

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP VINH

KHOA CÔNG NGHỆ

BỘ MÔN CÔNG NGHỆ THÔNG TIN

(LƯU HÀNH NỘI BỘ)

TRÍ TUỆ NHÂN TẠO

BIỂU DIỄN & TÌM KIẾM

TS. NGUYỄN NGỌC THUẦN

VINH, 2018

LỜI TỰA

Để có thêm lựa chọn “Tài liệu học tập” cho sinh viên trong quá trình học tập, nghiên cứu; chủ động trong việc giảng dạy cho giáo viên. Sau nhiều năm giảng dạy, chúng tôi biên tập lại một số vấn đề cơ bản của Môn học Trí tuệ nhân tạo trên tinh thần rút gọn để phù hợp với đối tượng - sinh viên ngành CNTT, trường Đại học Công nghiệp Vinh – trường Đại học định hướng thực hành và vừa đảm bảo những phần cơ bản nhất của môn học.

Với phương châm vừa giới thiệu lý thuyết vừa tăng cường các kỹ năng thực hành, tập bài giảng tập trung giới thiệu các vấn đề liên quan đến biểu diễn tri thức và các kỹ thuật tìm kiếm. Đây là giáo trình môn học Trí tuệ nhân tạo dành cho giáo viên, sinh viên ngành CNTT trong công tác giảng dạy, nghiên cứu và học tập tại trường Đại học Công nghiệp Vinh.

Giáo trình lựa chọn các vấn đề cơ bản từ các tài liệu chuyên ngành từ truyền thống đến hiện đại và được tổ chức thành 04 chương; Chương 1, giới thiệu các khái niệm cơ bản; Nêu vai trò, nhiệm vụ của môn học trong chương trình đào tạo Đại học và trong đời sống xã hội; Các xu thế nghiên cứu trong tương lai gần trong ứng dụng thực tiễn của TTNT. Chương 2, tập trung giới thiệu hai trong bốn cách biểu diễn tri thức, đó là biểu diễn tri thức nhờ Logic và biểu diễn tri thức sử dụng luật dẫn xuất; Còn biểu diễn tri thức sử dụng mạng ngữ nghĩa hoặc khung (Frame) chỉ giới thiệu định nghĩa và cấu trúc của phương pháp. Chương 3, mô tả không gian trạng thái với các bộ phận cấu thành nhằm phục vụ cho các giải pháp tìm kiếm được trình bày trong phần tiếp theo. Chương 4, ngoài việc đưa ra các giải pháp tìm kiếm truyền thống: Tìm kiếm theo chiều rộng, tìm kiếm theo chiều sâu, giáo trình tập trung các giải pháp nâng cao: tìm kiếm với đường đi cực tiểu, tìm kiếm với giá thành tối ưu – Phương pháp tìm kiếm tốt nhất đầu tiên (BFS); tìm kiếm leo đồi với hàm đánh giá được chọn. Cuối mỗi phần, đều có các ví dụ, bài tập minh họa và kiểm tra nhận thức của sinh viên.

Cùng với tập bài giảng, chúng tôi cung cấp tập các bài toán thực tiễn, giúp người học có thêm cơ hội rèn luyện, kiểm tra khả năng lập trình trên các ngôn ngữ lập trình bậc cao.

Chủ biên

TS. Nguyễn Ngọc Thuần

MỤC LỤC

	Nội dung	Trang
Chương 1	Tổng quan về Trí tuệ nhân tạo.	1-5
	1.1. Các khái niệm	1
	1.2. Đối tượng và mục tiêu nghiên cứu	2
	1.3. Vai trò của TTNT	2
	1.4. Các kỹ thuật TTNT	3
	1.5. Những vấn đề đặt ra trong tương lai của TTNT	4
	1.6. Câu hỏi và Bài tập	5
Chương 2	Phương pháp biểu diễn tri thức và các giải thuật.	6 – 49
	2.1. Biểu diễn tri thức nhờ Logic hình thức	6 - 36
	2.1.1. Lô gic mệnh đề	7 – 21
	2.1.2. Các luật đại số	10 – 11
	2.1.3. Một số thuật giải chứng minh	11 – 21
	Thuật toán Vương Hạo	12 – 14
	Thuật toán Robinson	14 – 15
	Các ví dụ	15 – 19
	2.1.4. Bài tập	19 – 21
	2.1.5. Logic vị từ	21 – 36
	Các Định nghĩa và khái niệm	21 – 23
	2.1.5.1. Cú pháp, ngữ nghĩa	24 – 29
	2.1.5.2. Chuẩn hóa công thức	29 - 33
	2.1.5.3. Ví dụ và Bài tập	33 – 36
	2.2. Biểu diễn tri thức sử dụng luật dẫn xuất	37 – 48
	2.2.1. Các khái niệm	37
	2.2.2. Cấu trúc	37
	2.2.3. Suy diễn trên luật sản xuất	39 – 43
	2.2.3.1. Khái niệm	39
	2.2.3.2. Các phương pháp suy diễn	39
	Phương pháp suy diễn tiến	39 – 41
	Phương pháp suy diễn lùi	41 – 43
	2.2.4. Ví dụ và Bài tập	44 – 48
	2.3. Biểu diễn tri thức sử dụng mạng ngữ nghĩa, khung (Frame)	49
Chương 3	Biểu diễn bài toán trong không gian trạng thái	50 – 61
	3.1. Các khái niệm	50 – 51
	3.2. Mô tả trạng thái	52 – 53
	3.3. Toán tử chuyển trạng thái	54 – 56
	3.4. Không gian trạng thái của bài toán	57
	3.5. Biểu diễn không gian trạng thái dưới dạng Graf	58 – 59

	3.6. Ví dụ và Bài tập	60 – 61
Chương 4	Các phương pháp tìm kiếm lời giải trong không gian trạng thái	62 – 90
	Mô tả bài toán	62 – 63
	4.1. Phương pháp tìm kiếm theo chiều rộng	63 – 69
	4.2. Phương pháp tìm kiếm theo chiều sâu	69 – 74
	4.3. Phương pháp tìm kiếm tốt nhất đầu tiên (Thuật toán BFS – Best First Search)	74 – 79
	4.4. Tìm kiếm đường đi có giá thành cực tiểu (Thuật toán ATK – Algorithm for Knowledgeable Tree Search)	80 – 84
	4.5. Tìm kiếm cực tiểu dùng hàm đánh giá (Thuật toán A*)	84 – 87
	4.6. Phương pháp tìm kiếm leo đồi (Hill-climbing search)	87 - 90

TRÍ TUỆ NHÂN TẠO.

TS. Nguyễn Ngọc Thuận

Chương 1 TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO (TTNT).

1.1. Các khái niệm.

Trí tuệ nhân tạo (AI: Artificial Intelligence) là một ngành của khoa học máy tính liên quan đến việc tự động hóa các hành vi thông minh. Trí tuệ nhân tạo (tương lai của nhân loại) không có gì khác hơn là mô phỏng hoạt động của não người - mạng nơ ron (Neural network). AI được xây dựng trên những nguyên lý lý thuyết vững chắc, có khả năng ứng dụng được.

Những nguyên lý đó bao gồm: Các cấu trúc dữ liệu dùng cho biểu diễn tri thức; Các thuật toán cần thiết để xử lý và áp dụng những tri thức đó; Các ngôn ngữ và kỹ thuật lập trình dùng cho việc cài đặt chúng.

+ Khái niệm: TTNT - AI: Artificial Intelligence; (Trí tuệ, Tri thức - knowledge)

CSTT (Cơ sở tri thức)-Tập hợp các tri thức.

+ Vai trò, Chức năng của AI: Tìm kiếm, tập hợp những dữ liệu (mà con người gọi là tri thức, kiến thức); Tìm kiếm CTDL thích hợp để biểu diễn các tri thức đó trong MT; Tìm kiếm các Luật suy diễn phù hợp để từ các CSTT - một tập hợp các tri thức được biểu diễn dưới dạng nào đó, suy ra được các hành động đúng.

+ Các bộ phận cấu thành Hệ TT: CSTT + Bộ suy diễn. Bộ suy diễn là thành phần cơ bản khác của các hệ tri thức. Như vậy hệ tri thức bảo trì một CSTT và được trang bị một thủ tục suy diễn.

+ Bài toán của TTNT gồm các dạng:

- Cách biểu diễn tri thức và đánh giá;
- Các thuật toán xử lý TT và các yếu tố vượt trội.
- Học máy (Machine Learning - khác với việc máy là công cụ lưu trữ):
Đưa dữ liệu vào máy; Hệ thống thuật toán xử lý; Đưa ra kết quả - đây là kiến thức thu/học được. Quá trình được lặp lại cho đến khi thu được kết quả mong đợi.

+ Mục tiêu của biểu diễn tri thức:

Để máy tính có thể sử dụng được tri thức, có thể xử lý tri thức, chúng ta cần biểu diễn tri thức dưới dạng thuận tiện cho máy tính.

Trong Công Nghệ Thông Tin, Trí Tuệ Nhân Tạo (Artificial Intelligence) là một ngành mới, nhưng đã và đang phát triển rất mạnh mẽ và đem lại nhiều kết quả to

lớn. Việc ra đời của máy tính điện tử vào những năm 50 của thế kỷ 20 đã đưa việc nghiên cứu trí tuệ nhân tạo (về các vấn đề lý thuyết và thực nghiệm) vào máy tính trên cả lĩnh vực nghiên cứu lý thuyết và ứng dụng thực tiễn.

1.2. **Đối tượng và mục tiêu nghiên cứu của trí tuệ nhân tạo.**

Hai mối quan tâm nền tảng nhất của các nhà nghiên cứu TTNT là:

Biểu diễn tri thức (knowledge representation) và *tìm kiếm* (search).

Biểu diễn tức là Sử dụng các công cụ: ngôn ngữ hình thức, lô gic mệnh đề, lô gic vị từ, Luật suy diễn, Khung,.....tùy theo bài toán cụ thể (một cách phù hợp) chuyển các bài toán thực tiễn (theo ngôn ngữ tự nhiên) về dạng các mô hình toán học.

Tìm kiếm ở đây được hiểu theo nghĩa: Sử dụng kết hợp các kỹ thuật đã biết để tìm các phương án (lời giải) trên không gian trạng thái bài toán (problem state).

Trí tuệ nhân tạo nghiên cứu về cách hành xử/hành vi thông minh (intelligent behaviour) với mục tiêu là xây dựng đầy đủ lý thuyết về thông minh để có thể giải thích được các hoạt động thông minh của sinh vật và áp dụng được các hiểu biết đó vào các công cụ/máy móc nói chung, nhằm phục vụ cho con người.

1.3. **Vai trò của Trí Tuệ Nhân Tạo.**

Trí tuệ nhân tạo nghiên cứu các lĩnh vực tổng quát như máy nhận biết, suy luận logic, đến các bài toán như chơi cờ, chứng minh định lý, các kỹ thuật hệ thống hoá và tự động hoá các xử lý tri thức cũng như các phương pháp thuộc lĩnh vực mang tính người.

Trí tuệ nhân tạo nghiên cứu kỹ thuật làm cho máy tính có thể “suy nghĩ một cách thông minh” và mô phỏng quá trình suy nghĩ của con người khi đưa ra những quyết định, lời giải. Trên cơ sở đó, thiết kế các chương trình cho máy tính để giải quyết bài toán.

Sự ra đời và phát triển của Trí tuệ nhân tạo đã tạo ra một bước nhảy vọt về chất trong kỹ thuật và kỹ nghệ xử lý thông tin. Điều này được thể hiện qua các mặt sau:

- Nhờ những công cụ hình thức hoá: Các mô hình logic ngôn ngữ, logic mờ,..., các tri thức thủ tục và tri thức mô tả có thể biểu diễn được trong máy. Do vậy quá trình giải bài toán được tiến hành hữu hiệu hơn.

- Logic ngôn ngữ đã mở rộng khả năng ứng dụng của máy tính trong lĩnh vực đòi hỏi tri thức ở trình độ cao, khó như: y học, sinh học, địa lý, tự động hóa.
- Đã có nhiều phần mềm TTNT thể hiện tính thích nghi và tính mềm dẻo đối với các lớp bài toán thuộc nhiều lĩnh vực khác nhau: Y học, Dự báo, Thiết kế, Xây dựng, Thương mại,.....

1.4. Các kỹ thuật Trí tuệ nhân tạo.

Có nhiều kỹ thuật nghiên cứu, phát triển Trí tuệ nhân tạo. Tuy vậy, các kỹ thuật Trí tuệ nhân tạo thường khá phức tạp khi thực hiện cài đặt cụ thể, vì nó liên quan đến xử lý các ký hiệu.

Các kỹ thuật Trí tuệ nhân tạo cơ bản bao gồm :

- **Lý thuyết giải bài toán và suy diễn thông minh:** Viết các chương trình giải câu đố, các trò chơi thông qua các suy luận mang tính người; các hệ thống chứng minh định lý.
- **Lý thuyết thể hiện tri thức và hệ chuyên gia:** Trí tuệ nhân tạo là khoa học về thể hiện và sử dụng tri thức.
- **Mạng ngữ nghĩa, lược đồ, logic vị từ, khung là các phương pháp thể hiện tri thức thông dụng.** Việc gắn liền cách thể hiện và sử dụng tri thức là cơ sở hình thành hệ chuyên gia.
- **Lý thuyết nhận dạng và xử lý tiếng nói:** Các phương pháp nhận dạng chính gồm: nhận dạng hình học, nhận dạng dùng tâm lý học, nhận dạng theo phương pháp hàm thế, dùng máy nhận dạng, nhận dạng chữ viết, âm thanh.
- **Người máy:** Cuối những năm 70, người máy trong công nghiệp đã đạt được nhiều tiến bộ. Người máy có bộ phận cảm nhận và các cơ chế hoạt động được nối ghép theo sự điều khiển thông minh.
- **Tâm lý học xử lý thông tin. Xử lý danh sách, kỹ thuật đệ quy, kỹ thuật quay lui và xử lý cú pháp hình thức,....** là những kỹ thuật cơ bản của tin học truyền thống có liên quan trực tiếp đến Trí tuệ nhân tạo.

1.5. Các khái niệm cơ bản về trí tuệ:

Trí tuệ con người (Human Intelligence):

Có hai khái niệm về trí tuệ con người được chấp nhận và sử dụng nhiều nhất, đó là:

- Khái niệm trí tuệ theo quan điểm của Turing

“Trí tuệ là những gì có thể đánh giá được thông qua các trắc nghiệm thông minh”

- Khái niệm trí tuệ đưa ra trong từ điển bách khoa toàn thư: “Trí tuệ là khả năng phản ứng một cách thích hợp những tình huống mới thông qua hiệu chỉnh hành vi một cách thích đáng. Hiểu rõ những mối liên hệ qua lại của các sự kiện của thế giới bên ngoài nhằm đưa ra những hành động phù hợp đạt tới một mục đích nào đó.

Theo tâm lý học, quá trình hoạt động trí tuệ của con người bao gồm 4 thao tác cơ bản:

- 1- Xác định tập đích (goals).
- 2- Thu thập các sự kiện (facts) và các luật suy diễn (inference rules) để đạt được đích đặt ra.
- 3- Thu gọn (pruning) quá trình suy luận nhằm xác định tập các suy diễn có thể sử dụng được.
- 4- Áp dụng các cơ chế suy diễn cụ thể (inference mechanisms) để đưa các sự kiện ban đầu đi đến đích.

Trí tuệ máy:

Không có một định nghĩa tổng quát, nhưng cũng có thể nêu các đặc trưng chính:

- 1- Khả năng học.
- 2- Khả năng mô phỏng hành vi của con người.
- 3- Khả năng trừu tượng hoá, tổng quát hoá và suy diễn .
- 4- Khả năng tự giải thích hành vi.
- 5- Khả năng thích nghi tình huống mới kể cả thu nạp tri thức và dữ liệu.
- 6- Khả năng xử lý các biểu diễn hình thức như các ký hiệu tượng trưng.
- 7- Khả năng sử dụng tri thức heuristic.
- 8- Khả năng xử lý các thông tin không đầy đủ, không chính xác

1.6. Những vấn đề đặt ra trong tương lai và các chuyên ngành của Trí tuệ nhân tạo.

Trong tương lai, những nghiên cứu và ứng dụng của TTNT tập trung vào các vấn đề lớn sau:

+ Nghiên cứu và ứng dụng các mạng Neuron; Các hệ thống Trí tuệ nhân tạo mô phỏng chức năng hoạt động của bộ não với các khả năng học, tự tổ chức, tự thích nghi, tổng quát hoá, xử lý song song, có khả năng diễn giải, xử lý thông tin liên tục và rời rạc.

+ Nghiên cứu và tạo lập các hệ thống có giao tiếp thân thiện giữa người và máy trên cơ sở nghiên cứu nhận thức máy, thu thập và xử lý tri thức, xử lý thông tin hình ảnh, tiếng nói.

+ Nghiên cứu các phương pháp biểu diễn tri thức và các phương pháp suy diễn thông minh, các phương pháp giải quyết vấn đề đối với những bài toán phụ thuộc không gian, thời gian.

Các chuyên ngành:

1. Các phương pháp tìm kiếm lời giải.
2. Hệ chuyên gia
3. Lý thuyết nhận dạng.
4. Các mô hình thần kinh.
5. Người máy.

. . .

1.7. Câu hỏi, Bài tập.

1. Nêu sự khái quát của riêng bạn về Trí tuệ nhân tạo. Trí tuệ người và máy?
2. Hãy so sánh những điểm khác nhau cơ bản trong khả năng giải quyết vấn đề giữa máy tính hiện đại với con người.
3. Hãy cho biết một số ứng dụng của TTNT trong cuộc sống xã hội mà bạn biết.

Chương 2 PHƯƠNG PHÁP BIỂU DIỄN TRI THỨC VÀ CÁC GIẢI THUẬT

Như ta đã biết, không thể có một 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*** và các ***phương pháp tìm kiếm lời giải trong không gian bài toán nhận được***.

Đầu tiên, chúng ta quan tâm đến vấn đề biểu diễn.

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

Để máy tính có thể *sử dụng được tri thức, có thể xử lý được tri thức*, chúng ta cần phải biểu diễn tri thức dưới dạng thuận tiện cho máy tính (để máy tính hiểu được).

Xét cho cùng, mọi chương trình máy tính đều gồm hai thành phần là: Các mã lệnh và Dữ liệu. Mã lệnh được ví như là phần cứng của chương trình còn Dữ liệu được xem là phần mềm (vì nó có thể được thay đổi bởi người dùng). Do đó, "mềm" hóa tri thức cũng đồng nghĩa với việc tìm các phương pháp để có thể *biểu diễn các loại tri thức của con người* bằng các *cấu trúc dữ liệu* mà máy tính có thể xử lý được. Đây cũng chính là ý nghĩa của thuật ngữ "biểu diễn tri thức".

Từ đây, Các phương pháp đưa tri thức vào máy tính được gọi là *biểu diễn tri thức* và là một phần quan trọng trong TTNT (AI).

Trong Chương này sẽ trình bày một số phương pháp biểu diễn và các phương pháp giải quyết vấn đề trên cách biểu diễn đó.

2.1. BIỂU DIỄN VẤN ĐỀ NHỜ LOGIC HÌNH THỨC

Một vài nét về ứng dụng của Logic hình thức:

- + Logic hình thức thường dùng để loại bỏ, thu gọn quá trình tìm kiếm lời giải. Nhờ phân tích logic, có thể chứng tỏ rằng một bài toán nào đó có thể giải được hay không trước khi giải quyết.

- + Kiểm tra các kết luận logic. Điều rất cần, ngay cả trong cách tiếp cận dựa trên không gian trạng thái và quy bài toán về bài toán con.

- + Logic hình thức có thể được sử dụng để giải quyết những bài toán chứng minh logic: 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

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 $1/2$ 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?

2.1.1. Logic mệnh đề (Nhắc lại)

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

1) 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 mà còn phụ thuộc vào thời gian, không gian và nhiều yếu tố khách quan khác. 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.)

2) 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 ($\neg, \sim, \bar{}$), 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 (cấu trúc lệnh IF ... THEN ... ELSE: $(p \wedge q \vee s) \Rightarrow r$) để biểu diễn các tri thức "cứng" trong máy tính !)

3) Bảng chân trị (bảng chân lý) Dùng để đánh giá giá trị của biểu thức logic.

p	q	$\neg p$	$p \vee q$	$p \wedge q$	$\neg p \vee q$	$p \rightarrow q$	$p \equiv q$
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:

$$p \rightarrow q \equiv \neg p \vee q$$

4) 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) $0 \rightarrow p$
- 3) $(p \vee q) \wedge (\neg p \vee r) \rightarrow q \vee r$

Xét ví dụ 3): Ta thấy rằng biểu thức có dạng $VT \rightarrow VP$ luôn có giá trị True (T) với mọi giá trị của a, b ; chỉ có một trường hợp để $a \rightarrow b$ 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.

I. 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) $((p \rightarrow q \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r))$

II. Hãy xây dựng các câu sau bằng các biểu thức logic mệnh đề

“Nếu trời mưa thì Lan mang theo dù”

“Nếu Lan mang theo dù thì Lan không bị ướt”

“Nếu trời không mưa thì Lan không bị ướt”

Giải.

Đặt: r : “Trời mưa” u : “Lan mang theo dù” w : “Lan bị ướt”

Lúc đó, ta có các biểu thức logic đối với các câu/mệnh đề trên như sau:

$$r \rightarrow u; u \rightarrow \neg w; \neg r \rightarrow \neg w.$$

III. Hãy biểu diễn các tri thức sau bằng logic mệnh đề

a. Nếu n là số nguyên chẵn và n là số nguyên tố thì $n=2$

b. Số n là chính phương thì n tận cùng bằng 1, 4, 5, 6 hoặc 9

Lời giải

- a. Gọi a : “là một số nguyên chẵn”,
 b : “là một số nguyên tố”,
 c : “số nguyên 2”.

Lúc đó, tri thức sẽ được biểu diễn như sau: $ab \rightarrow c$

- b. Gọi a : “là một số chính phương”,
 b : “số có chữ số tận cùng bằng 1”
 c : “số có chữ số tận cùng bằng 4”
 d : “số có chữ số tận cùng bằng 5”
 e : “số có chữ số tận cùng bằng 6”
 f : “số có chữ số tận cùng bằng 9”.

Lúc đó, tri thức sẽ được biểu diễn như sau: $a \rightarrow b+c+d+e+f$

2.1.2. Một số luật đại số (11 luật)

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 $\vee$ (tuyển): $p \vee 0 \equiv p$

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

g) Triệt tử

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

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

h) Tính lũy đẳng

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

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

i) Luật De Morgan

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

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

j) Một số luật khác cho phép kéo theo

$$(p \rightarrow q) \wedge (q \rightarrow p) \equiv (p \equiv q)$$

$$(p \equiv q) \rightarrow (p \rightarrow q)$$

$$p \rightarrow q \equiv \neg p \vee q$$

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

2.1.3. MỘT SỐ THUẬT GIẢI CHỨNG MINH (Logic mệnh đề)

Một trong những vấn đề khá quan trọng của logic mệnh đề là chứng minh tính đúng đắn của phép suy diễn ($a \sqsubseteq b$). Đây cũng chính là bài toán chứng minh thường gặp trong toán học.

Rõ ràng rằng với hai phép suy luận cơ bản của logic mệnh đề (Modus Ponens, Modus Tollens) cộng với các phép biến đổi hình thức (sẽ được đề cập ở Mục sau), ta cũng có thể chứng minh được phép suy diễn. Tuy nhiên, thao tác biến đổi hình thức là rất khó cài đặt được trên máy tính.

