

Các Hệ cơ sở tri thức

KBS: Knowledge Based Systems

Nguyễn Đình Thuận
Khoa Hệ thống Thông tin
ĐH Công nghệ thông tin

TP HCM 1-2011

Hệ cơ sở tri thức

Chương 1: Tổng quan về Hệ cơ sở tri thức

Chương 2: Biểu diễn và suy luận tri thức

Chương 3: Hệ MYCIN

Chương 4: Hệ học

Chương 5: Hệ thống mờ cho các biến liên tục

Chương 6: Công cụ tạo lập Hệ CSTT

Tài liệu tham khảo

- [1] **McGraw-Hill/Irwin.** Management information system The McGraw-Hill Companies, Inc. 2008 .
- [2] Robert I. Levine. Knowledge based systems. Wissenschafts Verlag, 1991
- [3] Hoàng Kiếm. Giáo trình Các hệ cơ sở tri thức. ĐHQG TP Hồ Chí Minh. 2007

Chương 1: Tổng quan về Hệ cơ sở tri thức

1.1 Khái niệm về Hệ Cơ sở tri thức

- Hệ cơ sở tri thức là chương trình máy tính được thiết kế để mô hình hoá khả năng giải quyết vấn đề của chuyên gia con người.
- Hệ CSTT là hệ thống dựa trên tri thức, cho phép mô hình hoá các tri thức của chuyên gia, dùng tri thức này để giải quyết vấn đề phức tạp thuộc cùng lĩnh vực.
- Hai yếu tố quan trọng trong Hệ CSTT là: tri thức chuyên gia và lập luận, tương ứng với hệ thống có 2 khối chính là Cơ sở tri thức và động cơ suy diễn.

1.1 Khái niệm về Hệ CSTT (Tiếp)

- Hệ Chuyên gia là một loại cơ sở tri thức được thiết kế cho một lĩnh vực ứng dụng cụ thể.

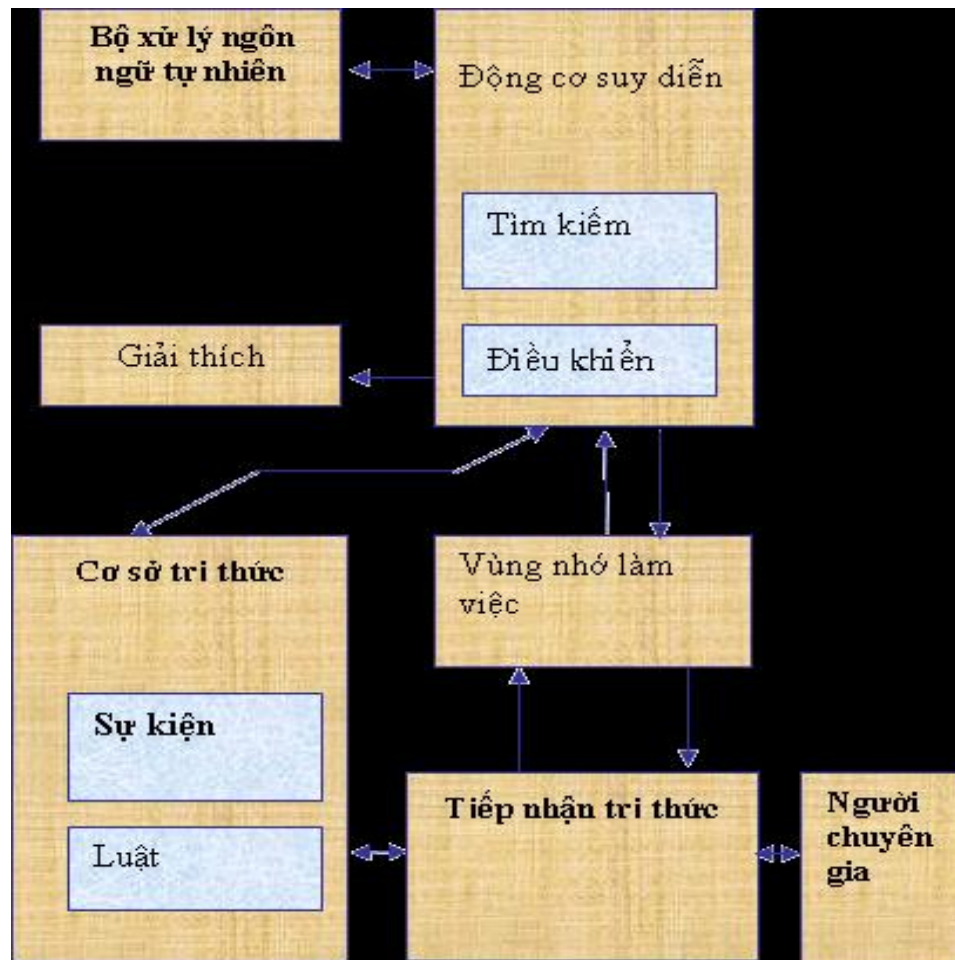
Ví dụ: Hệ Chuyên gia về chẩn đoán bệnh trong Y khoa, Hệ Chuyên gia chẩn đoán hồng học của đường dây điện thoại,...

- Hệ Chuyên gia làm việc như một chuyên gia thực thụ và cung cấp các ý kiến dựa trên kinh nghiệm của chuyên gia con người đã được đưa vào Hệ Chuyên gia.

1.1 Khái niệm về Hệ CSTT (Tiếp)

- Cơ sở tri thức: Chứa các tri thức chuyên sâu về lĩnh vực như chuyên gia. Cơ sở tri thức bao gồm: các sự kiện, các luật, các khái niệm và các quan hệ.
- Động cơ suy diễn: bộ xử lý tri thức theo mô hình hoá theo cách lập luận của chuyên gia. Động cơ hoạt động trên thông tin về vấn đề đang xét, so sánh với tri thức lưu trong cơ sở tri thức rồi rút ra kết luận.
- Kỹ sư tri thức (Knowledge Engineer): người thiết kế, xây dựng và thử nghiệm Hệ Chuyên gia.

1.2 Cấu trúc của Hệ CSTT



1.2 Cấu trúc của Hệ CSTT(tiếp)

1. **Giao diện người máy (User Interface):** Thực hiện giao tiếp giữa Hệ Chuyên gia và User. Nhận các thông tin từ User (các câu hỏi, các yêu cầu về lĩnh vực) và đưa ra các câu trả lời, các lời khuyên, các giải thích về lĩnh vực đó. Giao diện người máy bao gồm: Menu, bộ xử lý ngôn ngữ tự nhiên và các hệ thống tương tác khác.
2. **Bộ giải thích (Explanation system):** Giải thích các hoạt động khi có yêu cầu của User.
3. **Động cơ suy diễn (Inference Engine):** Quá trình trong Hệ Chuyên gia cho phép khớp các sự kiện trong vùng nhớ làm việc với các tri thức về lĩnh vực trong cơ sở tri thức, để rút ra các kết luận về vấn đề đang giải quyết.

1.2 Cấu trúc của Hệ CSTT(tiếp)

4. **Bộ tiếp nhận tri thức (Knowledge editor):** Làm nhiệm vụ thu nhận tri thức từ chuyên gia con người (human expert), từ kỹ sư tri thức và User thông qua các yêu cầu và lưu trữ vào cơ sở tri thức
5. **Cơ sở tri thức:** Lưu trữ, biểu diễn các tri thức mà hệ đảm nhận, làm cơ sở cho các hoạt động của hệ. Cơ sở tri thức bao gồm các sự kiện (facts) và các luật (rules).
6. **Vùng nhớ làm việc (working memory):** Một phần của Hệ Chuyên gia chứa các sự kiện của vấn đề đang xét.

1.3 Các đặc tính của Hệ CSTT

1. Tách tri thức ra khỏi điều khiển: cơ sở tri thức và động cơ suy diễn là các khối tách rời: một khối chứa tri thức còn khối kia là cơ chế giải quyết vấn đề.
2. Tri thức của chuyên gia: các tri thức được thu nạp từ nhiều chuyên gia hiểu sâu về lĩnh vực đang giải quyết.
Expert: “Người có kỹ năng giải quyết đúng vấn đề và giải quyết hiệu quả. Họ có khả năng lập luận hơn những người khác trong lĩnh vực đó.”
3. Tập trung nguồn chuyên gia: Mỗi chuyên gia chỉ giải quyết các vấn đề trong lĩnh vực của mình. Nên chia nhỏ bài toán chẩn đoán theo các hệ thống nhỏ hơn.

1.3 Các đặc tính của Hệ CSTT (tiếp)

4. Lập luận trên các ký hiệu: Chuyên gia dùng các ký hiệu để thể hiện tri thức, thay vì thực hiện việc xử lý số.
5. Lập luận may rủi: Kinh nghiệm giúp các chuyên gia nhanh chóng tìm đến các giải pháp.
6. Lập luận với thông tin không đầy đủ: chẳng hạn, bác sĩ khám bệnh nhân vào cấp cứu trong hoàn cảnh không hỏi được nhiều thông tin.
7. Chứa khả năng suy diễn, mức độ sâu sắc cao, có khả năng tự kiểm tra, tự học trong quá trình suy luận.

