ĐẠI HỌC QUỐC GIA
TRƯỜNG ĐẠI HỌC BÁCH KHOA THÀNH PHỐ
KHOA ĐIỆN VÀ ĐIỆN TỬ
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

BÀI GIẢNG MÔN HỌC :
Trí Tuệ Nhân Tạo Và Hệ Chuyên Gia

Thành phố Hồ Chí Minh Ngày 7 Tháng 01 Năm 2006

Biên soạn : Tiến sĩ Nguyễn Thiện Thành

Nội dung bài giảng:

Chương 1 : Tổng Quan Về Trí Tuệ Nhân Tạo

1.1) Trí tuệ nhân tạo là gì ?

Trí tuệ nhân tạo là lĩnh vực khoa học chuyên nghiên cứu các phương pháp chế tạo trí tuệ máy sao cho giống như trí tuệ con người.

Vài định nghĩa của trí tuệ nhân tạo điển hình là
- Hệ thống mà biết suy nghĩ như con người
- Hệ thống mà biết hành động như con người

Để hệ thống mà biết suy nghĩ và hành động như con người thì hệ thống đó phải được trang bị các công cụ như thính giác, tri thức, lý giải tự động, việc học, thị giác và di chuyển giống như con người.

Thông thường, cách giải quyết vấn đề của con người được thể hiện qua bốn thao tác cơ bản đó là
- Xác định tập hợp của các đích
- Thu thập các sự kiện và luật suy diễn
- Cơ chế tập trung
- Bộ máy suy diễn

Như vậy, trí tuệ máy là gì ? là các khả năng giải quyết vấn đề của máy đó là
- Hành động giống như con người.
- Suy nghĩ giống như con người.
- Học giống như con người.
- Xử lý thông tin giống như con người.
- Hành động và suy nghĩ trên cơ sở logic và chính xác.

1.2) Lịch sử phát triển trí tuệ nhân tạo :

Ý tưởng chế tạo trí tuệ máy đã có từ lâu nhưng mãi đến năm 1950, nhà tóan học người Anh công bố công trình khoa học của ông ta đó là "Máy tính và Thông minh", đây được xem như là mốc loch sử bắt đầu phát triển trí tuệ nhân tạo. Nối theo thời điểm này, các chương trình thông minh được công bố đó là

+ Năm 1956, chương trình giải bài tóan tổng quát đã được xuất hiện.
+ Năm 1958, chương trình chứng minh định lý hình học cũng được khám phá.

Đỉnh cao của việc phát triển ở lĩnh vực này phải nói đến những năm 1960. Dù rằng còn bị hạn chế về trang thiết bị nhưng những năm này có nhiều công trình được công bố như

+ Năm 1960, ngôn ngữ Lisp.
+ Năm 1961, chương trình giải các bài tóan đại số sơ cấp.
+ Năm 1963, chương trình trò chơi cờ vua.
+ Năm 1964, chương trình tính tích phân.
+ Năm 1966, chương trình phân tích và học nói.
+ Năm 1968, chương trình điều khiển Robot theo phương án mắt và tay.
+ Năm 1972, ngôn ngữ Prolog.

Từ những năm 1969 đến năm 1999, có nhiều chương trình được xây dựng trên các hệ cơ sở tri thức.

Thật vậy, lĩnh vực trí tuệ đã đi vào đời sống dân dụng từ những năm 1980 đến này.

1.3) Các thành phần cơ bản của trí tuệ nhân tạo :

Có hai thành phần cơ bản của trí tuệ nhân tạo đó là biểu diễn tri thức và tìm kiếm tri thức trong miền biểu diễn. Tri thức của bài tóan có thể được phân ra làm ba lọai tri thức cơ bản đó là tri thức mô tả, tri thức thủ tục và tri thức điều khiển.

+ Tri thức mô tả : là lọai tri thức mô tả những gì mà được biết về bài tóan. Lọai tri thức này bao gồm các sự kiện, các quan hệ và các tính chất của bài tóan.

+ Tri thức thủ tục : là lọai tri thức mô tả cách giải quyết bài tóan. Lọai tri thức này bao gồm luật suy diễn hợp lệ, chiến lược tìm kiếm và giải thuật tìm kiếm.

+ Tri thức điều khiển : là lọai tri thức được xem như là luật chủ chốt điều khiển quá trình lý giải để dẫn đến kết luận.

Để biểu diễn tri thức của bài tóan nhờ các phương pháp biểu diễn như

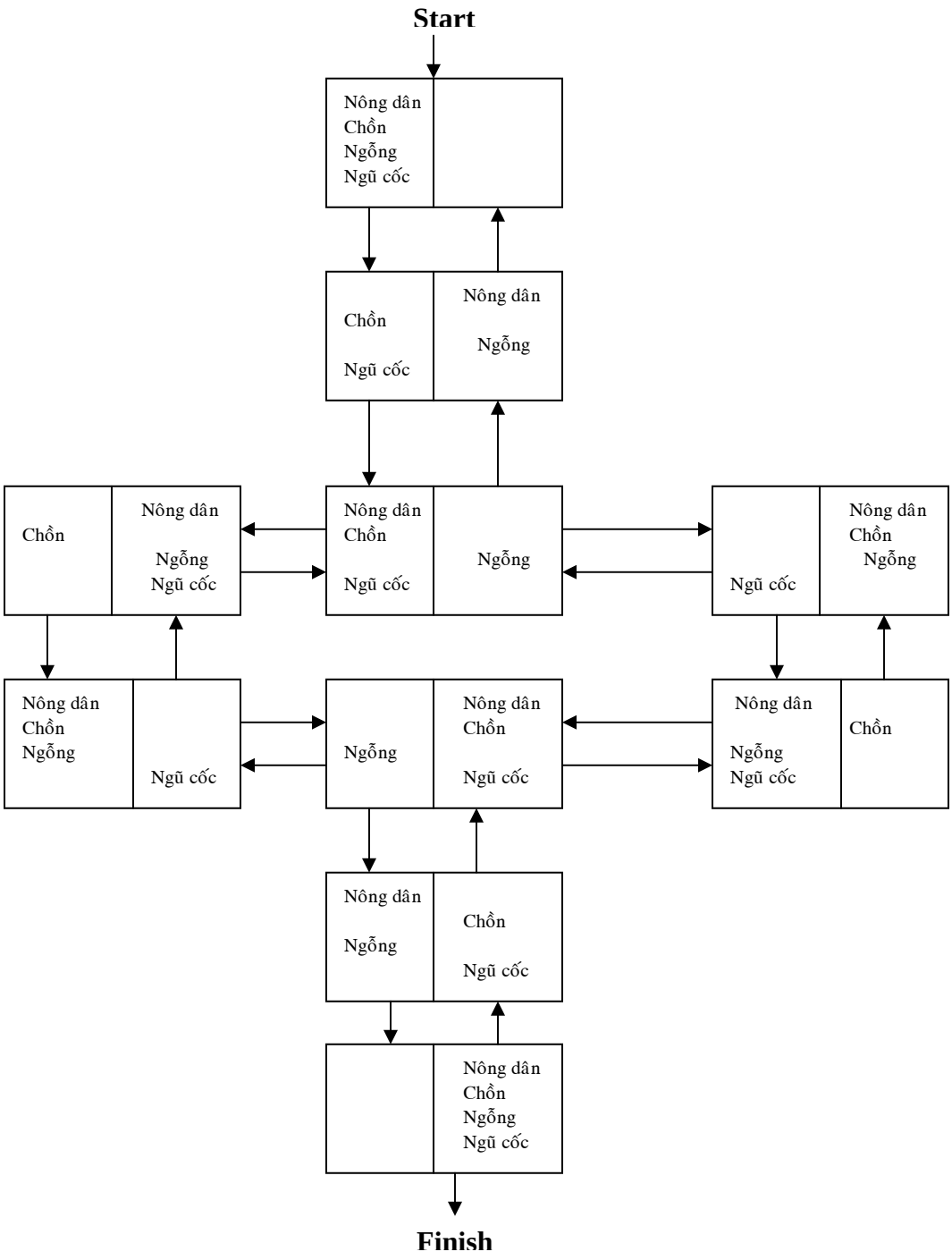
+ Phương pháp biểu diễn nhờ luật
+ Phương pháp biểu diễn nhờ mạng ngữ nghĩa
+ Phương pháp biểu diễn nhờ Frame
+ Phương pháp biểu diễn nhờ logic vị từ

Sau khi tri thức của bài tóan đã được biểu diễn, kỹ thuật giải bài tóan trong lĩnh vực trí tuệ nhân tạo là các phương pháp tìm kiếm trong miền đặc trưng tri thức về bài tóan đó.

Ví dụ : Xét bài tóan người nông dân, chồn, ngỗng và ngũ cốc. Bài tóan đặt ra là người nông dân muốn mang theo với mình một con chồn, một con ngỗng và một số ngũ cốc qua bên kia sông bằng một chiếc thuyền. Tuy nhiên, thuyền của ông ta quá bé chỉ có thể mang theo một thứ duy nhất với ông ta trên mỗi chuyến thuyền sang sông. Nếu ông ta để lại chồn và ngỗng bên này sông thì chồn sẽ ăn ngỗng và nếu ông ta để lại ngỗng và ngũ cốc thì ngỗng sẽ ăn hết số ngũ cốc. Hãy sắp xếp các chuyến thuyền sao cho người nông dân mang mọi thứ sang bên kia sông an tòan?

Với bài tóan này, ta có thể biểu diễn nhờ thông qua các phát biểu ngôn ngữ tự nhiên, tuy nhiên cách biểu diễn này không giúp ta vạch trần ra các ràng buộc vốn sẵn có trong bài tóan. Cách biểu diễn tốt nhất giúp ta có thể vạch trần các ràng buộc vốn sẵn có trong bài tóan là xây dựng một biểu đồ với các nút có đánh nhãn người nông dân mang theo thứ mà ông ta cần phải mang theo trên mỗi chuyến thuyền và các cạnh liên kết giữa các nút là các đường mũi tên chỉ các chuyến thuyền qua lại sông.

Cách biểu diễn này hàm chứa các thành phần như ngữ từ học, cấu trúc, thủ tục và ngữ nghĩa.
+ Ngữ từ học (Lexical) : là các từ vựng hợp lệ được sử dụng như là các ký hiệu trong biểu diễn.
+ Cấu trúc (Structure) : là các đường mũi tên liên kết giữa các nút chỉ định các chuyến thuyền qua lại sông.
+ Thủ tục (Procedure) : là mô tả cách giải bài tóan từ nút này đến nút kia nhờ thông các đường chỉ định mũi tên.
+ Ngữ nghĩa (Semantic) : là ý nghĩa của các nút và các cạnh liên kết thông qua cách giải bài tóan.
Biểu đồ biểu diễn bài tóan người nông dân, chồn, ngỗng và ngũ cốc được mô tả như hình


Start
Nông dân
Chồn
Ngỗng
Ngũ cốc
Chồn
Ngũ cốc
Nông dân
Ngỗng
Chồn
Nông dân
Ngỗng
Ngũ cốc
Nông dân
Chồn
Ngỗng
Ngũ cốc
Nông dân
Chồn
Ngỗng
Nông dân
Chồn
Ngỗng
Ngũ cốc
Nông dân
Chồn
Ngỗng
Ngũ cốc
Nông dân
Ngỗng
Chồn
Ngũ cốc
Nông dân
Chồn
Ngỗng
Ngũ cốc
Finish

Chương 2 : Các Phương Pháp Giải Quyết Vấn Đề Cơ Bản

2.1) Không Gian Bài Tóan :

Tri thức của bài tóan được chia ra làm ba loại tri thức cơ bản đó là tri thức mô tả, tri thức thủ tục và tri thức điều khiển, trong đó tri thức thủ tục định nghĩa không gian bài tóan. Không gian bài tóan có thể được biểu diễn bằng không gian trạng trạng thái đó là một biểu diễn bằng đồ thị định hướng gồm bốn thành phần như sau :

+ S : trạng thái ban đầu của bài tóan (dữ liệu ban đầu).

+ G : tập các trạng thái đích của bài tóan (dữ liệu đích).

+ N : tập các trạng thái khác được phát sinh từ trạng thái ban đầu đạt đến trạng thái đích đó là các nút của đồ thị.

+ A : Tập các trạng thái chuyển tiếp đó là các cung liên kết giữa các nút của đồ thị nhờ thông qua các luật áp dụng của bài tóan.

Luật áp dụng là luật mà vế điều kiện của nó hợp với trạng thái hiện có để vế kết luận của nó phát sinh ra các trạng thái mới.

Đường lời giải của bài tóan là đường bắt đầu từ trạng thái ban đầu thông qua các trạng thái khác được phát sinh đến một trạng thái nào đó trong tập các trạng thái đích.

Ví dụ 1: Không gian bài tóan bình đựng nước.

Cho hai bình đựng nước, một bình có dung tích 4 lít và một bình khác có dung tích 3 lít, cả hai bình không có dấu dung tích. Trạng thái ban đầu của hai bình là rỗng, dùng một bơm nước làm đầy nước với hai bình. Làm cách nào để có chính xác 2 lít nước trong bình 4 lít ?

Vậy, không gian trạng thái cho bài tóan này là gì ?

Giải :

Cho cặp biến số nguyên (x,y) biểu diễn các trạng thái trong không gian trạng thái cho bài tóan này, trong đó x là số lít nước trong bình 4 lít và y là số lít nước trong bình 3 lít.

Không gian trạng thái cho bài tóan được mô tả bằng các thành phần như sau :

+ Trạng thái ban đầu của bài tóan : hai bình đều rỗng đó là cặp số nguyên (0,0).

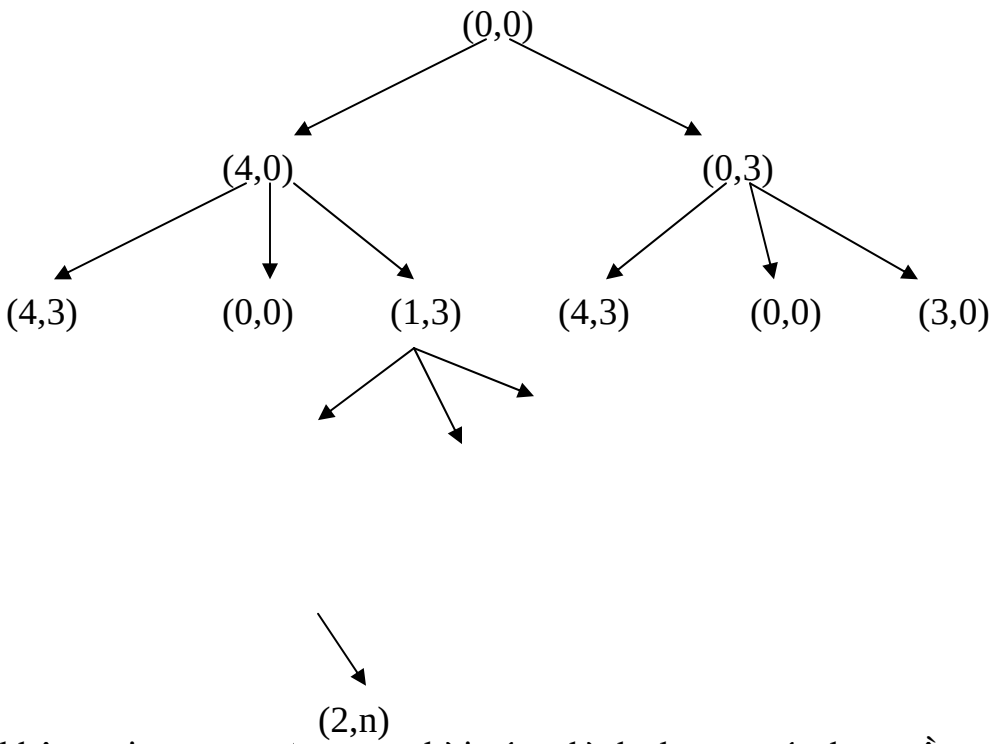
+ Trạng thái đích của bài tóan : cần có chính xác 2 lít nước trong bình 4 lít đó là cặp số nguyên (2,n), tronng đó n là số không xác định trong bình 3 lít.

+ Trạng thái khác của bài tóan : đó là cặp số nguyên (x,y) mô tả các trạng thái trong không gian bài tóan.

+ Trạng thái chuyển tiếp của bài tóan : đó la' bước chuyển tiếp từ trạng thái hiện có đến trạng thái mới nhờ thông luật áp dụng của bài tóan. Luật áp dụng là luật mà vế điều kiện của nó hợp với trạng thái hiện hữu để vế kết luận của nó phát sinh ra trạng thái mới. Tập các luật giải bài tóan bình đựng nước được liệt kê là

Luật 1 : $(x,y/\ x < 4) \rightarrow (4,y)$.
Luật 2 : $(x,y/\ y < 3) \rightarrow (x,3)$.
Luật 3 : $(x,y/\ x > 0) \rightarrow (0,y)$.
Luật 4 : $(x,y/\ y > 0) \rightarrow (x,0)$.
Luật 5 : $(x,y/\ x + y >= 4$ và $y > 0) \rightarrow (4,y - (4 - x))$.
Luật 6 : $(x,y/\ x + y >= 3$ và $x > 0) \rightarrow (x - (3 - y),3)$.
Luật 7 : $(x,y/\ x + y < 4$ và $y > 0) \rightarrow (x + y,0)$.
Luật 8 : $(x,y/x + y < 3$ và $x > 0) \rightarrow (0,x + y)$

Không gian trạng thái cho bài tóan này được biểu diễn bằng đồ thị như hình



Vậy, không gian trạng thái cho bài tóan bình đựng nước bao gồm trạng thái ban đầu, tất cả các trạng thái khác đạt được từ trạng thái ban đầu nhờ thông qua các luật ứng dụng (các trạng thái chuyển tiếp) và trạng thái đích của bài tóan.

Kích thước của không gian trạng thái cho bài tóan là số trạng thái được tạo ra nhờ thông qua các luật ứng dụng từ trạng thái ban đầu đến trạng thái đích của bài tóan.

Ví dụ 2 : Không gian bài tóan trò chơi 8 số.

Bài tóan trò chơi 8 số như một cái mâm hình vuông có ba hàng và ba cột gồm 9 ô, trong đó 8 ô chứa 8 viên ngói có đánh số từ 1 đến 8 và ô còn lại là ô trống.

Cho cấu hình trạng thái ban đầu và cấu hình trạng thái đích của bài tóan được cho như hình

2	8	3
1	6	4
7		5

Trạng Thái Ban Đầu

1	2	3
8		4
7	6	5

Trạng Thái Đích

Bài tóan đặt ra là trượt các viên ngói đến ô trống kề nó, không được phép trượt theo đường chéo sao cho cấu hình trạng thái ban đầu đạt đến cấu hình trạng thái đích của bài tóan.

Vậy, không gian bài tóan này là gì ?

Giải :

Không gian trạng thái cho bài tóan này được mô tả gồm các thành phần như sau :

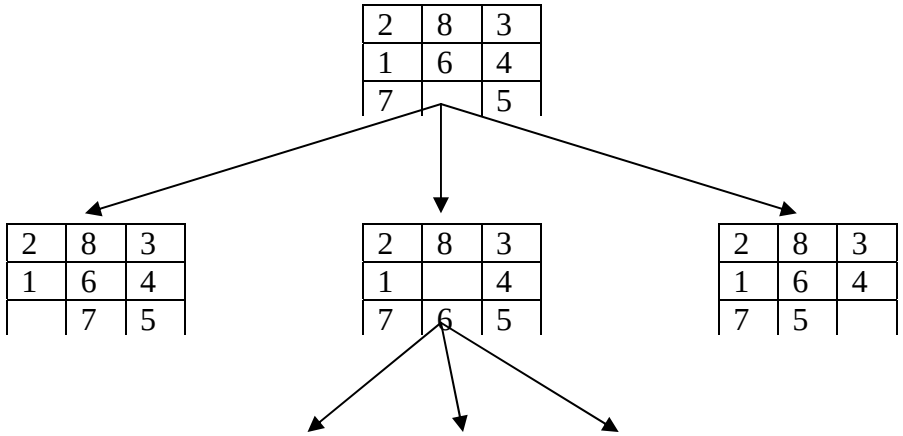
+ Trạng thái ban đầu của bài tóan : là cấu hình mảng hai chiều 3×3 chứa các viên ngói có đánh số cho trước.

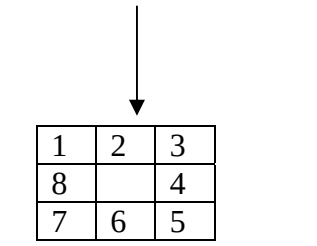
+ Trạng thái đích của bài tóan: cũng là cấu hình mảng hai chiều 3×3 chứa các viên ngói có đánh số cho trước.

+ Trạng thái khác của bài tóan : đó là cấu hình mảng hai chiều 3×3 chứa các viên ngói mô tả các trạng thái trong không gian bài tóan.

+ Trạng thái chuyển tiếp của bài tóan : đó là bước chuyển tiếp từ trạng thái hiện có đến trạng thái mới nhờ thông qua luật hợp lệ như trượt viên ngói đi lên↑, trượt viên ngói đi xuống ↓, trượt viên ngói sang trái ← hoặc trượt viên ngói sang phải →.

Không gian trạng thái cho bài tóan này có thể được biểu diễn bằng đồ thị như hình





Trạng Thái Đích

Vậy, không gian trạng thái cho bài tóan gồm có trạng thái ban đầu, tất cả các trạng thái đạt được từ trạng thái ban đầu đến trạng thái đích, tất các trạng thái chuyển tiếp và trạng thái đích của bài tóan.

Kích thước của không gian trạng thái này đó là số trạng thái đạt được từ trạng thái ban đầu đến trạng thái đích của bài tóan nhờ thông qua tất cả các trạng thái chuyển tiếp.

Ví dụ 3 : Không gian bài tóan ba tu sĩ và ba kẻ ăn thịt người.

Ba tu sĩ và ba kẻ ăn thịt người ở bên này sông muốn qua bên kia sông bằng một chiếc thuyền có sức chở tối đa là 2 thành viên. Bài tóan đặt ra là ở bất kỳ nơi nào bên này sông, trên thuyền hoặc bên kia sông, nếu số tu sĩ ít hơn số kẻ ăn thịt người thì số tu sĩ sẽ bị ăn thịt bởi số kẻ ăn thịt người. Hãy sắp xếp các chuyến thuyền qua lại sông sao cho đưa mọi người sang bên kia sông an tòan ?

Vậy, không gian bài tóan này là gì ?

Giải : Không gian trạng thái cho bài tóan này được mô tả bằng các thành phần như sau :

+ Trạng thái ban của bài tóan : Tất cả mọi người và thuyền ở bên này sông với cấu hình là (MMM, CCC, B), trong đó M là tu sĩ, C là kẻ ăn thịt người và B là thuyền.

+ Trạng thái đích của bài tóan : Tất cả mọi người và thuyền đều được qua bên kia kia sông an tòan, vì thế cấu hình đích bên này sông là (_, _, _).

+ Ràng buộc của bài tóan : Số tu sĩ phải là luôn luôn lớn hơn hoặc bằng số kẻ ăn thịt người ở bất cứ nơi nào bên này sông, trên thuyền hoặc bên kia sông.

+ Trạng thái khác của bài tóan : cấu hình số tu sĩ, số kẻ ăn thịt người và thuyền ở bên này sông hoặc ở bên kia sông.

+ Trạng thái chuyển tiếp của bài tóan : bước dịch chuyển thuyền đưa một vài thành viên qua lại sông.

Bài tóan ba tu sĩ và ba kẻ ăn thịt người được giải gồm các bước như sau :

	Bên này sông	**Bên kia sông**
0. Trạng thái ban đầu	**(MMM,CCC,B)**	**(_ , _ , _)**
1. Hai kẻ ăn thịt người qua bên kia sông.	**(MMM, C, _)**	**(_ , CC, B)**
2. Một kẻ ăn thịt người qua lại bên này sông.	(MMM, CC, B)	(_ , C, _)
3. Hai kẻ ăn thịt người qua bên kia sông.	(MMM, _, _)	(_, CCC, B)
4. Một kẻ ăn thịt người qua lại bên này sông.	(MMM, CC,B)	(_ , CC, _)
5. Hai kẻ tu sĩ qua bên kia sông.	(M, CC, _)	(MM, CC,B)
6. Một tu sĩ và một kẻ ăn thịt người qua lại bên này sông.	(MM,CC,B)	(M, C, -)
7. Hai tu sĩ qua bên kia sông	(_, CC, _)	(MMM, C, B)
8. Một kẻ ăn thịt người qua bên này sông.	(_ , CCC, B)	(MMM, _ , _)
9. Hai kẻ ăn thị người qua bên kia sông.	(_ , C , _ B)	(MMM, CC, B)
10. Một kẻ ăn thịt người qua qua lại bên này sông.	(_, CC, B)	(MMM, C, -)
11. Hai kẻ ăn thịt người qua bên kia sông.	(_ , _ , _) Đích.	(MMM,CCC,B)

Ví dụ 4 : Bài tóan rao số học (Cryarithmetic).

Bài tóan đặt ra là tìm các chữ số từ 0 đến 9 thay thế cho các chữ cái sao cho biểu thức số học tương ứng của nó là đúng điển hình là

FORTY	29786
TEN	850
TEN	850
SIXTY	31486

Vậy, không gian bài tóan này là ?

Ví dụ 5 : Bài tóan hành trình người bán hàng.

Bài tóan hành trình người bán hàng đặt ra là cho bản đồ của n thành phố, tìm đường đi ngắn nhất cho cuộc hành trình của người bán hàng bắt đầu từ một thành phố, viếng thăm mọi thành phố chính xác một lần và trở về lại thành phố bắt đầu.

2.2) Chiến Lược Tìm Kiếm :

Có hai chiến lược tìm kiếm trên không gian trạng thái bài tóan đó là tìm kiếm bắt đầu từ dữ liệu ban đầu về đích và tìm kiếm từ dữ liệu đích lùi về dữ liệu ban đầu.

Tìm kiếm bắt đầu từ dữ liệu ban đầu về đích được gọi là chiến lược tìm kiếm suy diễn tiến và tìm kiếm bắt đầu từ đích lùi về dữ liệu được gọi là chiến lược tìm kiếm suy diễn lùi.

1) Tìm kiếm suy diễn tiến :

Thủ tục tìm kiếm suy diễn tiến trên không gian trạng thái bài tóan được mô tả như sau :

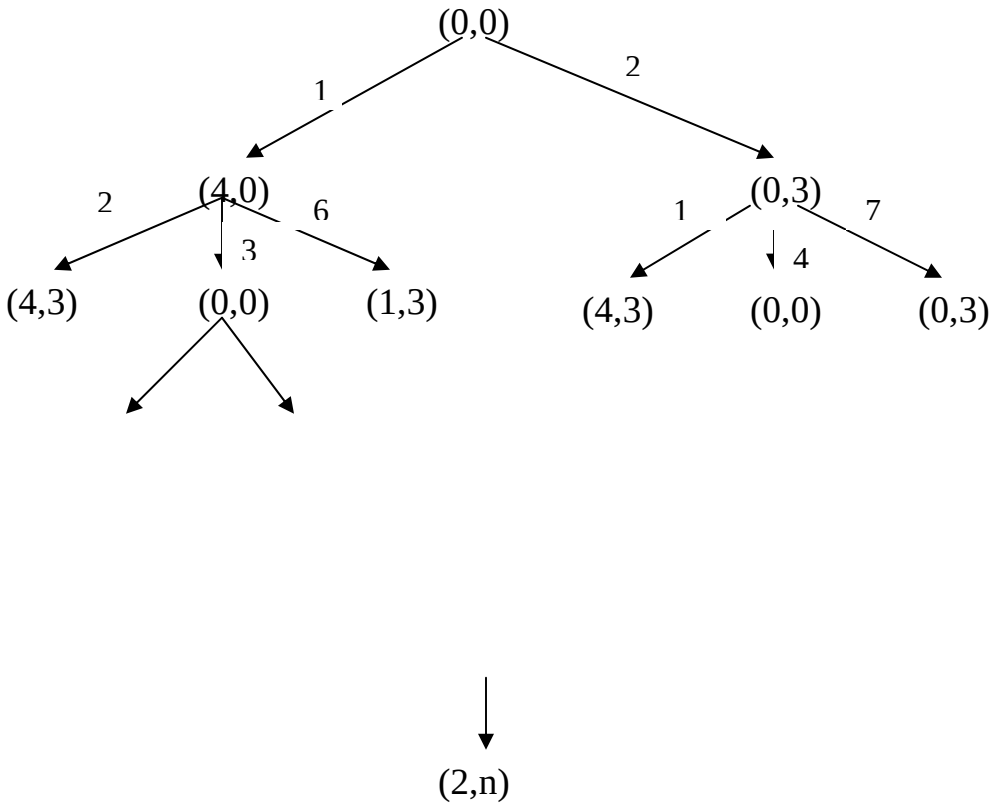
+ Bắt đầu tìm kiếm từ dữ liệu ban của bài tóan.

+ Chọn tất các các luật ứng dụng với vế điều kiện hợp với dữ liệu ban đầu của bài tóan để vế kết luận phát sinh ra các dữ liệu mới.

+ Tại mỗi điểm dữ liệu mới, chọn tất cả các luật ứng dụng với vế điều kiện hợp với dữ liệu mới để vế kết luận phát sinh ra các dữ liệu mới hơn.

+ Thủ tục này được lặp lại cho tất cả các dữ liệu mới cho đến khi dữ liệu đích được tìm thấy.

Ví dụ **: Chiến lược tìm kiếm suy diễn tiến cho bài tóan bình đựng nước trên không gian trạng thái bài tóan được mô tả như như hình**



2) Chiến lược tìm kiếm suy diễn lùi :

Thủ tục tìm kiếm suy diễn lùi trên không gian trạng thái bài tóan được mô tả như sau :

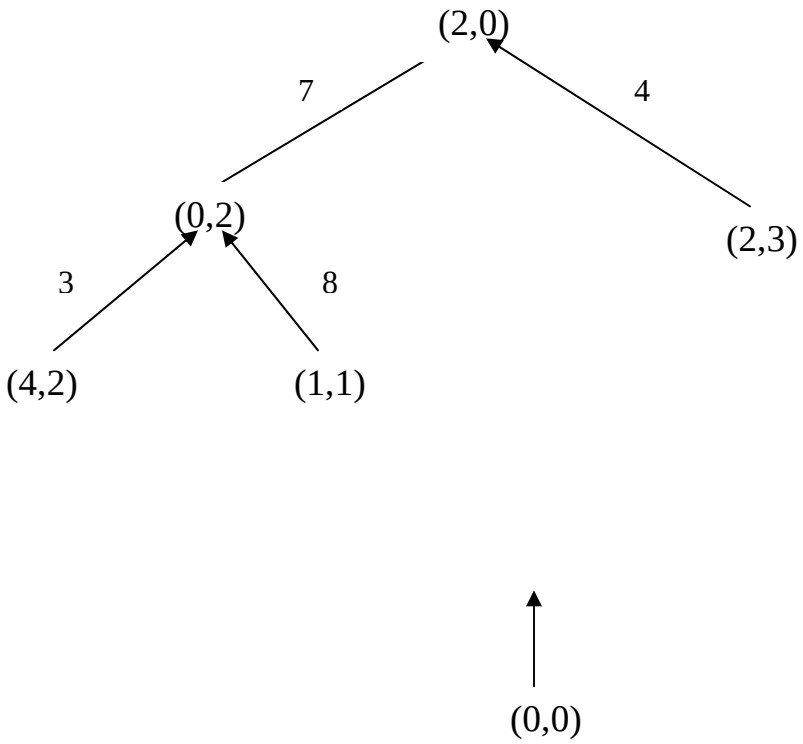
+ Thủ tục bắt đầu tìm kiếm từ dữ liệu đích của bài tóan.

+ Chọn tất cả các luật ứng dụng với vế kết luận hợp với dữ liệu đích, thiết lập dữ liệu ở vế điều kiện phát sinh ra đích làm dữ liệu đích mới.

+ Tại mỗi điểm dữ liệu đích mới, chọn tất cả các luật ứng dụng với vế kết luận hợp với đích mới, thiết lập dữ liệu ở điều kiện làm dữ liệu đích mới hơn.

+ Thủ tục này lặp lại cho tất cả các đích mới cho đến khi nào dữ liệu ban đầu của bài tóan được tìm thấy.

Ví dụ : **Chiến lược tìm kiếm suy diễn lùi trên không gian trạng thái bài tóan bình đựng nước được mô tả như hình**



2.3) Giải Thuật Tìm Kiếm :

Để giải bài tóan sử dụng chiến lược tìm kiếm suy diễn tiến hoặc chiến lược tìm kiếm suy diễn lùi, công việc tìm kiếm là phải tìm một đường từ trạng thái bắt đầu đến trạng thái đích thông qua không gian trạng thái của bài tóan. Công cụ thực hiện việc tìm kiếm này đó là giải thuật. Quá trình tìm kiếm được xem như cây tìm kiếm thông qua không gian trạng thái của bài tóan. Giải thuật bắt đầu từ nút gốc của cây tìm kiếm thăm dò qua các nút khác của cây trong không gian trạng thái của bài tóan. Nếu giải thuật tìm thấy đích thì giải thuật thiết lập đường lời giải bắt đầu từ nút gốc thông qua các nút liên kết đến nút đích của cây. Cấu trúc dữ liệu cho cây tìm kiếm ở đây là ta giả sử nút là một cấu trúc dữ liệu với năm thành phần như sau :

+ Trạng thái trong không gian trạng thái tương ứng với nút của cây.

