline{A}} \to A \equiv \alpha \to A \equiv \beta$ nếu ta chọn $A = \beta$.

Vậy tiên đề 4(c) độc lập với các tiên đề còn lại.

d) Chứng minh tính độc lập của các tiên đề trong nhóm (1)

Vì phép toán $\to$ trong nhóm (1) có mặt ở các nhóm khác nên để chứng minh tính độc lập của các tiên đề trong nhóm (1) ta cần định nghĩa các phép toán thoả mãn các điều kiện sau đây:

$$(a)\quad \begin{cases} A \wedge B \equiv B \wedge A,\ A \wedge \alpha \equiv A,\ A \wedge \beta \equiv \beta,\ A \wedge A \equiv A & (1) \\ A \vee B \equiv B \vee A,\ A \vee \alpha \equiv A,\ A \vee \beta \equiv \beta,\ A \vee A \equiv A & (2) \\ \left.\begin{array}{l} A \to A = \alpha,\ A \to \alpha = \alpha,\ \beta \to A = \alpha,\ \alpha \to \beta = \beta, \\ B \to \alpha = \alpha \text{ và } \alpha \to B \neq \alpha \text{ nếu B nhận giá trị } \neq \alpha \end{array}\right\} & (3) \\ \overline{\alpha} = \beta, \overline{\beta} = \alpha & (4) \end{cases}$$

Dễ thấy rằng các định nghĩa như trên là hợp lý bởi vì ta xem α là đúng, β là sai.

Với nhóm trên ta có thể chứng minh rằng, nếu A có tính chất $\mathcal{Q}$, $A \to B$ có tính chất $\mathcal{Q}$, thì B cũng có tính chất $\mathcal{Q}$. Nói cách khác, từ (a) ta chứng minh rằng nếu $A = \alpha$, $A \to B = \alpha$ thì $B = \alpha$.

Thật vậy, nếu $B \neq \alpha$ với bộ giá trị có thể nào đó, thì với bộ đó $A = \alpha$, Như vậy với bộ $A \to B \neq \alpha$ mâu thuẫn với giả thiết là $A \to B$ có tính chất $\mathcal{Q}$.

* *Tính độc lập của tiên đề 1(b):*

Đối với tiên đề này ta lấy bộ giá trị α, β, γ và định nghĩa bổ sung vào nhóm (a) các phép toán sau đây và gọi là nhóm (b):

(b) $\quad \alpha \to \gamma = \gamma,\ \gamma \to \beta = \gamma,\ \overline{\gamma} = \gamma$

Với cách định nghĩa trên mọi tiên đề (trừ tiên đề 1(b)) đều có tính chất $\mathcal{Q}$.

Ở đây ta chỉ thử lại tiên đề 2(a) (các tiên đề khác tương tự):

$$A \wedge B \to A$$

Nếu $A = \alpha$, theo tính giao hoán của (a) ta có:

$$A \wedge B \to A \equiv B \wedge A \to A \equiv B \wedge \alpha \to \alpha \equiv B \to \alpha$$

Nhưng theo (3) của (a) thì $B \to \alpha = \alpha$.

Nếu $A = \beta$, khi đó $A \wedge B \to A \equiv B \wedge A \to A \equiv B \wedge \beta \to \beta \equiv \beta \to \beta = \alpha$.

Nếu $A = \gamma$ thì $A \wedge B \to A$. Căn cứ vào (a) thì $A \wedge B$ hoặc là β hoặc là γ, mà không thể là α do đó ta có:

$$A \wedge B \to A \equiv \beta \to \gamma \text{ hoặc } A \wedge B \to A \equiv \gamma \to \gamma$$

Cả hai trường hợp ta đều có $A \wedge B \to A \equiv \alpha$ (theo nhóm 3 của a).

Vậy 2(a) có tính chất $\mathcal{Q}$.

Riêng 1(b) không có tính chất $\mathcal{Q}$.

Thật vậy, $(A \to (B \to C)) \to ((A \to B) \to (A \to C)) \neq \alpha$ nếu ta chọn $A = B = \gamma$, $C = \beta$. Rõ ràng

$$(\gamma \to (\gamma \to \beta)) \to ((\gamma \to \gamma) \to (\gamma \to \beta)) = (\gamma \to \gamma) \to (\alpha \to \gamma) = \alpha \to \gamma = \gamma$$

* *Tính độc lập của tiên đề 1(a):*

Ta dùng 4 ký hiệu: α, β, γ, δ.

Ta định nghĩa thêm vào nhóm (a), (b) và gọi là nhóm (c) sau đây:

$$\text{(c)} \quad \begin{cases} \alpha \to B = \beta \text{ (trong đó } B = \gamma, \delta, \beta) \\ \alpha \to \beta = \beta, \gamma \to \delta = \beta, \delta \to \beta = \beta, \delta \to \gamma = \alpha \\ \gamma \wedge \delta = \delta, \gamma \vee \delta = \gamma, \overline{\gamma} = \delta, \overline{\delta} = \gamma \end{cases}$$

Với định nghĩa trên mọi tiên đề (trừ tiên đề 1(a)) đều có tính chất $\mathcal{Q}$.

Ở đây ta chỉ chứng minh cho tiên đề 2(a) (các tiên đề khác có thể chứng minh tương tự):

$$A \wedge B \to A$$

Nếu A, B cùng nhận giá trị β thì $A \wedge B \equiv \beta$, do đó

$$A \wedge B \to A \equiv \beta \to \beta \equiv \alpha.$$

Nếu A nhận giá trị α thì

$$A \wedge B \to A \equiv B \wedge A \to A \equiv B \wedge \alpha \to \alpha \equiv B \to \alpha \equiv \alpha$$

Nếu B nhận giá trị α thì $A \wedge \alpha \to A \equiv A \to A \equiv \alpha$.

Nếu $A = B$ thì $A \wedge B \to A \equiv \alpha$.

Nếu $A \neq B$ và nhận các giá trị δ, γ do tính giao hoán chỉ cần xét hai trường hợp $\gamma \wedge \delta \to \gamma$ và $\gamma \wedge \delta \to \delta$ vì $\gamma \wedge \delta = \delta$ nên trường hợp thứ nhất:

$$\gamma \wedge \delta \to \gamma \equiv \delta \to \gamma \equiv \alpha \text{ (do nhóm (c))}$$

Trường hợp thứ hai: $\gamma \wedge \delta \to \delta \equiv \delta \to \delta \equiv \alpha$ (do nhóm (c)).

Tóm lại tiên đề 2(a) có tính chất $\mathcal{Q}$.

Riêng tiên đề 1(a) không có tính chất $\mathcal{Q}$.

Thật vậy, $A \to (B \to A)$ chọn $A = \delta$, $B = \alpha$ khi đó:

$$\delta \to (\alpha \to \delta) \equiv \delta \to \beta = \beta \text{ (do nhóm (c))}$$

Vậy tiên đề 1(a) độc lập với các tiên đề còn lại.

Định lý được chứng minh.

PHẦN III
ĐỒ THỊ VÀ ỨNG DỤNG

Chương 7
LÝ THUYẾT ĐỒ THỊ

§1. ĐỊNH NGHĨA ĐỒ THỊ, BIỂU DIỄN ĐỒ THỊ VÀ MỘT SỐ DẠNG ĐỒ THỊ THƯỜNG GẶP

1.1. Định nghĩa đồ thị

– Giả sử X là tập không rỗng các phần tử nào đó và $U \subseteq X \times X$, ở đây $U = \{(a, b) : a, b \in X\}$ gồm các cặp (a, b) sắp thứ tự, mỗi cặp (a, b) là một *cung*. Trong trường hợp này bộ G = <X, U> được gọi là *đồ thị có hướng*, với X là tập các đỉnh, U là tập các cung.

– Trường hợp $U = \{a, b) : a, b \in X\}$ gồm các cặp (a, b) không sắp thứ tự, mỗi cặp (a, b) gọi là một *cạnh* và trong trường hợp này bộ G = <X, U> được gọi là *đồ thị vô hướng*, với X là tập các đỉnh, U là tập các cạnh.

– Đồ thị G = <X, U> là đồ thị hữu hạn nếu X là tập hữu hạn.

– Hai đỉnh $a, b \in X$ $(a \neq b)$ được gọi là hai đỉnh kề nhau nếu (a, b) là một cạnh (một cung) của đồ thị.

– Hai cạnh (hai cung) được gọi là hai cạnh (hai cung) kề nhau nếu chúng có một đỉnh chung.

– Một cạnh (một cung) có thể bắt đầu và kết thúc tại cùng một đỉnh được gọi là nút hay khuyên.

– Nếu cung đi từ đỉnh a đến đỉnh b thì a gọi là đỉnh đầu, còn b gọi là đỉnh cuối của cung.

– Nếu cặp đỉnh (a, b) có từ hai cạnh (hai cung cùng hướng) thì cặp (a, b) gọi là cạnh (cung) bội.

1.2. Biểu diễn đồ thị

Ở đây đưa ra hai phương pháp biểu diễn đồ thị là: Phương pháp hình học và Phương pháp ma trận kề. Các phương pháp khác bạn đọc có thể xem trong tài liệu tham khảo ở cuối sách.

a) Phương pháp hình học

Biểu diễn đồ thị G = <X, U> dưới dạng hình học theo nguyên tắc sau đây:

– Với mỗi $x \in X$ đặt tương ứng với một điểm trên mặt phẳng hay một điểm trong không gian, đồng thời dùng ngay ký hiệu x để ghi nhãn cho chính điểm đó và gọi nó là đỉnh của đồ thị.

– Nếu $u = (x, y) \in U$ là một cạnh thì từ đỉnh x đến đỉnh y được nối với nhau bởi một đoạn thẳng hay một đoạn cong u mà không đi qua các đỉnh trung gian khác.

– Nếu $u = (x, y) \in U$ là một cung với đỉnh đầu là x và đỉnh cuối là y thì từ đỉnh đầu x đến đỉnh cuối y được nối với nhau bởi một đoạn thẳng hay một đoạn cong có hướng từ x đến y và không đi qua bất kỳ một đỉnh trung gian nào khác.

– Nếu $u = (x, x) \in U$ thì tại đỉnh x có một khuyên. Trường hợp $x \in X$ nhưng $(x, x) \notin U$ hoặc không có $y \in X$ nào để $(x, y) \in U$ thì đỉnh x được gọi là *đỉnh cô lập*.

b) Phương pháp ma trận kề

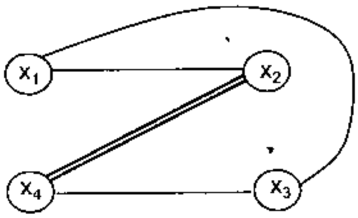
1) Trường hợp G = <X, U> là đồ thị vô hướng với $X = \{x_1, x_2, ..., x_n\}$ thì đồ thị G có thể biểu diễn bằng ma trận vuông cấp n (ký hiệu là $M_G^{n\times n}$), mà phần tử δ_{ij} ở hàng i, cột j được xác định như sau: $\delta_{ij} = d$ nếu cặp đỉnh x_i, x_j có d cạnh nối với nhau; khi cặp đỉnh x_i, x_j không có cạnh nào nối với nhau thì $\delta_{ij} = 0$ (i, j = 1, 2, ..., n). Bằng cách đó ta sẽ nhận được ma trận kề biểu diễn đồ thị đã cho.

Ma trận của đồ thị vô hướng là ma trận đối xứng, tức là các phần tử đối xứng qua đường chéo chính sẽ tương ứng bằng nhau.

2) Trường hợp G = <X, U> là đồ thị có hướng với $X = \{x_1, x_2, ..., x_n\}$ thì đồ thị G có thể biểu diễn bằng ma trận vuông cấp n ($M_G^{n\times n}$) mà phần tử ở hàng i, cột j của nó là δ_{ij} được xác định như sau: đối với mỗi cặp đỉnh (x_i, x_j), từ x_i đến x_j nếu có d cung (cùng hướng) thì $\delta_{ij} = d$; trong trường hợp ngược lại thì $\delta_{ij} = 0$. Nói chung ma trận biểu diễn đồ thị có hướng là ma trận không đối xứng.

Chú ý: Ma trận kề của đồ thị đơn là ma trận lôgic.

Ví dụ 1: Cho đồ thị vô hướng $G_1 = <X_1, U_1>$ như hình vẽ dưới đây:



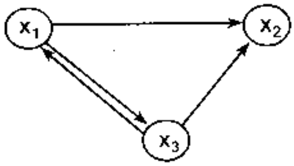
Ta có thể biểu diễn G_1 dưới dạng ma trận đối xứng như sau:

$$M_{G_1}^{4\times4} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix} \end{array} \quad (1)$$

Chú ý: Để cho gọn ta viết $M_{G_1}^{4\times4}$ trong (1) bởi

$$M_{G_1}^{4\times4} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}$$

Ví dụ 2: Cho đồ thị có hướng $G_2 = <X_2, U_2>$ như hình vẽ:



Ma trận kề của nó là

$$M_{G_2}^{3\times3} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

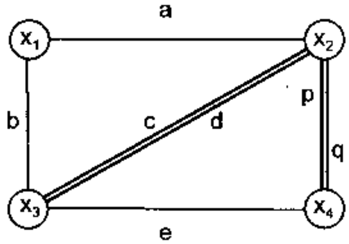
Đối với ma trận kề của đồ thị ta có:

Định lý 1. Nếu $G = <X, U>$ là một đồ thị với $M_G = (a_{ij})$ là ma trận kề, thì số các đường khác nhau đi từ đỉnh x_i đến đỉnh x_j có độ dài l bằng phần tử P_{ij} của ma trận tích

$$\underbrace{M_G.M_G...M_G}_{l \text{ lần}} = M_G^l = (P_{ij})$$

Chứng minh: Quy nạp theo $l \geq 1$ và dựa vào định nghĩa phép nhân ma trận.

Ví dụ 3: Cho đồ thị G = <X, U> được thể hiện ở hình vẽ bên.



Ma trận kề của G là

$$M_G^{4\times4} = M_G = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 2 \\ 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \end{pmatrix}.$$

Tìm tất cả các đường độ dài $l = 2$ đi từ x_1 đến x_4, từ x_3 đến x_4 và từ x_2 vào x_2. Trước hết tính ma trận tích

$$M_G^2 = M_G.M_G = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 2 \\ 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \end{pmatrix} . \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 2 \\ 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \end{pmatrix} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{pmatrix} 2 & 2 & 2 & 3 \\ 2 & 9 & 3 & 2 \\ 2 & 3 & 6 & 4 \\ 3 & 2 & 4 & 5 \end{pmatrix} \end{array}$$

Từ ma trận này ta thấy số đường độ dài 2 từ x_1 đến x_4 là 3, đó là các đường: $x_1ax_2px_4$, $x_1ax_2qx_4$ và $x_1bx_3ex_4$.

Số các đường độ dài 2 từ x_3 đến x_4 là 4, đó là các đường: $x_3cx_2px_4$, $x_3cx_2qx_4$, $x_3dx_2px_4$, $x_3dx_2qx_4$.

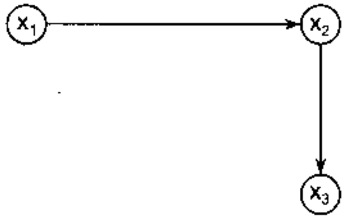
Số các đường độ dài 3 từ x_2 vào x_2 là 9, đó là các đường: $x_2px_4px_2$, $x_2qx_4qx_2$, $x_2px_4qx_2$, $x_2qx_4px_2$, $x_2cx_3cx_2$, $x_2dx_3dx_2$, $x_2cx_3dx_2$, $x_2dx_3cx_2$ và $x_2ax_1ax_2$.

Từ định lý 1 ta có hệ quả sau:

Hệ quả: Trong đồ thị G = <X, U> ta luôn có:

G chứa đường độ dài l từ đỉnh x_i sang x_j khi và chỉ khi $M_G^l \neq \theta$, ở đây θ là ma trận gồm toàn các phần tử 0.

Ví dụ 4: Xét G = <X, U> như hình bên.



Ma trận kề $M_G = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$ và ma trận tích $M_G^2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \neq \theta.$

Trong đồ thị có chứa đường đi độ dài 2 từ x_1 đến x_3.

$$M_G^3 = M_G.M_G.M_G = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

nên đồ thị không có đường đi nào có độ dài lớn hơn 3.

1.3. Một số dạng đồ thị thường gặp

Các dạng đồ thị dưới đây có nhiều ứng dụng trong lĩnh vực tin học và các lĩnh vực khoa học khác. Chính từ các ứng dụng đó mà người ta đưa ra các khái niệm về đồ thị dưới đây:

• ***Đơn đồ thị (đồ thị đơn)***

Đồ thị vô hướng G = <X, U> gọi là *đơn đồ thị* nếu mỗi cặp đỉnh khác nhau a, b ∈ X được nối với nhau bởi không quá một cạnh và mọi đỉnh đều không có khuyên.

• ***Đa đồ thị***

Đồ thị vô hướng G = <X, U> gọi là *đa đồ thị* nếu trong G = <X, U> có tồn tại hai đỉnh khác nhau a, b ∈ X được nối với nhau bởi hai cạnh trở lên và mỗi đỉnh đều không có khuyên.

• ***Giả đồ thị***

Đồ thị vô hướng G = <X, U> gọi là *giả đồ thị* nếu nó là đa đồ thị có khuyên.

• ***Đơn đồ thị có hướng***

Đồ thị có hướng G = <X, U> gọi là *đơn đồ thị có hướng* nếu với mỗi cặp đỉnh a, b ∈ X khác nhau đều được nối với nhau bởi không quá một cung. Đồ thị đơn có hướng có thể có khuyên.

• Đa đồ thị có hướng

Đồ thị có hướng G = <X, U> gọi là *đa đồ thị có hướng* nếu tồn tại một cặp đỉnh a, b $\in$ X khác nhau được nối với nhau bởi hai cung (cùng hướng) trở lên. Đa đồ thị có hướng có thể có khuyên.

• Đồ thị đầy đủ

Đồ thị vô hướng (có hướng) G = <X, U> gọi là *đồ thị đầy đủ* nếu nó là đồ thị đơn (đồ thị đơn có hướng) và với mỗi cặp đỉnh khác nhau a, b $\in$ X có đúng một cạnh (một cung) nối với nhau. Đồ thị đầy đủ n đỉnh ký hiệu là K_n.

• Đồ thị k đầy đủ

Đồ thị vô hướng (có hướng) G = <X, U> gọi là *đồ thị k đầy đủ* nếu với mỗi cặp đỉnh a, b $\in$ X khác nhau có đúng k cạnh (k cung với chiều tùy ý) nối với nhau.

• Đồ thị phân đôi

Đồ thị vô hướng G = <X, U> gọi là *đồ thị phân đôi* nếu X chia thành hai tập $X_1, X_2 \neq \varnothing$ sao cho: $X = X_1 \cup X_2$ và $X_1 \cap X_2 = \varnothing$, đồng thời mỗi cạnh (a, b) $\in$ U thoả mãn a $\in X_1$, b $\in X_2$. Ký hiệu đồ thị phân đôi là $G = \langle X_1 \cup X_2, U\rangle$.

• Đồ thị phân đôi đầy đủ

Đồ thị phân đôi đầy đủ là đồ thị phân đôi $G = \langle X_1 \cup X_2, U\rangle$ với $X = X_1 \cup X_2$ và với mỗi u = (x, y) $\in$ U ta luôn có x $\in X_1$ còn y $\in X_2$, hoặc ngược lại.

• Đồ thị phẳng

Đồ thị vô hướng G = <X, U> gọi là *đồ thị phẳng* nếu G = <X, U> có thể biểu diễn trên mặt phẳng sao cho các cạnh của nó chỉ cắt nhau ở các đỉnh mà thôi. Cách vẽ như vậy gọi là một biểu diễn phẳng của đồ thị G = <X, U>. Ký hiệu G_P là đồ thị biểu diễn phẳng của đồ thị G = <X, U>.

• Đồ thị bù

Đồ thị bù của đơn đồ thị G = <X, U> là đồ thị $\overline{G} = \langle X, \overline{U}\rangle$, ở đây:

$$\overline{U} = \{u = (a, b) : a, b \in X; (a, b) \notin U\}$$

§2. MỘT SỐ THUẬT NGỮ VÀ TÍNH CHẤT CỦA ĐỒ THỊ

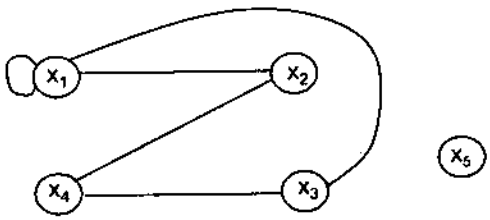
2.1. Một số thuật ngữ của đồ thị

• ***Bậc của đồ thị***

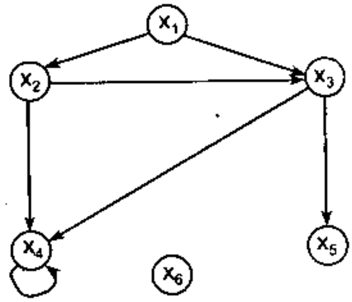
– Cho đồ thị vô hướng $G = \langle X, U\rangle$. Với $x \in X$, ta ký hiệu m(x) là số các cạnh thuộc đỉnh x và m(x) gọi là *bậc của đỉnh x* (nếu đỉnh x có khuyên thì được tính là 2). Nếu m(x) = 0 thì x được gọi là *đỉnh cô lập* và nếu m(x) = 1 thì x được gọi là *đỉnh treo*. Bậc của đồ thị vô hướng $G = \langle X, U\rangle$ được ký hiệu là $m(G) = \sum_{x \in X} m(x)$.

– Cho đồ thị có hướng $G = \langle X, U\rangle$. Với $x \in X$ là một đỉnh của đồ thị, ta ký hiệu $m^+(x)$ là số các cung vào của đỉnh x, còn $m^-(x)$ là số các cung ra khỏi đỉnh x. Khi đó ta gọi $m^+(x)$ là *bậc vào của đỉnh x*, còn $m^-(x)$ là *bậc ra của đỉnh x*. Bậc của đồ thị có hướng $G = \langle X, U\rangle$ được ký hiệu là $m(G) = \sum_{x \in X} m^+(x) + \sum_{x \in X} m^-(x)$. Với $x \in X$ mà $m^+(x) + m^-(x) = 0$ thì ta nói x là *đỉnh cô lập*, còn nếu có $m^+(x) = 0$ và $m^-(x) = 1$ hoặc $m^+(x) = 1$ và $m^-(x) = 0$ thì x được gọi là *đỉnh treo*.

Ví dụ 1: Cho đồ thị $G = \langle X, U\rangle$ như hình vẽ dưới đây và theo định nghĩa ta có: $m(x_1) = 4$, $m(x_2) = 3$, $m(x_3) = 2$, $m(x_4) = 1$, $m(x_5) = 0$. Khi đó x_4 là đỉnh treo, x_5 là đỉnh cô lập và m(G) = 10.



Ví dụ 2: Xét đồ thị có hướng: $G = \langle X, U\rangle$ (hình vẽ) ta có: $m^+(x_1) = 0$, $m^-(x_1) = 2$, $m^+(x_2) = 1$, $m^-(x_2) = 2$, $m^+(x_3) = 2$, $m^-(x_3) = 2$, $m^+(x_4) = 3$, $m^-(x_4) = 1$, $m^+(x_5) = 1$, $m^-(x_5) = 0$, $m^+(x_6) = m^-(x_6) = 0$. Khi đó x_5 là đỉnh treo, x_6 là đỉnh cô lập và bậc của đồ thị G là m(G) = 14.



• ***Đường và chu trình trong đồ thị***

Giả sử G = <X, U> là đồ thị vô hướng (hoặc có hướng). Một *đường đi* trong đồ thị là một dãy $x_{i1}u_{i1}x_{i2}u_{i2} \ldots x_{ij}u_{ij} \ldots x_{ik}u_{ik}x_{ik+1}$, trong đó x_{ij} là các đỉnh còn u_{ij} là các cạnh sao cho với $\forall j \in \{1, 2, \ldots, k\}$ thì đỉnh x_{ij} và đỉnh x_{ij+1} là hai đỉnh kề nhau của cạnh u_{ij}. Đường đi đó xuất phát từ đỉnh x_{i1} và kết thúc tại đỉnh x_{ik+1} (hoặc đi từ đỉnh x_{i1} đến đỉnh x_{ik+1}).

Độ dài của đường bằng số các cạnh (hoặc cung) trong đường đó.

Chu trình trong đồ thị là một đường đi có đỉnh xuất phát và kết thúc trùng nhau.

Đường (chu trình) trong đồ thị G = <X, U> được gọi là *đường (chu trình) đơn* nếu nó đi qua mỗi cạnh đúng một lần.

Đường (chu trình) được gọi là *đường (chu trình) sơ cấp* nếu nó đi qua mỗi đỉnh đúng một lần.

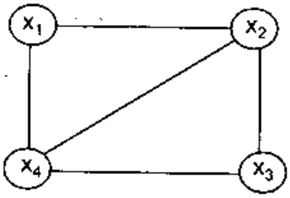
• ***Đồ thị con, đồ thị bộ phận và đồ thị liên thông***

– Đồ thị con, đồ thị bộ phận

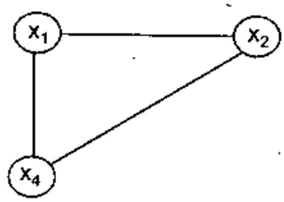
Nếu trong G = <X, U> ta bỏ đi một số đỉnh nào đó và các cạnh liên quan tới các đỉnh đó thì phần còn lại của đồ thị được gọi là *đồ thị con* của đồ thị G đã cho.

Nếu trong đồ thị G ta bỏ đi một số cạnh, giữ nguyên các đỉnh thì phần còn lại của đồ thị được gọi là *đồ thị bộ phận* của đồ thị G.

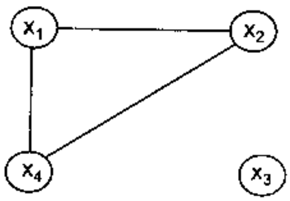
Ví dụ 3: Cho đồ thị G = <X, U> như hình a.



Hình a



Hình b



Hình c

Trong G bỏ đi x_3 và các cạnh kề đỉnh x_3 (hình b) ta được đồ thị G' = <X', U'> là đồ thị con của đồ thị G ban đầu.

Trong G nếu ta giữ lại các đỉnh, bỏ hai cạnh kề đỉnh x_3 ta được đồ thị G" = <X, U"> là đồ thị bộ phận của đồ thị G ban đầu.

– Đồ thị liên thông

Cho đồ thị vô hướng G = <X, U>. Hai đỉnh phân biệt x, y ∈ X được gọi là *liên thông* nếu tồn tại một đường đi nối hai đỉnh x, y với nhau.

Đồ thị vô hướng G được gọi là *liên thông* nếu với hai đỉnh phân biệt bất kỳ trong đồ thị đều là liên thông.

– Đồ thị liên thông mạnh và liên thông yếu

Cho đồ thị có hướng G = <X, U>. Ta nói G là *liên thông mạnh* nếu có đường đi từ x đến y và từ y đến x với mọi đỉnh x, y khác nhau của đồ thị.

Đồ thị có hướng G = <X, U> là *liên thông yếu* nếu giữa hai đỉnh bất kỳ đều có đường đi khi ta không quan tâm tới hướng của các cạnh. Đồ thị mà ta không quan tâm tới hướng gọi là đồ thị nền của đồ thị đã cho.

– Thành phần liên thông

Giả sử $G_1 = \langle X_1, U_1\rangle$ và $G_2 = \langle X_2, U_2\rangle$ là hai đồ thị liên thông với giả thiết $X_1 \cap X_2 = \varnothing$. Khi đó ta nói đồ thị G = <X, U> với $X = X_1 \cup X_2$, $U = U_1 \cup U_2$ có hai *thành phần liên thông* G_1 và G_2.

Lưu ý:

+ Khái niệm thành phần liên thông có thể mở rộng cho một số thành phần là một số nguyên dương k tùy ý.

+ Đồ thị liên thông mạnh cũng là đồ thị liên thông yếu.

2.2. Một số tính chất của đồ thị

Định lý 2: Giả sử G = <X, U> là đồ thị (đa đồ thị) vô hướng (có hướng). Khi đó bậc của đồ thị G gấp hai lần số cạnh của đồ thị, tức là $m(G) = 2|U|$ ($|U|$ là số cạnh của đồ thị).

Chứng minh: Giả sử $x, y \in X$ và $(x, y) \in U$.

Nhận xét:

– Giả sử $x \neq y$, khi đó nếu xoá cạnh (x, y) thì bậc của đồ thị sẽ giảm đi 2; nếu ta xoá tất cả các cạnh có dạng như trên thì đồ thị còn lại chỉ gồm các đỉnh cô lập hoặc các đỉnh có khuyên.

– Tại mỗi đỉnh x có khuyên, nếu ta xoá khuyên thì bậc của đồ thị cũng sẽ giảm đi 2.

– Như vậy, sau khi xoá một cạnh hoặc một khuyên thì bậc của đồ thị giảm đi 2 và sau khi xoá tất cả các cạnh và các khuyên của đồ thị thì bậc của đồ thị còn lại là bằng 0.

Từ nhận xét trên, hiển nhiên ta có đẳng thức

$$m(G) = 2|U|.$$

Đó là điều cần phải chứng minh.

Hệ quả: Đối với đồ thị có hướng G = <X, U> ta luôn có:

$$\sum_{x\in X} m^{+}(x) = \sum_{x\in X} m^{-}(x) = |U|$$

Định lý 3: Giả sử G = <X, U> là một đồ thị (đa đồ thị) có $X = \{x_1, x_2, ..., x_n\}$. Khi đó số các đỉnh bậc lẻ của đồ thị là một số chẵn.

Chứng minh: Không làm mất tính tổng quát, giả sử trong n đỉnh có k đỉnh bậc lẻ là $x_1, x_2, ..., x_k$. Các đỉnh còn lại có bậc chẵn là các đỉnh $x_{k+1}, x_{k+2}, ..., x_n$. Ở đây $m(x_i) = 2m_i + 1$ với $i = 1, ..., k$ và $m(x_j) = 2m_j$ với $j = k + 1, ..., n$; m_i, m_j là số nguyên dương. Theo định lý 2 thì:

$$m(G) = \sum_{i=1}^{k} m(x_i) + \sum_{j=k+1}^{n} m(x_j) = 2|U|$$

Do

$$\sum_{i=1}^{k} m(x_i) = \sum_{i=1}^{k}(2m_i + 1) = 2\sum_{i=1}^{k} m_i + k \text{ và } \sum_{j=k+1}^{n} m(x_j) = 2\sum_{i=1}^{n} m_j$$

Suy ra

$$m(G) = \sum_{i=1}^{k} m(x_i) + \sum_{j=k+1}^{n} m(x_j) = 2\sum_{i=1}^{k} m_i + k + 2\sum_{j=k+1}^{n} m_j$$

$$= 2\left(\sum_{i=1}^{k} m_i + \sum_{j=k+1}^{n} m_j\right) + k = 2|U|$$

Từ đó suy ra k là một số chẵn. Tức là số các đỉnh bậc lẻ trong đồ thị phải là một số chẵn.

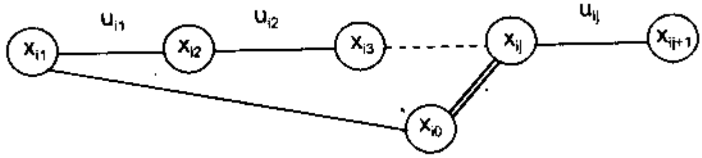
Trong trường hợp G = <X, U> là đồ thị có hướng điều khẳng định trên vẫn đúng (chứng minh tương tự).

Sau đây ta xét một khẳng định liên quan đến chu trình của đồ thị.

Định lý 4: Giả sử G = <X, U> là đồ thị vô hướng với $|X| = n \geq 3$ ($|X|$ là số đỉnh của đồ thị). Nếu trong đồ thị mà mỗi đỉnh $x \in X$ đều có bậc $m(x) \geq 2$ thì đồ thị có chu trình sơ cấp.

Chứng minh: Xét tất cả các đường sơ cấp có thể có trong đồ thị. Rõ ràng số các đường này là hữu hạn. Vì thế trong số các đường sơ cấp đó sẽ tồn tại một đường có độ dài lớn nhất. Giả sử đó là đường ω: $x_{i1}u_{i1}x_{i2}u_{i2} ... x_{ij}u_{ij}x_{ij+1}$.

Dưới dạng hình học thì đường ω có dạng:



Theo giả thiết $m(x_{i1}) \geq 2$ nên tồn tại ít nhất một đỉnh x_{i0} và một cạnh nối đỉnh x_{i1} với x_{i0}. Đỉnh x_{i0} phải trùng với một đỉnh, chẳng hạn là đỉnh x_{ij} trong đường ω, vì nếu không thì đường ω không phải là đường sơ cấp dài nhất, trái với giả thiết ban đầu ω là đường có độ dài lớn nhất. Điều đó chứng tỏ phải tồn tại một chu trình trong đồ thị đang xét. Đây là chu trình sơ cấp, vì các đường ta xét đều là đường sơ cấp.

Định lý được chứng minh.

Định lý 5: Đồ thị G = <X, U> là liên thông khi và chỉ khi nó có một thành phần liên thông.

Chứng minh: Điều kiện đủ là hiển nhiên. Ta chứng minh điều kiện cần bằng cách giả sử ngược lại G = <X, U> có không ít hơn hai thành phần G_1, G_2 liên thông. Lấy x, y ∈ X với x là đỉnh của G_1 còn y là đỉnh của G_2. Vì G_1, G_2 là hai thành phần liên thông của G và x, y không liên thông, vậy G không liên thông, mâu thuẫn với giả thiết.

Định lý 6: Cho G = <X, U> là đồ thị vô hướng với |X| n = ≥ 4. Nếu với mọi x ∈ X mà $m(x) \geq 3$ thì trong G = <X, U> có chu trình sơ cấp độ dài chẵn.

Chứng minh: Dựa vào định nghĩa đường sơ cấp.

Định lý 7: Giữa mọi cặp đỉnh phân biệt của một đồ thị hữu hạn, vô hướng, liên thông luôn luôn có đường đi đơn.

Chứng minh: Dựa vào số đường đi từ hai đỉnh phân biệt bất kỳ là hữu hạn nên có đường với độ dài ngắn nhất. Dùng phản chứng để chứng minh.

Định lý 8: Đồ thị đơn G = <X, U> với $|X| = n \geq 2$ mà tổng bậc của hai đỉnh khác nhau tùy ý đều không nhỏ hơn n là đồ thị liên thông.

Chứng minh: Giả sử a, b ∈ X là hai đỉnh bất kỳ thoả mãn $m(a) + m(b) \geq n$, nhưng a, b không liên thông. Khi đó có hai thành phần liên thông $G_1 = \langle X_1, U_1 \rangle$ có $|X_1| = n_1$ chứa đỉnh a và $G_2 = \langle X_2, U_2 \rangle$ có $|X_2| = n_2$ chứa đỉnh b. Rõ ràng $n_1 + n_2 \leq n$ và $m(a) + m(b) \leq (n_1 - 1) + (n_2 - 1) = n_1 + n_2 - 2 \leq n - 2 < n$. Vô lý với $m(a) + m(b) \geq n$. Vậy G = <X, U> là đồ thị liên thông.

Định lý 9: Nếu đồ thị G = <X, U> có đúng hai đỉnh bậc lẻ thì hai đỉnh này phải liên thông.

Chứng minh: Dùng phản chứng dẫn đến mâu thuẫn với định lý 3.

§3. SỐ ỔN ĐỊNH TRONG, SỐ ỔN ĐỊNH NGOÀI VÀ NHÂN CỦA ĐỒ THỊ

3.1. Số ổn định trong

Cho đồ thị vô hướng $G = <X, U>$ và tập $A \subseteq X$.

1) Tập A gọi là *tập ổn định trong* của đồ thị nếu hai đỉnh bất kỳ x, y trong A là không kề nhau, tức là không có một cạnh nào của đồ thị chứa hai đỉnh x và y.

2) Tập A là *tập ổn định trong cực đại* nếu nó thoả mãn hai điều kiện:

– A là tập ổn định trong.

– Nếu thêm vào A một đỉnh tùy ý không nằm trong A thì A sẽ không còn là tập ổn định trong.

3) Gọi L là họ tất cả các tập ổn định trong của đồ thị $G = <X, U>$. Khi đó ký hiệu $\alpha(G) = \max\{|A| : A \in L\}$ và $\alpha(G)$ được gọi là số *ổn định trong* của đồ thị G.

Nhận xét: Một tập ổn định trong thì mọi tập con của nó cũng là tập ổn định trong củá đồ thị. Với các tập ổn định trong ta chỉ quan tâm đến tập có số phần tử lớn nhất, vì lực lượng của tập đó chính là số ổn định trong của đồ thị.

3.2. Số ổn định ngoài

Cho đồ thị vô hướng $G = <X, U>$ và tập $B \subseteq X$.

1) Tập B được gọi là *tập ổn định ngoài* của đồ thị nếu với mỗi phần tử $y \in X \setminus B$ đều tồn tại $x \in B$ sao cho có cạnh nối giữa x và y. Đôi khi người ta gọi B là *tập thống trị* của đồ thị.

2) Tập B được gọi là *tập ổn định ngoài cực tiểu* nếu nó thoả mãn hai điều kiện sau:

– Tập B là tập ổn định ngoài.

– Nếu bỏ đi một phần tử bất kỳ của B thì B không còn là tập ổn định ngoài.

3) Gọi M là họ tất cả các tập ổn định ngoài của đồ thị $G = <X, U>$. Khi đó ký hiệu $\beta(G) = \min\{|B| : B \in M\}$ và $\beta(G)$ được gọi là *số ổn ngoài* của đồ thị G.

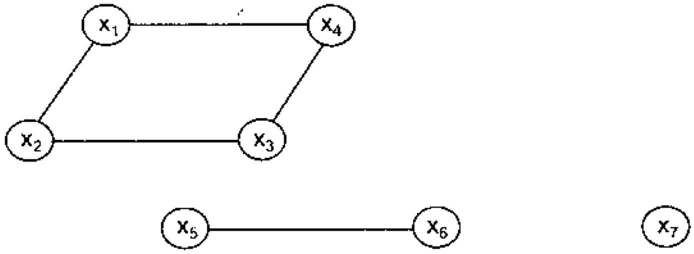
Nhận xét: Với các tập ổn định ngoài, ta thường quan tâm đến tập ổn định ngoài có số phần tử bé nhất vì lực lượng của nó liên quan tới số ổn định ngoài của đồ thị.

3.3. Nhân của đồ thị

Cho đồ thị vô hướng G = <X, U>. Nếu tập $A \subseteq X$ vừa là tập ổn định trong lại vừa là tập ổn định ngoài của đồ thị G thì A được gọi là *nhân* của đồ thị.

Nhận xét: Đối với nhân của đồ thị, ta quan tâm tới nhân có số phần tử bé nhất.

Cho đồ thị G = <X, U> dưới dạng



Các tập ổn định trong của đồ thị là:

$A_1 = \{x_1, x_5, x_7\}$ $A_6 = \{x_2, x_6, x_7\}$

$A_2 = \{x_1, x_6, x_7\}$ $A_7 = \{x_4, x_5, x_7\}$

$A_3 = \{x_3, x_5, x_7\}$ $A_8 = \{x_4, x_6, x_7\}$

$A_4 = \{x_3, x_6, x_7\}$ $A_9 = \{x_2, x_4, x_5, x_7\}$

$A_5 = \{x_2, x_5, x_7\}$ $A_{10} = \{x_2, x_4, x_6, x_7\}$

Trong đó chỉ có hai tập ổn định trong có 4 phần tử. Các tập ổn định trong còn lại có không quá 3 phần tử. Tập A_9 và A_{10} là các tập ổn định trong cực đại vì nếu thêm một đỉnh mới vào các tập đó thì chúng không còn là tập ổn định trong nữa. Số ổn định trong của đồ thị trên là $\alpha(G) = 4$.

Đối với đồ thị trên ta có các tập ổn định ngoài cực tiểu là $B_1 = A_1$, $B_2 = A_2$, $B_3 = A_3$, $B_4 = A_4$. Vì bản thân B_i $(i = \overline{1,4})$ là các tập ổn định ngoài và nếu bớt đi một trong các phần tử của chúng thì tập còn lại không còn là tập ổn định ngoài nữa. Số ổn định ngoài của đồ thị trên là $\beta(G) = 3$. Nhân của đồ thị trên là B_1, B_2, B_3, B_4 và B_i $(i = \overline{1,4})$ là các tập ổn định trong và đồng thời cũng là các tập ổn định ngoài. Số các nhân của đồ thị trên là 4.

3.4. Thuật toán tìm số ổn định ngoài

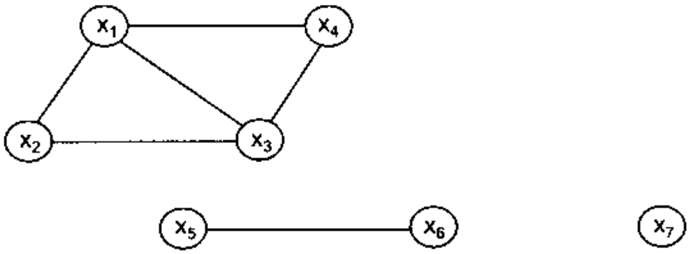
Bài toán: Cho đồ thị G = <X, U>, $X = \{x_1, x_2, ..., x_n\}$. Hãy tìm số ổn định ngoài của đồ thị G (tìm $\beta(G)$). Bài toán này còn được gọi là bài toán đặt vọng gác.

Thuật toán:

Bước 1: Xây dựng ánh xạ $\Delta: X \rightarrow 2^X$ theo cách: Với mỗi $x_i \in X$ thì có $\Delta(x_i) \subseteq X$ và được xác định: $\Delta(x_i) = \{y : y = x_i \text{ hoặc } y \text{ kề } x_i\}$

Bước 2: Giả sử bước 1 xác định được các tập $\Delta(x_1), \Delta(x_2), ..., \Delta(x_n)$. Cần xác định một tập $B \subseteq X$ có số phần tử ít nhất sao cho $\Delta(B) = X$. Khi đó B là tập ổn định ngoài cực tiểu của đồ thị và $\beta(G) = |B|$.

Ví dụ 1. Cho đồ thị G = <X, U>, với $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ dưới đây:



Hãy tìm số ổn định ngoài $\beta(G)$.

Giải:

– *Bước 1:* Xác định ánh xạ $\Delta: X \rightarrow 2^X$

$\Delta(x_1) = \{x_1, x_2, x_3, x_4\}$ $\quad$ $\Delta(x_2) = \{x_1, x_2, x_3\}$

$\Delta(x_3) = \{x_1, x_2, x_3, x_4\}$ $\quad$ $\Delta(x_4) = \{x_1, x_3, x_4\}$

$\Delta(x_5) = \{x_5, x_6\}$ $\quad$ $\Delta(x_6) = \{x_5, x_6\}$

$\Delta(x_7) = \{x_7\}$

– *Bước 2:* Dựa vào bước 1 ta có:

$$\Delta(x_1) \cup \Delta(x_5) \cup \Delta(x_7) = X$$

$$\Delta(x_1) \cup \Delta(x_6) \cup \Delta(x_7) = X$$

$$\Delta(x_3) \cup \Delta(x_5) \cup \Delta(x_7) = X$$

$$\Delta(x_3) \cup \Delta(x_6) \cup \Delta(x_7) = X$$

Mọi trường hợp khác đều không thoả mãn bài toán.

Vậy ta có 4 tập:

$B_1 = \{x_1, x_5, x_7\}$ $\quad$ $B_3 = \{x_3, x_5, x_7\}$

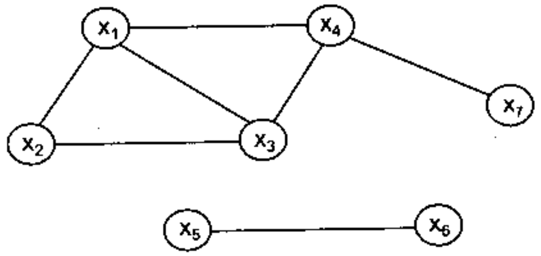
$B_2 = \{x_1, x_6, x_7\}$ $\quad$ $B_4 = \{x_3, x_6, x_7\}$

là các tập ổn định ngoài có số phần tử ít nhất (bằng 3) và thoả mãn

$$\Delta(B_i) = X, \text{ với } i = \overline{1,4}.$$

Ta có: $\beta(G) = \beta(B_i) = |B_i| = 3$.

Ví dụ 2: Cho đồ thị vô hướng $G = <X, U>$, với $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ dưới đây:



Hãy tìm số ổn định ngoài $\beta(G)$.

Giải:

– *Bước 1:* Xác định ánh xạ $\Delta: X \to 2^X$

$\Delta(x_1) = \{x_1, x_2, x_3, x_4\}$ $\quad$ $\Delta(x_2) = \{x_1, x_2, x_3\}$

$\Delta(x_3) = \{x_1, x_2, x_3, x_4\}$ $\quad$ $\Delta(x_4) = \{x_1, x_3, x_4, x_7\}$

$\Delta(x_5) = \{x_5, x_6\}$ $\quad$ $\Delta(x_6) = \{x_5, x_6\}$

$\Delta(x_7) = \{x_4, x_7\}$

– *Bước 2:* Ta có:

$$\Delta(x_4) \cup \Delta(x_5) \cup \Delta(x_2) = X$$

$$\Delta(x_4) \cup \Delta(x_6) \cup \Delta(x_2) = X$$

...

Một trong các tập ổn định ngoài có số phần tử ít nhất là: $B = \{x_2, x_4, x_5\}$.

Vậy $\beta(G) = 3$.

§4. SẮC SỐ CỦA ĐỒ THỊ

Sắc số của đồ thị G là số màu tối thiểu cần dùng để tô màu các đỉnh của đồ thị sao cho hai đỉnh kề nhau phải có màu khác nhau. Sắc số của đồ thị G ký hiệu là s(G).

4.1. Sắc số của đồ thị đầy đủ

Định lý 10: Cho đồ thị G = <X, U>. Nếu đồ thị là đầy đủ thì sắc số của nó bằng số đỉnh của đồ thị, tức là: s(G) = |X|.

Chứng minh: Sử dụng phép quy nạp theo số đỉnh (|X|) của đồ thị.

– Nếu |X| = 1, tức là đồ thị có một đỉnh. Khi đó cần tô một màu là đủ.

– Giả sử định lý đã đúng cho mọi đồ thị đầy đủ có số đỉnh không vượt quá k, tức là với đồ thị đầy đủ G = <X, U> có |X| = k (với $n \le k$) thì có s(G) = n.

– Xét một đồ thị đầy đủ bất kỳ gồm k + 1 đỉnh G' = <X', U'> với $X' = \{x_1, x_2, ..., x_k, x_{k+1}\}$.

Đặt $X = \{x_1, x_2, ..., x_k\}$ khi đó ta có:

$$X' = X \cup \{x_{k+1}\} = \{x_1, x_2, ..., x_k, x_{k+1}\}$$

Nếu trong đồ thị G' bỏ đi đỉnh x_{k+1} và các cạnh nối nó với các đỉnh khác thì ta nhận được đồ thị đầy đủ G = <X, U> có |X| = k. Vì G có k đỉnh nên theo giả thiết quy nạp cần có k màu khác nhau để tô các đỉnh trong G. Vì x_{k+1} là các đỉnh kề với các đỉnh $x_1, x_2, ..., x_k$ nên để tô đỉnh x_{k+1} cần phải dùng màu khác với các màu đã tô ở các đỉnh $x_1, x_2, ..., x_k$. Do đó s(G') = k + 1.

Định lý được chứng minh.

4.2. Sắc số của đồ thị không có chu trình độ dài lẻ

Định lý 11: Giả sử G = <X, U> là đồ thị vô hướng. s(G) = 2 khi và chỉ khi trong G không có chu trình độ dài lẻ.

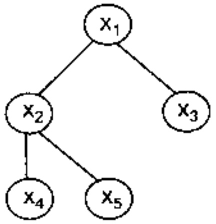
Chứng minh: Trước hết ta có thể giả thiết G là đồ thị liên thông. Vì nếu không ta sẽ chứng minh trên các thành phần liên thông của G.

Điều kiện đủ: Giả sử G không có chu trình độ dài lẻ. Ta chỉ ra s(G) = 2. Thật vậy, tô màu các đỉnh của G theo nguyên tắc sau: Nếu $x \in X$ được tô màu xanh thì các đỉnh kề của x là y, z, ... tô màu đỏ. Tiếp theo các đỉnh kề của y, z, ... tô màu xanh. Cứ như vậy, do số đỉnh hữu hạn và G liên thông nên tất cả các đỉnh trong X sẽ được tô màu hoặc xanh, hoặc đỏ và không có đỉnh nào được tô cả hai màu xanh, đỏ đồng thời (vì nếu có điều đó xảy ra thì sẽ có một chu trình độ dài lẻ đi qua x (trái với giả thiết)). Vậy s(G) = 2.

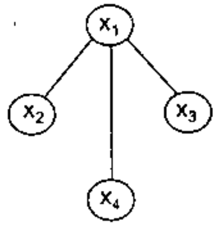
Điều kiện cần: Giả sử s(G) = 2. Dễ thấy rằng, nếu chỉ dùng 2 màu để tô các đỉnh của G thì trong G phải không có chu trình độ dài lẻ (Vì nếu có chu trình độ dài lẻ thì tô màu các đỉnh theo quy tắc trên sẽ có ít nhất một đỉnh được tô đồng thời cả 2 màu).

Định lý được chứng minh.

Ví dụ 1: Cho $G_1 = \langle X_1, U_1 \rangle$ không có chu trình độ dài lẻ, còn $G_2 = \langle X_2, U_2 \rangle$ có chu trình độ dài lẻ (xem hình vẽ).



G₁



G₂

Trong G_1 nếu ta tô x_1 bởi màu xanh, thì x_2 và x_3 tô màu đỏ. Vì x_2 màu đỏ nên x_4, x_5 tô màu xanh, hay $s(G_1) = 2$, ở đây G_1 không có chu trình độ dài lẻ. Xét G_2 có chu trình độ dài lẻ nên không thể tô bằng 2 màu, mà phải dùng 3 màu: xanh, đỏ và vàng.

4.3. Quan hệ giữa sắc số và số ổn định trong

Định lý 12: Giả sử $G = \langle X, U \rangle$ là đồ thị vô hướng với $|X| = n$. Khi đó số ổn định trong $\alpha(G)$ và sắc số $s(G)$ của G thoả mãn bất đẳng thức:

$$\alpha(G).s(G) \geq n.$$

Chứng minh: Đặt s(G) = s, theo định nghĩa của sắc số thì dùng s màu để tô các đỉnh trong X theo nguyên tắc hai đỉnh kề phải tô bằng 2 màu khác nhau.

Cách tô màu như trên lập nên một phân hoạch tương đương trên tập X: $X_1 \cup X_2 \cup \ldots \cup X_s = X$ với $X_i \cap X_j = \varnothing$ $(i \neq j)$, ở đây nếu ta đánh số các màu từ 1, 2, ..., s thì X_i gồm các đỉnh được tô màu i (i = 1, ..., s). Mặt khác, theo định nghĩa số ổn định trong $\alpha(G)$ thì $|X_i| \leq \alpha(G)$. Từ đó ta có đánh giá:

$$|X| = n = |X_1| + |X_2| + \ldots + |X_i| + \ldots + |X_s| \leq s.\alpha(G)$$

Hay $s(G).\alpha(G) \geq n$.

Định lý được chứng minh.

4.4. Sắc số của đồ thị có chu trình

Định lý 13: Một chu trình độ dài lẻ (chẵn) luôn có sắc số bằng 3 (sắc số bằng 2).

Chứng minh: Dùng quy nạp theo độ dài chu trình.

4.5. Sắc số của đồ thị đơn và đồ thị đơn phân đôi

• ***Đối với đồ thị đơn ta có thuật toán tô màu đồ thị như sau:***

Cho G = <X, U> là đồ thị đơn với $X = \{x_1, x_2, ..., x_n\}$. Tìm s(G).

Bước 1: Sắp xếp bậc của các đỉnh theo thứ tự giảm dần. Không mất tính tổng quát, giả sử:

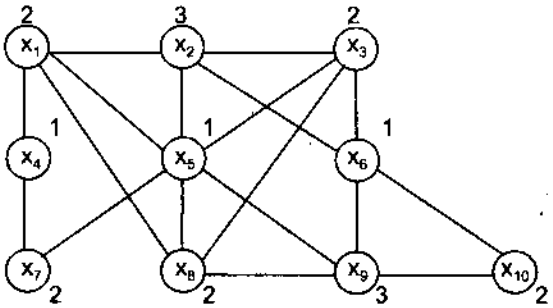
$$m(x_1) \geq m(x_2) \geq ... \geq m(x_n).$$

Bước 2: Gán màu 1 cho x_1 và các đỉnh không liền kề với x_1. Sau đó tiếp gán màu 1 cho các đỉnh không liền kề với các đỉnh đã gán màu 1. Thủ tục gán màu 1 chỉ dừng lại khi không thể gán màu 1 được nữa.

Bước 3: Gán màu 2 cho các đỉnh chưa được gán màu 1 theo thủ tục gán màu ở bước 2.

Thủ tục gán màu sẽ dừng lại khi tất cả các đỉnh đã được gán màu (do đồ thị hữu hạn).

Ví dụ 2: Cho G = <X, U> dưới dạng đồ thị



Ta có: $m(x_5) = 6$;

$$m(x_1) = m(x_2) = m(x_3) = m(x_6) = m(x_8) = m(x_9) = 4;$$

$$m(x_4) = m(x_7) = m(x_{10}) = 2$$

Gán màu 1 cho x_5 và các đỉnh không kề với x_5 là x_4 và x_6.

Gán màu 2 cho x_1 và các đỉnh không kề với x_1 là x_3, x_7, x_{10}.

Gán màu 3 cho x_2 và x_8.

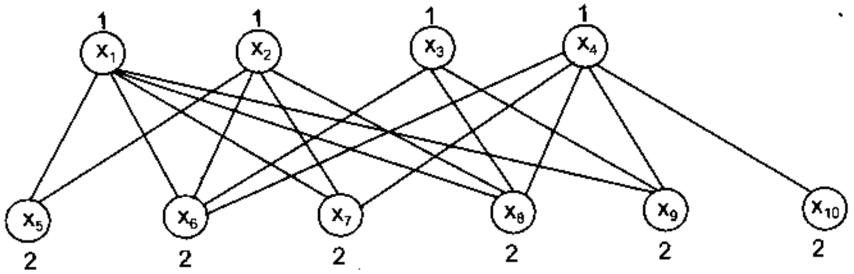
Vậy s(G) = 3.

b) Đối với đồ thị phân đôi G = <X, U>

Định lý 14: s(G) = 2 khi và chỉ khi G là đồ thị phân đôi và liên thông.

(Bạn đọc tự chứng minh)

Ví dụ 3: Cho G = <X, U> có dạng



Ở đây: $X_1 = \{x_1, x_2, x_3, x_4\}$

$X_2 = \{x_5, x_6, x_7, x_8, x_9, x_{10}\}$

Các đỉnh trong X_1 gán màu 1, còn các đỉnh trong X_2 gán màu 2.

Vậy s(G) = 2.

4.6. Bài toán tô màu bản đồ

a) Quan hệ giữa bản đồ và đồ thị .

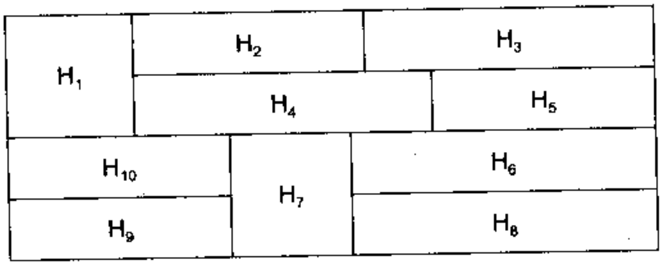
Một bản đồ trên mặt phẳng có thể biểu diễn bởi một đồ thị (dưới đây gọi là đồ thị đối ngẫu) được thiết lập theo các nguyên tắc sau:

– Mỗi miền của bản đồ được biểu diễn bởi một đỉnh.

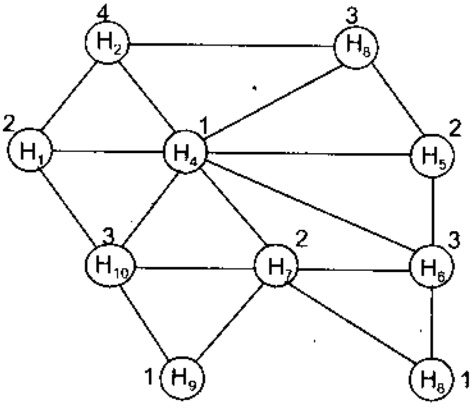
– Hai đỉnh được nối với nhau bởi một cạnh nếu hai miền tương ứng với hai đỉnh đó có biên giới chung trong bản đồ.

– Hai miền chung nhau chỉ một điểm không được xem là có biên giới chung..

Ví dụ 4: Bản đồ gồm 9 huyện là H_i (i = 1, 2, ..., 9) như sau:



Theo nguyên tắc trên, bản đồ có đồ thị đối ngẫu là



b) Tô màu bản đồ

Số màu tối thiểu dùng để tô màu bản đồ sao cho hai miền có chung biên giới phải tô bằng hai màu khác nhau. Sắc số của bản đồ là số màu tối thiểu dùng để tô bản đồ đó.

Sắc số của đồ thị đối ngẫu của bản đồ chính là số màu tối thiểu dùng để tô màu bản đồ theo nguyên tắc hai miền có biên giới chung phải tô bằng hai màu khác nhau.

Vì đồ thị đối ngẫu của một bản đồ là đồ thị phẳng, đối với đồ thị phẳng ta có định lý 4 màu sau đây:

Định lý 15 *(Định lý 4 màu):* Sắc số của một đồ thị phẳng không lớn hơn 4.

(Bạn đọc tự chứng minh)

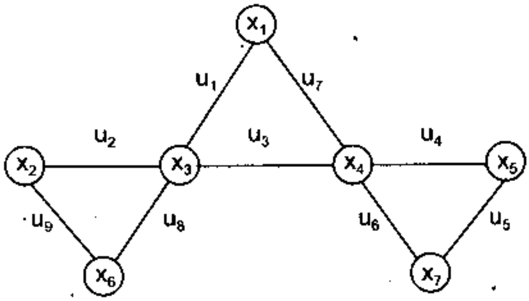
Từ định lý trên ta thấy sắc số của bản đồ không vượt quá 4. Chẳng hạn trong đồ thị đối ngẫu ở ví dụ trên sắc số của nó là 4. Tức là dùng 4 màu có thể tô được đồ thị đã cho.

§5. CHU TRÌNH EULER VÀ ĐƯỜNG EULER

5.1. Chu trình Euler

Giả sử G = <X, U> là đồ thị vô hướng hoặc có hướng. Một chu trình ω trong đồ thị G được gọi là *chu trình Euler* nếu nó đi qua tất cả các cạnh của G và đi qua mỗi cạnh đúng một lần.

Ví dụ 1: Cho đồ thị G = <X, U> có dạng



Chu trình Euler trong đồ thị trên là:

$$z = x_1u_1x_3u_8\ x_6u_9x_2u_2\ x_3u_3x_4u_4\ x_5u_5x_7u_6\ x_4u_7x_1$$

Định lý 16:

a) Đồ thị vô hướng G = <X, U> có chu trình Euler khi và chỉ khi G là liên thông và bậc của tất cả các đỉnh trong đồ thị G là số chẵn.

b) Đồ thị có hướng G = <X, U> có chu trình Euler khi và chỉ khi G liên thông yếu, bậc vào và bậc ra của mỗi đỉnh đều bằng nhau.

Chứng minh: a)

• *Điều kiện cần:* Giả sử đồ thị G = <X, U> có chu trình Euler. Ta cần chứng minh G là đồ thị liên thông và với mỗi $x \in X$ có $m(x) = 2k$ (k là một số nguyên dương nào đó).

Thật vậy, giả sử G = <X, U> không liên thông hay G có ít nhất hai thành phần liên thông $G_1 = <X_1, U_1>$ và $G_2 = <X_2, U_2>$. Trong đó $X_1 \cup X_2 = X$, $U_1 \cup U_2 = U$, giữa các đỉnh trong X_1 và trong X_2 không có cạnh hoặc đường nối với nhau. Giả sử ω là chu trình Euler trong G. Theo định nghĩa của chu trình Euler thì ω là chu trình đi qua tất cả các cạnh trong G, mỗi cạnh đúng một lần. Nếu ω có đỉnh chung với $G_1 = <X_1, U_1>$ thì ω là chu trình nằm trọn trong đồ thị G_1. Điều này mâu thuẫn với định nghĩa của ω. Chứng tỏ đồ thị G = <X, U> là liên thông.

Bây giờ ta chứng minh mỗi đỉnh $x \in X$ trong G đều có bậc chẵn, tức là cần chỉ ra $m(x) = 2k$, với $k \in \{1, 2, ...\}$. Trước hết thấy rằng $k \neq 0$, bởi vì nếu $k = 0$ thì x là điểm cô lập trong G, tức là G không liên thông, trái với điều đã chỉ ra. Giả sử ngược lại, tồn tại một đỉnh $x_i \in X$ mà $m(x_i)$ là một số lẻ, chẳng hạn $m(x_i) = 3$. Đối với x_i có ba cạnh đi vào nó, giả sử đó là các cạnh (x_i, x_k), (x_i, x_j) và $(x_j, x_l) \in U$. Chu trình Euler ω sẽ đi qua ba cạnh đó. Khi đó một trong ba cạnh trên có ít nhất một cạnh mà chu trình Euler ω đi qua hai lần.

Điều đó mâu thuẫn với định nghĩa của chu trình ω. Vậy m(x) là một số chẵn với mọi $x \in X$.

Điều kiện đủ: Giả sử G = <X, U> là đồ thị liên thông và mỗi đỉnh $x \in X$ đều có bậc chẵn: $m(x) = 2k$, $k \in \{1, 2, ...\}$. Ta chứng minh trong đồ thị G có tồn tại một chu trình Euler.

Thật vậy, với giả thiết trên, trước hết ta chứng minh tại mỗi đỉnh của G có tồn tại chu trình đơn (tức là chu trình đi qua các cạnh, mỗi cạnh đúng một lần). Để chứng minh điều đó, ta lưu ý rằng không thể có một đỉnh x mà $m(x) = 2$. Điều đó là đúng bởi vì khi đó tại đỉnh x có khuyên và do đó x cũng là một đỉnh cô lập, trái với giả thiết đồ thị G và liên thông.

Giả sử $x \in X$ là một đỉnh nào đó. Ta chỉ ra có chu trình đơn P qua x. Do $m(x) > 2$ suy ra tồn tại đỉnh x_1 sao cho $x_1 \neq x$ và x kề với x_1. Do $m(x_1) > 2$ suy ra tồn tại đỉnh x_2 sao cho $x_2 \neq x_1$ và x_1 kề với x_2. Tiếp tục như vậy, do $m(x_{i-1}) > 2$ suy ra tồn tại đỉnh x_i sao cho $x_i \neq x_{i-1}$ và x_i kề với x_{i-1}. Khi tới bước i ta có một đường đi từ x tới x_i qua các cạnh, mỗi cạnh đúng một lần. Quá trình trên sẽ dừng lại do tính hữu hạn của đồ thị G. Giả sử số bước hữu hạn đó là j. Điều đó chứng tỏ x và x_j là hai đỉnh kề nhau, tức là có cạnh nối x với x_j. Điều này là đúng vì j là bước cuối cùng.

Như vậy tại đỉnh x có chu trình đơn P đi qua.

Bây giờ ta chứng minh rằng trong đồ thị G = <X, U> có chu trình Euler. Theo chứng minh trên với đỉnh $x \in X$ có chu trình đơn đi qua là P_1 và P_1 là chu trình trong đồ thị G. Hãy "đánh dấu xoá" các cạnh trong P_1. Nếu sau khi "đánh dấu xoá" các cạnh trên đường P_1 tạo ra một số đỉnh cô lập mới thì hãy "đánh dấu loại bỏ" các đỉnh cô lập mới đó. Kết quả thu được sẽ là một đồ thị mới G_1 = <X_1, U_1> là đồ thị con của đồ thị G = <X, U> đã cho. Ta chỉ ra đồ thị G_1 thoả mãn một số tính chất sau:

– Chu trình P_1 trong đồ thị G và G_1 có đỉnh chung, bởi vì G là đồ thị liên thông.

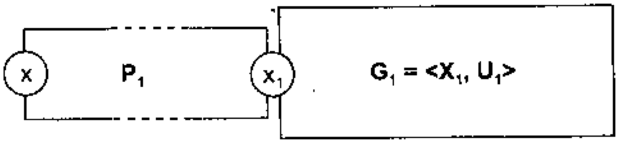
– Đồ thị G_1 gồm các đỉnh $x \in X_1$ có bậc chẵn.

Thật vậy, nếu $x \in X_1$ mà x không thuộc các đỉnh trong P_1 thì m(x) hiển nhiên là một số chẵn.

Còn nếu $x \in X_1$ mà x là đỉnh thuộc P_1 thì sau khi "đánh dấu xoá" hai cạnh của P_1 chứa đỉnh x thì bậc của đỉnh x sẽ giảm đi 2, do đó m(x) cũng là số chẵn.

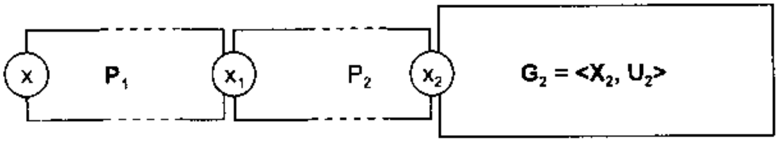
Tóm lại với $\forall x \in X_1$ thì m(x) là một số chẵn.

Ta có thể minh hoạ đồ thị G với chu trình đơn P_1 và đồ thị con G_1 như hình vẽ dưới đây:

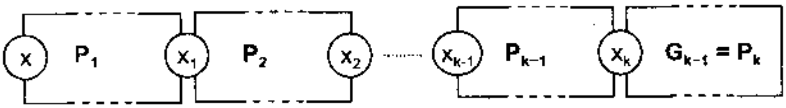


x_1 là đỉnh chung giữa P_1 và G_1. Đối với $G_1 = <X_1, U_1>$ tại đỉnh $x_1 \in X_1$ có tồn tại chu trình đơn P_2 mà cách xây dựng P_2 cũng như đối với P_1.

Trong P_2 bỏ tất cả các cạnh, giữ lại các đỉnh có cạnh hoặc đường nối với các đỉnh khác trong G_1 ta được đồ thị con $G_2 = <X_2, U_2>$ của G_1. Đồ thị này cũng có tính chất như G_1 là liên thông, $\forall x \in X_2$ đều có bậc chẵn, G_2 và P_2 có điểm chung chẳng hạn tại x_2:



Do tính hữu hạn của đồ thị $G = <X, U>$, quá trình xây dựng cá chu trình đơn sẽ dừng lại ở bước thứ k nào đó. Như vậy, trước khi sang bước thứ k ta đã có k − 1 chu trình đơn $P_1, P_2, ..., P_{k-1}$ và đồ thị $G_{k-1} = <X_{k-1}, U_{k-1}>$ là đồ thị con của đồ thị $G_{k-2} = <X_{k-2}, U_{k-2}>$. Đồ thị G_{k-1} là liên thông và mọi đỉnh $x \in X_{k-1}$ có bậc chẵn, đồng thời G_{k-1} và P_{k-1} có điểm chung là x_k. Vì quá trình trên dừng lại sau k bước nên đồ thị G_{k-1} là một chu trình đơn qua x_k và bao gồm hết các cạnh trong đồ thị G_{k-1}. Vì nếu không sẽ dẫn tới mâu thuẫn do k là bước cuối cùng.

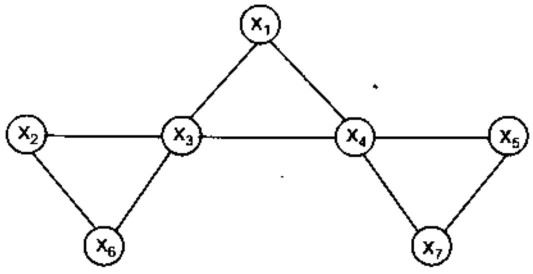


Ghép các chu trình đơn $P_1, P_2, ..., P_k$ tại các đỉnh chung ta được tập các chu trình Euler trong đồ thị $G = <X, U>$. Định lý được chứng minh.

b) Chứng minh dành cho bạn đọc. Cần chú ý định nghĩa bậc vào và bậc ra của đồ thị có hướng và chú ý hệ quả của định lý 2.

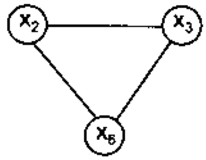
Ví dụ minh hoạ cho điều kiện đủ của định lý.

Xét đồ thị như hình dưới đây:

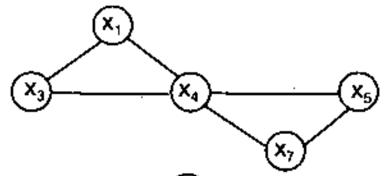


Đây là đồ thị liên thông và có các đỉnh đều bậc chẵn.

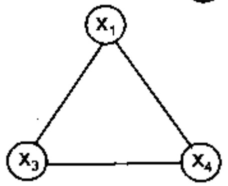
Xuất phát từ đỉnh x_2 ta có chu trình đơn P_1:



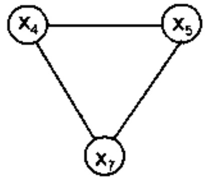
và G_1 là



Từ G_1 ta có P_2 là



và P_3 là



5.2. Thuật toán tìm chu trình Euler

Bài toán: Cho đồ thị G = <X, U>. Hãy tìm chu trình Euler của đồ thị G nếu có.

Thuật toán:

– *Bước 1:* Kiểm tra xem G cớ là đồ thị liên thông hay không. Nếu G là liên thông thì chuyển sang bước 2. Ngược lại thì thuật toán dừng và kết luận rằng đồ thị không có chu trình Euler.

– *Bước 2:* Kiểm tra xem tất cả các đỉnh của G đều có bậc chẵn hay không. Nếu tất cả các đỉnh đều có bậc là chẵn thì chuyển sang bước tiếp theo. Nếu không, hãy dừng lại và kết luận rằng đồ thị đã cho không có chu trình Euler.

– *Bước 3:* Xây dựng các chu trình đơn trong G sao cho tất cả các cạnh của đồ thị đều có các chu trình đơn đi qua và mỗi cạnh chỉ đi qua một lần. Ghép các chu trình đơn như trên tại các đỉnh chung nhau ta được tập các chu trình Euler cần tìm.

Các bước trên được thể hiện trong thủ tục sau:

Procedure Euler (G: đồ thị liên thông với tất cả các đỉnh bậc chẵn);

Chu trình := chu trình trong G bắt đầu tại một đỉnh chọn tùy ý và có cạnh được thêm vào để xây dựng đường đi qua các đỉnh và cuối cùng quay lại đỉnh được chọn ban đầu;

g := G với các cạnh của G sau khi bỏ đi chu trình;

While <g còn cạnh>

Begin

Chu trình con := chu trình trong g bắt đầu tại đỉnh trong g cũng là đỉnh đầu mút của một cạnh thuộc chu trình;

g := g với các cạnh của chu trình con và tất cả các đỉnh cô lập bị loại bỏ;

Chu trình := chu trình với chu trình con được chèn vào tại một đỉnh thích hợp;

End; {chu trình là chu trình Euler}

5.3. Đường Euler

Đường Euler trong đồ thị G = <X, U> là đường đi qua tất cả các cạnh của đồ thị, mỗi cạnh đi qua đúng một lần.

Nhận thấy rằng mỗi chu trình Euler là đường Euler đóng.

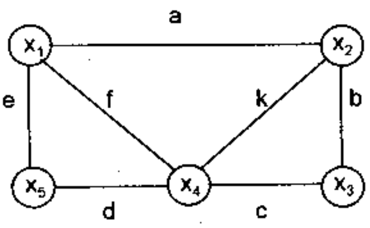
Định lý 17:

a) Cho G = <X, U> là đồ thị vô hướng. Điều kiện cần và đủ để đồ thị có đường Euler nhưng không có chu trình Euler là đồ thị G liên thông và số đỉnh bậc lẻ trong đồ thị là 2.

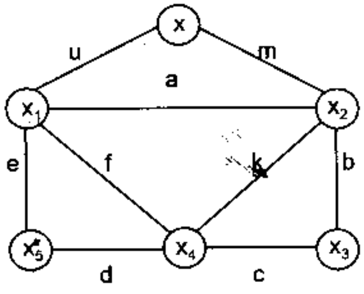
b) Cho G = <X, U> là độ thị có hướng. Điều kiện cần và đủ để G có đường Euler nhưng không có chu trình Euler là G liên thông yếu đồng thời bậc vào và bậc ra của tất cả các đỉnh đều bằng nhau; trừ hai đỉnh, một đỉnh có bậc vào lớn hơn bậc ra 1 đơn vị, còn đỉnh kia có bậc ra lớn hơn bậc vào 1 đơn vị.

Chứng minh (Chứng minh dành cho bạn đọc).

Ví dụ 2: Cho G = <X, U> liên thông và có hai đỉnh bậc lẻ là x_1 và x_2 (hình a).



Hình a



Hình b

Bổ sung thêm đỉnh mới x và hai cạnh u = (x, x_1), v = (x, x_2) vào đồ thị G ta thu được G' = <X', U'> (hình b). Đồ thị G' là liên thông và mọi đỉnh đều có bậc chẵn, do đó theo định lý 16 tồn tại chu trình Euler:

$$xux_1ex_5dx_4cx_3bx_2kx_4fx_1ax_2mx.$$

Bỏ đỉnh x và hai cạnh u, m ta được đường Euler trong G là:

$$x_1ex_5dx_4cx_3bx_2kx_4fx_1ax_2.$$

5.4. Thuật toán tìm đường Euler

Bài toán: Cho đồ thị G = <X, U>. Hãy tìm đường Euler.

Thuật toán:

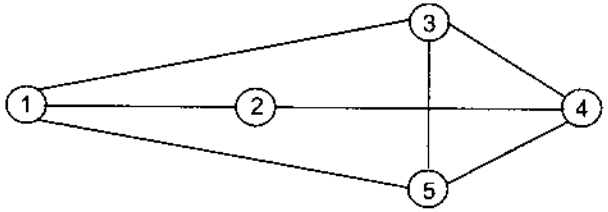
– *Bước 1:* Kiểm tra xem đồ thị G có liên thông hay không. Nếu có thì chuyển sang bước 2. Ngược lại, thì dừng thuật toán và khẳng định rằng không có đường Euler.

– *Bước 2:* Kiểm tra xem mọi đỉnh trong G đều có bậc chẵn hay không. Nếu có chuyển sang bước 4. Nếu không chuyển sang bước 3.

– *Bước 3:* Kiểm tra xem số đỉnh bậc lẻ có bằng 2 hay không. Nếu có chuyển sang bước 4. Nếu không thì dừng lại và kết luận không có đường Euler.

– *Bước 4:* Xây dựng đường Euler trong G.

Ví dụ 3: Đồ thị sau đây là liên thông, nhưng số đỉnh bậc lẻ là 4 do đó theo định lý 17 thì nó không có đường Euler.



§6. CHU TRÌNH HAMILTON VÀ ĐƯỜNG HAMILTON

6.1. Chu trình Hamilton

Giả sử G = <X, U> là đồ thị vô hướng. *Chu trình Hamilton* là chu trình đi qua tất cả các đỉnh của đồ thị, mỗi đỉnh đúng một lần.