Với công cụ máy tính, có thể cho rằng ta sẽ dễ dàng chứng minh được mọi bài toán bằng một phương pháp "thô bạo" là lập bảng chân trị. Tuy về lý thuyết, phương pháp lập bảng chân trị luôn cho được kết quả cuối cùng nhưng độ phức tạp của phương pháp này là quá lớn, $O(2^n)$ với n là số biến mệnh đề.

Sau đây sẽ nghiên cứu hai phương pháp chứng minh mệnh đề với độ phức tạp chỉ có $O(n)$.

Bài toán Cho tập các giả thiết dưới dạng các biểu thức logic mệnh đề $GT = \{GT_1, GT_2, \dots, GT_n\}$. Hãy chứng minh tập kết luận $KL = \{KL_1, KL_2, \dots, KL_m\}$.

Thuật giải Vương Hạo

Cơ sở lý luận: Cho các giả thiết $GT_1, GT_2, \dots, GT_n$. Để chứng minh tập kết luận $KL_1, KL_2, \dots, KL_m$, ta chứng minh $GT_1, GT_2, \dots, GT_n \rightarrow KL_1, KL_2, \dots, KL_m$: True

Thuật giải bao gồm các bước sau:

B1: Phát biểu lại giả thiết và kết luận của vấn đề theo **dạng chuẩn** sau :

$$GT_1, GT_2, \dots, GT_n \sqsubseteq KL_1, KL_2, \dots, KL_m$$

Trong đó các GT_i và KL_i là các biểu thức logic dạng chuẩn (chỉ chứa 3 phép toán cơ bản : $\neg$, $\vee$, $\wedge$)

B2: Nếu GT_i có phép $\neg$ thì tách thành hai dòng con.

Nếu ở KL_i có phép $\neg$ thì tách thành hai dòng con.

Ví dụ: $p \wedge (\neg p \vee q) \vee q$ thì tách thành 2 dòng:

$$p \wedge \neg p \vee q \quad \text{và} \quad p \wedge q \vee q$$

Hoặc nếu có $p \wedge q \rightarrow q \wedge r$ thì tách thành 2 dòng:

$$p \wedge q \rightarrow q \quad \text{và} \quad p \wedge q \rightarrow r$$

B3: Một dòng được chứng minh nếu:

1) Tồn tại chung một mệnh đề ở cả hai phía.

$$\text{Ví dụ: } p \wedge q \vee \neg q \vee r: \text{True}$$

2) Tồn tại 2 mệnh đề phủ định lẫn nhau (p và $\neg p$)

$$\text{Ví dụ: } p \wedge \neg p \vee q: \text{True}$$

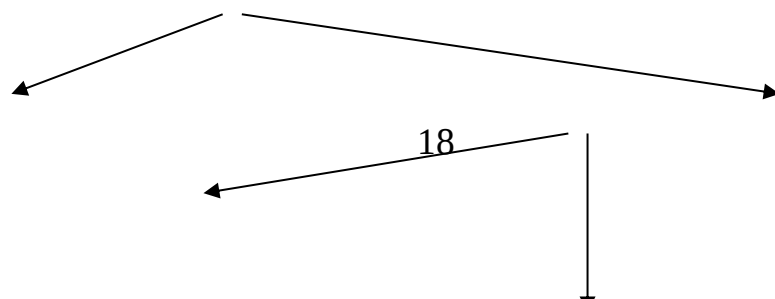
B4 : a) Nếu một dòng không còn phép nối $\vee$ hoặc $\wedge$ ở cả hai vế và ở 2 vế không có chung một biến mệnh đề thì dòng đó không được chứng minh.

b) Một vấn đề được chứng minh nếu tất cả dòng dẫn xuất từ dạng chuẩn ban đầu đều được chứng minh.

Ví dụ

1) Cần chứng minh rằng từ: $a \wedge b \rightarrow c$ và $b \wedge c \rightarrow d$ và a và b , suy ra d

$$(\neg a \vee \neg b \vee c) \wedge (\neg b \vee \neg c \vee d) \wedge a \wedge b \rightarrow d$$



$$\neg \mathbf{a} \wedge (\neg \mathbf{b} \vee \neg \mathbf{c} \vee \mathbf{d}) \wedge \mathbf{a} \wedge \mathbf{b} \rightarrow \mathbf{d}: \mathbf{T}$$

$$\rightarrow \mathbf{d}$$

$$(\neg \mathbf{b} \vee \mathbf{c}) \wedge (\neg \mathbf{b} \vee \neg \mathbf{c} \vee \mathbf{d}) \wedge \mathbf{a} \wedge \mathbf{b}$$

$$\neg \mathbf{b} \wedge (\neg \mathbf{b} \vee \neg \mathbf{c} \vee \mathbf{d}) \wedge \mathbf{a} \wedge \mathbf{b} \rightarrow \mathbf{d}: \mathbf{T}$$

$$\mathbf{c} \wedge (\neg \mathbf{b} \vee \neg \mathbf{c} \vee \mathbf{d}) \wedge \mathbf{a} \wedge \mathbf{b} \rightarrow \mathbf{d}$$

$$\mathbf{c} \wedge \neg \mathbf{b} \wedge \mathbf{a} \wedge \mathbf{b} \rightarrow \mathbf{d}: \mathbf{T}$$

$$\mathbf{c} \wedge (\neg \mathbf{c} \vee \mathbf{d}) \wedge \mathbf{a} \wedge \mathbf{b} \rightarrow \mathbf{d}$$

$$\mathbf{c} \wedge \neg \mathbf{c} \wedge \mathbf{a} \wedge \mathbf{b} \rightarrow \mathbf{d}: \mathbf{T}$$

$$\mathbf{c} \wedge \mathbf{d} \wedge \mathbf{a} \wedge \mathbf{b} \rightarrow \mathbf{d}: \mathbf{T}$$

Cây suy diễn trong giải thuật Vương Hạo

2) Xét các câu đúng sau:

“Nếu trời mưa thì Lan mang theo dù”

“Nếu Lan mang theo dù thì Lan không bị ướt”

“Nếu trời không mưa thì Lan không bị ướt”

a) Xây dựng các câu trên bằng các biểu thức logic mệnh đề

b) Hãy chứng minh rằng “Lan không bị ướt” bằng phương pháp Vương Hạo

Lời giải

a) r: “Trời mưa”; u: “Lan mang theo dù”; w: “Lan bị ướt”

Lúc đó, ta có các biểu thức logic đúng sau:

$$r \rightarrow u; u \rightarrow \neg w; \neg r \rightarrow \neg w$$

b). Ta phải chứng minh $(r \rightarrow u) \wedge (u \rightarrow \neg w) \wedge (\neg r \rightarrow \neg w) \rightarrow \neg w$: True

Thuật giải Robinson

Thuật giải này do Robinson đề xuất và hoạt động dựa trên *phương pháp chứng minh phản chứng*.

Phương pháp chứng minh phản chứng:

Bài toán: Chứng minh phép suy luận $(a \sqsubseteq b)$ là đúng (với a là giả thiết, b là kết luận).

Phản chứng: giả sử b sai, suy ra $\sqsubseteq b$ là đúng.

Bài toán được chứng minh, nếu a đúng và $\sqsubseteq b$ đúng sinh ra một mâu thuẫn.

B1 : Phát biểu lại giả thiết và kết luận của vấn đề dưới dạng chuẩn như sau:

$$GT_1, GT_2, \dots, GT_n \sqsubseteq KL_1, KL_2, \dots, KL_m$$

Trong đó : GT_i và KL_j là các biểu thức logic dạng chuẩn (chỉ chứa các phép toán : $\sqsubseteq, \neg, \wedge, \vee$)

B2 : Giả sử $\neg KL_1, \neg KL_2, \dots, \neg KL_m$ là đúng, lúc đó ta có các biểu thức logic đúng sau:

$$\{ GT_1, GT_2, \dots, GT_n, \neg KL_1, \neg KL_2, \dots, \neg KL_m \}$$

B3 : Sau đó đưa về dạng chuẩn hội (tích các tổng)

Ví dụ:

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

B4 : Xây dựng một mệnh đề mới bằng cách áp dụng đồng nhất đúng:

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

Nghĩa là: nếu $(\neg p \vee q) \wedge (\neg p \vee r)$: True thì có thêm biểu thức $q \vee r$: True và đưa vào tập giả thiết.

Lặp lại quá trình trên cho đến khi sinh ra 2 mệnh đề có chân trị đối nhau (có sự mâu thuẫn) và bài toán lúc đó kết luận là được chứng minh, hoặc không tạo thêm mệnh đề mới nào gây mâu thuẫn và lúc này kết luận không chứng minh được.

Ví dụ

3) Xét bài toán ở Ví dụ 1) bằng Thuật toán Robínson.

Chứng minh rằng từ: $a \wedge b \rightarrow c$ và $b \wedge c \rightarrow d$ và a và b , suy ra d .

Đưa về dạng chuẩn hội, ta viết các biểu thức (chỉ chứa phép $\vee$) trên từng dòng. Ta có:

1. $\neg a \vee \neg b \vee c$
2. $\neg b \vee \neg c \vee d$
3. a
4. b

Giả sử $\neg d$ đúng, ta có thêm các biểu thức đúng sau:

5. $\neg d$
6. $\neg b \vee c$ (Từ 1a, 3)
7. $\neg a \vee c$ (Từ 1b, 4)
8. $\neg c \vee d$ (Từ 2b, 4)
9. $\neg b \vee \neg c$ (Từ 2c, 5)
10. c (Từ 3, 7a)

11. $\neg c$ (Từ 4, 9b)

Mâu thuẫn giữa 10 và 11. Chứng minh xong.

2.1.4. Ví dụ và Bài tập.

Để trình bày gọn, chúng ta quy ước

- Phép và (hội hay tích)

Ví dụ: a và b được ký hiệu $a \wedge b$ hoặc ab

- *Phép hoặc (tuyển hay tổng)*

Ví dụ: a hoặc b được ký hiệu $a \vee b$ hoặc $a+b$

- Phép phủ định

Ví dụ: phủ định của a được ký hiệu $\neg a$

- *Phép kéo theo (suy ra)*

Ví dụ: a kéo theo b được ký hiệu $a \rightarrow b$

Ví dụ 1 Cho các khẳng định đúng với các ý nghĩa như sau:

a	một người có nhóm máu A
b	một người có nhóm máu B
c	một người có nhóm máu AB
o	một người có nhóm máu O
t	mẫu máu của một người có xét nghiệm T dương tính
s	mẫu máu của một người có xét nghiệm S dương tính

Hãy viết các biểu thức logic diễn tả những ý tưởng sau:

- Nếu xét nghiệm T dương tính thì người đó có nhóm máu A hoặc AB
- Nếu xét nghiệm S dương tính thì người đó có nhóm máu B hoặc AB
- Nếu một người có nhóm máu B thì xét nghiệm S sẽ dương tính
- Một người có nhóm máu A, B, AB hoặc O

Lời giải

- $t \rightarrow a+c$
- $s \rightarrow b+c$
- $b \rightarrow s$

d. $a + b + c + o$

Ví dụ 2. Hãy biểu diễn các tri thức sau bằng logic mệnh đề

a. Nếu n là số nguyên chẵn và n là số nguyên tố thì $n=2$

b. Số n là chính phương thì n tận cùng bằng 1, 4, 5, 6 hoặc 9

Lời giải

a. Gọi a : “là một số nguyên chẵn”,

b : “là một số nguyên tố”,

c : “số nguyên 2”.

Lúc đó, tri thức sẽ được biểu diễn như sau: $ab \rightarrow c$

b. Gọi a : “là một số chính phương”,

b : “số có chữ số tận cùng bằng 1”

c : “số có chữ số tận cùng bằng 4”

d : “số có chữ số tận cùng bằng 5”

e : “số có chữ số tận cùng bằng 6”

f : “số có chữ số tận cùng bằng 9”.

Lúc đó, tri thức sẽ được biểu diễn như sau: $a \rightarrow b + c + d + e + f$

Ví dụ 3 : Cho các biểu thức logic mệnh đề đúng sau

1. $a \rightarrow f$

2. $a \rightarrow (f \rightarrow p)$

3. $p + q \rightarrow d$

4. a

5. $ad \rightarrow g$

Hãy dùng phương pháp Robinson và Vương Hạo để chứng minh hoặc bác bỏ $g \equiv 1$

Lời giải

- **Chuyển về dạng chuẩn**

1). $a \rightarrow f \equiv \neg a + f$

2). $a \rightarrow (f \rightarrow p) \equiv \neg a + (f \rightarrow p) \equiv \neg a + (\neg f + p) \equiv \neg a + \neg f + p$

$$3). p+q \rightarrow d \equiv \neg(p+q)+d \equiv (\neg p \neg q)+d \equiv (\neg p + d)(\neg q + d) . \forall i$$

Luật De Morgan

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

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

Luật phân phối

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

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

$$5). ad \rightarrow g \equiv \neg(ad)+g \equiv \neg a + \neg d + g$$

Dùng phương pháp phân giải Robinson

Giả sử $\neg g \equiv 1$, ta có các biểu thức đúng sau

$$1. \neg a + f$$

$$2. \neg a + \neg f + p$$

$$3. \neg p + d$$

$$4. \neg q + d$$

$$5. a$$

$$6. \neg a + \neg d + g$$

$$7. \neg g$$

$$8. \neg a + \neg d \quad \text{Từ } (6,7)$$

$$9. \neg d \quad \text{Từ } (5, 8_1)$$

$$10. \neg a + \neg f + d \quad \text{Từ } (2,3,3_1)$$

$$11. \neg a + d \quad \text{Từ } (1,10_2)$$

$$12. d \quad \text{Từ } (5,11_1)$$

Ta thấy 9) kết hợp 12) sẽ cho ra câu rỗng (tức là có sự mâu thuẫn). Vì vậy $g \equiv 1$.

- Dùng phương pháp Vương Hạo

Ta cần chứng minh:

$$(a)(\neg a + f)(\neg a + \neg f + p)(\neg p + d)(\neg q + d)(\neg a + \neg d + g) \rightarrow g: \text{true (I)}$$

$$(1) \quad (2) \quad (3) \quad (4) \quad (5)$$

Để chứng minh (I) tách (1), biểu thức (I) trở thành:

$$I.1) \quad (a)(\neg a)(\neg a + \neg f + p)(\neg p + d)(\neg q + d)(\neg a + \neg d + g) \rightarrow g : \text{true}$$

$$I.2) \quad af(\neg a + \neg f + p)(\neg p + d)(\neg q + d)(\neg a + \neg d + g) \rightarrow g$$

Chứng minh I.2) tách 2), I.2) trở thành

$$I.2.1) \quad af(\neg a)(\neg p + d)(\neg q + d)(\neg a + \neg d + g) \rightarrow g : \text{true}$$

$$I.2.2) \quad af(\neg f)(\neg p + d)(\neg q + d)(\neg a + \neg d + g) \rightarrow g : \text{true}$$

$$I.2.3) \quad afp(\neg p + d)(\neg q + d)(\neg a + \neg d + g) \rightarrow g$$

Chứng minh I.2.3) tách 3), I.2.3) trở thành

$$I.2.3.1) \quad afp(\neg p)(\neg q + d)(\neg a + \neg d + g) \rightarrow g : \text{true}$$

$$I.2.3.2) \quad afpd(\neg q + d)(\neg a + \neg d + g) \rightarrow g$$

Chứng minh I.2.3.2) tách 5), I.2.3.2) trở thành

$$I.2.3.2.1) \quad afpd(\neg q + d)(\neg a) \rightarrow g : \text{true}$$

$$I.2.3.2.2) \quad afpd(\neg q + d)(\neg d) \rightarrow g : \text{true}$$

$$I.2.3.2.3) \quad afpd(\neg q + d)g \rightarrow g : \text{true}$$

(I) được chứng minh, vì vậy $g \equiv 1$

Nhóm bài tập Logic mệnh đề.

Bài tập1. Biểu diễn các tri thức sau dưới dạng logic mệnh đề

- Trong tam giác vuông, tổng bình phương chiều dài hai cạnh góc vuông bằng bình phương chiều dài cạnh huyền
- Một số nguyên dương có chữ số hàng đơn vị bằng 5 thì số đó chia hết cho 5
- Nếu x là số lẻ và bình phương của x tận cùng bằng 1 thì x tận cùng bằng 1 hoặc bằng 9
- Trong tam giác vuông, chiều dài của đường trung tuyến xuất phát từ góc vuông bằng nửa chiều dài của cạnh huyền

Bài tập 2. Cho các biểu thức logic mệnh đề đúng sau

$$1. \quad n + c + d \rightarrow p$$

$$2. \quad qp \rightarrow c$$

$$3. \quad qc \rightarrow f + \neg h$$

4. $\neg n + \neg p + h$

5. nq

Hãy dùng phương pháp Robinson và Vương Hạo để chứng minh hoặc bác bỏ $f \equiv 1$

Bài tập 3. Cho các biểu thức logic mệnh đề đúng sau

1. $abc \rightarrow c$

2. $abc \rightarrow p$

3. $as \rightarrow h$

4. $bcp \rightarrow s$

5. abd

Hãy dùng phương pháp Robinson và Vương Hạo để chứng minh hoặc bác bỏ $sh \equiv 1$

Bài tập 4. Ta có cơ sở tri thức của hệ chuyên gia về *bệnh cảm cúm* như sau:

- 1) “Nếu bệnh nhân rất họng và viêm nhiễm thì viêm họng và đi chữa họng”
- 2) “Nếu thân nhiệt $> 37^0$ thì sốt”
- 3) “Nếu ốm trên 7 ngày và sốt thì viêm nhiễm”
- 4) “Nếu sốt và ho và kèm theo khó thở hoặc kèm theo tằng ran thì viêm phổi”

a) Hãy biểu diễn các tri thức trên dưới dạng logic mệnh đề.

b) Có một bệnh nhân khai : “Thân nhiệt $> 37^0$ “ và “ốm trên 7 ngày”. Dùng phương pháp chứng minh Robinson và Vương Hạo để kết luận bệnh nhân này bị "viêm nhiễm".

Bài tập 5. Ta có cơ sở tri thức mô tả mối quan hệ của các thành phần trong một tam giác như sau:

- Nếu biết 3 cạnh của 1 tam giác ta có thể biết nửa chu vi của tam giác đó
- Nếu biết 2 cạnh và nửa chu vi của một tam giác thì ta có thể biết được cạnh còn lại của tam giác đó

- Nếu biết được diện tích và một cạnh của một tam giác thì ta có thể biết được chiều cao tương ứng với cạnh đó
- Nếu biết 2 cạnh và một góc kẹp giữa 2 cạnh đó của một tam giác thì ta có thể biết được cạnh còn lại của tam giác đó.
- Nếu biết 2 cạnh và một góc kẹp giữa 2 cạnh đó của một tam giác thì ta có thể biết được diện tích của tam giác đó
- Nếu biết ba cạnh và nửa chu vi của một tam giác thì ta biết được diện tích của tam giác đó
- Nếu biết diện tích và đường cao của một tam giác thì ta biết được cạnh tương ứng với đường cao của tam giác đó

Giả sử biết được 2 cạnh và và góc kẹp giữ hai cạnh đó. Bằng phương pháp Robinson, hãy chứng minh rằng ta có thể suy ra được đường cao tương ứng với cạnh còn lại

Hướng dẫn

Ký hiệu a : cạnh a của tam giác k : đường cao tương ứng với cạnh a
 b : cạnh b của tam giác l : đường cao tương ứng với cạnh b
 c : cạnh c của tam giác m : đường cao tương ứng với cạnh c
 A : góc tương ứng với cạnh a S : diện tích của tam giác
 B : góc tương ứng với cạnh b p : nửa chu vi của tam giác
 C : góc tương ứng với cạnh c

- Các tri thức đó được biểu diễn dưới dạng logic mệnh đề như sau:

- | | |
|-------------------------|--------------------------|
| 10) $abc \rightarrow p$ | 19) $bcA \rightarrow a$ |
| 11) $bpc \rightarrow a$ | 20) $abC \rightarrow S$ |
| 12) $apc \rightarrow b$ | 21) $acB \rightarrow S$ |
| 13) $abp \rightarrow c$ | 22) $bcA \rightarrow S$ |
| 14) $Sa \rightarrow k$ | 23) $abcp \rightarrow S$ |
| 15) $Sb \rightarrow l$ | 24) $Sk \rightarrow a$ |
| 16) $Sc \rightarrow m$ | 25) $Sl \rightarrow b$ |
| 17) $abC \rightarrow c$ | 26) $Sm \rightarrow c$ |

18) $acB \rightarrow b$

Sau đó dùng phương pháp Robinson ($GT=\{a, b\}$, $KL=\{m\}$)

2.1.5. Logic vị từ (Cấp 1)

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 một 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 dựa vào các *vị từ* (predicate).

Ngoài các kết nối logic như trong logic mệnh đề (5 phép toán), logic vị từ cấp một còn sử dụng các *lượng tử*. ($\forall$: **với mọi**, $\exists$: **tồn tại**) để tăng cường tính cấu trúc của một mệnh đề.

Phần 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.

Ví dụ: Khi đặt mệnh đề “hôm qua trời mưa” là P, thì chúng ta có thể tạo ra một vị từ chỉ thời tiết mô tả quan hệ giữa một ngày và thời tiết trong ngày ấy:

thời_tiết (hôm_qua, mưa).

Định nghĩa: Trong logic vị từ, một mệnh đề *được cấu tạo bởi* hai thành phần: 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 :

Mệnh đề: Cam có vị Ngọt $\Rightarrow$ Vị từ: *Vị* (Cam, Ngọt)

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

... “hôm qua trời mưa” $\Rightarrow$ *Thời_tiết* (hôm_qua, mưa).

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, đó 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) Mệnh đề : “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 C sao cho : *Bố* (A, C) và (*Anh*(C, B) hoặc *Anh*(B,C))

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", "Khang") có giá trị đúng (An là bố của Khang)

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

thì mệnh đề

c) Bô ("An", "Tú") sẽ có giá trị là đúng. (An là bô 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 B)" 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 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 :

$$\text{NgườiXấu}(x) = \forall y : \text{Bạn}(x,y) \text{ và } \text{NgườiXấu}(y)$$

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 (Programming Logic) và ngôn ngữ SQL. Logic vị từ cũng được sử dụng trong các hệ thống suy luận hoặc các hệ chuyên gia như: Các chương trình chẩn đoán tự động y khoa, các chương trình chứng minh định lý tự động,.....

2.1.5.1. Cú pháp và ngữ nghĩa của logic vị từ

1. 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 ($n \geq 0$). 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 ($n \geq 1$). Ví dụ: $\cos$, $\sin$ là hàm một biến

- **Lượng từ:** $\forall$ (với mọi), $\exists$ (tồn tại).

Ví dụ: $\forall X, p(X)$ nghĩa là p đúng với mọi giá trị của biến lượng giá X .

$\exists X, p(X)$ nghĩa là p đúng với ít nhất một giá trị của biến lượng giá X .

- **Các ký hiệu kết nối logic:** $\wedge$ (hội), $\vee$ (tuyển), $\neg$ (phủ định), $\Rightarrow$ (kéo theo), $\Leftrightarrow$ (tương đương).

- **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 $t_1, t_2, t_3, \dots, t_n$ là n hạng thức và f là một ký hiệu hàm n biến thì $f(t_1, t_2, t_3, \dots, t_n)$ 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 (câu đơn).**

Dùng để biểu diễn các tính chất của đối tượng, hoặc các quan hệ giữa các đối tượng.