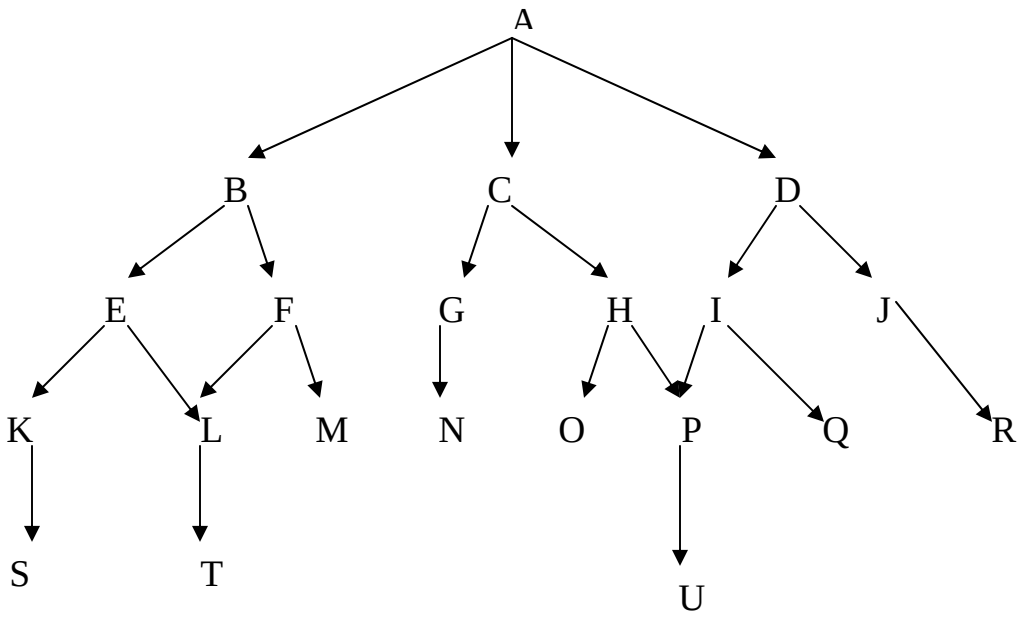
+ Nút trong cây tìm kiếm mà đã phát sinh ra nút mới thì nút này được gọi là nút cha và nút mới được gọi là nút con.
+ Luật hay lệnh hợp lệ được áp dụng để phát sinh ra nút.
+ Số lượng của các nút trên đường từ nút gốc của cây đến một nút ,được gọi là độ sâu của nút đó.
+ Giá chi phí của đường là tính từ nút gốc của cây đến nút đó.

Có hai loại giải thuật tìm kiếm cơ bản đó là giải thuật tìm kiếm theo chiều rộng và giải thuật tìm kiếm theo chiều sâu.

1) Giải thuật tìm kiếm theo chiều rộng ((Breadth First Search):

Giải thuật tìm kiếm theo chiều rộng là giải thuật tìm kiếm mức theo mức của cây. Giải thuật bắt đầu từ nút gốc của cây tìm kiếm qua tất cả các nút ở mức kế theo, nếu nó chưa tìm thấy đích thì nó tiếp tục tìm kiếm qua tất cả các nút ở mức sâu hơn, cứ như thế cho đến khi nó tìm thấy nút đích thì nó dừng thủ tục tìm kiếm và thiết lập đường lời giải bắt đầu từ nút gốc thông qua các nút liên kết đến nút đích.

Giả sử cho không gian trạng thái của bài tóan với các trạng thái được đánh nhãn bằng các chữ cái A, B, C, … được mô tả như hình



Thứ tự của các trạng thái tìm kiếm với giải thuật tìm kiếm theo chiều rộng là
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U.

Giải thuật tìm kiếm theo chiều rộng được trang bị bằng hai danh sách mở Open và đóng Closed, trong đó danh sách Open chứa các trạng thái đang chờ được duyệt và danh sách Closed chứa các trạng thái đã được duyệt qua. Giải thuật được mô tả như sau :

```
Procedure breadth_first_search
Begin
        Open  = [Start];
        Closed = [ ];
    While Open ≠ [ ]
    Begin
        + Loại bỏ nút đầu tiên từ danh sách Open và gọi nút này là X.
        If X = đích Then trả về thành công
        Else begin
        + Phát sinh các con của X dùng các luật áp dụng hợp với X;
        + Đặt X vào danh sách Closed;
        + Loại bỏ các con của X đã có mặt trên Open hoặc Closed;
        + Đặt các on của X chưa có mặt trên Open hoặc Closed vào cuối
        danh sách Open;
        end
    end;
end.
```

2) Giải thuật tìm kiếm theo chiều sâu (Depth First Search) :

Giải thuật tìm kiếm theo chiều sâu là giải thuật tìm kiếm nhánh theo nhánh của cây. Giải thuật bắt đầu từ nút gốc tìm kiếm đến con, cháu, chắc của gốc, nếu giải thuật tìm thấy đích thì dừng thủ tục tìm kiếm và thiết lập đường lời giải từ nút gốc thông qua các nút liên kết đến nút đích; mặt khác nếu giải thuật tìm thấy đường cụt thì nó lùi về tìm kiếm nút anh em.

Cho không trạng thái của bài tóan như hình trên, thứ tự của các trạng thái tìm kiếm với giải thuật tìm kiếm theo chiều sâu là A, B, E, K, S, L, T, F, M, C, G, N, H, O, P, U, D, I, Q, J, R.

Giải thuật cũng được trang bị bằng hai danh sách mơ Open và đóng Closed giống như giải thuật tìm kiếm theo chiều rộng. Giải thuật được mô tả như sau :

```
Procedure depth_first_search
Begin
        Open  = [Start];
        Closed = [ ];
    While Open ≠ [ ]
    Begin
        + Loại bỏ nút đầu tiên từ danh sách Open và gọi nút này là X.
        If X = đích Then trả về thành công
        Else begin
        + Phát sinh các con của X dùng các luật áp dụng hợp với X;
        + Đặt X vào danh sách Closed;
        + Loại bỏ các con của X đã có mặt trên Open hoặc Closed;
        + Đặt các on của X chưa có mặt trên Open hoặc Closed vào đầu
        danh sách Open;
        end
    end;
end.
```

3) Giải thuật tìm kiếm truyền lùi (Back Tracking search) :

Một giải thuật tìm kiếm khác được gọi là giải thuật tìm kiếm truyền lùi, cách tìm kiếm đích của giải thuật này cũng giống như cách tìm kiếm đích của giải thuật tìm kiếm theo chiều sâu. Giải thuật được trang bị bằng ba danh sách N, S và D, trong đó danh sách N chứa các trạng thái đang chờ sẽ được duyệt qua, danh sách S chứa các trạng thái đã được duyệt qua trên đường tìm kiếm và D là danh sách chứa các trạng thái của các đường cụt. Khi giải thuật tìm thấy đích, danh sách S được thiết lập với các trạng thái liên kết nhau từ nút gốc đến nút đích đó là đường lời giải của bài tóan. Giải thuật được mô tả như sau :

```
Function    backtracking
```

```
Begin
    N = [Start];
    S = [Start];
    D = [ ];
    C = Start;
    While N ≠ [ ]
    Begin
        If C = đích then return (S)
        Elseif  C không có thừa kế
            ( Không kể các thừa kế đã có mặt trên N, S hoặc D)
        begin
            while S ≠ [ ] và C là phần tử đầu tiên của S
            begin
                + Đặt C vào đầu danh sách D.
                + Loại bỏ nút đầu tiên của S.
                + Loại bỏ nút đầu tiên của N.
                + Đặt C = phần tử đầu tiên của N.
            end
        Đặt C vào đầu danh sách S.
        end
        Else
        begin
            + Khai triển các thừa kế của C dùng các luật ứng hợp với C.
            + Loại bỏ tất cả các thừa kế của C đã có mặt trên N, S, hoặc D.
            + Đặt các thừa kế của C chưa có mặt trên N, S, hoặc D vào đầu
            danh sách N.
            + Đặt C = phần tử đầu tiên của N.
            + Đặt C vào đầu danh sách S.
        end
    end;
end.
```

2.4) **Tìm Kiếm Heuristic** :

1) Heuristic là gì ?

Tri thức điều khiển của bài tóan còn được gọi là heuristic. Heuristic là luật chủ chốt điều khiển thuật tóan tìm kiếm bám theo đường có các trạng thái tốt nhất

để đạt đến đích. Heuristic có thể được thể hiện dưới dạng luật hoặc dưới dạng hàm số. Nếu nó được thể hiện dưới dạng luật thì nó được gọi là luật heuristic và nếu nó được thể hiện dưới dạng hàm thì nó được gọi là hàm heuristic.

Heuristic còn được gọi là tri thức nông cạn của bài tóan vì nó chỉ đóan bắt trạng thái tốt nhất ở bước kế theo trong quá trình giải quyết vấn. Do đó heuristic đôi lúc không thể đảm bảo tìm thấy lời giải tốt nhất nhưng hầu hết nó đảm bảo tìm thấy lời giải tương đối tốt nhất.

Nếu ta định nghĩa h(n) là hàm heuristic tại trạng thái n thì h(n) là một ước lượng tính từ trạng thái n đến trạng thái đích của bài tóan. Trạng thái nào có heuristic nhỏ nhất đó là trạng thái tốt nhất được chọn để tiếp diễn quá trình tìm kiếm.

+ Nếu trạng thái n không dẫn đến đường cụt thì heuristic của nó là h(n) >= 0.
+ Nếu trạng thái n dẫn đến đường cụt thì heuristic của nó là h(n) = ∞.
+ Nếu trạng thái n dẫn đến trạng thái đích của bài tóan thì heuristic của nó là h(n) = 0.

Ví dụ 1 : Cho bài tóan trò chơi 8 số với cấu hình trạng thái ban đầu và trạng thái đích như hình vẽ

2	8	3
1	6	4
7		5

Trạng Thái Ban Đầu

1	2	3
8		4
7	6	5

Trạng Thái Đích

Heuristic của bài tóan này là số các viên ngói đặt không đúng chổ tại mỗi trạng thái n so với trạng thái đích của bài tóan. Trạng thái nào có số viên ngói đặt không đúng chổ ít nhất đó là trạng thái tốt nhất được chọn để tiếp diễn quá trình tìm kiếm. Khi thuật tóan đạt đến đích, heuristic của bài tóan tiến đến zero.

2) Giải thuật tìm kiếm Best_First_Search :

Một trong các giải thuật tìm kiếm sử dụng heuristic đó là giải thuật Best_first_search. Giải thuật được trang bị bằng hai danh sách mở Open và đóng Closed cũng giống như giải thuật tìm kiếm theo chiều rộng và chiều sâu. Giải thuật bắt đầu tìm kiếm với nút gốc của cây, khai triển các thừa kế của gốc nhờ thông qua các luật ứng dụng, ước lượng heuristic cho tất cả các nút con của gốc, chọn nút có heuristic nhỏ nhất để đến viếng thăm và tháo bỏ tất cả các nút còn lại. Thủ tục này

được lặp lại cho tất cả các nút viếng thăm cho đến khi nào trạng thái đích của bài tóan được tìm thấy. Cách tìm kiếm này tạo ra một đường liên kết bám theo các trạng thái có thông tin heuristic nỏ nhất đó là các trạng thái được đánh giá là tốt nhất để đạt đến đích.

Giải thuật được mô tả như sau :

```
Procedure      best_first_search
Begin
        Open = [Start];
        Closed = [ ];
    While Open ≠ [ ]
    Begin
        + Lấy nút đầu tiên của danh sách Open , gọi nút này là X và loại bỏ X
        khỏi danh sách Open.
        + If X = đích Then return(success)
        Else
        Begin
        + Khai triển các thừa kế của X nhờ thông qua các luật ứng dụng.
        Cho mỗi thừa kế của X thực hiện một trong các trường hợp như sau :
        Case : Thừa kế chưa xuất hiện trên danh sách Open hoặc Closed.
                + Ước lượng heuristic cho thừa kế.
                + Đặt thừa kế vào danh sách Open.
        Case : Thừa kế đã có mặt trên danh sách Open.
                + Ước lượng heuristic cho thừa kế.
                + Nếu trạng thái mới xuất hiện là tốt hơn trạng thái cũ đã xuất
        hiện trên Open thì loại bỏ cũ khỏi danh sách Open và đặt mới vào danh
        sách Open; mặt khác giữ lại trạng thái cũ ở danh sách Open và loại bỏ
        trạng thái mới xuất hiện.
        Case : Thừa kế đã có mặt trên danh sách Closed.
                + Ước lượng heuristic cho thừa kế.
                + Nếu trạng thái mới xuất hiện là tốt hơn trạng thái cũ đã có mặt
        sẵn trên Closed thì loại bỏ cũ khỏi danh sách Closed và đặt mới vào
        danh sách Open; mặt khác giữ lại trạng thái cũ ở danh sách Closed và
        loại bỏ trạng thái mới xuất hiện.
        End
    + Đặt X vào danh sách Closed.
```

+ Sắp xếp lại các trạng thái trong danh sách Open theo thứ tự từ đầu danh sách đến cuối danh sách tương ứng với trạng thái tốt nhất đến trạng thái xấu nhất.

End;

End.

3) Hàm đánh giá heuristic :

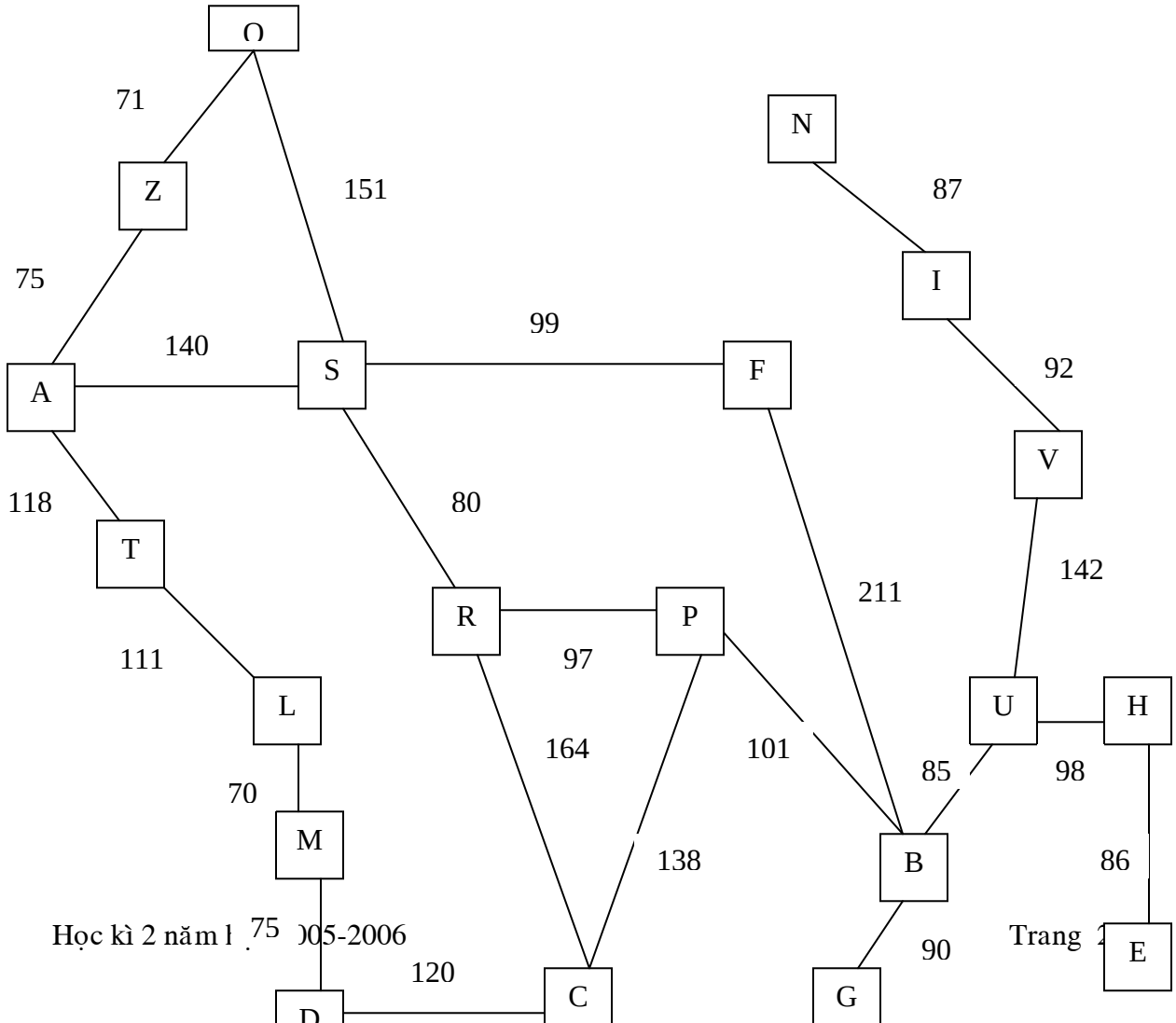
Giả sử quá trình tìm kiếm với thông tin heuristic trên khong gian bài tóan có hai hoặc nhiều trạng thái xuất hiện có cùng heuristic, trong trường hợp này, trạng thái nào là gần gốc nhất của cây đó là trạng thái tố nhất. Để đánh giá đầy đủ thông tin heuris cho các trường hợp như vậy, một hàm đánh giá heuristic được thiết lập gồm hai thành phần đó là

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

trong đó, h(n) là hàm heuristic tại mỗi trạng thái n và g(n) là hàm chi phí đo từ trạng thái gốc của cây đến nút trạng thái n.

Vì vậy, quá trình tìm kiếm, trạng thái nào có f(n) là nhỏ nhất đó là trạng thái tốt nhất được chọn để tiếp diễn tìm kiếm.

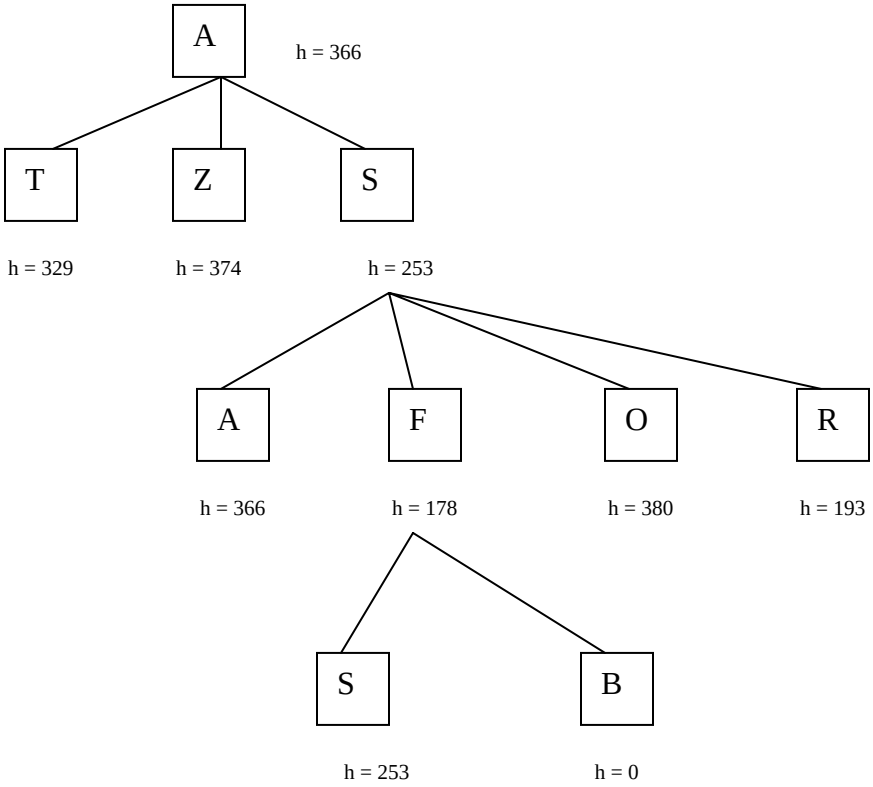
Ví dụ : Cho bản đồ của các thành phố như hình vẽ



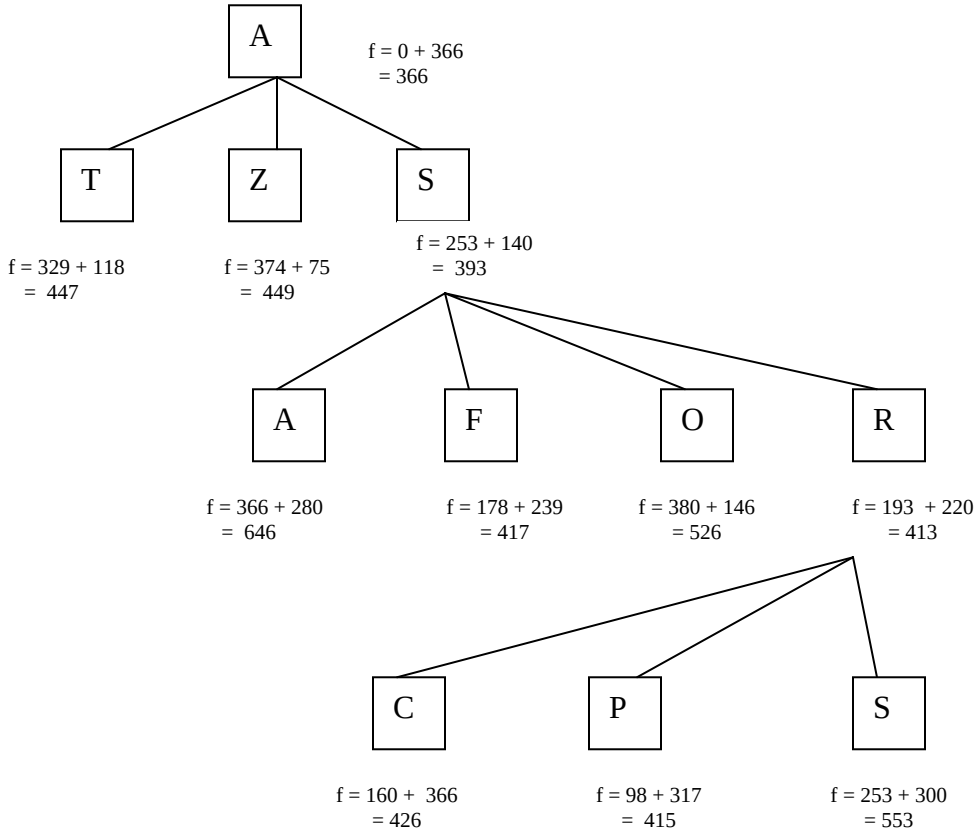
Trong đó, khỏang cách thực sự giữa các thành phố được đánh nhãn trên bản đồ và khỏang cách đường chim bay giữa các thành phố đến thành phố B được liệt kê như bảng

Khỏang các đường chim bay từ các thành phố đến thành phố B	
A	366 Km
B	0 Km
C	160 Km
D	242 Km
E	161 Km
F	178 Km
G	77 Km
H	151 Km
I	226 Km
L	244 Km
M	241 Km
N	234 Km
O	380 Km
P	98 Km
R	193 Km
S	253 Km
T	329 Km
U	80 Km
V	199 Km
Z	374 Km

Giải thuật tìm kiếm sử dụng thông tin heuristic tìm đường đi ngắn nhất từ thành phố A đến thành phố B được mô tả như hình vẽ



Giải thuật tìm kiếm sử dụng thông tin hàm đánh giá heuristic tìm đường đi ngắn nhất từ thành phố A đến thành phố B được mô tả như hình vẽ



2.5) Bài Tóan Ràng Buộc :

Ràng buộc là mối quan hệ giữa hai hoặc nhiều đối tượng. Bài tóan thỏa mãn ràng buộc là tìm các giá trị cho các đối tượng sao cho thỏa mãn nghiệm số của bài tóan đặt ra.

Ví dụ : Cho bài tóan hệ của hai phương trình tuyến tính là

$$x + y = 2$$
$$3x - y = 2$$

trong đó mỗi phương trình tuyến được gọi là một ràng buộc của bài tóan. Bài tóan thỏa mãn ràng buộc là tìm các giá trị cho các đối tượng x và y sao cho thỏa mãn nghiệm của hệ phương trình.

Xét bài tóan cộng rao số học (Cryarithmetic) ràng buộc điển hình là

$$\begin{array}{r} SEND \\ +\ MORE \\ \hline MONEY \end{array}$$

Bài tóan đặt ra làtìm các chữ số từ 0 đến 9 gán cho các đối tượng chữ cái tương ứng sao cho biểu thức số học của chúng là đúng.

Bài tóan có các ràng buộc là

+ Hai chữ cái là không cùng một chữ số
+ Các ràng buộc khác của bài tóan là

$S + M + C_3 = O$ nhớ C_4.
$E + O + C_2 = N$ nhớ C_3.
$N + R + C_1 = E$ nhớ C_2
$D + E = Y$.

Trạng thái ban đầu của bài tóan là

S = ?	C_1 = ?
E = ?	C_2 = ?
N = ?	C_3 = ?
D = ?	C_4 = ?
M = ?	
O = ?	
R = ?	
Y = ?	

Trạng thái đích của bài tóan là tìm các chữ số từ 0 đến 9 gán cho các đối tượng chữ cái sao cho biểu thức số học của chúng là thỏa mãn ràng buộc của bài tóan đặt ra điển hình là với S = 9, M = 1, O = 0, E = 5, N = 6, R = 8, D = 7 và Y = 2.

Lời giải của bài tóan có thể được tìm thấy là

$$\begin{array}{r} SEND \\ +\ MORE \\ \hline MONEY \end{array} \qquad \begin{array}{r} 9567 \\ +1085 \\ \hline 10652 \end{array}$$

Chương 3 : Hệ Chuyên Gia

3.1) Hệ chuyên gia là gì ?

Hệ chuyên gia là một chương trình cơ sở tri thức làm việc giống như một chuyên gia con người. Hệ chuyên gia các các đặc điểm như sau :

+ Tách tri thức của bài tóan khỏi cơ chế điều khiển : Hai thành phần quan trọng nhất của hệ chuyên gia đó là cơ sở tri thức và bộ máy suy diễn. Hai thành phần này tách biệt nhau trong hệ chuyên.

+ Tri thức chuyên gia : Tri thức giải bài tóan trong hệ chuyên gia là tri thức thu thập từ người chuyên gia.

+ Tập trung nguồn chuyên gia : Người chuyên gia chỉ có khả năng giải quyết các vấn đề trong lĩnh vực chuyên môn của họ, còn các vấn đề ngòai lĩnh vực chuyên mô này , họ không có khả năng. Giống như cách giải quyết vấn đề của người chuyên gia, hệ chuyên gia chỉ giải quyết được các vấn đề trong lĩnh vực hẹp chuyên môn.

+ Xử lý tri thức bằng ký hiệu : Tri thức giải bài tóan trong hệ chuyên gia được mã hóa bằng ký hiệu và xử lý những ký hiệu này trên cơ sở lập luận logic.

+ Xử lý tri thức với heuristic : Người chuyên gia có rất nhiều kinh nghiệm giải quyết vấn đề trong lĩnh vực chuyên môn của họ. Với kinh nghiệm này giúp họ giải quyến vấn đề rất nhanh. Giống như cách giải quyết vấn đề của người chuyên gia, các hệ chuyên gia hầu hết đều sử dụng thông tin heuristic thu thập được từ kinh nghiệm của người chuyên gia giúp hệ giải quyết vấn đề nhanh nhất và hiệu quả nhất.

+ Xử lý tri thức không chắc chắn : Hơn 80% ứng dụng trong thực tế không thể giải quyết được bằng các phương pháp lập luận chắc chắn. Hệ chuyên gia có thể giải quyết được những ứng dụng này nhờ vào các phương pháp xử lý tri thức không chắc chắn.

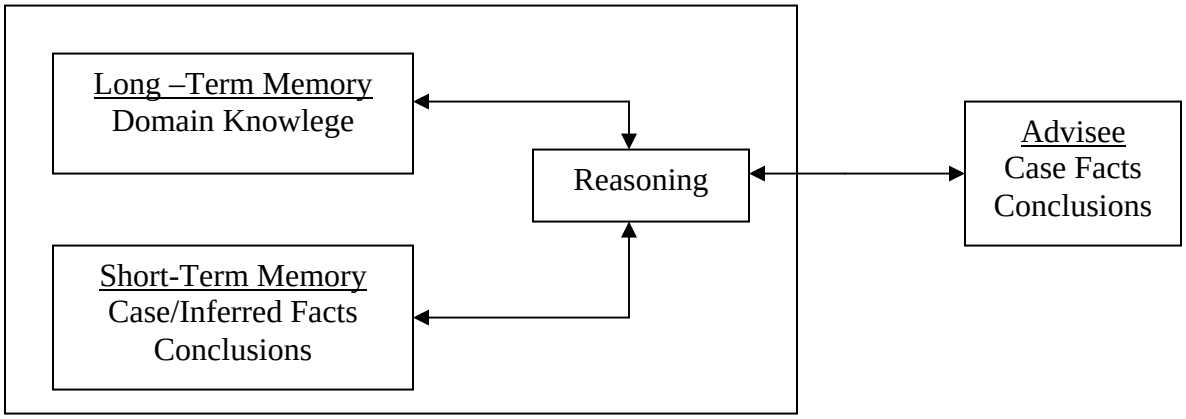
+ Bài tóan giải được : Hệ chuyên gia chỉ giải được bài tóan nào mà người chuyên gia giải được.

+ Mức phức tạp vừa phải : Không nên thiết kế một hệ chuyên gia để giải quyết vấn đề quá đơn giản và cũng không nên mong đợi hệ chuyên gia có thể giải quyết vấn đề quá phức tạp ngòai khả năng giải quyết vấn đề của người chuyên gia.

+ Chấp nhận sai lầm : Người chuyên gia giải quyết vấn đề đôi lúc cũng mắc phải sai lầm, vì thế ta phải chấp nhận một số rủi ro khi sử dụng hệ chuyên gia.

3.2) Cấu trúc hệ chuyên gia :

Xem xét người chuyên gia giải quyết vấn với miền tri thức của họ lưu trữ trong vùng nhớ dài hạn và quá trình lý giải với các sự kiện được phát sinh lưu trữ trong vùng nhớ ngắn hạn như hình vẽ



Nguyên tắc làm việc của người chuyên gia như sau :

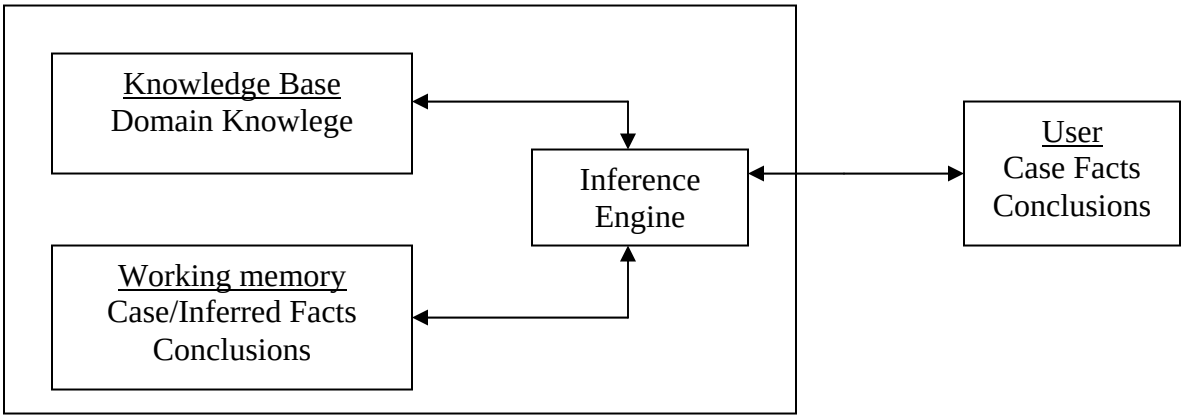
+ Người tham vấn tham vấn người chuyên gia về bài tóan, các sự kiện ban đầu của bài tóan được đưa đến lưu trữ trong vùng nhớ dài hạn.

+ Bộ máy suy diễn của người chuyên gia liên kết các sự kiện trong vùng nhớ ngắn hạn với tri thức giải bài tóan sẵn có trong vùng nhớ dài hạn để suy diễn ra các sự kiện mới.

+ Các sự kiện mới này được đưa vào lưu trữ trong vùng nhớ ngắn hạn.

+ Thủ tục này được lặp lại cho đến khi kết luận của bài tóan được tìm thấy.

Giống như cơ chế làm việc của người chuyên gia, cấu trúc hệ chuyên gia được mô tả như hình



+ Cơ sở tri thức : là phần của hệ chuyên gia chứa miền tri thức. Công việc của ta được xem như là người kỹ sư tri thức lấy tri thức giải bài tóan từ người chuyên gia và mã hóa nó trong vùng cơ sở tri thức.

+ Bộ nhớ làm việc : là phần của hệ chuyên gia chứa các sự kiện về bài tóan được khám phá để dẫn đến kết luận.

+ Bộ máy suy diễn : Hệ chuyên gia mô hình hóa quá trình xử lý lý giải như con người. Vì thế bộ máy suy diễn đó chính là bộ xử lý trong hệ chuyên gia hợp các sự kiện được chứa trong vùng nhớ làm việc và miền tri thức được chứa trong vùng cơ sở tri thức để dẫn đến kết luận về bài tóan.

3.3) Thiết Kế Hệ Chuyên Gia :

Có hai cách giải quyết vấn đề trong các hệ chuyên gia đó là giải quyết vấn đề theo hướng thuận và giải quyết vấn đề theo hướng nghịch. Hệ chuyên gia được thiết kế để giải quyết vấn đề theo hướng thuận được gọi là hệ chuyên gia suy diễn tiến và hệ chuyên gia được thiết kế để giải quyết vấn đề theo hướng nghịch được gọi là hệ chuyên gia suy diễn lùi.

1) **Hệ chuyên gia suy diễn tiến** :

Hệ chuyên gia suy diễn tiến là hệ chuyên gia giải quyết vấn đề bắt đầu lý giải từ dữ liệu ban đầu của bài tóan lập luận trên cơ sở logic để khám phá các sự kiện mới dẫn đến kết luận về bài tóan. Để thiết kế một hệ chuyên gia suy diễn tiến bao gồm các bước được mô tả như sau :

Bước 1 : Định nghĩa vấn đề.

Bước này gồm có phân tích vấn đề để nắm bắt hướng giải quyết vấn đề chi tiết và cụ thể đó là phải xác định được dữ liệu vào ra của hệ thống và phương thức xử lý số liệu vào ra của hệ thống.

Bước 2 : Định nghĩa dữ liệu vào của hệ thống.

Định nghĩa dữ liệu vào của hệ thống sao cho đầu từ dữ liệu ban đầu của bài tóan hợp với vế điều kiện của luật suy diễn thứ nhất để dữ liệu đích của nó là dữ liệu vào cho luật kế theo..

Bước 3 : Định nghĩa cấu trúc điều khiển của hệ thống.

Cấu trúc điều khiển dữ liệu suy diễn tiến của hệ thống đó là cơ sở luật suy diễn tiến bao gồm tất cả các luật mô tả tổng quát cách giải bài tóan được thể hiện dưới dạng luật If Then với vế điều kiện của luật đầu tiên hợp với dữ liệu ban đầu của bài tóan để vế phải của luật phát sinh ra đích thứ nhất, vế điều kiện của luật thứ 2 hợp với đích thứ nhất để vế phải của luật phát sinh ra đích thứ 2 và cứ như thế cho đến luật thứ n mà vế kết luận của nó đạt đến lời giải cuối cùng.

Bước 4 : Mã hóa cơ sở tri thức.

Cơ sở tri thức gồm cơ sở luật và cơ sở dữ liệu. Các thành phần này phải được mã hóa nhờ các phương pháp biểu diễn tri thức như logic vị từ, khung.

