Ta có nhận xét rằng, luật phân giải là một luật suy diễn tổng quát, nó bao gồm luật Modus Ponens, luật Modus Tollens, luật bắc cầu như các trường hợp riêng. (Bạn đọc dễ dàng chứng minh được điều đó).

TIÊN ĐỀ, ĐỊNH LÝ, CHỨNG MINH

Giả sử chúng ta có một tập nào đó các công thức. Các luật suy diễn cho phép ta từ các công thức đã có suy ra công thức mới bằng một dãy áp dụng các luật suy diễn. Các công thức đã cho được gọi là các ***tiên đề***. Các công thức được suy ra được gọi là các ***định lý***. Dãy các luật được áp dụng để dẫn tới định lý được gọi là một ***chứng minh*** của định lý. Nếu các luật suy diễn là tin cậy, thì các định lý là hệ quả logic của các tiên đề.

Ví dụ. Giả sử ta có các công thức sau:

$$Q \wedge S \Rightarrow G \wedge H \qquad (1)$$

$$P \Rightarrow Q \qquad (2)$$

$$R \Rightarrow S \qquad (3)$$

$$P \qquad (4)$$

$$R \qquad (5)$$

Giả sử ta cần chứng minh công thức G. Từ công thức (2) và (4), ta suy ra Q (Luật Modus Ponens). Lại áp dụng luật Modus Ponens, từ (3) và (5) ta suy ra S. Từ Q, S ta suy ra $Q \wedge S$ (luật đưa vào hội). Từ (1) và $Q \wedge S$ ta suy ra $G \wedge H$. Từ công thức $G \wedge H$ ta suy ra G.

Trong các hệ tri thức, chẳng hạn các hệ chuyên gia, hệ lập trình logic, ..., sử dụng các luật suy diễn người ta thiết kế nên các ***thủ tục suy diễn*** (còn được gọi là ***thủ tục chứng minh***) để từ các tri thức trong cơ sở tri thức ta suy ra các tri thức mới đáp ứng nhu cầu của người sử dụng.

Một ***hệ hình thức*** (formal system) bao gồm một tập các tiên đề và một tập các luật suy diễn nào đó (trong ngôn ngữ biểu diễn tri thức nào đó).

Một tập luật suy diễn được xem là ***đầy đủ***, nếu mọi hệ quả logic của một tập các tiên đề đều chứng minh được bằng cách chỉ sử dụng các luật của tập đó.

PHƯƠNG PHÁP CHỨNG MINH BÁC BỎ

Phương pháp chứng minh bác bỏ (refutation proof hoặc proof by contradiction) là một phương pháp thường xuyên được sử dụng trong các chứng minh toán học. Tư tưởng của phương pháp này là như sau: Để chứng minh P đúng, ta giả sử P sai (thêm $\rceil P$ vào các giả thiết) và dẫn tới một mâu thuẫn. Sau đây ta sẽ trình bầy cơ sở của phương pháp chứng minh này.

Giả sử chúng ta có một tập các công thức $\mathcal{G} = \{G_1, \ldots, G_m\}$ ta cần chứng minh công thức H là hệ quả logic của $\mathcal{G}$. Điều đó tương đương với chứng minh công thức $G_1 \wedge \ldots \wedge G_m \Rightarrow H$ là vững chắc. Thay cho chứng minh $G_1 \wedge \ldots \wedge G_m \Rightarrow H$ là vững chắc, ta chứng minh $G_1 \wedge \ldots \wedge G_m \wedge \rceil H$ là không thỏa mãn được. Tức là ta chứng minh tập $\mathcal{G}' = (G_1, \ldots, G_m, \rceil H)$ là không thỏa được. $\mathcal{G}'$ sẽ không thỏa được nếu từ $\mathcal{G}'$ ta suy ra hai mệnh đề đối lập nhau. Việc chứng minh công thức H là hệ quả logic của tập các tiên đề $\mathcal{G}$ bằng cách chứng minh tính không thỏa được của tập các tiên đề được thêm vào phủ định của công thức cần chứng minh, được gọi là chứng minh bác bỏ.