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 $t_1, t_2, t_3, \dots, t_n$ là n hạng thức và p là vị từ của n biến thì $p(t_1, t_2, t_3, \dots, t_n)$ 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, thì $love("Hoa", husband("Hoa"))$ là một công thức phân tử.

- **Các công thức (câu).**

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

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 $(G \wedge H)$, $(G \vee H)$, $(\neg G)$, $(G \Rightarrow H)$, $(G \Leftrightarrow H)$ 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 từ phổ dụng (universal quantifier) cho phép mô tả tính chất của cả một lớp các đối tượng chứ không phải của một đối tượng. Ví dụ: Vị từ $Voi(X)$ (đối tượng X là con voi) và vị từ $Màu(X, "Xám")$ (đối tượng X có màu xám) thì câu: “tất cả các con voi đều có màu xám” có thể biểu diễn bởi công thức: $\forall X (Voi(X) \Rightarrow Màu(X, "Xám"))$

Lượng từ tồn tại (existential quantifier) 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.

Ví dụ: Sử dụng các câu đơn $student(X)$ (X là sinh viên) và $Ở(X, "P301")$ (X ở trong phòng 301), ta có thể biểu diễn câu “Có một sinh viên ở phòng 301” bởi biểu thức: $\exists X (student(X) \wedge Ở(X, "P301"))$

Một công thức là công thức phân tử hoặc phủ định công thức phân tử được gọi là *literal*.

Ví dụ: $play(X, "Football")$, $\neg like("Lan", "Rose")$ là các literal. Một công thức là tuyển của các literal sẽ được gọi là câu tuyển. Chẳng hạn, $male(X) \vee \neg like(X, "Football")$ 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 từ nào cả (sự xuất hiện của Y gọi là sự xuất hiện tự do)

2. Ngữ nghĩa

Tương tự như phép tính mệnh đề, ngữ nghĩa của phép tính vị từ cung cấp một cơ sở để xác

định chân trị của các biểu thức dạng chuẩn. Chân trị của các biểu thức phụ thuộc vào ánh xạ

từ các hằng, các biến, các vị từ và các hàm vào các đối tượng và quan hệ trong lĩnh vực được

đề cập.

Nói đến ngữ nghĩa là chúng ta nói đến ý nghĩa của các công thức (câu) trong một thế giới hiện thực nào đó mà chúng ta sẽ gọi là một minh họa/thể hiện (intepretation). 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 thực mà ta quan tâm).

Vậy, ngữ nghĩa là ý nghĩa của một công thức (Câu) trong một thế giới thực cụ thể.

Trong một minh họa, các ký hiệu hằng sẽ được gán với các đối tượng cụ thể trong miền đối tượng, các ký hiệu hàm sẽ được gán 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.

Ví dụ: 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) gán 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 họa, các ký hiệu vị từ sẽ được gán với một thuộc tính hoặc một quan hệ cụ thể. Khi đó, mỗi công thức phân tử (không chứa biến) sẽ chỉ định một sự kiện cụ thể. Sự kiện này có thể là đúng (true) hoặc sai (false).

Ví dụ: Nếu ký hiệu “Lan” ứng với một cô gái cụ thể nào đó; Student(X) ứng với thuộc tính X là sinh viên, thì câu Student(“Lan”) có giá trị: **True** nếu Lan là sinh viên và là: **False** nếu Lan không phải là sinh viên.

- **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 các liên kết các câu đơn bởi các kết nối logic) như trong logic mệnh đề

Ví dụ: **Câu** $\text{student(“Lan”) } \wedge \text{ student(“An”)}$ 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 viên.

- **Ngữ nghĩa của các câu chứa các lượng từ**

Ngữ nghĩa của các câu $\forall X(G)$, trong đó G là một công thức nào đó, được xác định là ngữ nghĩa của **hội của tất cả** các công thức nhận được từ công thức G bằng cách thay X bởi một đối tượng trong miền đối tượng.

Ví dụ: Nếu miền đối tượng gồm ba người {Lan, An, Hoa} thì ngữ nghĩa của câu $\forall X, \text{student}(X)$ được xác định là ngữ nghĩa của câu $\text{student(“Lan”) } \wedge \text{ student(“An”) } \wedge \text{ student(“Hoa”)}$. Câu này đúng khi và chỉ khi cả ba câu thành phần đều đúng, tức là cả Lan, Hoa, An đều là sinh viên.

Như vậy, công thức $\forall X (G)$ là đúng nếu và chỉ nếu mọi công thức nhận được từ G bằng cách thay X bởi một đối tượng bất kỳ trong miền đối tượng đều đúng, tức là G đúng cho tất cả đối tượng X trong miền đối tượng

Ngữ nghĩa của công thức $\exists X (G)$ được xác định như là ngữ nghĩa của **tuyển của tất cả** các công thức nhận được từ công thức G bằng cách thay X bởi một đối tượng trong miền đối tượng.

Ví dụ: Nếu ngữ nghĩa của câu $\text{younger}(X, 20)$ là “X trẻ hơn 20 tuổi” và miền đối tượng gồm ba người { Lan, Hoa, An} thì ngữ nghĩa của câu $\exists X \text{ younger}(X, 20)$ là ngữ nghĩa của câu $\text{younger(“Lan”, 20)} \vee \text{ younger(“Hoa”, 20)} \vee \text{ younger(“An”, 20)}$. Câu này nhận giá trị True nếu và chỉ nếu ít nhất một trong ba người Lan, Hoa, An trẻ hơn 20 tuổi.

Như vậy, công thức $\exists X(G)$ là *đúng nếu và chỉ nếu một trong các công thức* nhận được từ G bằng cách thay X bởi một đối tượng trong miền đối tượng là đúng.

Các công thức tương đương

Cũng như trong logic mệnh đề, ta nói hai công thức G và H tương đương ($G \equiv H$) nếu chúng cùng đúng hoặc cùng sai trong mọi minh họa. Ngoài các tương đương đã biết trong logic mệnh đề, trong logic vị từ còn có các **tương đương khác liên quan tới các lượng từ**. Giả sử G là một công thức, thì $G(X)$ nói rằng công thức G có chứa các xuất hiện của biến X . Khi đó công thức $G(Y)$ là công thức nhận được từ $G(X)$ bằng cách thay tất cả các xuất hiện của X bởi Y . Ta nói $G(Y)$ là công thức nhận được từ $G(X)$ bằng cách đặt tên lại (biến X đổi tên lại là Y).

Chúng ta có các tương đương sau đây:

$$1. \forall X (G(X)) \equiv \forall Y (G(Y))$$

$$\exists X (G(X)) \equiv \exists Y (G(Y))$$

Đặt tên lại biến đi sau lượng từ phổ dụng/tồn tại, ta nhận được công thức tương đương.

$$2. \neg (\forall X (G(X))) \equiv \exists X (\neg G(X))$$

$$\neg \exists X (G(X)) \equiv \forall X (\neg G(X))$$

$$3. \forall X (G(X) \wedge H(X)) \equiv \forall X (G(X)) \wedge \forall X (H(X))$$

$$\exists X (G(X) \vee H(X)) \equiv \exists X (G(X)) \vee \exists X (H(X)).$$

3. Phép tính vị từ bậc nhất (First order predicate calculus)

Phép tính vị từ bậc nhất cho phép các biến lượng giá $\forall X, p(X), \exists X, p(X)$ tham chiếu đến các đối tượng trong miền của vấn đề đang đề cập nhưng **KHÔNG** được tham chiếu đến các vị từ và hàm.

Ví dụ: Ví dụ không hợp lệ: $\forall (\text{Likes}) \text{ Likes}(\text{helen}, \text{ice-cream})$

Ví dụ hợp lệ:

+ Nếu ngày mai trời không mưa, Tom sẽ đi biển.

$$\neg \text{weather}(\text{rain}, \text{tomorrow}) \Rightarrow \text{go}(\text{tom}, \text{sea})$$

+ Tất cả các cầu thủ bóng rổ đều cao.

$$\forall X (\text{basketball_player}(X) \Rightarrow \text{tall}(X))$$

+ Có người thích coca-cola.

$$\exists X \text{ person}(X) \wedge \text{likes}(X, \text{coca-cola})$$

+ Không ai thích thuế.

$$\neg \exists X \text{ likes}(X, \text{taxes})$$

Hầu hết bất kỳ câu đúng ngữ pháp nào cũng có thể biểu diễn trong phép tính vị từ bậc nhất

bằng cách sử dụng các ký hiệu, các phép kết nối và ký hiệu biến.

Bài tập

1) Cho biểu thức vị từ có 3 đối số $\text{csg}(C, S, G)$ biểu diễn câu: “Sinh viên S nhận điểm G trong học phần C”. Hãy diễn giải $\text{csg}(\text{“Anhvan”}, \text{“An”}, 8)$.

2) Giả sử ta có vị từ $\text{loves}(X, Y)$ được diễn giải là: “X yêu Y”, như vậy các câu sau (biểu diễn trong logic vị từ) được hiểu như thế nào?

$$\forall X (\exists Y (\text{loves}(X, Y)))$$

$$\exists Y (\forall X (\text{loves}(X, Y)))$$

3) Giả sử ta có các vị từ:

$\text{dog}(X)$ (“X là chó”), $\text{cat}(Y)$ (“Y là mèo”), $\text{animal}(Z)$ (“Z là động vật”). Hãy biểu diễn câu sau trong logic vị từ: “chó, mèo đều là động vật”.

2.1.5.2. Chuẩn hoá các công thức

Từ các câu phân tử, bằng cách sử dụng các kết nối logic và các lượng từ, ta có thể tạo ra các câu phức hợp có cấu trúc rất phức tạp. Để dễ dàng cho việc lưu trữ các câu trong bộ nhớ và thuận lợi cho việc xây dựng các thủ tục suy diễn, chúng ta cần chuẩn hoá các câu bằng cách đưa chúng về dạng chuẩn tắc hội (hội của các câu tuyển).

Trong mục này, chúng ta sẽ trình bày thủ tục chuyển một câu phức hợp thành một câu ở dạng chuẩn tắc hội tương đương.

Thủ tục chuẩn hoá các công thức bao gồm các bước sau:

1. Loại bỏ các kéo theo, tương đương.

Để loại bỏ các kéo theo, thay công thức :

$$P \Rightarrow Q \text{ bởi công thức tương đương } \neg P \vee Q;$$

Thay $P \Leftrightarrow Q$ bởi $(\neg P \vee Q) \wedge (P \vee \neg Q)$

2. Chuyển các phủ định tới các phân tử

Điều này được thực hiện bằng cách thay công thức ở vế trái bởi công thức ở vế phải trong các tương đương sau đây:

$$\neg(\neg P) \equiv P$$

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$

$$\neg(\forall X (P)) \equiv \exists X (\neg P)$$

$$\neg(\exists X (P)) \equiv \forall X (\neg P)$$

3. Loại bỏ các lượng từ tồn tại

Giả sử $p(X,Y)$ là các vị từ có nghĩa rằng “Y lớn hơn X” trong miền các số. Khi đó công thức $\forall x (\exists y (P(x,y)))$ có nghĩa là “với mọi số X, tồn tại Y sao cho số Y lớn hơn X”. Ta có thể xem Y trong công thức đó là hàm của đối số X, chẳng hạn $f(X)$ và loại bỏ lượng từ $\exists Y$, công thức đang xét trở thành $\forall X (P(X,f(X)))$. Ví dụ: $f(X)=X+1$, khi đó $\forall X (P(X,f(X))) \equiv \forall X (P(X, X+1))$ nghĩa là với mọi giá trị của X thì X+1 lớn hơn X.

Một cách tổng quát, giả sử $\exists Y(G)$ là một công thức con của công thức đang xét và nằm trong miền tác dụng của các lượng từ $\forall X_1, \dots, \forall X_n$. Khi đó ta có thể xem Y là hàm của n biến $X_1, \dots, X_n$, chẳng hạn $f(X_1, \dots, X_n)$. Sau đó ta thay các xuất hiện của Y trong công thức G bởi hạng thức $f(X_1, \dots, X_n)$ và loại bỏ các lượng từ tồn tại. Các hàm f được đưa vào để loại bỏ các lượng từ tồn tại được gọi là hàm Scholem.

Ví dụ: Xét công thức sau

$$\forall X (\exists Y (P(X,Y)) \vee \forall U (\exists V (Q(a,V) \wedge \exists Y (\neg R(X,Y)))) \quad (1)$$

Công thức con $\exists Y (P(X,Y))$ nằm trong miền tác dụng của lượng từ $\forall X$, ta xem Y là hàm của X: $f(X)$. Các công thức con $\exists V (Q(a,V))$ và $\exists Y (\neg R(X,Y))$ nằm trong miền tác dụng của các lượng từ $\forall X, \forall U$ ta xem V là hàm $g(X,U)$ và Y là hàm $h(X,U)$ của hai biến X, U. Thay các xuất hiện của Y và V bởi các hàm tương ứng, sau đó loại bỏ các lượng từ tồn tại, từ công thức (1) ta nhận được công thức:

$$\forall X (P(X, f(X)) \vee \forall U (Q(a, g(X, U)) \wedge \neg R(X, h(X, U)))) \quad (2)$$

4. Loại bỏ các lượng từ phổ dụng

Sau bước 3 trong công thức chỉ còn lại các lượng từ phổ dụng và mọi xuất hiện của biến đều nằm trong miền tác dụng của các lượng từ phổ dụng. Ta có thể loại bỏ tất cả lượng từ phổ dụng, công thức (2) trở thành công thức:

$$P(X, f(X)) \vee Q(a, g(X, U)) \wedge \neg R(X, h(X, U)) \quad (3)$$

Cần chú ý rằng, sau khi được thực hiện bước này tất cả các biến trong công thức được xem là chịu tác dụng của các lượng từ phổ dụng.

5. Chuyển các tuyển tới các literal

Bước này được thực hiện bằng cách thay các công thức dạng: $P \vee (Q \wedge R)$ bởi $(P \vee Q) \wedge (P \vee R)$ và thay $(P \wedge Q) \vee R$ bởi $(P \vee R) \wedge (Q \vee R)$. Sau bước này công thức trở thành hội của các câu tuyển nghĩa là ta nhận được các công thức ở dạng chuẩn tắc hội. Chẳng hạn, công thức (3) được chuyển thành công thức sau:

$$(P(X, f(X)) \vee Q(a, g(X, U))) \wedge (P(X, f(X)) \vee \neg R(X, h(X, U))) \quad (4)$$

6. Loại bỏ các hội

Một câu hội là đúng nếu và chỉ nếu tất cả các thành phần của nó đều là đúng. Do đó công thức ở dạng chuẩn tắc hội tương đương với tập các thành phần. Chẳng hạn, công thức (4) tương đương với tập hai câu tuyển sau:

$$\begin{aligned} P(f(X)) \vee Q(a, g(X, U)) \\ P(f(X)) \vee \neg R(X, h(X, U)) \end{aligned} \quad (5)$$

7. Đặt tên lại các biến

Đặt tên lại các biến sao cho các biến trong các câu khác nhau có tên khác nhau, chẳng hạn hai câu (5) có hai biến cùng tên là X , ta cần đổi tên biến X trong câu hai thành Z , khi đó các câu (5) tương đương với các câu sau:

$$\begin{aligned} P(f(X)) \vee Q(a, g(X, U)) \\ P(f(Z)) \vee \neg R(Z, h(Z, U)) \end{aligned} \quad (5')$$

Như vậy, khi tri thức là một tập hợp nào đó các công thức trong logic vị từ, bằng cách áp dụng thủ tục trên ta nhận được cơ sở tri thức chỉ gồm các câu tuyển (tức là ta luôn luôn có thể xem mỗi câu trong cơ sở tri thức là tuyển của các

literal). Hoàn toàn tương tự như trong logic mệnh đề, mỗi câu tuyển có thể biểu diễn dưới dạng một kéo theo, vế trái của kéo theo là hội của các câu phân tử, còn vế phải là tuyển của các câu phân tử. Dạng câu này được gọi là câu Kowlski, một trường hợp quan trọng của câu Kowlski là câu Horn (luật if- then)

- **Chứng minh bằng luật phân giải trên logic vị từ**

a). Chứng minh diễn dịch: Định lý phân giải (trong mục logic mệnh đề) vẫn còn đúng trong logic vị từ cấp một là thỏa được hay không thỏa được. Nếu một tập câu là không thỏa được thì qua một số bước áp dụng luật phân giải sẽ sinh ra một câu rỗng (tức là dẫn tới mâu thuẫn).

Ví dụ: Giả sử CSTT gồm các câu tuyển sau:

$$\neg P(W) \vee Q(W) \quad (1)$$

$$P(X) \vee R(X) \quad (2)$$

$$\neg Q(Y) \vee S(Y) \quad (3)$$

$$\neg R(z) \vee S(z) \quad (4)$$

Sau đây chúng ta sẽ đưa ra một chứng minh của câu $S(a)$ từ CSTT trên bằng luật phân giải. Áp dụng luật phân giải cho câu (2) và (4) với phép thế $[X/a, Z/a]$, ta suy ra câu sau:

$$P(a) \vee S(a) \quad (5)$$

Áp dụng luật phân giải cho câu (1) và (3) với phép thế $[w/a, y/a]$ ta nhận được câu:

$$\neg P(a) \vee S(a) \quad (6)$$

Áp dụng luật phân giải cho câu (5) và (6), ta suy ra câu $S(a) \vee S(a)$. Câu này tương đương với câu $S(a)$.

Chứng minh bằng cách áp dụng các luật suy diễn để dẫn tới điều cần phải chứng minh (như chứng minh trên) được gọi là chứng minh diễn dịch (deduction proof).

b). Chứng minh bác bỏ: Để chứng minh câu H là hệ quả logic của tập các câu $\{G_1, G_2, \dots, G_n\}$ (các tiên đề), ta có thể áp dụng phương pháp chứng minh bác bỏ, tức là chứng minh tập câu $\{G_1, G_2, \dots, G_n, \neg H\}$ không thỏa được. Mặt khác, ở trên

ta đã chỉ ra rằng luật phân giải cho phép ta xác định được một tập câu là thỏa được hay không thỏa được. Vì vậy luật phân giải được xem là luật đầy đủ cho bác bỏ.

Ví dụ

Từ CSTT gồm các câu (1) đến (4) trong ví dụ trên, ta có thể chứng minh được câu S(a) bằng phương pháp bác bỏ như sau. Thêm vào CSTT câu:

$$\neg S(a) \quad (7)$$

(lấy phủ định của câu cần chứng minh), áp dụng luật phân giải cho câu (3) và (7) với phép thế $[y/a]$, ta suy ra câu:

$$\neg Q(a) \quad (8)$$

Từ câu (1) và (8) với phép thế $[w/a]$, ta nhận được câu:

$$\neg P(a) \quad (9)$$

Từ câu (4) và (7) với phép thế $[z/a]$, ta suy ra câu:

$$\neg R(a) \quad (10)$$

Từ câu (2) và (10) với phép thế $[x/a]$, ta nhận được câu:

$$P(a) \quad (11)$$

Áp dụng luật phân giải cho câu (9) và (11) ta nhận được câu rỗng (mâu thuẫn: $\neg P(a)$ và $P(a)$).

2.1.5.3. Bài tập

1) Gọi vị từ $nt(X)$ có nghĩa là “X là số nguyên tố” và vị từ $sl(X)$ có nghĩa là “X là số lẻ”.

Hãy diễn giải ý nghĩa của công thức: $\exists X (nt(X) \text{ and } sl(X))$.

Viết lại công thức trên sau khi lấy phủ định và diễn giải ý nghĩa của công thức đó.

2) Biến đổi công thức sau về dạng chuẩn tắc hội

$$\exists X \exists Y ((b(X) \wedge c(X)) \vee (d(Y) \wedge b(Y)))$$

3) Gọi $p(X,Y,Z)$ có nghĩa là: $Z=X*Y$, là một vị từ 3 biến trên tập số thực.

Khi đó tính chất giao hoán của phép nhân $X*Y=Y*X$ được diễn tả như sau:

$$\forall X,Y (p(X,Y,Z)) \Rightarrow p(Y,X,Z)$$

Hãy chuẩn hóa công thức trên (đưa về dạng chuẩn tắc hội)

Thí dụ: Cho trước: (1) $\forall X (\text{man}(X) \Rightarrow \text{mortal}(X))$

(2) $\text{man}(\text{socrates})$

$\Rightarrow$ (3) $\text{man}(\text{socrates}) \Rightarrow \text{mortal}(\text{socrates})$ từ (1), (2) bằng luật UI (Luật Phổ dụng).

(4) $\text{mortal}(\text{socrates})$ từ (3) và (2) bằng luật MP (Modus Ponent).

(1) $\forall X (\text{đanông}(X) \Rightarrow \text{uốngrượu}(X))$

(2) $\text{đanông}(\text{An})$

$\Rightarrow$ (3) $\text{đanông}(\text{An}) \Rightarrow \text{uốngrượu}(\text{An})$

(4) $\text{uốngrượu}(\text{An})$.

Nhóm bài tập Logic vị từ.

Bài tập 6. Biểu diễn các tri thức sau dưới dạng logic vị từ

- a. *Bất kỳ người nào cũng có cha mẹ*
- b. *Mọi số nguyên tố lớn hơn 2 đều là số lẻ*
- c. *Chuồn chuồn bay thấp thì mưa*

Lời giải

a. Ký hiệu $\text{NGUOI}(X)$: nghĩa là X là người

$\text{CHAME}(X, Y)$: X là cha mẹ của Y

$\forall X (\text{NGUOI}(X) \rightarrow \exists Y: \text{CHAME}(Y, X))$

b. Ký hiệu $P(X)$: X là số nguyên tố lớn hơn 2

$Q(X)$: X là số lẻ

$\forall X (P(X) \rightarrow Q(X))$

c. Ký hiệu $\text{BAY}(X, Y)$: con vật X bay với độ cao Y

TROIMUA : trời mưa

$\text{BAY}(\text{"chuồn chuồn"}, \text{"thấp"}) \rightarrow \text{TROIMUA}$

Bài tập 7. (Có dùng Phép thế để loại trừ Phổ dụng)

Giả sử chúng ta biết các thông tin sau đây:

- 1) Ông Ba nuôi một con chó
- 2) Hoặc ông Ba hoặc ông An đã giết con mèo Bibi
- 3) Mọi người nuôi chó đều yêu quý động vật

4) Ai yêu quý động vật cũng không giết động vật

5) Chó mèo đều là động vật

Ai đã giết Bibi?

Lời giải

- **Biểu diễn các thông tin trên dưới dạng logic vị từ cấp một như sau**

Để biểu diễn các tri thức trên trong logic vị từ cấp một, chúng ta cần sử dụng:

Các hằng: D, Ba, An, Bibi.

Các vị từ: Dog(x) (x là chó), Cat(y) (y là mèo), Nuoi(u,v) (u nuôi v), AnimalLover(u) (u là người yêu quý động vật), Kill(u,v) (u giết v), Animal(x) (x là động vật).