Bước 5 : Thử nghiệm hệ thống.

Cho số liệu vào, quá trình xử lý của hệ thống cho số liệu ra với nhiều tình huống khác nhau bao trùm cả không gian vào.

Bước 6 : Thiết kế hệ thống giao diện người xử dụng hệ chuyên gia.

Bước 7 : Mở rộng hệ thống.

Mở rộng cơ sở tri thức của hệ sao cho giải quyết bài tóan càng linh hoạt, càng mềm dẽo là càng tốt đó là quá trình cải tiến hoặc thêm bớt luật suy diễn và cơ sở dữ liệu.

Bước 9 : Đánh giá hệ thống.

Bước này đưa hệ thống vào thou nghiệm trong các trường hợp thực kế để rút ra kết luận đánh giá chất lượng vận hành của hệ thống đáng tin cậy hoặc chưa đáng tin cậy.

Ví dụ 1 : Thiết kế hệ chuyên gia suy diễn tiến cố vấn sinh viên học tập.

+ Định nghĩa vấn đề : Bài tóan đặt ra ra là thiết kế hệ chuyên gia suy diễn tiến cố vấn sinh viên học tập giải quyết các vấn đề như sau :

1- Giải quyết các môn học mà sinh viên đã thi đậu cho qua.
2- Xử lý các môn học mà sinh viên được đặt cách cho qua.
3- Xử lý các môn học có các môn học tiên quyết.
4- Xử lý các môn học mà sinh viên được phép đăng ký học trong mỗi học kỳ.

+ Định nghĩa dữ liệu vào : Dữ liệu vào của bài tóan gồm có

1- Các môn học bắt buộc.
2- Các môn học tự chọn.
3- Các môn học có các môn học tiên quyết.
4- Các môn học mà sinh viên đã học xong.
5- Các môn học cho phép sinh viên được đăng ký trong mỗi học kỳ.

+ Cấu trúc điều khiển dữ liệu suy diễn tiến của hệ thống : Để xử lý số liệu vào ra của hệ thống, cơ sở luật của hệ thống được thiết lập gồm các luật là

Luật 1 : Nếu X là môn học mà sinh viên đã thi đậu cho qua thì sinh viên đã học xong môn học với X.

Luật 2 : Nếu X là môn học mà sinh viên đã được đặt cách cho qua thì sinh viên đã học xong môn học với X.

Luật 3 : Nếu sinh viên đã học xong môn học với X và Q là danh sách chứa các môn học mà sinh viên đã học xong thì Q chứa X.

Luật 4 : Nếu X có môn học tiên quyết Y thì môn học tiên quyết của X là Y.

Luật 5 : Nếu X có môn học tiên quyết Y và Y có môn học tiên quyết Z thì môn học tiên quyết của X là Z.

Luật 6 : Nếu môn học tiên quyết của X là Y và P là danh sách chứa các môn học tiên quyết thì của X P phải chứa Y.

Luật 7 : Nếu Q là danh sách chứa các môn học mà sinh viên đã học xong với X, P là danh sách chứa các môn học tiên quyết của X và P là tập con của Q thì sinh viên đã học xong tất cả với các môn học học tiên quyết của X.

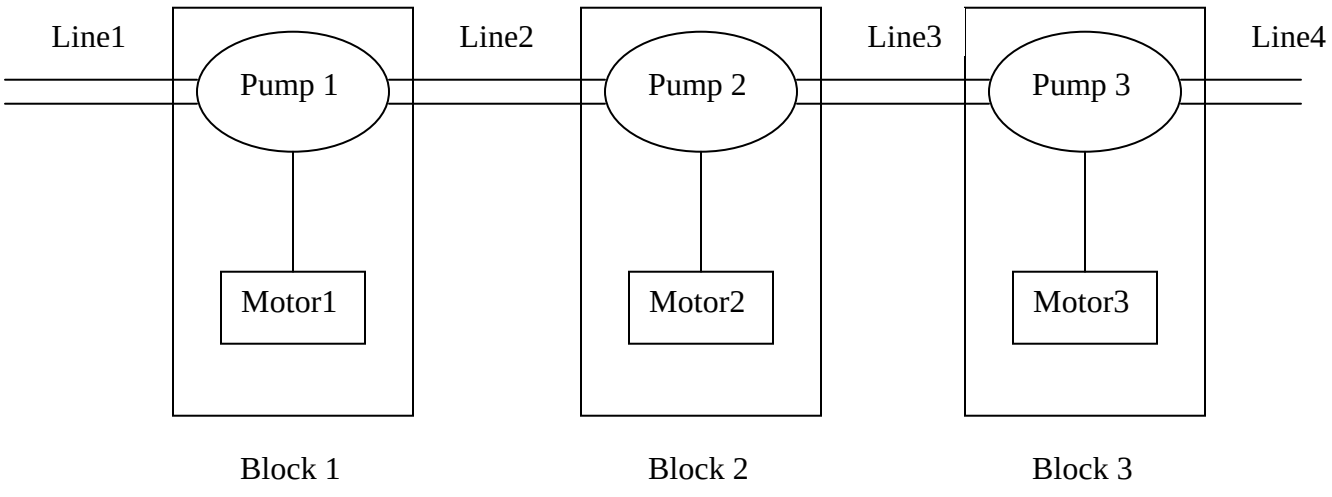
Luật 8 : Nếu X là môn học bắt buộc, sinh viên chưa học xong với X, sinh viên đã học xong tất cả với các môn học tiên quyết của X và X là môn học cho phép sinh viên đăng ký học trong học kỳ thì cho phép sinh viên đăng ký môn học với X.
Luật 9 : Nếu X là môn học tự chọn, sinh viên chưa học xong với X , sinh viên đã học xong tất cả với các môn học tiên quyết của X và X là môn học cho phép sinh viên đăng ký học trong học kỳ thì cho phép sinh viên đăng ký môn học với X.
+ Mã hóa cơ sở tri thức : sau đây là một ví dụ điển hình mã hóa cơ sở tri thức gồm cơ sở dữ liệu và cơ sở luật.

- Các môn học bắt buộc được mã hóa bằng ký hiệu điển hình là
  req("intro to computing").
  req("data structures").
  req("assembler").
  req("operating systems").
  Vân và vân vân.
- Các môn học tự chọn được mã hóa bằng ký hiệu điển hình là
  elec("information systems").
  elec("compilers").
  elec("algorithm analysis").
  Vân và vân vân.
- Các môn học tiên quyết được mã hóa bằng ký hiệu điển hình là
  impreq("data structures","intro to computing").
  impreq("calculus 2","calculus 1").
  impreq("operating systems","assempler").
  Vân và vân vân.
- Các môn học cho phép sinh viên được phép đăng ký học trong học kỳ được mã hóa bằng ký hiệu điển hình là
  given_now("intro to computing").
  given_now("calculus 2").
  Vân và vân vân.
- Các môn học mà sinh viên đã được đặt cách cho qua và thi đậu cho được mã hóa bằng ký hiệu điển hình là
  waived("intro to computing").
  waived("calculus 1").
  passed("data structures").
  passed("assempler").
  passed("calculus 2").

Vân và vân vân.

- Luật 1 được mã hóa bằng ký hiệu điển hình là
  if passed(X) then done_with(X).
- Luật 2 được mã hóa bằng ký hiệu điển hình là
  if waived(X) then done_with(X).
- Luật 3 được mã hóa bằng ký hiệu điển hình là
  if findall(Y, done_with(Y),X)) then all_done_with(X).
- Luật 4 được mã hóa bằng ký hiệu điển hình là
  if impreq(X,Y) then preq(X,Y).
- Luật 5 được mã hóa bằng ký hiệu điển hình là
  if impreq(X,Y) and preq(Y,Z) then preq(X,Z).
- Luật 6 được mã hóa bằng ký hiệu điển hình là
  if findall(Y,preq(X,Y),Z) then all_preq_for(X,Z).
- luật 7 được mã hóa bằng ký hiệu điển hình là
  if all_preq_for(X,Z) and all_done_with(Q) and subset(Z,Q)
  then have_preq_for(X).
- Luật 8 được mã hóa bằng ký hiệu điển hình là
  if req(X) and not(done_with(X)) and given_now(X) and
  have_preq_for(X) then pos_req_course(X).
- Luật 9 được mã hóa bằng ký hiệu điển hình là
  if elec(X) and not(done_with(X)) and given_now(X) and
  have_preq_for(X) then pos_elec_course(X).

Ví dụ 2 : Thiết kế hệ chuyên gia suy diễn tiến để giải quyết bài tóan chẩn đóan sự cố trên một trạm bơm nước như hình vẽ



Trạm bơm nước gồm có ba khối liên kết nhau qua các đường ống, trong đó mỗi khối có một máy bơm và một motor.

Vấn đề đặt ra của bài tóan là phát hiện sự cố vận hành nước trên trạm, nhận dạng khối có sự cố và chẩn đóan các thành phần bị hỏng gây ra sự cố như máy bơm, motor hoặc đường ống bị rỉ nức.

+ Điều kiện phát hiện sự cố vận hành nước trên trạm đó là áp suất ra trên các đường ống của trạm là thấp hơn áp suất vận hành nước bình thường.

+ Điều kiện khối có sự cố vận hành nước trên trạm là áp suất vào trên các đường ống của khối là bình thường trong khi đó áp suất ra trên các đường ống của khối là thấp hơn áp suất bình thường.

+ Các thành phần của khối là motor, máy bơm và đường ống. Khi đã xác định được khối có sự cố vận hành nước thì một trong các thành phần này của khối là nhân tố quyết định gây ra sự cố.

- Điều kiện motor vận hành yếu đó là nguyên nhân gây ra sự cố thì khi đó chỉ số vận hành motor được đọc về từ cảm biến là thấp.
- Điều kiện máy bơm bị hỏng đó cũng là nguyên nhân gây ra sự cố thì khi đó áp suất được đọc về từ cảm biến là áp suất vào của khối bằng áp suất ra của khối.
- Điều kiện đường ống bị rỉ nức đó là nguyên nhân gây ra sự cố thì khi đó áp suất được đọc về từ cảm biến là áp suất vào của khối phải là nhỏ hơn áp suất ra của khối.

Từ việc phân tích bài tóan nói trên, dữ liệu vào của hệ thống là áp suất và chỉ số vận hành của motor.

Giả sử áp suất vận hành nước bình thường của các đường ống là

+ Line 1 = 50 psi
+ Line 2 = 100 psi
+ Line 3 = 150 psi
+ Line 4 = 200 psi

và ta cũng giả sử rằng chỉ số vận hành bình thường của các motor là motor1 = motor2 = motor3 = 1.

Trên cơ sở đó, cấu trúc điều khiển dữ liệu vào ra của hệ thống xử lý hai khối trên trạm gồm các luật được thiết lập như sau :

Luật 1 : if line1 < 50 then line1 = low.

Luật 2 : if line1 >= 50 then line1 = normal.

Luật 3 : if line2 < 100 then line2 = low.
Luật 4 : if line2 >= 100 then line2 = normal.
Luật 5 : if line3 < 150 then line3 = low and display fault detected.
Luật 6 : if line3 >= 150 then line3 = normal.
Luật 7 : if mortor1 < 1 then motor1 = low.
Luật 8 : if motor2 >= 1 then motor1 = normal.
Luật 9 : if mortor2 < 1 then motor2 = low.
Luật 10 : if motor2 >= 1 then motor2 = normal.
Luật 11 : if line1 = normal and line2 = low then block1 = fault .
Luật 12 : if line2 = normal and line3 = low then block2 = fault.
Luật 13 : if block1 = fault and motor1 = low then motor1 = fault and display fault found.
Luật 14 : if block2 = fault and motor2 = low then motor2 = fault and display fault found.
Luật 15 : if block1 = fault and motor1 = normal and line1 pressure = line2 pressure then pump1 = fault and display fault found.
Luật 16 : if block2 = fault and motor2 = normal and line2 pressure = line3 pressure then pump2 = fault and display fault found.
Luật 17 : if block1 = fault and motor1 = normal and line1 pressure < line2 pressure then line2 = fault and display fault found.
Luật 18 : if block2 = fault and motor2 = normal and line2 pressure < line3 pressure then line3 = fault and display fault found.

2) Thiết kế hệ chuyên gia suy diễn lùi :

Hệ chuyên gia suy diễn lùi là hệ xử lý số liệu vào ra bắt đầu từ dữ liệu đích với cấu trúc điều khiển luật suy diễn móc xích lùi về dữ liệu ban đầu của bài tóan. Để thiết kế một hệ chuyên gia móc xích suy diễn lùi gồm các bước là

+ Định nghĩa bài tóan.
+ Định nghĩa các đích của bài tóan.
+ Thiết kế các luật móc xích đích suy diễn lùi giải quyết bài tóan.
+ Mở rộng hệ thống.
+ Cải tiến hệ thống.
+ Thiết kế giao diện người sử dụng hệ chuyên.
+ Đánh giá hệ thống.

Ví dụ : Thiết kế hệ chuyên gia suy diễn lùi tư vấn tài chánh bao gồm các công việc được mô tả như sau :

1) Định nghĩa vấn đề : Bài tóan tư vấn khách hàng về tài chánh nên đầu tư số tiền của họ vào một trong các loại phần vốn đầu tư như tiết kiệm, chứng khóan, công trái và tiết kiệm và thị trường chứng khóan điều đó phải phụ thuộc vào tình trạng bản thân và tình trạng tài chánh của khách hàng. Tình trạng bản thân của khách hàng phụ thuộc vào độ tuổi, công việc làm và tình trạng gia đình. Tình trạng tài chánh của khách hàng phụ thuộc vào tài sản hiện có và tình trạng gia đình. Như vậy, hai nhân tố quan trọng nhất để dẫn đến đích của bài tóan tư vấn khách hàng đầu tư phần tiền của họ vào các khỏan đầu tư đó là tình trạng bản thân và tình trạng tài chánh của khách hàng là ổn định hoặc không ổn định.
2) Định nghĩa các đích của bài tóan : Với bài tóan này, các đích của bài tóan được định nghĩa là
   + Phần vốn đầu tư loại 1 : 100% đầu tư vào tiết kiệm
   + Phần vốn đầu tư loại 2 : 60% thị trường chứng khóan, 30% thị trường công trái và 10% tiết kiệm.
   + Phần vốn đầu tư loại 3 : 20% thị trường chứng khóan, 40% thị trường công trái và 40% tiết kiệm.
   + Phần vốn đầu tư loại 4 : 100% đầu tư vào thị trường chứng khóan.
3) Thiết kế các luật suy diễn đích : Luật suy diễn đích của hệ chuyên gia suy diễn lùi là luật có cấu trúc nhìn từ đích lùi về dữ liệu. Điều đó có nghĩa là đích cuối cùng của bài tóan phải đượcđịnh nghĩa, từ đó nhìn về các nhân tố quyết định để dẫn đến đích và các nhân tố quyết định này được xem như là các đích mới để nhìn về các nhân tố quyết định khác dẫn đến đích mới này. Thủ tục thiết kế luật này được lặp lại cho đến khi nhân tố quyết định là ngõ vào từ dữ liệu ban đầu của bài tóan.

Hệ thống luật móc xích suy diễn đích để giải bài tóan tư vấn khách hàng về tài chánh được thiết lập là :

Luật 1 : Nếu số tiền của khách hàng là nhỏ hơn 1000 dollars thì tư vấn khách hàng nên đầu tư 100% số tiền của họ vào phần vốn đầu tư tiết kiệm.

Luật 2 : Nếu tình trạng bản thân của khách hàng là không ổn định và tình trạng thài chánh của khách hàng là không ổn định thì tư vấn khách hàng nên đầu tư 100% số tiền của họ vào phần vốn đầu tư tiết kiệm.

Luật 3 : Nếu tình trạng bả thân của khách hàng là không ổn định và tình trạng thài chánh của khách hàng là ổn định thì tư vấn khách hàng nên đầu 60% số tiền của họ

vào phần vốn đầu tư chứng khóan, 30% số tiền của họ vào phần vốn đầu tư công trái và 10% số tiền của họ vào phần vốn đầu tư tiết kiệm.

Luật 4 : Nếu tình trạng bản thân của khách hàng là ổn định và tình trạng tài chánh của khách hàng là không ổn định thì tư vấn khách hàng đầu tư 20% số tiền của họ vào phần vốn đầu tư chứng khóan, 40% số tiền của họ vào phần vốn đầu tư công trái và 40% số tiền của họ vào phần vốn đầu tư tiết kiệm.

Luật 5 : Nếu tình trạng bản thân của khách hàng là ổn định và tình trạng thài chánh của khách hàng là ổn định thì tư vấn khách hàng nên đầu tư 100% số tiền của họ vào phần vốn đầu tư chứng khóan.

Luật 6 : Nếu tuổi của khách hàng là lớn tuổi hoặc việc làm của khách hàng là không ổn định thì tình trạng bản thân của khách hàng là không ổn định.

Luật 7 : Nếu tuổi của khách hàng là trẻ tuổi vàviệc làm của khách hàng là ổn định và khách hàng có trẻ con thì tình trạng bản thân của khách hàng là không ổn định.

Luật 8 : Nếu tuổi của khách hàng là trẻ và việc làm của khách hàng là ổn định và khách hàng không có trẻ con thì tình trạng bản thân của khách hàng là ổn định.

Luật 9 : Nếu tuổi của khách hàng là lớn hơn 40 thì tuổi của khách hàng là lớn tuổi.

Luật 10 : Nếu tuổi của khách hàng là nhỏ hơn 40 thì tuổi của khách hàng là trẻ tuổi.

Luật 11 : Nếu thời gian hợp đồng làm việc của khách hàng là hơn 10 năm thì việc làm của khách hàng là ổn định.

Luật 12 : Nếu thời gian hợp đồng làm việc của khách hàng là từ 3 năm đến 10 năm và mức độ sa thải là thấp thì việc làm của khách hàng là ổn định.

Luật 13 : Nếu thời gian hợp đồng làm việc của khách hàng là từ 3 năm đến 10 năm và mức độ sa thải là cao thì việc làm của khách hàng là không ổn định.

Luật 14 : Nếu thời gian hợp đồng làm việc của khách hàng là ít hơn 3 năm thì việc làm của khách hàng là không ổn định.

Luật 15 : Nếu tổng số tài sản của khách hàng là nhỏ hơn tổng số nợ của khách hàng thì tình trạng tài chánh của khách hàng là không ổn định.

Luật 16 : Nếu tổng số tài sản của khách hàng là lớn hơn tổng số nợ của khách hàng và nhỏ hơn 2 lần tổng số nợ của khách hàng và khách hàng có trẻ con thì tình trạng tài chánh của khách hàng là không ổn định.

Luật 17 : Nếu tổng số tài sản của khách hàng là lớn hơn tổng số nợ của khách hàng thì tình trạng tài chánh của khách hàng là ổn định.

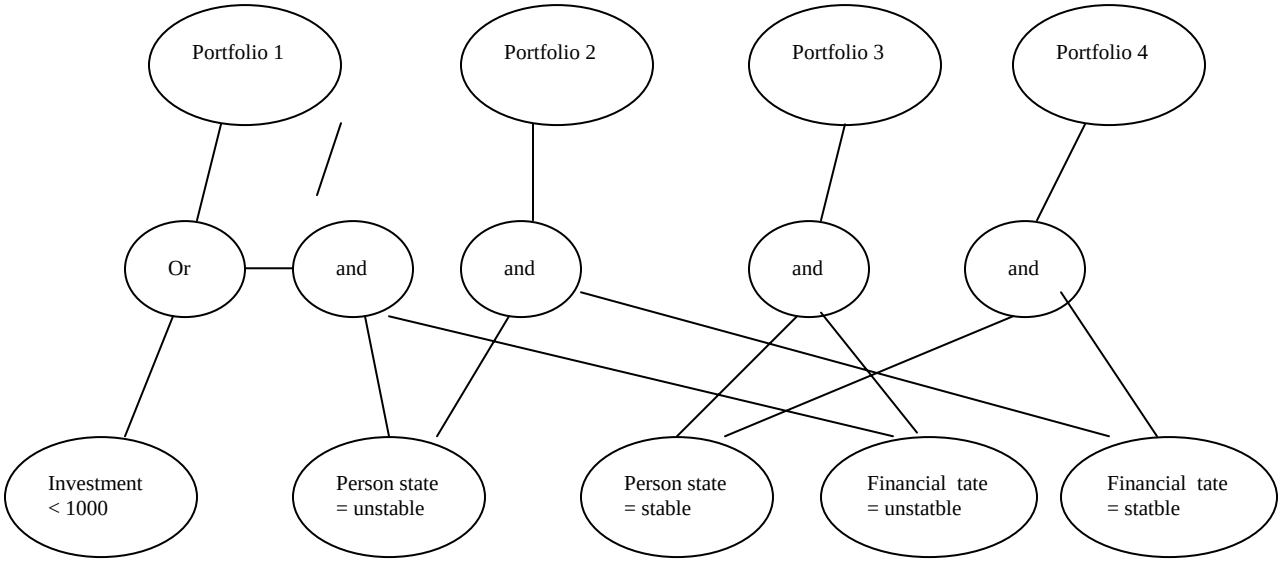
Chạy hệ chuyên gia này với các số liệu vào là

Số tiền đầu tư : 5000 dollars

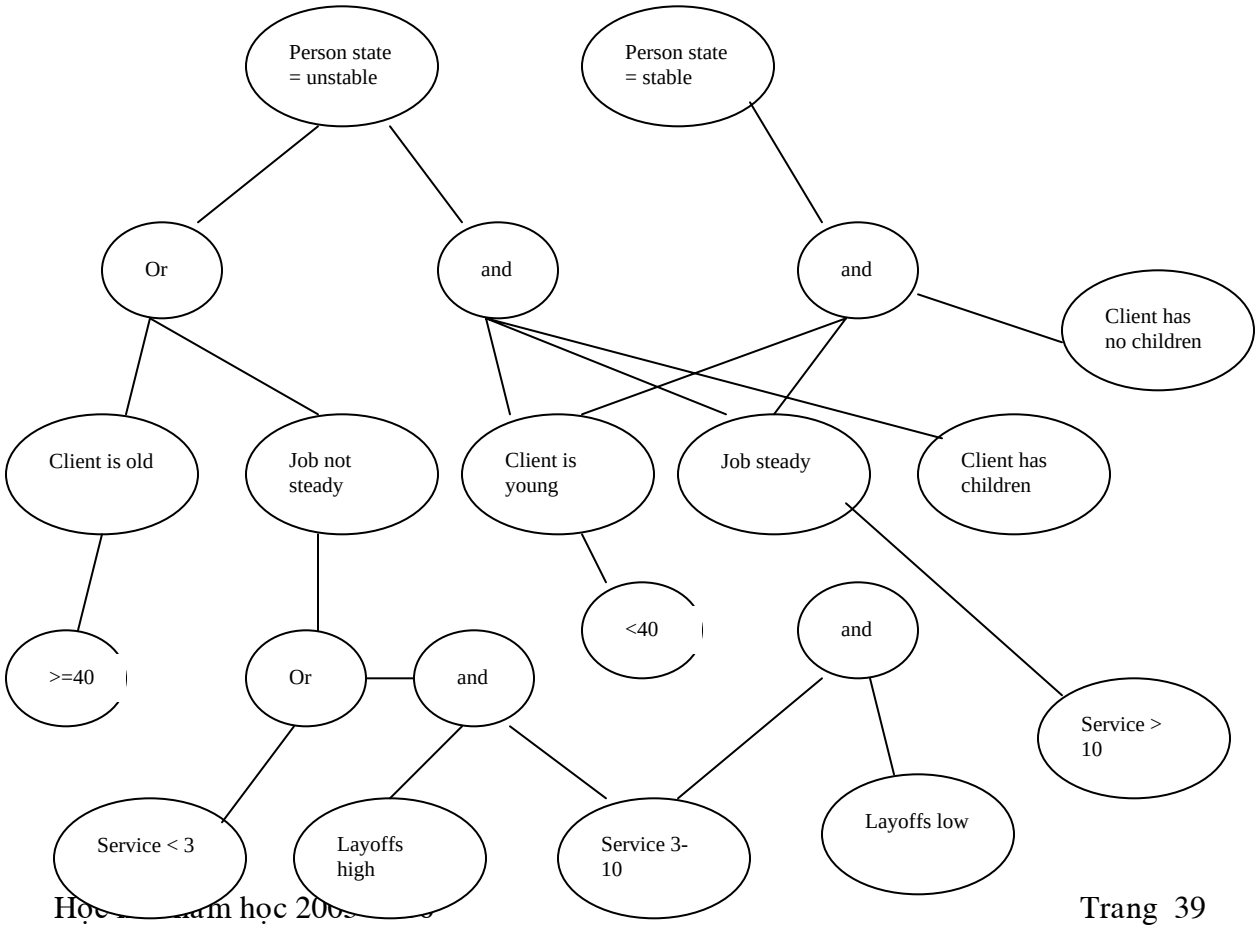
Tuổi của khách hàng : 30

Thời gian hợp đồng làm việc : 5 năm
Có trẻ con không : Có
Tổng số tài sản : 100000 dollars
Tổng số nợ : 20000 dollars.

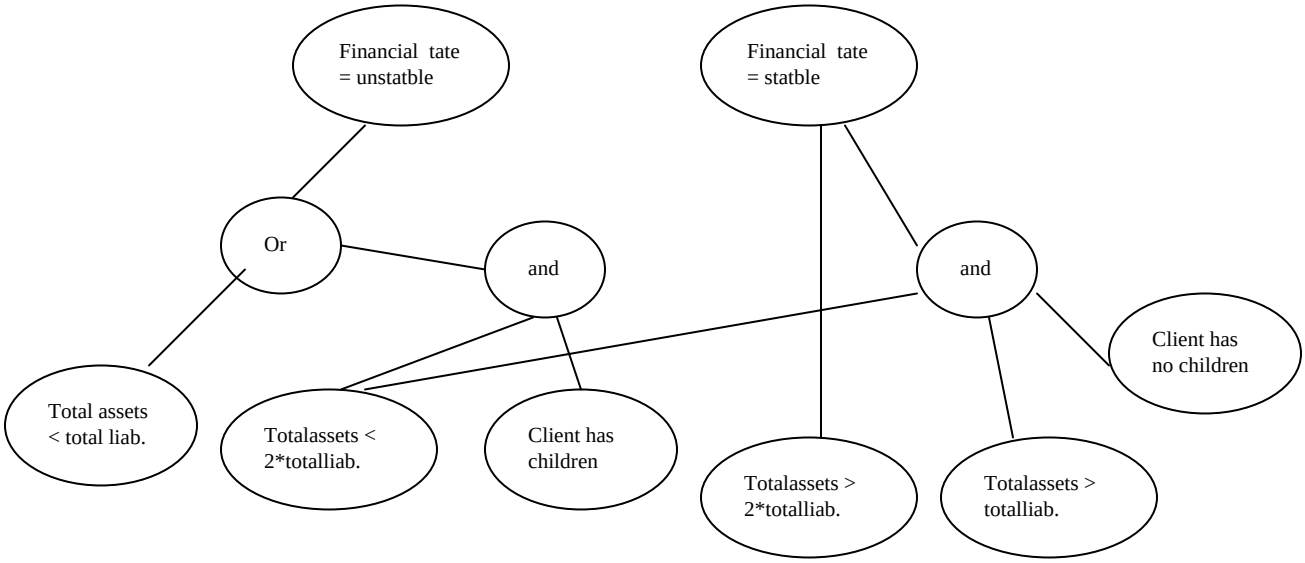
Mạng suy diễn luật đích của hệ chuyên gia suy diễn lùi tư vấn tài chánh được mô tả như hình



Mạng suy diễn tình trạng bản thân của khách hàng được mô tả như hình



Mạng suy diễn tình trạng trạng tài chánh của khách hàng được mô tả như hình



Chương 4 : Các Phương Pháp Biểu Diễn Tri Thức

4.1) Biểu Diễn Tri Thức Là Gì ?

Biểu diễn tri thức là cách thể hiện tri thức trong máy dưới dạng sao cho bài tóan có thể được giải tốt nhất. Biểu diễn tri thức trong máy phải :

+ Thể hiện được tất cả các thông tin cần thiết.
+ Cho phép tri thức mới được suy diễn từ tập các sự kiện và luật suy diễn.
+ Cho phép biểu diễn các nguyên lý tổng quát cũng như các tình huống đặc trưng.
+ Bắt lấy được ý nghĩa ngữ nghĩa phức tạp.
+ Cho phép lý giải ở mức tri thức cao hơn.

Có hai loại tri thức của bài tóan cần phải được biểu diễn đó là tri thức mô tả và tri thức thủ tục.

Tri thức mô tả là loại tri thức mô tả những gì được biết về bài tóan. Loại tri thức này bao gồm sự kiện, đối tượng, lớp của các đối tượng và quan hệ giữa các đối tượng.

Tri thức thủ tục là thủ tục tổng quát mô tả cách giải quyết bài tóan. Loại tri thức này bao gồm thủ tục tìm kiếm và luật suy diễn.

Có ba phương pháp biểu diễn tri thức mô tả cơ bản đó là phương pháp biểu diễn tri thức nhờ logic vị từ, phương pháp biểu diễn tri thức nhờ mạng ngữ nghĩa và phương pháp biểu diễn tri thức bằng khung.

Phương pháp biểu diễn tri thức nhờ logic vị từ đó là lớp biểu diễn sử dụng các biểu thức logic để biểu diễn cơ sở tri thức. Luật suy diễn và thủ tục chứng minh lý giải tri thức này trên cơ sở logic với các yêu cầu bài tóan đặt ra. Tuy nhiên, đó chỉ là một thành phần của biểu diễn logic được trang bị cho công việc ý tưởng lập trình của ngôn ngữ lập trình Prolog.

Phương pháp biểu diễn tri thức nhờ mạng ngữ nghĩa đó là dùng mạng để biểu diễn tri thức như là một cấu trúc dữ liệu, trong đó mối quan hệ giữa các đối tượng được thiết lập thông qua mạng biểu diễn tri thức.

Phương pháp biểu diễn tri thức nhờ frame còn được gọi là ngôn ngữ biểu diễn cấu trúc đó là sự mở rộng của mạng, trong đó mỗi nút của mạng là một cấu trúc dữ liệu chứa các slot với các giá trị của chúng được kèm theo và các thủ tục giải quyết vấn đề thực hiện trên các tác vụ frame này.

4.2) **Biểu Diễn Tri Thức Nhờ Logic Vị Từ** :

1) **Logic đề xuất** :

Logic đề xuất là tập của các đề xuất, trong đó mỗi đề xuất là một phát biểu mà nội dung của nó có thể là đúng hoặc là sai .

Cú pháp của logic đề xuất gồm có ký hiệu chân lý, ký hiệu đề xuất và tóan tử logic.

+ **Ký hiệu chân lý** : Ký hiệu chân lý là hai chữ cái in hoa T và F, trong đó T xác định nội dung của phát biểu là đúng và F xác định nội dung của phát biểu là sai.

+ **Ký hiệu đề xuất** : Ký hiệu đề xuất là các chữ cái in hoa như A, B, C, D, …. được sử dụng để biểu diễn đề xuất.

+ **Tóan tử logic** : Tóan tử logic gồm có các lọai tóan tư như :

- $\wedge$: tóan tử logic liên từ và.
- $\vee$: tóan tử logic giới từ hoặc.
- $\neg$: tóan tử logic phủ định.
- $\rightarrow$: Tóan tử logic kéo nếu.
- $\leftrightarrow$: tóan tử logic tương đương nếu và chỉ nếu.

+ **Câu đề xuất** : Câu đề xuất được định nghĩa như sau :

- Mọi ký hiệu đề xuất và ký hiệu chân lý là một câu. Ví dụ điển hình là T, F, Q, P, hoặc R là một câu.
- Phủ định của một câu là một câu. Ví dụ điển hình là $\neg$P là một câu.
- Tóan tử kết nối liên từ và của hai câu là một câu. Ví dụ điển hình là P$\wedge$Q là một câu.
- Tóan tử kết nối giới từ hoặc của hai câu là một câu. Ví dụ điển hình là P$\vee$Q là một câu.
- Tóan tử kéo theo của một câu cho một câu khác là một câu. Ví dụ điển hình là P$\rightarrow$Q là một câu.
- Sự tương của hai câu là một câu. Ví dụ điển hình là P$\vee$Q = R là một câu.

Tất cả các câu hợp lệ được xem như là các công thức dạng hòan thiện (WFFs).

- Ở biểu thức dạng P$\wedge$Q, trong đó P và Q được gọi là các liên từ.

- Ở biểu thức dạng P∨Q, trong P và Q được gọi là các giới từ.
- Ở biểu thức dạng P→Q, trong đó P được gọi là tiền điều kiện và Q được gọi là kết luận.

Trong câu logic đề xuất, các ký hiệu ( ) và [ ] được sử dụng để nhóm các biểu thức con trong câu.

Ví dụ : Cho công thức ((P∧Q) →R) = ¬P∨¬Q∨R đó là một dạng câu hòan thiện, bởi vì :

- P, Q và R là các đề xuất và vì thế chúng là các câu hòan thiện.
- P∧Q là liên từ của hai câu và vì thế nó là một câu hòan thiện.
- (P∧Q) →R là kéo theo của một câu cho một câu khác và vì thế nó là một câu.
- ¬P và ¬Q là phủ đinh của câu và vì thế chúng là câu.
- ¬P∨¬Q là giới từ của hai câu và vì thế nó là một câu hòan thiện.
- ¬P∨¬Q∨R là giới từ của hai câu và vì thế nó cũng là một câu hòan thiện.
- ((P∧Q) →R) = ¬P∨¬Q∨R là sự tương của hai câu và vì thế nó là một câu hòan thiện.

+ Ngữ nghĩa của logic đề xuất : Ngữ nghĩa của logic đề xuất đó chính là giá trị chân lý của các ký hiệu đề xuất. Giá trị chân lý đúng của một đề xuất được ký hiệu là T và giá trị chân lý sai của một đề xuất được ký hiệu là F.

- Giá trị chân lý của phủ định ¬, ¬P là F nếu P là T và ¬P là T nếu P là F.
- Giá trị chân lý của liên từ ∧ , là T chỉ khi nào giá trị chân lý của cả hai thành phần của nó là T; mặt khác giá trị chân lý của nó là F.
- Giá trị chân lý của giới từ ∨ , là F chỉ khi nào giá trị chân lý của cả hai thành phần của nó là F; mặt khác giá trị chân lý của nó là T.
- Giá trị chân lý của phép kéo theo →, là F nếu giá trị chân lý của vế tiền điều kiện là T và giá trị chân lý của vế kết luận là F; mặt khác giá trị chân lý của nó là T.
- Giá trị chân lý của phép tương đương ↔, là T chỉ khi nào hai thành phần của nó là có cùng giá trị chân lý; mặt khác giá trị chân lý của nó là F.