Đối với chu trình Hamilton ta có định lý sau đây:

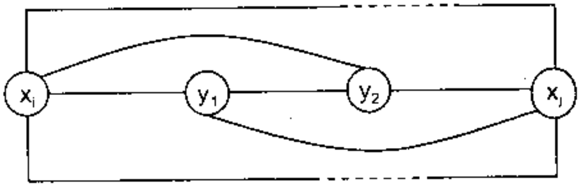
Định lý 18: Nếu đơn đồ thị liên thông G = <X, U> có bậc ở mỗi đỉnh không nhỏ hơn nửa số đỉnh của đồ thị (tức là với $\forall x \in X$ ta luôn có $m(x) \geq |X|/2$) thì trong đồ thị có tồn tại chư trình Hamilton, ở đây giả thiết $|X| \geq 3$.

Chứng minh: Theo giả thiết có $m(x) \geq |X|/2$ với $\forall x \in X$. Dùng phương pháp phản chứng. Giả sử trong G = <X, U> không tồn tại chu trình Hamilton. Bổ sung thêm các đỉnh mới vào đồ thị G và từ các đỉnh mới đều có cạnh nối tới tất cả các đỉnh $x \in X$ trong đồ thị G cho tới khi nào đồ thị mới G' = <U', X'> có chu trình Hamilton thì thôi.

Giả sử tối thiểu các đỉnh mới cần bổ sung vào đồ thị G = <X, U> để được đồ thị $G_1 = <X_1, U_1>$ có chu trình Hamilton là p đỉnh. Khi đó $|X_1| = |X| + p$ (*).

a) Đỉnh mới phải đứng xen kẽ hai đỉnh cũ trong chu trình Hamilton.

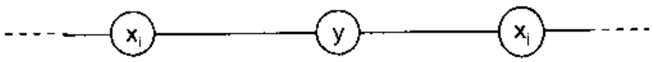
Thật vậy, nếu hai đỉnh mới y_1, y_2 là kề nhau trong chu trình Hamilton (xem hình vẽ)



với $x_i, x_j \in X$ còn $y_1, y_2 \in X_1 \setminus X$

Do y_1, y_2 đều có cạnh nối với đỉnh cũ nên ta có thể bỏ đi một trong hai đỉnh y_1, y_2 ta vẫn được chu trình Hamilton. Điều này trái với giả thiết rằng số đỉnh bổ sung thêm là tối thiểu. Vậy đỉnh mới đứng xen kẽ hai đỉnh cũ.

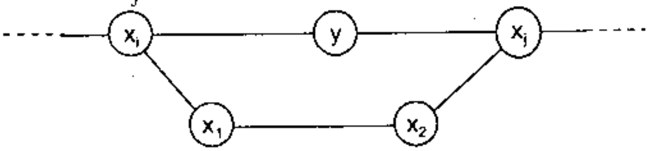
b) Từ khẳng định a) ta thấy chu trình Hamilton có dạng



với $x_i, x_j \in X$ còn $y \in X_1 \setminus X$

Trong trường hợp này ta cần chỉ ra số các đỉnh kề của đỉnh x_i sẽ nhỏ hơn hoặc bằng số các đỉnh không kề của đỉnh x_j.

Thật vậy, giả sử x_1 là đỉnh kề của x_i và x_2 là đỉnh kề của x_1. Nếu x_2 là đỉnh kề của đỉnh x_j thì chu trình Hamilton có dạng



Trong trường hợp này ta có thể bỏ đỉnh y và thay $x_i y x_j$ bởi $x_i x_1 x_2 x_j$. Điều đó trái với giả thiết rằng số đỉnh bổ sung là tối thiểu. Vậy x_2 không kề với x_j. Như vậy ta đã chỉ ra: Với đỉnh x_1 kề của đỉnh x_i thì tương ứng với đỉnh x_2 không kề với x_j. Khẳng định b) được chứng minh.

Dựa vào khẳng định b) ta có các bất đẳng thức sau:

$$|X_1| = |X| + p \geq s_1 + s_2. \qquad (1)$$

trong đó s_1 là tổng số đỉnh kề của x_j và số đỉnh không kề của x_j; s_2 là tổng số đỉnh kề của x_j và số đỉnh kề của x_i. Ta có số đỉnh kề của đỉnh $x_j = m(x_j) \geq |X|/2 + p$ và số đỉnh kề của đỉnh $x_i = m(x_i) \geq |X|/2 + p(2)$. Từ (1) và (2) ta có bất đẳng thức sau:

$$|X_1| = |X| + p \geq (|X|/2 + p) + (|X|/2 + p) = |X| + 2p$$

Kết hợp với (*) suy ra p = 0. Điều đó chứng tỏ không cần bổ sung đỉnh mới nào, đồ thị G = <X, U> vẫn có chu trình Hamilton.

Định lý được chứng minh.

6.2. Đường Hamilton

Đường Hamilton trong đồ thị G = <X, U> là đường đi qua tất cả các đỉnh, mỗi đỉnh đúng một lần.

Định lý 19: Giả sử G = <X, U> là đồ thị có hướng và đầy đủ. Khi đó trong đồ thị luôn luôn tồn tại đường Hamilton.

Chứng minh: Giả sử $\omega = x_{i1}x_{i2} \ldots x_{ik}x_{ik+1} \ldots x_{im-1}x_{im}$ là một đường sơ cấp bất kỳ trong đồ thị. Để đơn giản, ta chỉ viết các đỉnh mà không có các cạnh xen kẽ như định nghĩa.

Nếu trong đường sơ cấp ω mà tất cả các đỉnh trong X đều có mặt thì chính ω là đường Hamilton. Còn nếu có những đỉnh trong X nhưng chưa có mặt trong ω thì ta có thể bổ sung hết những đỉnh đó vào đường ω sơ cấp để nó trở thành đường Hamilton theo nguyên tắc sau đây:

Giả sử x ∈ X mà x không nằm trên đường sơ cấp ω. Các trường hợp sau đây xảy ra do tính đầy đủ của đồ thị G.

1. Nếu x có cung tới x_{i1} thì bổ sung x vào đầu ω và nó có dạng:

$$x x_{i1}x_{i2} \ldots x_{ik}x_{ik+1} \ldots x_{im-1}x_{im}$$

2. Nếu từ x_{ik} có cung tới x và từ x có cung tới x_{ik+1} thì ta bổ sung x vào giữa hai đỉnh x_{ik} và x_{ik+1}.

Khi đó ω có dạng:

$$x_{i1}x_{i2} \dots x_{ik}\, xx_{jk+1} \dots x_{im-1}x_{im}$$

3) Nếu từ x_{ik} và x_{ik+1} có cung đi từ x và từ x lại có cung đi tới x_{ik+2} thì ta bổ sung x vào giữa hai đỉnh x_{ik+1} và x_{ik+2} ta được ω có dạng:

$$x_{i1}x_{i2} \dots x_{ik}x_{ik+1}xx_{ik+2} \dots x_{im-1}x_{im}$$

4) Nếu với $\forall k \in [1, m-1]$ mà từ x_{ik} và x_{ik+1} có cung sang x thì ta bổ sung x vào cuối, khi đó ω có dạng:

$$x_{i1}x_{i2} \dots x_{ik}x_{ik+1} \dots x_{im-1}x_{im}x$$

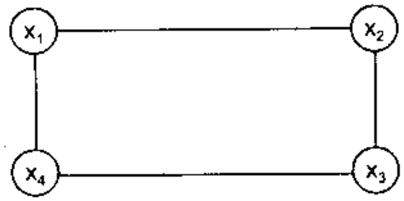
Bằng cách trên ta có thể bổ sung vào ω các đỉnh trong đồ thị mà chưa có mặt trong ω để nó trở thành đường Hamilton.

Định lý được chứng minh.

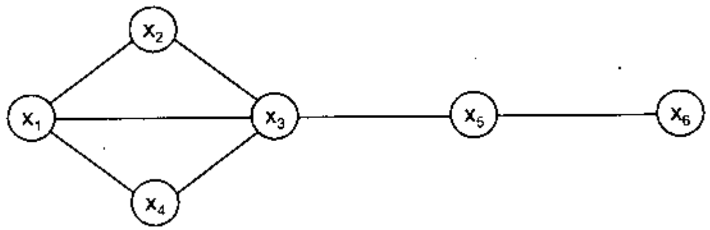
Nhận xét: Trong đồ thị có hướng đầy đủ, nếu ta thay cung bởi cạnh thì ta được đồ thị đầy đủ vô hướng.

Hệ quả: Giả sử G = <X, U> là đồ thị đầy đủ với $|X| \geq 3$. Khi đó trong đồ thị G luôn luôn tồn tại chu trình Hamilton. Điều ngược lại nói chung không đúng.

Ví dụ 1: Đồ thị dưới đây có chu trình Hamilton nhưng nó không phải là đồ thị đầy đủ.



Ví dụ 2: Đồ thị dưới đây không có đường Hamilton.



§7. ĐƯỜNG ĐI NGẮN NHẤT TRONG ĐỒ THỊ

7.1. Đường đi ngắn nhất trong đồ thị không trọng số

Đồ thị không có *trọng số* là đồ thị trên các cạnh không gán trọng số.

Bài toán: Cho đồ thị không có trọng số $G = <X, U>$ và hai đỉnh a, $b \in X$. Tìm đường đi ngắn nhất từ đỉnh a đến đỉnh b, tức là đường từ a đến b có số các cạnh (cung) là ít nhất.

Thuật toán:

Bước 1: Ghi chỉ số ở các đỉnh bằng quy nạp theo số bước

– Đỉnh a ghi nhãn 0: $A(0) = \{a\}$.

– Các đỉnh có cạnh đi từ đỉnh a đến ghi số 1.

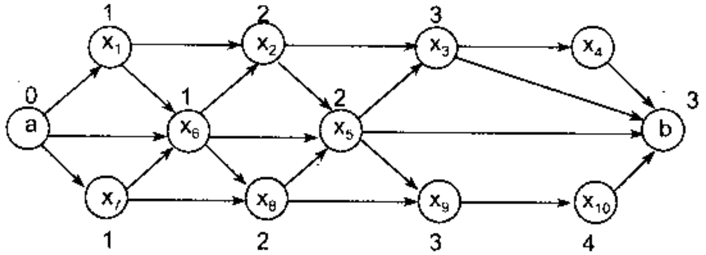
– Giả sử đã ghi nhãn tới bước thứ i: A(0), A(1), ..., A(i). Khi đó tập nhãn thứ i + 1 xác định như sau: $A(i + 1) = \{x : x \in X, x \notin A(k)\ (0 \le k \le i)$ và $\exists y \in A(i)$ sao cho y có cạnh (cung) tới x$\}$, ở đây bước 1 chỉ dừng lại khi đã xác định được tập A(m) các đỉnh, trong đó có đỉnh b được gán nhãn m (do đồ thị hữu hạn).

Bước 2. Tìm đường đi ngắn nhất từ a đến b. Bước này xuất phát từ đỉnh b đi ngược về đỉnh a theo các nguyên tắc sau:

– Tìm tất cả các cạnh (cung) tới b được gán nhãn m – 1, tức $A(m - 1) = \{x_{i_1}, x_{i_2}, ..., x_{i_k}\}$. Với mỗi $x \in A(m - 1)$ tìm tất cả các đỉnh có cạnh (cung) tới x được ghi nhãn m – 2. Thủ tục này sau một số bước sẽ gặp đỉnh có nhãn 0, đó chính là đỉnh a.

– Tất cả các đường xác định được trong bước 1 và 2 là các đường đi ngắn nhất từ a đến b có độ dài m.

Ví dụ 1: Cho đồ thị có hướng như hình dưới đây:



a) Tìm đường đi ngắn nhất từ a đến b.

Bước 1: Đỉnh a được đánh số 0 và có A(0) = {a}

$$A(1) = \{x_1, x_6, x_7\}$$

$$A(2) = \{x_2, x_5, x_8\}$$

$$A(3) = \{x_3, b, x_9\}$$

Bước 2: Từ bước 1 ta có $b \in A(3)$ nên từ a đến b là đường đi ngắn nhất có 3 cung. Tiếp theo ta xác định tất cả các đường đi ngắn nhất có độ dài 3:

– Đỉnh có cung tới b được đánh số 2 là x_5.

– Đỉnh có cung tới x_5 được đánh số 1 là x_6.

– Đỉnh có cung tới x_6 được đánh số 0 là a.

Đường cần tìm là ax_6x_5b.

b) Tìm đường đi ngắn nhất đi từ x_8 đến x_{10}

Bước 1: Đánh số các đỉnh

$$A(0) = \{x_8\}$$

$$A(1) = \{x_5, x_9\}$$

$$A(2) = \{x_3, b, x_{10}\}$$

Bước 2: Từ bước 1 ta có $x_{10} \in A(2)$, chứng tỏ đường đi từ đỉnh x_8 đến đỉnh x_{10} có độ dài bằng 2 là đường đi ngắn nhất. Ta xác định đường đó:

– Đỉnh có cung tới x_{10} được đánh số 1 là x_9.

– Đỉnh có cung tới x_9 được đánh số 0 là x_8.

Vậy đường cần tìm là $x_8x_9x_{10}$.

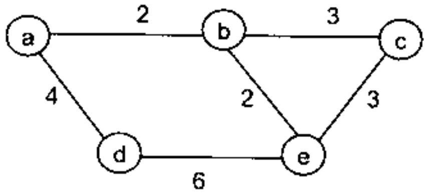
7.2. Đường đi ngắn nhất trong đồ thị có trọng số

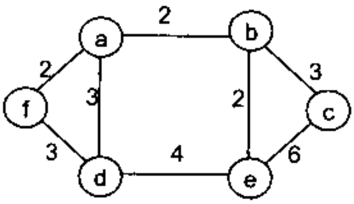
Thuật toán tìm đường đi ngắn nhất trong đồ thị có trọng số được ứng dụng rất rộng rãi trong lĩnh vực CNTT cũng như các bài toán ứng dụng phân luồng trong thực tế, đặc biệt là các bài toán định tuyến trong mạng máy tính (tìm đường đi ngắn nhất cho gói tin giữa hai máy tính). Nếu chúng ta coi tôpô một mạng máy tính (Computer Networks) hay một mạng nội bộ nào đó (Local Networks) là một đồ thị có trọng số. Các Routers, Workstations, ... ở đây đóng vai trò như các đỉnh trong đồ thị và khoảng cách (hay thời gian truyền trễ – Tranferdelay time) giữa chúng là trọng số của đồ thị... Khi đó, thuật toán tìm đường đi ngắn nhất trong đồ thị có trọng số sẽ giúp các máy tính có thể tìm được một đường đi ngắn nhất hay đường đi có độ trễ nhỏ nhất để truyền và trao đổi thông tin cho nhau...

a) Đồ thị có trọng số

Định nghĩa: Cho đồ thị hữu hạn $G = \{X, U\}$. Với mỗi cạnh $u \in U$ ta đặt tương ứng với một số thực $l(u)$ và gọi là *trọng số* của u. Đồ thị có các cạnh có trọng số như trên được gọi là *đồ thị có trọng số*.

Ví dụ 2: Hai đồ thị dưới đây là hai đồ thị có trọng số.





Ta ký hiệu D(a, b) là tập tất cả các đường nối đỉnh a với đỉnh b trong đồ thị $G = <X, U>$, gọi α là một đường nào đó trong G và giả sử

$$\alpha = x_{i1}u_{i1}x_{i2}u_{i2} \ldots x_{in-1}u_{in-1}x_{in},\ x_{ij} \in X,\ u_{ij} \in U\ (j = 1, 2, \ldots, n).$$

Khi đó ta ký hiệu $l(\alpha) = \sum_{j=1}^{n-1} l(u_{ij})$ và gọi là trọng số của đường α.

Bài toán: Cho đồ thị đơn liên thông có trọng số $G = <X, U>$ và $a, b \in X$ là hai đỉnh trong đồ thị. Tìm các đường α từ đỉnh a đến đỉnh b trong đồ thị $G = <X, U>$ có trọng số bé nhất, tức α phải thoả mãn

$$l(\alpha) = \min\{l(\beta) : \beta \in D(a, b)\}.$$

Đối với bài toán này tồn tại nhiều thuật toán khác nhau cho phép tìm ra lời giải. Có thể kể đến các thuật toán như:

– *Thuật toán Ford – Bellman:* Tìm đường đi ngắn nhất từ một đỉnh tới tất cả các đỉnh còn lại của đồ thị, áp dụng cho trường hợp trọng số tùy ý nhưng trong đồ thị không có chu trình âm (*chu trình âm* là chu trình có tổng trọng số các cạnh trong chu trình là một số âm).

– *Thuật toán Dijkstra:* Áp dụng cho trường hợp trọng số trên các cung là không âm. Thuật toán có độ phức tạp cỡ $O(n^2)$, với n là số đỉnh của đồ thị.

– *Thuật toán Floyd:* Cho phép ta tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh trong đồ thị với độ phức tạp tính toán cỡ $O(n^3)$.

Dưới đây chúng tôi trình bày các bước làm việc của thuật toán Dijkstra.

b) Thuật toán Dijkstra

Thuật toán Dijkstra áp dụng trong trường hợp trọng số không âm sẽ hữu hiệu hơn rất nhiều so với thuật toán Ford – Bellman. Thuật toán được xây dựng trên cơ sở gán cho các đỉnh các nhãn tạm thời. Nhãn của mỗi đỉnh cho biết cận trên của độ dài đường đi từ đỉnh a (đỉnh xuất phát) đến nó. Các nhãn này sẽ được biến đổi theo một thủ tục lặp, mà ở mỗi bước lặp có một nhãn tạm thời trở thành nhãn cố định. Nếu ở một nhãn của đỉnh nào trở thành nhãn cố định thì nó sẽ cho ta độ dài của đường đi ngắn nhất từ đỉnh a đến nó. Thuật toán có độ phức tạp cỡ $O(n^2)$. Thuật toán được mô tả cụ thể như sau:

• *Bước 1:* Đánh trọng số các đỉnh.

– Trọng số của đỉnh xuất phát a là $\sigma(a) = 0$.

– Tại các đỉnh còn lại ta ghi một trọng số dương σ đủ lớn sao cho nó lớn hơn trọng số của các đỉnh từ a tới.

• *Bước 2:* Thực hiện việc giảm trọng số các đỉnh.

Giả sử tại đỉnh x được ghi trọng số σ(x). Nếu tồn tại đỉnh y có trọng số σ(y) từ y sang x mà $\sigma(x) > \sigma(y) + l(y, x)$ thì ta thay trọng số của σ(x) bởi trọng số $\sigma'(x) = \sigma(y) + l(y, x)$. Trong trường hợp ngược lại thì giữ nguyên trọng số σ(x). Quá trình thực hiện cho tới khi trọng số của tất cả các đỉnh trong $G = \langle X, U \rangle$ đạt cực tiểu, tức là $\forall x \in X$ không tồn tại $y \in X$ kề với x mà $\sigma(x) > \sigma(y) + l(y, x)$.

• *Bước 3:* Xác định đường đi từ a đến b có trọng số bé nhất.

– Từ bước 2 ta xác định được trọng số của đỉnh b. Xuất phát từ b đi về đỉnh kề với b, chẳng hạn đó là đỉnh x_{in} có tính chất $\sigma(b) = \sigma(x_{in}) + l(x_{in}, b)$. Nếu không có đỉnh kề x_{in} như vậy thì ta đi về đỉnh kề với b có trọng số cạnh (cung) từ đỉnh đó về b là bé nhất.

– Từ đỉnh x_{in} ta đi ngược về đỉnh x_{in-1} có tính chất

$$\sigma(x_{in}) = \sigma(x_{in-1}) + l(x_{in-1}, x_{in}),$$

nếu không đi về đỉnh kề với x_{in} mà trọng số cạnh (cung) giữa chúng là bé nhất.

Bằng cách đó ta sẽ đi về đỉnh x_{i1} mà đỉnh kề là a sao cho

$$\sigma(x_{i1}) = \sigma(a) + l(a, x_{i1}) = l(a, x_{i1}), \text{ với } \sigma(a) = 0.$$

Đường $\alpha = ax_{i1}x_{i2} \dots x_{in-1}x_{in}b$ là đường đi từ a đến b có trọng số bé nhất trong số tất cả các đường đi từ a đến b.

Thật vậy,

$$\begin{aligned} l(\alpha) &= l(a, x_{i1}) + l(x_{i1}, x_{i2}) + \dots + l(x_{in-1}, x_{in}) + l(x_{in}, b) \\ &= [\sigma(x_{i1}) - \sigma(a)] + [\sigma(x_{i2}) - \sigma(x_{i1})] + \dots + [\sigma(x_{in}) - \sigma(x_{in-1})] \\ &\quad + [\sigma(b) - \sigma(x_{in})] = \sigma(b). \end{aligned}$$

Giả sử $\beta \in D(a, b)$ là một đường đi bất kỳ từ a đến b và có dạng $\beta = ax_{j1}x_{j2} \dots x_{jk-1}x_{jk}b$. Theo bước 2 ta có bất đẳng thức sau:

$$l(a, x_{j1}) \geq \sigma(x_{j1}) - \sigma(a) = \sigma(x_{j1})$$

$$l(x_{j1}, x_{j2}) \geq \sigma(x_{j2}) - \sigma(x_{j1})$$

...

$$l(x_{jk-1}, x_{jk}) \geq \sigma(x_{jk}) - \sigma(x_{jk-1})$$

$$l(x_{jk}, b) \geq \sigma(b) - \sigma(x_{jk})$$

Cộng hai vế ta có:

$$l(\beta) = l(a, x_{j1}) + l(x_{j1}, x_{j2}) + \dots + l(x_{jk-1}, x_{jk}) + l(x_{jk}, b) \geq \sigma(b)$$

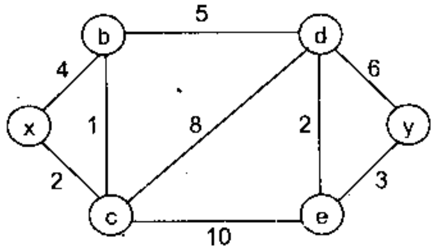
So sánh với trên ta có:

$$l(\alpha) = \min \{l(\beta) : \beta \in D(a, b)\}.$$

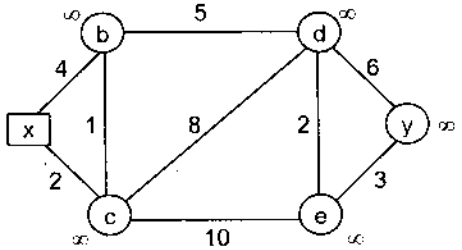
Dạng giả mã của thuật toán Dijkstra như sau:

```
Procedure Dijkstra (G: đồ thị liên thông có trọng số dương)
{G có các đỉnh a = v0, v1, ..., vn = b và trọng số l(vi, vj) = ∞ nếu (vi, vj) ∉ U của G}
For i := 1 to n
        σ(vi) := ∞;
σ(a) := 0;
S := ∅;
{ban đầu các nhãn được khởi tạo sao cho nhãn của a bằng 0, còn các đỉnh khác
bằng ∞, tập S là rỗng}
While b ∉ S
        Begin
                u := đỉnh không thuộc S có nhãn σ(u) nhỏ nhất;
                S := S ∪ {u};
                For <tất cả các đỉnh v không thuộc S>
                        if σ(u) + l(u, v) < σ(v) then σ(v) := σ(u) + l(u, v)
                {thêm vào S đỉnh có nhãn nhỏ nhất và sửa đổi nhãn của các
                đỉnh không thuộc S}
        End; {l(α) = l(a, b) = độ dài đường đi ngắn nhất từ a đến b}.
```

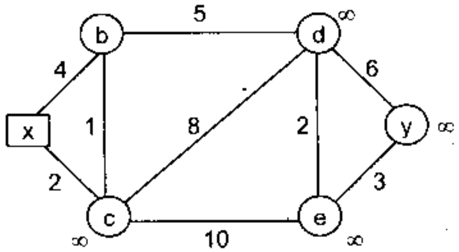
Ví dụ 3: Dùng thuật toán Dijkstra tìm đường đi độ dài ngắn nhất từ đỉnh x đến đỉnh y của đồ thị G = <X, U> có trọng số cho dưới dạng:



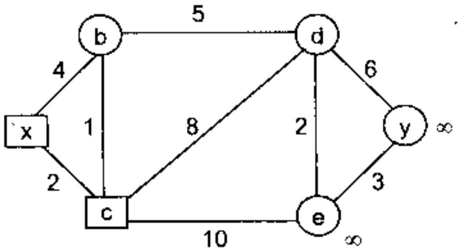
Giải: Các bước dùng thuật toán Dijkstra tìm độ dài đường đi ngắn nhất giữa đỉnh x và đỉnh y được thể hiện bởi các hình dưới đây:



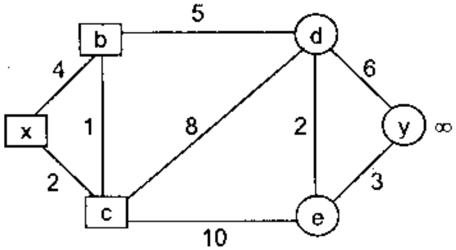
⇒

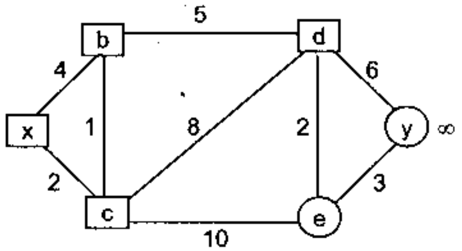


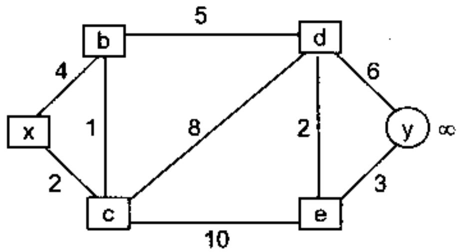
⇒

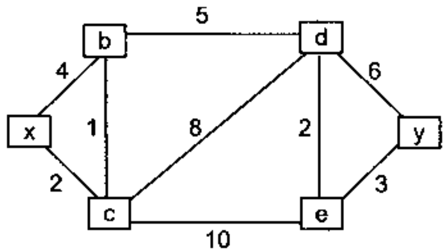


⇒



⇒ 

⇒ 

⇒ 

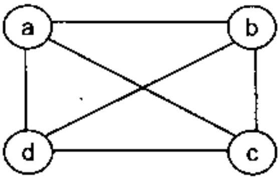
Tại mỗi bước lặp của thuật toán, các đỉnh của S được khoanh thành hình vuông. Đường đi ngắn nhất chỉ chứa các đỉnh đã thuộc vào tập S_k. Thuật toán kết thúc khi y được khoanh thành hình vuông, ta nhận được đường đi ngắn nhất từ x đến y là: xcbdey với độ dài là 13.

§8. MỘT SỐ TÍNH CHẤT CỦA ĐỒ THỊ PHẲNG

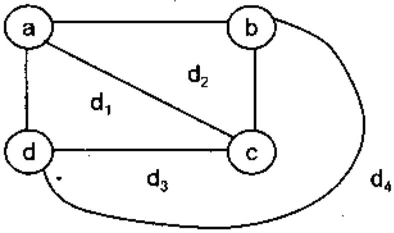
8.1. Khái niệm diện hữu hạn và diện vô hạn của đồ thị phẳng

Giả sử G = <X, U> là đồ thị phẳng. Khi đó G có thể biểu diễn trên mặt phẳng sao cho các cạnh chỉ cắt nhau tại đỉnh mà thôi.

Ví dụ 1: Cho đồ thị như hình a.



Hình a



Hình b

Đồ thị trên là đồ thị phẳng vì nó có cách biểu diễn trên mặt phẳng mà các cạnh chỉ cắt nhau ở đỉnh (hình b).

Biểu diễn phẳng này chia mặt phẳng thành 4 miền mà ta đánh số là d_1, d_2, d_3 và d_4. Ta gọi mỗi miền là một *diện*. Đồ thị trên có 4 diện d_1, d_2, d_3 và d_4, trong đó diện d_1, d_2 và d_3 là diện hữu hạn, còn diện d_4 là diện vô hạn. Chu trình của một diện là biên tạo nên diện đó, và mỗi một biên của diện được tạo nên bởi các cạnh của đồ thị phẳng.

Hai diện d_1, d_2 được gọi là kề nhau nếu biên của nó có ít nhất một cạnh chung (các diện chỉ tiếp xúc với nhau chỉ ở một đỉnh không coi là kề nhau).

Một cách tổng quát, nếu $G = <X, U>$ là một đồ thị phẳng thì biểu diễn phẳng G_P của $G = <X, U>$ chia mặt phẳng thành r diện $d_1, d_2, ..., d_r$, trong đó có $r - 1$ diện hữu hạn là $d_1, d_2, ..., d_{r-1}$ và một diện vô hạn là d_r.

Ký hiệu $m(d_i)$ là bậc của diện d_i và $m(d_i)$:= số các cạnh trong đồ thị tạo nên biên của diện d_i. Bậc của biểu diễn phẳng G_P của G ta ký hiệu là $m(G_P)$ và được định nghĩa như sau:

$$m(G_P) := \sum_{i=1}^{r} m(d_i)$$

ở đây: r là số diện được tạo ra trong biểu diễn phẳng G_P của $G = <X, U>$.

Dễ dàng kiểm tra lại: Đơn đồ thị phẳng và liên thông $G = <X, U>$ thoả mãn đẳng thức $m(G_P) = 2|U|$.

Hay bậc của biểu diễn phẳng của đồ thị đơn phẳng liên thông $G = <X, U>$ bằng hai lần số cạnh của đồ thị.

8.2. Chu số của đồ thị

Giả sử $G = <X, U>$ là một đồ thị (đa đồ thị) với $|X| = n$ đỉnh, $|U| = m$ cạnh và p thành phần liên thông. Ta ký hiệu chu số của đồ thị G là $c(G) := m - n + p$. Khi đó ta có kết quả sau đây:

Định lý 20: Số diện hữu hạn trong đồ thị đơn phẳng G = <X, U> luôn luôn bằng chu số của G = <X, U>.

Chứng minh: Dùng phương pháp quy nạp theo số diện hữu hạn h của G.

– Với h = 1, trong trường hợp này G là 1 diện hữu hạn, hay là một chu trình đơn duy nhất. Điều này chứng tỏ G có |X| = |U| và G liên thông. Vậy c(G) = m – n + p = 1 = h.

– Giả sử định lý đúng với h diện hữu hạn, tức là c(G) = m – n + p = h.

– Xét G có n đỉnh, m cạnh, p thành phần liên thông và h + 1 diện hữu hạn.

Chọn một cạnh nào đó trên biên của một diện hữu hạn mà nó tiếp xúc với diện vô hạn. Loại cạnh này, ta được đồ thị bộ phận G' của G. Trong G' có số diện hữu hạn là h. Theo giả thiết quy nạp thì c(G') = h. Vậy

$$(m - 1) - n + p = h \Leftrightarrow m - n + p = c(G) = h + 1.$$

Đó là điều cần chứng minh.

8.3. Công thức Euler

Định lý 21: Nếu đồ thị G = <X, U> phẳng, liên thông có |X| = n đỉnh, |U| = m cạnh và số diện trong biểu diễn phẳng là r thì ta luôn có:

$$r = m - n + 2 \text{ (công thức Euler)}$$

Chứng minh:

Theo định lý 20 thì c(G) = m – n + 1 = r – 1 (do trong r diện có 1 diện vô hạn, còn lại là diện hữu hạn). Rõ ràng

$$r = m - n + 2.$$

Từ định lý 21 ta có các hệ quả sau đây:

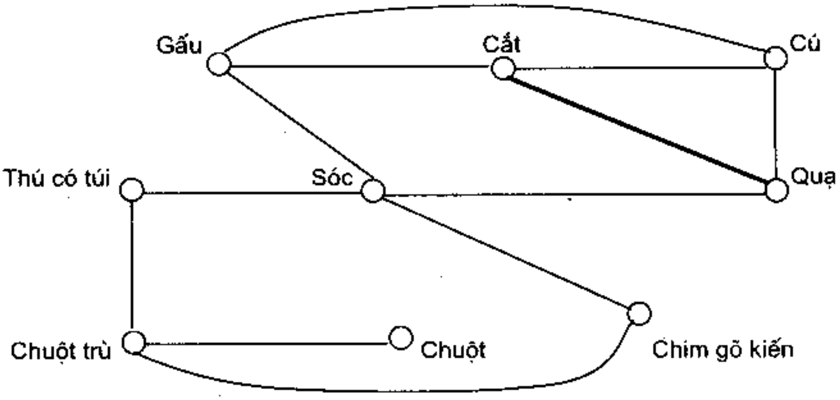
Hệ quả 1: Nếu G = <X, U> là đồ thị đơn, phẳng và liên thông có |X| = n đỉnh ($n \geq 3$), |U| = m cạnh thì $m \leq 3n - 6$.

Hệ quả 2: Nếu G = <X, U> là đồ thị đơn, phẳng, liên thông có |X| = n đỉnh ($n \geq 3$), |U| = m cạnh và không có chu trình độ dài 3 thì $m \leq 2n - 4$.

BÀI TẬP

1. *Đồ thị "lấn tổ" là một mô hình đồ thị biểu hiện sự cạnh tranh của các loài trong một hệ sinh thái nào đó. Mỗi loài vật được biểu diễn bởi một đỉnh. Mỗi cạnh vô hướng nối hai đỉnh nếu hai loài được biểu diễn bởi hai đỉnh này là cạnh tranh với nhau (tức là chúng cùng chung nguồn thức ăn).*

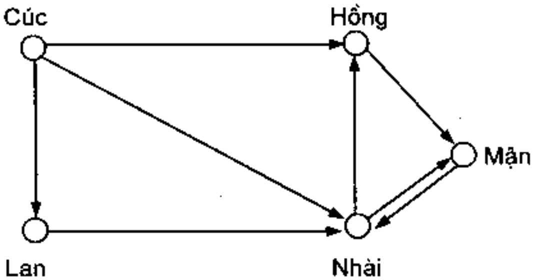
Cho đồ thị "lấn tổ" trong hệ sinh thái. Trong đồ thị ta thấy: Gấu và Sóc cạnh tranh; Cú và Quạ cạnh tranh nhau; Gấu và Quạ không cạnh tranh nhau...



Hãy xác định các loài thú cạnh tranh với Chuột trù và các loài thú không cạnh tranh với Chuột trù.

2. *Đồ thị ảnh hưởng: Khi nghiên cứu tính cách của một nhóm người ta thấy một số người có thể có ảnh hưởng lên suy nghĩ của một số người khác. Đồ thị có hướng có thể dùng để biểu thị bài toán này. Đồ thị có hướng như vậy được gọi là đồ thị ảnh hưởng.*

   Ví dụ: Cho đồ thị có hướng như hình dưới

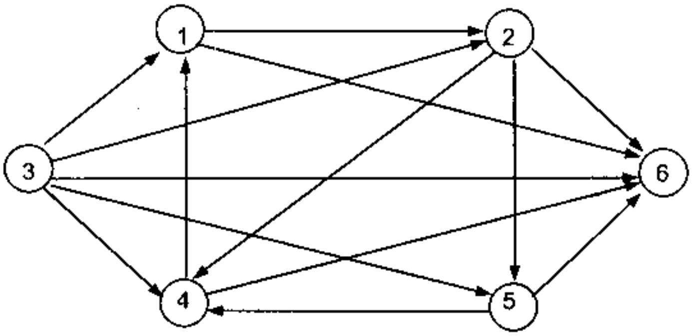


Cúc có ảnh hưởng tới Hồng, Lan và Nhài... Không ai có ảnh hưởng đến Cúc cả. Nhài và Mận có ảnh hưởng tới nhau.

Hãy xây dựng đồ thị ảnh hưởng của các thành viên lãnh đạo trong một trường đại học nếu biết: Hiệu trưởng có ảnh hưởng lên ba Hiệu phó và phòng Tổ chức cán bộ; Hiệu phó phụ trách đào tạo có ảnh hưởng lên phòng Đào tạo và phòng Đào tạo sau đại học; Hiệu phó phụ trách nghiên cứu khoa học có ảnh hưởng tới phòng Đào tạo sau đại học; Hiệu phó phụ trách cơ sở vật chất có ảnh hưởng tới phòng Hành chính quản trị; phòng Tài vụ không ảnh hưởng tới ai và cũng không có ai có thể ảnh hưởng tới phòng Tài vụ, còn phòng Tổ chức cán bộ có thể ảnh hưởng tới Hiệu trưởng.

3. *Mô hình đồ thị thi đấu vòng tròn: Một cuộc thi đấu thể thao, trong đó mỗi đội đấu với một đội khác đúng một lần được gọi là thi đấu vòng tròn. Cuộc thi đấu vòng tròn có thể biểu diễn bởi một đồ thị có hướng và mỗi đội tương ứng với một đỉnh. Một cạnh đi từ đỉnh a đến đỉnh b (theo hướng từ a sang b) nếu đội a thắng đội b. Trường hợp đội a hoà đội b thì giữa đỉnh a và đỉnh b không có cung nối với nhau.*

Ví dụ: Cho đồ thị như hình dưới mà mỗi đỉnh là một đội bóng:



Đội 3 thắng tất cả các đội. Đội 6 thua tất cả các đội. Đội 1 thắng đội 2 và 6, hoà với đội 5, nhưng thua các đội 3 và 4.

Trong trận đấu vòng tròn ở bảng A gồm các đội bóng Thể công (TC), Công an Hà Nội (CAHN), Sông lam Nghệ An (SLNA) và Công an Thành phố Hồ Chí Minh (CATPHCM) diễn ra theo kết quả sau:

TC thắng CAHN, hoà SLNA và thua CATPHCM,

CAHN thắng CATPHCM, nhưng thua TC và SLNA.

CATPHCM thắng SLNA và TC, ngưng thua CAHN.

Hãy dùng đồ thị có hướng để biểu diễn kết quả thi đấu của bốn đội trên.

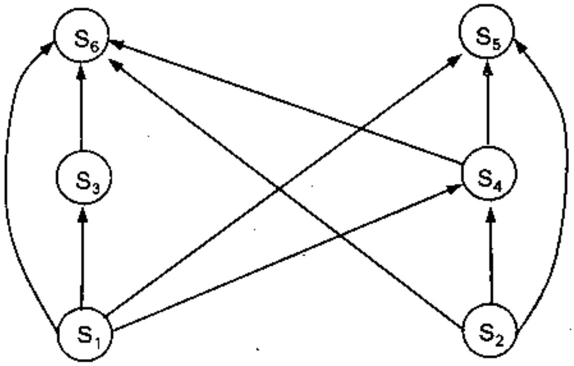
4. *Các chương trình máy tính có thể thực hiện nhanh hơn bằng cách thi hành đồng thời các câu lệnh nào đó, nhưng với điều kiện là không được thực hiện một câu lệnh đòi hỏi kết quả của các câu lệnh khác chưa được thực hiện.*

Sự phụ thuộc giữa các câu lệnh vào các câu lệnh trước có thể biểu diễn bằng đồ thị có hướng theo nguyên tắc sau: Mỗi câu lệnh đặt tương ứng *với một đỉnh. Nếu có một cung đi từ đỉnh a sang đỉnh b thì câu lệnh ở đỉnh b không thể thực hiện được trước khi câu lệnh ở đỉnh a thực hiện. Đồ thị này người ta gọi là đồ thị có ưu tiên trước sau.*

Ví dụ: Cho chương trình máy tính S và đồ thị có ưu tiên trước sau của S dưới dạng sau đây:

Chương trình S:
$$\begin{cases} s_1 : a := 0 \\ s_2 : b := 1 \\ s_3 : c := a + 1 \\ s_4 : d := b + a \\ s_5 : e := d + 1 \\ s_6 := e := c + d \end{cases}$$

Đồ thị của S:



s_5 không thể thực hiện được nếu chưa thực hiện s_1, s_2 và s_4. Cũng như vậy s_6 chưa thể thực hiện được nếu chưa thực hiện s_1, s_2, s_3 và s_4.

Xây dựng đồ thị có ưu tiên trước sau cho chương trình S bao gồm các câu lệnh dạng sau:

$s_1 : x := 0$

$s_2 : x := x + 1$

$s_3 : y := 2$

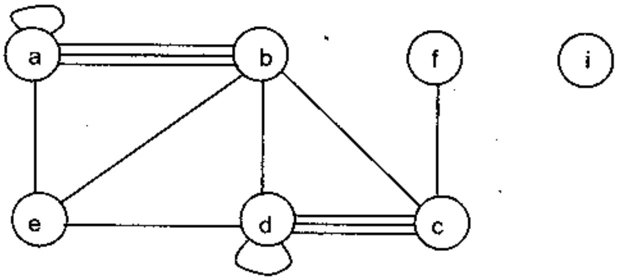
$s_4 : z := y$

$s_5 : x := x + 2$

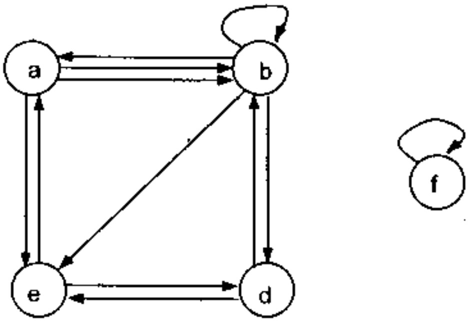
$s_6 : y := x + z$

$s_7 : z := 4.$

5. Cho đồ thị vô hướng



và có hướng

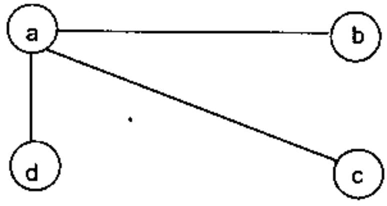


a) Hãy xác định bậc của các đồ thị.

b) Chỉ ra các đỉnh cô lập và đỉnh treo của các đồ thị.

c) Kiểm tra xem bậc của đồ thị có bằng hai lần số cạnh hay không.

6. Cho đồ thị như hình dưới



Hãy vẽ tất cả các đồ thị con của đồ thị trên.

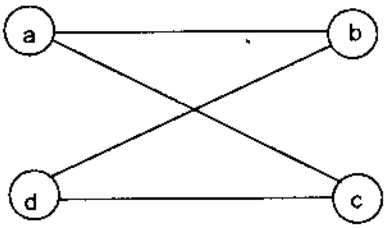
7. Trong một cuộc liên hoan, mọi người đều bắt tay nhau. Hãy chỉ ra rằng tổng số lượt người được bắt tay là một số chẵn (giả sử rằng không ai tự bắt tay mình).

8. Cho ma trận kề

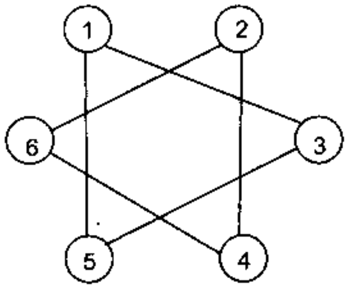
$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix}$$

Hãy vẽ đồ thị được biểu diễn bằng ma trận kề trên.

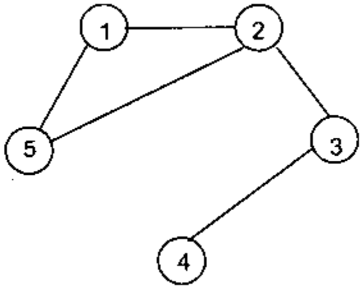
9. Có bao nhiêu đường đi độ dài 4 từ đỉnh a đến đỉnh d trong đồ thị sau:



10. Cho hai đồ thị sau:



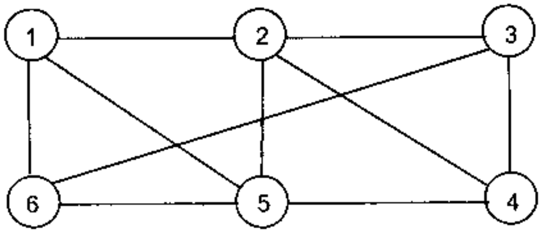
và



Đồ thị nào liên thông, đồ thị nào không liên thông?

11. Chỉ ra rằng, nếu đơn đồ thị G có k thành phần liên thông và các thành phần liên thông này có tương ứng $n_1, n_2, ..., n_k$ đỉnh, khi đó số cạnh của G không vượt quá $\sum_{i=1}^{k} C(n_i, 2)$.

12. Cho đồ thị:



Tìm số đường đi giữa hai đỉnh 3, 6 có độ dài:

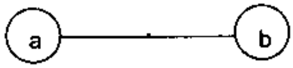
a) 2; b) 3; c) 4;

d) 5; e) 6; f) 7.

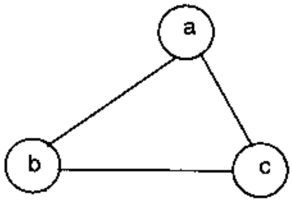
13. Cho các đồ thị đầy đủ K_n (n chỉ số đỉnh) với $1 \le n \le 6$:

K_1 :

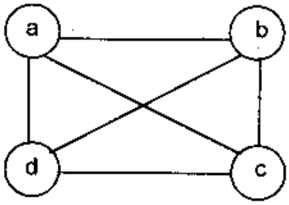
K_2:



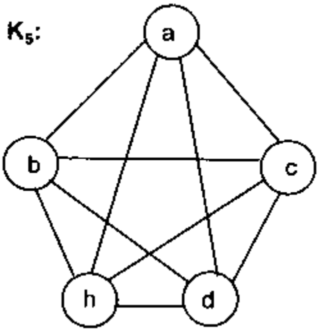
K_3:



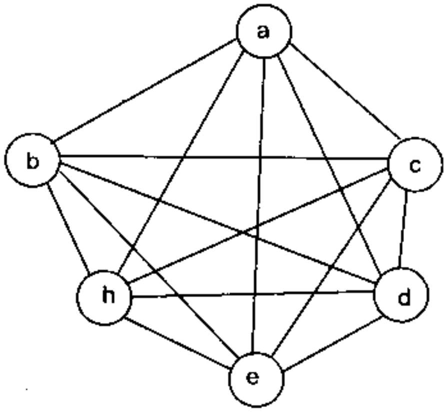
K_4:



K_5:



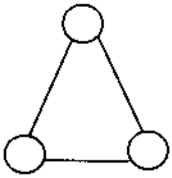
K_6:



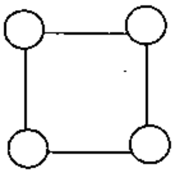
a) Hãy biểu diễn K_n bằng danh sách kề và ma trận kề.

b) Tìm số cạnh của đồ thị đầy đủ K_n.

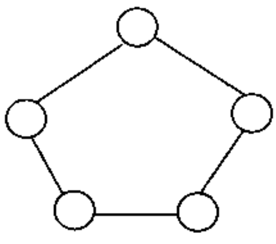
14. Chu trình C_n ($n \geq 3$) là một đồ thị có n đỉnh $v_1, v_2, ..., v_n$ và n cạnh $(v_1, v_2), (v_2, v_3), ..., (v_{n-1}, v_n)$ và (v_n, v_1). Các chu trình C_n ($n = 3, 4, 5, 6$) là



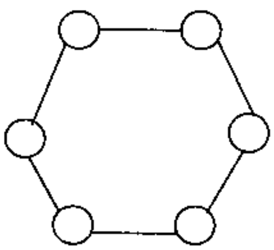
C_3



C_4



C_5



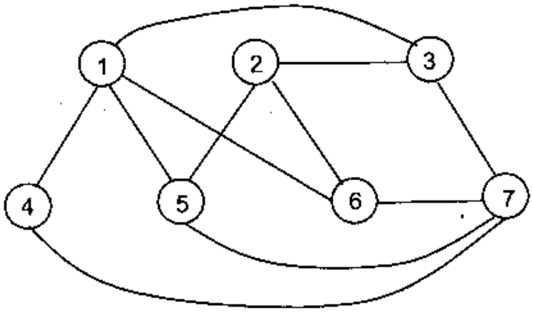
C_6

a) Có bao nhiêu đồ thị con, đồ thị bộ phận đối với đồ thị C_n (n = 3, 4, 5, 6)?

b) Chứng minh bậc của C_n được tính theo công thức $m(C_n) = 2n$ ($n \geq 3$).

c) Tìm số ổn định trong, ổn định ngoài và nhân của đồ thị C_n.

15. Cho đồ thị



a) Hãy chỉ ra nó là đồ thị phân đôi.

b) Tìm số ổn định trong, số ổn định ngoài và nhân của đồ thị trên.

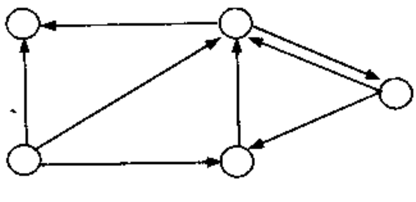
16. a) Nếu đồ thị đơn G có n đỉnh và m cạnh, khi đó $\overline{G}$ có bao nhiêu cạnh?

b) Chứng minh rằng nếu G là đồ thị đơn có n đỉnh thì $G \cup \overline{G} = K_n$.

17. Nghịch đảo của đồ thị có hướng $G = <X, U>$ được ký hiệu là G^{-1} là đồ thị có hướng được xác định như sau: $G^{-1} = <X, V>$ trong đó $(x, y) \in V$ khi và chỉ khi $(y, x) \in U$.

a) Chứng minh $(G^{-1})^{-1} = G$.

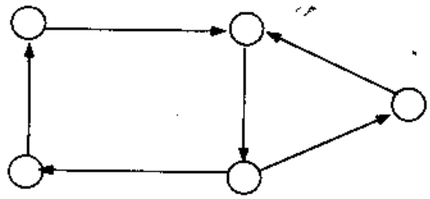
b) Hãy vẽ đồ thị nghịch đảo của đồ thị:



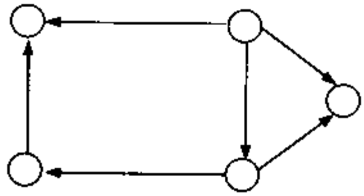
18. Dùng phương pháp đồ thị thể hiện việc bố trí lịch cho các lớp CNTT với 7 môn thi trong 7 ngày. Yêu cầu phải bố trí lịch thi sao cho hai môn thi của cùng một thầy giáo không được rơi vào hai ngày liên tiếp nhau. Biết rằng không có thầy giáo nào có nhiều hơn 4 môn thi.

19. *Bài toán lập lịch:* Hãy lập lịch thi trong một trường đại học sao cho không có sinh viên nào có hai môn thi cùng một lúc.

20. Cho hai đồ thị có hướng:



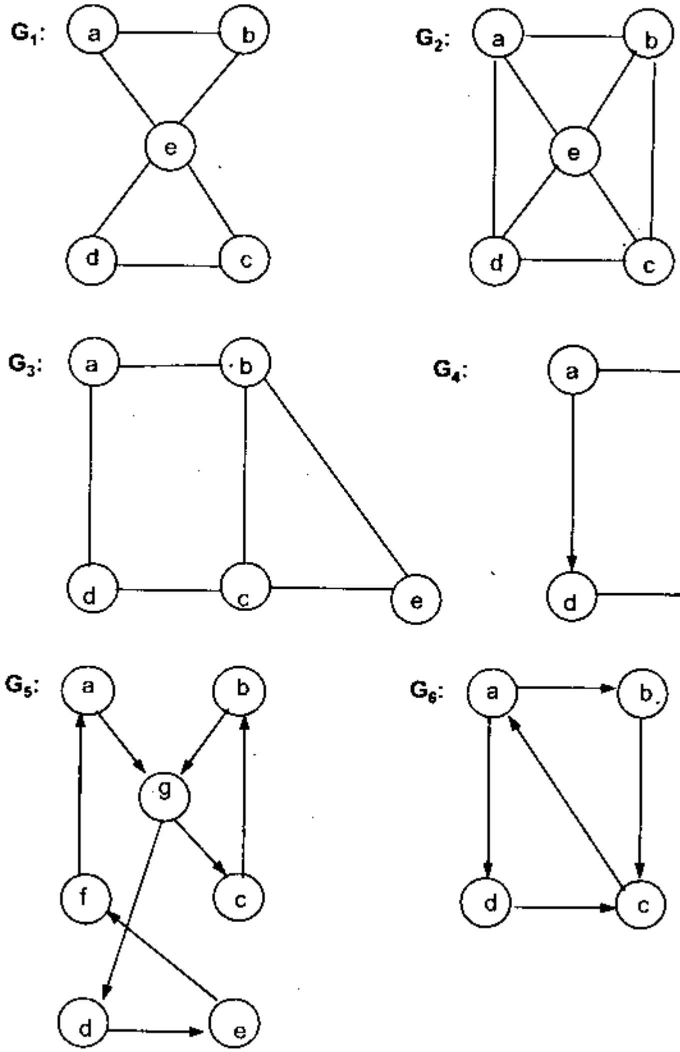
và



Đồ thị nào là liên thông mạnh? Đồ thị nào là liên thông yếu?

(Đồ thị có hướng là liên thông yếu nếu có đường đi giữa hai đỉnh bất kỳ của đồ thị vô hướng nền).

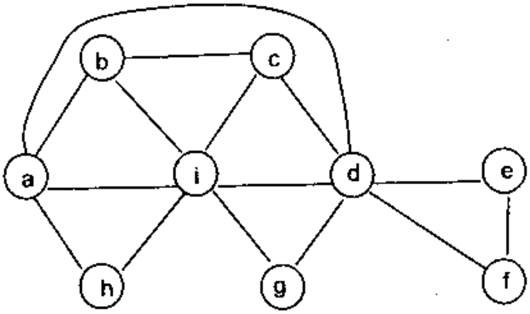
21. Cho các đồ thị:



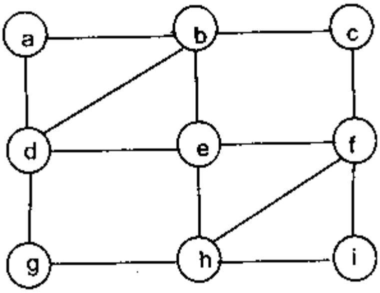
Kiểm tra xem đồ thị nào có chu trình và đường Euler, vì sao? Nếu có hãy chỉ ra.

22. Cho hai đồ thị:

G_1:

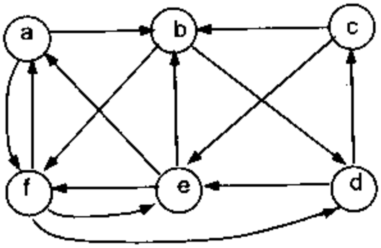


G_2:



Hai đồ thị trên, đồ thị nào là đồ thị Euler?

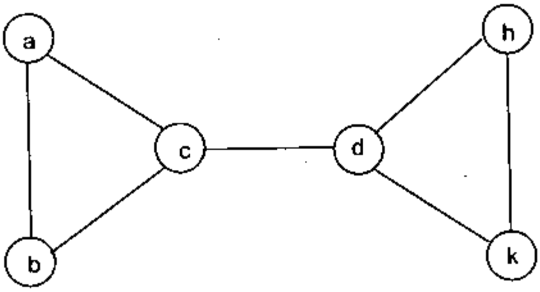
23. Cho đồ thị có hướng:



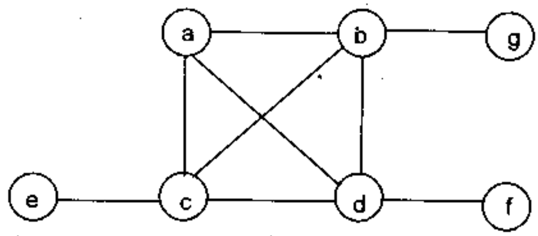
Kiểm tra xem đồ thị trên có chu trình Euler hay không? Nếu có hãy chỉ ra các chu trình Euler đó.

24. Cho các đồ thị:

G_1:



G_2:

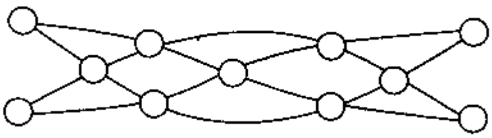


G_1, G_2 có đường và chu trình Hamilton không? Nếu có thì vẽ đường và chu trình Hamilton đó, nếu không giải thích vì sao?

25. Cho G = <X, U> là đồ thị liên thông. Chứng minh rằng nếu các đỉnh của đường đơn dài nhất tạo thành đồ thị con G' = <X', U'> có chu trình Hamilton thì G cũng có chu trình Hamilton.

26. Nhà vua mời 2n ($n \geq 2$) kỵ mã đến dự tiệc. Mỗi kỵ mã quen ít nhất n kỵ mã đến dự tiệc. Chứng minh rằng luôn luôn có thể xếp tất cả các kỵ mã ngồi xung quanh một bàn tròn, sao cho mỗi người ngồi giữa hai người mà mình quen biết.

27. Có thể vẽ bức tranh thanh mã tấu dưới đây bằng một nét liền (không nâng bút lên khỏi trang giấy trong quá trình vẽ) hay không?

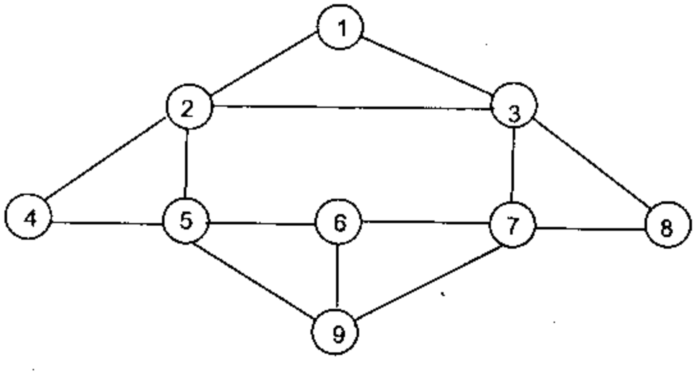


28. Một nước có 10 thành phố. Hãy thiết lập một mạng cầu hàng không sao cho:

a) Mỗi thành phố có cầu hàng không nối trực tiếp với đúng ba thành phố khác.

b) Từ mỗi thành phố có cầu hàng không đi tới một thành phố tùy ý sao cho trên đường hành trình tới đích có thể đi qua các thành phố khác, mỗi thành phố đi qua đúng một lần.

29. Bản đồ thành phố mà người đưa thư cần phải đi qua là đồ thị:

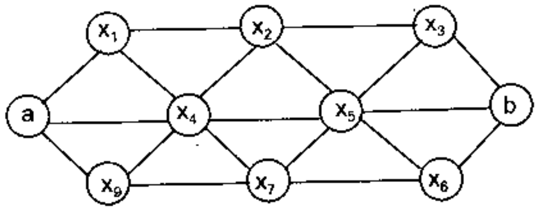


Người đưa thư xuất phát từ đỉnh 2 đi qua tất cả các thành phố (cạnh) để đưa thư rồi lại quay về nơi xuất phát.

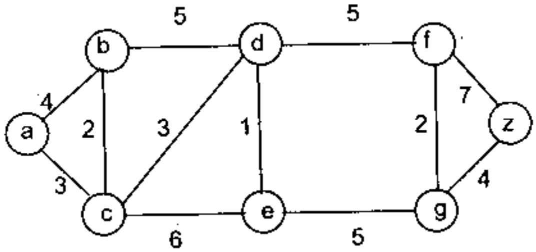
Hãy chỉ ra đường đi ngắn nhất của người đưa thư với giả thiết độ dài của mỗi cạnh là như nhau.

30. Khi nào có thể sơn vạch phân đôi các đường phố trong một thành phố mà không cần đi qua mỗi phố quá một lần? (Giả thiết tất cả các phố đều có đường hai chiều).

31. Tìm đường đi ngắn nhất trong đồ thị không có trọng số từ đỉnh a đến đỉnh b sau đây:



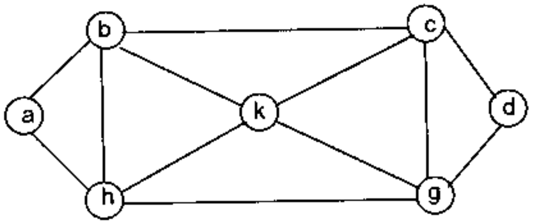
32. Cho đồ thị có trọng số sau



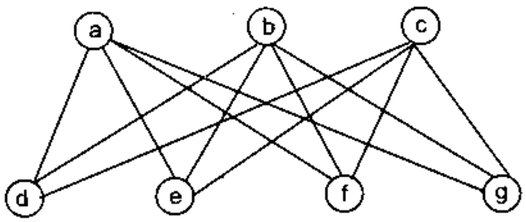
Tìm đường đi ngắn nhất giữa hai đỉnh a và z.

33. Tìm sắc số của các đồ thị sau:

G_1:

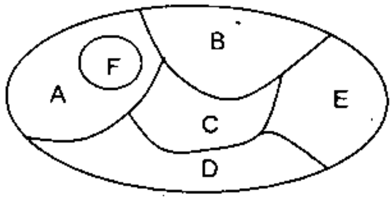


G_2:

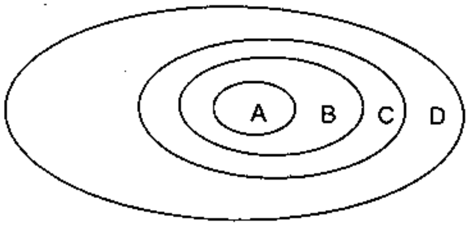


34. Cho các bản đồ của hai vùng:

a)



b)



Hãy vẽ đồ thị tương ứng của hai bản đồ trên và tìm sắc số của nó.

Chương 8
CÂY VÀ ỨNG DỤNG CỦA CÂY

§1. ĐỊNH NGHĨA VÀ CÁC VÍ DỤ VỀ CÂY

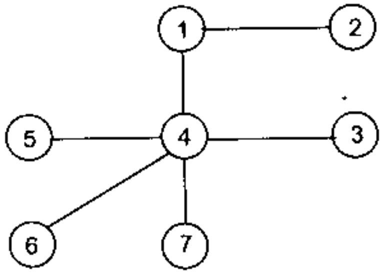
Cây có rất nhiều ứng dụng trong tin học. Người ta dùng cây để nghiên cứu, xây dựng các thuật toán như sắp xếp, định vị các phần tử trong danh sách. Cây dùng trong việc xây dựng mạng máy tính, dùng để tạo ra các mã lưu trữ dữ liệu...

1.1. Cây

Ta định nghĩa cây như sau: *Cây* là một đồ thị đơn, liên thông và không có chu trình.

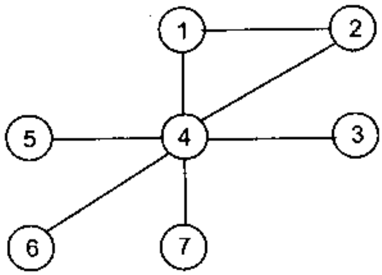
Ví dụ 1:

- $G_1 = <X_1, U_1>$:



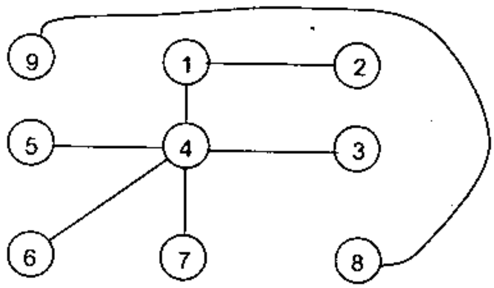
là một cây, vì G_1 là đồ thị đơn, liên thông không có chu trình.

- $G_2 = <X_2, U_2>$:



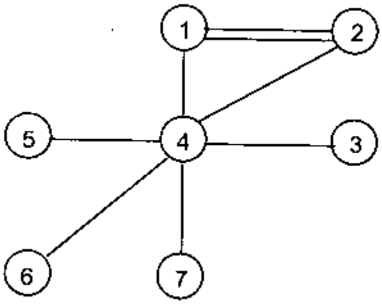
không phải là một cây, vì G_2 có chu trình mặc dù liên thông và đơn.

- $G_3 = <X_3, U_3>$:



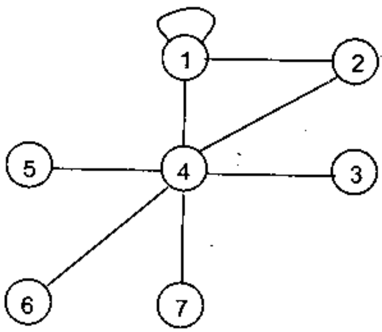
không phải là một cây, vì không liên thông mặc dù là đơn và không có chu trình.

- $G_4 = <X_4, U_4>$:



không phải là cây, vì nó không phải là đồ thị đơn mà là đa đồ thị.

- $G_5 = <X_5, U_5>$:

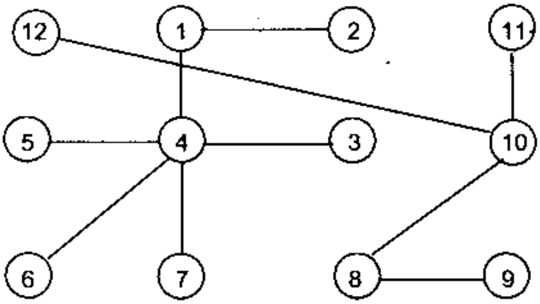


cũng không phải là cây, vì nó có khuyên nên không phải là đồ thị đơn.

1.2. Rừng cây

Một đồ thị vô hướng gồm k thành phần liên thông mà mỗi thành phần liên thông là một cây được gọi là *rừng cây*.

Đồ thị G = <X, U> có hai thành phần liên thông dưới đây là một rừng:

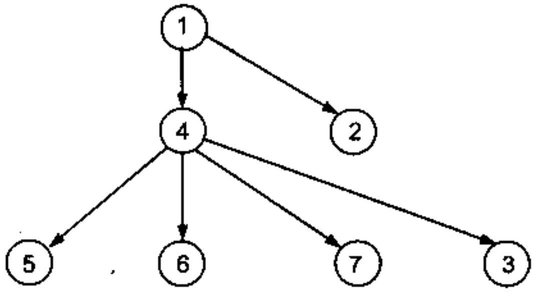


1.3. Cây có gốc

Trong một cây, nếu ta lấy một đỉnh nào đó làm gốc thì ta có thể gán cho mỗi cạnh của cây một hướng đi từ gốc ra. Cây như vậy gọi là *cây có gốc*. Cây có gốc là một đồ thị có hướng mà gốc là một đỉnh không có bậc vào, còn lá là đỉnh không có bậc ra. Thường cây có gốc có đỉnh gốc nằm phía trên còn các lá nằm phía dưới. Đỉnh của cây có gốc không phải là gốc cũng không phải là lá gọi là *đỉnh trong*. Nếu trong cây có gốc ta lấy một đỉnh trong làm gốc thì cây đó gọi là *cây con* có gốc của cây có gốc ban đầu. Đôi khi trong cây có gốc có thể bỏ hướng mũi tên đi từ gốc đến các lá, vì khi chọn gốc tức là hướng của các cạnh đã được xác định.

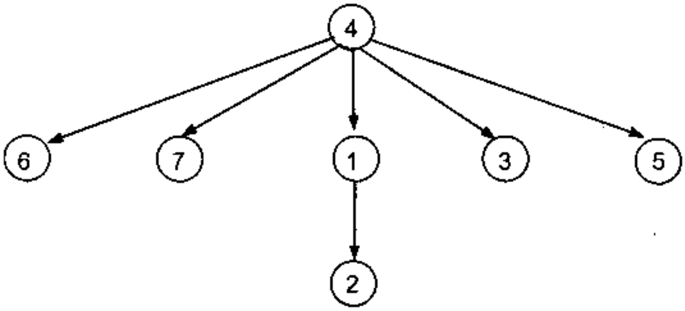
Ví dụ 2: Cho cây $G_1 = <X_1, U_1>$.

- Nếu ta chọn đỉnh 1 làm gốc thì cây có gốc có dạng:



với gốc là đỉnh 1, các lá là các đỉnh 2, 3, 5, 6, 7, còn đỉnh trong là đỉnh 4.

• Nếu ta chọn 4 làm đỉnh gốc thì ta nhận được cây có gốc là cây có dạng:



với đỉnh gốc là 4, các lá là 6, 7, 2, 3 và 5, còn 1 là đỉnh trong.

Chiều cao h của cây là mức cao nhất trong cây, hay là độ dài của đường đi lớn nhất từ gốc đến lá của cây.

1.4. Cây m – phân, cây m – phân đầy đủ và cây nhị phân

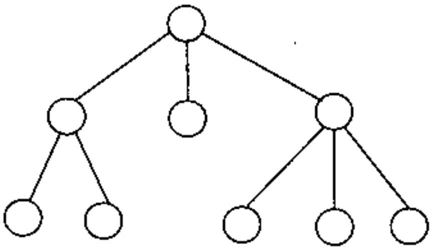
Cây có gốc được gọi là *cây m – phân* nếu tất cả các đỉnh của cây có không quá m con.

Cây m – phân đầy đủ là cây m – phân với mọi đỉnh trong (kể cả gốc) có đúng m con.

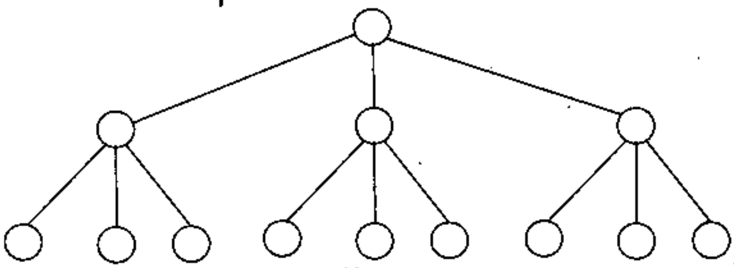
Cây m – phân có gốc và độ cao h được gọi là *cây cân đối* nếu tất cả các lá đều ở mức h hoặc h – 1.

Cây m – phân với m = 2 gọi là *cây nhị phân*.

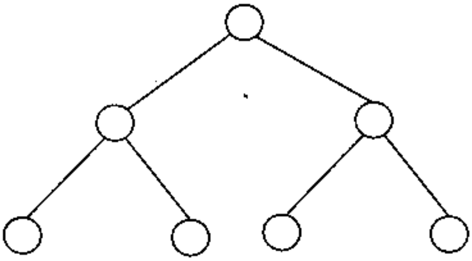
Ví dụ 3: Cho các cây:



H_1



H_2



H_3

Trong các cây trên, cây H_1 là cây 3 – phân; H_2 là cây 3 – phân đầy đủ; H_3 là cây nhị phân.

§2. MỘT SỐ TÍNH CHẤT CỦA CÂY

Định lý 1. Giả sử G = <X, U> là một cây. Khi đó các mệnh đề sau đây là tương đương:

1) G = <X, U> là đồ thị liên thông và không có chu trình.

2) G = <X, U> có n – 1 cạnh và không có chu trình.

3) G = <X, U> liên thông và có n – 1 cạnh.

4) G = <X, U> không có chu trình, nhưng nếu thêm vào một cạnh nối hai đỉnh không kề nhau thì xuất hiện duy nhất một chu trình.

5) G liên thông và nếu trong G = <X, U> bỏ đi một cạnh tùy ý thì đồ thị nhận được sẽ không liên thông.

6) Hai đỉnh bất kỳ của G = <X, U> được nối với nhau bởi một đường đi đơn.

Chứng minh: Ta chứng minh theo trình tự sau:

$$1) \Rightarrow 2) \Rightarrow 3) \Rightarrow 4) \Rightarrow 5) \Rightarrow 6 \Rightarrow 1)$$

Sử dụng công thức $c(G) = m - n + p$ là chu số của đồ thị G = <X, U>, ở đây $|X| = n$, $|U| = m$, p là số thành phần liên thông của G.

1) ⇒ 2): Vì G không có chu trình nên $c(G) = 0$ hay $m - n + p = 0$. Do G liên thông nên $p = 1$. Vậy $m - n + 1 = 0$, hay $m = n - 1$. Chứng tỏ G không có chu trình và số cạnh là $n - 1$.

2) ⇒ 3): Giả sử G không có chu trình và có đúng $n - 1$ cạnh. Ta chứng minh G liên thông và có $n - 1$ cạnh.

Thật vậy, giả sử ngược lại G không liên thông. Khi đó $p \geq 2$, $c(G) = m - n + p = 0$ (do G không có chu trình). Thay $m = n - 1$ vào đẳng thức trên ta có: $p - 1 = 0 \Rightarrow p = 1$; trái với $p \geq 2$. Vậy G liên thông và có số cạnh là $n - 1$.

3) ⇒ 4): Giả sử G liên thông có $n - 1$ cạnh. Ta chỉ ra G không có chu trình và nếu thêm vào một cạnh nối hai đỉnh không kề nhau thì xuất hiện duy nhất một chu trình.

Thật vậy, vì G liên thông nên $p = 1$, mặt khác $m = n - 1$ nên ta có:

$$c(G) = m - n + 1 = 0 \tag{1}$$

Hay G không có chu trình.

Nếu ta thêm vào G một cạnh ta được đồ thị G' với số cạnh là n, ta có

$$c(G') = n - n + 1 = 1 \tag{2}$$

Chứng tỏ nếu thêm vào G một cạnh thì G' xuất hiện duy nhất một chu trình.

Từ (1) và (2) ta suy ra 4).

4) ⇒ 5): Giả sử G không có chu trình và nếu thêm vào một cạnh nối hai đỉnh không kề nhau thì xuất hiện duy nhất một chu trình. Ta chỉ ra G liên thông và nếu trong G bỏ đi một cạnh tùy ý thì đồ thị nhận được sẽ không liên thông.

Giả sử ngược lại G không liên thông, tức là tồn tại cặp đỉnh (x, y) trong G mà không có đường nào nối x với y. Khi đó nối x với y bởi một cạnh đồ thị nhận được vẫn không có chu trình. Điều này mâu thuẫn với 4). Vậy G là liên thông.

Nếu trong G bỏ đi một cạnh mà đồ thị nhận được vẫn liên thông thì nếu khôi phục lại cạnh này đồ thị sẽ có chu trình. Điều này mâu thuẫn với 4). Toán lại ta có 5).

5) ⇒ 6): Giả sử ngược lại, nếu trong G có tồn tại cặp đỉnh (x, y) không nối với nhau bằng đường nào cả. Chứng tỏ G không liên thông. Mâu thuẫn với 5) là G liên thông, hay mỗi cặp đỉnh đều có một đường nối với nhau. Đường đó là đường đơn, vì nếu có hai đường nối cặp đỉnh x, y thì sau khi bỏ đi một đường đồ thị vẫn liên thông, trái với 5) là nếu bỏ đi một cạnh thì đồ thị sẽ không liên thông.

6) ⇒ 1): Với mỗi cặp đỉnh nối với nhau bởi một đường thì theo định nghĩa G là liên thông. Giả sử G có chu trình, xét cặp đỉnh trên chu trình đó. Khi đó đỉnh x và y có hai đường nối với nhau, trái với 6).

Định lý được chứng minh.

Định lý 2. Một cây có ít nhất 2 đỉnh treo.

Chứng minh: Ta chứng minh bằng phản chứng. Giả sử trong cây G số đỉnh treo nhỏ hơn 2. Có hai trường hợp xảy ra:

a) Số đỉnh treo = 0.

b) Số đỉnh treo = 1.

Ta chứng minh cả hai trường hợp đều không xảy ra.

Trường hợp a): Nếu số đỉnh treo = 0, tức là không có đỉnh treo, hay mỗi đỉnh trong G đều có ít nhất 2 cạnh liên quan đến nó, tức là bậc của nó ≥ 2. Xuất phát từ mỗi đỉnh vào 1 cạnh và đi ra 1 cạnh khác (không lặp lại), quá trình này là vô hạn, vì nếu hữu hạn thì xuất hiện đỉnh treo. Tính vô hạn này mâu thuẫn với tính hữu hạn của các cạnh trong đồ thị.

Trường hợp b): Giả sử có 1 đỉnh treo. Ta xuất phát từ đỉnh này đi ngược lại như quá trình trên thì cũng dẫn đến sự mâu thuẫn với tính hữu hạn của đồ thị.