Sử dụng các hằng và các vị từ trên, chúng ta có thể chuyển các thông tin trên thành các câu trong logic vị từ cấp một như sau:

- 1) $Dog("D") \wedge Nuoi("Ba", "D")$
- 2) $Cat("Bibi") \wedge (Kill("Ba", "Bibi") \wedge Kill("An", "Bibi"))$
- 3) $\forall X (\forall Y (Dog(Y) \wedge Nuoi(X,Y)) \rightarrow AnimalLover(X))$
- 4) $\forall U (AnimalLover(U) \rightarrow (\forall V Animal(V) \rightarrow \neg Kill(U,V)))$
- 5) $\forall Z (Dog(Z) \rightarrow Animal(Z)) \wedge \forall W (Cat(W) \rightarrow Animal(W))$

- **Chuyển về dạng chuẩn và dùng phương pháp phân giải Robinson**

- 1) $Dog("D")$
- 2) $Nuoi("Ba", "D")$
- 3) $Cat("Bibi")$
- 4) $Kill("Ba", "Bibi") \wedge Kill("An", "Bibi")$
- 5) $\neg Dog(Y) \vee \neg Nuoi(X,Y) \vee AnimalLover(X)$
- 6) $\neg AnimalLover(U) \vee \neg Animal(V) \vee \neg Kill(U,V)$
- 7) $\neg Dog(Z) \vee Animal(Z)$
- 8) $\neg Cat(W) \vee Animal(W)$

Giả sử $\neg Kill(T, "Bibi")$ đúng

- 9) $\neg Kill(T, "Bibi")$

Từ câu (4) và câu (9) với phép thế $[T/An]$, ta nhận được câu:

10) Kill(“Ba”, “Bibi”)

Từ câu (6) và câu (10) với phép thế [u/Ba, v/Bibi], ta nhận được câu:

11) $\neg \text{AnimalLover}(\text{“Ba”}) + \neg \text{Animal}(\text{“Bibi”})$

Từ câu (3) và câu (8) với phép thế [w/Bibi], ta nhận được câu:

12) Animal(“Bibi”)

Từ câu (11) và câu (12), ta nhận được câu:

13) $\neg \text{AnimalLover}(\text{“Ba”})$

Từ câu (1) và câu (5), với phép thế [y/D] ta nhận được câu:

14) $\neg \text{Nuoi}(\text{X}, \text{“D”}) + \text{AnimalLover}(\text{X})$

Từ câu (2) và câu (14), với phép thế [x/Ba] ta nhận được câu:

15) AnimalLover(“Ba”)

Từ câu (13) và câu (15) ta suy ra câu rỗng (có sự mâu thuẫn).

Như vậy ông An đã giết con mèo Bibi.

Bài tập 8. Giả sử chúng ta biết các thông tin sau đây:

- a. Mọi người đều chết
- b. Mọi phụ nữ đều chết
- c. Thần thánh không chết
- d. Tất cả cả những người bệnh phải được điều trị
- e. Beatrice là phụ nữ
- f. Christel là phụ nữ
- g. Marta là phụ nữ
- h. Socrate là người
- i. Zeus là thần thánh
- k. Socrate bị bệnh

Dùng phương pháp phân giải Robinson để có thể suy ra được Socrate có được điều trị hay không?

Bài tập 9) Chuyển các câu sau đây thành câu trong logic vị từ:

- a) Tất cả các con mèo đều là động vật.
- b) Không có con chó nào là loài bò sát.

- c) Tất cả các nhà khoa học máy tính đều thích một hệ điều hành nào đó.
- d) Mọi đứa trẻ đều thích Coca-cola.
- e) Không có đứa trẻ nào thích ăn rau.
- f) Một số người thích kẹo, một số khác thì không.
- g) Không có sinh viên nào học mà thi rớt môn này.

Bài tập 10) Cho một vấn đề được phát biểu như sau:

- a) John thích mọi loại thức ăn.
 - b) Táo là thức ăn.
 - c) Gà là thức ăn.
 - d) Tất cả mọi thứ ăn được mà vẫn còn sống thì đó là thức ăn.
 - e) Bill ăn đậu phộng và Bill vẫn còn sống.
 - f) Sue ăn mọi thứ mà Bill ăn.
1. Hãy biểu diễn vấn đề trên theo logic vị từ bậc nhất.
 2. Chứng minh “John thích đậu phộng”

Bài tập 11). Câu chuyện dưới đây được lấy từ quyển sách Algorithms + Data structures =

Programs(Thuật toán + Cấu trúc dữ liệu = Chương trình) của N. Wirth (1976):

“Tôi cưới một góa phụ(W), bà ta có một cô con gái đã lớn (D). Cha tôi (F), người thường

xuyên đến thăm chúng tôi đã phải lòng cô con riêng của vợ tôi và cưới cô ta. Vì thế cha tôi con rể tôi và con ghẻ tôi mẹ tôi. Vài tháng sau đó, vợ tôi sinh một đứa con trai (S1), nó trở thành em rể của bố tôi, cũng như trở thành chú tôi. Vợ của bố tôi, tức là con ghẻ của tôi, cũng sinh một đứa con trai (S2).”

Sử dụng phép tính vị từ, hãy tạo ra một tập hợp các biểu thức biểu diễn hoàn cảnh trong câu

chuyện trên. Hãy đưa ra các biểu thức định nghĩa các quan hệ gia đình cơ bản như định nghĩa bố vợ chẳng hạn, và sử dụng modus ponens trên hệ này mình kết luận “Tôi cũng chính là ông tôi”.

2.2. BIỂU DIỄN TRI THỨC SỬ DỤNG LUẬT DẪN XUẤT (LUẬT SINH)

2.2.1. Khái niệm

*Đây là một kiểu biểu diễn tri thức có cấu trúc. Ý tưởng cơ bản là tri thức có thể được cấu trúc bằng một cặp: **điều kiện – hành động**: "NẾU *điều kiện* xảy ra THÌ *hành động* sẽ được thi hành".*

Ví dụ: NẾU đèn giao thông là đỏ THÌ bạn không được đi thẳng

Phương pháp biểu diễn tri thức nhờ logic (logic mệnh đề và logic vị từ) khá trực quan song chỉ phù hợp khi không có quá nhiều luật suy diễn.

Ngày nay, các luật sinh đã trở nên phổ biến và được áp dụng rộng rãi trong nhiều hệ thống trí tuệ nhân tạo khác nhau. Luật sinh có thể là một công cụ mô tả để giải quyết các vấn đề thực tế thay cho các kiểu phân tích vấn đề truyền thống. Trong trường hợp này, các luật được dùng như là những chỉ dẫn (tuy có thể không hoàn chỉnh) nhưng rất hữu ích để trợ giúp cho các quyết định trong quá trình tìm kiếm, từ đó làm giảm không gian tìm kiếm. Một ví dụ khác là luật sinh có thể được dùng để bắt chước hành vi của những chuyên gia. Theo cách này, luật sinh không chỉ đơn thuần là một kiểu biểu diễn tri thức trong máy tính mà là một kiểu biểu diễn các hành vi của con người.

2.2.2. Cấu trúc.

Một cách tổng quát luật sinh có dạng như sau :

$$P_1 \square P_2 \square \dots \square P_n \square Q$$

(Đây là một câu Horn dạng chuẩn)

Tùy vào các vấn đề đang quan tâm mà luật sinh có những ngữ nghĩa hay cấu tạo khác nhau :

Trong logic vị từ : $P_1, P_2, \dots, P_n, Q$ là những biểu thức logic.

Trong ngôn ngữ lập trình, mỗi một luật sinh là một câu lệnh. IF (P_1 AND P_2 AND .. AND P_n) THEN Q.

Để biểu diễn một tập luật sinh, người ta thường phải chỉ rõ hai thành phần chính sau:

(1) **Tập các sự kiện F (Facts)**

$$F = \{ f_1, f_2, \dots, f_n \}$$

(2) **Tập các quy tắc R (Rules)** áp dụng trên các sự kiện dạng như sau :

$$f_1 \wedge f_2 \wedge \dots \wedge f_i \square q$$

Trong đó, các f_i , q đều thuộc F

Một câu Horn dạng tổng quát:

$$p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q_1 \vee q_2 \vee \dots \vee q_m$$

Lưu ý:

Nếu có luật dạng: $p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q_1 \vee q_2 \vee \dots \vee q_m$ thì tương đương với **m** luật sau:

$$p_1 \wedge p_2 \wedge \dots \wedge p_n \wedge \neg q_2 \wedge \dots \wedge \neg q_m \Rightarrow q_1$$

$$p_1 \wedge p_2 \wedge \dots \wedge p_n \wedge \neg q_1 \wedge \neg q_3 \wedge \dots \wedge \neg q_m \Rightarrow q_2$$

$$p_1 \wedge p_2 \wedge \dots \wedge p_n \wedge \neg q_1 \wedge \dots \wedge \neg q_{m-1} \Rightarrow q_m$$

$$(p \Rightarrow q_1 \vee q_2 \equiv \neg p \vee (q_1 \vee q_2) \equiv \neg p \vee q_1 \vee q_2 \equiv \neg(p \wedge \neg q_2) \vee q_1 \equiv p \wedge \neg q_2 \Rightarrow q_1).$$

Tuy nhiên ta chỉ xét câu Horn dạng chuẩn ($m=1$)

- Nếu $n=0$, $m=1$: câu Horn có dạng $\Rightarrow q$: gọi là sự kiện (fact) q.
- Nếu $n>0$, $m=1$: câu Horn có dạng: $p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q$: gọi là luật (rule).

Trong các hệ chuyên gia, cơ sở tri thức gồm 2 phần:

Tập các sự kiện (facts) và Tập luật (rules).

Ví dụ

1) Ta có các luật về kinh nghiệm dự báo thời tiết:

"Chuồn chuồn bay thấp thì mưa, bay cao thì nắng, bay vừa thì râm"

a: chuồn chuồn bay thấp, b: chuồn chuồn bay cao, c: chuồn chuồn bay vừa

d: trời mưa, e: trời nắng, f: trời râm.

Khi đó ta có: **Sự kiện**: (a, b, c, d, e, f) và các **luật** sau:

Rules: $a \Rightarrow d$

$b \Rightarrow e$

$c \Rightarrow f$

2) Nhiều định lý trong toán học có thể biểu diễn bởi các luật, ví dụ:

Nếu tam giác có một góc bằng 60^0 và tam giác có hai cạnh bằng nhau thì tam giác đó là tam giác đều.

2.2.3. SUY DIỄN TRÊN LUẬT SẢN XUẤT

2.2.3.1. Khái niệm

Suy diễn là quá trình suy luận dựa vào các quy luật đã cho, thiết lập các thông tin mới từ các thông tin đã biết. Suy diễn sẽ sử dụng tập sự kiện làm tiên đề.

Các phương pháp suy diễn dần dần chuyển từ các giả thiết về các kết luận bằng cách thêm vào giả thiết những sự kiện đã được khẳng định đúng, dựa trên 2 phương thức:

- Modus ponens:
$$\frac{A, A \Rightarrow B}{B}$$

nghĩa là nếu A đúng và $A \Rightarrow B$ đúng thì B cũng đúng

- Modus tollens
$$\frac{\neg B, A \Rightarrow B}{\neg A}$$

nghĩa là nếu B sai và biết rằng $A \Rightarrow B$ đúng thì A cũng sai.

Trong quá trình suy diễn, ta cần quan tâm đến các vấn đề sau:

- Xây dựng tập luật, câu hỏi nào được chọn để người sử dụng trả lời
- Chọn quá trình tìm kiếm như thế nào
- Thông tin nhận được có ảnh hưởng đến quá trình tìm kiếm không?

Bài toán

Cho tập sự kiện $F = \{f_1, f_2, \dots, f_n\}$ và tập luật $R = \{r_1, r_2, \dots, r_m\}$. Chứng minh tập kết luận G đúng.

2.2.3.2. Các phương pháp suy diễn

Quá trình suy diễn trong hệ luật sản xuất bao gồm 2 phương pháp cơ bản: suy diễn tiến và suy diễn lùi.

a) Suy diễn tiến (lập luận tiến - forward reasoning – hướng từ Dữ liệu)

(Tu tưởng cơ bản của suy diễn tiến là áp dụng luật suy diễn Modus Ponens tổng quát)

Đây là quá trình suy diễn bắt đầu từ tập sự kiện đã biết, rút ra những sự kiện mới và cứ như vậy cho đến khi có được sự kiện cần chứng minh hoặc không có luật nào sinh ra các sự kiện mới.

- Phương pháp

Gọi T là tập các sự kiện tại thời điểm đang xét (khởi tạo tập $T=F$: tập sự kiện đúng ban đầu).

Xét các luật r_i có dạng: $p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q$ và $p_j \in T \quad \forall_j = \overline{1, n}$ nghĩa là $\text{left}(r_i) \in T$ thì

$$T(\text{mới}) = T(\text{cũ}) + \text{right}(r_i)$$

quá trình lặp lại cho đến khi $G \subset T$ hoặc không có luật nào sinh ra thêm sự kiện mới.

- Giải thuật

Procedure suydiendien;

Begin

$T := F$;

$S := \text{loc}(R, T)$; { S : là tập luật có dạng $p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q$ sao cho $p_j \in T \quad \forall_j = \overline{1, n}$ }

While $G \not\subset T$ and $S \neq \emptyset$ do

Begin

$r := \text{get}(S)$;

$T := T + \text{right}(r)$;

$R := R \setminus \{r\}$;

$S := \text{loc}(R, T)$;

End;

If $G \subset T$ then write (“thành công”)

Else write (“không thành công”);

End;

Ví dụ

1) Cho trước tập sự kiện $F = \{a, b\}$. Sử dụng các luật:

$$r_1: a \Rightarrow c$$

$$r_2: b \Rightarrow d$$

$$r_3: c \Rightarrow e$$

$$r_4: a \wedge d \Rightarrow e$$

$$r_5: b \wedge c \Rightarrow f$$

$$r_6: e \wedge f \Rightarrow g$$

Cần suy ra g

(Ký hiệu: r : là luật đang xét; T : tập sự kiện đúng tại thời điểm đang xét; S : tập các luật có dạng các mệnh đề ở vế trái thuộc T (chưa xét tại thời điểm đó); R là tập luật tại thời điểm đang xét)

r	T	S	R
	a, b	r_1, r_2	$r_1, r_2, r_3, r_4, r_5, r_6$
r_1	a, b, c	r_2, r_3, r_5	$r_2, \dots, r_6$
r_2	a, b, c, d	r_3, r_4, r_5	$r_3, \dots, r_6$
r_3	a, b, c, d, e	r_4, r_5	r_4, r_5, r_6
r_4	a, b, c, d, e	r_5	r_5, r_6
r_5	a, b, c, d, e, f	r_6	r_6
r_6	a, b, c, d, e, f, g		

$g \in T$ nên bài toán được chứng minh (g : true)

Chú ý

- Quá trình suy diễn tiến là quá trình xem xét các luật, với mỗi luật ta xét phần điều kiện (ở vế trái) tới phần kết luận (ở vế phải) và khi mà tất cả các điều kiện của luật đều thỏa mãn thì ta suy ra sự kiện trong phần kết luận. Chính vì lẽ đó mà có tên là suy diễn tiến.
- Trong mỗi bước của thủ tục, người ta xét một luật trong tập luật. So sánh mỗi điều kiện (ở vế trái) của tập luật với các sự kiện trong cơ sở sự kiện, nếu tất cả các điều kiện của luật được thỏa mãn thì sự kiện trong phần kết luận được xem là sự kiện được suy ra. Nếu sự kiện này là sự kiện mới (không có

trong bộ nhớ làm việc) thì nó được đưa vào bộ nhớ làm việc. Quá trình trên cứ lặp lại cho đến khi nào không có luật nào sinh ra sự kiện mới.

- Quá trình suy diễn tiến không định hướng tới giải quyết một vấn đề nào cả, không hướng tới tìm ra câu trả lời cho một câu hỏi nào cả. Suy diễn tiến chỉ là quá trình suy ra các sự kiện mới từ các sự kiện có trong bộ nhớ làm việc.

b) Suy diễn lùi (lập luận lùi - backward reason – hướng từ mục tiêu)

Phương pháp này bắt đầu từ **mục tiêu** mà chúng ta muốn giải quyết, khảo sát xem có thể dùng những luật hợp thức nào để đạt đến mục tiêu này, đồng thời xác định xem các điều kiện nào phải được thỏa mãn để có thể áp dụng được chúng. Các điều kiện này sẽ trở nên những mục tiêu mới, hay còn gọi là các mục tiêu phụ(subgoals) trong tiến trình tìm kiếm. Quá trình tìm kiếm cứ tiếp tục như thế, hoạt động theo chiều ngược, qua hết các đích phụ kế tiếp nhau cho đến khi gặp các sự kiện thực (hoặc tập sự kiện con) của bài toán. Phương pháp này sẽ tìm ra một chuỗi các bước đi, hay các luật, dẫn từ các dữ liệu đến một đích, mặc dù nó thực hiện theo thứ tự ngược lại.

(Nghĩa là: để đưa ra kết luận b , ta thử tìm tất cả các luật có dạng: $a_1 \wedge \dots \wedge a_n \Rightarrow b$, để có b , phải đưa ra các kết luận $a_1, \dots, a_n$. Quá trình xác định a_i cũng tương tự như đối với b , nếu đến một lúc nào đó phát hiện được rằng có một a_i nào đó không dẫn xuất được từ các giả thiết thì quay lui sang các luật sản xuất khác sinh ra b có dạng $b_1 \wedge \dots \wedge b_m \Rightarrow b$. Ngược lại, nếu mọi a_i đều dẫn xuất được giả thiết thì quá trình dẫn xuất ra b là đúng)

- Giải thuật

Gọi T là tập các sự kiện cần chứng minh tại thời điểm đang xét (khởi tạo $T = G$, G là tập kết luận).

$S(p) = \{r_i \in R \mid \text{right}(r_i) = p\}$ (là tập các luật trong R sao cho vế phải chứa p)

Procedure suydienlui (g);

Begin

```

T:= {g};
If  $T \subset F$  then write ('g đã được chứng minh ')
Else
    Begin
        p:=get(T);
        If  $S(p) = \{\}$  then write ('g không chứng minh được ')
        Else
            For  $r_i \in S(p)$  do
                Begin
                     $T := T \setminus \text{right}(r_i)$ ;
                     $T := T + \text{left}(r_i)$ ;
                    For  $l \in T \setminus F$  do suydiennlui(l);
                End;
            End;
    End;
End;

```

Ví dụ

1) Cho tập sự kiện $F = \{p, r\}$, và tập luật R:

$r_1) p \Rightarrow q$

$r_2) q \wedge r \Rightarrow s$

Chứng minh s

p	r	T	S(p)
s	r_2	s	r_2
q	r_1	q, r r, p	r_1

2.2.3.3. Nhận xét

- **Suy diễn tiến:**

Ưu điểm:

i) Làm việc tốt khi bài toán có bản chất là đi thu thập thông tin rồi thấy điều cần suy diễn

ii) Cho ra khối lượng lớn các thông tin từ một số thông tin ban đầu. Nó sinh ra nhiều thông tin mới.

iii) Suy diễn tiến là tiếp cận lý tưởng đối với các loại bài toán cần giải quyết các nhiệm vụ như lập kế hoạch, điều hành, điều khiển và diễn dịch.

Nhược điểm:

i) Không cảm nhận được rằng chỉ cần một vài thông tin quan trọng. Hệ thống hỏi các câu hỏi có thể hỏi mà không biết rằng chỉ một ít câu đã đi đến kết luận được.

ii) Hệ thống có thể hỏi cả câu hỏi không liên quan. Có thể các câu trả lời cũng quan trọng nhưng làm người dùng lúng túng khi phải trả lời các câu chẳng dính đến chủ đề.

- **Suy diễn lùi:**

Ưu điểm:

i) Phù hợp với bài toán đưa ra giả thuyết và liệu giả thuyết đó có đúng hay không?

ii) Tập trung vào đích đã cho. Nó tạo ra một loạt câu hỏi chỉ liên quan đến vấn đề đang xét, thuận tiện đối với người dùng.

iii) Khi suy diễn một điều gì từ thông tin đã biết, nó chỉ tìm trên một phần của cơ sở tri thức thích đáng đối với bài toán đang xét.

iv) Suy diễn lùi được đánh giá cao trong các bài toán như là chẩn đoán, dự đoán và tìm lỗi.

Nhược điểm:

Nhược điểm cơ bản của loại suy diễn này là nó thường tiếp theo dòng suy diễn thay vì đúng ra phải dừng ở đó mà sang nhánh khác.

- **Như vậy**, dựa vào các ưu và nhược điểm của từng loại suy diễn mà ta nên chọn kỹ thuật suy diễn nào để áp dụng vào bài toán. Trước tiên, ta xem xét các chuyên gia giải nó như thế nào?. Nếu cần thu thập dữ liệu rồi mới quyết định suy diễn cái gì thì ta chọn suy diễn tiến. còn nếu đã có giả thuyết và cần chứng minh cái đích này thì ta dùng suy diễn lùi.

Ví dụ Một bác sĩ có thể hiểu hàng trăm vấn đề có thể xảy ra với một cá nhân, nhưng vẫn phải tìm hiểu hiện trạng của bệnh nhân, lúc đó cần suy diễn tiến.

Ngược lại bác sĩ hầu như thấy được bệnh (ví dụ như viêm họng) thì ông ta dùng suy diễn lùi.

Bài toán mô phỏng: Cho hai bình rỗng X và Y có thể tích lần lượt là VX và VY, hãy dùng hai bình này để đo ra z lít nước ($z \leq \min(VX, VY)$). Bằng cách sử dụng các luật dẫn xuất (luật sinh) bài toán được giải quyết như sau:

Ví dụ với: $VX = 5$ và $VY = 7$ và $z = 4$.

Sau một thời gian tính toán, ta có thể sẽ đưa ra một quy trình đổ nước như :

- *Mức đầy bình 7 ($VY=7; VX=0$)*
- *Trút qua bình 5 cho đến khi 5 đầy. ($VX=5; VY=7-5=2$)*
- *Đổ hết nước trong bình 5 ($VX=0$)*
- *Đổ hết nước còn lại từ bình 7 sang bình 5 ($VX=2; VY=0$)*
- *Mức đầy bình 7 ($VY=7; VX=2$)*
- *Trút qua bình 5 cho đến khi bình 5 đầy. ($VY=7-3=4; VX=2+3=5$)*



Phần còn lại ở bình 7 chính là số nước cần đo.

Chúng ta hãy phát biểu lại bài toán một cách hình thức hơn.

Không làm mất tính tổng quát, ta luôn có thể giả sử rằng $VX < VY$.

Gọi lượng nước chứa trong bình X là x ($0 \leq x \leq VX$)

Gọi lượng nước chứa trong bình Y là y ($0 \leq y \leq VY$)

Như vậy, điều kiện kết thúc của bài toán sẽ là :

$x = z$ hoặc $y = z$

Điều kiện đầu của bài toán là : $x = 0$ và $y = 0$

Quá trình giải được thực hiện bằng cách xét lần lượt các luật sau, luật nào thỏa mãn thì sẽ được áp dụng. Lúc này, các luật chính là các "kinh nghiệm" hay tri thức mà ta đã chuyển giao cho máy tính. Sau khi áp dụng luật, trạng thái của bài toán sẽ thay đổi, ta lại tiếp tục xét các luật kế tiếp, nếu hết luật, quay trở lại luật đầu tiên. Quá trình tiếp diễn cho đến khi đạt được điều kiện kết thúc của bài toán.

Ba luật này được mô tả như sau :

(L1) Nếu bình X đầy thì đổ hết nước trong bình X đi.

(L2) Nếu bình Y rỗng thì đổ đầy nước vào bình Y.

(L3) Nếu bình X không đầy và bình Y không rỗng thì hãy trút nước từ bình Y sang bình X (cho đến khi bình X đầy hoặc bình Y hết nước).

Chuyển đổi cách giải này thành chương trình như sau :

...

$x := 0; y := 0;$

WHILE ($(x < z)$ AND $(y > 0)$) DO BEGIN

IF $(x = V_x)$ THEN $x := 0;$

IF $(y = 0)$ THEN $y := V_y;$

IF $(y > 0)$ THEN

BEGIN

$k := \min(V_x - x, y);$

$x := x + k;$

$y := y - k;$

END;

END;

...