Cho P, Q và R là các biểu thức đề xuất, các biểu thức sau đây là các biểu thức logic tương đương :

- ¬(¬P) = P.
- (P∨Q) = (¬P→Q).

- Luật de Morgan : $\neg(P \vee Q) = (\neg P \wedge \neg Q)$.
- Luật de Morgan : $\neg(P \wedge Q) = (\neg P \vee \neg Q)$.
- Luật phân bố : $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$.
- Luật phân bố : $P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$.
- Luật giao hóan : $(P \wedge Q) = (Q \wedge P)$.
- Luật giao hóan : $(P \vee Q) = (Q \vee P)$.
- Luật kết hợp : $((P \wedge Q) \wedge R)) = (P \wedge (Q \wedge R))$.
- Luật kết hợp : $((P \vee Q) \vee R)) = (P \vee (Q \vee R))$.
- Luật tương phản : $(P \rightarrow Q) = (\neg Q \rightarrow \neg P)$.

Hai biểu thức logic được gọi là tương đương khi các giá trị chân lý của chúng là giống nhau.

2) **Logic vị từ** :

Logic vị từ là sự mở rộng của logic đề xuất, đôi lúc nó còn được gọi là logic bậc nhất. Trong logic đề xuất, mỗi ký nhiệu đề xuất như P, Q hoặc R được sử dụng để biểu diễn một đề xuất. Với cách biểu diễn này không cho phép ta truy cập các thành phần cá thể trong một đề xuất. Để khắc phục điều này nhờ đến logic vị từ. Cách biểu diễn các đề xuất dùng logic vị từ cho phép ta có thể truy cập các thành phân cá thể trong một đề xuất.

Ví dụ : Cho đề xuất là

It rained on Tuesday.

Cách biểu diễn đề xuất này dùng logic đề xuất là

R = It rained on Tuesday.

Với cách biểu diễn này, ta chỉ xác minh được giá trị chân lý của ký hiệu đề xuất R nhưng ta không thể truy cập các thành phần cá thể trong đề xuất như rained và tuesday đó là tình huống thời tiết và thời gian.

Cách biểu diễn đề xuất trên dùng logic vị từ là

weather(tuesday,rain).

Với cách biểu diễn này, ta có thể truy cập các thành phần cá thể trong đề xuất như tuesday và rain .

Trong cách biểu diễn này, ta cũng có thể thay thế biến X biểu diễn lớp của các đối tượng trong tuần với đạc trưng tuesday điển hình là

weather(X,rain).

+ **Cú pháp của logic vị từ** : gồm có ký hiệu chân lý, ký hiệu vị từ và các phép tóan logic. Vì logic vị từ là sự mở rộng của logic đề xuất, do đó ký hiệu chân lý và các

phép tóan logic của logic vị từ và logic đề xuất là giống nhau. Sự khác nhau giữa hai lọai logic này là ký hiệu vị từ và ký hiệu đề xuất.

Ký hiệu vị từ gồm có hằng vị từ, biến vị từ, hàm vị từ, vị từ và vị từ định lượng.

- Hằng vị từ : là chuỗi của các chữ cái in thường dùng để biểu diễn tên riêng hoặc thuộc tính riêng của đối tượng.

Ví dụ : john, tree, tall, blue là các ký hiệu hằng vị từ hợp lệ.

- Biến vị từ : là chuỗi của các chữ cái với ít nhất chữ cái đầu tiên của chuỗi phải là chữ cái in hoa dùng để biểu diễn lớp của các đối tượng.

Ví dụ : X là biến vị từ dùng để biểu diễn lớp của các đối tượng ngày trong tuần hoặc Breaker là biến vị từ dùng để biểu diễn lớp của các đối tượng máy cắt điện.

- Hàm vị từ : là ánh xạ từ một hoặc nhiều phần tử của tập hợp này đến một phần tử duy nhất trong một tập hợp khác. Hàm phải có tên riêng và các đối số vào của nó. Tên của hàm vị từ được qui ước là chuỗi của các chữ cái in thường. Cú pháp tổng quát của hàm là

Tên_hàm(Các đối số vào của hàm).

Hàm nhận các đối số vào từ một tập hợp này và trả về duy nhất một đối số ra trong một tập hợp khác.

Ví dụ : Cho đề xuất là

George is father of David.

Đề xuất này có thể được biểu diễn bằng hàm vị từ father là

father(david).

Hàm trả về giá trị ra của nó là george.

Cho một đề xuất khác là

2 plus 3 is equal to 5.

Đề xuất này có thể được biểu diễn bằng hàm vị từ plus là

plus(2,3).

Hàm sẽ trả về giá trị ra là 5.

Các đối số của hàm vị từ có thể là hằng vị từ hoặc biến vị từ.

- Vị từ : Vị từ là một dạng đặc biệt của hàm vị từ. Vị từ cũng phải có tên riêng và các đối số vào của nó. Tên vị từ thường là mối quan hệ giữa hai hoặc nhiều đối tượng trong một đề xuất. Qui ước đặt tên cho vị từ cũng

giống như hàm vị từ. Vị từ nhận các đối số vào từ một tập hợp và trả về đối số ra trong tập hợp giá trị chân lý đúng T hoặc sai F.

Ví dụ : Cho đề xuất là

George likes Kate.

Đề xuất này có thể được biểu diễn bằng vị từ likes là

likes(george,kate).

Vị từ likes sẽ trả về logic true(T) nếu George thích Kate; mặt khác vị từ trả về giá trị logic false(F).

Các đối số của vị từ có thể là hằng vị từ, biến vị từ hoặc hàm vị từ.

Ví dụ : Cho đề xuất là

X likes Kate, trong đó X là lớp của các đối tượng đàn ông.

Đề xuất này có thể được biểu diễn bằng vị từ likes là

likes(X, kate).

Vị từ sẽ trả về giá trị logic true (T) nếu that sự tất cả những người đàn ông thích Kate; mặt khác vị từ trả về giá trị logic false (F).

- Vị từ định lượng : Khi các đối số vào của hàm vị từ hoặc vị từ là biến vị từ khi đó để xác định phạm vi giá trị của biến, hai đại lượng đứng trước biến đó là $\forall$ và $\exists$, hai đại lượng này được gọi là các vị từ định lượng.

Vị từ định lượng $\forall$ đứng trước biến để xác định biểu thức là đúng cho mọi giá trị của biến.

Vị từ định lượng $\exists$, đứng trước biến để xác định biểu thức là đúng co một vài giá trị của biến.

Ví dụ : Cho đề xuất là

All humans are mortal.

Đề xuất này có thể được biểu diễn là

$(\forall X)$ (human(X) $\rightarrow$ mortal(X)).

Cho một đề xuất khác là

There is a student who is smart.

Đề xuất này có thể được biểu diễn là

$(\exists X)$(student(X) $\wedge$smart(X)).

Ví dụ ứng dụng : Sau đây là ví dụ minh chứng dùng logic vị từ biểu diễn cơ sở tri thức của bài tóan tư vấn tài chánh.

1. saving_account(inadequate) $\rightarrow$ investment(savings).
2. savings_count(adequate)$\wedge$income(adequate)$\rightarrow$investment(stocks).

3. savings_account(adequate)∧income(inadequate)→investmentcombination).
4. ∀X amount_save(X)∧∃Y(dependents(Y)∧ greater(X,minsavings(Y))) →savings_account(adequate).
5. ∀X amount_save(X)∧∃Y(dependents(Y)∧ ¬greater(X,minsavings(Y))) →savings_account(inadequate).
6. ∀X earnings(X,steady)∧∃Y(dependents(Y)∧ greater(X,minincom(Y))) →income(adequate).
7. ∀X earnings(X,steady)∧∃Y(dependents(Y)∧ ¬greater(X,minincom(Y))) →income(inadequate).
8. ∀X earnings(X,unsteady) →income(adequate).
9. amount_saved(22000).
10. earnings(2500,steady).
11. dependents(3).
12. income(inadequate).
13. savings_account(adequate).

3) Giải bài tóan bằng phương pháp hợp giải :

Có hai hướng giải bài tóan trong lĩnh vực trí tuệ nhân tạo đó là giải bài tóan theo hướng thuận và giải bài tóan theo hướng nghịch. Một trong các phương pháp giải bài bài tóan theo hướng nghịc đó là phương pháp hợp giải. Phương pháp hợp giải là phương pháp giải bài tóan bằng chứng minh phản đề. Để chứng minh đề xuất P có giá trị logic đúng từ tập các tiên đề, ta giả sử P là sai, điều đó dẫn đến phủ định của nó là đúng. Cộng tiên đề này vào tập các tiên đề và hợp giải chúng cho đến khi nào có mệnh đề rỗng xuất hiện. Mệnh đề rỗng này chứng tỏ rằng có sự mâu thuẩn giữa tiên đề đã giả định và tập các tiên sẵn có. Vì thế ta đi đến kết luận rằng phủ định của P là không tương thích và P là tương thích với tập các tiên đề. Qui trình chứng minh phản đề bằng phương pháp hợp giải được mô tả gồm các bước như sau :

Bước 1 : Chuyển tất cả các tiên đề ở dạng công thức logic hòan thiện sang dạng logic mệnh đề với dạng tổng quát là $a_1 \vee a_2 \vee a_3 \vee \ldots\ldots \vee a_n$ sử dụng các biểu thức thức logic tương đương là

$\neg(\neg P) = P$.

$P \rightarrow Q = \neg P \vee Q$.

$P \leftrightarrow Q = P \rightarrow Q \wedge Q \rightarrow P$.

$\neg(P \wedge Q) = \neg P \vee \neg Q$
$\neg(P \vee Q) = \neg P \wedge \neg Q.$
$\neg(\exists X)\ P(X) = (\forall X)\neg P(X).$
$\neg(\forall X)\ P(X) = (\exists X)\neg P(X).$

Bước 2 : Để chứng minh đề xuất P là đúng, ta giả sử P là sai, điều đó có nghĩa là phủ định của P là đúng và cộng phủ định này vào tập các tiên đề.

Bước 3 : Đưa tất cả các vị từ định lượng về đứng trước các mệnh đề và loại bỏ chúng khỏi tập các tiên đề.

Bước 4 : Chọn cặp mệnh đề, một có chứa P và một mệnh đề khác chứa ¬P, khử bỏ cặp phân tử này, vì chúng có mâu thuẩn. Các phần tử còn lại của hai mệnh đề hợp nhau tạo thành mệnh đề mới.

Bước 5 : Lặp lại bước 4 cho đến khi nào có mệnh đề rỗng xuất hiện thì dừng thủ tục chứng minh. Mệnh đề rỗng là mệnh đề không chứa bất kỳ một phân tử nào. Điều này chứng tỏ rằng có sự mâu thuẩn giữa tiên đề đã giả sử và tập các tiên đề, do đó ta đi đến kết luận rằng đề xuất P là tương thích với tập các tiên đề và phủ định của đề xuất P là không tương thích với tập các tiên đề.

Ví dụ : Cho tập các tiên đề là

1) All dogs are animals.
2) Fido is a dog.
3) All animals will die.

Hãy chứng minh rằng Fido will die bằng phương pháp hợp gải ?

Giải

Các tiên đề được biểu diễn bằng logic vị từ ở dạng hòan thiện là

1) $\forall(X)\ (dog(x) \rightarrow animal(X))$.
2) dog(fido).
3) $\forall(Y)(animal(Y) \rightarrow die(Y))$.

Để chứng minh đề xuất Fido will die là đúng, giả sử Fido wil die là sai và do đó phủ định của nó là đúng đó là ¬die(fido). Cho tiên đề này là tiên đề 4 và cộng nó vào tập các tên đề.

1) $\forall(X)\ (dog(x) \rightarrow animal(X))$.

2) dog(fido).
3) $\forall$(Y)(animal(Y) $\rightarrow$ die(Y)).
4) $\neg$die(fido).

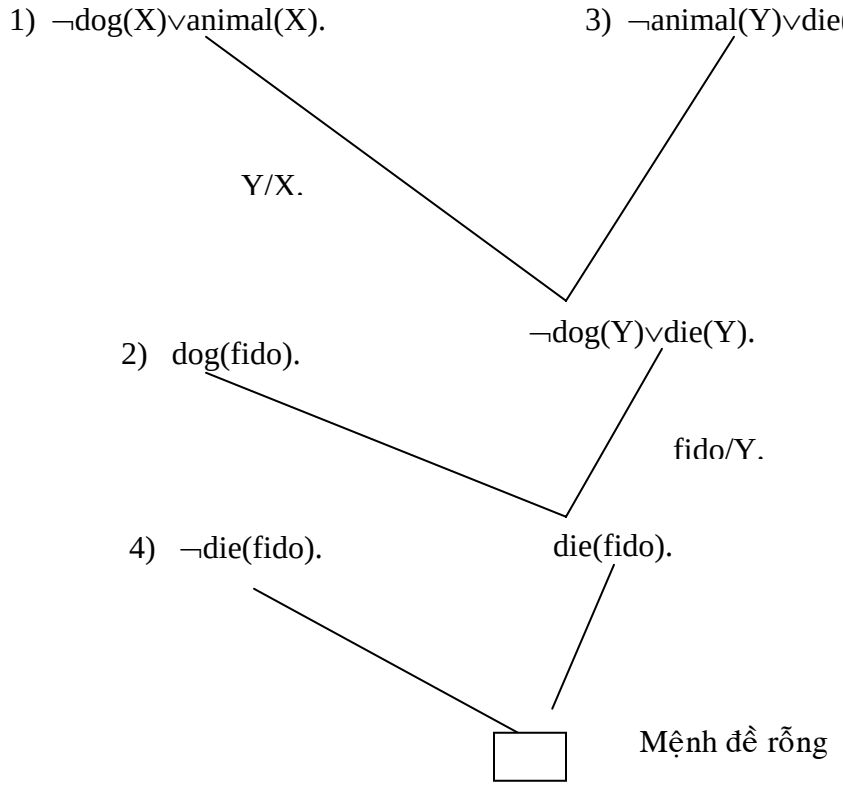
Chuyển các tiên đề ở dạng logic hòan thiện về dạng mệnh đề là

1) $\forall$(X)($\neg$dog(X)$\vee$animal(X)).
2) dog(fido).
3) $\forall$(Y)($\neg$animal(Y)$\vee$die(Y)).
4) $\neg$die(fido).

Khử bỏ tất cả các vị từ định lượng

1) $\neg$dog(X)$\vee$animal(X).
2) dog(fido).
3) $\neg$animal(Y)$\vee$die(Y).
4) $\neg$die(fido).

Qui trình chứng minh bằng cây hợp giả nhị phân được mô tả như hình



4.3) Biểu Diễn Tri Thức Nhờ Mạng Ngữ Nghĩa :

Một cách biểu diễn tri thức khác đó là mạng ngữ nghĩa. Cách biểu diễn này cho ta cái nhìn tổng thể về tri thức của bài tóan bằng bằng đồ thị. Tri thức của bài tóan có thể được thể hiện trên một đồ thị định hướng với tập các nút có dánh nhãn để biểu diễn các đối tượng và tập các cung của đồ thị là các đường mũi tên có đánh

nhãn để chỉ các quan hệ hoặc các tính chất giữa các đối tượng. Với cách biểu diễn này được xem như một cấu trúc dữ liệu mà các ràng buộc vốn sẵn có trong bài tóan có thể được vạch ra và hướng giải quyết vấn đề của bài tóan bằng các luật suy diễn cũng có thể được thiết lập nhờ thông qua các đường mũi tên liên kết giữa các đối tượng. Vì thế mạng còn được gọi là mạng suy diễn tri thức.

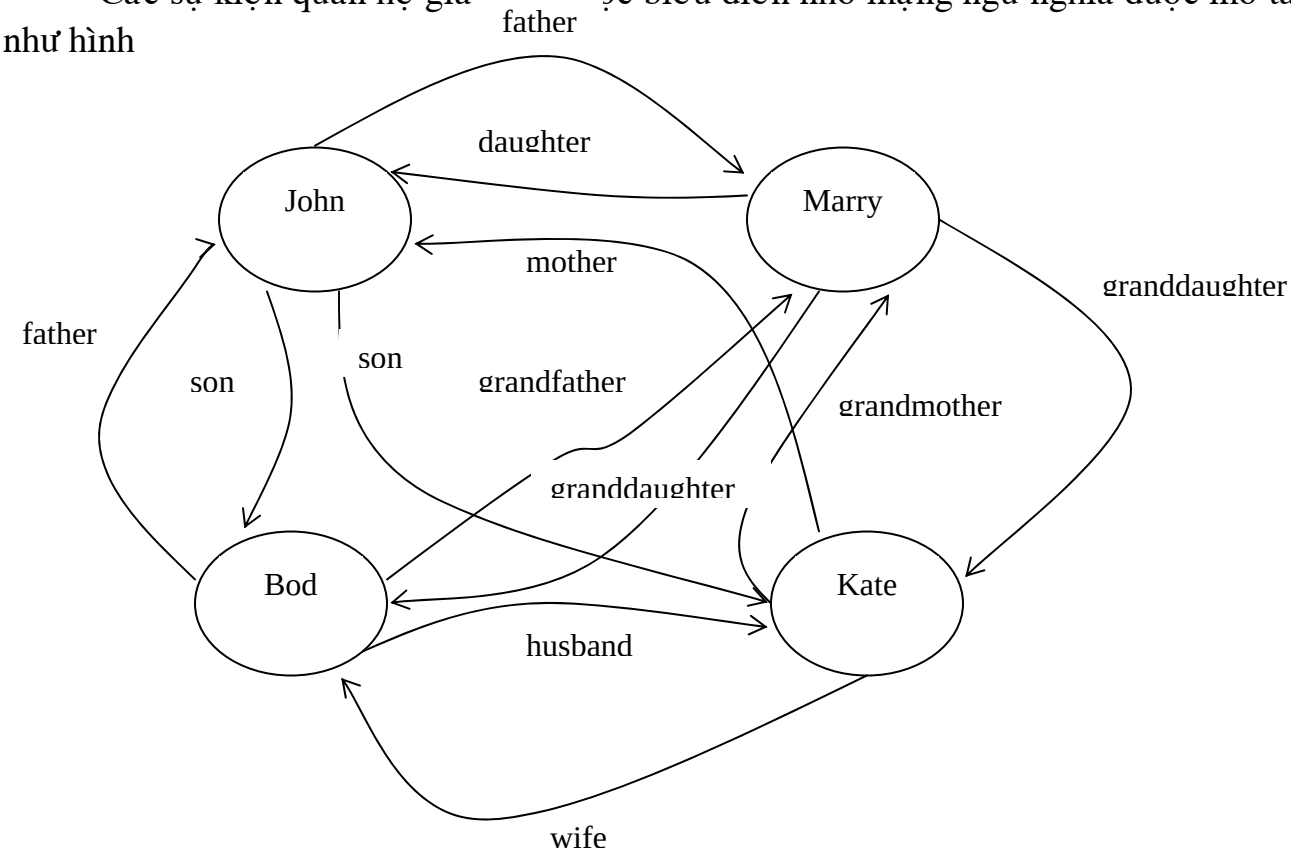
Ví dụ : Cho bài tóan quan hệ gia đình với các sự kiện là

1) John is father of Marry.
2) Mary is daughter of John.
3) Bod is father of John.
4) John is son of Bod.
5) Bod is husband of Kate.
6) Kate is wife of Bod.
7) John is son of Kate.
8) Bod is grandfather of Marry.
9) Kate is grandmother of Marry.
10) Marry is granddaughter of Bod and Kate.

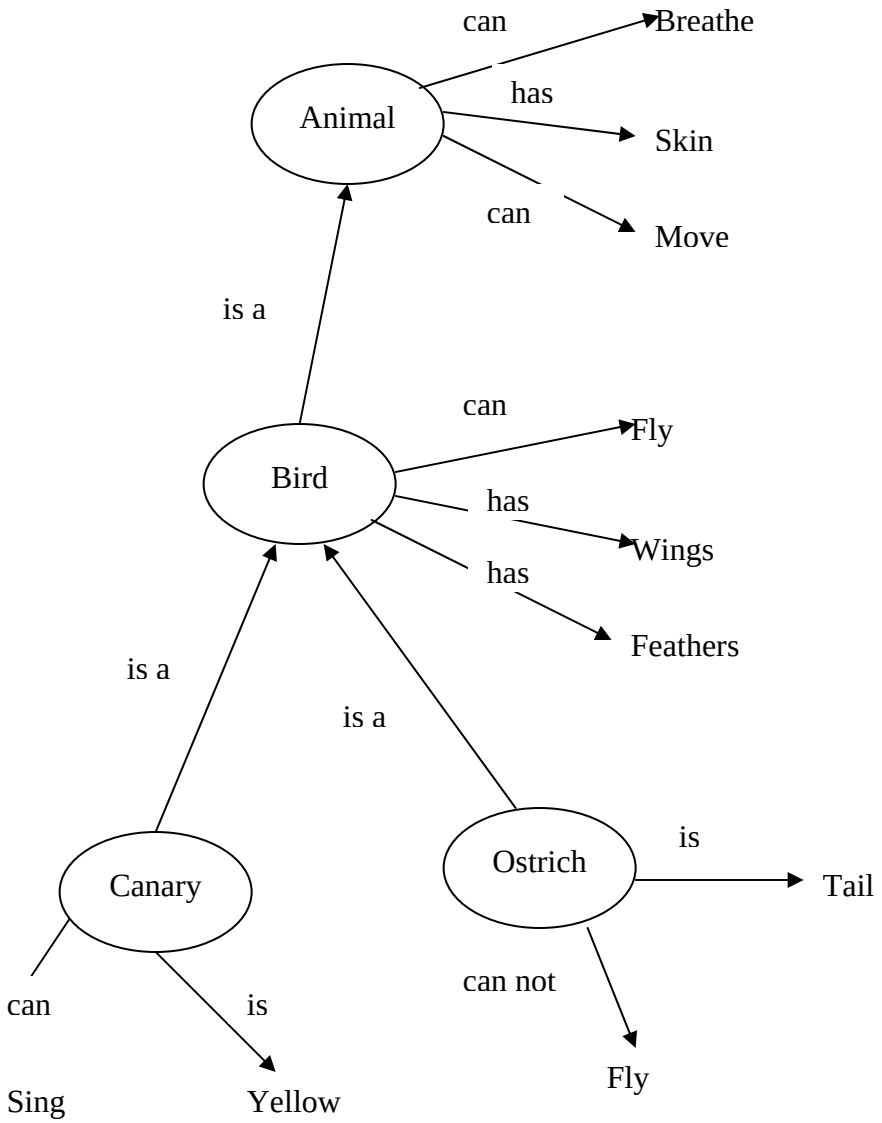
Hãy biểu diễn các sự kiện quan hệ gia đình này nhờ mạng ngữ nghĩa ?

Giải :

Các sự kiện quan hệ gia đình được biểu diễn nhờ mạng ngữ nghĩa được mô tả như hình



Một ví dụ minh chứng khác dùng mạng ngữ nghĩa để biểu diễn lớp động vật được mô tả như hình



4.4) **Biểu Diên Tri Thức Nhờ Frame** :

Mộp phương pháp biểu diễn tri thức khác đó là biểu diễn cấu trúc bằng cách sử dụng mạng ngữ nghĩa, trong đó mỗi nút của mạng là một cấu trúc dữ liệu chứa các slot với các giá trị của các slot được kèm theo. Cấu trúc tổng quát của một Frame dữ liệu được mô tả như hình

Frame <frame_name>
Slot : <property_name_1>
Value : <value of property_name_1>
Slot : <property_name_2>

Value : value of property_name_2>

Slot : <property_name_N>
Value : <value of property_name_N>.

Các slot trong mỗi frame chứa các thông tin như sau :

1) Thông tin nhận dạng frame.
2) Thông tin quan hệ của frame này với frame khác.
3) Các thành phần mô tả của frame.
4) Thông tin thủ thủ tục.
5) Thông tin mặc định frame.
6) Thông tin đề xuất mới.

Biểu diễn tri thức nhờ Frame cung cấp ý tưởng lập trình hướng đối tượng trong ngôn ngữ lập trình Prolog hoặc các ngôn ngữ lập trình khác như C++ và Visual Basic. Frame cho phép truy cập các thành phần của Frame đó là các slot và cho phép hưởng quyền thừa kế giữa Frame dữ liệu này và Frame dữ liệu khác.

Ví dụ : Cho mạng ngữ nghĩa biểu diễn các sự kiện về động vật như hình trên, các sự kiện này được tổ chức trong các khung như sau :

Frame animal
Slot : can
Value : breathe, move
Slot : has
Value : wings

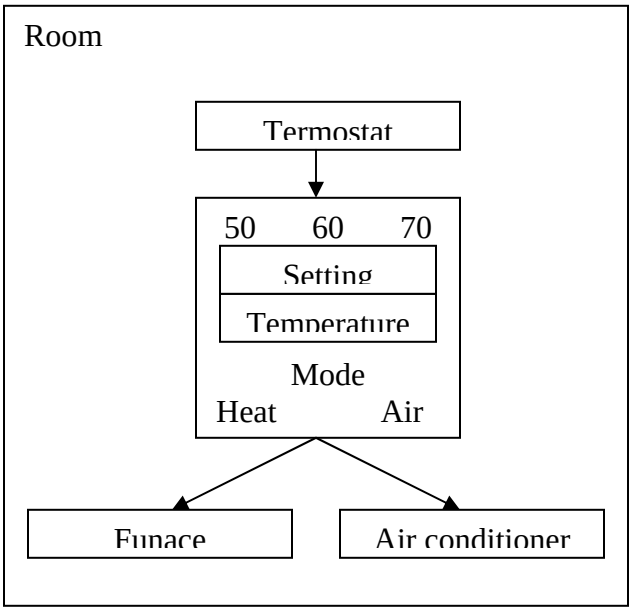
Frame bird
Slot : is_a
Value : animal
Slot : can
Value : fly
Slot : has
Value : wings, feathers

Frame canary
Slot : is_a
Value : bird
Slot : can
Value : sing
Slot : is
Value : yellow

Frame ostrich
Slot : is_a
Value : bird
Slot : can_not
Value : fly
Slot : is
Value : tail

Ví dụ ứng dụng : Thiết kế hệ chuyên gia điều khiển nhiệt độ môi trường trên cơ sở hệ thống Frame.

Xét hệ thống điều khiển nhiêt độ môi trường trong một căn nhà nhỏ gồm có ba phòng đó là phòng khách, phòng ngũ và phòng ăn, trong đó mỗi phòng có một nhiệt kế, một lò sưởi và một máy điều hòa có cấu trúc như hình.



Hệ thống gồm các frame như room, thermostat, funace và air conđioner được mô tả như sau :

Frame room
Slot : funace
Value : < funac1,funace2,funace3 >
Slot : air_conditioner
Value : < air_conditioner1,air_conditioner2, air_conditioner3 >
Slot : thermostat
Value : < thermostat1, thermost2, thermostat3 >
Slot : occupany
Value : < yes,no >

Frame thermostat
Slot : air_conditioner
Value : < air_conditioner1,air_conditioner2, air_conditioner3 >
Slot : funace
Value : < funac1,funace2,funace3 >
Slot : mode
Value : <heat,air>
Slot : setting
Value : 60
Slot : temperature
Value : 65
Slot : room
Value : <livingroom, bedroom, kitchenroom >

Frame air_conditioner
Slot : room
Value : <livingroom, bedroom, kitchenroom >
Slot : state
Value : <on,off>
Slot : thermostat
Value : < thermostat1,thermostat2, thermostat3 >

Frame funace
Slot : room
Value : <livingroom, bedroom, kitchenroom >
Slot : state
Value : <on,off>
Slot : thermostat
Value : < thermostat1,thermostat2, thermostat3 >

Cơ sở luật điều khiển nhiệt độ môi trường trong nhà trên cơ sở các thành phần của hệ thống frame được thiết lập gồm các luật là

Luật 1 : if (temperature < setting) and (funace state ≠ off) and (mode ≠ heat) and (roomoccupancy ≠ yes) then send message (funace state = on).

Luật 2 : if (temperature < setting - 5) and (funace state ≠ off) and (mode ≠ heat) and (roomoccupancy ≠ no) then send message (funace state = on).

Luật 3 : if (temperature >= setting) and (funace state ≠ on) and (mode ≠ heat) and (roomoccupancy ≠ yes) then send message (funace state = off).

Luật 4 : if (temperature >= setting - 5) and (funace state ≠ on) and (mode ≠ heat) and (roomoccupancy ≠ no) then send message (funace state = off).

Luật 5 : if (temperature < setting) and (air_conditioner state ≠ off) and (mode ≠ air) and (roomoccupancy ≠ yes) then send message (air_conditioner state = on).

Luật 6 : if (temperature < setting - 5) and (air_conditioner state ≠ off) and (mode ≠ air) and (roomoccupancy ≠ no) then send message (air_conditioner state = on).

Luật 7 : if (temperature >= setting) and (air_conditioner state ≠ on) and (mode ≠ air) and (roomoccupancy ≠ yes) then send message (air_conditioner state = off).

Luật 8 : if (temperature >= setting - 5) and (air_conditioner state ≠ on) and (mode ≠ air) and (roomoccupancy ≠ no) then send message (air_conditioner state = off).

4.5) **Giới Thiệu Về Ngôn Ngữ Lập Prolog** :

1) **Cấu trúc chương trình** :

Ngôn ngữ Prolog là ngôn ngữ lập trình trên cơ sở tóan học logic được thiết lập để giải quyết các vấn đề trong lĩnh vực trí tuệ nhân tạo. Cấu trúc cơ bản của ngôn ngữ lập trình Prolog được mô tả như sau :

```
domains
        /*
                domain declarations
        */
predicates
        /*
                predicate declarations
        */
clauses
        /*
                clauses ( rules and facts)
        */
goal
        /*
                subgoal_1
                subgoal_2
        /*
```

+ domains : là vùng để khai báo miền kiểu dữ liệu không chuẩn trong các predicate. Giống như các ngôn ngữ lập trình khác, các miền kiểu dữ liệu chuẩncủa Prolog là short, ushort, word, integer, unsigned, long, ulong, dword, real, string và symbol.

+ predicates : là vùng để khai báo các predicates. Predicate phải có tên riêng và các đối số vào của nó. Theo qui ước, tên của predicate phải là chuỗi của các chữ cái in thường và đối số của predicate là chữ in hoa nếu nó là biến; mặc khác nó là chuỗi các chữ cái in thường. Các đối số của predicate phải được khai báo với kiểu dữ liệu tương ứng của chúng.

+ clauses : là vùng cho phép thể hiện các sự kiện và luật suy diễn. Sự kiện được thể hiện dưới dạng vị từ với các đối số của nó là các hằng vị từ. Luật suy diễn được thể

hiện dưới dạng if condition_1 and condition_2, … and condition_N then conclusion với cú pháp tổng quát của Prolog là

conclusion : - condition_1, condition_2, …,condition_N.

+ goal : là vùng thực hiện các đích đề ra của bài tóan.

Ví dụ : Cho các sự kiện và luật suy diễn là

1) John likes wine.
2) Lance likes skiing.
3) Lance likes books.
4) Lance likes films.
5) If Z reads and Z is inquisitive then Z likes books.

Chương trình Prolog sau thể hiện các sự kiện, luật suy diễn để thỏa mãn hai thành phần đích con của John đó là John thích uống rượu, đọc sách và là người tìm tòi do đó john thích sách.

```
domains
        name, thing = symbol
predicates
        nondeterm likes(name, thing)
        reads(name)
        is_inquisitive(name)
clauses
        likes(john,wine).
        likes(lance,skiing).
        likes(lance,books).
        likes(lance,films).
        likes(Z,books) :-
        reads(Z),is_inquisitive(Z).
        reads(john).
        is_inquisitive(john).
goal
likes(X,wine),likes(X,books).
```

Cạy chương trình cho kết quả là

```
X = john.
1 solution.
```

2) Các loại tóan tử :

+ Tóan tử logic :

- Logic and : $\wedge$ với cú pháp Prolog là **,** (dấu phảy).
- Logic or : $\vee$ với cú pháp Prolog là **;** (dấu chấm phảy).
- Logic only if $\leftarrow$ với cú pháp Prolog là **:-**
- Logic not : $\neg$ với cú pháp prolog là not.

+ Tóan tử quan hệ :

- Less than với cú pháp Prolog là **<**
- **Less than or equal to với cú pháp Prolog là <=**
- **Equal với cú pháp Prolog là =**
- **Greater than với cúp pháp Prolog là >**
- **Greater than or equal với cú pháp prolog là >=**
- **Not equal với cú pháp prolog là <> hoặc ><**
- **Cut với cú pháp Prolog là !**

Ví dụ : chương trình Prolog sau là một ví dụ minh chứng sử dụng các tóan trên để giải phương trình bậc 2 có dạng $ax^2 + bx + c = 0$.

```
predicates
solve(real,real,real)
reply(real,real,real)
clauses
solve(A,B,C):-
D = B*B - 4*A*C,
reply(A,B,D),nl.
reply(_,_,D):-
D < 0,
write("No solution"),!.
reply(A,B,D):-
D = 0,
X = -B/(2*A),
write("X = ",X),!.
reply(A,B,D):-
X1 = (-B +sqrt(D))/(2*A),
X2 = (-B -sqrt(D))/(2*A),
write("X1 = ",X1," and X2 = ",X2).
```

```
goal
solve(1.0,2.0,1.0),
solve(1.0,1.0,4.0),
solve(1.0,-3.0,2.0).
```

3) **Xử lý danh sách trong ngôn ngữ lập trình Prolog** :

+ Thành viên của danh sách : Nếu X là thành viên của danh sách L thì mệnh đề clauses được thiết lập là

```
clauses
        member(X,[X|-]).
        Member(X,[_|L]) :- member(X,L).
```

Ví dụ : Chương trình Prolog sau là một ví dụ minh chứng khẳng định nếu đối tượng là thành viên của danh sách thì đáp án là yes; mặt khác đáp án là no.

```
domains
x = symbol
List = symbol*
predicates
member(X,List)
clauses
member(X,[X|_]):-!.
member(X,[_|T]) :- member(X,T).
goal
member(b,[a,b,c]).
```

Chạy chương trình này cho kết quả là yes, vì b là thành viên của danh sách [a,b,c].

+ Nối danh sách : Mệnh đề nối hai danh sách với clause được thiết lập là

```
clauses
        append([],List,List).
        Append([X|L1],L2,[X|L3]) :- append(L1,L2,L3).
```

Ví dụ : Chương trình Prolog sau là một ví dụ nối hai danh sách.

```
domains
s = symbol
List = symbol*
predicates
```

```
append(List,List,List)
clauses
append([],List,List).
append([X|T],L,[X|NL]) :- append(T,L,NL).
goal
append([a,b,c],[d,e],Y).
```

Chạy chương trình này cho kết quả là Y = [“a”,”b”,”c”,”d”,”e”].

