PHẠM ĐÌNH NGHIỆM

LOGIC CHUYÊN NGÀNH

Giáo trình dành cho sinh viên ngành triết học

TP. HỒ CHÍ MINH 2006

Chương I LOGIC MỆNH ĐỀ

I. Mệnh đề. Các phép toán trên mệnh đề

1. Mệnh đề

Trong Tiếng Việt (và các ngôn ngữ khác) có những câu - thường là câu tường thuật - mô tả sự vật và hiện tượng. Có những câu mô tả đúng, cũng có những câu mô tả sai sự vật và hiện tượng. Những câu như thế, cả câu đúng và câu sai, được gọi là mệnh đề[1]. Ví dụ, các câu sau:

(a) Nam là sinh viên;

(b) Khí hậu trái đất đang nóng dần lên;

(c) Bạn có thể thất vọng khi bị thất bại nhưng bạn sẽ không là gì cả nếu không nỗ lực hết mình (Beverly Silis);

(d) Nếu người vợ đẹp mà không phải là thiên thần thì người chồng vô cùng bất hạnh (J.J. Rousseau);

là các mệnh đề.

Không phải câu nào cũng hoặc đúng hoặc sai. Các câu hỏi, câu mệnh lệnh, câu cảm thán không mô tả cái gì nên không đúng mà cũng không sai. Có cả những câu tường thuật không thể xác định là đúng hay sai. Chẳng hạn, câu "Tôi nói dối" không thể là đúng, nhưng cũng không sai. Những câu không đúng, không sai như thế không phải là mệnh đề.

Các mệnh đề không thể tách ra thành các mệnh đề đơn giản hơn gọi là mệnh đề đơn. Các mệnh đề có thể tách thành các mệnh đề đơn giản hơn gọi là mệnh đề phức. Nói cách khác, mệnh đề phức được tạo thành từ các mệnh đề đơn. Các mệnh đề *(a)* và *(b)* trên đây là mệnh đề đơn, còn *(c), (d)* là các mệnh đề phức.

2. Các phép toán logic trên mệnh đề

Như trên kia đã nói, có thể xây dựng các mệnh đề phức tạp từ những mệnh đề đơn giản hơn. Việc này thực hiện được nhờ các phép toán (toán tử) logic.

Phủ định là một trong những phép toán đơn giản nhất trên mệnh đề. Đó là phép toán một ngôi. Mặc dầu trong ngôn ngữ tự nhiên một mệnh đề nào đó có thể bị phủ định bằng nhiều cách khác nhau, ở đây ta chỉ phủ định một mệnh đề bằng một cách duy nhất, bằng cách đặt dấu $\neg$ trước mệnh đề đó. Nếu A là một mệnh đề, thì $\neg A$ là phủ định của mệnh đề A.

Phép toán phủ định được định nghĩa bằng bảng chân lý sau:

Phủ định

A	$\neg A$

[1] Mệnh đề và câu, xét nghiêm ngặt, khác nhau. Nhưng trong chương trình này, để cho đơn giản, chúng tôi đồng nhất mệnh đề với câu tường thuật.

T	*F*
F	*T*

Các chữ cái *T* và *F* ở đây chỉ các giá trị chân lý *"đúng" (True)* và *"sai" (False)* tương ứng. Trong bảng trên, nếu *A* đúng thì phủ định của nó, $\neg A$, sai, và ngược lại, nếu *A* sai thì $\neg A$ là đúng.

Hội là phép toán phổ biến thứ hai trên mệnh đề. Người ta còn gọi nó là phép liên kết. Liên kết của hai mệnh đề *A* và B được ký hiệu bằng *A & B*. Bảng chân lý định nghĩa phép *hội* như sau (xem bảng). Mệnh đề *A & B* đúng khi và chỉ khi *A* đúng và *B* đúng. Các mệnh đề *A* và *B* được gọi là các thành phần liên kết của mệnh đề *A & B*.

Hội

A	*B*	*A & B*
T	*T*	*T*
T	*F*	*F*
F	*T*	*F*
F	*F*	*F*

Tuyển không nghiêm ngặt

A	*B*	$A \vee B$
T	*T*	*T*
T	*F*	*T*
F	*T*	*T*
F	*F*	*F*

Tuyển nghiêm ngặt

A	*B*	$A \veebar B$
T	*T*	*F*
T	*F*	*T*
F	*T*	*T*
F	*F*	*F*

Lựa chọn là phép tính phổ biến thứ ba trên mệnh đề. Người ta còn gọi nó là phép tuyển. Trong tiếng Việt phép toán này thường được biểu thị bằng từ "hoặc", "hoặc là", "hay", "hay là". Lựa chọn có thể được hiểu theo hai nghĩa khác nhau. Trong nghĩa thứ nhất "*A* hoặc *B*" (ký hiệu là $A \vee B$) được hiểu là đúng khi có ít nhất một trong hai thành phần A hoặc B đúng , hoặc là cả *A* và *B* cùng đúng. Trong nghĩa thứ hai "*A* hoặc *B*" (ký hiệu là $A \veebar B$) đúng khi *A* đúng, *B* sai, hoặc là khi *A* sai, B đúng. Nghĩa thứ nhất là phép tuyển không nghiêm ngặt, phép tuyển nghiêm ngặt ứng với nghĩa thứ hai. Phép tuyển nghiêm ngặt được ký hiệu là $\veebar$. Bảng chân lý của phép tuyển không nghiêm ngặt và nghiêm ngặt được dẫn ở trên.

Kéo theo là một phép toán hai ngôi được định nghĩa bằng bảng chân lý quan trọng nữa trên các mệnh đề. Với các mệnh đề A và B phép toán này cho phép tạo nên mệnh đề $A \supset B$. Nghĩa của mệnh đề này là "Nếu *A* thì *B*", hay là "*A* kéo theo *B*". Nghĩa này không được xác định rõ ràng trong những ứng dụng thông thường. Ta chỉ biết rằng "*A* kéo theo *B*" đúng có nghĩa là nếu *A* đúng thì *B* phải đúng. Trong tiếng Việt phép toán này thường được diễn đạt bằng các cụm từ "Nếu … thì … ", "Nếu … sẽ … ","Khi nào … thì … ", "Bao giờ … thì … ", "… thì …" và một số cụm từ khác. Ví dụ, các câu "Nếu không bảo vệ môi trường ngay từ bây giờ thì loài người sẽ không có tương lai" ; "Chuồn chuồn bay thấp thì mưa"; "Có nước thì có cá"; "Bao giờ chạch đẻ ngọn đa, sáo đẻ dưới nước thì ta lấy mình" … biểu đạt các mệnh đề dạng kéo theo. Trong ngôn ngữ thông thường, và cả trong các suy luận toán học hoặc các khoa học khác, nghĩa của cụm từ *"nếu ... thì ..."* và các cụm từ khác diễn đạt phép kéo theo được hiểu phụ thuộc vào văn cảnh. Câu "Nếu A thì B" trong tiếng Việt thường biểu thị một mối liên hệ giữa A và B về nội dung. Chẳng hạn, A là điều kiện, B là hệ quả (vì vậy mệnh đề loại này còn được gọi là mệnh đề điều kiện), hay A là nguyên nhân, B là kết quả. Nhưng trong logic mệnh đề chúng ta không quan tâm đến mối liên hệ về mặt nội dung đó, mà chỉ quan tâm đến mối liên hệ về giá trị chân lý của chúng mà thôi. Cụ thể là ta sẽ coi là "Nếu *A* thì *B*" chỉ sai khi *A* đúng mà *B* sai. Trong tất cả các trường hợp khác "Nếu *A* thì *B*" đúng.

Kéo theo

A	B	$A \supset B$
T	T	T
T	F	F
F	T	T
F	F	T

Tương đương

A	B	$A \equiv B$
T	T	T
T	F	F
F	T	F
F	F	T

Bảng chân lý của phép kéo theo được dẫn ở trên.

Nếu ký hiệu cụm từ "*A* tương đương *B*" là $A \equiv B$ thì ta có bảng chân lý cho phép tương đương như dẫn ở trên. $A \equiv B$ đúng khi và chỉ khi *A* và *B* có cùng một giá trị chân lý như nhau.

Độ ưu tiên thực hiện các phép toán được xác định theo thứ tự giảm dần như sau : $\neg$, &, $\vee$, $\supset$, $\equiv$. Cùng một phép toán thì chúng được kết hợp về bên phải[2], nghĩa là:

$p \vee q \vee r \quad \Leftrightarrow \quad p \vee (q \vee r)$

$p \,\&\, q \,\&\, r \quad \Leftrightarrow \quad p \,\&\, (q \,\&\, r)$

$p \supset q \supset r \quad \Leftrightarrow \quad p \supset (q \supset r)$

$\neg\neg p \quad \Leftrightarrow \quad \neg(\neg p)$

$p \equiv q \equiv r \quad \Leftrightarrow \quad p \equiv (q \equiv r)$

3. Định nghĩa các phép toán logic bằng phương pháp giải tích

Nếu ký hiệu *val(A)* là giá trị logic của công thức *A*, ký hiệu *val(A) = T* là *val(A) = 1* thì bảng định nghĩa các phép toán logic cho thấy :

val(A $\vee$ B) = max (val(A), val(B))= val (A) + val (B) (với chú ý: 1 + 1 = 1);

val(A & B) = min (val(A), val(B)) = val (A) . val (B);

val($\neg$A) = 1 – val(A);

val(A $\supset$ B) = val ($\neg$A $\vee$ B) = max(1 - val(A), val(B));

4. Công thức

[2] Không thể kết hợp về bên trái như trong toán học vì nếu như thế biểu thức $\neg\neg$A trở nên vô nghĩa.

Ta sẽ dùng thuật ngữ *công thức* để chỉ một loại biểu thức được xây dựng từ các mệnh đề đơn và các phép toán trên mệnh đề. Chính xác hơn:

(i) Tất cả các mệnh đề đơn $p, q, r, p_1, p_2, \ldots$ là các công thức.

(ii) Nếu A là công thức thì (A), $\neg A$ là công thức.

(iii) Nếu A, B là công thức thì $A \,\&\, B$, $A \vee B$, $A \supset B$, $A \equiv B$ là các công thức.

(iv) Ngoài ra không còn công thức nào khác.

Ví dụ công thức :

- p
- $p \vee (q \,\&\, r)$
- $(r \,\&\, q) \supset (((r \vee s) \,\&\, \neg q) \supset \neg s)$

Những biểu thức sau đây không phải là công thức :

- $p \,\& \vee q$,
- $\forall p \supset q$,
- p & (q ∨ r) ⊃ .

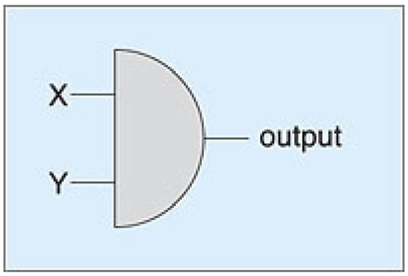
Mỗi công thức là một hàm của các biến (là các mệnh đề đơn thành phần của công thức đó) xác định trên tập các giá trị chân lý $\{T, F\}$. Hàm đó cũng nhận giá trị từ tập $\{T, F\}$. Mỗi sự phân bố các giá trị chân lý của các mệnh đề đơn cấu thành công thức A tương ứng với một giá trị chân lý của công thức A đó. Ví dụ, công thức $(p \vee q) \,\&\, (\neg r)$ có giá trị tương ứng với các phân bố giá trị chân lý của các mệnh đề đơn thành phần của nó như sau :

p	q	r	$p \vee q$	$\neg r$	$(p \vee q) \,\&\, (\neg r)$
T	*T*	*T*	*T*	*F*	*F*
T	*T*	*F*	*T*	*T*	*T*
T	*F*	*T*	*T*	*F*	*F*
T	*F*	*F*	*T*	*T*	*T*
F	*T*	*T*	*T*	*F*	*F*
F	*T*	*F*	*T*	*T*	*T*
F	*F*	*T*	*F*	*F*	*F*
F	*F*	*F*	*F*	*T*	*F*

Bảng liệt kê giá trị chân lý của công thức cùng với các phân bố giá trị của các mệnh đề đơn thành phần của nó như trong ví dụ trên đây gọi là *bảng chân lý* (hay *bảng chân trị)* – chúng ta sẽ khảo sát ở phần sau - của công thức.

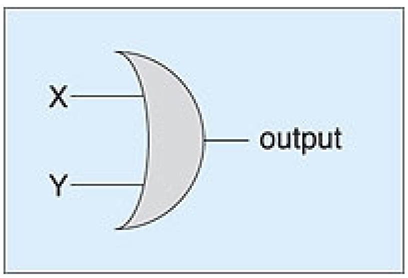
5. Các cổng logic trong kỹ thuật điện tử

Trong kỹ thuật điện tử người ta sử dụng các phần tử đặc biệt của mạch điện, gọi là các cổng logic. Các cổng logic thông thường là cổng AND, tương ứng với phép toán hội; cổng OR, tương ứng với phép tuyển không nghiêm ngặt; cổng XOR, tương ứng với phép tuyển nghiêm ngặt; cổng đảo NOT, tương ứng với phép phủ định; cổng NAND, tương ứng với phủ định của phép hội; cổng NOR, tương ứng với phủ định của phép tuyển; NXOR, tương ứng với phủ định của phép tuyển nghiêm ngặt.



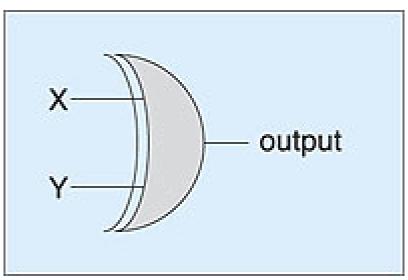
Cổng AND

Output = X & Y
(đầu ra có tín hiệu khi và chỉ khi cả hai đầu vào X và Y đều có tín hiệu)



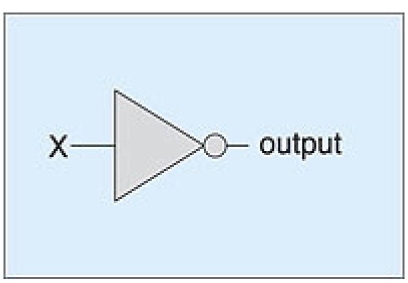
Cổng OR

Output = $X \vee Y$
(đầu ra có tín hiệu khi và chỉ khi có ít nhất một đầu vào X hoặc Y có tín hiệu)



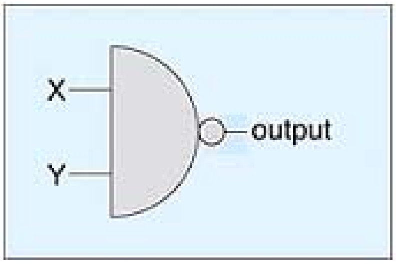
Cổng XOR

Output = $X \vee Y$
(đầu ra có tín hiệu khi và chỉ khi có đúng một đầu vào X hoặc Y có tín hiệu)



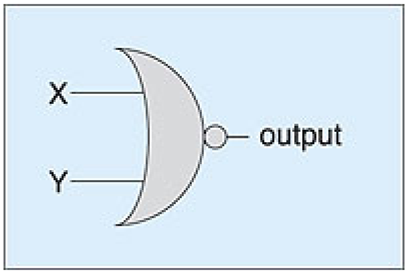
Cổng NOT (cổng đảo)

Output = $\neg X$
(đầu ra chỉ có tín hiệu khi đầu vào không có tín hiệu, và ngược lại)



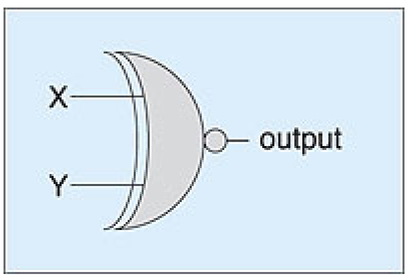
Cổng NAND

Output = $\neg (X \& Y)$
(đầu ra chỉ không có tín hiệu khi không đầu vào nào có tín hiệu, các trường hợp khác đầu ra đều có tín hiệu)



Cổng NOR

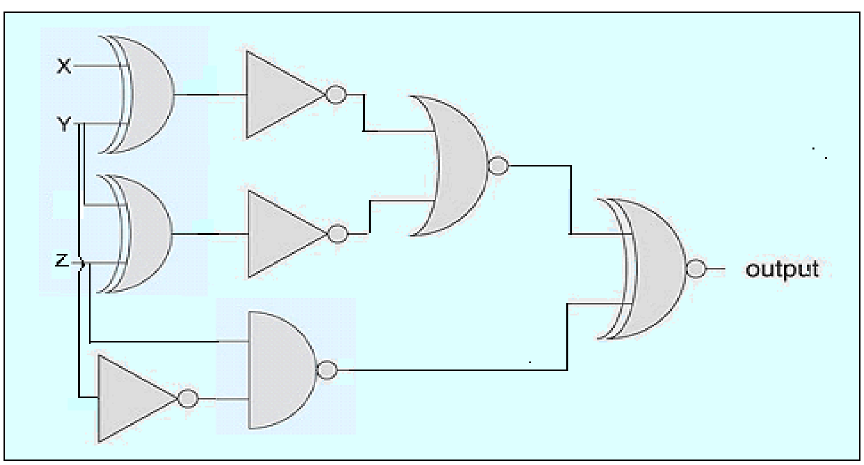
Output = $\neg (X \vee Y)$
(đầu ra chỉ có tín hiệu khi không đầu vào nào có tín hiệu)



Cổng NXOR

Output = $\neg (X \underline{\vee} Y)$
(đầu ra chỉ có tín hiệu khi không đầu vào nào có tín hiệu hoặc tất cả các đầu vào đều có tín hiệu)

Một mạch điện tử thiết kế từ những cổng logic này sẽ tương ứng với một công thức logic, và ngược lại, mỗi công thức logic tương ứng với một mạch điện tử thiết kế từ các cổng này.



Mạch điện tử trên đây tương ứng với công thức :
Output = $\neg(\neg(\neg(x \underline{\vee} y) \vee \neg(y \underline{\vee} z)) \underline{\vee} \neg (z \& \neg y))$

6. Hệ các phép toán đầy đủ

Vì các mệnh đề chỉ có thể nhận một trong hai giá trị chân lý là *T* và *F* nên số lượng các phép toán hai ngôi (khác nhau) trên mệnh đề có tất cả là $2^4 = 16$. Chúng được biểu diễn trong bảng sau:B

A	*B*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
T	*T*	*T*	*T*	*T*	*T*	*T*	*F*	*F*	*F*	*F*	*T*	*F*	*T*	*F*	*F*	*T*	*F*
T	*F*	*F*	*T*	*T*	*F*	*F*	*T*	*F*	*F*	*F*	*F*	*T*	*T*	*T*	*T*	*T*	*F*
F	*T*	*F*	*F*	*T*	*T*	*F*	*T*	*T*	*F*	*T*	*T*	*F*	*F*	*T*	*F*	*T*	*F*
F	*F*	*F*	*F*	*F*	*T*	*T*	*T*	*T*	*T*	*F*	*F*	*T*	*T*	*F*	*F*	*T*	*F*

Trong bảng trên các phép toán 1, 3, 4, 5 chính là các phép toán $\&$, $\vee$, $\supset$ và $\equiv$ tương ứng.

Nhận xét: phép kéo theo $(\supset)$ có thể được định nghĩa thông qua các phép phủ định và tuyển. Cụ thể là:

$(A \supset B) \;\; \Leftrightarrow \;\; (\neg A \vee B)$ (1)

Phép toán 14 có thể định nghĩa thông qua phép kéo theo và phủ định:
Ký hiệu nó bằng "|", ta có

$(A \,|B) \Leftrightarrow (\neg\, (A \supset B))$ (2)

và, từ (1), (2) ta thấy " |" có thể được xác định thông qua phép phủ định và tuyển:

$(A \mid B) \Leftrightarrow (\neg\, (\neg A \vee B))$

Có một câu hỏi rất tự nhiên là với một nhóm phép toán nào thì đủ để định nghĩa tất cả các phép toán còn lại trong 16 phép toán nêu trên? Định lý sau đây trả lời cho câu hỏi đó.

Định lý 1.1. *Bất cứ một phép toán nào trong số 16 phép toán nêu trong bảng trên đều có thể được cho thông qua các phép toán* $\neg$, $\&$ *và* $\vee$.

Chứng minh. Ta chứng minh định lý này bằng cách xác định từng phép toán trong số 16 phép toán trên qua các phép toán $\neg$, &, $\vee$.
Phép toán 1 chính là phép &, phép toán 3 là phép $\vee$. Phép kéo theo (4) và phép toán (14) được biểu diễn như trên kia đã nói. Phép toán (13) chính là phép tuyển nghiêm ngặt $\underline{\vee}$. Như đã biết,

A $\underline{\vee}$ B $\;\Leftrightarrow\;$ $(A \vee B)\ \&\ (\neg A \vee \neg B)$

Phép toán (5) chính là phép đồng nhất. Nó được biểu là :

(A = B) $\Leftrightarrow (\neg A \vee B)\ \&\ (\neg B \vee A)$

Phép toán thứ 8, ta ký hiệu nó bằng dấu L, được định nghĩa như sau:

$(A \text{ L } B) \;\; \Leftrightarrow \; (\neg A\ \&\ \neg B);$

phép toán 7, ta tạm ký hiệu nó bằng dấu $\rfloor$, có thể định nghĩa như sau:

$(A \rfloor B) \Leftrightarrow ((\neg A_1\ \&\ A_2) \vee (\neg A_1\ \&\ \neg A_2)).$

Chúng tôi dành phần còn lại cho bạn đọc, coi như bài tập.

Nếu cho trước một bảng chân lý thì nó còn cho phép ta xác định công thức có bảng chân lý đó.

Ví dụ: Có bảng chân lý

A_1	A_2	A_3	D
T	T	T	F
T	T	F	T
T	F	T	F
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	F

Công thức D ở đây là $D = (A_1 \& A_2 \& \neg A_3) \vee (A_1 \& \neg A_2 \& \neg A_3) \vee (\neg A_1 \& A_2 \& A_3)$.
Công thức D thu được bằng cách : Trong bảng chân trị của D chỉ sử dụng các dòng mà D có giá trị đúng (T). Tại các dòng đó, nếu biến có giá trị T thì lấy nguyên biến, nếu có giá trị F thì ấy phủ định của biến. Mỗi dòng đúng của bảng chân trị được biểu thị bằng một công thức, là hội của các biến hoặc phủ định biến chọn theo cách vừa trình bày. Các công thức tương ứng với dòng đúng được liên kết với nhau bằng dấu toán tuyển, kết quả là D.

Nhóm phép toán đủ để định nghĩa tất cả các phép toán khác được gọi là hệ các phép toán đầy đủ. Như ta thấy, định lý 1 khẳng định rằng $(\neg, \&, \vee)$ là một hệ các phép toán đầy đủ. Các cặp phép toán $(\supset, \neg)$; $(\neg, \vee)$ cũng là các hệ phép toán đầy đủ.

II. Quy luật và mâu thuẫn logic

1. Khái niệm quy luật và mâu thuẫn logic

Trong logic hai giá trị mà ta đang nghiên cứu thì một mệnh đề hoặc là đúng, hoặc là sai. Nếu mệnh đề phù hợp với thực tiễn thì nó đúng, nếu nó không phù hợp với thực tiễn thì nó sai. Nói chung, để xác định xem một mệnh đề có đúng hay không ta phải đối chiếu với thực tiễn. Thế nhưng có một số trường hợp không cần đối chiếu trực tiếp với hiện thực khách quan ta cũng có thể biết được mệnh đề là đúng hay sai. Ví dụ, ở một thời điểm nhất định thì mệnh đề *trời mưa hoặc không mưa* là mệnh đề đúng. Ta biết điều đó mà không cần phải xét xem trời mưa hay không mưa ở thời điểm đó. Nguyên nhân ở đây là mệnh đề đã nêu đúng trong cả hai trường hợp trời mưa và trời không mưa ở thời điểm đó. Mà ngoài hai trường hợp đó ra thì không còn trường hợp nào. Như vậy mệnh đề này đúng trong mọi trường hợp. Những mệnh đề đúng trong mọi trường hợp như vậy ta gọi là *mệnh đề hằng đúng,* hay *quy luật logic (tautology)*. Trái lại, ở thời điểm bất kỳ, mệnh đề *trời mưa và không mưa* sai. Nó sai trong trường hợp trên thực tế trời đang mưa, và sai cả trong trường hợp trên thực tế trời không mưa. Mà ngoài hai trường hợp đó ra thì không còn trường hợp nào khác. Nghĩa là mệnh đề này sai trong mọi trường hợp. Những mệnh đề sai trong mọi trường hợp như vậy gọi là *mệnh đề hằng sai,* hay *mâu thuẫn logic.*

Các khái niệm quy luật và mâu thuẫn logic vừa nêu có ý nghĩa rất quan trọng. Trong logic mệnh đề, một suy luận đúng và chỉ đúng khi công thức biểu thị nó là quy luật logic, và nó không thể nào đúng được khi công thức biểu thị nó là một mâu thuẫn logic. Quy luật logic cũng chính là các định lý trong các hệ tiên đề và hệ suy luận tự nhiên của logic mệnh đề mà ta sẽ nghiên cứu ở những phần sau.

2. Các phương pháp xác định quy luật và mâu thuẫn logic

a) Lập bảng chân lý

Theo định nghĩa ơ mục trên, mệnh đề là quy luật logic nếu nó đúng trong mọi trường hợp. Để ý rằng mỗi trường hợp tương ứng với một phân bố giá trị chân lý của các mệnh đề đơn. Thật vậy, chẳng hạn, với trường hợp "trời mưa" thì các mệnh đề đơn *trời mưa, đường ướt* có giá trị đúng; trong khi đó các mệnh đề *trời nắng*,… có giá trị sai. Nói cách khác, trường hợp "trời mưa" ứng với phân bố giá trị "đúng", "đúng", "sai", … cho các mệnh đề đơn *trời mưa, đường ướt, trời nắng* ... tương ứng. Như vậy mệnh đề là quy luật logic khi và chỉ khi tại tất cả các dòng trong bảng chân lý của công thức của nó đều có giá trị T (đúng). Tương tự như thế, mệnh đề là mâu thuẫn logic khi và chỉ khi tất cả các dòng trong bảng chân lý của công thức của nó đều có giá trị F (sai). Chính vì vậy lập bảng chân lý ta có thể xác định xem mệnh đề có phải là quy luật ogic hay không. Không những thế, bằng bảng chân lý ta còn có thể xác định xem mệnh đề có là mâu thuẫn logic hay không.

Cho trước một công thức. Căn cứ vào các phép toán đã biết, ta có thể lập bảng chân lý của công thức đó như sau.

Bước 1. Trước hết ta xác định xem trong công thức đã cho có bao nhiêu mệnh đề đơn khác nhau. Để ý rằng nếu một mệnh đề đơn nào đó xuất hiện nhiều lần ta cũng chỉ tính một lần. Nếu trong công thức có n mệnh đề đơn khác nhau thì bảng chân lý của công thức ấy có 2^n dòng. Mỗi dòng của bảng chứa một sự phân bố giá trị chân lý của các mệnh đề đơn trong công thức cùng với giá trị chân lý của các công thức xuất hiện khi xây dựng công thức khảo sát, và tất nhiên, cả giá trị chân lý của công thức khảo sát nữa. Ta kẻ ngay bên dưới công thức một bảng gồm 2^n dòng và mỗi mệnh đề đơn, mỗi dấu toán đều tương ứng với một cột.

Bước 2. Với mệnh đề đơn thứ nhất (thứ tự có thể chọn tùy ý) ta chia bảng thành hai phần trên dưới đều nhau. Tại cột của mệnh đề đó ở các dòng thuộc phần đầu ta ghi giá trị T (đúng), ở các dòng thuộc phần sau ghi giá trị F (sai). Với mệnh đề đơn thứ hai, hai phần của bảng lại được chia đoi. Bây giờ ta có bốn phần. Tại cột của mệnh đề này, ở các dòng phần lẻ ta ghi giá trị T, các dòng phần chẵn ghi giá trị F. Với các mệnh đề đơn còn lại làm tương tự : các phần đã có của bảng được chia thành hai phần trên dưới, ở các dòng phần lẻ ghi giá trị T, các dòng phần chẵn ghi giá trị F. Đây là bước gán giá trị cho các mệnh đề đơn. Để ý rằng trên cùng một dòng của bảng thì một mệnh đề đơn dù có thể xuất hiện nhiều lần nhưng bao giờ cũng có cùng một giá trị.

Bước 3. Ở bước này ta tính giá trị của các ô còn lại trong bảng, đây chính là giá trị của các công thức được tạo thành từ các mệnh đề đơn có mặt trong công thức ta đang khảo sát. Gia trị chân lý của các công thức tạo thành từ các mệnh đề đơn xét trong khuôn khổ công thức khảo sát được xác định tại mỗi dòng căn cứ vào giá trị các mệnh đề đơn trong dòng đó và các phép toán logic của nó. Lưu ý rằng các công thức nằm trong ngoặc đơn trong cùng phải được xác định trước, rồi sau đó căn cứ trên giá trị chân lý của chúng để xác định giá trị chân lý của các công thức có chứa chúng.

Cột giá trị được thực hiện cuối cùng là cột giá trị của công thức khảo sát. Căn cứ vào cột này có thể biết công thức có là quy luật logic hay không, nên nó được gọi là *cột đại diện*. Dấu toán tương ứng vơi cột đại diện gọi là *dấu toán chính* của công thức. Dòng có giá trị T ở cột đại diện gọi là *dòng đúng*, dòng có giá trị F ở cột đại diện gọi là *dòng sai*. Một công thức là *hằng đúng* (hay còn gọi là quy luật logic) neu trong bảng chân lý của nó, cột đại diện nó có giá trị *T* ở tất cả các hàng. Nói cách khác, công thức là hằng đúng nếu tất cả các dòng trong bảng chân lý của nó đều là dòng đúng. Nói cách khác, công thức là quy luật logic nếu bảng chân lý của nó không có dòng sai. Công thức là *hằng sai* (hay *mâu thuẫn logic*), nếu cột đại diện trong bảng chân lý của nó có giá trị *F* tại mỗi dòng, nghĩa là khi tất cả các dòng trong bảng chân lý đều là dòng sai. Hay cũng vậy, công thức là mâu thuẫn logic khi trong bảng chân lý của nó không có dòng đúng.

Ví dụ, bảng chân lý của công thức $(p \vee q)\ \&\ (\neg r)$ như sau:

Dòng đúng

Dòng sai

$(p$	$\vee$	$q)$	$\&$	$(\neg$	$r)$
T	*T*	*T*	***F***	*F*	*T*
T	*T*	*T*	***T***	*T*	*F*
T	*T*	*F*	***F***	*F*	*T*
T	*T*	*F*	***T***	*T*	*F*
F	*T*	*T*	***F***	*F*	*T*
F	*T*	*T*	***T***	*T*	*F*
F	*F*	*F*	***F***	*F*	*T*
F	*F*	*F*	***F***	*T*	*F*

Cột đại diện

Công thức có thể vừa không phải là quy luật logic, vừa không là mâu thuẫn logic. Công thức vừa xét trên đây là một công thức như vậy.

Ví dụ sau đây minh họa từng bước lập bảng chân lý của một công thức. Trong ví dụ này chúng tôi đánh số các phép toán có trong công thức theo thứ tự giảm dần độ ưu tiên để bạn đọc dễ theo dõi trình tự thực hiện chúng, các số được ghi trên đầu các dấu toán tương ứng. Cong thức khảo sát: $((p \vee q) \,\&\, (p \vee r)) \supset (\neg p \vee \neg r) \vee (\neg q \,\&\, \neg r)$.

Độ ưu tiên thực hiện các phép toán là (số càng nhỏ độ ưu tiên càng cao) :

	1		2		1		4	1		2	1		3	1		2	1	
((p	∨	q)	&	(p	∨	r))	⊃	(¬	p	∨	¬	r)	∨	(¬	q	&	¬	r)

Các phép toán có cùng độ ưu tiên có thể thực hiện theo thứ tự tuỳ ý.

Trong công thức này có ba mệnh đề đơn khác nhau là p, q và r. Vậy bảng chân lý của nó có $2^3 = 8$ dòng. Kẻ bảng và gán giá trị cho các mệnh đề đơn (coi p là mệnh đề đơn thứ nhất, q thứ hai và r thứ ba), ta được:

((p	∨	q)	&	(p	∨	r))	⊃	(¬	p	∨	¬	r)	∨	(¬	q	&	¬	r)
T		T		T		T			T			T			T			T
T		T		T		F			T			F			T			F
T		F		T		T			T			T			F			T
T		F		T		F			T			F			F			F
F		T		F		T			F			T			T			T
F		T		F		F			F			F			T			F
F		F		F		T			F			T			F			T
F		F		F		F			F			F			F			F

Thực hiện các phép toán có độ ưu tiên 1, ta được bảng sau :

((p	∨	q)	&	(p	∨	r))	⊃	(¬	p	∨	¬	r)	∨	(¬	q	&	¬	r)
T	T	T		T	T	T		F	T		F	T		F	T		F	T
T	T	T		T	T	F		F	T		T	F		F	T		T	F
T	T	F		T	T	T		F	T		F	T		T	F		F	T
T		F		T	T	F		F	T		T	F		T	F		T	F

F	T	T		F	T	T		T	F		F	T		F	T		F	T
F	T	T		F	F	F		T	F		T	F		F	T		T	F
F	F	F		F	T	T		T	F		F	T		T	F		F	T
F	F	F		F	F	F		T	F		T	F		T	F		T	F

Thực hiện các phép toán có độ ưu tiên 2, ta được :

$((p$	$\vee$	$q)$	$\&$	$(p$	$\vee$	$r))$	$\supset$	$(\neg$	p	$\vee$	$\neg$	$r)$	$\vee$	$(\neg$	q	$\&$	$\neg$	$r)$
T	T	T	T	T	T	T		F	T	F	F	T		F	T	F	F	T
T	T	T	T	T	T	F		F	T	T	T	F		F	T	F	T	F
T	T	F	T	T	T	T		F	T	F	F	T		T	F	F	F	T
T	T	F	T	T	T	F		F	T	T	T	F		T	F	T	T	F
F	T	T	T	F	T	T		T	F	T	F	T		F	T	F	F	T
F	T	T	F	F	F	F		T	F	T	T	F		F	T	F	T	F
F	F	F	F	F	T	T		T	F	T	F	T		T	F	F	F	T
F	F	F	F	F	F	F		T	F	T	T	F		T	F	T	T	F

Trong bảng trên giá trị tại mỗi cột đánh bóng đậm nhận được căn cứ vào giá trị tại hai cột đánh bóng mờ hơn hai bên nó.

Thực hiện phep toán tiếp theo, ta được :

$((p$	$\vee$	$q)$	$\&$	$(p$	$\vee$	$r))$	$\supset$	$(\neg$	p	$\vee$	$\neg$	$r)$	$\vee$	$(\neg$	q	$\&$	$\neg$	$r)$
T	T	T	T	T	T	T		F	T	F	F	T	F	F	T	F	F	T
T	T	T	T	T	T	F		F	T	T	T	F	T	F	T	F	T	F
T	T	F	T	T	T	T		F	T	F	F	T	F	T	F	F	F	T
T	T	F	T	T	T	F		F	T	T	T	F	T	T	F	T	T	F
F	T	T	T	F	T	T		T	F	T	F	T	T	F	T	F	F	T
F	T	T	F	F	F	F		T	F	T	T	F	T	F	T	F	T	F
F	F	F	F	F	T	T		T	F	T	F	T	T	T	F	F	F	T
F	F	F	F	F	F	F		T	F	T	T	F	T	T	F	T	T	F

Kết qủa mới nhận được trong cột đánh bóng đậm của bảng này căn cứ vào các cột đánh bóng mờ hơn.

Bây giờ thực hiện phép toán còn lại, tức phép toán chính, ta được :

((p	∨	q)	&	(p	∨	r))	⊃	(¬	p	∨	¬	r)	∨	(¬	q	&	¬	r)
T	T	T	T	T	T	T	F	F	T	F	F	T	F	F	T	F	F	T
T	T	T	T	T	T	F	T	F	T	T	T	F	T	F	T	F	T	F
T	T	F	T	T	T	T	F	F	T	F	F	T	F	T	F	F	F	T
T	T	F	T	T	T	F	T	F	T	T	T	F	T	T	F	T	T	F
F	T	T	T	F	T	T	T	T	F	T	F	T	T	F	T	F	F	T
F	T	T	F	F	F	F	T	T	F	T	T	F	T	F	T	F	T	F
F	F	F	F	F	T	T	T	T	F	T	F	T	T	T	F	F	F	T
F	F	F	F	F	F	F	T	T	F	T	T	F	T	T	F	T	T	F

cột đại diện – cột đánh bóng đậm, nhận được căn cứ vào các cột đánh bóng mờ - cho thấy bang có hai dòng sai và 7 dòng đúng. Như vậy công thức đã khảo sát không phải là quy luật logic, cũng không phải là mâu thuẫn logic.

Chúng ta vừa thấy việc lập bảng chân lý rất đơn giản. Với công thức nào của logic mệnh đề đều có thể lập bảng chân lý để xác định nó có phải là quy luật hay mâu thuẫn logic hay không. Bảng chân lý còn được sử dụng để giải quyết nhiều vấn đề khác.

Số dòng trong bảng chân lý của một công thức phụ thuộc vào số lượng mệnh đề đơn khác nhau tạo nên nó và tăng theo gấp đôi khi số mệnh đề đơn tăng lên một. Với công thức chứa 3 mệnh đề đơn thì số dòng là $2^3 = 8$, chứa 8 mệnh đề đơn thì số dòng đã là $2^8 = 256$! Bởi vậy, người ta phải tìm cách giảm khối lượng tính toán để có thể giải quyết được nhiều bài toán logic hơn. Ở đây ta nghiên cứu một trong những phương pháp như vậy. Đó là *phương pháp lập bảng ngữ nghĩa.*

b) Lập bảng ngữ nghĩa (bảng chân lý rút gọn)

Đây là phương pháp xác định xem công thức cho trước nào đó có phải là quy luật logic hay không bằng cách tìm xem trong bảng chân lý của nó có thể có dòng sai hay không, mặc dù không lập bảng chân lý của công thức. Nếu không có dòng sai nào trong bảng chân lý của nó thì công thức đã cho là quy luật logic. Còn nếu có thì công thức đã cho không phải là quy luật logic. Nếu như trong phương pháp lập bảng chân lý của công thức ta đi từ chỗ biết giá trị chân lý của các công thức thành phần đến việc xác lập giá trị của toàn bộ công thức, thì ở đây, ngược lại, ta đi từ chỗ biết giá trị của toàn bộ công thức đến việc xác định giá trị của các công thức thành phần của nó.

Để nghiên cứu phương pháp này ta xem xét vài ví dụ.

Ví dụ 1. Xét công thức

$((p \supset q) \,\&\, p) \supset q$

<u>Bước 1</u> Như đã nói ở trên, ta bắt đầu bằng cách giả định rằng công thức này không phải là quy luật logic. Vậy thì, theo định nghĩa, nó phải có giá trị F ở ít nhất một dòng trong bảng chân lý của nó. Ta viết giá trị F vào cột tương ứng với công thức đã cho ban đầu. Ở các bước tiếp theo ta sẽ cố gắng xác định xem một dòng như vậy có tồn tại không?

<u>Bước 2</u> Tiếp theo, theo định nghĩa phép $\supset$, công thức $((p \supset q) \,\&\, p) \supset q$ chỉ có thể có giá trị F khi các công thức $(p \supset q) \,\&\, p$ và q có các giá trị tương ứng là T và F. Vì vậy ta ghi các giá trị đó vào những vị trí tương ứng.

<u>Bước 3</u> $(p \supset q) \,\&\, p$ chỉ có thể có giá trị T khi cả $(p \supset q)$ và p đều có giá trị T. Ta ghi các giá trị đó vào chỗ của chúng. Ở bước 3 này ta còn ghi thêm giá trị F của mệnh đề đơn q đã biết ở bước 2 (nói chung ở bước thứ bất kỳ ta ghi cả giá trị của tất cả những mệnh đề đơn đã biết từ các bước trước nó).

<u>Bước 4</u> Công thức $(p \supset q)$, với giá trị T của p, chỉ có thể có giá trị T khi q có giá trị T. Ta ghi các giá trị vừa tìm ra đó vào bảng. Ta cũng ghi thêm, như đã nói ở phía trên, tất cả các giá trị chân lý đã biết ở các bước trước đó của các mệnh đề đơn.

Bước	*((p*	$\supset$	*q)*	&	*p)*	$\supset$	*q*
1						*F*	
2				*T*			*F*
3		*T*			*T*		*F*
4	*T*		*T*		*T*		*F*

Đến đây ta đã xác định được giá trị của tất cả các lần xuất hiện của các mệnh đề đơn trong công thức. Bảng đã lập xong. Dòng cuối cùng của bảng cho biết điều kiện mà một dòng trong bảng chân lý của công thức phải thỏa mãn để giá trị cua công thức trong dòng đó là sai. Ở dòng cuối cùng của bảng trên đây ta thấy mệnh đề đơn q vừa đúng lại vừa sai. Như vậy diều kiện mà ta xác định được là một diều kiện mâu thuẫn nên không dòng nào trong bảng chân lý của công thức có thể thoả mãn được. Nói cách khác, công thức là quy luật logic.

Bảng gọi là đóng nếu ở dòng cuối cùng của nó có nghịch lý, chẳng hạn như có những công thức vừa có giá trị đúng vừa có giá trị sai.

Ví dụ 2. Xét công thức

$((p \vee q) \,\&\, \neg q) \supset p$

<u>Bước 1</u> Ta giả định rằng công thức này không phải là quy luật logic. Vậy thì, theo định nghĩa, phải có giá trị F ở ít nhất một dòng trong bảng chân lý của nó. Ta viết giá trị F

vào cột tương ứng với công thức đã cho ban đầu. Ở các bước tiếp theo ta sẽ cố gắng xác định xem một dòng như vậy có tồn tại không?

Bước 2 Tiếp theo, theo định nghĩa phép $\supset$, công thức $((p \vee q) \& \neg q) \supset p$ chỉ có thể có giá trị F khi các công thức $(p \vee q) \& \neg q$ và p có các giá trị tương ứng là T và F. Vì vậy ta ghi các giá trị đó vào những vị trí tương ứng.

Bước 3 $(p \vee q) \& \neg q$ chỉ có thể có giá trị T khi cả $(p \vee q)$ và $\neg q$ đều có giá trị T. Ta ghi các giá trị đó vào chỗ của chúng. Ở bước 3 này ta còn ghi thêm giá trị F của mệnh đề đơn p đã biết ở bước 2.

Bước 4 Công thức $\neg q$ chỉ có thể có giá trị T khi q có giá trị F. Ta ghi các giá trị vừa tìm ra đó vào bảng. Ta cũng ghi thêm, như đã nói ở phía trên, tất cả các giá trị chân lý đã biết ở các bước trước đó của các mệnh đề đơn.

Bước 5 Công thức $(p \vee q)$ có thể có giá trị T trong hai trường hợp: Khi p có giá trị T và khi q có giá trị T. Để biểu thị điều này, ta phân đôi bảng, mỗi bảng con tương ứng với một trong hai trường hợp đã nêu trên:

Bước	*((p*	$\vee$	*q)*	*&*	$\neg$	*q)*	$\supset$	*p*
1							*F*	
2				*T*				*F*
3		*T*			*T*			*F*
4		*T*				*F*		*F*

Bảng con thứ nhất

5.1	*T*		*X*			*F*		*F*

Bảng con thứ hai

5.2	*X*		*T*			*F*		*F*

X trong bảng này có nghĩa là giá trị bất kỳ.

Cả hai bảng con của bảng ban đầu đều đóng, ta nói rằng bảng ban đầu là *bảng đóng.* Như đã thấy ở các bước 5.1 và 5.2, cả hai trường hợp p có giá trị T và q có giá trị T đều dẫn đến kết qua vô lý. Như vậy có nghĩa là không tồn tại bất cứ tổ hợp các giá trị chân lý nào của các mệnh đề đơn thoả mãn điều kiện để giá trị của công thức đã cho ban đầu là F. Vậy, ta có thể kết luận giả định ban đầu của ta rằng công thức $((p \vee q) \& q) \supset p$ không phải là quy luật logic đã là một giả định sai lầm. Và như vậy, $((p \vee q) \& q) \supset p$ phải là quy luật logic.

Bảng theo kiểu bảng mà ta vừa xây dựng được như trên cho một công thức nào đó gọi là *bảng ngữ nghĩa* của công thức đó.

Qua hai ví dụ trên ta thấy rằng bảng ngữ nghĩa của công thức có thể phân thành các bảng con (như trong ví dụ 2), hoặc không phân thành các bảng con (như trong ví dụ 1). Bảng ngữ nghĩa của công thức còn có thể phân chia thành các bảng con, rồi các bảng con đó, đến lượt no, cũng lại phân thành các bảng con nhỏ hơn nữa, ... Khi nào thì bảng phải phân chia ra thành các bảng con? Những suy luận nhằm tìm ra các giá trị của các công thức trong 2 ví dụ trên đây cho ta thấy rằng điều đó xảy ra khi ta từ giá trị đã xác định của một công thức cố gắng xác định giá trị của các công thức thành phần của nó. Và có phải phân chia bảng hay không là tuỳ thuộc vào dạng của công thức có các công thức con thành phần mà ta đang muốn xác định giá trị.

Chú ý: Ở ví dụ 2 trên đây, nếu ta sử dụng giá trị đã biết từ bước thứ 2 của biến p, hoặc nếu ta sử dụng giá trị đã biết từ bước số 4 của q để cùng với giá trị đã biết của công thức $p \vee q$ tiến hành xác định giá trị của các biến còn lại thì ta không cần phải phân chia bảng ra thành các bảng con. Bảng ngữ nghĩa của công thức ở ví dụ 2 khi đó có dạng như sau:

Bước	*((p*	$\vee$	*q)*	*&*	$\neg$	*q)*	$\supset$	*p*
1							*F*	
2				*T*				*F*
3		*T*			*T*			*F*
4		*T*				*F*		*F*
5	*F*	*T*				*F*		*F*
6	*F*		*T*			*F*		*F*

Ở dòng số 6 ta thấy biến q vừa có giá trị T vừa có giá trị F. Điều này chứng tỏ rằng không có dòng sai nào trong bảng chân lý của công thức đã khảo sát. Nói cách khác, công thức mà ta đã khảo sát là một quy luật logic.

Bạn đọc đã nhận thấy rằng trong ví dụ trên đây ở bước số 5 ta có thể sử dụng giá trị đã biết của biến p, mà cũng có thể sử dụng giá trị đã biết của biến q. Tổng quát hơn, khi đã biết giá trị của công thức dạng $A \otimes B$ (với $\otimes$ là một trong các phép toán mệnh đề $\supset$, $\&$, $\vee$, $\underline{\vee}$) và giá trị các công thức thành phần A và B của nó thì vấn đề đặt ra là nên chọn giá trị nào trong các giá trị đã biết đó và có thể sử dụng đồng thời cả hai giá trị đó hay không? Dựa vào bảng định nghĩa của các phép toán mệnh đề ta có câu trả lời như sau cho câu hỏi này:

* Nếu việc sử dụng cả hai giá trị của A và B không mâu thuẫn với giá trị đã biết của $A \otimes B$ thì ta dùng cả hai giá trị đó.

* Nếu việc sử dụng cả hai giá trị của A và B mâu thuẫn với giá trị đã biết của công thức $A \otimes B$ thì ta dùng một trong hai giá trị đó. Và phải sử dụng giá trị của thành phần nào mà nhờ đó cùng với giá trị đã biết của $A \otimes B$ có thể xác định được giá trị của thành phần kia. Nếu mới chỉ biết giá trị của một trong hai thàng phần thì ta sử dụng nó kết hợp với giá trị của toàn bộ công thức để xác định (nếu được) giá trị của thành phần còn lại.

* Ta cũng có thể coi như giá trị đã biết của A và B như chưa biết và không sử dụng giá trị nào trong số chúng (như trong ví dụ 2 trên đây).

Liên kết những điều đã trình bày trên đây với định nghĩa các phép toán logic, ta rút ra các quy tắc chung sau đây về cách xây dựng *bảng ngữ nghĩa* của công thức :

1. $$\left.\begin{array}{cc} \neg & A \\ T & \end{array}\right\} \Rightarrow A = F$$

2. $$\left.\begin{array}{cc} \neg & A \\ F & \end{array}\right\} \Rightarrow A = T$$

3. $$\left.\begin{array}{ccc} A & \& & B \\ & T & \end{array}\right\} \Rightarrow A = T, B = T$$

4. $$\left.\begin{array}{ccc} A & \vee & B \\ & F & \end{array}\right\} \Rightarrow A = F, B = F$$

5. $$\left.\begin{array}{ccc} A & \supset & B \\ & F & \end{array}\right\} \Rightarrow A = T, B = F$$

6. $$\left.\begin{array}{ccc} A & \& & B \\ F & F & \end{array}\right\} \Rightarrow B = F$$

7. $$\left.\begin{array}{ccc} A & \& & B \\ & F & F \end{array}\right\} \Rightarrow A = F$$

8. $$\left.\begin{array}{ccc} A & \vee & B \\ F & F & \end{array}\right\} \Rightarrow B = T$$

9. $$\left.\begin{array}{ccc} A & \vee & B \\ & F & F \end{array}\right\} \Rightarrow A = T$$

10. $$\left.\begin{array}{ccc} A & \supset & B \\ T & T & \end{array}\right\} \Rightarrow B = T$$

11. $$\left.\begin{array}{ccc} A & \supset & B \\ & T & F \end{array}\right\} \Rightarrow A = F$$

12. $$\left.\begin{array}{ccc} A & \& & B \\ & F & \end{array}\right\} \Rightarrow \begin{array}{l} \text{a) } A = F, B = X \\ \text{b) } B = F, A = X \end{array}$$

13. $$\left.\begin{array}{ccc} A & \vee & B \\ & T & \end{array}\right\} \Rightarrow \begin{array}{l} \text{a) } A = T, B = X \\ \text{b) } B = T, A = X \end{array}$$

14. $$\left.\begin{array}{ccc} A & \supset & B \\ & T & \end{array}\right\} \Rightarrow \begin{array}{l} \text{a) } A = F, B = X \\ \text{b) } B = T, A = X \end{array}$$

Các quy tắc từ số 1 đến số 5 tạo thành nhóm quy tắc I, nhóm II gồm các quy tắc từ số 6 đến số 11, nhóm III gồm các quy tắc còn lại.

Khi lập bảng ngữ nghĩa của công thức, mặc dù không bắt buộc, nhưng sẽ thuận tiện hơn nếu trước hết áp dụng các quy tắc nhóm I, nếu các quy tắc đó không áp dụng được mới áp dụng các quy tắc nhóm II, và chỉ khi không thể áp dụng các quy tắc thuộc hai nhóm đầu mới áp dụng các quy tắc nhom III.

Bây giờ, để chặt chẽ, ta đưa ra một số định nghĩa.

Định nghĩa 1. Một *bảng con tận cùng* (là bảng không có bảng con, bảng mẹ của bảng con này có thể cũng là bảng con của một bảng khác) trong bảng ngữ nghĩa của công thức bất kỳ được gọi là *đóng* nếu như nó chứa dòng trong đó có một (hoặc nhiều) nghịch lý (chẳng hạn như tồn tại mệnh đề đơn vừa có giá trị *T* vừa có giá trị *F,* hoặc công thức dạng A & B có giá trị *F*, trong khi cả A va B đề có giá trị *T, ...*) . *Bảng mẹ* được gọi là *đóng*, nếu như tất cả các bảng con của nó đều đóng.

Bảng ngữ nghĩa của công thức bao giờ cũng hoặc là một *bảng con tận cùng* hoặc là một *bảng mẹ*, nên định nghĩa 1 trên đây cũng cho ta khái niệm về *bảng ngữ nghĩa đóng* của công thức.

Dễ dàng chứng minh được rằng một công thức là quy luật logic bao giờ cũng có các *bảng ngữ nghĩa đóng* và chỉ các quy luật logic mới có bảng như thế.

Vì vậy, nếu sử dụng thuật ngữ vừa đưa ra này thì ta có:

Định lý 1. Công thức A là quy luật logic khi và chỉ khi A có ít nhất một *bảng ngữ nghĩa đóng*.

So sánh việc xây dựng *bảng ngữ nghĩa* với việc xây dựng bảng chân ly của một công thức để xác định xem công thức có phải là quy luật logic hay không thì ta thấy xây dựng *bảng ngữ nghĩa* đỡ phải tính toán hơn rất nhiều.

Ta xét thêm một ví dụ ứng dụng phương pháp lập bảng ngữ nghĩa.

Ví dụ 3   Theo truyền thuyết, người đốt thư viện Alecxandre là Omahr suy luận như sau: "Nếu sách của các ngài đúng với kinh Koran thì sách của các ngài thừa. Nếu sách của các ngài không đúng với kinh Koran thì sách của các ngài có hại. Sách thừa hoặc có hại thì cần phải đốt bỏ. Vậy sách của các ngài cần phải đốt bỏ" . Hãy xét xem suy luận đó của Omahr có đúng không.

Giải:

Suy luận của Omar có thể viết dưới dạng công thức thành:

$$(((p \supset q) \,\&\, (\neg p \supset r)) \,\&\, ((q \vee r) \supset s)) \supset s$$

Nếu công thức vừa dẫn trên đây (ta gọi là công thức Omar) là quy luật logic thì suy luận của Omar đúng. Ngược lại thì suy luận của Omar là sai. Ta lập bảng ngữ nghĩa của công thức Omar.

(((*p*	$\supset$	*q*)	&	($\neg$	*p*	$\supset$	*r*))	&	((*q*	$\vee$	*r*)	$\supset$	*s*))	$\supset$	*s*
														F	
								T							*F*
			T									*T*			*F*
	T					*T*						*T*	*F*		*F*
	T					*T*				*F*			*F*		*F*
	T					*T*			*F*		*F*		*F*		*F*
	T	*F*				*T*	*F*		*F*		*F*		*F*		*F*
F		*F*		*F*			*F*		*F*		*F*		*F*		*F*
F		*F*			***T***		*F*		*F*		*F*		*F*		*F*

Các dấu mũi tên trong bảng cho ta biết các giá trị mà mũi tên chỉ nhận được từ đâu.

Ở dòng cuối cùng của bảng ta thấy mệnh đề đơn *p* vừa có giá trị *F*,vừa có giá trị *T* (các giá trị đó được in đậm ở trong bảng). Vậy bảng đóng, nghĩa là suy luận của Omar đúng.

III. Biến đổi tương đương

Ta cũng có thể phát hiện ra quy luật logic bằng cách biến đổi tương đương công thức về thành một công thức khác mà ta đã biết rõ có là quy luật logic hay không. Ngoài việc ứng dụng để xác định quy luật logic, biến đổi tương đương công thức còn giúp phát hiện các công thức tương đương với nhau. Như đã biết, các công thức tương đương với nhau là các công thức có giá trị logic như nhau với bất cứ phân bố giá trị nào của các mệnh đề đơn thành phần của chúng. Trong phần này ta nghiên cứu phương pháp biến đổi của đại số boole.

1. Các ký hiệu và hằng đẳng thức

Trong đại số boole, các phép toán logic được ký hiệu như sau:

A & B ký hiệu là $A\,.\,B$, (hoặc AB) Gọi là phép nhân logic;

A $\vee$ B ký hiệu là $A + B$ Gọi là phép cộng logic;

$\neg$ A ký hiệu là $\overline{A}$ Gọi là phép bù logic;

Quy luật logic ký hiệu là *1*;

Mâu thuẫn logic ký hiệu là *0;*

từ đây A $\supset$ B được viết thành $\overline{A} + B$.

Dễ thấy rằng:

1.	$A + A = A$;	Luật đồng nhất, luật nuốt
2.	$A . A = A$;	Luật đồng nhất, luật nuốt
3.	$A + B = B + A$	Tính chất giao hoán của phép cộng
4.	$A + (B + C) = (A+ B) + C$	Tính chất kết hợp của phép cộng
5.	$A . B = B . A$	Tính chất giao hoán của phép nhân
6.	$A . (B . C) = (A . B) . C$	Tính chất kết hợp của phép nhân
7.	$A . (B + C) = A.B + A.C$	Tính phân phối của phép cộng đối với phép nhân
8.	$A + (B . C) = (A + B) . (A + C)$	Tính phân phối của phép nhân đối với phép cộng
9.	$\overline{A} + A = 1$;	Định nghĩa 1
10.	$\overline{A} . A = 0$;	Định nghĩa 0
11.	$\overline{\overline{A}} = A$	Luật hoàn nguyên
12.	$\overline{A+B} = \overline{A}.\overline{B}$	Luật De Moorgan
13.	$\overline{A.B} = \overline{A} + \overline{B}$	Luật De Moorgan
14.	$A.(A + B) = A$	Luật giản lược
15.	$A + (A.B) = A$	Luật giản lược

Trong bất kỳ một đẳng thức logic nào, nếu thay một biểu thức (tức là một công thức) nào đó bằng một biểu thức khác tương đương với nó thì đẳng thức vẫn xác lập, túc là vẫn đúng.

2. Các ví dụ

(1) $A + 0 = A$

Chứng minh

$A + 0 = A + (A.\overline{A}) = A$

(2) $A.0 = A$

Chứng minh

$A.0 = A.(A.\overline{A}) = (A.A).\overline{A} = A.\overline{A} = 0$

(3) $A + 1 = 1$

Chứng minh

$A + 1 = A + (A + \overline{A}) = (A + A) + \overline{A}) = A + \overline{A} = 1$

(4) $A.1 = A$

Chứng minh

$A.1 = A.(A+\overline{A}) = A$

(5) $A . B + \overline{A} . B = B$

Chứng minh:

$$A . B + \overline{A} . B = B(\overline{A} + A) = B . 1 = B$$

(6) $A + \overline{A} . B = A + B$

Chứng minh:

$$A + \overline{A} . B = (A + \overline{A}) . (A + B) = 1 . (A + B) = A + B$$

(7) $((A \& B) \vee (A \& \neg B)) \supset A$

$= \overline{A.B + A.\overline{B}} + A$

$= \overline{A(B + \overline{B})} + A$

$= \overline{A.1} + A = \overline{A} + A = 1$, là quy luật logic.

(8) $((\neg A \supset B) \& (\neg A \supset \neg B)) \supset A$

$= ((\overline{\overline{A}} + B) . (\overline{\overline{A}} + \overline{B})) \supset A$

$= ((A + B).(A + \overline{B})) \supset A$

$= \overline{((A + B).(A + \overline{B})} + A$

$= \overline{A + B} + \overline{A + \overline{B}} + A$

$= \overline{A} . \overline{B} + \overline{A}.B + A$

$= \overline{A}(\overline{B} + B) + A$

$= \overline{A} . 1 + A = \overline{A} + A = 1$

IV. Hệ tiên đề của logic mệnh đề

Phương pháp lập bảng chân lý cho phép giải quyết hàng loạt vấn đề cơ bản của logic mệnh đề, ví dụ như xét xem công thức có phải là quy luật logic hay không, hai công thức cho trước có tương đương hay không, công thức cho trước có phải là mâu thuẫn logic hay không v.v. Thế nhưng có một số vấn đề phức tạp hơn của logic mệnh đề rất khó hoặc không thể giải quyết được bằng phương pháp lập bảng chân lý. Bởi vậy, chúng ta xét phương pháp các lý thuyết hình thức.

1. Lý thuyết hình thức hóa (lý thuyết tiên đề hóa)

Để cho đơn giản, chúng tôi chọn hệ tiên đề có ngôn ngữ không chứa các ký hiệu $\vee$ và & mà E. Mendelson đã nêu trong cuốn "Introduction To Mathematical Logic" (hệ thống này được Hilbert nghiên cứu đầu tiên). Khái niệm lý thuyết hình thức hóa dưới đây và khái niệm hệ tiên đề trong logic vị từ ở chương sau cũng được trình bày dựa theo sách này của E. Mendelson.

Một lý thuyết hình thức hóa (lý thuyết tiên đề hóa) S được coi là xác định nếu như các điều kiện sau đây được thỏa mãn:

1) Có một tập đếm được các ký tự, gọi là các ký tự của lý thuyết S. Dãy hữu hạn các ký tự của lý thuyết S gọi là biểu thức của lý thuyết S.
2) Xác định một tập con của tập các biểu thức lý thuyết S. Tập đó gọi là tập các công thức của lý thuyết S.
3) Xác định một tập các công thức nào đó, gọi là các tiên đề.
4) Xác định một tập $R_1, R_2, ..., R_n$ các quan hệ giữa các công thức của lý thuyết S, gọi là các quy tắc suy diễn. Với mỗi quy tắc R_i tồn tại một số nguyên dương j sao cho với mỗi tập gồm j công thức và một công thức A bao giờ cũng có một thuật toán xác định được xem j công thức đó với công thức A có quan hệ R_i hay không. Nếu giữa chúng có quan hệ R_i thì người ta gọi A là hệ quả trực tiếp của j công thức đã cho đó theo quy tắc R_i.

Về mặt nội dung thì tiên đề của một lý thuyết là một khẳng định cơ sở của lý thuyết đó. Tiên đề không cần chứng minh và không thể chứng minh được trong khuôn khổ lý thuyết đó. Ví dụ, khẳng định "Qua một điểm nằm ngoài một đường thẳng có thể kẻ được một và chỉ một đường thẳng song song với đường thẳng đã cho" là tiên đề số 5 nổi tiếng trong hình học Euclid. Tiên đề này được công nhận trong hình học Euclid và không thể chứng minh hay bác bỏ trong khuôn khổ hình học đó.

Chuỗi suy luận trong lý thuyết S là một dãy hữu hạn các công thức $Q_1, Q_2, \ldots, Q_n$, trong đó với mỗi $1 \leq i \leq n$, Q_i hoặc là một tiên đề, hoặc là một giả định (hay giả thiết) hoặc là hệ quả trực tiếp của một số công thức nào đó đứng trước nó trong dãy $Q_1, Q_2, \ldots, Q_n$ theo một quy tắc suy diễn của lý thuyết S.

Công thức Q được gọi là một hệ quả của tập các công thức Γ trong lý thuyết S, nếu tồn tại một chuỗi suy luận với công thức cuối cùng là Q còn các giả định hay giả thiết đều là phần tử của Γ. Nói cách khác, Công thức Q được gọi là một hệ quả của tập các công thức Γ trong lý thuyết S, nếu tồn tại một dãy hữu hạn các công thức $Q_1, Q_2, \ldots, Q_n$ của lý thuyết S, trong đó Q_n chính là Q và mỗi công thức Q_i trong dãy đó hoặc là tiên đề của S , hoặc là công thức từ tập Γ, hoặc là hệ quả trực tiếp của một số công thức đứng trước nó trong dãy trên theo một quy tắc suy diễn nào đó của lý thuyết S. Dãy các công thức như vậy được gọi là phép suy diễn Q từ Γ.

Phép chứng minh trong lý thuyết S là một dãy hữu hạn các công thức $Q_1, Q_2, \ldots, Q_n$, trong đó với mỗi $1 \leq i \leq n$, Q_i hoặc là một tiên đề, hoặc là hệ quả trực tiếp của một số công thức nào đó đứng trước nó trong dãy $Q_1, Q_2, \ldots, Q_n$ theo một quy tắc suy diễn của lý thuyết S.

Công thức Q được gọi là định lý của lý thuyết S, nếu tồn tại một phép chứng minh $Q_1, Q_2, \ldots, Q_n$, trong đó Q_n chính là Q.

Dễ thấy rằng nếu Q là một hệ quả của tập các công thức Γ, nhưng Γ = ∅ thì ta có một phép chứng minh. Trong trường hợp đó ta có phép chứng minh của Q. Nếu có một phép suy diễn Q từ tập công công thức $\Gamma = \{B_1, B_2, \ldots, B_k\}$ thì B_i với $1 \le i \le k$ được gọi là các tiền đề, hoặc giả thiết. Người ta ký hiệu "Q là hệ quả của Γ" là $\Gamma \vdash Q$. Nếu $\Gamma = \{B_1, B_2, \ldots, B_k\}$ thì, thay vì viết $\{B_1, B_2, \ldots, B_k\} \vdash Q$, ta viết $B_1, B_2, \ldots, B_k \vdash Q$.

Người ta còn có thể mở rộng khái niệm phép suy diễn từ tập công thức cho trước bằng cách không đòi hỏi tập công thức đó phải hữu hạn. Trong trường hợp của chúng ta điều đó có nghĩa là tập Γ có thể vô hạn, $\Gamma = \{B_1, B_2, \ldots, B_k, \ldots\}$. Nếu tồn tại một phép chứng minh của Q thì, hiển nhiên, theo định nghĩa định lý của lý thuyết *S*, Q là định lý của *S*. Nếu Q là định lý thì ta viết $\vdash Q$, như vậy ta có hai ký hiệu tương đương cho định lý Q là $\vdash Q$ và $\varnothing \vdash Q$.

Từ định nghĩa nêu trên đây dễ nhận thấy các tính chất:

(1) Nếu $\Gamma \subseteq \Delta$ và $\Gamma \vdash Q$ thì $\Delta \vdash Q$;

(2) $\Gamma \vdash Q$ khi và chỉ khi tồn tại một tập Γ_1 hữu hạn sao cho $\Gamma_1 \subseteq \Gamma$ và $\Gamma_1 \vdash Q$.

Ở đây $\Gamma \vdash Q$ được hiểu theo nghĩa rộng, Γ có thể có vô số phần tử.

(3) Nếu $\Gamma_1 \vdash Q$ và $\Gamma \vdash B$ với mọi B từ tập Γ_1 thì $\Gamma \vdash Q$.

Tính chất (1) nói lên rằng một tập nhiều giả thiết hơn thì có nhiều hệ quả hơn. Tính chất (2) xuất phát từ tính chất (1) và nhận xét sau đây: Trong bất cứ phép rút ra hệ qủa từ một tập công thức cho trước số thì số công thức bao giờ cũng hữu hạn. Ý nghĩa của tính chất (3) cũng dễ hiểu: Nếu mỗi công thức của tập Γ_1 đều là hệ quả của tập công thức Γ thì mọi hệ quả của tập Γ_1 cũng là hệ quả của tập công thức Γ.

2. Lý thuyết S (Hệ tiên đề S)

Bây giờ chúng ta khảo sát lý thuyết hình thức S của logic mệnh đề. Lý thuyết S được xác định qua 4 phần sau đây:

(1) Γ, ⊃, (,) và các chữ cái la tinh in hoa có hoặc không có chỉ số: A, B, C, $A_1, A_2, \ldots, B_1, B_2, \ldots, C_1, C_2, \ldots$ là các ký tự của S. A, B, C, $A_1, A_2, \ldots, B_1, B_2, \ldots, C_1, C_2, \ldots$ ở đây là các mệnh đề đơn; ¬, ⊃ là các phép toán logic.

(2) *a)* Tất cả các mệnh đề đơn đều là công thức của S.

b) Nếu A, B là công thức của S thì (A), (B), ¬ A, A ⊃ B là các công thức của S.

c) Ngoài ra S không có công thức nào khác.

(3) Cho A, B, C là các công thức bất kỳ của hệ S. Khi đó các công thức sau đây là tiên đề của hệ S:

(A_1) $(A \supset (B \supset A))$;

(A_2) $((A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$;

(A_3) $(\neg B \supset \neg A) \supset ((\neg B \supset A) \supset B))$

(4) Quy tắc suy diễn duy nhất của S là Modus Ponens:

$$MP \quad \frac{A \supset B,\ A}{B}$$

Nhận xét: *Vì A, B, C trong các tiên đề trên có thể là các công thức bất kỳ nên S chứa một số lượng vô hạn các tiên đề.*

(A_3) có thể được thay thế bởi (A_3') với

(A_3'): $(\neg B \supset \neg A) \supset (A \supset B)$.

Các phép toán $\neg$ và $\supset$ được định nghĩa bởi hệ các tiên đề khung (A_1), (A_2), (A_3). Còn các phép toán logic khác có thể định nghĩa như sau:

$A \vee B \Leftrightarrow \neg A \supset B;$

$A \,\&\, B \Leftrightarrow \neg (A \supset \neg B);$

$A \equiv B \Leftrightarrow (A \supset B) \,\&\, (B \supset A)$

Ví dụ sau đây cho thấy phép chứng minh được thực hiện trong lý thuyết S nêu trên như thế nào:

Chứng minh rằng $A \supset A$.

Chứng minh:

1. $(A \supset ((A \supset A) \supset A))$	*tiên đề,*
2. $(A \supset ((A \supset A) \supset A)) \supset ((A \supset (A \supset A)) \supset (A \supset A))$	*tiên đề*
3. $(A \supset (A \supset A)) \supset (A \supset A))$	*từ 1, 2 và MP,*
4. $A \supset (A \supset A)$	*tiên đề*
5. $A \supset A$	*3, 4 và MP*

Ví dụ trên đây cho thấy thực hiện phép chứng minh trong hệ tiên đề là công việc rất khó khăn.

Định lý suy diễn sau đây là một tính chất rất quan trọng của lý thuyết S (các định lý về hệ S, gọi là các *siêu định lý*, hay *định lý meta*, nhưng để cho đơn giản, trong trường hợp không sợ bị nhầm lẫn với các định lý của chính hệ S, ta sẽ gọi là *định lý*). Định lý này giúp cho ta thực hiện các phép chứng minh trong S dễ dàng hơn. Chúng tôi không dẫn ra đây phép chứng minh siêu định lý này.

***Định lý 1.2**.* (Định lý suy diễn, được Erbran phát biểu và chứng minh năm 1930) Nếu Γ là tập các công thức, A và B là các công thức và $\Gamma, A \vdash B$ thì $\Gamma \vdash A \supset B$.

Hệ quả 1.3.

(1) $A \supset B, B \supset C \vdash A \supset C.$

(2) $A \supset (B \supset C),\ B \vdash A \supset C.$

Chứng minh (1):

1. $A \supset B$ *giả thiết*

2. $B \supset C$ *giả thiết,*

3. A *giả thiết,*

4. B *1, 3, MP*

5. C *2, 4, MP*

Như vậy $A \supset B,\ B \supset C,\ A \vdash C$. Từ đây, theo siêu định lý 1, ta có :
$A \supset B,\ B \supset C \vdash A \supset C$.

Chứng minh (2:)

1. $A \supset (B \supset C)$ *giả thiết,*

2. B *giả thiết,*

3. A *giả thiết,*

4. $B \supset C$ *1, 3, MP*

5. C *2, 4, MP*

Như vậy $A \supset (B \supset C), B, A \vdash C$. Từ đây, theo siêu định lý 1, ta có :
$A \supset (B \supset C), B \vdash A \supset C$.

3. *Các hệ tiên đề khác của logic mệnh đề*

Thay các tiên đề A_1, A_2, A_3 hoặc một số trong số đó bằng những tiên đề khác của logic mệnh đề, tương đương với hệ S . Ở đây chúng tôi chỉ xem xét hệ tiên đề, trong đó các tiên đề, ngoài các phép toán $\supset$ và $\neg$ như ở hệ S , còn có thể chứa các phép toán & và $\vee$. Một hệ tiên đề như vậy rất tiện lợi khi sử dụng, vì các phép toán & và $\vee$ trở thành các phép toán ban đầu, chứ không phải chỉ là các phép toán được định nghĩa thông qua $\supset$ và $\neg$.
Sau đây là hệ tiên đề đó, ta ký hiệu nó là CL (viết tắt chữ Classical Logic) – Hệ logic cổ điển.
Các phép toán cơ sở là $\supset$, $\neg$, & và $\vee$.
Với mọi công thức A, B, C, những công thức sau đây là tiên đề:

(C1) $A \supset (B \supset A)$;

(C2) $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$;

(C3) $(A \,\&\, B) \supset A$

(C4) $(A \,\&\, B) \supset B$;

(C5) $A \supset (B \supset (A \,\&\, B))$;

(C6) $A \supset (A \vee B)$;

(C7) $B \supset (A \vee B)$;

(C8) $(A \supset C) \supset ((B \supset C) \supset ((A \vee B) \supset C))$;

(C9) $(A \supset B) \supset ((A \supset \neg B) \supset \neg A)$;

(C10) $\neg\neg A \supset A$

Quy tắc $\qquad MP \quad \dfrac{A \supset B, A}{B}$

So sánh hai hệ S và CL ta thấy rằng tất cả các tiên đề và quy tắc của S đều là các tiên đề và quy tắc của hệ CL. Như vậy, S là hệ con của CL. Nhưng mặt khác, sử dụng các định nghĩa & và $\vee$ nhờ $\supset$ và $\neg$ ta cũng dễ dàng chứng minh được rằng tất cả các tiên đề của CL là tiên đề hoặc định lý của hệ S. Như vậy S và CL là tương đương.

Ví dụ chứng minh trong hệ CL định lý

$\vdash (p \,\&\, (q \vee r)) \supset ((p \,\&\, q) \vee (p \,\&\, r))$

Vì S là hệ con của CL nên siêu định lý 1 và hệ quả 1 của nó cũng đúng đối với CL. Chúng ta dùng các khẳng định đó để chứng minh định lý đã nêu.

1. $p \,\&\, (q \vee r)$	giả thiết
2. $(p \,\&\, (q \vee r)) \supset p$	tiên đề C3
3. $(p \,\&\, (q \vee r)) \supset (q \vee r)$	tiên đề C4
4. p	1, 2, MP
5. $q \vee r$	1, 3, MP
6. $p \supset (q \supset (p \,\&\, q))$	tiên đề C5
7. $p \supset (r \supset (p \,\&\, r))$	tiên đề C5
8. $(q \supset (p \,\&\, q))$	4, 6, MP
9. $(r \supset (p \,\&\, r))$	4, 7, MP
10. $(p \,\&\, q) \supset ((p \,\&\, q) \vee (p \,\&\, r))$	tiên đề C6
11. $(p \,\&\, r) \supset ((p \,\&\, q) \vee (p \,\&\, r))$	tiên đề C7
12. $(q \supset (p \,\&\, q)) \supset (((p \,\&\, q) \supset ((p \,\&\, q) \vee (p \,\&\, r))) \supset (q \supset ((p \,\&\, q) \vee (p \,\&\, r))))$	hệ quả 1 của định lý suy diễn
13. $(r \supset (p \,\&\, r)) \supset (((p \,\&\, r) \supset ((p \,\&\, q) \vee (p \,\&\, r))) \supset (r \supset ((p \,\&\, q) \vee (p \,\&\, r))))$	hệ quả 1 của định lý suy diễn
14. $((p \,\&\, q) \supset ((p \,\&\, q) \vee (p \,\&\, r))) \supset (q \supset ((p \,\&\, q) \vee (p \,\&\, r)))$	8, 12, MP
15. $q \supset ((p \,\&\, q) \vee (p \,\&\, r))$	10, 14, MP
16. $((p \,\&\, r) \supset ((p \,\&\, q) \vee (p \,\&\, r))) \supset (r \supset ((p \,\&\, q) \vee (p \,\&\, r)))$	9, 13, MP
17. $r \supset ((p \,\&\, q) \vee (p \,\&\, r))$	11, 15, MP
18. $(q \supset ((p \,\&\, q) \vee (p \,\&\, r))) \supset ((r \supset ((p \,\&\, q) \vee (p \,\&\, r))) \supset ((q \vee r) \supset ((p \,\&\, q) \vee (p \,\&\, r))))$	tiên đề C8
19. $(r \supset ((p \,\&\, q) \vee (p \,\&\, r))) \supset ((q \vee r) \supset ((p \,\&\, q) \vee (p \,\&\, r)))$	15, 18, MP
20. $(q \vee r) \supset ((p \,\&\, q) \vee (p \,\&\, r))$	17, 19, MP

21. (p & q) $\vee$ (p & r)　　　　　　　　　5, 20, MP

Như vậy : (p & (q $\vee$ r)) $\vdash$ (p & q) $\vee$ (p & r). Từ đây, theo định lý suy diễn, ta được điều phải chứng minh $\vdash$ (p & (q $\vee$ r)) $\supset$((p & q) $\vee$ (p & r)).

V. Hệ suy luận tự nhiên của logic mệnh đề

1. Các quy tắc

Hệ suy luận tự nhiên không có các tiên đề, mà chỉ bao gồm các quy tắc suy luận. Các quy tắc thường được chia ra làm hai loại: Loại đưa một phép toán vào công thức (gọi tắt là Quy tắc nhập), và loại khử bỏ phép toán từ công thức (gọi tắt là loại khử). Các quy tắc đó như sau:

Với *A*, *B* là các công thức bất kỳ :

Quy tắc nhập & (ký hiệu $\&_i$)　$\dfrac{A, B}{A \& B}$

Quy tắc khử & (ký hiệu $\&_e$)　$\dfrac{A \& B}{A}$;　$\dfrac{A \& B}{B}$

Quy tắc nhập $\vee$ (ký hiệu $\vee_i$)　$\dfrac{A}{A \vee B}$;　$\dfrac{B}{A \vee B}$

Quy tắc khử $\vee$ (ký hiệu $\vee_e$)　$\dfrac{A \vee B, \neg A}{B}$;　$\dfrac{A \vee B, \neg B}{A}$

Quy tắc nhập $\neg$ (ký hiệu $\neg_i$)　$\dfrac{B, \neg B}{\neg A}$　(*)

Quy tắc khử $\neg$ (ký hiệu $\neg_e$)　$\dfrac{\neg\neg A}{A}$

Quy tắc nhập $\supset$ (ký hiệu $\supset_i$)　$\dfrac{B}{A \supset B}$　(*)

Quy tắc khử $\supset$ (ký hiệu $\supset_e$)　$\dfrac{A \supset B, A}{B}$

2. Chuỗi suy diễn và phép chứng minh

Cũng như với hệ tiên đề, với hệ suy luận tự nhiên cũng có các chuỗi suy diễn và phép chứng minh.

Chuỗi suy diễn trong hệ suy luận tự nhiên là một dãy các công thức kế tiếp nhau, trong đó mỗi công thức hoặc là một giả thiết, hoặc là một giả định, hoặc được rút ra từ các công thức đứng trước nó trong dãy theo một trong các quy tắc của hệ suy luận tự nhiên.

Ví dụ chuỗi suy diễn:

1^+. $A \supset B$

2^+. $\neg B \vee C$

3^+. $\neg C$

4. $\neg B$ 2, 3, $\vee_e$

5^*. A

6. B 1, 5, $\supset_e$

(Trong các chuỗi suy diễn và phép chứng minh từ đây về sau dấu * phía trên bên phải của số thứ tự công thức nói rằng công thức đó là một giả định, dấu + ở vị trí như vậy cho biết công thức là một giả thiết).

Với quy tắc (*), để cho đơn giản khi xây dựng phép chứng minh hay rút hệ quả từ một tập công thức cho trước có thể đòi hỏi B là giả thiết sau cùng trong dãy công thức của suy luận đó và B chưa bị loại bỏ khỏi suy luận đó. Khái niệm công thức bị loại bỏ khỏi suy luận (khỏi chuỗi suy diễn) được định nghĩa như sau:

Nếu giả thiết *B* là công thức thứ *i* trong dãy. Ở bước *n* ta áp dụng công thức *B* với một hoặc hai công thức khác theo quy tắc $\dfrac{B, \neg B}{\neg A}$ hoặc $\dfrac{B}{A \supset B}$

thì tất cả các công thức kể từ thứ i đến $n-1$ đều gọi là bị loại khỏi dãy công thức của chuỗi suy diễn.

Để khỏi nhầm lẫn, người ta dùng dấu ngoặc vuông để tách riêng các công thức bị loại. Ví dụ, trong suy luận sau đây:

1^+. A

2^+. $A \supset B$

3. B 1, 2, $\supset_e$

4. $(A \supset B) \supset B$ 2, 3, $\supset_i$

5. $A \supset ((A \supset B) \supset B)$ 1, 4, $\supset_i$

(Ta dùng dấu + ở góc trên bên phải của số thứ tự công thức để ký hiệu rằng công thức đó là một giả thiết).

Ở bước 4, áp dụng quy tắc $\supset_i$ đối với các công thức 2 và 3, ta được công thức số 4. Đồng thời với việc áp dụng quy tắc như vậy, các công thức 2 và 3 bị loại bỏ (ta cho chúng vào trong dấu ngoặc vuông). Tiếp theo, ở bước 5, áp dụng quy tắc $\supset_i$ đối với các công thức 1 và 4, ta được công thức số 5. Khi đó các công thức 1 và 4 bị loại khỏi chuỗi suy diễn (các công thức 2 và 3 đã bị loại từ trước). các công thức đã bị loại bỏ ở bước thứ *i* thì không thể được sử dụng ở các bước sau đó nữa.

Một *phép chứng minh* công thức *A* trong hệ suy luận tự nhiên này là một dãy các công thức của hệ, trong đó mỗi công thức hoặc là một giả thiết (giả định), hoặc nhận được từ một số công thức đứng trước nó trong dãy theo một trong các quy tắc của hệ. Ngoài ra, công thức cuối cùng trong dãy là *A* , và bất cứ một giả thiết nào trong dãy cũng đều đã được loại bỏ khỏi chuỗi suy diễn ở một bước nào đó.

Công thức A được gọi là *hệ quả* của tập các công thức (tập giả thiết) Γ khi và chỉ khi tồn tại một dãy các công thức của hệ mà mỗi công thức trong số đó hoặc là phần tử của Γ, hoặc là nhận được từ các công thức đứng trước trong dãy khi áp dụng một trong số các quy tắc của hệ. Ngoài ra, công thức A là công thức cuối cùng của dãy đó, dãy là hữu hạn.

Nếu tồn tại một phép chứng minh của công thức A trong hệ này thì A được gọi là *định lý* của hệ.

Một vài ví dụ về phép chứng minh trong hệ suy luận tự nhiên.

Chứng minh các tiên đề của hệ S :

a) $A \supset (B \supset A)$

1^+.	A	
2^+.	B	
3.	A	1
4.	$A \supset B$	2, 3, $\supset_i$
5.	$A \supset (B \supset A)$	1, 4, $\supset_i$

b) $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$

1^+.	$A \supset (B \supset C)$	
2^+.	$A \supset B$	
3^+.	A	
4.	$B \supset C$	1, 3, $\supset_e$
5.	B	2, 3, $\supset_e$
6.	C	4, 5, $\supset_e$
7.	$A \supset C$	3, 6, $\supset_i$
8.	$(A \supset B) \supset (A \supset C)$	2, 7, $\supset_i$
9.	$(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$	1, 8, $\supset_i$

c) $(\neg B \supset \neg A) \supset ((\neg B \supset A) \supset B)$

1^+.	$\neg B \supset \neg A$	
2^+.	$\neg B \supset A$	
3^+.	$\neg B$	
4.	$\neg A$	1, 3, $\supset_e$
5.	A	2, 3, $\supset_e$
6.	$\neg\neg B$	3, 4, 5, $\neg_i$
7.	B	6, $\neg_e$
8.	$(\neg B \supset A) \supset B$	2, 7, $\supset_i$
9.	$(\neg B \supset \neg A) \supset ((\neg B \supset A) \supset B)$	1, 8, $\supset_i$

Chứng minh định lý Church: $((p \supset q) \supset p) \supset p$

1^+. $(p \supset q) \supset p$

2^+. $\neg p$

3^+. p

4^+. $\neg q$

5. $\neg\neg q$	2, 3, 4, $\neg_i$
6. q	5, $\neg_e$
7. $p \supset q$	3, 6, $\supset_i$
8. p	1, 76, $\supset_e$
9. $\neg\neg p$	2, 8, $\neg_i$
10. p	9, $\neg_e$
11. $((p \supset q) \supset p) \supset p$	1, 10, $\supset_i$

Như trên đây ta thấy các tiên đề của lý thuyết S đều là các định lý của hệ suy luận tự nhiên. Còn quy tắc MP của S trùng với các quy tắc $\supset_e$ của hệ suy luận tự nhiên. Từ đây suy ra rằng tất cả các định lý của lý thuyết S đều là định lý của hệ suy luận tự nhiên. Thật vậy, giả sử A là một định lý của lý thuyết S . Khi đó tồn tại một phép chứng minh A trong S , nghĩa là tồn tại một dãy hữu hạn các công thức của S , trong đó A là công thức cuối cùng và mỗi công thức của dãy hoặc là tiên đề, hoặc là nhận được từ các công thức đứng trước nó trong dãy theo quy tắc MP. Nếu ta thay các tiên đề trong dãy đó bằng các phép chứng minh chúng trong hệ suy luận tự nhiên thì ta được một phép chứng minh công thức A trong hệ suy luận tự nhiên. Ngược lại, cũng dễ dàng chứng minh được rằng tất cả các định lý của hệ suy luận tự nhiên (hệ suy luận tự nhiên không có tiên đề) đều là các định lý của lý thuyết S. Ký hiệu $\vdash_L$ A là công thức A chứng minh được trong hệ L, và NS là hệ suy luận tự nhiên, ta có:

Định lý 1.4

$$\vdash_{NS} A \Leftrightarrow \vdash_S A.$$

3. Tính không mâu thuẫn và đầy đủ của các hệ S và NS

Tính không mâu thuẫn và tính đầy đủ là những tính chất đặc biệt quan trọng của một số hệ logic. Tính không mâu thuẫn được hiểu theo hai nghĩa: Không mâu thuẫn nội tại (còn gọi là không mâu thuẫn cú pháp (Syntax)) và không mâu thuẫn ngữ nghĩa (Semantics). Hệ L gọi là không mâu thuẫn nội tại, nếu trong L không thể chứng minh được một công thức A nào đó, đồng thời chứng minh được phủ định $\neg$A của nó. Nghĩa là L không mâu thuẫn cú pháp khi không tồn tại A sao cho

$\vdash A$ và $\vdash \neg A$.

Định lý 1.5: *Nếu A là định lý của S (hoặc NS) thì A nhận toàn giá trị T trong bảng chân lý của nó (A là quy luật logic).*

Tính không mâu thuẫn Semantics (soundness) của hệ S và NS có nghĩa rằng mọi định lý của hệ S (và NS) đều là các quy luật logic, nghĩa là các công thức nhận toàn giá trị T trong bảng chân lý của nó.

Chứng minh. Ta chỉ cần chỉ rõ điều này với hệ S, hệ NS tương đương với S nên có kết luận tương tự. Trước hết ta thấy tất cả các tiên đề của S đều là quy luật logic. Quy tắc MP bảo toàn giá trị đúng, thật vậy, nếu X ⊃ Y đúng, Y đúng thì theo định nghĩa của phép kéo theo ⊃, Y cũng có giá trị đúng.

Giả sử A là định lý của S, khi đó có một phép chứng minh với A là công thức cuối cùng. Công thức đầu tiên không phải là tiên đề trong phép chứng minh này phải rút ra được từ các công thức trước nó là tiên đề theo MP, và vì vậy, nó là quy luật logic. Công thức tiếp sau đó nếu không là tiên đề thì cũng nhận được từ các quy luật logic theo MP, vậy là quy luật logic. Cứ như vậy, ta sẽ đến công thức cuối cùng A, và A phải là quy luật logic vì nhận được từ các quy luật logic theo MP.

Định lý 1.6 *Hệ S và NS không mâu thuẫn cú pháp , nghĩa là không tồn tại công thức A sao cho A và ¬A là định lý của S, hoặc A và ¬A là định lý của NS.*

Chứng minh Giả sử tồn tại công thức A sao cho cả A và ¬ A đều là định lý của S (NS). Khi đó, theo siêu định lý 3, cả A và ¬ A đều là quy luật logic. Nhưng điều này vô lý, vậy không tồn tại công thức A sao cho cả A và ¬ A đều chứng minh được trong S (NS).

Chúng ta thừa nhận tính đầy đủ S và NS trong siêu định lý sau đây:

Định lý 1.7 *Nếu A là quy luật logic thì A là định lý của hệ S (và NS).*

Các định lý 1.6 và 1.7 nói lên quan hệ giữa các định lý của hệ S (NS) và các quy luật logic (tức là các công thức chỉ nhận giá trị T trong bảng chân lý của mình). Như vậy, các định lý đó xác định ý nghĩa của các hệ logic S và NS.

BÀI TẬP CHƯƠNG 1

1. Hãy chứng minh các hệ phép toán sau đây là đầy đủ :
   a. $\{\neg, \supset\}$
   b. $\{\neg, \vee, \&\}$
2. Hãy dùng các phương pháp bảng chân trị và bảng ngữ nghĩa để xác định xem các công thức sau đây có là quy luật hay mâu thuẫn logic không ?
   a. $((p \supset q) \ \&\ \neg p) \supset \neg q$
   b. $((p \supset q) \ \&\ q) \supset p$
   c. $\neg (p \ \&\ q) \supset (\neg p \ \&\ \neg q)$

d. $\neg(p \vee q) \supset (\neg p \vee \neg q)$
e. $(p \supset \neg p) \supset q$
f. $((p \supset q) \supset p) \supset p$
g. $(p \supset q) \vee (q \supset p)$

3. Hãy dùng các phương pháp bảng chân trị và bảng ngữ nghĩa để xác định xem các công thức sau đây có là quy luật hay mâu thuẫn logic không ?
   a. $(p \supset (q \,\&\, r)) \supset ((p \supset q) \,\&\, (p \supset r))$
   b. $(p \supset (q \vee r)) \supset ((p \supset q) \vee (p \supset r))$
   c. $\neg((p \,\&\, q) \vee r) \supset ((\neg p \vee \neg q) \,\&\, \neg r)$
   d. $\neg((p \,\&\, q) \vee r) \supset ((\neg p \,\&\, \neg q) \vee \neg r)$
   e. $\neg((p \,\&\, q) \vee r) \supset (\neg r \supset (\neg p \vee \neg q))$

4. Hãy dùng các phương pháp bảng chân trị và bảng ngữ nghĩa để xác định xem các công thức sau đây có là quy luật hay mâu thuẫn logic không ?
   a. $((\neg p \,\&\, \neg q) \supset r) \supset ((\neg p \supset r) \,\&\, (\neg q \supset r))$
   b. $((\neg p \,\&\, \neg q) \supset r) \supset ((r \supset \neg p) \,\&\, (r \supset \neg q))$
   c. $((\neg p \,\&\, \neg q) \supset r) \supset ((\neg r \supset p) \,\&\, (\neg r \supset q))$
   d. $((\neg p \vee \neg q) \supset r) \supset ((\neg r \supset p) \vee (\neg r \supset q))$
   e. $((\neg p \vee \neg q) \supset r) \supset ((\neg p \supset r) \vee (\neg q \supset r))$

5. Hãy dùng các phương pháp bảng chân trị và bảng ngữ nghĩa để xác định xem các công thức sau đây có là quy luật hay mâu thuẫn logic không ?
   a. $((p \supset q) \,\&\, (q \supset r)) \supset (\neg r \supset \neg p)$
   b. $((p \vee q) \,\&\, (p \supset r) \,\&\, (q \supset s)) \supset (\neg r \supset s)$
   c. $((p \vee q) \,\&\, (p \supset r) \,\&\, (q \supset s)) \supset (r \vee s)$
   d. $((p \,\&\, q) \,\&\, (p \supset r) \,\&\, (q \supset s)) \supset (r \,\&\, s)$

6. Hãy dùng các phương pháp bảng chân trị và bảng ngữ nghĩa để xác định xem các công thức sau đây có là quy luật hay mâu thuẫn logic không ?
   a. $(p \vee q \vee r) \equiv (\neg r \,\&\, \neg q \,\&\, \neg p)$
   b. $(p \,\&\, q \,\&\, r) \equiv (\neg r \vee \neg q \vee \neg p)$
   c. $((p \,\&\, q) \supset r) \equiv (\neg r \supset (\neg p \vee \neg q))$
   d. $(p \supset (q \vee r)) \equiv (\neg p \supset (\neg q \,\&\, \neg r))$
   e. $(p \supset (q \,\&\, r)) \equiv ((\neg p \vee q) \,\&\, (\neg p \vee r))$

7. Hãy rút gọn các công thức sau đây :
   a. $p \supset (p \supset q)$
   b. $((p \supset q) \,\&\, (p \supset \neg q)) \supset r$
   c. $(p \supset q) \supset ((q \supset r) \supset (p \supset r))$
   d. $(p \supset q) \,\&\, (p \supset r)$
   e. $(p \supset q) \vee (p \supset r)$
   f. $(p \vee q \vee r) \supset (s \vee u \vee p)$

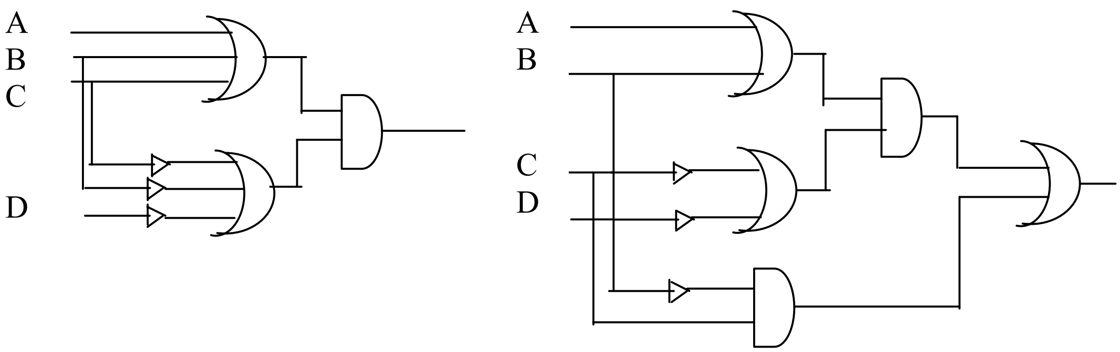
8. Hãy rút gọn các công thức sau đây :
   a. $p.q + q.r + p.r + p.q.r$
   b. $(p.q + p.r) + p.q.r$
   c. $p.(q + r) + q.(p + r) + r.(p + q)$

d. $(p + q.r) + (q + p.r) + (r + p.q)$

e. $\overline{ab} + \overline{bc} + \overline{ca} + abc$

f. $\overline{abc} + a\overline{bc} + \overline{a}\,b\,\overline{c} + \overline{ab}\,c + abc$

9. Hãy chứng tỏ rằng các mạch điện tử sau đây tương đương với nhau:



10. Chứng minh các định lý sau đây trong hệ tiên đề logic mệnh đề:
    a. $p \supset p$
    b. $\neg\neg p \supset p$
    c. $p \supset \neg\neg p$
    d. $(p \supset (q \supset r)) \supset (q \supset (p \supset r))$
    e. $p \supset ((p \supset r) \supset r)$
    f. $\neg p \supset (p \supset q)$

11. Chứng minh các định lý sau đây trong hệ tiên đề logic mệnh đề:
    a. $(p \supset \neg p) \supset \neg p$
    b. $(\neg p \supset \neg q) \supset (q \supset p)$
    c. $(p \supset q) \supset ((q \supset r) \supset (p \supset r))$
    d. $((p \supset p) \supset p) \supset p$
    e. $(p \supset q) \supset ((\neg p \supset q) \supset q)$
    f. $p \supset (\neg q \supset \neg (p \supset q))$

12. Chứng minh các dạng thức suy luận sau đây trong hệ suy luận tự nhiên logic mệnh đề:
    a. $((p \supset q) \& p) \supset q$
    b. $((p \supset q) \& \neg q) \supset \neg p$
    c. $((p \vee q) \& \neg p) \supset q$
    d. $((p \vee q) \& (p \supset r) \& (q \supset r)) \supset r$
    e. $((p \vee q) \& (p \supset r) \& (q \supset s)) \supset (r \vee s)$
    f. $\neg(p \& q) \equiv (\neg p \vee \neg q)$
    g. $\neg(p \vee q) \equiv (\neg p \& \neg q)$

13. Chứng minh các dạng thức suy luận sau đây trong hệ suy luận tự nhiên logic mệnh đề:

    a. $(((p \& q) \supset r) \& \neg r) \supset (\neg p \vee \neg q)$
    b. $(((p \vee q) \supset r) \& \neg r) \supset (\neg p \& \neg q)$

c. $((p \supset (q \vee r)) \,\&\, (\neg q \,\&\, \neg r)) \supset \neg p$
d. $((p \supset (q \,\&\, r)) \,\&\, (\neg q \vee \neg r)) \supset \neg p$
e. $(((p \vee q) \supset r) \,\&\, \neg r) \supset (\neg p \,\&\, \neg q)$
f. $p \supset (q \supset (p \,\&\, q))$

14. Chứng minh các dạng thức suy luận sau đây trong hệ suy luận tự nhiên logic mệnh đề:
   a. $p \supset p$
   b. $\neg\neg p \supset p$
   c. $p \supset \neg\neg p$
   d. $p \supset (q \supset p)$
   e. $(p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$
   f. $(p \supset (q \supset r)) \supset (q \supset (p \supset r))$
   g. $p \supset ((p \supset r) \supset r)$
   h. $(p \supset \neg p) \supset \neg p$
   i. $(\neg p \supset \neg q) \supset (q \supset p)$
   j. $(p \supset q) \supset ((q \supset r) \supset (p \supset r))$
   k. $(p \supset q) \supset ((p \supset \neg q) \supset \neg p)$
15. Chứng minh các định lý sau đây trong hệ suy luận tự nhiên logic mệnh đề
   a. $p \vee \neg p$
   b. $(p \supset q) \vee (q \supset p)$
   c. $((p \supset q) \,\&\, (\neg p \supset r) \,\&\, ((q \vee r) \supset s)) \supset s$
   d. $((p \,\&\, q) \supset r) \supset (\neg r \supset (\neg p \vee \neg q))$
   e. $(p \vee (q \,\&\, r)) \equiv ((p \vee r) \,\&\, (q \vee p))$
   f. $(p \,\&\, (q \vee r)) \equiv ((p \,\&\, r) \vee (q \,\&\, p))$
16. Hãy dùng hệ suy luận tự nhiên của logic mệnh đề để chứng tỏ rằng những suy luận sau đây là đúng :
   a. Con người bao giờ cũng ở một trong hai trạng thái : đang sống hoặc đã chết. Nếu con người đang sống thì chưa có cái chết nên không cần sợ cái chết. Ngược lại, nếu con người đã chết thì chẳng còn biết gì nữa, nên tất nhiên cũng chẳng cần sợ cái chết. Như vậy, chẳng cần sợ chết.
   b. Minh sẽ được nhận vào làm việc tại doanh nghiệp Nhật nếu anh biết tiếng Nhật và anh biết nghiệp vụ xuất nhập khẩu. Nếu sẽ được nhận vào làm việc tại doanh nghiệp Nhật hoặc tìm được học bổng thì anh có thể đi du học ở nước ngòai. Biết rằng Minh khong thể đi du học ở nước ngòai. Vậy Minh không biết nghiệp vụ xuất nhập khẩu.
   c. Khi các ngôi sao, các thiên hà chạy ra xa chúng ta, tức là khi vũ trụ giãn nở, ánh sáng của chúng sẽ càng ngày càng chuyển về phần đỏ tren dãy quang phổ (hiệu ứng dịch chuyển về phần đỏ). Vũ trụ chỉ giãn nở vì có một vụ nổ lớn ban đầu gọi là Big Bang. Năm 1929 nhà thiên văn học người Mỹ Hubble đã quan sát thấy hiệu ứng dịch chuyển về phần đỏ. Như vậy, nếu hiệu ứng dịch chuyển về phần đỏ chỉ có thể do sự giãn nở của vũ trụ gây ra thì Big Bang là có thật.

aChương II HỢP GIẢI TRONG LOGIC MỆNH ĐỀ

Hợp giải là một phương pháp logic hiện đại để rút ra kết luận từ một tập hợp các tiền đề cho trước[1]. Phương pháp này được nhà logic người Mỹ J.A. Robinson đề xuất vào đầu những năm 60 của thế kỷ XX. Hiện nay phương pháp này được sử dụng nhiều trong tin học, đặc biệt là trong lĩnh vực trí tuệ nhân tạo. Nó cũng là nền tảng logic của ngôn ngữ lập trình PROLOG. Chúng ta đã làm quen với hợp giải trong phần logic nhập môn, ở đây ta xét kỹ hơn dạng đơn giản của phương pháp này, cụ thể là dạng ứng với logic mệnh đề, dạng đầy đủ của phương pháp này, dạng ứng với logic vị từ, sẽ được xem xét trong chương 4.

I. Công thức dạng tuyển

1. Định nghĩa

Định nghĩa 2.1. Đơn tử – còn gọi là *Literal* - là một mệnh đề đơn, hoặc là phủ định của một mệnh đề đơn.

Ví dụ, cho p là một mệnh đề đơn, khi đó p và ¬p là các literal.

Định nghĩa 2.2. Công thức dạng tuyển là công thức dạng $X_1 \vee X_2 \vee \ldots \vee X_n$, trong đó X_i với i = 1, 2, …, n, là literal.

Ví dụ, với p, q, r là các mệnh đề đơn thì các biểu thức sau đây là các công thức dạng tuyển :

$p \vee q \,\&\, \neg r$, $\neg p \vee \neg q \vee \neg r$, $\neg p \vee q \vee \neg r$;

các biểu thức sau đây không phải là công thức dạng tuyển :

$\neg(p \,\&\, q) \vee r$, $\neg p \vee \neg(q \vee r)$, …

Dạng tuyển còn được biểu thị dưới dạng tập hợp. Dạng tuyển $X_1 \vee X_2 \vee \ldots \vee X_n$ khi đó được thay thế bằng tập hợp $\{ X_1, X_2, \ldots, X_n\}$.

Khi n =0, công thức dạng tuyển $X_1 \vee X_2 \vee \ldots \vee X_n$ được ký hiệu là □, dạng tập hợp của nó là tập rỗng ∅, { }. □, hay { } là mâu thuẫn logic.

Định lý 2.1. *Mỗi tập công thức của logic mệnh đề đều có tập công thức dạng tuyển tương đương*.

Chúng ta chứng minh định lý vừa nêu trong mục tiếp theo, bằng cách nêu ra một quy trình biến đổi công thức A bất kỳ thành hội của một số công thức dạng tuyển, mỗi bước của quy trình đó đều biến đổi công thức thành một công thức hoặc một tập hợp công thức tương đương với công thức ban đầu (ta nói công thức A tương đương với tập hợp công thức Γ khi và chỉ khi A tương đương với công thức B, với B là hội của tất cả các công thức trong Γ).

[1] Chính xác hơn thì đây là phương pháp cho phép kiểm tra xem có thể rút ra kết luận nhất định nào đó từ tập tiền đề cho trước hay không, và là phương pháp chứng minh định lý tự động.

2. *Quy trình INDO*

Quy trình INDO là một quy trình gồm các bước I, N, D, O (được xác định dưới đây), giúp biến đổi công thức bất kỳ thành một hoặc một số công thức dạng tuyển.

I – loại bỏ phép kéo theo (Implication Out) :

$A \supset B \qquad\qquad\qquad \Rightarrow \neg A \vee B$

N – Đưa dấu phủ định vào đứng trước các mệnh đề đơn (Negation In) :

$\neg \neg A \Rightarrow A$

$\neg (A \,\&\, B) \Rightarrow \neg A \vee \neg B$

$\neg (A \vee B) \Rightarrow \neg A \,\&\, \neg B$

D – Đưa dấu tuyển vào sâu hơn dấu hội (Disjunctions in):

$A \vee (B \,\&\, C) \Rightarrow (A \vee B) \,\&\, (A \vee C)$

O – Loại bỏ dấu ∨ và & (Operators Out):

$A \vee B \Rightarrow \{A, B\}$

$A \,\&\, B \Rightarrow \{A\}$

$\qquad\quad \Rightarrow \{B\}$

Nếu muốn biểu đạt công thức dạng tuyển dưới dạng $X_1 \vee X_2 \vee \ldots \vee X_n$ thì ở bước O không cần loại bỏ dấu hội &, còn để biểu đạt dưới dạng tập hợp thì cần loại bỏ dấu &.

Ví dụ 1. $(p \supset q) \vee ((q \,\&\, \neg r) \supset p)$

I: $(\neg p \vee q) \vee (\neg (q \,\&\, \neg r) \vee p)$

N: $((\neg p \vee q) \vee ((\neg q \vee \neg\neg r) \vee p)$

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

D: $\neg p \vee q \vee \neg q \vee r \vee p$

O: $\{\neg p, q, \neg q, r, p\}$

Ví dụ 2. $(p \supset q) \,\&\, (\neg p \supset r) \,\&\, ((p \vee r) \supset s)$

I: $(\neg p \vee q) \,\&\, (\neg\neg p \vee r) \,\&\, (\neg (q \vee r) \vee s))$

N: $(\neg p \vee q) \,\&\, (p \vee r) \,\&\, (\neg (q \vee r) \vee s))$

$(\neg p \vee q) \,\&\, (p \vee r) \,\&\, ((\neg q \,\&\, \neg r) \vee s)$

D: $(\neg p \vee q) \,\&\, (p \vee r) \,\&\, ((\neg q \vee s) \,\&\, (\neg r \vee s))$

O: $\{\neg p, q\}$

$\{p, r\}$

$\{\neg q, s\}$

. $\{\neg r, s\}$.

II. Quy tắc hợp giải

Với các công thức A, B bất kỳ :

$$\frac{A \vee B, \neg A \vee C}{B \vee C}$$

Từ các tiền đề $\neg A \vee B$ và $A \vee C$ ta có thể rút ra kết luận $B \vee C$. Kết luận này được gọi là *resolvent.* Trong trường hợp không có thành phần B và C thì được *resolvent rỗng*, ký hiệu bằng hình vuông nhỏ $\square$. Như vậy resolvent rỗng $\square$ xuất hiện khi có hai tiền đề mâu thuẫn với nhau.

Ở phần logic nhập môn, ngòai quy tắc hợp giải nêu trên đây , chúng ta còn gặp các quy tắc :

$$\frac{\neg A, \quad A}{\square} \qquad\qquad \frac{\neg A \vee B, \quad A \vee B}{\mathrm{B}}$$

Dễ thấy rằng thực ra các quy tắc này chỉ là các trường hợp riêng của quy tắc đã nêu trên kia mà thôi, nên ở đây chúng tôi không nêu chúng.

Sau đây là một số ví dụ áp dụng các quy tắc hợp giải.

Ví dụ 1. $p \vee q \vee \neg r, \quad \neg q \vee \neg s$

$p \vee \neg r \vee \neg s$

Ví dụ 2. $\neg p \vee q \vee r \qquad \neg q \vee r$

$\neg p \vee r$

Với dạng tập hợp, ta có quy tắc hợp giải chặt chẽ hơn, như sau :

$$\frac{\begin{array}{c}\{\varphi_1, \varphi_2, \ldots, \varphi_i, \varphi, \varphi_{i+1}, \ldots, \varphi_n\} \\ \{\psi_1, \psi_2, \ldots, \psi_j, \neg\varphi, \psi_{j+1}, \ldots, \psi_n\}\end{array}}{\{\varphi_1, \varphi_2, \ldots \varphi_n, \psi_1, \psi_2, \ldots \psi_n\}}$$

Ví dụ 1: từ 1. $\{p, \neg q, r\}$

2. $\{p, q, s\}$

rút ra 3. $\{p, r, s\}$

Ví dụ 2: từ 1. $\{q\}$

2. $\{\neg q\}$

rút ra 3. $\{\ \}$

Lưu ý : từ $\{p, q\}$ và $\{\neg p, \neg q\}$ không rút ra được $\{\ \}$.

Người ta sử dụng các quy tắc hợp giải để kiểm tra xem một tập hợp các công thức có mâu thuẫn hay không. Ngoài ra còn để xác định xem từ một tập các công thức cho trước có thể rút ra được một công thức nhất định nào đó hay không.

III. Phương pháp hợp giải

Thực chất của phương pháp hợp giải là chứng minh bằng phản chứng. Để chứng minh rằng từ một tập tiền đề $\{A_1, A_2, \dots, A_n\}$ cho trước có thể rút ra kết luận B, ta thêm $\neg B$ vào tập tiền đề này, được tập mới $\{A_1, A_2, \dots, A_n, \neg B\}$. Khi đó nếu trong tập mới nhận được có mâu thuẫn thì phép chứng minh đã hoàn tất. Nếu tập mới không có mâu thuẫn thì không thể rút ra được $\neg B$ từ $\{A_1, A_2, \dots, A_n\}$. Phương pháp hợp giải áp dụng quy tắc hợp giải để xác định xem tập công thức có mâu thuẫn hay không.

Để xác định xem tập công thức cho trước $\{A_1, A_2, \dots, A_n, \neg B\}$ có mâu thuẫn không, ta áp dụng quy tắc hợp giải cho các cặp công thức của tập này. Các resolvent nhận được sẽ được thêm vào tập công thức, nếu chúng chưa có trong tập công thức đó. Quá trình này được tiếp tục cho đến khi xảy ra một trong các trường hợp sau:

Trường hợp 1: Trong tập công thức xuất hiện resolvent rỗng □. Kết luận: Tập công thức đã xét có mâu thuẫn. Nghĩa là có thể rút ra B từ tập công thức $\{A_1, A_2, \dots, A_n\}$.

Trường hợp 2: Không thể áp dụng quy tắc hợp giải cho bất kỳ cặp công thức nào nữa. Kết luận: Tập công thức đã xét không có mâu thuẫn. Nghĩa là không thể rút ra B từ tập công thức $\{A_1, A_2, \dots, A_n\}$.

Trường hợp 3: Việc áp dụng quy tắc hợp giải không làm thay đổi tập công thức nữa. Kết luận: Tập công thức đã xét không có mâu thuẫn. Nghĩa là không thể rút ra B từ tập công thức $\{A_1, A_2, \dots, A_n\}$.

Ví dụ 1. Xét xem từ tập tiền đề $\{p \vee q, \neg p \vee r, s, \neg q \vee r\}$ có thể rút ra kết luận *r* không?

Giải: Thêm $\neg r$ vào tập tiền đề. Sau đó áp dụng quy tắc hợp giải, ta được:

$$\{p \vee q,\ \neg p \vee r,\ s,\ \neg q \vee r,\ \neg r\}$$

$$\{p \vee q,\ \neg p \vee r,\ s,\ \neg q \vee r,\ \neg r,\ q \vee r,\ p \vee r,\ \neg p,\ \neg q\}$$

$$\{p \vee q,\ \neg p \vee r,\ s,\ \neg q \vee r,\ \neg r,\ q \vee r,\ p \vee r,\ \neg q,\ \neg p,\ q,\ p,\ r,\ \ldots\}$$

$$\{p \vee q,\ \neg p \vee r,\ s,\ \neg q \vee r,\ \neg r,\ q \vee r,\ p \vee r,\ \neg q,\ \neg p,\ q,\ p,\ \square,\ \ldots\}.$$

Ví dụ 2. Xét xem từ tập tiền đề $\{\neg p \vee r \vee s,\ \neg q \vee r,\ p\}$ có thể rút ra kết luận r không?
Giải: Thêm $\neg$ r vào tập tiền đề, rồi áp dụng các quy tắc hợp giải, ta được:

$$\{\neg p \vee r \vee s,\ \neg q \vee r,\ p,\ \neg r\}.$$

$$\{\neg p \vee r \vee s,\ \neg q \vee \neg r,\ p,\ \neg r,\ \neg p \vee s,\ \neg q,\ r \vee s\}$$

$$\{\neg p \vee r \vee s,\ \neg q \vee r,\ p,\ \neg r,\ \neg q,\ r \vee s,\ s,\ \neg p \vee s,\ \neg q \vee s\}.$$

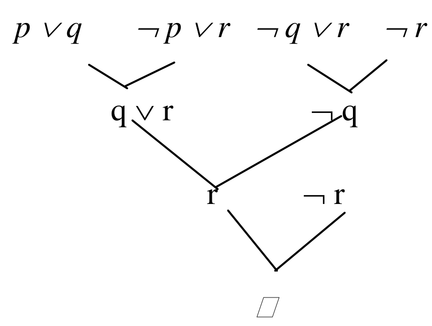
Đến đây ta thấy có thể áp dụng tiếp các quy tắc hợp giải cho một số cặp công thức, tuy nhiên các resolvent nhận được không mới, đã có sẵn trong tập công thức trên đây. Như vậy, tập công thức này không có mâu thuẫn, nghĩa là không thể rút ra r từ tập tiền đề $\{\neg p \vee r \vee s,\ \neg q \vee r,\ p\}$.

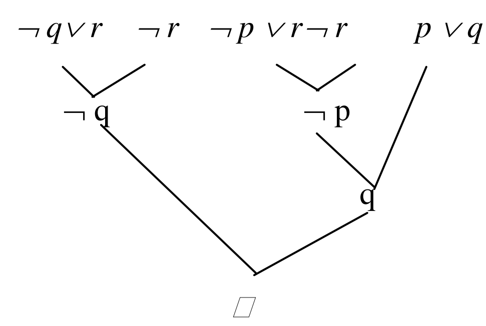
Ví dụ 3. Xét xem từ tập tiền đề $\{\neg p \vee \neg s,\ \neg q \vee r\}$ có thể rút ra kết luận *s* không?

Giải: Thêm $\neg$ s vào tập tiền đề đã cho, ta được tập $\{\neg p \vee \neg s,\ \neg q \vee r,\ \neg s\}$. Không thể áp dụng các quy tắc hợp giải vào các cặp công thức trong tập này. Vậy tập công thức này không mâu thuẫn, không thể rút ra *s* từ tập $\{\neg p \vee \neg s,\ \neg q \vee r,\ \}$.

IV. Cây hợp giải. Hợp giải tuyến tính

Các lời giải bài toán bằng phương pháp hợp giải có thể biểu diễn dưới dạng hình vẽ dạng cây, trong đó chỉ nêu ra các công thức cần thiết để đi đến kết luận, những công thức khác không cần nêu lên. Chẳng hạn, lời giải trên đây và một lời giải khác của ví dụ 1 được biểu diễn dạng cây thành:

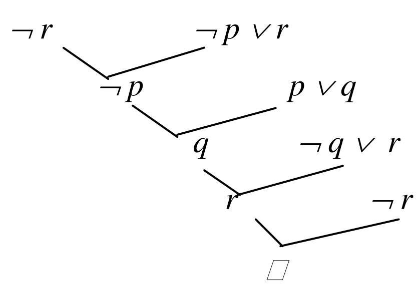


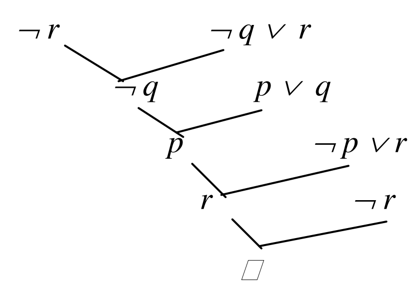


Dạng biểu diễn cây của các lời giải như thế được gọi là cây hợp giải. Mỗi lời giải của bài toán tương ứng với một cây hợp giải. Robinson đã chứng minh được định lý: *Từ tập tiền đề $\{A_1, A_2, \ldots, A_n\}$ có thể rút ra kết luận B khi và chỉ khi tồn tại ít nhất một cây hợp giải cho tập $\{A_1, A_2, \ldots, A_n, \neg B\}$.*

Phương pháp hợp giải như đã trình bày trên đây có nhược điểm là ở các bước có thể xuất hiện những resolvent không cần thiết đối với việc đi đến kết luận. Khi áp dụng quy tắc hợp giải vào tất cả các cặp công thức có thể áp dụng được, số lượng các resolvent tăng lên rất nhanh chóng, xảy ra bùng nổ tổ hợp. Để tránh điều này, R.A. Kowalski đưa ra phương pháp hợp giải tuyến tính. Ở đây, khác với hợp giải thông thường, trước hết ta xác định một công thức từ tập $\{A_1, A_2, \ldots, A_n, \neg B\}$ có thể cùng với $\neg B$ áp dụng quy tắc hợp giải. Được resolvent B_1, thêm nó vào tập công thức đã có, lại xác định một công thức từ tập $\{A_1, A_2, \ldots, A_n, \neg B, B_1\}$ có thể cùng B_1 áp dụng quy tắc này. Cứ tiếp tục như thế cho đến khi được resolvent rỗng, hoặc không thể tiếp tục vì không tìm ra công thức cần tìm, hoặc việc tiếp tục chỉ lặp lại các kết quả đã có. Cây hợp giải tương ứng được gọi là cây hợp giải tuyến tính. Kowalski đã chứng minh được định lý : *Từ tập tiền đề không mâu thuẫn $\{A_1, A_2, \ldots, A_n\}$ có thể rút ra kết luận B khi và chỉ khi tồn tại ít nhất một cây hợp giải tuyến tính cho tập $\{A_1, A_2, \ldots, A_n, \neg B\}$.*

Ví dụ 1 trên kia có các cây hợp giải tuyến tính sau :





Để tìm lời giải của bài toán, nghĩa là để xây dựng cây hợp giải tuyến tính, người ta sử dụng *kỹ thuật quay lui (backtracking)*.

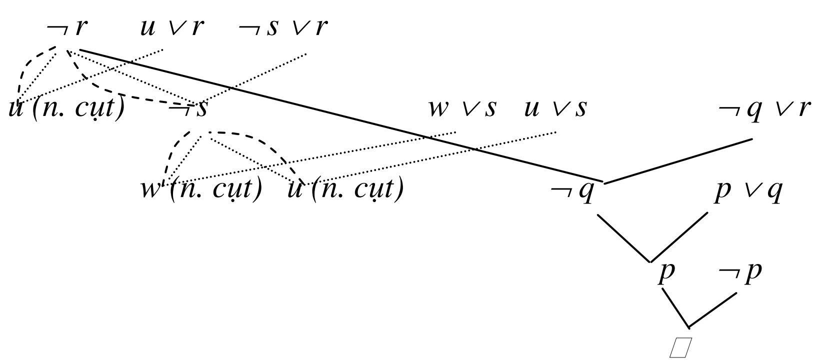
Dãy liên tục các resolvent trong hợp giải tuyến tính gọi là một nhánh. Nhánh này gọi là nhánh cụt, hay nhánh thất bại, nếu nó kết thúc bằng một công thức nào đó khác □. Nhánh này gọi là nhánh tuần hoàn, nếu đến một lúc nào đó bắt đầu lặp lại các resolvent đã có từ trước. Nhánh tuần hoàn cũng là nhánh thất bại. Nhánh kết thúc bằng □ gọi là nhánh thành công.

Giả sử việc áp dụng quy tắc hợp giải vào cặp công thức B_{i-1} với một công thức khác cho ta kết quả B_i. Khi đó từ tập các công thức đang khảo sát ở bước này xác định một tập con các công thức có thể cùng với B_i tạo thành cặp để áp dụng quy tắc hợp giải. Ta chọn trong tập con này một công thức, áp dụng quy tắc hợp giải cho cặp công thức vừa chọn và B_i, được resolvent B_{i+1}. Với B_{i+1} lại xác định tập con công thức có thể tạo cặp để áp dụng quy tắc hợp giải. Quá trình cứ vậy tiếp diễn. Nếu tất cả các nhánh con bắt đầu từ B_{i+1} đều thất bại thì quay trở lại với B_i. Bây giờ ta chọn công thức khác tạo cặp với B_i để áp dụng quy tắc hợp giải. Nếu tất cả các nhánh con bắt đầu từ B_i cũng đều thất bại, thì tiếp tục quay lui đến $B_{i-1.}$ Bằng cách này sẽ tìm được nhánh thành công, tức là xây dựng được cây hợp giải tuyến tính, nếu nó tồn tại.

Ví dụ 4. Xây dựng cây hợp giải tuyến tính rút ra r từ tập công thức

$\{\neg s \vee r, \neg p \vee q, \neg q \vee r, p, u \vee r, w \vee s\}$

Giải. Sơ đồ tìm kiếm lời giải như sau, trong sơ đồ này các dấu mũi tên vòng chỉ các quay lui.



Sơ đồ trên đây cho thấy lúc đầu ta đi từ $\neg r$ đến *u*. Đây là nhánh cụt, vì thế quay trở lại $\neg r$. Từ đây đi đến $\neg s$, từ $\neg s$ đi đến *w*, rồi lại quay về *s* vì là nhánh cụt. Từ *s* đi theo hướng khác đến *u*, đây cũng là nhánh cụt, nên quay về *s*. Vì các khả năng khác đi từ *s* đã hết, nên quay tiếp về $\neg r$. Từ đây đi đến $\neg q$. Từ $\neg q$ đi đến *p*, đi tiếp đến □, đây là nhánh thành công. Cây hợp giải tuyến tính cần xây dựng được biểu diễn bằng các đường kẻ liền trong hình.

V. Giản lược tiền đề

Các ví dụ hợp giải trên kia đã cho thấy có những tiền đề hoàn toàn không cần đến trong quá trình đi đến resolvent rỗng. Lại có cả những tiền đề mà bất cứ nhánh hợp giải nào dùng đến chúng đều không thể đi đến resolvent rỗng. Những tiền đề như vậy hoàn toàn không cần đến trong quá trình rút ra kết luận cần thiết, trái lại, chúng là cho quá trình đó trở nên khó khăn hơn. Vì vậy nên loại bỏ chúng trước khi tiến hành hợp giải. Trong mục này chúng ta xem xét một số trường hợp lược bỏ tiền đề như vậy.

Để cho đơn giản chúng ta giả định rằng tập hợp các tiền đề gồm các phần tử là công thức dạng tuyển.

1. Giản lược tiền đề là quy luật logic

Công thức dạng tuyển là quy luật logic nếu nó chứa cặp đơn tử trái ngược nhau, chẳng hạn như p và $\neg p$.

Ví dụ : $p \vee q \vee \neg p$; $p \vee q \vee \neg q \neg r$; $\neg p \vee \neg r \vee q \vee r \vee \neg q; s \vee \neg s$;
Đều là các quy luật logic.
Cho một tập hợp tiền đề S, giản lược hết tất cả các tiền đề là quy luật logic ta được tập S’. Dễ thấy rằng tập S mâu thuẫn khi và chỉ khi tập S’ mâu thuẫn. Để chứng minh điều này chúng ta chỉ cần chỉ ra rằng nếu với các tiền đề có trong S chúng ta có thể rút ra dạng tuyển rỗng thì khi bỏ bớt đi mộ tiền đề là quy luật logic ta vẫn rút ra được dạng tuyển rỗng. Thật vậy, nếu để rút ra □ lúc đầu đã không cần đến tiền đề $p \vee \neg p \vee A$ (trong đó A là một công thức dạng tuyển) thì việc loại bỏ nó hiển nhiên không ảnh hưởng gì đến việc rút ra □. Ta xét trường hợp lúc đầu để rút ra □ đã dùng đến tiền đề chứa $p \vee \neg p \vee A$. Bấy giờ, để đi đến □ ta phải triệt tiêu được p và $\neg p$. Điều này chỉ có thể thực hiện được nếu trong S có tiền đề nào đó chứa $\neg p$ và tiền đề chứa p. Khi áp dụng quy tắc hợp giải để loại bỏ các đơn tử p và $\neg p$ trong quy luật logic đang đề cập với các tiền đề này, ta được công thức $A \vee B \vee C$. Nhưng nếu áp dụng quy tắc hợp giải với cặp tiền đề $p \vee B$ và $\neg p \vee C$ thì ta được công thức $B \vee C$. Rõ ràng là nếu từ tiền đề $A \vee B \vee C$ có thể đi đến □ thì từ $B \vee C$ cũng có thể đi đến □ được (bằng cách bỏ bớt các bước áp dụng quy tắc hợp giải để loại bỏ A). Như vậy cả tiền đề $p \vee \neg p \vee A$ cũng không cần thiết để rút ra dạng tuyển rỗng □. Như vậy, ta đã chứng minh được định lý sau :

Định lý 2.2. *Cho một tập hợp tiền đề S, tập S^* là kết quả giản lược hết tất cả các tiền đề là quy luật logic trong S, khi đó S mâu thuẫn khi và chỉ khi S^* mâu thuẫn.*

***Hệ quả 2.3. có thể giản lược các tiền đề là quy luật logic*.**

2. Giản lược tiền đề một chiều

Cho tập tiền đề S. Nếu một đơn tử, chẳng hạn p (hay $\neg p$), xuất hiện trong S, mà trong S không có đơn tử $\neg p$ (hay p), thì đơn tử p được gọi là đơn tử một chiều trong S. Tiền đề một chiều trong S là tiền đề chứa đơn tử một chiều trong S (từ đây về sau, ở những chỗ không sợ nhầm lẫn chúng tôi sẽ gọi ngắn gọn là đơn tử một chiều và tiền đề một chiều).
Ví dụ : Cho S = $\{p \vee q; \neg q \vee r \vee s; \neg s \vee \neg p; r \vee s \vee q\}$
R là đơn tử một chiều, vì thế $\neg q \vee r \vee s$, $r \vee s \vee q$ là các tiền đề một chiều.

Nếu đơn tử p là một chiều thì nó không thể bị triệt tiêu trong hợp giải, và vì vậy các nhánh hợp giải chứa tiền đề một chiều cũng không thể dẫn đến □, tức đều sẽ là nhánh thất bại. Như vậy tiền đề một chiều hoàn toàn không giúp đi đến dạng tuyển rỗng □. Điều này có nghĩa là ta đã chứng minh định lý sau đây:

Định lý 2.4. *Cho một tập hợp tiền đề S, tập S^* là kết quả giản lược hết tất cả các tiền đề một chiều trong S, khi đó S mâu thuẫn khi và chỉ khi S^* mâu thuẫn.*

Hệ quả 2.5. có thể giản lược các tiền đề một chiều.

3. Giản lược tiền đề yếu

Nếu trong tập S có tiền đề A thì tiền đề dạng $A \vee B$ trong S được gọi là tiền đề yếu trong S.

Ví dụ : $S = \{p \vee q \vee r; p \vee s ; s \vee \neg q; q \vee r\}$

$p \vee q \vee r$ là tiền đề yếu trong S.

Với tiền đề yếu ta dễ dàng chứng minh được định lý sau đây:

Định lý 2.6. *Gọi S^* là kết quả việc loại bỏ toàn bộ các tiền đề yếu của S. Khi đó S mâu thuẫn khi và chỉ khi S^* mâu thuẫn. (Bạn đọc hãy tự chứng minh điều này).*

Hệ quả 2.7. Có thể giản lược các tiền đề yếu.

Ví dụ: Xét xem có thể rút ra kết luận *s* từ tập hợp tiền đề sau đây không

$\{\neg p \vee q \vee r, p \vee s, q \vee s, \neg r \vee q, \neg q \vee \neg s \vee p, r \vee s \vee \neg r\}$

Giải: Thêm $\neg s$ vào tập tiền đề đã cho, ta được tập hợp S

$S = \{\neg p \vee q \vee r, p \vee s, q \vee s, \neg r \vee p, \neg q \vee \neg s \vee p, r \vee s \vee \neg r, \neg s\}$

Trong S ta thấy: $r \vee s \vee \neg r$ là quy luật logic, có thể lược bỏ.

$\neg q \vee \neg s \vee p$ là tiền đề yếu, có thể lược bỏ.

Sau khi lược bỏ hai tiền đề đã nêu, xuất hiện các tiền đề yếu:

$\neg p \vee q \vee r$ và $q \vee s$,

Sau khi lược bỏ hai tiền đề yếu đã nêu, xuất hiện các tiền đề yếu mới:

$p \vee s$, và $\neg r \vee p$

Lược bỏ các tiền đề đó, tiền đề cuối cùng còn lại, $\neg s$, cũng trở thành tiền đề yếu. Loại bỏ nó, tập hợp tiền đề bây giờ rỗng, tập hợp đó không mâu thuẫn.

Như vậy, không thể rút ra kết luận *s* từ tập hợp tiền đề đã cho.

BÀI TẬP CHƯƠNG 2

1. Hãy đưa các công thức sau đây về dạng tuyển :
   a. $p \supset (q \,\&\, r)$
   b. $(p \vee q) \supset r$
   c. $((p \supset q) \,\&\, (r \supset s)) \supset (s \,\&\, r)$
   d. $(p \vee q \vee r) \supset ((\neg p \,\&\, q) \supset \neg r)$
   e. $(r \vee (p \,\&q)) \supset ((p \supset \neg q) \,\&\, (r \supset \neg p))$
2. Hãy xác định và giản lược các tiền đề yếu, tiền đề một chiều, các quy luật logic trong các tập hợp tiền đề sau đây:
   a. $\{p \vee r,\ s \vee \neg r \vee q,\ \neg p \vee s \vee q,\ q \vee s\}$
   b. $\{p \vee q \vee \neg r,\ s \vee r,\ \neg p \vee s,\ \neg q \vee s \vee u,\ \neg u\}$
   c. $\{p \vee q \vee r,\ \neg r,\ s \vee r,\ \neg p \vee s,\ \neg q \vee s \vee u,\ p\}$
   d. $\{q \vee s \vee p,\ q \vee \neg p,\ p \vee r \vee \neg s,\ s \vee q \vee \neg s\}$
   e. $\{p,\ q,\ \neg p \vee q \vee \neg r,\ r \vee s \vee \neg q,\ q \vee p \vee s,\ \neg p \vee u,\ \neg q \vee u\ ,\ \neg u \vee r,\ \neg u \vee s\ \}$
3. Hãy xác định và giản lược các tiền đề yếu, tiền đề một chiều, các quy luật logic trong các tập hợp tiền đề sau đây:
   a. $\{p \supset (q \vee r),\ (r \,\&\, s) \supset q,\ p \vee q \vee \neg r,\ p \supset (\neg q \supset p)\}$
   b. $\{(p \vee q \vee \neg r) \supset u\ ,\ s \vee r,\ \neg p \vee u,\ q \supset u,\ s \supset u,\ \neg u\}$
   c. $\{p \supset r,\ q \supset r,\ \neg s \vee w,\ \neg r \vee s,\ q \vee p \vee s\ ,\ \neg w\ \}$
   d. $\{p \supset r,\ q \supset r,\ s \vee w,\ \neg r \vee s,\ q \vee p\ ,\ \neg r \vee w\ \}$
   e. $\{p \supset (p \vee q),\ q \supset r,\ q \supset s,\ (s \vee r \vee u) \supset q,\ \neg u \vee r \vee p,\ \neg r \supset (q \vee u)\}$
4. a. Cho tập hợp các tiền đề :
   $\{p \vee \neg q \vee \neg r,\ p \supset s,\ w \supset s,\ r \vee s,\ \neg s \supset q\ \}$
   Từ tập tiền đề đã cho có thể rút ra kết luận *s* không ?

   b. Cho các tiền đề $\{p \vee q \vee \neg r,\ s \vee r,\ \neg p \vee s,\ \neg q \vee s \vee u,\ \neg u\}$. Hãy xác định xem từ tập tiền đề đã cho có thể rút ra kết luận $u \vee s$ hay không.

   c. Cho các tiền đề $\{(p \,\&\, q) \supset r,\ \neg r,\ s \vee r,\ \neg p \vee s,\ \neg q \vee s \vee u,\ p\}$. Hãy xác định xem từ tập tiền đề đã cho có thể rút ra kết luận $u \vee \neg q$ hay không.

   d. Cho các tiền đề $\{(p \vee q \vee \neg r) \supset u\ ,\ s \vee r,\ \neg p \vee u,\ q \supset u,\ s \supset u,\ \neg u\}$. Hãy xác định xem từ tập tiền đề đã cho có thể rút ra kết luận u hay không.

   e. Cho các tiền đề $\{(p \,\&\, q) \supset r,\ \neg r \,\&\, (s \vee r),\ \neg p \,\&\, s,\ (\neg q \vee s) \supset u,\ q\}$. Hãy xác định xem từ tập tiền đề đã cho có thể rút ra kết luận $u \vee \neg q$ hay không.

   f. Cho các tiền đề $\{(p \,\&\, q \,\&\, \neg r) \supset (u \vee r),\ s \vee r \vee \neg p \vee u,\ (q \supset u) \,\&(s \supset u),\ \neg u\}$. Hãy xác định xem từ tập tiền đề đã cho có thể rút ra kết luận $u \vee r$ hay không.

Chương 3 LOGIC VỊ TỪ

Khi xây dựng các chuỗi suy diễn hoặc phép chứng minh, logic mệnh đề không xét cấu trúc bên trong (chẳng hạn như cấu trúc chủ từ - vị từ) của các mệnh đề đơn. Cho mệnh đề đơn "Mọi loài chim đều biết bay", khi đó logic mệnh đề ký hiệu mệnh đề này bằng chữ cái nào đó, chẳng hạn *p*, sau đó coi p như không có cấu trúc, nghĩa là không hề tìm hiểu cấu trúc bên trong của *p*, và, tất nhiên, không hề sử dụng thông tin chứa trong cấu trúc đó. Thế nhưng có những suy luận đòi hỏi nhất thiết phải sử dụng đến cấu trúc bên trong của các mệnh đề. Ví dụ, cho suy luận:

Thịt của tất cả các loài vật bốn chân đều ăn được, bò là loài vật bốn chân, vậy thịt bò ăn được.

Với logic mệnh đề, ta được suy luận $p, q \rightarrow r$.

Tuy nhiên công thức biểu thị suy luận này, $(p \,\&\, q) \supset r$ lại không phải là công phán đoán hằng đúng. Từ đây logic mệnh đề cho rằng suy luận đã cho sai. Thế nhưng đây lại là một suy luận hoàn toàn đúng !

Tính đúng đắn của suy luận vừa nêu không chỉ dựa trên phụ thuộc hàm giữa các giá trị chân lý của các mệnh đề thành phần trong suy luận, mà còn dựa trên cấu trúc bên trong của các mệnh đề đó.

Logic vị từ là hệ logic nghiên cứu những suy luận như vậy. Nó là sự mở rộng logic mệnh đề.

I. Ngôn ngữ logic vị từ

Logic vị từ sử dụng ngôn ngữ hình thức cùng tên. Việc hiểu và dịch câu của ngôn ngữ tự nhiên sang ngôn ngữ logic vị từ dựa trên sự phân tích ngôn ngữ tự nhiên. Vì vậy, trước hết chúng ta tiến hành phân tích ngôn ngữ tự nhiên.

1. Phân tích ngôn ngữ tự nhiên

Ngôn ngữ tự nhiên là ngôn ngữ của các dân tộc, ví dụ như tiếng Việt, tiếng Anh, tiếng Pháp,… Các ngôn ngữ này hình thành dần dần trong lịch sử một cách tự nhiên, thông qua hoạt động nhận thức và cải tạo thực tiễn của các dân tộc. Các ngôn ngữ tự nhiên hình thành và phát triển một cách tự phát – nghĩa là ngôn ngữ tự nhiên không phải là kết qủa hoạt động tự giác nhằm tạo ra chúng của một người hay một nhóm người nào đó. Các quy tắc hình thành ngôn ngữ tự nhiên, chẳng hạn quy tắc ngữ pháp, cú pháp ,… vì thế nhiều khi không được xác định ở dạng tường minh.

a) Các tính chất cơ bản của ngôn ngữ tự nhiên

i) Đa nghĩa. Một từ hoặc một cụm từ (từ đây về sau ta sẽ gọi ngắn gọn là một *biểu thức ngôn ngữ*) trong ngôn ngữ tự nhiên có thể có nhiều nghĩa khác nhau, tùy thuộc vào ngữ cảnh trong đó nó được sử dụng.

Ví dụ 1: Từ "ngày mai" có thể được hiểu là tương lai, mà cũng có thể được hiểu là ngày hôm sau.

Ví dụ 2: Trong câu "Diêu bông hỡi diêu bông sao em nỡ vội lấy chồng" (Lời bài hát "Ngẫu hứng Lá Diêu Bông" của Trần Tiến) "Diêu bông" có thể hiểu là "Em", mà cũng có thể hiểu là một thán từ, kiểu than "Trời ơi!".

Tính đa nghĩa là một tính chất rất đáng quý của ngôn ngữ trong giao tiếp hàng ngày, trong văn học và nghệ thuật. Tuy nhiên tính chất này lại gây ra khá nhiều khó khăn cho việc sử dụng ngôn ngữ tự nhiên trong khoa học, kỹ thuật, luật pháp, … - những lĩnh vực có đòi hỏi đầu tiên là trình bày vấn đề một cách rõ ràng, chính xác, tránh hiểu nhầm.

ii*) Giàu khả năng biểu đạt.* Tất cả các ngôn ngữ tự nhiên đều rất giàu khả năng biểu đạt. Người ta có thể dùng ngôn ngữ tự nhiên trong rất nhiều lĩnh vực. Có thể dùng chúng để trò chuyện, trao đổi thường ngày; có thể dùng chúng để làm thơ, viết văn, để bàn luận về thời sự, về chính trị, về luật pháp; có thể dùng chúng để nghiên cứu và trình bày các tư tưởng và công trình khoa học,… Ngoài ra, với ngôn ngữ tự nhiên, cùng một sự vật hoặc hiện tượng có thể được mô tả, được biểu đạt bằng các cách khác nhau, bằng các biểu thức ngôn ngữ khác nhau. Ví dụ: Các cụm từ "Lên xe hoa", "Đi lấy chồng",… biểu thị cùng một sự việc. Các cụm từ như "Chào đời", "Ra đời",… cũng biểu thị cùng một sự việc.

iii) Đóng về ngữ nghĩa. Trong ngôn ngữ tự nhiên vừa có bộ phận từ và câu nói về các đối tượng bên ngoài ngôn ngữ, nói về thế giới bên ngoài ngôn ngữ, ví dụ, nói về thời tiết, về kinh tế, về các vật dụng, … và có cả những bộ phận từ và câu nói về các đối tượng của bản thân ngôn ngữ, ví dụ, nói về ngữ pháp, về cú pháp, về danh từ, động từ, câu, … Sự có mặt của cả hai thành phần như vậy trong ngôn ngữ được gọi là tính đóng về ngữ nghĩa của nó. Tính chất này chính là các nguyên nhân gây nên các nghịch lý về ngữ nghĩa như nghịch lý kẻ nói dối sau đây. Có người nói rằng anh ta đang nói dối. Ta cần xác định xem lúc nói như vậy là anh ta đang nói dối hay đang nói thật. Nếu như khi nói như vậy anh ta đang nói thật thì hóa ra anh ta nói thật rằng mình đang nói dối, và nghĩa là anh ta đang nói dối ! Ngược lại, nếu khi đó anh ta đang nói dối thì có nghĩa là anh ta đang nói dối rằng mình đang nói dối. Nhưng như thế lại có nghĩa là trên thực tế anh ta đang nói thật ! Như vậy không thể nói rằng anh ta đang nói dối và cũng không thể khẳng định rằng anh ta đang nói thật. Ta có nghịch lý ở đây vì một câu nói khẳng định về tính đúng sai của chính nó. Rõ ràng là điều này chỉ có thể xảy ra đối với các ngôn ngữ đóng về ngữ nghĩa.

iv) Có nhiều cấp độ ngôn ngữ. Trong cùng một đoạn văn hoặc một câu của ngôn ngữ tự nhiên từ ngữ có thể thuộc về nhiều cấp độ khác nhau. Chẳng hạn, trong câu nói của Socrate "Tôi chỉ biết rằng mình không biết gì" hai lần xuất hiện của từ "biết" thuộc về hai cấp độ ngôn ngữ khác nhau. Từ "biết" thứ hai là biết về toàn bộ thế giới khách quan, ngoại trừ về khả năng hiểu biết của chính mình, nó thuộc cấp độ thứ nhất. Từ "biết" thứ nhất lại thuộc cấp độ thứ hai, biết về khả năng hiểu biết của mình, nghĩa là biết về cái biết thuộc cấp độ thứ nhất. Nếu không phân biệt các cấp độ ngôn ngữ khác nhau như vậy thì ta sẽ cho rằng đây là câu nói chứa đựng nghịch lý.

v) Một phần thông tin không được biểu đạt tường minh. Thông tin chứa đựng trong các câu, các đoạn văn trong ngôn ngữ tự nhiên có thể chỉ có một phần được biểu đạt dưới dạng tường minh, còn phần khác được ngầm hiểu. Ví dụ: câu "Trở về nhà, anh ta lục tung căn phòng của mình để tìm tấm ảnh" chứa đựng những thông tin không được biểu thị tường minh như : anh ta mới đi đâu đó; có tấm ảnh. Ví dụ khác: "Con chó này chỉ có hai chân" có một thông tin được ngầm hiểu là : bình thường chó có nhiều hơn hai chân. Phần thông tin được biểu đạt tường minh ta gọi là *hiển ngôn*, phần thông tin không được biểu đạt tường minh gọi là *hàm ngôn*. Hàm ngôn có thể là *tiền giả định* hay *hàm ý* [2]. Để suy luận đúng đắn ta cần phải xác định được toàn bộ nội dung thông tin mà câu hoặc đoạn văn chứa, cả hiển ngôn và hàm ngôn.

Như đã nói, ngôn ngữ tự nhiên rất thuận tiện cho quá trình trao đổi trong cuộc sống hàng ngày. Nó cũng rất thuận lợi cho các hoạt động văn học nghệ thuật. Tuy nhiên, nếu dùng ngôn ngữ tự nhiên để nghiên cứu và trình bày các vấn đề khoa học kỹ thuật thì ta gặp phải nhiều khó khăn vì tính đa nghĩa của nó. Thêm vào đó, vì ngôn ngữ tự nhiên đóng về ngữ nghĩa nên nó có thể chứa các nghịch lý. Điều này khiến ta không thể dùng nó để xây dựng các lý thuyết khoa học chặt chẽ bởi lẽ khoa học không được phép chứa đựng các nghịch lý.

Những lý do nêu trên buộc các nhà khoa học phải sáng tạo ra ngôn ngữ hình thức để giải quyết các vấn đề của mình. Ngôn ngữ hình thức là ngôn ngữ được người ta tạo ra một cách tự giác để làm công cụ giải quyết những vấn đề nhất định nào đó (chủ yếu là của khoa học và kỹ thuật). Các quy tắc xây dựng ngôn ngữ hình thức, tỉ như quy tắc cú pháp, … được xác định ngay từ đầu ở dạng tường minh.

b) Một số loại ký hiệu và phạm trù ngữ nghĩa của ngôn ngữ tự nhiên

**) Tên gọi. Hằng đối tượng*

Tên gọi là từ hay cụm từ dùng để chỉ, thay thế, đại diện cho một đối tượng hoặc tập hợp đối tượng nào đó trong giao tiếp ngôn ngữ.

Ví dụ, từ "sinh viên" trong giao tiếp ngôn ngữ dùng thay thế, đại diện cho tập hợp học sinh đại học và cao đẳng – "sinh viên" là tên của tập hợp đó. "Hồ Chí Minh" là tên của người sáng lập ra Nước Việt Nam Dân Chủ Cộng Hòa, và tên này được dùng thay, dùng đại diện cho Người trong giao tiếp ngôn ngữ.

Tên có thể chia thành tên chung và tên riêng. Tên riêng là tên chỉ một đối tượng đơn lẻ nào đó, tên chung là tên chỉ một tập hợp đối tượng. Ví dụ, tên "Trường Đại học khoa học xã hội và nhân văn thành phố Hồ Chí Minh" là một tên riêng, còn tên "Học sinh đại học" lại là một tên chung.

Cũng có thể chia tên gọi thành tên đơn và tên phức (hay còn gọi là tên mô tả). Tên đơn là tên không được tạo thành từ những tên khác. Ví dụ, "Việt Nam", "Sông Lam", "học sinh", … là những tên đơn. Tên phức, hay tên mô tả, là tên được tạo thành từ nhiều tên khác. Ví dụ, "con sông lớn nhất Việt Nam" là một tên phức, nó được tạo thành từ các tên "con sông", "Việt Nam".

Tên gọi là một ký hiệu, và cũng như mọi ký hiệu khác, tên gọi có hai đặc trưng quan trọng là *nghĩa thực (denotation)*, hay còn gọi là *sự biểu hiện*[1], và *ngữ nghĩa*, hay còn gọi đơn giản là *nghĩa*.

Nghĩa thực của tên là đối tượng hay tập hợp đối tượng mà tên đó chỉ. " Sự biểu hiện của một từ ngữ là thuộc loại tất cả những sự vật có thật hay đang tồn tại mà từ ấy đã thích nghi một cách đúng đắn … Một từ ngữ không chỉ ra một cái gì có thật là mang sự biểu hiện số không …"[2]. Ví dụ, tên "Thành phố Hồ Chí Minh" có nghĩa thực, hay sự biểu hiện, là thành phố lớn nhất Việt Nam.

Tên có thể có hoặc không có nghĩa thực. Các tên "Số tự nhiên lớn nhất", "Hình vuông tròn"[3], "Vua hiện nay của nước Pháp",… không chỉ bất cứ một đối tượng nào trên thực tế nên không có nghĩa thực. Còn các tên như "Mặt trời", "Thái bình dương" chỉ những đối tượng tồn tại trên thực tế nên có nghĩa thực. Nhiều tên khác nhau có thể có cùng một nghĩa thực. Ví dụ, các tên "Sao Hôm" và "Sao Mai" cùng chỉ một hành tinh nên có cùng một nghĩa thực; các tên "Logic học" và "Môn khoa học nghiên cứu các hình thức và quy luật của tư duy" chỉ cùng một bộ môn khoa học nên có cùng một nghĩa thực. Trong ngôn ngữ tự nhiên, vì tính đa nghĩa nên một tên có thể có nhiều nghĩa thực khác nhau. Ví dụ, tên "Vật chất" có nghĩa thực là thực tại khách quan được đưa lại cho con người trong cảm giác (nếu hiểu theo nghĩa triết học), lại cũng có nghĩa thực là các vật thể cụ thể (nếu hiểu theo nghĩa vật lý).

Ngữ nghĩa của tên là toàn bộ những thông tin có trong tên, nhờ đó mà có thể xác định được nghĩa thực của nó. Theo Frege thì nghĩa của tên là cái chứa đựng các phương thức hiện ra của đối tượng. Tên có thể không có nghĩa thực, nhưng bao giờ cũng có ngữ nghĩa. Chúng ta thấy các câu chứa tên không có cái biểu hiện vẫn có ý nghĩa là bởi vì các tên đó vẫn có nghĩa. Hai tên có cùng cái biểu hiện có thể chứa những thông tin khác nhau và vì vậy có nghĩa khác nhau. Ví dụ, đối với một người không am tường địa lý thì các câu "SEA Games 19 được tổ chức tại Jakarta" và "SEA Games 22 được tổ chức tại Thủ đô nước Philippin" chứa những thông tin hoàn toàn khác nhau vì các tên "Manila" và "Thủ đô nước Philippin" chứa các thông tin khác nhau.

Các ngôn ngữ hình thức thường được xây dựng sao cho ngữ nghĩa của tên xác định duy nhất nghĩa thực của tên, tuy nhiên điều ngược lại không bắt buộc phải có.

Trong các ngôn ngữ hình thức, việc sử dụng tên phải tuân theo ba quy tắc sau đây:

---

[1] Xem, ví dụ, Hoàng Trinh *Từ ký hiệu học đến thi pháp học*, Đà Nẵng, 1997, trang 39-41.

[2] C. Lewis, dẫn theo Hoàng Trinh, Sđd, tr. 40.

[3] Một số tác giả cho rằng nếu cụm từ không chỉ đối tượng nào trên thực tế thì nó không phải là tên. Xem, ví dụ B. Russell *"Quán từ mô tả (description)"* trong sách "Cái mới trong ngôn ngữ học nước ngoài", cuốn 13, Moskva, 1982, tiếng Nga.

Quy tắc hướng đối tượng. Khi sử dụng một tên là ta muốn nói đến đối tượng mà tên đó chỉ, nghĩa là muốn nói đến nghĩa thực của nó, chứ không phải là muốn nói đến bản thân cái tên.

Ví dụ, nói "Hà Nội là thành phố nằm trên bờ sông Hồng" là ta muốn nói về Thủ đô của nước ta, chứ không muốn nói đến bản thân cái tên "Hà Nội".

Quy tắc có nghĩa thực duy nhất. Mỗi tên chỉ được chỉ một đối tượng hoặc một tập hợp đối tượng duy nhất, nghĩa là chỉ được quyền có một nghĩa thực duy nhất.

Tính đa nghĩa của ngôn ngữ tự nhiên làm cho nó không tuân theo quy tắc này.

Quy tắc thay thế. Hai tên có cùng nghĩa thực phải thay thế được cho nhau trong mọi trường hợp.

Trong ngôn ngữ tự nhiên các tên có cùng nghĩa thực có thể thay thế được cho nhau trong một số trường hợp và không thể thay thế cho nhau trong một số trường hợp khác. Ví dụ, tên "Sao Hôm" thay thế được cho tên "Sao Mai" trong câu "Sao Mai là một ngôi sao rất sáng" (khi thay ta được câu "Sao Hôm là một ngôi sao rất sáng"), nhưng không thể thay thế được cho nó trong câu "Ông cha ta không biết rằng Sao Hôm chính là Sao Mai" (khi thay ta được câu "Ông cha ta không biết rằng Sao Hôm chính là Sao Hôm"!).

Hằng đối tượng là biểu thức ngôn ngữ chỉ một đối tượng nào đó không đổi trong suốt qúa trình tư duy được khảo sát. Trong ngôn ngữ tự nhiên hằng đối tượng thông thường là tên riêng. Ví dụ, "Hoa hồng" là một hằng đối tượng trong câu "Hoa hồng đẹp"; "Thỏ" là hằng đối tượng trong câu "Thỏ là một loài gặm nhấm".

**). Biến đối tượng. Hàm đối tượng.*

Biến đối tượng là một biểu thức ngôn ngữ chạy trên tập hợp các đối tượng, nghĩa là có thể nhận những giá trị là các đối tượng khác nhau. Biến đối tượng có thể coi là sự khái quát hóa của khái niệm biến số trong toán học. Trong ngôn ngữ tự nhiên các biến đối tượng không được biểu thị một cách tường minh, mà thường không được tách riêng khỏi biểu thức ngôn ngữ biểu thị tập hợp các đối tượng mà chúng có thể nhận giá trị.

Hàm đối tượng là một biểu thức ngôn ngữ (thường là một tên chung) mà khi dùng kết hợp với một hoặc một số hằng đối tượng thì xác định một hằng đối tượng khác. Hàm đối tượng còn được dùng cặp với các biến đối tượng. Hàm đối tượng dùng cặp với n biến hoặc hằng đối tượng thì gọi là hàm n ngôi. Ta có thể coi khái niệm hàm đối tượng là sự khái quát hóa của khái niệm hàm số trong toán học.

Ví dụ: Biểu thức "Đại học Quốc gia" là một hàm đối tượng. Khi kết hợp nó với hằng đối tượng "Thành phố Hồ Chí Minh", ta được hằng đối tượng mới là "Đại học Quốc gia thành phố Hồ Chí Minh", còn nếu kết hợp nó với hằng đối tượng "Hà Nội" ta lại được hằng đối tượng mới là "Đại học Quốc gia Hà Nội".

**). Vị từ (predicate).* Đó là những biểu thức ngôn ngữ biểu thị một tính chất nào đó ở một đối tượng hoặc biểu thị một mối quan hệ nào đó giữa một số đối tượng.

Ví dụ: Trong câu “Logic học là một khoa học quy phạm” thì cụm từ “khoa học quy phạm” thể hiện một tính chất của logic học, như vậy nó là một vị từ. Trong câu “5 lớn hơn 3” cụm từ “lớn hơn” biểu thị một quan hệ giữa các đối tượng 5 và 3, vậy nó cũng là một vị từ.

Vị từ chỉ tính chất gọi là vị từ một ngôi, vị từ chỉ mối quan hệ giữa n đối tượng gọi là vị từ n ngôi.

**). Lượng từ (quantifier) và các liên từ logic. Lượng* từ là những từ chỉ đặc trưng về lượng của câu như : tất cả, mọi, tồn tại, một số, có những, đa số, thiểu số, … và những từ hoặc cấu trúc ngôn ngữ tương đương. “Lượng từ là các tác tử trỏ lượng tác động lên các đối mà nó chi phối”[4]. Ở đây các đối là các biến hoặc hằng đối tượng. Các liên từ logic là các từ như : *và*, *hay là*, *hoặc là* , *nếu … thì …*, *kéo theo*, *khi và chỉ khi*, *tương đương*, *không là*, *không phải là*, … và những từ hoặc cấu trúc ngôn ngữ tương đương với chúng.

Lưu ý. Khái niệm lượng từ mà ta dùng ở đây không phải là khái niệm số từ mà ta dùng thường ngày. Ví dụ, không có lượng từ trong câu: *Dải Ngân Hà có khoảng 400 tỉ ngôi sao.*

**). Mệnh đề đơn (proposition).* Mệnh đề là biểu thức ngôn ngữ có giá trị đúng hoặc sai. Mệnh đề đơn là biểu thức ngôn ngữ khẳng định hay phủ định một tính chất nhất định ở một đối tượng, hoặc khẳng định hay phủ định một mối quan hệ nhất định giữa một số đối tượng nào đó. Mệnh đề đơn là mệnh đề mà bất cứ thành phần nào của nó cũng không phải là mệnh đề.

Ví dụ, câu “Mọi số chẵn đều chia hết cho 2” là một mệnh đề đơn. Câu “Nếu số a chẵn thì số a chia hết cho 2” không phải là mệnh đề đơn, vì thành phần “số a chẵn” của nó đã là một mệnh đề đơn.

Cần lưu ý rằng trong ngôn ngữ tự nhiên một biểu thức ngôn ngữ xác định có thể là hằng đối tượng, là biến đối tượng, là hàm đối tượng hoặc là vị từ, tùy thuộc vào ngữ cảnh. Ta xét một số ví dụ phân tích về mặt logic các biểu thức ngôn ngữ tự nhiên:

Ví dụ 1. Sinh viên học môn logic.
Trong câu này “sinh viên” là tên chung, tên đơn, và là hằng đối tượng. “Học môn logic” là vị từ.

Ví dụ 2. Vợ nhà thơ Tú Xương là một người phụ nữ rất đảm đang.
Trong câu này “nhà thơ Tú Xương”, “vợ nhà thơ Tú Xương” là các hằng đối tượng; “là một người phụ nữ rất đảm đang” là vị từ một ngôi chỉ tính chất; “vợ” là hàm đối tượng.

Ví dụ 3. Mọi sinh viên đều học môn logic.
Ở đây “sinh viên” và “môn logic” không phải là các hằng đối tượng. Trong ví dụ 1 “sinh viên” là hằng đối tượng, vì nó chỉ một tập hợp đối tượng mà ta coi như một đối

[4] Nguyễn Đức Dân, “Lôgích và tiếng Việt”, NXB Giáo dục, 1996, tr.71.

tượng, và đối tượng đó xác định, không thay đổi trong qúa trình tư duy ta đang xét. "Sinh viên" trong ví dụ 3 có vai trò khác hẳn. Ở đây nó không chỉ một đối tượng cụ thể, mà có thể chỉ bất cứ đối tượng nào từ tập hợp sinh viên vì đi sau lượng từ "mọi". Vì vậy, "sinh viên" ở đây là một biến đối tượng. Hơn nữa, biến đối tượng này chỉ xác định trên tập sinh viên, nghĩa là các đối tượng mà biến này có thể nhận giá trị đều có tính chất "sinh viên". Bởi vậy, "sinh viên" trong ví dụ này còn là một vị từ chỉ tính chất.

Ví dụ 4. $3 + 4 = 7$.

Ở đây "3", "4", "7" là các hằng đối tượng; "=" là vị từ hai ngôi, "+" (chính xác hơn là "… + …") là hàm đối tượng hai ngôi, và vì vậy "3 + 4" cũng là hằng đối tượng.

f. Liên từ logic. Có thể kết nối hai hoặc nhiều mệnh đề đơn lại với nhau nhờ những từ gọi là liên từ logic, kết quả việc kết nối đó gọi là mệnh đề phức hợp. Đó thông thường là những từ và cụm từ « và », « hoăc là », « hay là », « nếu … thì … », « tương đương », « khi và chỉ khi », « không phải là », và những cụm từ hay từ tương đương khác.

2. Hệ ký tự

- $p, q, r, s, p_1, p_2, \ldots$ — Các ký tự chỉ mệnh đề đơn;
- $a, b, c, d, a_1, a_2, \ldots$ — Các ký tự chỉ hằng đối tượng;
- $x, y, z, u, v, w, x_1, x_2, \ldots$ — Biến đối tượng;
- $f, g, h, f_1, f_2, \ldots$ — Các ký tự chỉ hàm đối tượng;
- $\neg, \vee, \&, \supset, \equiv$ — Các liên từ (phép toán) logic;
- $\forall, \exists$ — Các lượng từ;
- $(,), \ldots$ — Các dấu kỹ thuật.

3. Hạn từ (term)

- Hạn từ trong ngôn ngữ logic vị từ có vai trò tương tự như danh từ hoặc cụm từ đóng vai trò danh từ trong ngôn ngữ tự nhiên, nó được định nghĩa đệ quy như sau:
- Các ký tự chỉ hằng và biến đối tượng là các hạn từ ;
- Nếu $t_1, t_2, \ldots, t_k$ là các hạn từ, f^k là hàm đối tượng k ngôi (hàm k biến, k đối), thì $f^k(t_1, t_2, \ldots, t_k)$ là hạn từ;
- Ngoài ra không còn hạn từ nào khác.

4. Công thức (WFF – Well Formed Formula)

Công thức trong ngôn ngữ logic vị từ có vai trò tương tự như câu (hay mệnh đề) trong ngôn ngữ tự nhiên, công thức cũng được định nghĩa đệ quy:

- Các ký tự chỉ mệnh đề đơn là công thức;
- Nếu P_k là vị từ k ngôi, $t_1, t_2, \ldots, t_k$ là các hạn từ, thì $P_k(t_1, t_2, \ldots, t_k)$ là công thức (gọi là công thức nguyên tử – atom);

- Nếu A và B là các công thức thì (A), (B), $\neg$ A, $\neg$ B, A $\vee$ B, A & B, A $\supset$ B, A $\equiv$ B là các công thức;
- Nếu A là công thức chứa biến đối tượng x (khi đó ta viết $A(x)$) thì $\forall x\ A$, $\exists x\ A$ (hay viết $\forall x\ A(x)$, $\exists x\ A(x)$) là các công thức;
- Ngoài ra không còn công thức nào khác.

5. *Các ví dụ*

a) Ví dụ hạn từ (term):

- Cho f là hàm một ngôi, x là biến đối tượng. Khi đó $f(x)$ là hạn từ. Nếu a là hằng đối tượng thì $f(a)$ cũng là hạn từ.
- Giả sử f là hàm một ngôi, g là hàm hai ngôi, t_1 và t_2 là hai hạn từ. Khi đó:
  - t_1, t_2 là hạn từ;
  - $g(t_1, t_2)$ là hạn từ;
  - $f(t_1), f(t_2)$ là hạn từ;
  - $f(g(t_1, t_2))$ là hạn từ;
  - $g(f(t_1), g(f(t_2), x))$ là hạn từ.
- a, b là các hằng đối tượng, bởi vậy là hạn từ;
- x là biến đối tượng, vậy x là hạn từ;
- $f(a, b)$ là hạn từ;
- $f(g(x), c)$ là hạn từ;
- *Các biểu thức sau đây không phải là hạn từ :*
  - $f(a, f(b));$
  - $a + x;$
  - $P(f(x));$
  - $f(P(a));$
  - $\forall x P(x);$ ...

b) Ví dụ công thức

- $p\ \&\ (q \vee r);$
- $\exists x\ Q(x) \supset P(a)$
- $p\ \&\ \forall x\ R(x);$
- $\forall x\ \exists y\ (P(x) \supset Q(y))$
- $\forall x\ (p\ \&\ R(x));$
- $\exists x\ P2(x, a)\ \&\ \forall x\ Q(x).$
- *Các biểu thức sau đây không phải là công thức :*
  - $P\ \&\ Q;$
  - $P(P(a));$
  - $P(P(x, a));$
  - $f(P(a));$
  - $R \vee Q(a, b, x);$

- $Q(a, b, c) \supset f(a, b, c)$;

Ngôn ngữ logic vị từ mà ta vừa xác định, như đã thấy, rất đơn giản, nhưng khả năng biểu đạt của nó, tuy không thể sánh được với ngôn ngữ tự nhiên, vẫn rất lớn. Nếu như không tồn tại một tiêu chuẩn cú pháp hình thức nào để xác định một biểu thức trong ngôn ngữ tự nhiên có phải là một câu hay không, thì trong ngôn ngữ logic vị từ ta thấy rõ có thể xác định một cách dễ dàng một biểu thức ngôn ngữ nào đó có phải là công thức hay không. Cũng tương tự như vậy với danh từ hoặc cụm từ đóng vai trò danh từ trong ngôn ngữ tự nhiên và hạn từ trong ngôn ngữ logic vị từ. Chính vì vậy, việc sử dụng ngôn ngữ logic vị từ thay cho ngôn ngữ tự nhiên trong nhiều trường hợp (đặc biệt là trong các hệ thống hình thức, hệ thống máy móc) thuận tiện hơn rất nhiều.

6. Biểu thị tư tưởng bằng ngôn ngữ logic vị từ

Các phán đoán và suy luận thông thường bây giờ có thể được viết dưới dạng các công thức trong ngôn ngữ logic vị từ. Việc này có ý nghĩa rất lớn, vì nó giúp xác định rõ ràng, chính xác ý nghĩa của các phán đoán và suy luận, tránh được sự hiểu lầm, mập mờ hoặc nhiều nghĩa của câu. Hơn thế nữa, khi đã biểu thị tư tưởng, suy luận, v.v. , ta có thể sử dụng logic vị từ để kiểm tra được tính đúng đắn của các suy luận.

Muốn vậy, trước hết phải "dịch" các suy luận từ ngôn ngữ thông thường sang ngôn ngữ logic vị từ. Cấu trúc các câu trong ngôn ngữ tự nhiên vô cùng phong phú, vì vậy không có các quy tắc chung bao quát được tất cả các trường hợp cần dịch. Sau đây chúng tôi nêu một số quy tắc hướng dẫn dịch một số dạng câu. Lưu ý rằng các hướng dẫn này chưa bao quát hết mọi trường hợp cần dịch, và ngay cả các dạng câu được đề cập cũng không loại trừ các trường hợp ngoại lệ.

Phương pháp dịch câu (mệnh đề) từ ngôn ngữ tự nhiên sang ngôn ngữ logic vị từ

Với mệnh đề đơn cần thực hiện các bước sau :

- *Phân tích câu để xác định vị từ và các hạn từ tương ứng với nó. Nếu một hạn từ được cấu thành từ một hàm đối tượng và một số hạn từ khác thì nó được biểu diễn bằng cách viết hàm đối tượng trước, sau đó liệt kê vào trong cặp ngoặc đơn mở đóng các hạn từ tương ứng, nếu số này nhiều thì dùng dấu phẩy để ngăn cách chúng.*

- *Viết vị từ, liệt kê các hạn từ tương ứng vào trong cặp ngoặc đơn để ngay sau vị từ. Nếu có nhiều hạn từ thì dùng dấu phẩy để phân cách chúng. Ta sẽ gọi cách biểu thị câu như thế này là cách viết vị từ, hay dạng vị từ của câu.*

- *Thay thế vị từ và các hạn từ trong cách viết vị từ bằng các ký hiệu tương ứng quy định trong phần hệ ký tự của ngôn ngữ logic vị từ.*

Ví dụ : Cho mệnh đề *"Mẹ Mai là bác sĩ"*. Trước hết, cần phân tích câu để xác định các thành phần ngữ nghĩa của nó. Rõ ràng câu này là câu đơn. Ở đây *"Mẹ"* là hàm đối tượng, *"Mai"* là hằng đối tượng, nên *"Mẹ(Mai)"* là hạn từ ; *"là bác sĩ"* là vị từ (tính chất *"là bác sĩ"* và tính chất *"bác sĩ"* như nhau, nên về sau ta sẽ lược bỏ *"là"*, ta cũng lược bỏ như vậy với các vị từ khác). Vị từ *"bác sĩ"* tương ứng với hạn từ

"Mẹ(Mai)". Vậy mệnh đề ban đầu được viết ở dạng vị từ thành *"bác sĩ (Mẹ(Mai))"*. Thay vị từ *"bác sĩ"*, hàm đối tượng *"Mẹ"* và hằng đối tượng *"Mai"* bằng các ký hiệu được phép như quy định trong hệ ký tự của ngôn ngữ logic vị từ. Kết quả ta được công thức tương đương mệnh đề đã cho : $P(f(a))$.

Với mệnh đề được tạo thành từ hai hoặc nhiều mệnh đề đơn, ta thực hiện các bước :

- *Xác định các mệnh đề đơn thành phần;*
- *Dịch riêng từng mệnh đề đơn thành phần. Lưu ý, các vị từ, hằng, hàm đối tượng xuất hiện trong nhiều mệnh đề đơn thành phần phải được thay thế bằng các ký tự giống nhau của ngôn ngữ logic vị từ;*
- *Dùng các dấu liên từ logic thay cho các cụm từ tương ứng để nối các mệnh đề đơn thành phần với nhau.*

Ví dụ, cho mệnh đề *"Hằng là sinh viên và Hằng với Mai là chị em"*. Ở đây có hai mệnh đề đơn thành phần *"Hằng là sinh viên"*, *"Hằng với Mai là chị em"*. Dịch riêng chúng, ta được các công thức $P(a)$, $Q(a, b)$. Nối chúng với nhau bằng dấu *&* - dấu tương ứng với liên từ *"và"*, ta được công thức biểu diễn mệnh đề đã cho ban đầu : $P(a) \,\&\, Q(a,b)$.

Với mệnh đề phổ quát đơn giản :

- *Chuyển câu về một trong hai dạng "Mọi S là P" hoặc "Mọi S không là P".*
- *Mọi S là P* dịch thành $\forall x(S(x) \supset P(x))$.
- *Mọi S không là P* dịch thành $\forall x(S(x) \supset \neg P(x))$

Ví dụ, mệnh đề *"Mọi sinh viên đều học logic"* tương đương với mệnh đề *"Mọi sinh viên đều là người học logic"*. Mệnh đề này có dạng *"Mọi S là P"*, trong đó *S = "Sinh viên", P = "người học logic"*. Vậy nó được dịch sang ngôn ngữ logic vị từ thành công thức $\forall x(S(x) \supset P(x))$.

Với mệnh đề bộ phận đơn giản :

- *Chuyển câu về thành một trong hai dạng "Một số S là P" hoặc "Một số S không là P".*
- *Một số S là P* dịch thành $\exists x(S(x) \,\&\, P(x))$.
- *Một số S không là P* dịch thành $\exists x(S(x) \,\&\, \neg P(x))$

Ví dụ. Câu *"Một số loài chim di cư về Phương Nam"* tương đương với câu *"Một số loài chim là loài di cư về Phương Nam"*[3]. Nó có dạng *"Một số S là P"*, với *S = "loài chim"*, *P = "loài di cư về Phương Nam"*. Vậy công thức tương ứng là $\exists x(S(x) \,\&\, P(x))$.

Sau đây ta xét thêm một số ví dụ.

Mệnh đề đơn

Ví dụ 1 *Thỏ là một loài gặm nhấm.*

"Thỏ" – hằng đối tượng, ta ký hiệu là a; "là một loài gặm nhấm" – vị từ một ngôi, ta ký hiệu là P. Kết qủa: $P(a)$.

Ví dụ 2 *Hằng cao hơn Mai.*

"Hằng" và *"Mai"* – các hằng đối tượng, ta ký hiệu tương ứng là a và b; "cao hơn" – vị từ hai ngôi, ta ký hiệu là P. Kết qủa: $P(a,b)$.

Ví dụ 3 *Hằng cao bằng chị của Mai.*

"Hằng" và *"Mai"* – các hằng đối tượng, ta ký hiệu tương ứng là a và b; "chị" – hàm đối tượng, ta ký hiệu là f ; "cao bằng" – vị từ hai ngôi, ta ký hiệu là P. Kết qủa:

$P(a, f(b))$.

Nếu trong câu này ta lấy các hằng đối tượng *"Hằng"* và *"chị của Mai",* ký hiệu chúng là a và c, thì kết qủa là: $P(a,c)$.

Mệnh đề có một lượng từ.

Ví dụ 4 *Mọi sinh viên đều học môn logic.*

"Mọi" – lượng từ, ký hiệu $\forall$; *"sinh viên"* – biến đối tượng, ký hiệu x; *"sinh viên"* – vị từ một ngôi, ký hiệu P; *"học môn logic"* – vị từ, ký hiệu Q. Kết qủa:

$\forall x\ (P(x) \supset Q(x))$.

Ví dụ 5 *Một số sinh viên học ngành tin học.*

"Một số" - lượng từ, ta ký hiệu $\exists$; *"sinh viên"* – biến đối tượng, ta ký hiệu x; *"sinh viên"* – vị từ một ngôi, ký hiệu P; "học ngành tin học" – vị từ, ký hiệu Q. Kết qủa:

$\exists x\ (P(x)\ \&\ Q(x))$.

Ví dụ 6 *Mọi sinh viên học giỏi toán đều học giỏi logic.*

"Mọi" – lượng từ, ký hiệu $\forall$; *"sinh viên học giỏi toán"* – biến đối tượng, ký hiệu x; *"sinh viên"* – vị từ một ngôi, ký hiệu P; *"học giỏi toán"* – vị từ, ký hiệu Q; *"học giỏi logic"* – vị từ, ký hiệu R. Kết qủa:

$\forall x\ ((P(x)\ \&\ Q(x)) \supset R(x))$.

Mệnh đề có nhiều lượng từ.

Ví dụ 7 *Mọi người đều có người để yêu mến.*

"Mọi" – lượng từ, ký hiệu $\forall$; *"người"* – biến đối tượng, ký hiệu x; *"người"* – vị từ một ngôi, ký hiệu P; *"có"* – lượng từ $\exists$, *"người"* – biến đối tượng, ký hiệu y ;*"yêu mến"* – vị từ hai ngôi, ký hiệu Q. Kết qủa:

$\forall x\ (P(x) \supset \exists y\ (P(y)\ \&\ Q(x,y)))$

Nếu chỉ đề cập đến những con người, và vì vậy không sợ nhầm lẫn thì các thành phần P(x), P(y) trong công thức này không cần thiết. Khi đó có thể viết đơn giản :

$\forall x\ \exists y\ Q(x,y)$.

Ví dụ 8 *Có người mà mọi người đều yêu mến.*

Phân tích tương tự câu trên, kết qủa:

$\exists y\ (P(y)\ \&\ \forall x\ (P\ (x) \supset Q(x,y)))$

Nếu không sợ nhầm lẫn vì đang chỉ đề cập đến con người thì ta có thể viết câu này đơn giản:

$\exists y\ \forall x\ Q(x,y)$.

Ví dụ 9 *Nếu Nam là sinh viên tin học thì Nam học môn logic.*

Nam là sinh viên tin học: $P(a)$;
Nam học môn logic: $Q(a)$;
Liên từ *"nếu ... thì ..."*: $\supset$
Kết quả: $P(a) \supset Q(a)$.

Ví dụ 10 *Một số sinh viên được học bổng, một số sinh viên không được.*

Một số sinh viên được học bổng: $\exists x\ (P(x)\ \&\ Q(x))$;
Một số sinh viên không được học bổng $\exists y\ (P(y)\ \&\ \neg Q(y))$;
Dấu phẩy: $\&$
Kết quả: $\exists x\ (P(x)\ \&\ Q(x))\ \&\exists y\ (P(y)\ \&\ \neg Q(y))$;

Nếu chỉ sử dụng cách viết của ngôn ngữ logic vị từ mà không thay thế các hằng và hàm đối tượng, các vị từ bằng ký hiệu, vẫn giữ nguyên chúng ở dạng ngôn ngữ tự nhiên thì ta có ngôn ngữ logic vị từ ứng dụng.

Trong tin học ngôn ngữ logic vị từ được sử dụng rất rộng rãi. Nó được sử dụng để biểu thị tri thức trong các hệ chuyên gia hoặc trí tuệ nhân tạo, dạng tương tự với nó được dùng làm ngôn ngữ hỏi trong các hệ quản trị cơ sở dữ liệu, người ta cũng dùng một phần đặc biệt của ngôn ngữ này làm ngôn ngữ lập trình thuận tiện cho lĩnh vực trí tuệ nhân tạo (ngôn ngữ Prolog), …

Ví dụ: Chuyển sang ngôn ngữ của logic vị từ ứng dụng các mệnh đề sau.
(i). *Mọi loài chim đều biết bay*.
Trong câu này *"Mọi"* là lượng từ. *"loài chim"* vừa là biến đối tượng (ký hiệu *x*), vừa là vị từ tương ứng với *x*. *"biết bay"* là vị từ tương ứng với *x*.
Vậy công thức tương ứng trong ngôn ngữ logic vị từ ứng dụng sẽ là :

$\forall x(\text{loàichim}(x) \supset \text{biết bay}(x))$

(ii) Công thức ở ví dụ 10 trên đây có thể viết thành :

$\exists x\ (\text{sinh viên}(x)\ \&\ \text{đượchọcbổng}(x))\ \&\exists y\ (\text{sinhviên}(y)\ \&\ \neg\ \text{đượchọcbổng}(y))$;

Viết dưới dạng này công thức trở nên dễ hiểu hơn. Công thức này có thể đọc là: *"với mọi x, nếu x là chim thì x biết bay"*.

7. Biến tự do và biến buộc

Trong biểu thức $\forall x\ A(x)$, $A(x)$ gọi là vùng tác động của của lượng từ $\forall x$. Nếu biến *x* xuất hiện trong một vùng tác động của lượng từ $\forall x$ (trong một công thức lượng từ $\forall x$ có thể xuất hiện nhiều lần, và vì thế có thể có nhiều vùng tác động khác nhau của $\forall x$

trong một công thức) thì lần xuất hiện đó của x được gọi là xuất hiện không tự do (còn gọi là buộc). Ngược lại thì gọi là xuất hiện tự do. Một biến có thể xuất hiện tự do trong công thức, có thể xuất hiện không tự do trong công thức, và có thể vừa xuất hiện tự do, vừa xuất hiện không tự do trong cùng một công thức.

Với lượng từ $\exists$x (tồn tại) cũng hoàn toàn tương tự. Chính xác hơn, nếu ở những điều vừa nói trên đây về sự xuất hiện tự do và buộc của biến trong công thức mà ta thay lượng từ $\forall$x (với mọi x) bằng lượng từ $\exists$x (tồn tại), thì những điều đó vẫn đúng.

Ví dụ về sự xuất hiện tự do và xuất hiện buộc của biến.

Trong công thức

$$\forall x \ (P(x) \supset P(y)) \ \& \ P(a)$$

xuất hiện của biến x là buộc, còn biến y xuất hiện tự do.

Trong công thức

$$\forall x \ (P(x, y) \supset \exists y \ (Q(y, x)))$$

cả hai lần xuất hiện của x đều là xuất hiện buộc, biến y vừa xuất hiện tự do (lần đầu), vừa xuất hiện buộc (lần sau), vì lần xuất hiện đầu của biến y nằm ngoài miền tác động của các lượng từ $\forall y$ và $\exists y$, còn lần xuất hiện thứ hai, vì nằm trong vùng tác động của lượng từ $\exists$y nên là xuất hiện buộc.

Biến x *tự do* trong công thức nếu nó có xuất hiện tự do trong công thức. Nếu x không có xuất hiện tự do trong công thức, nghĩa là mọi xuất hiện của nó trong công thức đều là xuất hiện buộc thì x là *biến buộc* trong công thức đó.

Ví dụ, biến x tự do và biến y là biến buộc trong công thức sau đây :

$$\forall y(P(x, y) \supset \exists x Q(y,x))$$

Giả sử $x_1, x_2, \ldots, x_k$ là các biến, A – là công thức. Không quan tâm đến việc trong công thức A các biến đó tự do hay là biến buộc và ngoài ra có còn các biến tự do khác hay không, ta ký hiệu công thức A bằng $A(x_1, x_2, \ldots, x_k)$ để sau đó có thể ký hiệu kết quả phép thế các hạn từ $t_1, t_2, \ldots, t_k$ tương ứng vào các chỗ xuất hiện tự do (nếu có) của các biến $x_1, x_2, \ldots, x_k$ là $A(t_1, t_2, \ldots, t_k)$.

Hạn từ t gọi là *tự do đối với biến x* trong công thức A, nếu như không một lần xuất hiện tự do nào của biến x trong A nằm vào vùng tác động của lượng từ $\forall y$, $\exists y$ với y là biến có trong t.

Ví dụ.

- Hạn từ x_1 tự do đối với x_2 trong công thức $A(x_2)$, vì trong công thức này xuất hiện tự do duy nhất của x_2 không nằm trong miền tác động của bất kỳ lượng từ nào của biến x_1- biến duy nhất trong hạn từ x_1.
- Hạn từ $f(x_1, x_3)$ tự do đối với x_1 trong công thức $\forall x_3 \ \exists x_1 \ (A_1(x_2, x_3) \ \& \ A_3(x_1, x_2)$, vì trong công thức này không có xuất hiện tự do nào của biến x_1.
- Hạn từ $f(x_1, x_3)$ không tự do đối với x_1 trong công thức $\exists x_3 \ (A_1(x_1, x_3) \supset A_2(x_2, x_3))$, vì xuất hiện tự do của x1 trong công thức này nằm trong miền tác động của lượng từ $\exists x_3$, mà x_3 là một biến trong hạn từ $f(x_1, x_3)$.

Từ định nghĩa ta dễ dàng rút ra :

(a) Mọi biến đều tự do đối với chính nó trong mọi công thức.

(b) Nếu hạn từ không chứa biến thì hạn từ đó tự do đối với mọi biến trong mọi công thức;

(c) Hạn từ *t* tự do đối với mọi biến trong công thức *A*, nếu như các biến của *t* không có xuất hiện buộc trong *A*,

(d) x_i tự do đối với x_j trong công thức *A*, nếu trong A không có xuất hiện tự do nào của x_j.

II. Diễn giải (Interpretation). Mô hình (Model)

1. Diễn giải

Diễn giải và *mô hình* là những khái niệm rất quan trọng trong logic vị từ. Để bạn đọc dễ nắm được những khái niệm này, trước hết chúng tôi trình bày một cách phi hình thức nội dung của khái niệm diễn giải.

Các công thức chỉ có nghĩa khi có một sự giải thích, giải nghĩa nào đó các thuật ngữ của nó, diễn giải các ký tự, các từ có ở trong công thức đó. Chẳng hạn, xét công thức $P(a) \supset (Q(a) \,\&\, Q(b))$. Vì đây là công thức của ngôn ngữ logic vị từ, nên chúng ta biết rằng *a, b* là những ký tự hằng đối tượng, nghĩa là chúng chỉ những đối tượng nào đó; *P* và *Q* là các ký tự vị từ một ngôi, nghĩa là chúng chỉ đến những tính chất nào đó của đối tượng *a* và đối tượng *b*. Tuy nhiên, chúng ta vẫn không biết công thức này nói gì. Ý nghĩa của công thức đã cho chỉ sáng tỏ sau khi biết *P* và *Q* chỉ những tính chất nào, *a, b* chỉ những đối tượng nào. Và chỉ sau khi xác định được ý nghĩa của công thức ta mới có thể xác định nó đúng hay sai. Nói cách khác, một công thức có thể được giải nghĩa có nghĩa là nó có thể nhận được *giá trị chân lý*. Như vậy, *diễn giải* một công thức là cho biết các ký tự trong công thức đó như các ký tự vị từ, ký tự hàm đối tượng, ký tự biến đối tượng chỉ những tính chất (hay quan hệ) nào, những hàm đối tượng và đối tượng nào. Thao tác cho biết này chúng ta sẽ gọi là *gán*. Cho biết *a* chỉ đối tượng *Mặt trời* có nghĩa là gán *Mặt trời* cho *a*. Ngoài ra, nếu trong công thức có chứa các biến đối tượng thì cũng phải cho biết các biến đối tượng đó nhận giá trị từ miền nào, hay chỉ đối tượng nào. Mặc dù khi diễn giải công thức ta có thể gán các đối tượng tùy ý cho các ký tự hằng đối tượng, gán các quan hệ hoặc tính chất tùy ý cho các ký tự vị từ , … nhưng diễn giải công thức không phải hoàn toàn tùy tiện, mà phải thỏa mãn một số đòi hỏi nhất định. Nếu trong công thức có nhiều ký tự hằng đối tượng khác nhau thì có thể diễn giải rằng các hằng đối tượng đó chỉ cùng một đối tượng, nhưng nếu một ký tự hằng đối tượng xuất hiện nhiều lần trong công thức thì không thể diễn giải khác nhau các lần xuất hiện đó. Trong công thức đã khảo sát trên đây hai ký tự *a* và *b* có thể được gán hai đối tượng khác nhau hoặc được gán cùng một đối tượng; nhưng ký tự hằng đối tượng *a* xuất hiện hai lần, ta không thể diễn giải rằng hai lần xuất hiện đó của *a* chỉ hai đối tượng khác nhau. Cũng hoàn toàn tương tự với các loại ký tự khác trong công thức như ký tự vị từ, biến đối tượng, hàm đối tượng. Diễn giải thoả mãn được đòi hỏi như vậy gọi là diễn giải có tính hệ thống. Nếu diễn giải nhiều công thức cùng lúc thì tính hệ thống phải được mở rộng cho tập hợp các công thức đó. Cụ thể, các xuất hiện của cùng một ký tự trong các công thức khác nhau phải cùng được gán cùng một đối tượng hay cùng một quan hệ, … như nhau. Diễn giải phải mang tính hệ thống.

Như ta thấy, công thức của logic vị từ không chỉ được tạo thành từ các công thức con, mà còn được tạo thành từ các hạn từ. Vì vậy, để diễn giải công thức, ta cần giải nghĩa các hạn từ của nó. Có thể hiểu hạn từ một cách trực quan nhất như là một *đối tượng*. Như vậy, diễn giải phải xác định, phải đặc tả một tập hợp đối tượng nhất định, gọi là *miền xác định của diễn giải*, hay ngắn gọn hơn là *miền diễn giải*.

Bây giờ chúng ta xem xét khái niệm diễn giải một cách chặt chẽ, chính xác hơn. Người đầu tiên đưa ra khái niệm diễn giải chính xác đối với ngôn ngữ logic vị từ là nhà logic người Mỹ gốc Ba lan Tarski.

Diễn giải là một bộ bốn $< D, \Pi, \Phi, \Psi, \nu >$, trong đó:

- D là một tập hợp không rỗng, gọi là *miền diễn giải*.
- Π là một hàm đặt tương ứng với mỗi ký tự vị từ P^k một quan hệ k-ngôi trên D, nghĩa là đặt tương ứng với mỗi ký tự vị từ P^k một tập con của tập D^k (xin nhắc lại là mỗi quan hệ n ngôi trên tập D là một tập hợp con của tập D^n. Với D^n là tích Descartes $D \times D \times \ldots \times D$ (n lần)).
- Φ là một hàm xác định một tương quan nào đó giữa các hằng đối tượng *a, b, c, ...* các với các phần tử của D : mỗi hằng đối tượng *a, b, c, ...* tương ứng với một phần tử của D. Nói cách khác, Φ là mộp phép gán giá trị cho các hằng đối tượng.
- Ψ là một hàm xác định một tương quan nào đó giữa tập các ký tự hàm f^i, g^j, … với các phép toán i , j ngôi , … trên D: mỗi hàm f^i tương ứng với một phép toán trên tập D có i ngôi.

  Lưu ý : người ta cũng có thể kết hợp hai hàm Φ và Ψ thành một hàm duy nhất, khi đó người ta coi hằng đối tượng là hàm đối tượng có số ngôi bằng không (hàm không biến).
- ν là một phép gán giá trị cho các biến (variable assignment), nghĩa là một ánh xạ từ tập các biến đối tượng lên tập D.

2. Giá trị chân lý của công thức trong diễn giải

Trên cơ sở diễn giải $I = < D, \Pi, \Phi, \Psi, \nu >$ có thể xác định một hàm I đặt tương ứng cho mỗi công thức A một giá trị chân lý $I(A)$ và đặt tương ứng với mỗi hạn từ t một phần tử $I(t)$ của miền D như sau.

- Nếu x là biến thì I(x) = ν(x).
- Nếu a là một hằng đối tượng thì I(a) = Φ(a).
- Nếu f là ký tự hàm đối tượng k ngôi, t1, t2, …, tk là các hạn từ thì
- I(f(t1, t2, …, tk) i2Ψ (f)(I(t1), I(t2), …, I(tk)).
- Nếu P là ký tự vị từ n ngôi, t1, t2, …, tn là các hạn từ thì I(P(t1, t2, …, tn)) đúng (ký hiệu I(P(t1, t2, …, tn)) = T) khi và chỉ khi bộ <I(t1), I(t2), …, I(tn)> là một phần tử của tập hợp Π(P); và I(P(t1, t2, …, tn)) sai khi và chỉ khi bộ <I(t1), I(t2), …, I(tn)> không phải là một phần tử của tập hợp Π(P).

- Nếu A và B là các công thức thì :
  - $I(\neg A)$ đúng khi và chỉ khi $I(A)$ sai.
  - $I(A\&B)$ đúng khi và chỉ khi $I(A)$ và $I(B)$ cùng đúng.
  - $I(A\vee B)$ đúng khi và chỉ khi $I(A)$ đúng hay $I(B)$ đúng.
  - $I(A\supset B)$ đúng khi và chỉ khi $I(A)$ sai hay $I(B)$ đúng.
  - $I(A\equiv B)$ đúng khi và chỉ khi cả $I(A)$ và $I(B)$ cùng đúng hoặc cả $I(A)$ và $I(B)$ cùng sai.

Các lượng từ $\forall$ và $\exists$ được diễn giải như cách hiểu thông thường. Điều đó được biểu đạt chặt chẽ như sau. Để xác định *I* cho các công thức chứa lượng từ chúng ta sử dụng thêm một khái niệm mới. Nếu *I* là một diễn giải, $I = < D_I, \Pi_I, \Phi_I, \Psi_I, \nu_I>$, *d* là một phần tử của D_I, còn *x* là một biến đối tượng, khi đó $I_{x/d}$ là ký hiệu cho diễn giải mới J, trong đó x đã được gán giá trị mới là d, còn các biến đối tượng khác vẫn được gán giá trị như trước. Cụ thể là $J = < D_J, \Pi_J, \Phi_J, \Psi_j, \nu_J>$, trong đó $D_J = D_I$, $\Pi_J = \Pi_I$, $\Phi_J = \Phi_I$, $\Psi_j = \Psi_I$, $\nu_J(x) = d$, $\nu_J(y) = \nu_J(y)$ vơi mọi biến tự do *y* khác *x*. Như vậy *J* là một diễn giải giống như *I* ở mọi điểm, ngoại trừ (có thể) cách gán giá trị cho biến *x* : *J* gán cho *x* giá trị *d*, còn *I* có thể gán *d* cho *x*, cũng có thể gán giá trị khác cho *x*.

- Nếu A là công thức và x là biến đối tượng thì I($\forall$xA) đúng khi và chỉ khi Ix/d(A) đúng với mọi d $\in$ D.
- Nếu A là công thức và x là biến đối tượng thì I($\exists$xA) đúng khi và chỉ khi Ix/d(A) đúng ít nhất với một phần tử d của D.

Nếu *A* là một công thức và *I* là một diễn giải, thì ta ký hiệu *A đúng trong I* và *A sai trong I* (hay *A không đúng trong I*) lần lượt là $I(A) = T$ và $I(A) = F$.

Ví dụ 1. Cho công thức *P(a,f(b,c))*.
Giả sử *I* là diễn giải, $I = < D, \Pi, \Phi, \Psi, \nu>$. Trong đó *D* là tập các số tự nhiên, Π đặt tương ứng *P* với quan hệ *bằng nhau* quen thuộc, trong công thức này không có các ký tự vị từ khác nên không cần phần còn lại của Π (về sau ta xác định Π bằng cách xác định các quan hệ ứng với các vị từ trong công thức, chẳng hạn như với công thức đang xét ta xác định Π bằng cách nói rằng *P* là "bằng nhau". Ta cũng xác định Φ và Ψ bằng cách tương tự): a là số *10, b* là số *5, c* là số 5, *f* là *phép cộng* theo nghĩa của toán học. Trong công thức không có biến nên không cần xác định ν. Với diễn giải *I* công thức đang xét có nghĩa *là "10 = 5 + 5"*. Vì quan hệ bằng nhau và phép toán cộng hiểu như trong toán học, nên mệnh đề này đúng trong *I*.
Để ý rằng trong diễn giải này các ký tự *b* và *c* được gán cùng một đối tượng - số *5*.
Nếu J là diễn giải chỉ khác *I* ở chỗ *J* gán cho *f* phép toán nhân thì với *J* công thức đang xét có nghĩa là *"10 = 5 * 5"*. Như vậy, trong diễn giải *J* nó sai.
Như vậy một công thức có thể trở thành mệnh đề đúng trong diễn giải này và sai trong diễn giải khác.
Ví dụ 2. Cho công thức *R(a, f(x))*.

Giả sử diễn giải $I = < D, \Pi, \Phi, \Psi, \nu >$ được xác định như sau[5] :

D là một tập hợp người, $D = \{Hải, Minh, Hạnh, Thảo\};$

$\Phi(a) = Hải;$

$\Psi(f)(Hải) = Minh;$

$\Psi(f)(Minh) = Hạnh;$

$\Psi(f)(Hạnh) = Thảo;$

$\Psi(f)(Thảo) = Thảo;$

$\Pi(R) = \{<Hải, Minh>, <Hải, Thảo>, <Hạnh, Thảo>, <Minh, Thảo>\}$

$\nu(x) = Hạnh.$

Với diễn giải I ta có :

$I(a) = \Phi(a) = Hải;$

$I(f(x)) = \Psi(f)(\nu(x)) = \Psi(f)(Hạnh) = Thảo.$

Vì cặp $<Hải, Thảo>$ là một phần tử của $\Pi(R)$, nên $I(R(a, f(x)))$ đúng, nghĩa là công thức $R(a, f(x))$ đúng trong I.

Cặp $<Thảo, Hải>$ không thuộc $\Pi(R)$, nên $I(R(f(x),a))$ sai, tức là công thức $R(f(x),a)$ không đúng, sai trong I.

Ta xét thêm tính đúng sai của các công thức $\exists x(R(f(a), x)$ và $\forall x(R(a, x))$ trong I.

Với I ta có :

$I(a) = \Phi(a) = Hải;$

$I(f(a)) = \Psi(f)(\Psi(a)) = \Psi(f)(Hải) = Minh;$

$I(x) = \nu(x) = Hạnh.$

Vì vậy, với $I_{x/\ Thảo} = J = < D_J, \Pi_J, \Phi_J, \Psi_j, \nu_J >$ ta có :

$I_{x/\ Thảo}(f(a)) = I(f(a)) = Minh;$

$I_{x/\ Thảo}(\mathrm{x}) = Thảo;$

$\Pi_J(R) = \Pi(R) = \{<Hải, Minh>, <Hải, Thảo>, <Hạnh, Thảo>, <Minh, Thảo>\}$

Rõ ràng cặp $<Minh, Thảo>$ là một phần tử của $\Pi_J(R)$, nên $I_{x/\ Thảo}(R(f(a), x))$ đúng, nghĩa là $R(f(a), x)$ đúng trong $I_{x/\ Thảo}$.

Vậy $\exists x(R(f(a), x)$ đúng trong I.

Trong khi đó, với $I_{x/Thảo} = K = < D_K, \Pi_K, \Phi_K, \Psi_k, \nu_K >$ ta có :

$I_{x/\ Hạnh}(a) = I(a) = Hải;$

$I_{x/\ Hạnh}(\mathrm{x}) = Hạnh;$

$\Pi_K(R) = \Pi(R) = \{<Hải, Minh>, <Hải, Thảo>, <Hạnh, Thảo>, <Minh, Thảo>\}$

Rõ ràng cặp $<Hải, Hạnh>$ không phải là phần tử của $\Pi_J(R)$, nên $J(R(a, x))$ sai, $I_{x/Hạnh}(R(a, x))$ sai, nghĩa là $R(a, x)$ sai trong $I_{x/Hạnh}$.

Vậy $\forall x(R(a, x)$ sai trong I.

Ví dụ 3. Cho công thức $\forall \mathrm{x}(\mathrm{x}, \mathrm{f}(\mathrm{x}))$.

---

[5] Ở đây chỉ xác định phần D, Π, Φ, Ψ, ν tương ứng với các ký tự trong công thức đang xét.

Giả sử diễn giải U = $< D, \Pi, \Phi, \Psi, \nu>$ được xác định như sau :

$D = \{\alpha, \beta, \gamma\};$

$\Psi(f)(\alpha) = \beta ;$

$\Psi(f)(\beta) = \gamma ;$

$\Psi(f)(\gamma) = \alpha ;$

$\Pi (R) = \{<\alpha, \alpha >, <\alpha, \beta >, <\beta, \gamma >, <\gamma, \gamma >, <\gamma, \alpha >\};$

$\nu(x) = \alpha .$

Với U ta có :

$U(x) = \nu(x) = \alpha ;$

$U(f(x)) = \Psi(f)(U(x)) = \Psi(f)(\alpha) = \beta ;$

Vì vậy : Với $U_{x/\alpha} = J$ ta có

$J(x) = U(x) = \alpha ;$

$J(f(x)) = U(f(x)) = \Psi_J(f)(U_{x/\alpha} (x)) = \Psi_J(f)(\nu_J (x)) = \Psi_J(f)(\alpha) = \beta ;$

$\Pi_J (R) = \Pi (R) = \{<\alpha, \alpha >, <\alpha, \beta >, <\beta, \gamma >, <\gamma, \gamma >, <\gamma, \alpha >\};$

$<\alpha, \beta> \in \Pi_J (R)$

Từ đây $U_{x/\alpha} (R(x, f(x))) = J(R(x, f(x)))$ đúng.

Hoàn toàn tương tự, ta xác định được rằng công thức *R(x, f(x))* cũng đúng trong các diễn giải $U_{x/\beta}$ và $U_{x/\gamma}$.

Miền diễn giải D_U của U chỉ có các đối tượng α, β, γ. Như thế *U(R(x, f(x)))* đúng, hay *R(x, f(x)) đúng trong U.*

Nhận xét về variable assignment ν **:** Ví dụ 2 và ví dụ 3 trên đây cho thấy mặc dù hàm ν được xác định trên toàn bộ tập hợp biến, nhưng khi xét tính đúng sai của một công thức trong một diễn giải thì ν chỉ có ý nghĩa khi công thức đó chứa biến tự do (như trong ví dụ 2), và hoàn toàn không có ý nghĩa nào khi công thức không chứa biến tự do (như trong các công thức ở ví dụ 3). Sở dĩ như vậy là do nếu trong công thức biến *x* chỉ xuất hiện buộc thì khi xét tính đúng sai của nó trong giải nghĩa có variable assignment ν ta chỉ cần xét đến các diễn giải có variable assignment có thể khác ν ở giá trị của *x*. Chính vì vậy đôi khi người ta còn nói đến ν như là phép gán giá trị cho các biến tự do.

Từ định nghĩa diễn giải và định nghĩa về giá trị chân lý của công thức trong diễn giải có thể rút ra các khẳng định:

Định lý 3.1.

(1) Không công thức nào vừa đúng vừa sai trong cùng một diễn giải.

(2) Nếu trong diễn giải I các công thức A và $A \supset B$ *đúng, thì công thức B cũng đúng trong I.*

(3) Nếu $\forall xA$ *đúng trong diễn giải I thì* $\exists x$ *cũng đúng trong I.*

(4) Nếu $\forall xA$ *đúng trong diễn giải I thì A cũng đúng trong I.*

(5) Nếu A đúng trong diễn giải I thì $\exists xA$ *cũng đúng trong I.*

3. Mô hình (model). Quy luật logic

Mô hình của một công thức là diễn giải trong đó công thức ấy đúng. Diễn giải U trong ví dụ 3 trên đây là mô hình của công thức $R(x, f(x))$. Diễn giải I trong ví dụ 2 trên đây không phải là mô hình của công thức $\forall x R(a,x)$.
Không phải công thức nào cũng có mô hình. Trái lại, có những công thức mà diễn giải nào cũng là mô hình của chúng.

Công thức được gọi là *hằng đúng (*hay *quy luật logic)* nếu nó đúng trong mọi diễn giải, và được gọi là *hằng sai (*hay *mâu thuẫn logic)* nếu nó không đúng trong bất cứ một diễn giải nào. Như vậy công thức là hằng đúng nếu như mọi diễn giải đều là mô hình của nó; và công thức là hằng sai, nếu nó không có mô hình nào. Công thức hằng đúng A được ký hiệu là $\models A$. Công thức đúng trong một số diễn giải, sai trong một số diễn giải khác gọi là công thức *trung hòa*.

Dễ dàng chứng minh được những khẳng định sau đây :

(1) $\models A$ khi và chỉ khi $\models \forall x\, A$.
(2) $\models (A \vee \neg A)$.
(3) Công thức $A \,\&\, \neg A$ là mâu thuẫn logic.
(4) $\models (\forall x A \supset A)$
(5) $\models (A \supset \exists x A)$
(6) $\models \exists x A \equiv \neg (\forall x \neg A)$
(7) $\models \forall x A \equiv \neg (\exists x \neg A)$

Chứng minh khẳng định (1).
Từ trái qua phải. Giả sử $\models A$, nghĩa là $I(A) = T$ với mọi diễn giải I. Giả sử $\forall x\, A$ không phải là công thức hằng đúng. Khi đó có diễn giải, tạm gọi là J, sao cho $J(\forall x A) = F$. Điều này có nghĩa là có diễn giải $J_{x/d}$ sao cho $J_{x/d}(A) = F$. Nhưng điều này mâu thuẫn với điều giả sử ban đầu. Vậy, với mọi diễn giải J ta có $J(\forall x A) = T$, nghĩa là $\models \forall x A$.
Chiều ngược lại. Giả sử $\models \forall x A$, nghĩa là $I(\forall x A) = T$ với diễn giải I tùy ý. Giả sử variable assignment của I là v_I, và $v_I(x) = a$. Khi đó $I = I_{x/a}$. Nhưng $I(\forall x A) = T$ lại có nghĩa là $I_{x/d}(A) = T$ với mọi $d \in D_I$, từ đây $I_{x/a}(A) = T = I(A)$. A đúng trong diễn giải I bất kỳ, vậy $\models$ A. Chứng minh xong.

Chứng minh khẳng định (2) :
Với diễn giải I tùy ý A đúng trong I hoặc A sai trong I. Nếu A đúng trong I thì theo định nghĩa diễn giải, $A \vee \neg A$ đúng trong I. Ngược lại, nếu A sai trong I thì theo định nghĩa diễn giải, $\neg A$ đúng trong I, vì thế $A \vee \neg A$ cũng đúng trong I. Như thế $A \vee \neg A$ luôn luôn đúng trong I. Vậy, $A \vee \neg A$ là công thức hằng đúng.

Chứng minh khẳng định (6) :

Ta cần chỉ ra rằng với mọi diễn giải I, nếu $\exists xA$ đúng trong I thì $\neg\forall x\neg A$ đúng trong I, và ngược lại, nếu $\neg\forall x\neg A$ đúng trong I thì $\exists xA$ đúng trong I.
Lấy diễn giải I tùy ý. Nếu $\exists xA$ đúng trong I thì, theo định nghĩa, ta có ít nhất một phần tử d thuộc miền diễn giải D của I sao cho A đúng trong $I_{x/d}$. Từ đây, $\neg A$ sai trong $I_{x/d}$. Vì thế $\forall x\neg A$ sai trong $I_{x/d}$. Vậy, theo định nghĩa, $\neg\forall x\neg A$ đúng trong I.
Ngược lại, Nếu $\neg\forall x\neg A$ đúng trong I thì $\forall x\neg A$ sai trong I. Nói cách khác, tồn tại phần tử c thuộc miền diễn giải D của I sao cho $\neg A$ sai trong $I_{x/c}$. Vì $\neg A$ sai trong $I_{x/c}$ nên A đúng trong $I_{x/c}$. Vậy, theo định nghĩa, $\exists xA$ đúng trong I.

Các khẳng định còn lại bạn đọc hãy tự chứng minh.

Để chứng tỏ rằng công thức là một quy luật logic, cần phải chỉ ra rằng nó đúng trong mọi diễn giải. Kiểm tra tính đúng sai của công thức trong mọi diễn giải là một việc nói chung rất khó khăn, vì có những diễn giải có miền diễn giải là tập hợp vô tận hoặc tập hợp gồm rất nhiều phần tử. Nhưng để chứng tỏ rằng một công thức không phải là quy luật logic thì chỉ cần đưa ra một diễn giải trong đó nó sai. Người ta thường sử dụng điều này để chứng tỏ công thức không phải là quy luật logic, đó chính là phương pháp dùng phản ví dụ.

Công thức *A kéo theo công thức B* (hay còn nói là *B là hệ quả logic của A*) nếu trong mọi diễn giải mà A đúng thì B cũng đúng. Tổng quát hơn, công thức B là hệ quả của tập hợp công thức Γ (ký hiệu $\Gamma \models B$), nếu như B đúng trong mỗi diễn giải mà tất cả các công thức từ tập Γ đúng.

III. Diễn giải Herbrand

Chúng ta đã thấy một công thức có thể đúng trong diễn giải này, và sai trong diễn giải khác. Với một diễn giải nhất định sẽ có những công thức đúng trong nó và các công thức khác sai trong nó. Rõ ràng hai diễn giải I và J khác nhau nếu tập hợp các công thức đúng trong I khác với tập hợp các công thức đúng trong J. Nói cách khác, I và J khác nhau nếu có một công thức A nào đó có giá trị chân lý trong I và trong J khác nhau. Tuy nhiên, nếu tập hợp công thức đúng trong diễn giải I và tập hợp công thức đúng trong diễn giải J như nhau thì cũng chưa thể khẳng định I trùng với J. Thật vậy, ví dụ, tập hợp công thức đúng trong hai diễn giải I và J sau đây là như nhau:

Diễn giải $I = \langle D_I, \Pi_I, \Phi_I, \Psi_I, v_I \rangle$

$D_I = \{Nam, Bình\}$

$\Phi_{iI}(a) = Nam$

$\Phi_{iI}(b) = Bình$

$\Psi_{iI}(f)(Nam) = Nam$

$\Psi_{iI}(f)(Bình) = Nam$

$\Pi_{iI}(P) = \{\langle Nam \rangle, \langle Bình \rangle\}$

$\Pi_{iI}(Q) = \Pi_{iI}(R) = \Pi_{iI}(S) = \ldots = \{\ \}$

$v_I(x) = v_I(y) = v_I(z) = \ldots = Nam$

Diễn giải $J = < D_J, \Pi_J, \Phi_J, \nu_J >$

$D_J = \{Nam, Bình\}$

$\Phi_J(a) = Bình$

$\Psi_J(b) = Nam$

$\Psi_J(f)(Nam) = Bình$

$\Psi_J(f)(Bình) = Bình$

$\Pi_J(P) = \{<Nam>, <Bình>\}$

$\Pi_J(Q) = \Pi_J(R) = \Pi_J(S) = ... = \{ \}$

$\nu_J(x) = \nu_J(y) = \nu_J(z) = ... = Bình$

Vì rõ ràng $\Phi_I \neq \Phi_J$ và $\nu_I \neq \nu_J$, nên I ≠ J.

Như vậy, bằng các tập hợp công thức chúng ta không phân biệt được các diễn giải với nhau.

Ta đã biết rằng theo định nghĩa, một công thức là quy luật logic khi và chỉ khi nó đúng trong mọi diễn giải. Vì vậy, một trong các phương pháp xác định xem công thức có phải là quy luật logic hay không là xét xem nó có đúng trong từng diễn giải hay không. Ta cũng đã biết rằng công thức B là hệ quả logic của tập hợp các công thức Γ khi và chỉ khi B đúng trong mọi diễn giải mà tất cả các công thức trong Γ đều đúng. Như vậy, để xác định xem B có phải là hệ quả logic của tập hợp công thức Γ hay không, phải xét hết các diễn giải mà trong đó tất cả các công thức từ Γ đều đúng. Sự kiện có những diễn giải mà tập hợp công thức đúng trong chúng như nhau gợi ý tưởng rằng khi xét xem công thức có phải là quy luật logic hay không ta không thật sự cần xét hết các diễn giải, mà chỉ cần xét cho mỗi nhóm diễn giải có chung tập hợp công thức đúng một đại diện mà thôi. Cũng vậy, để xét xem một công thức có là hệ quả logic của một tập hợp các công thức khác hay không ta cũng chỉ cần xét một số các diễn giải là đại diện cho các nhóm diễn giải trong đó tất cả các công thức trong tập hợp đã cho đều đúng. Chẳng hạn, chỉ cần xét một trong hai diễn giải I hoặc J trên đây là đủ thay vì xét cả hai. Nếu như ta có thể chia tập hợp tất cả các diễn giải thành từng nhóm có chung tập công thức đúng thì khi đó chỉ cần xét một phần tử nào đó bất kỳ cho mỗi nhóm như vậy mà thôi. Tuy nhiên điều này đòi hỏi trước hết phải tiến hành phân nhóm các diễn giải. Mà đây là một vấn đề vô cùng khó thực hiện. May mắn thay, có một hướng khác : trong một số trường hợp có thể xác định tất cả các đại diện cho các nhóm diễn giải có chung tập hợp công thức đúng (mặc dù chưa xác định được các nhóm như vậy), rồi chỉ xét các đại diện đó mà thôi. Các diễn giải đại diện như vậy là diễn giải herbrand.

1. *Miền herbrand*

Miền herbrand của công thức φ là tập hợp H_φ, nhỏ nhất trong số các tập hợp thỏa mãn các điều kiện sau đây :

- Mọi hằng đối tượng *a* có mặt trong công thức φ đều là phần tử của H_φ.
- Nếu φ không chứa ký tự hằng đối tượng nào thì H_φ chứa một hằng herbrand, ký hiệu là δ.

- Với mỗi ký tự hàm đối tượng n-ngôi f trong φ, nếu $h_1, h_2, ..., h_n$ đều là phần tử của H_φ, thì $f(h_1, h_2, ..., h_n)$ cũng là phần tử của H_φ.

Các ví dụ sau đây giúp chúng ta hiểu rõ hơn về miền herbrand.

Ví dụ 1. Với công thức P(x)
Miền herbrand là tập hợp gồm duy nhất một phần tử $\{\delta\}$.

Ví dụ 2. Với công thức R(a, b)
Miền herbrand là tập hợp hai phần tử $\{a, b\}$.

Ví dụ 3. Với công thức $P(f(x))$
Miền herbrand là tập hợp vô hạn phần tử $\{\delta, f(\delta), f(f(\delta)), f(f(f(\delta))), ...\}$.

- *Ví dụ 4.* Với công thức $P(f(a))$
Miền herbrand là tập hợp vô hạn phần tử $\{a, f(a), f(f(a)), f(f(f(a))), ...\}$.

Ví dụ 5. Với công thức $R(a, g(b))$
Miền herbrand là tập hợp vô hạn phần tử $\{a, b, g(a), g(b), g(g(a)), g(g(b)), ...\}$.

Lưu ý : Các phần tử của miền herbrand chỉ đơn thuần là các cấu trúc cú pháp, không phải là các đối tượng hiện thực cụ thể. Nếu công thức không chứa hằng đối tượng thì miền herbrand tương ứng chỉ gồm một phần tử δ duy nhất. Nếu công thức không chứa các hàm đối tượng và chứa *n* hằng đối tượng khác nhau thì miền herbrand của nó gồm *n* phần tử. Nếu công thức có chứa hàm đối tượng thì miền herbrand của nó là một tập hợp vô hạn.

2. Định nghĩa diễn giải herbrand và ví dụ

Diễn giải herbrand của công thức φ là diễn giải $I = < D, \Pi, \Phi, \Psi, \nu>$, thỏa mãn các điều kiện sau đây :

- Miền diễn giải của I là miền herbrand Hφ .
- Φ đặt mỗi hằng đối tượng a tương ứng với chính nó.
- Ψ đặt mỗi hàm đối tượng n-ngôi f tương ứng với hàm F xác định như sau :

$$F : \quad H_\varphi^n \rightarrow H_\varphi$$
$$<h_1, h_2, ..., h_n> \rightarrow F(h_1, h_2, ..., h_n)$$

Sau đây là một số ví dụ diễn giải herbrand.

Ví dụ 1. Cho công thức $\varphi = P(x)$
Khi đó , vì φ không chứa hằng đối tượng, nên miền herbrand H_φ tương ứng là tập hợp $\{\delta\}$.

$D = H_\varphi = \{\delta\}$
$\Pi(P) = \{\delta\}$
$\nu(x) = \delta$

Một diễn giải herbrand khác của công thức đã cho là :

$D = H_\varphi = \{\delta\}$

$\Pi(P) = \{\ \}$

$\nu(x) = \delta$.

Ngoài ra công thức đã cho không còn diễn giải herbrand nào khác.

Ví dụ 2. Cho công thức $\varphi = P(a)$

Khi đó miền herbrand H_φ tương ứng là tập hợp $\{a\}$.

$D = H_\varphi = \{a\}$

$\Pi(P) = \{<a>\}$

$\nu(x) = a$

Một diễn giải herbrand khác của công thức đã cho là :

$D = H_\varphi = \{a\}$

$\Pi(P) = \{\ \}$

$\nu(x) = a$.

Ngoài ra công thức đã cho không còn diễn giải herbrand nào khác.

Ví dụ 3. Cho công thức $\varphi = R(a, b)$

Khi đó miền herbrand H_φ tương ứng là tập hợp $\{a, b\}$.

$D = H_\varphi = \{a, b\}$

$\Pi(R) = \{<a,a, a>, <a,b, b>, <b, a, b>\}$

$\nu(x) = a$

Một diễn giải herbrand khác của công thức đã cho là :

$D = H_\varphi = \{a, b\}$

$\Pi(P) = \{<a, a, a>, <b, b, b>\}$

$\nu(x) = a$

Công thức này còn nhiều diễn giải herbrand khác nữa. Số các diễn giải herbrand ở đây là hữu hạn.

Ví dụ 4. Xét công thức $P(a) \vee Q(f(x))$.

Miền herbrand tương ứng với công thức này là tập hợp vô hạn $\{a, f(a), f(f(a)), \ldots\}$.

$D = \{a, f(a), f(f(a)), \ldots\}$.

$\Pi(P) = \{<a>,<f(f(a))>, <f(f(f(f(a))))>\}$

$\Pi(Q) = \{<f(a)>, <f(f(a)), <f(f(f(a)))>\}$

$\Psi(f)(a) = F(a) = a$

$\Psi(f)(f(a)) = F(f(a)) = f(a)$

$\Psi(f)(f(f(a))) = F(f(f(a))) = f(f(a))$

...

$\nu(x) = f(a)$

Vì miền herbrand của nó vô hạn, nên công thức này có vô số diễn giải herbrand.

Khái niệm miền herbrand và diễn giải herbrand có thể mở rộng ra cho tập hợp công thức bất kỳ như sau : Tạo công thức mới là hội của tất cả các công thức trong tập hợp. Khi đó miền herbrand và diễn giải herbran của tập hợp công thức chính là miền herbrand và diễn giải herbrand của công thức mới đã nêu.

3. Mô hình herbrand

Mô hình herbrand của một công thức là diễn giải herbrand trong đó công thức đúng. *Mô hình herbrand của một tập hợp công thức* là diễn giải herbrand của tập hợp đó, và mọi công thức từ tập hợp đều đúng trong nó.

IV. Hệ tiên đề của logic vị từ

Tương tự như logic mệnh đề, người ta cũng lập nên các hệ tiên đề cho logic vị từ. Logic vị từ bao hàm toàn bộ logic mệnh đề. Điều đó có thể thấy rõ khi toàn bộ các tiên đề của logic mệnh đề đều là tiên đề của logic vị từ, quy tắc của logic mệnh đề cũng là quy tắc của logic vị từ.

1. Các tiên đề và quy tắc

Với mọi công thức *A, B, C*, và biến *x* bất kỳ, các biểu thức *A1, A2, A3* sau đây là tiên đề của logic vị từ:

A1. $A \supset (B \supset A);$

A2. $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C));$

A3. $(\neg A \supset \neg B) \supset ((\neg A \supset B) \supset A);$

A4. $\forall x\, A(x) \supset A(t),$ với *A(x)* là công thức, *t* là hạn từ, tự do đối với *x* trong công thức *A(x).*

A5. $\forall x\, (A \supset B) \supset (A \supset \forall x\, B),$ nếu trong công thức *A* không có xuất hiện tự do của *x*.

Các quy tắc:

1) $MP \quad \dfrac{A \supset B,\ A}{B}$ (Modus ponens)

2) $Gen \quad \dfrac{A}{\forall xA}$ (Quy tắc tổng quát hóa)

Người ta thường ký hiệu quy tắc 1 là MP, và công thức 2 là Gen (từ từ tiếng Anh Generalization).

2. Chuỗi suy diễn, phép chứng minh

Các khái niệm này tương tự như trong logic mệnh đề. Cụ thể :

- Chuỗi suy diễn là một dãy hữu hạn các công thức kế tiếp nhau, trong đó mỗi công thức đều hoặc là một giả thiết òn gọi là tiền đề, tức là công thức cho trước), hoặc là một tiên đề, hoặc là nhận được từ các công thức đứng trước nó theo quy tắc MP hoặc Gen. Công thức cuối của chuỗi gọi là kết luận. Công thức kết luận là hệ quả của các giả thiết có trong chuỗi.
- Phép chứng minh là chuỗi suy diễn trong đó không có giả thiết nào. Công thức cuối cùng của một phép chứng minh được gọi là định lý. Với một nghĩa rộng hơn, người ta cũng gọi chuỗi suy diễn như đã định nghĩa trên đây là một phép chứng

minh. Trong trường hợp đó ta có phép chứng minh cho khẳng định rằng từ các giả thiết có trong suy chuỗi có thể rút ra được công thức kết luận của nó.

- Ta ký hiệu, - giống như trong logic mệnh đề - $B_1, B_2, ..., B_n \vdash A$, là "công thức A là hệ quả của các công thức $B_1, B_2, ..., B_n$ ". $\vdash A$ có nghĩa là A là định lý.
- Khi chứng minh được một định lý, ta có thể sử dụng nó trong các chuỗi suy diễn và phép chứng minh khác. Khi đó khái niệm chuỗi suy diễn được mở rộng như sau : Chuỗi suy diễn là một dãy hữu hạn các công thức kế tiếp nhau, trong đó mỗi công thức đều hoặc là một giả thiết (tức là công thức cho trước), hoặc là một tiên đề, hoặc là một định lý đã được chứng minh, hoặc là nhận được từ các công thức đứng trước nó theo quy tắc MP hoặc Gen. Khái niệm phép chứng minh được mở rộng tương ứng.

Ví dụ : dãy công thức sau đây là một chuỗi suy diễn

1.	$\forall x(P(x) \supset Q(x))$	*tiền đề*
2.	$\forall x(Q(x) \supset R(x))$	*tiền đề*
3.	$P(a)$	*tiền đề*
4.	$\forall x(P(x) \supset Q(x)) \supset (P(a) \supset Q(a))$	*tiên đề*
5.	$P(a) \supset Q(a)$	*1, 4, MP*
6.	$Q(a)$	*3, 5, MP*
7.	$\forall x(Q(x) \supset R(x)) \supset (Q(a) \supset R(a))$	*tiên đề*
8.	$Q(a) \supset R(a)$	*2, 7, MP*
9.	$R(a)$	*6, 8, MP*

Chuỗi suy diễn trên đây có kết luận là công thức số 9 - *R(a)*. Như vậy *R(a)* là hệ quả của các công thức 1, 2,và 3. Chuỗi này không phải là một phép chứng minh, vì nó có các tiền đề.

Chứng minh các định lý trong hệ tiên đề của logic vị từ rất khó khăn. Rất may mắn là định lý suy diễn mà chúng ta đã quen biết trong logic mệnh đề cũng đúng cho logic vị từ, và với định lý này thì việc chứng minh các định lý trong hệ tiên đề của logic vị từ đơn giản hơn rất nhiều.

Ví dụ, chứng minh $\forall x(P(x) \supset Q(x)) \supset (\forall x(Px) \supset \forall xQ(x))$

1.	$\forall x(P(x) \supset Q(x))$	*giả thiết*
2.	$\forall xP(x)$	*giả thiết*
3.	$\forall x(P(x) \supset Q(x)) \supset (P(a) \supset Q(a))$	*tiên đề*
4.	$P(a) \supset Q(a)$	*1, 3, MP*
5.	$\forall xP(x) \supset P(a)$	*tiên đề*
6.	$P(a)$	*2, 5, MP*
7.	$Q(a)$	*4, 6, MP*
8.	$\forall xQ(x)$	*7, Gen*

Như vậy ta có $\forall x(P(x) \supset Q(x)),\ \forall xP(x) \vdash \forall xQ(x)$

Áp dụng định lý suy diễn hai lần liên tiếp, ta được công thức cần chứng minh.

Trong hệ thống tiên đề trên đây chúng ta chỉ sử dụng các phép toán logic *phủ định* ($\neg$) và *kéo theo* ($\supset$) , và cũng chỉ sử dụng lượng từ $\forall$. Để cho thuận tiện, ta có thể đưa vào thêm các phép toán và lượng từ khác thông qua các định nghĩa:

$A \vee B =_{df} \neg A \supset B;$

$A \,\&\, B =_{df} \neg (A \supset \neg B);$

$A \equiv B =_{df} (A \supset B) \,\&\, (B \supset A);$

$\exists x\, A(x) =_{df} \neg (\forall x\, \neg A(x));$

3. Các tính chất cơ bản của hệ tiên đề logic vị từ

Người ta chứng minh được các định lý sau đây ***về*** logic vị từ (những định lý như vậy gọi là *siêu định lý – Metatheorems*, nhưng để cho đơn giản, trong trường hợp không lẫn lộn với các định lý ***của*** bản thân logic vị từ, ta sẽ gọi là *định lý*):

Định lý 3.2. Hệ logic vị từ không mâu thuẫn, nghĩa là không tồn tại công thức A sao cho vừa A, vừa $\neg$A là định lý trong hệ logic vị từ.

Định lý 3.3. Mọi định lý của logic vị từ đều là công thức hằng đúng. (Là công thức đúng trong mọi interpretation). Và ngược lại, mọi công thức hằng đúng đều là định lý của logic vị từ. (Định lý về tính đủ).

Hệ quả 3.4. Công thức A đúng trong mọi model của hệ logic vị từ khi và chỉ khi A là định lý của hệ.

Phép chứng minh của các định lý và hệ quả trên đây khá dài dòng và phức tạp, vì thế ở đây chúng tôi chỉ dẫn ra phép chứng minh định lý 3.1.

Bây giờ chúng ta bắt đầu phép chứng minh. Chúng tôi dẫn ra đây phép chứng minh ghi trong cuốn sách đã nêu trên của E. Mendelson. Với công thức A bất kỳ ta ký hiệu bằng h(A) biểu thức thu được bằng cách loại bỏ khỏi A tất cả các lượng từ và các hạn từ (cùng với các dấu ngoặc và dấu phẩy tương ứng). Ví dụ, $h(\forall x\, (P(x) \supset Q(x)))$ sẽ là $P \supset Q$, $h(\exists x\, A(x) \supset \neg B(x,y))$ sẽ là $A \supset \neg B$. Về thực chất thì h(A) bao giờ cũng là một công thức của logic mệnh đề. Rõ ràng là $h(\neg A) = \neg\, h(A)$ và $h(A \supset B) = h(A) \supset h(B)$. Với mọi tiên đề A của logic vị từ, h(A) là công thức hằng đúng (quy luật logic) hiểu theo nghĩa của đại số mệnh đề. Điều này rõ ràng với 3 tiên đề đầu. Tiên đề thứ tư là $\forall(x_i)\, A(x_i) \supset A(t)$, vậy $h(\forall(x_i)\, A(x_i) \supset A(t))$, bằng $A \supset A$. Rõ ràng là hằng đúng. Tiên đề 5 là $\forall x_i\, (A \supset B) \supset (A \supset \forall x_i\, B)$ nên $h(\forall x_i\, (A \supset B) \supset (A \supset \forall x_i\, B) = (A \supset B) \supset (A \supset B)$, cũng là công thức hằng đúng. Cuối cùng, nếu $h\,(A \supset B)$ và h(A) là công thức hằng đúng thì h (B) cũng là hằng đúng; và nếu h(A) là hằng đúng thì $h(\forall x_i\, A)$ cũng là hằng đúng, vì $h(A) = h(\forall x_i\, A)$. Như vậy, nếu A là định lý của logic vị từ, thì h(A) là hằng đúng. Nếu như tồn tại công thức A sao cho cả A và $\neg$A đều là định lý của logic vị từ thì cả h(A) và $h(\neg A)$, tức là cả h(A) và $\neg h(A)$ đều là hằng đúng. Điều đó không thể có được. Như vậy, logic vị từ không mâu thuẫn.

Nếu cùng với các tiên đề của logic vị từ ta còn đưa ra các tiên đề riêng khác thì ta được một hệ gọi là hệ lý thuyết bậc1. Các lý thuyết như vậy là lý thuyết hình thức hóa, rất chặt chẽ, và có tác dụng lớn trong việc nghiên cứu tính không mâu thuẫn, tính đầy đủ và cả các bài toán giải được trong lý thuyết đó của các lý thuyết. Điều này mở ra những ứng dụng rộng rãi và rất hữu ích của logic vị từ.

V. Hệ suy luận tự nhiên của logic vị từ

Cũng như logic mệnh đề, người ta xác định hệ suy luận tự nhiên cho logic vị từ. Hệ suy luận này tương đương với hệ tiên đề của logic vị từ, nghĩa là với công thức A bất kỳ, A có thể chứng minh được trong hệ này khi và chỉ khi A chứng minh được trong hệ tiên đề.

Hệ suy luận tự nhiên gồm có các quy tắc đưa vào và khử các phép toán logic và lượng từ. Hệ không có một tiên đề nào. Trong hệ suy luận tự nhiên phép chứng minh được thực hiện đơn giản hơn rất nhiều so với trong hệ tiên đề.

1. ***Các quy tắc***

Với mọi mệnh đề A, B :

Quy tắc nhập & (ký hiệu $\&_i$) $\dfrac{A, B}{A \,\&\, B}$

Quy tắc khử & (ký hiệu $\&_e$) $\dfrac{A \,\&\, B}{A}$; $\dfrac{A \,\&\, B}{B}$

Quy tắc nhập $\vee$ (ký hiệu $\vee_i$) $\dfrac{A}{A \vee B}$; $\dfrac{B}{A \vee B}$

Quy tắc khử $\vee$ (ký hiệu $\vee_e$) $\dfrac{A \vee B, \neg A}{B}$; $\dfrac{A \vee B, \neg B}{A}$

Quy tắc nhập $\neg$ (ký hiệu $\neg_i$) $\dfrac{B, \neg B}{\neg A}$ (*)

Quy tắc khử $\neg$ (ký hiệu $\neg_e$) $\dfrac{\neg\neg A}{A}$

Quy tắc nhập $\supset$ (ký hiệu $\supset_i$) $\dfrac{B}{A \supset B}$ (*)

Quy tắc khử $\supset$ (ký hiệu $\supset_e$) $\dfrac{A \supset B, A}{B}$

Quy tắc nhập $\forall$ (ký hiệu $\forall_i$) $\dfrac{A(x)}{\forall x A(x)}$ x không xuất hiện tự do trong các giả thiết và giả định trước A(x) trong chuỗi suy diễn.

Quy tắc khử ∀(ký hiệu $\forall_e$) $\dfrac{\forall x A(x)}{A(t)}$ t tự do đối với x trong A(x)

Quy tắc nhập ∃ (ký hiệu $\exists_i$): $\dfrac{A(t)}{\exists x A(x)}$ t tự do đối với x trong A(x)

Quy tắc khử ∃ (ký hiệu $\exists_e$) $\dfrac{\exists x A(x)}{A(c)}$ c là hằng đối tượng mới

Trong hệ này các quy tắc đối với các lượng từ đều đòi hỏi những điều kiện nhất định mới áp dụng được, một số điều kiện như thế đã được nêu cùng với quy tắc, một số khác liên quan đến chuỗi suy diễn, được nêu dưới đây.

2. Chuỗi suy diễn, phép chứng minh

Các khái niệm suy luận, hệ quả logic và chứng minh được định nghĩa như ở hệ suy luận tự nhiên của logic mệnh đề, với những bổ sung sau đây:

(1) Thêm các quy tắc $\forall_i$, $\forall_e$, $\exists_i$, $\exists_e$;

(2) Quy tắc $\forall_i$ chỉ có thể áp dụng vào công thức A(x) với điều kiện x không có xuất hiện tự do trong các giả thiết và giả định đứng trước A(x) trong chuỗi suy diễn. Quy tắc này cũng không áp dụng cho biến y nếu y có xuất hiện tự do dù chỉ trong một công thức dạng ∃xA(x) và ở các bước trước trong chuỗi suy diễn đã có áp dụng quy tắc ∃e cho công thức dạng ∃xA(x) đã nêu trên;

(3) Trong quy tắc $\exists_e$ c phải là hằng đối tượng mới, chưa hề xuất hiện ở các bước trước đó trong chuỗi suy diễn;

(4) Công thức là kết quả của suy luận hoặc là công thức đã chứng minh phải là công thức đóng, nghĩa là không chứa biến tự do.

3. Một số ví dụ

Chứng minh các công thức

a) $\forall x\, A(x) \supset \exists x\, A(x)$;

b) $\forall x\, (A(x)\ \&\ B(x)) \supset (\forall x\, A(x)\ \&\ \forall y\, B(y))$

c) $\exists x\, (A(x)\ \&\ B(x)) \supset (\exists x\, A(x)\ \&\ \exists x\, B(x)$;

d) $\forall x\, A(x)\ \&\ \forall y\, B(y)) \supset \forall x\, (A(x)\ \&\ B(x))$

Chứng minh

a)

1^+.	$\forall x\, (A(x)$	
2.	$A(t)$	$1,\ \forall_e$
3.	$\exists x\, A(x)$	$2,\ \exists_i$
4.	$\forall x\, (A(x) \supset \exists x\, A(x)$	$1, 3, \supset_i$

b)	1^+. $\forall x\ (A(x)\ \&\ B(x))$	
	2. $A(x)\ \&\ B(x)$	1, $\forall_e$
	3. $A(x)$	2, $\&_e$
	3. $B(x)$	2, $\&_e$
	4. $\forall x\ A(x)$	3, $\forall_i$
	5. $\forall y\ B(y)$	4, $\forall_i$
	6. $\forall x\ (A(x)\ \&\ \forall y\ B(y)$	4, 5, $\&_i$
	7. $\forall x\ (A(x)\ \&\ B(x)) \supset (\forall x\ A(x)\ \&\ \forall y\ B(y))$	1, 6, $\supset_i$
c)	1^+. $\exists x\ (A(x)\ \&\ B(x))$	
	2. $A(c)\ \&\ B(c)$	1, $\exists_e$, *c là hạn từ mới*
	3. $A(c)$	2, $\&_e$
	4. $\exists x\ A(x)$	3, $\exists_i$
	5. $B(c)$	2, $\&_e$
	6. $\exists x\ B(x)$	5, $\exists_i$
	7. $\exists x\ (A(x)\ \&\ \exists x\ B(x)$	4, 6, $\&_i$
	8. $\exists x\ (A(x)\ \&\ B(x)) \supset (\exists x\ (A(x)\ \&\ \exists x\ B(x))$	1, 7, $\supset_i$
d)	1^+. $\forall x A(x)\ \&\ \forall y\ B(y)$	
	2. $\forall x\ A(x)$	1, $\&_e$
	3. $A(x)$	2, $\forall_e$
	4. $\forall y\ B(y)$	1, $\&_e$
	5. $B(x)$	4, $\forall_e$
	6. $A(x)\ \&\ B(x)$	3, 5, $\&_i$
	7. $\forall x\ (A(x)\ \&\ B(x))$	6, $\forall_i$
	8. $(\forall x\ (A(x)\ \&\ \forall y\ B(y)) \supset \forall x\ (A(x)\ \&\ B(x))$	1, 7, $\supset_i$

Trong các phép chứng minh trên đây nếu ta bỏ các bước cuối cùng, và cùng với nó là đường kẻ loại ra khỏi suy luận các công thức tương ứng, thì ta có được các suy luận:

a) từ $\forall x\ A(x)$ suy ra $\exists x(A(x)$

b) từ $\forall x\ (A(x)\ \&\ B(x))$ suy ra $\forall x\ A(x)\ \&\ \forall y\ B(y)$

c) từ $\exists x\ (A(x)\ \&\ B(x))$ suy ra $\exists x\ A(x)\ \&\ \exists x\ B(x))$

d) từ $(\forall x A(x)\ \&\ \forall y\ B(y))$ suy ra $\forall x\ (A(x)\ \&\ B(x))$

Ví dụ các phép chứng minh và chuỗi suy diễn sai:

a) Chứng minh $(\exists x\, A(x)\ \&\ \exists x\, B(x)) \supset \exists x\, (A(x)\ \&\ B(x))$

$1^+\ \ \exists x\, A(x)\ \&\ \exists x\, B(x)$

$2.\ \exists x\, A(x)$ $\quad$ $1,\ \&_e$

$3.\ A(c)$ $\quad$ $2,\ \exists_e$

$4.\ \exists x\, B(x)$ $\quad$ $1,\ \&_e$

$5.\ B(c)$ $\quad$ $4,\ \exists_e$

$6.\ A(c)\ \&\ B(c)$ $\quad$ $3, 5,\ \&_i$

$7.\ \exists x\, (A(x)\ \&\ B(x))$ $\quad$ $6,\ \exists_i$

$8.\ (\exists x\, A(x)\ \&\ \exists x\, B(x)) \supset \exists x\, (A(x)\ \&\ B(x))$ $\quad$ $1, 7,\ \supset_i$

Phép chứng minh này sai, vì hạn từ c ở bước 5 không phải là hạn từ mới như đòi hỏi, c đã xuất hiện ở bước 3 (Nếu như nó là đúng thì câu kết luận có thể được diễn giải thành, ví dụ, " Nếu có người say và có người không uống rượu thì có người không uống rượu mà say").

b) $\forall x\, A(x)$ *từ* $\exists x\, A(x)$

$1^+.\ \exists x\, A(x)$

$2\ .\ A(c)$ $\quad$ $1,\ \exists_e$, *c là hạn từ mới*

$3.\ \forall x\, A(x)$ $\quad$ $2,\ \forall i$

Bước 3 trong suy luận trên đây sai, vì quy tắc $\forall$i được áp dụng vào công thức A(c), chứ không phải A(x) như đòi hỏi.
(Nếu không có sự hạn chế đó thì ta có thể thu được từ tiên đề : "Có một người say" kết luận "Tất cả mọi người đều say" , và những suy luận sai tương tự).

3. Tính không mâu thuẫn và đầy đủ của hệ suy luận tự nhiên

Ký hiệu hệ tiên đề của logic vị từ là Svt , hệ suy luận tự nhiên của logic vị từ là SNvt
Người ta đã chứng minh được :

Định lý 3.4.

$$\vdash_{NSvt}\ A \Leftrightarrow\ \vdash_{Svt}\ A.$$

Định lý 3.4 khẳng định hệ tiên đề của logic vị từ và hệ suy luận tự nhiên của logic vị từ hoàn toàn tương đương với nhau, theo nghĩa là mọi định lý trong hệ tiên đề đều là định lý trong hệ suy luận tự nhiên và ngược lại.Vì vậy, các tính chất như tính không mâu thuẫn, tính đủ mà hệ tiên đề có thì hệ suy luận tự nhiên cũng có.

BÀI TẬP CHƯƠNG 3

1. Hãy xác định xem các biểu thức sau đây có phải là công thức trong ngôn ngữ logic vị từ hay không :
   a. $P(a,x,f(g(x)))$
   b. $R(a, f(a),f(a,f(x)))$
   c. $R(x) \supset (Q(a)\ \&\ \neg P(a,x))$
   d. $\forall x \exists y(P(x) \supset Q(x,a)) \vee P(b \vee a)$

e. $\forall x \neg \exists y(P(x,a) \supset R(a,y))$ & $P(x,x)$

2. Dịch các câu sau sang ngôn ngữ logic vị từ :
   a. Bình là nhà báo.
   b. Mai không là nhà báo.
   c. Bà ngoại của Mai là nhà giáo.
   d. Một số người rất thích sầu riêng.
   e. Có những người không muốn nói về mình.
   f. Một số loài gặm nhấm là loài có ích.
   g. Có những người mà mọi người đều yêu mến.
   h. Mai là sinh viên báo chí và Hằng cũng thế.
   i. Mẹ Mai là bác sĩ nhưng không làm việc ở bệnh viện.
   j. Nếu anh bắn vào quá khứ bằng súng lục thì tương lai sẽ bắn vào anh bằng đại bác.
   k. Người ta phải dè chừng con ngựa ở trước mặt, con chó ở sau lưng và con người ở tứ phía.

3. Dịch các đoạn văn sau đây sang ngôn ngữ logic vị từ :

   a. Chuột là loài có hại. Rắn bắt chuột. Mọi loài bắt loài có hại đều là loài có ích. Vậy rắn là loài có ích.

   b. Một số loài chim di cư vào mùa đông là loài quý hiếm. Hồng hạc là loài quý hiếm. Như vậy hồng hạc cũng là loài chim di cư vào mùa đông.

4. Hãy viết bằng ngôn ngữ logic vị từ các phán đoán nhất quyết đơn sau đây :
   a. S a P
   b. M e S
   c. P i M
   d. M o S
   e. S e P

5. Với các công thức sau đây hãy đưa ra một diễn giải trong đó công thức đã cho đúng và một diễn giải trong đó công thức đã cho không đúng.
   a. $\forall x \exists y(P(a,x) \supset Q(f(y),x))$
   b. $\exists x \exists y(P(f(a),x)\ \&\ \neg Q(f(y),x))$
   c. $\forall x \forall y \forall z((P(x,y)\ \&\ P(x,z)) \supset P(y,z))$
   d. $\exists x \exists y \exists z((P(x,y)\ \&\ P(x,z)) \supset P(y,z))$
   e. $\exists x \forall y \forall z((P(x,y)\ \&\ P(x,z)) \supset P(y,z))$
   f. $\forall x \forall y \forall z((P(x,y)\ \&\ P(y,z)) \supset P(x,z))$
   g. $\exists x \exists y \exists z((P(x,y)\ \&\ P(y,z)) \supset P(x,z))$

6. Hãy xác định xem các công thức sau đây có là quy luật logic hay không:
   a. $(\forall x\, A(x)\ \&\ \forall x\, B(x)) \equiv \forall x\, (A(x)\ \&\ B(x))$
   b. $(\forall x\, A(x) \vee \forall x\, B(x)) \supset \forall x\, (A(x) \vee B(x))$
   c. $\forall x\, (A(x) \supset B(x)) \supset (\forall x\, A(x) \supset \forall x\, B(x))$
   d. $\forall x\, (A(x) \equiv B(x)) \supset (\forall x\, A(x) \equiv \forall x\, B(x))$
   e. $\exists x\, (A(x)\ \&\ B(x)) \equiv (\exists x\, A(x)\ \&\ \exists x\, B(x))$
   f. $(\exists x\, A(x) \vee \exists x\, B(x)) \equiv \exists x\, (A(x) \vee B(x))$
   g. $\exists x\, (A(x) \supset B(x)) \equiv (\exists x\, A(x) \supset \exists x\, B(x))$
   h. $\forall x\, \neg A(x) \equiv \neg \exists x\, A(x)$

7. Hãy chứng minh trong hệ tiên đề các công thức sau đây:

a. $\forall x\,(A(x) \supset B(x)) \supset (\forall x\, A(x) \supset \forall x\, B(x))$.
b. $\forall x(A(x) \supset A(x))$
c. $\forall x A(x) \supset \exists x A(x)$
d. $\forall x A(x) \supset \neg\exists y(\neg A(y))$

8. Hãy chứng minh trong hệ suy luận tự nhiên các công thức sau đây (A ≡ B là viết tắt của (A ⊃ B) & (B ⊃ A)):
a. $(\forall x\, A(x)\ \&\ \forall x\, B(x)) \equiv \forall x\,(A(x)\ \&\ B(x))$
b. $(\forall x\, A(x) \vee \forall x\, B(x)) \supset \forall x\,(A(x) \vee B(x))$
c. $\forall x\,(A(x) \equiv B(x)) \supset (\forall x\, A(x) \equiv \forall x\, B(x))$
d. $\exists x\,(A(x)\ \&\ B(x)) \supset (\exists x\, A(x)\ \&\ \exists x\, B(x))$
e. $\exists x\,(A(x) \vee B(x)) \supset (\exists x\, A(x) \vee \exists x\, B(x))$
f. $(\exists x\, A(x) \supset \exists x\, B(x)) \supset \exists x\,(A(x) \supset B(x))$
g. $\forall x\, \neg A(x) \equiv \neg\exists x\, A(x)$
h. $\exists x \neg A(x) \equiv \neg\forall x\, A(x)$
i. $\forall x P(x) \supset \exists x P(x)$
j. $\exists y \forall x P(x,y) \supset \forall x \exists y P(x,y)$

9. Cho chuỗi suy diễn

1^{+} $\forall x\exists y P(x,y)$	
2. $\exists y P(x,y)$	1, $\forall_e$
3. $P(x,c)$	2, $\exists_e$
4. $\forall x P(x,c)$	3, $\forall_i$
5. $\exists y\forall x P(x,y)$	4, $\exists_i$

Chuỗi suy diễn trên có đúng không, vì sao ?

10. Hãy chứng minh các tam đoạn luận đơn sau đây trong hệ suy luận tự nhiên:
a. Kiểu AAA ở hình 1
b. Kiểu EAE ở hình 1
c. Kiểu EIO ở hình 1
d. Kiểu AII ở hình 1
e. Kiểu AEE ở hình 2
f. Kiểu AOO ở hình 2
g. Kiểu EIO ở hình 2
h. Kiểu IAI ở hình 3
i. Kiểu OAO ở hình 3
j. Kiểu AEE ở hình 4

Chương 4 HỢP GIẢI TRONG LOGIC VỊ TỪ

I. Công thức dạng tuyển

1. Định nghĩa

Một *đơn tử vị từ* (từ nay về sau gọi ngắn gọn là đơn tử) – còn gọi là *literal* - là một công thức nguyên tử hoặc phủ định của nó. Ví dụ, P(x,a), ¬Q(a) là các literal; các công thức P(x,a) ∨ Q(a), ¬(P(a,b) & Q(a)), … không phải là các literal. *Công thức dạng tuyển* là công thức dạng $p_1 \vee p_2 \vee \ldots \vee p_n$, trong đó *n* là một số tự nhiên, còn p_i *(i = 1, , …, n)* là các literal. Khi n = 0, ta có công thức dạng tuyển rỗng, ký hiệu bằng một hình vuông nhỏ Σ. Công thức dạng tuyển trên đây còn được biểu thị dưới dạng tập hợp $\{p_1, p_2, \ldots, p_n\}$, khi đó công thưc dạng tuyển rỗng được biểu diễn bằng tập hợp rỗng { }.
Ví dụ công thức dạng tuyển :
P(a,b) ∨ Q(x) ∨ ¬R(x,a)
Q(f(a)) ∨ ¬P(b)
Các công thức này viết dưới dạng tập hợp như sau:
{P(a,b), Q(x), ¬R(x,a)}
{Q(f(a)), ¬P(b)}
Các công thức sau đây không ở dạng tuyển:
P(a) ∨ ¬(Q(a,x) ∨ P(b))
P(x) & ¬P(a) ∨ Q(a,x).

2. Quy trình INSEADOR

Định lý 4.1 *Một công thức bất kỳ của logic vị từ tương đương với một tập hợp các công thức dạng tuyển.*
Chúng ta chứng minh được điều đó bằng cách sử dụng một quy trình đưa bất cứ công thức nào của logic vị từ về dạng tuyển. Đó là quy trình INSEADOR, với các bước giữ nguyên tính đúng của công thức: với A là một công thức, A* là kết quả áp dụng một bước bất kỳ của quy trình INSEADOR vào A, Khi đó |= A khi và chỉ khi |= A*. Đây là quy trình mở rộng của quy trình INDO mà ta đã gặp ở phần logic mệnh đề.

Các bước như sau:

I - Loại bỏ dấu kéo theo và tương đương (Implication Out)
$A \supset B \;\Rightarrow\; \neg A \vee B$
$A \equiv B \;\Rightarrow\; (\neg A \vee B) \;\&\; (\neg B \vee A)$
N - Đưa dấu ¬ vào đứng trước công thức nguyên tử (Negation In)
$\neg\neg A \;\Rightarrow\; A$
$\neg(A \vee B) \;\Rightarrow\; \neg A \;\&\; \neg B$
$\neg(A \;\&\; B) \Rightarrow \neg A \vee \neg B$
$\neg\forall x A(x) \;\Rightarrow\; \exists x \neg A(x)$
$\neg\exists x A(x) \;\Rightarrow\; \forall x \neg A(x)$
S - Chuẩn hoá (Standardization). Ở bước này ta đổi tên các biến buộc sao cho trong công thức cùng một biến không vừa buộc vừa tự do và lượng từ theo một biến bất kỳ không nằm trong miền tác động của một lượng từ khác với cùng biến đó. Chẳng hạn, trong công thức ∀x(P(x) & Q(a)) ⊃ Q(x) biến x vừa buộc vừa tự do, khi đó ta đổi tên biến buộc thành biến y, và được

công thức mới $\forall y(P(y) \& Q(a)) \supset Q(x)$; trong công thức $\forall x(P(x) \supset \exists xQ(x,a))$ ta thấy lượng từ $\exists x$ nằm trong miền tác động của lượng từ $\forall x$, vậy ta đổi thành $\forall x(P(x) \supset \exists yQ(y,a))$.

E - Loại bỏ lượng từ tồn tại (Exist Out)

Ta có hai trường hợp loại bỏ lượng từ tồn tại. Nếu lượng từ $\exists x$ không nằm trong miền tác động của bất cứ lượng từ toàn thể nào thì ta bỏ lượng từ $\exists x$ đi và thay tất cả các xuất hiện của x trong miền tác động của lượng từ này bằng một hằng đối tượng chưa có trong công thức. Hằng đối tượng như vậy gọi là *hằng scolem*. Chẳng hạn : $\exists xP(x,a)$ được biến đổi thành $P(b,a)$, với b là hằng scolem.

Nếu lượng từ $\exists x$ nằm trong miền tác động của lượng từ $\forall y$ thì bỏ $\exists x$ và thay hạn từ f(y) vào chỗ của x trong miền tác động của lượng từ $\exists x$ nêu trên. Nếu lượng từ $\exists x$ nằm trong miền tác động của nhiều lượng từ toàn thể $\forall x_1, \forall x_2, \ldots, \forall x_k$ thì bỏ $\exists x$ và thay hạn từ $f(x_1, x_2, \ldots, x_k)$ vào vị trí của các biến x trong miền tác động của $\exists x$. Ở đây f là một ký tự hàm đối tượng mới, chưa có trong công thức. Hàm được tạo theo cách này gọi là hàm scolem. Ví dụ : $\forall x(P(x) \supset \forall z\exists yQ(x,y,z))$ biến đổi thành $\forall x(P(x) \supset \forall zQ(x,f(x,z),z))$.

D – Đưa dấu tuyển vào đứng trước các literal (Disjunction In)

$A \vee (B \& C) \Rightarrow (A \vee B) \& (A \vee C)$

$(B \& C) \vee A \Rightarrow (A \vee B) \& (A \vee C)$

$A \vee (B \vee C) \Leftrightarrow (A \vee B) \vee C$

$A \& (B \& C) \Leftrightarrow (A \& B) \& C$

Hai dòng cuối của bước D trên đây nói rằng nếu trong công thức chỉ có các dấu tuyển hoặc chỉ có các dấu hội thì không cần sử dụng đến dấu ngoặc đơn.

O – Operator Out (Loại bỏ các dấu & và $\vee$)

$A \& B \Rightarrow A,$
$ B$

$A \vee B \Rightarrow \{A, B\}$

R – Rename (Đổi tên biến). Ở bước này ta đổi tên các biến tự do sao cho các tập hợp khác nhau (tức là các công thức dạng tuyển khác nhau) có các tên biến riêng của mình, không trùng lặp với các tên biến đã có trong các tập hợp khác.

Ví dụ, đưa công thức $\forall x((P(x) \vee Q(a,x)) \supset \exists yQ(x,y))$ về dạng tuyển

$\forall x((P(x) \vee Q(a,x)) \supset \exists yQ(x,y))$

I: $\forall x(\neg(P(x) \vee Q(a,x)) \vee \exists yQ(x,y))$

N: $\forall x((\neg P(x) \& \neg Q(a,x)) \vee \exists yQ(x,y))$

S: $\forall x((\neg P(x) \& \neg Q(a,x)) \vee \exists yQ(x,y))$

E: $\forall x((\neg P(x) \& \neg Q(a,x)) \vee Q(x,f(x)))$

A: $(\neg P(x) \& \neg Q(a,x)) \vee Q(x,f(x))$

D: $(\neg P(x) \vee Q(x,f(x))) \& (\neg Q(a,x) \vee Q(x,f(x)))$

O: $\neg P(x) \vee Q(x,f(x)),$
$\neg Q(a,x) \vee Q(x,f(x))$
$\{\neg P(x), Q(x,f(x))\},$
$\{\neg Q(a,x), Q(x,f(x))\}$

R: $\{\neg P(x), Q(x,f(x))\},$
$\{\neg Q(a,y), Q(y,f(y))\}.$

II. Phép thế

1. Định nghĩa

Phép thế là một ánh xạ hữu hạn từ một tập các biến đối tượng đến một tập hạn từ. Chúng ta ghi phép thế dưới dạng $\{x_1\leftarrow t_1, x_2 \leftarrow t_2, \ldots, x_n \leftarrow t_n\}$, hoặc $\{x_1 / t_1, x_2 / t_2, \ldots, x_n / t_n\}$, trong đó x_i là biến đối tượng, t_i là hạn từ, với i = 1, 2, …, n. Ví dụ, $\{x\leftarrow a, y\leftarrow b\}$, $\{x\leftarrow a, y\leftarrow x, z\leftarrow f(a,x)\}$ là các phép thế. Với phép thế $\gamma = \{x_1\leftarrow t_1, x_2 \leftarrow t_2, \ldots, x_n \leftarrow t_n\}$ ta gọi tập hợp $\{x_1, x_2 , \ldots, x_n \{$ là *miền xác định (domain)* của γ và $\{t_1, t_2, \ldots, t_n\}$ là *miền giá trị (range)* của γ, các hạn từ $t_1, t_2, \ldots, t_n$ gọi là các *phần tử thế* của γ. Nếu các phần tử thế của phép thế γ đều là biến đối tượng thì γ được gọi là *phép đổi tên biến*, hay ngắn gọn hơn, là *phép đổi biến*. Ví dụ, $\tau = \{x\leftarrow y,y\leftarrow z\}$ là một phép đổi biến.

2. Áp dụng phép thế

Giả sử γ là phép thế $\{x_1\leftarrow t_1, x_2 \leftarrow t_2, \ldots, x_n \leftarrow t_n\}$, A là một công thức, khi đó Aγ là kết quả việc áp dụng phép thế γ vào công thức A, nhận được từ A bằng cách thay thế tất cả các xuất hiện của biến x_i trong công thức A bằng hạn từ t_i. Ví dụ, với $\delta = \{x\leftarrow a,y\leftarrow b,z\leftarrow c\}$ và công thức $A = \forall x(P(x,y) \supset Q(a,x))$ thì $A\delta = P(a,b) \supset Q(a,a)$.

3. Tính bất biến của phép thế

Kết quả sẽ ra sao nếu áp dụng nhiều lần cùng một phép thế vào một công thức ? Với công thức A và phép thế δ vừa xét trên đây ta có $A\delta\delta = (P(a,b) \supset Q(a,a)) \{x\leftarrow a,y\leftarrow b,z\leftarrow c\} = P(a,b) \supset Q(a,a) = A\delta$, nghĩa là áp dụng liên tục hai lần phép thế δ vào công thức A ta cũng được kết quả như chỉ áp dụng một lần. Tuy nhiên với phép thế $\tau = \{x\leftarrow a,y\leftarrow x\}$ thì ta có $A\tau = \forall x(P(x,y) \supset Q(a,x)) \{x\leftarrow a,y\leftarrow x\} = P(a,x) \supset Q(a,a)$, trong khi đó $A\tau\tau = (P(a,x) \supset Q(a,a)) \{x\leftarrow a,y\leftarrow x\} = P(a,a) \supset Q(a,a) \neq A\tau$. Nghĩa là trong trường hợp này áp dụng nhiều lần cùng một phép thế vào một công thức sẽ cho kết quả khác với chỉ áp dụng một lần duy nhất. Nếu áp dụng nhiều lần phép thế γ vào bất cứ công thức nào cũng cho kết quả như chỉ áp dụng γ duy nhất một lần vào công thức đó thì ta nói phép thế γ có tính bất biến (Idempotency).

Những phép thế nào có tính bất biến? Để trả lời câu hỏi này chúng ta hãy xem xét một số ví dụ công thức và phép thế. Trước hết, chúng ta dễ dàng nhận thấy là với phép thế γ tùy ý, nếu công thức A không chứa biến thì áp dụng bao nhiêu lần γ vào A kết quả cũng giống nhau, và bằng A. Nhận xét này của chúng ta không có giá trị tự thân, vì tính bất biến của phép thế được xác định trong mối quan hệ với mọi công thức, chứ không phải chỉ trong mối quan hệ với một số công thức nào đó. Tuy nhiên nó cũng giúp tìm ra được các phép thế có tính bất biến. Xuất phát từ nhận xét trên ta nhận thấy nếu phép thế γ có tính chất sao cho mọi công thức Aγ đều không chứa biến thì γ có tính bất biến. Mọi công thức Aγ đều sẽ không chứa biến, nếu γ thay thế tất cả các biến bằng các hạn từ không chứa biến. Nhưng lúc bấy giờ γ là vô hạn, vì số lượng các biến của ngôn ngữ logic vị từ là vô hạn. Để giải quyết khó khăn này chúng ta sử dụng một nhận xét thứ hai : phép thế γ không làm thay đổi công thức A nếu như tập các biến có trong A và tập các biến mà γ thế giá trị (tức miền xác định của γ) không có các phần tử giống nhau. Như vậy γ bất biến, nếu Aγ và miền xác định của γ không có biến chung. Mà Aγ gồm các biến mà γ không thế giá trị và các biến mới xuất hiện khi thay các biến của A bằng các phần tử thế tương ứng trong γ. Vì thế Aγ và miền xác định của γ không có biến chung khi γ có các phần tử thế không chứa biến thuộc miền xác định của nó. Ta gọi phép thế có các phần tử thế không chứa biến thuộc miền xác định là *phép thế thuần tuý (pure)*. Ví dụ, $\{x\leftarrow a,y\leftarrow f(a)\}$, $\{x\leftarrow a,y\leftarrow f(z,a)\}$ là các phép thế thuần tuý; các phép thế $\{x \leftarrow a, y \leftarrow x\}$, $\{x \leftarrow y, y \leftarrow z, z \leftarrow a\}$ không phải là phép thế thuần tuý. Ta đã lập luận rằng các phép thế thuần túy đều có tính bất biến.

4. *Phép thế hợp*

Ta xét ví dụ: cho A = $P(x,y,f(y,z))$, các phép thế $\delta = \{x \leftarrow y, y \leftarrow c, z \leftarrow b\}$ và $\tau = \{x \leftarrow a, y \leftarrow b\}$. Khi đó $A\delta = P(x,y,f(y,z))\ \{x \leftarrow y, y \leftarrow c, z \leftarrow b\} = P(y,c,f(c,b))$, và $A\delta\tau = P(y,c,f(c,b))\{x \leftarrow a, y \leftarrow b\} = P(b,c,f(c,b))$, và $A\gamma = P(x,y,f(y,z))\ \{x \leftarrow b, y \leftarrow c, z \leftarrow b\} = P(b, c, f(c,b))$. Nghĩa là $A\gamma = A\delta\tau$. Như vậy áp dụng γ vào công thức A sẽ được kết quả như khi áp dụng liên tiếp δ và τ vào A. Điều này cũng đúng với bất kỳ công thức B nào khác, chứ không chỉ đúng với A. Rõ ràng áp dụng chỉ một phép thế γ vào một công thức sẽ nhanh chóng và thuận tiện hơn nhiều so với áp dụng liên tiếp hai phép thế khác vào công thức đó. Với hai phép thế δ và τ bao giờ ta cũng xác định được một phép thế γ sao cho áp dụng γ vào công thức A bất kỳ sẽ được kết quả như khi áp dụng liên tiếp δ và τ vào A. Phép thế γ như vậy gọi là phép thế hợp của δ và τ. Hợp của δ và τ ta ký hiệu $\delta\tau$. Có một phương pháp đơn giản để tìm hợp của hai phép thế cho trước δ và τ. Phương pháp đó như sau : Trước hết áp dụng τ vào range của δ, nghĩa là áp dụng τ vào các phần tử thế của δ, gọi kết quả là δ'; sau đó bổ sung thêm các cặp biến-hạn từ có trong τ mà cặp với biến đó chưa có trong δ'; sau cùng, lược bỏ đi các cặp biến-hạn từ có biến trùng với hạn từ, kết quả là hợp của δ và τ. Ví dụ, cho $\delta = \{x \leftarrow z, y \leftarrow f(a,x)\}$, $\tau = \{x \leftarrow b, z \leftarrow a\}$. Ta áp dụng τ vào cho range của δ, được $\delta' = \{x \leftarrow a, y \leftarrow f(a,b)\}$. Cặp $z \leftarrow a$ là cặp có biến z, chưa có trong δ', vì vậy ta bổ sung cặp này vào δ', được kết quả cuối cùng $\delta\tau = \{x \leftarrow a, y \leftarrow f(a,b), z \leftarrow a\}$.

Tính chất của phép hợp

Hợp của hai phép thế không có tính chất giao hoán. Chẳng hạn, với δ và τ vừa xét ta có $\tau\delta = \{x \leftarrow b, z \leftarrow a\}\{x \leftarrow z, y \leftarrow f(a,x)\} = \{x \leftarrow b, y \leftarrow f(a,x), z \leftarrow a\} \neq \tau\delta$.

Nhưng phép hợp có tính kết hợp, tức là $\delta(\tau\gamma) = (\delta\tau)\gamma$.

Phép hợp cũng có đơn vị, đó là phép thế rỗng { }, vì với mọi phép thế γ, $\{\}\gamma = \gamma\{\} = \gamma$.

5. *Quan hệ sắp xếp*

Phép thế δ được gọi là lớn hơn phép thế τ (ký hiệu $\delta > \tau$) nếu tồn tại một phép thế ε không rỗng sao cho $\delta\varepsilon = \tau$. Ví dụ, phép thế $\delta = \{x \leftarrow a, y \leftarrow z\}$ lớn hơn phép thế $\tau = \{x \leftarrow a, y \leftarrow b, z \leftarrow b\}$, vì với phép thế $\varepsilon = \{z \leftarrow b\}$ ta có $\delta\varepsilon = \tau$. Dễ dàng nhận thấy quan hệ lớn hơn giữa các phép thế có tính chất bắc cầu, nghĩa là $(\delta > \tau)\ \&\ (\tau > \varepsilon) \Rightarrow \delta > \varepsilon$

III. Đồng nhất thể

1. *Định nghĩa*

Cho các công thức A và B, phép thế γ được gọi là *đồng nhất thể* (unifier) của A và B nếu $A\gamma = B\gamma$. Ví dụ, với A = $P(x,a)$ và B = $P(b,y)$ thì $\gamma = \{x \leftarrow b, y \leftarrow a\}$ là đồng nhất thể của A và B, vì $A\gamma = P(x,a)\ \{x \leftarrow b, y \leftarrow a\} = P(b,a) = P(b,y)\ \{x \leftarrow b, y \leftarrow a\} = B\gamma$. Nếu tồn tại một đồng nhất thể cho các công thức A và B thì A và B được gọi là *khả đồng nhất*. Hai công thức khả đồng nhất có nhiều đồng nhất thể. Chẳng hạn, ngoài đồng nhất thể γ trên đây, các công thức A = $P(x,a)$ và B = P(b,y) còn có đồng nhất thể $\{x \leftarrow b, y \leftarrow a, z \leftarrow c\}$ và nhiều đồng nhất thể khác. Các phép thế $\delta = \{x \leftarrow b, y \leftarrow b\}$, $\tau = \{x \leftarrow c, y \leftarrow c\}$, $\varepsilon = \{x \leftarrow y\}$, … đều là các đồng nhất thể của các công thức $Q(x,b)$ và $Q(y,b)$.

2. *Đồng nhất thể lớn nhất*

Đồng nhất thể lớn nhất (ký hiệu *mgu*, từ cụm từ *most general unifier*) của hai công thức là phép thế lớn nhất trong tất cả các đồng nhất thể của hai công thức đó.

Người ta đã chứng minh được định lý làm cơ sở cho phương pháp hợp giải sau đây.

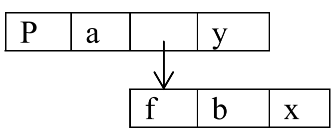
Định lý 4.2 *Với hai công thức bất kỳ, nếu chúng khả đồng nhất thì chúng có đồng nhất thể lớn nhất. Đồng nhất thể lớn nhất đó duy nhất tính cho đến phép đổi biến - nghĩa là khi không phân biệt hai phép thế* δ *và* γ *nếu tồn tại phép đổi biến* τ *sao cho* $(range(\gamma))\tau = (range(\delta))\tau$ *và* $(domain(\gamma))\tau = (domain(\delta))\tau$.

Ta có một thuật toán đơn giản để tìm đồng nhất thể lớn nhất của hai công thức bất kỳ, nếu đồng nhất thể lớn nhất đó tồn tại, nghĩa là nếu hai công thức đó khả đồng nhất.

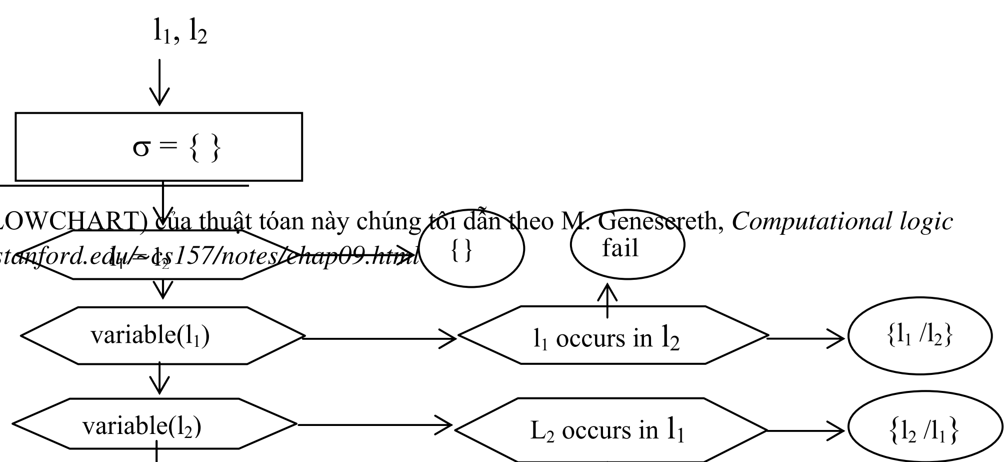
Trước hết ta hãy biểu diễn một công thức bất kỳ dưới dạng một chuỗi ký tự. Các ký tự vị từ, hằng đối tượng, biến đối tượng có mặt trong công thức và không nằm dưới dấu hàm đối tượng được coi là phần tử của chuỗi. Thứ tự sắp xếp của chuỗi tính từ trái qua phải. Các hạn từ được cấu tạo từ ký hiệu hàm đối tượng được coi là một chuỗi con của chuỗi ban đầu. Như vậy một công thức có thể được biểu diễn thành một chuỗi có chứa hoặc không chứa các chuỗi con. Các chuỗi con cũng có thể chứa các chuỗi con của mình. Ví dụ, công thức P(a,x,y) tương ứng với chuỗi

P	a	x	y

còn công thức P(a,f(b,x),y) thì tương ứng với chuỗi:



Các ký tự hàm đối tượng, biến đối tượng, vị từ là các hằng. Thuật toán xác định đồng nhất thể lớn nhất của hai công thức bất kỳ đã được biểu diễn thành hai chuỗi ký tự l_1 và l_2 như sau [1] :



[1] Lưu đồ (FLOWCHART) của thuật tóan này chúng tôi dẫn theo M. Genesereth, *Computational logic* *http://logic.stanford.edu/~cs157/notes/chap09.html*

Với các công thức đầu vào đã được biểu diễn dưới dạng các chuỗi l_l và l_2 quy trình bắt đầu với đồng nhất thể lớn nhất $\sigma = \{\}$. Nếu $l_1 = l_2$ thì quy trình kết thúc, kết quả là đồng nhât thể rỗng. Nếu l_1 và l_2 đều là hằng (hằng đối tượng, ký tự vị từ, ký tự hàm đối tượng, ký tự chỉ các liên từ logic) thì quy trình cũng kết thúc, l_1 và l_2 không khả đồng nhất. Nếu l_1 là biến thì quy trình sẽ kiểm tra xem l_l có trong l_2 hay không. Nếu l_1 có trong l_2 thì l_1 và l_2 không khả đồng nhất, quy trình kết thúc. Còn nếu l_1 không có trong l_2 thì đồng nhất thể lớn nhất là $\{l_1/l_2\}$. Cũng tương tự như thế với trường hợp l_2 là biến. Trường hợp còn lại là trường hợp khi cả l_1 và l_2 đều là các biểu thức phức tạp. Khi đó quy trình sẽ xử lý từng cặp ký tự tính từ bên trái bằng cách gọi đệ quy chính mình. Nếu lần gọi đệ quy thành công với kết quả là phép thế τ thì σ được thay bằng τ, τ cũng được áp dụng vào phần còn lại của các chuỗi l_1 và l_2. Nếu lần gọi đệ quy thất bại thì toàn bộ quy trình kết thúc thất bại, l_1 và l_2 không khả đồng nhất.

IV. Quy tắc hợp giải

Quy tắc hợp giải trong logic vị từ cũng tương tự như trong logic mệnh đề, nhưng có tính đến sự có mặt của các biến đối tượng, và vì thế cần đến các đồng nhất thể để đồng nhất các literal trước khi áp dụng.

$$\frac{A \vee B, \neg A^{*} \vee C}{(B \vee C)\gamma}$$

Trong quy tắc này A và A^* khác nhau (ở hợp giải mệnh đề chúng như nhau), nhưng khả đồng nhất, và γ là một đồng nhất thể của chúng.

Ví dụ : $$\frac{R(a,x) \vee Q(b) \vee P(y), \neg P(z) \vee R(b,y)}{R(a,x) \vee Q(b) \vee R(b,c)}$$

Ở đây hai đơn tử P(y) và ¬P(z) khả đồng nhất, γ = {y/c, z/c} là một trong những đồng nhất thể của chúng (không phải là đồng nhất thể lớn nhất).
Để tiện sử dụng, ta có thể đòi hỏi γ trong quy tắc trên là đồng nhất thể lớn nhất của hai đơn tử được chọn triệt tiêu trong các tiền đề, tức là γ = mgu(A, A^*).
Với đòi hỏi này, vì mgu(P(y), P(z)) = {y/z}, nên ví dụ trên trở thành :

$$\frac{R(a,x) \vee Q(b) \vee P(y), \neg P(z) \vee R(b,y)}{R(a,x) \vee Q(b) \vee R(b,z)}$$

Quy tắc hợp giải trên đây có tính khái quát cao, vì nó cho phép làm việc cả với các công thức không phải dạng tuyển.

Quy tắc hợp giải còn được cho chính xác hơn, dưới dạng tập hợp, như sau.
Nếu công thức dạng tuyển Ψ chứa literal φ, công thức dạng tuyển Φ chứa literal $\neg\phi$, φ và ϕ khả đồng nhất với *mgu* γ, thì từ Φ và Ψ ta được kết quả theo sơ đồ sau :

$$\frac{\begin{array}{l}\{\varphi_1, \varphi_2, \ldots, \varphi, \ldots, \varphi_m\} \\ \{\phi_1, \phi_2, \ldots, \neg\phi, \ldots, \phi_n\}\end{array}}{\{\varphi_1, \varphi_2, \ldots, \varphi_m, \phi_1, \phi_2, \ldots, \phi_n\}\gamma}$$

trong đó $\gamma = mgu(\varphi,\phi)$.

Ví dụ : Công thức thứ nhất chứa literal *P(a, x),* công thức thứ hai chứa literal *¬P(y,b).* Vì $\gamma = \{x \leftarrow b, y \leftarrow a\}$ là đồng nhất thể lớn nhất của *P(a,x)* và *P(b,y)* nên với hai công thức này có thể áp dụng quy tắc hợp giải, chẳng hạn :

$$\frac{\begin{array}{l}\{P(a,x), R(a), Q(y)\} \\ \{R(x), \neg P(y,b), S(f(x))\}\end{array}}{\{R(a), Q(a), R(b), S(f(b))\}}$$

Dạng này của quy tắc hợp giải chỉ áp dụng cho các công thức dạng tuyển.

Hai literal *P(a,x)* và *P(x,b)* không khả đồng nhất vì x không thể vừa bằng *a* lại vừa bằng *b*, nên không thể sử dụng *P(a,x)* và *¬P(x,b)* làm cặp đơn tử triệt tiêu nhau cho các công thức chứa chúng. Trong khi đó chúng ta biết rằng hai literal này tương ứng với các công thức $\forall x P(a,x)$ và $\forall x \neg P(x,b)$ và các công thức rõ ràng là loại trừ lẫn nhau, tức là phải triệt tiêu với nhau.

Để ý rằng biến *x* trong công thức $\forall x P(a,x)$ và biến *x* trong $\forall x \neg P(x,b)$ thực ra không hề liên quan đến nhau, nên ta có thể đổi tên biến ở một trong hai công thức này để tránh khó khăn khi sử dụng chúng cùng lúc. Sau đó mới áp dụng quy tắc hợp giải.

Ví dụ : Cho hai công thức $P(a,x) \vee R(x) \vee \neg Q(f(y),c)$ và $\neg P(x,b) \vee R(a) \vee S(z)$. Ta đổi biến ở công thức thứ nhất, được $P(a,v) \vee R(v) \vee \neg Q(f(y),c\}$ (phép đổi biến $\tau = \{x \leftarrow v\}$). Rõ ràng $P(a,v)$ và $P(x,b)$ có đồng nhất thể lớn nhất là $\gamma = \{x \leftarrow a,\ v \leftarrow b\}$. Như vậy ta có:

$P(a,v) \vee R(v) \vee \neg Q(f(y),c)$ $\qquad$ $\neg P(x,b) \vee R(a) \vee S(z)$

$\{x/a,\ v/b\}$

$R(b) \vee \neg Q(f(y),c) \vee R(a) \vee S(z)$

Tuy nhiên đổi tên biến rồi mới áp dụng quy tắc hợp giải vẫn chưa giải quyết được loại khó khăn sau đây. Chẳng hạn, ta có các tiền đề $\forall x P(x) \vee \forall y P(y)$ và $\forall v(\neg P(v)) \vee \forall u(\neg P(u))$. Rõ ràng các tiền đề này tương đương với các công thức $\forall x P(x)$ và $\forall v(\neg P(v))$ và các công thức này phủ định lẫn nhau. Như thế từ chúng ta phải rút ra được công thức dạng tuyển rỗng. Thế nhưng dạng tuyển của các tiền đề ban đầu tương ứng là $\{P(x), P(y)\}$ và $\{\neg P(v), \neg P(u)\}$. Từ đây, áp dụng quy tắc hợp giải ta không được dạng tuyển rỗng, mà được $\{P(y), \neg P(u)\}$ hoặc $\{P(x), \neg P(u)\}$ hoặc $\{P(y), \neg P(v)\}$ hoặc $\{P(x), \neg P(v)\}$.

Để giải quyết vấn đề này, ta sử dụng nguyên lý hợp giải 3. Trước hết, chúng ta xem xét khái niệm *factor* . Cho công thức dạng tuyển, nghĩa là một tập hợp các literal Φ. Nếu các literal trong một tập con nào đó của Φ khả đồng nhất với nhau với đồng nhất thể lớn nhất γ, thì $\Phi' = \Phi\gamma$ là một factor của Φ. Chẳng hạn, $P(a)$ và $P(x)$ có đồng nhất thể lớn nhất là $\gamma = \{x \leftarrow a\}$, vậy $\Phi' = \{Q(a,a), P(a), R(y)\}$ là một factor của $\Phi = \{Q(x,a, P(a), P(x), R(y)\}$. Rõ ràng mỗi công thức dạng tuyển là một factor của chính nó. Bây giờ chúng ta có nguyên lý hợp giải hoàn chỉnh như sau. Cho hai công thức dạng tuyển Φ và Ψ; Φ' và Ψ' lần lượt là hai factor tương ứng của chúng. Nếu Φ' chứa literal ϕ, Ψ' chứa literal $\neg\varphi$, và mgu của ϕ và φ là γ thì từ hai công thức dạng tuyển đó ta rút ra được công thức dạng tuyển $((\Phi' - \{\phi\}) \cup (\Psi' - \{\neg\varphi\}))\gamma$.

$$\frac{\begin{array}{l}\Phi\\ \Psi\end{array}}{((\Phi' - \{\phi\}) \cup (\Psi' - \{\neg\varphi\}))\gamma}$$

Trong đó $\phi \in \Phi'$, $\neg\varphi \in \Psi'$, Φ' là một factor của Φ,
Ψ' là một factor của Ψ, $\gamma = \mathrm{mgu}(\phi, \neg\varphi)$.

Để tránh trường hợp trùng biến giữa hai công thức dạng tuyển ta có thể đổi tên biến của một trong hai công thức đó.

Ví dụ, cho hai công thức $\Phi = \{R(x), Q(x,a), P(b,y), Q(b,a)\}$ và $\Psi = \{R(a), \neg Q(x,z)\}$. Rõ ràng $Q(x,a)$ và $Q(b,a)$ có đồng nhất thể lớn nhất $\gamma = \{x \leftarrow b\}$, vì thế một trong những factor của Φ là $\Phi' = \{R(b), Q(b,a), P(b,y)\}$; Ψ là một factor của chính nó; $mgu(Q(x,z), Q(b,a)) = \{x \leftarrow b,\ z \leftarrow a\}$. Như thế, ta được :

$\{R(x), Q(x,a), P(b,y), Q(b,a)\}$

$\{R(a), \neg Q(x,z)\}$

$\Downarrow$

$\{R(b), Q(b,a), P(b,y)\}$

$$\frac{\{R(a), \neg Q(x,z)\}}{\{R(b), P(b,y), R(a)\}}$$

Ba trường hợp hợp giải mà chúng ta vừa khảo sát trên đây là ba phương án (phương án sau hoàn thiện hơn phương án trước) của cùng một quy tắc hợp giải. Sử dụng phương án một đơn giản hơn sử dụng phương án 2, sử dụng phương án sau này lại đơn giản hơn sử dụng phương án 3. Vì thế, trong trường hợp không cần đến phương án phức tạp thì có thể sử dụng phương án đơn giản hơn.

V. Suy diễn hợp giải (chuỗi hợp giải) và phép chứng minh

Một suy diễn hợp giải, hay chuỗi hợp giải (resolution derivation) của công thức φ từ tập hợp công thức Δ là một dãy các công thức nối tiếp nhau, trong đó mỗi một công thức hoặc là (1) một phần tử của Δ, hoặc là (2) rút ra được từ các công thức đứng trước nó theo nguyên lý hợp giải; công thức cuối cùng của dãy là φ. Một công thức φ được coi là được chứng minh bằng hợp giải nhờ tập tiền đề Δ khi và chỉ khi có một suy diễn hợp giải của dạng tuyển rỗng từ tập $\{\neg\varphi\} \cup \Delta$. Người ta đã chứng minh được rằng công thức φ chứng minh được bằng hợp giải nhờ tập tiền đề Δ khi và chỉ khi φ là một hệ quả logic của Δ, nghĩa là khi A_1 & A_2 & … & A_n & … ⊃ φ (trong đó A_i , với i = 1, 2, 3, … là phần tử của Δ) là một quy luật logic.

Ví dụ: *Δ = {P(b)}*

{R(a,x), ¬ R(b,x)}

{R(b,y), ¬ P(y)}

{Q(a,z), Q(b,z), ¬ P(x)}

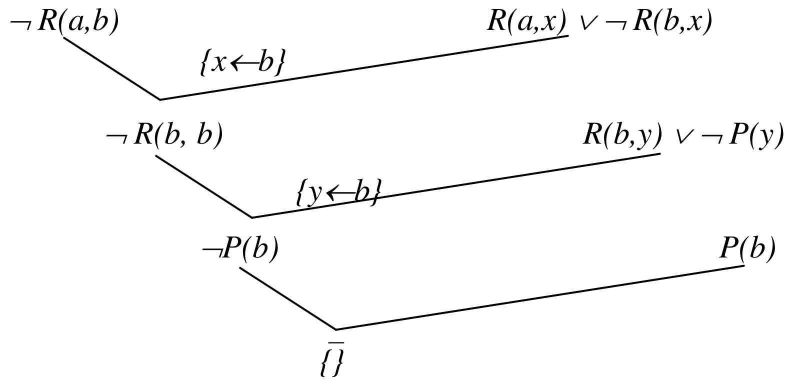
φ = {R(a,b)}

Khi đó φ chứng minh được nhờ Δ, vì ta có chuỗi suy diễn hợp giải sau đây :

1.	*{P(b)}*	tiền đề
2.	*{R(a,x), ¬ R(b,x)}*	tiền đề
3.	*{R(b,y), ¬ P(y)}*	tiền đề
4.	*{Q(a,z), Q(b,z), ¬ P(x)}*	tiền đề
5.	*{¬ R(a,b)}*	¬ φ
6.	*{¬ R(b,b)}*	2, 5 {x←b}
7.	*{¬ P(b)}*	3, 6 {y←b}
8.	*{}*	1, 7

Qua ví dụ trên ta thấy không nhất thiết phải sử dụng tất cả các tiền đề cho trước. Cũng có trường hợp một tiền đề được sử dụng nhiều lần.

Hợp giải trong logic mệnh đề là một trường hợp riêng của hợp giải trong logic vị từ. Dựa trên hợp giải trong logic mệnh đề bạn sẽ dễ nắm bắt ý tưởng của hợp giải trong logic vị từ hơn. Để bạn đọc dễ liên hệ với hợp giải trong logic mệnh đề, chúng tôi trình bày lại suy diễn hợp giải trên đây dưới dạng cây như trong logic mệnh đề. Để ý rằng công thức dạng tuyển $\{A_1, A_2, \ldots, A_n\}$ có thể viết dưới dạng $A_1 \vee A_2 \vee \ldots \vee A_n$.



VI. Áp dụng

Hợp giải trong logic vị từ có thể được ứng dụng để giải quyết nhiều vấn đề khác nhau như chứng minh tính mâu thuẫn của một tập hợp công thức, xác định xem một công thức nào đó có thể rút ra được từ một tập hợp tiền đề cho trước hay không. Nó cũng được dùng để tìm kiếm câu trả lời, chính xác hơn là tìm kiếm các đối tượng thỏa mãn các đòi hỏi nào đó nêu trong câu hỏi.

*1. **Xác định tính mâu thuẫn của một tập công thức***

Cho tập hợp công thức Δ. Nếu xây dựng được một chuỗi hợp giải của {} từ Δ. Vậy, để chứng minh rằng một tập hợp công thức nào đó là mâu thuẫn, ta chỉ cần tìm một chuỗi hợp giải công thức rỗng {} từ tập công thức đó. Một chuỗi hợp giải của công thức rỗng {} từ tập công thúc Δ gọi là một *chuỗi phản bác hợp giải*, hay ngắn gọn hơn là *chuỗi phản bác* của Δ.

Có thể dùng chuỗi phản bác để chứng minh rằng một công thức φ nào đó là hệ quả logic của một tập hợp tiền đề Δ (bằng cách xây dựng chuỗi phản bác cho tập hợp $\Delta \cup \{\neg\varphi\}$). Ví dụ vừa mới xét trên kia là một phép chứng minh như vậy.

2. ***Trả lời câu hỏi đúng, sai***

Có thể sử dụng chuỗi phản bác để trả lời cho các câu hỏi dạng đúng hay sai. Chẳng hạn, người ta cho biết mẹ của Hải là bà Bình, mẹ của Nam là bà Ba, Mẹ của Hoa là bà Bình, Hai người cùng mẹ là anh em. Hỏi Hải và Hoa có là anh em hay không ? Để trả lời câu hỏi này bằng hợp giải, trước hết chúng ta chuyển các tiền đề và phủ định của câu hỏi về dạng tuyển, rồi sau đó áp dụng nguyên lý hợp giải. Nếu từ tập các tiền đề và phủ định của câu hỏi đó có thể rút ra được công thức rỗng thì câu trả lời là "đúng", ngược lại thì câu trả lời là "sai". Với ví dụ này câu hỏi là Anh_em(Hải, Hoa) ?

1. {Mẹ(bà Bình, Hải)}	tiền đề
2. {Mẹ(bà Ba, Nam)}	tiền đề
3. {Mẹ(bà Bình, Hoa)}	tiền đề

4. {Anh_em(x, y), ¬Mẹ(z,x), ¬Mẹ(z,y)} tiền đề

5. {¬ Anh_em(Hải, Hoa)} phủ định của câu hỏi

6. {¬ Mẹ(z, Hải), ¬ Mẹ(z, Hoa)} 4, 5 {x←Na, y←Hoa}

7. {¬ Mẹ(bà Bình, Hoa)} 1, 6 {z←bà Bình}

8. {} 3, 7 {}

3. Tìm kiếm câu trả lời

Trong ví dụ trên đây, nếu ta muốn biết ai là anh em của Hoa thì câu hỏi sẽ được viết dưới dạng Anh_em(x, Hoa), còn muốn biết những ai là anh em của nhau thì câu hỏi sẽ là Anh_em(x,y). Nói cách khác, câu hỏi sẽ dành những chỗ trống – được thể hiện dưới dạng các biến. Qúa trình xây dựng chuỗi phản bác về sau sẽ điền vào chỗ trống trong các câu hỏi các giá trị được thế tương ứng cho các biến trong câu hỏi. Những giá trị biến như thế, nghĩa là những giá trị được điền vào chỗ trống đó chính là các câu trả lời ta cần.

Ví dụ, từ ví dụ tên đây, ta đi tìm xe ai là anh em của Hoa. Khi đó ta có:

1. {Mẹ(bà Bình, Hải)} tiền đề

2. {Mẹ(bà Ba, Nam)} tiền đề

3. {Mẹ(bà Bình, Hoa)} tiền đề

4. {Anh_em(x, y), ¬Mẹ(z,x), ¬Mẹ(z,y)} tiền đề

5. {¬ Anh_em(v, Hoa)} phủ định của câu hỏi

6. {¬ Mẹ(z, v), ¬ Mẹ(z, Hoa)} 4, 5 {x←Na, y←Hoa}

7. {¬ Mẹ(bà Bình, Hoa)} 1, 6 {z←bà Bình, v←Hải}

8. {} 3, 7 {}

Kết quả : Hải là anh em của Hoa, vì Hải chính là giá trị được thế vào cho biến *v* của câu hỏi (bước 7).

VII. Giản lược tiền đề

Cũng như hợp giải trong logic mệnh đề, hợp giải trong logic vị từ cũng sử dụng các chiến lược nhằm giản lược tập tiền đề cho trước và giảm thiểu các kết quả trung gian. Ở đây chúng ta cũng sử dụng các cách loại bỏ tiền đề một chiều, quy luật logic và tiền đề yếu. Hợp giải tuyến tính cũng được sử dụng tương tự như ở hợp giải trong logic mệnh đề. Tất cả những chiến lược này được sử dụng với những điều chỉnh nhỏ cho phù hợp với sự có mặt của các biến tự do trong các literal. Lập luận tương tự như trong phần tương ứng của logic mệnh đề, ta có các định lý tương tự các định lý 2.2, 2.4, 2.6.

1. Giản lược tiền đề là quy luật logic

Công thức dạng tuyển có chứa cùng lúc literal φ và ¬φ nào đó gọi là quy luật logic. Ví dụ, các công thức dạng tuyển {P(x), Q(x, a, f(a)), ¬ P(x)}, {R(x, a), ¬R(x, a)} là các quy luật logic. Công thức {P(x,a), P(b, a)} không phải là quy luật logic. Các tiền đề là quy luật logic có thể được loại bỏ khỏi tập tiền đề mà không ảnh hưởng đến tập các hệ quả logic của tập tiền đề đó.

Định lý 4.2. *Cho một tập hợp tiền đề S, tập S^* là kết quả giản lược hết tất cả các tiền đề là quy luật logic trong S, khi đó S mâu thuẫn khi và chỉ khi S^* mâu thuẫn.*

Hệ quả 4.3. Có thể giản lược các tiền đề là quy luật logic.

2. Giản lược tiền đề một chiều

Hai literal φ và $\neg \phi$ gọi là tương phản với nhau, nếu φ và ϕ có đồng nhất thể lớn nhất γ. Chẳng hạn, P(x,a) và P(b,y) có đồng nhất thể lớn nhất là $\{x \leftarrow b, y \leftarrow a\}$ nên P(x,a) và $\neg$P(b,y) tương phản với nhau; R(a,x) và R(x,b) không khả đồng nhất, nên R(a,x) và $\neg$R(x,b) không phải là cặp literal tương phản với nhau. Một literal trong một công thức dạng tuyển của một cặp tiền đề được gọi là một chiều, nếu như không có công thức dạng tuyển nào khác trong tập tiền đề đó có chứa literal tương phản với nó. Công thức dạng tuyển trong một tập hợp tiền đề là một chiều khi và chỉ khi nó chứa literal một chiều.

Định lý 4.4. *Cho một tập hợp tiền đề S, tập S^* là kết quả giản lược hết tất cả các tiền đề là một chiều trong S, khi đó S mâu thuẫn khi và chỉ khi S^* mâu thuẫn.*

***Hệ quả 4.5. Có thể giản lược các tiền đề một chiều*.**

Có thể loại bỏ các tiền đề một chiều khỏi tập hợp tiền đề mà không ảnh hưởng đến tập hệ quả logic của nó. Ví dụ, cho tập tiền đề

$$
\begin{aligned}
\Delta = \quad & \{\{P(a), Q(a,x)\}; \\
& \{\neg P(x), R(a,y)\}; \\
& \{S(a, f(x), c), \neg R(a,b)\}; \\
& \{\neg S(y,f(b),c), R(x,z)\}\}
\end{aligned}
$$

Rõ ràng literal Q(a,x) trong Δ là một chiều. Vậy có thể giản lược tiền đề chứa nó, được:

$$
\begin{aligned}
\Delta = \quad & \{\{\neg P(x), R(a,y)\}; \\
& \{S(a, f(x), c), \neg R(a,b)\}; \\
& \{\neg S(y,f(b),c), R(x,z)\}\}
\end{aligned}
$$

Sau khi giản lược tiền đề một chiều {P(a), Q(a,x)} ta thấy literal $\neg$ P(x) lại trở thành một chiều, vì thế lại có thể giản lược tiền đề chứa nó. Ta được :

$$
\begin{aligned}
\Delta = \quad & \{\{S(a, f(x), c), \neg R(a,b)\}; \\
& \{\neg S(y,f(b),c), R(x,z)\}\}
\end{aligned}
$$

Tập tiền đề Δ bây giờ không còn có literal một chiều nào nữa.

3. Giản lược tiền đề yếu

Công thức dạng tuyển ψ là yếu hơn công thức dạng tuyển Φ khi và chỉ khi tồn tại một phép thế δ sao cho $\Phi\delta \subseteq \Psi$.

Định lý 4.6. *Gọi S^* là kết quả việc loại bỏ toàn bộ các tiền đề yếu của S. Khi đó S mâu thuẫn khi và chỉ khi S^* mâu thuẫn.*

Hệ quả 4.7. Có thể giản lược các tiền đề yếu.

Một công thức dạng tuyển yếu hơn một công thức dạng tuyển khác trong tập tiền đề thì có thể được loại bỏ khỏi tập tiền đề mà không hề ảnh hưởng đến tập hợp các hệ quả logic của tập tiền đề đó.

Ví dụ giản lược tiền đề. Xét xem tập hợp tiền đề sau đây có thể rút ra kết luận $\neg P(a)$ không:

1. $\neg P(x) \vee Q(x)$
2. $\neg P(y) \vee R(y)$
3. $\neg Q(z) \vee S(z) \vee P(a)$
4. $S(x_1) \vee \neg R(x_1)$

Thêm phủ định của kết luận vào tập tiền đề:

5. $P(a)$

Ta thấy tiền đề 3 chứa công thức 5, vậy 3 là tiền đề yếu, có thể giản lược. Bây giờ ta có tập hợp:

1. $\neg P(x) \vee Q(x)$
2. $\neg P(y) \vee R(y)$
4. $S(x_1) \vee \neg R(x_1)$
5. $P(a)$

Trong tập hợp tiền đề này các đơn tử $Q(x)$ và $S(x_1)$ là đơn tử một chiều. Như vậy các tiền đề 1 và 3 là một chiều,có thể giản lược. Tập tiền đề còn lại:

2. $\neg P(y) \vee R(y)$
5. $P(a)$

Rõ ràng $R(y)$ là đơn tử một chiều, vậy có thể giản lược tiền đề 2. Nhưng lúc đó tiền đề 5 còn lại cũng là một chiều. Vậy tất cả các tiền đề đều được lược bỏ.

Từ tập tiền đề đã cho không rút ra được kết luận $\neg P(a)$.

BÀI TẬP CHƯƠNG 4

1. Hãy áp dụng phép thế $\gamma = \{x \leftarrow a, y \leftarrow b, z \leftarrow f(b), w \leftarrow f(y)\}$ vào các công thức cho sau đây:
   a. $P(a, f(a), g(b)) \vee Q(x, a, f(y))$
   b. $P(f(x), a, y, w, g(z))$
   c. $P(f(g(b, w)), y, g(a, z), z)$
   d. $P(a, g(y, z), x) \vee Q(y, f(b), x)$
   e. $R(a, x, x) \vee P(y, f(y), y)$
   f. $\neg R(f(g(a)), y, g(y, z), f(w))$
2. Tìm phép thế hợp của các phép thế

a. $\{x \leftarrow a, y \leftarrow b, z \leftarrow c\}$ và $\{x \leftarrow a, u \leftarrow b, w \leftarrow c\}$
b. $\{x \leftarrow u, y \leftarrow f(x), z \leftarrow c\}$ và $\{x \leftarrow a, u \leftarrow b, w \leftarrow y\}$
c. $\{x \leftarrow f(u), y \leftarrow f(x), z \leftarrow w\}$ và $\{x \leftarrow a, u \leftarrow b, z \leftarrow b, w \leftarrow f(y)\}$
d. $\{y \leftarrow f(x), z \leftarrow f(u)\}$ và $\{x \leftarrow a, u \leftarrow b, z \leftarrow b, w \leftarrow f(y)\}$
e. $\{x \leftarrow z, y \leftarrow b\}, \{z \leftarrow a\}$ và $\{y \leftarrow x, z \leftarrow b, u \leftarrow a, w \leftarrow f(u)\}$
f. $\{x \leftarrow z, y \leftarrow b, z \leftarrow f(x)\}, \{y \leftarrow f(z), z \leftarrow f(a)\}$ và $\{u \leftarrow a, w \leftarrow a\}$

3. Tìm các đồng nhất thể (nếu có) cho các cặp công thức
   a. $P(a, x)$ và $P(y, b)$
   b. $Q(a, x, y)$ và $\neg Q(x, y, b)$
   c. $P(f(x), a, y)$ và $P(f(b), y, z)$
   d. $P(a, x, b) \vee Q(f(y), a, x)$ và $P(y, z, x) \vee Q(f(b), a, a)$
   e. $P(a, b, x) \vee Q(y, y, x)$ và $P(a, y, x) \vee Q(y, f(b), x)$
   f. $R(a, c, c) \vee P(y, f(y), a)$ và $R(a, c, x)$
   g. $P(a, f(a), g(b)) \vee Q(x, a, f(y))$ và $P(a, f(y), g(x)) \vee Q(x, a, f(a))$
   h. $P(a, f(a), g(b)) \vee Q(x, a, f(y))$ và $P(a, f(y), g(x)) \vee Q(x, f(a), g(z))$

4. Hãy tìm đồng nhất thể lớn nhất (nếu có) cho các cặp công thức cho trong bài 33 trên đây.

5. Hãy chuyển các công thức sau đây về dạng tuyển
   a. $\forall x(P(x, a) \supset \exists y P(x, y))$
   b. $\exists x(P(x, y) \,\&\, \forall x(Q(x) \supset P(a, y))$
   c. $\exists x \forall y(R(x, y) \supset R(y, x))$
   d. $\forall x(P(x, y) \supset \exists z R(x, y, z))$
   e. $(\forall x P(a) \,\&\, \exists y Q(b, y)) \vee \neg (P(x) \supset Q(x))$
   f. $\neg \exists x(P(x) \,\&\, \neg Q(x))$
   g. $\neg \forall x \exists y(R(x, a) \supset R(a, y))$

6. Cho các tiền đề
   (i). $\forall x \forall y(P(y, z) \supset \neg P(z, y))$
   (ii). $\forall x(P(b, x) \supset P(a, x))$
   (iii). $P(b, c) \vee P(a, c)$

   Chứng minh rằng từ các tiền đề đã cho có thể rút ra $\neg P(c, a)$.

7. Cho các tiền đề (ở đây các chữ cái in hoa là ký hiệu vị từ, x, y, z là các biến đối tượng, a, b, c là các hằng đối tượng) :
   i. $\{\neg H(x), \neg D(y), F(x, y)\}$
   ii. $\{G(a)\}$
   iii. $\{\neg R(z), F(a, z)\}$
   iv. $\{\neg G(y), D(y)\}$
   v. $\{\neg F(x, y), \neg F(y, z), F(x, z)\}$
   vi. $\{H(b)\}$
   vii. $\{R(c)\}$
   Hãy dùng hợp giải để xác định xem từ các tiền đề này có thể rút ra kết luận $F(b, c)$ không. Nếu được, hãy rút ra kết luận đó.

8. Hãy dùng hợp giải để chứng minh tính đúng đắn của các tam đoạn luận đơn ở bài 30.

9. Mai là em của Bình. Bình là con của bà Ba và ông Bảy. Hạnh là con của ông Vinh và là em của Hải. Hải và Bình là anh em cùng mẹ. Biết rằng hai người cùng cha hoặc cùng mẹ là anh em (hay chị em), hãy xác định xem:
   a. Mai và Hạnh có là chị em (anh em) không ?
   b. Những người nào là anh em với Bình ?
   c. Những ai là con của bà Ba ?

10. Hãy dùng hợp giải để kiểm tra xem các suy luận sau đây có đúng không:
   a. Mọi loài cá đều sống dưới nước. Cá voi sống dưới nước. Vậy cá voi là cá.
   b. Sinh viên nào học giỏi logic cũng học giỏi tin học và giỏi toán. Một số sinh viên giỏi toán nhưng không thích tin học. Ai không thích tin học thì không học giỏi tin học. Những người học giỏi tin học dễ xin được việc làm. Mai không thích học logic, nhưng thích học toán. Bình thích học toán nhưng học tin học dở. Vậy Mai dễ xin được việc làm, Bình không dễ xin được việc làm.

BÀI TẬP CHƯƠNG 4

11. Hãy áp dụng phép thế $\gamma = \{x \leftarrow a, y \leftarrow b, z \leftarrow f(b), w \leftarrow f(y)\}$ vào các công thức cho sau đây:
   g. $P(a, f(a), g(b)) \vee Q(x, a, f(y))$
   h. $P(f(x), a, y, w, g(z))$
   i. $P(f(g(b, w)), y, g(a, z), z)$
   j. $P(a, g(y, z), x) \vee Q(y, f(b), x)$
   k. $R(a, x, x) \vee P(y, f(y), y)$
   l. $\neg R(f(g(a)), y, g(y, z), f(w))$

12. Tìm phép thế hợp của các phép thế
   g. $\{x \leftarrow a, y \leftarrow b, z \leftarrow c\}$ và $\{x \leftarrow a, u \leftarrow b, w \leftarrow c\}$
   h. $\{x \leftarrow u, y \leftarrow f(x), z \leftarrow c\}$ và $\{x \leftarrow a, u \leftarrow b, w \leftarrow y\}$
   i. $\{x \leftarrow f(u), y \leftarrow f(x), z \leftarrow w\}$ và $\{x \leftarrow a, u \leftarrow b, z \leftarrow b, w \leftarrow f(y)\}$
   j. $\{y \leftarrow f(x), z \leftarrow f(u)\}$ và $\{x \leftarrow a, u \leftarrow b, z \leftarrow b, w \leftarrow f(y)\}$
   k. $\{x \leftarrow z, y \leftarrow b\}, \{z \leftarrow a\}$ và $\{y \leftarrow x, z \leftarrow b, u \leftarrow a, w \leftarrow f(u)\}$
   l. $\{x \leftarrow z, y \leftarrow b, z \leftarrow f(x)\}, \{y \leftarrow f(z), z \leftarrow f(a)\}$ và $\{u \leftarrow a, w \leftarrow a\}$

13. Tìm các đồng nhất thể (nếu có) cho các cặp công thức
   i. $P(a, x)$ và $P(y, b)$
   j. $Q(a, x, y)$ và $\neg Q(x, y, b)$
   k. $P(f(x), a, y)$ và $P(f(b), y, z)$
   l. $P(a, x, b) \vee Q(f(y), a, x)$ và $P(y, z, x) \vee Q(f(b), a, a)$
   m. $P(a, b, x) \vee Q(y, y, x)$ và $P(a, y, x) \vee Q(y, f(b), x)$
   n. $R(a, c, c) \vee P(y, f(y), a)$ và $R(a, c, x)$
   o. $P(a, f(a), g(b)) \vee Q(x, a, f(y))$ và $P(a, f(y), g(x)) \vee Q(x, a, f(a))$
   p. $P(a, f(a), g(b)) \vee Q(x, a, f(y))$ và $P(a, f(y), g(x)) \vee Q(x, f(a), g(z))$

14. Hãy tìm đồng nhất thể lớn nhất (nếu có) cho các cặp công thức cho trong bài 33 trên đây.

15. Hãy chuyển các công thức sau đây về dạng tuyển
   h. $\forall x(P(x, a) \supset \exists yP(x, y))$
   i. $\exists x(P(x, y) \& \forall x(Q(x) \supset P(a, y))$
   j. $\exists x \forall y(R(x, y) \supset R(y, x))$
   k. $\forall x(P(x, y) \supset \exists zR(x, y, z))$
   l. $(\forall xP(a) \& \exists yQ(b, y)) \vee \neg (P(x) \supset Q(x))$
   m. $\neg\exists x(P(x) \& \neg Q(x))$
   n. $\neg \forall x\exists y(R(x, a) \supset R(a, y))$

16. Cho các tiền đề
   (iv). $\forall x \forall y(P(y, z) \supset \neg P(z, y))$
   (v). $\forall x(P(b, x) \supset P(a, x))$
   (vi). $P(b, c) \vee P(a, c)$

   Chứng minh rằng từ các tiền đề đã cho có thể rút ra $\neg P(c, a)$.

17. Cho các tiền đề (ở đây các chữ cái in hoa là ký hiệu vị từ, x, y, z là các biến đối tượng, a, b, c là các hằng đối tượng) :
   i. $\{\neg H(x), \neg D(y), F(x, y)\}$
   ii. $\{G(a)\}$
   iii. $\{\neg R(z), F(a, z)\}$
   iv. $\{\neg G(y), D(y)\}$
   v. $\{\neg F(x, y), \neg F(y, z), F(x, z)\}$
   vi. $\{H(b)\}$
   vii. $\{R(c)\}$
   Hãy dùng hợp giải để xác định xem từ các tiền đề này có thể rút ra kết luận $F(b, c)$ không. Nếu được, hãy rút ra kết luận đó.

18. Hãy dùng hợp giải để chứng minh tính đúng đắn của các tam đoạn luận đơn ở bài 30.

19. Mai là em của Bình. Bình là con của bà Ba và ông Bảy. Hạnh là con của ông Vinh và là em của Hải. Hải và Bình là anh em cùng mẹ. Biết rằng hai người cùng cha hoặc cùng mẹ là anh em (hay chị em), hãy xác định xem:
   d. Mai và Hạnh có là chị em (anh em) không ?
   e. Những người nào là anh em với Bình ?
   f. Những ai là con của bà Ba ?

20. Hãy dùng hợp giải để kiểm tra xem các suy luận sau đây có đúng không:
   c. Mọi loài cá đều sống dưới nước. Cá voi sống dưới nước. Vậy cá voi là cá.
   d. Sinh viên nào học giỏi logic cũng học giỏi tin học và giỏi toán. Một số sinh viên giỏi toán nhưng không thích tin học. Ai không thích tin học thì không học giỏi tin học. Những người học giỏi tin học dễ xin được việc làm. Mai không thích học logic, nhưng thích học toán. Bình thích học toán nhưng học tin học dở. Vậy Mai dễ xin được việc làm, Bình không dễ xin được việc làm.

Chương 5 RELEVANT LOGIC

I. Dẫn nhập

Relevant logic là một ngành logic học hiện đại, bắt đầu phát triển từ khi công trình "Bergrundung einer Strengen Implikation" của W. Ackermann được công bố vào năm 1956[1].
Ackermann đã đặt ra vấn đề xác định quan hệ suy diễn logic (quan hệ giữa tiền đề và kết luận trong suy luận logic) như là quan hệ về mặt nội dung giữa các mệnh đề. Vấn đề này đã định hướng và liên kết các nghiên cứu tiếp theo của nhiều nhà logic.
Ngoài vấn đề mà Ackermann đã nêu nói trên, đối với Relevant logic còn một vấn đề hết sức quan trọng nữa – đó là vấn đề xác định, vấn đề chuẩn hóa khái niệm liên hệ điều kiện ("nếu … thì …"). Mà thật ra, hai vấn đề này có mối liên hệ rất chặt chẽ với nhau. Hơn thế nữa, mối liên hệ điều kiện được nói đến trong, ví dụ, các quy luật khoa học (nơi nó biểu thị quan hệ xác định lẫn nhau giữa các tính chất, trạng thái) chính là mối quan hệ suy diễn logic, vốn gắn chặt với quan hệ suy diễn relevant.
Vì vấn đề chuẩn hóa, xác định quan hệ suy diễn logic là một trong những vấn đề trọng tâm, quan trọng (nếu không nói là quan trọng nhất) của khoa học logic, nên ngành logic relevant là một trong những ngành logic quan trọng và có ý nghĩa nhất. Điều này còn thể hiện rõ hơn trong thời đại tin học, khi con người muốn xây dựng các hệ thống máy tính, chế tạo các các robot "thông minh", tư duy giống như con người.

II. Nghịch lý của suy diễn logic và phép toán kéo theo

1. Các nghịch lý

Trong các hệ logic, thông thường quan hệ suy diễn logic gắn liền với phép toán kéo theo (implication). Quy luật logic (tức là công thức hằng đúng) $A \supset B$ có nghĩa là B suy ra được từ A, B là hệ quả của A. (Nhắc lại, để kiểm tra xem một suy luận nào đó có đúng hay không, ta viết các tiền đề và kết luận của suy luận đó ra dưới dạng các công thức. Nối các công thức tiền đề với nhau bởi dấu toán hội, ta được phần tiền đề A. Phần kết luận được biểu thị bởi công thức B. Ta nối phần tiền đề và phần kết luận bằng dấu kéo theo (implication), và được công thức biểu thị toàn bộ suy luận $A \supset B$. Nếu $A \supset B$ là công thức hằng đúng,- tức là quy luật logic -, thì suy luận là đúng, hay nói cách khác, thì B suy ra được từ A.

Cơ sở của cách hiểu này về quan hệ suy diễn logic bắt nguồn từ tư tưởng của Frege G., được ông trình bày lần đầu tiên trong tác phẩm "Phép tính khái niệm", công bố năm 1879.

Chúng ta đã biết rằng, phép toán kéo theo chỉ được định nghĩa như là một phụ thuộc hàm giữa các giá trị chân lý của A và của B trong $A \supset B$, chứ không thể hiện được quan hệ "nếu …

[1] Tc phẩm ñy cơng bố trong Journal of Symbol Logic (J. S. L.) 1956, Vol. 21

thì …” của ngôn ngữ tự nhiên. Chính vì vậy $A \supset B$ chỉ sai trong một trường hợp duy nhất, khi A đúng, nhưng B sai. Ở đây vấn đề liệu giữa A và B có mối quan hệ về mặt nội dung nào không hoàn toàn không quan trọng.

Vậy nên khi dùng implication để biểu thị quan hệ suy diễn logic thì ta gặp phải các nghịch lý.

Các nghịch lý quan trọng nhất là:

a) *Quy luật logic có thể suy ra từ bất cứ mệnh đề nào.*

Ví dụ: Ta rút ra quy luật triệt tam $A \vee \neg A$ từ mệnh đề bất kỳ B.

1^{+} B	
2^{+} $\neg (A \vee \neg A)$	
3^{+} A	
4. $A \vee \neg A$	3, $\vee_i$
5. $\neg A$	2, 3, 4, $\neg_i$
6. $A \vee \neg A$	5, $\vee_i$
7. $\neg \neg (A \vee \neg A)$	2, 6, $\neg_i$
8. $A \vee \neg A$	7, $\neg_e$

Từ đây $B \vdash A \vee \neg A$

b). *Từ mâu thuẫn logic có thể suy ra bất cứ mệnh đề nào.*

Ví dụ: Ta suy ra mệnh đề B từ mâu thuẫn $A \wedge \neg A$

1^{+} $A \wedge \neg A$	
2^{+} $\neg B$	
3. A	1, $\wedge_e$
4. $\neg A$	1, $\wedge_e$
5. $\neg \neg B$	2, 3, 4, $\neg_i$
6. B	5, $\neg_e$

Từ đây $A \wedge \neg A \vdash B$

2. Các cố gắng giải quyết nghịch lý

Để giải quyết các nghịch lý nêu trên của quan hệ kéo theo cổ điển, người ta đã xây dựng rất nhiều hệ thống logic phi cổ điển, với các quan hệ kéo theo khác. Đáng lưu ý nhất là các hệ logic trực quan, các hệ logic với quan hệ kéo theo chặt (strenge implication) của Ackermann và hình thức sửa đổi của nó trong hệ E (entailment) của Anderson A. R. và Belnap N. D., các hệ kéo theo chặt của Lewis C. I.[2], hệ Analytic Implication (kéo theo phân tích) của Parry W. T.

Trong các hệ của Anderson và Belnap, cùng với các nghịch lý, một số quy tắc suy luận khác vốn thường được coi là hiển nhiên đúng, cũng bị loại khỏi hệ.

[2] Là các hệ logic tình thái $S_2 - S_4$.

Ví dụ, công thức $((A \vee B) \wedge \neg A) \supset B$ (thường gọi là *quy tắc* γ) không đúng trong các hệ này.

Khi xây dựng các hệ logic của mình, Lewis đã xuất phát từ một quan điểm rất đúng đắn, theo đó thì nếu chỉ đòi hỏi "từ các tiền đề đúng phải suy ra các kết luận đúng" để xác định khái niệm suy diễn logic thì chưa đủ. Cần phải đòi hỏi thêm rằng giữa các tiền đề và kết luận phải có mối liên hệ về nội dung! Tiếc rằng Lewis đã không làm rõ khái niệm "liên hệ về nội dung". Ngoài ra, các hệ logic của Lewis vẫn chứa những nghịch lý tương tự như các nghịch lý của phép toán kéo theo cổ điển.

Trong nền tảng của hệ logic với analytic implication của Parry có ý tưởng rất hay: Nếu công thức B suy ra được từ công thức A thì công thức B chỉ chứa các biến mệnh đề có trong A. Bằng cách đó Parry đã thể hiện được đòi hỏi rằng giữa tiền đề và kết luận phải có mối liên hệ về mặt nội dung.

Write, Smily và Gite đưa ra tiêu chuẩn (gọi là tiêu chuẩn WGS) đối với quan hệ suy diễn như sau:

Từ A suy ra B khi và chỉ khi:

i) $A \supset B$ là một trường hợp riêng của công thức hằng đúng $A' \supset B'$, và

ii) A' không phải là mâu thuẫn logic, còn

iii) B' không phải là quy luật logic.

Tất cả các cách giải quyết nghịch lý nêu trên đây đều không đáp ứng được yêu cầu chuẩn hóa khái niệm suy diễn logic, vì chúng không thể đảm bảo cho các hệ mới tránh được nghịch lý. Lý do là ta vẫn chưa xác định được nguyên nhân gây ra nghịch lý.

III. Suy diễn logic – một quan hệ về nội dung, hay là về thông tin giữa các mệnh đề

1. Các đặc điểm của quan hệ suy diễn logic

Các đặc điểm chủ yếu sau đây của quan hệ suy diễn logic được thừa nhận rộng rãi:

Thứ nhất: Quan hệ suy diễn logic giữa các mệnh đề không phụ thuộc vào các từ mô tả trong các mệnh đề đó. Nói cách khác, đó là quan hệ chỉ phụ thuộc vào hình thức logic của các mệnh đề, nghĩa là quan hệ giữa nội dung logic của các mệnh đề.

Thứ hai: $A \models B$ có nghĩa là với mọi nghĩa của các từ mô tả trong các hình thức logic A và B, B luôn luôn đúng nếu A đúng.

Thứ ba: $A \models B$ khi và chỉ khi nôi dung logic của B là một phần nội dung logic của A.

Xuất hiện vấn đề: phải xác định cái gì là *nội dung logic* của mệnh đề.

2. Thông tin của mệnh đề. Nội dung logic là thông tin

Ta định nghĩa nội dung logic là thông tin ngữ nghĩa chứa trong hình thức logic, nghĩa là chứa trong công thức logic. Đó là thông tin được xác định bởi định nghĩa của các phép toán logic có mặt trong mệnh đề.

<u>Định nghĩa 1</u>. Thông tin của mệnh đề A là mức độ hạn chế được tập hợp M tất cả các khả năng thành tập hợp các khả năng M_A, $M_A \subseteq M$, khi thừa nhận A đúng (nói cách khác: khi có thông báo A).

Ví dụ: Ta gieo một con xúc xắc. Khi chưa có một thông báo nào về kết quả, nghĩa là chưa có thông tin nào, thì ta chỉ biết rằng một trong sáu trường hợp sẽ xảy ra. Đó là trường hợp được mặt 1 chấm, được mặt 2 chấm, …, 6 chấm. Tập hợp M tất cả các khả năng ban đầu ở đây là:

M = {được mặt 1 chấm, được mặt 2 chấm, được mặt 3 chấm, được mặt 4 chấm, được mặt 5 chấm, được mặt 6 chấm}

Giả sử A là thông báo: "Đã được mặt có số chấm chẵn". Khi có A ta có thể hạn chế M xuống còn M_A :

M_A = { được mặt 2 chấm, được mặt 4 chấm, được mặt 6 chấm}

Giả sử B là thông báo: "Đã được mặt có số chấm chẵn, chia hết cho 3", thì ta có thể, sau khi thừa nhận B, hạn chế tập M xuống còn tập M_B :

M_B = { được mặt 6 chấm}.

Thông báo cho phép ta hạn chế được tập hợp M càng nhiều thì chứa càng nhiều thông tin. Trong ví dụ trên đây, thông báo B cho phép hạn chế tập M nhiều hơn so với thông báo A, nên B chứa nhiều thông tin hơn.

Rõ ràng là chỉ có thể so sánh thông tin của hai mệnh đề khi tập M ban đầu là chung cho cả hai mệnh đề đó. Ở đây người ta so sánh được thông tin của A (ký hiệu inf(A)) với thông tin của thông báo B (ký hiệu là inf(B)) vì tập hợp M các khả năng ban đầu là chung cho cả hai thông báo A và B. Như vậy, ta có thể nói đến thông tin của thông báo A ứng với tập hợp các khả năng ban đầu M, ký hiệu là inf(A,M), và biểu thị nó bằng cặp $< M_A, M >$, trong đó $M_A \subseteq M$, và A đúng với mọi trường hợp thuộc M_A. Vì chúng ta quan tâm đến thông tin của các mệnh đề, và chỉ sử dụng ngôn ngữ logic mệnh đề, nên có thể sử dụng tập hợp tất cả các mô tả trạng thái (state description) làm tập hợp các khả năng M.

Định nghĩa 2. Giả sử trong ngôn ngữ của chúng ta chỉ có tập đếm được các biến mệnh đề p_1, p_2, …, p_n, … Khi đó tập hợp $\{\overline{p_1}, \overline{p_2}, \overline{p_3}, ..., \overline{p_n}\}$, với $\overline{p_i} = p_i$ hoặc $\overline{p_i} = \neg p_i$ (i = 1, 2,…), gọi là một mô tả trạng thái (cổ điển).

Một state discription như vậy chính là một mô tả về một thế giới có thể (possible world) nào đó. Ví dụ: Một lớp học có 3 sinh viên. Khi đó nếu muốn mô tả sự có mặt hay vắng mặt của sinh viên lớp học này ta chỉ cần sử dụng các mệnh đề p_1, p_2, p_3 và các mệnh đề phủ định của chúng.

p_1 có nghĩa là sinh viên thứ nhất có mặt

$\neg p_1$ có nghĩa là sinh viên thứ nhất vắng mặt

p_i có nghĩa là sinh viên thứ i có mặt,

$\neg p_i$ có nghĩa là sinh viên thứ i vắng mặt.

Với ngôn ngữ này, mô tả trạng thái $\{ p_1, p_2, \neg p_3\}$ cho biết các sinh viên thứ nhất và thứ 2 có mặt, còn sinh viên thứ 3 vắng mặt. Còn mô tả trạng thái $\{p_1, \neg p_2, \neg p_3\}$ cho biết sinh viên thứ nhất có mặt, hai sinh viên còn lại vắng mặt.

Trong ví dụ này, nếu mô tả thêm tình hình học tập của các sinh viên thì ta có thể thêm các mệnh đề $q_1, q_2, q_3, \neg q_1, \neg q_2, \neg q_3$, trong đó q_i có nghĩa là sinh viên thứ i học tốt, $\neg q_i$ có nghĩa là sinh viên thứ i học không tốt (i = 1, 2, 3). Lúc này, mô tả trạng thái

$\{ p_1, p_2, \neg p_3, q_1, \neg q_2, q_3\}$ cho biết: sinh viên thứ nhất và thứ 2 có mặt, sinh viên thứ 3 vắng mặt. Sinh viên thứ nhất và thứ 3 học tốt, sinh viên thứ 2 học không tốt. Tập hợp các khả năng ban đầu là tập tất cả các mô tả trạng thái:

$M = \{\alpha : \alpha$ là mô tả trạng thái$\}$.

Nếu trong ngôn ngữ có cả thảy n mệnh đề đơn, không kể các mệnh đề phủ định, thì số lượng các state description là 2^n.

Vì tập M của ta là chung cho tất cả các mệnh đề, nên thay vì viết inf(A,M), ta chỉ cần viết inf(A) như đã từng dùng trên kia.

Định nghĩa 3: Ký hiệu TA/α và FA/α tương ứng là mệnh đề A đúng trong mô tả trạng thái (thế giới có thể có) α, và mệnh đề A sai trong mô tả trạng thái (thế giới có thể có) α. Khi đó:

Với mọi biến mệnh đề p_i

3.1. $Tp_i / \alpha \Leftrightarrow p_i \in \alpha$

3.2. $Fp_i / \alpha \Leftrightarrow \neg p_i \in \alpha$

Với mọi công thức A và B:

3.3. $T(A \wedge B) / \alpha \Leftrightarrow TA / \alpha$ và TB / α

3.4. $F(A \wedge B) / \alpha \Leftrightarrow FA / \alpha$ hay FB / α

3.5. $T(A \vee B) / \alpha \Leftrightarrow TA / \alpha$ hay TB / α

3.6. $T(A \vee B) / \alpha \Leftrightarrow TA / \alpha$ và TB / α

3.7. $T(\neg A) / \alpha \Leftrightarrow FA / \alpha$

3.8. $F(\neg A) / \alpha \Leftrightarrow TA / \alpha$

3.9. $T(A \supset B) / \alpha \Leftrightarrow$ nếu TA / α thì TB / α [3]

3.10. $F(A \supset B) / \alpha \Leftrightarrow TA / \alpha$ và FB / α

Dễ thấy rằng định nghĩa 3 trên đây tương đương với định nghĩa thông qua bảng chân lý của các phép toán logic tương ứng trong logic mệnh đề cổ điển.

Định nghĩa 4. Giả sử A, B là các mệnh đề, khi đó:

4.1 $M_A = \{\alpha : TA/ \alpha\}$, α là state description;

4.2 $Inf(A,M) = Inf(A) = (M_A, M)$

4.3 $Inf(B) \subseteq Inf(A) \Leftrightarrow M_A \subseteq M_B$

[3] Nhiều ťc giả cho rằng dấu $\supset$ chỉ được pĥp cĩ mặt một lần duy nhất trong cơng thức, ňn khơng cần các định nghĩa 3.9 ṽ 3.10.

4.4 $A \models B \Leftrightarrow Inf(B) \subseteq Inf(A)$.

3. Nguyên nhân của nghịch lý suy diễn logic (cổ điển)

Các định nghĩa 2, 3 và 4 ở phần trên đã cho phép ta xác định được nguyên nhân, hay nguồn gốc, của các nghịch lý của quan hệ suy diễn logic.

Theo định nghĩa 2, ta thấy với mọi mô tả trạng thái α, mọi mệnh đề đơn p_i,:

$p_i \in \alpha$ hoặc $\neg p_i \in \alpha$ (*)

$p_i \notin \alpha$ hoặc $\neg p_i \notin \alpha$ (**)

Từ đây, theo định nghĩa 3 dễ dàng suy ra, với mọi mô tả trạng thái α, và với mọi mệnh đề A, ta có TA / α hoặc FA / α. "Hoặc" ở đây được hiểu nghiêm ngặt, nghĩa là không thể có đồng thời TA / α và FA / α.

Chính vì vậy, quy luật $A \vee \neg A$ đúng trong mọi mô tả trạng thái α. Nói cách khác:

$M_{(A \vee \neg A)} = M$ (Theo định nghĩa 4.1).

Nghĩa là khi có thông báo $A \vee \neg A$, ta không hề hạn chế được tập M. Và như vậy, quy luật $A \vee \neg A$ không hề chứa thông tin !

Hơn nữa, công thức mâu thuẫn $A \wedge \neg A$ không đúng trong bất cứ mô tả trạng thái nào, nghĩa là theo định nghĩa 4.1. thì $M_{(A \wedge \neg A)} = \phi$. Tập hợp rỗng là tập con của bất cứ tập hợp nào, nên ta có $\phi \subseteq M_{(B)}$ với mọi mệnh đề B. Từ đây, $Inf(B) \subseteq inf(A \wedge \neg A)$, với B bất kỳ.

Vậy, $A \wedge \neg A \models B$.

Rõ ràng vì có (*) nên $A \vee \neg A$ mới không chứa thông tin. Nói chính xác hơn, $A \vee \neg A$ không chứa thêm điều gì mới so với thông tin đã có trong bản thân khái niệm mô tả trạng thái cổ điển, thể hiện ra qua (*). Tương tự như vậy, vì có (**) nên $(A \wedge \neg A)$ mới chứa nhiều thông tin hơn bất cứ mệnh đề nào.

Vậy, để loại bỏ các nghịch lý, chỉ cần loại bỏ (*) và (**) từ định nghĩa mô tả trạng thái. (*) và (**) chính là nguồn gốc của các nghịch lý này. Khi đòi hỏi mô tả trạng thái phải thỏa mãn (*) và (**), ta đã không còn xác định thông tin của chính các mệnh đề nữa, mà xác định nội dung mới của chúng so với nội dung đã có trong bản thân các mô tả trạng thái.

IV. Quan hệ suy diễn logic relevant

Như phần trên ta đã thấy, nếu tập M tất cả các khả năng ban đầu là tập hợp tất cả các mô tả trạng thái cổ điển thì một chân lý logic không chứa thông tin, và vì vậy có thể suy ra được từ bất kỳ một mệnh đề nào khác. Ngược lại, một mâu thuẫn logic, vì không đúng trong bất kỳ một mô tả trạng thái nào, nên chứa toàn bộ lượng thông tin có thể chuyển đạt được bằng ngôn ngữ đang xét. Vì vậy, từ một mâu thuẫn logic có thể suy ra bất cứ mệnh đề nào.

Để giải quyết vấn đề này,- cũng có nghĩa là để biến quan hệ suy diễn logic cổ điển thành quan hệ suy diễn relevant -, cần phải thay đổi tập M. Giáo sư Voisvillo đưa ra cách giải quyết như sau: Vẫn coi tập M là tập tất cả các mô tả trạng thái, nhưng bản thân khái niệm mô tả trạng thái thì phải được thay đổi, loại bỏ các yêu cầu (*) và (**).

Định nghĩa 5. Giả sử ngôn ngữ có các mệnh đề $p_1, p_2, p_3, \ldots$ (một tập hợp đếm được). Ta thiết lập tập hợp $K = \{ p_1, \neg p_1, p_2, \neg p_2, p_3, \neg p_3, \ldots\}$. Khi đó *một mô tả trạng thái relevant* (còn gọi là *mô tả trạng thái mở rộng*) là tập hợp con bất kỳ của tập hợp K.

So sánh với mô tả trạng thái cổ điển ta thấy:

i) Mọi mô tả trạng thái cổ điển là mô tả trạng thái mở rộng;

ii) Các yêu cầu (*) và (**) không đúng với mô tả trạng thái mở rộng. Ví dụ, $\alpha = \{p_1\}$ là mô tả trạng thái mở rộng, nhưng $p_2 \notin \alpha$, $\neg p_2 \notin \alpha$.

Tập hợp tất cả các khả năng M là tập hợp tất cả các mô tả trạng thái mở rộng. Nghĩa là: $M = \{\alpha / \alpha \subseteq K\}$.

Ký hiệu TA / α hoặc FA / α như trong định nghĩa 3 (với hiệu chỉnh α là mô tả trạng thái mở rộng) ta có :

Định nghĩa 6. Với mọi biến mệnh đề p_i, mọi mô tả trạng thái mở rộng α, mọi mệnh đề A, B:

6.1. $Tp_i /\alpha \Leftrightarrow p_i \in \alpha$

6.2. $Fp_i /\alpha \Leftrightarrow p_i \notin \alpha$

6.3. $T\neg p_i /\alpha \Leftrightarrow \neg p_i \in \alpha$

6.4. $F\neg p_i /\alpha \Leftrightarrow \neg p_i \notin \alpha$

6.5. $T(A \wedge B) / \alpha \Leftrightarrow TA / \alpha$ và TB / α

6.6. $F(A \wedge B) / \alpha \Leftrightarrow FA / \alpha$ hoặc FB / α

6.7. $T(A \vee B) / \alpha \Leftrightarrow TA / \alpha$ hoặc TB / α

6.8. $F(A \vee B) / \alpha \Leftrightarrow FA / \alpha$ và FB / α

6.9. $T\neg A / \alpha \Leftrightarrow FA / \alpha$

6.10. $F\neg A / \alpha \Leftrightarrow TA / \alpha$

6.11. $T(A \supset B) / \alpha \Leftrightarrow TA / \alpha \Rightarrow TB / \alpha$

6.12. $F(A \supset B) / \alpha \Leftrightarrow TA / \alpha$ và FB / α

Định nghĩa 4 ở phần trên được giữ nguyên cho quan hệ suy diễn relevant, chỉ thay tập M trong nó bằng tập M tất cả các mô tả trạng thái mở rộng.

Từ các định nghĩa 5 và 6 ta có các nhận xét:

iii) $\forall p_i \exists \alpha (\alpha \in M \ \& \ Tp_i / \alpha)$

iv) $\forall p_i \exists \alpha (\alpha \in M \ \& \ Fp_i / \alpha)$

v) $\forall p_i \exists \alpha (\alpha \in M \ \& \ Tp_i / \alpha \ \& \ Fp_i / \alpha)$.

Thật vậy, vì $\alpha = \{p_i, \neg p_i\} \in M$ nên theo định nghĩa 6, Tp_i / α và Fp_i / α.

Quy nạp theo số lượng dấu toán logic trong công thức, ta dễ dàng suy ra từ đây: Nếu A là công thức logic cổ điển , nghĩa là nếu A không chứa dấu kéo theo $\supset$ (dấu $\supset$ được coi là cách viết tắt thay cho các dấu $\neg$ và $\vee$), thì *vi)* $\forall A\ \exists\alpha\ (\alpha \in M\ \&\ TA/\alpha\ \&\ FA/\alpha)$.

Như vậy, mọi công thức cổ điển đều mang một thông tin xác định: Rõ ràng từ nhận xét *vi)* với mọi công thức A ta có $M_A \neq \varnothing$, $M_A \neq M$. Từ đây, theo mục 4.2 của định nghĩa 4, ta có $inf(A) \neq 0$ và $inf(A) \neq U$ (Với U là toàn bộ thông tin có thể biểu thị được bằng ngôn ngữ mà ta đang dùng).

V. Xác định về mặt hình thức nguyên nhân nghịch lý của khái niệm kéo theo logic cổ điển. Hệ logic tự nhiên relevant

1. Nguyên nhân nghịch lý của khái niệm kéo theo logic cổ điển về mặt hình thức

Ở phần trên ta đã phân tích nguồn gốc nghịch lý của khái niệm kéo theo logic từ quan điểm thông tin, nghĩa là từ mặt ngữ nghĩa. Để có thể xây dựng hệ logic phù hợp với quan hệ kéo theo relevant, ta phải xác định nguồn gốc của các nghịch lý đó từ mặt hình thức. Nói cách khác, ta phải xác định xem tại sao khi sử dụng các quy tắc suy luận rất đơn giản, và có vẻ như rất hiển nhiên của hệ suy luận tự nhiên cổ điển thì lại làm xuất hiện các nghịch lý.

Giáo sư Voisvillo E. K. là người đầu tiên đã xây dựng các hệ logic cổ điển với khái niệm phụ thuộc của công thức vào giả định. Ông phân tích các hệ này để tìm ra nguồn gốc nghịch lý. Vấn đề này được ông trình bày rõ ràng nhất trong tác phẩm Voisvillo E. K. "Các khía cạnh triết học – phương pháp luận của logic relevant" [4].

Như đã nói, khi xây dựng các hệ này, ta sử dụng khái niệm " sự phụ thuộc của công thức vào các giả định và giả thiết". Trong các chuỗi suy diễn những sự phụ thuộc này được biểu thị dưới dạng tường minh bằng cách gán cho mỗi công thức một đặc điểm phụ thuộc.

Đặc điểm phụ thuộc của công thức A trong chuỗi suy diễn là một tập hợp các giả thiết và giả định mà công thức đó phụ thuộc vào trong suy luận. Như vậy, mỗi bước trong suy diễn được viết dưới dạng $A[\ \Gamma\ ]$ (đọc là: công thức A có đặc điểm phụ thuộc Γ).

Để ý rằng, chuỗi suy diễn trong hệ suy luận tự nhiên bao giờ cũng bắt đầu bằng giả thiết hay giả định. Sau đây là một số hệ logic với đặc điểm phụ thuộc do giáo sư Voisvillo e. K xây dựng.

Hệ K_1: Đặc điểm phụ thuộc (của công thức A nào đó trong chuỗi suy diễn vào các giả định và giả thiết) là một tập hợp các giả định, giả thiết (có thể là tập rỗng). Khái niệm công thức phụ thuộc vào giả định trong chuỗi suy diễn được định nghĩa quy nạp:

a) Giả định (giả thiết) A phụ thuộc vào chính nó (và nghĩa là đặc điểm phụ thuộc của nó là tập hợp $\{A\}$);
b) Sự phụ thuộc của các công thức khác được xác định theo các quy tắc suy luận.

---

[4] Voisvillo E. K. *Các khía cạnh triết học – phương pháp luận của logic relevant.* Moskva 1988. (Tiếng nga)

Trong các quy tắc sau đây ta ký hiệu A, B, C, D là công thức bất kỳ; Γ, Δ là tập các công thức (có thể rỗng). Dấu phẩy giữa các tập hợp có nghĩa là dấu hội; "Γ \ Δ" có nghĩa là kết quả việc loại bỏ các thành phần thuộc Δ ra khỏi Γ :

$\wedge_i$: $$\frac{A[\Gamma], B[\Delta]}{(A \wedge B)[\Gamma, \Delta]}$$ $\wedge_e$: $$(\frac{(A \wedge B)[\Gamma]}{A[\Gamma]}; \quad \frac{(A \wedge B)[\Gamma]}{B[\Gamma]};$$

$\vee_i$: $$\frac{A[\Gamma]}{(A \vee B)[\Gamma]}; \quad \frac{B[\Gamma]}{(A \vee B)[\Gamma]};$$ $\vee_e$: $$\frac{C[\Gamma, A], C[\Delta, B]}{C[\Gamma, \Delta, A \vee B]}$$

Trong đó A $\vee$ B là giả định có trước khi sử dụng quy tắc này trong chuỗi suy diễn.

$\supset_i$: $$\frac{B[\Gamma]}{(A \supset B)[\Gamma \setminus \{A\}]}$$ với A là giả định bất kỳ trong chuỗi suy diễn.

$\supset_e$: $$\frac{(A \supset B)[\Gamma], A[\Delta]}{B[\Gamma, \Delta]};$$ $\neg_e$: $$\frac{\neg\neg A[\Gamma]}{A[\Gamma]}$$

$\neg_e$: $$\frac{\neg B[\Gamma], B[\Delta]}{\neg A[\Gamma, \Delta \setminus \{A\}]}$$ với A là giả định bất kỳ trong chuỗi suy diễn.

Chuỗi suy diễn là dãy các công thức (với đặc điểm phụ thuộc), trong đó mỗi công thức hoặc là một giả định, một giả thiết, hoặc là nhận được từ các công thức trước nó bằng cách sử dụng các quy tắc của hệ. Công thức cuối cùng được gọi là kết luận. Nếu kết luận là công thức A [Γ] thì :

Nếu Γ ≠ ∅, Γ = {B_1, B_2, …, B_n}, ta nói A là hệ quả của các công thức B_1, B_2, …, B_n. Viết Γ $\models$ A.

Nếu Γ = ∅ thì A là định lý, chuỗi suy diễn được gọi là phép chứng minh của công thức A.

Trong hệ này các giả định có thể được sử dụng bao nhiêu lần cũng được, và chúng không bị loại bỏ khỏi suy luận.

Ta nhận thấy rằng các quy tắc $\wedge_i$, $\supset_i$ và $\neg_e$ không đảm bảo tính liên hệ về nội dung giữa tiền đề và kết luận. Thật vậy:

Với $\wedge_i$, để ý rằng nếu trong chuỗi suy diễn, công thức giả định A không hề được sử dụng, thì kết luận của suy luận không phụ thuộc vào A. Thế nhưng bao giờ ta cũng có thể làm cho kết luận phụ thuộc vào một công thức bất kỳ, bằng cách sử dụng quy tắc $\wedge_i$, $\wedge_e$. Như thế mặc dầu không cần đến giả định A khi ta rút ra B, nhưng vẫn có thể coi là B phụ thuộc vào A.

Với $\supset_i$, nếu A $\notin \Gamma$ thì trong tiền đề B không phụ thuộc vào A, có nghĩa là B không được rút ra nhờ có A; còn ở kết luận $\Gamma \setminus \{A\} = \Gamma$, nhưng lại có $B \supset A\ [\Gamma \setminus \{A\}]$, hay $A \supset B\ [\Gamma]$ và có nghĩa là B là hệ qủa của A trong điều kiện có Γ. Nghịch lý ở đây thể hiện khá rõ ràng.

Tương tự với $\neg_e$. Nếu $A \notin \Gamma$ và $A \notin \Delta$ thì $B[\Gamma]$, $\neg B[\Delta]$ cho biết A không phải là một trong số các nguyên nhân làm xuất hiện mâu thuẫn B và $\neg$B. Thế nhưng từ đó quy tắc này lại cho phép rút ra $\neg A\ [\Gamma, \Delta \setminus \{A\}]$, cứ như A là một trong những nguyên nhân gây ra mâu thuẫn B và $\neg$B vậy. Nghịch lý đã rõ ràng.

2. Hệ suy luận tự nhiên relevant

Để chuyển sang relevant logic, ta xét hệ K_2 (tương đương K_1), trong đó những vi phạm nguyên lý relevant (nguyên lý đảm bảo sự liên hệ về nội dung giữa tiền đề và kết luận) được gom về trong một quy tắc duy nhất.

Hệ K_2: Đặc điểm phụ thuộc bây giờ được hiểu là dãy các giả định (giả thiết); Γ, Δ là dãy bất kỳ của các giả định (công thức), và cũng có thể là dãy trống.

Quy tắc:

$\wedge_i$: $\dfrac{A[\Gamma], B[\Gamma]}{(A \wedge B)[\Gamma]}$ $\qquad$ $\wedge_e$: $(\dfrac{(A \wedge B)[\Gamma]}{A[\Gamma]}$; $\dfrac{(A \wedge B)[\Gamma]}{B[\Gamma]}$;

$\vee_i$: $\dfrac{A[\Gamma]}{(A \vee B)[\Gamma]}$; $\dfrac{B[\Gamma]}{(A \vee B)[\Gamma]}$; $\qquad$ $\vee_e$: $\dfrac{C[\Gamma, A], C[\Gamma, B]}{C[\Gamma, A \vee B]}$

$\supset_i$: $\dfrac{B[\Gamma, A]}{(A \supset B)[\Gamma]}$ $\qquad$ $\supset_e$: $\dfrac{(A \supset B)[\Gamma], A[\Delta]}{B[\Gamma, \Delta]}$

$\neg_e$: $\dfrac{\neg B[\Gamma, A], B[\Delta, C]}{\neg A[\Gamma, \Delta, C^*]}$ $\qquad$ $\neg_e$: $\dfrac{\neg\neg A[\Gamma]}{A[\Gamma]}$

trong đó C^* là C nếu như C không trùng với A, và C^* là dãy trống trong trường hợp ngược lại.

$\dfrac{A[\Gamma]}{A[B, \Gamma]}$ $\qquad$ Quy tắc đơn điệu, với B là giả định bất kỳ trong chuỗi suy diễn.

$$\frac{A[\Gamma, B, B, \Delta]}{A[\Gamma, B, \Delta]}$$ Rút gọn đặc điểm phụ thuộc

$$\frac{A[\Gamma, B, C, \Delta]}{A[\Gamma, C, B, \Delta]}$$ Hoán vị đặc điểm phụ thuộc

Các khái niệm chuỗi suy diễn và phép chứng minh định nghĩa như cũ.
Nếu có chuỗi suy diễn với kết luận A [Γ] thì ta nói rằng A suy ra được từ tập Δ bất kỳ chứa tất cả các phần tử của dãy Γ.
Bây giờ nếu ta thay khái niệm trên đây thành: Nếu có chuỗi suy diễn với kết luận A[Γ] thì ta nói rằng A suy ra được từ tập công thức Δ, với Δ chứa tất cả các phần tử của Γ và Γ không rỗng (nghĩa là Γ chứa ít nhất một phần tử).
Trong hệ thống này quy tắc duy hất không đảm bảo tính liên hệ về nội dung giữa tiền đề và kết luận (tính relevant) là quy tắc đơn điệu. Như vậy, loại bỏ quy tắc này, ta sẽ tránh được các nghịch lý. Hệ logic relevant thu được từ hệ K_2 sau khi loại bỏ quy tắc đơn điệu được gọi là hệ RAO[5].

Thay đổi chút ít hệ K_2 ta được một trong những hệ logic relevant quan trọng nhất là hệ R[6]: Loại bỏ khỏi K_2 quy tắc gây nghịch lý là quy tắc đơn điệu và thêm vào quy tắc phân phối của phép hội đối với phép tuyển

$$\frac{(A \wedge (B \vee C))[\Gamma]}{((A \wedge B) \vee (A \wedge C))[\Gamma]}$$

Hệ E_n (tương đương với một hệ logic relevant nổi tiếng khác là hệ E) nhận được từ hệ trên bằng cách thay quy tắc hoán vị trong đặc điểm phụ thuộc bằng quy tắc hạn chế hoán vị trong đặc điểm phụ thuộc như sau:

$$\frac{A[\Gamma, B, (C \supset D), \Delta]}{A[\Gamma, (C \supset D), B, \Delta]}$$

Chúng tôi cho rằng cơ sở lý luận thật sự của lập trình logic chính là logic relevant. Bởi vậy, việc nghiên cứu logic relevant lại càng có ý nghĩa.

[5] Giáo sư tiến sĩ Smirnov V.A. là người đưa ra thuật ngữ ǹy, ̀v ơng cũng là người đầu tín nghĩn cứu hệ RAO.
[6] Chính xác hơn là ta được hệ suy luận tự nhiên tương đương với hệ R.

BẢNG ĐỊNH NGHĨA CÁC PHÉP TOÁN LOGIC

Phủ định

A	$\neg A$
T	F
F	T

Hội

A	B	$A \& B$
T	T	T
T	F	F
F	T	F
F	F	F

Tuyển không nghiêm ngặt

A	B	$A \vee B$
T	T	T
T	F	T
F	T	T
F	F	F

Kéo theo

A	B	$A \supset B$
T	T	T
T	F	F
F	T	T
F	F	T

Tương đương

A	B	$A \equiv B$
T	T	T
T	F	F
F	T	F
F	F	T

Tuyển nghiêm ngặt

A	B	$A \underline{\vee} B$
T	T	F
T	F	T
F	T	T
F	F	F

MỘT SỐ HẰNG ĐẲNG THỨC TRONG LOGIC MỆNH ĐỀ

1. $A + A = A$; — Luật đồng nhất, luật nuốt
2. $A . A = A$; — Luật đồng nhất, luật nuốt
3. $A + B = B + A$ — Tính chất giao hoán của phép cộng
4. $A + (B + C) = (A+ B) + C$ — Tính chất kết hợp của phép cộng
5. $A . B = B . A$ — Tính chất giao hoán của phép nhân
6. $A . (B . C) = (A . B) . C$ — Tính chất kết hợp của phép nhân
7. $A . (B + C) = A.B + A.C$ — Tính phân phối của phép cộng đối với phép nhân
8. $A + (B . C) = (A + B) . (A + C)$ — Tính phân phối của phép nhân đối với phép cộng
9. $\overline{A} + A = 1$; — Định nghĩa 1
10. $\overline{A} . A = 0$; — Định nghĩa 0
11. $\overline{\overline{A}} = A$ — Luật hoàn nguyên
12. $\overline{A + B} = \overline{A} . \overline{B}$ — Luật De Moorgan
13. $\overline{A.B} = \overline{A} + \overline{B}$ — Luật De Moorgan
14. $A.(A + B) = A$ — Luật giản lược
15. $A + (A.B) = A$ — Luật giản lược

CÁC KÝ HIỆU

Ký hiệu	Tên gọi	Ví dụ sử dụng
$\neg$	Phủ định	$\neg A$
&	Hội	A & B
$\vee$	Tuyển không nghiêm ngặt	$A \vee B$
$\underline{\vee}$	Tuyển nghiêm ngặt	$A \underline{\vee} B$
$\supset$	Kéo theo	$A \supset B$
$\equiv$	Tương đương	$A \equiv B$
$\rightarrow$	Kéo theo relevant, rút ra kết quả	$A \rightarrow B$; $\{p \vee q, \neg p \vee r\} \rightarrow q$
$\bar{\ }$	Phủ định (trong đại số Boole)	$\overline{A}$
.	Hội (trong đại số Boole)	A.B
+	Tuyển (trong đại số Boole)	A + B
$\forall$	Lượng từ toàn thể	$\forall xA(x)$
$\exists$	Lượng từ tồn tại	$\exists xA(x)$
$\leftarrow$, /	Thế bằng	$x \leftarrow t$, x/t
term	Hạn từ	*a* là một term
WFF	Công thức	$A \vee B$ là một WFF
mgu	Đồng nhất thể lớn nhất	mgu(A,B)
inf	Thông tin	Inf(A)
U	Toàn bộ thông tin	inf(A) $\neq$ U
T	Đúng	$T(A \rightarrow B)/\alpha$
F	Sai	$F(A \rightarrow B)/\alpha$
$\subseteq$	Là tập hợp con của hoặc bằng	$M_A \subseteq M_B$
$\in$	Thuộc về	$p_i \in \alpha$
$\notin$	Không thuộc về	$p_i \notin \alpha$
$\varnothing$	Tập hợp rỗng	$M_A \neq \varnothing$
$\models$	Là quy luật logic, hệ quả	$\models A$; $\Gamma, A \models B$
$\vdash$	Được chứng minh, định lý, rút ra	$\vdash A$; $\Gamma, A \vdash B$
$\Leftrightarrow$	Nghĩa là, tương đương với	Inf(A) $\subseteq$ inf (B) $\Leftrightarrow M_A \subseteq M_B$
Fail	Thất bại (trong hợp giải)	$\dfrac{p \vee q, q \vee r \vee \neg s}{fail}$
$\square$	Resolvent rỗng	$\dfrac{\neg p, \quad p}{\square}$

CÁC TIÊN ĐỀ VÀ QUY TẮC LOGIC MỆNH ĐỀ

(1) Cho A, B, C là các công thức bất kỳ của hệ S . Khi đó các công thức sau đây là tiên đề của hệ S :

(A_1) $(A \supset (B \supset A));$

(A_2) $((A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C));$

(A_3) $(\neg B \supset \neg A) \supset ((\neg B \supset A) \supset B))$

(2) Quy tắc suy diễn duy nhất của S là Modus Ponens:

$$MP \quad \frac{A \supset B, A}{B}$$

CÁC QUY TẮC CỦA HỆ SUY LUẬN TỰ NHIÊN TRONG LOGIC MỆNH ĐỀ

Quy tắc nhập & (ký hiệu $\&_i$) $\quad \dfrac{A, B}{A \& B}$

Quy tắc khử & (ký hiệu $\&_e$) $\quad \dfrac{A \& B}{A}; \quad \dfrac{A \& B}{B}$

Quy tắc nhập $\vee$ (ký hiệu $\vee_i$) $\quad \dfrac{A}{A \vee B}; \quad \dfrac{B}{A \vee B}$

Quy tắc khử $\vee$ (ký hiệu $\vee_e$) $\quad \dfrac{A \vee B, \neg A}{B}; \quad \dfrac{A \vee B, \neg B}{A}$

Quy tắc nhập $\neg$ (ký hiệu $\neg_i$) $\quad \dfrac{B, \neg B}{\neg A}$ (*)

Quy tắc khử $\neg$ (ký hiệu $\neg_e$) $\quad \dfrac{\neg\neg A}{A}$

Quy tắc nhập $\supset$ (ký hiệu $\supset_i$) $\quad \dfrac{B}{A \supset B}$ (*)

Quy tắc khử $\supset$ (ký hiệu $\supset_e$) $\quad \dfrac{A \supset B, A}{B}$

CÁC TIÊN ĐỀ VÀ QUY TẮC LOGIC VỊ TỪ

Các tiên đề

Với mọi công thức *A, B, C*, và biến *x* bất kỳ, các biểu thức *A1, A2, A3* sau đây là tiên đề của logic vị từ:

A1. $A \supset (B \supset A)$;

A2. $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$;

A3. $(\neg A \supset \neg B) \supset ((\neg A \supset B) \supset A)$;

A4. $\forall x\, A(x) \supset A(t)$, với *A(x)* là công thức, *t* là hạn từ, tự do đối với *x* trong công thức *A(x)*.

A5. $\forall x\, (A \supset B) \supset (A \supset \forall x\, B)$, nếu trong công thức *A* không có xuất hiện tự do của *x*.

Các quy tắc

1) $MP \quad \dfrac{A \supset B,\ A}{B}$ (Modus ponens)

2) $Gen \quad \dfrac{A}{\forall xA}$ (Quy tắc tổng quát hóa)

CÁC QUY TẮC CỦA HỆ SUY LUẬN TỰ NHIÊN LOGIC VỊ TỪ

Các quy tắc của hệ suy luận tự nhiên logic mệnh đề

Quy tắc nhập ∀ (ký hiệu $\forall_i$) $\quad \dfrac{A(x)}{\forall xA(x)}$ x không xuất hiện tự do trong các giả thiết và giả định trước A(x) trong chuỗi suy diễn.

Quy tắc khử ∀(ký hiệu $\forall_e$) $\quad \dfrac{\forall xA(x)}{A(t)}$ t tự do đối với x trong A(x)

Quy tắc nhập ∃ (ký hiệu $\exists_i$): $\quad \dfrac{A(t)}{\exists xA(x)}$ t tự do đối với x trong A(x)

Quy tắc khử ∃ (ký hiệu $\exists_e$) $\quad \dfrac{\exists xA(x)}{A(c)}$ c là hằng đối tượng mới

QUY TẮC HỆ SUY LUẬN TỰ NHIÊN LOGIC RELEVANT (HỆ K_2)

Đặc điểm phụ thuộc được hiểu là dãy các giả định (giả thiết), ghi trong cặp dấu ngoặc vuông; Γ, Δ là dãy bất kỳ của các giả định (công thức), và cũng có thể là dãy trống.

$\wedge_i$: $\dfrac{A[\Gamma], B[\Gamma]}{(A \wedge B)[\Gamma]}$

$\wedge_e$: $(\dfrac{(A \wedge B)[\Gamma]}{A[\Gamma]}$; $\dfrac{(A \wedge B)[\Gamma]}{B[\Gamma]}$;

$\vee_i$: $\dfrac{A[\Gamma]}{(A \vee B)[\Gamma]}$; $\dfrac{B[\Gamma]}{(A \vee B)[\Gamma]}$;

$\vee_e$: $\dfrac{C[\Gamma, A], C[\Gamma, B]}{C[\Gamma, A \vee B]}$

$\supset_i$: $\dfrac{B[\Gamma, A]}{(A \supset B)[\Gamma]}$

$\supset_e$: $\dfrac{(A \supset B)[\Gamma], A[\Delta]}{B[\Gamma, \Delta]}$

$\neg_e$: $\dfrac{\neg B[\Gamma, A], B[\Delta, C]}{\neg A[\Gamma, \Delta, C^*]}$

$\neg_e$: $\dfrac{\neg\neg A[\Gamma]}{A[\Gamma]}$

trong đó C^* là C nếu như C không trùng với A, và C^* là dãy trống trong trường hợp ngược lại.

$\dfrac{A[\Gamma]}{A[B, \Gamma]}$ Quy tắc đơn điệu, với B là giả định bất kỳ trong chuỗi suy diễn.

$\dfrac{A[\Gamma, B, B, \Delta]}{A[\Gamma, B, \Delta]}$ Rút gọn đặc điểm phụ thuộc

$\dfrac{A[\Gamma, B, C, \Delta]}{A[\Gamma, C, B, \Delta]}$ Hoán vị đặc điểm phụ thuộc

MỤC LỤC

IV. Hệ tiên đề của logic mệnh đề

1. ***Lý thuyết hình thức hóa (lý thuyết tiên đề hóa)***
2. ***Lý thuyết S (Hệ tiên đề S)***
3. ***Các hệ tiên đề khác của logic mệnh đề***

V. Hệ suy luận tự nhiên của logic mệnh đề

1. ***Các quy tắc***
2. ***Chuỗi suy diễn và phép chứng minh***
3. ***Tính không mâu thuẫn và đầy đủ của các hệ S và NS***

Chương II HỢP GIẢI TRONG LOGIC MỆNH ĐỀ

I. Công thức dạng tuyển

1. ***Định nghĩa***
2. ***Quy trình INDO***

II. Quy tắc hợp giải

III. Phương pháp hợp giải

IV. Cây hợp giải. Hợp giải tuyến tính

V. Giản lược tiền đề

1. ***Giản lược tiền đề là quy luật logic***
2. ***Giản lược tiền đề một chiều***
3. ***Giản lược tiền đề yếu***

Chương 3 LOGIC VỊ TỪ

I. Ngôn ngữ logic vị từ

1. ***Phân tích ngôn ngữ tự nhiên***
2. ***Hệ ký tự***
3. ***Hạn từ (term)***
4. ***Công thức (WFF – Well Formed Formula)***
5. ***Các ví dụ***
6. ***Biểu thị tư tưởng bằng ngôn ngữ logic vị từ***
7. ***Biến tự do và biến buộc***

II. Diễn giải (Interpretation). Mô hình (Model)

1. Diễn giải
2. Giá trị chân lý của công thức trong diễn giải
3. Mô hình (model). Quy luật logic

III. Diễn giải Herbrand

1. Miền herbrand
2. Định nghĩa diễn giải herbrand và ví dụ
3. Mô hình herbrand

IV. Hệ tiên đề của logic vị từ

1. Các tiên đề và quy tắc
2. Chuỗi suy diễn, phép chứng minh
3. Các tính chất cơ bản của hệ tiên đề logic vị từ

V. Hệ suy luận tự nhiên của logic vị từ

1. Các quy tắc
2. Chuỗi suy diễn, phép chứng minh
3. Một số ví dụ
4. Tính không mâu thuẫn và đầy đủ của hệ suy luận tự nhiên

Chương 4 HỢP GIẢI TRONG LOGIC VỊ TỪ

I. Công thức dạng tuyển

1. Định nghĩa
2. Quy trình INSEADOR

II. Phép thế

1. Định nghĩa
2. Áp dụng phép thế
3. Tính bất biến của phép thế
4. Phép thế hợp
5. *Quan hệ sắp xếp*

III. Đồng nhất thể

1. Định nghĩa
2. Đồng nhất thể lớn nhất

IV. Quy tắc hợp giải

V. Suy diễn hợp giải (chuỗi hợp giải) và phép chứng minh

VI. Áp dụng

1. Xác định tính mâu thuẫn của một tập công thức

2. Trả lời câu hỏi đúng, sai

3. Tìm kiếm câu trả lời

VII. Giản lược tiền đề

1. Giản lược tiền đề là quy luật logic

2. Giản lược tiền đề một chiều

3. Giản lược tiền đề yếu

Chương 5 RELEVANT LOGIC

I Dẫn nhập

II. Nghịch lý của suy diễn logic và phép toán kéo theo

1. Các nghịch lý

2. Các cố gắng giải quyết nghịch lý

III. Suy diễn logic – một quan hệ về nội dung, hay là về thông tin giữa các mệnh đề

1. Các đặc điểm của quan hệ suy diễn logic

2. Thông tin của mệnh đề. Nội dung logic là thông tin

3. Nguyên nhân của nghịch lý suy diễn logic (cổ điển)

IV Quan hệ suy diễn logic relevant

V Xác định về mặt hình thức nguyên nhân nghịch lý của khái niệm kéo theo logic cổ điển. Hệ logic tự nhiên relevant

1. Nguyên nhân nghịch lý của khái niệm kéo theo logic cổ điển về mặt hình thức

2. Hệ suy luận tự nhiên relevant

BÀI TẬP

DANH MỤC TÀI LIỆU THAM KHẢO

TÀI LIỆU THAM KHẢO

1. Nguyễn Đức Dân. *Logic và Tiếng Việt*, NXB Giáo dục, Hà Nội 1996
2. Nguyễn Đức Đồng, Nguyễn Văn Vĩnh. *Logic toán*, NXB Thanh Hóa
3. Phạm Đình Nghiệm. *Cơ sở phương pháp luận của lập trình logic*, Luận án Phó tiến sĩ triết học, Moskva 1991 (Tiếng Nga).
4. E. Mendencon. *Nhập môn logic toán*, NXB Khoa học, (Tiếng Nga).
5. M. Genesereth. *Computational logic* . http://logic.stanford.edu/~cs157/notes/
6. E.K. Voisvillo. *Khía cạnh triết lý-nhận thức luận của logic relevant*, Moskva 1988 (Tiếng Nga).
7. A.N. Kongomorov, A.G. Dragalin. *Nhập môn logic toán*, NXB Đại học Tổng hợp Moskva, 1982 (Tiếng Nga).
8. Huge. *Introduction to logic*
9. P. Tidman, H. Kanane. *Logic and Philosophy. A modern introduction*, Wadsworth Publishing Company.