Thử "chạy" chương trình trên với số liệu cụ thể là :

$V_x = 3, V_y = 4$ và $z = 2$

Ban đầu : $x = 0, y = 0$

Luật (L2) $\rightarrow x = 0, y = 4$

Luật (L3) $\rightarrow x = 3, y = 1$

Luật (L1) $\rightarrow x = 0, y = 1$

Luật (L3) $\rightarrow x = 1, y = 0$

Luật (L2) $\rightarrow x = 1, y = 4$

Luật (L3) $\rightarrow x = 3, y = 2$

3 luật mà chúng ta đã cài đặt trong chương trình ở trên được gọi là **cơ sở tri thức**. Còn cách thức tìm kiếm lời giải bằng cách tìm tuần tự từng luật và áp dụng nó được gọi là **cơ sở suy diễn**.

(Bài tập: Viết và thực hiện chương trình cho bài toán trên trong NNLT C)

2.2.4. Ví dụ và Bài tập.

Bài tập 1. Cho các biểu thức logic mệnh đề đúng sau:

1) ac

2) $ab \rightarrow f$

3) $(d + b)f \rightarrow i$

4) $\neg h + \neg a + f$

5) $fgh \rightarrow i$

6) $(\neg a + d + \neg c)$

7) $ad \rightarrow gh$

Chứng minh hoặc bác bỏ mệnh đề i bằng phương pháp suy diễn tiến và suy diễn lùi

Lời giải

- Biểu diễn các biểu thức đúng đã cho bằng luật sản xuất

(xác định tập luật, tập sự kiện ban đầu, tập sự kiện cần chứng minh)

Quá trình biến đổi

$$\begin{aligned} 3) \quad (d+b)f \rightarrow i &\equiv \neg((d+b)f) + i \equiv \neg(d+b) + \neg f + i \equiv (\neg d \neg b) + \neg f + i \equiv \\ &(\neg d + \neg f + i)(\neg b + \neg f + i) \equiv (df \rightarrow i)(bf \rightarrow i). \text{ Áp dụng Luật PP: } (q.r) + p = (q+p) \\ &(\mathbf{r+p}) \end{aligned}$$

$$4) \quad \neg h + \neg a + f \equiv \neg(ha) + f \equiv ha \rightarrow f$$

$$6) \quad (\neg a + d + \neg c) \equiv \neg(ac) + d \equiv ac \rightarrow d$$

$$7) \quad ad \rightarrow gh \equiv \neg(ad) + (gh) \equiv (\neg(ad) + g)(\neg(ad) + h) \equiv (ad \rightarrow g)(ad \rightarrow h)$$

Tập sự kiện $F=\{a, c\}$, tập sự kiện cần chứng minh $G=\{i\}$

Tập luật R:

- | | |
|---------------------------|--------------------------|
| $r_1) ab \rightarrow f$ | $r_5) fgh \rightarrow i$ |
| $r_2) (df \rightarrow i)$ | $r_6) ac \rightarrow d$ |
| $r_3) (bf \rightarrow i)$ | $r_7) ad \rightarrow g$ |
| $r_4) ha \rightarrow f$ | $r_8) ad \rightarrow h$ |

- Suy diễn tiến (tiến hành lập bảng sau)

r	T	S	R
	a, c	r_6	$r_1, r_2, r_3, r_4, r_5, r_6, r_7,$
r_6	a, c, d	r_7, r_8	r_8
r_7	a, c, d, g	r_8	$r_1, \dots, r_5, r_7, r_8$
r_8	a, c, d, g, h	r_4	$r_1, \dots, r_5, r_8$
r_4	a, c, d, g, h, f	r_2, r_5	$r_1, \dots, r_5$
r_2	a, c, d, g, h, f, i		r_1, r_2, r_3, r_5

(trong đó: r: là luật đang xét, T: tập sự kiện đúng tại thời điểm đang xét, S: tập các luật có dạng các mệnh đề ở vế trái thuộc T. R là tập luật tại thời điểm đang xét)

Vì $i \in T$ (là tập sự kiện đúng). Vậy i được chứng minh

- Suy diễn lùi (tiến hành lập bảng sau)

p	r	T	S(p)
		i	r_2, r_3, r_5
i	r_2	f, d	r_1, r_4
f	r_1	a, b	$\emptyset$
b			
quay lui			r_1, r_4
f	r_4	h, a	r_8
h	r_8	a, d	r_6
d	r_6	a, c	Kết thúc

--	--	--	--

Đường đi: **a,c đúng $\mapsto$ d đúng $\mapsto$ h đúng $\mapsto$ f đúng $\mapsto$ i đúng** theo chiều ngược lại.

Vậy i được chứng minh

Bài tập 2. Cho cơ sở tri thức được biểu diễn bằng các biểu thức logic đúng sau

- | | |
|----------------------------|------------------------|
| 1) $pt \rightarrow a$ | 5) $p \rightarrow t$ |
| 2) $qt \rightarrow s$ | 6) $apq \rightarrow c$ |
| 3) $pq \rightarrow b$ | 7) $bc \rightarrow t$ |
| 4) $\neg b \rightarrow st$ | 8) pq |

Biểu diễn tri thức đã cho dưới dạng luật sản xuất và dùng phương pháp suy diễn tiến và suy diễn lùi để chứng minh hoặc bác bỏ sự kiện $s \equiv 1$.

Bài tập 3. Cho cơ sở tri thức được biểu diễn bằng các biểu thức logic đúng sau

- 1) $(a+c)b \rightarrow f$
- 2) $\neg e + \neg f + a$
- 3) $gfh \rightarrow i$
- 4) $(e+ f)b \rightarrow gi$
- 5) $(\neg a+ e + \neg c)abc$

Dùng phương pháp suy diễn tiến và suy diễn lùi để chứng minh hoặc bác bỏ sự kiện $i \equiv 1$.

Bài tập 4.. Cho cơ sở tri thức được biểu diễn bằng các biểu thức logic đúng sau

- 1) efh
- 2) $\neg a + g + d$
- 3) $\neg h + c + d$
- 4) $af \rightarrow bg$
- 5) $ke \rightarrow d$
- 6) $(ef \rightarrow a)(\neg c + \neg e + \neg f)$

- Biểu diễn tri thức đã cho dưới dạng luật sản xuất

- Dùng phương pháp suy diễn tiên đề chứng minh sự kiện $d \equiv 1$ đúng.

Bài tập 5.. Trong một lớp học, có một nhóm học sinh gồm 10 bạn có tên lần lượt là: A, B, C, D, E, F, G, H, I và J. Giữa các bạn học sinh đó có mối quan hệ gọi là quan hệ ảnh hưởng. Ví dụ: nếu ta viết $AB > C$ thì có nghĩa là hai bạn đồng thời cùng thuyết phục bạn C tham gia một hoạt động nào đó. Giả sử ban đầu có bốn bạn E, F, H, I tham gia dự thi sản phẩm phần mềm do nhà trường tổ chức và ta cũng biết được rằng:

- 1) $ACH > B$ 2) $BH > ACD$ 3) $ABCI > BDI$ 4) $ADEI > BCG$
 5) $CGI > AJE$ 6) $H > BC$.

Hãy dùng phương pháp suy diễn tiên đề chứng minh rằng cả 10 bạn trong nhóm trên đều tham gia dự thi sản phẩm phần mềm.

2.3. BIỂU DIỄN TRI THỨC SỬ DỤNG MẠNG NGỮ NGHĨA

2.3.1. Khái niệm

Mạng ngữ nghĩa là phương pháp biểu diễn tri thức trực quan nhất.

Phương pháp này sẽ biểu diễn tri thức dưới dạng một đồ thị, trong đó đỉnh là các đối tượng (khái niệm) còn các cung là mối quan hệ giữa các đối tượng (khái niệm) này.

2.3.2. Cấu trúc

Mạng ngữ nghĩa cấu trúc như một đồ thị định hướng $G = (V, E)$ gồm tập đỉnh V và tập cung E :

- + Đỉnh chứa công thức hoặc thuộc tính (đối tượng)
- + Cung: là mối quan hệ giữa các đối tượng.

2.4. BIỂU DIỄN TRI THỨC BẰNG FRAME (KHUNG)

2.4.1. Khái niệm.

Khung thực chất là sự tổng quát hoá của cấu trúc bản ghi trong Pascal và tương tự như cấu trúc đối tượng trong C^{++} .

Khung là một cấu trúc dữ liệu chứa đựng tất cả những tri thức liên quan đến một đối tượng cụ thể nào đó. Frame "đóng gói" toàn bộ một đối tượng, tình huống hoặc

cả một vấn đề phức tạp thành một thực thể duy nhất có cấu trúc. Một frame bao hàm trong nó một khối lượng tương đối lớn tri thức về một đối tượng, sự kiện, vị trí, tình huống hoặc những yếu tố khác. Do đó, frame có thể giúp ta mô tả khá chi tiết một đối tượng.

2.4.2. Cấu trúc.

Một khung được mô tả bởi cấu trúc:

- Tên khung: Định danh đối tượng mô tả
- Các khe (slot): trên mỗi khe lưu trữ các thông tin, n\miền giá trị, thuộc tính và chiều mũi tên chỉ đến các khung khác. Mỗi slot có thể chứa một hoặc nhiều **facet** (mặt, giao diện). Các facet đặc tả một số thông tin hoặc thủ tục liên quan đến thuộc tính được mô tả bởi slot.

Chương 3. BIỂU DIỄN BÀI TOÁN TRONG KHÔNG GIAN TRẠNG THÁI

3.1. Đặt vấn đề.

Vấn đề tìm kiếm, một cách tổng quát, có thể hiểu là: *Tìm một đối tượng thỏa mãn một số đòi hỏi nào đó, trong một tập hợp rộng lớn các đối tượng.* Chúng ta có thể kể ra rất nhiều vấn đề mà việc giải quyết nó được quy về vấn đề tìm kiếm.

Các trò chơi: Cờ vua, cờ carô có thể xem như vấn đề tìm kiếm. Trong số rất nhiều nước đi được phép thực hiện, ta phải tìm ra các nước đi dẫn tới tình thế kết cuộc mà ta là người thắng cuộc.

Chúng minh định lý cũng có thể xem như vấn đề tìm kiếm. Cho một tập các tiên đề và các luật suy diễn, trong trường hợp này mục tiêu của ta là tìm ra một chứng minh (một dãy các luật suy diễn được áp dụng) để đưa đến được công thức mà ta cần chứng minh.

Trong các lĩnh vực nghiên cứu của **Trí Tuệ Nhân Tạo**, chúng ta thường xuyên phải đối đầu với vấn đề tìm kiếm. Đặc biệt trong lập kế hoạch và học máy, tìm kiếm đóng vai trò rất quan trọng.

Một khi chúng ta muốn giải quyết một vấn đề nào đó bằng tìm kiếm, đầu tiên ta phải xác định không gian tìm kiếm. Không gian tìm kiếm bao gồm tất cả các đối tượng mà ta cần tìm kiếm. Nó có thể là không gian liên tục, chẳng hạn không gian các vectơ thực n chiều; nó cũng có thể là không gian các đối tượng rời rạc.

Trong mục này, ta sẽ xét việc biểu diễn một vấn đề trong không gian trạng thái sao cho việc giải quyết vấn đề được quy về việc tìm kiếm trong không gian trạng thái đó.

Một phạm vi rộng lớn các vấn đề, đặc biệt các câu đố, các trò chơi, có thể mô tả bằng cách sử dụng khái niệm trạng thái và toán tử (phép biến đổi trạng thái).

Ví dụ: Một khách du lịch có trong tay bản đồ mạng lưới giao thông nối các thành phố trong một vùng lãnh thổ, du khách đang ở thành phố A và muốn tìm đường đi tới thành phố B. Trong bài toán này, các thành phố có trong các bản đồ là các trạng thái, thành phố A là trạng thái ban đầu, B là trạng thái kết thúc. Khi đang ở thành phố D anh ta có thể đi theo các con đường nối tới các thành phố C, F và G. Các con đường nối các thành phố sẽ được biểu diễn bởi các toán tử. Một toán tử thực hiện việc biến đổi một trạng thái thành một trạng thái khác. Chẳng hạn, ở trạng thái D sẽ có 3 toán tử dẫn trạng thái D tới các trạng thái C, F và G. Vấn đề của du khách bây giờ là tìm một dãy toán tử để đưa trạng thái ban đầu A tới trạng thái kết thúc B.

Một ví dụ khác: Trong trò chơi cờ vua, mỗi cách bố trí các quân trên bàn cờ là một trạng thái. Trạng thái ban đầu là sự sắp xếp các quân lúc bắt đầu cuộc chơi. Mỗi nước đi hợp lệ là một toán tử, nó biến đổi một cảnh huống trên bàn cờ thành một cảnh huống khác.

Phương pháp biểu diễn vấn đề phù hợp là sử dụng các khái niệm **trạng thái** (state) và **toán tử** (operator).

Phương pháp giải quyết vấn đề dựa trên khái niệm trạng thái và toán tử được gọi là cách tiếp cận giải quyết vấn đề nhờ **không gian trạng thái**.

Như vậy muốn biểu diễn một vấn đề trong không gian trạng thái, ta cần xác định các yếu tố sau:

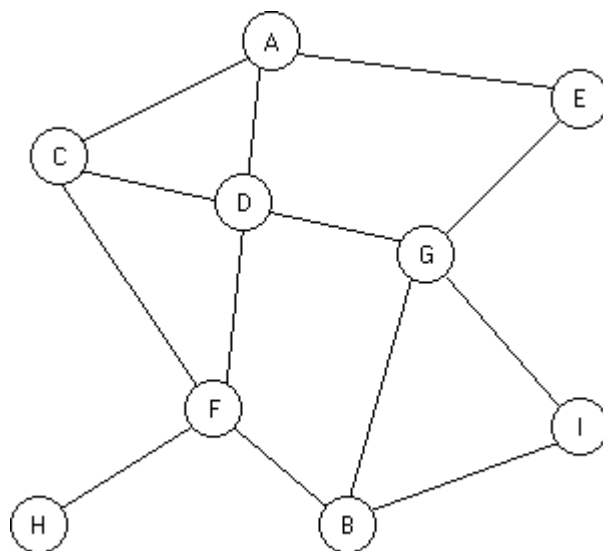
- Trạng thái ban đầu.
- Một tập hợp các toán tử. Trong đó mỗi toán tử mô tả một hành động hoặc một phép biến đổi có thể đưa một trạng thái tới một trạng thái khác.
- Không gian trạng thái: Tập hợp tất cả các trạng thái có thể đạt tới từ trạng thái ban đầu bằng cách áp dụng một dãy toán tử.

Ta sẽ ký hiệu không gian trạng thái là U , trạng thái ban đầu là u_0 ($u_0 \in U$). Mỗi toán tử R có thể xem như một ánh xạ $R: U \rightarrow U$. Nói chung R là một ánh xạ không xác định khắp nơi trên U .

- Trạng thái kết thúc: Một tập hợp T các trạng thái kết thúc (trạng thái đích). T là tập con của không gian U .

Trong vấn đề của du khách trên, chỉ có một trạng thái đích, đó là thành phố B. Nhưng trong nhiều vấn đề (chẳng hạn các loại cờ) có thể có nhiều trạng thái đích và ta không thể xác định trước được các trạng thái đích. Nói chung trong phần lớn các vấn đề, ta chỉ có thể mô tả các trạng thái đích là các trạng thái thỏa mãn một số điều kiện nào đó.

Khi chúng ta biểu diễn một vấn đề thông qua các trạng thái và các toán tử, thì việc tìm nghiệm của bài toán được quy về việc tìm đường đi từ trạng thái ban đầu tới trạng thái đích. (Một đường đi trong không gian trạng thái là một dãy toán tử dẫn một trạng thái tới một trạng thái khác).



Hình 1.1 Tìm đường đi từ A đến B

Chúng ta có thể biểu diễn không gian trạng thái bằng đồ thị định hướng, trong đó mỗi đỉnh của đồ thị tương ứng với một trạng thái. Nếu có toán tử R biến đổi trạng thái u thành trạng thái v , thì có cung gán nhãn R đi từ đỉnh u tới đỉnh v . Khi đó một đường đi trong không gian trạng thái sẽ là một đường đi trong đồ thị này.

3.2. Mô tả trạng thái

Giải bài toán trong không gian trạng thái, trước hết phải xác định dạng mô tả trạng thái bài toán sao cho bài toán trở nên đơn giản hơn, phù hợp bản chất vật lý của bài toán (Có thể sử dụng các xâu ký hiệu, vectơ, mảng hai chiều, cây, danh sách).

Mỗi trạng thái chính là mỗi hình trạng của bài toán, các hình trạng ban đầu và hình trạng cuối của bài toán gọi là trạng thái đầu và trạng thái cuối.

Ví dụ 1. Bài toán đóng nước

Cho 2 bình có dung tích lần lượt là m và n (lit). Với nguồn nước không hạn chế, dùng 2 bình trên để đóng k lit nước. Không mất tính tổng quát có thể giả thiết $k \leq \min(m, n)$.

Tại mỗi thời điểm xác định, lượng nước hiện có trong mỗi bình phản ánh bản chất hình trạng của bài toán ở thời điểm đó.

- Gọi x là lượng nước hiện có trong bình dung tích m và y là lượng nước hiện có trong bình dung tích n . Như vậy bộ có thứ tự (x,y) có thể xem là trạng thái của bài toán. Với cách mô tả như vậy, các trạng thái đặc biệt của bài toán sẽ là:

- Trạng thái đầu: $(0,0)$
- Trạng thái cuối: (x,k) hoặc (k,y) , $0 \leq x \leq m$, $0 \leq y \leq n$

Ví dụ 2. Bài toán trò chơi 8 số

Trong bảng ô vuông 3 hàng, 3 cột, mỗi ô chứa một số nằm trong phạm vi từ 1 đến 8 sao cho không có 2 ô có cùng giá trị, có một ô trong bảng bị trống (không chứa giá trị nào cả). Xuất phát từ một sắp xếp nào đó các số trong bảng, hãy dịch chuyển ô trống sang phải, sang trái, lên trên hoặc xuống dưới (nếu có thể được) để đưa từ bảng ban đầu về bảng qui ước trước.

Ví dụ: Hình 1 là bảng xuất phát và Hình 2. là bảng mà ta phải thực hiện các bước di chuyển ô trống để đạt được.

2	8	3
1	6	4
7		5

Hình 1.

1	2	3
8		4
7	6	5

Hình 2.

Giá trị các phần tử trong bảng xác định trạng thái bài toán. Vì vậy có thể mô tả trạng thái của bài toán bằng một ma trận $A_{3 \times 3} = (a_{ij})$, $a_{ij} \in \{0..8\}$, $a_{ij} \neq a_{kl}$, $\forall i \neq k, j \neq l$

- Trạng thái đầu của bài toán là ma trận

$$\begin{pmatrix} 2 & 8 & 3 \\ 1 & 6 & 4 \\ 7 & 0 & 5 \end{pmatrix}$$

- Trạng thái cuối là ma trận

$$\begin{pmatrix} 1 & 2 & 3 \\ 8 & 0 & 4 \\ 7 & 6 & 5 \end{pmatrix}$$

Có thể phát biểu dạng tổng quát của bài toán này (Trò chơi n^2-1 số)

Ví dụ 3. Bài toán tháp Hà Nội

Cho ba cọc 1, 2, 3. Ở cọc 1 ban đầu có n đĩa sắp xếp theo thứ tự to dần từ dưới lên trên. Hãy dịch chuyển n đĩa đó sang cọc thứ 3 sao cho:

- Mỗi lần chỉ chuyển một đĩa.
- Trong mỗi cọc không cho phép đĩa to nằm trên đĩa nhỏ hơn.

Bài toán xác định khi biết được từng đĩa đang nằm ở cọc nào. Hay nói cách khác, có hai cách xác định:

1- Cọc 1 hiện đang chứa những đĩa nào? Cọc 2 hiện đang chứa những đĩa nào? Và cọc 3 đang chứa những đĩa nào.

2- Đĩa lớn thứ i hiện đang nằm ở cọc nào? ($i = 1 \dots n$)

Như vậy cách mô tả trạng thái bài toán không duy nhất, vấn đề là chọn cách mô tả nào để đạt được mục đích dễ dàng nhất.

Theo trên, với cách thứ nhất ta phải dùng 3 danh sách động vì số đĩa trên mỗi cọc là khác nhau trong từng thời điểm khác nhau.

Cách thứ hai, nhìn qua thì khó mô tả nhưng dựa vào khái niệm về bộ có thứ tự trong toán học, cách này mô tả bài toán hiệu quả hơn. Thật vậy, nếu gọi x_i là cọc chứa đĩa lớn thứ i , trong đó $x_i \in \{1, 2, 3\}$, $i \in \{1 \dots n\}$. Khi đó bộ có thứ tự $(x_1, x_2, \dots, x_n)$ có thể dùng làm dạng mô tả trạng thái đang xét của bài toán. Với cách mô tả này thì : Trạng thái đầu là $(1, 1, \dots, 1)$; Trạng thái cuối là $(3, 3, \dots, 3)$.

3.3. Toán tử chuyển trạng thái.

Toán tử chuyển trạng thái thực chất là các phép biến đổi đưa từ trạng thái này sang trạng thái khác. Có hai cách dùng để biểu diễn các toán tử:

- Biểu diễn như một hàm xác định trên tập các trạng thái và nhận giá trị cũng trong tập này.
- Biểu diễn dưới dạng các quy tắc sản xuất : $S \rightarrow A$ có nghĩa là nếu có trạng thái S thì có thể đưa đến trạng thái A .

Ví dụ 1. Bài toán đong nước

Các thao tác sử dụng để chuyển trạng thái này sang trạng thái khác gồm:

- + Đổ đầy một bình;
- + Đổ hết nước trong một bình (nếu đầy) ra ngoài;
- + Đổ nước từ bình này sang bình khác.

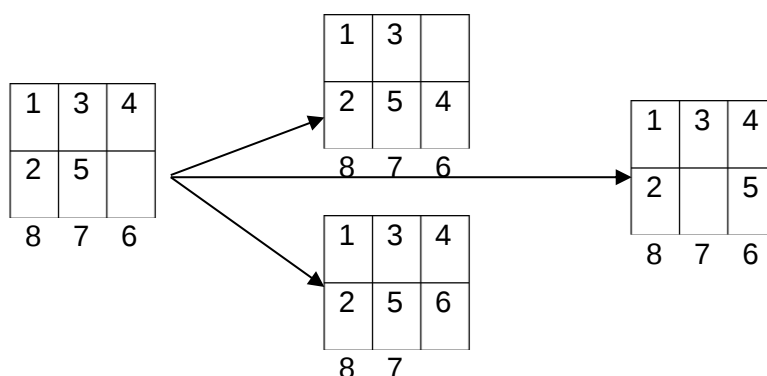
Như vậy, nếu trạng thái đang xét là (x,y) thì các trạng thái kế tiếp có thể chuyển đến sẽ là:

$$(x,y) \longrightarrow \left\{ \begin{array}{l} (m,y) \\ (x,n) \\ (0,y) \\ (x,0) \\ (0, x+y) \text{ nếu } x+y \leq n \\ (x+y -n,n) \text{ nếu } x+y > n \\ (x+ y,0) \text{ nếu } x+y \leq m \\ (m, x+y-m) \text{ nếu } x+y > m \end{array} \right.$$

Ví dụ 2. Trò chơi 8 số

Các thao tác để chuyển trạng thái tương ứng với việc chuyển ô trống sang phải, sang trái, lên, xuống nếu có thể được.