+ Hiển thị danh sách : Mệnh đề hiển thị hanh sách ra màn hình với clauses được thiết lập là

```
clauses
    writelist([]).
    writelist([H|T) :-  write(H),nl,writelist(T).
```

+ Đảo ngược thứ tự trong danh sách : mệnh đề đảo ngược thứ tự các phần tử trong danh sách với clauses được thiết lập là

```
clauses
    reverse_writelist([]).
    Reverse_writelist([H|T]) :- reverse_writelist(T),write(H),nl.
```

Ví dụ 1 : Cho bài tóan quan hệ gia đình với các sự kiện và các luật suy diễn là

1) John is son of Dan.
2) Mary is sister of Suzan.
3) Harold is brother of Larry.
4) John married Mary.
5) Larry married Sue.
6) If B is son of A then A is father of B.
7) If A is father of C and C is father of B then A is grandfather of B.
8) If A married C and C is sister of B then A is sister in law of B.
9) If A is brother of C and C married B then A is sister in law of B.

Chương trình Prolog là một ví dụ minh chứng giải quyết bài tóan mối quan hệ gia đình này.

```
database - tmp
  son(STRING,STRING)
  sister(STRING,STRING)
```

```
  brother(STRING,STRING)
  married(STRING,STRING)

clauses
  son("John","Dan").
  sister("Mary","Suzan").
  brother("Harold","Larry").
  married("John","Mary").
  married("Larry","Sue").

predicates
  father(STRING father,STRING child)
  grandfather(STRING grandfather,STRING grandchild)
 nondeterm sister_in_law(STRING,STRING)

clauses
  father(A,B):-
        son(B,A).

  grandfather(A,B):-
        father(A,C),
        father(C,B).

  sister_in_law(A,B):-
        married(A,C),
        sister(C,B).
  sister_in_law(A,B):-
        brother(A,C),
        married(C,B).

goal
  sister_in_law("John",Z),
  format(Msg,"sister_in_law(\"John\",%)",Z),
  write(Msg).
```

Ví dụ 2 : Chương trình Prolog sau là một ví dụ điển giải bài tóan người nông dân sắp xếp các chuyến thuyền qua lại sông.

```
domains
LOC   = east;
        west
 STATE = state(LOC farmer,LOC wolf,LOC goat,LOC cabbage)
 PATH  = STATE*
predicates
 go(STATE,STATE)              % Start of the algorithm
 path(STATE,STATE,PATH,PATH) % Finds a path from one state to another
 nondeterm move(STATE,STATE)  % Transfer a system from one side to
another
 opposite(LOC,LOC)            % Gives a location on the opposite side
 nondeterm unsafe(STATE)     % Gives the unsafe states
 nondeterm member(STATE,PATH) % Checks if the state is already visited
 write_path(PATH)
 write_move(STATE,STATE)

 clauses

 go(StartState,GoalState):-
        path(StartState,GoalState,[StartState],Path),
        write("A solution is:\n"),
        write_path(Path).

 path(StartState,GoalState,VisitedPath,Path):-
        move(StartState,NextState),                 % Find a move
        not(member(NextState,VisitedPath)),        % Check that we have not
had this situation before
        path(NextState,GoalState,[NextState|VisitedPath],Path),
        !.
 path(GoalState,GoalState,Path,Path).                % The final state is
reached
```

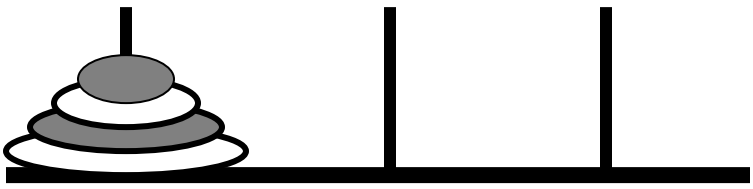
```
 move(state(X,X,G,C),state(Y,Y,G,C)):-
        opposite(X,Y),not(unsafe(state(Y,Y,G,C))).
move(state(X,W,X,C),state(Y,W,Y,C)):-
        opposite(X,Y),not(unsafe(state(Y,W,Y,C))).
move(state(X,W,G,X),state(Y,W,G,Y)):-
        opposite(X,Y),not(unsafe(state(Y,W,G,Y))).
move(state(X,W,G,C),state(Y,W,G,C)):-
        opposite(X,Y),not(unsafe(state(Y,W,G,C))).

opposite(east,west).
opposite(west,east).

unsafe(state(F,X,X,_)):-        % The wolf eats the goat
        opposite(F,X),
        !.
unsafe(state(F,_,X,X)):-        % The goat eats the cabbage
        opposite(F,X),
        !.

member(X,[X|_]):-
        !.
member(X,[_|L]):-
        member(X,L).

write_path([H1,H2|T]):-
        write_move(H1,H2),
        write_path([H2|T]).
write_path([]).

write_move(state(X,W,G,C),state(Y,W,G,C)):-
        !,
        write("The farmer crosses the river from ",X," to ",Y),
        nl.
write_move(state(X,X,G,C),state(Y,Y,G,C)):-
        !,
        write("The farmer takes the Wolf from ",X," of the river to ",Y),
```

```
        nl.
    write_move(state(X,W,X,C),state(Y,W,Y,C)):-
        !,
        write("The farmer takes the Goat from ",X," of the river to ",Y),
        nl.
    write_move(state(X,W,G,X),state(Y,W,G,Y)):-
        !,
        write("The farmer takes the cabbage from ",X," of the river to ",Y),
        nl.

goal
  go(state(east,east,east,east),state(west,west,west,west)),
  write("solved").
```

Chạy chương trình này cho kết quả là

```
A solution is:
The farmer takes the Goat from west of the river to east
The farmer crosses the river from east to west
The farmer takes the cabbage from west of the river to east
The farmer takes the Goat from east of the river to west
The farmer takes the Wolf from west of the river to east
The farmer crosses the river from east to west
The farmer takes the Goat from west of the river to east
no
```

Ví dụ 3 : Cho bài tóan tháp Hà Nội như hình vẽ



Mục tiêu của bài tóan là di chuyển tất cả các đĩa từ cột bên trái sang cột bên phải nhờ thông qua cột trung gian ở giữa mỗi lần di chuyển một đĩa không được phép đĩa lớn nằm trên đĩa nhỏ.

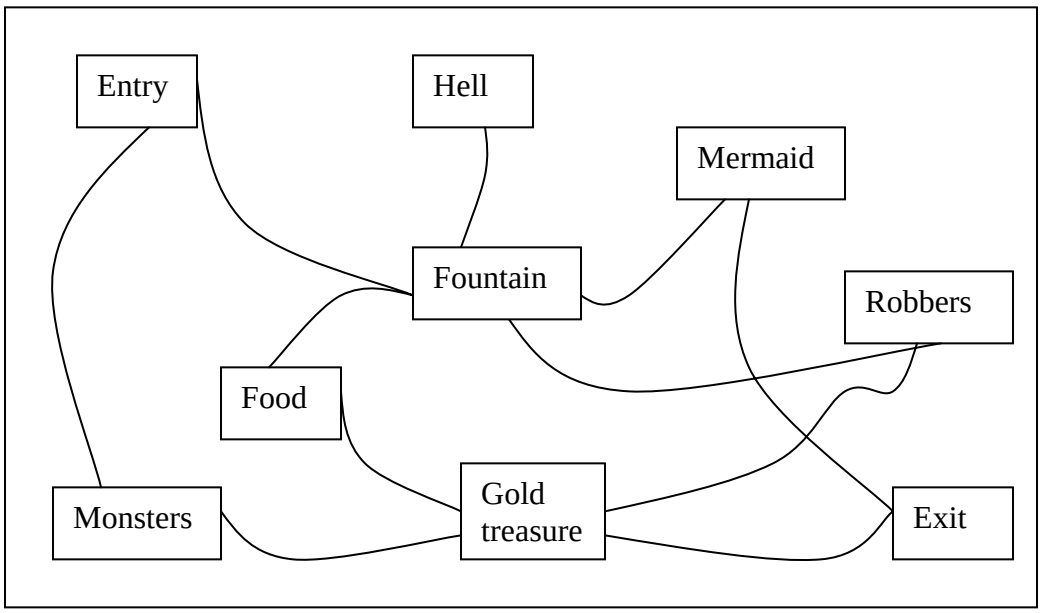
Chương trình Prolog sau là một ví dụ minh chứng giải bài tóan tháp Hà Nội này.

```
domains
  loc =right;middle;left

predicates
  hanoi(integer)
  move(integer,loc,loc,loc)
  inform(loc,loc)

clauses
  hanoi(N):-
        move(N,left,middle,right).
        move(1,A,_,C):-
        inform(A,C),
        !.

  move(N,A,B,C):-
        N1=N-1,
        move(N1,A,C,B),
        inform(A,C),
        move(N1,B,A,C).

  inform(Loc1, Loc2):-
        write("\nMove a disk from ", Loc1, " to ", Loc2).

GOAL
  hanoi(3).
```

Chạy chương trình này cho kết quả là

```
Move a disk from left to right
Move a disk from left to middle
Move a disk from right to middle
Move a disk from left to right
Move a disk from middle to left
Move a disk from middle to right
Move a disk from left to rightyes
```

Ví dụ 4 : Cho bản đồ chỉ đường đến kho báu chứa vàng như hình vẽ



Trên đường tìm đến kho báu chứa vàng phải qua các hang động nguy hiểm như monsters (quái vật) và robbers (những kẻ cướp).
Chương trình Prolog sau là một ví dụ minh chứng tìm đường an tòan đến hang động kho báu chứa vàng.

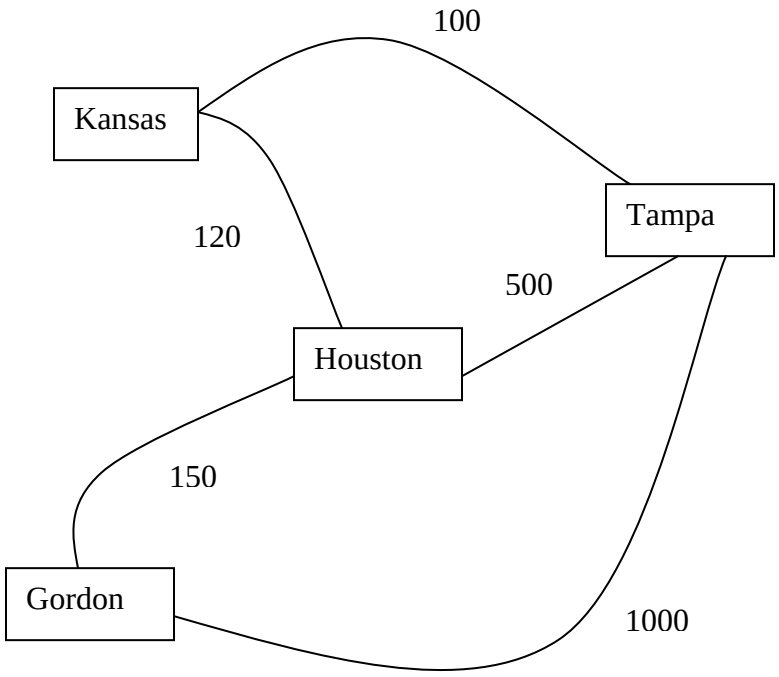
```
domains
        room = symbol
        roomlist = room*
predicates
        nondeterm gallery(room,room)
        nondeterm neighborroom(room,room)
        avoid(roomlist)
        nondeterm go(room,room)
        nondeterm route(room,room,roomlist)
        nondeterm member(room,roomlist)
clauses
        gallery(entry,monsters).
        gallery(entry,fountain).
        gallery(fountain,hell).
        gallery(fountain,food).
        gallery(exit,gold_treasure).
        gallery(fountain,mermaid).
```

```
        gallery(robbers,go_treasure).
        gallery(fountain,robbers).
        gallery(food,gold_treasure).
        gallery(mermaid,exit).
        gallery(monsters,gold_treasure).
        gallery(gold_treasure,exit).
        gallery(mermaid,gold_treasure).
        neighborroom(X,Y) :- gallery(X,Y).
        neighborroom(X,Y) :- gallery(Y,X).
        avoid([monsters,robbers]).
        go(Here,There) :- route(Here,There,[Here]).
        go(_,_).
route(Room,Room,VisitedRooms) :- member(gold_treasure,VisitedRooms),
                                 write(VisitedRooms),nl,fail.
route(Room,Way_out,VisitedRooms) :- neighborroom(Room,Nextroom),
                                 avoid(DangerousRooms),
                                 not(member(NextRoom,DangerousRooms)),
                                 not(member(NextRoom,VisitedRooms)),
route(NextRoom,Way_out,[NextRoom|VisitedRooms]).
member(X,[X|_]):-!.
  member(X,[_|L]):-
    member(X,L).
goal
go(entry,exit).
```

Chạy chương trình này cho kết quả là

```
["exit","gold_treasure","food","fountain","entry"]
["exit","gold_treasure","food","fountain","entry"]
yes
```

Ví dụ 5 : Cho bản đồ của các thành phố như hình vẽ



Chương trình Prolog sau là một ví dụ minh chứng giải bài tóan tìm đường đi ngắn nhất từ thành phố Gordon đến thành phố Tampa.

```
domains
  town     = symbol
  townlist = town*
  distance = integer

predicates
  nondeterm road(town,town,distance)
clauses
  road(tampa,houston,500).
  road(gordon,tampa,1000).
  road(houston,gordon,150).
  road(houston,kansas_city,120).
  road(tampa,kansas_city,100).

predicates
  nondeterm connected(town,town,distance)
clauses
  connected(X,Y,Dist):-
```

```
  road(X,Y,Dist).
 connected(X,Y,Dist):-
  road(Y,X,Dist).

predicates
 determ member(town,townlist)
clauses
 member(X,[X|_]):-!.
 member(X,[_|L]):-
  member(X,L).

predicates
 nondeterm route(town,town,townList,townList,distance)
clauses
 route(Town,Town,VisitedTowns, VisitedTowns, 0) :-
  !.
 route(Town1,Town2,VisitedTowns,ResultVisitedTowns,Distance):-
  connected(Town1,X,Dist1),
  not(member(X,VisitedTowns)),
  route(X,Town2,[X|VisitedTowns],ResultVisitedTowns,Dist2),
  Distance=Dist1+Dist2.

predicates
 showAllRoutes(town,town)
 write_rote(town FirstTown,townList,distance)
 reverse_list(townList InList, townList Tmp, townList Reversed)

clauses
 showAllRoutes(Town1,Town2):-
  write("All routes from ",Town1," to ",Town2," are:\n"),
  route(Town1,Town2, [Town1] ,VisitedTowns, Dist),
  write_rote(Town1,VisitedTowns,Dist),nl,
  fail.
 showAllRoutes(_,_).
 write_rote(Town1,[Town1|VisitedTowns],Dist):-
  !,
```

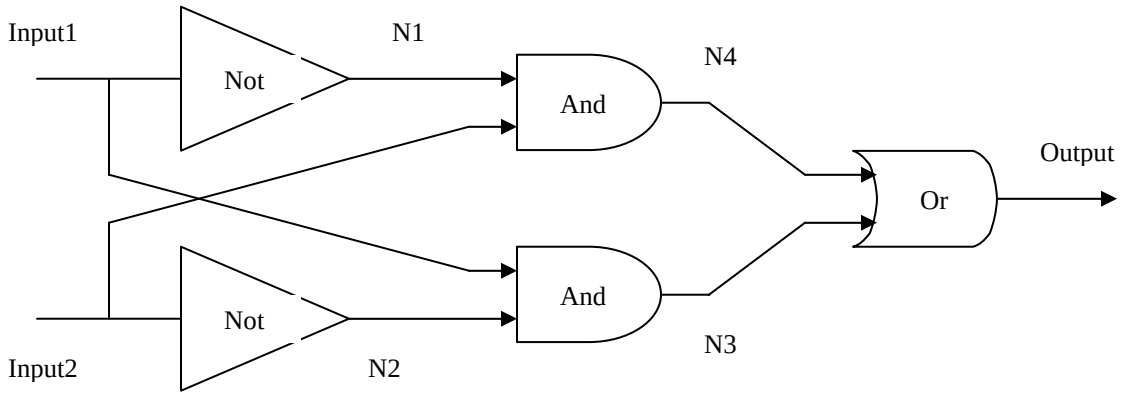
```
    Towns = [Town1|VisitedTowns],
    write("  ",Towns," --> ",Dist),nl.
  write_rote(_,VisitedTowns,Dist):-
    reverse_list(VisitedTowns, [], VisitedTowns_Reversed),
    write("  ",VisitedTowns_Reversed," --> ",Dist),nl.

  reverse_List([],LIST,LIST):-!.
  reverse_List([H|SeenListRest],Interm,SeenList):-
        reverse_List(SeenListRest,[H|Interm],SeenList).

predicates
  showShortestRoutes(town,town)
  determ shorterRouteExist(town,town,distance)

clauses
  showShortestRoutes(Town1,Town2):-
    write("Shortest routes between ",Town1," to ",Town2," is:\n"),
    route(Town1,Town2, [Town1] ,VisitedTowns, Dist),
    not(shorterRouteExist(Town1,Town2,Dist)),
    write_rote(Town1,VisitedTowns,Dist),nl,
    fail.
  showShortestRoutes(_,_).

  shorterRouteExist(Town1,Town2,Dist):-
    route(Town1,Town2, [Town1] ,_, Dist1),
    Dist1<Dist,!.

goal
  showAllRoutes("gordon", "tampa"),nl,
  showShortestRoutes("gordon", "tampa").
```

Chạy chương trình này cho kết quả là

```
All routes from gordon to tampa are:
  ["gordon","tampa"] --> 1000

  ["gordon","houston","kansas_city","tampa"] --> 370
```

```
  ["gordon","houston","tampa"] --> 650

Shortest routes between gordon to tampa is:
  ["gordon","houston","kansas_city","tampa"] --> 370
yes
```

Ví dụ 6 : Mô phỏng phần cứng. Cho mạch số Xor như hình vẽ



Chương trình Prolog sau là một ví dụ minh chứng kiểm tra cách vận hành của mạch.

```
domains
  d = integer

predicates
  not_(D,D)
  and_(D,D,D)
  or_(D,D,D)
  xor_(D,D,D)

clauses
  not_(1,0).
  not_(0,1).
  and_(0,0,0).
  and_(0,1,0).
  and_(1,0,0).
```

```
and_(1,1,1).
or_(0,0,0).
or_(0,1,1).
or_(1,0,1).
or_(1,1,1).

% See the documentarion for the XOR circuit
xor_(Input1,Input2,Output):-
        not_(Input1,N1),
        not_(Input2,N2),
        and_(Input1,N2,N3),
        and_(Input2,N1,N4),
        or_(N3,N4,Output).

goal
 xor_(Input1,Input2,Output),   % Use GOAL mode to see results !!!
 format(Msg," xor_(%,%,%)",Input1,Input2,Output),
 write(Msg).
```

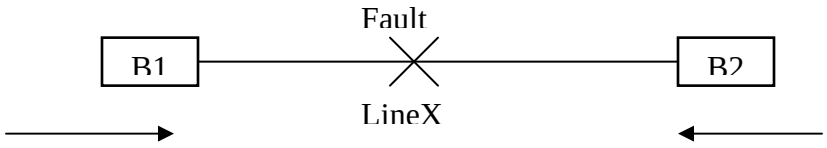
Chạy chương trình này cho kết quả là

```
xor_(1,1,0)Input1=1, Input2=1, Output=0, Msg= xor_(1,1,0)
 xor_(1,0,1)Input1=1, Input2=0, Output=1, Msg= xor_(1,0,1)
 xor_(0,1,1)Input1=0, Input2=1, Output=1, Msg= xor_(0,1,1)
 xor_(0,0,0)Input1=0, Input2=0, Output=0, Msg= xor_(0,0,0)
4 Solutions
```

Chương 5 : Vài Ứng Dụng Trí Tuệ Nhân Tạo

5.1) Ứng Dụng trí Tuệ Nhân Tạo Phân Tích Bảo Vệ Hệ Thống Năng Lượng điện :

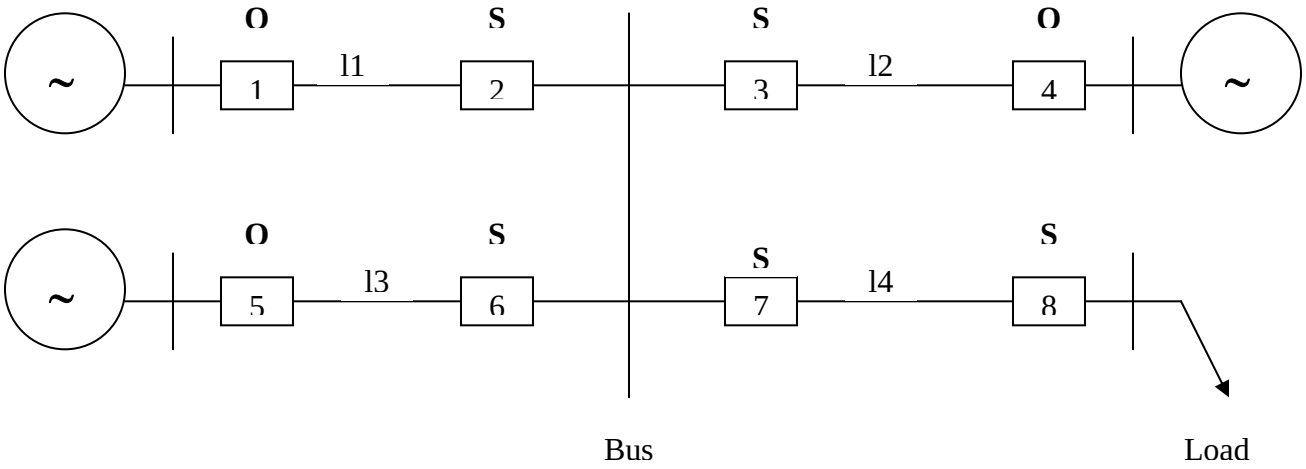
Cho hệ thống năng lượng điện đơn giản gồm hai cắt và một đường dẫn như hình vẽ



Máy cắt B1 được đặt ở đầu cuối bên trái và máy cắt B2 được đặt ở đầu cuối bên phải của đường dẫn LineX. Máy cắt B1 bảo vệ đường LineX theo hướng nhìn từ trái sang phải và máy cắt B2 bảo vệ đường dẫn LineX theo hướng nhìn từ phải sang trái.

Nếu có sự cố trên đường dẫn LineX thì một hoặc cả hai máy cắt ở hai đầu cuối của đường sẽ vận hành ngắt dòng bảo vệ đường.

Bài tóan đặt ra là có một hệ thống năng lượng phức tạp hơn gồm nhiều đường dẫn và nhiều máy cắt như hình vẽ



Ký hiệu O là máy cắt vận hành ngắt dòng và ký hiệu S là máy cắt có chức năng sai không vận hành ngắt dòng.

Với một hệ thống phức tạp như vậy, khi có sự cố trên đường dẫn LineX, thì việc phân tích bảo vệ hệ thống là rất cần thiết. Công việc phân tích bảo vệ hệ thống được đặt ra mấy vấn đề như sau :

1) Xác định vị trí xảy ra sự cố trên hệ thống.
2) Xác định trạng thái vận hành và trạng thái không vận hành của các máy cắt trên hệ thống.
3) Chỉ rõ các thành phần của hệ thống.
4) Xác định hệ thống của các máy cắt dự phòng cho các máy cắt không vận hành trên hệ thống.

Cho hệ thống năng lượng gồm 8 máy cắt và 4 đường dẫn với các sự kiện hiện có trên hệ thống như đã được mô tả trên, công việc phân tích bảo vệ hệ thống này là giả sử rằng nếu có sự cố xảy ra trên mỗi đường dẫn LineX thì việc phân tích bảo vệ hệ thống phải xác định được vị trí đường dẫn LineX, các máy cắt bảo vệ đường dẫn LineX vận hành hoặc không vận hành và nếu máy cắt không vận hành thì phải có máy cắt khác dự phòng vận hành để bảo vệ hệ thống.

Kỹ thuật trí tuệ nhân tạo được ứng dụng để thiết kế hệ thống phân tích bảo vệ hệ thống năng lượng bao gồm các công việc như sau :

+ Công việc mô tả các sự kiện hiện có của hệ thống như nguồn cung cấp năng lượng, hai máy cắt bảo vệ đường một đường dẫn, máy cắt vận hành và máy cắt không vận hành, các máy cắt liên thông qua thanh góp.

+ Công việc thiết kế cơ sở luật suy diễn từ các sự kiện hiện có được mô tả trên hệ thống như luật suy diễn liên thông giữa hai máy cắt, luật suy diễn xác định máy cắt cùng bảo vệ đường dẫn, luật suy diễn máy cắt có nguồn, luật suy diễn máy cắt dự phòng cho một máy cắt khác, luật suy diễn máy cắt dự phòng cho một máy cắt khác không vận hành, luật suy diễn máy cắt mất nguồn và luật suy diễn xác định đường dẫn có sự cố.

Để mô tả các sự kiện hiện có trên hệ thống năng lượng điện, các vị từ tổng quát được thiết lập là

+ Vị từ generation(B) : mô tả máy cắt B nối trực tiếp với nguồn.

+ Vị từ protected_by(LineX, B1, B2) : Đường dẫn LineX được bảo vệ bởi hai máy cắt B1 và B2.

+ Vị từ connect(B1,B2) : mô tả máy cắt B1 là liên thông với máy cắt B2 qua thanh góp.

+ Vị từ operate(B) : mô tả máy cắt vận hành.

Để thiết kế hệ cơ sở luật suy diễn giải quyết bài tóan phân tích bảo vệ hệ thống năng lượng như được mô tả trên, các luật suy diễn được thiết lập là

+ Luật xử lý các máy cắt liên thông qua thanh góp.

If connect(B1, B2) then connection(B1, B2).

If connect(B2, B1) then connection(B1,B2).

+ Luật xử lý máy cắt cùng bảo vệ đường dẫn.

If protected_by(LineX, B1, B2) then other_breaker(B1, B2).

If protected_by(LineX, B2, B1) then other_breaker(B1, B2).

+ Luật xử lý máy cắt có nguồn.

If generation(B) then has_gen(B).

If connection(B, B1) and other_breaker(B1, B2) and has_gen(B2) then has_gen(B).

+ Luật xử lý máy cắt dự phòng cho một máy cắt khác.

If not(generation(B1)) and connection(B1, B3) and other_breaker(B3,B2) and has_gen(B2) then back_up(B1, B2).

+ Luật xử lý máy cắt dự phòng cho một máy cắt khác không vận hành.

If back_up(B1, B2) and not(operate(B2)) then backup_did_not_work(B1, B2).

+ Luật xử lý máy cắt mất nguồn.

If not(has_gen(B)) then no_source_coming(B).

If has_gen(B) and operate(B) then no_source_coming(B).

If back_up(B1, B2) and not(backup_did_not_work(B1, B2)) then No_source_coming(B1).

+ Luật xử lý xác định đường dẫn LineX có sự cố.

If no_source_coming(B1) and no_source_coming(B2) then fault(LineX, B1, B2).

Chương trình Prolog sau là một ví dụ minh chứng giải quyết bài tóan phân tích bảo vệ hệ thống nặng lượng với mô hình topo hình học đã cho trên.

```
database -tmp
protected_by(STRING,STRING,STRING)
connect(STRING,STRING)
operate(STRING)
generation(STRING)
predicates
```

```
nondeterm connection(STRING,STRING)
nondeterm other_breaker(STRING,STRING)
nondeterm has_gen(STRING)
nondeterm back_up(STRING,STRING)
nondeterm backup_did_not_work(STRING,STRING)
nondeterm no_source_coming(STRING)
        fault(STRING,STRING,STRING)
printbackup(STRING)
printout(STRING)
run
clauses
protected_by("line1","1","2").
protected_by("line2","3","4").
protected_by("line3","5","6").
protected_by("line4","7","8").
connect("2","3").
connect("2","6").
connect("2","7").
connect("3","6").
connect("3","7").
connect("6","7").
generation("1").
generation("4").
generation("5").
operate("1").
operate("4").
operate("5").
connection(B1,B2) :- connect(B1,B2).
connection(B1,B2) :- connect(B2,B1).
other_breaker(B1,B2) :- protected_by(_,B1,B2).
other_breaker(B1,B2) :- protected_by(_,B2,B1).
has_gen(B) :- generation(B),!.
has_gen(B) :- connection(B,B1),other_breaker(B1,B2),has_gen(B2),!.
back_up(B1,B2)                                                    :-
not(generation(B1)),connection(B1,B3),other_breaker(B3,B2),has_gen(B2).
backup_did_not_work(B1,B2) :- back_up(B1,B2),not(operate(B2)).
```

```
no_source_coming(B1) :- not(has_gen(B1)).
no_source_coming(B1) :- has_gen(B1),operate(B1).
no_source_coming(B1) :- back_up(B1,_),not(backup_did_not_work(B1,_)).
fault(_,B1,B2) :- no_source_coming(B1),no_source_coming(B2),!.
printbackup(B) :- back_up(B,B1),operate(B1),
                  write("Breaker"),
                  write(B1),
                  write(" Operated correctly as a backup breaker"),nl,fail.
printout(B) :- has_gen(B),operate(B),
                  write("Breaker"),write(B),
                  write(" operated correctly"),nl,!.
printout(B) :- has_gen(B),not(operate(B)),
                  write("Breaker"),
                  write(B),
                  write(" Malfunctioned"),nl,
                  not(printbackup(B)),!.
run :-
        protected_by(L,B1,B2),
        fault(L,B1,B2),
        write("Possible Fault Location is on "),
        write(L),nl,
        printout(B1),
        printout(B2),nl,nl,fail.
goal
  run.
```

Khi chạy chương trình này cho kết quả là

```
        Possible Fault Location is on line1
        Breaker1 operated correctly
        Breaker2 Malfunctioned
        Breaker4 Operated correctly as a backup breaker
        Breaker5 Operated correctly as a backup breaker

        Possible Fault Location is on line2
        Breaker3 Malfunctioned
        Breaker5 Operated correctly as a backup breaker
```

Breaker1 Operated correctly as a backup breaker
Breaker4 operated correctly

Possible Fault Location is on line3
Breaker5 operated correctly
Breaker6 Malfunctioned
Breaker1 Operated correctly as a backup breaker
Breaker4 Operated correctly as a backup breaker

Possible Fault Location is on line4
Breaker7 Malfunctioned
Breaker1 Operated correctly as a backup breaker
Breaker4 Operated correctly as a backup breaker
Breaker5 Operated correctly as a backup breaker
No

5.2) Bài Tóan Robot Tìm Vàng :

Cho bài tóan robot tìm vàng trên mặt phẳng kẻ lưới hai chiều như hình vẽ

(1,4) Stench	(2,4)	(3,4)	(4,4)
(1,3) Wumpus	(2,3) Gold gliter	(3,3)	(4,3)
(1,2) Stench	(2,2)	(3,2) Breeze	(4,2)
(1,1) Agent	(2,1) Breeze	(3,1) Pits	(4,1)

Bài tóan đặt ra là giả sử rằng robot đang ở tại vị trí ô có chỉ số (1,1) viếng thăm qua các ô khác để tìm ô chứa vàng, lấy vàng và mang vàng trở về lại nhà là ô (1,1). Quá trình thăm dò qua các ô, robot phải đối mặt với các ô chứa các chướng ngại vật như hầm bẩy và kẻ trông coi vàng. Robot sẽ bị nguy hiểm nếu nó đi vào các ô này. Trước khi robot đi vào các ô chứa các đối tượng này, nó có thể đánh mùi các đối tượng này ở các ô kề của chúng. Hãy xây dựng hệ cơ sở tri thức cho robot có thể thực hiện các thao tác thăm dò qua các ô biết suy nghĩ tránh được các ô chứa các chướng ngại vật và tìm đường an tòan đến ô chứa vàng, lấy vàng và mang vàng trở về lại nhà là ô (1, 1) ?

Các ký hiệu sử dụng với bài tóan này có nghĩa như sau :

+ Agent : robot.
+ Gold : vàng.
+ Wumous : kẻ trông coi vàng.
+ Pits : hầm bẩy.
+ Stench : mùi kẻ trông coi vàng.
+ Breeze : mùi hầm bẩy.
+ gliter : mùi có vàng.

Để xây dựng một hệ thống cơ sở tri thức cho robot có thể thực hiện được các yêu cầu đề ra như trên, các công việc sau cần phải được xem xét đó là

+ Mô tả các sự kiện về robot và các sự kiên liên quan với robot như thao tác di chuyển và thao tác lấy vàng của robot, vị trí và tình huống của robot, vị trí ô kề đối mặt với robot đến thăm dò hoặc không đến thăm dò, vị trí ô chứa chướng ngại vật và các ô kề chứa mùi chướng ngại vật.

+ Hệ thống cơ sở luật suy diễn cho robot biết suy nghĩ tính tóan để thực hiện các thao tác cần thiết của nó.

Để mô tả các sự kiện về robot và các sự kiện liên quan với robot, các vị từ và các hàm vị từ sau đây được thiết lập là

+ Thao tác di chuyển và thao tác lấy vàng của robot.

- turn(left) : lệnh quẹo trái.
- turn(right) : lệnh quẹo phải.
- forward : lệnh đi tới.
- grab : lệnh lấy vàng.

+ Vị trí, tình huống và định hướng nhìn của robot.
  - result(Action, S_i) = S_{i+1} : hàm trả về tình huống S_{i+1} khi thực hiện thao tác Action tại tình huống S_i .
  - at(Object, Location, Situation) : mô tả đối tượng tại vị Location với tình huống Situation.
  - orientation(Agent, Situation) = D : hàm trả về góc D định hướng nhìn của robot với tình huống situation. Theo qui ước, D quay tròn 360^0, D = 0, mặt của robot nhìn về hướng đông; D = 90, mặt của ro bot nhìn về hướng bắc; D = 180 , mặt của robot nhìn về hướng tây và D = 270, mặt của robot nhìn về hướng nam.
  - locationtoward([X, Y], D) = Location : hàm trả về vị trí chỉ số Location của ô kề đối mặt với ô (X, Y) được xác định bởi góc định hướng D.

Hệ thống cơ sở luật suy diễn cho robot có khả năng suy nghĩ tính tóan để thực hiện các thao tác cần thiết tránh chướng ngại vật và bám theo đường trên các ô an tòan tìm đến ô chứa vàng, lấy vàng và mang vàng về ô (1,1) được thiết lập gồm các luật suy diễn như sau :

+ Luật xử lý vị trí ô đối mặt với robot.

at(Agent, L, S)$\rightarrow$ locationAhead(A, S) =
locationtoward(L,orientation(Agent, S)).