5.5. LUẬT PHÂN GIẢI, CHỨNG MINH BÁC BỎ BẰNG LUẬT PHÂN GIẢI

Để thuận tiện cho việc sử dụng luật phân giải, chúng ta sẽ cụ thể hoá luật phân giải trên các dạng câu đặc biệt quan trọng.

- ***Luật phân giải trên các câu tuyển***

$$\frac{\begin{array}{c} A_1 \vee \ldots \vee A_m \vee C, \\ \rceil C \vee B_1 \vee \ldots \vee B_n \end{array}}{A_1 \vee \ldots \vee A_m \vee B_1 \vee \ldots \vee B_n}$$

trong đó A_i, B_j và C là các literal.

- ***Luật phân giải trên các câu Horn:***

Giả sử P_i, R_j, Q và S là các literal. Khi đó ta có các luật sau:

$$\frac{\begin{array}{c} P_1 \wedge \ldots \wedge P_m \wedge S \Rightarrow Q, \\ R_1 \wedge \ldots \wedge R_n \Rightarrow S \end{array}}{P_1 \wedge \ldots \wedge P_m \wedge R_1 \wedge \ldots \wedge R_n \Rightarrow Q}$$

Một trường hợp riêng hay được sử dụng của luật trên là:

$$\frac{\begin{array}{r} P_1 \wedge \ldots \wedge P_m \wedge S \Rightarrow Q, \\ S \end{array}}{P_1 \wedge \ldots \wedge P_m \Rightarrow Q}$$

Khi ta có thể áp dụng luật phân giải cho hai câu, thì hai câu này được gọi là ***hai câu phân giải được*** và kết quả nhận được khi áp dụng luật phân giải cho hai câu đó được gọi là ***giải thức*** của chúng. Giải thức của hai câu A và B được kí hiệu là res(A, B). Chẳng hạn, hai câu tuyển phân giải được nếu một câu chứa một literal đối lập với một literal trong câu kia. Giải thức của hai literal đối lập nhau (P và $\rceil P$) là câu rỗng, chúng ta sẽ ký hiệu câu rỗng là . câu rỗng không thỏa được.

Giả sử $\mathcal{G}$ là một tập các câu tuyển (bằng cách chuẩn hoá ta có thể đưa một tập các công thức về một tập các câu tuyển). Ta sẽ ký hiệu R($\mathcal{G}$) là tập câu bao gồm các câu thuộc $\mathcal{G}$ và tất cả các câu nhận được từ $\mathcal{G}$ bằng một dãy áp dụng luật phân giải.

Luật phân giải là luật đầy đủ để chứng minh một tập câu là không thỏa được. Điều này được suy từ định lý sau.

ĐỊNH LÝ PHÂN GIẢI

Một tập câu tuyển là không thỏa được nếu và chỉ nếu câu rỗng $\in R(\mathcal{G})$.

Định lý phân giải có nghĩa rằng, nếu từ các câu thuộc $\mathcal{G}$, bằng cách áp dụng luật phân giải ta dẫn tới câu rỗng thì $\mathcal{G}$ là không thỏa được, còn nếu không thể sinh ra câu rỗng bằng luật phân giải thì $\mathcal{G}$ thỏa được. Lưu ý rằng, việc dẫn tới câu rỗng có nghĩa là ta đã dẫn tới hai literal đối lập nhau P và $\rceil P$ (tức là dẫn tới mâu thuẫn).

Từ định lý phân giải, ta đưa ra thủ tục sau đây để xác định một tập câu tuyển $\mathcal{G}$ là thỏa được hay không. Thủ tục này được gọi là thủ tục phân giải.

Procedure ***Resolution***;

Input: tập $\mathcal{G}$ các câu tuyển ;

begin

1. **Repeat**

   1.1. Chọn hai câu A và B thuộc $\mathcal{G}$;

   1.2. **if** A và B phân giải được **then** tính Res(A,B);

   1.3. **if** Res(A,B) là câu mới **then** thêm Res(A,B) vào $\mathcal{G}$;

   until nhận được câu rỗng hoặc không có câu mới xuất hiện;

2. **if** nhận được câu rỗng **then** thông báo $\mathcal{G}$ không thỏa được

   else thông báo $\mathcal{G}$ thỏa được;

end;

Chúng ta có nhận xét rằng, nếu $\mathcal{G}$ là tập hữu hạn các câu thì các literal có mặt trong các câu của $\mathcal{G}$ là hữu hạn. Do đó số các câu tuyển thành lập được từ các literal đó là hữu hạn. Vì vậy chỉ có một số hữu hạn câu được sinh ra bằng luật phân giải. Thủ tục phân giải sẽ dừng lại sau một số hữu hạn bước.

Chỉ sử dụng luật phân giải ta không thể suy ra mọi công thức là hệ quả logic của một tập công thức đã cho. Tuy nhiên, sử dụng luật phân giải ta có thể chứng minh được một công thức bất kì có là hệ quả của một tập công thức đã cho hay không bằng phương pháp chứng minh bác bỏ. Vì vậy luật phân giải được xem là ***luật đầy đủ cho bác bỏ.***

Sau đây là thủ tục chứng minh bác bỏ bằng luật phân giải

Procedure ***Refutation_Proof;***

input: Tập $\mathcal{G}$ các công thức;

Công thức cần chứng minh H;

begin

1. Thêm $\rceil$H vào $\mathcal{G}$;
2. Chuyển các công thức trong $\mathcal{G}$ về dạng chuẩn hội;
3. Từ các dạng chuẩn hội ở bước hai, thành lập tập các câu tuyển $\mathcal{G}'$;
4. Áp dụng thủ tục phân giải cho tập câu $\mathcal{G}'$;

5. **if** $\mathcal{G}'$ không thỏa được **then** thông báo H là hệ quả logic
   else thông báo H không là hệ quả logic của $\mathcal{G}$;

end;

Ví dụ. Giả giử $\mathcal{G}$ là tập hợp các câu tuyển sau

$$\rceil A \vee \rceil B \vee P \qquad (1)$$

$$\rceil C \vee \rceil D \vee P \qquad (2)$$

$$\rceil E \vee C \qquad (3)$$

$$A \qquad (4)$$

$$E \qquad (5)$$

$$D \qquad (6)$$

Giả sử ta cần chứng minh P. Thêm vào $\mathcal{G}$ câu sau:

$$\rceil P \qquad (7)$$

Áp dụng luật phân giải cho câu (2) và (7) ta được câu:

$$\rceil C \vee \rceil D \qquad (8)$$

Từ câu (6) và (8) ta nhận được câu:

$$\rceil C \qquad (9)$$

Từ câu (3) và (9) ta nhận được câu:

$$\rceil E \qquad (10)$$

Tới đây đã xuất hiện mâu thuẫn, vì câu (5) và (10) đối lập nhau. Từ câu (5) và (10) ta nhận được câu rỗng .

Vậy P là hệ quả logic của các câu (1) -- (6).

CHƯƠNG 6
LOGIC VỊ TỪ CẤP MỘT

Logic mệnh đề cho phép ta biểu diễn các sự kiện, mỗi kí hiệu trong logic mệnh đề được minh họa như là một sự kiện trong thế giới hiện thực. sử dụng các kết nối logic ta có thể tạo ra các câu phức hợp biểu diễn các sự kiện mang ý nghĩa phức tạp hơn. Như vậy khả năng biểu diễn của logic mệnh đề chỉ giới hạn trong phạm vi thế giới các sự kiện.

Thế giới hiện thực bao gồm các ***đối tượng,*** mỗi đối tượng có những ***tính chất*** riêng để phân biệt nó với các đối tượng khác. Các đối tượng lại có ***quan hệ*** với nhau. Các mối quan hệ rất đa dạng và phong phú. Chúng ta có thể liệt kê ra rất nhiều ví dụ về đối tượng, tính chất, quan hệ.

- *Đối tượng:* một cái bàn, một cái nhà, một cái cây, một con người, một con số....
- *Tính chất:* Cái bàn có thể có tính chất: có bốn chân, làm bằng gỗ, không có ngăn kéo. Con số có thể có tính chất là số nguyên, số hữu tỉ, là số chính phương...
- *Quan hệ:* cha con, anh em, bè bạn (giữa con người); lớn hơn, nhỏ hơn, bằng nhau (giữa các con số); bên trong, bên ngoài, nằm trên, nằm dưới (giữa các đồ vật)...
- *Hàm:* Một trường hợp riêng của quan hệ là quan hệ hàm. Chẳng hạn, vì mỗi người có một mẹ, do đó ta có quan hệ hàm ứng mỗi người với mẹ của nó.

Logic vị từ cấp một là mở rộng của logic mệnh đề. Nó cho phép ta mô tả thế giới với các đối tượng, các thuộc tính của đối tượng và các mối quan hệ giữa các đối tượng. Nó sử dụng các biến (biến đối tượng) để chỉ các đối tượng trong một miền đối tượng nào đó. Để mô tả các thuộc tính của đối tượng, các quan hệ giữa các đối tượng, trong logic vị từ, người ta đưa vào các ***vị từ*** (predicate). Ngoài các kết nối logic như trong logic mệnh đề, logic vị từ cấp một còn sử dụng các ***lượng tử.*** Chẳng hạn, lượng tử $\forall$ (với mọi) cho phép ta tạo ra các câu nói tới mọi đối tượng trong một miền đối tượng nào đó.

Chương này dành cho nghiên cứu logic vị từ cấp một với tư cách là một ngôn ngữ biểu diễn tri thức. Logic vị từ cấp một đóng vai trò cực kì quan trọng trong biểu diễn tri thức, vì khả năng biểu diễn của nó (nó cho phép ta biểu diễn tri thức về thế giới với các đối tượng, các thuộc tính của đối tượng và các quan hệ của đối tượng), và hơn nữa, nó là cơ sở cho nhiều ngôn ngữ logic khác.

6.1. CÚ PHÁP VÀ NGỮ NGHĨA CỦA LÔGIC VỊ TỪ CẤP MỘT

6.1.1. Cú pháp

CÁC KÝ HIỆU

Logic vị từ cấp một sử dụng các loại ký hiệu sau đây.

- Các ký hiệu hằng: a, b, c, An, Ba, John, ...
- Các ký hiệu biến: x, y, z, u, v, w, ...
- Các ký hiệu vị từ: P, Q, R, S, Like, Havecolor, Prime, ...

Mỗi vị từ là vị từ của n biến ($n \geq 0$). Chẳng hạn Like là vị từ của hai biến. Prime là vị từ một biến. Các ký hiệu vị từ không biến là các ký hiệu mệnh đề.

- Các ký hiệu hàm: f, g, cos, sin, mother, husband, distance, ...

  Mỗi hàm là hàm của n biến ($n \geq 1$). Chẳng hạn, cos, sin là hàm một biến. distance là hàm của ba biến.
- Các ký hiệu kết nối logic: $\wedge$ (hội), $\vee$ (tuyển), $\rceil$ (phủ định), $\Rightarrow$ (kéo theo), $\Leftrightarrow$ (kéo theo nhau).
- Các ký hiệu lượng tử: $\forall$ (với mọi), $\exists$ (tồn tại).
- Các ký hiệu ngăn cách: dấu phẩy, dấu mở ngoặc và dấu đóng ngoặc.

CÁC HẠNG THỨC

Các hạng thức (term) là các biểu thức mô tả các đối tượng. Các hạng thức được xác định đệ quy như sau.

- Các ký hiệu hằng và các ký hiệu biến là hạng thức.
- Nếu $t_1, t_2, \ldots, t_n$ là n hạng thức và f là một ký hiệu hàm n biến thì $f(t_1, t_2, \ldots, t_n)$ là hạng thức.

Một hạng thức không chứa biến được gọi là một ***hạng thức cụ thể*** (ground term). Chẳng hạn, *An* là ký hiệu hằng, mother là ký hiệu hàm một biến, thì mother(*An*) là một hạng thức cụ thể.

CÁC CÔNG THỨC PHÂN TỬ

Chúng ta sẽ biểu diễn các tính chất của đối tượng, hoặc các quan hệ giữa các đối tượng bởi các ***công thức phân tử (câu đơn)***.

Các công thức phân tử (câu đơn) được xác định đệ quy như sau.

- Các ký hiệu vị từ không biến (các ký hiệu mệnh đề) là công thức phân tử.
- Nếu $t_1, t_2, \ldots, t_n$ là n hạng thức và P là vị từ của n biến thì $P(t_1, t_2, \ldots, t_n)$ là công thức phân tử.

Chẳng hạn, Hoa là một ký hiệu hằng, Love là một vị từ của hai biến, husband là hàm của một biến, thì Love(Hoa, husband(Hoa)) là một công thức phân tử.

CÁC CÔNG THỨC

Từ công thức phân tử, sử dụng các kết nối logic và các lượng tử, ta xây dựng nên các công thức (các câu).

Các công thức được xác định đệ quy như sau:

- Các công thức phân tử là công thức.
- Nếu G và H là các công thức, thì các biểu thức $(G \wedge H)$, $(G \vee H)$, $(\rceil G)$, $(G \Rightarrow H)$, $(G \Leftrightarrow H)$ là công thức.
- Nếu G là một công thức và x là biến thì các biểu thức $(\forall\ x\ G)$, $(\exists\ x\ G)$ là công thức.

Các công thức không phải là công thức phân tử sẽ được gọi là các câu phức hợp. Các công thức không chứa biến sẽ được gọi là ***công thức cụ thể.*** Khi viết các công thức ta sẽ bỏ đi các dấu ngoặc không cần thiết, chẳng hạn các dấu ngoặc ngoài cùng.

Lượng tử phổ dụng cho phép mô tả tính chất của cả một lớp các đối tượng, chứ không phải của một đối tượng, mà không cần phải liệt kê ra tất cả các đối tượng trong lớp. Chẳng hạn sử dụng vị từ Elephant(x) (đối tượng x là con voi) và vị từ Color(x, Gray) (đối tượng x có mầu xám) thì câu "tất cả các con voi đều có mầu xám" có thể biểu diễn bởi công thức $\forall x$ (Elephant(x) $\Rightarrow$ Color(x, Gray)).

Lượng tử tồn tại cho phép ta tạo ra các câu nói đến một đối tượng nào đó trong một lớp đối tượng mà nó có một tính chất hoặc thỏa mãn một quan hệ nào đó. Chẳng hạn bằng cách sử dụng các câu đơn Student(x) (x là sinh viên) và Inside(x, P301), (x ở trong phòng 301), ta có thể biểu diễn câu "Có một sinh viên ở phòng 301" bởi biểu thức $\exists x$ (Student(x) $\wedge$ Inside(x,P301)).

Một công thức là công thức phân tử hoặc phủ định của công thức phân tử được gọi là ***literal***. Chẳng hạn, Play(x, Football), $\rceil$ Like(Lan, Rose) là các literal. Một công thức là tuyển của các literal sẽ được gọi là ***câu tuyển.*** Chẳng hạn, Male(x) $\vee$ $\rceil$ Like(x, Foodball) là câu tuyển.

Trong công thức $\forall x\ G$, hoặc $\exists x\ G$ trong đó G là một công thức nào đó, thì mỗi xuất hiện của biến x trong công thức G được gọi là ***xuất hiện buộc***. Một công thức mà tất cả các biến đều là xuất hiện buộc thì được gọi là ***công thức đóng.***

Ví dụ. Công thức $\forall x\ P(x, f(a, x)) \wedge \exists y\ Q(y)$ là công thức đóng, còn công thức $\forall x\ P(x, f(y, x))$ không phải là công thức đóng, vì sự xuất hiện của biến y trong công thức này không chịu ràng buộc bởi một lượng tử nào cả (Sự xuất hiện của y gọi là ***sự xuất hiện tự do***).

Sau này chúng ta chỉ quan tâm tới các công thức đóng.

6.1.2. Ngữ nghĩa

Cũng như trong logic mệnh đề, nói đến ngữ nghĩa là chúng ta nói đến ý nghĩa của các công thức trong một thế giới hiện thực nào đó mà chúng ta sẽ gọi là ***một minh họa.***

Để xác định một minh họa, trước hết ta cần xác định một miền đối tượng (nó bao gồm tất cả các đối tượng trong thế giới hiện thực mà ta quan tâm).

Trong một minh hoạ, các ký hiệu hằng sẽ được gắn với các đối tượng cụ thể trong miền đối tượng, các ký hiệu hàm sẽ được gắn với một hàm cụ thể nào đó. Khi đó, mỗi hạng thức cụ thể sẽ chỉ định một đối tượng cụ thể trong miền đối tượng. Chẳng hạn, nếu An là một ký hiệu hằng, Father là một ký hiệu hàm, nếu trong minh họa An ứng với một người cụ thể nào đó, còn Father(x) gắn với hàm: ứng với mỗi x là cha của nó, thì hạng thức Father(An) sẽ chỉ người cha của An.

NGỮ NGHĨA CỦA CÁC CÂU ĐƠN

Trong một minh hoạ, các ký hiệu vị từ sẽ được gắn với một thuộc tính, hoặc một quan hệ cụ thể nào đó. Khi đó mỗi công thức phân tử (không chứa biến) sẽ chỉ định một sự kiện cụ thể. Đương nhiên sự kiện này có thể là đúng (True) hoặc sai (False). Chẳng hạn, nếu trong minh hoạ, ký hiệu hằng Lan ứng với một cô gái cụ thể nào đó, còn Student(x) ứng với thuộc tính "x là sinh viên" thì câu Student (Lan) có giá trị chân lý là True hoặc False tùy thuộc trong thực tế Lan có phải là sinh viên hay không.

NGỮ NGHĨA CỦA CÁC CÂU PHỨC HỢP

Khi đã xác định được ngữ nghĩa của các câu đơn, ta có thể xác định được ngữ nghĩa của các câu phức hợp (được tạo thành từ các câu đơn bằng cách liên kết các câu đơn bởi các kết nối logic) như trong logic mệnh đề.

Ví dụ. Câu Student(Lan) $\wedge$ Student(An) nhận giá trị True nếu cả hai câu Student(Lan) và Student(An) đều có giá trị True, tức là cả Lan và An đều là sinh viện.

Câu Like(Lan, Rose) $\vee$ Like(An, Tulip) là đúng nếu câu Like(Lan, Rose) là đúng hoặc câu Like(An, Tulip) là đúng.

NGỮ NGHĨA CỦA CÁC CÂU CHỨA CÁC LƯỢNG TỬ

Ngữ nghĩa của các câu $\forall x\ G$, trong đó G là một công thức nào đó, được xác định như là ngữ nghĩa của công thức là hội của tất cả các công