- Biểu diễn theo quy tắc sản xuất:



- Biểu diễn theo một hàm

Gọi hàm f_u là hàm biểu diễn cho toán tử chuyển ô trống lên trên; gọi B ($B = (b_{ij})$) là trạng thái sau khi di chuyển ô trống ở trạng thái A ($A = (a_{ij})$) lên trên, nghĩa là: $B = f_u(A)$, giả sử ô trống đang ở vị trí (i_0, j_0) (hay nói cách khác $a_{i_0 j_0} = 0$) thì hàm f được xác định như sau:

$$f_u(a_{ij}) = \begin{cases} a_{ij} & \forall (i, j) \quad \text{nếu } i_0 = 1 \\ a_{ij} & \text{nếu } (i, j) \neq (i_0-1, j_0) \text{ và } (i, j) \neq (i_0, j_0) \text{ và } i_0 > 1 \\ a_{i_0-1, j_0} & \text{nếu } (i, j) = (i_0, j_0), i_0 > 1 \\ a_{i_0, j_0} & \text{nếu } (i, j) = (i_0-1, j_0), i_0 > 1 \end{cases}$$

Tương tự, có thể xác định các phép chuyển ô trống xuống dưới f_d , qua trái f_l , qua phải f_r như sau:

$$f_d(a_{ij}) = \begin{cases} a_{ij} & \forall (i, j) \quad \text{nếu } i_0 = 3 \\ a_{ij} & \text{nếu } (i, j) \neq (i_0+1, j_0) \text{ và } (i, j) \neq (i_0, j_0) \text{ và } i_0 < 3 \\ a_{i_0-1, j_0} & \text{nếu } (i, j) = (i_0, j_0), i_0 < 3 \\ a_{i_0, j_0} & \text{nếu } (i, j) = (i_0+1, j_0), i_0 < 3 \end{cases}$$

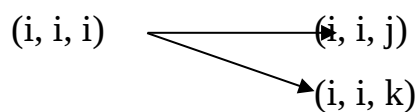
$$f_l(a_{ij}) = \begin{cases} a_{ij} & \forall (i, j) \quad \text{nếu } j_0 = 1 \\ a_{ij} & \text{nếu } (i, j) \neq (i_0, j_0-1) \text{ và } (i, j) \neq (i_0, j_0) \text{ và } j_0 > 1 \\ a_{i_0-1, j_0} & \text{nếu } (i, j) = (i_0, j_0), j_0 > 1 \\ a_{i_0, j_0} & \text{nếu } (i, j) = (i_0, j_0-1), j_0 > 1 \end{cases}$$

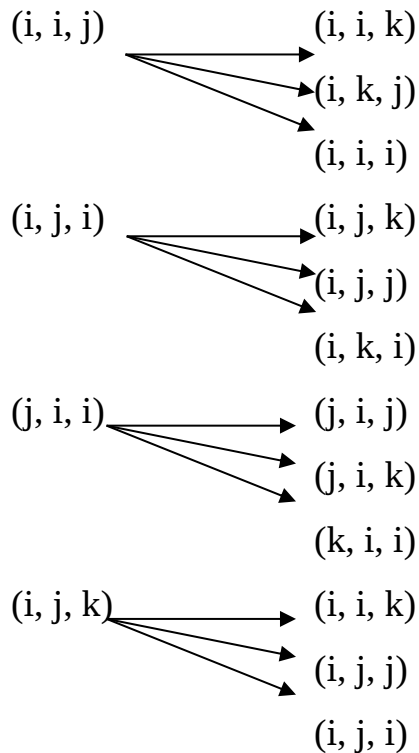
$$f_r(a_{ij}) = \begin{cases} a_{ij} & \forall (i, j) \quad \text{nếu } j_0 = 3 \\ a_{ij} & \text{nếu } (i, j) \neq (i_0, j_0+1) \text{ và } (i, j) \neq (i_0, j_0) \text{ và } j_0 < 3 \\ a_{i_0-1, j_0} & \text{nếu } (i, j) = (i_0, j_0), j_0 < 3 \\ a_{i_0, j_0} & \text{nếu } (i, j) = (i_0, j_0+1), j_0 < 3 \end{cases}$$

Ví dụ 3. Bài toán Tháp Hà Nội với $n=3$.

Mỗi trạng thái là một bộ ba (i, j, k) . Có các trường hợp như sau:

- Ba đĩa cùng nằm trên một cọc: (i, i, i)
- Hai đĩa cùng nằm trên một cọc: $(i, i, j), (i, j, i), (j, i, i)$
- Ba đĩa nằm trên ba cọc phân biệt: (i, j, k)





3.4. Không gian trạng thái của bài toán.

Không gian trạng thái là tập tất cả các trạng thái có thể có và tập các toán tử của bài toán.

Không gian trạng thái là một bộ bốn. Ký hiệu: $K = (T, S, G, F)$. Trong đó,

T: Tập tất cả các trạng thái có thể có của bài toán

S: Trạng thái đầu

G: Tập các trạng thái đích

F: Tập các toán tử

Ví dụ 1. Không gian trạng thái của bài toán đong nước là bộ bốn (T, S, G, F) xác định như sau:

$$T = \{ (x, y) \mid 0 \leq x \leq m; 0 \leq y \leq n \}$$

$$S = (0, 0)$$

$$G = \{ (x, k) \text{ hoặc } (k, y) \mid 0 \leq x \leq m; 0 \leq y \leq n \}$$

$$F = \{ \text{Đong đầy; Đổ hết ra; Đổ sang bình khác (thực hiện trên một bình)} \}.$$

Ví dụ 2. Không gian trạng thái của bài toán Tháp Hà nội với $n = 3$:

$$T = \{ (x_1, x_2, x_3) \mid x_i \in \{1, 2, 3\} \}$$

$$S = (1, 1, 1)$$

$$G = \{(3, 3, 3)\}$$

F = Tập các khả năng có thể chuyển đĩa đã xác định trong phần trước.

Ví dụ 3. Không gian trạng thái của bài toán trò chơi 8 số:

$$T = \{ (a_{ij})_{3 \times 3} \mid 0 \leq a_{ij} \leq 8 \text{ và } a_{ij} \neq a_{kl} \text{ với } i \neq j \text{ hoặc } k \neq l \}$$

S = Ma trận xuất phát của bài toán,

G = Ma trận cuối cùng của bài toán (các số nằm theo vị trí yêu cầu)

$$F = \{f_l, f_r, f_u, f_d\}$$

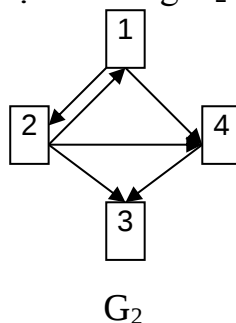
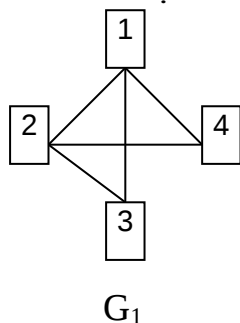
Tìm kiếm lời giải trong không gian trạng thái là quá trình tìm kiếm xuất phát từ trạng thái ban đầu, dựa vào toán tử chuyển trạng thái để xác định các trạng thái tiếp theo cho đến khi gặp được trạng thái đích.

3.5. Biểu diễn không gian trạng thái dưới dạng đồ thị

3.5.1. Các khái niệm

- Đồ thị $G = (V, E)$ trong đó V : tập đỉnh, E : tập cung ($E \subset V \times V$)
- G là đồ thị vô hướng thì (i, j) là một cạnh cũng như là (j, i) (tức là: $(i, j) \in E$ thì $(j, i) \in E$)
- Nếu G là đồ thị có hướng thì cung (i, j) hoàn toàn khác với cung (j, i) .

Ví dụ xét đồ thị vô hướng G_1 và đồ thị có hướng G_2



- Tập đỉnh kề:

$\forall n \in V, T(n) = \{m \in V \mid (n, m) \in E\}$ được gọi là tập các đỉnh kề của n

- Đường đi:

$p = (n_1, \dots, n_k)$ được gọi là đường đi từ đỉnh $n_1 \rightarrow n_k$ nếu $n_i \in V, \forall i=1, k$;

$$(n_i, n_{i+1}) \in E \quad \forall i=1, k-1$$

- Cây là đồ thị có đỉnh gốc $n_0 \in V$ thỏa mãn:

Một đồ thị $G=(V, E)$ gọi là cây nếu tồn tại một đỉnh $n_0 \in V$ có những tính chất sau:

- + $\forall n \in V, n \in T(n_0)$, trong đó $T(n_0)$: tập các đỉnh thuộc dòng dõi của n_0 (n_0 là tổ tiên của n)
- + $\forall n \in V, \exists! m \in V$ sao cho $n \in T(m)$; m được gọi là cha của n .

3.5.2. Biểu diễn không gian trạng thái bằng đồ thị

Theo ngôn ngữ đồ thị, không gian trạng thái tương ứng với một đồ thị định hướng trong đó: Các trạng thái tương ứng với các đỉnh trong đồ thị, nếu tồn tại toán tử chuyển trạng thái thì có cung (s, t)

Để thấy rõ mối tương quan, ta có bảng sau

KGTT	Đồ thị
Trạng thái	Đỉnh
Toán tử	Cung
Dãy các trạng thái liên tiếp	Đường đi

3.5.3. Biểu diễn đồ thị

Cho đồ thị $G = (V, E)$, giả sử $V = \{1, 2, \dots, n\}$. Có hai cách thường dùng để biểu diễn đồ thị G lưu trữ trong máy tính.

i) Biểu diễn bằng ma trận kề

Đồ thị G được biểu diễn bởi ma trận kề $A = (a_{ij})_{n \times n}$, với n là số đỉnh của đồ thị, trong đó:

$$a_{ij} = \begin{cases} 1 & \text{nếu } (i, j) \in E \\ 0 & \text{trong trường hợp ngược lại} \end{cases}$$

Nếu G là đồ thị vô hướng thì ma trận kề A là ma trận đối xứng.

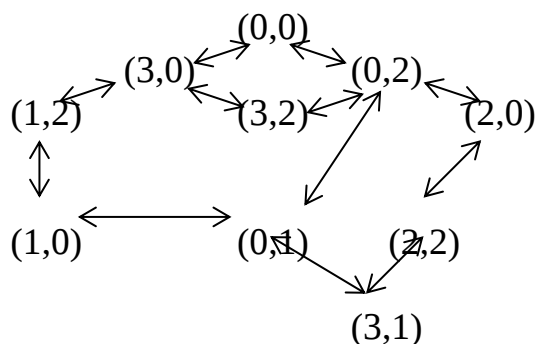
Ví dụ Với đồ thị vô hướng G_1 và đồ thị có hướng G_2 ở trên ta có các ma trận kề sau:

G1					G2			
0	1	1	1		0	1	0	1
1	0	1	1		1	0	1	1
1	1	0	0		0	0	0	0
1	1	0	0		0	0	1	0

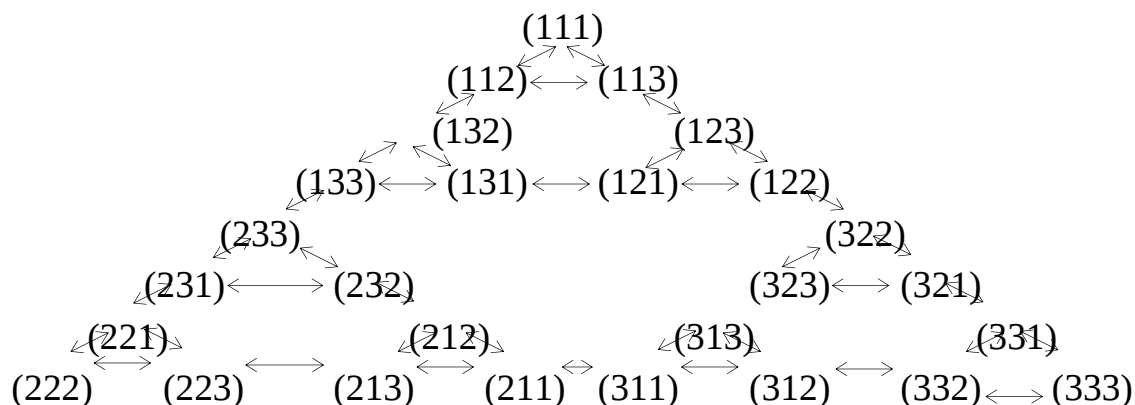
ii) Biểu diễn bằng danh sách kề.

Với mỗi đỉnh i của đồ thị, ta có một danh sách tất cả các đỉnh kề với i , ta ký hiệu là $List(i)$. Để thể hiện $List(i)$ ta có thể dùng mảng, kiểu tập hợp hay kiểu con trỏ. Ví dụ với đồ thị $G1$, ta có $List(1) = [2, 3, 4]$

Ví dụ 1. Bài toán đong nước $m=3, n=2, k=1$



Ví dụ 2. Tháp Hà Nội với $n = 3$



3.6. BÀI TẬP

Xây dựng không gian trạng thái đối với các bài toán sau:

1. Cho n thành phố đánh số từ 1 đến n . Giao thông đường bộ giữa hai thành phố i và j được cho bởi giá trị a_{ij} như sau: $a_{ij} = -1$ có nghĩa là không có đường bộ đi từ

thành phố i sang thành phố j và $a_{ij} = 1$ nếu có đường đi trực tiếp từ thành phố i sang thành phố j . Tìm đường đi từ thành phố i_0 sang thành phố j_0 .

2. Cho k và n là 2 số nguyên dương. Có 2^k viên sỏi, được phân bố trong n đồng, đồng thứ nhất có a_1 viên, đồng thứ 2 có a_2 viên, ..., đồng thứ n có a_n viên và tất nhiên $a_1 + a_2 + \dots + a_n = 2^k$. Người ta cần san sỏi từ các đồng để dồn sỏi trở về 1 đồng. Quy tắc san sỏi như sau: mỗi lần san áp dụng cho 2 đồng sỏi, giả sử 1 đồng có a viên và đồng kia có b viên (không giảm tổng quát, có thể giả thiết $a \geq b$) thì san sỏi từ đồng có a viên sang đồng có b viên để thành một đồng có $a-b$ viên và đồng kia $2*b$ viên.

Hướng dẫn: Trạng thái của bài toán phải xác định được số sỏi hiện có trong mỗi đồng.

3. Một dãy các số nguyên dương $a_1, a_2, \dots, a_n$ được gọi là hợp lý nếu thỏa mãn hai điều kiện:

- a_n là số nguyên tố.
- $a_{i+1} = a_i + 1$ hoặc $2*a_i$

Cho trước số a_1 , hãy tìm dãy hợp lý $a_1, a_2, \dots, a_n$.

4. Bài toán người đưa hàng.

Người đưa hàng cần phải xác định được hành trình ngắn nhất sao cho mỗi thành phố đi đến đúng một lần và quay trở lại thành phố xuất phát. Giả sử thành phố xuất phát là thành phố 1, có tất cả n thành phố đánh số từ 1 đến n .

Hướng dẫn:

- Mỗi trạng thái cho bởi danh sách các thành phố đã đi qua cho đến thời điểm hiện tại, trong đó không cho phép một thành phố nào được xuất hiện nhiều hơn một lần trừ thành phố 1 sau khi đã liệt kê tất cả các thành phố còn lại.
- Các toán tử tương ứng với các hành động
 - 1- đi tới thành phố 1
 - 2- đi tới thành phố 2

3- đi tới thành phố 3.

...

5. Bài toán phân tích cú pháp.

Văn phạm G là bộ bốn $G = (N, T, P, S)$, N là tập ký hiệu không kết thúc, T là tập ký hiệu kết thúc, $S \in N$ là ký hiệu đầu và P là tập sản xuất có dạng $\alpha \rightarrow \beta$, ở đây $\alpha, \beta \in (NUT)$. Ngôn ngữ sinh ra bởi văn phạm G được định nghĩa bởi:

$L(G) = \{ \omega \in T \mid S \Rightarrow \omega \}$, $S \Rightarrow \omega$ có nghĩa là $\exists \omega_1, \dots, \omega_n \in (NUT)$ sao cho $\omega_i \Rightarrow \omega_{i+1}$, $\omega_1 = S$ và $\omega_n = \omega$.

Bài toán phân tích cú pháp được phát biểu :Cho trước văn phạm G với xâu $\omega \in T$ đã cho. Hãy xác định xem $\omega \in L(G)$ hay không ?

Chương 4. CÁC PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI TRONG KHÔNG GIAN TRẠNG THÁI

Quá trình tìm kiếm lời giải của bài toán được biểu diễn trong không gian trạng thái được xem như quá trình dò tìm trên đồ thị, xuất phát từ trạng thái ban đầu, thông qua các toán tử chuyển trạng thái, lần lượt đến các trạng thái tiếp theo cho đến khi gặp được trạng thái đích hoặc không còn trạng thái nào có thể tiếp tục được nữa. Khi áp dụng các phương pháp tìm kiếm trong không gian trạng thái , người ta thường quan tâm đến các vấn đề sau:

- Kỹ thuật tìm kiếm lời giải
- Phương pháp luận của việc tìm kiếm
- Cách thể hiện các nút trong quá trình tìm kiếm (mô tả trạng thái bài toán)
- Việc chọn các toán tử chuyển trạng thái nào để áp dụng và có khả năng sử dụng phương pháp may rủi trong quá trình tìm kiếm.

Tuy nhiên, không phải các phương pháp này có thể áp dụng cho tất cả các bài toán phức tạp mà cho từng lớp bài toán. Việc chọn chiến lược tìm kiếm cho bài toán cụ thể phụ thuộc nhiều vào các đặc trưng của bài toán.

Các thủ tục tìm kiếm điển hình bao gồm:

- Tìm kiếm theo chiều rộng (Breadth – First Search)
- Tìm kiếm theo chiều sâu (Depth – First Search)
- Tìm kiếm leo đồi (Hill-climbing search – A*)
- Tìm kiếm cực tiểu (AT và ATK).
- Tìm kiếm với tri thức bổ sung (Heuristic Search – ATK và A*).
- Tìm kiếm có ràng buộc và phương pháp sinh – thử.

Các thủ tục tìm kiếm nói trên đều sử dụng các nguyên lý cơ sở sau:

♦ **Nguyên lý vết cạn thông minh:** Trong một bài toán tìm kiếm nào đó, khi không gian tìm kiếm lớn, ta thường tìm cách giới hạn lại không gian tìm kiếm hoặc thực hiện một kiểu dò tìm đặc biệt dựa vào đặc thù của bài toán để nhanh chóng tìm ra mục tiêu.

♦ **Nguyên lý tham lam (Greedy):** Lấy tiêu chuẩn tối ưu (trên phạm vi toàn cục) của bài toán để làm tiêu chuẩn chọn lựa hành động cho phạm vi cục bộ của từng bước (hay từng giai đoạn) trong quá trình tìm kiếm lời giải.

♦ **Nguyên lý thứ tự:** Thực hiện hành động dựa trên một cấu trúc thứ tự hợp lý của không gian khảo sát nhằm đạt được một lời giải tốt.

♦ **Hàm Heuristic:** Trong việc xây dựng các thuật giải Heuristic, người ta thường dùng các hàm Heuristic. Đó là các hàm đánh giá thô, giá trị của hàm phụ thuộc vào trạng thái hiện tại của bài toán tại mỗi bước giải. Nhờ giá trị này, ta có thể chọn được cách hành động tương đối hợp lý trong từng bước của thuật giải.

Cấu trúc chung của bài toán tìm kiếm

Cho trước hai trạng thái T_0 và T_G . Hãy xây dựng chuỗi trạng thái $T_0, T_1, T_2, \dots, T_{n-1}, T_n = T_G$ sao cho:

$$\sum_1^* \text{cost}(T_{i-1}, T_i) \text{ thỏa mãn một điều kiện cho trước (thường là nhỏ nhất).}$$

Trong đó:

- $T_i \in U$ (gọi là không gian trạng thái – state space);

- $\text{cost}(T_{i-1}, T_i)$ là chi phí để biến đổi từ trạng thái T_{i-1} sang trạng thái T_i .

Đa số các bài toán thuộc dạng mà chúng ta đang mô tả đều có thể được biểu diễn dưới dạng đồ thị. Trong đó, một trạng thái là một đỉnh của đồ thị.

Giả sử dùng danh sách L để lưu các trạng thái đã được sinh ra và chờ được phát triển. Mục tiêu của tìm kiếm trong không gian trạng thái là tìm đường đi từ trạng

thái ban đầu tới trạng thái đích, do đó ta cần lưu lại vết của đường đi. Ta có thể sử dụng hàm **father** để lưu lại cha của mỗi đỉnh trên đường đi.

$\text{father}(v) = u$ nếu cha của đỉnh v là u .

4.1. Phương pháp tìm kiếm theo chiều rộng.

4.1.1. Kỹ thuật tìm kiếm rộng.

Kỹ thuật tìm kiếm rộng là: *Tìm kiếm trên tất cả các nút của một mức trong không gian bài toán trước khi chuyển sang các nút của mức tiếp theo.*

- Trong tìm kiếm theo bề rộng, trạng thái nào được sinh ra trước sẽ được phát triển trước, do đó các danh sách được xử lý như hàng đợi.

Kỹ thuật tìm kiếm rộng bắt đầu từ mức thứ nhất của không gian bài toán, chẳng hạn “đi từ trái sang phải”. Nếu không thấy lời giải tại mức này, nó chuyển xuống mức sau để tiếp tục ... đến khi định vị được lời giải nếu có.

4.1.2. Giải thuật.

Input:

Cây/Đồ thị $G = (V, E)$ với đỉnh gốc là n_0 (trạng thái đầu); Tập đích: Goals.

Output:

Một đường đi p từ n_0 đến một đỉnh $n^* \in \text{Goals}$.

Phương pháp:

Sử dụng hai danh sách hoạt động theo nguyên tắc FIFO (queue):

+ MO lưu tập đỉnh đưa vào để xét.

+ DONG lưu tập đỉnh đã xét, được liệt kê theo thứ tự - Vết – Đường đi.

Procedure BrFS; (Breadth First Search)

Begin

Append(MO, n_0)

DONG=null;

While MO<> null do

begin

n:= Take(MO);

if $n \in \text{DICH}$ then exit;

Append(DONG, n);

```

For  $m \in T(n)$  and  $m \notin \text{DONG} + \text{MO}$  do
    Append(MO, m);
end;

Write ('Không có lời giải');

End;

```

Chú ý: Thủ tục Append(MO, n_0) bổ sung một phần tử vào queue MO.

Hàm Take(MO) lấy một phần tử từ queue MO.

- Nếu G là cây thì không cần dùng danh sách DONG.

4.1.3. Đánh giá độ phức tạp của giải thuật tìm kiếm rộng.

Giả sử rằng, mỗi trạng thái khi được xét sẽ sinh ra k trạng thái kế tiếp. Khi đó ta gọi k là nhân tố nhánh. Nếu bài toán tìm được nghiệm theo phương pháp tìm kiếm rộng có độ dài d . Như vậy, đỉnh đích sẽ nằm ở mức $d+1$, do đó số đỉnh cần xét lớn nhất là:

$$k^0 + k^1 + k^2 + \dots + k^d.$$

Như vậy độ phức tạp thời gian của giải thuật là $O(k^d)$. Độ phức tạp không gian cũng là $O(k^d)$, vì tất cả các đỉnh của cây tìm kiếm ở mức $d+1$ đều phải lưu vào danh sách.

4.1.4. Ưu và nhược điểm của phương pháp tìm kiếm rộng.

4.1.4.1. Ưu điểm.

- Kỹ thuật tìm kiếm rộng là kỹ thuật vét cạn không gian trạng thái bài toán vì vậy sẽ tìm được lời giải nếu có.
- Đường đi tìm được đi qua ít đỉnh nhất.