+Luật xử lý vị trí các ô kề liên kết.

adjacent(L_1, L_2) $\leftrightarrow$ $\exists$D L_1 = locationtoward(L_2, D).

+ Luật xử lý xác định vị trí các ô chứa các đường biên.

wall(X, Y) $\leftrightarrow$ (X = 0 $\vee$ X = 5 $\vee$ Y = 0 $\vee$ Y = 5).

+ Luật xử lý thực hiện lệnh forward đi tới.

at(Agent, L, result(Action, S)) $\leftrightarrow$ Action = forward$\wedge$L =
locationAhead(Agent, S)$\wedge\neg$wall(L).

+ Luật xử lý thực hiện lệnh queo trái.

orientation(Agent, result(Action, S)) = D $\leftrightarrow$ Action = turn(left)$\wedge$
D = Mod(orientation(Agent, S) + 90, 360).

+ Luật xử lý thực hiện lệnh queo phải.

orientation(Agent, result(Action, S)) = D $\leftrightarrow$ Action = turn(right)$\wedge$
D = Mod(orientation(Agent, S) - 90, 360).

+ Luật xử lý vị trí ô chứa mùi hầm bẩy và mùi kẻ trông coi vàng.

at(Agent, L, S) $\wedge$breeze(S) $\rightarrow$ breezy(L).
at(Agent, L, S)$\wedge$stench(S) $\rightarrow$ smelly(L).

+ Luật xử lý vị trí các ô kề chứa mùi các đối tượng hầm bẩy và kẻ trông coi vàng.

at(Wumpus, L_1, S)$\wedge$adjacent(L_1, L_2) $\rightarrow$ smelly(L_2).
at(Pitts, L_1, S)$\wedge$adjacent(L_1, L_2) $\rightarrow$ breezy(L_2).

+ Luật xác định vị trí các ô có khả năng chứa hầm bẩy và kẻ trông coi vàng.

smelly(L_1) $\rightarrow$ $\exists L_2$ at(Wumpus, L_2, S)$\wedge L_2 = L_1 \vee$adjacent(L_1, L_2).
breezy(L_1) $\rightarrow$ $\exists L_2$ at(Pitts, L_2, S)$\wedge L_2 = L_1 \vee$adjacent(L_1, L_2).

+ Luật xác định vị trí các ô an tòan.

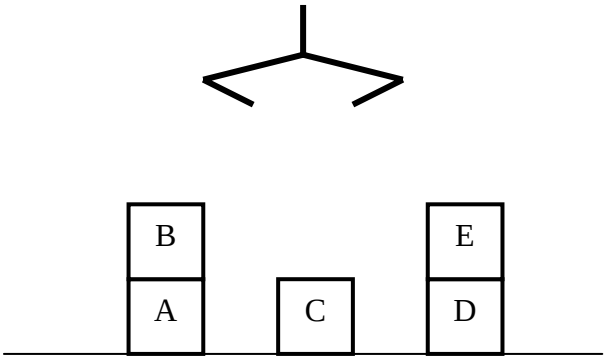
at(Agent, L_1, S)$\wedge$adjacent(L_1, L_2) $\rightarrow$ ok(L_2).
$\neg$at(Wumpus, L, S)$\wedge\neg$at(Pitts, L,S) $\rightarrow$ ok(L).

+ Luật xác định vị trí và tình huống của robot tìm thấy vàng, lấy vàng và mang vàng về lại ô (1, 1).

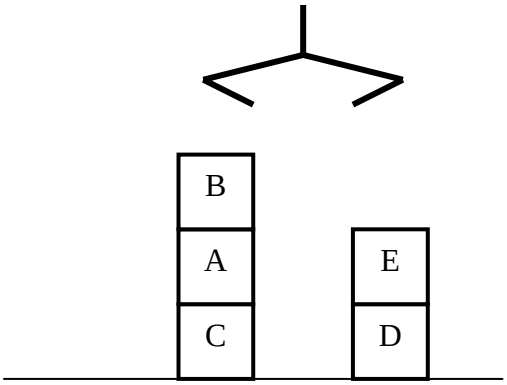
at(Agent, L, S)$\wedge$gliter(S) $\rightarrow$ atGold(S).
atGold(S) $\rightarrow$ present(Gold, S).
present(Gold, S)$\wedge$portable(Gold, S) $\rightarrow$ holding(Gold, result(grab, S)).
holding(Gold, S) $\rightarrow$ goallocation([1, 1], S).

5.3) Bài Tóan Lập Phương Án Cho Cánh Tay Robot Xếp Khối :

Cho bài tóan khối trên mặt bàn và cánh tay robot với trạng thái ban đầu và trạng thái đích như hình vẽ



Trạng thái ban đầu



Trạng thái đích

Bài tóan đặt ra là lập phương án cho cánh tay robot dời các khối từ trạng thái ban đầu của bài tóan sang trạng thái đích của bài tóan. Để làm được việc này, cánh tay robot phải thực hiện các thao tác là

+ goto(X, Y, Z) : di chuyển cánh tay robot đến vị trí có tọa độ X, Y, Z.
+ pickup(X) : thực hiện lệnh nhặt khối X lên từ mặt bàn.
+ putdown(X) : thực hiện lệnh đặt khối X xuống mặt bàn.
+ takeoff(X, Y) : thực hiện lệnh lấy khối X từ đỉnh của khối Y.
+ puton(X, Y) : thực hiện lệnh đặt khối X lên đỉnh của khối Y.

Để biểu diễn các trạng thái của bài tóan, các vị từ tổng quát được thiết lập là

+ location(W, X, Y, Z) : mô tả khối W định vị tại vị trí có tọa độ X, Y, Z.
+ on(X, Y) : mô tả khối X name trên khối Y.
+ clear(X) : mô tả làm sạch khối X (không có khối bất kỳ nằm trên khối X).
+ hold(X) : mô tả cánh tay robot cầm giữ khối X.
+ hold() : mô tả cánh tay robot rỗng.
+ ontable(X) : mô tả khối X name trên mặt bàn.

Hệ thống cơ sở luật suy diễn điều khiển cánh tay dời các khối từ trạng thái ban đầu đến trạng thái của bài tóan được thiết lập gồm các luật là

+ Luật xác định làm sạch khối.

$\forall X(clear(X) \leftarrow \neg\exists Y(on(Y, X)))$.

+ Luật xác định khối name trên mặt bàn.

$\forall X\ \forall Y\ (\neg on(Y, X) \leftrightarrow ontable(Y))$.

+ Luật xác định cánh tay robot rỗng.

$\forall Y$ (hold() $\leftrightarrow \neg$hold(Y)).

+ Luật thay đổi trạng thái khi thực hiện lệnh pickup.

$\forall X$ (pickup(X) $\rightarrow$ (hold(X) $\leftarrow$ (hold()$\wedge$ontable(X)$\wedge$clear(X)))).

+ Luật thay đổi trạng thái khi thực hiện lệnh putdown.

$\forall X$ (putdown(X) $\rightarrow$ (hold()$\wedge$ontable(X)$\wedge$clear(X)) $\leftarrow$ hold(X))).

+ Luật thay đổi trạng thái khi thực hiện lệnh puton.

$\forall X \forall Y$ (puton(X, Y) $\rightarrow$ ((hold()$\wedge$on(X, Y)$\wedge$clear(X))$\leftarrow$ (hold(X)$\wedge$clear(Y)))).

+ Luật thay đổi trạng thái khi thực hiện lệnh takeoff.

$\forall X \forall Y$ (takeoff(X, Y) $\rightarrow$ ((hold(X)$\wedge$clear(Y))$\leftarrow$ (hold()$\wedge$on(X,Y)$\wedge$clear(X)))).

+ Luật xử lý ràng buộc khi thực hiện lệnh takeoff.

$\forall X \forall Y \forall Z$ (takeoff(Y, Z) $\rightarrow$ (ontable(X)$\leftarrow$ontable(X))).

+ Luật xử lý ràng buộc khi thực hiện lệnh puton.

$\forall X \forall Y \forall Z$ (puton(Y, Z) $\rightarrow$ (ontable(X)$\leftarrow$ontable(X))).

Tương tự với hai luật ràng buộc trên, các luật ràng buộc khác cho quan hệ on và clear phải được thiết lập.

Do có nhiều luật ràng buộc khung kết hợp với các luật thay đổi trạng thái phát sinh ra một không gian trạng thái tìm kiếm của bài tóan là quá lớn và quá phức tạp tạo ra nhiều đường khác nhau dẫn về đích của bài tóan, trong đó mỗi đường là một phương án có thể điều khiển cánh tay robot dời các khối từ trạng thái ban đầu sang trạng thái đích.

Vì sử dụng quá nhiều luật ràng buộc khung với các lệnh thay đổi trạng thái của bài tóan tạo ra một không gian trạng thái tìm kiếm là quá phức tạp, điều này dẫn đến việc lập phương án cho cánh tay robot dời khối từ trạng thái ban đầu đạt đến trạng thái đích là rất khó khăn. Để khắc phục điều này, công việc loại bỏ việc sử dụng các luật ràng buộc khung, hệ thống luật thay đổi trạng thái của bài tóan có thể được cải tiến với luật ba thành phần là

$$pickup(X): \begin{cases} P: hold() \wedge ontable(X) \wedge clear(X) \\ A: hold(X) \\ D: hold() \wedge ontable(X) \wedge clear(X) \end{cases}$$

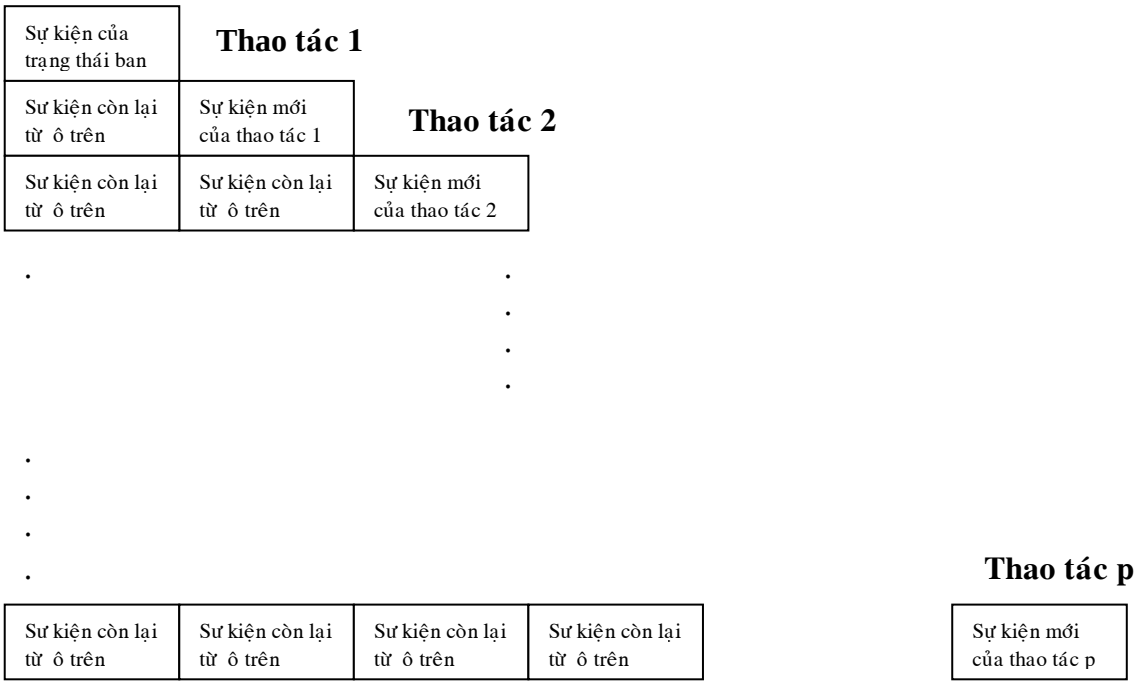
$$putdown(X):\begin{cases}P: hold(X)\\A: hold() \wedge ontable(X) \wedge clear(X)\\D: hold(X)\end{cases}$$

$$puton(X,Y):\begin{cases}P: hold(X) \wedge clear(Y)\\A: hold() \wedge on(X,Y) \wedge clear(X)\\D: hold(X) \wedge clear(Y)\end{cases}$$

$$takeoff(X,Y):\begin{cases}P: hold() \wedge on(X,Y) \wedge clear(X)\\A: hold(X) \wedge clear(Y)\\D: hold() \wedge on(X,Y) \wedge clear(X)\end{cases}$$

Trong đó, P là danh sách chứa các tiền điều kiện, A là danh sách chứa các sự kiện mới và D là danh sách chứa các tiền điều kiện đã sử dụng và được hủy bỏ.

<u>Bảng biểu tam giác</u> : Để nhớ lại các thao tác của một phương án, một cấu trúc dữ liệu mới được đề xuất đó là bảng biểu tam giác. Nếu phương án được thiết lập cho cánh tay robot dời khối từ trạng thái ban đầu có số p thao tác, thì bảng biểu tam giác được thiết lập với p + 1 hàng và p + 1 cột như bảng.



Cách thiết lập bảng biểu tam giác để nhớ lại các thao tác của phương án được thiết lập cho cánh robot dời các khối từ trạng thái ban đạt đến trạng thái đích của bài tóan khối được mô tả như sau :

+ Phương án có p thao tác, bảng biểu tam giác được thiết lập là p + 1 hàng và p + 1 cột.

+ Ô đầu tiên của bảng biểu với chỉ số (0, 0) chứa các sự kiện mô tả trạng thái ban đầu của bài tóan.

+ Ô có chỉ số (n, n) với n > 0 chứa các sự kiện mới của thao tác thứ n.

+ Ô có chỉ số (n, m) với m < n chứa các sự kiện còn lại từ ô (n-1, m) tức lã loại bỏ một số tiền điều kiện mà thao tác thứ n đã sử dụng ở ô(n-1, m) và số các sự kiện còn lại ở ô (n-1, m) được ghi xuống ô (n, m) với m < n.

Chương 6 : Xử Lý Tri Thức Không Chắc Chắn

6.1) Lý Giải Dưới Điều Kiện Không Chắc Chắn :

Tri thức của bài tóan đã được xử lý trước đây đó là loại tri thức chắc chắn. Để xử lý loại tri thức này sử dụng logic rõ hay còn được gọi là logic hai chữ số đó là logic vị từ được mở rộng từ logic đề xuất.

Tri thức chắn chắn là loại tri thức mà miền giá trị chân lý logic của nó là logic true và logic false ứng với hai chữ số 1 và 0.

Một loại tri thức khác của bài tóan đó là tri thức không chắc chắn. Tri thức không chắc chắn là loại tri thức mà miền giá trị chân lý của nó là không chắc chắn đúng và không chắc chắn sai. Điều đó có nghĩa là miền giá trị chân lý của nó là ở trong khỏang 0 và 1. Loại tri thức này thường được phát biểu với các nhóm không chắc chắn là

+ Tuyệt đối sai.
+ Hầu như không chắc chắn.
+ Có lẽ không chắc chắn.
+ Có thể không chắc chắn.
+ Chưa biết.
+ Có thể chắc chắn.
+ Có lẽ chắc chắn.
+ Hầu như chắc chắn.
+ Tuyệt đối chắc chắn.

Ví dụ : Cho luật suy diễn là

$$P \rightarrow Q.$$

Nếu suy diễn là tri thức chắc chắn thì giá trị chân lý của tiền điều kiện P là 1 hoặc 0 và giá trị chân lý của suy diễn $P \rightarrow Q$ cũng là 1 hoặc 0; do đó, ta có thể xác định được giá trị chân lý của kết luận Q đó là 1 hoặc 0.

Nếu suy diễn là tri thức không chắc chắn thì giá trị chân lý của tiền điều kiện P là ở trong khỏang 0 và 1 và giá trị chân lý của suy diễn cũng là ở trong khỏang 0 và 1; vậy thì bằng cách nào để xác định giá trị chân lý của kết luận Q ?.

Để lý giải với loại tri thức không chắc chắn sử dụng lý thuyết không chắc chắn đó là lý thuyết xác suất hay lý thuyết logic mờ. Hai loại lý thuyết này còn được gọi là logic nhiều chữ số ở giữa 0 và 1.

6.2) **Xử Lý Tri Thức Không Chắc Chắn Dùng Lý Thuyết Xác Suất** :

1) **Lý thuyết xác suất** :

Lý thuyết xác suất là bắt nguồn từ thực nghiệm, điều đó có nghĩa là thông qua thực nghiệm, có tồn tại một vài đại lượng P(E) được gọi là xác suất của biến cố E đó là độ tin cậy của E với các ràng buộc là

$$0 \le P(E) \le 1 \text{ và } P(E) + P(\neg E) = 1.$$

Giả sử có một cái túi lớn chứa nhiều quả bóng, trong đó một số quả bóng có đánh nhãn chữ cái a, một số quả bóng có đánh nhãn chữ cái b, một số quả bóng khác có đánh nhãn chữ cái a và b, và mộ số quả bóng không có đánh nhãn.

Bằng thực nghiệm, trộn đều các quả bóng trong túi, lấy các quả bóng ra từ túi và bỏ ngược chúng lại vào túi. Đếm số lần lặp lại của các quả bóng có nhãn a, số lần lặp lại của các quả bóng có nhãn b và số lần lặp lại của các quả bóng có nhãn a và b.

Cho n1 là số lần lặp lại của các quả bóng có nhãn a, n2 là số lần lặp lại của các quả bóng có nhãn b, n3 là số lần lặp lại của các quả bóng có nhãn a và b và n là tổng số của các quả bóng chứa trong túi.

Xác suất của hai biến cố a và b xảy ra độc lập trên cơ sở luật giao hóan được định nghĩa là

+ Xác suất của a ký hiệu là $P(a) = n1/n$.
+ Xác suất của được ký hiệu là $P(b) = n2/n$.
+ Xác suất của a và b được ký hiệu là $P(a \wedge b) = n3/n$.
+ Xác suất điều kiện a cho bởi biến cố b được ký hiệu là

$$P(a \backslash b) = n3/n2 = P(a \wedge b)/P(b).$$

+ Xác suất điều kiện b cho bởi biến cố a được ký hiệu là

$$P(b \backslash a) = n3/n1 = P(a \wedge b)/P(a).$$

Xác suất của hai biến cố a hoặc b xảy ra phụ thuộc trên cơ sở luật giao hợp được định nghĩa là

+ Xác suất của a là $P(a) = n1/n + n3/n = P(\neg b \wedge a) + P(a \wedge b)$.
+ Xác suất của b là $P(b) = n2/n + n3/n = P(\neg a \wedge b) + P(a \wedge b$.

+ Xác suất của a hoặc b là

$$P(a \vee b) = n1/n + n2/n + n3/n = P(a) + P(b) - P(a \wedge b).$$

Lý giải với tri thức không chắc chắn sử dụng lý thuyết xác suất để xác định giá trị xác suất của kết luận a hoặc b với các phương trình là

+ $P(a) = P(\neg b \wedge a) + P(a \wedge b) = P(\neg b) \times P(a \backslash \neg b) + P(b) \times P(a \backslash b)$.

+ $P(b) = P(\neg a \wedge b) + P(a \wedge b) = P(\neg a) \times P(b \backslash \neg a) + P(a) \times P(b \backslash a)$.

2) **<u>Lý giải chính xác dưới điều kiện không chắc chắn dùng xác suất</u>** :

Để lý giải chính xác dưới điều kiện không chắc chắn, mỗi bằng chứng và mỗi suy diễn phải được kèm theo số đo xác suất đó là độ tin cậy của bằng chứng và suy diễn.

Giả sử có luật suy diễn với dạng là

If a then b.

Cách tính xác xuất của kết luận b với luật suy diễn này là

$$P(b) = P(a) \times P(b \backslash a) + P(\neg a) \times P(b \backslash \neg a)$$

trong đó, P(a) là xác suất của có mặt bằng chứng a, $P(b \backslash a)$ là xác suất điều kiện b cho bởi có mặt bằng chứng a đó chính là xác suất của suy diễn if a then b, $P(\neg a)$ là xác suất của không có mặt bằng chứng a và $P(b \backslash \neg a)$ là xác suất điều kiện b cho bởi không có mặt bằng chứng a .

Giả sử cho luật suy diễn với dạng là

If (a and b) then c.

Cách tính xác suất của kết luận c với luật suy diễn này là

$$P(c) = P(c \backslash a \wedge b) \times p(a \wedge b) + P(c \backslash \neg(a \wedge b)) \times P(\neg(a \wedge b))$$

trong đó, $P(c \backslash a \wedge b)$ là xác suất điều kiện c cho bởi bằng chứng a và b, $p(a \wedge b)$ là xác suất của bằng chứng a và b, $P(c \backslash \neg(a \wedge b))$ là xác suất điều kiện c cho bởi không có bằng chứng a và b và $P(\neg(a \wedge b))$ là xác suất của không có bằng chứng a và b.

Giả sử cho luật suy diễn với dạng là

If (a or b) then c.

Cách tính xác suất của kết luận c với luật suy diễn này là

$$\begin{aligned} P(c) = {} & P(c \backslash a \wedge b) \times p(a \wedge b) \\ & + P(c \backslash a \wedge \neg b) \times p(a \wedge \neg b) \\ & + P(c \backslash \neg a \wedge b) \times P(\neg a \wedge b) \\ & + P(c \backslash \neg a \wedge \neg b) \times P(\neg a \wedge \neg b). \end{aligned}$$

Ví dụ : Cho luật suy diễn là

Nếu số người có bệnh tim thì trong số đó sẽ có một số người bị bệnh phổi.

Cho H là số người có bệnh tim và C là một số người trong số đó sẽ bị bệnh phổi, vậy thì luật suy diễn trên có thể được viết lại là

$$H \to C.$$

Qua thực nghiệm khảo sát cho thấy rằng :

+ Cứ 100 người, trong đó có 10 người bị bệnh tim. Vì thế xác suất của số người có bệnh tim là $P(H) = 0,1$.

+ Cứ 100 người, trong đó có 90 người không bị bệnh tim. Vì thế xác suất của những người không có bệnh tim là $P(\neg H) = 0,9$.

+ Cứ 100 người có bệnh tim thì trong số đó có 90 người bị bệnh phổi. Vì thế xác suất điều kiện số người bị bệnh phổi cho bởi số người có bệnh tim là $P(C\backslash H) = 0,9$.

+ Cứ 100 người không có bệnh tim thì trong số đó có 95 người không bị bệnh phổi. Do đó, xác suất điều kiện số người không bị bệnh phổi cho bởi số người không có bệnh tim là $P(\neg C\backslash\neg H) = 0,95$.

+ Cứ 100 người không có bệnh tim thì trong số có 5 người bị bệnh phổi. Do đó, xác suất điều kiện số người bị bệnh phổi cho bởi số người không có bệnh tim là $P(C\backslash\neg H) = 0,05$.

Ta có xác suất của luật suy diễn $H \to C$ đó chính là xác suất điều kiện C cho bởi bằng chứng H đó là $P(C\backslash H) = 0,9$.

Công thức tính xác suất của kết luận C với dạng luật suy diễn $H \to C$ là

$$P(C) = P(H)\times P(C\backslash H) + P(\neg H)\times P(C\backslash\neg H).$$

Vậy thì ta có xác suất của kết luận C là

$$P(C) = 0,1\times 0,9 + 0,9\times 0.05 = 0,135 \text{ hay } 13,5\%.$$

Với lý giải chính xác dưới điều kiện không chắc chắn dùng xác suất cho các luật suy diễn dạng phức tạp, công việc tính xác suất của vế kết luận sẽ xuất hiện nhiều ẩn số xác suất chưa biết trong công thức tính xác suất. Để khắc phục điều này, công

việc tính xấp xỉ cải tiến từ công thức tính xác suất của định luật Baye được thiết lập.

3) Lý thuyết chắc chắn :

Giả sử cho luật suy diễn là

If a then b.

Xác suất có mặt của kết luận b là P(b) và xác suất không có mặt của kết luận b là P(¬b). Vậy thì, tổng giá trị của hai loại xác suất này phải là P(b) + P(¬b) = 1. Xác suất điều kiện b cho bởi a là P(b\a).

Công việc lý giải dưới điều kiện không chắc chắn là cách xác định độ tin cậy của kết luận b với mỗi bằng chứng a. Độ tin cậy này có thể tăng hoặc giảm điều đó còn phụ thuộc vào độ tin cậy của mỗi bằng chứng a.

Với ý tưởng này, hai đại lượng số đo độ tin cậy mới được đề xuất cho kết luận b đó là MB và MD. Hai đại lượng này bị chặn bởi 0 và 1 đó là

$$0 \le \text{MB} \le 1 \text{ và } 0 \le \text{MD} \le 1$$

trong đó, MB là số đo độ tin cậy của kết luận b và MD là số đo độ không tin cậy của kết luận b.
Vậy thì, cho mỗi bằng chứng a, hai đại lượng số đo độ tin cậy và độ không tin của kết luận b này được thiết lập là

$$MB(b,a) = \begin{cases} 1 & if \quad P(b) = 1 \\ \dfrac{\max[P(b \setminus a), P(b)] - P(b)}{1 - P(b)} \end{cases}$$

$$MD(b,a) = \begin{cases} 1 & if \quad P(b) = 0 \\ \dfrac{\min[P(b \setminus a), P(b)] - P(b)}{-P(b)} \end{cases}$$

Trên cơ sở số đo độ tin cậy và số đo độ không tin cậy của kết luận b, một đại lượng số đo độ tin cậy khác được đề xuất đó là số đo chắc chắn của kết luận b với mỗi bằng chứng a. Số đo này bị chặn bởi –1 và 1 đó là $-1 \le \text{CF(b,a)} \le 1$ và được thiết lập là

CF(b,a) = MB(b,a) – MD(b,a)

+ Nếu số đo chắc chắn của kết luận b với bằng chứng a là CF(b,a) = -1 thì kết luận rằng b là sai.

+ Nếu số đo chắc chắn của kết luận b với bằng chứng a là CF(b,a) = 0 thì kết luận rằng b là chưa biết.

+ Nếu số đo chắc chắn của kết luận b với bằng chứng a là CF(b,a) = 1 thì kết luận rằng b là đúng.

Khảo sát các phương trình trên với các trường hợp là

+ **Trường hợp 1** : Bằng chứng a dẫn đến kết luận b là đúng hay nói cách khác, xác suất điều kiện b cho bởi a là đúng.

Với trường hợp này, ta có P(b\a) = 1 và P(b) = 1; do đó ta có MB(b,a) = 1 và MD(b,a) = 0. Vậy thì CF(b,a) = 1; do đó ta kế luận rằng b là đúng.

+ **Trường hợp 2** : Bằng chứng a dẫn đến kết luận b là sai hay nói cách khác, xác suất điều kiện không có mặt b cho bởi a là đúng.

Với trường hợp này, ta có P(¬b\a) = 1 và P(b) = 0; do đó ta có MB(b,a) = 0 và MD(b,a) = 1. Vậy thì CF(b,a) = -1; do đó ta có thể kết luận rằng b là sai.

+ **Trường hợp 3** : Không có mặt bằng chứng a dẫn đến kết luận b.

Với trường hợp này, ta có P(b\a) = P(b); do đó MB(b,a) = 0 và MD(b,a) = 0.

Vậy thì CF(b,a) = 0 và do đó ta kết luận rằng b là chưa biết.

+ **Trường hợp 4** : Bằng chứng khả thi a dẫn đến kết luận b.

Với trường hợp này, ta tcó xác suất điều kiện b cho bởi a bị chặn bởi là

P(b) < P(b\a) < 1.

Vì thế MB và MD được xác định là

$$MB(b,a) = \frac{P(b \setminus a) - P(b)}{1 - P(b)}$$

và MD(b,a) = o.

Do đó, CF(b,a) = MB(b,a) là một số dương. Điều này chứng tỏ rằng kết luận b là khả thi.

+ **Trường hợp 5** : Bằng chứng không khả thi dẫn đến kết luận b.

Với trường hợp này, xác suất điều kiện b cho bởi a bị chặn bởi là

$$0 < P(b\backslash a) < P(b).$$

Ví thế MB và MD được xác định là

$$MB(b,a) = 0$$

Và $$MD(b,a) = \frac{P(b) - P(b \backslash a)}{P(b)}.$$

Do đó, ta có CF(b,a) = - MD(b,a) là một số âm. Điều này chứng tỏ rằng kết luận b là không khả thi.

4) <u>Lý giải xấp xỉ dưới điều kiện không chắc chắn dùng lý thuyết số đo chắc chắn :</u>

Để lý giải xấp xỉ dưới điều kiện không chắc chắn dùng số đo chắc chắn, mỗi bằng chứng và mỗi luật suy diễn phải được kèm theo số đo chắc chắn. Theo lý thuyết, số đo chắc chắn của mỗi bằng chứng hoặc luật suy diễn phải bị chặn bởi là $-1 \le CF \le 1$.

Cho luật suy diễn với dạng là

If a then b

Với số đo chắc chắn của bằng chứng a được kèm theo là CF(a) và số đo chắc chắn của luật suy diễn được kèm theo là CF(rule). Vậy thì, số đo chắc chắn của kết luận b với dạng luật suy diễn này có thể được tính bằng công thức là

$$CF(b,a) = CF(a) \times CF(rule).$$

Cho luật suy diễn với dạng là

If a_1 and a_2 . . . and a_m then b

Với các số đo chắc chắn của các bằng chứng a_1, a_2 . . . . . a_m được kèm theo là $CF(a_1)$, $CF(a_2)$, ...,$CF(a_m)$ và số đo chắc chắn của luật suy diễn được kèm theo là CF(rule). Vậy thì, số đo chắc chắn của kết luận b với dạng luật suy diễn này được tính bằng công thức là

$$CF(b, a_1 \text{ and } a_2, \ldots \text{and } a_m) = \min\{CF(a_i)\} \times CF(rule).$$

Trong đó, min là hàm trả về giá trị cực tiểu của các số đo chắc chắn của các bằng chứng a_i.

Cho luật suy diễn với dạng là

If a_1 or a_2 or . . . or a_m then b

Với các số đo chắc chắn của các bằng chứng và luật suy diễn được kèm theo là như trên. Vậy thì, số đo chắc chắn của kết luận b với dạng luật này được tính bằng công thức là

$$CF(b, a_1 \text{ or } a_2, \ldots \text{ or } a_m) = \max\{CF(a_i)\} \times CF(\text{rule}).$$

Trong đó, max là hàm trả về giá trị cực đại của các số đo chắc chắn của các bằng chứng a_i.

+ Cách tính số đo chắc chắn của kết luận b được hổ trợ từ hai hoặc nhiều nguồn luật suy diễn khác nhau có cùng kết luận b :

Giả sử ta có hai luật suy diễn là

Rule1: If a_1 then b

Rule2: If a_2 then b

Với trường hợp này, số đo chắc chắn tổng hợp của kết luận b được tính bằng công thức là

$$CF(CF(b,a_1), CF(b.,a_2)) = \begin{cases} CF(b,a_1) + CF(b,a_2) \times (1 - CF(b,a_1)) & \text{if} \quad both > 0. \\ \dfrac{CF(b,a_1) + CF(b,a_2)}{1 + \min\{|CF(b,a_1)|, |CF(b,a_2)|\}} & \text{if} \quad one\ of\ them < 0. \\ CF(b,a_1) + CF(b,a_2) \times (1 + cf(b,a_1)) & \text{if} \quad both < 0. \end{cases}$$

Trong đó, CF(b,a_1) là số đo chắc chắn của kết luận b với rule1 và CF(b,a_2) là số đo chắc chắn của kết luận b với rule2.

6.3) **<u>Xử Lý Tri Thức Không Chắc Chắn Dùng Logic Mờ</u>** :

Một phương pháp xử lý tri thức không chắc chắn khác đó là logic mờ. Một hệ thống xử lý tri thức không chắc chắn dùng logic được mô tả bằng lưu đồ khối như hình

Một hệ thống xử lý tri thức không chắn dùng logic mờ gồm có biến vào ra X, Y của hệ thống, khâu mờ hóa, cơ sở tri thức mờ, kỹ thuật suy diễn mờ và khậu giải mờ.

+ Khâu mờ hóa : chuyển đại lượng rõ từ ngõ vào X sang đại lượng mờ $\mu_A(X)$.

+ Cơ ở tri thức mờ : gồm cơ sở dữ liệu mờ và cơ sở luật suy diễn mờ. Cơ sở dữ liệu mờ là các tập mờ vào ra của hệ thống và cơ sở luật suy diễn mờ là tập các luật suy diễn mờ được thể hiện dưới dạng luật If-Then đó là tập luật mô tả tổng quát cách giải một bài tóan mờ.

+ Kỹ thuật suy diễn mờ : phương pháp xác định tập mờ ngõ ra của hệ thống.

+ Khâu giải mờ : chuyển đại lượng mờ $\mu_B(Y)$ sang đại lượng ro Ỹ.

1) **<u>Tập mờ và các phép tóan trên các tập mờ</u>** :

+ **Tập rõ** : Cho x là phần tử của cơ sở X và A là tập con của X. A được gọi là tập rõ trong X, nếu A được định nghĩa bằng hàm liên thuộc là

$$\mu_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

+ **Tập mờ** : Cho x là phần tử của cơ sở X và A là tập con của X. A được gọi là tập mờ trong X, nếu A được định nghĩa bằng hàm liên thuộc của nó sao cho bị chặn giữa 0 và 1 đó là

$$0 \le \mu_A(X) \le 1.$$

+ **Biểu diễn tập mờ** :

- Nếu X là tập cơ sở liên tục, tập mờ A trong X được biểu diễn là

$$A = \int_X \frac{\mu(x)}{x} dx$$

Trong đó, ký hiệu $\int$ là tóan tử hợp và – là tóan tử kết hợp giữa đại lượng rõ và đại lượng mờ.

- Nếu X là tập cơ sở rời rạc, thì tập mờ A trong X được biểu diễn là

$$A = \sum_{i=1}^{n} \mu_A(x_i) / x_i$$

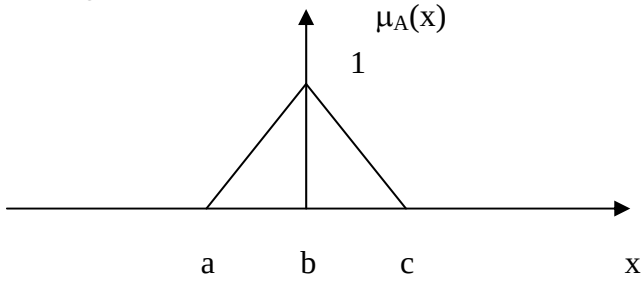
Trong đó, ký hiệu $\sum$ là tóan tử hợp và ký hiệu / là tóan tử kết hợp giữa giá trị rõ và giá trị mờ tương ứng.

+ **Hàm liên thuộc** : Có hai cách xây dựng hàm liên thuộc cho tập mờ A đó là xây dựng hàm liên thuộc dưới dạng bảng và xây dựng hàm liên thuộc dưới dạng hàm.

- Hàm liên thuộc dưới dạng bảng gồm hai cột và nhiều hàng, cột thứ nhất chứa giá trị rõ và cột chứa các giá trị mờ tương ứng được mô tả tổng quát như bảng

Đại lượng rõ x_i	Đại lượng mờ $\mu_A(x_i)$
x_1	$\mu_A(x_1)$
x_2	$\mu_A(x_2)$
x_n	$\mu_A(x_n)$

- Hàm liên thuộc dưới dạng hàm có nhiều hàm khác nhau nhưng hàm liên thuộc dạng tam giác là được sử dụng phổ biến nhất. Cho đồ thị biểu diễn tập mờ A dạng tam giác như hình



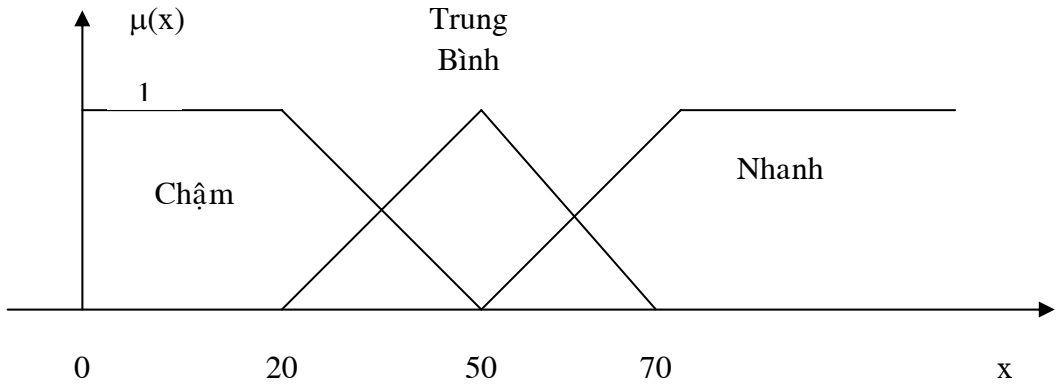
Hàm liên thuộc dạng tam giác được thiết lập là

$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a} & if\ a \le x \le b. \\ \dfrac{c-x}{c-b} & if\ b \le x \le c. \end{cases}$$

trong đó, a là cận trái, b là tâm và c là cận phải của tam giác trên trục hòanh x.

+ **Biến ngôn ngữ** : Các biến rõ vào ra của hệ thống mờ được gọi là các biến ngôn ngữ, vì chúng được mô tả dưới dạng ngôn ngữ tự nhiên như nhanh, chậm, ít, nhiều vân vân. Các đại lượng ngôn ngữ này đó chính là các tập mờ vào ra được định nghĩa trên các biến vào ra của hệ thống.

Ví dụ : Cho x là biến ngôn ngữ biểu diễn tốc độ của xe được mô tả bằng các tập mờ như nhanh, trung bình và chậm được biểu diễn bằng đồ thị như hình



+ **Các phép tóan trên các tập mờ** : Để làm việc trên các tập mờ, có các phép tóan là

- **Phép tóan giao** : Cho A và B là hai tập mờ trong tập cơ sở X. Tập mờ của phép tóan giao A và B cũng là tập mờ trong X với hàm liên thuộc là
$$\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$$
- **Phép tóan hợp** : Cho A và B là hai tập mờ trong X. Tập mờ của phép tóan hợp A và B cũng là tập mờ trong X với hàm liên thuộc là
$$\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$$
- **Phép tóan bù** : Cho $\bar{A}$ là tập bù của tập mờ A trong tập cơ sở X. $\bar{A}$ cũng là tập mờ trong X với hàm liên thuộc là
$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

2) **Quan hệ mờ và các phép tóan trên quan hệ mờ** :

+ **Tập tích của hai tập cơ sở** : cho X và Y là hai tập cơ sở với x∈X và y∈Y. Tập tích của hai tập cơ sở X và Y được định nghĩa là
$$X \times Y = \{(x, y) / x \in X, y \in Y\}$$

+ **Quan hệ rõ** : Cho R là tập con của tập tích X×Y, R được gọi là quan hệ rõ trong X×Y, nếu R được định nghĩa bằng hàm liên thuộc là

$$\mu_R(x,y) = \begin{cases} 1 & if\ (x,y) \in R \\ 0 & if\ (x,y) \notin R \end{cases}$$

+ **Quan hệ mờ** : Cho R là tập con của tập tích X×Y, R được gọi là quan hệ mờ trong X×Y, nếu R được định nghĩa bằng hàm liên thuộc của nó sao cho bị chặn giữa 0 và 1 đó là

$$0 \le \mu_R(x,y) \le 1.$$

+ **Biểu diễn quan hệ mờ** : Quan hệ mờ có thể được biểu diễn dưới dạng ma trận là

$$R(x,y) = \begin{bmatrix} \mu_R(x_1,y_1) & \mu_R(x_1,y_2). & . & . & \mu_R(x_1,y_n) \\ & & & & \\ & & & & \\ \mu_R(x_m,y_1) & \mu_R(x_m,y_2). & . & . & \mu_R(x_m,y_n) \end{bmatrix}$$

+ **Các phép tóan trên các quan hệ mờ** : Cho P là quan hệ mờ trong tập tích X×Y và Q là quan hệ mờ trong tập tích Y×Z. Quan hệ mờ trong tập tích X×Z được xác định bằng phương trình là

$$R = PoQ$$

Trong đó ký hiệu o là tóan tử hợp thành mờ.

Có nhiều loại tóan tử hợp thành mờ, tuy nhiên hai loại tóan tử hợp thành mờ thông dụng nhất đó là tóan tử max-min và tóan tử max-product.

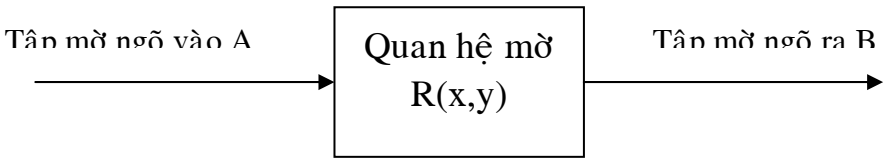
- Tóan tử max-min được thiết lập là

$$\mu_R(x,z) = \mu_{P\circ Q}(x,z) = \max\min\left\{\mu_P(x,y), \mu_Q(y,z)\right\}$$

- Tóan tử max-product được thiết lập là

$$\mu_R(x,z) = \mu_{P\circ Q}(x,z) = \max\left\{\mu_P(x,y) \times \mu_Q(y,z)\right\}$$

+ Phương trình quan hệ mờ : Cho A là tập mờ ngõ vào trên biến ngôn ngữ vào X, R là quan hệ mờ trong tập tích X×Y và B là tập mờ ngõ ra trên biến ngôn ngữ ngõ ra Y. Quan hệ vào ra của hệ thống mờ này được mô tả bằng lưu đồ khối như hình



Phương trình quan hệ mờ xác định tập mờ ngõ ra của hệ thống được thiết lập là

$$B = AoR$$

Trong đó, ký hiệu o là tóan tử hợp thành mờ max-min hoặc max-product như đã được thiết lập trên.

3) **Logic mờ và lý giải xấp xỉ mờ :**

+ Logic mờ : Logic mờ là logic mà giá trị chân lý của đề xuất không bị hạn chế bởi hai chữ số 0 và 1 như logic rõ hai chữ số. Giá trị chân lý của một đề xuất trong logic mờ có thể được gán cho một giá trị bất kỳ giữa 0 và 1.

Cho đề xuất P với $x \in A$, trong đó A là tập mờ trong tập cơ sở X với hàm liên thuộc là $\mu_A(x)$. Khi đó giá trị chân lý của đề xuất P là

$$T(P) = \mu_A(x)$$

trong đó, $\mu_A(x)$ là bị chặn bởi giữa khỏang 0 và 1 đó là

$$0 \le \mu_A(x) \le 1.a2$$

- **Phép tóan phủ định của đề xuất P** :

Cho đề xuất P với $x \in A$, trong đó A là tập mờ trong tập cơ sở X với hàm liên thuộc là $\mu_A(x)$. Phủ định của đề xuất P là $x \notin A$. Do đó, giá trị chân lý của $\neg P$ được thiết lập là

$$T(\neg P) = 1 - T(P).$$

- **Phép tóan logic hợp của đề xuất P và Q** :

Cho đề xuất P với $x \in A$ và đề xuất Q với $x \in B$, trong đó A và B là hai tập mờ trong tập cơ sở X với các hàm liên thuộc là $\mu_A(x)$ và $\mu_B(x)$. Khi đó phép tóan logic hợp của P và Q là

$P \vee Q$: $x \in A$ hoặc $x \in B$.

Do đó giá trị chân lý của phép tóan hợp P và Q được thiết lập là

$$T(P \vee Q) = \max\{T(P), T(Q)\}.$$

- **Phép tóan logic giao của đề xuất P và Q** :

Cho đề xuất P với $x \in A$ và đề xuất Q với $x \in B$, trong đó A và B là hai tập mờ trong tập cơ sở X với các hàm liên thuộc là $\mu_A(x)$ và $\mu_B(x)$. Khi đó phép tóan logic giao của P và Q là

$P \wedge Q$: $x \in A$ và $x \in B$.

Do đó, giá trị chân lý của phép tóan giao P và Q được thiết lập là

$$T(P \wedge Q) = \min\{T(P), T(Q)\}.$$

- **Phép tóan logic kéo theo** :

Cho đề xuất P với $x \in A$ và đề xuất Q với $x \in B$, trong đó A và B là hai tập mờ trong tập cơ sở X với các hàm liên thuộc là $\mu_A(x)$ và $\mu_B(x)$. Khi đó phép tóan logic kéo theo P cho Q là

$$P \rightarrow Q : \quad x \in A \rightarrow x \in B.$$

Do đó, giá trị chân lý của phép tóan kéo theo P cho Q được thiết lập là

$$T(P \rightarrow Q) = T(\neg P \vee Q) = \max\{T(\neg P), T(Q)\}.$$

Xét luật suy diễn mờ với dạng là

$$P \rightarrow Q \text{ if x is A then y is B,}$$

trong đó, A là tập mờ ngõ vào trong tập cơ sở ngõ vào X với hàm liên thuộc là $\mu_A(x)$ và B là tập mờ ngõ ra trong tập cơ sở ngõ ra Y với hàm liên thuộc là $\mu_A(x)$.

Mô hình luật suy diễn mờ này là tương đương với quan hệ mờ là

$$R = (A \times B) \vee (\neg A \times Y).$$

Do đó hàm liên thuộc của nó được thiết lập là

$$\mu_R(x,y) = \max[\mu_A(x) \wedge \mu_B(y), (1 - \mu_A)].$$

<u>Ví dụ</u> : Cho X là tập cơ sở ngõ vào biểu diễn tốc độ động cơ và A là tập mờ ngõ vào biểu diễn tốc độ động cơ an tòan trong X được thu thập từ thực nghiệm là

$$A = \{0.3/20 + 0.6/30 + 0.8/40 + 1/50 + 0.7/60 + 0.4/70\}.$$

Cho Y là tập cơ sở ngõ ra biểu diễn điện áp động cơ và B là tập mơ ngõ rà biểu diễn điện áp động cơ bình thường được thu thập từ thực nghiệm là

$$B = \{0.1/1 + 0.3/2 + 0.8/3 + 1/4 + 0.7/5 + 0.4/6 + 0.2/7\}.$$

Quan hệ mờ giữa tốc độ động cơ an tòan và điện áp động cơ bình thường được thiết lập là

$$R = x \in A \rightarrow y \in B = (A \times B) \vee (\neg A \times Y).$$

Từ đây, ta có quan hệ mờ R là

$$R = \begin{bmatrix} 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \\ 0.4 & 0.4 & 0.6 & 0.6 & 0.6 & 0.4 & 0.2 \\ 0.2 & 0.3 & 0.8 & 0.8 & 0.7 & 0.4 & 0.2 \\ 0.1 & 0.3 & 0.8 & 1.0 & 0.7 & 0.4 & 0.2 \\ 0.3 & 0.3 & 0.7 & 0.7 & 0.7 & 0.4 & 0.2 \\ 0.6 & 0.6 & 0.6 & 0.6 & 0.6 & 0.6 & 0.6 \end{bmatrix}$$

+ **Lý giải xấp xỉ mờ** :

Giả sử ta có luật suy diễn mờ với dạng là

R = if x is A then y is B,

trong đó, A và B là hai đề xuất mờ biểu diễn tốc độ động cơ an tòan và điện áp động cơ bình thường với quan hệ mờ R được xác định là

$$R = \begin{bmatrix} 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \\ 0.4 & 0.4 & 0.6 & 0.6 & 0.6 & 0.4 & 0.2 \\ 0.2 & 0.3 & 0.8 & 0.8 & 0.7 & 0.4 & 0.2 \\ 0.1 & 0.3 & 0.8 & 1.0 & 0.7 & 0.4 & 0.2 \\ 0.3 & 0.3 & 0.7 & 0.7 & 0.7 & 0.4 & 0.2 \\ 0.6 & 0.6 & 0.6 & 0.6 & 0.6 & 0.6 & 0.6 \end{bmatrix}$$

Cho một luật suy diễn mờ khác với dạng là

if x is A' then y is B',

trong đó, A' là đề xuất mờ biểu diễn tốc độ động cơ hơi chậm và B' là đề xuất mờ biểu diễn điện áp động cơ hơi chậm.

Nếu biết tập mờ ngõ vào A' và quan hệ mờ R thì tập mờ ngõ ra B' có thể được xác định bằng phương trình là

B' = A'oR

Trong đó, ký hiệu o là tóan tử hợp thành mờ.

Giả sử cho tập mờ ngõ vào A' là

A' = {0.4/20 + 0.7/30 + 1/40 + 0.6/50 + 0.3/60 + 0.1/70}.

Khi đó, tập mờ ngõ ra B' được xác định với phép tóan hợp thành mờ max-min là

B' = A'oR = {0.4/1 + 0.4/2 + 0.8/3 + 0.8/4 + 0.7/5 + 0.4/6 + 0.4/7}.

4) Cơ sở tri thức mờ :

Cơ sở tri thức mờ gồm có cơ sở dữ liệu mờ và cơ sở luật suy diễn mờ.

+ Cơ sở dữ liệu mờ bao gồm các tập mờ và các hàm liên thuộc của các tập mờ được định nghĩa trên các biến ngôn ngữ vào ra của hệ thống.

+ Cơ sở luật suy diễn mờ đó là bao gồm tất cả các luật suy diễn mờ thể hiện dưới dạng If-then mô tả đặc tính động học của hệ thống vạch ra cách giải quyết một bài tóan mờ. Mô hình luật suy diễn mờ tổng quát nhất của luật thứ i là

R_i : If x_1 is A_{i1} and x_2 is A_{i2} . . . . and x_j is A_{ij} and. . . and x_m is A_{im} then y is B_i.
Trong đó, A_{ij} là các tập mờ ngõ vào với hàm liên thuộc là $\mu_{A_{ij}}(x_j)$ và B_i là tập mờ ngõ ra của hệ thống với hàm liên thuộc là $\mu_{B_i}(y)$.
Với mô hình luật dạng thể loại này, số đo mờ của vế điều kiện được xác định bởi công thức là

$$\alpha_i = \min\left\{\mu_{A_{ij}}(x_j)\right\} \quad for\ j = 1,\ldots,m$$

5) Kỹ thuật suy diễn mờ :

Kỹ thuật suy diễn mờ là phương pháp xác định tập mờ ngõ ra của hệ thống. Có hai phương pháp pháp xác định tập mờ ngõ ra của hệ thống đó là kỹ thuật suy diễn mờ max-min và thuật suy diễn mờ max-product.

Cho hệ thống mờ gồm có số n luật suy diễn mờ, kỹ thuật suy diễn mờ là lần lượt xác định tập mờ ngõ ra của từng luật theo thứ tự từ luật thứ nhất đến luật thứ n dùng phép tóan min hoặc product và sau đó, tập hợp của tất cả các tập mờ ngõ ra đó chính là tập mờ ngõ ra của hệ thống dùng phép tóan max.

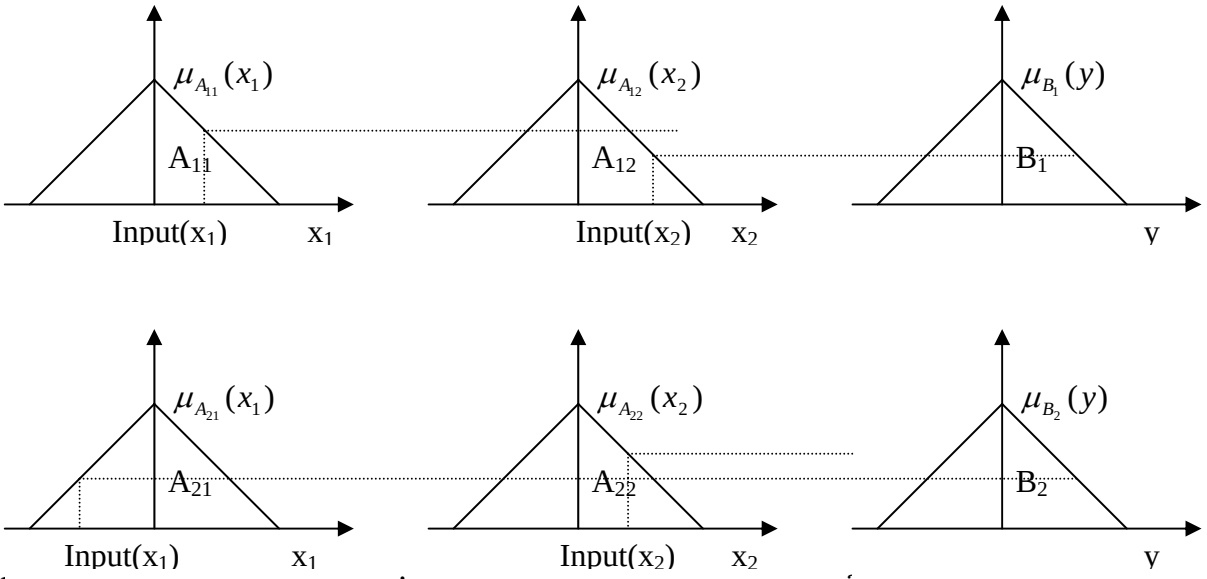
Giả sử cho hệ thống mờ gồm hai luật với mô hình luật dạng là

R_1 : If x_1 is A_{11} and x_2 is A_{12} then y is B_1

R_2 : If x_1 is A_{21} and x_2 is A_{22} then y is B_2

Trong đó, A_{11}, A_{12}, A_{21}, A_{22} là các tập mờ ngõ vào của hệ thống với các hàm liên thuộc là $\mu_{A_{11}}(x_1)$, $\mu_{A_{12}}(x_2)$, $\mu_{A_{21}}(x_1)$, $\mu_{A_{22}}(x_2)$ và B_1, B_2 là các tập mờ ngõ ra của hệ thống với các hàm liên là $\mu_{B_1}(y)$, $\mu_{B_2}(y)$.

+ **Kỹ thuật suy diễn mờ max-min** : Giả sử các hàm liên thuộc vào ra của hệ thống là dạng tam giác, kỹ thuật suy diễn mờ max-min được mô tả bằng đồ thị như hình



Kỹ thuật suy diên mờ max-min xác định tập mờ ngõ ra của hệ thông được mô tả như sau :

- Tập mờ ngõ ra B_1' của luật thứ nhất được xác định với hàm liên thuộc của nó là

$$\mu_{B_1'}(y) = \min\{\alpha_1, \mu_{B_1}(y)\}$$

trong đó, α_1 là số đo mờ ở vế điều kiện của luật 1 được xác định là

$$\alpha_1 = \min\{\mu_{A_{11}}(x_1), \mu_{A_{12}}(x_2)\}$$

- Tập mờ ngõ ra B_2' của luật thứ 2 được xác định với hàm liên thuộc của nó là

$$\mu_{B_2'}(y) = \min\{\alpha_2, \mu_{B_2}(y)\}$$

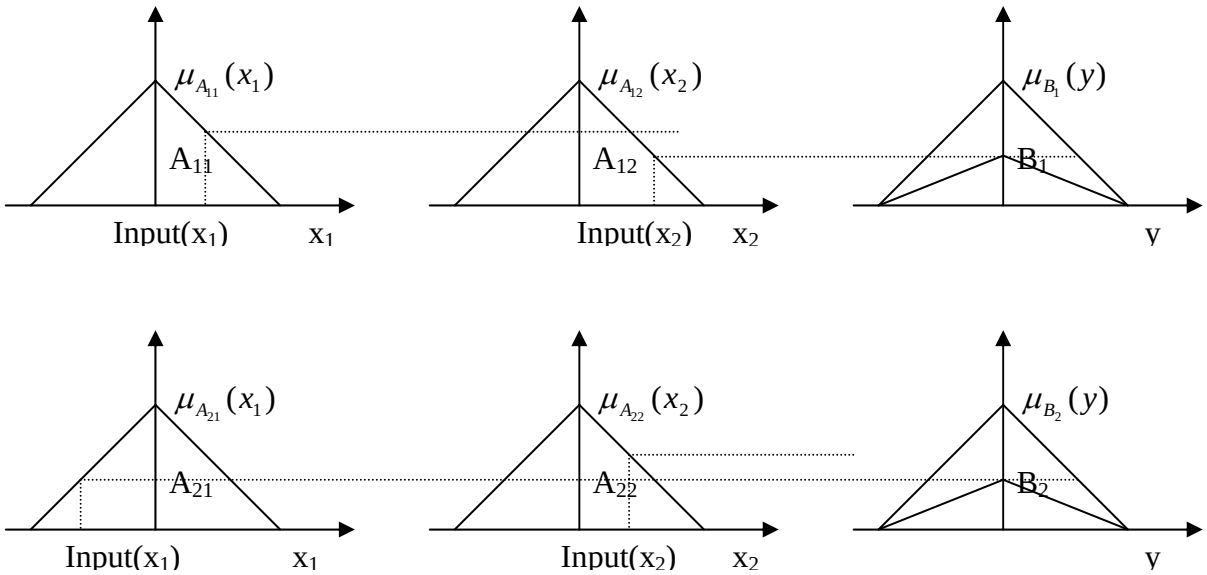
trong đó, α_2 là số đo mờ ở vế điều kiện của luật 2 được xác định là

$$\alpha_2 = \min\{\mu_{A_{21}}(x_1), \mu_{A_{22}}(x_2)\}$$

- Tập mờ ngõ ra B' của hệ thống đó chính là tập hợp B' của hai tập mờ ngõ ra B_1' và B_2' trước đó của hai luật là B' = $B_1' \vee B_2'$ và nó được định bằng hàm liên thuộc của nó là

$$\mu_{B'}(y) = \max\{\mu_{B_1'}(y), \mu_{B_2'}(y)\}$$

+ **<u>Kỹ thuật suy diễn mờ max-product</u>** : Cũng giống như kỹ thuật suy diễn mờmax-min, kỹ thuật suy diễn mờ max-product được mô tả bằng đồ thị như hình



Kỹ thuật suy diễn mờ max-product xác định tập mờ ngõ ra của hệ thống được mô tả như sau :

- Tập mờ ngõ ra B_1' của luật thứ nhất được xác định với hàm liên thuộc của nó là

$$\mu_{B_1'}(y) = \alpha_1 \times \mu_{B_1}(y)$$

trong đó, α_1 là số đo mờ ở vế điều kiện của luật 1 được xác định là

$$\alpha_1 = \min\left\{\mu_{A_{11}}(x_1), \mu_{A_{12}}(x_2)\right\}$$

- Tập mờ ngõ ra B_2' của luật thứ 2 được xác định với hàm liên thuộc của nó là

$$\mu_{B_2'}(y) = \alpha_2 \times \mu_{B_2}(y)$$

trong đó, α_2 là số đo mờ ở vế điều kiện của luật 2 được xác định là

$$\alpha_2 = \min\left\{\mu_{A_{21}}(x_1), \mu_{A_{22}}(x_2)\right\}$$

- Tập mờ ngõ ra B' của hệ thống đó chính là tập hợp B' của hai tập mờ ngõ ra B_1' và B_2' trước đó của hai luật là B' = $B_1' \vee B_2'$ và nó được định bằng hàm liên thuộc của nó là

$$\mu_{B'}(y) = \max\left\{\mu_{B_1'}(y), \mu_{B_2'}(y)\right\}$$

Chương 7 : Việc Học Máy

7.1) Việc Học Máy Là Gì ?

Con người có nhiều cách học như học ký ức, học các sự kiện nhờ thông qua sự quan sát và thăm dò, học cải thiện kỹ xảo thông qua thực tiễn, học nhờ sự phát triển của hệ thần kinh sinh học con người và học nhờ gen di truyền từ các thế hệ trước.

Dù cách học nào đi chăng nữa, mục tiêu của việc học là thu thập tri thức mới và xử lý tri thức mới sao cho thích nghi với tình huống mới.

Giống như cách học của con người, người ta muốn xây xựng các chương trình học cho máy sao cho máy có khả năng thu thập tri thức mới và xử lý tri thức mới sao cho thích nghi với tình huống mới.

Giống như cách học của con người, máy có các thể loại học như học giám sát, học củng cố và học không giám sát.

+ Học giám sát : học giám sát là thể loại học với quá trình học có tín hiệu hướng dẫn vào ra chính xác của thầy giáo. Với thể loại học này, dữ liệu học vào ra mong muốn của hệ thống học phải được thiết lập trước. Sau quá trình học, hệ thống sẽ tìm ra một luật thích hợp để thực hiện tốt công việc dự báo ngõ ra được kết hợp với ngõ vào mới của hệ thống.

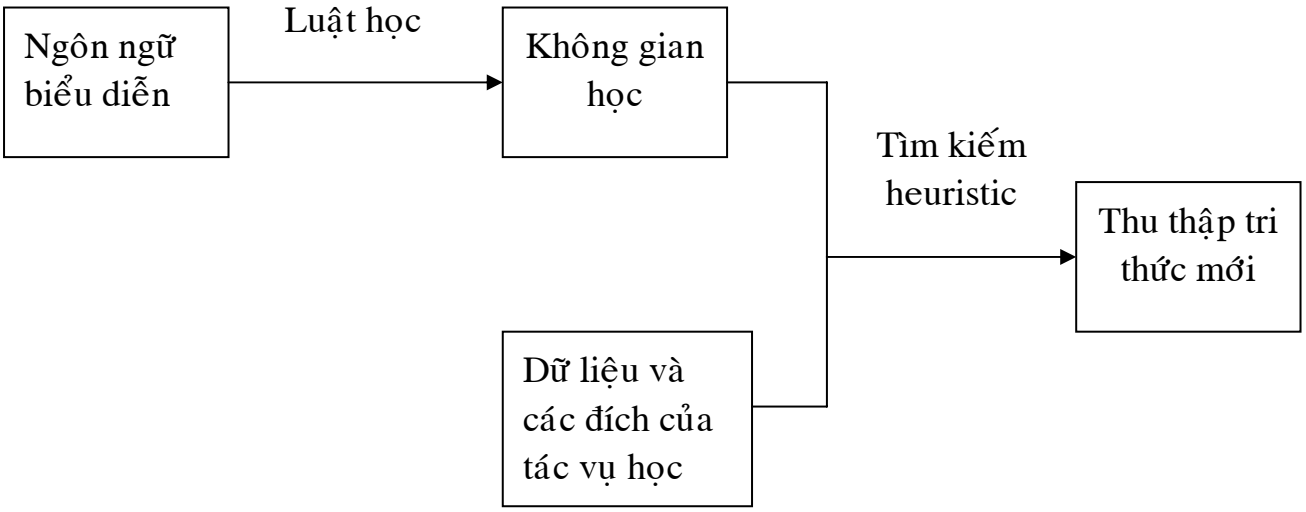
+ Học củng cố : học củng cố cũng là thể loại học giám sát; tuy nhiên tín hiệu hướng dẫn của thầy giáo là tín hiệu củng cố. Với thể loại học này, tín hiệu học của thầy giáo là tín hiệu thưởng tương ứng với tín hiệu đúng hoặc phạt tương ứng với tín sai trên cơ sở tri thức sẵn có của hệ thống cho tập các mẫu dữ liệu học vào mong muốn. Quá trình học, hệ thống sẽ tìm ra một luật thích hợp để củng cố những hành động ra quyết định đúng của hệ thống.

+ Học không giám sát : còn được gọi là thể loại học tự học, với thể loại học này, quá trình học không có sự trợ giúp bất kỳ thông tin hướng hẫn nào của thầy giáo, hệ thống tự khám phá ra một luật thích nghi để thực hiện tốt công việc ngõ ra được kết hợp với ngõ vào mới từ tập các mẫu dữ liệu học ngõ vào mong muốn.

Có ba lĩnh vực học máy đó là học trên cơ sở tri thức, học nhờ mạng neuron nhân tạo và học nhờ giải thuật học di truyền.

7.2) Mô Hình **Học Máy Trên Cơ Sở Tri Thức** :

Việc học máy trên cơ sở tri thức với mô hình tổng quát của quá trình học được mô tả bằng lưu đồ khối như hình



Mô hình học trên cơ sở tri thức gồm các thành phần như dữ liệu học và các đích của tác vụ học, ngôn ngữ biểu diễn tri thức học, luật học, không gian học và tìm kiếm heuristic.

+ **Dữ liệu và các đích của việc học** : công đoạn đầu tiên của việc học là phải xác định được đặc thù của các bài tóan học căn cứ theo đích của người học và dữ liệu học được thiết lập. Ví dụ điển hình là các thuật tóan học quy nạp, dữ liệu học là tập các mẫu ví dụ và đích của việc học là suy diễn một định nghĩa tổng quát để nhận dạng lớp của các đối tượng.

+ **Biểu diễn tri thức học** : công đoạn thứ hai của mô hình học trên cơ sở tri thức là chọn ngôn ngữ biểu diễn thích hợp để mã hóa tri thức học. Đó là ngôn ngữ biểu diễn nhờ logic vị từ và ngôn ngữ biểu diễn nhờ frame đã được khảo sát trước đây.

+ Luật học : công đoạn thứ ba là luật học, cho dữ liệu học, người học phải xây dựng một luật học sao cho thỏa mãn các đích của việc học.

+ Không gian học : ngôn ngữ biểu diễn tri thức học kết hợp với luật học định nghĩa một không gian học, người học phải tìm kiếm trong không gian này để tìm ra một khái niệm mong muốn học.

+ Tìm kiếm heuristic : hầu hết các chương trình học sử dụng thông tin heuristic để giúp quá trình học nhanh và có hiệu quả.

Với mô hình học trên cơ sở tri thức này, hệ thống có thể thu thập được tri thức mới từ những tri thức sẵn có của hệ thống.

1) **Giải thuật học gám sát hướng đặc trưng đến tổng quát và ngược lại** :

Mục tiêu của hai loại giải thuật học này là tìm ra một định nghĩa tổng quát để nhận dạng được tất cả các đối tượng của lớp . Giải thuật sử dụng dữ liệu học gồm hai tập mẫu dữ liệu huấn luyện dương P và âm N. Dữ liệu huấn luyện dương là dữ liệu cung cấp thông tin bổ ích được biết về các đối tượng của lớp muốn học và dữ liệu âm là dữ liệu cung cấp thông tin không bổ ích được biết về các đối tượng của lớp .

Giải thuật học hướng đặc trưng đến tổng quát hóa là quá trình học, hệ thống bắt đầu từ đối tượng với các thành phần đặc trưng nhất, tổng quát hóa các thành phần đặc trưng này sao cho đạt đến một định nghĩa tổng quát mà có thể nhận dạng được tất cả các đối tượng của lớp. Luật học của giải thuật này là tóan tử tổng quát hóa đó là tóan tử thay thế các thành phần hằng số của đối tượng với biến số.

Giải thuật học hướng tổng quát đến đặc trưng là quá trình học, hệ thống bắt đầu từ đối tượng với các thành phần tổng quát hóa nhất, đặc trưng các thành phần này sao cho đạt đến một định nghĩa tổng quát mà có thể nhận dạng được tất cả các đối tượng của lớp. Luật học của giải thuật này là tóan tử đặc trưng hóa đó là tóan tử thay thế các thành phần biến số của đối tượng với hằng số.

Giải thuật học hướng đặc trưng đến tổng quát được mô tả là

Begin

- Cho danh sách S chứa mẫu huấn luyện dương đặc trưng nhất.
- Cho N là tập chứa các mẫu huấn luyện âm.
- Cho mỗi mẫu huấn luyện dương p

  Begin

  - Cho mọi mẫu $s \in S$ không hợp với p, thì thay thế các thành phần đặc trưng của s với biến số sao cho hợp với p.
  - Loại bỏ tất cả các mẫu tổng quát hơn một vài mẫu khác trong S.
  - Loại bỏ tất cả các mẫu trong S mà hợp với mẫu âm n được giám sát trước đó.

  End ;
- Cho mỗi mỗi mẫu âm n

  Begin

  - Loại bỏ tất cả các thành viên của S hợp với n
  - Cộng n vào tập N để giám sát các mẫu quá tổng quát khác trong quá trình học.

  End;

End.

Giải thuật học hướng tổng quát hóa đến đặc trưng hóa được mô tả là

Begin

- Cho danh sách G chứa mẫu với các thành phần tổng quát nhất đó là các biến số mô tả các thành phần của đối tượng.
- Cho P là danh sách chứa các mẫu huấn luyện dương.
- Cho mỗi mẫu huấn luyện âm n

  Begin

  - Cho mỗi mẫu g∈G hợp với n thì thay thế các thành phần tổng quát của g với các thành phần đặc trưng sao cho không hợp với n.
  - Loại bỏ tất cả các mẫu đặc trưng hơn một vài mẫu khác trong G.
  - Loại bỏ tất cả các mẫu không hợp với vài mẫu dương p trong P.

  End;
- Cho mỗi mẫu dương p

  Begin

  - Loại bỏ tất cả các mẫu không hợp với p trong G.
  - Cộng p vào tập P để giám sát các mẫu quá đặc trưng trong quá trình học.

  End;

End.

Ví dụ : Học nhận dạng các đối tượng của lớp quả bóng sử dụng giải thuật học hướng đặc trưng và hướng tổng quát.

Cho miền của các đối tượng với các giá trị là

Kích_thước = {lớn, nhỏ}.