1.3 Các đặc tính của Hệ CSTT(tiếp)

Đối tượng so sánh	Chương trình bình thường	Các hệ CSTT
Xử lý	Số	Ký hiệu
Sử dụng	Thuật toán	Kinh nghiệm
Tổ chức thông tin	Thông tin và điều khiển được tích hợp	Tri thức tách rời điều khiển
Khả năng thay đổi	Khó thay đổi	Dễ thay đổi
Tính chắc chắn	Thông tin chính xác	Thông tin không chính xác
Giao diện	Giao diện câu lệnh	Hội thoại tự nhiên có giải thích
Loại kết quả	Kết quả cuối cùng	Khuyến cáo kèm theo giải thích
Tính tối ưu	Nghiem tối ưu	Lời giải chấp nhận được

Bảng 1.1: So sánh Hệ chuyên gia và chương trình bình thường

1.4 Hệ hỗ trợ ra quyết định

DSS (Decision Support System)

- Chức năng: Hỗ trợ ra quyết định
- Hoạt động theo cách tương tác với người sử dụng

Các tính chất của DSS:

- Hướng đến các quyết định của người quản lý
- Uyển chuyển với hoàn cảnh
- Trả lời câu hỏi trong tình huống
- Do người sử dụng khởi động và kiểm soát

1.5 Hệ học

- Trong nhiều tình huống, sẽ không có sẵn tri thức như:
 - Kỹ sư tri thức cần thu nhận tri thức từ chuyên gia lĩnh vực.
 - Cần biết các luật mô tả lĩnh vực cụ thể.
 - Bài toán không được biểu diễn tường minh theo luật, sự kiện hay các quan hệ.
- Có hai tiếp cận cho hệ thống học:
 - Học từ ký hiệu: bao gồm việc hình thức hóa, sửa chữa các luật tường minh, sự kiện và các quan hệ.
 - Học từ dữ liệu số: được áp dụng cho những hệ thống được mô hình dưới dạng số liên quan đến các kỹ thuật nhằm tối ưu các tham số. Học theo dạng số bao gồm mạng Neural nhân tạo, thuật giải di truyền, bài toán tối ưu truyền thống. Các kỹ thuật học theo số không tạo ra CSTT tường minh.

1.6 Hệ điều khiển mờ

- Mờ hóa: Chuyển đổi giá trị rõ đầu vào thành các vector mờ
- Xác định các luật hợp thành và thuật toán xác định giá trị mờ
- Giải mờ: Phương pháp điểm trọng tâm

1.7 Ứng dụng của Hệ Cơ sở tri thức

1. Diễn giải (Interpretation): Mô tả tình huống các dữ liệu thu thập được
2. Dự báo (Prediction): đưa ra các tri thức về dự báo một tình huống: dự báo giá cả, ...
3. Thiết kế (Design): Lựa chọn cấu hình phù hợp, ví dụ: sắp xếp công việc.
4. Chẩn đoán (Diagnosis): Dựa vào các dữ liệu quan sát được, xác định các lỗi hỏng hóc.

1.7 Ứng dụng của Hệ Cơ sở tri thức (tiếp)

5. Vạch kế hoạch (Planing): tạo lập các phương án hành động.
6. Dẫn dắt (Monotoring): So sánh dữ liệu và các kết quả hoạt động.
7. Gỡ rối (Debugging): Mô tả các phương pháp khắc phục của hệ thống.
8. Giảng dạy (Instruction): Sửa chữa các lỗi của người học trong quá trình học tập.
9. Điều khiển (Control): dẫn dắt dáng điệu tổng thể của hệ thống.

Thông tin về môn học

Đánh giá

<i>Phương pháp đánh giá</i>	<i>Trọng số[%]</i>
Chuyên cần, bài tập trên lớp	15%
Cài đặt	15%
Tiểu luận, báo cáo	20%
Thi cuối học kỳ	50%

Chương 2: Biểu diễn và suy luận tri thức

2.1. Mở đầu

- *tri thức, lĩnh vực và biểu diễn tri thức.*

2.2. Các loại tri thức: được chia thành 5 loại

1. Tri thức thủ tục: mô tả cách thức giải quyết một vấn đề. Loại tri thức này đưa ra giải pháp để thực hiện một công việc nào đó. Các dạng tri thức thủ tục tiêu biểu thường là các luật, chiến lược, lịch trình và thủ tục.
2. Tri thức khai báo: cho biết một vấn đề được thấy như thế nào. Loại tri thức này bao gồm các phát biểu đơn giản, dưới dạng các khẳng định logic đúng hoặc sai. Tri thức khai báo cũng có thể là một danh sách các khẳng định nhằm mô tả đầy đủ hơn về đối tượng hay một khái niệm nào đó.

2.2. Các loại tri thức (tiếp)

3. **Siêu tri thức:** mô tả *tri thức về tri thức*. Loại tri thức này giúp lựa chọn tri thức thích hợp nhất trong số các tri thức khi giải quyết một vấn đề. Các chuyên gia sử dụng tri thức này để điều chỉnh hiệu quả giải quyết vấn đề bằng cách hướng các lập luận về miền tri thức có khả năng hơn cả.
4. **Tri thức heuristic:** mô tả các "*kinh nghiệm*" để dẫn dắt tiến trình lập luận. Tri thức heuristic là *tri thức* không đảm bảo hoàn toàn 100% chính xác về kết quả giải quyết vấn đề. Các chuyên gia thường dùng các tri thức khoa học như sự kiện, luật, ... sau đó chuyển chúng thành các tri thức heuristic để thuận tiện hơn trong việc giải quyết một số bài toán.
5. **Tri thức có cấu trúc:** mô tả tri thức theo cấu trúc. Loại tri thức này mô tả mô hình tổng quan hệ thống theo quan điểm của chuyên gia, bao gồm khái niệm, khái niệm con, và các đối tượng; diễn tả chức năng và mối liên hệ giữa các tri thức dựa theo cấu trúc xác định.

Ví dụ: Hãy phân loại các tri thức sau

1. Nha Trang là thành phố đẹp.
2. Bạn Lan thích đọc sách.
3. Modus Ponens.
4. Modus Tollens.
5. Thuật toán tìm kiếm BFS
6. Thuật toán tìm kiếm DFS
7. Thuật toán A^{KT}
8. Thuật giải Greedy

Ví dụ: Hãy phân loại các tri thức sau(tiếp)

9. Một số cách chiếu tướng trong việc chơi cờ tướng.
10. Hệ thống các khái niệm trong hình học.
11. Cách tập viết chữ đẹp.
12. Tóm tắt quyển sách về Hệ chuyên gia.
13. Chọn loại cổ phiếu để mua cổ phiếu.
14. Chọn các thông tin về các loại cổ phiếu trên thị trường.

2.3. CÁC KỸ THUẬT BIỂU DIỄN TRI THỨC

2.3.1 Bộ ba Đối tượng-Thuộc tính-Giá trị

2.3.2 Các luật dẫn

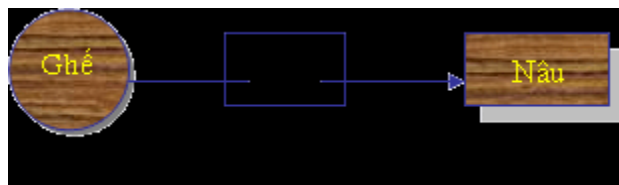
2.3.3 Mạng ngữ nghĩa

2.3.4 Frames

2.3.5 Logic

2.3.1 Bộ ba Đối tượng-Thuộc tính-Giá trị

- Một sự kiện có thể được dùng để xác nhận giá trị của một thuộc tính xác định của một vài đối tượng. Ví dụ, mệnh đề "quả bóng màu đỏ" xác nhận "đỏ" là giá trị thuộc tính "màu" của đối tượng "quả bóng". Kiểu sự kiện này được gọi là bộ ba Đối tượng-Thuộc tính-Giá trị (O-A-V – Object-Attribute-Value).



Hình 2.1. Biểu diễn tri thức theo bộ ba O-A-V

2.3.1 Bộ ba Đối tượng-Thuộc tính-Giá trị (tiếp)

- Trong các sự kiện O-A-V, một đối tượng có thể có nhiều thuộc tính với các kiểu giá trị khác nhau. Hơn nữa một thuộc tính cũng có thể có một hay nhiều giá trị. Chúng được gọi là các sự kiện *đơn trị* (single-valued) hoặc *đa trị* (multi-valued). Điều này cho phép các hệ tri thức linh động trong việc biểu diễn các tri thức cần thiết.
- Các sự kiện không phải lúc nào cũng bảo đảm là đúng hay sai với độ chắc chắn hoàn toàn. Ví thế, khi xem xét các sự kiện, người ta còn sử dụng thêm một khái niệm là *độ tin cậy*. Phương pháp truyền thống để quản lý thông tin không chắc chắn là sử dụng nhân tố chắc chắn CF (certainly factor). Khái niệm này bắt đầu từ hệ thống MYCIN (khoảng năm 1975), dùng để trả lời cho các thông tin suy luận. Khi đó, trong sự kiện O-A-V sẽ có thêm một giá trị xác định độ tin cậy của nó là CF.

2.3.2 Các luật dẫn

- Luật là cấu trúc tri thức dùng để liên kết thông tin đã biết với các thông tin khác giúp đưa ra các suy luận, kết luận từ những thông tin đã biết.
- Trong hệ thống dựa trên các luật, người ta thu thập các tri thức lĩnh vực trong một tập và lưu chúng trong cơ sở tri thức của hệ thống. Hệ thống dùng các luật này cùng với các thông tin trong bộ nhớ để giải bài toán. Việc xử lý các luật trong hệ thống dựa trên các luật được quản lý bằng một module gọi là *bộ suy diễn*.

2.3.2 Các luật dẫn (tiếp)

Các dạng luật cơ bản: 7 dạng

1. Quan hệ:

IF Bình điện hỏng

THEN Xe sẽ không khởi động được

2. Lời khuyên:

IF Xe không khởi động được

THEN Đi bộ

3. Hướng dẫn

IF Xe không khởi động được AND Hệ thống nhiên liệu tốt

THEN Kiểm tra hệ thống điện

2.3.2 Các luật dẫn (tiếp)

4. Chiến lược

IF Xe không khởi động được

THEN Đầu tiên hãy kiểm tra hệ thống nhiên liệu, sau đó kiểm tra hệ thống điện

5. Diễn giải

IF Xe nổ AND tiếng giong

THEN Động cơ hoạt động bình thường

6. Chẩn đoán

IF Sốt cao AND hay ho AND Họng đỏ

THEN Viêm họng

7. Thiết kế

IF Là nữ AND Da sáng

THEN Nên chọn Xe Spacy AND Chọn màu sáng

2.3.2 Các luật dẫn(tiếp)

Mở rộng cho các luật

Trong một số áp dụng cần thực hiện cùng một phép toán trên một tập hay các đối tượng giống nhau. Lúc đó cần các *luật có biến*.