4.1.4.2. Nhược điểm.

- Tìm kiếm lời giải theo thuật toán đã định trước, do vậy tìm kiếm một cách máy móc; khi không có thông tin hỗ trợ cho quá trình tìm kiếm, không nhận ra ngay lời giải.
- Không phù hợp với không gian bài toán kích thước lớn. Đối với loại bài toán này, phương pháp tìm kiếm rộng đối mặt với các nhu cầu:
 - Cần nhiều bộ nhớ theo số nút cần lưu trữ.

- Cần nhiều công sức xử lý các nút, nhất là khi các nhánh cây dài, số nút tăng.
- Dễ thực hiện các thao tác không thích hợp, thừa, đưa đến việc tăng đáng kể số nút phải xử lý.
- Không hiệu quả nếu lời giải ở sâu. Phương pháp này không phù hợp cho trường hợp có nhiều đường dẫn đến kết quả nhưng đều sâu.
- Giao tiếp với người dùng không thân thiện. Do duyệt qua tất cả các nút, việc tìm kiếm không tập trung vào một chủ đề.

4.1.5. Các ví dụ.

Ví dụ 1. Bài toán đong nước với $m = 5$, $n = 4$, $k = 3$. **Nhiệm vụ:** Thiết lập Không gian trạng thái $K = (T, S, G, F)$ (Biểu diễn d/d cây); Mô tả Cây tìm kiếm.

$F = \{ \text{đong đầy, đổ hết ra, đổ sang bình khác} \}$. Ta có thể dùng tập luật sau:

1. Nếu bình X đầy thì đổ hết nước trong bình X đi.
2. Nếu bình Y rỗng thì đổ đầy nước vào bình Y .
3. Nếu bình X không đầy và bình Y không rỗng thì hãy trút nước từ bình Y sang bình X (cho đến khi bình X đầy hoặc bình Y hết nước).

Mức 1: Trạng thái đầu (0;0)

Mức 2: Các trạng thái (5;0), (0;4),

Mức 3: (5;4), (1;4), (4;0)

Mức 4: (1;0), (4;4)

Mức 5: (0;1), (5;3)

Ở mức 5 ta gặp trạng thái đích là (5;3) vì vậy có được lời giải như sau:

(0;0) $\rightarrow$ (0;4) $\rightarrow$ (4;0) $\rightarrow$ (4;4) $\rightarrow$ (5;3)

Có thể trình bày quá trình tìm kiếm dưới dạng bảng sau:

Mức	i	T(i) Các trạng thái - Nút	$\uparrow$ MO $\downarrow$ <i>Queue.</i>	DONG
			(0;0)	
1	(0;0)	(5;0) (0;4)	(5;0) (0;4)	(0;0)
2	(5;0)	(5;4) (0;0) (1;4)	(0;4) (5;4) (1;4)	(0;0) (5;0)
2	(0;4)	(5;4) (0;0) (4;0)	(5;4) (1;4) (4;0)	(0;0) (5;0) (0;4)
3	(5;4)	(0;4) (5;0)	(1;4) (4;0)	(0;0) (5;0) (0;4) (5;4)
3	(1;4)	(1;0) (5;0)	(4;0) (1;0)	(0;0) (5;0) (0;4) (5;4) (1;4)
3	(4;0)	(4;4) (0;4)	(1;0) (4;4)	(0;0) (5;0) (0;4) (5;4) (1;4) (4;0)
4	(1;0)	(1;4) (0;1)	(4;4) (0;1)	(0;0) (5;0) (0;4) (5;4) (1;4) (4;0)

				(1;0)
4	(4;4)	(4;0) (5;3)	(0;1) (5;3)	(0;0) (5;0) (0;4) (5;4) (1;4) (4;0) (1;0) (4;4)
5	(0;1)	(5;1) (1;0)	(5;3) (5;1)	(0;0) (5;0) (0;4) (5;4) (1;4) (4;0) (1;0) (4;4) (0;1).
5	(5;3)			(0;0) (5;0) (0;4) (5;4) (1;4) (4;0) (1;0) (4;4) (0;1) (5;3).

Ví dụ 2. Bài toán trò chơi 8 số với Bảng xuất phát là:

2	8	3
1	6	4
7		5

Bảng kết thúc

1	2	3
8		4
7	6	5

Mức 1: Có một trạng thái (Bắt đầu)

2	8	3
1	6	4
7		5

Mức 2: Có ba trạng thái

2	8	3
1		4
7	6	5

f_u

2	8	3
1	6	4
	7	5

f_l

2	8	3
1	6	4
7	5	

f_r

Mức 3: Có năm trạng thái. (Áp dụng riêng rẽ cho từng trạng thái của mức 2).

Đối với trạng thái đầu:

f_l

2	8	3
	1	4

f_r

2	8	3
1	4	

f_u

2		3
1	8	4

7	6	5
---	---	---

7	6	5
---	---	---

7	6	5
---	---	---

Đối với 2 trạng thái tiếp theo:

2	8	3
	6	4
1	7	5

f_u

2	8	3
1	6	
7	5	4

f_u

Mức 4: Có mười trạng thái

	8	3
2	1	4
7	6	5

2	8	3
7	1	4
	6	5

2	8	
1	4	3
1	7	5

2	8	3
1	4	5
7	6	

	2	3
1	8	4
7	6	5

2	3	
1	8	4
7	6	5

	8	3
2	6	4
1	7	5

2	8	3
6		4
1	7	5

2	8	
1	6	3
7	5	4

2	8	3
1	6	4
7	5	

Mức 5: Có 12 trạng thái

1	2	3
	8	4
7	6	5

2	3	4
1	8	
7	6	5

8		3
2	1	4
7	6	5

2	8	3
7	1	4
6		5

2		8
1	4	3
7	6	5

2	8	3
1	4	5
7		6

8		3
2	6	4
1	7	5

2		3
6	8	4
1	7	5

2	8	3
6	7	4
1		5

2		8
1	6	3
7	5	4

2		3
1	8	3
7	5	4

2	8	3
1	5	6
7		4

Mức 6: Có 24 trạng thái

1	2	3
---	---	---

1	2	3
---	---	---

8		4
7	6	5

7	8	4
	6	5

...

Ở mức này ta gặp được trạng thái đích.

1	2	3
8		4
7	6	5

4.2. Phương pháp tìm kiếm theo chiều sâu.

4.2.1. Kỹ thuật tìm kiếm sâu.

Trong tìm kiếm theo độ sâu, *trạng thái được chọn để phát triển thường là trạng thái được sinh ra sau cùng trong số các trạng thái chờ phát triển*. Do đó thuật toán tìm kiếm theo độ sâu là hoàn toàn tương tự như thuật toán tìm kiếm theo bề rộng, chỉ có một điều khác là, ta xử lý danh sách *L* các trạng thái chờ phát triển không phải như hàng đợi mà như *ngăn xếp (Stack)*.

Tìm kiếm sâu trong không gian bài toán: Bắt đầu từ một nút được chọn rồi tiếp tục cho đến khi hoặc đến ngõ cụt (thì quay lại một mức trên đồ thị và tìm theo hướng khác) hoặc đến đích.

Thuật toán tìm kiếm theo chiều sâu được hình dung như việc khảo sát một cây bắt đầu từ gốc đi theo mọi cành có thể được, khi gặp cành cụt thì quay lại xét cành chưa đi qua.

Tổng quát, giả sử đang xét đỉnh *i*, khi đó các đỉnh kề với *i* có các trường hợp:

- + Nếu tồn tại đỉnh *j* kề *i* chưa được xét thì xét đỉnh này (nó trở thành đỉnh đã xét) và bắt đầu từ đó tiếp tục quá trình tìm kiếm với đỉnh này,...
- + Nếu với mọi đỉnh kề với *i* đều đã được xét thì *i* coi như duyệt xong và quay trở lại tìm kiếm từ đỉnh mà từ đó ta đi đến được *i*.

4.2.2. Giải thuật.

Input:

Cây/Đồ thị $G = (V, E)$ với đỉnh gốc là n_0 (trạng thái đầu)

Tập đích Goals

Output:

Một đường đi p từ n_0 đến một đỉnh $n^* \in \text{Goals}$

Method:

Sử dụng hai danh sách hoạt động theo nguyên tắc LIFO (Stack) MO và DONG

Procedure DFS; (Depth First Search)

Begin

Push (MO, n_0)

DONG=null;

While MO <> null do

begin

n:=pop (MO);

if $n \in \text{DICH}$ then exit;

push (DONG, n);

For $m \in T(n)$ and $m \notin \text{DONG} + \text{MO}$ do

Push (MO, m);

end;

Write ('Không có lời giải');

End;

Chú ý: Thủ tục Push(MO, n_0) thực hiện việc bổ sung n_0 vào stack MO

Hàm Pop(MO) lấy phần tử đầu tiên trong Stack MO.

4.2.3. Đánh giá độ phức tạp của thuật toán tìm kiếm sâu.

Giả sử nghiệm của bài toán là đường đi có độ dài d, cây tìm kiếm có nhân tố nhánh là k. Có thể xảy ra nghiệm là đỉnh cuối cùng được xét ở mức d+1. Khi đó độ phức tạp thời gian của thuật toán tìm kiếm theo chiều sâu trong trường hợp xấu nhất là $O(k^d)$.

4.2.4. Ưu và nhược điểm của phương pháp tìm kiếm sâu.

4.2.4.1. Ưu điểm.

- Nếu bài toán có lời giải, phương pháp tìm kiếm sâu bảo đảm tìm ra lời giải.
- Kỹ thuật tìm kiếm sâu tập trung vào đích, con người cảm thấy hài lòng khi các câu hỏi tập trung vào vấn đề chính.
- Do cách tìm của kỹ thuật này, nếu lời giải ở rất sâu, kỹ thuật tìm sâu sẽ tiết kiệm thời gian.

4.2.4.2. Nhược điểm.

- Tìm sâu khai thác không gian bài toán để tìm lời giải theo thuật toán đơn giản một cách cứng nhắc. Trong quá trình tìm nó không có thông tin nào hỗ trợ để phát hiện lời giải. Nếu chọn nút ban đầu không thích hợp có thể không dẫn đến đích của bài toán.

4.2.5. Các ví dụ.

Ví dụ 1. Bài toán đong nước với $m = 5$, $n = 4$, $k = 3$

Nếu ta chọn nhánh ưu tiên đồ đầy bình thứ hai, thì sẽ tìm thấy lời giải rất nhanh.

Quá trình tìm kiếm có thể trình bày bằng bảng dưới đây

Mức	i	T(i)	MO $\downarrow\uparrow$ - LIFO	DONG
			(0;0)	
1	(0;0)	(5;0) (0;4)	(0;4) (5;0)	(0;0)
2	(0;4)	(5;4) (0;0) (4;0)	(4;0) (5;4) (5;0)	(0;0) (0;4)
3	(4;0)	(0;4) (4;4)	(4;4) (5;4) (5;0)	(0;0) (0;4) (4;0)
4	(4;4)	(4;0) (5;3)	(5;3) (5;4) (5;0)	(0;0) (0;4) (4;0) (4;4)
5	(5;3)			(0;0) (0;4) (4;0) (4;4) (5;3)

Lời giải tìm được: $(0;0) \rightarrow (0;4) \rightarrow (4;0) \rightarrow (4;4) \rightarrow (5;3)$

Ví dụ 2. Bài toán Tháp Hà nội với $n = 3$.

Dùng bộ ba $(x_1; x_2; x_3)$ biểu diễn trạng thái, với x_i là cọc chứa đĩa lớn thứ i .

i	T(i)	MO $\downarrow\uparrow$	DONG
		(1;1;1)	
(1;1;1)	(1;1;2) (1;1;3)	(1;1;2) (1;1;3)	(1;1;1)
(1;1;3)	(1;1;1)(1;1;2)(1;2;3)	(1;1;2)(1;2;3)	(1;1;1)(1;1;3)

Các bước thực hiện tìm kiếm rộng:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U.

Chúng ta thực hiện tìm kiếm rộng bằng cách dùng các danh sách MO(mở) và DONG (đóng) để theo dõi tiến độ trong không gian trạng thái đó. Danh sách MO sẽ liệt kê các trạng thái vừa được sinh ra, nhưng con của chúng chưa được khảo sát. Thứ tự mà các trạng thái bị loại ra khỏi MO sẽ xác định thứ tự tìm kiếm. Danh sách DONG thì ghi các trạng thái đã được xem xét rồi.

1. MO = [A]; DONG = []
2. MO = [B,C,D]; DONG = [A]
3. MO = [C,D,E,F]; DONG = [B,A]
4. MO = [D,E,F,G,H]; DONG = [C,B,A]
5. MO = [E,F,G,H,I,J]; DONG = [D,C,B,A]
6. MO = [F,G,H,I,J,K,L]; DONG = [E,D,C,B,A]
7. MO = [G,H,I,J,K,L,M]; (vì L đã có trong MO); DONG = [F,E,D,C,B,A]
- 8.....

Các bước thực hiện tìm kiếm sâu: (Ở đây giả thiết rằng: Lấy phần tử phát sinh đầu tiên)

A, B, E, K, S, L, T, F, M, C, G, N, H, O, P, U, D, I, Q, J, R

1. MO = [A]; DONG = []
2. MO = [B,C,D]; DONG = [A]
3. MO = [E,F,C,D]; DONG = [B,A]
4. MO = [K,L,F,C,D]; DONG = [E,B,A]
5. MO = [S,L,F,C,D]; DONG = [K,E,B,A]
6. MO = [L,F,C,D]; DONG = [S,K,E,B,A]
7. MO = [T,F,C,D]; DONG = [L,S,K,E,B,A]

8. MO = [F,C,D]; DONG = [T,L,S,K,E,B,A]

.....

Hai chiến lược tìm kiếm kinh nghiệm quan trọng nhất là tìm kiếm tốt nhất - đầu tiên (best-first search) và tìm kiếm leo đồi (hill-climbing search). Có thể xác định các chiến lược này như sau:

Tìm kiếm tốt nhất đầu tiên = Tìm kiếm theo bề rộng + Hàm đánh giá

Tìm kiếm leo đồi = Tìm kiếm theo độ sâu + Hàm đánh giá

4.3. Phương pháp tìm kiếm tốt nhất đầu tiên (Best First Search).

Hai kỹ thuật tìm kiếm rộng và tìm kiếm sâu đều là phương pháp cơ bản, vét cạn không gian để tìm ra lời giải theo thủ tục xác định trước. Mặc dù có sử dụng tri thức về trạng thái của bài toán để định hướng tìm kiếm nhưng không nhiều. Cho dù có một số ưu điểm, nhưng chúng là những kỹ thuật thực hiện một cách máy móc. Chính vì vậy chúng bị gán tên là “*kỹ thuật tìm kiếm mù*”.

Ưu điểm của tìm kiếm theo chiều sâu là không quan tâm đến sự mở rộng của tất cả các nhánh.

Ưu điểm của tìm kiếm chiều rộng là không bị sa vào các đường dẫn bế tắc (các nhánh cụt).

4.3.1. Kỹ thuật tìm kiếm tốt nhất đầu tiên.(BFS)

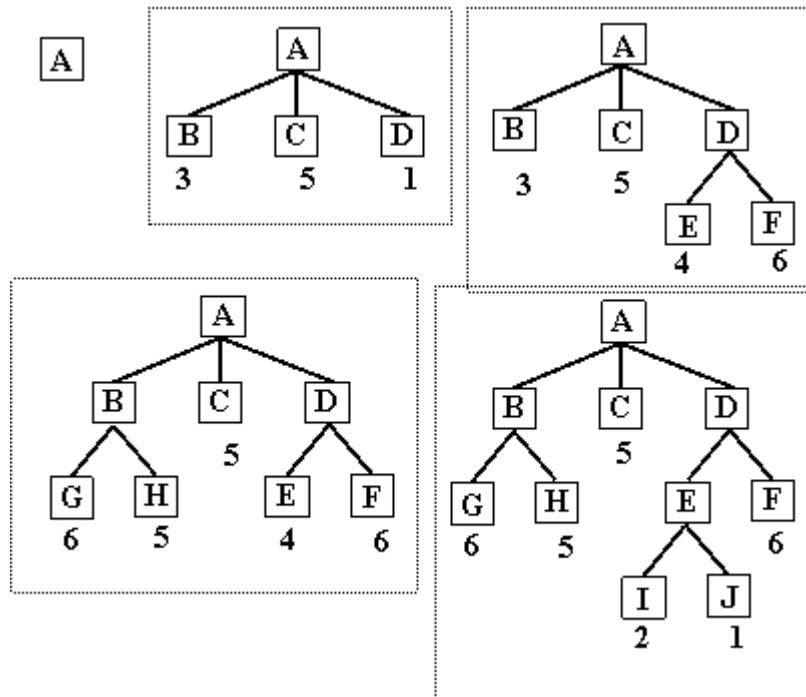
Kỹ thuật tìm kiếm tốt nhất đầu tiên sẽ kết hợp 2 phương pháp trên cho phép ta đi theo *một con đường duy nhất tại một thời điểm*, nhưng đồng thời vẫn "quan sát" được những hướng khác. Nếu con đường đang đi "có vẻ" không triển vọng bằng những con đường ta đang "quan sát", ta sẽ chuyển sang đi theo một trong số các con đường này.

Tìm kiếm tốt nhất đầu tiên = Tìm kiếm theo bề rộng + Hàm đánh giá

1. Tại mỗi bước của tìm kiếm BFS, ta chọn đi theo trạng thái có *khả năng cao nhất* trong số các trạng thái đã được xét *cho đến thời điểm đó*.
2. Ta sẽ ưu tiên đi vào những nhánh tìm kiếm có khả năng nhất. (giống tìm kiếm leo đồi dốc đứng).

Trong thủ tục này, chúng ta sử dụng danh sách L để lưu các trạng thái chờ phát triển, danh sách L được sắp theo thứ tự tăng dần của hàm đánh giá sao cho trạng thái có giá trị hàm đánh giá nhỏ nhất ở đầu danh sách.

Ví dụ:



Hình Minh họa thuật giải Best-First Search

Khởi đầu, chỉ có một nút (trạng thái) A nên nó sẽ được mở rộng tạo ra 3 nút mới B, C và D. Các con số được ghi dưới nút là giá trị cho biết độ tốt của nút. Con số càng nhỏ, nút càng tốt. Do D là nút có khả năng nhất nên nó sẽ được mở rộng tiếp sau nút A và sinh ra 2 nút kế tiếp là E và F. Đến đây, ta lại thấy nút B có vẻ có khả năng nhất (trong các nút B, C, E, F) nên ta sẽ chọn mở rộng nút B và tạo ra 2 nút G và H. Nhưng lại một lần nữa, hai nút G, H này được đánh giá ít khả năng hơn E, vì thế sự chú ý lại trở về E. E được mở rộng và các nút được sinh ra từ E là I và J. Ở bước kế tiếp, J sẽ được mở rộng vì nó có khả năng nhất. Quá trình này tiếp tục cho đến khi tìm thấy một lời giải.

Để cài đặt các thuật giải tìm kiếm BFS, người ta thường cần dùng 2 tập hợp sau:

MO: tập chứa các trạng thái đã được sinh ra nhưng chưa được xét đến (vì ta đã chọn một trạng thái khác). Thực ra, **MO** là một loại hàng đợi ưu tiên (priority queue) mà trong đó, phần tử có độ ưu tiên cao nhất là phần tử *tốt nhất*.

DONG: tập chứa các trạng thái đã được xét đến. Chúng ta cần lưu trữ những trạng thái này trong bộ nhớ để đề phòng trường hợp khi một trạng thái mới được tạo ra lại trùng với một trạng thái mà ta đã xét đến trước đó. Nếu không gian tìm kiếm có dạng cây thì không cần dùng tập này.

Trong các chương trình trí tuệ nhân tạo, kỹ thuật tìm kiếm tốt nhất đầu tiên sử dụng *hàm đánh giá*. Hàm này dùng các thông tin hiện tại về mức độ quan trọng của bài toán tại nút đó để gán giá trị cho nút này, gọi là trọng số của nút và giá trị này được xem xét trong lúc tìm kiếm. Thông thường, nút có trọng số nhỏ/ lớn nhất sẽ được chọn trong quá trình tìm kiếm.

4.3.2. Hàm đánh giá

Giả sử u là một *trạng thái đạt tới* (có đường đi từ trạng thái ban đầu u_0 tới u). Ta thường xác định hai hàm đánh giá sau:

- $g(u)$ là đánh giá độ dài đường đi ngắn nhất từ u_0 tới u (Đường đi từ u_0 tới trạng thái u không phải là trạng thái đích được gọi là *đường đi một phần*, để phân biệt với *đường đi đầy đủ*, là đường đi từ u_0 tới trạng thái đích).
- $h(u)$ là đánh giá độ dài đường đi ngắn nhất từ u tới trạng thái đích.

Hàm $h(u)$ được gọi là *chấp nhận được* (hoặc đánh giá thấp) nếu với mọi trạng thái u , $h(u) \leq$ độ dài đường đi ngắn nhất thực tế từ u tới trạng thái đích. Chẳng hạn trong bài toán tìm đường đi ngắn nhất trên bản đồ giao thông, ta có thể xác định $h(u)$ là độ dài đường chim bay từ u tới đích.

Để tăng hiệu quả tìm kiếm, ta có thể sử dụng hàm đánh giá mới:

$$f(u) = g(u) + h(u)$$

Tức là, $f(u)$ là đánh giá độ dài đường đi ngắn nhất qua u từ trạng thái ban đầu tới trạng thái kết thúc.

Với mỗi trạng thái u . Hàm $h(u)$, $g(u)$ hoặc $f(u)$ **được gọi là hàm đánh giá**.

Phương pháp tìm kiếm kinh nghiệm là phương pháp tìm kiếm có sử dụng đến hàm đánh giá. *Trong quá trình tìm kiếm, tại mỗi bước ta sẽ chọn trạng thái kế tiếp là trạng thái có nhiều hứa hẹn dẫn tới đích nhiều nhất.*

Quá trình tìm kiếm trong không gian trạng thái có sử dụng hàm đánh giá bao gồm các bước cơ bản sau:

- Biểu diễn thích hợp các trạng thái và các toán tử chuyển trạng thái
- Xây dựng hàm đánh giá
- Thiết kế chiến lược chọn trạng thái ở mỗi bước

4.3.3. Ưu và nhược điểm của phương pháp tìm kiếm tốt nhất đầu tiên.

4.3.3.1. Ưu điểm.

- Phương pháp tìm kiếm tốt nhất đầu tiên tổ hợp các ưu điểm của phương pháp tìm kiếm rộng và tìm kiếm sâu.

- Ưu điểm chủ yếu của phương pháp tìm kiếm tốt nhất đầu tiên là dùng tri thức để dẫn dắt việc tìm kiếm. Tri thức này giúp người ta bắt đầu từ đâu là tốt nhất và cách tốt nhất để tiến hành tìm lời giải.

- Tìm kiếm tốt nhất đầu tiên tuân theo cách suy lý của một chuyên gia. Do đó có thể thấy rõ đường đi hơn tìm kiếm rộng và tìm kiếm sâu.

4.3.3.2. Nhược điểm.

- Quá trình tìm kiếm có thể đi xa khỏi lời giải. Kỹ thuật này chỉ xét một phần của không gian và coi đó là phần hứa hẹn hơn cả.

4.3.4. Giải thuật AT (Algorithm for Tree Search).

Dữ liệu tương tự như giải thuật tìm kiếm rộng và sâu, sử dụng danh sách MO để lưu các đỉnh sẽ xét.