Màu = {đỏ, trắng, xanh}.

Hình = {quả_bóng, viên_gạch, hộp_phấn}.

Dữ liệu học cho các đối tượng này được thiết lập là

+ Tập các mẫu dữ liệu huấn luyện dương P gồm các mẫu là

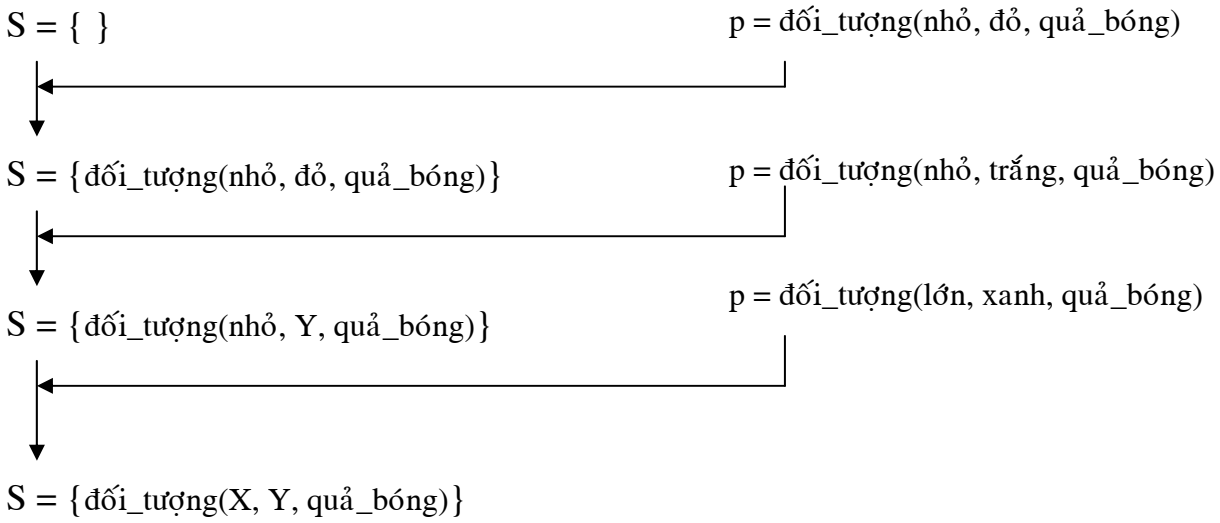
P = {đối_tượng(nhỏ, đỏ, quả_bóng), đối_tượng(lớn, đỏ, quả_bóng), đối_tượng(nhỏ, trắng, quả_bóng), đối_tượng(lớn, trắng, quả_bóng), đối_tượng(nhỏ, xanh, quả_bóng), đối_tượng(lớn, xanh, quả_bóng)}.

+ Tập các mẫu dữ liệu huấn luyện âm N gồm các mẫu là

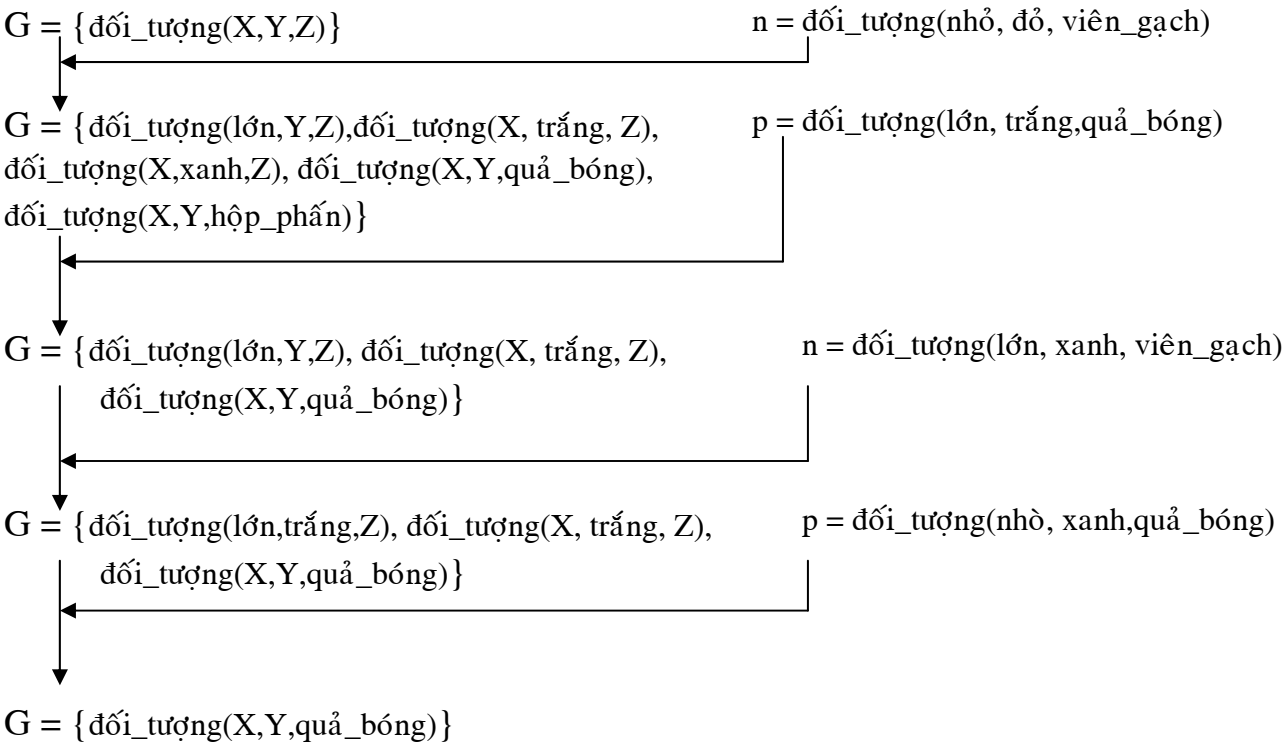
N = { đối_tượng(nhỏ, đỏ, viên_gạch), đối_tượng(lớn, đỏ, viên_gạch), đối_tượng(nhỏ, trắng, viên_gạch), đối_tượng(lớn, trắng, viên_gạch), đối_tượng(nhỏ, xanh, viên_gạch), đối_tượng(lớn, xanh, viên_gạch), đối_tượng(nhỏ, đỏ, hộp_phấn), đối_tượng(lớn, đỏ, hộp_phấn),

đối_tượng(nhỏ, trắng, hộp_phấn), đối_tượng(lớn, trắng, hộp_phấn),
đối_tượng(nhỏ, xanh, hộp_phấn), đối_tượng(lớn, xanh, hộp_phấn)}.

+ Quá trình học để nhận dạng các đối tượng của lớp quả bóng dùng giải thuật học hướng đặc trưng được mô tả như hình



Quá trình học để nhận dạng các đối tượng của lớp quả bóng dùng giải thuật học hướng tổng quát được mô tả như hình



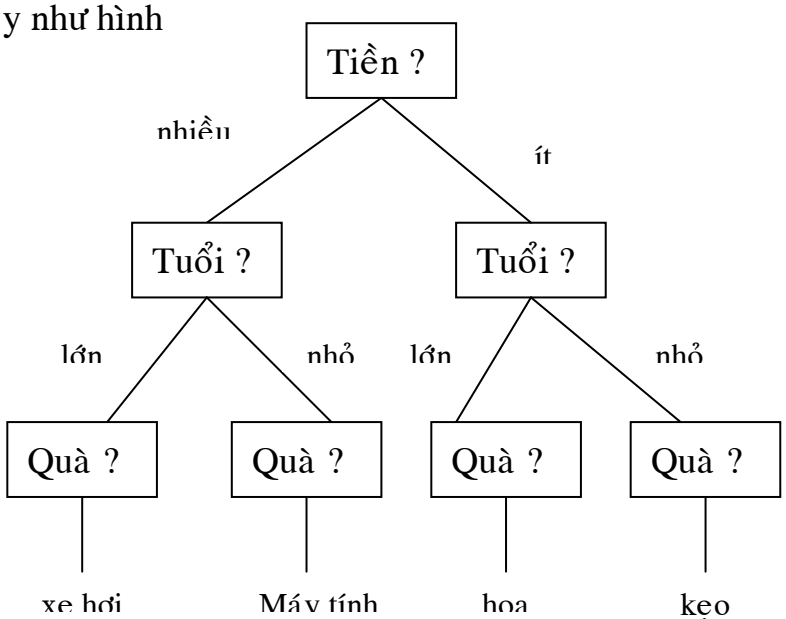
2) **Giải thuật học quy nạp cây quyết định** :

Một loại giải thuật học khác đó là giải thuật học quy nạp cây quyết định. Giải thuật học sử dụng dữ liệu học với các mẫu dữ liệu thu thập được dưới dạng bảng. Bảng chứa các mẫu dữ liệu thu thập được với số cột tương ứng với các thuộc tính mô tả các thành phần của đối tượng và số hàng tương ứng với số mẫu dữ liệu thu thập được. Mục tiêu của giải thuật học là xây dựng một cây quyết định để phân lớp dữ liệu từ dữ liệu thu thập được nhờ thông qua thực nghiệm. Giải thuật chọn một thuộc tính bất kỳ làm gốc của cây để từ đó phân lớp dữ liệu theo các nhánh với các giá trị tương ứng của thuộc tính. Thủ tục này được đệ quy cho mỗi cây con cho đến khi có một cây hòan chỉnh.

Ví dụ : Cho bảng dữ liệu thu thập được từ kinh nghiệm mua quà là

Các Nhân Tố Quyết Định			Kết quả
Stt	Tiền	Tuổi	Quà
1	nhiều	lớn	xe hơi
2	nhiều	nhỏ	máy tính
3	ít	lớn	hoa
4	ít	nhỏ	kẹo

Quá trình học mua quà của giải thuật học quy nạp cây quyết từ bảng dữ liệu được mô tả bằng cây như hình



Cho Example_set là bảng chứa tất cả các mẫu dữ liệu thu thập được và Properties là danh sách chứa các thuộc tính tương ứng trong bảng dữ liệu. Giải thuật học quy nạp cây quyết định được mô tả như sau :

```
Function    induce_tree(Example_set, Properties)
    Begin
        If ( Tất cả các thành viên trong Example_set là cùng lớp )
            Then ( tạo ra nút lá đánh nhãn với lớp đó)
        Elseif ( Properties là danh sách rỗng) Then ( Trả về nút lá có
        đánh nhãn giới từ hoặc của tất cả các lớp trong Example_set )
        Else begin
        -Chọn một thuộc tính P bất kỳ trong danh sách Properties làm gốc
         của cây và loại bỏ thuộc tính này khỏi danh sách.
        - Cho mỗi giá trị V của thuộc tính P
         Begin
           -  Tạo ra một nhánh của cây có đánh nhãn V.
           -  Đặt Partition chứa tất cả các mẫu có giá trị V.
           -  Thủ tục đệ quy cho mỗi cây con bằng cách gọi hàm
           induce_tree(Partition, Properties), nối kết quả vào nhánh V.
           End;
        End;
    End.
```

Ví dụ : Cho dữ liệu thu thập được về việc cho con nợ vay vốn như bảng

stt	Rủi Ro	Uy Tín	Khỏan Nợ	Thế Chấp	Thu Nhập
1	cao	xấu	nhiều	không	thấp
2	cao	chưa biết	nhiều	không	trung bình
3	vừa	chưa biết	ít	không	trung bình
4	cao	chưa biết	ít	không	thấp
5	thấp	chưa biết	ít	không	cao
6	thấp	chưa biết	ít	có	cao
7	cao	xấu	ít	không	thấp
8	vừa	xấu	ít	có	cao
9	thấp	tốt	ít	không	cao
10	thấp	tốt	nhiều	có	cao
11	cao	tốt	nhiều	không	thấp
12	vừa	tốt	nhiều	không	trung bình
13	thấp	tốt	nhiều	không	cao
14	cao	xấu	nhiều	không	trung bình

Hãy học xây dựng cây quyết định đánh giá rủi ro khi cho con nợ vay vốn ?

Giải thuật học quy nạp cây quyết định được sử dụng rất phổ biến trong nhiều lĩnh vực khác nhau như dự báo, đánh giá, nhận dạng và điều khiển bằng kinh nghiệm. Giải thuật giúp người học tìm kiếm nhanh mục đích muốn học từ cây quyết định. Giải thuật cũng giúp người học thiết kế hệ chuyên gia với dữ liệu thu thập được bằng kinh nghiệm. Sau quá trình học, cây quyết định đã được hình thành, thủ tục thiết kế hệ chuyên gia từ cây quyết định này đó là mỗi nhánh của cây có số liệu dẫn đến kết luận đó là một luật suy diễn của hệ chuyên gia. Vế điều kiện của luật là các nhân tố quyết định kết nối nhau từ gốc đến ngọn thông qua các phép tóan giao liên từ và, vế kết luận của luật nhân tố kết quả muốn học.

3) **Học heuristic với giải thuật học quy nạp cây quyết định** :

Cho bảng dữ liệu nhiều hàng và nhiều cột thu thập được từ thực nghiệm. Để giúp giải thuật học nhanh và có hiệu quả, theo lý thuyết thông tin, nhân tố quyết định nào trong bảng dữ liệu giành được thông tin lớn nhất đó là nhân tố quyết định tốt nhất được chọn làm gốc của cây trong quá trình học.

Cách tính thông tin giành được của các nhân tố quyết định trong bảng dữ liệu thu thập được là như sau :

+ Thông tin về nhân tố muốn học M đối với bảng dữ liệu C được tính bằng công thức là

$$I(C) = \sum_{i=1}^{n} - p(m_i) \log_2 (p(m_i))$$

trong đó, m_i là giá trị thứ i của nhân tố muốn học M và $p(m_i)$ là xác suất của mảnh thông tin m_i đối với bảng dữ liệu C đó chính là số mẫu trong bảng dữ liệu C chứa mảnh thông tin m_i chia cho tổng số mẫu trong bảng dữ liệu C.

+ Nếu ta chọn Q làm gốc của cây trong quá trình học thì bảng dữ liệu C sẽ được chia ra nhiều bảng dữ liệu con C_i trong đó mỗi của chúng chứa các mẫu có giá trị tương ứng của thuộc tính Q. Vì thế thông tin về nhân tố quyết Q nếu chọn Q làm gốc của cây được tính bằng công thức là

$$E(Q) = \frac{\sum_{i=1}^{n} |C_i| I(C_i)}{|C|}$$

trong đó, $|C_i|$ là tổng số mẫu chứa trong bảng dữ liệu con C_i , $|C|$ là tổng số mẫu chứa trong bảng dữ liệu C và $I(C_i)$ là thông tin về nhân tố muốn học đối với bảng dữ liệu C_i.

+ Thông tin giành được của nhân tố quyết định Q nếu ta chọn Q làm gốc của cây trong quá trình học được tính bằng công thức là

$$gain(Q) = I(C) - E(Q).$$

Nếu nhân tố quyết định nào có thông tin giành được là lớn nhất đó là nhân tố quyết định quan trọng nhất được chọn làm gốc của cây trong quá trình học. Đơn vị của thông tin là bit.

4) **Khái niệm về học củng cố và học không giám của mô hình học trên cơ sở tri thức** :

+ **Học củng cố** : Học củng cố cũng là dạng học giám sát, tuy nhiên dữ liệu học gồm mảnh nhỏ thông tin đơn giản và tri thức sẵn có của hệ thống. Đích của việc học là sau quá trình học, tìm ra một định nghĩa tổng quát nhất từ mảnh nhỏ thông tin đơn giản này đó là tín hiệu học củng cố của thầy giáo.

Ví dụ : Học tìm ra một luật suy diễn tổng quát để dẫn đến kết rằng X là lớp của các đối tượng quả bóng nhờ sự giải thích thông qua thể loại học củng cố.

- **Dữ liệu học** : gồm mảnh nhỏ thông tin đơn giản và tri thức sẵn có của hệ thống được thiết lập là
  1) **quả_bóng(đt)**.
  2) **vật_đá_được(X)∧vật_hình_cầu(X) → quả_bóng(X).**
  3) **vật_làm_bằng_nhựa(X)∧vật_nhẹ(X) → vật_đá_được(X).**
  4) **vật_có_mặt_lồi(X)∧vật_có_mặt_tròn(X) → vật_hình_cầu(X).**

- **Đích của việc học** : đích của việc học là tìm một luật suy diễn tổng quát nhất với dạng là

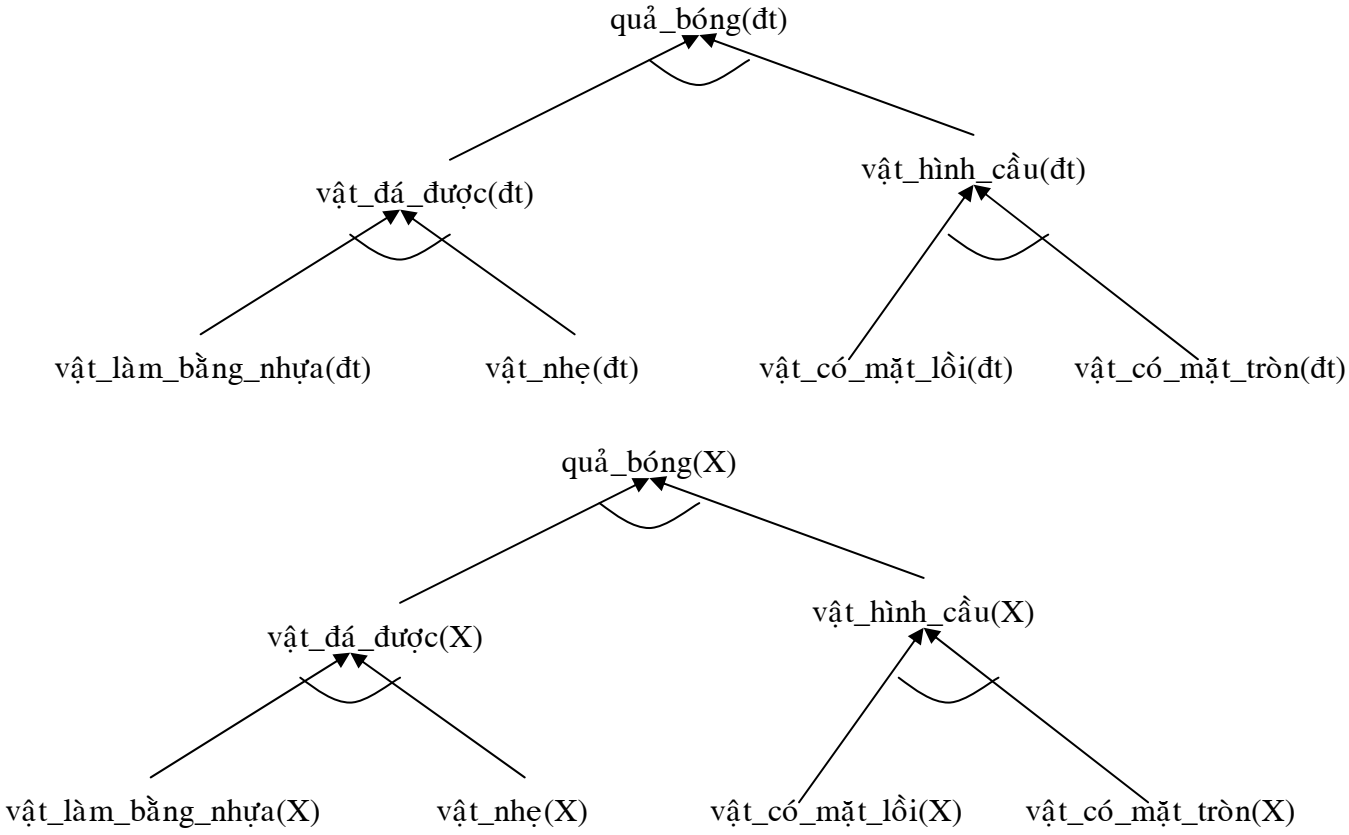
tiên_đề(X) → quả_bóng(X).

Quá trình học, xây dựng các nhân tố quyết định cho tiên_đề(X) để dẫn đến kết luận rằng X là lớp của các đối tượng quả bóng.

Thể loại học củng cố này là dạng học giải thích dựa trên cơ sở tri thức sẵn có của hệ thống và vì thế, hệ thống học phải trải qua hai giai đoạn. Giai đoạn đặc trưng hóa từ tri thức sẵn có của hệ thống đó là giai đoạn giải thích về

các thuộc tính đặc trưng của đối tượng. Giai đọan tổng quát hóa đó là tổng quát hóa các thuộc tính đặc trưng của đối tượng đã được giải thích với biến số X để tìm ra các nhân tố quyết định tổng quát nhất cho tiên_đề(X) dẫn đến kết luận rằng X là lớp của các đối tượng quả bóng.

Quá trình học với thể loại này được mô tả bằng cây như hình



Luật suy diễn tổng quát cho tiên_đề(X) là một biểu thức của các phép tóan giao liên từ và với các thành phần của nó là các nút lá của cây tổng quát. Luật được thiết lập là

$$\text{vật\_làm\_bằng\_nhựa}(X) \wedge \text{vật\_nhẹ}(X) \wedge \text{vật\_có\_mặt\_lồi}(X) \wedge \text{vật\_có\_mặt\_tròn}(X) \rightarrow \text{quả\_bóng}(X).$$

+ **<u>Học không giám sát</u>** : Học không giám sát còn được gọi là thể loại tự học. Hệ thống không được cung cấp bất kỳ một thông tin tín hiệu học nào của thầy giáo. Hệ

thống tự khám phá ra một vài thông tin bổ ích trong quá trình học. Thể loại học này thường sử dụng dữ liệu học không phân lớp và quá trình học tự khám phá để phân lớp dữ liệu.

Ví dụ : Học xếp lớp của các đối tượng quả bóng. Dữ liệu học không phân lớp của các quả bóng được thiết lập là

đt1 = { nhỏ, đỏ, nhựa, quả_bóng}.
đt2 = { nhỏ, xanh, nhựa, quả_bóng}.
đt3 = { lớn, đen, gổ, quả_bóng}.

Đích của việc học là quá trình học xếp lớp của các đối tượng quả bóng dựa trên cơ sở các số đo tương tự.

Cách học là lần lượt tính số đo tương tự của từng cặp dữ liệu, chọn cặp của các đối tượng có số đo tương tự là lớn nhất đưa về một lớp và còn các đối tượng khác có số đo tương tự là nhỏ hơn đưa về một lớp khác. Số đo tương tự của mỗi cặp dữ liệu đó là số thuộc tính giống nhau của hai đối tượng chia cho tổng số các thuộc tính của đối tượng và vì thế số đo tương tự của mỗi cặp dữ liệu cho trên được thiết lập như sau :

- Số đo tương tự của cặp dữ liệu đt1 và đt2 là 3/4.
- Số đo tương tự của cặp dữ liệu đt1 và đt3 là 1/4.
- Số đo tương tự của cặp dữ liệu đt2 và đt3 là 1/4.

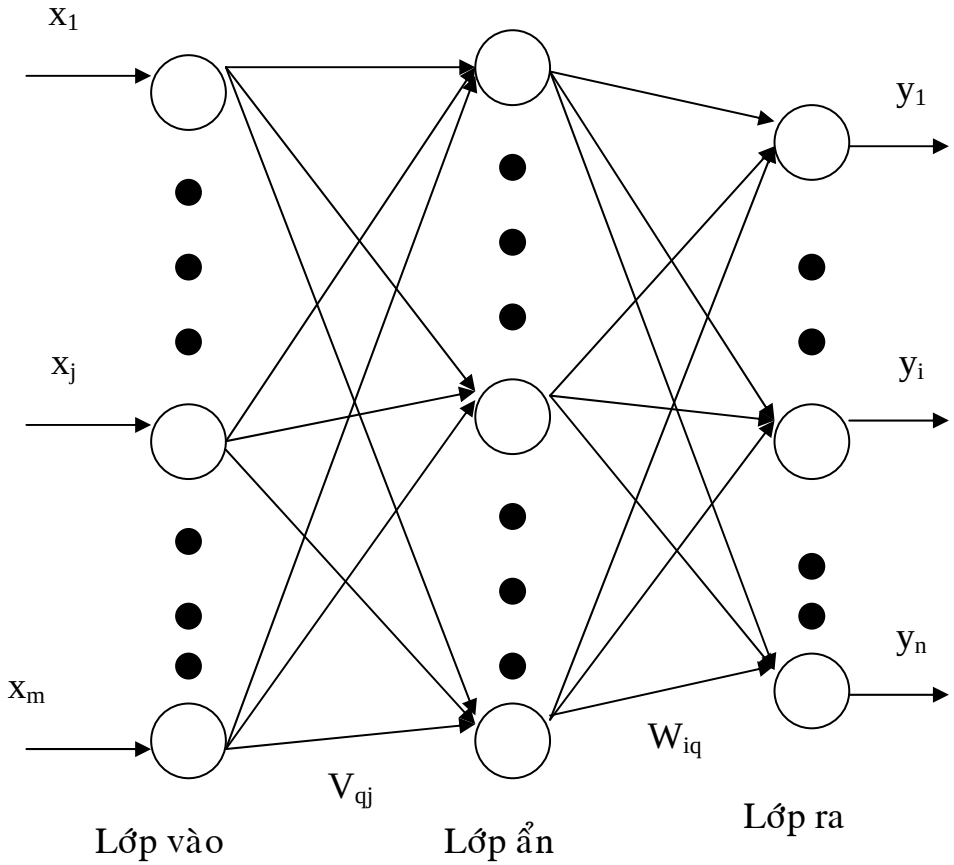
Với các số đo tương tự này, đối tượng đt1 và đt2 là ở cùng lớp và đt3 là ở một lớp khác.

7.3) Mô hình Học Máy Nhờ Mạng Neuron Nhân Tạo :

1) Tổng quan về mạng neuron nhân tạo :

Trái với mô hình học máy trên cơ sở tri thức, mô hình học máy nhờ mạng neuron nhân tạo đó là mô hình học bằng cách mô phỏng lại cấu trúc và nguyên lý làm việc của hệ neuron sinh học con người. Hệ neuron sinh học con người được thừa nhận có khoảng 10^{10} hoặc 10^{12} tế bạo neuron gồm nhiều lớp đó là lớp vào, các lớp ẩn và lớp ra. Lớp vào nối với các phần tử cảm biến như tai, mắt, miệng, mũi, da vân vân, lớp ra nối với các phần tử cơ bắp như chân, tay vân vân và các lớp ẩn chứa các đơn vị xử lý xử các thông tin nhận được từ lớp vào và gởi quyết định đến lớp ra để điều khiển các phần tử cơ bắp như chân và tay vân vân. Mỗi neuron sinh học có nhiều ngõ vào và một ngõ ra và ngõ ra của neuron này được kết nối với ngõ vào của neuron khác. Tín hiệu truyền từ neuron này đến neuron khác là dưới dạng điện áp. Nếu tín hiệu truyền giữa hai neuron là điện áp dương thì hai nuron được kết nối dưới dạng kích thích. Nếu tín hiệu truyền giữa hai neuron là điện áp âm thì

hai neuron được kết nối dưới dạng ức chế. Nếu tín hiệu truyền giữa hai neuron là điện áp zero thì hai neuron là không có sự kết nối. Lượng điện áp truyền giữa các neuron được gọi là cường độ kết nối. Trên cơ sở của hệ neuron sinh học con người như được mô tả, một mạng neuron nhân tạo nhiều lớp được thiết lập như hình



Có ba thành phần cơ bản của các mạng neuron nhân tạo đó là mô hình kết nối, đơn vị xử lý và luật học.

+ **<u>Mô hình kết nối</u>** : Có hai mô hình kết nối đó là kết nối truyền thẳng và kết nối hồi quy. Mô hình kết nối truyền thẳng được gọi là mạng truyền thẳng đó là cấu trúc mạng được kết nối chuyển tiếp tín hiệu từ lớp vào thông qua lớp ẩn và đến lớp ra. Mô hình kết nối hồi quy được gọi là mạng hồi quy đó là cấu trúc mạng được kết nối chuyển tiếp tín hiệu từ lớp vào thông qua lớp ẩn đến lớp ra và đồng thời hồi tiếp tín hiệu về đơn vị xử lý chính nó hoặc các đơn vị xử khác trong lớp hoặc ở lớp khác.

+ **Đơn vị xử lý** : Một mạng neuron nhân tạo có nhiều lớp đó là lớp vào, các lớp ẩn và lớp ra. Lớp vào chứa các neuron được xem như nơi chứa các tín hiệu vào. Các lớp ẩn chứa các neuron được xem như các đơn vị xử lý. Lớp ra chứa các neuron được xem như các đơn vị xử lý ra quyết định. Kết hợp với mỗi đơn vị xử lý có hàm tổng hợp và hàm kích hoạt. Hàm tổng hợp có chức năng tổng hợp tất cả các thông tin từ các ngõ vào của đơn vị và hàm kích hoạt có chức năng tạo tín hiệu ra của đơn vị khi nhận được tín hiệu vào từ hàm tổng hợp.

- Hàm tổng hợp dạng tuyến tính của mỗi đơn vị xử lý thứ i được thiết lập là

$$f_i = \sum_{j=1}^{m} W_{ij} x_j - \theta_i$$

trong đó, W_{ij} là trọng số kết nối giữa đơn vị j và đơn vị i, x_j là ngõ ra của đơn vị j đó chính là ngõ vào của đơn vị i và θ_i là đơn vị ngưỡng của đơn vị xử lý i.

- Hàm kích hoạt tạo tín hiệu ra của đơn vị xử lý thứ i được thiết lập một trong các dạng là

* Hàm bậc thang đơn vị : $$a(f_i) = \begin{cases} 1 & if \quad f_i \geq 0 \\ 0 & if \quad f_i \leq 0 \end{cases}$$

* Hàm unipolar sigmoid : $$a(f_i) = \frac{1}{1 + e^{-\lambda f_i}}$$

* Hàm hyperbolic tangent : $$a(f_i) = \frac{e^{f_i} - e^{-f_i}}{e^{f_i} + e^{-f_i}}$$

+ **Luật học** : Có hai cách học trong các mạng neuron nhân tạo đó là học cấu trúc và học thông số. Học cấu trúc là quá trình học thay đổi cấu trúc bên trong của mạng. Học thông số là quá trình học cập nhật các trọng số kết nối giữa các đơn vị xử lý trong mạng sao cho xấp xỉ với bộ trọng số mong muốn để có được ánh xạ vào ra như mong muốn.

Cho W_{ij} là trọng số kết nối giữa đơn vị thứ j và đơn vị thứ i, luật học cập nhật trọng trọng số tại thời điểm t+1 được thiết lập là

$$W_{ij}(t+1) = W_{ij}(t) + \eta \Delta W_{ij}(t)$$

Trong đó, η là hằng số dương đó được gọi là tốc độ học và $\Delta W_{ij}(t)$ là lượng gia tăng trọng số tại thời điểm t.

Có ba thể loại học trong các mạng neuron nhân tạo đó là học giám sát, học củng cố và học không giám sát.

- **Học giám sát** : cho tập dữ liệu vào ra mong muốn, quá trình học cập nhật các trọng số kết nối giữa các phần tử xử lý trong mạng sao cho ngõ ra that sự của mạng xấp xỉ với ngõ ra mong muốn của mạng.

- **Học củng cố** : cũng là thể loại học giám sát, tuy nhiên, tín hiệu ra mong muốn của mạng là tín hiệu củng cố đó là tín hiệu thưởng và phạt. Quá trình học, cập nhật các trọng số kết nối giữa các đơn vị xử lý sao cho ngõ ra thật sự của mạng xấp xỉ với ngõ ra mong muốn thưởng với độ tin cậy càng cao càng tốt.

- **Học không giám sát** : là thể loại học không có dữ liệu ra mong muốn, quá trình học với tập dữ liệu vào mong muốn, mạng tự cập nhật các trọng số kết nối dựa trên cơ sở tập dữ liệu vào mong muốn sao cho ngõ ra thực sự của mạng thích nghi với ngõ vào mong muốn.

2) Mạng truyền thẳng và giải thuật học lan truyền ngược :

Cho mạng truyền thẳng ba lớp như được mô tả trên, và với hàm chi phí đo tín hiệu sai số giữa ngõ ra thực sự y_i (k) của mạng và ngõ ra mong muốn d_i (k) của mạng cho bởi mỗi mẫu tín hiệu vào mong muốn X(k) là

$$E(k) = \frac{1}{2}\sum_{i=1}^{n}(d_i(k) - y_i(k)).$$

Giải thuật học lan truyền ngược cập nhật trọng số kết nối trong mạng được mô tả gồm các bước sau :

Bước 0 : Nhập tập mẫu huấn luyện vào ra mong {X(k), d(k), cho k = 1, . . . ,p}, trong đó X(k) là véctơ mẫu vào và d(k) là véctơ mẫu ra. Thiết lập tốc độ học η, sai số cho phép hội tụ E_{max} , trọng số khởi tạo, E = 0 và k = 1.

Bước 1 : Truyền tín hiệu chuyển tiếp từ lớp vào, qua lớp ẩn và đến lớp ra.

Cho q = 1 đến l

$$net_q(k) = \sum_{j=1}^{m} V_{qj}(k)x_j(k)$$

$$z_q(k) = a(net_q(k)).$$

Cho i = 1 đến n

$$net_i(k) = \sum_{q=1}^{l} W_{iq}(k)z_q(k)$$

$$y_i(k) = a(net_i(k)).$$

Bước 2 : Tính sai số chuan 2 giữa ngõ ra mong muốn và ngõ ra thực sự của mạng.

$$E(k) = \frac{1}{2}\sum_{i=1}^{n}(d_i(k) - y_i(k))$$

Bước 3 : Lan truyền ngược cập nhật trọng số kết nối giữa các lớp.

Cho i = 1 đến n

$$\delta_i(k) = (d_i(k) - y_i(k)) \times a'(net_i(k))$$

Cho i = 1 đến n

Cho q = 1 đến l

$$W_{iq}(k+1) = W_{iq}(k) + \eta \times \delta_i(k) \times z_q(k).$$

Cho q = 1 đến l

$$\delta_q(k) = a'(net_q(k)) \times \sum_{i=1}^{n} W_{qi}(k)\delta_i(k).$$

Cho q = 1 đến l

Cho j = 1 đến m

$$V_{qj}(k+1) = V_{qj}(k) + \eta \times \delta_q(k) \times x_j(k).$$

Bước 4 : Kiểm tra nếu k < p thì tăng k = k + 1 và quay về bước 1; mặt khác đến bước 5.

Bước 5 : Kiểm tra nếu E < E_{max} thì dừng thủ tục huấn luyện; mặt khác thiết lập E = 0 và k = 1 và quay về bước 1 thực hiện một thế hệ huấn luyện khác.