Tóm lại, định lý được chứng minh.

Dưới đây ta đưa ra một số kết quả của cây m – phân:

Định lý 3:

1) Cây m – phân đầy đủ với i đỉnh trong sẽ có tất cả $n = m.i + 1$ đỉnh.

2) Cây m – phân đầy đủ với n đỉnh sẽ có $i = \frac{n-1}{3}$ đỉnh trong và $l = \frac{(m-1)n+1}{m}$ đỉnh ngoài (lá).

3) Cây m – phân đầy đủ với i đỉnh trong sẽ có $n = m.i + 1$ đỉnh và $l = (m-1)i + 1$ lá.

4) Cây m – phân đầy đủ có $n = \frac{ml-1}{m-1}$ đỉnh và $i = \frac{l-1}{m-1}$ đỉnh trong.

Định lý 4:

1) Có nhiều nhất h lá trong cây m – phân với độ cao h.

2) Nếu cây m – phân có độ cao h và có l lá thì $h \geq [\log_m l]$.

3) Nếu cây m – phân đầy đủ cân đối thì $h = [\log_m l]$.

Ở đây [x] là số nguyên nhỏ nhất lớn hơn hay bằng x.

Bổ đề 1: Số lượng các đỉnh ở mức i của cây nhị phân đầy đủ là 2^i với $i = 0, 1, 2, \ldots$

Chứng minh: Chứng minh bằng quy nạp theo i.

– Với i = 0: Đỉnh thuộc mức 0 là gốc, mà mỗi cây chỉ có một gốc nên bổ đề đúng.

– Giả sử bổ đề đúng với i = k, tức là ở mức k cây có 2^k đỉnh.

– Xét mức k + 1 và chỉ ra số đỉnh ở mức này là 2^{k+1}.

Thật vậy, mỗi đỉnh ở mức k có đúng 2 con nên 2^k đỉnh ở mức k có $2 \times 2^k = 2^{k+1}$ con, hay số đỉnh ở mức k + 1 là 2^{k+1}.

Bổ đề 2: Số lượng các đỉnh trong và đỉnh ngoài của cây nhị phân đầy đủ có độ cao h là: $2^{h+1} - 1$.

Chứng minh: Suy ra từ bổ đề 1:

Số các đỉnh trong và đỉnh ngoài ở mức h = số các đỉnh ở mức 0 + số các đỉnh ở mức 1 + . . . + số các đỉnh ở mức h = $2^0 + 2^1 + ... + 2^h = 2^{h+1} - 1$.

Định lý 5: Cho cây nhị phân đầy đủ với L = số các đỉnh ngoài. Khi đó số các đỉnh trong của cây là L – 1.

Chứng minh: Gọi chiều cao của cây là h. Theo bổ đề 1 thì số các đỉnh ngoài $L = 2^h$. Còn số các đỉnh trong và ngoài theo bổ đề 2 là $(2^{h+1} - 1)$. Nếu đặt K = số đỉnh trong thì $K + L = 2^{h+1} - 1$.

Vậy $K = (2 \times 2^h - 1) - 2^h = 2L - 1 - L = L - 1$.

Định lý 6. Cây nhị phân đầy đủ có n đỉnh thì chiều cao h của nó được tính theo công thức: $h = \log_2(n + 1) - 1$.

Chứng minh: Theo bổ đề 2 thì:

$$n = 2^{h+1} - 1 \Leftrightarrow n + 1 = 2^{h+1}$$

Loga cơ số 2 hai vế đẳng thức trên ta được:

$$\log_2(n + 1) = h + 1 \text{ hay } h = \log_2(n + 1) - 1.$$

§3. ỨNG DỤNG CỦA CÂY

Xét ba bài toán bằng mô hình cây:

Bài toán 1: Các phần tử trong một danh sách được lưu trữ như thế nào để có thể dễ dàng định vị được chúng?

Bài toán 2: Hãy xác định dãy các quyết định để tìm đối tượng có tính chất x trong tập hợp các đối tượng nào đó.

Bài toán 3: Cần phải mã hoá tập các chữ cái bằng các dãy nhị phân như thế nào để có hiệu quả sử dụng tốt nhất?

3.1. Cây tìm kiếm nhị phân đối với bài toán 1

Tìm kiếm một phần tử trong một danh sách là công việc thường gặp trong Tin học. Để giải quyết nó cần một thuật toán tìm kiếm có hiệu quả. Đó là cây tìm kiếm nhị phân. Cây tìm kiếm nhị phân là một cây nhị phân, trong đó mỗi con của một đỉnh hoặc là con bên phải hoặc là con bên trái, mỗi đỉnh được gán một khoá sao cho với mỗi khoá chỉ xác định được một phần tử và khoá của đỉnh lớn hơn khoá của các đỉnh cây con bên trái và nhỏ hơn khoá của các đỉnh ở cây con bên phải của nó.

Thủ tục đệ quy sau đây cho phép tạo lập cây tìm kiếm nhị phân cho một danh sách các phần tử:

– *Bước 1:* Bắt đầu với cây có đúng 1 đỉnh là đỉnh gốc: Phần tử đầu trong danh sách là khoá của đỉnh này.

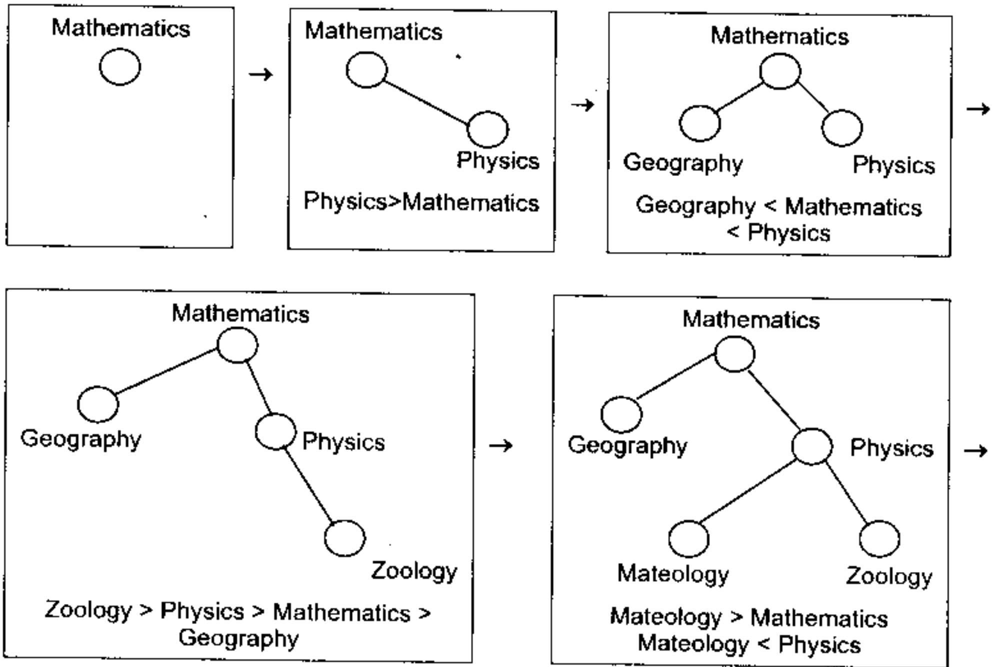
– *Bước 2:* Để thêm một phần tử mới (đỉnh mới) ta so sánh nó với khoá của các đỉnh đã có trên cây bắt đầu từ gốc và đi sang trái nếu phần tử nhỏ hơn khoá của đỉnh tương ứng, hoặc đi sang phải nếu phần tử này lớn hơn khoá của đỉnh tương ứng.

– *Bước 3:* Nếu cây nhị phân tìm kiếm được xây dựng mà không phải là cây nhị phân đầy đủ thì có thể thêm các đỉnh mới không nhãn để tạo ra cây nhị phân đầy đủ (mỗi khoá đều có 2 con), điều này giúp cho việc định vị hoặc thêm vào phần tử mới như là khoá của đỉnh mà không phải thêm vào một đỉnh mới.

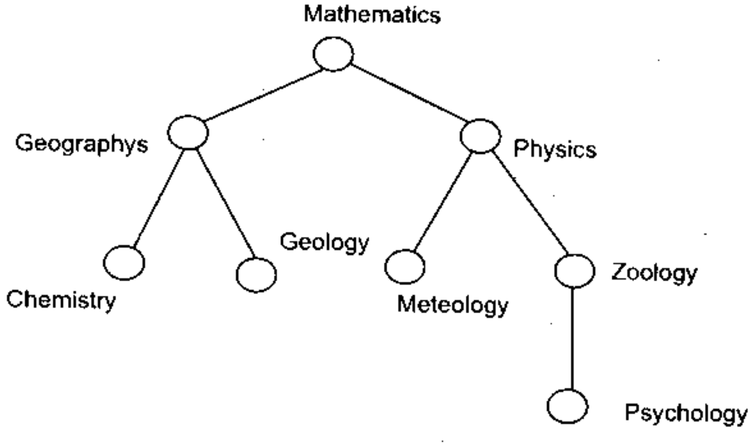
Như vậy, cây tìm kiếm nhị phân dùng để định vị các phần tử dựa trên một loạt các so sánh, trong đó mỗi so sánh cho biết có định vị được phần tử đó hay chưa, hoặc sẽ đi theo cây con bên phải hay bên trái.

Ví dụ 1: Hãy tạo cây tìm kiếm nhị phân sao cho các từ: mathematics, physics, geography, zoology, meteology, geology, psychology và chemistry theo thứ tự từ điển.

Giải. Thủ tục tạo dựng cây tìm kiếm nhị phân theo thứ tự từ điển từ danh sách trên được mô tả thông qua các bước dưới đây:



... và cuối cùng ta có cây tìm kiếm nhị phân của danh sách trên theo thứ tự từ điển tiếng Anh là:



(Chemistry < Geographys < Geology < Mathematics < Meteology < Physics < Psychology < Zoology)

3.2. Cây quyết định đối với bài toán 2

Cây có gốc có thể dùng để mô hình các bài toán trong đó có một dãy các quyết định dẫn đến lời giải của bài toán. Cụ thể cây có gốc, trong đó mỗi đỉnh trong ứng với một quyết định và mỗi cây con tại các đỉnh này ứng với một kết cục có thể của quyết định được gọi là *cây quyết định*. Những lời giải có thể của bài toán tương ứng với các đường đi tới các lá của cây có gốc. Chúng ta sẽ đề cập thuật toán sắp xếp bằng cây quyết định trong các bài tiếp theo.

Ví dụ 2: Cần phải cân bao nhiêu lần bằng một chiếc cân 2 đĩa (cân thăng bằng) để tìm đồng xu giả trong 4 đồng xu (có 3 đồng thật, một đồng giả). Biết đồng xu giả có trọng lượng nhẹ hơn hoặc nặng hơn các đồng xu thật. Mô tả thuật toán tìm đồng xu giả với số lần cân tìm được.

Giải:

Có ba khả năng xảy ra đối với mỗi lần cân: 2 đĩa có trọng lượng như nhau, đĩa thứ nhất nặng hơn đĩa thứ hai và đĩa thứ nhất nhẹ hơn đĩa thứ 2.

Do đó cây quyết định cho một dãy các lần cân là cây 3 – phân, và có ít nhất 4 lá trong cây quyết định vì có 4 kết cục có thể và mỗi kết cục phải được biểu diễn bằng ít nhất 1 lá. Số lần cân nhiều nhất để xác định đồng xu giả là chiều cao của cây 3 – phân. Thủ tục cân như sau:

– Đầu tiên cân đồng xu 1 với đồng xu 2, nếu chúng bằng nhau thì cân đồng xu 1 với đồng xu 3. Nếu đồng xu 1 và 3 bằng nhau thì đồng xu giả là đồng xu 4. Nếu đồng xu 1 khác đồng xu 3 thì đồng xu 3 là giả.

– Nếu đồng xu 1 khác đồng xu 2 thì ta cân đồng xu 1 với đồng xu 3. Nếu đồng xu 1 và đồng xu 3 bằng nhau thì đồng xu 2 là giả, còn nếu không cân bằng thì đồng xu 1 là giả.

3.3. Các mã tiền tố đối với bài toán 3

a) Bài toán mã hoá các chữ cái tiếng Anh bằng các dãy nhị phân

Xâu nhị phân độ dài 5 có thể dùng để mã hoá một chữ cái tiếng Anh. Vì có tất cả 32 xâu nhị phân khác nhau độ dài 5, trong khi đó số chữ cái tiếng Anh chỉ có 26. Mã hoá tuân thủ theo các nguyên tắc sau:

– Các chữ cái xuất hiện thường xuyên hơn sẽ được mã hoá bằng các xâu nhị phân ngắn, các xâu nhị phân dài hơn dùng để mã các chữ cái xuất hiện ít hơn.

– Khi các chữ cái được mã bằng số bit thay đổi cần phải có cách xác định xem các bit ứng với mỗi chữ bắt đầu và kết thúc ở đâu. Chẳng hạn, nếu chữ e được mã bằng 0, chữ a được mã bằng 1 và chữ t được mã bằng 01, khi đó xâu nhị phân 0101 có thể tương ứng với eat, eaea hoặc tt.

– Để đảm bảo không có xâu nhị phân nào ứng với hơn một dãy chữ cái, xâu nhị phân ứng với một chữ không bao giờ xuất hiện như là phần đầu của xâu nhị phân ứng với chữ khác.

Mã có các tính chất như trên được gọi là *mã tiền tố*.

Ví dụ 3: Mã chữ e bằng 0, mã chữ a bằng 10, còn mã chữ t bằng 11 là mã tiền tố. Xâu 10110 là mã của từ ate. Để giải mã ta thấy số 1 đầu tiên không biểu diễn chữ nào, nhưng 10 lại biểu diễn chữ a (nó không phải là phần đầu của xâu nhị phân biểu diễn chữ khác). Sau đó số 1 tiếp theo không biểu diễn một chữ, nhưng 11 là biểu diễn chữ t và số 0 cuối cùng biểu diễn chữ e.

b) Cây nhị phân dùng biểu diễn mã tiền tố

Nguyên tắc xây dựng cây nhị phân biểu diễn mã tiền tố:

– Lá của cây nhị phân được gán nhãn là một ký tự.

– Các cạnh của cây nhị phân được gán nhãn sao cho cạnh dẫn tới con bên trái được gán nhãn 0, còn cạnh dẫn tới con bên phải được gán nhãn 1.

– Xâu nhị phân mã hoá một chữ là dãy các nhãn của các cạnh thuộc đường đi duy nhất từ gốc tới lá có nhãn của chữ cái đó.

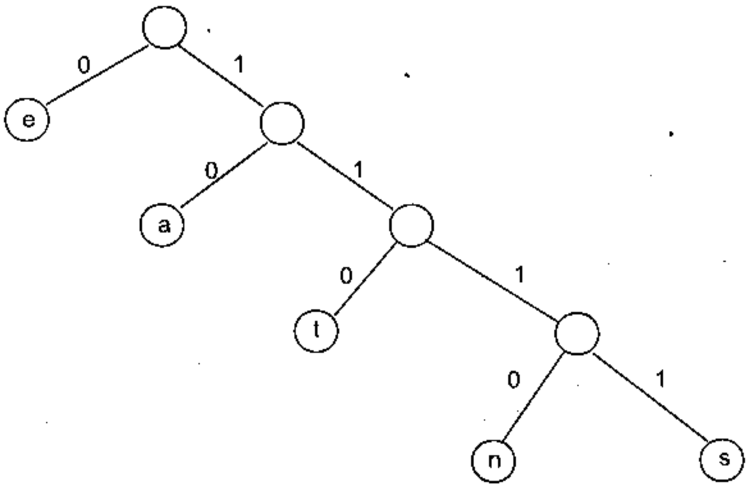
– Khi giải mã thì dùng chính cây nhị phân này để giải mã dãy nhị phân mà nó biểu diễn.

Như vậy, ta có thể xây dựng mã tiền tố bằng một cây nhị phân bất kỳ có cạnh trái của mỗi đỉnh được gán nhãn 0, còn cạnh phải được gán nhãn 1. Dĩ nhiên cây nhị phân này phải tuân thủ các nguyên tắc trong mục a).

Ví dụ 4: Xây dựng cây nhị phân mã tiền tố của các chữ e mã bằng 0, chữ a mã bằng 10, chữ t mã bằng 110, chữ n mã bằng 1110 và chữ s mã bằng 1111. Dùng cây nhị phân vừa xây dựng giải mã từ có mã tiền tố là: 11111011100.

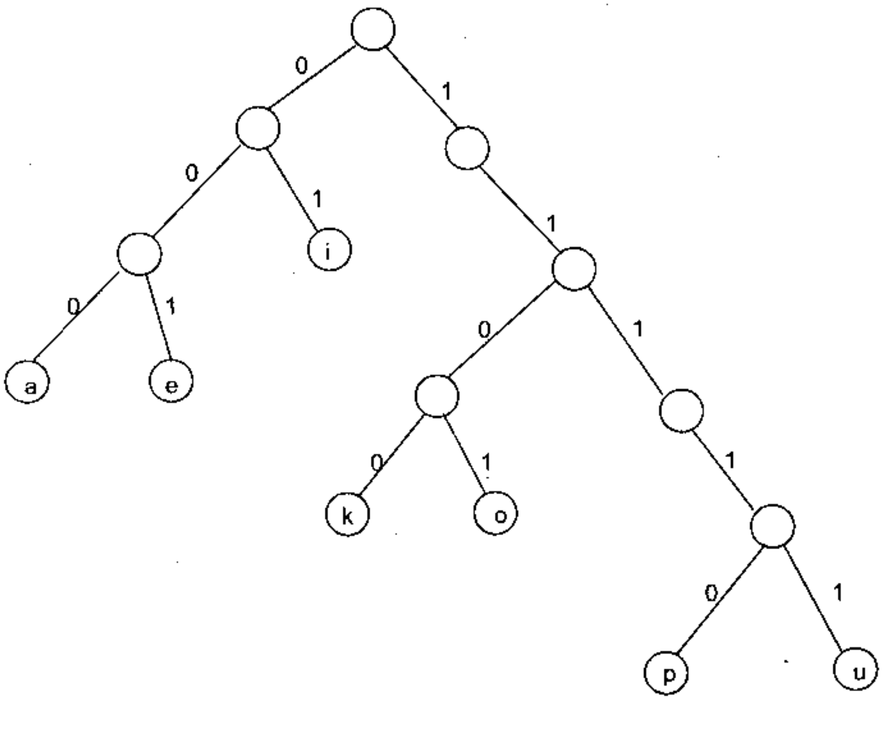
Giải:

Theo nguyên tắc xây dựng cây nhị phân mã tiền tố ta có cây nhị phân cần xác định là:



Giải mã xâu 11111011100 như sau: 1111 là xâu đi từ gốc tới lá s, hay xâu 1111 là mã của chữ s. Tiếp theo xét bit thứ 5 và 6 là 10, đi từ gốc thì 10 là mã của chữ a. Tiếp theo là từ bit 7 đến bis thứ 10 là 1110, đi từ gốc dẫn tới lá n, hay n có mã tiền tố là 1110. Bit thứ 11 trong dãy là 0, bit này đi từ gốc sẽ dẫn tới lá e. Vậy xâu 11111011100 là mã của xâu ký tự sane.

Ví dụ 5: Xác định mã tiền tố của a, e, i, k, o, p và u nếu sơ đồ mã được cho bởi cây nhị phân dưới đây:



Giải:

Từ cây nhị phân mã tiền tố ta có kết quả sau:

– a có mã là 000; – e có mã là 001;

– i có mã là 01; – k có mã là 1100;

– o có mã là 1101; – p có mã là 11110.

– u có mã là 11111.

§4. CÁC PHƯƠNG PHÁP DUYỆT CÂY

Cây có gốc được sắp thứ tự dùng để lưu trữ thông tin. Chúng ta cần có các thủ tục "Viếng thăm" các đỉnh của cây để truy nhập dữ liệu. Cây có gốc và được sắp thứ tự còn sử dụng để biểu diễn các biểu thức khác nhau, như biểu thức số học chứa các số, các biến và các phép toán. Nó không những dùng để biểu diễn các biểu thức số học mà nó còn rất hữu ích trong việc tính giá trị của các biểu thức này.

4.1. Hệ địa chỉ phổ dụng

Thủ tục duyệt tất cả các đỉnh của cây có gốc và được sắp thứ tự đều dựa trên việc sắp thứ tự các đỉnh con theo thứ tự từ trái sang phải.

Để làm được việc này, trước tiên cần phải gán nhãn cho tất cả các đỉnh bằng phương pháp truy hồi sau đây:

– Gán nhãn cho gốc bởi số 0, sau đó k đỉnh con của nó (ở mức 1) từ trái sang phải được gán nhãn 1, 2, ..., k.

– Với mọi đỉnh v ở mức n có nhãn là A thì k đỉnh con của nó từ trái sang phải được gán nhãn A1, A2, ..., Ak_v.

– Theo thủ tục trên thì đỉnh v ở mức $n \geq 1$ có nhãn $x_1x_2 \ldots x_n$, trong đó đường đi duy nhất từ gốc tới v sẽ đi qua đỉnh x_1 ở mức 1, đỉnh x_2 ở mức 2, ..., và đỉnh x_n ở mức n.

Cách gán nhãn như trên được gọi là *hệ địa chỉ phổ dụng* của một cây có gốc và được sắp. Ta có thể sắp tất cả các đỉnh của cây theo thứ tự từ điển của các nhãn trong hệ địa chỉ phổ dụng.

Đỉnh có nhãn $x_1x_2 \ldots x_n$ là nhỏ hơn đỉnh có nhãn $y_1y_2 \ldots y_m$ khi và chỉ khi $x_1 = y_1$, $x_2 = y_2$, ..., $x_{i-1} = y_{i-1}$ và $x_i < y_i$ với $0 \leq i \leq n$.

4.2. Thuật toán duyệt cây

a) Thuật toán duyệt tiền thứ tự

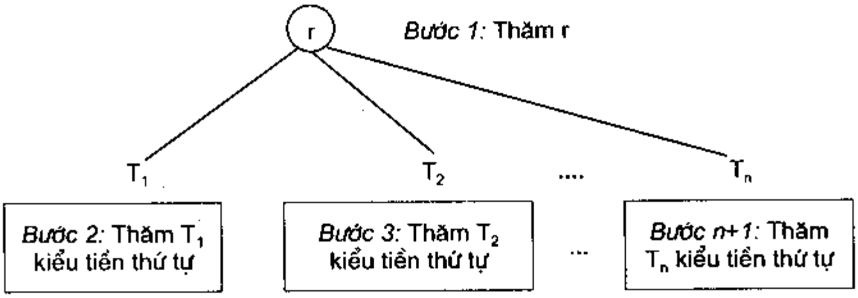
Thuật toán được thực hiện theo các bước sau:

– Giả sử T là một cây có gốc và được sắp thứ tự với gốc là r.

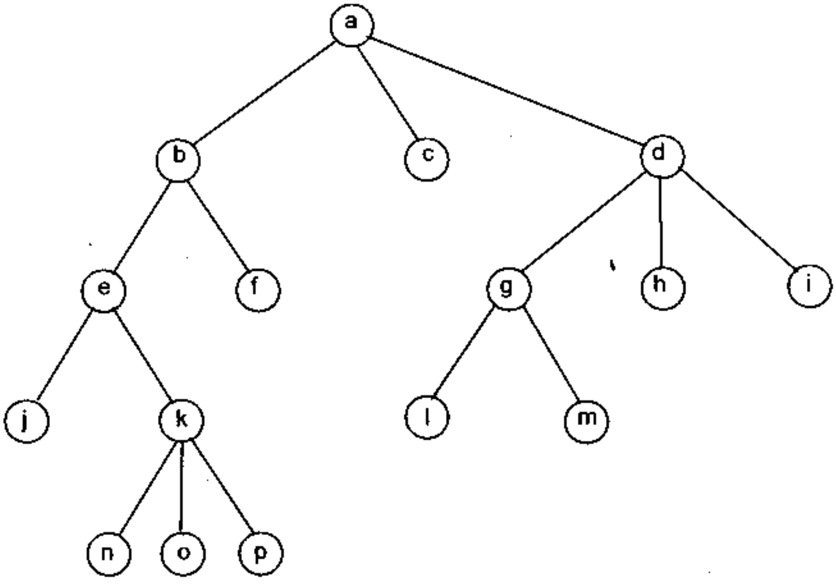
– Nếu T chỉ có r thì r là cách duyệt tiền thứ tự của T.

– Nếu tại đỉnh r còn có các cây con T_1, T_2, ..., T_n theo thứ tự từ trái sang phải. Duyệt tiền thứ tự sẽ đến thăm r đầu tiên. Tiếp tục duyệt T_1 theo kiểu tiền thứ tự. Sau đó duyệt T_2 theo kiểu tiền thứ tự, ... Cuối cùng đến T_n được duyệt theo kiểm tiền thứ tự.

Thủ tục trên mô tả dưới dạng cây sau:



Ví dụ 1: Cho cây có gốc được sắp T:



Cách duyệt tiền thứ tự cây T trên theo thứ tự là: a, b, e, j, k, n, o, p, f, c, d, g, l, m, h, i.

Thuật toán duyệt kiểu tiền thứ tự dưới dạng giả mã đối với cây T có gốc và được sắp như sau:

```
Procedure preorder (T: cây có gốc và được sắp);
r := gốc của cây T;
liệt kê r;
For <mỗi cây con c của r từ trái sang phải>
        begin
                T(c) := cây con với gốc c;
                preorder (T(c))
        end.
```

b) Thuật toán duyệt trung thứ tự

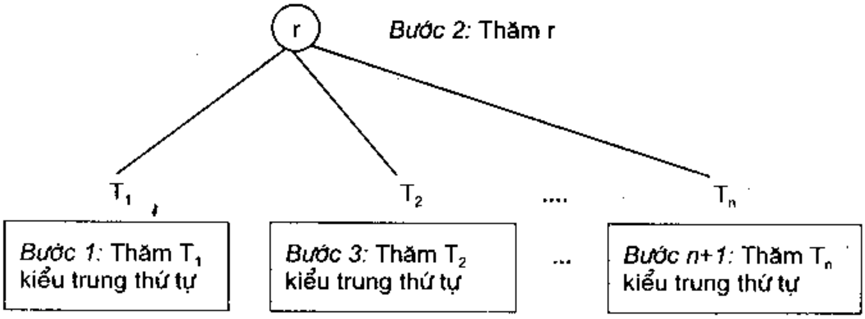
Thuật toán được thực hiện theo các bước sau:

– Giả sử T là một cây có gốc và được sắp thứ tự với gốc là r.

– Nếu T chỉ có r thì r là cách duyệt trung thứ tự của T.

– Nếu tại đỉnh r còn có các cây con T_1, T_2, ..., T_n theo thứ tự từ trái sang phải. Duyệt trung thứ tự sẽ bắt đầu bằng việc duyệt T_1 theo kiểu trung thứ tự, sau đó viếng thăm r. Tiếp tục duyệt T_2 theo kiểu trung thứ tự, ... Cuối cùng duyệt T_n theo kiểu trung thứ tự.

Thủ tục trên mô tả dưới dạng cây sau:



Thuật toán duyệt kiểu trung thứ tự dưới dạng giả mã đối với cây T có gốc và được sắp như sau:

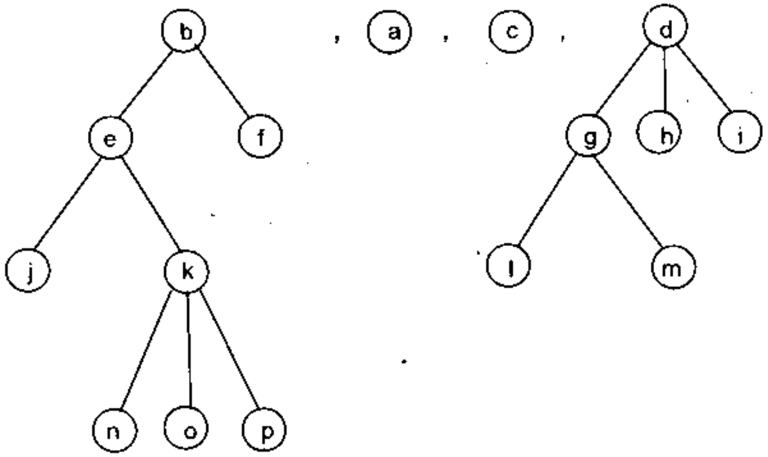
```
Procedure inorder (T: cây có gốc và được sắp);
r := gốc của cây T;
If <t là lá> then
        liệt kê r
```

```
else
        begin
                l := con đầu tiên từ trái sang phải của r;
                T(l) := Cây con với gốc l;
                inorder (T(l));
                liệt kê r;
                For <mỗi cây con c của r từ trái sang phải trừ l>
                        T(c) := Cây con với gốc c;
                inorder (T(c))
        end.
```

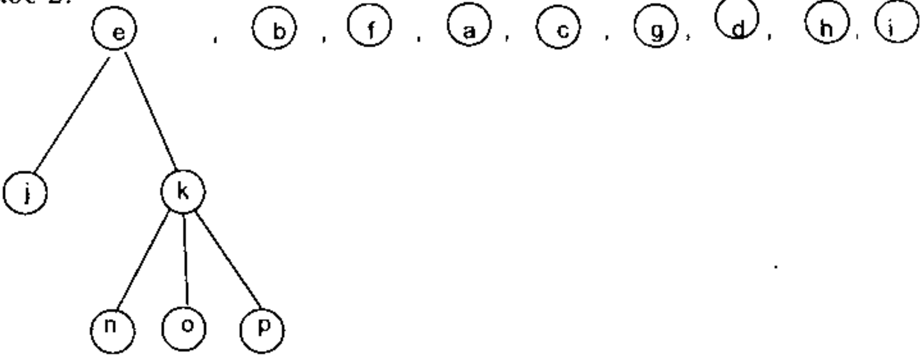
Ví dụ 2: Cho cây T có gốc và được sắp như trong ví dụ 1. Cách duyệt trung thứ tự của cây T theo thứ tự nào?

Giải:

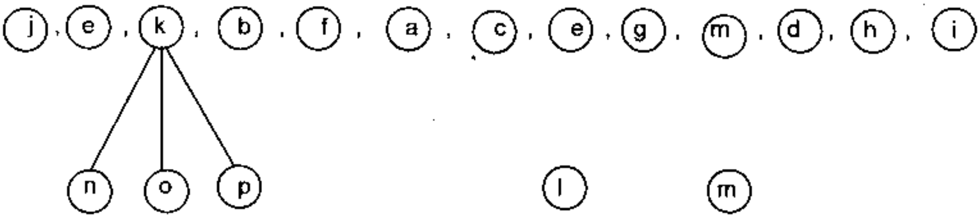
– *Bước 1:* Viếng thăm cây con cực trái, thăm gốc, thăm các cây con khác từ trái sang phải lần lượt theo thuật toán duyệt trung thứ tự đối với cây T ta được:



– *Bước 2:*



– Bước 3:



– Bước 4

j, e, n, k, o, p, b, f, a, c, l, g, m, d, h, i

Tóm lại, duyệt trung thứ tự cây T trong ví dụ 1 là lần lượt viếng thăm các đỉnh được liệt kê từ j đến i trong bước 4.

c) Thuật toán duyệt hậu thứ tự

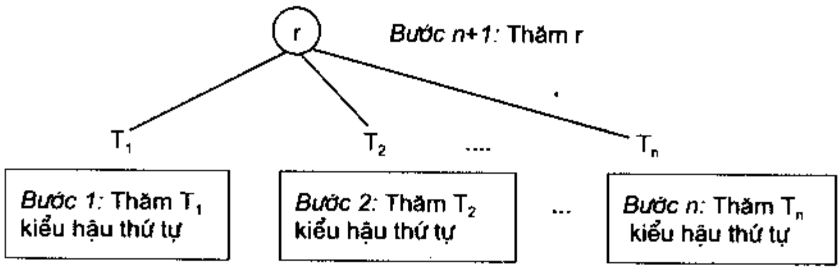
Thuật toán được thực hiện theo các bước sau:

– Giả sử T là một cây có gốc và được sắp thứ tự với gốc là r.

– Nếu T chỉ có r thì r là cách duyệt hậu thứ tự của T.

– Nếu không phải thì r còn có các cây con T_1, T_2, ..., T_n theo thứ tự từ trái sang phải. Duyệt hậu thứ tự sẽ bắt đầu bằng duyệt T_1 theo kiểu hậu thứ tự, sau đó duyệt T_2 theo kiểu hậu thứ tự và tiếp tục cho đến khi T_n được duyệt theo kiểu hậu thứ tự. Cuối cùng kết thúc bằng việc viếng thăm đỉnh gốc r của T.

Thủ tục trên mô tả dưới dạng cây:



Thuật toán duyệt hậu thứ tự dưới dạng giả mã như sau:

```
Procedure postorder (T: Cây có gốc và được sắp);
r := gốc của T;
For <mỗi cây con c của r từ trái sang phải>
Begin
        T(c) := cây con với gốc c;
        Postorder (T(c))
End.
```

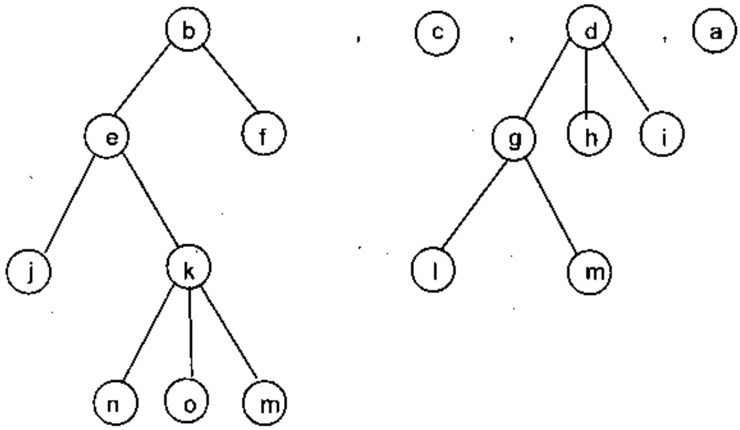
Ví dụ 3: Cách duyệt kiểu hậu thứ tự đối với cây có gốc và được sắp cho trong ví dụ 1 được cho theo thứ tự nào?

Giải:

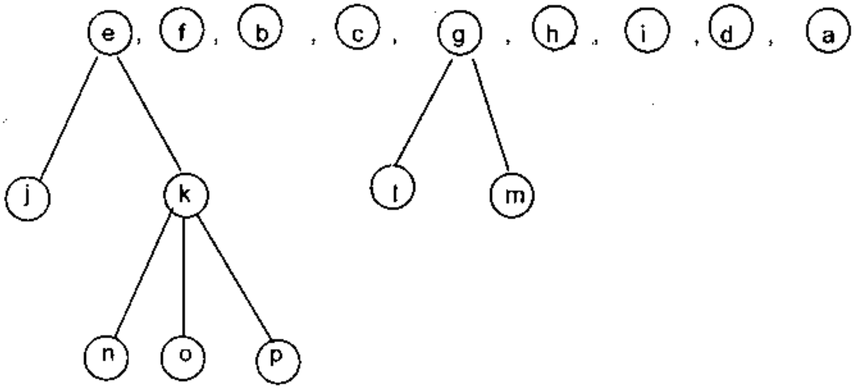
Duyệt kiểu hậu thứ tự đối với cây T bắt đầu bằng cách duyệt hậu thứ tự cây con với gốc b, duyệt hậu thứ tự cây con với gốc c (chính là c) và duyệt hậu thứ tự cây con với gốc d và cuối cùng là gốc a.

Cụ thể gồm các bước duyệt như sau:

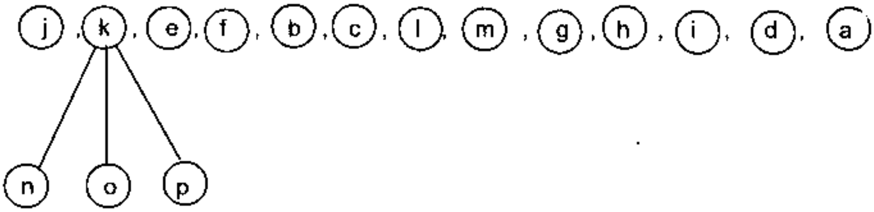
– *Bước 1:*



– *Bước 2:*



– *Bước 3:*



– *Bước 4*

j, n, o, p, k, e, f, b, c, l, m, g, h, i, d, a

là kết quả duyệt theo kiểu hậu thứ tự đối với cây T trong ví dụ 1.

§5. CÂY VÀ CÁC BÀI TOÁN SẮP XẾP

Bài toán sắp xếp các phần tử của một tập hợp được ứng dụng nhiều và thường gặp trong cuộc sống. Chẳng hạn, để lập danh bạ điện thoại cần phải sắp xếp tên của những người thuê bao theo thứ tự từ điển...

Sắp xếp là sự sắp đặt lại các phần tử vào một danh sách theo thứ tự nào đó. Ví dụ, sắp xếp các phần tử 7, 2, 4, 3, 1, 6, 5 thành một danh sách theo thứ tự tăng dần: 1, 2, 3, 4, 5, 6, 7. Hay sắp xếp các chữ cái d, h, c, a, f theo thứ tự từ điển ta được a, c, d, f, h.

5.1. Thuật toán sắp xếp nhị nguyên

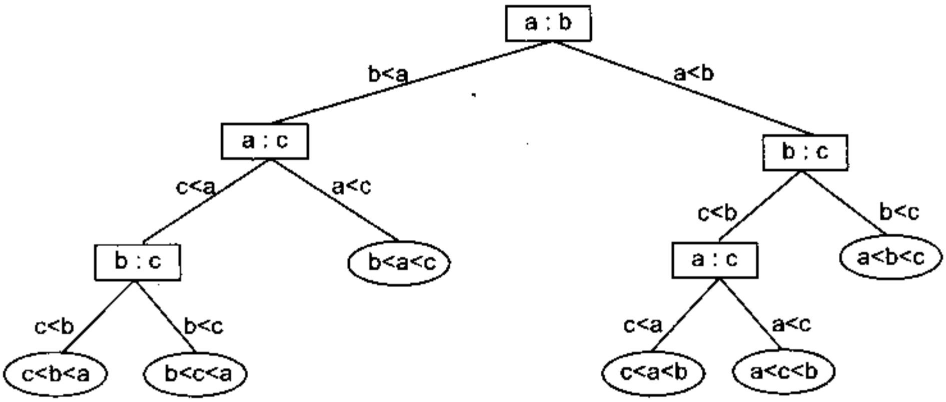
Phần lớn các công việc của máy tính là sắp xếp lại các đối tượng thuộc các loại khác nhau. Chính vì vậy, các nhà nghiên cứu không thể không nghiên cứu và phát triển thuật toán sắp xếp sao cho có hiệu quả nhất.

Giả sử có một tập gồm n phần tử. Khi đó ta có n! cách sắp xếp dựa trên các phép so sánh nhị nguyên (tức là mỗi lần so sánh hai phần tử với nhau). Kết quả sau mỗi lần so sánh số phần tử cần sắp xếp thu hẹp dần lại.

Thuật toán sắp xếp một tập gồm n phần tử dựa trên so sánh nhị nguyên được biểu diễn bằng cây quyết định nhị phân với các đỉnh trong chứa các phép so sánh hai phần tử, còn đỉnh ngoài (lá) chứa một trong n! hoán vị của n phần tử.

Độ phức tạp của thuật toán sắp xếp dựa trên so sánh nhị nguyên tính bằng số các phép toán so sánh được sử dụng. Nó đòi hỏi ít nhất [logn!] phép so sánh.

Ví dụ 1: Cây quyết định nhị phân biểu diễn cách sắp xếp danh sách ba phần tử a, b, c có dạng:



5.2. Thuật toán sắp xếp kiểu nổi bọt

Sắp xếp kiểu nổi bọt bao gồm các thao tác cơ bản:

– Đổi chỗ phần tử lớn hơn với phần tử nhỏ hơn đi sau bắt đầu từ đầu danh sách và duyệt qua toàn bộ danh sách.

– Thủ tục trên lặp lại cho tới khi việc sắp xếp hoàn thành.

Chú ý: Ta hãy tưởng tượng khi các phần tử được đặt vào một cột thì sắp xếp kiểu nổi bọt các phần tử nhỏ hơn sẽ nổi lên trên, vì chúng đổi chỗ với các phần tử lớn hơn. Các phần tử lớn hơn sẽ "chìm" xuống đáy cột.

Thuật toán sắp xếp nổi bọt có độ phức tạp trong trường hợp xấu nhất là $O(n^2)$.

Thuật toán sắp xếp theo kiểu nổi bọt dưới dạng giả mã như sau:

```
Procedure bubblesort (a1, a2, ..., an)
For i := 1 to n – 1
    begin
        for j := 1 to n – i
            if aj > aj+1 then đổi chỗ aj và aj+1
    end   {a1, a2, ..., an được sắp theo thứ tự tăng dần}
```

Ví dụ 2: Hãy sắp xếp 3, 2, 4, 1, 5 theo thứ tự tăng dần bằng thuật toán sắp xếp kiểu nổi bọt và chỉ rõ các danh sách nhận được ở mỗi bước.

Giải:

– *Bước 1.* So sánh 3 và 2, vì 3 > 2 không đúng vị trí nên đổi chỗ 3 và 2 cho nhau ta được 2, 3, 4, 1, 5. Vì 3 < 4 đúng vị trí nên tiếp tục so sánh 4 với 1. Vì 4 > 1 không đúng vị trí nên đổi cho 4 và 1 ta được 2, 3, 1, 4, 5. Vì 4 < 5 đúng vị trí nên bước 1 được hoàn thành và 5 là phần tử lớn nhất được đặt đúng vị trí trong dãy.

Bước 1 được mô tả bằng sơ đồ cột với ký hiệu:

$\begin{matrix}x\\y\end{matrix}$: đổi chỗ x và y, $\begin{matrix}x\\y\end{matrix}$: đúng vị trí.

3	2	2	2
2	3	3	3
4	4	4	1
1	1	1	4
5	5	5	$\boxed{5}$

– *Bước 2*. So sánh 2 và 3, vì 2 < 3 đúng vị trí nên so sánh tiếp 3 và 1. Vì 3 > 1 không đúng vị trí nên đổi chỗ 3 và 1 ta được 2, 1, 3, 4, 5. So sánh 3 và 4. Vì 3 < 4 đúng vị trí nên so sánh tiếp 4 và 5. Vì 4 < 5 đúng vị trí nên bước 2 được hoàn thành và 4 được đặt đúng vị trí trong dãy.

Bước 2 được mô tả bằng sơ đồ cột:

2	2	2
3	3	1
1	1	3
4	4	4
$\boxed{5}$	$\boxed{5}$	$\boxed{5}$

– *Bước 3*. So sánh 2 và 1, vì 2 > 1 không đúng vị trí nên đổi chỗ chúng cho nhau ta được: 1, 2, 3, 4, 5. Vì 2 < 3 đúng vị trí nên không cần tiếp tục so sánh 3 với 4 và 4 với 5 nữa (vì chúng đã đúng vị trí).

Bước 3 được hoàn thành với 3 đặt đúng vị trí.

2	1
1	2
3	3
$\boxed{4}$	$\boxed{4}$
$\boxed{5}$	$\boxed{5}$

– *Bước 4*. So sánh 1 và 2, vì 1 < 2 đúng vị trí nên không phải so sánh 2 với 3 nữa, vì 2 đã đặt đúng vị trí. Bước 4 được hoàn thành.

Bước 4 được mô tả bằng sơ đồ cột là :

1
2
$\boxed{3}$
$\boxed{4}$
$\boxed{5}$

Ví dụ 3: Dùng thuật toán sắp xếp kiểu nổi bọt hãy sắp xếp danh sách 3, 1, 5, 7, 4 theo thứ tự tăng dần và chỉ rõ các danh sách nhận được ở mỗi bước.

Giải:

Tương tự như ví dụ 2 ta có các bước của thuật toán sắp xếp kiểu nổi bọt dưới dạng sơ đồ cột sau:

– *Bước 1*

3	1	1	1	1
1	3	3	3	3
5	5	5	5	5
7	7	7	7	4
4	4	4	4	$\boxed{7}$

– *Bước 2*

1	1	1	1
3	3	3	3
5	5	5	4
4	4	4	$\boxed{5}$
$\boxed{7}$	$\boxed{7}$	$\boxed{7}$	$\boxed{7}$

– *Bước 3*

1	1	1
3	3	3
4	4	$\boxed{4}$
$\boxed{5}$	$\boxed{5}$	$\boxed{5}$
$\boxed{7}$	$\boxed{7}$	$\boxed{7}$

– *Bước 4*

1
3
$\boxed{4}$
$\boxed{5}$
$\boxed{7}$

5.3. Thuật toán sắp xếp kiểu hoà nhập

Thuật toán này được thực hiện theo các bước sau đây:

– Phân đôi liên tiếp các danh sách thành hai danh sách con bằng nhau (hoặc hơn kém nhau một phần tử) cho tới khi mỗi danh sách con chỉ gồm một phần tử.

– Dãy các danh sách con được biểu diễn bằng cây nhị phân cân đối.

– Tiếp tục hoà nhập lần lượt các cặp danh sách đã có thứ tự tăng dần thành một danh sách lớn với các phần tử được sắp theo thứ tự tăng dần cho tới khi toàn bộ danh sách ban đầu được sắp theo thứ tự tăng dần.

– Dãy danh sách hoà nhập được biểu diễn bằng cây nhị phân cân đối (ngược với cây nhị phân trong bước một).

Số các phép so sánh cần thiết trong thuật toán sắp xếp kiểu hoà nhập một danh sách n phần tử là O(nlogn).

Thuật toán hoà nhập hai danh sách theo bảng dưới dạng giả mã như sau:

Procedure merge (L_1, L_2 : danh sách)
L := danh sách rỗng
While <cả L_1 và L_2 đều không rỗng>
Begin
Xoá phần tử nhỏ hơn trọng hai phần tử đầu của L_1 và L_2 của danh sách chứa nó và đặt nó vào cuối danh sách hoà nhập L.
if <việc xoá một phần tử làm cho một danh sách thành rỗng>
then xoá tất cả các phần tử khỏi danh sách kia và nối chúng vào cuối L
End
{L là danh sách hoà nhập với các phần tử được sắp theo thứ tự tăng dần}

Ví dụ 4: Sắp xếp kiểu hoà nhập theo bảng với $L_1 = \{2, 3, 5, 6\}$ và $L_2 = \{1, 4\}$. Hoà nhập L_1 và L_2 để được danh sách $L = \{1, 2, 3, 4, 5, 6\}$.

Giải:

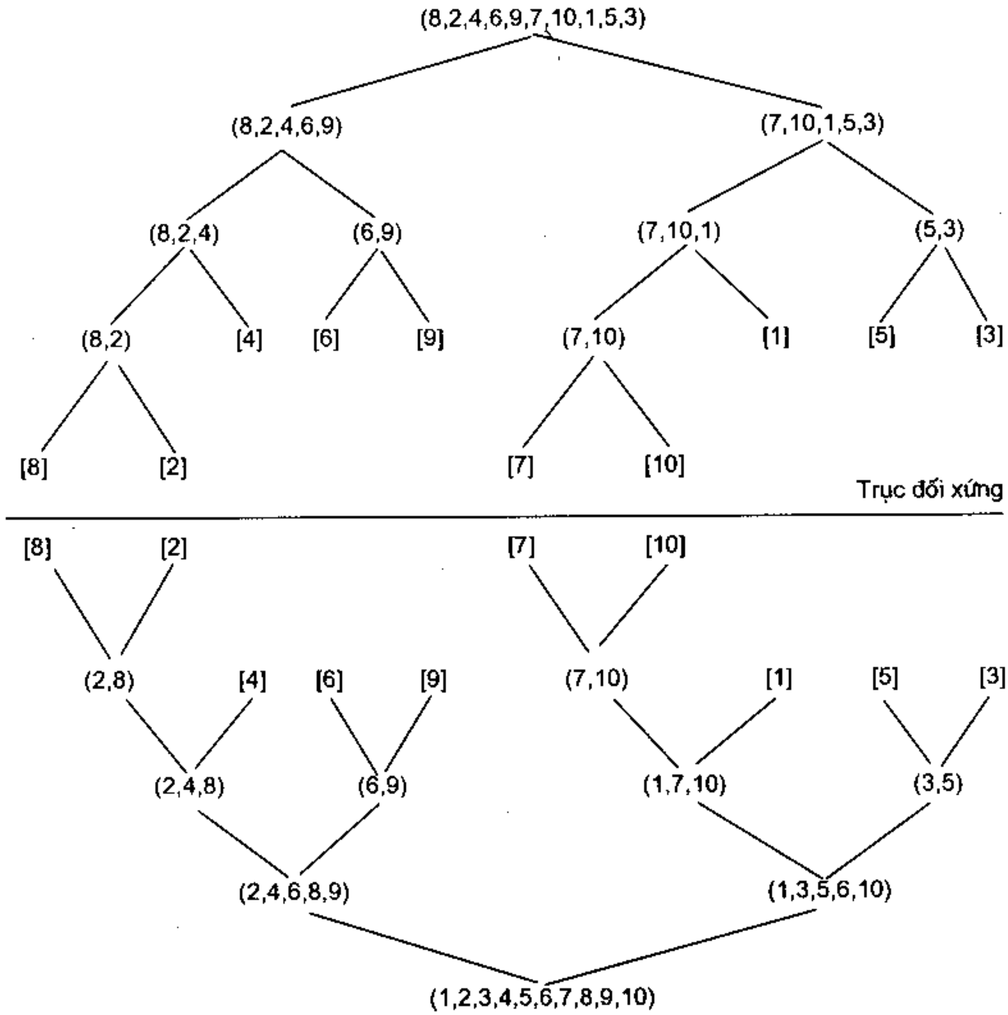
Kết quả áp dụng thuật toán trên đối với L_1 : 2, 3, 5, 6 và L_2 : 1, 4 được cho dưới bảng dưới đây:

Bảng hoà nhập danh sách L_1 : 2, 3, 5, 6 và L_2: 1, 4

Danh sách L_1	Danh sách L_2	Danh sách hoà nhập L	So sánh
2 3 5 6	1 4		1 < 2
2 3 5 6	4	1	2 < 4
3 5 6	4	1 2	3 < 4
5 6	4	1 2 3	4 < 5
5 6		1 2 3 4	
		1 2 3 4 5 6	

Ví dụ 5: Sắp xếp danh sách L: 8, 2, 4, 6, 9, 7, 10, 1, 5, 3 theo thứ tự tăng dần bằng cách chia đôi danh sách đã cho thành hai danh sách con có số phần tử bằng nhau hoặc hơn kém nhau một phần tử và biểu diễn các dãy con như vậy bởi cây nhị phân cân đối.

Giải: Gốc của cây nhị phân là L. Giả sử L được chia thành hai danh sách con là L_1 và L_2 với đỉnh con bên trái và đỉnh con bên phải tương ứng. Thủ tục đó dừng lại khi cây nhị phân có các lá là các phần tử trong danh sách L (|L| = số lá của cây nhị phân) lấy cây nhị phân đối ngẫu (đối xứng) với cây nhị phân xây dựng ở trên thì đỉnh gốc của cây nhị phân đối xứng chứa danh sách L được sắp theo thứ tự tăng dần (xem hình dưới). Ký hiệu (...) chỉ đỉnh trong, [...] chỉ lá.

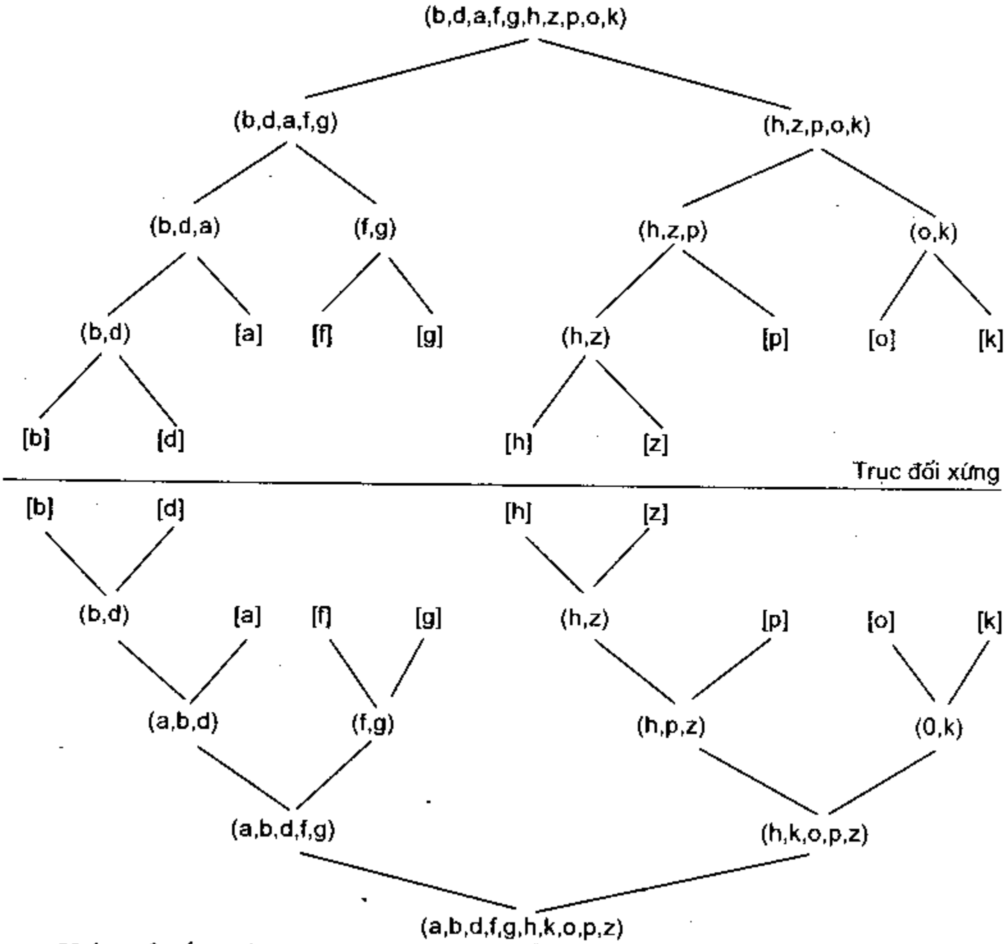


Kết quả L: 8, 2, 4, 6, 9, 7, 10, 1, 5, 3 cho ta danh sách được sắp là:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

Ví dụ 6: Tương tự ví dụ 5 sắp xếp danh sách L := b, d, a, f, g, h, z, p, o, k theo thứ tự từ điển.

Giải:



Kết quả sắp xếp L theo thứ thự từ điển ta được danh sách:

a, b, d, f, g, h, k, o, p, z.

Sau đây chúng ta xét một số thuật toán sắp xếp khác.

• ***Sắp xếp kiểu chọn lọc:*** Để sắp xếp dãy L := a_1, a_2, ..., a_n ta bắt đầu bằng việc chọn phần tử nhỏ nhất trong danh sách. Phần tử này được xếp lên đầu danh sách. Tiếp theo chọn phần tử nhỏ nhất trong danh sách còn lại và đặt phần tử đó sau phần tử nhỏ nhất ở bước trước. Thủ tục này được lặp lại cho đến khi toàn bộ danh sách được sắp xếp.

Áp dụng: Sắp xếp danh sách sau đây bằng thuật toán sắp xếp kiểu chọn lọc: 3, 5, 4, 1, 2.

Giải:

– *Bước 1:* Đặt 1 lên đầu danh sách ta được 1, 3, 5, 4, 2.

– *Bước 2:* Đặt 2 sau 1 ta được danh sách: 1, 2, 3, 5, 4.

– *Bước 3:* Vì 3 đã đứng sau 1, 2 nên danh sách ở bước 3 là: 1, 2, 3, 5, 4.

– *Bước 4:* Đặt 4 sau 1, 2, 3 ta được danh sách: 1, 2, 3, 4, 5.

Kết quả của bước 4 là kết quả cần tìm.

• ***Sắp xếp nhanh:*** Để sắp xếp L: $a_1, a_2, ..., a_n$, thuật toán bắt đầu bằng việc lấy ra phần tử a_1 và tạo ra hai danh sách L_1 gồm các phần tử nhỏ hơn a_1 theo thứ tự xuất hiện của chúng trong L; L_2 gồm các phần tử lớn hơn a_1 theo thứ tự xuất hiện của chúng trong L. Khi đó a_1 được đặt ở cuối danh sách L_1. Thủ tục này được lặp lại một cách đệ quy cho mỗi danh sách con cho đến khi mỗi danh sách chứa chỉ một phần tử theo thứ tự xuất hiện của chúng.

Áp dụng: Cho L : 3, 5, 7, 8, 1, 9, 2, 4, 6. Dùng thuật toán sắp xếp nhanh để sắp L theo thứ tự tăng dần.

Giải:

– *Bước 1:* Tập các phần tử nhỏ hơn 3 là L_1: 1, 2.
Tập các phần tử lớn hơn 3 là L_2 : 5, 7, 8, 9, 4, 6
Đặt 3 vào cuối danh sách L_1 được L_1 : 1, 2, 3.

– *Bước 2:* Đối với danh sách L_2: 5, 7, 8, 9, 4, 6 (trong bước 1)
Tập các phần tử nhỏ hơn 5 là L_1 : 4
Tập các phần tử lớn hơn 5 là L_2 : 7, 8, 9, 6.
Đặt 5 vào cuối danh sách L_1 được L_1 : 4, 5.

– *Bước 3:* Đối với danh sách L_2: 7, 8, 9, 6 (trong bước 2)
Tập các phần tử nhỏ hơn 7 là L_1 : 6
Tập các phần tử lớn hơn 7 là L_2 : 8, 9.
Đặt 7 vào cuối danh sách L_1 được L_1 : 6, 7.

– *Bước 4:* Đối với danh sách L_2: 8, 9 (trong bước 3)
Tập các phần tử nhỏ hơn 8 là L_1 : rỗng.
Tập các phần tử lớn hơn 8 là L_2 : 9.
Đặt 8 vào cuối danh sách L_1 được L_1 : 8.

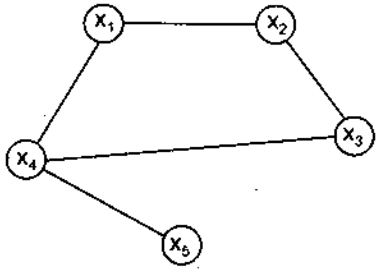
Vậy danh sách được sắp xếp theo thuật toán sắp xếp nhanh là: L : 1, 2, 3, 4, 5, 6, 7, 8, 9; ở đây: 1, 2, 3 do bước 1 tạo ra; 4, 5 do bước 2 tạo ra; 6, 7 do bước 3 tạo ra; còn 8, 9 do bước 4 tạo ra.

§6. CÂY KHUNG CỦA ĐỒ THỊ

6.1. Cây khung của đồ thị không có trọng số

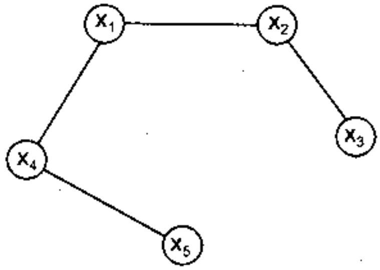
Cho G = <X, U> là một đồ thị vô hướng. Nếu đồ thị bộ phận G' = <X, U'> của G = <X, U> (U' ⊆ U) là một cây, tức là G' liên thông không có chu trình thì ta nói G' = <X, U> là cây khung của đồ thị G = <X, U>.

Ví dụ 1: Cho G = <X, U> có dạng

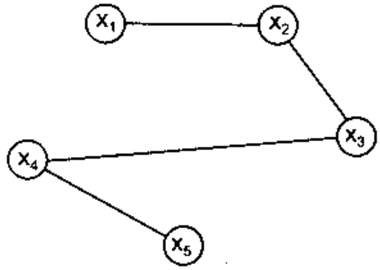


là đồ thị liên thông, nhưng không phải là một cây vì nó có chu trình.

– Nếu bỏ cạnh nối giữa x_4 và x_3 thì ta được đồ thị bộ phận G' = <X, U'> là



– Nếu bỏ cạnh nối giữa x_1 và x_4 thì ta được đồ thị bộ phận G" = <X, U"> là



G' là một cây vì G' liên thông và không có chu trình, theo định nghĩa G' là cây khung của G. Một đồ thị có thể có nhiều cây khung, ví dụ G" là cây khung khác của G.

Định lý 7: Đồ thị vô hướng G = <X, U> với $|X| = n \geq 2$ có cây khung khi và chỉ khi nó liên thông.

Chứng minh:

Điều kiện cần: Giả sử G có cây khung G'. Ta chỉ ra G là liên thông. Thật vậy, nếu G không liên thông thì tồn tại cặp đỉnh x, y mà giữa chúng không được nối bởi cạnh nào; mà x, y cũng là các đỉnh của G'. Chứng tỏ G' không liên thông, trái với giả thiết G' là một cây.

Điều kiện đủ: Giả sử G là đồ thị liên thông.

– Nếu trong G không có chu trình thì theo định nghĩa G là một cây, đó là cây khung.

– Nếu trong G có chu trình thì ta bỏ đi một cạnh trong chu trình đó ta được G' là liên thông và không có chu trình.

Kết quả dưới đây cho thấy số lượng cây khung của một đồ thị là rất lớn.

Định lý 8 (Cayley). Số cây khung của đồ thị K_n ($n \geq 2$) là n^{n-2}.

Dưới đây ta xét hai thuật toán tìm kiếm cây khung của đồ thị không trọng số.

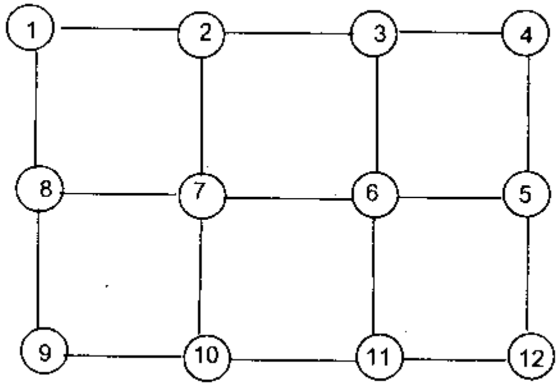
a) Thuật toán 1 (Thuật toán ưu tiên chiều sâu)

Cho G = <X, U> là một đồ thị vô hướng liên thông. Thuật toán ưu tiên chiều sâu tìm cây khung của G = <X, U> thực hiện theo các bước sau đây:

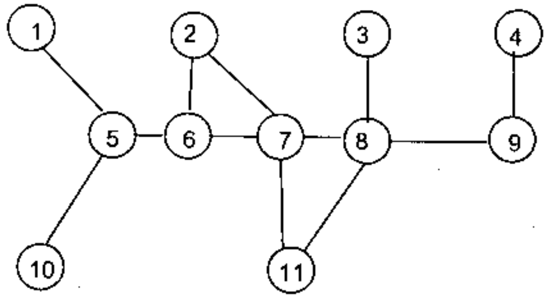
- Chọn tùy ý một đỉnh trong G = <X, U> làm gốc của cây.
- Xây dựng đường đi từ đỉnh này bằng cách lần lượt ghép các cạnh vào, sao cho mỗi cạnh mới ghép sẽ nối đỉnh cuối cùng trên đường đi với một đỉnh còn chưa thuộc đường đi. Thủ tục này chỉ dừng lại khi không thể ghép thêm vào bất kỳ cạnh khác được nữa, vì nếu ghép sẽ tạo ra chu trình.
- Nếu cây được tạo ra bởi các bước trên có tập đỉnh là tập đỉnh X của G thì cây đó là cây khung của đồ thị G.
- Trường hợp ngược lại, lùi lại một đỉnh trước đỉnh dừng lại cuối cùng của đường đi và nếu có thể được thì xây dựng đường đi mới xuất phát từ đỉnh này đi qua các đỉnh còn chưa thuộc đường đi. Nếu điều đó không thể làm được thì lùi tiếp lại một đỉnh nữa và tại điểm này lại xây dựng đường đi mới.
- Các thủ tục trên sau một số bước sẽ dừng lại và cho ta cây khung của G = <X, U>.

Ví dụ 2: Dùng thuật toán ưu tiên chiều xâu, tìm cây khung của các đồ thị sau:

G_1:



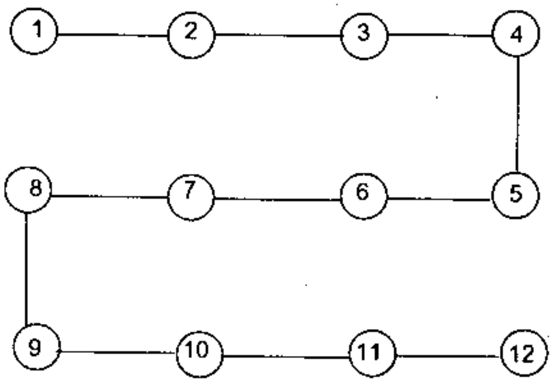
và G_2:



Giải:

- Với G_1: Xuất phát từ đỉnh 1 đi đến đỉnh 12 và dừng lại ta được cây khung của G_1 có dạng:

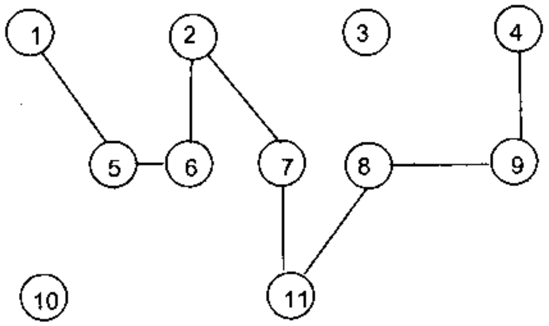
G_1:



Cây khung này thực hiện đúng một bước theo thuật toán ưu tiên chiều sâu.

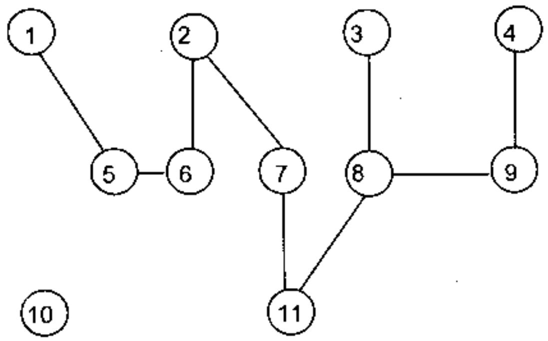
- Với G_2: Chẳng hạn xuất phát từ đỉnh 1 ta có các cây tạo ra của từng bước như sau:

Bước 1.

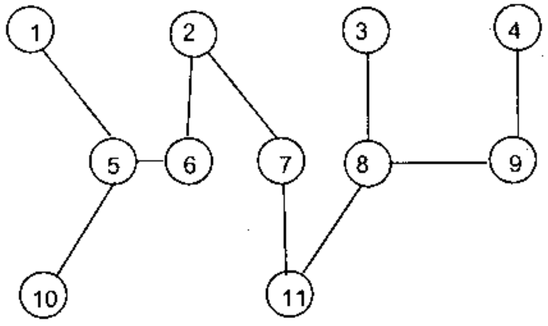


Cây T_1 chưa phải là cây khung vì còn hai đỉnh 3 và 10 nằm ngoài.

Bước 2. Ta đang ở đỉnh 4, lùi lại đỉnh 9 không tạo ra cây mới. Lùi tiếp về 8 và vẽ cạnh nối với đỉnh 3 ta được T_2 nhưng T_2 vẫn chưa là cây khung do còn điểm 10 vẫn nằm ngoài T_2



Bước 3..Ta đang ở đỉnh 3 trong T_2. Lùi từ 3 về 8, lùi từ 8 về 11, lùi từ 11 về 7, lùi từ 7 về 2, lùi từ 2 về 6 và lùi từ 6 về 5. Đến đây ghép cạnh (5, 10) vào cây T_2 ta được cây khung của G_2 có dạng:



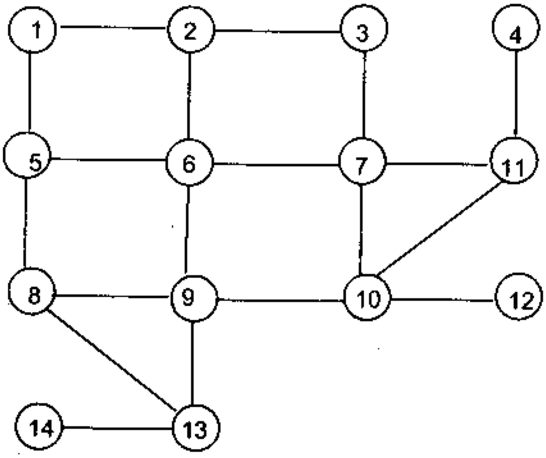
b) Thuật toán 2 (Thuật toán ưu tiên chiều rộng)

Cho G = <X, U> là một đồ thị vô hướng liên thông. Thuật toán ưu tiên chiều rộng tìm cây khung của G = <X, U> thực hiện theo các bước sau đây:

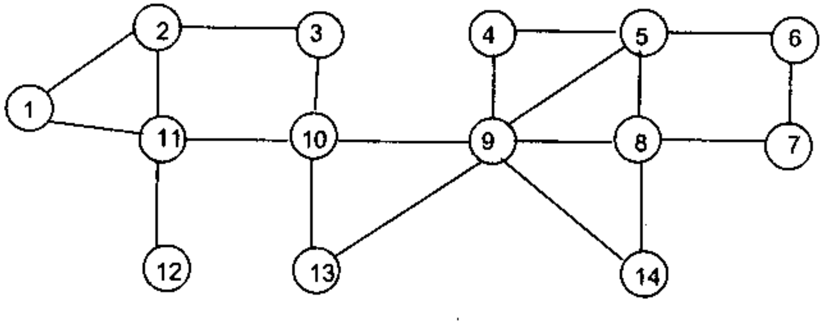
- Chọn một đỉnh bất kỳ trong đồ thị vô hướng liên thông G = <X, U> làm gốc của cây khung, gốc gán mức 0.
- Ghép tất cả các cạnh liên thuộc với đỉnh gốc sao cho các đỉnh kề với đỉnh gốc nằm ở mức 1 với thứ tự tùy ý.
- Tiếp tục với mỗi đỉnh ở mức 1, ta ghép các cạnh liên thuộc với chúng sao cho các đỉnh kề với các đỉnh có mức 1 không tạo ra chu trình và được nằm ở mức 2 của cây với thứ tự tùy ý.
- Quá trình này dừng lại sau một số bước làm việc (do tính hữu hạn của đồ thị) và tất cả các đỉnh của đồ thị G = <X, U> đã được ghép vào cây.

Ví dụ 3: Dùng thuật toán ưu tiên chiều rộng tìm cây khung của các đồ thị sau đây:

a) G_1:



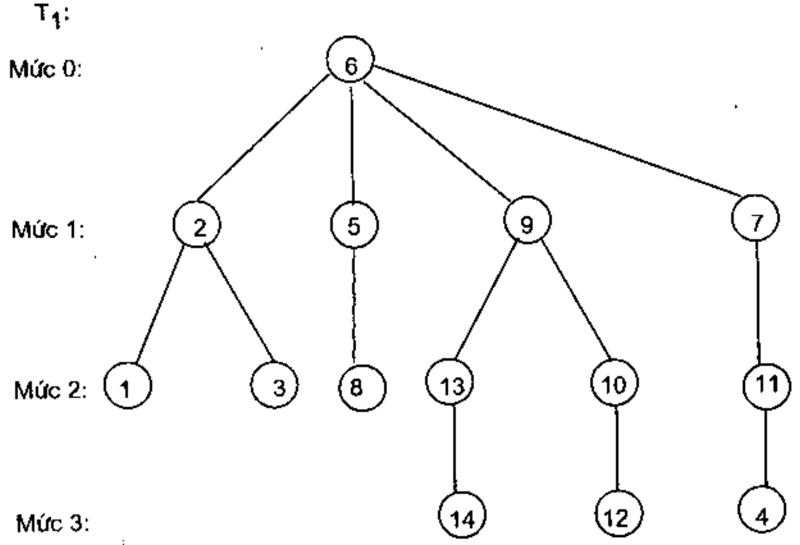
b) G_2:



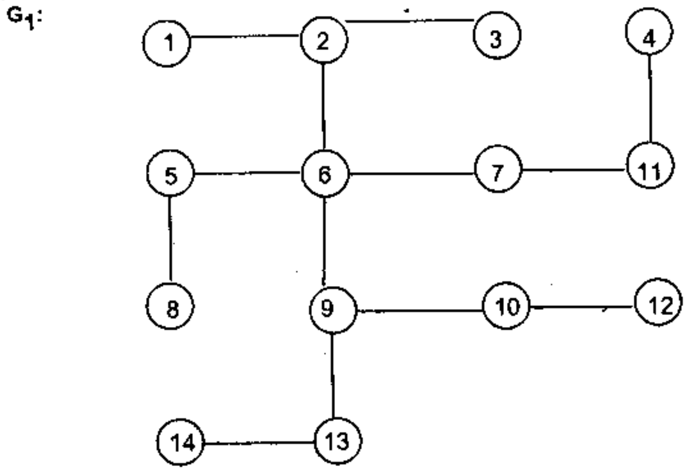
Giải:

a) Chọn đỉnh 6 làm gốc (gán nhãn 0). Ghép các cạnh liền kề với 6 và các đỉnh 2, 5, 7, 9 nằm trên mức 1. Xét các đỉnh trên mức 1 thì các đỉnh 1, 3, 8, 13, 10, 11 có mức 2. Đối với các đỉnh có mức 2 thì các đỉnh nằm trên mức 3 là 14, 12, 4.

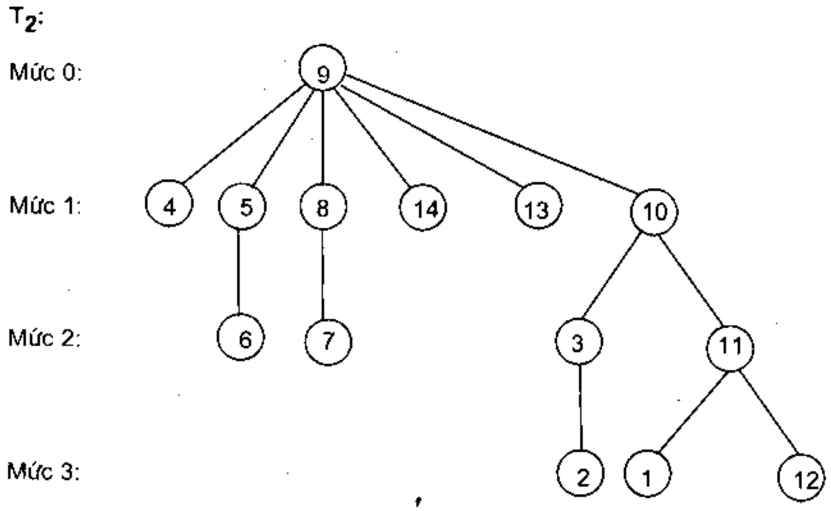
Cây có gốc T_1 dưới đây chứa đầy đủ các đỉnh của G_1 có dạng:



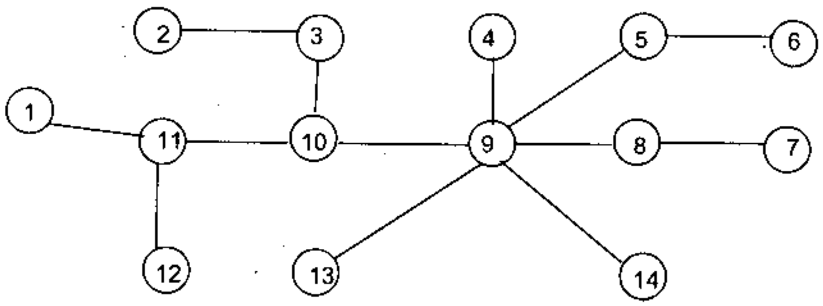
T_1 chính là cây khung của G_1. Hay T_1 viết dưới dạng có cấu trúc vị trí đỉnh của G_1 là



b) Chọn 9 làm gốc. Tương tự như câu a ta có cây khung của G_2 là T_2 có dạng:



Hay T_2 có cấu trúc vị trí đỉnh của G_2 là:

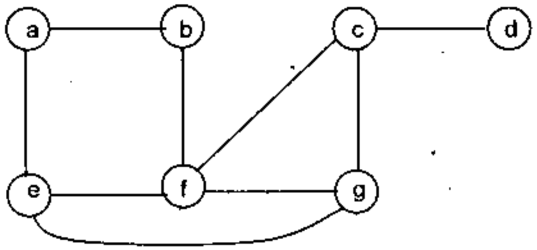


c) Thuật toán 3 (Xoá cạnh tạo ra chu trình)

Cho G = <X, U> là đơn đồ thị liên thông. Thuật toán xoá cạnh tạo ra chu trình thực hiện theo các bước sau đây:

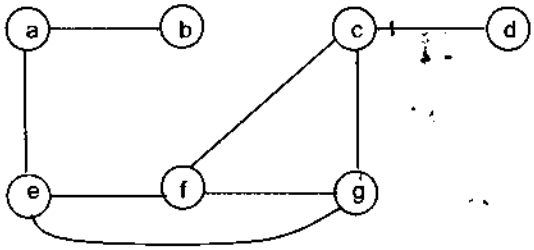
- Nếu G = <X, U> không có chu trình thì G = <X, U> là cây khung.
- Nếu G = <X, U> có chu trình thì bỏ đi một cạnh của chu trình trong đồ thị đó. Thủ tục bỏ đi các cạnh trong chu trình chỉ dừng lại khi đồ thị nhận được không còn chu trình nữa. Do đồ thị có hữu hạn cạnh nên sau một số hữu hạn bước thuật toán dừng và cho cây khung của đồ thị G = <X, U>.

Ví dụ 4: Cho đồ thị G = <X, U> dưới dạng

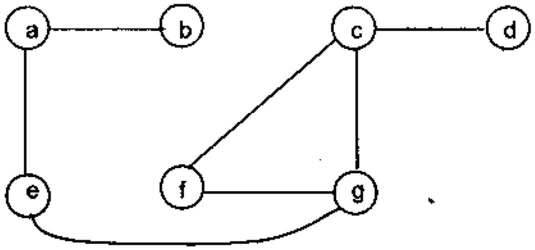


Tìm cây khung của đồ thị trên bằng cách xoá cạnh tạo ra chu trình.

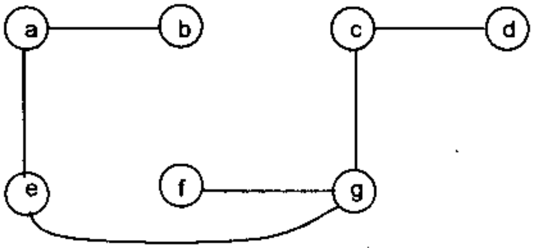
Giải: Trước hết khẳng định G có cây khung vì G liên thông. Bỏ cạnh (b, f) ta bỏ được một chu trình và nhận được G_1 dưới dạng



Xoá tiếp cạnh (e, f) ta được G_2 dưới dạng



Xoá tiếp cạnh (f, c) ta được G_3 dưới dạng

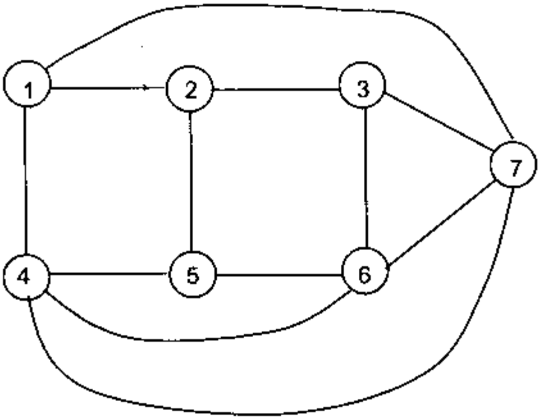


Đây là cây khung của đồ thị G.

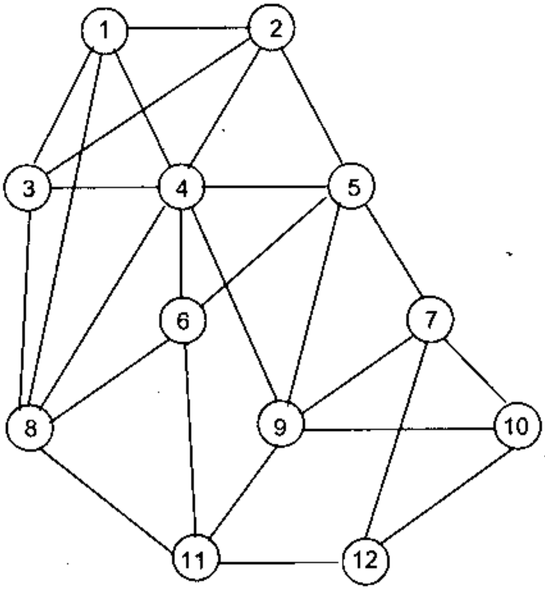
Chú ý: Nếu trong G_2 không xoá cạnh (f, c) mà xoá cạnh (f, g) hoặc (c, g) thì ta có các cây khung khác ngoài cây khung G_3.

Ví dụ 5: Trong các đồ thị dưới đây, hãy tìm cây khung của nó bằng cách xoá các cạnh trong các chu trình của đồ thị.

a) G_1:



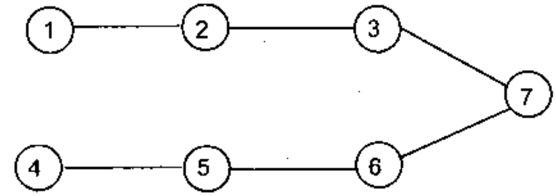
b) G_2:



Giải:

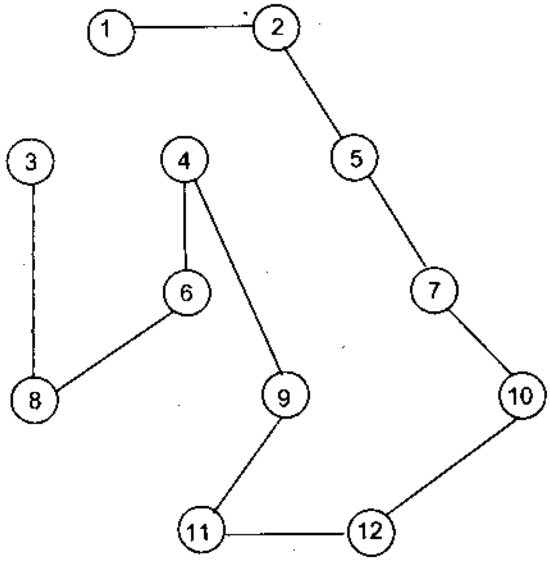
a) Như cách giải của ví dụ 4, chọn gốc là đỉnh 1 ta có cây khung T_1 của G_1 là:

T_1:



b) Cây khung T_2 của G_2 với gốc ta chọn là đỉnh 1 có dạng:

T_2:



Chú ý: Thuật toán 1 và 2 tạo ra cây khung của đồ thị dưới dạng cây có gốc. Thuật toán ưu tiên chiều sâu càng nhiều cạnh càng tốt trong mỗi bước thực hiện của thuật toán. Còn thuật toán ưu tiên chiều rộng thì cây có mức, mức càng nhiều đỉnh càng tốt.

6.2. Cây khung của đồ thị có trọng số

Trong thực tế có nhiều bài toán tìm cây khung bé nhất để giảm chi phí xây dựng và vận chuyển (đối với bài toán giao thông); giảm chi phí xây dựng và thiết kế điện năng (đối với nơi cấp điện và nơi tiêu thụ điện). Chẳng hạn, có một nguồn điện cần cung cấp cho n người tiêu thụ. Hãy tìm cách mắc điện để cho tổng độ dài dây dẫn là bé nhất. Giải bài toán này thực chất là tìm cây khung bé nhất giữa nơi cấp điện và nơi tiêu thụ.

Bài toán tìm cây khung bé nhất:

Cho G = <X, U> là một đồ thị liên thông có trọng số (không âm). Theo định lý 8 (Cayley) thì G có thể có nhiều cây khung.

Trọng số của đồ thị G là $l(G) = \sum_{u \in X} l(u)$, ở đây $l(u)$ là trọng số của cạnh u.

Khi đó *cây khung bé nhất* của G = <X, U> là cây khung có trọng số bé nhất trong tất cả các cây khung của đồ thị G = <X, U>.

Dưới đây trình bày hai thuật toán tìm cây khung bé nhất của đồ thị có trọng số liên thông.

a) Thuật toán 1 (PRIM)

Bài toán:

Đầu vào: $G = <X, U>$ có trọng số và liên thông với $|X| = n$ đỉnh;

Đầu ra: Cây khung G_o có trọng số bé nhất.

Thuật toán:

- Chọn một cạnh bất kỳ có trọng số bé nhất trong U, chẳng hạn đó là cạnh u_1, đặt $U_1 = \{u_1\}$.
- Chọn cạnh có trọng số bé nhất trong số $U \setminus U_1$ các cạnh liền kề với u_1 không tạo ra chu trình với U_1, chẳng hạn đó là u_2, đặt $U_2 = U_1 \cup \{u_2\} = \{u_1, u_2\}$.
- Chọn cạnh có trọng số bé nhất trong các cạnh $U \setminus U_2$ liền kề với các cạnh trong U_2 mà không tạo ra chu trình, chẳng hạn đó là u_3, đặt $U_3 = U_2 \cup \{u_3\} = \{u_1, u_2, u_3\}$.

...

Thủ tục trên dừng lại khi đã chọn đủ n – 1 cạnh, tức là ở bước thứ n – 1 xây dựng được tập $U_{n-1} = U_{n-2} \cup \{u_{n-1}\} = \{u_1, u_2, u_3, ..., u_{n-1}\}$, ở đây cạnh u_{n-1} là cạnh có trọng số bé nhất trong số các cạnh $U \setminus U_{n-2}$ liền kề và không tạo ra chu trình với các cạnh được chọn.

Định lý 9: Cây khung tạo ra theo thuật toán 1 (thuật toán PRIM) là cây khung bé nhất của đồ thị liên thông có trọng số $G = <X, U>$ với $|X| = n$ đỉnh.

Chứng minh:

Trước hết từ cách xây dựng thuật toán trên ta có $G' = <X, U_{n-1}>$ là cây khung của G (do tính chất 2, định lý 1). Ta chỉ ra G' là cây khung bé nhất. Thật vậy, giả sử $G'' = <X, U''>$ là cây khung bé nhất của G, tức là $l(G'') = \min\{l(G') : G' \in \boldsymbol{G}$, ở đây $\boldsymbol{G}$ là tập cây khung của G$\}$. Ta sẽ dẫn đến mâu thuẫn là tồn tại cây khung $G''' = <X, W>$ bé nhất $\neq G''$. Thật vậy, vì G'' là cây khung bé nhất nên $U'' \neq U_{n-1}$. Giả sử $u_k \in U_{n-1}$ nhưng $u_k \notin U''$. Khi đó trong đồ thị $<X, U'' \cup \{u_k\}>$ sẽ xuất hiện chu trình duy nhất (do tính chất 4, định lý 1). Giả sử ω là chu trình đó trong $<X, U'' \cup \{u_k\}>$. Vì trong $G' = <X, U_{n-1}>$ không có chu trình, mà $u_k \in U_{n-1}$ nên ∃ cạnh $u_o \notin U_{n-1}$ (u_k là cạnh trong ω). Xét đồ thị G''' = <X, W>, ở đây $W = U'' \cup \{u_k\} \setminus \{u_0\}$. Rõ ràng G''' không có chu trình (vì ω là chu trình duy nhất trong $<X, U'' \cup \{u_k\}>$). Do G'' = <X, U''> là cây khung nên |U''| = n – 1 (do tính chất 2 trong định lý 1). Vậy |W| = n – 1. Từ đó ta có G''' là cây khung trong G. Mặt khác $u_0 \notin U_{n-1}$ nên $l(u_0) > l(u_k)$, $u_k \in U_{n-1}$.

Ta có:

$$l(G''') = \sum_{u \in W} l(u) = \sum_{u \in U''} l(u) + \underbrace{l(u_k) - l(u_0)}_{<0} < \sum_{u \in U''} l(u) = l(G'')$$

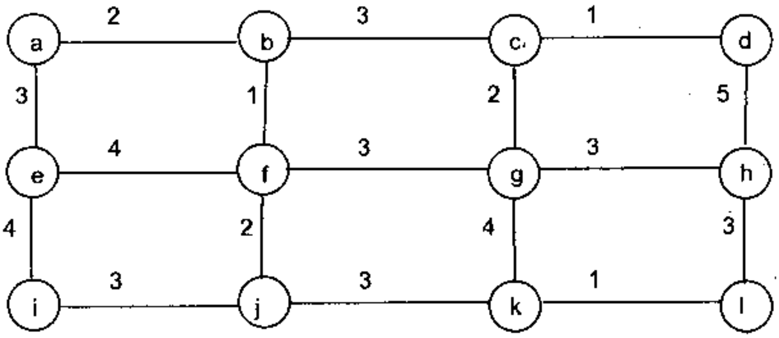
Trái với giả thiết G'' là cây khung bé nhất.

Định lý được chứng minh.

Thuật toán Prim tìm cây khung bé nhất của đồ thị liên thông có trọng số với n đỉnh bằng giả mã có dạng:

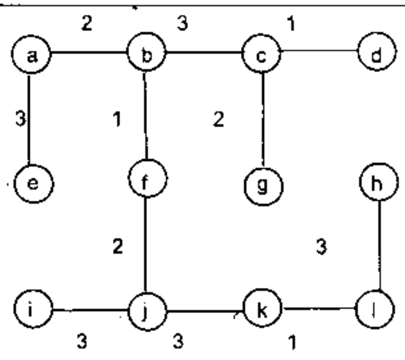
```
Procedure Prim (G: đồ thị liên thông có trọng số với n đỉnh)
T := cạnh có trọng số bé nhất.
For i := 1 to n – 1
begin
        u :=  cạnh có trọng số bé nhất liên thuộc với một đỉnh trong T
              và không tạo ra chu trình trong T nếu ghép nó vào T.
        T := T với u được ghép vào.
end     {T là cây khung bé nhất của G}.
```

Ví dụ 6: Cho đồ thị G = <X, U> dưới dạng

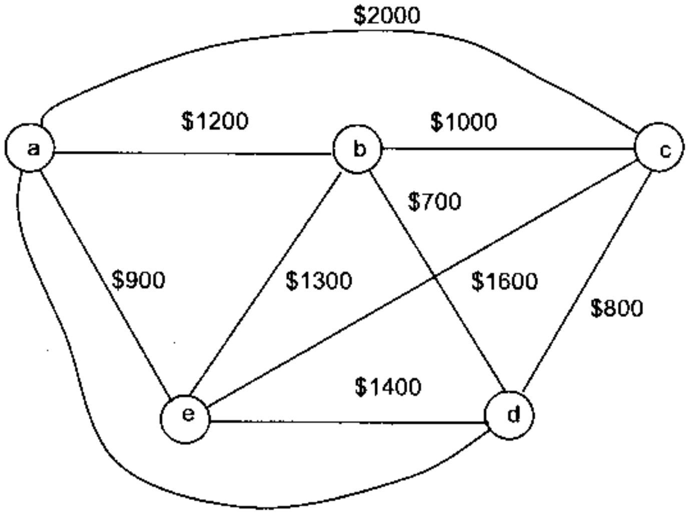


Tìm cây khung bé nhất theo thuật toán Prim.

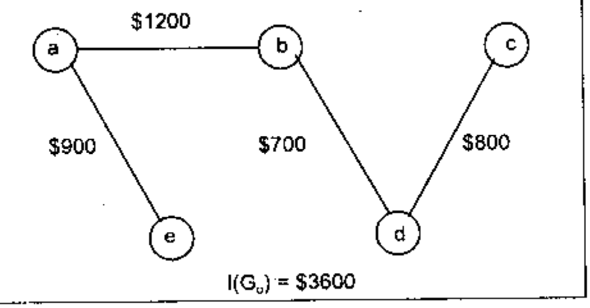
Giải: Ta trình bày thuật toán Prim theo dạng bảng.

Bước chọn	Cạnh chọn	Trọng số cạnh chọn	Cây khung bé nhất G_o
1	(c, d)	1	
2	(c, g)	2	
3	(c, b)	3	
4	(b, f)	1	
5	(b, a)	2	
6	(f, j)	2	
7	(j, k)	3	
8	(k, l)	1	
9	(j, i)	3	$l(G_o) = 24.$
10	(a, e)	3	
11	(l, h)	3	

Ví dụ 7: Dùng thuật toán Prim thiết kế một mạng truyền thông có giá trị tối thiểu để nối các trung tâm máy tính được biểu diễn bởi đồ thị với đỉnh là trung tâm máy tính, còn cạnh nối giữa hai đỉnh là đường truyền thông nối hai trung tâm máy tính tương ứng với hai đỉnh đó và trọng số của cạnh là tiền thuê bao phải trả hàng tháng.



Giải:

Bước chọn	Cạnh chọn	Trọng số cạnh chọn	Cây khung bé nhất G_o
1	(b, d)	\$700	
2	(d, c)	\$800	
3	(b, a)	\$1200	
4	(e, a)	\$900	

b) Thuật toán 2 (KRUSKAL)

Bài toán:

Đầu vào: G = <X, U> liên thông có trọng số với |X| = n đỉnh;

Đầu ra: G_o = <X, U'> cây khung có trọng số bé nhất.

Thuật toán:

- Chọn cạnh có cùng trọng số bé nhất của đồ thị G = <X, U> làm cạnh của cây khung, miễn là các cạnh đó không tạo ra chu trình.
- Ghép thêm các cạnh có cùng trọng số bé nhất trong số các cạnh còn lại vào cây khung, miễn là chúng không tạo ra chu trình với các cạnh đã cho.
- Thủ tục ghép như vậy sẽ dừng lại nếu n – 1 cạnh đã được chọn.

Định lý 10: Cây khung tạo ra theo thuật toán 2 (thuật toán KRUSKAL) là cây khung bé nhất của đồ thị liên thông có trọng số G = <X, U> với |X| = n đỉnh.

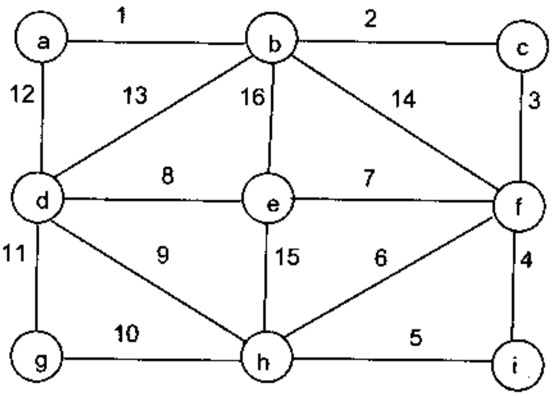
(Bạn đọc tự chứng minh)

Thuật toán Kruskal dưới dạng giả mã như sau:

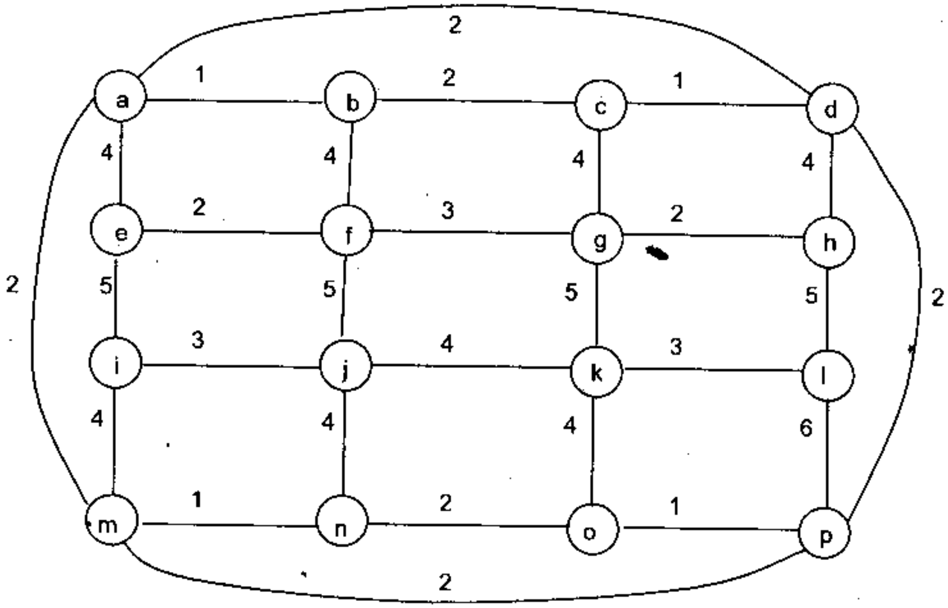
```
Procedure Kruskal (G: đồ thị n đỉnh, liên thông có trọng số)
T := đồ thị rỗng
For i := 1 to n – 1
begin
        u :=  một cạnh bất kỳ của G với trọng số bé nhất và không tạo
              ra chu trình khi ghép nó vào T.
        T := T với cạnh u đã được ghép thêm vào
end {T là cạnh khung bé nhất}
```

Ví dụ 8: Dùng thuật toán:

a) Kruskal đối với G_1



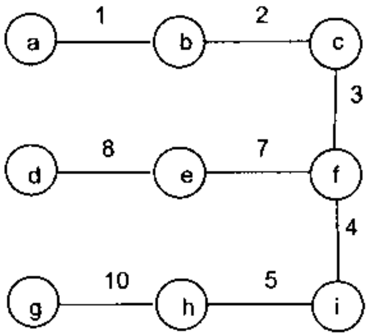
b) Prim và Kruskal đối với G_2



Giải:

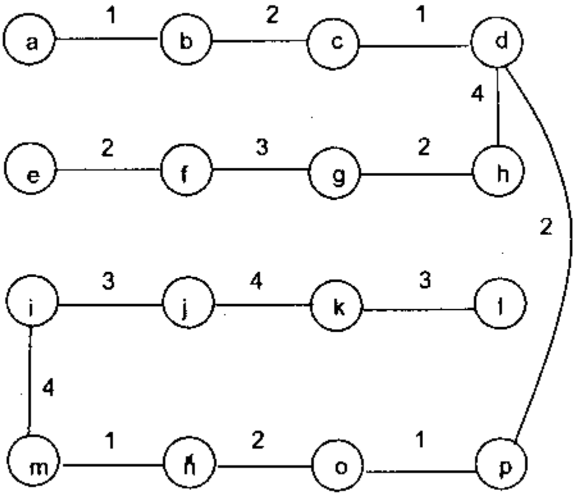
a) Kết quả áp dụng thuật toán Kruskal đối với G_1 để tìm cây khung bé nhất cho ở bảng 1 dưới đây:

Bảng 1

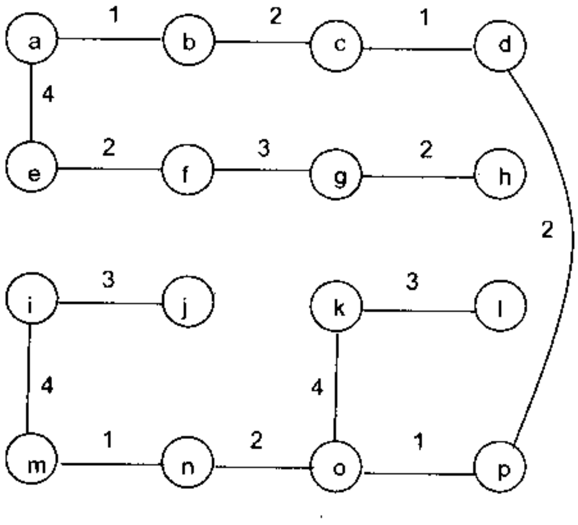
Bước chọn	Cạnh chọn	Trọng số cạnh chọn	Cây khung bé nhất G_o
1	(a, b)	1	
2	(b, c)	2	
3	(c, f)	3	
4	(f, i)	4	
5	(i, h)	5	
6	(f, e)	7	
7	(e, d)	8	
8	(h, g	10	$l(G_o) = 40$

b) Kết quả áp dụng thuật toán Prim đối với G_2 (xem bảng 2) và áp dụng thuật toán Kruscal đối với G_2 (xem bảng 3) dưới đây:

Bảng 2

Bước chọn	Cạnh chọn	Trọng số cạnh chọn	Cây khung bé nhất G_o
1	(a, b)	1	
2	(b, c)	2	
3	(c, d)	1	
4	(d, p)	2	
5	(p, o)	1	
6	(o, n)	2	
7	(n, m)	1	
8	(m, i)	4	
9	(d, h)	4	
10	(h, g)	2	
11	(g, f)	3	$l(G_o) = 35$
12	(f, e)	2	
13	(i, j)	3	
14	(j, k)	4	
15	(k, e)	3	

Bảng 3

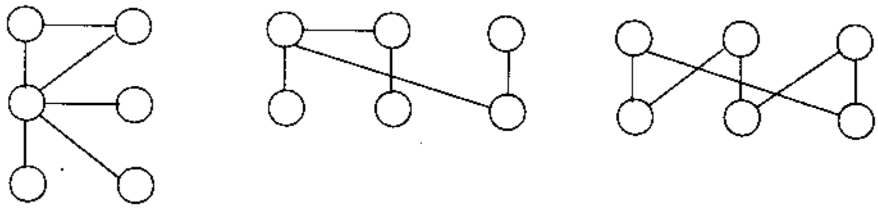
Bước chọn	Cạnh chọn	Trọng số cạnh chọn	Cây khung bé nhất G_o
1	(a, b)	1	
	(c, d)	1	
	(m, n)	1	
	(o, p)	1	
2	(e, f)	2	
	(g, h)	2	
	(b, c)	2	
	(n, o)	2	
	(d, p)	2	
3	(f, g)	3	
	(i, j)	3	
	(k, e)	3	
4	(m, i)	4	
	(k, o)	4	
	(a, e)	4	$l(G'_o) = 35$

Chú ý: Nếu đồ thị G = <X, U> liên thông và có trọng số các cạnh khác nhau từng cặp thì G = <X, U> có duy nhất một cây khung bé nhất. Trong trường hợp này các bước làm việc của thuật toán PRIM và thuật toán KRUSKAL là như nhau.

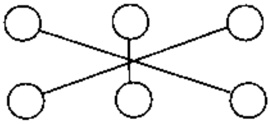
Trường hợp ngược lại, G = <X, U> trọng số các cạnh không nhất thiết khác nhau từng cặp thì G = <X, U> có thể có nhiều cây khung khác nhau về cấu trúc nhưng cùng một trọng số bé nhất.

BÀI TẬP

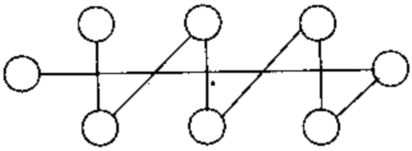
1. Đồ thị nào trong các đồ thị dưới đây là cây?



G_1 G_2 G_3

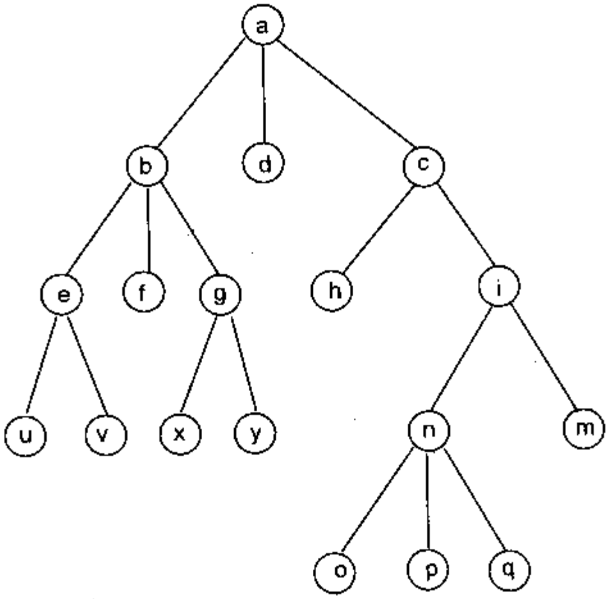


G_4



G_5

2. Cho cây có gốc G = <X, U>:



Trong cây trên:

a) Đỉnh nào là đỉnh gốc?

b) Đỉnh nào là đỉnh trong?

c) Đỉnh nào là lá?

d) Chiều cao của G bằng bao nhiêu?

e) Mức bé nhất (lớn nhất) của G bằng bao nhiêu và gồm những đỉnh nào?

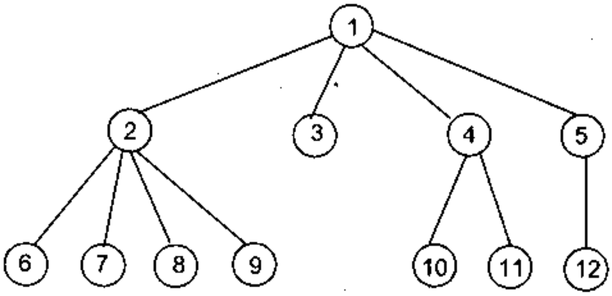
f) Cây G được gọi là cây 3 – phân có đúng không? Vì sao?

g) Muốn nhận được cây nhị phân đầy đủ (cây nhị phân cân đối) từ cây G = <X, U> ở trên cần phải bỏ đi những đỉnh nào? Và chiều cao của cây nhị phân đầy đủ, nhị phân cân đối là bao nhiêu?

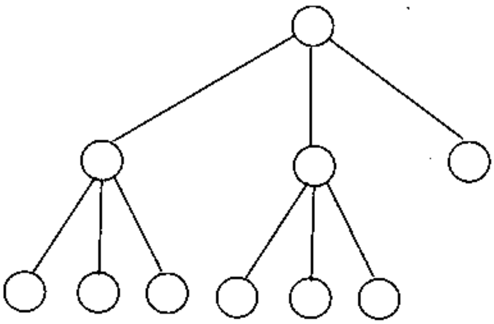
h) Chỉ ra các cây con của đỉnh a.

3. Trong các đồ thị phân đôi đầy đủ $K_{n,m}$, với n, m như thế nào thì $K_{n,m}$ là cây.

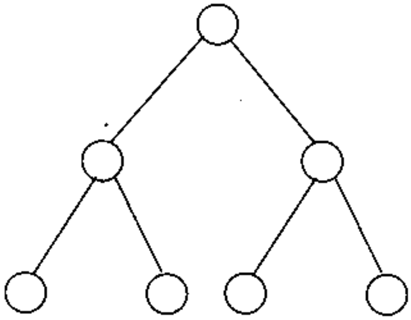
4. Cho các cây có gốc G_1, G_2 và G_3 dưới đây



G_1



G_2



G_3

a) Gọi tên G_1, G_2 và G_3.

b) Tìm mức của mỗi đỉnh trong G_1.

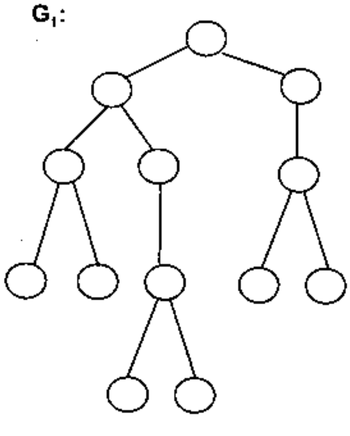
5. Cây ngũ phân đầy đủ G với 1000 đỉnh trong. Hỏi G có bao nhiêu đỉnh?

6. Cây nhị phân đầy đủ với 1000 đỉnh trong có bao nhiêu cạnh? Tìm chiều cao của cây.

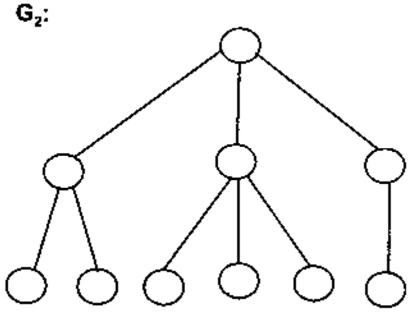
7. Hoặc là vẽ cây m – phân đầy dủ với 84 lá và có chiều cao bằng 3 hoặc chỉ ra không tồn tại cây như vậy.

8. Cây nào trong các cây dưới đây là cân đối:

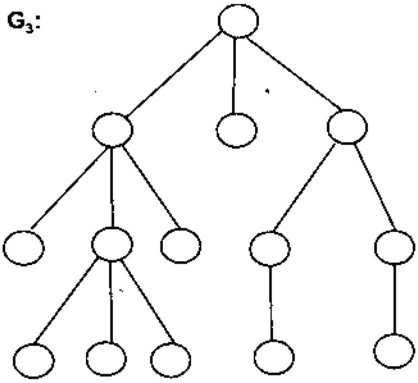
G_1:



G_2:



G_3:



9. Cho cây m – phân đầy đủ T với 81 lá và có chiều cao bằng 4.

a) Tìm cận trên và cận dưới của m.

b) Giá trị m là bao nhiêu nếu T là cây cân đối.

10. Cây m – phân hoàn toàn với chiều cao bằng h có bao nhiêu đỉnh và bao nhiêu lá.

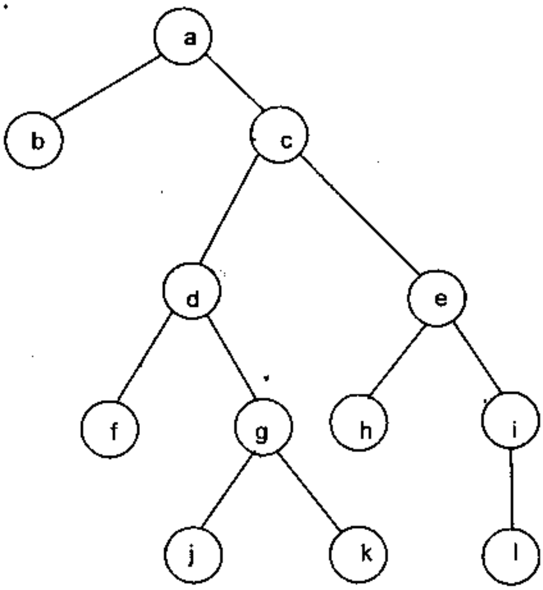
11. Vẽ cây nhị phân hoàn toàn có 15 đỉnh biểu thị mạng kết nối cây có 15 bộ xử lý.

12. Một rừng có k cây với tổng số n đỉnh thì có bao nhiêu cạnh?

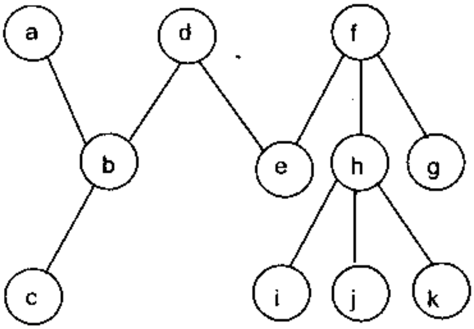
13. Tâm sai của một đỉnh trong cây không gốc là độ dài của đường đi đơn dài nhất bắt đầu từ đỉnh này.

Một đỉnh gọi là tâm nếu không có đỉnh nào trong cây có tâm sai nhỏ hơn tâm sai của đỉnh này.

Cho cây T_1:



và T_2:



Tìm mọi đỉnh là tâm trong các cây T_1, T_2 ở trên.

14. Chứng tỏ rằng một cây hoặc có 1 tâm hoặc có 2 tâm liền kề nhau.

15. Xây dựng cây tìm kiếm nhị phân cho các từ banana, peach, apple, pear, coconut, mango và papaya theo thứ tự từ điển.

16. Xây dựng cây tìm kiếm nhị phân cho các từ: oenology, phrenology, campanology, ornithology, ichthyology, limnology, alchemy và astrology theo thứ tự từ điển.

17. Xây dựng cây nhị phân với mã tiền tố biểu diễn các lược đồ mã hoá sau đây:

a) a : 11, e : 0, t : 101, s : 100;

b) a : 1, e : 01, t : 001, s : 0001, n : 00001.

18. Xác định cái nào là mã tiền tố trong các sơ đồ mã sau đây:

a) a : 11, e : 00, t : 10, s : 01;

b) a : 0, e : 1, t : 01, s : 001.

19. Xác định cái nào là mã tiền tố trong các sơ đồ mã sau:

a) a : 101, e : 11, t : 001, s : 011, n : 010;

b) a : 010, e : 11, t : 011, s : 1011, n : 1001, i : 10101

20. Dựng cây nhị phân với mã tiền tố biểu diễn các lược đồ mã hoá sau:

a : 1010, e : 0, t : 11, s : 1011, n : 1001, i : 100001.

21. Cho sơ đồ mã

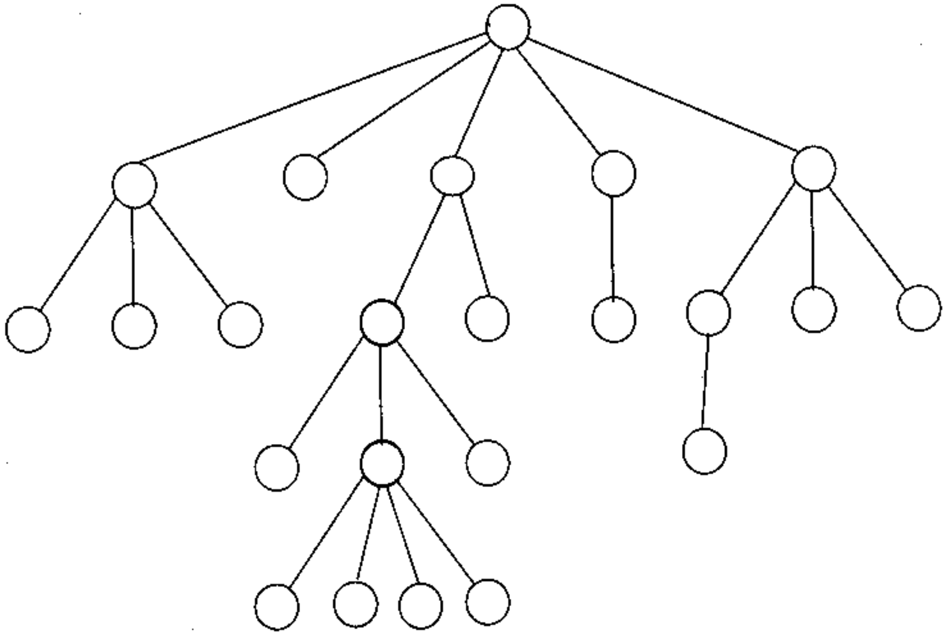
a : 001, b : 0001, e : 1, r : 0000, s : 0100, t : 011, x : 01010.

Hãy tìm các từ được biểu diễn bởi

a) 01110100011; b) 0001110000;

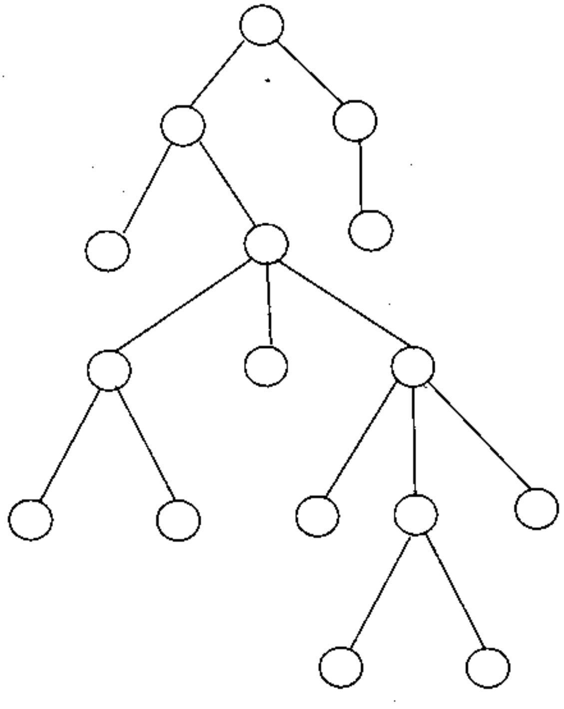
c) 0100101010; d) 01100101010.

22. Cho cây ngũ nguyên dưới đây



Hãy gán nhãn theo địa chỉ phổ dụng cho tất cả các đỉnh của cây trên.

23. Xây dựng hệ địa chỉ phổ dụng cho cây có gốc và được sắp sau:



24. Các lá của cây có gốc và được sắp có thể có danh sách địa chỉ phổ dụng như sau được không? Nếu có, hãy xây dựng cây có gốc đó:

a) 1.1.1, 1.1.2, 1.2, 2.1.1.1, 2.1.2, 2.1.3, 2.2, 3.1.1, 3.1.2.1, 3.1.2.2, 3.2;

b) 1.1, 1.2.1, 1.2.2, 1.2.3, 2.1, 2.2.1, 2.3.1, 2.3.2, 2.4.2.1, 2.4.2.2, 3.1, 3.2.1, 3.2.2;

c) 1.1, 1.2.1, 1.2.2, 1.2.2.1, 1.3, 1.4, 2, 3.1, 3.2, 4.1.1.1.

25. Để biểu diễn biểu thức số học chứa các toán tử + (cộng), – (trừ), * (nhân), / (chia) và ↑ (lũy thừa) (ta dùng các dấu ngoặc biểu thị thứ tự các phép toán) bằng cây có gốc được sắp, trong đó các đỉnh trong biểu thị các phép toán, các lá biểu thị các số hay các biến. Mỗi một phép toán tác động lên các cây con bên trái và cây con bên phải của nó đều theo thứ tự này.

a) Tìm cây có gốc biểu diễn biểu thức: $((x + y)\uparrow 2) + ((x - 4)/3)$.

b) Tìm dạng tiền tố của biểu thức trên.

c) Tìm dạng hậu tố của biểu thức trên.

26. a) Hãy tìm cây có gốc và được sắp biểu diễn mệnh đề lôgic phức hợp:

$$(\neg (p \wedge q) \leftrightarrow (\neg p \vee \neg q).$$

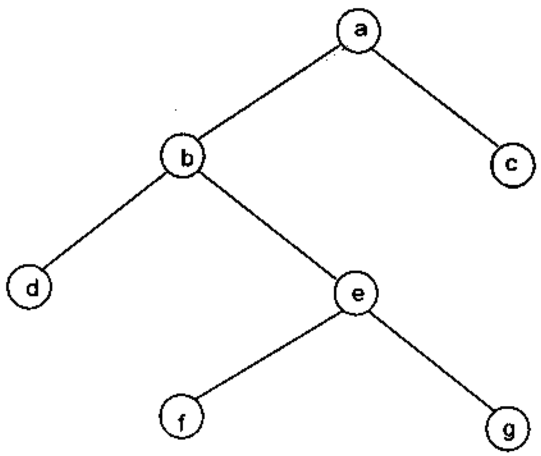
b) Tìm dạng tiền tố, hậu tố và trung tố của biểu thức trên.

27. a) Tìm giá trị của biểu thức tiền tố: + – * 2 3 5 / ↑ 2 3 4.

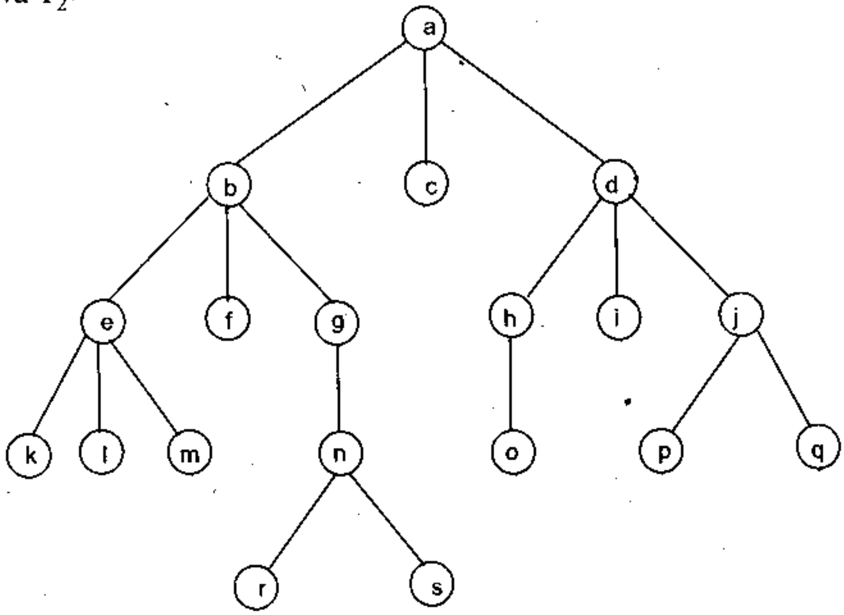
b) Tìm giá trị của biểu thức hậu tố: 7 2 3 * – 4 ↑ 9 3 / +.

28. Cho hai cây có gốc và được sắp:

T_1:



và T_2:



Xác định thứ tự mà các đỉnh của hai cây nếu ta duyệt nó theo kiểu:

a) Tiền thứ tự;

b) Trung thứ tự;

c) Hậu thứ tự.

29. a) Dùng thuật toán sắp xếp kiểu nổi bọt, hãy sắp xếp danh sách L: 3, 1, 5, 7, 4, 6 và chỉ rõ các danh sách nhận được ở mỗi bước.

a) Dùng thuật toán sắp xếp kiểu nổi bọt, hãy sắp xếp danh sách L: d, f, m, k, a, b và chỉ rõ các danh sách nhận được ở mỗi bước.

30. Dùng thuật toán hoà nhập hai danh sách, hãy hoà nhập hai danh sách L_1: 1, 3, 5, 7, 9 và L_2: 2, 4, 6, 8, 10 bằng phương pháp lập bảng.

31. Hãy viết thuật toán sắp xếp kiểu chọn lọc ở dạng giả mã.

32. Sắp xếp danh sách 7, 8, 1, 2, 4, 3, 5, 6, 9 bằng thuật toán sắp xếp nhanh.

33. Mô tả thuật toán sắp xếp nhanh dưới dạng giả mã.

34. a) Tính số lớn nhất các phép so sánh cần thiết để sắp xếp một danh sách có 4 phần tử bằng thuật toán sắp xếp nhanh.

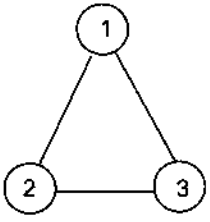
b) Tìm số ít nhất các phép so sánh cần thiết để sắp xếp một danh sách có 4 phần tử bằng thuật toán sắp xếp nhanh.

35. Chứng minh rằng một đơn đồ thị là liên thông khi và chỉ khi nó có cây khung.

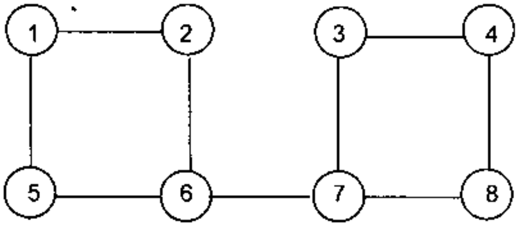
36. Tìm cây khung của các đồ thị K_5, $K_{4,4}$ và $K_{1,6}$.

37. Vẽ tất cả các cây khung của mỗi đồ thị dưới đây:

a) G_1:



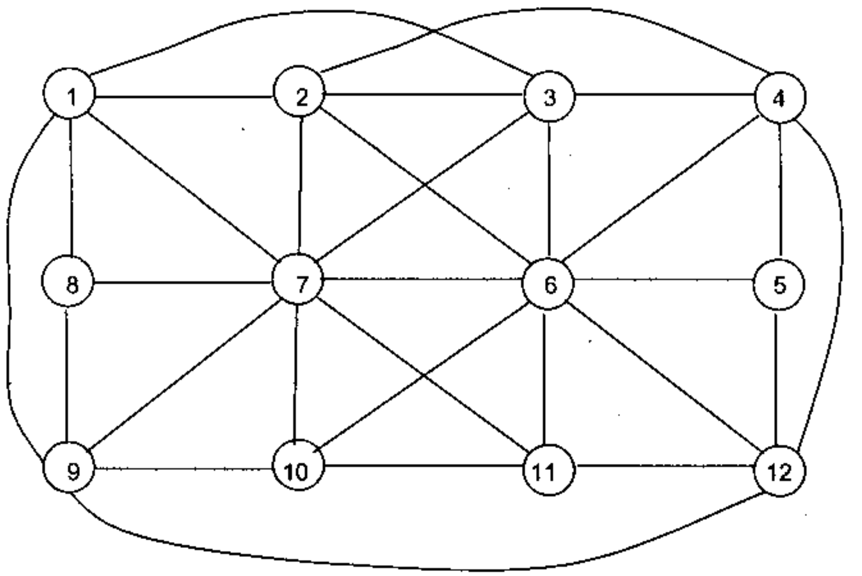
b) G_2:



38. Các đồ thị sau đây có bao nhiêu cây khung khác nhau?

a) K_n ($n \geq 2$); b) C_n ($n \geq 3$).

39. Giả sử hãng hàng không cần giảm bớt lịch bay để tiết kiệm tiền. Nếu ban đầu các đường bay được minh hoạ trong hình vẽ dưới. Có thể hủy bỏ các chuyến bay nào mà vẫn giữ được giao thông giữa hai thành phố bất kỳ.



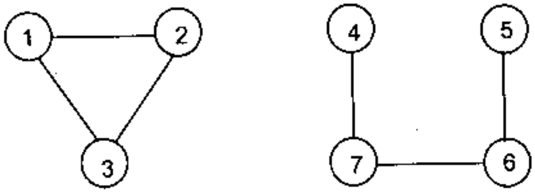
12 thành phố đánh số từ 1 đến 12 (mỗi đỉnh là một thành phố). Cạnh nối giữa các đỉnh là đường bay giữa các thành phố với nhau.

40. Chứng minh rằng độ dài của đường đi ngắn nhất giữa các đỉnh x, y trong một đơn đồ thị liên thông bằng số mức của x trong cây khung ưu tiên chiều rộng của đồ thị G với gốc chọn là đỉnh y.

41. Hãy viết thủ tục tìm kiếm ưu tiên chiều sâu dưới dạng giả mã.

42. Rừng khung của đồ thị G là rừng chứa mọi đỉnh của G sao cho hai đỉnh thuộc cùng một cây của rừng nếu giữa chúng có một đường đi trong G. Chứng minh rằng mọi đơn đồ thị hữu hạn đều có rừng khung.

43. Cho đồ thị G = <X, U> dưới dạng

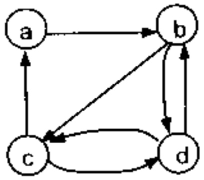


Tìm rừng khung của nó.

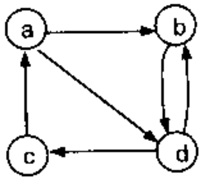
44. Cây khung có gốc của một đồ thị có hướng được định nghĩa là cây có gốc chứa các cạnh của đồ thị sao cho mọi đỉnh của đồ thị đều là một điểm đầu mút của một trong các cạnh của cây.

Tìm cây khung có gốc của các đồ thị cho dưới đây:

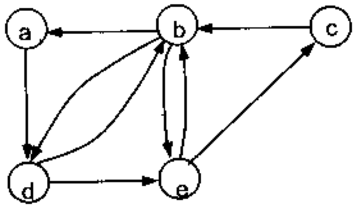
a)



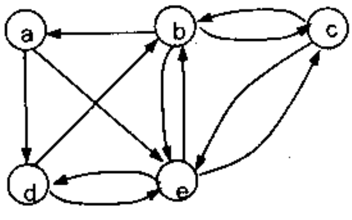
b)



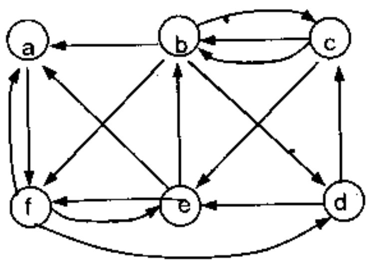
c)



c)

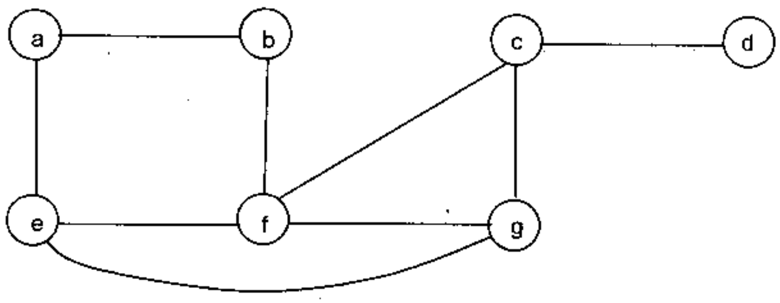


d)



45. Gọi T_1 và T_2 là 2 cây khung của đồ thị G = <X, U>. Khoảng cách giữa T_1 và T_2 được định nghĩa là số các cạnh trong T_1 và T_2 mà không là cạnh chung của chúng.

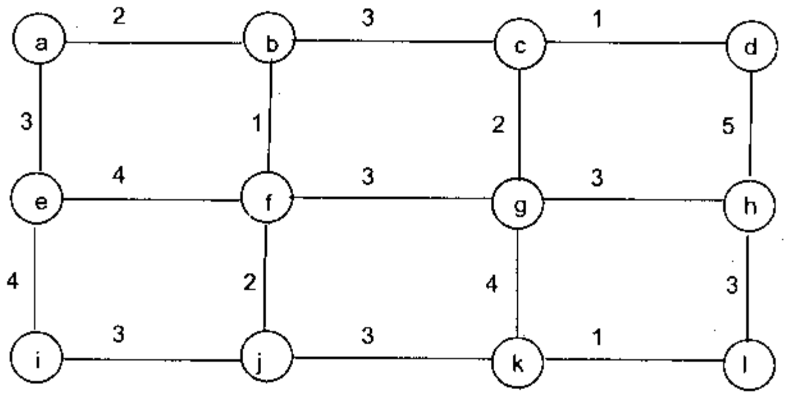
Cho đồ thị G = <X, U> có dạng



a) Chỉ ra 5 cây khung của G

b) Hãy chỉ ra khoảng cách giữa các cặp cây khung đó.

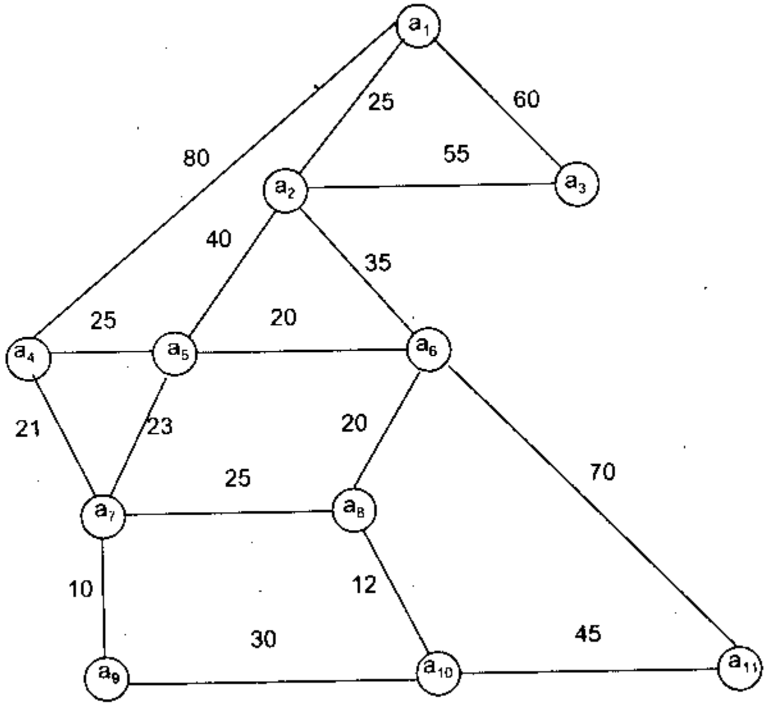
46. Cho đồ thị G = <X, U> dưới dạng



Tìm cây khung bé nhất theo thuật toán Prim.

47. Các con đường được biểu diễn trên đồ thị dưới đây là hoàn toàn chưa được trải nhựa. Độ dài của con đường được biểu thị bằng trọng số của cạnh (km). Cần phải trải nhựa những đoạn nào để vẫn có đường đi được trải nhựa giữa hai thành phố bất kỳ mà độ dài đường trải nhựa là tối thiểu?

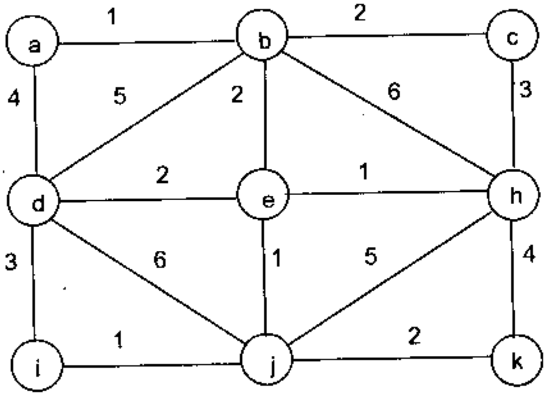
(Mỗi thành phố là một đỉnh)



48. Cây khung cực đại của một đồ thị liên thông có trọng số là cây khung có trọng số lớn nhất.

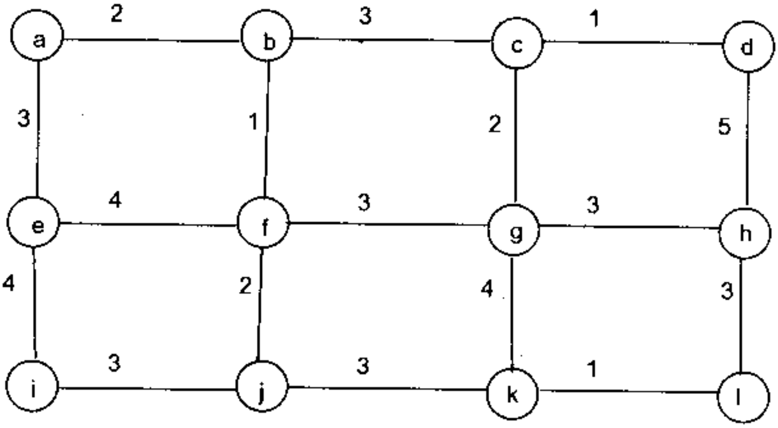
a) Hãy đề xuất một thuật toán tương tự như thuật toán Prim xây dựng cây khung có trọng số lớn nhất trong đồ thị liên thông có trọng số.

b) Cho đồ thị G = <X, U> dưới đây:



Áp dụng thuật toán vừa xây dựng trong phần a, hãy tìm cây khung cực đại của đồ thị liên thông có trọng số trên.

49. Cho đồ thị G = <X, U> dưới dạng:



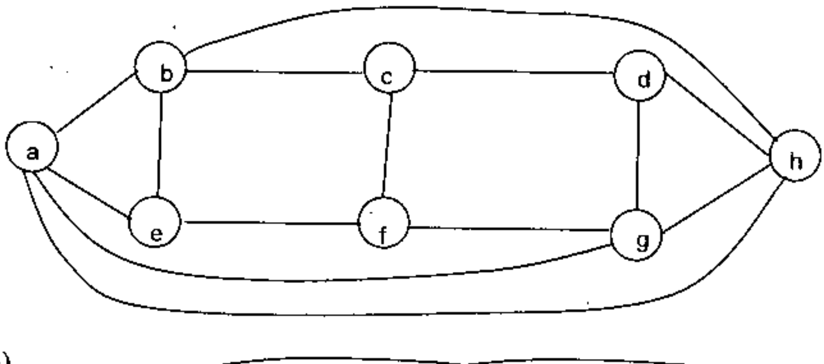
a) Tìm cây khung bé nhất có chứa các cạnh (e, i) với trọng số 4 và (g, k) với trọng số 4.

b) Tìm cây khung cực đại của đồ thị trên theo thuật toán tựa Kruscal.

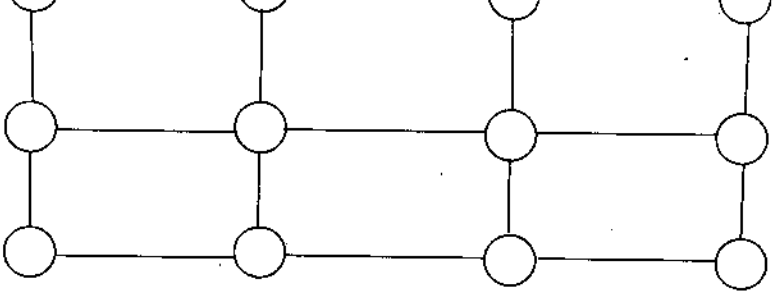
c) Tìm cây khung bé nhất theo thuật toán Prim đối với đồ thị trên.

50. Hãy tạo ra cây khung của các đồ thị sau đây bằng cách xoá cạnh tạo ra chu trình trong đồ thị:

a) G_1:



b)



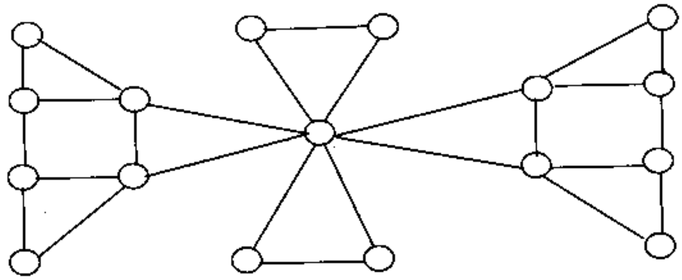
51. a) Cần phải xoá đi bao nhiêu cạnh khỏi đồ thị liên thông với n đỉnh và m cạnh để nhận được một cây khung?

b) Hãy tìm cây khung cho đồ thị K_5.

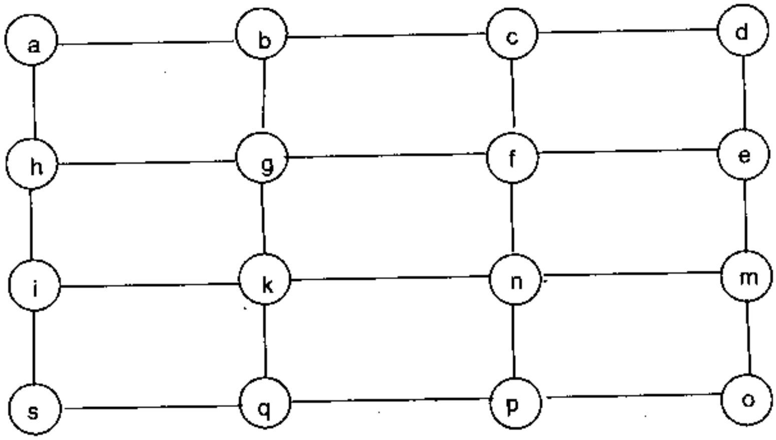
c) Đồ thị K_3, K_4 có bao nhiêu cây khung khác nhau?

52. Tìm cây khung của đồ thị đơn liên thông dưới đây theo thuật toán ưu tiên chiều sâu:

a)



b)



53. Hãy giải thích cách dùng kỹ thuật quay lui để tìm đường đi hoặc chu trình Hamilton trong một đồ thị.

54. Hãy đề xuất một thuật toán xây dựng rừng khung của một đồ thị bằng kỹ thuật tìm kiếm ưu tiên chiều sâu và chiều rộng.

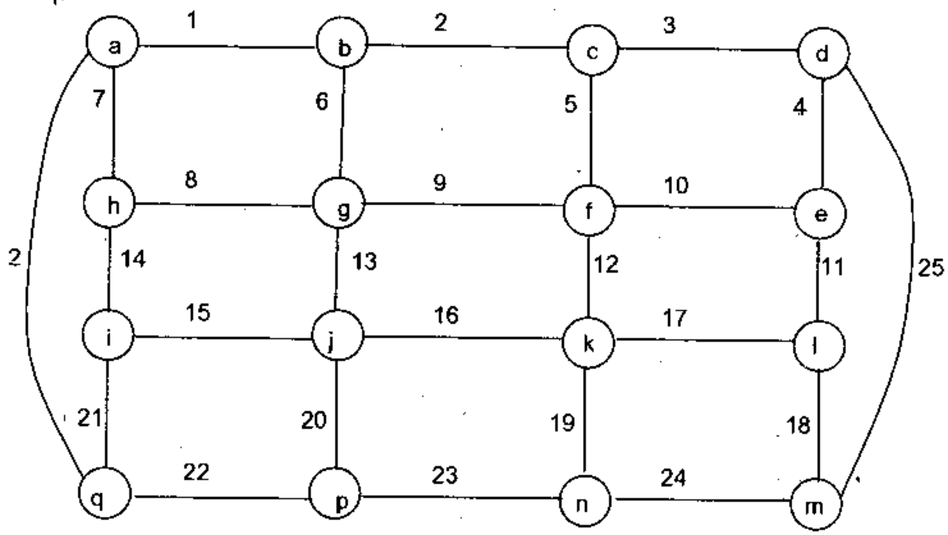
55. Giả sử T_1, T_2, T_3 là các cây khung của đồ thị G. Chứng minh rằng khoảng cách giữa T_1 và T_3 không vượt quá tổng khoảng cách giữa T_1 và T_2 và khoảng cách giữa T_2 và T_3.

56. a) Chỉ ra rằng đồ thị có hướng liên thông trong đó mỗi đỉnh có bậc ra và bậc vào như nhau sẽ có cây khung có gốc.

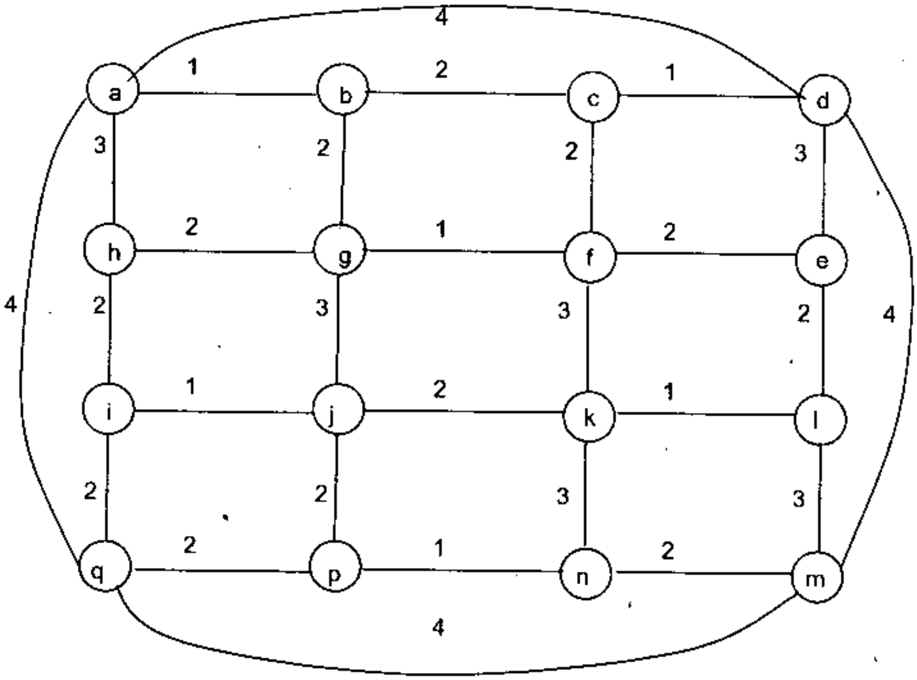
b) Hãy đề xuất một thuật toán xây dựng cây khung có gốc của một đồ thị có hướng liên thông trong đó mỗi đỉnh có bậc ra và bậc vào bằng nhau.

57. a) Dùng thuật toán Prim, tìm cây khung bé nhất của mỗi đồ thị cho dưới đây:

G_1:



G_2:



b) Dùng thuật toán Prim tìm cây khung bé nhất trong đồ thị G đi qua hai cạnh (a, d) có trọng số 4 và (q, m) có trọng số 4.

58. Tìm một đơn đồ thị liên thông có trọng số với một số tối thiểu các cạnh sao cho nó có nhiều hơn một cây khung bé nhất.

59. Hãy viết thuật toán tìm cây khung bé nhất trong đồ thị chứa một tập xác định các cạnh trong đồ thị đơn vô hướng, liên thông và có trọng số.

60. Hãy chỉ ra rằng có duy nhất một cây khung bé nhất trong đồ thị liên thông có trọng số nếu trọng số của các cạnh khác nhau từng cặp.

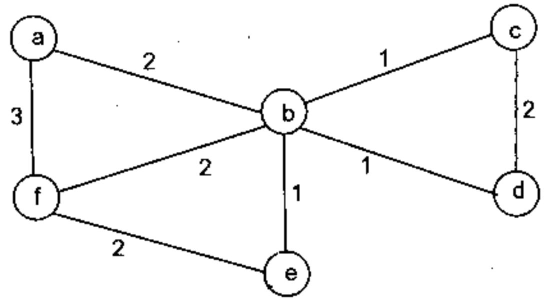
61. Ba cặp vợ chồng đi tới bờ một con sông. Mỗi bà vợ đều hay ghen và không tin chồng khi để anh ta đứng một mình với các bà kia. Làm thế nào 6 người có thể qua sông bằng một chiếc thuyền chỉ chở được không quá 2 người, sao cho không có ông chồng nào ở một mình với các bà không là vợ mình? Hãy dùng lý thuyết đồ thị giải bài toán trên.

62. Giả sử u ∈ U là một cạnh của đồ thị G = <X, U>, cạnh này liên thuộc với đỉnh x ∈ X sao cho trọng số của cạnh u không vượt quá trọng số của bất kỳ cạnh nào khác liên thuộc với đỉnh x. Chỉ ra rằng có cây khung bé nhất chứa cạnh u trong đồ thị liên thông G ở trên.

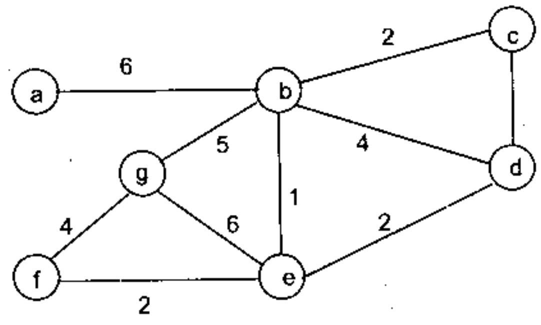
63. Cây khung hạn chế một đồ thị đơn G = <X, U> là cây khung có tính chất: bậc của một đỉnh trong cây này không thể vượt quá một giới hạn xác định nào đó. Cây khung như vậy có nhiều ứng dụng trong thực tế.

Hãy tìm cây khung bé nhất của mỗi một trong các đồ thị có trọng số sau đây, trong đó bậc của mỗi đỉnh trong cây khung bé nhất không vượt quá 2.

G_1:



G_2:



PHẦN IV
NGÔN NGỮ HÌNH THỨC

Chương 9
VĂN PHẠM VÀ NGÔN NGỮ SINH BỞI VĂN PHẠM

§1. KHÁI NIỆM CHUNG VỀ NGÔN NGỮ

Thuật toán là phương tiện giao tiếp. Sự giao tiếp ở đây có thể là sự giao tiếp giữa người với nhau, hoặc giữa người với máy, hoặc giữa máy với máy. Ngôn ngữ để con người có thể giao tiếp với nhau được gọi là *ngôn ngữ tự nhiên*, chẳng hạn như tiếng Anh, tiếng Nga, tiếng Việt là các ngôn ngữ tự nhiên. Như chúng ta đã biết, các quy tắc cú pháp của ngôn ngữ tự nhiên rất phong phú, đa dạng và phức tạp. Tuy nhiên, những yêu cầu nghiêm ngặt về ngữ nghĩa trong các ngôn ngữ tự nhiên chưa cao, chẳng hạn như cùng một từ hoặc cùng một câu ta có thể hiểu chúng theo những nghĩa khác nhau tuỳ theo những ngữ cảnh cụ thể. Để có sự giao tiếp giữa người với máy, hoặc giữa máy với máy cần phải có một ngôn ngữ mà các quy tắc cú pháp chặt chẽ hơn. Nói cách khác là với một từ hoặc một câu thì nghĩa của chúng phải là duy nhất. Những ngôn ngữ như thế được gọi là *ngôn ngữ hình thức*.

Để xây dựng một ngôn ngữ hình thức cần có một tập hợp hữu hạn khác trống các ký hiệu nào đó gọi là *bảng chữ cái*. Dãy hữu hạn các phần tử của bảng chữ cái được gọi là một *từ* hay một *xâu* trên bảng chữ cái. Một tập hợp các từ trên bảng chữ cái được gọi là *ngôn ngữ*.

Trong chương này chúng ta đề cập đến một số khái niệm và kết quả cơ bản liên quan đến ngôn ngữ hình thức.

1.1. Bảng chữ cái

Cho Σ là một tập hợp hữu hạn khác trống các ký hiệu nào đó mà ta gọi là một *bảng chữ cái*. Mỗi phần tử trong Σ được gọi là một *ký tự*.

Ví dụ 1:

– Bộ chữ cái của ngôn ngữ tiếng Việt gồm 26 ký tự, ký hiệu là

$$\Sigma = \{a, b, ..., y\}.$$

– Để có các xâu nhị phân ta thường dùng bảng chữ cái gồm hai ký hiệu 0, 1; tức là $\Sigma = \{0, 1\}$.

1.2. Xâu ký tự

Nói đến một *xâu* ký tự hay một *từ* trên bảng chữ cái Σ là nói đến một dãy hữu hạn các ký tự trên bảng Σ mà chúng được viết liền nhau.

Ví dụ 2: Beautiful là một xâu trên bảng chữ cái

$$\Sigma = \{a, b, c, ..., z\}.$$

Độ dài của xâu là số các ký tự có trong xâu đó.

Xâu 010110 là một xâu trên bảng chữ cái $\Sigma = \{0, 1\}$ có độ dài là 6. Với ω là một xâu thì ta ký hiệu $|\omega|$ (hoặc $l(\omega)$) là *độ dài* của xâu đó. Người ta quy ước xâu rỗng là xâu có độ dài bằng 0. Xâu rỗng được ký hiệu bởi ^.

1.3. Ngôn ngữ

Mỗi tập các từ trên bảng chữ cái được gọi là *ngôn ngữ* trên bảng chữ cái đó. Cho bảng chữ cái Σ, khi đó tất cả các từ trên bảng chữ cái Σ kể cả các xâu rỗng được ký hiệu bởi Σ^*. Tập tất cả các từ trên bảng chữ cái Σ mà mọi từ trong nó đều có độ dài khác 0 được ký hiệu bởi Σ^+. Từ đó ta có: $\Sigma^+ = \Sigma^* \setminus \{\wedge\}$. Các tập $\varnothing$ và tập $\{\wedge\}$ được xem là các ngôn ngữ trên bảng chữ cái bất kỳ.

Đối với một ngôn ngữ $L \subseteq \Sigma^*$, người ta thường quan tâm đến một số vấn đề sau:

– Với một xâu $\omega \in \Sigma^*$ bất kỳ cho trước, một vấn đề đặt ra là xâu ω có thuộc ngôn ngữ L cho trước hay không (vấn đề đoán nhận)?

– Với xâu bất kỳ ω trong L, làm thế nào để sinh ra ω (vấn đề sinh)?

Đó là những vấn đề liên quan đến biểu diễn ngôn ngữ. Người ta thường biểu diễn ngôn ngữ nhờ các văn phạm (vấn đề sinh), hoặc các máy hình thức (vấn đề đoán nhận).

§2. VĂN PHẠM VÀ NGÔN NGỮ SINH BỞI VĂN PHẠM

2.1. Định nghĩa văn phạm

Ta có thể hình dung một văn phạm như một "thiết bị tự động" mà nó có khả năng sinh ra một tập hợp các từ trên một bảng chữ cái cho trước. Mỗi từ được sinh ra sau một số hữu hạn bước thực hiện các quy tắc của văn phạm.

Việc xác định một ngôn ngữ trên bảng chữ cái cho trước có thể được thực hiện bằng một trong các cách thức sau:

Cách 1: Đối với mỗi xâu thuộc ngôn ngữ đã cho ta có thể chọn một quy cách hoạt động của "thiết bị tự động" để sau một số hữu hạn bước làm việc nó dừng và sinh ra chính xâu đó.

Cách 2: "Thiết bị tự động" có khả năng lần lượt sinh ra tất cả các xâu trong ngôn ngữ đã cho.

Cách 3: Với một xâu ω cho trước, "thiết bị tự động" có thể cho biết xâu đó có thuộc ngôn ngữ đã cho hay không.

Trong lý thuyết văn phạm, người ta đã chứng minh được rằng, ba cách thức trên là tương đương nhau, hay văn phạm làm việc theo các cách trên là tương đương nhau.

Vì vậy, ở đây ta quan tâm đến cách thứ nhất, tức là xét văn phạm như là một "thiết bị tự động" sinh ra các từ. Vì lẽ đó mà người ta còn gọi các "thiết bị tự động" đó là *văn phạm sinh.*

Việc sinh ra các từ có thể được thực hiện bằng nhiều cách khác nhau. Các từ có thể được sinh ra bởi các văn phạm, bởi các ôtômat, bởi các máy hình thức như máy Turing, ... Ở đây ta đề cập đến cách của Chomsky đưa ra vào những năm 1956 – 1957.

Định nghĩa 1: Văn phạm G là một bộ sắp thứ tự gồm bốn thành phần $<\Sigma, \Delta, I, R>$, trong đó:

– Σ, Δ là các tập hữu hạn, khác rỗng, không giao nhau. Tập Σ được gọi là *từ điển cơ bản*, mỗi phần tử của nó được gọi là *ký hiệu cơ bản* hay *ký hiệu kết thúc* (terminal). Tập Δ được gọi là *từ điển hỗ trợ*, các phần tử của nó được gọi là các *ký hiệu hỗ trợ* hay còn gọi là *ký hiệu không kết thúc* (nonterminal). Đặt $V = \Sigma \cup \Delta$, V được gọi là *từ điển đầy đủ.*

– Ký hiệu $I \in \Delta$ được gọi là *ký hiệu ban đầu.*

– R là tập các quy tắc mà mỗi phần tử của nó có dạng $\alpha \to \beta$, ở đây α, β là các từ trên từ điển đầy đủ V, ký hiệu $\to \notin V$.

Định nghĩa 2: Cho văn phạm $G = <\Sigma, \Delta, I, R>$ và một quy tắc $r = \alpha \to \beta \in R$. Ký hiệu $* \notin V$, một xâu có dạng $\xi_1 * \alpha * \xi_2$ được gọi là một *vị trí* của α trong xâu $\omega = \xi_1\alpha\xi_2$, ở đây ω là một xâu trên từ điển đầy đủ V. Khi đó ta nói xâu $\eta = \xi_1\beta\xi_2$ là nhận được từ xâu $\omega = \xi_1\alpha\xi_2$ nhờ việc áp dụng quy tắc $r = \alpha \to \beta$ vào vị trí $\xi_1 * \alpha * \xi_2$ của α trong xâu $\omega = \xi_1\alpha\xi_2$. Xâu η gọi là xâu *dẫn được trực tiếp* từ xâu ω trong văn phạm G và được ký hiệu là $\omega \models_G \eta$, hay đơn giản hơn là $\omega \models \eta$.

Định nghĩa 3: Dãy các xâu $D = (\omega_0, \omega_1, ..., \omega_k)$ được gọi là một *dẫn xuất* của xâu ω_k từ ω_0 trong văn phạm $G = <\Sigma, \Delta, I, R>$ nếu $\omega_{i-1} \models \omega_i$, với $i = 1, 2, ..., k$. Số k được gọi là *độ dài* của dẫn xuất D.

Dẫn xuất của ω_k từ ω_0 trong văn phạm G được ký hiệu là $\omega_0 \vdash_G \omega_k$, hay đơn giản hơn là $\omega_0 \vdash \omega_k$. Nếu ω_i nhận được từ ω_{i-1} bằng cách áp dụng quy tắc $r = \alpha \to \beta$ thì ta nói quy tắc r được áp dụng ở bước thứ i.

2.2. Ngôn ngữ của văn phạm

Định nghĩa 4: Cho văn phạm $G = <\Sigma, \Delta, I, R>$ và $D = (\omega_0, \omega_1, ..., \omega_k)$ là một dẫn xuất của xâu ω_k từ ω_0 trong văn phạm G. Nếu $\omega_0 = I$ và $\omega_k \in \Sigma^*$ (tập tất cả các xâu trên Σ) thì ω_k được gọi là một *xâu (từ)* sinh bởi văn phạm G và khi đó dẫn xuất $D = (I = \omega_0, \omega_1, ..., \omega_k = P)$ với $P \in \Sigma^*$ được gọi là *dẫn xuất đầy đủ* trong G.

Ngôn ngữ sinh ra bởi văn phạm G được ký hiệu bởi L(G) và được định nghĩa là:

$$L(G) = \{\omega : \omega \in \Sigma^* \text{ và } I \vdash_G \omega\}.$$

Ví dụ 1: Cho $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a, b\}$, $\Delta = \{I\}$, $R = \{I \to aIb, I \to ab\}$.

Khi đó thấy rằng: $L(G) = \{a^nb^n : n \in \mathbb{N}, n \geq 1\}$.

Thật vậy, dẫn xuất đầy đủ của a^nb^n là:

$$D = \{I, aIb, a^2Ib^2, ..., a^{n-1}Ib^{n-1}, a^nb^n\}, \text{ hay } I \vdash a^nb^n.$$

Định nghĩa 5: Hai văn phạm G và G' được gọi là *tương đương* nhau nếu $L(G) = L(G')$.

Ví dụ 2: Cho hai văn phạm $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a, b\}$, $\Delta = \{I\}$,

$R = \{I \to aIb, I \to \wedge\}$;

$G' = <\Sigma', \Delta', I', R'>$ với $\Sigma' = \{a, b\}$, $\Delta' = \{I'\}$, $R' = \{I' \to aI'b, I' \to ab\}$.

Dễ thấy rằng, $L(G') = \{a^n b^n : n \geq 1\}$, do đó G và G' là hai văn phạm tương đương với nhau (sai khác từ ^).

§3. PHÂN LOẠI VĂN PHẠM CỦA CHOMSKY

Chomsky phân loại văn phạm thành bốn nhóm trên cơ sở định nghĩa văn phạm nói chung:

– Nhóm 0: Văn phạm ngữ cấu;

– Nhóm 1: Văn phạm cảm ngữ cảnh;

– Nhóm 2: Văn phạm phi ngữ cảnh;

– Nhóm 3: Văn phạm chính quy.

Dưới đây ta định nghĩa văn phạm từ nhóm 0 đến nhóm 3 như sau:

Cho văn phạm $G = <\Sigma, \Delta, I, R>$. Đặt $V = \Sigma \cup \Delta$ là từ điển đầy đủ, V^* là tập các từ trên V, V^+ là lập các từ có độ dài khác không, tức là $V^+ = V^* \setminus \{^\wedge\}$.

Định nghĩa 6: Văn phạm $G = <\Sigma, \Delta, I, R>$ được gọi là *văn phạm ngữ cấu* nếu mọi quy tắc $r \in R$ đều có dạng $r = \alpha \to \beta$, trong đó $\alpha \in V^+$, $\beta \in V^*$. Quy tắc của văn phạm ngữ cấu gọi là quy tắc *ngữ cấu*. Ngôn ngữ do văn phạm ngữ cấu sinh ra được gọi là *ngôn ngữ ngữ cấu* (NNNC).

Chú ý: α có chứa ít nhất một ký tự trong Δ.

Định nghĩa 7: Văn phạm $G = <\Sigma, \Delta, I, R>$ được gọi là *văn phạm cảm ngữ cảnh* nếu với mọi quy tắc $r \in R$ đều có dạng $r = \alpha \to \beta$, ở đây $\alpha \in V^+$, $\beta \in V^*$ và $|\alpha| \leq |\beta|$.

Mỗi quy tắc của văn phạm cảm ngữ cảnh gọi là quy tắc *cảm ngữ cảnh*. Ngôn ngữ do văn phạm cảm ngữ cảnh sinh ra gọi là một *ngôn ngữ cảm ngữ cảnh* (NNCNC).

Ví dụ 1: Cho văn phạm $G = <\Sigma, \Delta, I, R>$ với

$$\Sigma = \{a, b\}, \Delta = \{I\}, R = \{I \to aIb, I \to bIa, I \to II, I \to ab\}$$

là một văn phạm cảm ngữ cảnh.

Ví dụ 2: Cho văn phạm $G = <\Sigma, \Delta, I, R>$, $\Sigma = \{a, b, c\}$, $\Delta = \{I, A, B\}$, $I \in \Delta$ và $R = \{I \to abc, I \to aAbc, Ab \to bA, Ac \to Bbc, bB \to Bb, aB \to aaA, aB \to aa\}$. Khi đó G là văn phạm cảm ngữ cảnh.

Định nghĩa 8: Văn phạm $G = <\Sigma, \Delta, I, R>$ được gọi là văn *phạm phi ngữ cảnh* nếu mọi quy tắc $r \in R$ đều có dạng $r = A \rightarrow \theta$, ở đây ký hiệu $A \in \Delta$ và $\theta \in V^*$.

Mỗi quy tắc của văn phạm phi ngữ cảnh gọi là quy tắc *phi ngữ cảnh*. Ngôn ngữ do văn phạm phi ngữ cảnh sinh ra gọi là *ngôn ngữ phi ngữ cảnh* (NNPNC).

Cho xâu $\omega = a_1a_2 \ldots a_{n-1}a_n$, khi đó *ảnh gương* của nó là xâu $\hat{\omega} = a_na_{n-1} \ldots a_2a_1$. Ảnh gương của ngôn ngữ L sẽ là tập $\{\hat{\omega}: \omega \in L\}$.

Ảnh gương của ký tự a là $a = \hat{a}$, ảnh gương của từ rỗng ($\wedge$) là chính nó.

Ví dụ 3: Xét văn phạm $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a, b\}$, $\Delta = \{I\}$,

$$R = \{I \rightarrow \wedge, I \rightarrow aIa, I \rightarrow bIb, I \rightarrow aa, I \rightarrow bb\}.$$

G là văn phạm phi ngữ cảnh và nó sinh ra ngôn ngữ phi ngữ cảnh $L(G) = \left\{\omega\,\hat{\omega} \,\middle|\, \omega \in V^+\right\}$.

Chẳng hạn, với xâu ω = abab, khi đó ta có $\hat{\omega}$ = baba và xâu $\omega\hat{\omega}$ = ababbaba có dẫn xuất là D = (I, aIa, abIba, abaIaba, ababbaba = $\omega\hat{\omega}$).

Định nghĩa 9: Văn phạm $G = <\Sigma, \Delta, I, R>$ gọi là *văn phạm chính quy* nếu tập R gồm các quy tắc có dạng: $A \rightarrow aB$, $A \rightarrow a$; ở đây $A, B \in \Delta$, $a \in \Sigma$.

Quy tắc của văn phạm chính quy gọi là *quy tắc chính quy*. Ngôn ngữ do văn phạm chính quy sinh ra được gọi là *ngôn ngữ chính quy* (NNCQ).

Ví dụ 4: Giả sử $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a, b, c\}$, $\Delta = \{I\}$,

$$R = \{I \rightarrow aI, I \rightarrow bI, I \rightarrow cI, I \rightarrow a, I \rightarrow b, I \rightarrow c\}.$$

Khi đó G là một văn phạm chính quy.

Dễ thấy rằng, xâu $a^mb^nc^k \in L(G)$ vì dẫn xuất đầy đủ của $\omega = a^mb^nc^k$ là $D = (I, aI, a^2I, \ldots, a^mI, a^mbI, a^mb^2I, \ldots, a^mb^nI, a^mb^ncI, a^mb^nc^2I, \ldots, a^mb^nc^{k-1}I, a^mb^nc^k = \omega)$.

Nhận xét: Từ định nghĩa ta thấy khái niệm văn phạm ngữ cấu là khái niệm tổng quát nhất, nghĩa là lớp các văn phạm 0 là lớp rộng nhất, nó chứa đựng văn phạm cảm ngữ cảnh; lớp văn phạm cảm ngữ cảnh chứa lớp các văn phạm phi ngữ cảnh; lớp các văn phạm phi ngữ cảnh chứa lớp văn phạm chính quy. Vì vậy, đối với lớp các ngôn ngữ do chúng sinh ra thoả mãn bao hàm thức:

$$\text{NNCQ} \subset \text{NNPNC} \subset \text{NNCNC} \subset \text{NNNC}.$$

Ta cũng thấy về mặt cấu trúc ngữ pháp thì các quy tắc sinh của các văn phạm phi ngữ cảnh, văn phạm chính quy là đơn giản hơn cả và chúng có nhiều ứng dụng trong việc thiết kế các ngôn ngữ lập trình và trong nghiên cứu về chương trình dịch, ... Vì vậy, trong phần tiếp theo chúng ta dành thêm sự quan tâm tới hai lớp văn phạm đó, nhưng trước hết hãy xét một số ví dụ về văn phạm.

§4. MỘT SỐ VÍ DỤ VỀ VĂN PHẠM

Ví dụ 1: Cho văn phạm $G = <\Sigma, \Delta, I, R>$, trong đó: $\Sigma = \{a_1, a_2, ..., a_n\}$ là bảng ký hiệu cơ bản; $\Delta = \{I, A_1, A_2, ..., A_{n-1}\}$ là bảng ký hiệu hỗ trợ; $I \in \Delta$ là ký hiệu ban đầu; $R = \{I \to a_1A_1, A_1 \to a_2A_2, A_2 \to a_3A_3, ..., A_{n-2} \to a_{n-1}A_{n-1}, A_{n-1} \to a_n\}$ $(n \geq 2)$ là tập quy tắc.

Theo định nghĩa đây là văn phạm chính quy. Văn phạm này sinh ra ngôn ngữ chính quy $L(G) = \{\omega = a_1a_2...a_n\}$.

Thật vậy, với từ ω ta có dẫn xuất đầy đủ trong G là:

$$D = (I, a_1A_1, a_1a_2A_2, ..., a_1a_2...a_{n-1}A_{n-1}, a_1a_2...a_n\}.$$

Ví dụ 2: Cho văn phạm $G = <\Sigma, \Delta, I, R>$, trong đó $\Sigma = \{x_1, x_2, ..., x_n\}$, $\Delta = \{I\}$, $R = \{I \to aI, I \to a : a \in \Sigma\}$. Đây cũng là văn phạm chính quy và nó sinh ra ngôn ngữ chính quy:

$$L(G) = \{\omega : \omega \in \Sigma^+\}, \text{ tức là } L(G) = \Sigma^+.$$

Chẳng hạn, với $\omega = x_{i1}x_{i2}...x_{ik} \in \Sigma^+$, ta thấy ω do G sinh ra vì nó có dẫn xuất đầy đủ là:

$$D = (I, x_{i1}I, x_{i1}x_{i2}I, x_{i1}x_{i2}...x_{ik-1}I, x_{i1}x_{i2}...x_{ik}).$$

Ví dụ 3: Ngôn ngữ $\varnothing$ sinh ra bởi văn phạm chính quy sau:

$$G = <\Sigma, \Delta, I, R>, \text{ trong đó } \Sigma = \{a\}, \Delta = \{I\}, R = \{I \to aI\}.$$

Thật vậy, G làm việc không bao giờ dừng, tức là không có xâu $\omega \in \Sigma^*$ và $\omega \neq \wedge$ mà G sinh ra, hay G sinh ra ngôn ngữ $\varnothing$.

Ví dụ 4: Văn phạm $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a, b\}$, $\Delta = \{I, B\}$, $R = \{I \to aI, I \to aB, B \to bB, B \to b\}$ cũng là văn phạm chính quy và $L(G) = \{a^nb^m : n, m \geq 1\}$ là ngôn ngữ chính quy do G sinh ra.

Thật vậy, với $\omega = a^n b^m$ ta có dẫn xuất đầy đủ trong G là

$$D = (I, aI, a^2I, ..., a^{n-1}I, a^nB, a^nbB, ..., a^nb^{m-1}B, a^nb^m).$$

Ví dụ 5: Ngôn ngữ $\{a^nb^n : n \geq 1\}$ sinh ra bởi văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a, b\}$, $\Delta = \{I\}$, $R = \{I \to aIb, I \to ab\}$.

Thật vậy, từ a^nb^n với $n \geq 1$ có dẫn xuất đầy đủ trong G là

$$D = (I, aIb, a^2Ib^2, ..., a^{n-1}Ib^{n-1}, a^nb^n).$$

Ví dụ 6: Cho ngôn ngữ $\{a^nb^na^m : n, m \geq 1\}$.

Ta xét văn phạm $G = <\Sigma, \Delta, I, R>$ là phi ngữ cảnh với $\Sigma = \{a, b\}$, $\Delta = \{I, A\}$, $R = \{I \to Ia, A \to aAb, A \to ab, I \to Aa\}$.

Khi đó ta thấy xâu $a^nb^na^m$ $(n, m \geq 1)$ sẽ có dẫn xuất đầy đủ trong G là

$$D = (I, Ia, Ia^2, ..., Ia^{m-1}, Aa^m, aAba^m, a^2Ab^2a^m, ..., a^{n-1}Ab^{n-1}a^m, a^nb^na^m).$$

Nhận xét: Nếu trong văn phạm G ta thay tập quy tắc R bởi $R' = \{I \to aI, A \to bAa, A \to ba, I \to aA\}$ thì ta nhận được văn phạm $G' = <\Sigma, \Delta, I, R'>$ là phi ngữ cảnh sinh ra ngôn ngữ $\{a^mb^na^n : n, m \geq 1\}$.

Khẳng định là đúng vì:

$$D = (I, aI, a^2I, ..., a^{m-1}I, a^mA, a^mbAa, ..., a^mb^{n-1}Aa^{n-1}, a^mb^na^n)$$

là một dẫn xuất đầy đủ của $a^mb^na^n$ trong G'.

Ví dụ 7: Ngôn ngữ L trên từ điển {a, b} gồm tất cả các từ khác ^ và mỗi từ trong L chứa số ký tự a bằng số ký tự b. Khi đó L là một ngôn ngữ phi ngữ cảnh. Thật vậy, xét văn phạm

$$G = <\Sigma, \Delta, I, R>, \text{ trong đó } \Sigma = \{a, b\}, \Delta = \{I\},$$
$$R = \{I \to II, I \to aIb, I \to bIa, I \to ab, I \to ba\}.$$

Rõ ràng văn phạm G là phi ngữ cảnh và nó sinh ra ngôn ngữ L.

Chẳng hạn, với từ a^mb^m ta có dẫn xuất đầy đủ là

$$D = (I, aIb, a^2Ib^2, ..., a^{m-1}Ib^{m-1}, a^nb^n).$$

Với từ ω = ababaabb có dẫn xuất đầy đủ là

$$D = (I, II, aIbI, ababI, ababaIb, ababaabb).$$

Ví dụ 8: Ngôn ngữ $L = \{\omega\hat{\omega} : \omega \in \Sigma^+\}$ được sinh ra bởi văn phạm phi ngữ cảnh:

$$G = <\Sigma, \Delta, I, R> \text{ với } \Sigma = \{a_1, a_2, ..., a_n\}, \Delta = \{I\};$$
$$I = \{I \to aIa, I \to aa\}, a \in \Sigma.$$

Ví dụ 9: Cho tập $\Sigma = \{a_1, a_2, ..., a_k, b\}$.

Khi đó ngôn ngữ $L = \{\omega b \omega : \omega \in \Sigma^*\}$ được sinh ra bởi văn phạm $G = <\Sigma, \Delta, I, R>$ với Σ như trên, $\Delta = \{I, A_1, A_2, ..., A_k\}$, $R = \{I \to IA_ia_i, I \to b, a_iA_j \to A_ja_i, bA_i \to a_ib\}$; $i, j = 1, 2, ... k$.

Để minh hoạ ta xét trường hợp đơn giản khi k = 2. Ta có

$$\Sigma = \{a_1, a_2, b\}, \Delta = \{I, A_1, A_2\},$$
$$R = \{I \to IA_1a_1, I \to IA_2a_2, I \to b, a_1A_1 \to A_1a_1, a_1A_2 \to A_2a_1,$$
$$a_2A_1 \to A_1a_2, a_2A_2 \to A_2a_2, bA_1 \to a_1b, bA_2 \to a_2b\}.$$

Giả sử với $\omega = a_2a_1$ và $P = \omega b \omega$, khi đó ta có dẫn xuất đầy đủ trong G là

$D = (I, IA_1a_1, IA_2a_2A_1a_1, bA_2a_2A_1a_1, a_2ba_2A_1a_1, a_2bA_1a_2a_1, a_2a_1ba_2a_1 = \omega b \omega)$.

Ví dụ 10: Cho văn phạm $G = <\Sigma, \Delta, I, R>$, ở đây

$$\Sigma = \{a, b\}, \Delta = \{I, A, B, C, D\},$$
$$R = \{I \to ABC, AB \to iADi, Dij \to jDi, DiC \to BiC, iB \to Bi,$$
$$AB \to \wedge, C \to \wedge\} \text{ với } i, j \in \Sigma.$$

Ta sẽ chỉ ra $L(G) = \{\omega\omega : \omega \in \Sigma^*\}$.

Thật vậy, $\wedge = \wedge\wedge \in L(G)$, vì $\wedge\wedge$ có dẫn xuất đầy đủ là

$D = (I, ABC, \wedge C, \wedge\wedge)$. Vậy $\omega\omega \in L(G)$ với $\omega = \wedge$.

Trường hợp khi $\omega = a$, ta chỉ ra $\omega\omega = aa \in L(G)$.

Thật vậy, $D = (I, ABC, aADaC, aABaC, a\wedge aC, a\wedge a\wedge = aa = \omega\omega)$ là dẫn xuất đầy đủ của $\omega\omega = aa$ trong G.

Trường hợp $\omega = b$, khi đó $\omega\omega = bb$ có dẫn xuất đầy đủ trong G là:

$D = (I, ABC, bADbC, bABbc, b\wedge bC, b\wedge b\wedge = bb)$.

Một cách tổng quát nếu có $\omega \in \Sigma^*$ thì sẽ có một dẫn xuất đầy đủ cho $\omega\omega$ là

$D = (I, ABC, ..., \omega AB\omega C, \omega\wedge\omega C, \omega\wedge\omega\wedge = \omega\omega)$.

Ví dụ 11: Xây dựng một văn phạm sinh ra tập các số chính phương $\{n^2 : n = 1, 2, ...\}$.

Vì $n = |^n$ nên văn phạm $G = <\Sigma, \Delta, I, R>$ sinh ra tập $\{n^2 : n = 1, 2, ...\}$ được cho dưới dạng sau:

$\Sigma = \{|\}, \Delta = \{I, A, B, C, D, E, F\}$,

$R = \{I \to ABBDF$ (1)

$BD \to DCB$ (2)

$$BC \to CB \qquad (3)$$
$$AD \to AAE \qquad (4)$$
$$EC \to AE \qquad (5)$$
$$EB \to BE \qquad (6)$$
$$EF \to BBDF \qquad (7)$$
$$DF \to | \qquad (8)$$
$$B \to | \qquad (9)$$
$$A \to | \qquad (10)$$
$$| \to | \qquad (11)$$

}

Ta chỉ ra $L(G) = \{n^2 : n = 1, 2, ...\}$. Thật vậy:

- Khi $n = 1 : 1^2 = 1, \omega = |$.

  Dẫn xuất đầy đủ của ω là $D = (I, |)$.

- Khi $n = 2 : 2^2 = 4, \omega = |^4$

  Dẫn xuất đầy đủ của ω là

  $$D = (I, ABBDF, ABB|, AB||, A|||, ||||).$$

- Khi $n = 3 : 3^2 = 9, \omega = |^9$

  Dẫn xuất đầy đủ của ω là

  $D = (I, ABBDF, ABDCBF, ADCBCBF, AAECBCBF, AAECCBBF, AAECBBF, AAAAEBBF, A^4BEBF, A^4BBEF, A^4BBBBDF, A^4B^4|, A^4|^4, |^4|^4| = |^9)$.

Một cách tổng quát với $\omega = \underbrace{| \dots |}_{n^2 \text{ lần}} = |^{n^2}$.

Ta có dẫn xuất đầy đủ cho ω là

$$D = (I, ABBDF, A^{(n-1)^2}B^{2(n-1)}DF, A^{(n-1)^2}B^{2(n-1)}|, A^{(n-1)^2}|^{2(n-1)}|, |^{(n-1)^2}|^{2(n-1)}| = |^{(n-1)+2(n-1)+1} = |^{n^2}).$$

§5. MỘT SỐ TÍNH CHẤT CỦA VĂN PHẠM

Định nghĩa 10: (Đồng lực của dẫn xuất)

Cho văn phạm G = <Σ, Δ, I, R> và hai dãy dẫn xuất $D = (\omega_0, \omega_1, ..., \omega_k)$ và $D' = (\omega_0', \omega_1', ..., \omega_m')$ trong văn phạm G. Hai dẫn xuất trên được gọi là *đồng lực* nếu $\omega_0 = \omega_0'$ và $\omega_k = \omega_m'$.

Định nghĩa 11: (Dẫn xuất không lặp)

Cho văn phạm G = <Σ, Δ, I, R> và dẫn xuất $D = (\omega_0, \omega_1, ..., \omega_n)$. Dẫn xuất D được gọi là *dẫn xuất không lặp* nếu không tồn tại cặp ω_i, ω_j $(i \neq j)$ mà $\omega_i = \omega_j$.

Định lý 1: Đối với một dẫn xuất bất kỳ trong văn phạm tuỳ ý, tồn tại một dẫn xuất không lặp đồng lực với nó.

Chứng minh: Giả sử $D = (\omega_0, \omega_1, ..., \omega_{i-1}, \omega_i, \omega_{i+1}, ..., \omega_k)$. Rõ ràng đây là một dẫn xuất đồng lực với chính nó. Xét các trường hợp sau:

– Trong D không có cặp ω_i, ω_j với $i \neq j$ mà $\omega_i = \omega_j$. Khi đó D chính là dẫn xuất không lặp đồng lực với nó.

– Trong D có một cặp $\omega_i = \omega_j$ mà $i \neq j$. Khi đó xét dẫn xuất:

$$D' = (\omega_0, \omega_1, ..., \omega_{i-1}, \omega_j, \omega_{j+1}, ..., \omega_k).$$

Rõ ràng đây là một dẫn xuất đồng lực với D vì D' nhận được từ D bằng cách bỏ đi $\omega_i, \omega_{i+1}, ..., \omega_{j-1}$ còn ω_0, ω_k vẫn giữ nguyên.

Nếu trong D' vẫn còn cặp $\omega_h = \omega_p$ $(h \neq p)$ thì lặp lại quá trình trên cho tới khi trong D' gồm các xâu khác nhau từng đôi một. Khi đó ta được một dẫn xuất không lặp với dẫn xuất D.

Định lý 2: Đối với văn phạm G bất kỳ bao giờ cũng xây dựng được một văn phạm G' tương đương với G, tức là L(G) = L(G').

Chứng minh: G = <Σ, Δ, I, R> là một văn phạm nào đó.

Với mỗi $a \in \Sigma$ ta lập ký hiệu mới $\bar{a} \notin \Sigma \cup \Delta$.

Đặt $\Sigma_1 = \{\bar{a} : a \in \Sigma\}$ và $R_1 = \{\bar{a} \to a : a \in \Sigma\}$. Còn R_2 là tập các quy tắc nhận được từ R bằng cách thay thế các ký hiệu cơ bản a có trong quy tắc của R bởi các ký hiệu mới tương ứng $\bar{a} \notin \Sigma \cup \Delta$.

Khi đó ta xây dựng văn phạm G' = <Σ', Δ', I', R'> như sau:

$$\Sigma' = \Sigma,\ \Delta' = \Sigma_1 \cup \Delta,\ I' = I,\ R' = R_1 \cup R_2.$$

Khi đó ta có L(G) = L(G').

Thật vậy, trước hết chứng minh bao hàm thức $L(G) \subseteq L(G')$.

Xét bất kỳ $P \in L(G)$, khi đó tồn tại dẫn xuất đầy đủ

$$D = (I, \omega_1, \omega_2, ..., \omega_k = P) \text{ trong } G.$$

Nếu trong D ta thay đổi mỗi quy tắc thuộc R mà nó được áp dụng bởi quy tắc tương ứng trong R_2 thì ta được $D' = (I, \omega_1', \omega_2', ..., \omega_k' = P')$ là một dẫn xuất trong G'. Tiếp tục áp dụng các quy tắc trong R_1 của G' đối với P' ta có dẫn xuất $D'' = (I, \omega_1', \omega_2', ..., \omega_k' = P', ..., \omega_{k+m}' = P)$. Nhận thấy D'' là một dẫn xuất đầy đủ trong G', do đó $P \in L(G')$. Vì P là tuỳ ý trong L(G) suy ra $L(G) \subseteq L(G')$.

Ta cần chứng minh bao hàm thức ngược lại, $L(G') \subseteq L(G)$. Giả sử $P \in L(G')$, khi đó tồn tại trong G' một dẫn xuất đầy đủ

$$D' = (I, \omega_1', \omega_2', ..., \omega_k' = P), \text{ ở đây } P \in \Sigma^*.$$

Trong D' ta thay mọi ký hiệu $\bar{a} \in \Sigma_1$ bởi $a \in \Sigma$. Khi đó sẽ nhận được $D = (I, \omega_1, \omega_2, ..., \omega_k = P)$ là dẫn xuất của P trong G, tức là $P \in L(G)$. Từ đó suy ra $L(G) \subseteq L(G')$. Vậy ta đã chứng minh được L(G) = L(G'), có nghĩa là G tương đương với G'.

Giả sử Σ là bảng chữ cái nào đó. Như đã định nghĩa, một ngôn ngữ trên bảng chữ cái Σ là một tập nào đó các xâu trên Σ, tức là một tập con bất kỳ của Σ^* là một ngôn ngữ trên Σ.

Trên lớp các ngôn ngữ ta có thể định nghĩa phép hợp các ngôn ngữ và phép giao các ngôn ngữ tương tự như trong lý thuyết tập hợp. Nghĩa là nếu L_1, L_2 là các ngôn ngữ trên Σ thì phép hợp của chúng được ký hiệu là $L_1 \cup L_2$ và được xác định bởi:

$$L_1 \cup L_2 = \{\omega : \omega \in L_1 \text{ hoặc } \omega \in L_2\}$$

và giao của chúng được ký hiệu là $L_1 \cap L_2$ và được xác định bởi:

$$L_1 \cap L_2 = \{\omega : \omega \in L_1 \text{ và } \omega \in L_2\}.$$

Bây giờ ta định nghĩa phép nhân ngôn ngữ L_1 với ngôn ngữ L_2 là ngôn ngữ được ký hiệu bởi $L_1.L_2$ và được định nghĩa như sau:

$$L_1.L_2 = \{\alpha\beta : \alpha \in L_1 \text{ và } \beta \in L_2\}$$

trong đó $\alpha\beta$ là kết quả nối ghép của hai xâu α và β.

Chú ý: Nói chung $L_1.L_2 \neq L_2.L_1$, hay phép nhân không có tính giao hoán nhưng nó có tính kết hợp, tức là $L_1.(L_2.L_3) = (L_1.L_2).L_3$ và có tính phân bố với phép hợp và phép giao, tức là:

$$L_1.(L_2 \cup L_3) = L_1.L_2 \cup L_1.L_3;$$
$$(L_1 \cup L_2).L_3 = L_1.L_3 \cup L_2.L_3;$$
$$L_1.(L_2 \cap L_3) = (L_1.L_2) \cap (L_1.L_3);$$
$$(L_1 \cap L_2).L_3 = (L_1.L_3) \cap (L_2.L_3).$$

Định nghĩa 12: (Tính đóng đối với phép hợp, giao và nhân)

Giả sử L_1, L_2 là hai ngôn ngữ bất kỳ được sinh bởi văn phạm và θ là một phép tính nào đó trên lớp các ngôn ngữ. Nếu thực hiện các phép tính θ đối với hai ngôn ngữ L_1, L_2 ta thu được một ngôn ngữ mới, ký hiệu là $L_1\theta L_2$, ngôn ngữ mới này cũng được sinh ra bởi văn phạm nào đó thì ta nói phép tính θ là đóng đối với lớp ngôn ngữ do văn phạm sinh ra.

Định lý 3: Lớp ngôn ngữ sinh ra bởi văn phạm là đóng đối với các phép hợp, giao và nhân ngôn ngữ.

Chứng minh: Giả sử rằng L_1, L_2 là các ngôn ngữ được sinh ra bởi văn phạm G_1, G_2, hay $L_1 = L(G_1)$ và $L_2 = L(G_2)$. Ta cần chỉ ra là có các văn phạm G_3, G_4, G_5 sao cho:

$$L(G_3) = L_1 \cup L_2;\ L(G_4) = L_1 \cap L_2;\ L(G_5) = L_1.L_2.$$

Giả sử $G_1 = \langle\Sigma_1, \Delta_1, I_1, R_1\rangle$ và $G_2 = \langle\Sigma_2, \Delta_2, I_2, R_2\rangle$.

a) Ta xây dựng văn phạm G_3 sinh ra ngôn ngữ $L_1 \cup L_2$ như sau:

$$G_3 = \langle\Sigma, \Delta, I, R\rangle,$$

ở đây: $\Sigma = \Sigma_1 \cup \Sigma_2$;

$\Delta = \Delta_1 \cup \Delta_2 \cup \{I\}$;

$R = R_1 \cup R_2 \cup \{I \to I_1, I \to I_2\}$.

Ta chứng minh $L(G_3) = L_1 \cup L_2$.

Thật vậy, trước hết chứng minh bao hàm thức $L(G_3) \subseteq L_1 \cup L_2$. Giả sử $P \in L(G_3)$, khi đó tồn tại dẫn xuất đầy đủ $D = (\omega_0, \omega_1, ..., \omega_k)$, ở đây $\omega_0 = I$, $\omega_k = P \in \Sigma^* = (\Sigma_1 \cup \Sigma_2)^*$. Từ cách xây dựng văn phạm G_3 ta suy ra $\omega_1 = I_1$ hoặc $\omega_1 = I_2$.

Giả sử $\omega_1 = I_1$, khi đó dẫn xuất $(\omega_1, \omega_2, ..., \omega_k = P)$ có dạng $(I_1, \omega_2, \omega_3, ..., \omega_k = P)$ là dẫn xuất đầy đủ của P trong văn phạm $G_1 = \langle\Sigma_1, \Delta_1, I_1, R_1\rangle$, hay $P \in L_1 = L(G_1)$. Vậy $P \in L_1 \cup L_2$, tức là $L(G_3) \subseteq L_1 \cup L_2$.

Trường hợp khi $\omega_1 = I_2$ thì $P \in L_2$ và do đó $P \in L_1 \cup L_2$, ta cũng có $L(G_3) \subseteq L_1 \cup L_2$.

Bây giờ ta chứng minh bao hàm thức ngược lại: $L_1 \cup L_2 \subseteq L(G_3)$.

Thật vậy, giả sử $P \in L_1 \cup L_2$, khi đó $P \in L_1$ hoặc $P \in L_2$.

Chẳng hạn, $P \in L_1 = L(G_1)$, theo định nghĩa ta có dẫn xuất đầy đủ $D = (I_1, \omega_1, \omega_2, ..., \omega_k = P)$ với $P \in \Sigma_1^*$. Khi đó dẫn xuất $D' = (I, I_1, \omega_1, ..., \omega_k = P)$ là dẫn xuất đầy đủ trong văn phạm G_3 vì $P \in \Sigma_1^*$ thì $P \in (\Sigma_1 \cup \Sigma_2)^*$. Vậy $L_1 \cup L_2 \subseteq L(G_3)$.

Tóm lại: $L(G_3) = L_1 \cup L_2$, tức là lớp ngôn ngữ sinh ra bởi văn phạm là đóng đối với phép hợp.

b) Từ các văn phạm G_1, G_2 ta xây dựng văn phạm G_4 sao cho $L(G_4) = L_1 \cap L_2$ như sau :

$$G_4 = <\Sigma, \Delta, I, R> \text{ với } \Sigma = \Sigma_1 \cap \Sigma_2;$$
$$\Delta = \Delta_1 \cup \Delta_2 \cup \Gamma_1 \cup \Gamma_2 \cup \{I\};$$
$$R = R'_1 \cup R'_2 \cup \{I \to I_1I_2\} \cup R''_1 \cup R''_2$$

ở đây: – $\Gamma_1 = \left\{\overline{a} : a \in \Sigma_1\right\}$ và $\Gamma_2 = \left\{\overline{b} : b \in \Sigma_2\right\}$ $(\Gamma_1 \cap \Gamma_2 = \varnothing)$;

– R''_1 là tập các quy tắc nhận được từ tập quy tắc R_1 bằng cách thay các ký hiệu cơ bản trong Σ_1 mà chúng có mặt trong quy tắc của R_1 bởi ký hiệu mới đối ngẫu với nó.

– R''_2 là tập các quy tắc nhận được từ tập quy tắc R_2 bằng cách thay các ký hiệu cơ bản trong Σ_2 mà chúng có mặt trong quy tắc của R_2 bởi ký hiệu mới đối ngẫu với nó.

– Các tập quy tắc R'_1, R'_2 được xác định như sau:

$$R'_1 = \left\{\overline{a}\,\overline{b} \to \overline{b}\,\overline{a} : a \in \Sigma_1,\ b \in \Sigma_2\right\},$$

$$R'_2 = \left\{\overline{a}\,\overline{a} \to a : a \in \Sigma_1 \cap \Sigma_2\right\}.$$

Kiểm tra lại ta thấy: $L(G_4) = L(G_1) \cap L(G_2) = L_1 \cap L_2$.

Vậy lớp ngôn ngữ do văn phạm sinh ra đóng đối với phép giao.

c) Từ các văn phạm G_1, G_2 ta xây dựng văn phạm G_5 sinh ra lớp ngôn ngữ $L_1.L_2$ như sau:

$G_5 = <\Sigma, \Delta, I, R>$, ở đây: $\Sigma = \Sigma_1 \cup \Sigma_2$; $\Delta = \Delta_1 \cup \Delta_2 \cup \{I\}$;

$$R = R_1 \cup R_2 \cup \{I \to I_1I_2\}.$$

Ta sẽ chứng minh $L(G_5) = L_1.L_2$.

Trước hết ta chỉ ra bao hàm thức $L(G_5) \subseteq L_1.L_2$.

Thật vậy, xét bất kỳ $P \in L(G_5)$, khi đó trong G_5 tồn tại dẫn xuất đầy đủ $D = (I, I_1, I_2, \omega_1 I_2, \omega_2 I_2, \omega_3 I_2, ..., P_1 I_2, P_1\omega_1', P_1\omega_2', ..., P_1P_2 = P)$.

Ở đây, đoạn từ I đến P_1I_2 chỉ áp dụng các quy tắc trong R_1 nên $P_1 \in L(G_1) = L_1$; còn đoạn từ $P_1\omega_1'$ đến $P_1P_2 = P$ có được là do áp dụng các quy tắc trong R_2 nên $P_2 \in L(G_2) = L_2$. Vậy $P = P_1P_2 \in L_1.L_2$.

Vậy, bao hàm thức $L(G_5) \subseteq L_1.L_2$ đã được chứng minh.

Tiếp theo ta chứng minh bao hàm thức ngược lại: $L_1.L_2 \subseteq L(G_5)$.

Giả sử $P \in L_1.L_2$, do đó tồn tại $P_1 \in L_1$, $P_2 \in L_2$ sao cho $P = P_1P_2$. Vì $P_1 \in L_1 = L(G_1)$ nên tồn tại dẫn xuất đầy đủ $D_1 = (I_1, \omega_1, \omega_2, ..., \omega_k = P_1)$ trong G_1 với $P_1 \in \Sigma_1^*$; $P_2 \in L_2 = L(G_2)$ nên cũng tồn tại dẫn xuất đầy đủ $D_2 = (I_2, \omega_1', \omega_2', ..., \omega_m' = P_2)$ trong G_2 với $P_2 \in \Sigma_2^*$.

Lập dẫn xuất $D = (I, I_1I_2, \omega_1 I_2, \omega_2 I_2, ..., P_1I_2, P_1\omega_1', P_1\omega_2', ..., P_1P_2 = P)$. Đây là dẫn xuất đầy đủ trong văn phạm G_5 và $P = P_1P_2 \in \Sigma^* = (\Sigma_1 \cup \Sigma_2)^*$, hay $P \in L(G_5)$. Vậy ta có $L_1.L_2 \subseteq L(G_5)$.

Tóm lại: $L_1.L_2 = L(G_1).L(G_2) = L(G_5)$.

Vậy lớp ngôn ngữ do văn phạm sinh ra đóng đối với phép nhân.

Định lý được chứng minh.

Ta minh hoạ định lý trên bằng ví dụ sau đây:

Ví dụ 1: Cho văn phạm $G_1 = <\Sigma_1, \Delta_1, I_1, R_1>$ với:

$$\Sigma_1 = \{a, b\};\ \Delta_1 = \{I_1\};\ R_1 = \{I_1 \to aI_1b,\ I_1 \to ab\}$$

và $G_2 = <\Sigma_2, \Delta_2, I_2, R_2>$ với $\Sigma_2 = \{c\}$;

$$\Delta_2 = \{I_2\},\ R_2 = \{I_1 \to cI_2,\ I_2 \to c\}.$$

Xây dựng văn phạm $G_3 = <\Sigma, \Delta, I, R>$ sao cho

$L(G_3) = L(G_1) \cup L(G_2)$ và G_5 sao cho $L(G_5) = L(G_1).L(G_2)$.

Ta đã biết: $L(G_1) = \left\{a^n b^n : a^n b^n \in \Sigma_1^*,\ n \geq 1\right\}$;

$$L(G_2) = \left\{c^m : c^m \in \Sigma_2^*,\ m \geq 1\right\}.$$

a) $L(G_3) = L(G_1) \cup L(G_2) = \left\{a^n b^n, c^m : n, m \geq 1\right\}$.

Ta xây dựng văn phạm G_3 như sau: $\Sigma = \Sigma_1 \cup \Sigma_2 = \{a, b, c\}$;

$\Delta = \Delta_1 \cup \Delta_2 \cup \{I\}$, $I \notin \Delta_1 \cup \Delta_2$ được gọi là ký hiệu ban đầu của G_3;

$R = R_1 \cup R_2 \cup \{I \to I_1, I \to I_2\}$

$= \{I_1 \to aI_1b, I_1 \to ab, I_1 \to cI_2, I_2 \to c, I \to I_1, I \to I_2\}$.

Ta chứng minh $L(G_3) = L(G_1) \cup L(G_2) = \left\{a^nb^n, c^m : n, m \geq 1\right\}$.

– Xét bao hàm thức $L(G_3) \subseteq L(G_1) \cup L(G_2)$.

Giả sử $P \in L(G_3)$, tức tồn tại dẫn xuất đầy đủ $D = (I, \omega_1, \omega_2, ..., \omega_n = P)$, ở đây $\omega_1 = I_1$ hoặc $\omega_1 = I_2$ và $P \in \Sigma^*$. Trường hợp $\omega_1 = I_1$, khi đó $D = (I, I_1, \omega_2 = aI_1b, \omega_3 = a^2I_1b^2, ..., \omega_n = a^{n-1}I_1b^{n-1}, \omega_{n+1} = a^nb^n = P)$ là dẫn đầy đủ của $P = a^nb^n$ trong G_1, nên $P \in L(G_1)$ hay $P \in L(G_1) \cup L(G_2)$. Trường hợp $\omega_1 = I_2$, khi đó $D = (I, I_2, \omega_2 = cI_2, \omega_3 = c^2I_2, ..., \omega_m = c^{m-1}I_2, \omega_{m+1} = c^m = P)$ là dẫn xuất đầy đủ của P trong G_2 hay $P \in L(G_2)$, tức là $P \in L(G_1) \cup L(G_2)$.

Vậy: $\quad L(G_3) \subseteq L(G_1) \cup L(G_2) \quad (1)$

– Xét bao hàm thức ngược lại: $L(G_1) \cup L(G_2) \subseteq L(G_3)$.

Giả sử $P \in L(G_1) \cup L(G_2)$. Không giảm tính tổng quát, giả sử $P \in L(G_1)$. Khi đó dẫn xuất đầy đủ của P trong G_1 có dạng

$$D' = (I_1, \omega_1 = aI_1b, \omega_2 = a^2I_1b^2, ..., \omega_n = a^nb^n = P) \text{ và } P \in \Sigma^*.$$

Từ D' ta có dẫn xuất đầy đủ của P là D trong G_3 là

$$D = (I, I_1, \omega_1 = aI_1b, \omega_2 = a^2I_1b^2, ..., \omega_n = a^nb^n = P).$$

Vậy $P \in G_3$. Trường hợp $P \in G_2$ chứng minh tương tự.

Tóm lại ta có: $\quad L(G_1) \cup L(G_2) \subseteq L(G_3) \quad (2)$

Từ (1) và (2) ta có $L(G_3) = L(G_1) \cup L(G_2)$.

b) Xây dựng $G_5 = \langle\Sigma, \Delta, I, R\rangle$ sao cho

$$L(G_5) = L(G_1).L(G_2) = \left\{a^nb^nc^m : m, n \geq 1\right\}.$$

Ta xây dựng G_5 như sau:

$\Sigma = \Sigma_1 \cup \Sigma_2 = \{a, b, c\}$; $\Delta = \Delta_1 \cup \Delta_2 \cup \{I\}$, $I \notin \Delta_1 \cup \Delta_2$;

$R = R_1 \cup R_2 \cup \{I \to I_1I_2\}$.

– Chứng minh bao hàm thức $L(G_5) \subseteq L(G_1).L(G_2)$.

Giả sử $P \in L(G_5)$, khi đó tồn tại dẫn xuất đầy đủ D của P trong G_5 là

$D = (I, I_1I_2, aI_1bI_2, a^2I_1b^2I_2, ..., a^nb^nI_2 = P_1I_2, P_1cI_2, P_1c^2I_2, ..., P_1c^m = P_1P_2)$

ở đây: $P_1 = a^nb^n \in L(G_1)$; $P_2 = c^m \in L(G_2)$, hay $P = P_1P_2 \in L(G_1).L(G_2)$.

Vậy bao hàm thức $L(G_5) \subseteq L(G_1).L(G_2)$ được chứng minh.

– Chứng minh bao hàm thức ngược lại: $L(G_1).L(G_2) \subseteq L(G_5)$.

Thật vậy, giả sử $P \in L(G_1).L(G_2)$ hay $P = P_1P_2$, ở đây: $P_1 \in L(G_1)$; $P_2 \in L(G_2)$.

Vì $P_1 \in L(G_1)$ nên tồn tại dẫn xuất đầy đủ

$D_1 = (I_1, \omega_1 = aI_1b, \omega_2 = a^2I_1b^2, ..., \omega_n = a^nb^n = P_1)$ trong G_1, ở đây $P_1 \in \Sigma_1^*$.

$P_2 \in L(G_2)$ nên cũng tồn tại dẫn xuất đầy đủ

$$D_2 = (I_2, cI_2, c^2I_2, ..., c^m = P_2) \text{ trong } G_2, \text{ ở đây } P_2 \in \Sigma_2^*.$$

Lập dẫn xuất $D = (I, I_1I_2, aI_1bI_2, a^2I_1b^2I_2, ..., a^nb^nI_2 = P_1I_2, P_1cI_2, P_1c^2I_2, ..., P_1c^m = P_1P_2)$, ở đây: $P_1 = a^nb^n \in L(G_1)$; $P_2 = c^m \in L(G_2)$ hay $P = P_1P_2 \in L(G_1).L(G_2)$.

Vậy bao hàm thức $L(G_1).L(G_2) \subseteq L(G_5)$ đã được chứng minh.

Tóm lại, với G_5 xây dựng như trên ta có: $L(G_5) = L(G_1).L(G_2)$.

Chú ý: Vì $\Sigma_1 \cap \Sigma_2 = \varnothing$ nên ở đây ta không xét văn phạm G_4 trong định lý 3.

BÀI TẬP

1. Cho bảng chữ cái $\Sigma = \{0, 1\}$. Ta ký hiệu Σ^* là tập tất cả các xâu (kể cả xâu rỗng ($\wedge$)).

Hãy viết ngôn ngữ Σ^* dưới dạng liệt kê theo thứ tự độ dài tăng dần và trong các xâu có cùng độ dài thì theo thứ tự từ điển.

2. Tìm cách biểu diễn hữu hạn cho các ngôn ngữ vô hạn sau:

a) $L_1 = \{\wedge, ab, aabb, aaabbb, ...\}$.

b) $L_2 = \{\wedge, 0, 1, 00, 01, 11, 000, 001, 010, 011, 100, 101, 110, 111, ...\}$.

3. Ngôn ngữ L_1 và L_2 trong bài tập 2 có biểu diễn thông qua văn phạm. Hãy xây dựng hai văn phạm đó.

4. Cho các văn phạm sau đây:

1) $G = <\Sigma, \Delta, I, R>$ với tập quy tắc sinh là

$$R = \{I \to ABC, AB \to iADj, Dij \to jDi, DiC \to BiC, iB \to Bi, AB \to \wedge, C \to \wedge\} \text{ với } i, j \in \{a, b\} = \Sigma.$$

2) $G = <\Sigma, \Delta, I, R>$ với tập quy tắc sinh là

$$R = \{I \to II, I \to aIb, I \to bIa, I \to ab, I \to ba\}.$$

3) $G = <\Sigma, \Delta, I, R>$ với tập quy tắc sinh là

$$R = \{I \rightarrow aI, I \rightarrow a : a \in \Sigma\}.$$

Trong các văn phạm trên:

a) Văn phạm nào là văn phạm chính quy, văn phạm nào là văn phạm phi ngữ cảnh, văn phạm nào là văn phạm ngữ cấu?

b) Viết lại từng văn phạm dưới dạng đầy đủ theo định nghĩa của văn phạm.

c) Tìm các ngôn ngữ do các văn phạm trên sinh ra.

5. Cho ngôn ngữ $L = \{\omega\hat{\omega} : \omega \in \{0, 1\}^*, \hat{\omega}$ là ảnh gương của $\omega\}$.

Xây dựng văn phạm phi ngữ cảnh G sao cho $L(G) = L$.

6. Cho ngôn ngữ $L = \{a^n b^n c^m : n, m \geq 1\}$.

Xây dựng văn phạm phi ngữ cảnh G sao cho $L(G) = L$.

7. Ngôn ngữ L trên từ điển $\{a, b\}$ gồm tất cả các xâu không rỗng mà trong xâu đó số ký tự a và số ký tự b là bằng nhau. Vì sao ngôn ngữ L là ngôn ngữ phi ngữ cảnh?

8. Cho văn phạm phi ngữ cảnh với tập quy tắc sinh là

$$R = \{I \rightarrow aIa, I \rightarrow aa : a \in \Sigma = \{a_1, a_2, ..., a_n\}\}.$$

a) Tìm ngôn ngữ do văn phạm sinh ra.

b) Hãy chỉ ra dẫn xuất đầy đủ của xâu $\omega = a_3a_2a_3a_1a_2a_2a_3a_2a_3$ trong văn phạm trên.

9. Cho ngôn ngữ $L = \{\omega b\omega : \omega \in \Sigma^* = \{a_1, a_2, ..., a_k, b\}^*\}$.

Xây dựng một văn phạm phi ngữ cảnh G sao cho $L(G) = L$.

10. Xây dựng các văn phạm chính quy sinh ra các tập các số tự nhiên $L_1 = \{1, 2, 3, ..., n\}$ và tập $L_2 = \{\text{if } a = b \text{ then } c\}$ tương ứng.

11. Cho các văn phạm:

1) G_1 với tập quy tắc $R_1 = \{I \rightarrow aI, I \rightarrow Ib, I \rightarrow aIb, I \rightarrow c\}$.

2) G_2 với tập quy tắc $R_2 = \{I \rightarrow II, I \rightarrow a, I \rightarrow b\}$.

3) G_3 với tập quy tắc $R_3 = \{I \rightarrow aA, I \rightarrow bB, I \rightarrow c, A \rightarrow Ia, B \rightarrow Ib\}$.

4) G_4 với tập quy tắc $R_4 = \{I \rightarrow AB, A \rightarrow Ic, A \rightarrow a, B \rightarrow dB, B \rightarrow b\}$.

5) G_5 với tập quy tắc $R_5 = \{I \rightarrow IaI, I \rightarrow b\}$.

6) G_6 với tập quy tắc $R_6 = \{I \rightarrow aII, I \rightarrow b\}$.

7) G_7 với tập quy tắc $R_7 = \{I \rightarrow AA, A \rightarrow aAa, A \rightarrow bAb, A \rightarrow c\}$.

a) Hãy phân loại 7 văn phạm trên theo nhóm 0, 1, 2, 3 của Chomsky.

b) Tìm $L(G_i)$ ($i = \overline{1,7}$) và gọi tên ngôn ngữ theo phân loại trên.

12. *Bài toán 1:* Cho văn phạm $G = <\Sigma, \Delta, I, R>$. Hãy xây dựng văn phạm $G' = <\Sigma', \Delta', I', R'>$ sao cho $L(G') = L(G)$, hay $G' \approx G$.

Ta xây dựng $G' = <\Sigma', \Delta', I', R'>$ theo các bước sau:

- Bước 1: Đặt $\Sigma' = \Sigma$;
- Bước 2: Chọn $I' = I$;
- Bước 3: Đặt $\Delta' = \Delta \cup \overline{\Sigma}$, ở đây $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$ ($\overline{a}$ là phần tử đối ngẫu của a, dĩ nhiên $\overline{a} \notin \Delta \cup \Sigma$);
- Bước 4: Đặt $R' = \hat{R} \cup \overline{R}$, ở đây $\hat{R}$ = tập các quy tắc trong R mà mọi ký tự $a \in \Sigma$ được thay bởi $\overline{a} \in \Sigma$; còn $\overline{R} = \{\overline{a} \to a \mid a \in \Sigma\}$. Khi đó $G' \approx G$.

Áp dụng bài toán 1 để tìm các văn phạm tương đương G' với các văn phạm G cho trong các bài từ 4 đến 11.

13. *Bài toán 2:* Cho hai văn phạm $G_1 = <\Sigma_1, \Delta_1, I_1, R_1>$ và $G_2 = <\Sigma_2, \Delta_2, I_2, R_2>$. Xây dựng văn phạm $G = <\Sigma, \Delta, I, R>$ sao cho $L(G) = L(G_1) \cup L(G_2)$.

Các bước xây dựng G như sau:

- Bước 1: Đặt $\Sigma = \Sigma_1 \cup \Sigma_2$;
- Bước 2: Đặt $\Delta = \Delta_1 \cup \Delta_2 \cup \{I\}$;
- Bước 3: Đặt: $R = R_1 \cup R_2 \cup \{I \to I_1, I \to I_2\}$.

Khi đó ta có $L(G) = L(G_1) \cup L(G_2)$.

a) Phát biểu bài toán 2 cho trường hợp tổng quát: Cho các văn phạm $G_i = <\Sigma_i, \Delta_i, I_i, R_i>$ ($i = \overline{1, n}$). Xây dựng $G = <\Sigma, \Delta, I, R>$ sao cho
$$L(G) = \bigcup_{i=1}^{n} L(G_i).$$

b) Cho hai văn phạm $G_1 = <\Sigma_1, \Delta_1, I_1, R_1>$ với $\Sigma_1 = \{a, b\}$; $\Delta_1 = \{I_1\}$; $R_1 = \{I_1 \to aI_1b, I_1 \to ab, I_1 \to ba\}$ và $G_2 = <\Sigma_2, \Delta_2, I_2, R_2>$ với $\Sigma_2 = \{c, d, h\}$; $\Delta_2 = \{I_2, A\}$; $R_2 = \{I_2 \to I_2h, I_2 \to A, A \to cAd, A \to cd\}$. Xây dựng $G = <\Sigma, \Delta, I, R>$ sao cho $L(G) = L(G_1) \cup L(G_2)$.

c) Cho $G_3 = <\Sigma_3, \Delta_3, I_3, R_3>$ với $\Sigma = \{0, 1\}$; $\Delta_3 = \{I_3, A, B\}$;
$$R_3 = \{I_3 \to 0I_3, I_3 \to A, A \to 1A, A \to B, B \to 0B, B \to 0\}.$$
Xây dựng $G = <\Sigma, \Delta, I, R>$ sao cho

$$L(G) = L(G_1) \cup L(G_2) \cup L(G_3),$$

ở đây G_1 và G_2 được cho trong câu b.

d) Hãy xây dựng văn phạm $G = <\Sigma, \Delta, I, R>$ sao cho $L(G) = \bigcup_{i=1}^{7} L(G_i)$,

với G_i cho trong bài 11.

14. *Bài toán 3:* Cho hai văn phạm $G_1 = <\Sigma_1, \Delta_1, I_1, R_1>$ và $G_2 = <\Sigma_2, \Delta_2, I_2, R_2>$. Xây dựng văn phạm $G = <\Sigma, \Delta, I, R>$ sao cho $L(G) = L(G_1).L(G_2)$.

Việc xây dựng G được thực hiện theo các bước sau:

– Bước 1: Đặt $\Sigma = \Sigma_1 \cup \Sigma_2$;

– Bước 2: Đặt $\Delta = \Delta_1 \cup \Delta_2 \cup \{I\}$;

– Bước 3: Đặt $R = R_1 \cup R_2 \cup \{I \to I_1 I_2\}$.

Khi đó ta có $L(G) = L(G_1).L(G_2)$.

a) Phát biểu bài toán trên cho trường hợp tổng quát: Cho $G_i = <\Sigma_i, \Delta_i, I_i, R_i>$ ($i = 1, 2, ..., n$). Xây dựng $G = <\Sigma, \Delta, I, R>$ sao cho

$$L(G) = L(G_1).L(G_2)...L(G_n).$$

b) Xây dựng văn phạm G sao cho $L(G) = L(G_1).L(G_2)$, ở đây G_1 và G_2 được cho trong câu b của bài 13.

c) Xây dựng văn phạm G sao cho $L(G) = L(G_3).L(G_1).L(G_2)$, ở đây G_1 và G_2 được cho trong câu b của bài 13, còn G_3 cho trong câu c của bài 13.

d) Xây dựng văn phạm G sao cho

$$L(G) = L(G_1).L(G_2) \cup L(G_3).L(G_1).L(G_2),$$

ở đây: G_1, G_2, G_3 là các văn phạm ở phần b và c của bài 13.

e) Hãy xây dựng văn phạm G sao cho

$$L(G) = L(G_1).L(G_2).L(G_3).L(G_4).L(G_5).L(G_6).L(G_7),$$

ở đây G_i ($i = \overline{1,7}$) được cho trong bài 11.

15. *Bài toán 4:* Cho hai văn phạm $G_1 = <\Sigma_1, \Delta_1, I_1, R_1>$ và $G_2 = <\Sigma_2, \Delta_2, I_2, R_2>$. Xây dựng văn phạm $G = <\Sigma, \Delta, I, R>$ sao cho $L(G) = L(G_1) \cap L(G_2)$.

Việc xây dựng G theo các bước sau đây:

– Bước 1: Đặt $\Sigma = \Sigma_1 \cap \Sigma_2$;

– Bước 2: Đặt $\Delta = \Delta_1 \cup \Delta_2 \cup \overline{\Sigma}_1 \cup \overline{\Sigma}_2 \cup \{I\}$;

– Bước 3: Đặt $R = \hat{R}_1 \cup \hat{R}_2 \cup \{I \to I_1 I_2\} \cup R' \cup R''$, ở đây:

$\overline{\Sigma}_1 = \left\{\overline{a} \mid a \in \Sigma_1\right\}$, $\overline{\Sigma}_2 = \left\{\overline{b} \mid b \in \Sigma_2\right\}$;

$\hat{R}_1$ = tập tất cả các quy tắc trong R_1 mà mọi ký tự $a \in \Sigma_1$ đều được thay tương ứng với $\overline{a} \in \Sigma_1$;

$\hat{R}_2$ = tập tất cả các quy tắc trong R_2 mà mọi ký tự $b \in \Sigma_2$ đều được thay bởi ký tự $\overline{b} \in \Sigma_2$;

$R' = \left\{\overline{a}\,\overline{b} \to \overline{b}\,\overline{a} \mid a \in \Sigma_1, b \in \Sigma_2\right\}$;

$R'' = \left\{\overline{a}\,\overline{a} \to a \mid a \in \Sigma_1 \cap \Sigma_2\right\}$.

a) Áp dụng bài toán 4: Cho:

$G_1 = <\Sigma_1, \Delta_1, I_1, R_1>$ với: $\Sigma_1 = \{a, b\}$; $R_1 = \{I_1 \to aI_1b, I_1 \to ba\}$;
$\Delta_1 = \{I_1\}$.

$G_2 = <\Sigma_2, \Delta_2, I_2, R_2>$ với: $\Sigma_2 = \{a, b, c\}$;
$R_2 = \{I_2 \to I_2c, I_2 \to aI_2b, I_2 \to ba\}$;
$\Delta_2 = \{I_2\}$.

Xây dựng $G = <\Sigma, \Delta, I, R>$ sao cho $L(G) = L(G_1) \cap L(G_2)$.

b) Phát biểu bài toán 4 trong trường hợp tổng quát:

Cho $G_i = <\Sigma_i, \Delta_i, I_i, R_i>$ ($i = \overline{1\ n}$). Xây dựng $G = <\Sigma, \Delta, I, R>$ sao cho

$$L(G) = L(G_1) \cap L(G_2) \cap \ldots \cap L(G_n) = \bigcap_{i=1}^{n} L(G_i).$$

c) Xây dựng các văn phạm phi ngữ cảnh G_1, G_2, G_3. Tiếp theo xây dựng G sao cho $L(G) = L(G_1) \cap L(G_2) \cap L(G_3)$.

16. Cho: $G_1 = <\Sigma_1, \Delta_1, I_1, R_1>$ với:

$\Sigma_1 = \{a\}$; $\Delta_1 = \{I_1\}$; $R_1 = \{I_1 \to aI_1, I_1 \to a\}$.

$G_2 = <\Sigma_2, \Delta_2, I_2, R_2>$ với:

$\Sigma_2 = \{a, b\}$; $\Delta_2 = \{I_2\}$; $R_2 = \{I_2 \to aI_2b, I_2 \to a\}$.

a) Xây dựng $G = <\Sigma, \Delta, I, R>$ sao cho:

1) $L(G) = L(G_1) \cup L(G_2)$;

2) $L(G) = L(G_1) \cap L(G_2)$;

3) $L(G) = L(G_1) \cup L(G_2) \cup L(G_1) \cap L(G_2)$.

b) Xây dựng văn phạm tương đương với các văn phạm G trong câu a.1, a.2 và a.3.

Chương 10
ÔTÔMAT HỮU HẠN
VÀ NGÔN NGỮ ĐOÁN NHẬN CỦA NÓ

§1. ÔTÔMAT HỮU HẠN (FINITE AUTOMATA – FA)

Ôtômat hữu hạn có thể xem như "máy trừu tượng" để đoán nhận ngôn ngữ. Chúng ta sẽ thấy rằng lớp ngôn ngữ được đoán nhận bởi ôtômat hữu hạn là khá đơn giản, đó chính là lớp ngôn ngữ chính quy do các văn phạm chính quy sinh ra.

1.1. Định nghĩa ôtômat hữu hạn

Định nghĩa 1:

Ôtômat hữu hạn là bộ $M = <\Sigma, Q, \delta, q_0, F>$, trong đó:

Σ là tập hữu hạn khác rỗng các ký hiệu vào;

Q là tập hữu hạn khác rỗng các trạng thái;

$q_0 \in Q$ được gọi là trạng thái ban đầu;

$F \subseteq Q$ được gọi là tập các trạng thái kết thúc;

δ được gọi là hàm chuyển có dạng:

1. Nếu hàm chuyển là ánh xạ $\delta: Q \times \Sigma \to Q$, khi đó M được gọi là *ôtômat đơn định* (Deterministic Finite Automata, viết tắt là DFA).
2. Nếu hàm chuyển là ánh xạ $\delta: Q \times \Sigma \to 2^Q$, khi đó M được gọi là *ôtômat không đơn định* (Nondeterministic Finite Automata, viết tắt là NDFA).

Ta có thể mô tả các bước làm việc của ôtômat hữu hạn $M = <\Sigma, Q, \delta, q_0, F>$ khi cho xâu vào $\omega = x_1x_2x_3 \ldots x_n \in \Sigma^*$ như sau:

$$
\begin{array}{ll}
\text{Xâu vào } \omega: & x_1 \quad x_2 \ldots x_{n-1} \quad x_n \\
 & \uparrow \\
 & q_0 \to \ldots\ldots\ldots\ldots\ldots
\end{array}
$$

Khi bắt đầu làm việc máy ở trạng thái đầu q_0 và đầu đọc đang nhìn vào ô có ký hiệu x_1. Tiếp theo máy chuyển từ trạng thái q_0 dưới tác động của ký hiệu vào x_1 về trạng thái mới $\delta(q_0, x_1) = q_1 \in Q$ và đầu đọc chuyển sang phải 1 ô, tức nhìn vào ô có ký hiệu x_2. Sau đó ôtômat M lại tiếp tục chuyển từ trạng thái q_1 nhờ hàm chuyển δ về trạng thái mới

$$q_2 = \delta(q_1, x_2) = \delta(\delta(q_0, x_1), x_2) = \delta(q_0, x_1x_2) \in Q.$$

Quá trình đó sẽ tiếp tục cho tới khi đầu đọc nhìn vào ký hiệu x_n với trạng thái của máy là $q_{n-1} = \delta(q_0, x_1x_2 \ldots x_{n-1})$.

Hàm chuyển lại đưa máy từ trạng thái q_{n-1} dưới tác động của x_n về trạng thái $q_n = \delta(q_{n-1}, x_n) = \delta(q_0, x_1x_2 \ldots x_n) \in Q$, khi đó ôtômat dừng lại. Nếu $q_n \in F$ thì ta nói rằng ôtômat đã đoán nhận xâu ω. Trong trường hợp ngược lại thì nói rằng ôtômat không đoán nhận xâu ω.

Tập $L(M) = \{\omega : \omega \in \Sigma^* \text{ mà } \delta(q_0, \omega) \in F\}$ được gọi là *ngôn ngữ được đoán nhận bởi ôtômat M*. Tập trạng thái Q trong quá trình tính toán được coi như là bộ nhớ của một ôtômat. Vì Q là hữu hạn nên M được gọi là *ôtômat hữu hạn*.

1.2. Phương pháp biểu diễn ôtômat hữu hạn

Cho ôtômat thực chất là cho hàm chuyển của nó. Hàm chuyển có thể cho dưới dạng bảng chuyển, hoặc cho dưới dạng đồ thị.

a) Phương pháp cho bảng chuyển

Trạng thái	Ký hiệu			
	x_1	x_2	...	x_n
q_1	$\delta(q_1, x_1)$	$\delta(q_1, x_2)$	...	$\delta(q_1, x_n)$
q_2	$\delta(q_2, x_1)$	$\delta(q_2, x_2)$	...	$\delta(q_2, x_n)$
q_3	$\delta(q_3, x_1)$	$\delta(q_3, x_2)$	...	$\delta(q_3, x_n)$
...	...			
...	...			
q_k	$\delta(q_k, x_1)$	$\delta(q_k, x_2)$	...	$\delta(q_k, x_n)$

Nếu $M = <\Sigma, Q, \delta, q_0, F>$ là ôtômat đơn định thì $\delta(q_i, x_j)$ là một trạng thái trong Q, còn nếu nó là ôtômat không đơn định thì $\delta(q_i, x_j)$ là một tập các trạng thái trong Q.

Ví dụ 1: Cho ôtômat đơn định M = <Σ, Q, δ, q_0, F> với Σ = {0, 1}, Q = {q_0, q_1, q_2, q_3}, F = {q_0}, trạng thái ban đầu là q_0, còn hàm chuyển cho dưới dạng bảng:

Trạng thái	Ký hiệu vào	
	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

Cho xâu vào ω = 110101.

Quá trình hoạt động của ôtômat M diễn ra theo quá trình sau:

$$\delta(q_0, 1) = q_1, \quad \delta(q_1, 1) = q_0, \quad \delta(q_0, 0) = q_2,$$
$$\delta(q_2, 1) = q_3, \quad \delta(q_3, 0) = q_1, \quad \delta(q_1, 1) = q_0 \in F.$$

Điều đó chứng tỏ ôtômat M đoán nhận xâu ω = 110101. Dãy trạng thái của M khi cho xâu vào ω = 110101 được biểu diễn như sau:

$$\begin{array}{cccccccc} \omega = & 1 & 1 & 0 & 1 & 0 & 1 & \\ & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ & q_0 \to & q_1 \to & q_0 \to & q_2 \to & q_3 \to & q_1 \to & q_0 \end{array}$$

Hay quá trình trên tương đương với diễn đạt sau:

$$\begin{aligned} \delta(q_0, 110101) &= \delta(\delta(q_0, 1), 10101) = \delta(q_1, 10101) \\ &= \delta(\delta(q_1, 1), 0101) = \delta(q_0, 0101) \\ &= \delta(\delta(q_0, 0), 101) = \delta(q_2, 101) \\ &= \delta(\delta(q_2, 1), 01) = \delta(q_3, 01) \\ &= \delta(\delta(q_3, 0), 1) = \delta(q_1, 1) = q_0 \in F. \end{aligned}$$

Ví dụ 2: Cho ôtômat đơn định M = <Σ, Q, δ, q_0, F> với Σ = {0, 1}, Q = {q_0, q_1, q_2, q_3}, q_0 là trạng thái ban đầu, còn F = {q_3} là tập trạng thái kết thúc.

Hàm chuyển trạng thái được cho bởi bảng sau:

Trạng thái	Ký hiệu vào	
	0	1
q_0	q_1	q_2
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_3	q_3

Giả sử xét các xâu $\omega_1 = 10110$, $\omega_2 = 110$, $\omega_3 = 10011$. Hỏi rằng ôtômat M có đoán nhận các xâu đó hay không?

a) Dãy trạng thái của ôtômat M khi cho ω_1 vào là:

$$\begin{array}{llllll} \omega_1 = 1 & 0 & 1 & 1 & 0 & \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \end{array}$$

$$q_0 \to q_2 \to q_0 \to q_2 \to q_3 \to q_3 \in F.$$

Do đó ôtômat M đoán nhận xâu $\omega_1 = 10110$.

b) Với xâu vào $\omega_2 = 110$, khi đó dãy trạng thái M đối với xâu vào ω_2 sẽ là:

$$\begin{array}{llll} \omega_2 = 1 & 1 & 0 & \\ \uparrow & \uparrow & \uparrow & \uparrow \end{array}$$

$$q_0 \to q_2 \to q_3 \to q_3 \in F$$

và ôtômat cũng đoán nhận xâu $\omega_2 = 110$.

c) Với xâu vào $\omega_3 = 10011$. Ta lập dãy trạng thái của ôtômat M đối với ω_3 như sau:

$$\begin{array}{llllll} \omega_3 = 1 & 0 & 0 & 1 & 1 & \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \end{array}$$

$$q_0 \to q_2 \to q_0 \to q_1 \to q_0 \to q_2 \notin F$$

nên máy M không đoán nhận xâu vào $\omega_3 = 10011$.

Từ các ví dụ trên ta thấy quá trình đoán nhận một xâu nào đó đối với ôtômat M đơn định là quá trình biến đổi trạng thái cuối cùng của ôtômat có phải là trạng thái kết thúc hay không.

Ta có thể mô tả quá trình đoán nhận xâu vào của ôtômat đơn định M như sau:

Thuật toán mô phỏng ôtômat hữu hạn đoán nhận xâu vào:

- Đầu vào:
  - Một xâu ω, kết thúc bởi ký hiệu hết tệp (File) là eof.
  - Một ôtômat hữu hạn M với trạng thái ban đầu q_0 và tập trạng thái kết thúc là F.
- Đầu ra: "Đúng" nếu M đoán nhận xâu ω;

  "Sai" nếu M không đoán nhận xâu ω.
- Thuật toán:

```
Begin
    S := q0;
    C := ký hiệu tiếp theo;
    While C <> eof do
        begin
            S := δ(S, C);
            C := ký hiệu tiếp theo;
        end;
    if S in F return (True)
    else return (False);
End.
```

Ví dụ 3: Cho ôtômat không đơn định $M = <\Sigma, Q, \delta, q_0, F>$ với $\Sigma = \{0, 1\}$, $Q = \{q_0, q_1, q_2, q_3\}$, q_0 là trạng thái ban đầu, $F = \{q_2, q_4\}$ là tập hợp trạng thái kết thúc.

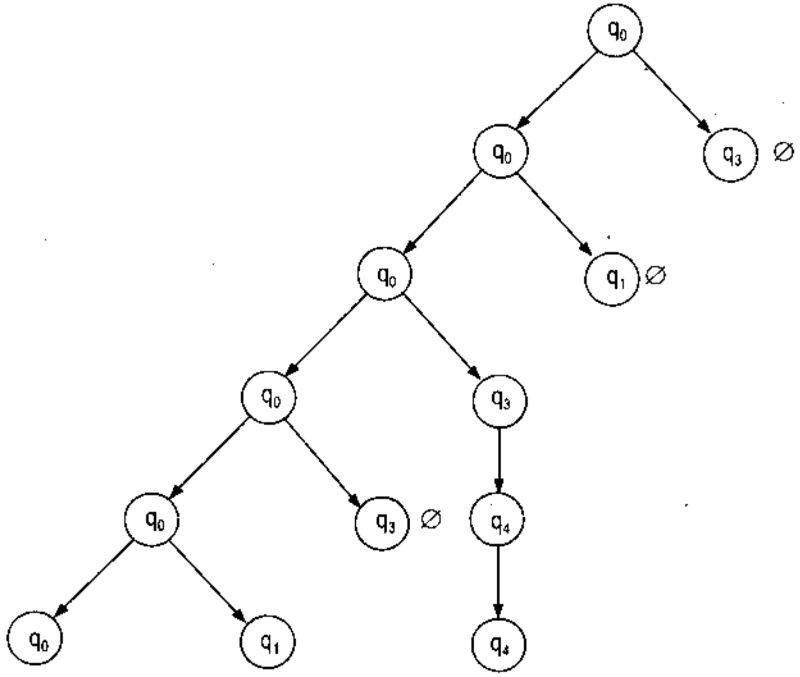
Hàm chuyển $\delta : Q \times \Sigma \rightarrow 2^Q$ được cho ở bảng sau:

Trạng thái	Ký hiệu vào	
	0	1
q_0	$\{q_0, q_3\}$	$\{q_0, q_1\}$
q_1	$\varnothing$	q_2
q_2	q_2	q_2
q_3	q_4	$\varnothing$
q_4	q_4	q_3

Sau đây là quá trình đoán nhận một xâu vào của máy M.

Vì đối với ôtômat không đơn định thì hàm chuyển δ khi máy ở trạng thái q và tín hiệu a là $\delta(q, a) \subseteq Q$ nên sẽ xuất hiện tình trạng rẽ nhánh. Để cho tiện ta có thể lập dãy trạng thái của máy ứng với xâu vào dưới dạng cây mà gốc của nó là trạng thái ban đầu q_0. Trong cây này nếu có một đường đi từ gốc q_0 đến một lá chứa trạng thái kết thúc thì ta nói rằng máy M đoán nhận xâu vào đang xét. Ngược lại, nếu không có một lá nào trong cây chứa trạng thái kết thúc thì ta nói M không đoán nhận xâu vào đó.

Chẳng hạn, khi xâu vào $\omega_1 = 01001$ đối với máy M thì ta có cây đoán nhận xâu ω_1 như sau:



Trong cây trên có một đường đi từ q_0 đến $q_4 \in F$ nên xâu $\omega_1 = 01001$ là xâu đoán nhận được bởi máy M.

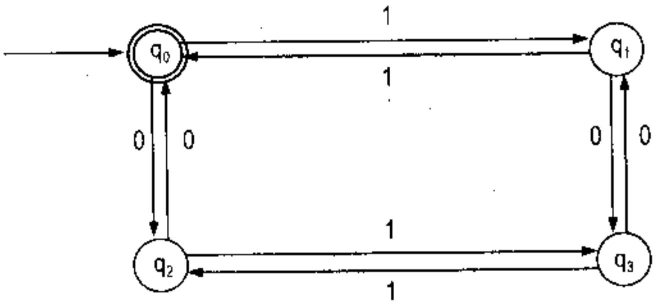
b) Phương pháp cho ôtômat bằng đồ thị chuyển

Cho ôtômat hữu hạn $M = <\Sigma, \delta, q_0, F>$. Hàm chuyển δ có thể cho bằng đồ thị chuyển có hướng theo nguyên tắc sau: Mỗi trạng thái $q \in Q$ là một đỉnh của đồ thị. Nếu $a \in \Sigma$ và từ trạng thái q_i chuyển sang trạng thái q_j do đẳng thức $\delta(q_i, a) = q_j$ (đối với ôtômat đơn định) thì sẽ có một cung có hướng từ đỉnh q_i sang đỉnh q_j và trên cung đó được gán nhãn a.

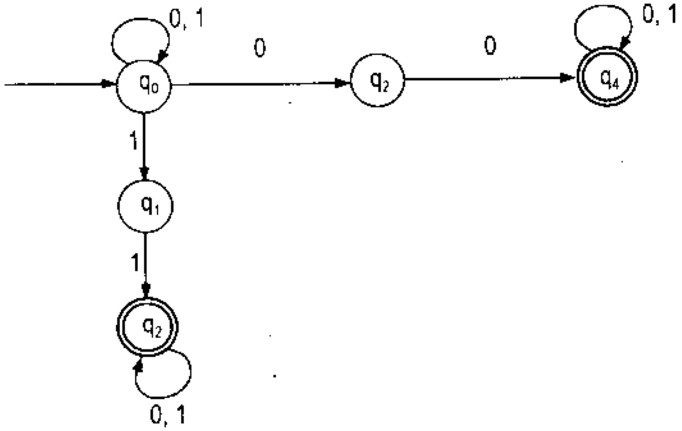
Nếu $\delta(q_i, a) = \{q_{j_1}, q_{j_2}, ..., q_{j_k}\}$ (đối với ôtômát không đơn định) thì từ q_i sang $q_{j_1}, q_{j_2}, ..., q_{j_k}$ có k cung gán cùng một nhãn a.

Đỉnh vào của đồ thị chuyển là đỉnh ứng với trạng thái ban đầu q_0. Các đỉnh có các trạng thái kết thúc được khoanh bằng vòng tròn kép. Các đỉnh còn lại khoanh bởi vòng tròn đơn.

Ví dụ 4: Cho ôtômat đơn định M = <Σ, Q, δ, q_0, F>, $\Sigma = \{0,1\}$, $Q = \{q_0, q_1, q_2, q_3\}$, $F = \{q_0\}$, hàm chuyển $\delta: Q \times \Sigma \rightarrow Q$ được cho dưới dạng đồ thị chuyển như sau:



Ví dụ 5: Cho ôtômat không đơn định M = <Σ, Q, δ, q_0, F>, $\Sigma = \{0,1\}$, $Q = \{q_0, q_1, q_2, q_3\}$, $F = \{q_2, q_4\}$, còn hàm chuyển $\delta: Q \times \Sigma \rightarrow 2^Q$ cho dưới dạng đồ thị như sau:



Đối với ôtômat M ta đưa vào khái niệm "*chuyển dịch* $\wedge$" được định nghĩa như sau:

Hàm ∧–Closure(T) cho tập trạng thái đạt được từ các trạng thái $s \in T$ qua phép chuyển dịch ∧ duy nhất. Lớp T có thể chỉ gồm một phần tử. Nó được tính như sau:

```
Đẩy tất cả trạng thái trong T vào Stack,
Đặt ∧–Closure(T) là T
While <stack không rỗng> do
    begin
        lấy t là phần tử trên cùng của stack ra;
        for <mỗi trạng thái mà có đường đi từ t đến u có nhãn ^> do
            if <u không thuộc ∧–Closure(T)> then
                begin
                    thêm u vào ∧–Closure(T);
                    đặt u vào Stack;
                end;
    end;
```

Thuật toán mô phỏng ôtômat không đơn định đoán nhận xâu vào:

- Đầu vào: Một xâu vào ω, kết thúc bởi ký hiệu hết file là eof. Một ôtômat không đơn định N với trạng thái ban đầu q_0 và tập trạng thái kết thúc $F \subseteq Q$.
- Đầu ra: Trả lời "True" nếu N đoán nhận xâu ω; "False" trong trường hợp ngược lại:
- Thuật toán:

```
Begin
    S := ∧–Closure(q0);
    C := ký tự tiếp;
    While C <> eof do
        begin
            S := ^–Closure(Move(S,C));
            C := ký tự tiếp;
        end;
    if S ∩ F ≠ ∅ then reture True else return False
End;
```

Chú ý rằng, hàm Move(T, a) là tập trạng thái mà N có thể đạt được khi thực hiện dịch chuyển với ký hiệu vào a từ một trạng thái thuộc T.

1.3. Sự tương đương giữa ôtômat đơn định và không đơn định

Theo định nghĩa của ôtômat đơn định M và ôtômat không đơn định N thì M cũng là N nên lớp các ngôn ngữ đoán nhận được bởi M cũng là lớp ngôn ngữ đoán nhận bởi N. Ký hiệu các lớp ngôn ngữ được đoán nhận bởi M và N tương ứng là L(M) và L(N).

Định lý 1: Lớp ngôn ngữ đoán nhận được bởi ôtômat đơn định trùng với lớp ngôn ngữ đoán nhận được bởi ôtômat không đơn định.

Chứng minh: Theo định nghĩa ôtômat đơn định và ôtômat không đơn định, thì lớp ngôn ngữ đoán nhận được bởi ôtômat đơn định nằm trong lớp ngôn ngữ đoán nhận được bởi ôtômat không đơn định. Ta chứng minh bao hàm thức ngược lại: Lớp ngôn ngữ đoán nhận được bởi ôtômat không đơn định nằm trong lớp ngôn ngữ do ôtômat đơn định đoán nhận. Thật vậy, giả sử $p \in \Sigma^*$ đoán nhận được bởi ôtômat không đơn định $N = <\Sigma, Q, \delta, q_0, F>$ hay $p \in L(N)$.

Ta xây dựng ôtômat đơn định $M = <\Sigma', Q', \delta', s_0, F>$ đoán nhận p như sau: $\Sigma' = \Sigma$; $Q' = 2^Q$; $s_0 = \{q_0\}$; $F' = \{U \subseteq Q : U \cap F \neq \varnothing\}$; còn δ': $Q' \times \Sigma \to Q'$ được xác định như sau:

$$\forall U \in Q, \forall x \in \Sigma \text{ thì } \delta'(U, x) = \bigcup_{q \in U} \delta(q, x)$$

Với M định nghĩa như trên thì ta có:

$$p \in L(M) \text{ hay } L(N) \subseteq L(M).$$

Ví dụ 6: Cho ôtômat không đơn định $N = <\Sigma, Q, \delta, q_0, F>$ với $\Sigma = \{a, b\}$, $Q = \{s_0, s_1\}$, $F = \{s_1\}$, còn hàm chuyển δ: $Q \times \Sigma \to 2^Q$ được cho bởi bảng chuyển sau:

Trạng thái	Ký hiệu vào	
	a	b
s_0	$\{s_0, s_1\}$	s_1
s_1	$\varnothing$	$\{s_0, s_1\}$

Khi đó ôtômat đơn định tương ứng với ôtômat không đơn định N được xây dựng như sau:

$M = <\Sigma', Q', \delta', q_2, F'>$ với $\Sigma' = \Sigma = \{a, b\}$; $Q' = \{q_1, q_2, q_3, q_4\}$, ở đây $q_1 = \varnothing$, $q_2 = \{s_0\}$, $q_3 = \{s_1\}$, $q_4 = \{s_0, s_1\}$, còn $F' = \{q_3, q_4\}$.

Hàm chuyển δ' cho dưới dạng bảng chuyển sau:

Trạng thái	Ký hiệu vào	
	a	b
q_1	$\varnothing$	$\varnothing$
q_2	q_1	q_3
q_3	$\varnothing$	q_4
q_4	q_4	q_4

Ta có L(N) = L(M) hay N tương ứng với M.

§2. NGÔN NGỮ CHÍNH QUY VÀ BIỂU THỨC CHÍNH QUY

2.1. Ngôn ngữ chính quy

Giả sử Σ là bảng chữ hữu hạn không rỗng, Σ^* là tập tất cả xác xâu (kể cả xâu rỗng) được xây dựng trên Σ.

Như đã định nghĩa trước đây thì $\Sigma^+ = \Sigma^* \setminus \{\wedge\}$.

Tập con $E \subseteq \Sigma^+$ được gọi là một ngôn ngữ trên bảng Σ.

Trên tập tất cả các ngôn ngữ ta có thể định nghĩa các phép toán hợp, giao, phần bù, nhân và lặp các ngôn ngữ.

Trong các phép toán trên, ta đặc biệt chú ý ba phép toán sau đây:

a) Phép hợp: Cho hai ngôn ngữ E_1, E_2 trên tập Σ, ta định nghĩa phép hợp $E_1 \cup E_2 = \{\omega : \omega \in E_1 \text{ hoặc } \omega \in E_2\}$.

b) Phép nhân: Cho hai ngôn ngữ E_1, E_2 trên Σ, phép nhân E_1 với E_2 là $E_1.E_2 = \{\alpha\beta : \alpha \in E_1, \beta \in E_2\}$.

c) Phép lặp: Với ngôn ngữ E trên Σ ta định nghĩa phép lặp của E là

$$E^+ = E \cup E^2 \cup E^3 \cup \ldots = \bigcup_{n=1}^{\infty} E^n \text{, ở đây } E^n = EE\ldots E \text{ (n lần)}.$$

Giả sử $\Sigma = \{a_1, a_2, \ldots, a_n\}$, ngôn ngữ $\varnothing$ và ngôn ngữ $\{a_i\}$ $(i = 1, 2, \ldots, n)$ được gọi là *ngôn ngữ sơ cấp* trên Σ.

Định nghĩa 2:

a) Các ngôn ngữ sơ cấp trên Σ gọi là *ngôn ngữ chính quy* trên Σ.

b) Nếu E và F là hai ngôn ngữ chính quy trên Σ thì $E \cup F$, E.F và E^+ cũng là ngôn ngữ chính quy trên Σ.

c) Không có ngôn ngữ chính quy nào khác trên Σ ngoài các ngôn ngữ chính quy được định nghĩa trong các bước a) và b) ở trên.

Thông thường để diễn đạt các ngôn ngữ chính quy người ta đưa vào biểu thức chính quy.

2.2. Biểu thức chính quy

Trên bảng Σ ta định nghĩa biểu thức chính quy theo các bước đệ quy sau:

1) $\varnothing$ là biểu thức chính quy, nó biểu diễn ngôn ngữ rỗng.

2) Nếu $a \in \Sigma$ thì a là biểu thức chính quy, nó biểu diễn ngôn ngữ $\{a\}$.

3) Nếu r, s là hai biểu thức chính quy trên Σ biểu diễn hai ngôn ngữ R và S tương ứng. Khi đó:

- $(r) \cup (s)$ là biểu thức chính quy biểu diễn ngôn ngữ $R \cup S$.
- (r).(s) là biểu thức chính quy biểu diễn ngông ngữ R.S.
- $(r)^+$ là biểu thức chính quy biểu diễn ngôn ngữ R^+.

Từ định nghĩa về ngôn ngữ chính quy và biểu thức chính quy trên Σ ta có:

Định lý 2: Mọi ngôn ngữ chính quy trên bảng Σ đều nhận được từ các ngôn ngữ hữu hạn bằng cách áp dụng một số hữu hạn lần các phép toán hợp, nhân và lặp.

Định lý 3: Một ngôn ngữ trên Σ là chính quy khi và chỉ khi nó biểu diễn được bởi một biểu thức chính quy.

Chú ý: a) Một ngôn ngữ chính quy là vô hạn khi và chỉ khi biểu thức chính quy biểu diễn nó có dấu + (phép lặp).

b) Lớp tất cả các ngôn ngữ chính quy trên Σ là lớp bé nhất chứa các tập hữu hạn và đóng đối với các phép toán hợp, nhân và lặp.

2.3. Thuật toán Thompson

Bài toán: Cho một biểu thức chính quy. Hãy xây dựng ôtômat không đơn định N đoán nhận ngôn ngữ chính quy từ biểu thức chính quy đã cho.

Thuật toán:

Đầu vào: Một biểu thức chính quy r trên bộ Σ.

Đầu ra: Một ôtômat N đoán nhận ngôn ngữ L(r).

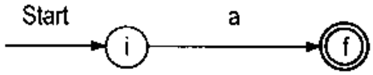
Sau đây là mô tả thuật toán của Thompson:

Trước hết tách biểu thức chính quy r thành các biểu thức chính quy thành phần $r_1, r_2, \ldots, r_k$. Sau đó áp dụng luật 1, luật 2 để xây dựng các ôtômat không đơn định $N_1, N_2, \ldots, N_k$ tương ứng đoán nhận các ngôn ngữ $L(r_1)$, $L(r_2), \ldots, L(r_k)$. Cuối cùng dùng luật 3 để xây dựng ôtômat không đơn định N đoán nhận ngôn ngữ L(r).

– *Luật 1:* Đối với ký hiệu ^, xây dựng ôtômat không đơn định N đoán nhận ngôn ngữ {^} như sau:

với i là trạng thái ban đầu còn f là trạng thái kết thúc.

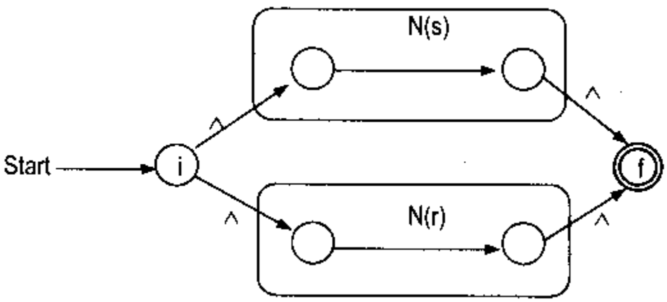
– *Luật 2:* Với a ∈ Σ, xây dựng ôtômat không đơn định đoán nhận {a} như sau:



với i là trạng thái ban đầu, f là trạng thái kết thúc.

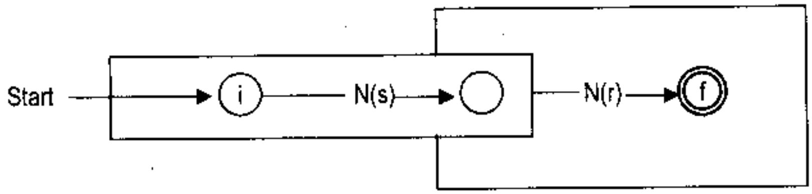
– *Luật 3:* Giả sử N(s), N(r) là các ôtômat thành phần ứng với các biểu thức chính quy s và r:

a) Đối với biểu thức chính quy (s) ∪ (r) ta xây dựng ôtômat không đơn định N đoán nhận ngôn ngữ S ∪ R như sau:



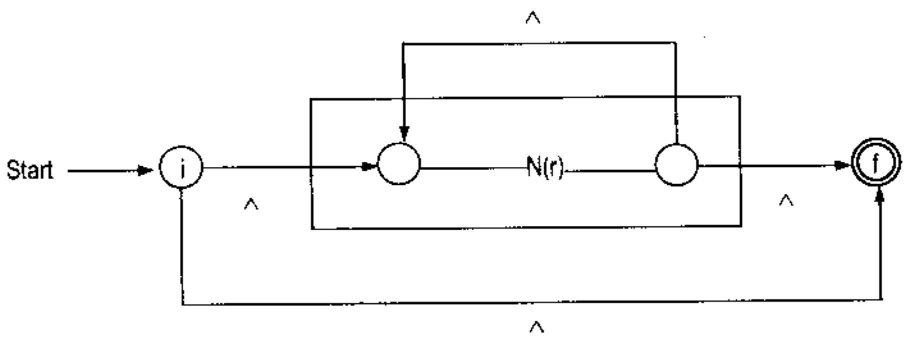
ở đây: i là trạng thái ban đầu; f là trạng thái kết thúc của N và N nhận được bằng cách tổ hợp hai ôtômat thành phần N(s) và N(r) theo sơ đồ bảng chuyển trên.

b) Đối với biểu thức chính quy (s).(r) thì ôtômat không đơn định N đoán nhận ngôn ngữ chính quy S.R được cho dưới dạng bảng chuyển sau đây:



với: i là trạng thái ban đầu; f là trạng thái kết thúc và N nhận được từ N(s) và N(r) bằng cách lấy trạng thái ban đầu của N(s) làm trạng thái ban đầu của N, trạng thái kết thúc của N(r) làm trạng thái kết thúc của N và đồng nhất trạng thái kết thúc của N(s) với trạng thái ban đầu của N(r).

c) Đối với biểu thức chính quy $(r)^+$ biểu diễn ngôn ngữ chính quy R^+ ta xây dựng ôtômat không đơn định N đoán nhận R^+ theo bảng chuyển dưới đây:



ở đây: i là trạng thái ban đầu; f là trạng thái kết thúc của N.

2.4. Tính chất của ngôn ngữ chính quy

Định lý 4: Lớp tất cả các ngôn ngữ chính quy trên Σ là đóng đối với các phép toán: hợp, nhân (giao), hiệu, lấy phần bù, nhân ghép và lặp.

Chứng minh: Tính đóng đối với các phép toán hợp, nhân ghép và lặp chứng minh theo định nghĩa biểu thức chính quy.

Ta chứng minh tính đóng đối với các phép toán còn lại sau đây:

• Phép hiệu:

R_1 và R_2 là hai ngôn ngữ chính quy trên Σ. Giả sử $R = R_1 \setminus R_2$, ở đây: $R_1 = L(M_1)$ (M_1 là ôtômat đoán nhận R_1 với $M_1 = <\Sigma, Q_1, \delta_1, q_1, F_1>$); $R_2 = L(M_2)$ (M_2 là ôtômat đoán nhận R_2 với $M_2 = <\Sigma, Q_2, \delta_2, q_2, F_2>$).

Ta xây dựng ôtômat M đoán nhận R như sau:

$$M = <\Sigma, Q, \delta, q_0, F> \text{ với } Q = Q_1 \times Q_2,$$
$$q_0 = (q_1, q_2) \in Q; F = F_1 \times (Q_2 \setminus F_2);$$

còn $\delta: Q \times \Sigma \to Q$ được xác định như sau:

$\delta((q, q'), x) = <\delta_1(q, x), \delta_2(q', x)>$ với $q \in Q_1, q' \in Q_2, x \in \Sigma$.

Kiểm tra lại ta có $L(M) = R$, hay R là ngôn ngữ chính quy trên Σ.

• Phép lấy phần bù:

Nếu R là ngôn ngữ chính quy trên Σ thì ta có $\overline{R}$ cũng ngôn ngữ chính quy trên Σ.

Thật vậy, giả sử $M = <\Sigma, Q, \delta, q_0, F>$ là ôtômat đoán nhận R. Khi đó ôtômat $M' = <\Sigma, Q, \delta, q_0, F'>$ với $F' = Q \setminus F$ là ôtômat đoán nhận ngôn ngữ $\overline{R}$, hay $\overline{R}$ là ngôn ngữ chính quy trên Σ.

• Phép toán giao giữa hai ngôn ngữ chính quy R_1 và R_2:

Vì $R_1 \cap R_2 = \overline{\overline{R_1} \cup \overline{R_2}}$ nên tính đóng đối với phép giao được suy ra từ tính đóng đối với phép toán hợp và phép lấy phần bù.

Định lý được chứng minh.

2.5. Quan hệ giữa ôtômat hữu hạn và ngôn ngữ chính quy

Định nghĩa 3: Văn phạm $G = <\Sigma, \Delta, I, R>$ được gọi là *văn phạm chính quy suy rộng* nếu mọi quy tắc trong R đều có dạng $A \to aB$, $A \to a$ hoặc $A \to \wedge$; ở đây $A, B \in \Delta$, $a \in \Sigma$, $\wedge$ là ký hiệu xâu rỗng.

Dễ dàng nhận thấy rằng:

Định lý 5: Đối với mỗi văn phạm chính quy suy rộng G bao giờ cũng xây dựng được văn phạm chính quy G' sao cho $L(G') = L(G) \setminus \{\wedge\}$.

Định nghĩa 4: Văn phạm chính quy suy rộng là *văn phạm chính quy mẫu* nếu nó không có quy tắc $A \to a$.

Định lý 6: Đối với văn phạm chính quy suy rộng có thể xây dựng được văn phạm chính quy mẫu tương đương với nó.

Chứng minh: Giả sử $G = <\Sigma, \Delta, I, R>$ là văn phạm chính quy suy rộng. Thêm ký hiệu K vào Δ và đặt $\Delta' = \Delta \cup \{K\}$.

Đối với mỗi quy tắc $A \to a \in R$ thay bởi cặp quy tắc $A \to aK$, $K \to \wedge$ ta được R'. Rõ ràng văn phạm $G' = <\Sigma, \Delta', I, R'>$ là văn phạm chính quy mẫu và $L(G) = L(G')$.

Định nghĩa 5: Cho văn phạm chính quy mẫu

$$G = <\Sigma, \Delta, I, R>.$$

Khi đó G có thể biểu diễn qua đồ hình được định nghĩa như sau:

Các đỉnh của đồ hình là các ký hiệu hỗ trợ trong Δ. Đối với cặp đỉnh A, B $\in \Delta$, số cung đi từ A sang B sẽ bằng số quy tắc có dạng $A \to aB$ của G. Mỗi một cung được đánh dấu bởi ký hiệu cơ bản $a \in \Sigma$ trong quy tắc tương ứng. Đỉnh ứng với ký hiệu ban đầu I gọi là *đỉnh ban đầu*, còn đỉnh A ứng với quy tắc $A \to$ ^ gọi là *đỉnh kết thúc*.

Đa đồ thị vừa được định nghĩa như trên gọi là đồ hình của văn phạm.

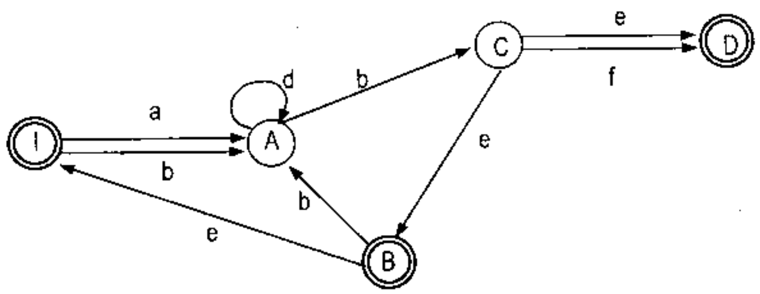
Ví dụ 1: Cho văn phạm chính quy mẫu:

$G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a, b, c, d, e, f\}$; $\Delta = \{I, A, B, C, D\}$;

$R = \{I \to aA, I \to bA, A \to dA, A \to bC, C \to cB, B \to bA,$

$C \to eD, C \to fD, B \to eI, I \to \wedge, B \to \wedge, D \to \wedge\}$

Đồ hình của G có dạng như hình dưới đây:



Đỉnh đầu là I, các đỉnh kết thúc là I, B, D. Nếu $A_0, A_1, \ldots, A_n$ là đường trong đa đồ thị của văn phạm chính quy mẫu đã cho sao cho với mỗi $i = \overline{1, n}$ có cung đi từ A_{i-1} sang A_i được đánh dấu bởi ký hiệu a_i thì ta nói rằng đường đã cho sinh ra xâu $a_1a_2 \ldots a_n$.

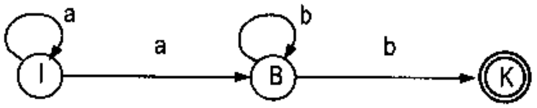
Đường bất kỳ có độ dài 0 theo định nghĩa nó sinh ra xâu ^. Rõ ràng xâu x sinh ra bởi một đường bắt đầu từ đỉnh A và kết thúc ở đỉnh B khi và chỉ khi $A \models xB$ (ký hiệu $\models$ nghĩa là hoặc dẫn được từ A hoặc trùng với A).

Ta gọi đường $A_0, A_1, \ldots, A_n$ là *chu trình* nếu $A_0 = A_n$ và là *đường đầy đủ* nếu $A_0 = I$ còn A_n là đỉnh kết thúc (tức $A_n \to$ ^ là một quy tắc trong R của G).

Từ các khái niệm trên ta suy ra: Tập dẫn xuất đầy đủ trong văn phạm G và tập các đường đầy đủ trong đa đồ thị của G có tương ứng 1 – 1, đồng thời xâu $x \in L(G)$ khi và chỉ khi x được sinh ra bởi đường đầy đủ trong đa đồ thị của G.

Chú ý: Đối với văn phạm chính quy suy rộng không phải là mẫu thì cũng tương tự như trên ta xây dựng được đồ hình của nó.

Ví dụ 2: Cho $G = <\Sigma, \Delta, I, R>$ là văn phạm chính quy suy rộng với $R = \{I \to aI, I \to aB, B \to bB, B \to b\}$. Đồ hình của G được xây dựng như sau: Đưa G về văn phạm mẫu G' bằng cách thay quy tắc $B \to b$ bởi hai quy tắc $B \to bK$ và $K \to$ ^. Đồ hình của văn phạm mẫu G' cũng là đồ hình của G có dạng dưới đây:



ở đây I là đỉnh ban đầu, K là đỉnh kết thúc.

Định nghĩa 6: Tương tự như đồ hình của văn phạm chính quy suy rộng, ta có thể định nghĩa đồ hình của ôtômat hữu hạn theo nguyên tắc sau đây:

– Các đỉnh của đồ hình là các trạng thái của ôtômat.

– Nếu trong ôtômat $M = <\Sigma, Q, \delta, q_0, F>$ mà $\delta(q, a) = q'$ với $q \in Q$, $a \in \Sigma$ thì trong đồ hình từ đỉnh q đến đỉnh q' có một cung và trên cung đó được gán mã a, q_0 là đỉnh ban đầu.

– Đỉnh q cũng được gọi là đỉnh ban đầu nếu ứng với nó ta có $\delta(q_0, \#) = q$. Các đỉnh đánh dấu bằng các trạng thái kết thúc gọi là các đỉnh kết thúc (# là ký hiệu biên bên trái).

Khái niệm một xâu sinh ra bởi một đường, khái niệm chu trình, khái niệm đường đầy đủ... cũng được định nghĩa như trong đồ thị của văn phạm ôtômat suy rộng.

Chú ý: Từ định nghĩa đồ hình của ôtômat hữu hạn ta suy ra: Một xâu sinh ra bởi đường đầy đủ trong đồ hình của ôtômat khi và chỉ khi xâu đó được đoán nhận bởi ôtômat đó. Ngoài ra, ôtômat đoán nhận xâu rỗng khi và chỉ khi giao giữa tập các đỉnh ban đầu và tập các đỉnh kết thúc trong đồ hình là khác rỗng.

Từ khái niệm đồ hình của văn phạm và đồ hình của ôtômat hữu hạn ta suy ra định lý sau đây:

Định lý 7:

1) Đối với văn phạm chính quy suy rộng bất kỳ có thể xây dựng được ôtômat hữu hạn tương đương với nó.

2) Đối với ôtômat hữu hạn bất kỳ có thể xây dựng được văn phạm chính quy suy rộng tương đương với nó.

Từ các kết quả trên ta có định lý quan trọng sau đây:

Định lý 8:

1) Lớp các ngôn ngữ chính quy trùng với lớp ngôn ngữ do văn phạm chính quy sinh ra.

2) Ngôn ngữ trên bảng Σ là đoán nhận được bởi ôtômat hữu hạn khi và chỉ khi ngôn ngữ đó là ngôn ngữ chính quy.

BÀI TẬP

1. Ôtômat M được cho dưới dạng bảng chuyển:

Trạng thái	Ký hiệu vào	
	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_4	q_2
q_4	q_1	q_4

Với trạng thái ban đầu và trạng thái kết thúc là q_0.

a) Viết đầy đủ theo định nghĩa của M.

b) M là ôtômat đơn định hay không đơn định? Vì sao?

c) Đưa ôtômat M về dạng đồ thị chuyển.

d) Kiểm tra lại sự đoán nhận của M đối với các xâu sau:

$\omega_1 = 001110111000$;

$\omega_2 = 110111101111$.

theo định nghĩa $L(M) = \{\omega : \omega \in \Sigma^* \text{ và } \sigma(q_0, \omega) \in F\}$.

2. Ôtômat M được cho dưới bảng chuyển:

Trạng thái	Ký hiệu vào	
	a	b
q_0	$\{q_0, q_3\}$	$\{q_0, q_1\}$
q_1	$\varnothing$	q_2
q_2	$\varnothing$	q_2
q_3	q_4	$\varnothing$
q_4	q_5	$\varnothing$
q_5	q_5	$\varnothing$

Với trạng thái ban đầu là q_0 còn trạng thái kết thúc là q_2 và q_5.

a) Viết đầy đủ M dưới dạng định nghĩa.

b) M là ôtômat đơn định hay không đơn định? Vì sao?

c) Viết M dưới dạng đồ thị chuyển.

d) Mô tả sự đoán nhận của M đối với xâu ω theo cách xây dựng cây đoán nhận với:

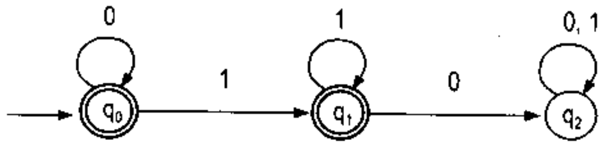
$\omega = a^n b a^n$ (với $n \geq 1$),

$\omega = a^n b^m$ ($n \geq 1$, $m \geq 2$),

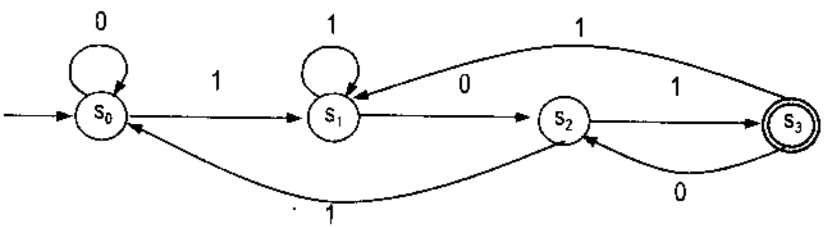
$\omega = b^n a^m$ ($n \geq 1$, $m \geq 3$).

3. Cho các ôtômat hữu hạn dưới dạng đồ thị chuyển như sau:

a) M_1:



b) M_2:

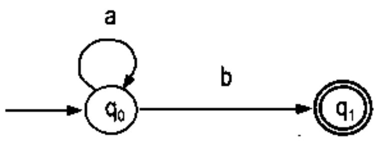


a) Viết đầy đủ theo định nghĩa M_1 và M_2 dưới dạng bảng chuyển.

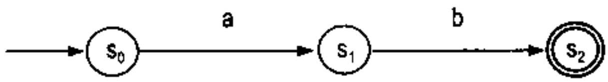
b) Mô tả ngôn ngữ đoán nhận của M_1 và M_2.

4. Cho các ôtômat hữu hạn dưới dạng đồ thị chuyển:

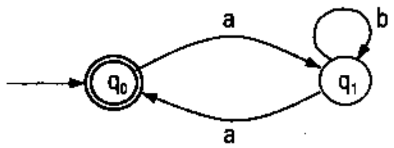
M_1:



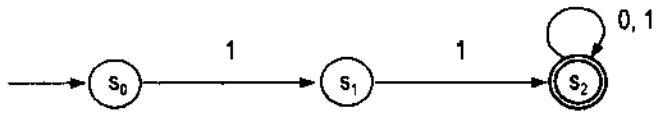
M_2:



M_3:



M_4:



Tìm $L(M_i)$ ($i = 1, 2, 3, 4$).

5. Tìm ôtômat hữu hạn đoán nhận các ngôn ngữ sau:

a) $L = \{\omega : \omega \in \{0, 1\}^*$ và ω có chẵn số 0 và lẻ số 1$\}$.

b) $L = \{a^n b^m : n \geq 1, m \geq 1\}$.

c) $L = \{\omega 00 : \omega \in \{0,1\}^*\}$.

d) $L = \{\omega 11 : \omega \in \{0,1\}^*\}$.

e) $L = \{\omega_1 000 \omega_2 : \omega_i \in \{0,1\}^*, i = 1, 2\}$.

f) $L = \{a^n b^m a^l : n, m, l \geq 0\}$.

g) $L = \{11\omega : \omega \in \{0,1\}^*\}$.

6. *Xây dựng ôtômat đơn định tương đương với ôtômat không đơn định cho trước.*

Cho ôtômat không đơn định $M = <\Sigma, Q, \sigma, q_0, F>$. Xây dựng ôtômat đơn định $M' = <\Sigma', Q', \sigma', s_0, F'>$ sao cho $L(M') = L(M)$ hay $M' \approx M$.

Xây dựng M' theo các bước sau:

Bước 1: $\Sigma' = \Sigma$;

Bước 2: $Q' = 2^Q$ (tập tất cả các tập con của Q);

Bước 3: $s_0 = \{q_0\}$;

Bước 4: $F' = \{U : U \subseteq Q$ mà $U \cap F \neq \varnothing\}$;

Bước 5: σ': $Q' \times \Sigma^* \to Q'$ xác định theo công thức:

$$\delta'(U, x) = \bigcup_{q \in U} \sigma(q, x) \text{ với } U \in Q' \text{ và } x \in \Sigma.$$

Với M' xác định như trên ta có L(M') = L(M).

Áp dụng bài toán trên hãy xây dựng các ôtômat đơn định tương đương với các ôtômat không đơn định được cho trong các trường hợp sau:

a) M_1:

Trạng thái	Ký hiệu vào	
	a	b
s_0	$\{s_0, s_1\}$	s_1
s_1	$\varnothing$	$\{s_0, s_1\}$

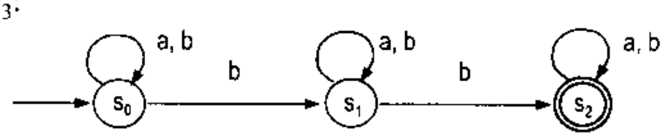
Với trạng thái ban đầu là s_0, trạng thái kết thúc là s_1.

a) M_2:

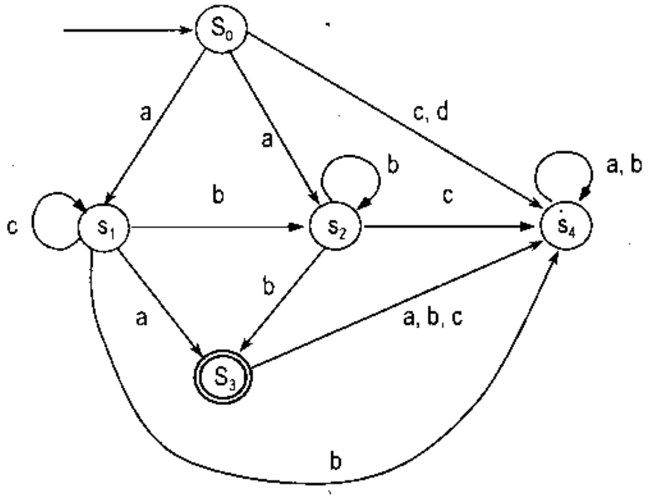
Trạng thái	Ký hiệu vào	
	a	b
s_0	$\{s_0, s_1\}$	s_1
s_1	$\varnothing$	s_2
s_2	$\varnothing$	$\varnothing$

Với trạng thái ban đầu là s_0 còn trạng thái kết thúc là s_2.

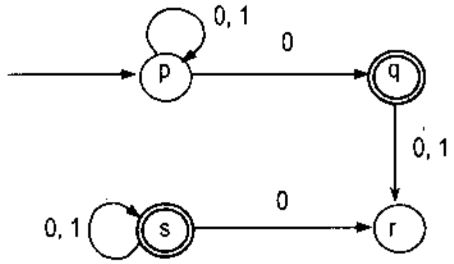
c) M_3:



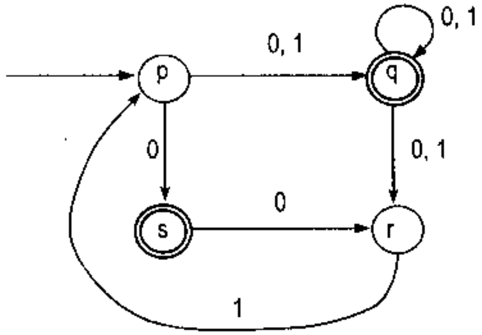
d) M_4:



e) M_5:



f) M_6:



7. Cho các văn phạm chính quy sau đây:

$G_1 = <\Sigma_1, \Delta_1, I_1, R_1>$ với $R_1 = \{I_1 \to aI_1, I_1 \to aB, B \to bB, B \to b\}$ và $G_2 = <\Sigma_2\ \Delta_2, I_2, R_2>$ với $R_2 = \{I_2 \to cI_2, I_2 \to d\}$.

a) Tìm $L(G_1)$ và $L(G_2)$.

b) Xây dựng các ôtômat M_1 và M_2 đoán nhận $L(G_1)$ và $L(G_2)$ tương ứng.

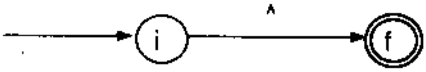
c) Cho các ôtômat M_1, M_2, M_3, M_4 như trong bài 4. Hãy xây dựng các văn phạm chính quy G_i sao cho $L(G_i) = L(M_i)$, i = 1, 2, 3, 4.

8. *Thuật toán Thompson xây dựng ôtômat không đơn định từ một biểu thức chính quy.*

Hướng dẫn:

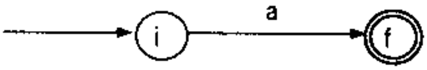
1) Với ký hiệu ^:

Ôtômat đoán nhận {^} có dạng



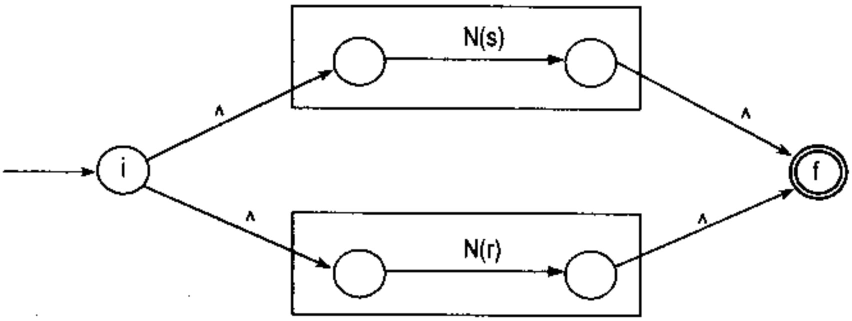
2) Với $a \in \Sigma$:

Ôtômat đoán nhận {a} có dạng

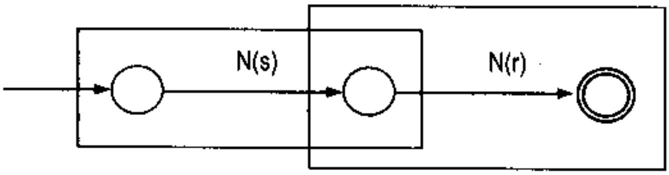


3) Giả sử N(s) và N(r) là hai ôtômat có thành phần ứng với các biểu thức chính quy s và r.

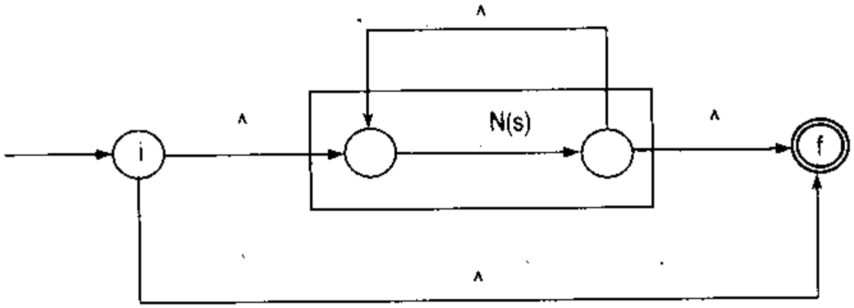
a) Ôtômat đoán nhận ngôn ngữ chính quy $S \cup R$, ở đây $L(N(s)) = S$, $L(N(r)) = R$, có dạng sau:



b) Ôtômat đoán nhận ngôn ngữ chính quy S.R có dạng:



c) Ôtômat đoán nhận ngôn ngữ chính quy S* = L(N(s))* có dạng:



Áp dụng thuật toán Thompson làm các bài tập từ bài 9 đến 20.

9. 00 là biểu thức chính quy nó chỉ định tập {00}.

a) Xây dựng văn phạm chính quy G sao cho L(G) = {00}.

b) Xây dựng ôtômat hữu hạn M đoán nhận {00}.

10. (0 + 1)* là biểu thức chính quy chỉ định mọi xâu 0 và 1.

a) Hãy xây dựng văn phạm phi ngữ cảnh G sao cho L(G) = (0 + 1)*.

b) Xây dựng ôtômat hữu hạn đoán nhận ngôn ngữ chính quy (0 + 1)*.

11. Cho ngôn ngữ chính quy L được xác định bởi biểu thức chính quy (0 + 1)* 00 (0 + 1)*.

a) Xây dựng văn phạm phi ngữ cảnh G sao cho L(G) = L.

b) Xây dựng ôtômat hữu hạn đoán nhận ngôn ngữ L.

Lưu ý: Biểu thức chính quy (0 + 1)* 00 (0 + 1)* chỉ định tập tất cả các xâu 0 và 1, trong đó có chứa 00.

12. Biểu thức (0 + 1)*011 chỉ định tập các xâu 0 và 1 kết thúc bởi 011.

Xây dựng ôtômat M đoán nhận ngôn ngữ chính quy trên.

13. Biểu thức 0* 1* 2* chỉ định tập các xâu gồm một số số 0, tiếp đến một số số 1 rồi đến một số số 2.

Xây dựng ôtômat M đoán nhận ngôn ngữ có biểu thức chính quy trên.

14. Xây dựng ôtômat đơn định đoán nhận ngôn ngữ L cho trong các trường hợp sau trên bộ chữ cái Σ = {0, 1}:

a) L = tập mọi xâu trên bộ {0, 1} và kết thúc bởi 00.

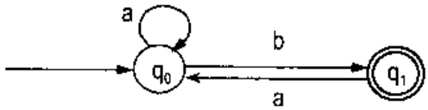
b) L = tập mọi xâu trên bộ {0, 1} có chứa ba số 0 liên tiếp.

c) L = tập mọi xâu mà ký hiệu thứ ba (kể từ mút phải) là 1.

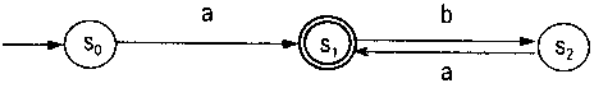
d) L = tập mọi xâu mà bất cứ xâu nào có độ dài 5 đều có chứa ít nhất hai số 0.

15. Cho các ôtômat:

M_1:



M_2:



a) Tìm $L(M_i)$ với $i = 1, 2$.

b) Dùng thuật toán Thompson xây dựng các ôtômat N_1, N_2, N_3 sao cho:

$$L(N_1) = L(M_1) \cup L(M_2);$$
$$L(N_2) = L(M_1).L(M_2);$$
$$L(N_3) = L(M_2)^*.$$

16. Cho ôtômat M_1, M_2 như trong bài 3. Hãy xây dựng ôtômat M sao cho:

a) $L(M) = L(M_1) \cup L(M_2)$;

b) $L(M) = L(M_1).L(M_2)$;

c) $L(M) = L(M_1)^*$.

17. Cho các ôtômat M_1, M_2 và M_3 như trong bài 4. Hãy xây dựng ôtômat M sao cho:

a) $L(M) = L(M_1) \cup L(M_2) \cup L(M_3)$;

b) $L(M) = L(M_3).L(M_1).L(M_3) \cup L(M_1)$;

c) $L(M) = L(M_3)^*$;

d) $L(M) = (L(M_1) \cup L(M_2)^*$.

18. Cho M_i ($i = \overline{1,6}$) trong bài 6. Xây dựng ôtômat M sao cho:

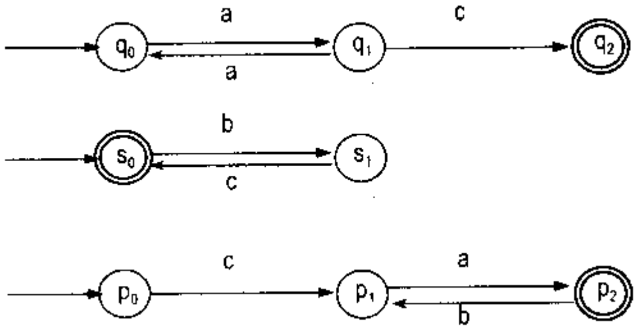
a) $L(M) = \bigcup_{i=1}^{6} L(M_i)$;

b) $L(M) = \bigcap_{i=1}^{6} L(M_i)$;

c) $L(M) = L(M_5)^*$.

d) $L(M) = L(M_1) \cup L(M_2) \cup L(M_3).L(M_4).L(M_5) \cup L(M_6)^*$.

19. Cho các ôtômat M_1, M_2, M_2 tương ứng có đồ thị chuyển là:



a) Tìm $L(M_i)$ (i = 1, 2, 3).

b) Xây dựng các văn phạm G_i sao cho $L(G_i) = L(M_i)$ (i = 1, 2, 3).

c) Xây dựng M sao cho:

1) $L(M) = L(M_1) \cup L(M_2) \cup L(M_3)$;

2) $L(M) = L(M_3).L(M_2).L(M_1)$;

3) $L(M) = L(M_3)^*$.

d) Từ G_1, G_2, G_3 xây dựng trong phần b. Hãy xây dựng văn phạm G sao cho:

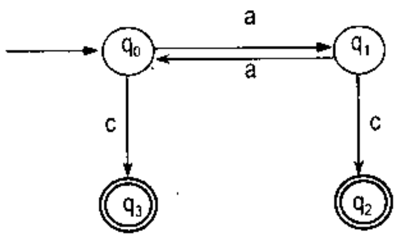
1) $L(G) = L(G_1) \cup L(G_2) \cup L(G_3)$;

2) $L(G) = L(G_3).L(G_2).L(G_1)$.

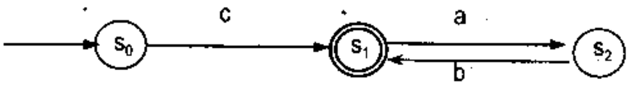
e) Có nhận xét gì về ôtômat M trong phần c.1, c.2 với văn phạm G trong phần d.1, d.2 tương ứng.

20. Cho ôtômat M_1 và M_2 dưới đây:

M_1:



M_2:



Xây dựng ôtômat M sao cho

$$L(M) = (L(M_1) \cup L(M_1).L(M_2) \cup L(M_2)^*)^*.$$

Chương 11
ÔTÔMAT ĐẨY XUỐNG ĐOÁN NHẬN NGÔN NGỮ PHI NGỮ CẢNH

Đối với các lớp văn phạm được phân loại theo Chomsky, lớp văn phạm phi ngữ cảnh có vai trò quan trọng nhất trong việc ứng dụng để xây dựng các ngôn ngữ lập trình và chương trình dịch.

Trong quá trình dịch từ chương trình nguồn ra chương trình đích, người ta sử dụng cấu trúc cú pháp của văn phạm phi ngữ cảnh để phân tích các xâu vào. Cấu trúc cú pháp của một xâu vào được xác định từ chuỗi các dẫn xuất suy từ xâu đó. Dựa vào chuỗi các dẫn xuất đó, bộ phân tích cú pháp của chương trình dịch sẽ cho biết xâu vào đang xét có thuộc vào xâu do văn phạm phi ngữ cảnh sinh ra hay không. Nói cách khác là, với xâu vào ω và một văn phạm phi ngữ cảnh G. Hỏi rằng $\omega \in L(G)$ hay không? Nếu có thì hãy tìm cách biểu diễn ω bằng văn phạm, tức là tìm các quy tắc sinh của văn phạm phi ngữ cảnh để sinh ra xâu ω.

§1. VĂN PHẠM PHI NGỮ CẢNH VÀ CÂY DẪN XUẤT CỦA NÓ

1.1. Định nghĩa văn phạm phi ngữ cảnh và quy tắc về ký hiệu

Bộ $G = <\Sigma, \Delta, I, R>$ được gọi là văn phạm phi ngữ cảnh nếu:

– Σ là từ điển cơ bản, các phần tử của nó gọi là phần tử cơ bản hay phần tử kết thúc;

– Δ là từ điển hỗ trợ, các phần tử của nó là phần tử hỗ trợ hay phần tử kết thúc;

– $I \in \Delta$ là ký hiệu phụ và được gọi là ký hiệu ban đầu;

– R là tập các quy tắc sinh (gọi tắt là quy tắc) có dạng:

$$R = \{A \to \theta : A \in \Delta, \theta \in (\Sigma \cup \Delta)^*\}.$$

Để dễ dàng cho việc diễn đạt các phần tiếp theo ta đưa vào một số quy ước sau:

– Mỗi ký hiệu trong từ điển hỗ trợ Δ là một chữ cái in hoa nào đó và có thể có cả chỉ số. Ví dụ như A, B, C, A_1, A_2, ..., B_1, B_2, ..., C_1, C_2, ...

– Mỗi ký hiệu trong từ điển cơ bản Σ là một chữ thường a, b, c, đôi khi cả chỉ số như a_1, a_2, ...

– Các chữ in hoa X, Y, Z có thể cả chỉ số để chỉ các ký hiệu thuộc bảng từ điển đầy đủ $V = \Sigma \cup \Delta$. Như vậy X, Y, Z có thể là ký hiệu cơ bản hoặc cũng có thể là ký hiệu phụ.

– Các chữ cái thường ω, u, v, w, x, y, z dùng để ký hiệu các xâu trong Σ^*.

– Các chữ α, β là các xâu trong từ điển đầy đủ $V = \Sigma \cup \Delta$, tức là các xâu có thể chứa cả các ký hiệu cơ bản và các ký hiệu phụ.

1.2. Cây dẫn xuất đầy đủ trong văn phạm phi ngữ cảnh

Định nghĩa cây dẫn xuất (suy diễn, phân tích) đầy đủ:

Cho văn phạm phi ngữ cảnh $G = \langle\Sigma, \Delta, I, R\rangle$.

Cây dẫn xuất đầy đủ trong văn phạm phi ngữ cảnh là một cây được thành lập từ một dẫn xuất đầy đủ nào đó trong G. Mỗi xâu $x \in \Sigma^*$ mà $I \models_G x$ sẽ tương ứng với một số cây dẫn xuất đầy đủ nào đó. Xâu x được gọi là kết quả của cây dẫn xuất đầy đủ, ta có thể định nghĩa cây dẫn xuất đầy đủ như sau:

Cây dẫn xuất đầy đủ trong văn phạm phi ngữ cảnh $G = \langle\Sigma, \Delta, I, R\rangle$ là một cây thoả mãn bốn yêu cầu sau:

1. Ở mỗi đỉnh được gán một nhãn. Nhãn gán ở đỉnh là các ký hiệu trong tập $\Sigma \cup \Delta$ {^}. Gốc của cây được gán nhãn là I.

2. Mỗi đỉnh trong được gán nhãn là một ký hiệu nào đó trong Δ.

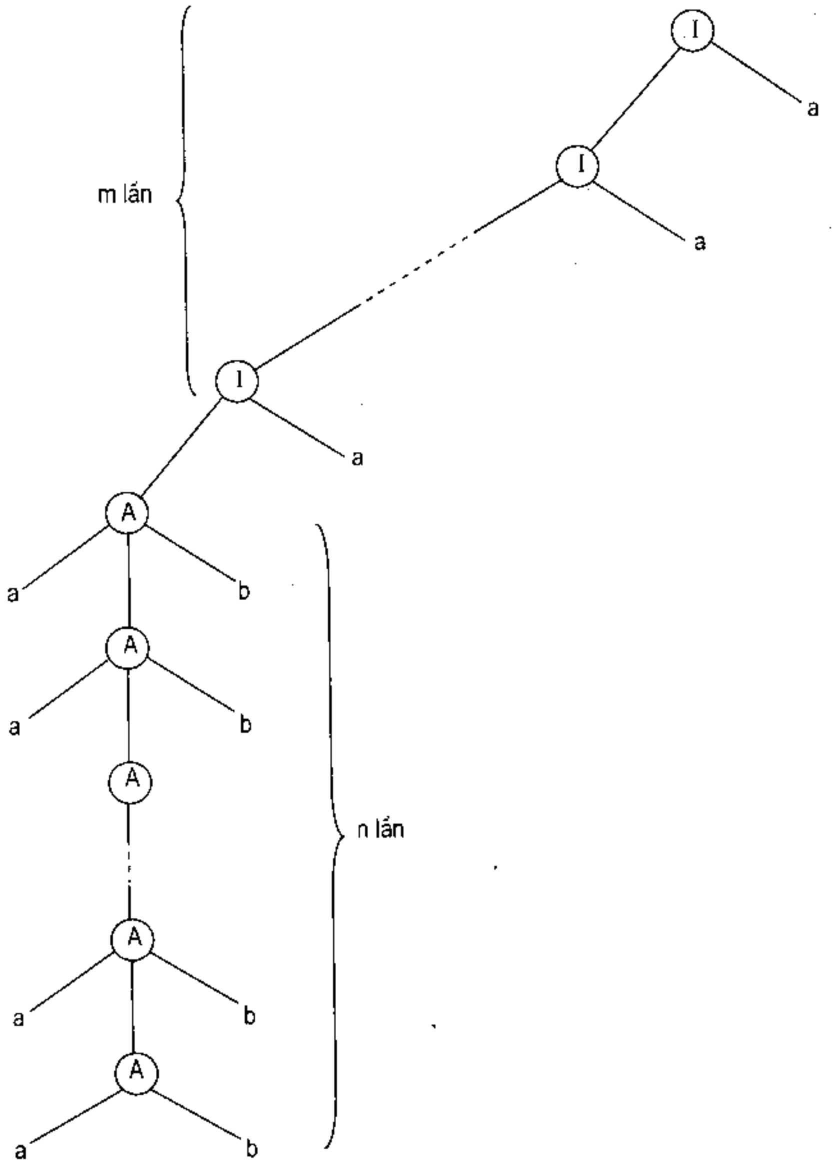
3. Mỗi đỉnh ngoài (lá của cây) được gán nhãn là một ký hiệu trong tập $\Sigma \cup \{\wedge\}$.

4. Nếu đỉnh m được gán nhãn là $A \in \Delta$, còn các đỉnh n_1, n_2, ..., n_k là các con của đỉnh m theo thứ tự từ trái sang phải và được gán nhãn X_1, X_2, ..., X_k tương ứng thì $A \to X_1, X_2, \ldots, X_k$ là một quy tắc trong R của văn phạm G.

Nếu đọc tất cả nhãn ở các lá (lá được gán nhãn trong tập $\Sigma \cup \{\wedge\}$) theo thứ tự từ trái qua phải ta sẽ nhận được một xâu nào đó. Xâu đó là một phần tử trong L(G) và được gọi là kết quả dẫn xuất đầy đủ của G.

Ví dụ 1: Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$, $\Sigma = \{a, b\}$, $\Delta = \{I, A\}$ và $R = \{I \rightarrow Ia, I \rightarrow Aa, I \rightarrow aAb, I \rightarrow ab\}$.

Cây dẫn xuất đầy đủ của xâu $a^n b^n a^m$ là cây:



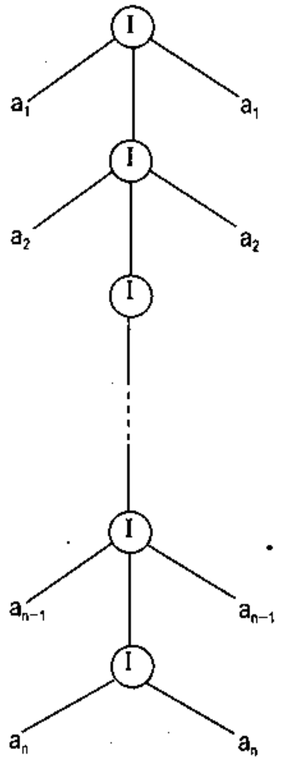
Ví dụ 2: Cho văn phạm phi ngữ cảnh G = <Σ, Δ, I, R>, trong đó:

$\Sigma = \{a_1, a_2, \ldots, a_n\}$,

$\Delta = \{I\}$ và $R = \{I \to aIa, I \to aa\}$, $a \in \Sigma$.

Cho $\omega = a_1a_2 \ldots a_n$, ta ký hiệu $\hat{\omega} = a_na_{n-1} \ldots a_2a_1$

Khi đó xâu $\omega\hat{\omega} = a_1a_2 \ldots a_{n-1}a_na_na_{n-1} \ldots a_2a_1 \in L(G)$ và cây dẫn xuất đầy đủ của nó có dạng



Ví dụ 3: Cho văn phạm phi ngữ cảnh G = <Σ, Δ, I, R>, trong đó:

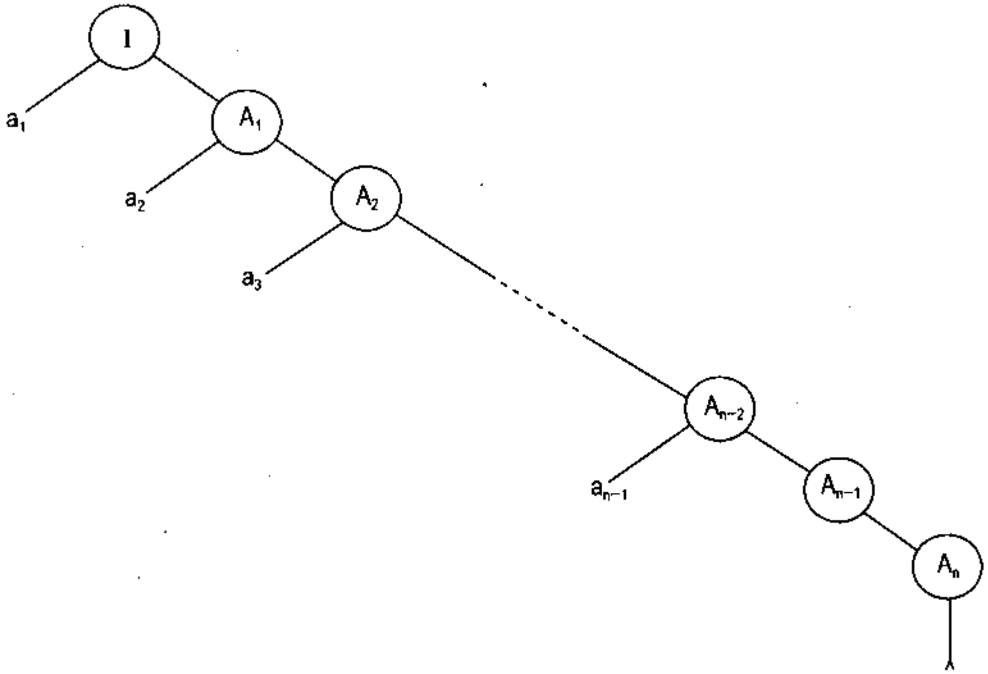
$\Sigma = \{a_1, a_2, \ldots, a_n\}$,

$\Delta = \{I, A_1, A_2, \ldots, A_n\}$,

$R = \{I \to a_1A_1, A_1 \to a_2A_2, \ldots, A_{n-1} \to a_nA_n, A_n \to \wedge\}$,

$L(G) = \{a_1a_2 \ldots a_n\}$.

Cây dẫn xuất của $\omega = a_1a_2 \ldots a_n$ là:



1.3. Sự nhập nhằng trong ngôn ngữ phi ngữ cảnh

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$, ta nói văn phạm G là *nhập nhằng* nếu tồn tại một xâu ω là kết quả của hai cây dẫn xuất đầy đủ khác nhau trong G.

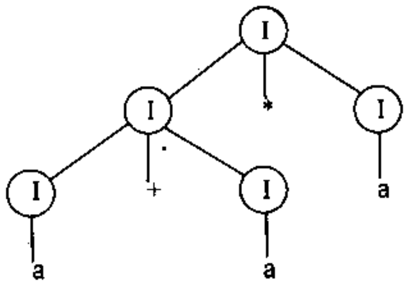
Ngôn ngữ do văn phạm G sinh ra là *ngôn ngữ nhập nhằng* nếu G là văn phạm nhập nhằng.

Ví dụ 4: Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$, trong đó:

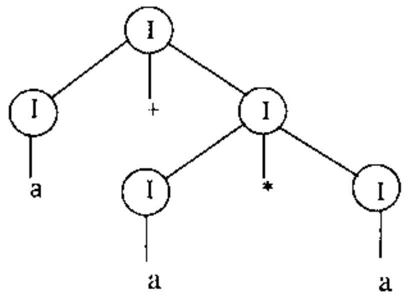
$$\Sigma = \{+, a, *\}, \Delta = \{I\}, R = \{I \to I + I, I \to I*I, I \to a\},$$

ở đây ký hiệu + là phép cộng, * là phép nhân.

Xâu a + a * a là kết quả của cây dẫn xuất đầy đủ (thực hiện phép * trước):



Xâu a + a * a cũng là kết quả của cây dẫn xuất đầy đủ (thực hiện phép + trước):



Rõ ràng xâu a + a * a là kết quả của hai cây dẫn xuất khác nhau trong cùng một văn phạm phi ngữ cảnh. Nếu $\omega \in L(G)$ là kết quả của nhiều cây dẫn xuất đầy đủ, thì xâu ω có nhiều cách phân tích cú pháp khác nhau. Mỗi một cây là một cách phân tích cú pháp. Văn phạm trong ví dụ này là văn phạm phi ngữ cảnh nhập nhằng.

§2. GIẢN LƯỢC CÁC VĂN PHẠM PHI NGỮ CẢNH

Trong một văn phạm thì ngữ cảnh có thể có nhiều yếu tố thừa, ví dụ có những ký hiệu không hề tham gia vào quá trình sinh các ngôn ngữ, hoặc có những quy tắc dạng $A \rightarrow B$ chỉ làm mất thời gian trong quá trình hình thành các xâu của ngôn ngữ. Vì lẽ đó cần loại bỏ những yếu tố dư thừa không có ích trong việc sinh ngôn ngữ, sao cho việc loại bỏ đó không làm ảnh hưởng tới quá trình sinh ngôn ngữ. Điều đó có nghĩa là chỉ cần giữ lại các ký hiệu và các quy tắc có ích trong văn phạm G mà chúng thực sự là cần thiết trong quá trình sinh ngôn ngữ mà thôi.

2.1. Ký hiệu có ích và ký hiệu thừa

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$. X được gọi là ký hiệu có ích nếu tồn tại dẫn xuất $I \vdash \alpha X \beta \vdash w$, ở đây α, $B \in (\Sigma \cup \Delta)^*$, $X \in \Sigma \cup \Delta$ và $w \in \Sigma^*$.

Nếu ký hiệu X không thoả mãn điều kiện trên thì X được gọi là *ký hiệu thừa*. Như vậy X là ký hiệu thừa nếu từ X không thể dẫn ra một xâu $w \in \Sigma^*$. Ký hiệu X có tính chất như thế còn được gọi là *ký hiệu vô sinh*. Nếu từ ký hiệu ban đầu I không dẫn được một xâu nào có chứa ký hiệu X thì ta nói ký hiệu X là *ký hiệu không đến được*. Như vậy, một ký hiệu là thừa nếu nó là ký hiệu vô sinh hoặc là ký hiệu không đến được.

Bổ đề 1 (loại ký hiệu vô sinh)

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ với $L(G) \neq \varnothing$. Ta có thể xây dựng được văn phạm phi ngữ cảnh G' tương đương với G, ở đây $G' = <\Sigma, \Delta', I, R'>$ sao cho mỗi $A \in \Delta'$ có một xâu $w \in \Sigma^*$ để $\vdash_G w$.

Chứng minh: Từ tập quy tắc R của G ta xây dựng Δ' như sau:

– Nếu trong R có quy tắc dạng $A \rightarrow w$ với $A \in \Delta$, $w \in \Sigma^*$ thì kết nạp A vào Δ'.

– Nếu $A \rightarrow X_1X_2 \ldots X_n$ là quy tắc trong R mà $X_i \in \Sigma$ hoặc X_i là biến đã được kết nạp vào Δ' thì đưa A vào Δ'.

Cứ tiếp tục xét các quy tắc trong R ta sẽ xây dựng các ký hiệu cho tập Δ'. Vì R là hữu hạn nên quá trình sẽ được dừng lại sau một số hữu hạn bước. Khi đó ta xây dựng được tập Δ'.

Ta xây dựng tiếp tập quy tắc R' gồm các quy tắc trong R mà các ký hiệu có mặt trong đó đều thuộc tập $\Sigma \cup \Delta'$.

Bổ đề 2 (loại ký hiệu không đến được)

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$, ta có thể xây dựng được văn phạm phi ngữ cảnh G' tương đương với G, ở đây $G' = <\Sigma', \Delta', I, R'>$ sao cho với mỗi $X \in \Sigma' \cup \Delta'$ đều tồn tại $\alpha, \beta \in (\Sigma' \cup \Delta')$ để cho $I \vdash \alpha X\beta$.

Chứng minh:

• Xây dựng tập Σ' và Δ' như sau:

Đưa ký hiệu I vào Δ'. Nếu một biến A đã được kết nạp vào Δ' và $A \rightarrow \alpha$ (ở đây $\alpha \in (\Sigma' \cup \Delta')$) thì ta kết nạp các ký hiệu biến trong α vào Δ', còn các ký hiệu kết thúc trong α thì kết nạp vào Σ'.

Thủ tục kết nạp trên sẽ ngừng khi không còn bổ sung thêm được bất kỳ ký hiệu nào vào các tập Σ' và Δ'.

• Tập quy tắc R' được xây dựng như sau:

R' bao gồm mọi quy tắc trong R mà các ký hiệu thuộc tập $\Sigma' \cup \Delta'$. Với cách xây dựng đó ta có $L(G) = L(G')$, trong đó G' gồm các ký hiệu đến được. Từ hai bổ đề trên ta có:

Định lý 1: Mọi ngôn ngữ phi ngữ cảnh không rỗng đều có thể được sinh ra từ một văn phạm phi ngữ cảnh không có ký hiệu thừa.

Ví dụ 1: Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$, ở đây $\Sigma = \{a\}$, $\Delta = \{I, A, B\}$, $R = \{I \to AB, I \to a, A \to a\}$. Văn phạm G sinh ra ngôn ngữ $L(G) = \{a\}$ (tập gồm một xâu a).

Văn phạm phi ngữ cảnh G' nhận được từ G bằng cách áp dụng các kết quả trên là văn phạm phi ngữ cảnh không có ký hiệu thừa và có dạng:

$G' = <\Sigma, \Delta', I, R'>$ với $\Sigma = \{a\}$, $\Delta' = \{I\}$, $R' = \{I \to a\}$ và rõ ràng $L(G') = \{a\} = L(G)$.

2.2. Các $\wedge$– quy tắc ($\wedge$ chỉ xâu rỗng)

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$. Nếu trong R có quy tắc $A \to \wedge$, $A \in \Delta$ thì ta nói trong G có $\wedge$– quy tắc.

Bây giờ ta tìm cách loại các $\wedge$– quy tắc khỏi tập quy tắc R của văn phạm G.

Nếu L(G) không chứa từ $\wedge$ thì có thể loại hết $\wedge$– quy tắc trong R ta được R' và $G' = <\Sigma, \Delta, I, R'>$ vẫn tương đương với $G = <\Sigma, \Delta, I, R>$. Nếu $\wedge \in L(G)$ thì ta không thể loại hết toàn bộ các $\wedge$ – quy tắc.

Định lý 2: Giả sử $G = <\Sigma, \Delta, I, R>$ là văn phạm phi ngữ cảnh và $L = L(G)$ là ngôn ngữ của văn phạm G. Khi đó $L \setminus \{\wedge\}$ được sinh ra bởi các văn phạm phi ngữ cảnh $G' = <\Sigma, \Delta', I, R'>$ không chứa các ký hiệu thừa và không chứa các $\wedge$– quy tắc.

Chứng minh:

• Từ tập Δ ta xây dựng Δ' như sau:

Ký hiệu $A \in \Delta$ được gọi là ký hiệu triệt tiêu nếu tồn tại dẫn xuất $A \vdash \wedge$. Ta có thể xác định các ký hiệu triệt tiêu bằng phương pháp sau:

– Nếu $A \to \wedge$ là một quy tắc trong R thì A là ký hiệu triệt tiêu.

– Nếu $B \to \alpha$ là một quy tắc trong R mà α gồm các ký hiệu triệt tiêu thì B cũng là ký hiệu triệt tiêu. Lặp lại quá trình trên cho tới khi trong Δ không còn tìm được các ký hiệu triệt tiêu mới. Bằng cách loại tất cả các ký hiệu triệt tiêu tìm được trong Δ ta sẽ nhận được tập Δ' mà trong nó không còn ký hiệu triệt tiêu.

• Tập R' được xác định như sau:

Nếu $A \to X_1X_2 \ldots X_n$ là một quy tắc trong R thì ta đưa vào R' quy tắc dạng $A \to \alpha_1\alpha_2 \ldots \alpha_n$, trong đó:

– Nếu X_i là ký hiệu không triệt tiêu thì đặt $\alpha_i = X_i$.

– Nếu X_i là ký hiệu triệt tiêu thì đặt $\alpha_i = \wedge$.

– Nếu mọi X_i là ký hiệu triệt tiêu thì thay $\alpha_i = \wedge$, với mọi i = 1, ..., n.

Khi đó $L(G') = L(G) \setminus \{\wedge\}$.

Ví dụ 2: Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$, trong đó:

$\Sigma = \{a, b\}$, $\Delta = \{I, A, B\}$, $R = \{I \to AB, A \to aA, A \to \wedge, B \to bB, B \to \wedge\}$.

Hãy loại bỏ $\wedge$– quy tắc ra khỏi văn phạm phi ngữ cảnh trên.

Giải:

Rõ ràng $\wedge \in L(G)$.

Theo định lý trên thì các quy tắc $A \to \wedge$ và $B \to \wedge$ có thể được loại trước tiên. Sau đó từ $I \to AB$ mà $B \to \wedge$ và $A \to \wedge$ nên ta có $I \to A$, $I \to B$. Cũng từ $A \to aA$, $B \to bB$ và $A \to \wedge$, $B \to \wedge$ ta có các quy tắc $A \to a$ và $B \to b$. Tóm lại:

$R' = \{I \to AB, I \to A, I \to B, A \to aA, A \to a, B \to bB, B \to b\}$.

Rõ ràng văn phạm $G' = <\Sigma, \Delta, I, R'\}$ không có $\wedge$– quy tắc và $L(G') = L(G) \setminus \{\wedge\}$. Muốn $G \approx G'$, do $\wedge \in L(G)$ nên cần bổ sung thêm quy tắc $I \to \wedge$ vào R'.

2.3. Các quy tắc đơn

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$. Nếu quy tắc trong R có dạng $A \to B$, ở đây $A, B \in \Delta$ thì ta nói $A \to B$ là quy tắc đơn. Quy tắc đơn có tác dụng làm kéo dài quá trình sinh ra ngôn ngữ, vì vậy ta sẽ tìm cách loại quy tắc đơn mà không làm ảnh hưởng tới quá trình sinh ra ngôn ngữ của văn phạm đã cho.

Chú ý: Quy tắc $A \to a$ với $A \in \Delta$ và $a \in \Sigma$ không phải là quy tắc đơn.

Định lý 3: Mọi ngôn ngữ phi ngữ cảnh không chứa xâu rỗng đều có thể sinh ra từ một văn phạm phi ngữ cảnh không có ký hiệu thừa, không có các $\wedge$ – quy tắc và quy tắc đơn.

Chứng minh: Giả sử cho văn phạm $G = <\Sigma, \Delta, I, R>$ là văn phạm phi ngữ cảnh và $L = L(G)$ là ngôn ngữ phi ngữ cảnh do G sinh ra. Ta xây dựng văn phạm phi ngữ cảnh $G^* = <\Sigma, \Delta^*, I, R^*>$ sinh ra ngôn ngữ $L \setminus \{\wedge\} = L(G^*)$, trong đó G^* không có ký hiệu thừa, không có $\wedge$– quy tắc và quy tắc đơn.

Theo định lý 2, từ văn phạm G ta xây dựng được văn phạm $G' = <\Sigma, \Delta', I, R'>$ là văn phạm phi ngữ cảnh sao cho $L(G') = L \setminus \{\wedge\}$, trong đó Δ' không có ký hiệu thừa, còn R' không có $\wedge$– quy tắc. Đặt $\Delta^* = \Delta'$. Xây dựng quy tắc R^* như sau:

Đưa tất cả các quy tắc không đơn của R' vào R^*.

Nếu trong R' có $A \to B$ với $A, B \in \Delta'$, khi đó sẽ tồn tại dẫn xuất $I \vdash \alpha A\beta \vdash \alpha B\beta \vdash \alpha w\beta$, ở đây $\alpha, \beta \in (\Sigma \cup \Delta)^*$, $w \in \Sigma^*$ do Δ' gồm các ký hiệu không thừa.

Vậy thay cho $A \to B$ ta đưa vào R^* quy tắc $I \to \alpha A\beta$ và $A \to w$ đều là các quy tắc không đơn nhưng chức năng sinh ngôn ngữ tương đương với quy tắc $A \to B$.

Rõ ràng $L(G^*) = L(G') = L \setminus \{\wedge\}$.

Ví dụ 3: Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$, trong đó: $\Sigma = \{a\}$, $\Delta = \{I, A, B\}$, $R = \{I \to I + A, I \to A, A \to A^*B, A \to B, B \to a\}$.

Hãy loại các quy tắc đơn trong văn phạm trên.

Ta xây dựng $G^* = <\Sigma, \Delta, I, R^*>$ với $R^* = \{I \to I + A, A \to A^*B, B \to a, I \to A^*B, A \to a, I \to a\}$ không còn quy tắc đơn và $G \approx G^*$.

§3. VĂN PHẠM CHUẨN CỦA CHOMSKY

Định nghĩa 1: Văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ được gọi là *văn phạm chuẩn của Chomsky* nếu mọi quy tắc trong tập R đều có dạng $A \to BC$ hoặc $A \to a$, ở đây $A, B, C \in \Delta$, $a \in \Sigma$.

Định lý 4: Đối với văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ bao giờ cũng tồn tại một văn phạm phi ngữ cảnh dạng chuẩn $G_1 = <\Sigma, \Delta_1, I, R_1>$ tương đương với nó, tức là $L(G) = L(G_1)$.

Chứng minh: Giả sử $G = <\Sigma, \Delta, I, R>$ là văn phạm phi ngữ cảnh nào đó. Theo định lý 3 thì có thể giả thiết trong R không chứa các $\wedge$– quy tắc và quy tắc đơn $A \to B$ với $A, B \in \Delta$.

Ta có thể giả thiết trong R chỉ gồm các quy tắc có dạng $A \to a$, hoặc $A \to B_1B_2 \dots B_n$ (ở đây $a \in \Sigma$, còn $A, B_1, B_2, \dots, B_n \in \Delta$).

Nếu G không thoả mãn các điều kiện trên thì trong G còn quy tắc dạng $A \to X$, với X là một xâu gồm cả các ký kiệu chính và các ký hiệu hỗ trợ. Trong trường hợp này, ta xây dựng G' theo G sao cho $L(G) = L(G')$ và G' thoả mãn điều kiện $R' = \{A \to a, A \to B_1B_2 \dots B_n\}$ như sau:

1) Với mỗi $a \in \Sigma$ ta đặt tương đương với ký hiệu mới $\bar{a}$. Sau đó đặt $\Delta' = \Delta \cup \{\bar{a} : a \in \Sigma\}$.

2) Trong tất cả các quy tắc của R mà trong đó vế phải của quy tắc có chứa ký hiệu chính $a \in \Sigma$ thì trong quy tắc đó ta thay a bởi $\bar{a}$ ta được $\bar{R}$.

Đặt $R' = \bar{R} \cup \{\bar{a} \to a : a \in \Sigma\}$.

Khi đó ta nhận được văn phạm $G' = <\Sigma, \Delta', I, R'>$ tương đương với G nhưng R' chỉ gồm quy tắc dạng $A \to a$, $A \in \Delta'$, $a \in \Sigma$; $A \to B_1B_2 \ldots B_n$, $A, B_i \in \Delta'$, $i = 1, 2, \ldots, n$.

Vậy ta có thể giả thiết văn phạm $G = <\Sigma, \Delta, I, R>$ chỉ chứa các quy tắc dạng $A \to a$ hoặc $A \to B_1B_2 \ldots B_n$, trong đó

$$a \in \Sigma;\ A, B_1, B_2, \ldots, B_n \in \Delta.$$

Có hai trường hợp sau xảy ra đối với G:

Trường hợp 1. Nếu tập R gồm các quy tắc dạng:

$$A \to a,\ A \to BC,\ a \in \Sigma \text{ còn } A, B, C \in \Delta.$$

Khi đó định lý được chứng minh.

Trường hợp 2. Nếu tập R gồm các quy tắc $A \to a$ hoặc $A \to B_1B_2 \ldots B_n$ với $n > 2$, thì $A \to B_1B_2 \ldots B_n$ có thể thay bởi các quy tắc $A \to B_1K_1$, $K_1 \to B_2K_2$, $K_2 \to B_3K_3$, $\ldots$, $K_{n-3} \to B_{n-2}K_{n-2}$, $K_{n-2} \to B_{n-1}B_n$, ở đây $K_1, K_2, \ldots, K_{n-2}$ là ký hiệu phụ mới.

Đặt $\Delta_1 = \Delta' \cup \{K_1, K_2, \ldots, K_{n-2}\}$, $R_1 = \{A \to a, A \to B_1K_1, K_1 \to B_2K_2, K_2 \to B_3K_3, \ldots, K_{n-3} \to B_{n-2}K_{n-2}, K_{n-2} \to B_{n-1}B_n\}$

Khi đó $G_1 = <\Sigma, \Delta_1, I, R_1> \approx G = <\Sigma, \Delta, I, R>$ thoả mãn điều kiện định lý. Định lý đã được chứng minh.

Ví dụ 1: Cho văn phạm $G = <\Sigma, \Delta, I, R>$, trong đó: $\Sigma = \{a, b\}$, $\Delta = \{I, A\}$, $R = \{I \to aAI, I \to a, A \to IbA, A \to II, A \to ba\}$.

Ta xây dựng văn phạm $G_1 = <\Sigma, \Delta_1, I, R_1>$ tương đương với văn phạm G sao cho G_1 là dạng chuẩn Chomsky. Trước hết ta xây dựng văn phạm $G' = <\Sigma, \Delta', I, R'>$ với $\Delta' = \{I, A, \bar{a}, \bar{b}\}$, $R' = \{I \to a, I \to \bar{a}AI, A \to I\bar{b}A, A \to II, A \to \bar{b}\,\bar{a}, \bar{a} \to a, \bar{b} \to b\}$, trong đó R' gồm các quy tắc có dạng $A \to a$ và $A \to B_1B_2 \ldots B_n$, với $a \in \Sigma$ còn $A, B_1, B_2, \ldots, B_n \in \Delta'$.

Tiếp theo ta xây dựng văn phạm chuẩn Chomsky $G_1 = <\Sigma, \Delta_1, I, R_1>$ có $\Sigma = \{a, b\}$, $R_1 = \{I \to a, I \to \bar{a}B_1, B_1 \to AI, A \to IB_2, B_2 \to \bar{b}A, A \to II, A \to \bar{b}\,\bar{a}, \bar{a} \to a, \bar{b} \to b\}$.

Rõ ràng $L(G') = L(G_1)$. Vì $L(G') = L(G)$, suy ra $L(G) = L(G_1)$ với G_1 là văn phạm ở dạng chuẩn Chomsky.

Ví dụ 2: Văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a, b\}$; $\Delta = \{I, A, B\}$; $R = I \to bA, I \to aB, A \to bAA, A \to aI, A \to a, B \to aBB, B \to bI, B \to b\}$.

Tìm dạng chuẩn.

Giải:

Xây dựng văn phạm $G' = <\Sigma, \Delta', I, R'>$ với:

$\Sigma = \{a, b\}, \Delta' = \Delta \cup \{\bar{a}, \bar{b}\} = \{I, A, B, \bar{a}, \bar{b}\}$,

$R' = \{I \to \bar{b}A, I \to \bar{a}B, A \to \bar{b}AA, A \to \bar{a}I, A \to a, B \to \bar{a}BB, B \to \bar{b}I, B \to b, \bar{a} \to a, \bar{b} \to b\}$.

Rõ ràng $L(G') = L(G)$.

Từ G' ta xây dựng dạng chuẩn Chomsky $G_1 = <\Sigma, \Delta_1, I, R_1>$ như sau:

$\Sigma = \{a, b\}, \Delta_1 = \{I, A, B, \bar{a}, \bar{b}, F_1, F_2\}$,

$R_1 = \{I \to \bar{b}A, I \to \bar{a}B, A \to \bar{a}F_1, F_1 \to AA, A \to \bar{a}I, A \to a, B \to \bar{a}F_2, F_2 \to BB, B \to \bar{b}I, B \to b, \bar{a} \to a, \bar{b} \to b\}$.

Rõ ràng $L(G') = L(G_1)$ hay $L(G) = L(G_1)$.

§4. ÔTÔMAT ĐẨY XUỐNG (PUSHDOWN AUTOMATA)

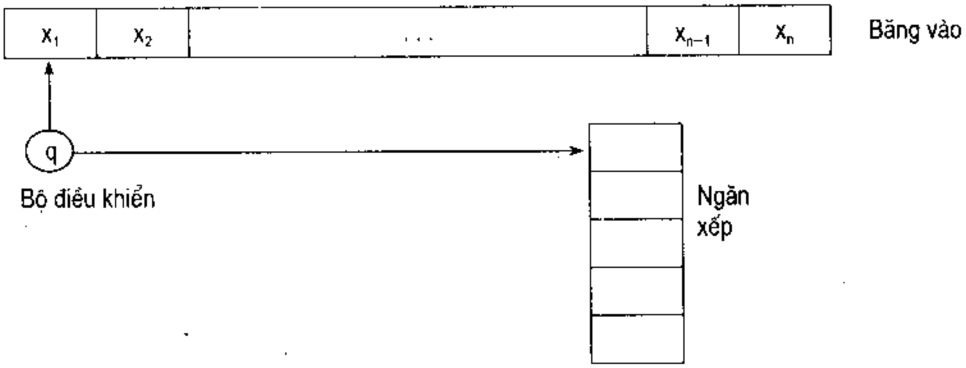
Như ta đã biết, lớp các ngôn ngữ chính quy do văn phạm chính quy suy rộng sinh ra cũng trùng với lớp các ngôn ngữ được đoán nhận bởi ôtômat hữu hạn (đơn định hoặc không đơn định).

Tương tự chúng ta sẽ thấy trong chương này là lớp ngôn ngữ phi ngữ cảnh do các văn phạm phi ngữ cảnh sinh ra sẽ trùng với lớp các ngôn ngữ được đoán nhận bởi ôtômat đẩy xuống không đơn định.

Đáng lưu ý chỉ có lớp ôtômat đẩy xuống không đơn định mới có thể đoán nhận hết lớp ngôn ngữ phi ngữ cảnh. Còn ôtômat đẩy xuống đơn định chỉ có khả năng đoán nhận được lớp con thực sự của lớp ngôn ngữ phi ngữ cảnh mà thôi. Tuy vậy, lớp ngôn ngữ được đoán nhận bởi lớp các ôtômat đẩy xuống đơn định là khá rộng, nó bao gồm phần lớn các ngôn ngữ lập trình hiện nay. Ở đây chúng ta chỉ đề cập tới ôtômat đẩy xuống không đơn định, gọi tắt là NDPA hay gọn hơn là PA.

4.1. Ôtômat đẩy xuống

PA bao gồm một băng vào, một ngăn xếp và một bộ điều khiển như hình dưới đây:



Ôtômat đẩy xuống cũng như ôtômat hữu hạn có bộ điều khiển là tập hữu hạn các trạng thái. Đầu đọc của ôtômat cho phép đọc lần lượt các ký hiệu trên băng vào từ trái sang phải. Ngoài ra PA còn có thêm băng làm việc (ngăn xếp) hay stack, nhờ có nó mà bộ nhớ của PA được tăng lên so với khả năng nhớ của ôtômat hữu hạn. Ngăn xếp được tổ chức theo nguyên tắc ký hiệu vào sau thì ra trước (giống như ổ nạp đạn). Khi đưa ký hiệu vào ngăn xếp thì ký hiệu đó được trở thành ký hiệu đầu của ngăn xếp. Khi ngăn xếp đọc thì ký hiệu đó là ký hiệu trên cùng và khi ký hiệu đó được đọc xong thì nó sẽ bị loại khỏi ngăn xếp. Một ngăn xếp như vậy còn được gọi là một danh sách đẩy xuống.

Bước chuyển của PA:

Căn cứ vào trạng thái hiện tại của bộ điều khiển, ký hiệu vào mà đầu đọc đang quan sát là ký hiệu đầu của ngăn xếp, ôtômat đẩy xuống sẽ chuyển sang một trạng thái mới nào đó và đồng thời đầu đọc có thể được chuyển sang ô bên phải. Nếu đầu đọc chuyển sang ô bên phải thì ta gọi quá trình trên là một bước chuyển. Ngược lại, nếu ký hiệu vào không ảnh hưởng tới bước chuyển thì ta gọi đó là bước chuyển "nhắm mắt" và trong bước chuyển đó đầu đọc vẫn đứng yên tại chỗ. Thực chất của bước chuyển "nhắm mắt" là sự tạm ngừng quan sát băng vào để chấn chỉnh lại ngăn xếp.

Có hai cách đoán nhận xâu vào của PA:

Cách 1: Xâu vào được đọc xong và PA chuyển được về một trạng thái kết thúc nào đó.

Cách 2: Xâu vào được đọc xong và ngăn xếp trở thành rỗng.

Sau này ta sẽ chỉ ra hai cách đoán nhận trên là tương đương.

Ví dụ 1: Cho văn phạm phi ngữ cảnh $G = \langle\Sigma, \Delta, I, R\rangle$, với $\Sigma = \{0, 1, c\}$, $\Delta = \{I\}$, $R = \{I \to 0I0, I \to 1I1, I \to c\}$.

Dễ nhận thấy rằng, các xâu có dạng $wc\hat{w}$ (với $w \in \{0, 1\}^*$, $\hat{w}$ là xâu viết các ký hiệu của xâu w theo một thứ tự ngược lại) được sinh ra bởi văn phạm G. Chẳng hạn, đối với xâu w = 010011 thì xâu $wc\hat{w}$ = 010011c110010 sẽ có dẫn xuất đầy đủ sau đây:

$$D = (I, 0I0, 01I10, 010I010, 0100I0010, 01001I10010,$$
$$010011I110010, 010011c110010) \text{ hay } wc\hat{w} \in L(G).$$

Mặt khác xâu $wc\hat{w}$ có thể được đoán nhận bởi PA như sau:

Trước hết đặt xâu $x = wc\hat{w}$ lên băng vào. Đầu đọc dịch chuyển từ trái sang phải. Ban đầu ngăn xếp rỗng. PA có hai trạng thái p, q, trong đó p là trạng thái ban đầu. Khi ở trạng thái ban đầu p, đầu đọc đọc ký hiệu trên băng vào là 0 hoặc 1 và nó đưa ký hiệu đó vào ngăn xếp. Trạng thái của PA lúc đó vẫn là p. Đầu đọc dịch chuyển sang bên phải 1 ô và đọc ký hiệu ở ô mới này. Nếu ký hiệu này là 0 hoặc 1 thì PA đưa ký hiệu đó vào ngăn xếp, trạng thái của ôtômat vẫn là p và đầu đọc lại được chuyển sang phía phải 1 ô.

Quá trình đó vẫn tiếp tục cho tới khi đầu đọc gặp ký hiệu c. Khi đó ôtômat sẽ chuyển trạng thái p sang trạng thái q và đầu đọc chuyển sang phải 1 ô. Tại thời điểm này ngăn xếp xuất hiện xâu w. Ký hiệu bên phải nhất của w nằm trên cùng của ngăn xếp, còn trạng thái của ôtômat là q. Đầu đọc đang chỉ ô bên phải ký hiệu c. Nếu ký hiệu của ô này là ký hiệu ở ô trên cùng của ngăn xếp thì PA loại ký hiệu trên cùng ra khỏi ngăn xếp, ký hiệu dưới nó được đẩy lên vị trí đầu của ngăn xếp, PA chuyển đầu đọc sang phải 1 ô, còn trạng thái vẫn là q. Quá trình đó cứ tiếp tục và có hai khả năng xảy ra:

1) PA đọc hết xâu x và ngăn xếp trở thành rỗng thì PA dừng lại và đoán nhận được xâu $x = wc\hat{w}$.

2) Các ký hiệu ở ngăn xếp chưa bị loại hết thì đầu đọc gặp ký hiệu trên xâu vào khác ký hiệu trên cùng của ngăn xếp. Trong trường hợp này PA không đoán nhận xâu x.

Tóm lại, ta đã xây dựng được ôtômat đẩy xuống đoán nhận ngôn ngữ phi ngữ cảnh gồm và chỉ gồm các xâu có dạng $wc\hat{w}$.

Nhận xét: Nhờ có ngăn xếp mà PA có khả năng nhớ được nửa đầu của xâu $x = wc\hat{w}$ với w có độ dài tuỳ ý và sau đó nó so sánh dần với nửa cuối $\hat{w}$ của x. Ôtômat hữu hạn không có khả năng này.

Bây giờ ta định nghĩa một cách hình thức ôtômat đẩy xuống như sau:

Định nghĩa 2: Một ôtômat đẩy xuống PA là một bộ sắp thứ tự $M = <\Sigma, Q, \Gamma, \delta, q_0, Z_0, F>$, ở đây:

Σ là tập hữu hạn, khác rỗng các ký hiệu vào;

Q là tập hữu hạn, khác rỗng các trạng thái sao cho $\Sigma \cap Q = \varnothing$,

Γ là tập hữu hạn, khác rỗng các ký hiệu mà ta gọi là các ký hiệu của ngăn xếp;

$Z_0 \in \Gamma$ là ký hiệu đặc biệt, gọi là ký hiệu đáy của ngăn xếp (còn gọi là ký hiệu đầu của danh sách đẩy xuống);

$q_0 \in Q$ là trạng thái ban đầu;

$F \subseteq Q$ là tập trạng thái kết thúc;

$\delta: \Sigma \cup \{\wedge\} \times Q \times \Gamma^* \to 2^{Q\times\Gamma^*}$ là hàm chuyển của M.

Bước chuyển của PA:

a) Đẳng thức dạng:

$$\delta(a, q, Z) = \{(q_1, \gamma_1), (q_2, \gamma_2), \ldots, (q_m, \gamma_m)\},$$

trong đó $Z \in \Gamma$, $a \in \Sigma$, $q_i \in Q$, $\gamma_i \in \Gamma^*$ $(i = \overline{1,m})$ được gọi là một bước chuyển của PA. Nó đang ở trạng thái q, đọc ký hiệu a ở băng vào và Z là ký hiệu ở đỉnh ngăn xếp. Khi đó nó chuyển sang trạng thái q_i, thay ký hiệu Z ở đỉnh ngăn xếp bằng xâu γ_i $(i = \overline{1,m})$, đồng thời chuyển đầu đọc sang bên phải 1 ô. Quy ước rằng khi đưa γ_i vào ngăn xếp, ký hiệu bên trái nhất của γ_i sẽ nằm ở phần dưới, còn ký hiệu bên phải nhất của γ_i sẽ nằm ở ô đầu ngăn xếp.

b) Đẳng thức có dạng:

$$\delta(\wedge, q, Z) = \{(q_1, \gamma_1), (q_2, \gamma_2), \ldots, (q_m, \gamma_m)\}$$

diễn tả một bước chuyển "nhắm mắt" của PA: Ôtômat ở trạng thái q, ký hiệu Z ở đỉnh ngăn xếp. Khi đó PA chuyển trạng thái q về q_i và thay $Z \in \Gamma$ ở đỉnh ngăn xếp bởi xâu γ_i $(i = \overline{1,m})$, còn đầu đọc thì không dịch chuyển.

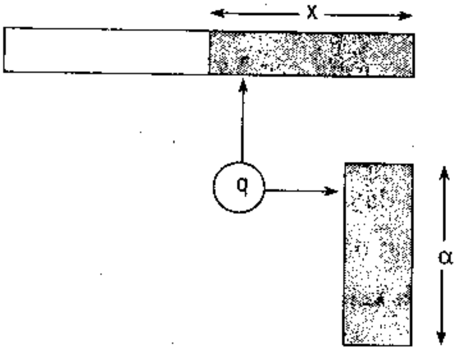
4.2. Ngôn ngữ đoán nhận của PA

Ở một thời điểm, tình huống tức thời của PA xác định bởi ba yếu tố:

1) Xâu $\alpha \in \Gamma^*$ trong ngăn xếp (với quy ước là ký hiệu bên trái nhất (bên phải nhất) của α nằm ở đáy (ở đỉnh) ngăn xếp).

2) Trạng thái $q \in Q$.

3) Phần xâu vào $x \in \Sigma^*$ chưa được đọc đến trên băng vào (với quy ước: ký hiệu bên trái nhất của x là ký hiệu sẽ được đọc tiếp). Hình trạng hiện thời của PA có thể mô tả như ở hình dưới đây:



Hình trạng hiện thời của PA ta ký hiệu là K thì $K = (x, q, \alpha)$. Quá trình đoán nhận xâu vào $\omega \in \Sigma^*$ của PA là quá trình biến đổi hình trạng bắt đầu từ hình trạng ban đầu (ω, q_0, z_0) thông qua các dẫn xuất trực tiếp ($\vdash:$) và dẫn xuất gián tiếp ($\vdash$) được định nghĩa như sau:

Định nghĩa 3: Giả sử có hai hình trạng $K = (\omega, q, v)$ và $K' = (\omega', q', v')$.

1) Ta nói hình trạng K' dẫn xuất trực tiếp từ hình trạng K (ký hiệu $K \vdash: K'$) nếu có $\omega = x\omega_0$, $v = v_0k$, $v' = v_0m$ thoả mãn:

$$x \in \Sigma \cup \{\wedge\},\ \omega' = \omega_0 \text{ đồng thời } (q', m) \in \delta(x, q, k).$$

2) Ta nói hình trạng K' dẫn xuất gián tiếp từ hình trạng K (ký hiệu $K \vdash K'$) nếu có một dãy các dẫn xuất trực tiếp $K = K_0, K_1, ..., K_t$, ở đây $K_i \vdash: K_{i+1}$ $(i = 0, 1, ..., t - 1)$.

Sự đoán nhận ngôn ngữ của PA có thể thực hiện được theo hai cách. Giả sử M là một PA. Ký hiệu T(M) là tập các ngôn ngữ đoán nhận được bởi M theo tập trạng thái kết thúc. Null(M) là các ngôn ngữ đoán nhận được bởi M theo kiểu ngăn xếp (stack) rỗng. Ta có thể định nghĩa T(M) và Null(M) như sau:

$T(M) = \{\omega : \omega \in \Sigma^*$ sao cho $(\omega, q_0, Z_0) \vdash (\wedge, q, v)$ với $q \in F, v \in \Gamma^*\}$

$Null(M) = \{\omega : \omega \in \Sigma^*$ sao cho $(\omega, q_0, Z_0) \vdash (\wedge, q, \wedge)$ với $q \in Q\}$

Lưu ý rằng, nếu đoán nhận ngôn ngữ theo ngăn xếp rỗng thì trong $M = <\Sigma, Q, \Gamma, \delta, q_0, Z_0>$ cho $F = \varnothing$.

Bây giờ ta phát biểu và chứng minh định lý trọng tâm sau:

Định lý 5: Đối với mọi ngôn ngữ phi ngữ cảnh L có tồn tại một ôtômat đẩy xuống không đơn định M đoán nhận L.

Chứng minh: Giả sử $G = <\Sigma, \Delta, I, R>$ là một văn phạm phi ngữ cảnh sinh ra ngôn ngữ phi ngữ cảnh L hay L(G) = L.

Ta xây dựng ôtômat đẩy xuống không đơn định $M = <\Sigma, Q, \Gamma, \delta, q_0, Z_0, F>$ đoán nhận L với:

Σ là bảng các ký hiệu vào (chính là Σ của G);

$Q = \{q_0, q_1, q_2\}$ là tập các trạng thái;

$\Gamma = \Sigma \cup \Delta \cup \{\%\}$ là tập các ký hiệu ngăn xếp;

$Z_0 = I$ là ký hiệu đầu của ngăn xếp;

$q_0 \in Q$ là trạng thái ban đầu;

$F = \{q_2\}$ là tập trạng thái kết thúc;

Hàm chuyển δ: $\Sigma \cup \{\wedge\} \times Q \times \Gamma^* \to 2^{Q\times\Gamma^*}$ định nghĩa qua các biểu thức sau:

1) $\delta(\wedge, q_1, \xi) = \{(q_1, \hat{w}) : \xi \to w \in R, \xi \in \Delta, w \in \Gamma^*\}$ ($\hat{w}$ là ký hiệu xâu được thành lập từ xâu w với thứ tự ngược lại (ảnh gương)).

2) $\delta(a, q_1, a) = \{(q_1, \wedge)\}$ đối với mọi $a \in \Sigma$.

3) $\delta(\wedge, q_1, \%) = \{(q_2, \%)\}$.

4) $\delta(\wedge, q_0, I) = \{(q_1, \%I)\}$.

Ký hiệu % là ký hiệu đáy ngăn xếp, không thuộc tập $\Sigma \cup \Delta$.

Giả sử $w \in L = L(G)$, khi đó với w có tồn tại dẫn xuất đầy đủ trong G là $D = (I, u_1\xi_1v_1, u_1u_2\xi_2v_2, ..., u_1u_2...u_{n-1}\xi_{n-1}v_{n-1}, u_1, u_2, ..., u_n = w)$. Ở đây $\xi_i \in \Delta, u_i \in \Sigma, v_i \in \Sigma \cup \Delta$ $(1 \le i \le n-1)$ và $u_n \in \Sigma^*$.

Dựa vào các quy tắc của G sử dụng trong dẫn xuất đầy đủ D của xâu $w = u_1u_2 ... u_n$, ôtômat đẩy xuống M đoán nhận w theo nguyên tắc sau:

Trên băng vào của M ta đặt xâu $w = u_1u_2 ... u_n$.

Dựa trên M áp dụng hàm chuyển trong các biểu thức 4, 1, 2 ta có

$$(w, q_0, Z_0) = (w, q_0, I) \vdash^{(4)}: (w, q_1, \%I) = (u_1u_2 \dots u_n, q_1, \%I)$$
$$\vdash^{(1)}: (u_1u_2 \dots u_n, q_1, \% \hat{v}_1\xi_1\hat{u}_1) \vdash^{(2)}: (u_2u_3 \dots u_n, q_1, \% \hat{v}_1 \xi_1),$$

chú ý $\hat{u}_1$, $\hat{v}_1$ là xâu thành lập từ xâu u_1, v_1 theo thứ tự ngược lại.

Giả sử các quy tắc tiếp theo (sau quy tắc $I \to u_1\xi_1v_1$) trong D là $\xi_i \to Z_i \in R$ ($i = 1, 2, \dots, n-2$) với $\xi_1 \in \Delta$, $Z_i \in \Sigma \cup \Delta$ sao cho $Z_iv_i = u_{i+1}\xi_{i+1}v_{i+1}$. Khi đó M sau thời điểm thực hiện quy tắc $I \to u_1\xi_1v_1$ nó có hình trạng $(u_2u_3 \dots u_n, q_1, \% \hat{v}_1 \xi_1)$. Hình trạng của M khi áp dụng quy tắc $\xi_i \to Z_i$ ($i = 1, 2, \dots, n-2$) biến đổi như sau:

$$(u_2u_3 \dots u_n, q_1, \% \hat{v}_1 \xi_1) \vdash^{(1)}: (u_3 \dots u_n, \% \hat{v}_1\hat{Z}_1) =$$
$$= (u_2u_3 \dots u_n, q_1, \% \hat{v}_2\xi_2\hat{u}_2) \vdash^{(2)}: (u_3 \dots u_n, q_1, \% \hat{v}_2\xi_2) \vdash: \dots$$
$$\vdash: (u_n, q_1, \% \hat{v}_{n-1}\xi_{n-1}) \quad (*)$$

Trong D ta sử dụng quy tắc $\xi_{n-1} \to Z_{n-1} \in R$ với $Z_{n-1}v_{n-1} = u_n$ ta có hình trạng của M là:

$$(u_n, q_1, \% \hat{v}_{n-1}\xi_{n-1}) \vdash^{(1)}: (u_n, q_1, \% \hat{v}_{n-1}\hat{Z}_{n-1}) =$$
$$= (u_n, q_1, \% \hat{u}_n) \vdash^{(2)}: (\wedge, q_1, \%) \vdash^{(3)}: (\wedge, q_2, \%) \quad (**)$$

Từ (*) và (**) ta có dẫn xuất gián tiếp của w trong M là

$$(w, q_0, Z_0) \vdash (\wedge, q_2, \%)$$

mà $q_2 \in F$ nên đoán nhận được xâu w. Định lý đã được chứng minh.

Bây giờ ta chỉ ra sự đoán nhận của ôtômat đẩy xuống M theo tập trạng thái kết thúc và theo ngăn xếp rỗng là tương đương nhau.

Định lý 6: Đối với mỗi ôtômat đẩy xuống M_1 đều tồn tại một ôtômat đẩy xuống M_2 sao cho $Null(M_2) = T(M_1)$.

Chứng minh: Giả sử $M_1 = <\Sigma_1, Q_1, \Gamma_1, \delta_1, q_0^1, Z_0^1, F_1>$ là ôtômat đẩy xuống nào đó.

Ta xây dựng ôtômat đẩy xuống $M_2 = <\Sigma_2, Q_2, \Gamma_2, \delta_2, q_0^2, Z_0^2, F_2>$ sao cho $Null(M_2) = T(M_1)$.

Muốn vậy ta đưa thêm vào ký hiệu trạng thái mới $q_1^1, q_2^1 \notin Q_1$ và ký hiệu ngăn kéo mới $\% \notin \Gamma_1$ và đặt:

$\Sigma_2 = \Sigma_1$;

$Q_2 = Q_1 \cup \{q_1^1, q_2^1\}$;

$\Gamma_2 = \Gamma_1 \cup \{\%\}$;

$q_0^2 = q_1^1$;

$Z_0^2 = \%$;

$F_2 = \varnothing$;

δ_2: $\Sigma \cup \{\wedge\} \times Q_2 \times \Gamma_2^* \to 2^{Q_2 \times \Gamma_2^*}$ được định nghĩa như sau:

$$\delta_2(\wedge,\ q_1^1, \%) = \left\{(q_0^1, \%, Z_0^1)\right\},$$

$$\delta_2(x, q, \alpha) = \delta_1(x, q, \alpha) \text{ với } x \in \Sigma_1, q \in Q, \alpha \in \Gamma_1.$$

$$\delta_2(\wedge, q, \alpha) = \begin{cases} \delta(\wedge, q, \alpha) & \text{nếu } q \in Q_1 \setminus F_1, \alpha \in \Gamma_1 \\ \delta(\wedge, q, \alpha) \cup \left\{(q_2^1, \wedge)\right\} & \text{nếu } q \in F_1, \alpha \in \Gamma_1 \end{cases}$$

$$\delta_2(\wedge, q, \%) = \left\{(q_2^1, \wedge)\right\} \text{ với } q \in F_1,$$

$$\delta_2(\wedge,\ q_2^1, \alpha) = \left\{(q_2^1, \wedge)\right\} \text{ với } \alpha \in \Gamma_1 \cup \{\%\}.$$

Bây giờ ta chỉ ra $\mathrm{Null}(M_2) = T(M_1)$.

Giả sử $w \in \mathrm{Null}(M_2)$, khi đó theo cách làm việc của M_2 thì ta có một dãy các hình trạng sau:

$$(w,\ q_0^2, Z_0^2) = (w,\ q_1^1, \%) \overset{M_2}{\vdash} K_1 \overset{M_2}{\vdash} K_2 \ldots \overset{M_2}{\vdash} K_t = (\wedge, q, \wedge)$$

với $t \geq 2$, $K_i = (w_i, q_i, v_i)$, $w_i \in \Sigma_1^*$, $q_i \in Q_2$, $v_i \in \Gamma_2^*$ $(i = 1, 2, ..., t)$.

Theo cách xây dựng của M_2 thì

$$K_1 = (w,\ q_0^1, \%Z_0^1),\ K_t = (\wedge,\ q_2^1, \wedge) \text{ và } w \in \mathrm{Null}(M_2).$$

Từ đó $\exists\ i < t$ sao cho:

$$K_i = (\wedge, q_i, \%\, v'_i) \overset{M_2}{\vdash} (\wedge,\ q_2^1, \%v'_{i+1}),$$

trong đó: $q_i \in F_i$, v'_i, $v'_{i+1} \in \Gamma_1^*$ và

$$K_j = (w_j, q_j, \%v'_j) \overset{M_2}{\vdash} (w_{j+1}, q_{j+1}, \%v'_{j+1}),$$

ở đây: $w_j \in \Sigma_1^*$, $q_j \in Q_1$, $v'_j \in \Gamma_1^*$.

Cũng như $K_j = (\wedge, q_2^1, \%v'_j)$, ở đây $v'_j \in \Gamma_1^*$ $(i < j < t)$.

Từ đó $(w, q_0^1, Z_0^1) \vdash (\wedge, q_i, \%v_i)$ và do $q_i \in F_1$ suy ra $w \in T(M_1)$.

Tóm lại ta có bao hàm thức $Null(M_2) \subseteq T(M_1)$.

Bao hàm thức ngược lại $T(M_1) \subseteq Null(M_2)$ được suy ra trực tiếp từ cách xây dựng M_2 từ M_1.

Để kết thúc phần này ta chứng minh định lý sau đây:

Định lý 7: Đối với mỗi ôtômat đẩy xuống M bao giờ cũng xây dựng được một văn phạm phi ngữ cảnh G sao cho Null(M) = L(G).

Chứng minh: Giả sử $M = <\Sigma, Q, \Gamma, \delta, q_0, Z_0, F>$ là một ôtômat đẩy xuống. Theo định lý 6 ta cần chỉ ra có văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ sao cho Null(M) = L(G). Không giảm tính tổng quát giả sử $\wedge \notin Null(M)$. Ta xây dựng văn phạm G ở trên như sau:

Σ là Σ của M;

$\Delta = \{I\} \cup \{(q_1, \alpha, q_2) : q_1, q_2 \in Q, \alpha \in \Gamma \cup \Sigma\}$;

I là ký hiệu hỗ trợ ban đầu của G;

$R = R_1 \cup R_2 \cup R_3$, ở đây:

$R_1 = \{I \to (q_0, \alpha, q) : q \in Q, \alpha \in \Gamma\}$,

$R_2 = \{(t_1, \alpha, t_2) \to x(t_3, \alpha_n, q_{n-1})(q_{n-1}, \alpha_{n-1}, q_{n-1})...$
$(q_2, \alpha_2, q_1)(q_1, \alpha_1, t_2) : n \in \{1, 2, ...\}, q_i \in Q\ (1 \le i < n)$,
$t_1, t_2, t_3 \in Q, x \in \Sigma \cup \{\wedge\}$ và $(t_3, \alpha_1\alpha_2 ... \alpha_n) \in \delta(x, t_1, \alpha)\}$.

$R_3 = \{(t_1, \alpha, t_2) \to x : (t_2, \wedge) \in \delta(x, t_1, \alpha)$ với $\alpha \in \Gamma, t_1, t_2 \in Q$,
$x \in \Sigma \cup \{\wedge\}\}$.

Người ta đã chỉ ra được với G định nghĩa như trên là văn phạm phi ngữ cảnh mà L(G) = Null(M), tức là lớp ngôn ngữ do G sinh ra đoán nhận bởi ôtômat đẩy xuống M.

Định lý 8: Lớp ngôn ngữ chính quy là một tập con thực sự ở trong ngôn ngữ phi ngữ cảnh.

Chứng minh: Ta chỉ cần chỉ ra một ngôn ngữ phi ngữ cảnh (do văn phạm phi ngữ cảnh sinh ra) nhưng không phải là ngôn ngữ chính quy. Một trong những ngôn ngữ như vậy là ngôn ngữ của văn phạm phi ngữ cảnh sau đây sinh ra:

$G = <\Sigma, \Delta, I, R>$, ở đây $\Sigma = \{a, b\}$, $\Delta = \{I\}$ và $R = \{I \to aIb, I \to \lambda\}$.

Rõ ràng $L(G) = \{a^n b^n : n \geq 0\}$ là ngôn ngữ phi ngữ cảnh. Ta chỉ ra $L(G)$ không phải là ngôn ngữ chính quy.

Thật vậy, giả sử $L(G)$ là ngôn ngữ chính quy, khi đó sẽ có một ôtômat hữu hạn tất định M đoán nhận nó: $M = <\Sigma, Q, F, q_0, \sigma>$.

Giả sử $N = |Q|$ là số trạng thái trong M và M sẽ đoán nhận xâu $a^N b^N$. Khi dùng xâu $a^N b^N$ như một đầu vào của M ta có:

$$q_1 = \sigma(q_0, a),\ q_2 = \sigma(q_1, a), \ldots$$

$$q_N = \sigma(q_{N-1}, a),\ q_{N+1} = \sigma(q_N, b), \ldots$$

$$q_{2N} = \sigma(q_{2N-1}, b).$$

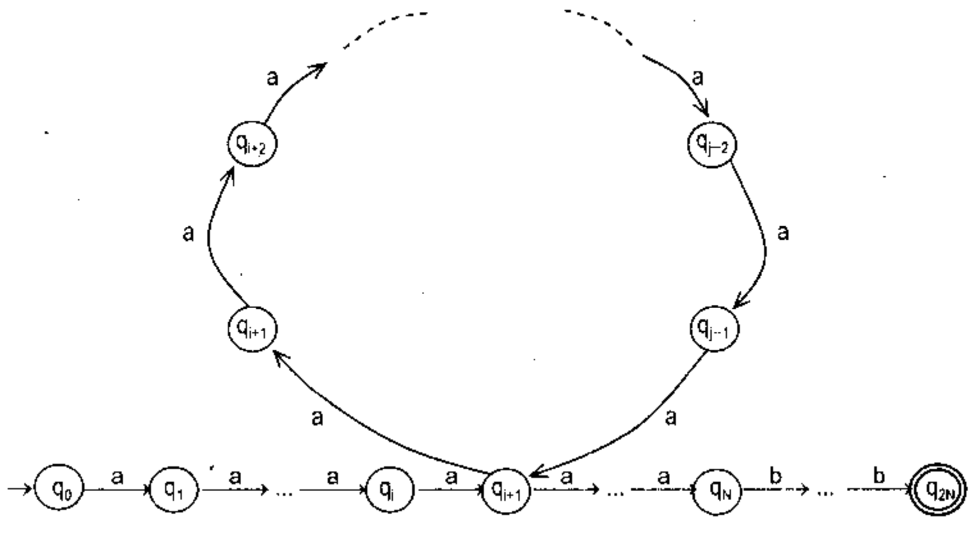
Trạng thái kết thúc là q_{2N}.

Mà M chỉ có N trạng thái. Theo nguyên lý Dirichlet, ít nhất có hai trạng thái trong N + 1 trạng thái trong dãy $q_0, q_1, \ldots, q_{2N}$ phải đồng nhất bằng nhau.

Giả sử q_i và q_j là hai trạng thái đồng nhất đó, với $N \geq j > i \geq 0$. Suy ra

$$q_j = \sigma(q_i, a^t), t = j - 1$$

Suy ra vòng dẫn từ q_i quay lại chính nó nhận được bằng cách dùng đầu vào a tổng cộng t lần (xem hình vẽ).



Đường đi tạo bởi xâu $a^N b^N$

Xét xâu đầu vào là $a^{N+1}b^{N}$. Xâu này không được M đoán nhận vì không có dạng $a^{N}b^{N}$. Tuy nhiên, khi dùng xâu này như một đầu vào của M ta vẫn sẽ kết thúc ở chính trạng thái q_{2N} như trước. Đó là vì t ký tự a trong xâu này sẽ tạo thêm một vòng dẫn nữa từ q^{i} trở lại chính nó.

Chính mâu thuẫn này chứng tỏ $L(G) = \{a^n b^n : n = 0, 1, ...\}$ không phải là ngôn ngữ chính quy. Vậy ngôn ngữ chính quy là tập con thực sự của ngôn ngữ phi ngữ cảnh.

§5. PHƯƠNG PHÁP PHÂN TÍCH TẤT ĐỊNH TRÊN LỚP NGÔN NGỮ PHI NGỮ CẢNH

Trong lý thuyết ngôn ngữ, một trong những vấn đề người ta quan tâm là: Cho L là một ngôn ngữ, ω là một xâu nào đó trên bảng ký tự Σ. Làm thế nào để biết được $\omega \in L$ hay không?

Dưới đây ta xét bài toán trong khuôn khổ văn phạm phi ngữ cảnh với phương pháp phân tích Top – Down, Bottom – Up và phương pháp phân tích tất định.

Phân tích Top – Down và Bottom – Up có nhược điểm là khi đưa xâu vào độ dài n thì số bước mà thuật toán cần phải phân tích là C^n (C – hằng số nào đó).

Nói cách khác, khi chiều dài của xâu vào tăng thì thời gian phân tích của thuật toán tăng rất nhanh. Vì vậy, trong phần này chúng ta đề cập tới bộ phân tích làm việc hiệu quả hơn, cụ thể thời gian phân tích tối đa là C.n (n là độ dài xâu vào, C là hằng số). Nhưng cách phân tích này chỉ làm việc được với một lớp con các văn phạm phi ngữ cảnh (lớp con văn phạm phi ngữ cảnh này chứa khá nhiều ngôn ngữ lập trình). Lớp ngôn ngữ có thể dùng phân tích tất định là lớp văn phạm phi ngữ cảnh không nhập nhằng và không có quy tắc lặp trái.

5.1. Phân tích tất định

Cho văn phạm phi ngữ cảnh $G = \langle\Sigma, \Delta, I, R\rangle$ và $A \in \Delta$. A gọi là đệ quy trái nếu trong R có tồn tại $A \to A\alpha$ với $\alpha \in V^*$. Ta cũng gọi $A \to A\alpha$ là quy tắc đệ quy trái. Ta có thể thay quy tắc đệ quy trái bằng các quy tắc không đệ quy trái nhưng vẫn tương đương về mặt ngôn ngữ. Chẳng hạn, nếu trong G có một quy tắc đệ quy trái $A \to A\alpha$ và một quy tắc không đệ quy trái $A \to \beta$, ở đây giả thiết xâu β không bắt đầu bởi A, thì ta có thể thay cặp $A \to A\alpha$ và $A \to \beta$ bởi ba quy tắc không đệ quy trái sau:

$$A \to \beta A',\ A' \to \alpha A'' \text{ và } A'' \to \wedge.$$

Chú ý: Nếu sau một số bước dẫn xuất trực tiếp mới xuất hiện quy tắc đệ quy trái thì việc thay thế không đơn giản.

Cơ sở của phương pháp phân tích tất định là phân tích Top – Down và ôtômat đẩy xuống.

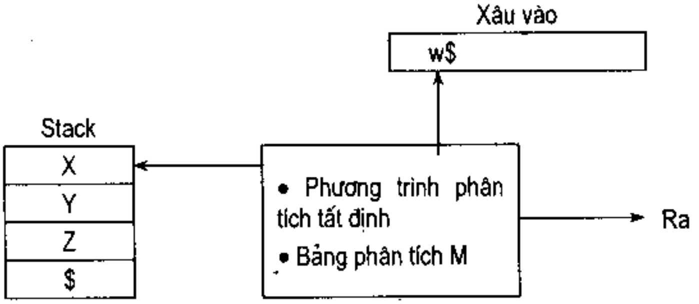
• ***Mô hình của phân tích tất định,*** gồm:

– Một băng vào, chứa xâu cần phân tích với ký hiệu kết thúc của xâu là $.

– Một stack (hay một danh sách đẩy xuống) chứa các ký hiệu của văn phạm và đáy được đánh dấu bởi $.

– Một bộ phận điều khiển chứa chương trình phân tích và bảng phân tích M, ở đây M là mảng hai chiều M[A, a] với A là ký hiệu không kết thúc, a là ký hiệu kết thúc.

Một đầu ra (xem hình dưới đây):



• ***Hoạt động của bộ phân tích tất định***

Hoạt động của bộ phân tích tất định H được xác định từ ký hiệu X trên đỉnh stack và ký hiệu vào a của xâu vào. Các khả năng sau có thể xảy ra:

Nếu X là ký hiệu kết thúc:

1) $X = a = \$$, bộ phân tích dừng, phân tích thành công.

2) $X = a \neq \$$, bộ phân tích lấy X ra khỏi stack và dịch con trỏ xâu vào sang phải 1 ô.

3) Nếu X là ký hiệu không kết thúc, bộ điều khiển sẽ xét vị trí M[X, a] của bảng phân tích M. Vị trí này có thể là:

– Một quy tắc dạng: $M[X, a] = X \to uvw$. Trong trường hợp này lấy X ra khỏi stack và thay vào đó là wvu (u nằm trên cùng của stack), bộ phân tích thực hiện chức năng mở rộng cây và in ra quy tắc vừa được sử dụng là $X \to uvw$.

– Nếu M[X, a] = error (vị trí lỗi), bộ phân tích sẽ gọi hàm khôi phục lỗi.

Ta có thể mô tả thuật toán tất định như sau:

Thuật toán:

Đầu vào (input): Xâu vào w và bảng phân tích M của văn phạm G.

Đầu ra (output): Nếu $w \in L(G)$ thì đưa ra dẫn xuất trái nhất của w, nếu không sẽ thông báo lỗi.

(ở trạng thái đầu stack được đặt ký hiệu I$ còn xâu vào là w$)

```
Repeat
    Đặt X là ký hiệu của đỉnh stack và a là ký hiệu tiếp theo;
    if X = $ then X là ký hiệu kết thúc
            if X = a then
                    Lấy X từ đỉnh stack và loại a ra khỏi w
            else error()
    else {X không phải là ký hiệu kết thúc}
            if M[X,a] = X → Y1Y2... Yk then
                    Begin
                            Lấy X từ stack;
                            Đưa YkYk-1... Y1 vào stack; {Y1 ở đỉnh}
                    End
            else error();
Until X = $; {stack rỗng}
```

Ta xét ví dụ minh hoạ sau đây:

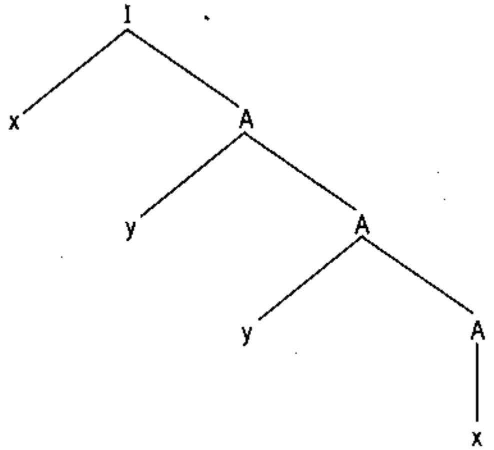
Ví dụ 1: Cho văn phạm phi ngữ cảnh G = <Σ, Δ, I, R>, ở đây:

$$\Sigma = \{x, y, z\}, \Delta = \{I, A\}, R = \{I \to xA, A \to z \mid yA\}.$$

G có đoán nhận xâu vào $xyyz \in L(G)$ hay không?

Stack	Input	Output
$I	xyyz$	
$Ax	xyyz$	$I \to xA$
$A	yyz$	
$Ay	yyz$	$A \to yA$
$A	yz$	
$Ay	yz$	$A \to yA$
$A	z$	
$z	z$	$A \to z$
$	$	

Vậy xyyz ∈ L(G) và cây dẫn xuất trái (Top – Down) có dạng:



Xâu xyyz ∈ L(G) đoán nhận được bởi phương pháp phân tích tất định.

5.2. Các hàm FIRST, FOLLOW

Phân tích Top – Down đòi hỏi văn phạm không có đệ quy trái (vì nếu có thì ta có thể dùng thuật toán để xoá đệ quy trái, tức là thay quy tắc đệ quy trái bằng quy tắc không đệ quy trái, nhưng văn phạm vẫn tương đương).

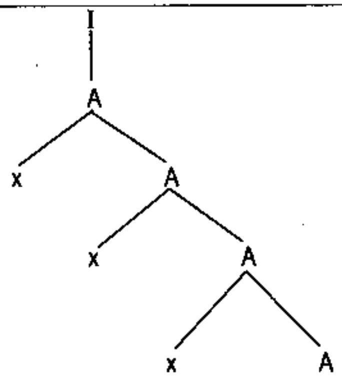
Đối với phân tích tất định người ta chỉ sử dụng văn phạm *không đệ quy trái* và *không nhập nhằng*. Thế nào là văn phạm không nhập nhằng?

Ta xét ví dụ về văn phạm sau đây:

Cho G = <Σ, Δ, I, R>, trong đó: Σ = {x, y, z}, Δ = {I, A, B},

$$R = \{I \to A|B,\ A \to xA|y,\ B \to xB|z\}$$

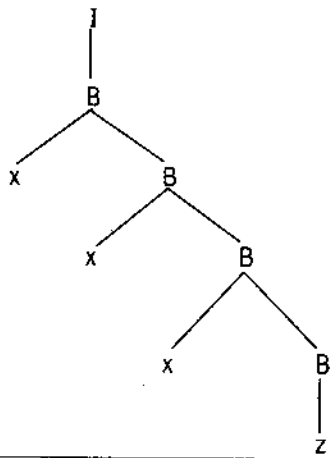
Đoán nhận xâu vào xxxz ∈ L(G) hay không?

Stack	Input	Output	Cây dẫn xuất
$I	xxxz$		
$A	xxxz$	I → A	
$Ax	xxxz$	A → xA	
$A	xxz$		
$Ax	xxz$	A → xA	
$A	xz$		
$Ax	xz$	A → xA	
A	z$		

Cây dẫn xuất trên không phải là cây dẫn xuất Top – Down (vì lá có chứa $A \in \Delta$) của xxxz.

Vì không có quy tắc $A \to z$ nên việc đoán nhận không thực hiện được và thuật toán dừng lại không cho kết quả.

Tuy nhiên, nếu ta đoán nhận theo trình tự sau đây thì thực hiện được:

Stack	Input	Output	Cây dẫn xuất
\$I	xxxz\$		
\$B	xxxz\$	$I \to B$	
\$Bx	xxxz\$	$B \to xB$	
\$B	xxz\$		
\$Bx	xxz\$	$B \to xB$	
\$B	xz\$		
\$Bx	xz\$	$B \to xB$	
\$B	z\$		
\$z	z\$	$B \to z$	
\$	\$		

Cây dẫn xuất trên là cây dẫn xuất Top – Down của xâu xxxz.

Hiện tượng trên gọi là hiện tượng quay lui (Back – tracking). Văn phạm có hiện tượng quay lui khi áp dụng phương pháp Top – Down cũng gọi là văn phạm nhập nhằng.

Làm thế nào để biết được một văn phạm phi ngữ cảnh là không nhập nhằng. Muốn giải quyết điều đó, ta cần đưa vào khái niệm hàm FIRST và FOLLOW.

• ***Định nghĩa hàm FIRST***

Giả sử trong văn phạm có quy tắc dạng $A \to \xi_1|\xi_2| \dots |\xi_k$. Khi đó ta định nghĩa:

First(A) = tập các ký hiệu kết thúc ở vị trí đầu tiên của xâu có thể dẫn xuất được từ ξ_i ($i = \overline{1, k}$). Nếu $A \to {}^*\wedge$ thì $\wedge \in$ First(A).

Cách tính hàm First:

1) Nếu $a \in \Sigma$, $\xi \in (\Sigma \cup \Delta)^*$ thì $\text{First}(a\xi) = \{a\}$.

2) Nếu $A \to \xi_1|\xi_2| \dots | \xi_k$ thì $\text{First}(A) = \bigcup_{i=1}^{k} \text{First}(\xi_i)$.

• ***Định nghĩa hàm Follow***

Follow(A) = tập các ký hiệu kết thúc A mà chúng có thể xuất hiện ngay bên phải A ở trong một số dạng câu, tức là tập các ký hiệu kết thúc a sao cho tại dẫn xuất dạng I → *αAaβ đối với α, β là các xâu bất kỳ. Nếu A là bên phải nhất trong dạng câu thì thêm $ (ký hiệu kết thúc) vào Follow(A).

Cách tính hàm Follow:

Nếu các quy tắc của văn phạm có dạng $X_i \rightarrow \xi_i A \eta_i$ ($i = \overline{1, k}$) thì

$$\text{Follow}(A) = \bigcup_{i=1}^{k} \text{First}(\eta_i)$$

5.3. Điều kiện để văn phạm phi ngữ cảnh không nhập nhằng và không đệ quy trái

Giả sử văn phạm G = <Σ, Δ, I, R> có các quy tắc A → α|β. Văn phạm G là không nhập nhằng và không đệ quy trái nếu nó thoả mãn hai điều kiện sau:

a) First(α) ∩ First(β) = ∅

b) First(A) ∩ Lollow(A) = ∅

Xét một ví dụ sau:

Ví dụ 1: Cho G = <Σ, Δ, I, R>, trong đó: Σ = {x, y, z}, Δ = {I, A}, R = {I → xA, A → z | yA}.

Trong văn phạm này có quy tắc dạng: A → z | yA.

Ta có: First(A) = First(z) ∪ First(yA) = {z} ∪ {y} = {z, y}, ở đây First(z) ∩ First(yA) = {z} ∩ {y} = ∅, thoả mãn điều kiện a.

Ta lại có First(A) = {y, z} và Follow(A) = {^}.

Vậy First(A) ∩ Follow(A) = {y, z} ∩ {^} = ∅, thoả mãn điều kiện b.

Văn phạm trên là văn phạm không nhập nhằng và không đệ quy trái. Do đó có thể đoán nhận được xâu xyyz ∈ L(G) bởi phân tích tất định (xem bảng phân tích ở ví dụ trên).

Ví dụ 2: Cho G = <Σ, Δ, I, R>, trong đó: Σ = {x, y, z}, Δ = {I, A, B}, R = {I → A|B, A → xA|y, B → xB|z}

Trong R có chứa quy tắc I → A|B.

Ta tính First(I) = First(A) $\cup$ First(B)

$= \{$ First(xA) $\cup$ First(y)$\}$ $\cup$ $\{$ First(xB) $\cup$ First(z)$\}$

$= \{x\} \cup \{y\} \cup \{z\} = \{z, y, z\}$.

Tuy nhiên, First(A) = $\{x, y\}$, First(B) = $\{x, z\}$ suy ra

First(A) $\cap$ First(B) = $\{x, y\} \cap \{x, z\} = \{x\} \neq \varnothing$, vi phạm điều kiện a.

Do Follow(I) = $\{\wedge\}$ nên First(I) $\cap$ Follow(I) = $\varnothing$ thoả mãn điều kiện b.

Văn phạm trên là nhập nhằng, sự nhập nhằng ở đây là do dẫn xuất từ $I \rightarrow A$ hay $I \rightarrow B$ là không rõ ràng.

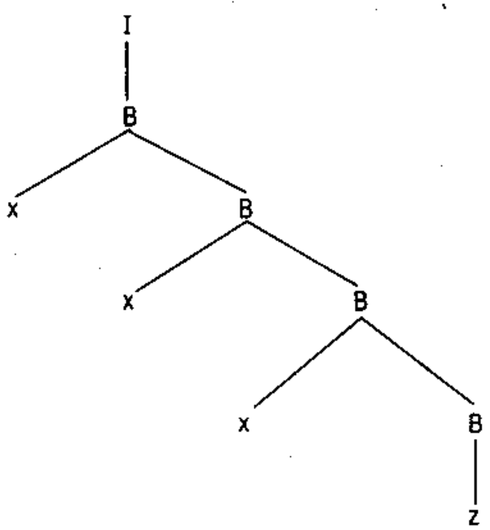
Ta có thể xây dựng một văn phạm $G' \approx G$ nhưng G' không nhập nhằng trong ví dụ sau:

Ví dụ 3: Cho $G = <\Sigma, \Delta, I, R>$ như trong ví dụ ở mục 5.2. Hãy xây dựng $G' \approx G$ mà G' không nhập nhằng.

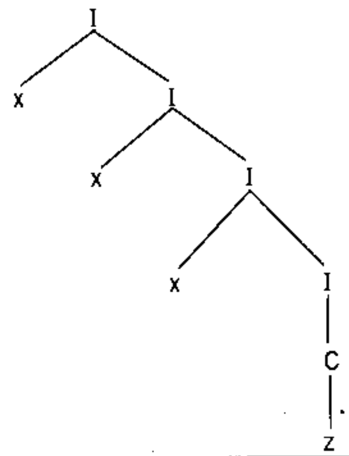
Giải: Xây dựng $G' = <\Sigma', \Delta', I, R'>$, trong đó: $\Sigma' = \{x, y, z\} = \Sigma$, $\Delta' = \{I, C\}$, $R' = \{I \rightarrow C|xI, C \rightarrow y|z\}$.

Ta chứng minh $G' \approx G$.

Thật vậy ta đã biết trong ví dụ trước đây là xâu xxxz $\in$ L(G) với cây dẫn xuất trái (Top – Down) (xem hình dưới đây).



Ta chỉ ra xxxz $\in$ L(G').

Stack	Input	Output	Cây dẫn xuất
\$I	xxxz\$		
\$Ix	xxxz\$	$I \to xI$	
\$I	xxz\$		
\$Ix	xxz\$	$I \to xI$	
\$I	xz\$		
\$Ix	xz\$	$I \to xI$	
\$I	z\$		
\$C	z\$	$I \to C$	
\$z	z\$	$C \to z$	
\$	\$		

Chứng tỏ xxxz ∈ L(G').

Dễ dàng chỉ ra điều ngược lại.

Ta chứng minh G' là văn phạm không nhập nhằng. Trong G' có các nhóm lệnh:

$$I \to C \mid xI: \begin{cases} \text{First}(I) \cap \text{Follow}(I) = \varnothing \\ \text{First}(C) \cap \text{Follow}(xI) = \varnothing \end{cases}$$

$$C \to y \mid z: \begin{cases} \text{First}(y) \cap \text{Follow}(z) = \varnothing \\ \text{First}(C) \cap \text{Follow}(C) = \varnothing \end{cases}$$

First(I) = First(C) ∪ First(xI) = {y, z} ∪ {x} = {x, y, z}

First(C) ∩ First(xI) = {y, z} ∩ {x} = ∅, thoả mãn điều kiện a.

Follow(I) = {^}.

Vậy First(I) ∩ Follow(I) = {x, y, z} ∩ {^} = R, thoả mãn điều kiện b.

Ta lại có: First(y) = {y}, First(z) = {z}.

Vậy: First(y) ∩ First(z) = ∅, thoả mãn a

First(C) = {y, z}

Follow(C) = {^}.

Suy ra First(C) ∩ Follow(C) = ∅, thoả mãn b.

Vậy G' là không nhập nhằng (để phân tích cú pháp trong văn phạm ta nên và chỉ nên sử dụng G').

BÀI TẬP

1. Cho văn phạm phi ngữ cảnh G = <Σ, Δ, I, R> với

$$R = \{I \to aAI, I \to a, A \to IbA, A \to II, A \to ba\}$$

Lấy xâu ω = aabbaa.

a) Xây dựng cây dẫn xuất đầy đủ của ω trong văn phạm trên.

b) Cho một cây dẫn xuất con của cây dẫn xuất đầy đủ ở trên.

2. Cho văn phạm phi ngữ cảnh với tập quy tắc sinh gồm các quy tắc:

$$\{I \to I + I, I \to I * I, I \to a\}.$$

Tìm cây dẫn xuất đầy đủ của xâu ω = a*a*a trong văn phạm trên.

3. ***Bài toán 1*** (Xây dựng văn phạm phi ngữ cảnh chuẩn)

Cho trước văn phạm phi ngữ cảnh bất kỳ G = <Σ, Δ, I, R>.

Xây dựng văn phạm phi ngữ cảnh chuẩn G_1 = <Σ, Δ_1, I, R_1>, tức là văn phạm mà mọi quy tắc có dạng: $A \to BC$, $A \to a$, ở đây $A, B, C \in \Delta_1$, còn $a \in \Sigma$ sao cho $G_1 \approx G$.

Ta xây dựng G_1 = <Σ, Δ_1, I, R_1> theo các bước sau:

Bước 1: Xây dựng G' = <Σ, Δ', I, R'> ≈ G = <Σ, Δ, I, R>.

a) $\Delta' = \Delta \cup \bar{\Sigma}$, ở đây $\bar{\Sigma} = \{\bar{a} : a \in \Sigma\}$.

b) $R' = \hat{R} \cup \bar{R}$, ở đây $\hat{R}$ = tập các quy tắc trong R mà mọi ký hiệu $a \in \Sigma$ đều được thay bởi $\bar{a} \le \bar{\Sigma}$; còn $\bar{R} = \{\bar{a} \to a : a \in \Sigma\}$.

G' xây dựng như trên sẽ tương ứng với G và mọi quy tắc trong R' đều có dạng $A \to BC$, hoặc $A \to B_1B_2... B_n$ $(n \ge 3)$, hoặc $A \to \bar{a}$ và $\bar{a} \to a$, ở đây $A, B, C, B_1, B_2, ..., B_n, \bar{a} \in \Delta'$ và $a \in \Sigma$.

Bước 2: Xây dựng văn phạm phi ngữ cảnh chuẩn G_1 = <Σ, Δ_1, I, R_1>.

a) $\Delta_1 = \Delta' \cup \{F_1, F_2, ..., F_{n-2}\}$.

b) $R_1 = \{\bar{a} \to a : a \in \Sigma\} \cup \{A \to a\} \cup \{A \to B_1F_1, F_1 \to B_2F_2, ...,$
$F_{n-3} \to B_{n-2}F_{n-2}, F_{n-2} \to B_{n-1}B_n\}$.

Lưu ý: Trong R_1 thì tập $\{\bar{a} \to a : a \in \Sigma\}$ chính là $\bar{R}$, tập $\{A \to a\}$ chính là tập quy tắc tương đương với quy tắc không chuẩn $A \to \bar{a}$ (kết hợp với $\bar{a} \to a$), còn tập $\{A \to B_1F_1, ..., F_{n-2} \to B_{n-1}B_n\}$ chính là tập quy tắc chuẩn tương đương về mặt sinh ngôn ngữ với quy tắc không chuẩn $A \to B_1B_2 \dots B_n$ ($n \ge 3$).

Rõ ràng $G_1 \approx G' \approx G$ và G_1 là dạng chuẩn cần tìm.

Ví dụ: Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a, b\}$, $\Delta = \{I\}$, $R = \{I \to aIb, I \to ab\}$.

Hãy xây dựng văn phạm phi ngữ cảnh chuẩn $G_1 = <\Sigma, \Delta_1, I, R_1> \approx G$.

Áp dụng bài toán 1, tìm các văn phạm phi ngữ cảnh chuẩn của các văn phạm phi ngữ cảnh G_1, G_2, G_3, G_4, G_5, G_6 và G_7 trong bài tập 11 chương 9.

4. ***Bài toán 2*** (Loại ký hiệu vô sinh)

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$. Xây dựng văn phạm phi ngữ cảnh $G' = <\Sigma, \Delta', I, R'> \approx G$ nhưng trong G' mọi $X \in \Delta'$ đều là ký hiệu hữu sinh, tức là tồn tại dẫn xuất $\alpha X\beta \vdash \omega$, ở đây $\omega \in \Sigma^*$ còn $\alpha, \beta \in (V)^*$.

Bước 1: Xây dựng Δ' theo thủ tục sau:

– Kết nạp I vào Δ';

– Nếu $A \to \omega$ mà $\omega \in \Sigma^*$ thì kết nạp A vào Δ';

– Nếu $A \to \theta$ mà mọi ký hiệu trong θ hoặc là thuộc Σ hoặc là thuộc Δ' thì kết nạp A vào Δ'.

Bước 2: Xây dựng R' theo thủ tục sau:

Duyệt tất cả các quy tắc trong R, quy tắc nào có vế trái và vế phải gồm toàn ký hiệu thuộc $\Sigma \cup \Delta'$ thì kết nạp vào R'.

Do số quy tắc của R là hữu hạn nên thủ tục này sẽ dừng lại khi duyệt hết các phần tử trong R và ta có R'.

Sau hai bước làm việc ta có $G' = <\Sigma, \Delta', I, R'>$ là văn phạm phi ngữ cảnh không có ký hiệu vô sinh và $G' \approx G$.

Ví dụ: Cho $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a\}$, $\Delta = \{I, A, B\}$,
$R = \{I \to AB, I \to a, A \to a\}$.

Xây dựng $G' = <\Sigma, \Delta', I, R'\}$ không có ký hiệu vô sinh và $G' \approx G$.

Bước 1:

+ $I \to AB$ và $A \to a$ cho ta A là hữu sinh còn B là vô sinh. Đưa A vào Δ' và loại B khỏi Δ'.

+ $I \to a$, chứng tỏ I là hữu sinh, đưa I vào Δ' ta có $\Delta' = \{I, A\}$.

Bước 2:

+ Quy tắc $I \to AB$ không kết nạp vào R' vì $B \notin \Sigma \cup \Delta' = \{I, A, a\}$.

+ Quy tắc $A \to a$ và $I \to a$ kết nạp vào R', tức là $R' = \{I \to a, A \to a\}$ vì các ký hiệu $A, a, I \in \Sigma \cup \Delta'$.

Văn phạm $G' = <\Sigma, \Delta', I, R'> \approx G$ và G' không có ký hiệu vô sinh.

a) Áp dụng bài toán 2, hãy loại ký hiệu vô sinh trong các văn phạm phi ngữ cảnh sau:

1) G_1 với tập quy tắc $R_1 = \{I \to A, I \to a, A \to AB, B \to b\}$;

2) G_2 với tập quy tắc $R_2 = \{I \to A, I \to a, A \to aB, A \to bI, A \to b, B \to AB, B \to Ba, C \to AI, C \to b\}$;

3) G_3 với tập quy tắc $R_3 = \{I \to AB, I \to CA, B \to BC, B \to AB, A \to a, C \to aB, C \to b\}$.

b) Viết giải thuật tìm Δ' trong bài toán trên.

5. ***Bài toán 3*** (Loại ký hiệu không đến được)

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$. Xây dựng văn phạm phi ngữ cảnh $G' = <\Sigma', \Delta', I, R'> \approx G$ mà trong G' không có ký hiệu không đến được, hay G' toàn ký hiệu đến được, tức là với mọi $X \in \Sigma' \cup \Delta'$ ta luôn có $I \vdash \alpha X\beta$, ở đây $\alpha, \beta \in (\Sigma' \cup \Delta')^*$.

$G' = <\Sigma', \Delta', I, R'>$ được xây dựng theo các bước sau:

Bước 1: Xây dựng Σ' và Δ':

– Kết hợp I vào Δ';

– Xét quy tắc $A \to \theta$ với $A \in \Delta'$. Với quy tắc này, kết nạp những ký hiệu kết thúc có mặt trong θ vào Σ', các ký hiệu không kết thúc trong θ kết nạp vào Δ';

Thủ tục này dừng lại khi duyệt hết các quy tắc trong R.

Bước 2: Xây dựng R' theo thủ tục sau:

Duyệt tất cả các quy tắc $A \to \theta$ trong R và kết nạp các quy tắc này vào R' nếu tất cả các ký hiệu có mặt trong quy tắc đều thuộc vào tập $\Sigma' \cup \Delta'$.

Sau hai bước làm việc ta nhận được $G' = <\Sigma', \Delta', I, R'> \approx G$, trong đó G' gồm toàn ký hiệu đến được.

Ví dụ: Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a\}$, $\Delta = \{I, A\}$, $R = \{I \to a, A \to a\}$.

Xây dựng $G' = <\Sigma', \Delta', I, R'>$ không có ký hiệu không đến được và $G' \approx G$.

Bước 1: Duyệt các quy tắc $I \to a$, $A \to a$ ta có $\Sigma' = \{a\}$, $\Delta' = \{I\}$ (ký hiệu A bị loại vì không đến được).

Bước 2: Chỉ có quy tắc $I \to a$ là gồm các ký hiệu thuộc $\Sigma' \cup \Delta' = \{I, a\}$ nên $R' = \{I \to a\}$.

Tóm lại $G' = <\Sigma', \Delta', I, R'> \approx G$ và G' gồm toàn ký hiệu đến được.

Áp dụng bài toán 3 để loại các ký hiệu không đến được trong các văn phạm phi ngữ cảnh G_1, G_2, G_3 trong câu a bài 4.

6. Văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ được gọi là văn phạm phi ngữ cảnh không có ký hiệu thừa nếu với mọi ký hiệu $X \in V = \Sigma \cup \Delta$ đều là ký hiệu có ích, tức là ký hiệu thoả mãn dẫn xuất $I \vdash \alpha X \beta \vdash \omega$, với $\omega \in \Sigma^*$; $\alpha, \beta \in V^*$.

Bài toán 4: Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$. Xây dựng văn phạm phi ngữ cảnh gồm các ký hiệu có ích $G' = <\Sigma', \Delta', I, R'>$ tương đương với G.

G' được xây dựng theo các bước sau:

Bước 1: Xây dựng văn phạm $G_1 = <\Sigma_1, \Delta_1, I, R_1>$ gồm các ký hiệu hữu sinh tương đương với G (Bài toán 2 – Loại các ký hiệu vô sinh).

Bước 2: Xây dựng văn phạm $G' = <\Sigma', \Delta', I, R'> \approx G_1 = <\Sigma_1, \Delta_1, I, R_1>$ mà G' gồm toàn ký hiệu đến được (Bài toán 3 – Loại các ký hiệu không đến được).

Khi đó G' là văn phạm phi ngữ cảnh gồm toàn ký hiệu có ích và tương đương với G.

Ví dụ: Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a\}$, $\Delta = \{I, A, B\}$, $R = \{I \to AB, I \to a, A \to a\}$.

Xây dựng $G' = <\Sigma', \Delta', I, R'> \approx G$ và gồm toàn ký hiệu có ích (tức không còn ký hiệu vô sinh và ký hiệu không đến được).

Bước 1: Áp dụng thuật toán loại ký hiệu vô sinh ta được $G_1 = <\Sigma_1, \Delta_1, I, R_1>$ với $\Sigma_1 = \Sigma = \{a\}$; $\Delta_1 = \{I, A\}$; $R_1 = \{I \to a, A \to a\}$.

Bước 2: Áp dụng thuật toán loại ký hiệu không đến được trong G_1 ta được $G' = <\Sigma', \Delta', I, R'> \approx G_1$, với $\Sigma' = \Sigma_1 = \{a\}$; $\Delta' = \{I\}$; $R' = \{I \to a\}$.

Rõ ràng $L(G') = L(G_1) = L(G) = \{a\}$.

Áp dụng bài toán 4 để xây dựng các văn phạm phi ngữ cảnh gồm toàn ký hiệu có ích và tương đương với các văn phạm phi ngữ cảnh G_1, G_2, G_3 cho trong câu a bài 4.

7. Xây dựng các văn phạm phi ngữ cảnh không có ký hiệu thừa tương đương với các văn phạm phi ngữ cảnh sau:

a) G_1 với tập quy tắc

$$R_1 = \{I \to ABC, A \to B, B \to a, B \to b, I \to cab\};$$

b) G_2 với tập quy tắc

$$R_2 = \{I \to B, B \to C, C \to c, I \to ACI, I \to ab\};$$

c) G_3 với tập quy tắc

$$R_3 = \{I \to AB, B \to a, B \to C, I \to b\};$$

d) G_4 với tập quy tắc

$$R_4 = \{I \to A, I \to a, A \to AB, B \to b\};$$

e) G_5 với tập quy tắc

$$R_5 = \{I \to A, I \to B, A \to aB, A \to bI, A \to b, B \to AB,$$
$$B \to Ba, C \to AI, C \to b\};$$

f) G_6 với tập quy tắc

$$R_6 = \{I \to AB, I \to CA, B \to BC, B \to AB, A \to a, C \to aB, C \to \}.$$

8. ***Bài toán 5*** (Loại $\wedge$–quy tắc)

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$. Xây dựng văn phạm phi ngữ cảnh $G' = <\Sigma, \Delta, I, R'> \approx G$ (sai khác nhau xâu $\wedge$ nếu $\wedge \in L(G)$) nhưng G' không có $\wedge$–quy tắc, tức là trong R' không có quy tắc $A \to \wedge$ $(A \in \Delta)$.

Trước hết, ta đưa vào khái niệm ký hiệu triệt tiêu:

– Nếu $A \to \wedge$ $(A \in \Delta)$ thì A là ký hiệu triệt tiêu trực tiếp;

– Nếu $A \to \theta$ mà θ gồm một xâu toàn ký hiệu triệt tiêu, tức là $A \vdash \wedge$ thì A gọi là ký hiệu triệt tiêu gián tiếp.

Thuật toán được mô tả như sau:

Xét các quy tắc trong R.

Bước 1: Nếu R có quy tắc $A \to \wedge$ thì loại quy tắc này ra khỏi R.

Bước 2: Nếu trong R không có quy tắc nào có ký hiệu triệt tiêu gián tiếp thì $R' = R \setminus \{A \to \wedge\}$ và $G' \approx G$.

Bước 3:

– Nếu trong R có quy tắc $A \to \theta = X_1X_2 \ldots X_n$ mà có X_{i_0} là ký hiệu triệt tiêu gián tiếp thì ta thay quy tắc $A \to X_1X_2 \ldots X_n$ bởi các quy tắc $A \to X_1X_2 \ldots X_n$ và quy tắc $A \to \theta = X_1X_2 \ldots X_{i_0-1}, X_{i_0+1} \ldots X_n$.

– Nếu $A \to \theta = X_1X_2 \ldots X_n$ mà mọi X_i ($i = \overline{1\ n}$) đều là ký hiệu triệt tiêu gián tiếp thì không được thay $X_i = \wedge$ ($i = \overline{1\ n}$).

Thuật toán dừng lại khi đã duyệt hết các quy tắc trong R và ta có R' gồm các quy tắc không có $\wedge$–quy tắc và $G' = <\Sigma, \Delta, I, R'> \approx G$ (sai khác nhau xâu $\wedge$).

Ví dụ: Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ với $\Sigma = \{a, b\}$, $\Delta = \{I, A, B\}$, $R = \{I \to AB, A \to aA, A \to \wedge, B \to bB, B \to \wedge\}$.

Xây dựng văn phạm $G' = <\Sigma, \Delta, I, R'> \approx G$ không có $\wedge$–quy tắc.

Trước hết loại các quy tắc đơn $A \to \wedge$ và $B \to \wedge$ khỏi R. Từ các quy tắc $I \to AB$ và $B \to \wedge$, $A \to \wedge$ ta thêm vào quy tắc $I \to A$ và $I \to B$. Cũng từ $A \to aA$, $B \to bB$ và $A \to \wedge$, $B \to \wedge$ ta thay vào các quy tắc $A \to a$ và $B \to b$.

Như vậy, ta có $R' = \{I \to AB, I \to A, I \to B, A \to aA, A \to a, B \to Bb, B \to b\}$. Rõ ràng G' không có $\wedge$–quy tắc và $G' \approx G$ (sai khác nhau xâu $\wedge$).

Áp dụng bài toán 5 để xây dựng văn phạm phi ngữ cảnh không có $\wedge$–quy tắc tương đương với các văn phạm phi ngữ cảnh sau:

a) G_1 với $R_1 = \{I \to aIb, I \to aABb, A \to B, B \to \wedge, A \to c\}$.

b) G_2 với $R_2 = \{I \to A, I \to B, A \to C, A \to D, B \to D, B \to I, C \to I,$
$C \to a, C \to \wedge, D \to I, D \to b, I \to I, I \to c, I \to \wedge\}$.

c) G_3 với $R_3 = \{I \to aIbI, I \to bIaI, I \to \wedge\}$.

9. ***Bài toán 6*** (Loại quy tắc đơn)

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$. Xây dựng văn phạm phi ngữ cảnh $G' = <\Sigma, \Delta, I, R'>$ không có quy tắc đơn và tương đương với G.

Trước hết, ta có thể giả thiết G là văn phạm gồm các ký hiệu có ích (vì nếu không ta áp dụng bài toán 1, 2 để được văn phạm tương đương với G gồm toàn ký hiệu có ích).

Với giả thiết trên, nếu $A \to B$ là một quy tắc đơn trong G với $A, B \in \Delta$ thì ta luôn luôn có dẫn xuất:

$$I \vdash \alpha A\beta \vdash \alpha B\beta \vdash \omega, \text{ với } \alpha, \beta \in (\Sigma \wedge \Delta)^* \text{ còn } \omega \in \Sigma^*.$$

Như vậy, ta có thể thay quy tắc đơn $A \to B$ bởi cặp quy tắc không đơn $I \to \alpha A\beta$ và $A \to \omega$ mà vẫn bảo đảm tính tương đương về sinh ngôn ngữ của văn phạm.

Vì vậy, tập quy tắc R' được xây dựng như sau:

$R' = R_1 \cup R_2$, ở đây R_1 = tập tất cả các quy tắc không đơn trong R; còn R_2 là tập các cặp quy tắc không đơn được thay thế bởi các quy tắc đơn trong R.

Rõ ràng $G' = <\Sigma, \Delta, I, R'> \approx G = <\Sigma, \Delta, I, R>$, trong đó R' không còn các quy tắc đơn.

Áp dụng bài toán 6 để xây dựng các văn phạm phi ngữ cảnh không có quy tắc đơn tương đương với các văn phạm phi ngữ cảnh sau đây và so sánh độ dài của dẫn xuất đầy đủ của cùng một xâu trong hai văn phạm có quy tắc đơn và không có quy tắc đơn:

a) G_1 với $R_1 = \{I \to I + A, I \to A, A \to A{*}B, A \to B, B \to a\}$.

b) G_2 với $R_2 = \{I \to A + B, A \to B * C, A \to B, B \to a, B \to C, C \to b\}$.

c) G_3 với $R_3 = \{I \to B, I \to C, B \to 0B, B \to 1B, B \to 011, C \to 0D, C \to 1C, C \to \wedge, D \to 0C, D \to 1D\}$.

d) G_4 với $R_4 = \{I \to T + I, I \to T, T \to F * T, T \to F, F \to a, F \to b\}$.

10. Cho một văn phạm phi ngữ cảnh có các ký hiệu thừa, có quy tắc đơn và có $\wedge$–quy tắc. Hãy dùng các thuật toán trong các bài toán 2, 3, 5 và 6 để xây dựng văn phạm phi ngữ cảnh tương đương với nó nhưng không có ký hiệu thừa, không có quy tắc đơn và không có $\wedge$–quy tắc.

11. Xây dựng ôtômat đẩy xuống không đơn định đoán nhận các ngôn ngữ L trong các trường hợp sau:

a) $L = \{\omega c \hat{\omega} : \omega \in \{0,1\}^*$, $\hat{\omega}$ là ảnh gương của ω còn $c \notin \{0, 1\}$.

b) $L = \{\omega \hat{\omega} : \omega \in \{0, 1\}^*\}$.

c) $L = \{\omega : \omega \in \{a, b\}^*$ và $\omega|_a = \omega|_b\}$.

d) $L = \{\omega : \omega \in \{a, b\}^*$ và $\omega|_a = 2\omega|_b\}$.

Ở đây $\omega|_x$ chỉ số ký hiệu x có mặt trong ω;

e) $L = \{1^m 0^n : n > m > 0\}$.

12. Chứng minh rằng lớp ngôn ngữ đoán nhận bởi ôtômat đẩy xuống không đơn định rộng hơn thực sự lớp ngôn ngữ do ôtômat đẩy xuống đơn định đoán nhận.

13. Hãy đưa các văn phạm phi ngữ cảnh có quy tắc đệ quy trái về văn phạm tương đương không có quy tắc đệ quy trái:

a) G_1 với $R_1 = \{I \rightarrow I + A, I \rightarrow A, A \rightarrow A * B, A \rightarrow B, B \rightarrow a\}$;

b) G_2 với $R_2 = \{I \rightarrow Aa, I \rightarrow b, A \rightarrow Ac, A \rightarrow Id, A \rightarrow e\}$.

14. Xây dựng một thuật giải về xoá đệ quy trái trong một văn phạm phi ngữ cảnh.

15. ***Thuật toán phân tích từ trên xuống*** (Top – Down)

Input: Văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ không đệ quy trái và xâu $\omega = a_1a_2 \dots a_n \in \Sigma^*$.

Output: Cây dẫn xuất từ trên xuống của ω ($\omega \in L(G)$) và báo lỗi nếu không xây dựng được cây dẫn xuất từ trên xuống của ω ($\omega \notin L(G)$).

Thuật toán xây dựng cây dẫn xuất của ω như sau:

Bước 1: Gốc của cây là I và đưa con trỏ chỉ vào ký hiệu đầu tiên a_1 của xâu ω.

Bước 2: Giả sử X là đỉnh hiện tại và con trỏ đang trỏ vào ký hiệu hiện tại trên ω:

– Nếu $X \in \Delta$ thì chọn quy tắc có vế trái là X trong R, chẳng hạn chọn quy tắc $X \rightarrow \theta$:

+ Nếu $\theta = \wedge$ thì đỉnh tiếp theo là đỉnh bên phải của X;

+ Nếu $\theta \neq \wedge$ và $\theta = X_1X_2 \dots X_k$ thì đỉnh tiếp theo là X_1.

– Nếu $X \in \Sigma$, chẳng hạn X = a thì so sánh a với ký hiệu hiện tại trên ω:

+ Nếu ký hiệu hiện tại trên ω là X thì đỉnh tiếp theo là đỉnh bên phải của X và con trỏ dịch sang phải một ký hiệu trên xâu ω;

+ Nếu ký hiệu hiện tại trên ω khác a thì quay về đỉnh xét trước đó và lặp lại bước 1.

Thủ tục trên, sau một số bước làm việc sẽ dừng lại và có hai khả năng:

1) Xây dựng được cây dẫn xuất từ trên xuống của xâu ω, tức là cây mà khi đọc các lá từ trái sang phải ta thu được xâu ω (trường hợp này $\omega \in L(G)$).

2) Nếu duyệt mọi khả năng mà không xây dựng được xây phân tích của xâu ω thì báo lỗi và $\omega \notin L(G)$.

Ví dụ: Cho $G = <\Sigma, \Delta, I, R>$, với $\Sigma = \{a, +, *\}$, $\Delta = \{I, T, F\}$,

$$R = \{I \to T + I, I \to T, T \to F * T, T \to F, F \to a\}.$$

Tìm cây dẫn xuất từ trên xuống của $\omega = a*a*a$.

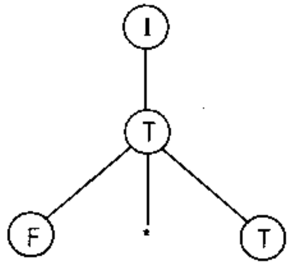
Trước tiên ta đánh số các quy tắc của R:

$I \to T + I$	(1)	$I \to T$	(2)
$T \to F * T$	(3)	$T \to F$	(4)
$F \to a$	(5)		

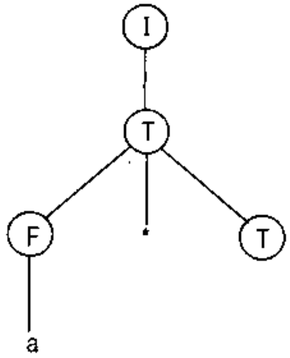
Gốc là I và con trỏ chỉ vào ký hiệu đầu tiên của ω: $a * a * a$

Do $I \in \Delta$ nên chọn quy tắc (2) ta có cây dẫn xuất

Do $T \in \Delta$ nên chọn quy tắc (3) và cây có dạng



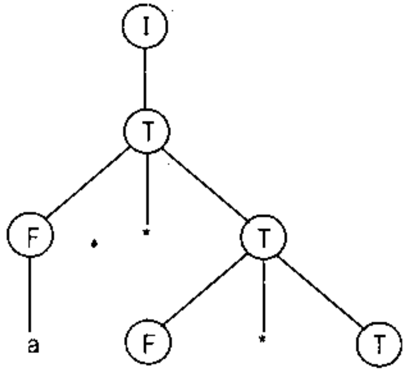
Đỉnh tiếp theo là F, chọn quy tắc (5) ta có



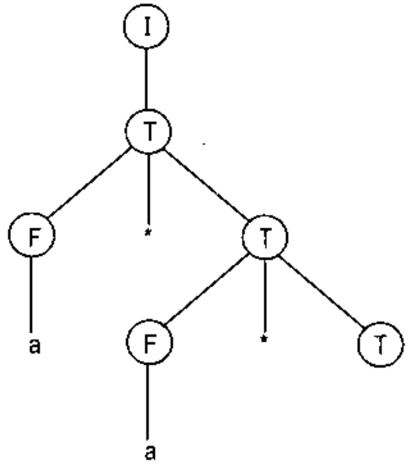
Vì đỉnh a ∈ Σ nên so sánh a với ký hiệu hiện tại cũng là a cho nên đỉnh tiếp theo là * và con trỏ dịch sang phải mộ ký hiệu: a * a * a hay * là ký hiệu hiện tại.

Do đỉnh * ∈ Σ nên đỉnh tiếp theo là đỉnh T và dịch con trỏ sang phải một ký hiệu, ký hiệu hiện tại là a: a * a * a

Do T ∈ Δ nên chọn quy tắc (3) ta có



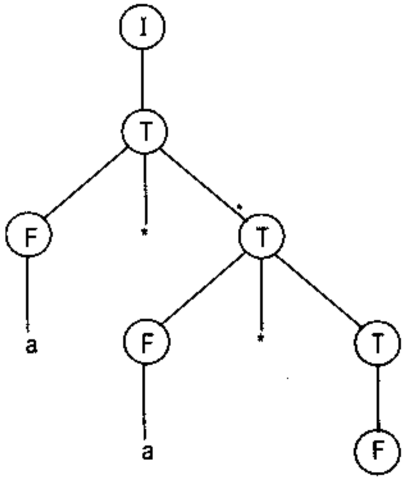
Do F ∈ Δ nên chọn quy tắc (5) ta được



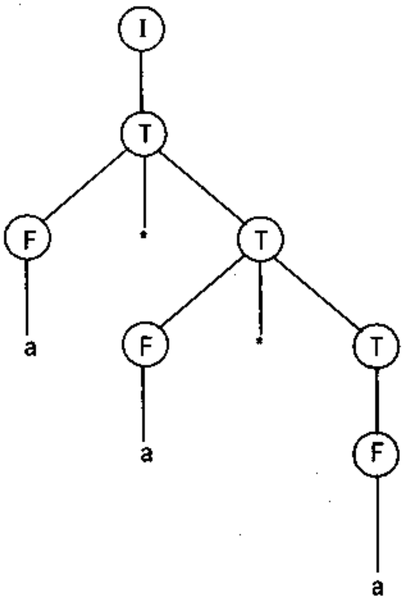
Do a ∈ Σ nên so sánh với ký hiệu hiện tại cũng là a nên đỉnh tiếp theo là * và dịch con trỏ sang phải một ký hiệu: a * a * a

Do * ∈ Σ và * cũng là ký hiệu hiện tại nên đỉnh tiếp theo là T và dịch con trỏ sang phải một ký hiệu: a * a * a

Vì T ∈ Δ nên chọn quy tắc (4) và ta có cây



Cũng do F ∈ Δ nên chọn quy tắc (5) ta có



Đỉnh tiếp theo là a. Vì a ∈ Σ và a cũng là ký hiệu hiện tại cuối cùng trên ω nên thuật toán dừng lại và cây cuối cùng là cây dẫn xuất từ trên xuống của ω = a * a * a hay ω ∈ L(G).

16. Áp dụng thuật toán tìm cây dẫn xuất từ trên xuống (bài 15) để giải bài toán khi cho văn phạm phi ngữ cảnh G với

$$R = \{I \to T + I, I \to T, T \to F * T, T \to F, F \to a, F \to b\}$$

và các xâu: $\omega_1 = a * b + b + a$; $\omega_2 = a * a * a + b * b * b$.

17. Tìm cây dẫn xuất từ trên xuống của ω = aacbc và α = aaccbb đối với văn phạm phi ngữ cảnh G với tập quy tắc sinh

$$R = \{I \to aIbI, I \to aI, I \to c\}.$$

18. Tìm cây dẫn xuất từ trên cuống trong văn phạm phi ngữ cảnh G với tập quy tắc sinh $R = \{I \to aIb, I \to c\}$ của các xâu $\omega = a^3cb^3$ và $\alpha = a^2cb^3$.

19. Cho $G = <\Sigma, \Delta, I, R>$ với: $\Sigma = \{a, b, c\}$, $\Delta = \{I, A\}$.

$$R = \{I \to aA, A \to c, A \to bA\}.$$

Cho xâu $\omega = ab^5c$ và xâu $\alpha = acbc$. Dùng thuật toán tìm cây dẫn xuất từ trên xuống để cho biết xâu nào do văn phạm ở trên sinh ra?

20. ***Thuật toán phân tích từ dưới lên*** (Bottom – Up)

Input: Văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ không chứa quy tắc đệ quy $A \to A$, $A \to \wedge$ và xâu $\omega = a_1a_2 \dots a_k \in \Sigma^*$.

Output: Cây dẫn xuất từ dưới lên của xâu ω (nếu $\omega \in L(G)$); ngược lại thì báo lỗi (tức $\omega \notin L(G)$).

Hướng dẫn: *Thuật toán được mô tả như sau:*

Bước 1: Duyệt lần lượt các ký hiệu của xâu ω lên đỉnh danh sách (stack):

a) Nếu các ký hiệu được quét lên đỉnh danh sách là vế phải của một quy tắc trong R thì thu gọn nó bởi vế trái của quy tắc đó. Nếu có nhiều quy tắc có cùng vế trái thì đánh số các quy tắc đó để lựa chọn.

b) Ngược lại, nếu không có quy tắc nào có vế phải là các ký hiệu được duyệt lên đỉnh danh sách thì gạt ký hiệu tiếp theo của ω lên đỉnh danh sách và quay lại a).

Thủ tục này tiếp tục chừng nào chưa gạt hết các ký hiệu của ω lên đỉnh danh sách.

Bước 2: Nếu đã gạt hết các ký hiệu của ω lên đỉnh danh sách mà không có một thu gọn nào được thực hiện thì quay lại bước chuyển dịch trước đó để chọn quy tắc thu gọn tiếp theo và quay lại bước 1.

Bước 3: Quá trình thu gọn thực hiện từ lá chứa ký hiệu thuộc Σ đến các đỉnh trong chứa ký hiệu thuộc Δ và cuối cùng là gốc được thu gọn bởi ký hiệu I.

Nếu ta đọc các lá từ trái sang phải mà ta thu được xâu ω thì cây đó chính là cây dẫn xuất từ dưới lên của xâu ω (trong trường hợp này $\omega \in L(G)$).

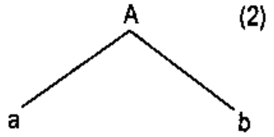
Bước 4: Nếu mọi lựa chọn có thể đã được thực hiện thông qua bước 1, 2, 3 mà vẫn không xây dựng được cây dẫn xuất từ dưới lên của ω thì thuật toán báo lỗi và $\omega \notin L(G)$.

Ví dụ: Tìm cây dẫn xuất từ dưới lên của câu ω = ababa trong văn phạm phi ngữ cảnh với tập quy tắc

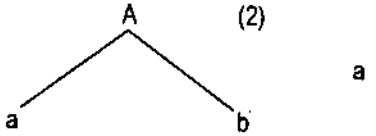
$$R = \left\{ I \xrightarrow{(1)} AB,\ A \xrightarrow{(2)} ab,\ B \xrightarrow{(3)} aba \right\}.$$

– Gạt a vào đỉnh danh sách, trong tập quy tắc R không có quy tắc nào có vế phải là a.

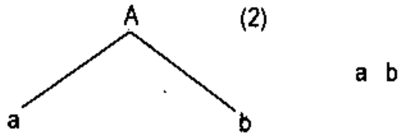
– Gạt tiếp b vào đỉnh danh sách, ta được ab, quy tắc có vế phải là ab là quy tắc (2): A → ab nên thu gọn ab bởi A ta được cây



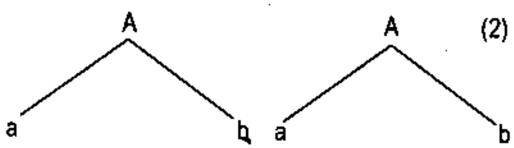
– Gạt tiếp a vào đỉnh danh sách ta được



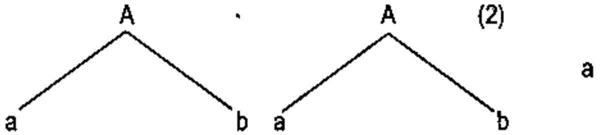
và a không thể thu gọn được bằng bất kỳ quy tắc nào trong R nên ta lại gạt tiếp b vào đỉnh danh sách ta được



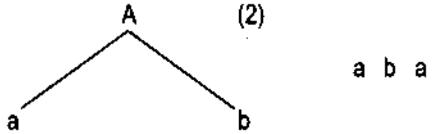
– Thu gọn ab bởi A ta được cây:



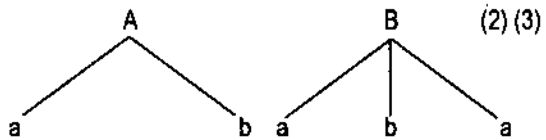
– Gạt tiếp ký hiệu a cuối cùng trong ω vào đỉnh danh sách ta được



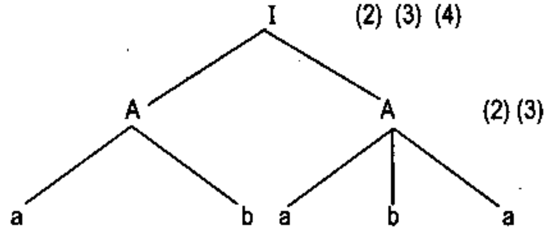
và không có quy tắc nào trong R có thể thu gọn được nên ta quay lại bước chuyển dịch trước đó và bỏ đi bước thu gọn ab bởi A ta được cây



– Thu gọn aba bởi B và áp dụng quy tắc B → aba ta được cây



– Cuối cùng thu gọn AB bởi I sẽ nhận được cây



và đọc các lá từ trái qua phải ta được ω = ababa. Vậy ω ∈ L(G). Các quy tắc và thứ tự thực hiện là (2), (3), (1) và (1).

Áp dụng thuật toán phân tích từ dưới lên để giải các bài tập 21, 22, 23, 24, 25.

21. Cho văn phạm phi ngữ cảnh G với tập quy tắc

$$R = \{I \to ABC, A \to a, B \to ab, C \to c\}$$

Tìm cây dẫn xuất từ dưới lên của xâu ω = aabc. Xâu α = abba có cây dẫn xuất từ dưới lên trong G hay không? Vì sao?

22. Dùng thuật toán phân tích từ dưới lên để chứng tỏ xâu ω = a * a ∈ L(G). Xâu α = a * a * a + a + a + a có thuộc L(G) hay không, nếu G được cho bởi tập quy tắc $R = \{I \to I + T, I \to T, T \to T * F, T \to F, F \to a\}$?

23. Cho văn phạm phi ngữ cảnh G với tập quy tắc

$$R = \{I \rightarrow ABC, A \rightarrow a, B \rightarrow ab, C \rightarrow c\} \text{ và xâu } \omega = aabc.$$

Tìm cây dẫn xuất từ dưới lên của ω.

24. Cho văn phạm phi ngữ cảnh G với tập quy tắc

$$R = \{I \rightarrow ABC, A \rightarrow a, B \rightarrow ba, C \rightarrow bc\}.$$

Xây dựng các cây dẫn xuất từ dưới lên của xâu $\omega = ababc$ và $\alpha = cabba$.

25. Cho văn phạm phi ngữ cảnh G với $R = \{I \rightarrow AB, I \rightarrow BC, A \rightarrow BA, A \rightarrow a, B \rightarrow CC, B \rightarrow b, C \rightarrow AB, C \rightarrow a\}$ và các xâu $\omega = baaba$, $\alpha = bbaaba$.

Tìm cây dẫn xuất từ dưới lên của ω và α.

26. ***Thuật toán phân tích tất định***

Input: Văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ không nhập nhằng, không đệ quy trái và xâu vào $\omega \in \Sigma^*$.

Output: Đưa ra bảng phân tích tất định và cây dẫn xuất tất định nếu $\omega \in L(G)$; ngược lại $\omega \notin L(G)$ thì báo lỗi.

Hướng dẫn: Thuật toán được mô tả như sau:

Bước 1: Tại thời điểm ban đầu stack được đặt ký hiệu I\$ ($I \in \Delta$ là ký hiệu ban đầu của văn phạm G), còn trên bảng làm việc đặt xâu ω\$ và đầu đọc nhìn vào ký hiệu đầu tiên của xâu ω. Ký hiệu mà đầu đọc nhìn vào ta gọi là ký hiệu hiện tại. \$ là ký hiệu đáy stack (ký hiệu đọc hết xâu ω trên băng làm việc).

Bước 2: Bộ phân tích của thuật toán so sánh ký hiệu trên cùng của stack là X và ký hiệu hiện tại trên băng vào.

1) Nếu $X \in \Sigma$ thì:

   – Nếu X = \$ thì phân tích kết thúc;

   – Nếu $X = a \in \Sigma$ và ký hiệu hiện tại cũng là a thì bộ phân tích đưa X ra khỏi stack và dịch đầu đọc sang phải một ký hiệu;

   – Nếu $X = a \in \Sigma$ còn ký hiệu hiện tại không phải là a thì báo lỗi.

2) Nếu $X \in \Delta$ thì bộ phân tích chọn:

   – Quy tắc có vế trái là X (chẳng hạn $X \rightarrow UVW$), trong trường hợp này lấy X ra khỏi stack, đưa xâu WVU vào stack (U nằm trên cùng của stack), sau đó bộ phân tích đưa ra quy tắc $X \rightarrow UVW$ và quay lại 1.

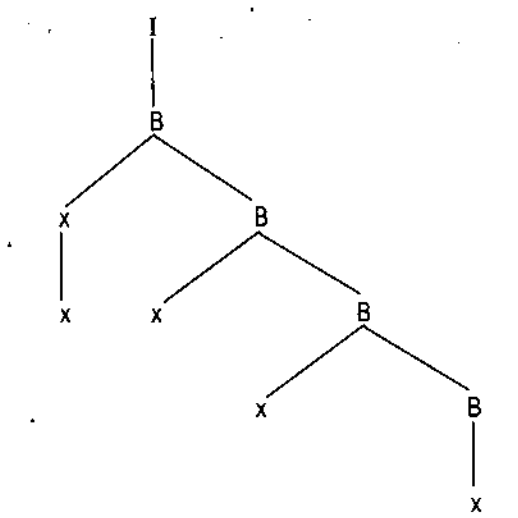
   – Trường hợp không có quy tắc nào có vế trái là X thì báo lỗi.

Bước 3: Các thủ tục trên thực hiện cho tới khi nào trong stack chỉ còn ký hiệu \$ và trên băng làm việc đầu đọc nhìn vào ký hiệu \$. Bộ phân tích đưa ra cây dẫn xuất tất định của ω, tức là $\omega \in L(G)$. Ngược lại báo lỗi và $\omega \notin L(G)$.

Ví dụ: Cho G với tập quy tắc $R = \{I \to B, A \to y, B \to xB, B \to z\}$.

Xây dựng cây dẫn xuất tất định của ω = xxxz.

Bảng phân tích và cây dẫn xuất tất định:

Stack	**Input**	**Output**	**Cây dẫn xuất tất định**
\$I	xxxz\$		
\$B	xxxz\$	$I \to B$	
\$Bx	xxxz\$	$B \to xB$	
\$B	xxz\$		
\$Bx	xxz\$	$B \to xB$	
\$B	xz\$		
\$Bx	xz\$	$B \to xB$	
\$B	\$z		
\$z	\$z	$B \to z$	
\$	\$		

27. Văn phạm G được cho như trong ví dụ của bài 26. Các xâu ω = xxxxxz và α = xxxxx có thuộc vào tập ngôn ngữ L(G) hay không? Vì sao?

28. Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ với

$$\Sigma = \{x, y, z\}, \Delta = \{I, A\}, R = \{I \to xA, A \to z, A \to yA\}$$

a) Chứng minh G là văn phạm không nhập nhằng.

b) Dùng thuật toán phân tích tất định để kiểm tra xem các xâu ω = xyyz và α = yyyyz có phải do văn phạm trên sinh ra hay không?

29. Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ với

$$\Sigma = \{x, y, z\}, \Delta = \{I, A\}, R = \{I \to A, A \to xI, A \to y, A \to z\}$$

a) Chứng minh G là văn phạm không nhập nhằng.

b) Xây dựng bảng phân tích và cây dẫn xuất tất định của các xâu: ω = xxxxxy; α = xxxz; β = yxx.

30. Cho văn phạm phi ngữ cảnh G = <Σ, Δ, I, R> với

$$\Sigma = \{x\}, \Delta = \{I\}, R = \{I \rightarrow x, I \rightarrow Ix\}.$$

a) Tìm L(G).

b) Chứng minh G là văn phạm nhập nhằng.

c) Xây dựng văn phạm phi ngữ cảnh G' = <Σ, Δ, I, R'> tương đương với G nhưng G' không nhập nhằng.

d) Xây dựng bảng phân tích tất định và cây dẫn xuất tất định của xâu $\omega = x^n$ ($n \geq 1$) đối với G'.

31. Cho văn phạm với tập quy tắc

$$R = \{I \rightarrow aIbI, I \rightarrow bIaI, I \rightarrow \wedge\}.$$

Văn phạm này có nhập nhằng hay không?

32. Cho văn phạm với tập quy tắc

$$R = \{I \rightarrow TE, E \rightarrow +I, E \rightarrow \wedge, T \rightarrow FT', T' \rightarrow T, T' \rightarrow \wedge, F \rightarrow PF', F' \rightarrow *F', F' \rightarrow \wedge, P \rightarrow a, P \rightarrow b, P \rightarrow \wedge\}.$$

a) Văn phạm trên có nhập nhằng hay không?

b) Tìm L(G).

c) Xây dựng bảng phân tích và cây dẫn xuất tất định của một xâu $\omega \in L(G)$ nào đó nếu G không nhập nhằng.

Chương 12
MÁY TURING KHÔNG ĐƠN ĐỊNH ĐÀN HỒI ĐOÁN NHẬN NGÔN NGỮ VĂN PHẠM

§1. MÁY TURING ĐƠN ĐỊNH

Định nghĩa 1: Máy Turing đơn định là một bộ gồm các thành phần <X, A, Φ, a_0, F>, trong đó:

X – bảng chữ cái hữu hạn (gọi là bảng tín hiệu vào);

A – bảng chữ cái hữu hạn (gọi là bảng trạng thái);

Φ – hàm ánh xạ từ tập $A \times X$ vào tập $A \times X \times \{0, 1, 2\}$;

a_0 – trạng thái ban đầu, $a_0 \in A$.

F – tập các trạng thái kết thúc và $F \subseteq A$.

Mô tả cách làm việc của máy:

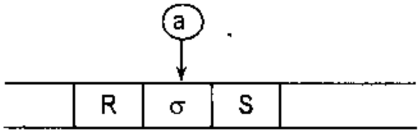
Giả sử có một băng dài vô hạn, được chia ra từng ô và có một bộ phận gọi là đầu đọc – ghi (xem hình vẽ).

Trên mỗi ô có thể viết một tín hiệu hoặc không viết tín hiệu nào. Để tiện lợi, thường những ô không viết tín hiệu nào ta viết vào tín hiệu $x_0 \in X$ nào đấy. Mặc dù băng dài vô hạn, nhưng tại mỗi thời điểm số tín hiệu trên băng là hữu hạn.

Đầu đọc có khả năng đọc tín hiệu trên ô đối diện với nó hoặc xóa tín hiệu trên ô đó và viết vào ô đó tín hiệu mới. Đầu đọc còn có khả năng chuyển dịch tùy ý sang ô bên phải hoặc ô bên trái nó, hoặc không chuyển dịch.

Định nghĩa 2: Trạng thái toàn phần của máy Turing là một từ, ta ký hiệu là $K = Ra\sigma S$, trong đó $R, S \in X^8$, $a \in A$, $\sigma \in X$.

Biểu diễn trạng thái trên băng:



Máy Turing làm việc theo từng bước, sau đây là một bước làm việc của nó: $\Phi(a, \sigma) = (a', \sigma', \alpha)$ có nghĩa là:

1) Xóa σ ở ô đối diện và viết σ' vào ô đó.

2) Máy Turing chuyển từ trạng thái a sang trạng thái trong a'. Nếu $a' \in F$ thì máy dừng lại, còn $a' \notin F$ thì máy chuyển sang bước sau.

3) Tùy theo giá trị α mà:

– Nếu $\alpha = 0$, thì đầu đọc – ghi chuyển sang trái một ô.

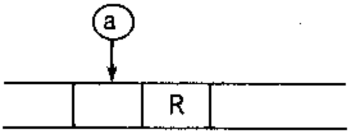
– Nếu $\alpha = 1$, thì đầu đọc – ghi đứng yên.

– Nếu $\alpha = 2$, thì đầu đọc – ghi chuyển sang phải một ô.

Sau một bước ta được trạng thái toàn phần mới K': $K \models K'$.

Định nghĩa 3: Trạng thái toàn phần K được gọi là có dạng chuẩn nếu $K = aR$, $R \in X^*$, $a \in A$.

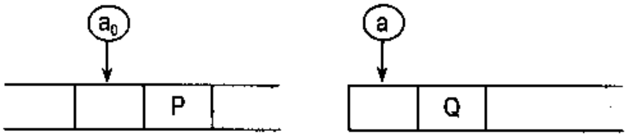
Minh họa trạng thái toàn phần dạng chuẩn trên băng:



Dưới đây để cho gọn ta nói máy Turing thay cho máy Turing đơn định.

Định nghĩa 4: Giả sử M là một máy Turing; P, Q là những từ thuộc X^*, nếu dãy $M(a_0P)$ hữu hạn (tức là máy gặp trạng thái kết thúc) và phần tử cuối cùng của dãy này có dạng aQ thì ta nói máy Turing M biến từ P thành Q và ký hiệu là: $M(P) = Q$.

Minh họa:



Cho máy Turing, tức là cho hàm $\Phi: A \times X \to A \times X \times \{0, 1, 2\}$.

Có hai cách cho máy Turing:

1) Cho dưới dạng bảng:

Tín hiệu vào / Trạng thái	x
a	$\Phi(a, x)$

2) Cho dưới dạng lệnh: $ax \to a'x'\alpha$

Mỗi một dòng như trên được gọi là một lệnh của mấy.

§2. MÁY TURING KHÔNG ĐƠN ĐỊNH ĐÀN HỒI

2.1. Mô tả sự hoạt động của máy Turing đơn định không đàn hồi

Máy Turing đơn định là máy mà trong các lệnh của nó không chứa lệnh dạng:

$$\left.\begin{aligned}\alpha \to \beta \\ \alpha \to \gamma\end{aligned}\right\} (\beta \neq \gamma)$$

Máy Turing đơn định dùng để biểu diễn thuật toán và sự làm việc của thuật toán là quá trình tiền định, tức là bước sau của nó được xác định hoàn toàn nhờ vào bước trước, đặc biệt kết quả của toàn bộ thuật toán được xác định dựa vào điều kiện ban đầu.

Trong mục này ta xây dựng máy Turing không đơn định với băng làm việc "đàn hồi" mà máy Turing đơn định chỉ là trường hợp riêng của nó.

Sự khác nhau giữa máy Turing đơn định và máy Turing không đơn định với băng làm việc đàn hồi (gọi tắt là máy Turing không đơn định đàn hồi) thể hiện qua các điểm sau đây:

1) Trong máy Turing đơn định không chứa các lệnh $\alpha \to \beta$, $\alpha \to \gamma$ với $\beta \neq \gamma$, còn trong máy Turing không đơn định thì có thể chứa các lệnh dạng trên. Dĩ nhiên tại một thời điểm nó chỉ được phép chọn một trong các lệnh để làm việc.

2) Máy Turing đơn định chỉ có một băng vào, còn máy Turing không đơn định lại làm việc trên hai băng: một băng gọi là băng vào còn một băng gọi là băng làm việc.

Trên băng vào ghi từ xuất phát mà trong quá trình làm việc nó không thay đổi, hơn nữa một ký hiệu chỉ đọc một lần. Toàn bộ công việc còn lại máy làm việc trên băng thứ hai.

3) Tính đàn hồi của băng làm việc (băng thứ 2) thể hiện:

– "Tính co lại của băng làm việc": Trong khi xóa ký hiệu ở ô làm việc, máy hủy bỏ luôn chính ô này, sao cho hai ô cạnh nó lại trở thành hai ô kề nhau.

– "Tính dãn ra của băng làm việc": Giữa hai ô bất kỳ thuộc băng làm việc, máy có thể lập thêm ô mới và có thể ghi ngay trên ô này ký hiệu cần dùng.

Ích lợi của "tính đàn hồi" là ở chỗ nó cho phép trong khi mô hình hoá các văn phạm bằng máy có thể thực hiện vòng quanh mà không cần chuyển dịch toàn bộ nội dung của băng.

Để hiểu xâu sắc hơn lý thuyết văn phạm người ta phải thông qua lý thuyết máy Turing, đồng thời khi nghiên cứu văn phạm thì máy Turing là phương tiện kỹ thuật rất tiện lợi.

Vì lý do đó ta cần xây dựng máy Turing không đơn định với băng làm việc đàn hồi.

Định nghĩa 5: Máy Turing không đơn định với băng làm việc đàn hồi gồm có:

1) Hai băng hữu hạn được chia thành từng ô. Một băng gọi là "băng vào" và một băng gọi là "băng làm việc".

2) Hai bảng chữ cái V, W và được gọi tương ứng là bảng chữ vào và bảng chữ làm việc. Các phần tử của chúng cũng được gọi một cách tương ứng là các "ký hiệu vào" và các "ký hiệu làm việc".

3) Các ký hiệu $\#, \neq \notin V \cup W$, # gọi là ký hiệu biên trái, còn $\neq$ gọi là ký hiệu biên phải.

Đặt $V' = V \cup \{\#, \neq\}$ và $W' = W \cup \{\#\}$.

4) Đầu đọc vào và đầu đọc làm việc được chuyển dịch trên băng vào và băng làm việc tương ứng.

5) Tập hữu hạn các trạng thái $Q = \{q_1, ..., q_n\}$, trong đó lấy ra phần tử q_1 gọi là trạng thái ban đầu, và tập $Q_0 \subseteq Q$ gọi là tập trạng thái kết thúc.

6) Chương trình làm việc của máy gồm các lệnh là các từ trong $V' \cup W \cup Q \cup \{T, P, \rightarrow\}$. Cụ thể gồm năm lệnh có dạng sau đây:

(i) $q_\alpha a \rightarrow q_\beta$; (ii) $Aq_\alpha \rightarrow q_\beta$;

(iii) $q_\alpha \rightarrow Aq_\beta$; (iv) $q_\alpha \rightarrow q_\beta T$;

(v) $q_\alpha \rightarrow q_\beta P$.

Ở đây $a \in V'$, $A \in W$, q_α và $q_\beta \in Q$.

Giải thích nội dung năm lệnh trên:

Mỗi lệnh thực hiện một số hoạt động và nó phụ thuộc vào các yếu tố sau đây:

– Trạng thái của máy khi bắt đầu thực hiện lệnh;

– Các ký hiệu được đọc bởi đầu đọc;

– Vị trí của đầu đọc làm việc.

Cụ thể, nếu vế trái của lệnh có ký hiệu là q_α, vế phải có ký hiệu là q_β thì chuyển từ trạng thái q_α sang trạng thái q_β, đồng thời với sự làm việc ở một trong các trường hợp sau đây:

+ Trường hợp lệnh dạng (i): $q_\alpha a \rightarrow q_\beta$

Nếu a không phải là ≠ thì chuyển đầu đọc sang phải một ô, nếu như ô đang đọc có ghi ký hiệu là a.

Nếu a là ≠ thì đọc vào và đầu đọc làm việc giữ nguyên tại chỗ.

+ Trong trường hợp (ii): $Aq_\alpha \rightarrow q_\beta$

Nếu trên ô làm việc đang đọc có ghi ký hiệu A thì hủy bỏ ô này và chuyển đầu đọc sang ô nằm sát bên trái ô vừa bị hủy (tính co lại của băng làm việc).

+ Trong trường hợp (iii): $q_\alpha \rightarrow Aq_\beta$

Lập thêm ô mới ở sát ngay bên phải ô làm việc đang được đọc và ghi trên ô này ký hiệu A, đầu đọc chuyển đến ô này (tính dãn ra của băng làm việc).

+ Trường hợp (iv): $q_\alpha \rightarrow q_\beta T$

Chuyển đầu đọc làm việc về phía trái một ô, nếu nó không ở biên trái #.

+ Trường hợp (v): $q_\alpha \to q_\beta P$

Chuyển đầu đọc làm việc về phía phải một ô, nếu nó không ở biên phải #.

Ta gọi tập hợp các hoạt động được chỉ ra ở trên là một bước làm việc của máy.

Do tính không đơn định mà tại một thời điểm nào đó máy có thể sử dụng cho nhiều lệnh khác nhau, nhưng nó chỉ được thực hiện một lệnh, do đó nó phải chọn một lệnh trong số các lệnh (lấy một lệnh để làm việc).

Định nghĩa 6 (Trạng thái toàn phần)

Trạng thái toàn phần của máy Turing không đơn định với băng làm việc đàn hồi là bộ sắp thứ tự (q_α, x', x", X', X"), ở đây $q_\alpha \in Q$, $x' \in (V')^*$, $x'' \in (V')^+$, $X' \in (W')^*$, $X'' \in (W')^+$, x'x" là một từ trên băng vào với x' là phần nằm bên trái đầu đọc và x" nằm bên phải đầu đọc. X'X" là từ viết trên băng làm việc. X' là phần nằm bên trái đầu đọc làm việc, còn X" nằm bên phải đầu đọc làm việc.

Định nghĩa 7 (Trạng thái toàn phần ban đầu và kết thúc)

– Trạng thái toàn phần có dạng (q_1, $\wedge$, #x$\neq$, $\wedge$, #) với $x \in V^*$ được gọi là trạng thái toàn phần ban đầu, ở đây q_1 là ký hiệu trạng thái ban đầu của máy (gọi tắt là: x – trạng thái toàn phần ban đầu).

– Trạng thái toàn phần có dạng (q_f, #x, $\neq$, #, $\wedge$) với $q_f \in Q_0$, $x \in V^*$ gọi là trạng thái toàn phần kết thúc (x – trạng thái toàn phần kết thúc).

Định nghĩa 8: Nếu trạng thái toàn phần S' nhận được từ trạng thái toàn phần S bằng cách áp dụng một lệnh nào đó của máy thì ta nói rằng S' đạt được trực tiếp từ S trong máy M (ký hiệu $S \underset{M}{\models} S'$, hay gọn hơn là $S \models S'$).

Định nghĩa 9:

1) Dãy các trạng thái toàn phần $C = (S_0, S_1, ..., S_n)$ $(n \geq 1)$ được gọi là một tính toán của máy M nếu với mỗi i ($i = \overline{1, n}$) ta có $S_{i-1} \models S_i$. Số n gọi là độ dài của tính toán C.

2) Trong tính toán C thì trạng thái toàn phần S_n đạt được từ trạng thái toàn phần S_0 nên ta viết $S_0 \vdash S_n$.

Nói cách khác, nếu trong tính toán $C = (S_0, S_1, ..., S_n)$ mà $S_{i-1} \models S_i$ ($\forall i = 1, 2, ..., n$) thì ta nói $S_0 \vdash S_n$.

3) Nếu một tính toán bắt đầu bởi x – trạng thái toàn phần ban đầu thì ta nói tính toán đó là x – tính toán.

4) Nếu một tính toán bắt đầu bởi x – trạng thái toàn phần ban đầu và kết thúc bằng x – trạng thái toàn phần kết thúc thì tính toán đó gọi là x – tính toán đầy đủ.

5) [x, y] – tính toán là một tính toán bắt đầu bằng x – trạng thái toàn phần ban đầu và kết thúc bởi trạng thái toàn phần có dạng (q_f, #x, $\neq$, $\wedge$, y).

Chú ý: Từ cách định nghĩa trên ta có x – tính toán đầy đủ chính là [x, $\wedge$] – tính toán.

Định nghĩa 10: Nếu tồn tại x – tính toán đầy đủ của máy M thì ta nói rằng máy M đoán nhận từ x.

Tập các từ mà máy M đoán nhận được gọi là ngôn ngữ của máy M và ký hiệu là L(M).

Chú ý:

– Trong quá trình x – tính toán các biên được ghi bên trái của ô làm việc không bao giờ bị triệt tiêu và nó cũng không xuất hiện ở ô làm việc nào khác. Vì lý do đó, khi nói "trên băng làm việc ghi ký hiệu A" và "băng làm việc triệt tiêu" có nghĩa là trên băng làm việc ghi từ #A và trên băng làm việc tất cả các ô đều bị triệt tiêu (trừ ô có ghi ký hiệu #).

– Một tính toán đầy đủ bắt đầu và kết thúc không có băng làm việc thì toàn bộ từ vào phải được đọc cho tới tận cùng của tính toán đầy đủ.

Định nghĩa 11: (Mã của tính toán)

Nếu một lệnh của máy đàn hồi M được đặt tương ứng 1 – 1 với ký hiệu C_j nào đó. Lập từ $C_{j1} \ldots C_{j2} \ldots C_{jn}$, ở đây ($C_{ji}$ ($i = \overline{1,n}$) là ký hiệu được đặt tương ứng với lệnh thực hiện ở bước thứ i của tính toán C = (S_0, S_1, ..., S_n). Khi đó từ $C_{j1} \ldots C_{j2} \ldots C_{jn}$ được gọi là mã của tính toán C.

Chú ý: Mỗi tính toán C có đúng duy nhất một mã và C khôi phục lại được nhờ vào mã của nó và trạng thái ban đầu của C.

2.2. Mô hình máy một băng

Định nghĩa 12: (Mô hình máy một băng)

Giả sử M là máy Turing không đơn định đàn hồi với bảng vào V, bảng làm việc W.

Ta sẽ gọi máy Turing đàn hồi $\widetilde{M}$ với bảng làm việc $W' \supseteq V \cup W \cup \{d\}$ ($d \notin V \cup W$) là mô hình một băng trực tiếp của M nếu [x, y] – tính toán của máy M tồn tại khi và chỉ khi trạng thái toàn phần (q_f, ∧, #≠, #x, dy) với $q_f \in Q_0$ đạt được từ trạng thái toàn phần (q_1, ∧, #≠, ∧, #xd) trong $\widetilde{M}$.

Định lý 1: Đối với bất kỳ máy Turing không đơn định đàn hồi bao giờ cũng xây dựng được mô hình một băng trực tiếp của nó.

Chứng minh: Ta mô tả cách làm việc của mô hình một băng trực tiếp $\widetilde{M}$ của M như sau: Trong "khu vào" (bên trái d) và "khu làm việc" (bên phải d) của băng làm việc, máy $\widetilde{M}$ làm việc hoàn toàn như máy M trong băng làm việc. Nhưng mỗi khi $\widetilde{M}$ chuyển từ "bước vào" sang "bước làm việc" hoặc ngược lại thì đầu đọc của $\widetilde{M}$ sẽ chuyển từ khu này sang khu khác, đồng thời nó đánh dấu ô vừa được đọc để có thể tìm được ô này khi quay trở lại. Với cách làm việc của $\widetilde{M}$ như trên ta thấy ngay [x, y] – tính toán của M suy ra (q_1, ∧, #≠, ∧, #xd) $\vdash$ (q_f, ∧, #≠, #x, dy) trong $\widetilde{M}$ và ngược lại.

Ví dụ: Nếu [x, y] – tính toán của M có dạng:

$C = (S_0, S_1, ..., S_n)$, trong đó $S_0 = (q_1, \wedge, \#x\neq, \wedge, \neq)$,

$S_1 = (q_2, \#x', x''\neq, \wedge, \#y'), ..., S_n = (Q_f, \#x, \neq, \wedge, \neq y)$.

Với x = x'x", ứng với S_0 của M ta có $\tilde{S}_0$ của $\widetilde{M}$ là:

$$\tilde{S}_0 = (q_1, \wedge, \#\neq, \wedge, \#xd).$$

Theo cách làm việc của M, nếu có $S_0 \models S_1$ trong M thì trong $\widetilde{M}$ có $\tilde{S}_0 \models \tilde{S}_1$ với $\tilde{S}_1 = (q_2, \wedge, \#\neq, \#x', x''dy')$. Nếu trong M có $S_1 \models S_2$ thì trong $\widetilde{M}$ có $\tilde{S}_1 \models \tilde{S}_2$, trong đó $S_2 = (q_3, \#x'x_1'', x_2''\neq, \wedge, y'y'')$ với $x'x_1''x_2'' = x$. Khi đó $\tilde{S}_2 = (q_3, \wedge, \#\neq, \#x'x_1'', x_2''dy'y''), ...$

Cuối cùng nếu trong M có $S_{n-1} \models S_n$ mà $S_n = (q_f, \#x, \neq, \wedge, \#y)$ thì trong $\widetilde{M}$ ta cũng có $\tilde{S}_{n-1} \models \tilde{S}_n$ với $\tilde{S}_n = (q_f, \wedge, \#\neq, \#x, dy)$.

Tóm lại $\widetilde{M}$ là mô hình một băng trực tiếp của M.

§3. SỰ TƯƠNG ĐƯƠNG GIỮA MÁY TURING KHÔNG ĐƠN ĐỊNH ĐÀN HỒI VÀ VĂN PHẠM CHOMSKY

Định nghĩa 13: Ta nói máy Turing không đơn định đàn hồi M và văn phạm G là tương đương nhau nếu L(M) = L(G).

Định lý 2:

1) Đối với máy Turing không đơn định đàn hồi có thể xây dựng được văn phạm tương đương với nó.

2) Đối với văn phạm bất kỳ có thể xây dựng được máy Turing không đơn định đàn hồi tương đương với nó.

Chứng minh:

1) Giả sử M là máy không đơn định đàn hồi với bảng vào V và M_1 là mô hình một băng trực tiếp của M. ta xây dựng M_2 có hai trạng thái q', q" sao cho với mọi từ $x \in V^*$ thì trạng thái toàn phần (q", $\wedge$, #$\neq$, $\wedge$, #) đạt được từ trạng thái toàn phần (q', $\wedge$, #$\neq$, $\wedge$, #x), tức là (q', $\wedge$, #$\neq$, $\wedge$, #x) $\models$ (q", $\wedge$, #$\neq$, $\wedge$, #) khi và chỉ khi $x \in L(M)$.

Xây dựng văn phạm G = <V, W, I, R>, ở đây V = V (bảng vào của M), $W := (W \setminus V) \cup Q \cup \{I\}$, với W, Q tương ứng là bảng làm việc, tập trạng thái trong của M_2, còn $I \notin V \cup W \cup Q$.

R gồm các quy tắc $I \to \#q$, $\#q' \to \wedge$ và các quy tắc được xác định ứng với các lệnh của máy M_2. Cụ thể nếu M_2 có lệnh dạng $\delta q_\alpha \to q_\beta$ (ii) thì cho tương ứng với quy tắc $q_\beta \to q_\alpha\delta$, lệnh dạng $q_\alpha \to \delta q_\beta$ (iii) cho ứng với quy tắc $\delta q_\beta \to q_\alpha$, lệnh $q_\alpha \to q_\beta T$ (iv) cho ứng với quy tắc $q_\beta\gamma \to \gamma q_\alpha$ ($\gamma \in W$), còn lệnh $q_\alpha \to q_\beta P$ cho ứng với quy tắc $\gamma q_\beta \to q_\alpha\gamma$ ($\gamma \in W$). Như vậy R của G gồm các lệnh của M_2 nhưng ở dạng ngược lại.

Rõ ràng với $x \in V^*$ và $D = (I, \omega_1, \omega_2, ..., \omega_{n-1}, x)$ là dẫn xuất đầy đủ của x trong G khi và chỉ khi trong M_2 từ trạng thái toàn phần (q', $\wedge$, #$\neq$, $\wedge$, #x) đạt được từ trạng thái toàn phần (q", $\wedge$, #$\neq$, $\wedge$, #) (sở dĩ như vậy vì các quy tắc của G là các lệnh ngược trong M_2, do đó nếu trong G mà dẫn được từ I thì trong M_2 x sẽ lại triệt tiêu trong quá trình làm việc trên).

Nhưng trong M_2 thì (q', $\wedge$, #$\neq$, $\wedge$, #x) $\models$ (q", $\wedge$, #$\neq$, $\wedge$, #) khi và chỉ khi $x \in L(M)$. Vậy L(M) = L(G).

2) Giả sử G = <V, W, I, R> là một văn phạm tùy ý nào đó. Ta xây dựng máy Turing M như sau:

Bảng vào của M là V của G.

Bảng làm việc của M là $V \cup W$.

M làm việc như sau: Nó ghi lên bảng làm việc từ vào, sau đó trên băng làm việc nó thực hiện dẫn xuất tùy ý của văn phạm G với thứ tự ngược lại. Nếu kết quả dẫn xuất ngược lại đó dừng lại ở ký hiệu cuối cùng là I thì máy M tự huỷ băng làm việc và chuyển vào trạng thái kết thúc. Trường hợp ngược lại, cho máy M không bao giờ gặp trạng thái kết thúc. Từ quá trình làm việc trên ta suy ra L(M) = L(G). Định lý được chứng minh.

Chú ý: Từ định lý trên ta thấy lớp các ngôn ngữ do máy Turing không đơn định đàn hồi sinh ra trùng với lớp ngôn ngữ do văn phạm sinh ra.

Phụ lục

MỘT SỐ ĐỀ THI TUYỂN SINH SAU ĐẠI HỌC (ĐHQGHN)

Môn thi: TOÁN RỜI RẠC – NGÀNH CÔNG NGHỆ THÔNG TIN

NĂM 2000

(Thời gian 180 phút)

Câu 1: 1. Hãy lập bảng giá trị của công thức mệnh đề sau:

$$((P \to Q) \to R) \to S.$$

2. Hãy biến đổi tương đương để đưa công thức sau về dạng: không có các dấu tương đương ($\equiv$); không có các dấu kéo theo ($\to$); không kể các dấu lượng từ thì nó là tuyển của các thành phần, mà mỗi thành phần này lại là hội của các công thức không chứa dấu tuyển ($\vee$) và dấu hội ($\wedge$):

$\sim(\forall x)P(x, y) \equiv Q(y)$, ở đây dấu ~ chỉ toán tử phủ định.

Câu 2: Cho ngôn ngữ chính quy trên bảng chữ cái {0, 1}, gồm tất cả các từ chứa ít nhất một ký hiệu 1.

1. Hãy biểu diễn ngôn ngữ đó bởi biểu thức chính quy.
2. Hãy tìm văn phạm tuyến tính sinh ra ngôn ngữ đó.
3. Hãy xác định ôtômat hữu hạn đoán nhận ngôn ngữ đó.

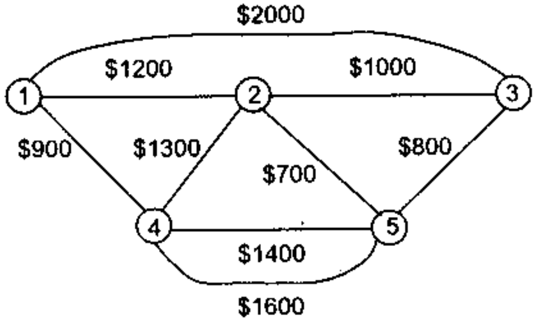
Câu 3: Cho ngôn ngữ phi ngữ cảnh $L = \{a^n0^kb^nc^md^m \mid n, m > 0, k \geq 0\}$.

1. Hãy tìm văn phạm phi ngữ cảnh sinh ra ngôn ngữ L.
2. Hãy đưa ra một dẫn suất chứng tỏ từ $a^30^2b^3c^2d^2$ được sinh ra bởi văn phạm vừa tìm được ở trên.

Câu 4: 1. Định nghĩa cây khung nhỏ nhất trong đồ thị liên thông có trọng số.

2. Mô tả các bước làm việc của thuật toán Prim tìm cây khung nhỏ nhất của đồ thị liên thông có trọng số.

3. Cho đồ thị G = <X, U> liên thông có trọng số (hình 1). Mỗi đỉnh i là một trung tâm máy tính (i = 1, 2, 3, 4, 5). Mỗi cạnh là một đường truyền thông được thuê bao, còn trọng số của mỗi cạnh là tiền thuê bao hàng tháng (tính bằng đôla) của đường truyền thông được biểu thị bằng cạnh đó. Hãy thiết kế lại mạng truyền thông nối 5 trung tâm máy tính trên sao cho tiền thuê bao hàng tháng là tối thiểu.



Hình 1

4. Hãy tìm cây khung nhỏ nhất đi qua cạnh nối đỉnh 4 và 5 với trọng số cạnh là $1600 của đồ thị G = <X, U> trên.

Câu 5: 1. Mô tả các bước làm việc của thuật toán tìm cây khung cực đại trong đồ thị liên thông có trọng số, ở đây cây khung cực đại của đồ thị liên thông có trọng số là cây khung có trọng số lớn nhất trong tất cả các cây khung có trọng số của đồ thị.

2. Tìm cây khung cực đại của đồ thị cho trong phần 3 của câu 4.

NĂM 2002

(Thời gian 180 phút)

Câu 1: 1. Cho công thức $(\forall x)P(x, 4) \vee \overline{(\forall x)(\exists y)P(x,y)}$, với $x \in \{1, 2, 3\}$. Hãy biến đổi tương đương công thức trên thành công thức không còn các lượng từ, mà chỉ có các phép toán hội ($\wedge$), tuyển ($\vee$) và phép phủ định (–). Riêng phép phủ định chỉ liên quan trực tiếp tới các vị từ cụ thể trên trường $\{1, 2, 3\} \times \{4, 5\}$.

2. Cho công thức

$$A \equiv ((\exists x)P(x) \vee (\exists x)Q(x) \vee (X \rightarrow (Y \rightarrow X))) \rightarrow \overline{(\forall x)(P(x) \vee Q(x))}$$

với P(x), Q(x) là vị từ một biến; còn X, Y là các mệnh đề sơ cấp. Thực hiện các phép biến đổi tương đương sau đây đối với A:

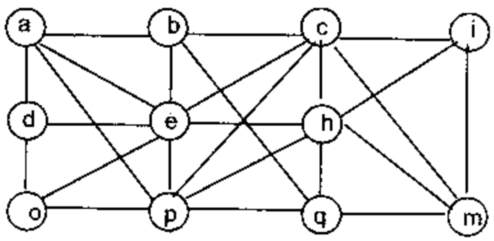
a) Khử phép toán kéo theo ($\rightarrow$).

b) Đưa phép toán phủ định (–) về trực tiếp liên quan tới các vị từ và các mệnh đề sơ cấp.

c) Đưa lượng từ $\forall$, $\exists$ lên đứng trước mọi phép toán lôgic khác.

Câu 2: 1. Phát biểu (không chứng minh) điều kiện cần và đủ để một đồ thị vô hướng có chu trình Euler. Cho ví dụ minh họa.

2. Cho đồ thị như hình 1 và cho công thức Euler r = e – v + 2, ở đây e là số cạnh, v là số đỉnh của đồ thị đơn, phẳng và liên thông G, còn r là số miền trong biểu diễn phẳng của G. Tìm r có đồ thị cho ở hình 1.



Hình 1

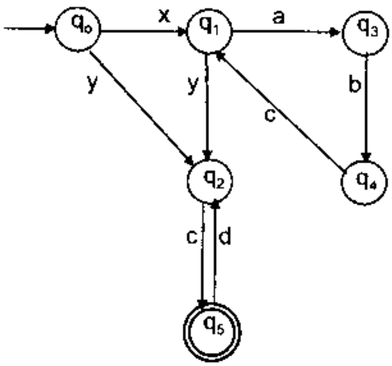
3. Phát biểu thuật toán tìm đường đi ngắn nhất từ đỉnh x đến đỉnh y trong đồ thị liên thông không có trọng số, ở đây độ dài của đường đi từ đỉnh x đến đỉnh y là số cạnh có mặt trong đường đó. Áp dụng thuật toán vừa phát biểu ở trên, hãy chỉ ra đường đi ngắn nhất từ đỉnh a đến đỉnh m trong đồ thị (hình 1). Có bao nhiêu đường như vậy.

Câu 3: Cho ôtômat hữu hạn trạng thái M với hàm chuyển trạng thái như hình 2.

1. Tìm ngôn ngữ đoán nhận L(M) của M.

2. Xây dựng văn phạm chính quy $G = <\Sigma, \Delta, I, R>$ sao cho ngôn ngữ sinh L(G) = L(M) (văn phạm chính quy là văn phạm có tập quy tắc $R = \{A \rightarrow aB, D \rightarrow b \mid A, B, D \in \Delta;\ a, b \in \Sigma$ và $\Sigma \cap \Delta = \varnothing\}$, ở đây Δ là tập các ký tự không kết thúc, Σ là tập các ký tự kết thúc, $I \in \Delta$ là ký tự ban đầu của văn phạm).

3. Cho xâu ω = xabcycdc. Hãy chỉ ra xâu ω do văn phạm chính quy G sinh ra và M đoán nhận xâu đó.



Hình 2

NĂM 2003

(Thời gian 180 phút)

Câu 1: 1. Cho công thức

$$A \equiv \overline{((\forall x)P(x) \to (\exists x)(\overline{P}(x) \wedge Q(x))) \to ((\forall x)(R(x) \to F(x)) \wedge (\forall x)F(x)) \vee (X \to X)}.$$

Thực hiện các phép biến đổi tương đương sau đây đối với A:

a) Khử phép toán kéo theo ($\to$).

b) Đưa phép toán phủ định ($-$) về trực tiếp liên quan tới từng biến mệnh đề X và từng biến vị từ P, Q, R, F.

c) Đưa các lượng từ $\forall$, $\exists$ lên trước các phép toán lôgic $\wedge$, $\vee$, $-$.

d) Tìm dạng chuẩn tắc hội và dạng chuẩn tắc tuyển của A, từ đó viết dạng chuẩn tắc hội và chuẩn tắc tuyển của $\overline{A}$..

2. Cho các vị từ P(x): "$3 \le x \le 5$"; Q(x): "$x \ge 2$" và F(x): "$x \le 8$" trên trường M = {1, 2, 3...}.

a) Tìm tập các giá trị của x trên trường M để công thức $P(x) \wedge Q(x) \wedge F(x)$ đúng.

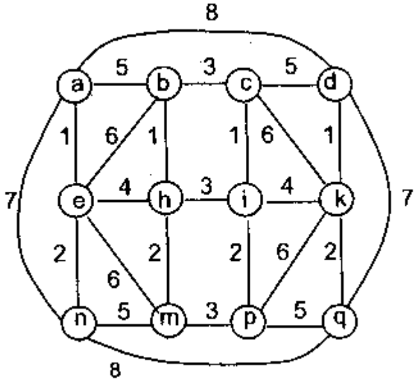
b) Tìm giá trị bé nhất của x trên trường M để công thức $(\forall x)P(x) \to (F(x) \wedge Q(x))$ đúng.

3. a) Chỉ ra suy diễn dưới đây là đúng:

$$\begin{array}{l} X_1 \to X_2 \\ \overline{X}_1 \to X_3 \\ X_3 \to X_4 \\ \hline \therefore \overline{X}_2 \to X_4 \end{array}$$

b) Chuyển mô hình suy diễn trên về dạng công thức hằng đúng tương đương.

Câu 2: 1. Chứng tỏ rằng, nếu đồ thị có đúng hai đỉnh bậc lẻ thì hai đỉng bậc lẻ này phải liên thông.



Hình 1

2. Cho mạng truyền thông như hình 1. Trong mạng truyền thông này, mỗi đỉnh là một trung tâm máy tính, mỗi cạnh nối hai đỉnh là một đường thuê bao và trọng số của mỗi cạnh là tiền thuê bao phải trả hàng tháng giữa hai trung tâm máy tính tương ứng với hai đỉnh đó. Áp dụng thuật toán Prim để thiết kế lại mạng truyền thông cho ở hình 1, sao cho tiền thuê bao hàng tháng phải trả là ít nhất.

3. Xem mạng truyền thông ở hình 1 như là một đồ thị. Tìm số màu tối thiểu để tô màu các đỉnh của đồ thị trong hình 1, sao cho hai đỉnh kề nhau được tô bởi hai màu khác nhau.

Câu 3: Cho ngôn ngữ phi ngữ cảnh L = $\{\omega d \hat{\omega} \mid \omega \in \{a,b,c\}^* \; \& \; \hat{\omega}$ là ảnh gương của xâu $\omega\}$, ở đây $\hat{\omega}$ nhận được từ ω bằng cách viết theo thứ tự ngược lại.

1. Xây dựng văn phạm phi ngữ cảnh G = <Σ, Δ, I, R> (đầy đủ theo định nghĩa) sao cho ngôn ngữ sinh L(G) = L.

2. Xây dựng ôtômat đẩy xuống (Pushdown Automata) M, sao cho ngôn ngữ đoán nhận L(M) = L(G) = L theo ngăn xếp rỗng, với L(M) là ngôn ngữ đoán nhận của ôtômat đẩy xuống M.

3. Cho hai xâu: ω_1 = aababbccdccbbabaa và ω_2 = ababbcccdcccbbabaa.

Chỉ ra $\omega_1 \in L(G)$ và $\omega_1 \in L(PA)$, còn xâu $\omega_2 \notin L(G)$ và $\omega_2 \notin L(PA)$.

4. Xây dựng cây cú pháp của xâu ω_1 đối với văn phạm phi ngữ cảnh G cho ở trên.

NĂM 2004

(Thời gian 180 phút)

Câu 1: 1. Cho mô hình suy diễn trong lôgic vị từ:

$$\begin{array}{l}(\forall x)(\overline{P}(x)\rightarrow Q(x))\\ (\exists x)\overline{P}(x)\\ (\forall x)(Q(x)\rightarrow R(x))\\ (\forall x)(S(x)\rightarrow \overline{R}(x))\\ \hline \therefore (\exists x)S(x)\end{array} \quad (*)$$

ở đây P(x), Q(x), R(x), S(x) là các vị từ một biến sinh ra trên trường M.

a) Viết công thức cơ sở của mô hình suy diễn (*) và tìm dạng chuẩn tắc tuyển của công thức cơ sở đó.

b) Mô hình suy diễn trên có đúng trên trường M không, những quy tắc suy diễn nào được áp dụng trong mô hình đó?

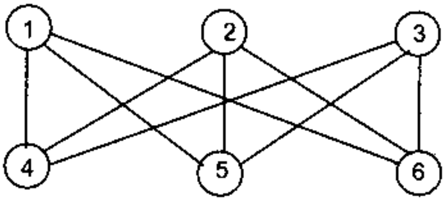
2. Hãy diễn đạt định nghĩa giới hạn $\lim\limits_{x\to x_0} f(x) = L$ dưới dạng một công thức vị từ.

3. Chỉ ra rằng công thức $\overline{(\forall x)P(x)}$ tương đương với công thức $(\exists x)\overline{P}(x)$ trên trường $M = \{a_1, a_2, ..., a_n\}$.

Câu 2: 1. Định nghĩa đồ thị đơn, đồ thị phẳng, các khái niệm liên thông đối với đồ thị vô hướng và đồ thị có hướng.

2. Cho G = <X, U> là đồ thị đơn, phẳng và liên thông, có m cạnh, n đỉnh và không có chu trình độ dài 3. Chứng minh rằng ta luôn có $m \le 2n - 4$ nếu $n \ge 3$.

3. Cho đồ thị G = <X, U> dưới dạng



Hình 1

Bằng cách áp dụng phần 2 câu 2 chứng minh rằng đồ thị G = <X, U> cho ở hình 1 là không phẳng.

Câu 3: 1. Cho văn phạm phi ngữ cảnh G = <Σ, Δ, I, R>, với tập quy tắc sinh R = {I → aIa, I → bIb, I → cIc, I → d}. Hãy viết đầy đủ văn phạm G theo định nghĩa và tìm ngôn ngữ sinh L(G) của G.

2. Xây dựng văn phạm phi ngữ cảnh chuẩn G' = <Σ, Δ', I, R'> tương đương với văn phạm G = <Σ, Δ, I, R> với tập quy tắc sinh R cho ở trên.

3. Xây dựng ôtômat đẩy xuống (Pushdown Automata) M đoán nhận ngôn ngữ L(G) trong phần 1, câu 3 theo ngăn xếp rỗng và thỏa mãn ngôn ngữ đoán nhận L(M) = L(G).

NĂM 2006

(Thời gian 180 phút)

Câu 1: 1. Ngôn ngữ L trên bảng ký tự được mô tả như sau: L gồm các xâu bắt đầu bởi lẻ các số 0, được tiếp theo bởi nhiều hơn một số 1 đứng trước một số chẵn các số 0.

a) Viết ngôn ngữ L dưới dạng ngôn ngữ chính quy trên bảng ký tự {0, 1}.

b) Viết biểu thức chính quy của ngôn ngữ chính quy L.

c) Chỉ ra văn phạm chính quy suy rộng $G = <\Sigma, \Delta, I, R>$ sao cho: L(G) = L.

d) Chỉ ra ôtômat $M = <\Sigma, Q, F, Q_0, \sigma>$ sao cho: L(M) = L(G) = L.

2. Cho dạng biên dịch dùng để định nghĩa biểu thức số học như sau:

<biểu thức> := <biểu thức> + <biểu thức>/<biểu thức> * <biểu thức>/(<biểu thức>/a/b/c

a) Chỉ ra văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ của biên dịch trên.

b) Xây dựng cây cú pháp của xâu ω = a * (b + c) + a trong văn phạm phi ngữ cảnh G của dạng biên dịch trên.

Câu 2: 1. Cho mô hình suy diễn trong lôgic vị từ cấp 1:

$$\begin{array}{l} (\forall x)(P(x) \rightarrow Q(x)) \\ (\forall x)(\overline{Q}(x) \wedge R(x)) \\ (\forall x)(\overline{R}(x) \vee F(x)) \\ (\forall x)(\overline{F}(x) \vee H(x)) \\ \underline{(\exists x)\overline{H}(x) \qquad\qquad} \\ \quad \therefore (\exists x)\overline{P}(x) \end{array} \qquad (*)$$

a) Viết công thức cơ sở của mô hình suy diễn (*). Công thức cơ sở này có hằng đúng không? Vì sao?

b) Mô hình suy diễn (*) có đúng không và những quy tắc nào được áp dụng?

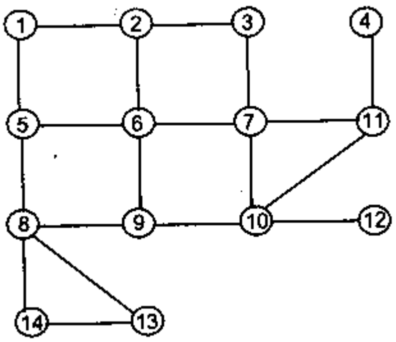
2.a) Hãy nêu các ví dụ trong cuộc sống có vận dụng mô hình suy diễn sau:

$$\begin{array}{l} A \vee B \\ \underline{\overline{A} \qquad} \\ \therefore B \end{array} \; ; \; \begin{array}{l} A \rightarrow B \\ \underline{\overline{B} \qquad} \\ \therefore \overline{A} \end{array}$$

b) Dùng phương pháp lập bảng và phương pháp biến đổi tương đương để chỉ ra hai công thức $(X \wedge Y) \rightarrow Z$ và $X \rightarrow (Y \rightarrow Z)$ là đồng nhất bằng nhau.

Câu 3: 1.a) Định nghĩa cây khung của đơn đồ thị liên thông.

b) Dùng thuật toán "Tìm kiếm tối ưu theo chiều rộng" xác định cây khung của đồ thị (hình 1) bằng cách chọn đỉnh 6 làm gốc của cây.



Hình 1

2. Nếu đồ thị đơn G = <X, U> có |X| = n đỉnh và |U| = m cạnh thì đồ thị bù của G là $\overline{G} = <X, \overline{U}>$ có bao nhiêu cạnh? (ở đây: $\overline{U} = \{(a, b) \mid (a, b) \notin U\}$)

3. Chứng minh rằng: Điều kiện cần và đủ để đồ thị G = <X, U> liên thông là đồ thị G = <X, U> có đúng một thành phần liên thông.

NĂM 2007

(Thời gian 180 phút)

Câu 1: 1. Cho công thức lôgic vị từ:

$$A \equiv (\forall x)(P_1(x) \to P_2(x)) \wedge (\forall x)(P_3(x) \to P_4(x)) \wedge (\forall x)(P_1(x) \vee P_3(x)) \to (\forall x)(P_2(x) \vee P_4(x))$$

trên trường M. Hãy kiểm tra tính hằng đúng của công thức A trên trường M bằng hai phương pháp sau đây:

a) Biến đổi tương đương.

b) Mô hình suy diễn.

2. Chỉ ra một ví dụ cụ thể trong cuộc sống hoặc trong lĩnh vực toán học có sử dụng mô hình suy diễn đặc biệt hoá phổ dụng $\dfrac{(\forall x)P(x)}{\therefore P(a)}$ (a là phần tử bất kỳ trong trường M).

3. Chứng minh:

a) $(\forall x)P(x) \wedge (\forall x)Q(x) \equiv (\forall x)(P(x) \wedge Q(x))$ trên trường $M = \{a, b\}$.

b) $(\exists x)(\exists y)P(x,y) \equiv (\exists y)(\exists x)P(x,y)$ trên trường $M' = \{a, b\} \times \{c, d\}$.

Câu 2: 1.a) Cho đồ thị phân đôi đầy đủ $K_{n,m}$. Với điều kiện nào của m và n thì đồ thị $K_{n,m}$ có chu trình Hamitơn? Hãy chỉ ra chu trình Hamiltơn đó (nếu có).

b) Chứng minh rằng, nếu đồ thị đơn $G = <X, U>$ thoả mãn với mỗi $x \in X$ ($|X| = n \geq 2$) ta luôn có $m(x) \geq \dfrac{n}{2}$ thì đồ thị $G = <X, U>$ là liên thông.

2. Đồ thị bù của đồ thị đơn $G = <X, U>$ ta định nghĩa là đồ thị $\overline{G} = < X, \overline{U} >$ với

$$\overline{U} = \{(a, b) \mid a, b \in X \text{ và } (a, b) \notin U\})$$

a) Cho đồ thị đơn $G = <X, U>$ có $|X| = n$ đỉnh và $|U| = m$ cạnh. Hỏi đồ thị $\overline{G} = < X, \overline{U} >$ có bao nhiêu cạnh?

b) Cho đồ thị $K_{3,3}$. Vẽ đồ thị bù của đồ thị $K_{3,3}$.

Câu 3: 1. Cho biểu thức chính quy: $x((0 \cup 1)(0 \cup 1))^* x \cup y(0 \cup 1)^* y$.

a) Tìm ngôn ngữ chính quy L của biểu thức chính quy cho ở trên.

b) Xây dựng ôtômat $M = <\Sigma, Q, F, Q_0, \sigma>$ (đầy đủ theo định nghĩa) sao cho ngôn ngữ đoán nhận của M là L(M) = L trong phần a (không chứng minh).

c) Xây dựng văn phạm chính quy $G = <\Sigma, \Delta, I, R>$ sao cho ngôn ngữ sinh bởi văn phạm G là L(G) = L(M) (không chứng minh).

2. Số nguyên có dấu là một số nguyên không dấu được đặt trước một dấu + hoặc một dấu –. Dạng biên dịch BNF (Backus – Naur Form) dùng để định nghĩa số nguyên có dấu được cho như sau:

<số nguyên có dấu> := <dấu> <số nguyên>
<dấu> := + | –
<số nguyên> := <chữ số> | <chữ số> <số nguyên>
<chữ số> := 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

a) Chỉ ra văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, I, R>$ (đầy đủ theo định nghĩa) có dạng biểu diễn BNF cho ở trên.

b) Xây dựng cây cú pháp của xâu ω = +9987 trong văn phạm chỉ ra trong câu a.

TÀI LIỆU THAM KHẢO CHÍNH

[1] *Toán rời rạc ứng dụng trong Tin học* (dịch từ : Mc Graw – Hill. *Discrete Mathematics its Applications,* 1994). NXB Khoa học và Kỹ thuật, Hà Nội, 2000.

[2] *Toán rời rạc cho các nhà khoa học máy tính* (tài liệu dịch từ : *Discrete Mathematics for Computer Scientits*). Khoa CNTT – Trường Đại học Khoa học Tự nhiên (ĐHQG Hà Nội), 1998.

[3] Nguyễn Hữu Anh. *Toán rời rạc.* NXB Giáo dục, Hà Nội, 1999.

[4] Nguyễn Văn Ba. *Ngôn ngữ hình thức.* NXB Khoa học và Kỹ thuật, Hà Nội, 2002.

[5] Phan Đình Diệu. *Lý thuyết ôtômat và thuật toán.* NXB Đại học và Trung học chuyên nghiệp, Hà Nội, 1977.

[6] Nguyễn Hữu Ngự. *Lý thuyết đồ thị.* NXB Đại học Quốc gia, Hà Nội, 2001.

[7] Nguyễn Tô Thành, Nguyễn Hữu Nghĩa. *Toán rời rạc.* NXB Giáo dục, Hà Nội, 1997.

[8] Đặng Huy Ruận. *Lý thuyết đồ thị và ứng dụng.* NXB Khoa học và Kỹ thuật, Hà Nội, 2000.

[9] Đặng Huy Ruận. *Lý thuyết ngôn ngữ hình thức và ôtômat.* NXB Đại học Quốc gia, Hà Nội, 2002.

[10] Đặng Huy Ruận. *Bảy phương pháp giải các bài toán lôgic.* NXB Khoa học và Kỹ thuật, Hà Nội, 2002.

[11] Đỗ Đức Giáo, Đặng Huy Ruận. *Văn phạm và ngôn ngữ hình thức.* NXB Khoa học và Kỹ thuật, Hà Nội, 1991.

[12] Đỗ Đức Giáo. *Cơ sở toán trong lập trình.* NXB Khoa học và Kỹ thuật, Hà Nội, 1998.

[13] Đỗ Đức Giáo. *Toán rời rạc.* NXB Đại học Quốc gia, Hà Nội, 2004.

[14] Đỗ Đức Giáo. *Hướng dẫn giải bài tập Toán rời rạc.* NXB Giáo dục, 2006.

Chịu trách nhiệm xuất bản :
Chủ tịch HĐQT kiêm Tổng Giám đốc NGÔ TRẦN ÁI
Phó Tổng Giám đốc kiêm Tổng biên tập NGUYỄN QUÝ THAO

Tổ chức bản thảo và chịu trách nhiệm nội dung :
Chủ tịch HĐQT kiêm Giám đốc CT CP Sách ĐH – DN
TRẦN NHẬT TÂN

Biên tập nội dung và sửa bản in :
ĐỖ HỮU PHÚ

Thiết kế mỹ thuật và trình bày bìa :
ĐINH XUÂN DŨNG

Thiết kế sách và chế bản :
ĐỖ PHÚ

TOÁN RỜI RẠC ỨNG DỤNG TRONG TIN HỌC

Mã số : 7B705Y8 – DAI

In 1.000 cuốn (QĐ : 33), khổ 16 x 24cm. In tại Công ty CP In Thái Nguyên.
Địa chỉ : Phường Quang Trung, TP. Thái Nguyên.
Số ĐKKH xuất bản : 113 – 2008/CXB/59 – 175/GD.
In xong và nộp lưu chiểu tháng 7 năm 2008.

CÔNG TY CỔ PHẦN SÁCH ĐẠI HỌC - DẠY NGHỀ
HEVOBCO
25 HÀN THUYÊN – HÀ NỘI
Website : www.hevobco.com.vn

TÌM ĐỌC SÁCH THAM KHẢO ĐẠI HỌC BỘ MÔN TOÁN
CỦA NHÀ XUẤT BẢN GIÁO DỤC

1. Giải tích hàm	*Nguyễn Xuân Liêm*
2. Bài tập giải tích hàm	*Nguyễn Xuân Liêm*
3. Giải tích (hai tập)	*Nguyễn Xuân Liêm*
4. Toán học cao cấp (3 tập)	*Nguyễn Đình Trí (Chủ biên)*
5. Bài tập toán học cao cấp (3 tập)	*Nguyễn Đình Trí (Chủ biên)*
6. Mở đầu lí thuyết xác suất và ứng dụng	*Đặng Hùng Thắng*
7. Bài tập xác suất	*Đặng Hùng Thắng*
8. Lí thuyết xác suất	*Nguyễn Duy Tiến – Vũ Viết Yên*
9. Xác suất thống kê	*Nguyễn Văn Hộ*

Bạn đọc có thể mua tại các Công ty Sách - Thiết bị trường học ở các địa phương hoặc các Cửa hàng sách của Nhà xuất bản Giáo dục :

Tại Hà Nội : 25 Hàn Thuyên; 187B Giảng Võ; 232 Tây Sơn; 23 Tràng Tiền.
Tại Đà Nẵng : Số 15 Nguyễn Chí Thanh; Số 62 Nguyễn Chí Thanh.
Tại Thành phố Hồ Chí Minh : Cửa hàng 451B - 453, Hai Bà Trưng, Quận 3;
240 Trần Bình Trọng – Quận 5.
Tại Thành phố Cần Thơ : Số 5/5, đường 30/4.
Website : www.nxbgd.com.vn

8 934980 813003

Giá : 47.500 đ