Procedure BFS; {Best First Search}

Begin

Push(MO, n_0);

while MO $\neq$ null do

begin

i := Pop(MO);

```

        if  $i \in \text{Goals}$  then exit;
        for  $j \in T(i)$  do
            Push(MO,j);
        Sort(MO); {theo thứ tự của hàm đánh giá}
    end;
write('Không có lời giải');
end;

```

1. Thuật giải AT (*Algorithm for Tree Search*)

Thuật giải AT là một phương pháp tìm kiếm theo kiểu BFS với độ tốt của nút là giá trị hàm **g()** – tổng chiều dài con đường đã đi từ trạng thái bắt đầu đến trạng thái hiện tại.

🔗 Thuật giải AT

1. Đặt **MO** chứa trạng thái khởi đầu.
2. Cho đến khi tìm được trạng thái đích hoặc không còn nút nào trong **MO**, thực hiện:
 - 2.a. Chọn trạng thái (**Tmax**) có **giá trị g nhỏ nhất** trong **MO** (và xóa **Tmax** khỏi **MO**)
 - 2.b. Nếu **Tmax** là trạng thái kết thúc thì thoát.
 - 2.c. Ngược lại, tạo ra các trạng thái kế tiếp **Tk** có thể có từ trạng thái **Tmax**. Đối với mỗi trạng thái kế tiếp **Tk** thực hiện:

$$g(Tk) = g(Tmax) + \text{cost}(Tmax, Tk);$$

Thêm **Tk** vào **MO**.

4.3.5. Các ví dụ.

Ví dụ 1 Trong bài toán tìm kiếm đường đi trên bản đồ giao thông, ta có thể lấy độ dài của đường chim bay từ một thành phố đang xét tới một thành phố đích làm giá trị của hàm đánh giá của thành phố đang xét.

Ví dụ 2 Bài toán 8 số. Chúng ta có thể đưa ra hai cách đánh giá

u =

Đích=

3	2	8		1	2	3
---	---	---	--	---	---	---

	6	4		8		4
7	1	5		7	6	5

- Hàm h_1 : Với mỗi trạng thái u thì $h_1(u)$ là số quân không nằm đúng vị trí của nó trong trạng thái đích.

Với ví dụ trên, ta có $h_1(u)=4$

- Hàm h_2 : Gọi $h_2(u)$ là tổng khoảng cách giữa vị trí của các quân trong trạng thái u và vị trí của nó trong trạng thái đích. Ở đây, khoảng cách được hiểu là số lần dịch chuyển ít nhất theo hàng hoặc cột để đưa một quân ở vị trí hiện tại tới trạng thái đích. (Khoảng cách Manhattan)

Với ví dụ trên, ta có: $h_2(u)=2+3+1+3=9$ (vì quân 3 cần ít nhất 2 dịch chuyển, quân 8 cần ít nhất 3 dịch chuyển, quân 6 cần ít nhất 1 dịch chuyển và quân 1 cần ít nhất 3 dịch chuyển).

Nếu chọn $h_1(u)$, thì phương án tiếp theo được chọn là trạng thái có $h_1(u)$ nhỏ nhất.

$f_u(a_{ij})$				$f_d(a_{ij})$				$f_r(a_{ij})$		
3	2	8	→		2	8		3	2	8
	6	4		3	6	4		6		4
7	1	5		7	1	5		7	1	5
h=4				h=5				h=4		

4.4. Tìm kiếm đường đi có giá thành cực tiểu - Thuật toán ATK

(Algorithm for Knowledgeable Tree Search)

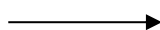
Cho đồ thị $G = (V, E)$ biểu diễn bài toán với đỉnh xuất phát n_0 và tập đích DICH xác định.

Với mỗi phép chuyển trạng thái $n_i \rightarrow n_{i+1}$ tốn chi phí $c(n_i, n_{i+1})$ ký hiệu $c(u)$ với $u = (n_i, n_{i+1}) \in E$

$c(u)$

n_i

n_{i+1}



Vấn đề:

Tìm đường đi $p: n_0 \rightarrow n^* \in DICH$ sao cho $c(p) = \sum_{u \in p} c(u) \rightarrow \min$

Chẳng hạn trong bài toán tìm đường đi trong bản đồ giao thông, giá của cung (i,j) chính là độ dài của đường nối thành phố i với thành phố j . Độ dài đường đi được xác định là tổng độ dài các cung trên đường đi. Vấn đề đặt ra là tìm đường đi ngắn nhất từ trạng thái ban đầu đến trạng thái đích.

- **Phương pháp giải**

Gọi $g(n)$ là giá của đường đi cực tiểu từ đỉnh n_0 đến n , khi đó bài toán có thể phát biểu như sau:

Tìm đường đi từ đỉnh $n_0 \rightarrow n_k \in DICH$ **sao cho:** $g(n_k) = \min \{g(n) / n \in DICH\}$

Lúc đó, ta có: $g(n_0) = 0$

$$g(m) = \min_{(n,m) \in E} \{g(n) + c(n,m)\}$$

Tại mỗi thời điểm, chọn đỉnh n trong MO để xét là đỉnh có chi phí $g(n)$ nhỏ nhất của đường đi từ đỉnh xuất phát đến đỉnh n .

Dùng 2 danh sách MO, DONG như trên.

- **Thuật toán ATK** (*Algorithm for Knowledgeable Tree Search*)

Input:

Đồ thị $G = (V,E)$, Đỉnh xuất phát n_0

Hàm chi phí $c: E \rightarrow R^+$

$c(i,j)$: xác định chi phí chuyển từ đỉnh i sang đỉnh j với $(i,j) \in E$

Tập các đỉnh đích: DICH

Output:

Đường đi từ n_0 đến $n^* \in DICH$ sao cho $g(n^*) = c(p) = \min \{g(n) | n \in DICH\}$.

Procedure ATK

{ Dùng $g^0(n)$ là chi phí cực tiểu của đường đi từ đỉnh xuất phát đến đỉnh n tại thời điểm đang xét và xem như hàm g }

Begin

$g(n_0) := 0;$

push(MO, n_0);

```

While MO<>null do
    begin
         $g(n) := \min_{m \in MO} g(m)$ 
        if  $n \in DICH$  then
            exit {xay dung duong di cuc tieu}
        push(DONG, n);
        if  $T(n) \neq null$  then
            for  $m \in T(n)$  do
                if  $m \notin MO + DONG$  then
                    begin
                        push(MO, m);
                         $g(m) := g(n) + c(n, m)$ ;
                        cha(m) := n; (Ghi lai Vết)
                    end
                else
                    if  $g(m) > g(n) + c(n, m)$  then
                        begin
                             $g(m) := g(n) + c(n, m)$ ;
                            cha(m) := n;
                        end;
            end;
        writeln('Khong co duong di');
    End;

```

Ví dụ 1. Bài toán Tháp Hà Nội -với chi phí chuyển đĩa như sau:

Chi phí chuyển đĩa nhỏ giữa 2 cọc gần 1; Chi phí chuyển đĩa nhỏ giữa 2 cọc xa 3
 Chi phí chuyển đĩa vừa giữa 2 cọc gần 2; Chi phí chuyển đĩa vừa giữa 2 cọc xa 5
 Chi phí chuyển đĩa lớn giữa 2 cọc gần 4; Chi phí chuyển đĩa lớn giữa 2 cọc xa 8
 Xuất phát từ đỉnh (1,1,1), ta có $g(1,1,1) = 0$.

Khi xét đỉnh **(1,1,1)** ta có các đỉnh kề và chi phí tương ứng :

$g(1,1,2) = 1$; $g(1,1,3) = 3$; như vậy đỉnh **(1,1,2)** được chọn

Các đỉnh kề của (1,1,2) có giá trị hàm g:

$g(1,1,3) = 2$ (ở đây giá của đỉnh (1,1,3) được tính lại); $g(1,3,2) = 5$; chọn đỉnh **(1,1,3)**, ta lại tính tiếp giá trị hàm g của các đỉnh kề với đỉnh này:

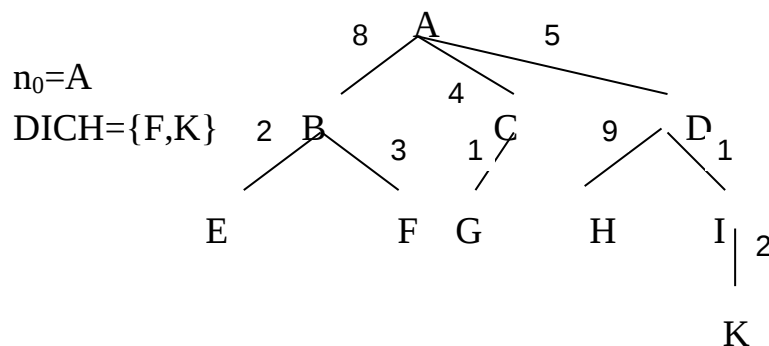
$g(1,2,3) = 2$; lại chọn đỉnh **(1,2,3)**; chi phí của các đỉnh kề với nó:

$g(1,2,1) = 2 + 3 = 5$; $g(1,2,2) = 2 + 1 = 3$; chọn đỉnh **(1,2,2)**

$g(1,2,1) = 3 + 1 = 4$ (được tính lại); $g(3,2,2) = 3 + 8 = 11$, chọn đỉnh (1,2,1)

Cứ tiếp tục như vậy cho đến khi xét đỉnh (3,3,3).

Ví dụ 2



Có thể trình bày quá trình tìm kiếm bằng bảng dưới đây.

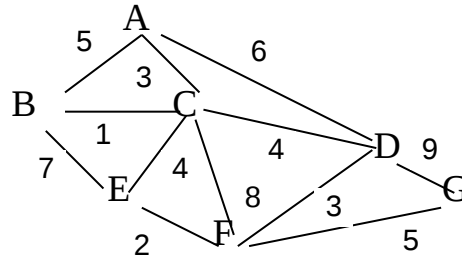
Số được ghi tại các đỉnh là giá trị $g(n)$ tương ứng đỉnh n: $n_{g(n)}$

i	T(i)	MO	DONG
		A_0	
A	B C D	$B_8 C_4 D_5$	A
C	G	$B_8 D_5 G_5$	A C
D	H I	$B_8 G_5 H_{14} I_6$	A C D
G		$B_8 H_{14} I_6$	A C D G
I	K	$B_8 H_{14} K_8$	A C D G
B	E F	$H_{14} K_8 E_{10} F_{11}$	A C D G B
K			

Lời giải của bài toán là $A \rightarrow D \rightarrow I \rightarrow K$ và chi phí của đường đi tìm được là 8

Ví dụ 3.

$n_0 = A$; DICH = {G}



i	T(i)	MO	DONG
		A_0	
A	B C D	$B_5 C_3 D_6$	A
C	A B E F D	$B_4 D_6 E_7 F_{11}$ (Tính lại B)	A C
B	A C E	$D_6 E_7 F_{11}$	A C B
D	A C F G	$E_7 F_9 G_{15}$ (Tính lại F)	A C B D
E	B C F	$F_9 G_{15}$	A C B D E
F	C D E G	G_{14} (Tính lại G)	A C B D E F
G			

Đường đi tìm được p: $A \rightarrow D \rightarrow F \rightarrow G$. Chi phí của đường đi là 14.

4.5. Tìm kiếm cực tiểu sử dụng hàm đánh giá - Thuật toán A^* :

Thuật toán A^* là thuật toán sử dụng kỹ thuật tìm kiếm tốt nhất đầu tiên với hàm đánh giá $f(u)$. (Khái niệm tốt nhất ở đây do hàm đánh giá Heuristic quyết định)

Đối với việc tìm kiếm đường đi với chi phí cực tiểu, người ta sử dụng hàm đánh giá Heuristic như sau:

Gọi $g(n)$: giá cực tiểu đường đi từ $n_0 \rightarrow n$. Tại đỉnh n , $g(n)$ xác định được.

Gọi $h(n)$: giá cực tiểu đường đi từ $n \rightarrow$ DICH, $h(n)$ không xác định được $\Rightarrow$ người ta tìm cách ước lượng giá trị này.

Đặt $f^0(n) = g^0(n) + h^0(n)$: dự đoán chi phí cực tiểu của đường đi từ $n_0 \rightarrow$ DICH có đi qua đỉnh n .

$g^0(n)$ là chi phí của đường đi từ đỉnh xuất phát đến đỉnh n tại thời điểm đang xét. $h^0(n)$ là ước lượng (dự đoán) chi phí đường đi từ đỉnh n đến đích. Việc chọn

giá trị xấp xỉ $h^0(n)$ của $h(n)$ không có một phương pháp tổng quát và được xem như một nghệ thuật. Giá trị này sẽ do các chuyên gia đưa ra.

Lúc này giải thuật tìm kiếm cực tiểu sẽ thay việc xét hàm g bởi hàm f .

Tuy nhiên, người ta cũng đã chứng minh được 2 kết quả như sau:

Kết quả 1: Nếu $h^0(n)$ có tính chất: $0 \leq h^0(n) \leq h(n) \forall n$ và $c(u) > 0 \forall u \in E$ thì thủ tục TKCT sử dụng hàm $f^0(n)$ để chọn phần tử trong MO ra xét (thay $g(n)$) sẽ cho đường đi từ $n_0 \rightarrow n^* \in DICH$ sao cho $g(n_*) = \min_{n \in DICH} g(n)$

Kết quả 2: Giả sử dùng 2 hàm ước lượng h_1^0 và h_2^0 thỏa tính chất: $h_2^0(m) - h_2^0(n) \leq h(m, n)$ (giá cực tiểu của đường đi từ $m \rightarrow n$) và $\forall n \in N, 0 \leq h_1^0(n) \leq h_2^0(n) \leq h(n)$. Khi đó $\#DONG_2 \leq \#DONG_1$

Nhận xét:

$h^0 \equiv h \Rightarrow$ phương án tốt nhất

$h^0 \equiv 0 \Rightarrow$ **phương án tồi nhất**

Thuật toán A^*

Input:

Đồ thị $G = (V, E)$, Đỉnh xuất phát n_0

Hàm chi phí $c: E \rightarrow R^+$

$c(i, j)$: xác định chi phí chuyển từ đỉnh i sang đỉnh j với $(i, j) \in E$

$h: V \rightarrow R^+$; $h(n)$ xác định dự đoán chi phí tối ưu của đường đi từ đỉnh n đến đích. (ký hiệu h thay cho h^0 , (tương tự g))

Tập các đỉnh đích: DICH

Output:

Đường đi từ đỉnh n_0 đến đỉnh $n^* \in DICH$

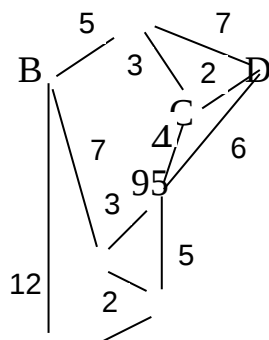
Procedure A^* ;

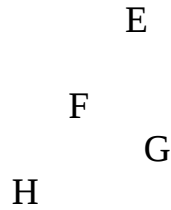
Begin

$g(n_0) := 0;$

push(MO, n_0);

While MO $\neq$ null do





Tìm đường đi từ đỉnh A đến đỉnh H.

Trước tiên đỉnh A được đưa vào danh sách MO

$$g(A) = 0; h(A) = 14; f(A) = 14$$

Xét đỉnh A, (đưa A vào danh sách DONG) ta có các đỉnh kề B, C, D:

$$g(B) = 5; \mathbf{f(B) = 15}; g(C) = 3; \mathbf{f(C) = 13}; g(D) = 7; \mathbf{f(D) = 12} \rightarrow \text{chọn đỉnh D.}$$

Xét đỉnh D (đưa D vào danh sách DONG) có các đỉnh kề A, C, E. Đỉnh A đã ở trong danh sách DONG, ta tính lại $f(C)$ và tính $f(E)$:

$f(C)$ không thay đổi; $f(E) = g(D) + c(D,E) + H(E) = 7 + 6 + 5 = 18$; $\mathbf{f(E) = 18}$, chọn đỉnh C, có các đỉnh kề A, D, E. Tính lại $\mathbf{f(E) = 12}$, chọn E. Các đỉnh kề của E là C, D, F, G. Tính $\mathbf{f(F) = 14}$; $\mathbf{f(G) = 16}$, chọn F. Các đỉnh kề của F là E, G, B và $f(B)$, $f(E)$, $f(G)$ không đổi, chọn B. Các đỉnh kề của B là F, H. $\mathbf{f(H) = 17}$, chọn G. Tính lại $f(H) = 15$ và dừng.

Đường đi tìm được là $p: A \rightarrow C \rightarrow E \rightarrow G \rightarrow H$ với chi phí đường đi là 15

4.6. Phương pháp tìm kiếm leo đồi (Hill-climbing search)

4.6.1. Kỹ thuật tìm kiếm leo đồi.

Tìm kiếm leo đồi = Tìm kiếm theo độ sâu + Hàm đánh giá

Tìm kiếm leo đồi là tìm kiếm theo độ sâu được định hướng bởi hàm đánh giá. Song khác với tìm kiếm theo độ sâu, khi phát triển một đỉnh u thì bước tiếp theo ta chọn trong số các đỉnh con (kề) của u, đỉnh có hứa hẹn nhiều nhất để phát triển, đỉnh này được xác định bởi hàm đánh giá.

4.6.2. Giải thuật.

Input:

Đồ thị $G = (V, E)$, đỉnh xuất phát n_0 .

Hàm đánh giá $h(n)$ đối với mỗi đỉnh n.

Tập đỉnh đích DICH

Output:

Đường đi từ đỉnh n_0 đến DICH

Procedure HCS; {Hill Climbing Search}

begin

Push(MO, n_0);

while MO $\neq$ null do

begin

$i = \text{Pop}(\text{MO})$;

 if $T(i) \cap \text{DICH} \neq \text{null}$ then

 begin

$L := \text{null}$;

 for $j \in T(i)$ do

 if j chưa xét then

 đưa j vào danh sách L

 sắp xếp L theo thứ tự hàm đánh giá;

 chuyển danh sách L vào đầu danh sách MO ;

 end;

 write ('Không có lời giải');

end;

4.6.3. Nhận xét.

Phương pháp tìm kiếm leo đồi chú trọng tìm hướng đi dễ dẫn đến trạng thái đích nhất. Cách đó được đưa ra nhằm làm giảm công sức tìm kiếm. Thuật toán tìm kiếm leo đồi thực chất là thuật toán tìm kiếm theo chiều sâu, song tại mỗi bước ta sẽ ưu tiên chọn một trạng thái có hứa hẹn nhanh tới đích nhất để phát triển trước. Thông thường ta gán mỗi trạng thái của bài toán với một số đo (hàm đánh giá) nào đó nhằm đánh giá mức độ gần đích của nó. Điều đó có nghĩa là nếu trạng thái hiện thời là u thì trạng thái v sẽ được phát triển tiếp theo nếu v kề với u và hàm đánh giá của v đạt giá trị max (hoặc min).

Tuy nhiên phương pháp này không được cải thiện so với các phương pháp khác trong một số trường hợp sau:

- **Cực trị địa phương:** nút đang xét tốt hơn các nút lân cận, nhưng đó không phải là phương án tốt nhất trong toàn thể, ví vậy có thể phải quay lui về nút trước để đi theo hướng khác.
- **Cao nguyên bằng phẳng:** Các giá trị của các phương án như nhau, không xác định được ngay hướng nào là tốt hơn trong vùng lân cận.

4.6.4. Các ví dụ.

Ví dụ 1. Bài toán trò chơi 8 số.

trạng thái đầu	2	8	3
	1	6	4
	7		5

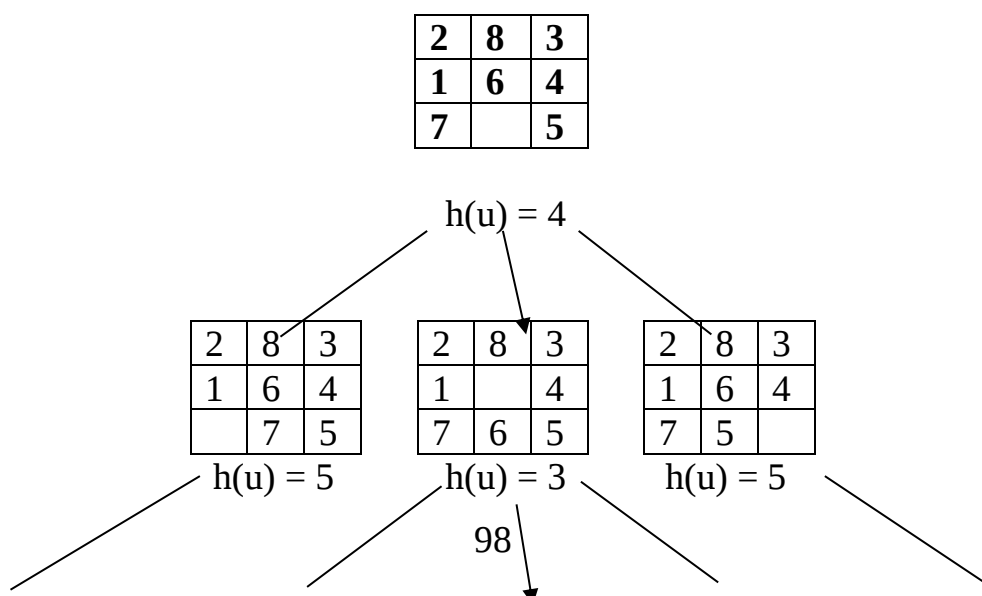
1	2	3
8		4
7	6	5

Trong bài toán này, ta sử dụng hàm đánh giá h như sau:

$h(u)$ là số các chữ số trong trạng thái u không trùng với vị trí của nó trong trạng thái đích. Trạng thái có tiềm năng dẫn đến đích nhanh nhất (được ưu tiên phát triển trước) là trạng thái có hàm đánh giá h đạt giá trị min..

Trạng thái được chọn đi tiếp ở hướng mũi tên. Ở mức 3 chúng ta thấy có hai trạng thái cùng giá trị hàm đánh giá ($= 3$). Đây là trường hợp “cao nguyên bằng phẳng” như nhận xét trên, nếu ta chọn phương án kia thì chắc chắn quá trình tìm kiếm sẽ khác đi nhiều. (Bài tập dành cho Sinh viên).

Minh hoạ cây tìm kiếm cho bài toán này theo giải thuật leo đồi như sau :



	8	3
	1	4
7	6	5

$$h(u) = 3$$

2		3
1	8	4
7	6	5

$$h(u) = 3$$

2	8	3
1	4	
7	6	5

$$h(u) = 4$$

	2	3
1	8	4
7	6	5

$$h(u) = 2$$

2	3	
1	8	4
7	6	5

$$h(u) = 4$$

1	2	3
	8	4
7	6	5

$$h(u) = 1$$

1	2	3
8		4
7	6	5

TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Thanh Thủy, *Trí tuệ nhân tạo - Các phương pháp giải quyết vấn đề và kỹ thuật xử lý tri thức*, NXB GD, 1999
- [2]. Bạch Hưng Khang, Hoàng Kiếm, *Trí tuệ nhân tạo - Các phương pháp và ứng dụng*, Nhà xuất bản Khoa học kỹ thuật, 1989
- [3]. Đỗ Trung Tuấn, *Trí tuệ nhân tạo*, Nhà xuất bản giáo dục, 1998.
- [4]. Bùi Xuân Toại, Trương Gia Việt (Biên dịch), *Trí tuệ nhân tạo - Các cấu trúc và chiến lược giải quyết vấn đề*, NXB Thống kê, 2000
- [5]. Deepak Khemani, *First Course in Artificial Intelligence*, MC Graw-Hill Education, New Delhi, 2014.