Ví dụ: IF X là nhân viên AND Tuổi của X > 65
THEN X có thể nghỉ hưu

- Khi mệnh đề phát biểu về sự kiện, hay bản thân sự kiện có thể không chắc chắn, người ta dùng hệ số chắc chắn CF. Luật thiết lập quan hệ không chính xác giữa các sự kiện giả thiết và kết luận được gọi là *luật không chắc chắn*.

Ví dụ: IF Lạm phát CAO THEN Hài như chắc chắn lãi suất sẽ CAO
Luật này được viết lại với giá trị CF có thể như sau:

IF Lạm phát cao THEN Lãi suất cao, CF = 0.8

- **Dạng luật tiếp theo là *siêu luật*:**

Một luật với chức năng mô tả cách thức dùng các luật khác. Siêu luật sẽ đưa ra chiến lược sử dụng các luật theo lĩnh vực chuyên dụng, thay vì đưa ra thông tin mới.

Ví dụ: IF Xe không khởi động AND Hệ thống điện làm việc bình thường
THEN Có thể sử dụng các luật liên quan đến hệ thống điện

Ví dụ: Giải quyết vấn đề bằng cách biểu diễn tri thức

Bài toán: Cho hai bình rỗng X và Y có thể tích lần lượt là V_X và V_Y , hãy dùng hai bình này để đong ra z lít nước (với $z \leq \min(V_X, V_Y)$).

- Xét trường hợp cụ thể của bài toán, chẳng hạn: $V_X=5$ và $V_Y=7$ và $z = 4$, có thể đưa ra quy trình sau:
 - Múc đầy bình 7
 - Trút hết qua bình 5 cho đến khi 5 đầy.
 - Đổ hết nước trong bình 5
 - Đổ hết nước còn lại từ bình 7 sang bình 5
 - Múc đầy bình 7
 - Trút hết qua bình 5 cho đến khi bình 5 đầy.
 - Phần còn lại chính là số nước cần đong.

Ví dụ: Giải quyết vấn đề bằng cách biểu diễn tri thức (tt)

- Lời giải trên được phát biểu tổng quát hơn:
 - Không mất tính tổng quát, giả sử rằng $V_X < V_Y$.
 - Gọi lượng nước chứa trong bình X là x ($0 \leq x \leq V_X$)
 - Gọi lượng nước chứa trong bình Y là y ($0 \leq y \leq V_Y$)
 - Điều kiện kết thúc của bài toán sẽ là: $x=z$ hoặc $y=z$
 - Điều kiện đầu của bài toán là: $x=0$ và $y=0$
- Quá trình giải được thực hiện bằng cách xét lần lượt các luật sau, luật nào thỏa mãn thì sẽ được áp dụng.
 - (L1) Nếu bình X đầy thì đổ hết nước trong bình X đi.
 - (L2) Nếu bình Y rỗng thì đổ đầy nước vào bình Y.
 - (L3) Nếu bình X không đầy và bình Y không rỗng thì hãy trút nước từ bình Y sang bình X (cho đến khi bình X đầy hoặc bình Y hết nước).

Ví dụ: Giải quyết vấn đề bằng cách biểu diễn tri thức (tt)

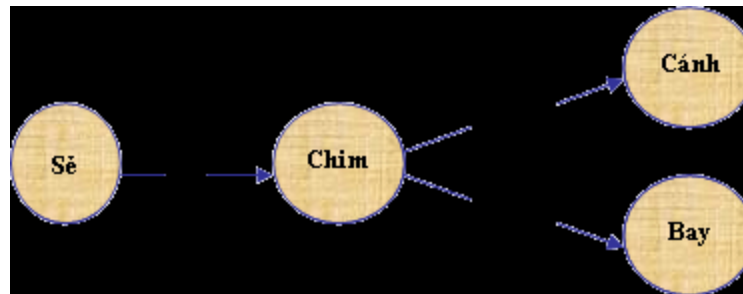
Lời giải trên được minh họa như sau:

```
...
x:= 0; y := 0;
WHILE ((x <> z) AND (y<>z)) DO BEGIN
  IF (x = Vx) THEN x := 0;
  IF (y = 0) THEN (y:= Vy);
  IF (y > 0) THEN BEGIN
    k:= min(Vx - x, y);
    x := x + k;
    y := y - k;
  END;
END;
```

- Cách giải quyết vấn đề theo kiểu này *không đưa ra một trình tự giải quyết vấn đề cụ thể* mà chỉ đưa ra các quy tắc dưới dạng các luật
- Ghi chú: Người ta đã chứng minh được rằng, bài toán đong nước chỉ có lời giải khi số nước cần đong là một bội số của ước số chung lớn nhất của thể tích hai bình.

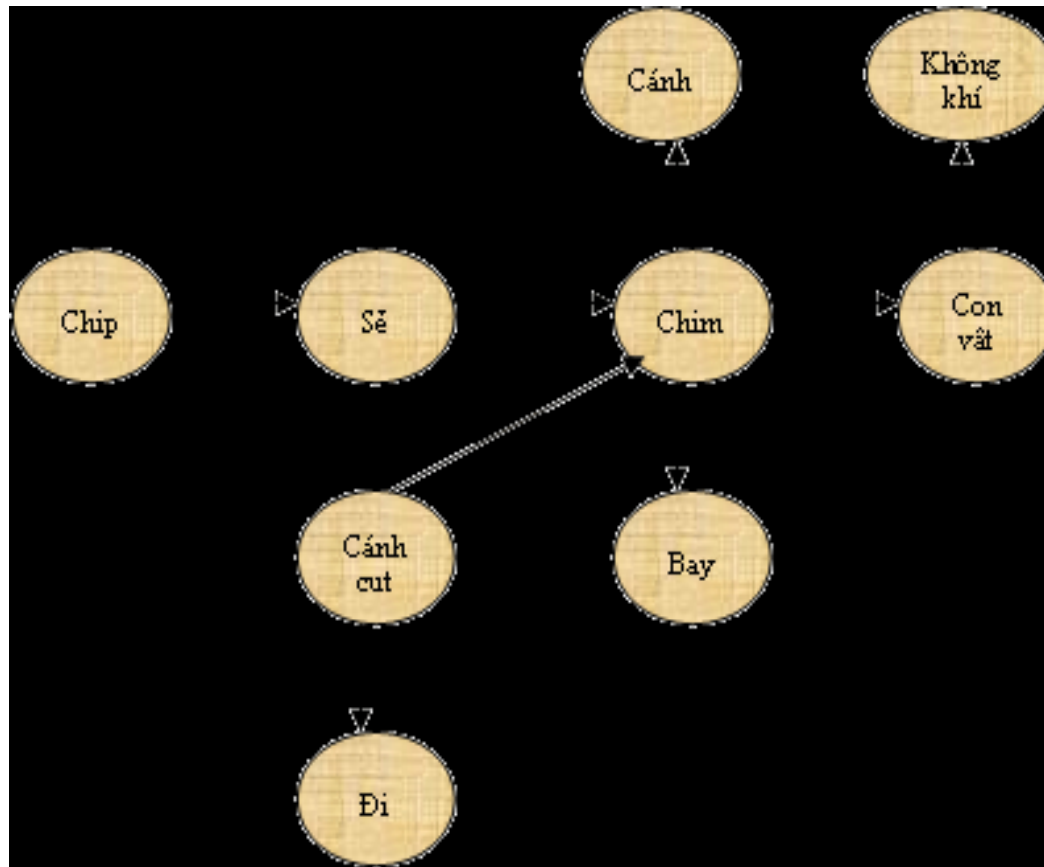
2.3.3 Mạng ngữ nghĩa

Mạng ngữ nghĩa là một phương pháp biểu diễn tri thức dùng đồ thị trong đó nút biểu diễn đối tượng và cung biểu diễn quan hệ giữa các đối tượng.



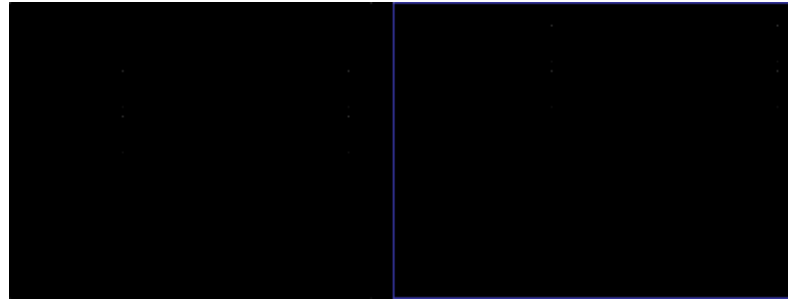
Hình 2.3. "Sẻ là Chim" thể hiện trên mạng ngữ nghĩa

2.3.3 Mạng ngữ nghĩa (tiếp)

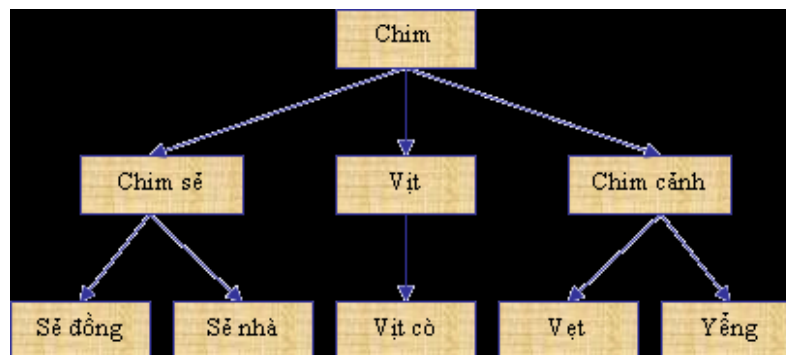


Hình 2.4. Phát triển mạng ngữ nghĩa

2.3.4 Frame



Hình 2.6. Cấu trúc frame



Hình 2.7. Nhiều mức của frame mô tả quan hệ phức tạp hơn

Ví dụ: Giải bài toán tam giác tổng quát

- Có 22 yếu tố của tam giác. Như vậy có $C_{22}^3 - 1$ cách để xây dựng hay xác định một tam giác.
- Theo thống kê, có khoảng 200 công thức liên quan đến cạnh và góc 1 tam giác.
- Để giải bài toán này bằng công cụ mạng ngữ nghĩa, sử dụng khoảng 200 đỉnh để chứa công thức và khoảng 22 đỉnh để chứa các yếu tố của tam giác. Mạng ngữ nghĩa cho bài toán này có cấu trúc như sau :
 - Đỉnh của đồ thị bao gồm hai loại :
 - Đỉnh chứa công thức (ký hiệu bằng hình chữ nhật)
 - Đỉnh chứa yếu tố của tam giác (ký hiệu bằng hình tròn)
 - Cung : chỉ nối từ đỉnh hình tròn đến đỉnh hình chữ nhật cho biết yếu tố tam giác xuất hiện trong công thức nào
- * Lưu ý : trong một công thức liên hệ giữa n yếu tố của tam giác, ta giả định rằng nếu đã biết giá trị của n-1 yếu tố thì sẽ tính được giá trị của yếu tố còn lại. Chẳng hạn như trong công thức tổng 3 góc của tam giác bằng 180° thì khi biết được hai góc, ta sẽ tính được góc còn lại.

Ví dụ: Giải bt tam giác tổng quát (tt)

- **B1** : Kích hoạt những **đỉnh hình tròn** đã cho ban đầu (những yếu tố đã có giá trị)
- **B2** : Lặp lại bước sau cho đến khi kích hoạt được tất cả những đỉnh ứng với những yếu tố cần tính hoặc không thể kích hoạt được bất kỳ đỉnh nào nữa.
- **Nếu** một đỉnh hình chữ nhật có cung nối với n đỉnh hình tròn mà $n-1$ đỉnh hình tròn đã được kích hoạt **thì** kích hoạt đỉnh hình tròn còn lại (và tính giá trị đỉnh còn lại này thông qua công thức ở đỉnh hình chữ nhật).

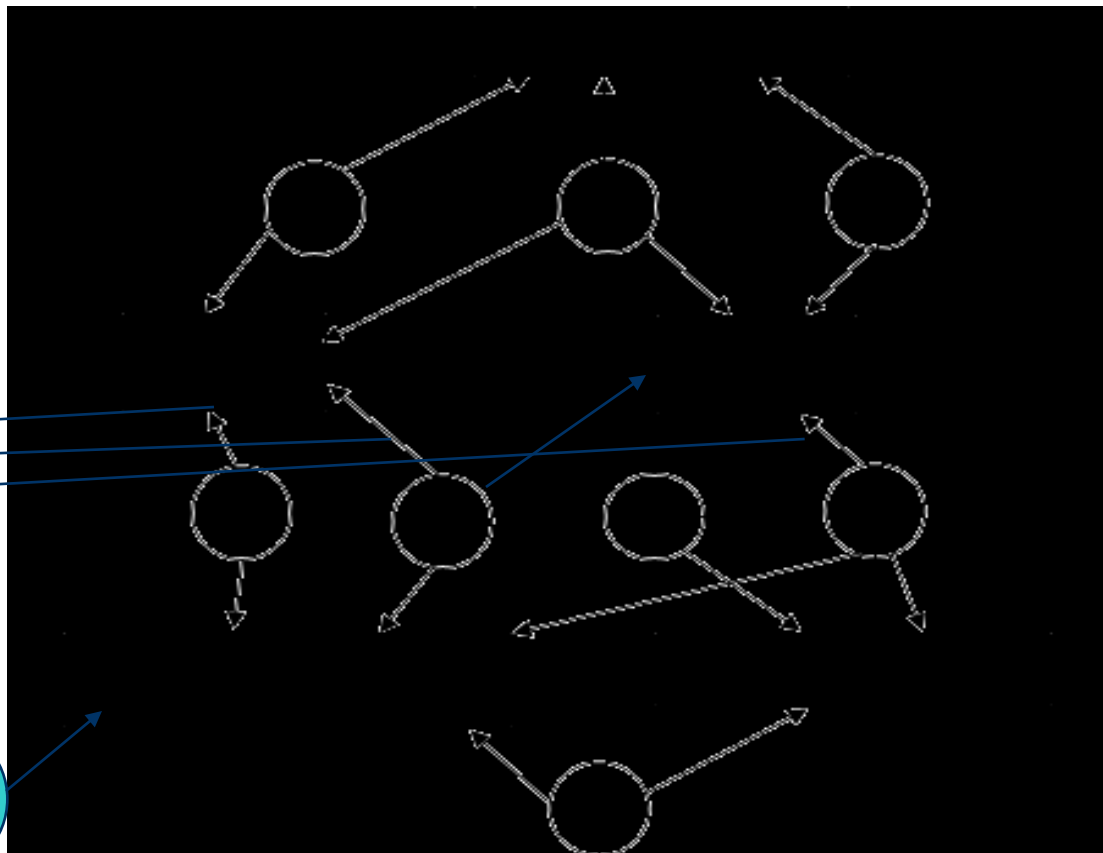
Ví dụ: Giải bt tam giác tổng quát (tt)

Ví dụ : "Cho hai góc ■■■■ và chiều dài cạnh a của tam giác. Tính chiều dài đường cao h_c ".

(6)

$$p = (a+b+c)/2$$

p



2.3.5 Logic

1. Logic mệnh đề

IF Xe không khởi động được (A)

AND Khoảng cách từ nhà đến chỗ làm là xa (B)

THEN Sẽ trễ giờ làm (C)

- Luật trên có thể biểu diễn lại như sau: $A \wedge B \rightarrow C$

2. Logic vị từ

- Logic vị từ, cũng giống như logic mệnh đề, dùng các ký hiệu để thể hiện tri thức. Những ký hiệu này gồm **hằng số**, **vị từ**, **biến** và **hàm**.

2.4 SUY DIỄN DỮ LIỆU

1. Modus ponens

1. E1

2. E1 \rightarrow E2

3. E2

Nếu có tiên đề khác, có dạng E2 \rightarrow E3 thì E3 được đưa vào danh sách.

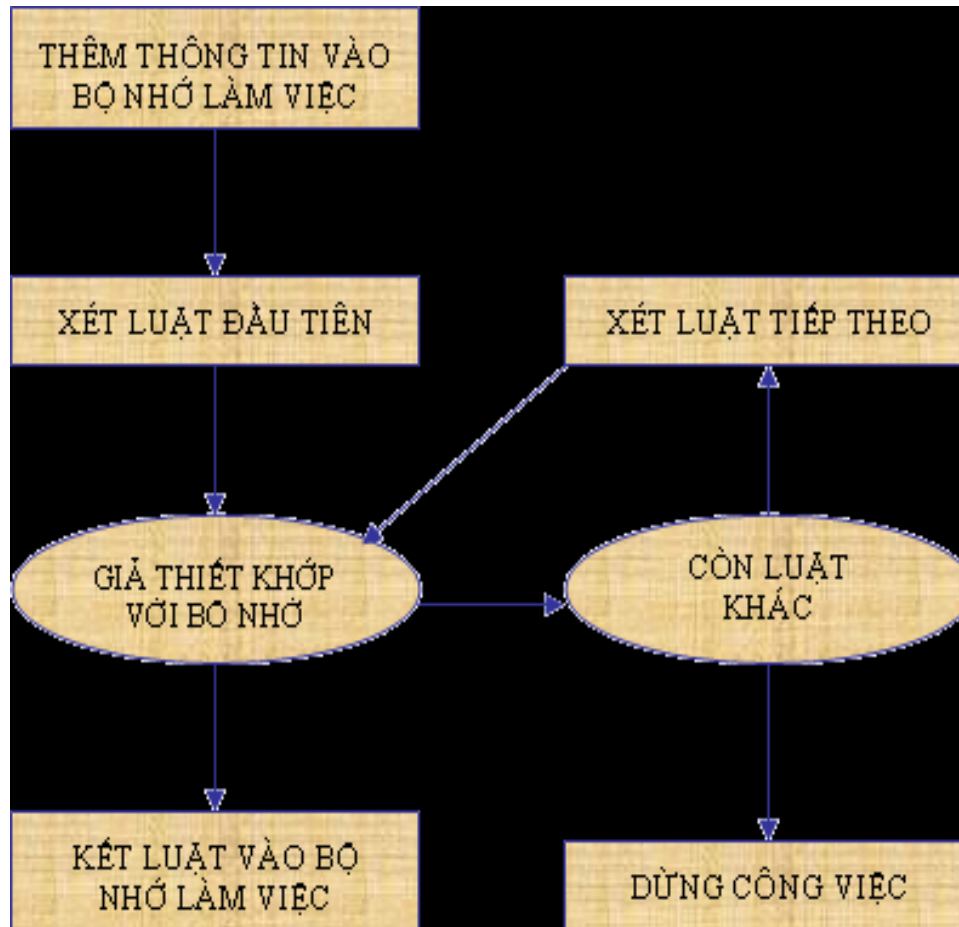
2. Modus tollens

1. \neg E2

2. E1 \rightarrow E2

3. \neg E1

2.4.2 Các hoạt động của Hệ thống Suy diễn tiến



Ví dụ về Suy diễn tiến

Luật 1. IF Bệnh nhân rát họng **AND** Nghi viêm nhiễm
THEN Tin rằng bệnh nhân viêm họng, đi chữa họng.

Luật 2. IF Nhiệt độ bệnh nhân quá 37 độ
THEN Bệnh nhân bị sốt

Luật 3. IF Bệnh nhân ốm trên 1 tuần **AND** Bệnh nhân sốt
THEN Nghi bệnh nhân viêm nhiễm.

Thông tin từ bệnh nhân là:

- Bệnh nhân có nhiệt độ 39 độ
- Bệnh nhân đã ốm hai tuần
- Bệnh nhân họng rát

Khi hệ thống thấy giả thiết của luật khớp với thông tin trong bộ nhớ, câu kết luận của luật được bổ sung vào bộ nhớ.

Minh hoa

Ví dụ suy diễn lùi

Cơ chế suy diễn

Suy diễn với logic mệnh đề:

1. Thuật toán suy diễn tiến

Input: - Tập luật Rule = $\{r_1, r_2, \dots, r_m\}$
- GT, KL

Output: Thông báo “thành công” nếu GT
Ngược lại, thông báo “không thành công”

Method:

```
TD=GT;
T=Loc(Rule, TD);
While (KL AND TD) AND (T) Do
{
    r = Get(T);
    TD=TD  $\setminus$  {r}; // r:left
    Rule = Rule  $\setminus$  {r};
    T=Loc(Rule, TD);
}
If KL THEN Return “True”
else Return “False”
```

Ví dụ: Rule = $\{r_1:a, r_2:b, r_3:a, r_4:a, r_5:b, r_6:e\}$
Hỏi a

Thuật toán suy diễn lùi

```
If KL  $\neq$  T THEN Return "True"
Else {TĐích=Vết = First=1; Quaylui= False;}
For Each q  $\in$  L DO TĐích=TĐích(q,0)};
Repeat
  first ++;
  ((f,i)=Get(TĐích);
  If (f  $\neq$  T) THEN
  {
    j = Tìmluật(f,i,Rule);          // rj: Leftj
    If (Tìm có rj) THEN
      { Vet = Vet(f,j)};
      For Each t  $\in$  Leftj \ GT) DO TĐích = TĐích(t,0)};
    else
      { Quaylui=True;
        While (f  $\in$  L) AND Quaylui DO
          {
            Repeat          { (g,k)=Get(Vết);
                             TĐích = TĐích \ Leftk;}
            Until f  $\neq$  leftk;
            l=Tìmluật(g,k,Rule);
```

Thuật toán suy diễn lùi

```

If (Tìm có ri) THEN
    { TĐích = TĐích \ Lefti ;
      For Each t (Lefti \ GT) DO
          TĐích = TĐích (t,0);
          Vết = Vết (g,l);
          Quaylui = False;
        }
    else f=g;
  }
}

```

Until (TĐích = **TRue** OR ((f **TRue** L) and (First>2));

If (f **TRue** L) then Return False else Return TRue;

Ví dụ: Rule = {r₁:a **TRue** r₂:b **TRue** r₃:a **TRue** r₄:a **TRue** r₅:b **TRue** r₆:e **TRue**}

Hỏi a **TRue** **TRue** , a **TRue**

2.4.3 Ưu điểm

* Suy diễn tiến

- Ưu điểm chính của suy diễn tiến là làm việc tốt khi bài toán về bản chất đi thu thập thông tin rồi thấy điều cần suy diễn.
- Suy diễn tiến cho ra khối lượng lớn các thông tin từ một số thông tin ban đầu. Nó sinh ra nhiều thông tin mới.
- Suy diễn tiến là tiếp cận lý tưởng đối với loại bài toán cần giải quyết các nhiệm vụ như lập kế hoạch, điều hành điều khiển và diễn dịch.

* Suy diễn lùi

- Một trong các ưu điểm chính của suy diễn lùi là phù hợp với bài toán đưa ra giả thuyết rồi xem hiệu quả giả thiết đó có đúng không.
- Suy diễn lùi tập trung vào đích đã cho. Nó tạo ra một loạt câu hỏi chỉ liên quan đến vấn đề đang xét, đến hoàn cảnh thuận tiện đối với người dùng.
- Khi suy diễn lùi muốn suy diễn cái gì đó từ các thông tin đã biết, nó chỉ tìm trên một phần của cơ sở tri thức thích đáng đối với bài toán đang xét.

2.4.4 Nhược điểm

* Suy diễn tiến

- Một nhược điểm chính của hệ thống suy diễn tiến là không cảm nhận được rằng chỉ một vài thông tin là quan trọng. Hệ thống hỏi các câu hỏi có thể hỏi mà không biết rằng chỉ một ít câu đã đi đến kết luận được.
- Hệ thống có thể hỏi cả câu không liên quan. Có thể các câu trả lời cũng quan trọng, nhưng làm người dùng lúng túng khi phải trả lời các câu không dính đến chủ đề.

* Suy diễn lùi

- Nhược điểm cơ bản của suy diễn này là nó thường tiếp theo dòng suy diễn, thay vì đúng ra phải đúng ở đó mà sang nhánh khác. Tuy nhiên có thể dùng nhân tố tin cậy và các luật meta để khắc phục.

Chương 3: Hệ MYCIN

3.1 Giới thiệu

MYCIN là một hệ lập luận trong y học được hoàn tất vào năm 1970 tại đại học Stanford, Hoa Kỳ. Đây là một hệ chuyên gia dựa trên luật và sự kiện. MYCIN sử dụng cơ chế lập luận gần đúng để xử lý các luật suy diễn dựa trên độ đo chắc chắn. Tiếp theo sau MYCIN, hệ EMYCIN ra đời. EMYCIN là một hệ chuyên gia tổng quát được tạo lập bằng cách loại phần cơ sở tri thức ra khỏi hệ MYCIN. EMYCIN cung cấp một cơ chế lập luận và tùy theo bài toán cụ thể sẽ bổ sung tri thức riêng của bài toán đó để tạo thành hệ chuyên gia.

Các đặc điểm chính:

- Sử dụng kỹ thuật suy diễn lùi.
- Có khả năng phân tích tri thức và điều khiển.
- Có tích hợp Meta-Rule.
- Có thể dùng khi thiếu thông tin hoặc thông tin không chắc chắn.
- Dễ sử dụng, giao tiếp bằng tiếng Anh.
- Cung cấp các chức giải thích: HOW, WHY.

Một phiên làm việc cụ thể với MYCIN:

?Thông tin về bệnh nhân

- MYCIN: Họ và tên?
- User: Jack Durkin.
- MYCIN: Tuổi?
- User: 24.
- MYCIN: Giới tính?
- User: Nam.
- MYCIN: Chủng tộc?
- User: CAUCASIAN.

?Xác định bản chất lây nhiễm

- MYCIN: Có bất kỳ phép cấy vi khuẩn cho Jack Durkin liên quan đến bệnh hiện tại?
- User: Có.
- MYCIN: Mẫu vật lấy từ đâu?
- User: CSF.
- MYCIN: Ngày giờ thực hiện cấy:
- User: 5-6-77 8:09 AM
-
- MYCIN: Gần đây Jack Durkin có các triệu
- chứng như: choáng váng, hôn mê không?
- User: Có.
-

CÁC THÀNH PHẦN CỦA HỆ MYCIN

1. Chương trình tư vấn: Cung cấp cho các Bác sĩ các lời khuyên để chọn phương pháp điều trị thích hợp bằng cách xác định rõ cách thức điều trị bởi các dữ liệu lấy ra từ các phòng thí nghiệm lâm sàng thông qua các câu trả lời của bác sĩ cho câu hỏi của máy tính.
2. Khả năng giải thích có tác động qua lại: Cho phép chương trình tư vấn giải thích các kiến thức của nó về các phương pháp điều trị và chứng minh các chú thích về các phương pháp điều trị đặc biệt.
3. Thu nạp tri thức: cho phép các chuyên gia con người trong lĩnh vực điều trị các căn bệnh truyền nhiễm dạy cho MYCIN các luật quyết định theo phương pháp điều trị mà họ tìm thấy trong thực tế lâm sàng.

PHẠM VI SỬ DỤNG CỦA HỆ MYCIN

1. Chẩn đoán nguyên nhân gây bệnh: đối với các bác sĩ điều trị, khi xét nghiệm cho bệnh nhân để có kết quả chẩn đoán chắc chắn mất 24-48 giờ. Nhiều trường hợp phải điều trị cả ngay khi chưa có kết luận hoàn chỉnh. MYCIN giúp chẩn đoán nguyên nhân gây bệnh nhanh hơn: khi gọi chương trình MYCIN, các bác sĩ trả lời các câu hỏi về tiền sử bệnh nhân, bệnh án, các kết quả xét nghiệm, các triệu chứng, ... từ đó MYCIN đưa ra chẩn đoán bệnh.
2. Tạo ra phương pháp điều trị: Sau khi nhận được các câu trả lời của bác sĩ về tình trạng bệnh nhân thông qua đối thoại. Trong trường hợp câu trả lời không biết hoặc biết không chắc chắn, thì MYCIN sẽ suy luận từ các thông tin không hoàn chỉnh.
3. Dự đoán diễn biến của bệnh: Bằng các câu hỏi “HOW, WHY”, MYCIN sẽ giải thích các nguyên nhân và lý do cho các bác sĩ. Sau khi việc chẩn đoán bệnh và kê đơn hoàn tất, bác sĩ có thể theo dõi toàn bộ quá trình chẩn đoán bệnh của MYCIN và qua đó theo dõi diễn biến của bệnh

NGUYÊN NHÂN THÀNH CÔNG CỦA MYCIN

1. Sự cần thiết của việc tư vấn dùng kháng sinh của các bác sĩ: vào thời điểm này việc lạm dụng kháng sinh đã đem lại không ít phản ứng phụ.
2. Cơ sở tri thức của MYCIN được thu nạp từ các chuyên gia xuất sắc nhất trong lĩnh vực.
3. MYCIN không bao giờ đi đến ngay kết luận để luôn có thêm các thông tin cốt yếu qua mỗi bước.
4. MYCIN được hình thành từ một chương trình trí tuệ nhân tạo đã được áp dụng thực tế (DENDRAL) và đã được thực hiện tại trung tâm y tế nổi tiếng với các tri thức mới nhất về bệnh học và dược học.

3.2 LÝ THUYẾT VỀ SỰ CHẮC CHẮN

MB (Measure of Belief in): độ đo sự tin cậy

MD (Measure of Disbelief in): độ đo sự không tin cậy

CF (Certainly Factor): Hệ số chắc chắn

Gọi: **MB**(H/E) là độ đo sự tin cậy của giả thuyết H khi có chứng cứ E.

MD(H/E) là độ đo sự không tin cậy của giả thuyết H khi có chứng cứ E.

Khi đó: $0 < \mathbf{MB}(H/E) < 1$ trong khi $\mathbf{MD}(H/E) = 0$

$0 < \mathbf{MD}(H/E) < 1$ trong khi $\mathbf{MB}(H/E) = 0$

Độ đo chắc chắn **CF**(H/E) được tính bằng công thức:

$$\mathbf{CF}(H/E) = \mathbf{MB}(H/E) - \mathbf{MD}(H/E)$$

3.2 LÝ THUYẾT VỀ SỰ CHẮC CHẮN (tiếp)

1. Luật đơn giản: $r: \text{If}(e) \text{ then } (c)$

$CF(e)$ là độ đo chắc chắn của chứng cứ.

$CF(r)$ là độ đo chắc chắn của luật suy diễn.

Khi đó: $CF(c)$ là độ đo chắc chắn của kết luận sẽ được tính bằng công thức:

$$CF(c) = CF(e) * CF(r)$$

2. Luật phức tạp:

- $\text{If}(e1 \text{ AND } e2) \text{ then } (c)$

$$CF(e1 \text{ AND } e2) = \text{MIN}(CF(e1), CF(e2))$$

- $\text{if } (e1 \text{ OR } e2) \text{ then } (c)$

$$CF(e1 \text{ OR } e2) = \text{MAX}(CF(e1), CF(e2))$$

3.2 LÝ THUYẾT VỀ SỰ CHẮC CHẴN (tiếp)

3. Với luật: if ((e1 AND e2) OR e3) then (c)

$$CF((e1 \text{ AND } e2) \text{ OR } e3) = \text{MAX}(\text{MIN}(CF(e1), CF(e2)), CF(e3))$$

4. $CF(\text{NOT } e) = -CF(e)$

5. Kết hợp nhiều luật có cùng kết luận:

- Luật 1: If(e1) then (c) với $CF(r1)$: độ đo chắc chắn của luật 1
- Luật 2: If(e2) then (c) với $CF(r2)$: độ đo chắc chắn của luật 2

- Với $CF(t1)$, $CF(t2)$ là CF của kết luận của luật 1 và 2, khi $CF(t1)$ và $CF(t2)$ đều dương thì:

$$C_{\text{tổng}} = CF(t1) + CF(t2) - CF(t1) * CF(t2)$$

- Khi $CF(t1)$ và $CF(t2)$ đều âm thì:

$$C_{\text{tổng}} = CF(t1) + CF(t2) + CF(t1) * CF(t2)$$

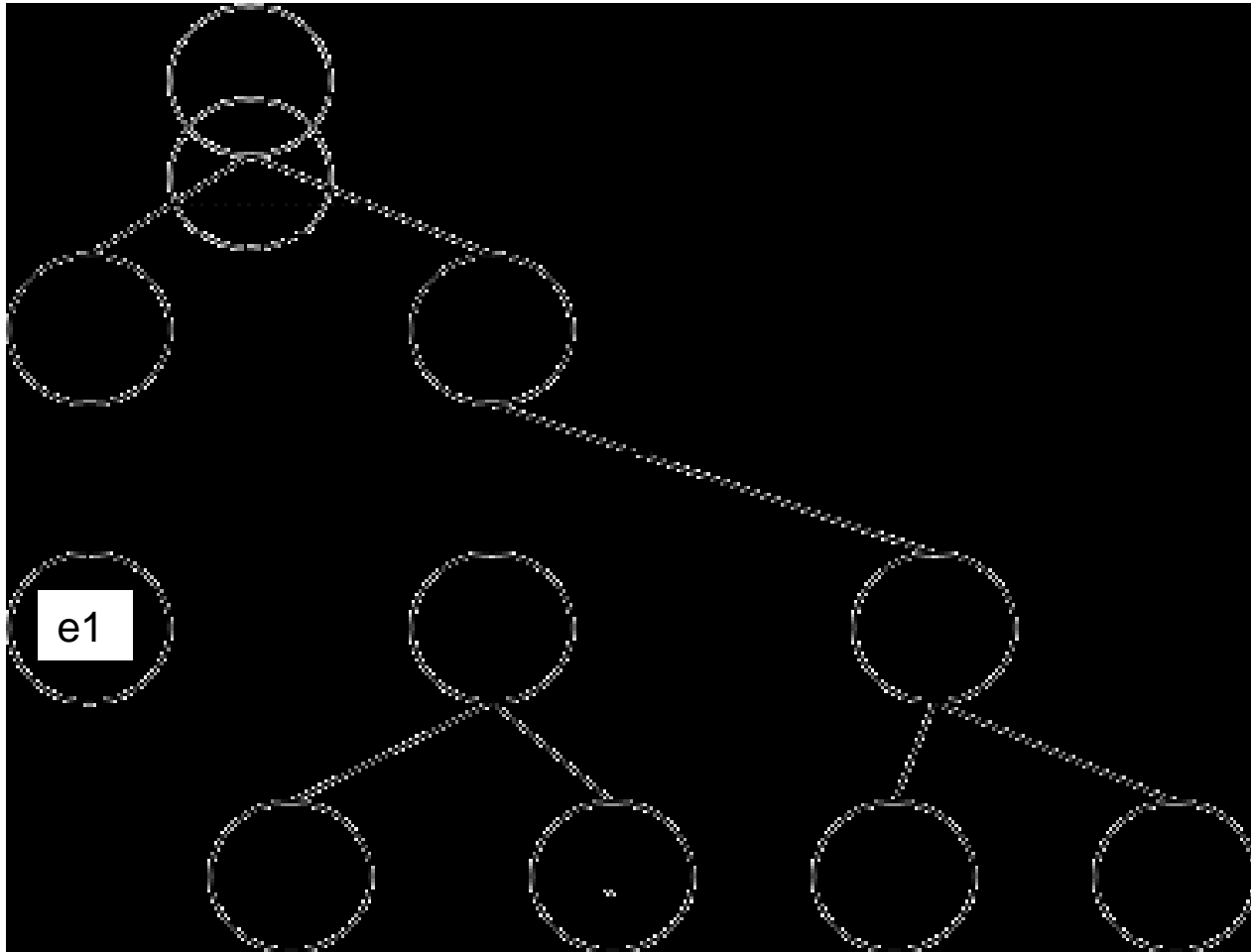
- Nếu $CF(t1)$ khác dấu với $CF(t2)$ thì:

$$C_{\text{tổng}} = (CF(t1) + CF(t2)) / (1 - \text{MIN}(\text{ABS}(CF(t1)), \text{ABS}(CF(t2))))$$

Ví dụ về lập luận trong Hệ MYCIN

- Ví dụ: Có 7 luật sau đây:
 - r1: **If**(e1) **Then** (c1) **CF**(r1) = 0,8
 - r2: **If** (e2) **Then** (c2) **CF**(r2) = 0,9
 - r3: **If** (e3) **Then** (c2) **CF**(r3) = 0,7
 - r4: **If** (e4) **Then** (c3) **CF**(r4) = 0,6
 - r5: **If** (**NOT** e5) **Then** (c3) **CF**(r5) = 0,5
 - r6: **If** (c2 **AND** c3) **Then** (c4) **CF**(r6) = 0,9
 - r7: **If** (c1 **OR** c4) **Then** (c5) **CF**(r7) = 0,8
- Bảng luật này tạo thành mạng suy diễn ở hình 3.1 với c5 là giả thuyết cần hướng đến.

Hình 3.1. Mạng suy diễn



Lập luận trên mạng suy diễn

- Giả sử các chứng cứ e_1, e_2, e_3, e_4, e_5 có độ đo chắc chắn như sau:

$$CF(e_1) = 0,9$$

$$CF(e_2) = 0,9$$

$$CF(e_3) = -0,3$$

$$CF(e_4) = 0,4$$

$$CF(e_5) = -0,3$$

Lập luận trên mạng suy diễn (tiếp)

- Chúng ta sẽ lập luận từ các **CF** của chứng cứ dần lên giả thuyết c5 như sau:
- Dựa vào luật r1 tính được **CF(c1)**:
- **CF(c1) = CF(e1) * CF(r1) = 0,8 * 0,9 = 0,72**
- Dựa vào luật r2, r3 tính được **CF(c2)**
- Với luật r2: **CF(c2) = CF(e2) * CF(r2) = 0,9 * 0,9 = 0,81**
- Với luật r3: **CF(c2) = CF(e3) * CF(r3) = -0,3 * 0,7 = -0,21**
- Do **CF(c2)** của r2 trái dấu với **CF(c2)** của r3, nên:
CF(c2)_{tổng} = (0,81 + (-0,21)) / (1 - MIN(0,81, 0,21)) = 0,74

Lập luận trên mạng suy diễn (tiếp)

- Dựa vào luật r4, r5 ta tính được $CF(c3)$
- Với luật r4:
- $CF(c3) = CF(e4) * CF(r4) = 0,4 * 0,6 = 0,24$
- Với luật r5:
- $CF(c3) = CF(\text{NOT } e5) * CF(r5) = -CF(e5) * CF(r5) = 0,3 * 0,5 = 0,15$
- Do $CF(c3)$ của r4 và $CF(c3)$ của r5 cùng dương nên $CF(c3)_{\text{tổng}} = 0,24 + 0,15 - 0,24 * 0,15 = 0,354$
- Dựa vào luật r6 ta tính được $CF(c4)$:
- $CF(c4) = \text{MIN}(CF(c2), CF(c3)) * CF(r6) = \text{MIN}(0,74, 0,354) * 0,9 = 0,354 * 0,9 = 0,3186$
- Dựa vào luật r7 ta tính được $CF(c5)$
- $CF(c5) = \text{MAX}(CF(c1), CF(c4)) * CF(r7) = \text{MAX}(0,72, 0,3186) * 0,8 = 0,576$
- Như thế độ chắc chắn của giả thuyết c5 là 0,576.

Chương 4 Hệ học

4.1 MỞ ĐẦU

- Các chương trước đã thảo luận về biểu diễn và suy luận tri thức. Trong trường hợp này giả định đã có sẵn tri thức và có thể biểu diễn tường minh tri thức.
- Tuy vậy trong nhiều tình huống, sẽ không có sẵn tri thức như:
 - Kỹ sư tri thức cần thu nhận tri thức từ chuyên gia lĩnh vực.
 - Cần biết các luật mô tả lĩnh vực cụ thể.
 - Bài toán không được biểu diễn tường minh theo luật, sự kiện hay các quan hệ.
- Có hai tiếp cận cho hệ thống học:
 - Học từ ký hiệu: bao gồm việc hình thức hóa, sửa chữa các luật tường minh, sự kiện và các quan hệ.
 - Học từ dữ liệu số: được áp dụng cho những hệ thống được mô hình dưới dạng số liên quan đến các kỹ thuật nhằm tối ưu các tham số. Học theo dạng số bao gồm mạng Neural nhân tạo, thuật giải di truyền, bài toán tối ưu truyền thống. Các kỹ thuật học theo số không tạo ra CSTT tường minh.

4.2 CÁC HÌNH THỨC HỌC

1. **Học vẹt:** Hệ tiếp nhận các khẳng định của các quyết định đúng. Khi hệ tạo ra một quyết định không đúng, hệ sẽ đưa ra các luật hay quan hệ đúng mà hệ đã sử dụng. Hình thức học vẹt nhằm cho phép chuyên gia cung cấp tri thức theo kiểu tương tác.
2. **Học bằng cách chỉ dẫn:** Thay vì đưa ra một luật cụ thể cần áp dụng vào tình huống cho trước, hệ thống sẽ được cung cấp bằng các chỉ dẫn tổng quát. Ví dụ: "gas hầu như bị thoát ra từ van thay vì thoát ra từ ống dẫn". Hệ thống phải tự mình đề ra cách biến đổi từ trừu tượng đến các luật khả dụng.
3. **Học bằng qui nạp:** Hệ thống được cung cấp một tập các ví dụ và kết luận được rút ra từ từng ví dụ. Hệ liên tục lọc các luật và quan hệ nhằm xử lý từng ví dụ mới.

4.2 CÁC HÌNH THỨC HỌC (Tiếp)

4. **Học bằng tương tự:** Hệ thống được cung cấp đáp ứng đúng cho các tác vụ tương tự nhưng không giống nhau. Hệ thống cần làm thích ứng đáp ứng trước đó nhằm tạo ra một luật mới có khả năng áp dụng cho tình huống mới.
5. **Học dựa trên giải thích:** Hệ thống phân tích tập các lời giải ví dụ (và kết quả) nhằm ấn định khả năng đúng hoặc sai và tạo ra các giải thích dùng để hướng dẫn cách giải bài toán trong tương lai.
6. **Học dựa trên tình huống:** Bất kỳ tính huống nào được hệ thống lập luận đều được lưu trữ cùng với kết quả cho dù đúng hay sai. Khi gặp tình huống mới, hệ thống sẽ làm thích nghi hành vi đã lưu trữ với tình huống mới.
7. **Khám phá hay học không giám sát:** Thay vì có mục tiêu tường minh, hệ khám phá liên tục tìm kiếm các mẫu và quan hệ trong dữ liệu nhập. Các ví dụ về học không giám sát bao gồm gom cụm dữ liệu, học để nhận dạng các đặc tính cơ bản như cạnh từ các điểm ảnh.

Ví dụ về CÁC HÌNH THỨC HỌC

Ví dụ:

- **Hệ MYCIN**
- **Mạng Neural nhân tạo**
- **Thuật toán học Quinland**
- **Bài toán nhận dạng**
- **Máy chơi cờ carô, cờ tướng**

4.3 THUẬT GIẢI Quinlan

- Là thuật toán học theo quy nạp dùng luật, đa mục tiêu.
- Do Quinlan đưa ra năm 1979.
- Ý tưởng: Chọn thuộc tính quan trọng nhất để tạo cây quyết định.
- Thuộc tính quan trọng nhất là thuộc tính phân loại Bảng quan sát thành các bảng con sao cho từ mỗi bảng con này dễ phân tích để tìm quy luật chung.

4.3.1 THUẬT GIẢI A. Quinlan

STT	Size	Nationality	Family	Conclusion
1	Small	German	Single	A
2	Large	French	Single	A
3	Large	German	Single	A
4	Small	Italian	Single	B
5	Large	German	Married	B
6	Large	Italian	Single	B
7	Large	Italian	Married	B
8	Small	German	Married	B

Với mỗi thuộc tính của bảng quan sát:

- Xét vector V : có số chiều bằng số phân loại
 - $V_{(\text{Size}=\text{Small})} = (A_{\text{Small}}, B_{\text{Small}})$
 - $A_{\text{Small}} = \text{Số quan sát A có Size là Small} / \text{Tổng số quan sát có Size=Small}$
 - $B_{\text{Small}} = \text{Số quan sát B có Size là Small} / \text{Tổng số quan sát có Size=Small}$
 - $V_{(\text{Size}=\text{Small})} = (1/3, 2/3)$
 - $V_{(\text{Size}=\text{Large})} = (2/5, 3/5)$
- Với thuộc tính Nationality
 - $V_{(\text{Nat} = \text{German})} = (2/4, 2/4)$
 - $V_{(\text{Nat} = \text{French})} = (1, 0)$
 - $V_{(\text{Nat} = \text{Italian})} = (0/3, 3/3) = (0, 1)$
- Thuộc tính Family:
 - $V_{(\text{Family}=\text{Single})} = (3/5, 2/5)$
 - $V_{(\text{Family} = \text{Married})} = (0/3, 3/3) = (0, 1)$

Với mỗi thuộc tính của bảng quan sát:

Chỉ còn xét German

- Thuộc tính Size:

$$V_{(\text{Size}=\text{Small})} = (1/2, 1/2)$$

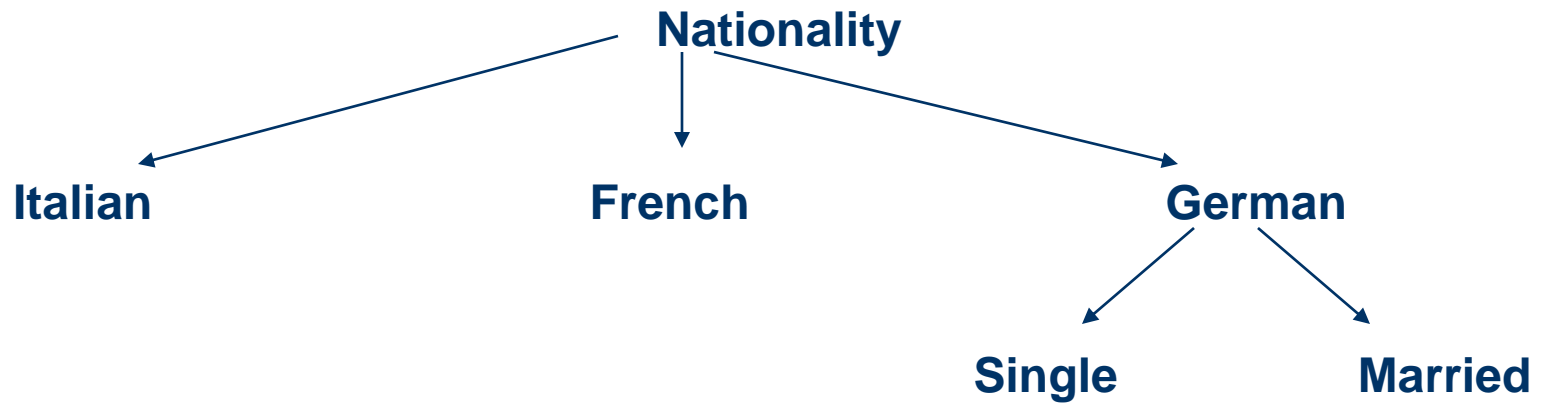
$$V_{(\text{Size}=\text{Large})} = (1/2, 1/2)$$

- Thuộc tính Family:

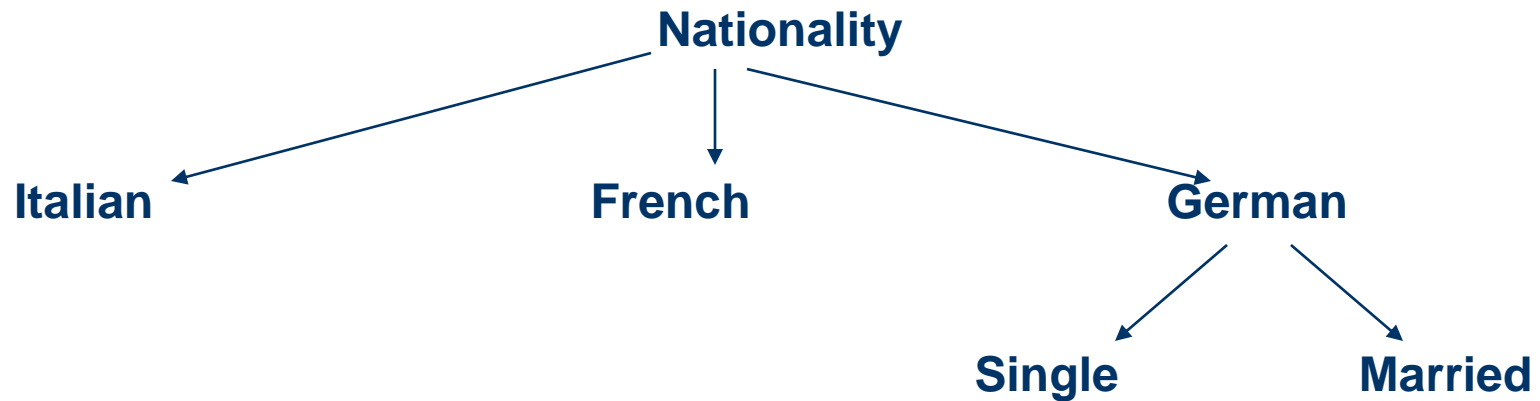
$$V_{(\text{Family}=\text{Single})} = (2/2, 0, 2) = (1, 0)$$

$$V_{(\text{Family}=\text{Married})} = (0/2, 2/2) = (0, 1)$$

STT	Size	Family	Conclusion
1	Small	Single	A
2	Large	Single	A
3	Large	Married	B
4	Small	Married	B



Với mỗi thuộc tính của bảng quan sát(tiếp)



Rule 1: If (Nationality IS Italian) then (Conclusion IS B)

Rule 2: If (Nationality IS French) then (Conclusion IS A)

**Rule 3: If (Nationality IS German) AND (Family IS Single)
then (Conclusion IS A)**

**Rule 4: If (Nationality IS German) AND (Family IS Married)
then (Conclusion IS B)**

Thuật giải: Học theo độ bất định

Stt	Age	Competition	Type	Profit
1	Old	No	Software	Down
2	Midle	Yes	Software	Down
3	Midle	No	Hardware	Up
4	Old	No	Hardware	Down
5	New	No	Hardware	Up
6	New	No	Software	Up
7	Midle	No	Software	Up
8	New	Yes	Software	Up
9	Midle	Yes	Hardware	Down
10	Old	Yes	Hardware	Down

Học theo độ bất định(tiếp)

- Độ bất định của X: $E(X) = \sum_{i=1}^n p_i \log_2 p_i$
- Tính Entropy cho mỗi thuộc tính và chọn thuộc tính có Entropy nhỏ nhất.

$$E(C / A) = \sum_{i=1}^n p(c_i, a_i) \log_2 p(c_i, a_i)$$

$$E(C / \text{Competition}_{No}) = \frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} = 0.918$$

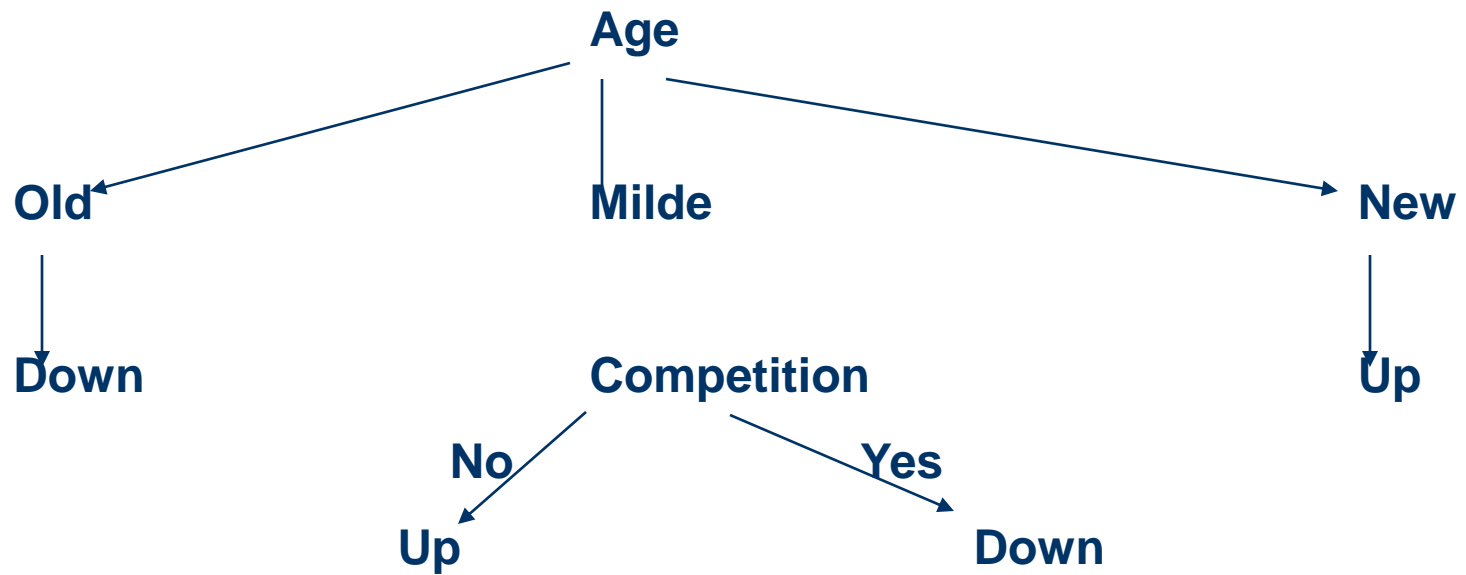
$$E(C / \text{Competition}_{Yes}) = \frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4} = 0.811$$

$$E(C / \text{Competition}) = 0.6 * 0.918 + 0.4 * 0.811 = 0.8752$$

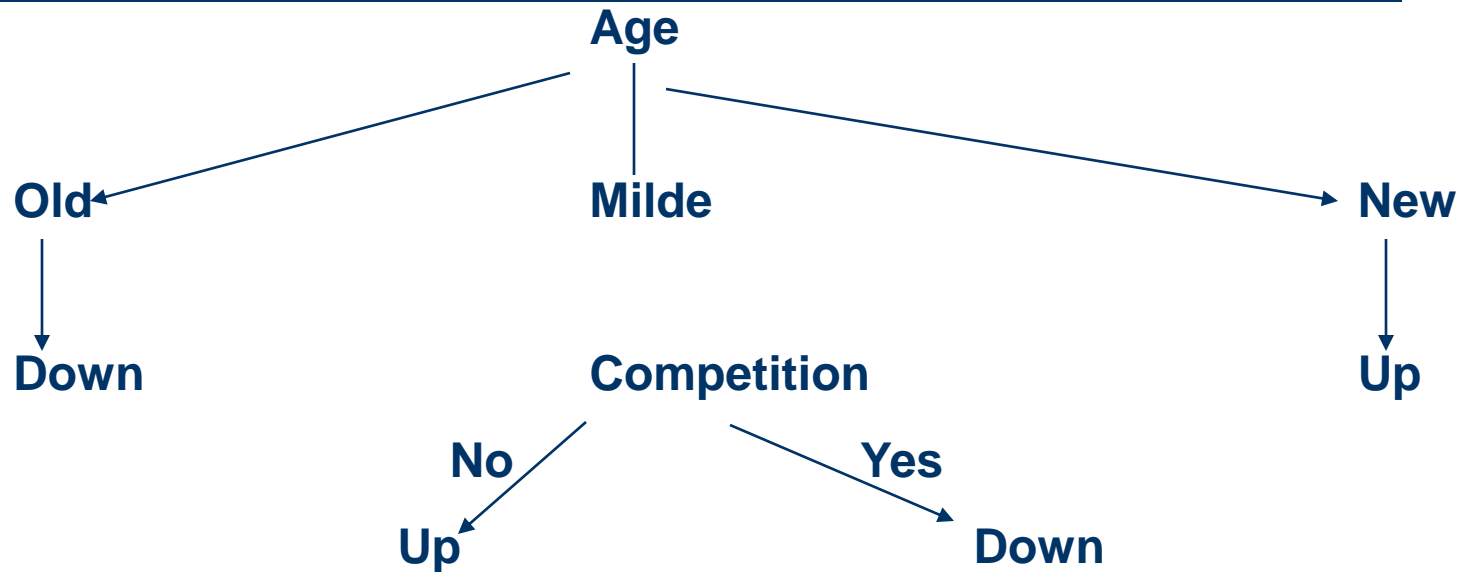
Học theo độ bất định(tiếp)

- Tương tự:
 $E(C/Age) = 0.4$
 $E(C/Type) = 1$
- Age cho nhiều thông tin nhất

STT	Competition	Type	Profit
1	Yes	Software	Down
2	No	Hardware	Up
3	No	Software	Up
4	Yes	Hardware	Down



Học theo độ bất định(tiếp)



Rule 1: If (Age IS Old) then (Profit IS Down)

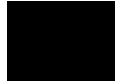
Rule 2: If (Age IS New) then (Profit IS Up)

**Rule 3: If (Age IS Midle) And (Competition IS No)
then (Profit IS Up)**

**Rule 4: If (Age IS Midle) And (Competition IS Yes)
then (Profit IS Down)**

4.4 THUẬT GIẢI ILA(Inductive Learning Algorithm)

Xác định dữ liệu

1. Tập mẫu được liệt kê trong một bảng, với mỗi dòng tương ứng một mẫu, và mỗi cột thể hiện một thuộc tính trong mẫu.
2. Tập mẫu có m mẫu, mỗi mẫu gồm k thuộc tính và có một thuộc tính quyết định. Tổng số n các giá trị của thuộc tính quyết định chính là số lớp của tập mẫu.
3. Tập luật R có giá trị khởi tạo là 
4. Tất cả các dòng trong bảng ban đầu chưa được đánh dấu (kiểm tra).

4.4 THUẬT GIẢI ILA(tiếp)

- *Bước 1:* Chia bảng m mