Chương 4

BIỂU DIỄN BÀI TOÁN BẰNG LOGIC VÀ CÁC PHƯƠNG PHÁP CHỨNG MINH

Như ta đã biết, không thể có phương pháp giải quyết vấn đề tổng quát cho mọi bài toán. Có thể phương pháp này phù hợp cho bài toán này, nhưng lại không phù hợp cho lớp bài toán khác. Điều này có nghĩa là khi nói tới một bài toán, ta phải chú ý đến ***phương pháp biểu diễn nó*** cùng với các ***phương pháp tìm kiếm trong không gian bài toán nhận được***.

1. Biểu diễn bài toán nhờ không gian trạng thái (có các chiến lược tìm kiếm trên đồ thị biểu diễn vấn đề)
2. Quy về các bài toán con
3. Biểu diễn vấn đề nhờ logic hình thức (có các phương pháp suy diễn logic)

....

và trong phần này sẽ trình bày phương pháp biểu diễn vấn đề nhờ logic hình thức và các phương pháp giải quyết vấn đề trên cách biểu diễn này.

Logic hình thức thường dùng để thu gọn quá trình tìm kiếm lời giải. Trước khi giải quyết vấn đề, nhờ phân tích logic, có thể chứng tỏ rằng một bài toán nào đó có thể giải được hay không?.

Ngoài ra, các kết luận logic rất cần ngay cả trong cách tiếp cận dựa trên không gian trạng thái và quy bài toán về bài toán con. Chẳng hạn, trong các phương pháp dựa trên không gian trạng thái, các kết luận logic dùng để kiểm tra một trạng thái nào đó có phải là trạng thái đích hay không?,....

Ngoài ra, logic hình thức có thể được sử dụng để giải quyết những bài toán chứng minh logic, chẳng hạn như chứng minh một khẳng định nào đó là đúng khi biết những tiền đề ban đầu và các luật suy diễn. Đây là một dạng quen thuộc nhất và được các chuyên gia TTNT quan tâm ngay từ đầu.

Ví dụ

Ta có thể dùng các biểu thức logic để mô tả mối quan hệ của các thành phần trong 1 tam giác như sau:

1) $a \wedge b \wedge c \Rightarrow p$

2) $b \wedge p \wedge c \Rightarrow a$

3) $a \wedge p \wedge c \Rightarrow b$

4) $a \wedge b \wedge p \Rightarrow c$

5) $S \wedge c \Rightarrow hc$

6) $a \wedge b \wedge C \Rightarrow c$

7) $a \wedge b \wedge C \Rightarrow S$

8) $a \wedge b \wedge c \wedge p \Rightarrow S$

9) $S \wedge hc \Rightarrow c$

(***Trong đó:*** *a, b, c là ký hiệu các cạnh, A, B, C là ký hiệu các góc tương ứng, p là ký hiệu nữa chu vi, và hc là đường cao xuất phát từ đỉnh C của tam giác*)

Giả sử ta biết các cạnh a, b và một góc C. Ta có thể có kết luận về đường cao hc không?

1. BI ỂU DI ỄN VẤN ĐỀ NHỜ LOGIC HÌNH THỨC

1.1. Logic mệnh đề

Đây là kiểu biểu diễn tri thức đơn giản nhất và gần gũi nhất đối với chúng ta.

a) ***Mệnh đề*** là một khẳng định, một phát biểu mà giá trị của nó chỉ có thể hoặc là đúng hoặc là sai.

Ví dụ

phát biểu "1+1=2" (có giá trị đúng)

phát biểu "Trời mưa"

(Giá trị của mệnh đề không chỉ phụ thuộc vào bản thân mệnh đề đó. Có những mệnh đề mà giá trị của nó luôn đúng hoặc sai bất chấp thời gian nhưng cũng có những mệnh đề mà giá trị của nó lại phụ thuộc vào thời gian, không gian và nhiều yếu tố khác quan khác. Chẳng hạn như mệnh đề : "Con người không thể nhảy cao hơn 5m với chân trần" là đúng khi ở trái đất , còn ở những hành tinh có lực hấp dẫn yếu thì có thể sai.)

b) Biểu thức logic

- Ta ký hiệu mệnh đề bằng những chữ cái la tinh như **a, b, c, ...** và các ký hiệu này được *gọi là biến mệnh đề*

- *Biểu thức logic* được định nghĩa đệ quy như sau:

- Các hằng logic (True, False) và các biến mệnh đề là các biểu thức logic
- Các biểu thức logic kết hợp với các toán tử logic (phép tuyển ($\vee$), phép hội ($\wedge$), phủ định ($\neg$, ~, $\bar{\ }$), phép kéo theo ($\Rightarrow$, $\rightarrow$), phép tương đương ($\Leftrightarrow$, $\equiv$)) là các biểu thức logic.

Tức là nếu E và F là các biểu thức logic thì $E \wedge F$, $E \vee F$, $E \rightarrow F$, $E \equiv F$ cũng là các biểu thức logic

Thứ tự ưu tiên của các phép toán logic: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\equiv$

Ví dụ Một số biểu thức logic:

1)True

2)$\neg$ p

3)p $\wedge$ (p $\vee$ r)

.....

- *Biểu thức logic dạng chuẩn:* là biểu thức được xây dựng từ các biến mệnh đề và các phép toán $\neg$, $\wedge$, $\vee$.

Ví dụ $p \wedge (\neg p \vee r)$

(Chúng ta đã từng sử dụng logic mệnh đề trong chương trình rất nhiều lần (như trong cấu trúc lệnh IF ... THEN ... ELSE) để biểu diễn các tri thức "cứng" trong máy tính !)

c) Bảng chân trị (bảng chân lý) Dùng để dánh giá giá trị của biểu thức logic.

p	**q**	$\neg p$	$p \vee q$	$p \wedge q$	$\neg p \vee q$	$p \to q$	$p \equiv q$
T	T	F	T	T	T	T	T
T	F	F	T	F	F	F	F
F	T	T	T	F	T	T	F
F	F	T	F	F	T	T	T

Nhận xét

- Mọi biểu thức logic đều có thể chuyển về các biểu thức logic dạng chuẩn nhờ vào:

$$p \to q \equiv \neg p \vee q$$

- Nếu có n biến mệnh đề trong biểu thức logic thì bảng chân trị sẽ có 2^n trường hợp khác nhau đối với các biến mệnh đề.

d) Đồng nhất đúng

Một đồng nhất đúng là một biểu thức logic luôn luôn có giá trị True với bất kỳ giá trị nào của các biến mệnh đề trong biểu thức logic đó.

Ví dụ (Có thể kiểm tra bằng cách dùng bảng chân trị)

1) $p \vee \neg p$

2) $0 \to p$

3) $(p \vee q) \wedge (\neg p \vee r) \to q \vee r$

Ta thấy rằng biểu thức có dạng VT→VP luôn có giá trị True (T) với mọi giá trị của a, b; chỉ có một trường hợp để a →b có giá trị False (F) là a: True và b: False. Như vậy, để chứng minh biểu thức 3) là một đồng nhất đúng, ta chỉ cần chứng minh nếu b: F thì a: F, không có trường hợp a: T và b: F.

Thật vậy, giả sử VP: F nghĩa là q: F và r: F. Xét 2 trường hợp của p:

- Nếu p: T thì VT: F
- Nếu p: F thì VT: F

Do đó biểu thức 3) là một đồng nhất đúng

Bài tập. Biểu thức nào trong số các biểu thức sau đây là đồng nhất đúng?

1) $p \wedge q \wedge r \rightarrow p \vee q$

2) $(p \rightarrow q) \rightarrow p$

3) $((p \rightarrow q \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$

1.2. Một số luật đại số

Sau đây là một số đồng nhất đúng thường gặp

a) Luật phản xạ (cho phép tương đương): $p \equiv p$

b) Luật giao hoán

- phép tương đương: $p \equiv p$
- phép hội: $p \wedge q \equiv q \wedge p$
- phép tuyển: $p \vee q \equiv q \vee p$

c) Luật bắc cầu:

- phép kéo theo: $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$
- phép tương đương: $(p \equiv q) \wedge (q \equiv r) \rightarrow (p \equiv r)$

d) Luật kết hợp:

- phép hội: $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$
- phép tuyển: $p \vee (q \vee r) \equiv (p \vee q) \vee r$

e) Luật phân phối:

- phép $\wedge$ trên phép $\vee$: $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
- phép $\vee$ trên phép $\wedge$: $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$

f) Phần tử trung hoà:

- 0 (False) là phần tử trung hoà cho phép $\vee$: $p \vee 0 \equiv p$
- 1 (true) là phần tử trung hoà cho phép $\wedge$: $p \wedge 1 \equiv p$

g) Triệt tử

- 0 (False) là triệt tử cho phép $\wedge$: $p \wedge 0 \equiv 0$
- 1 (true) là triệt tử cho phép $\vee$: $p \vee 1 \equiv 1$

h) Tính luỹ đẳng

- của phép $\wedge$: $p \wedge p \equiv p$
- của phép $\vee$: $p \vee p \equiv p$

i) Luật Demorgan

$\neg(p \vee q) \equiv \neg p \wedge \neg q$

$\neg(p \wedge q) \equiv \neg p \vee \neg q$

j) Một số luật khác cho phép kéo theo

- $(p \to q) \wedge (q \to p) \equiv (p \equiv q)$
- $(p \equiv q) \to (p \to q)$
- $p \to q \equiv \neg p \vee q$

k) $\neg(\neg p) \equiv p$

1.3. Logic vị từ

Biểu diễn tri thức bằng mệnh đề gặp phải một trở ngại cơ bản là ta không thể can thiệp vào cấu trúc của một mệnh đề. Hay nói một cách khác là mệnh đề *không có cấu trúc* . Điều này làm hạn chế rất nhiều thao tác suy luận .

Do đó, người ta đã đưa vào ***khái niệm vị từ và lượng từ*** (**$\forall$:với mọi, $\exists$: tồn tại**) để tăng cường tính cấu trúc của một mệnh đề.

Trong logic vị từ, một mệnh đề *được cấu tạo bởi* hai thành phần là các ***đối tượng tri thức*** và ***mối liên hệ giữa chúng*** *(gọi là vị từ)*. Các mệnh đề sẽ được biểu diễn dưới dạng:

Vị từ (<đối tượng 1>, <đối tượng 2>, …, <đối tượng n>)

Ví dụ

Để biểu diễn vị của các trái cây, các mệnh đề sẽ được viết lại thành :

Cam có vị Ngọt $\Rightarrow$ Vị (Cam, Ngọt)

Cam có màu Xanh $\Rightarrow$ Màu (Cam, Xanh)

...

Kiểu biểu diễn này có hình thức tương tự như hàm trong các ngôn ngữ lập trình, các đối tượng tri thức chính là các tham số của hàm, giá trị mệnh đề chính là kết quả của hàm (thuộc kiểu BOOLEAN).

Với vị từ, ta có thể biểu diễn các tri thức dưới dạng các mệnh đề tổng quát, là những mệnh đề mà giá trị của nó được xác định thông qua các đối tượng tri thức cấu tạo nên nó.

Ví dụ

1) Chẳng hạn tri thức : *"A là bố của B nếu B là anh hoặc em của một người con của A"* có thể được biểu diễn dưới dạng vị từ như sau :

Bố (A, B) = Tồn tại Z sao cho : Bố (A, Z) và (Anh(Z, B) hoặc Anh(B,Z))

Trong trường hợp này, mệnh đề Bố(A,B) là một mệnh đề tổng quát

Như vậy nếu ta có các mệnh đề cơ sở là :

a) Bố ("An", "Bình") có giá trị đúng (Anh là bố của Bình)

b) Anh("Tú", "Bình") có giá trị đúng (Tú là anh của Bình)

thì mệnh đề **c)** Bố ("An", "Tú") sẽ có giá trị là đúng. (An là bố của Tú).

Rõ ràng là nếu chỉ sử dụng logic mệnh đề thông thường thì ta sẽ không thể tìm được một mối liên hệ nào giữa c và a,b bằng các phép nối mệnh đề $\wedge$, $\vee$, $\neg$. Từ đó, ta cũng không thể tính ra được giá trị của mệnh đề **c.** Sở dĩ như vậy vì ta không thể thể hiện tường minh tri thức *"(A là bố của B) nếu có Z sao cho (A là bố của Z) và (Z anh hoặc em C)"* dưới dạng các mệnh đề thông thường. Chính đặc trưng của vị từ đã cho phép chúng ta thể hiện được các tri thức dạng tổng quát như trên.

2) Câu cách ngôn *"Không có vật gì là lớn nhất và không có vật gì là bé nhất!"* có thể được biểu diễn dưới dạng vị từ như sau :

LớnHơn(x,y) = x>y

NhỏHơn(x,y) = x<yy

∀x,× ∃y : LớnHơn(y,x) và ∀x, ∃y : NhỏHơn(y,x)

3) Câu châm ngôn *"Gần mực thì đen, gần đèn thì sáng"* được hiểu là "chơi với bạn xấu nào thì ta cũng sẽ thành người xấu" có thể được biểu diễn bằng vị từ như sau :

NgườiXấu (x) = ∀y : Bạn(x,y) và NgườiXấu(y)

Sử dụng vị từ làm toán hạng nguyên tử thay vì các biến mệnh đề đã đưa ra một ngôn ngữ mạnh mẽ hơn so với các biểu thức chỉ chứa mệnh đề. Thực sự, logic vị từ đủ khả năng diễn tả để tạo cơ sở cho một số ngôn ngữ lập trình rất có ích như Prolog (Programing Logic) và ngôn ngữ SQL. Logic vị từ cũng được sử dụng trong các hệ thống suy luận hoặc các hệ chuyên gia chẳng hạn các chương trình chẩn đoán tự động y khoa, các chương trình chứng minh định lý tự động

1.3.1. *Cú pháp và ngữ nghĩa của logic vị từ*

a. Cú pháp

- ***Các ký hiệu***

- **Hằng**: được biểu diễn bằng chuỗi ký tự bắt đầu bằng chữ cái thường hoặc các chữ số hoặc chuỗi ký tự đặt trong bao nháy. Ví dụ: a,b, c, “An”, “Ba”,...

- **Biến**: tên biến luôn bắt đầu bằng chữ cái viết hoa. Ví dụ: X, Y, Z, U, V,...

- **Vị từ**: được biểu diễn bằng chuỗi ký tự bắt đầu bằng chữ cái thường. Ví dụ: p, q, r, s, like,...

Mỗi vị từ là vị từ của n biến (n≥ 0). Các ký hiệu vị từ không có biến là các ký hiệu mệnh đề

Ví dụ: like(X,Y) là vị từ của hai biến

u(X) là vị từ một biến

r là vị từ không biến

- **Hàm:** f, g, cos, sin, mother,...

Mỗi hàm là hàm của n biến (n≥ 1). Ví dụ: cos, sin là hàm một biến

- Lượng từ: ∀(với mọi), ∃ (tồn tại).

Ví dụ: ∀X, p(X) nghĩa là với mọi giá trị của biến X đều làm cho biểu thức p đúng.

∃X, p(X) nghĩa là có ít nhất một giá trị của biến X để làm cho biểu thức p đúng.

- Các ký hiệu kết nối logic: ∧(hội), ∨(tuyển), ¬(phủ định), ⇒(kéo theo), ⇔(kéo theo nhau).

- 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 $t_1, t_2, t_3,...,t_n$ là n hạng thức và f là một ký hiệu hàm n biến thì $f(t_1, t_2, t_3,...,t_n)$ là hạng thức. Một hạng thức không chứa biến được gọi là một hạng thức cụ thể (ground term).

Ví dụ: An là một 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 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 $t_1, t_2, t_3,...,t_n$ là n hạng thức và p là vị từ của n biến thì $p(t_1, t_2, t_3,...,t_n)$ là công thức phân tử.

Ví dụ: Hoa là một ký hiệu hằng, love là một vị từ hai biến, husband là hàm của một biến thế thì love("Hoa", husband("Hoa")) là một công thức phân tử.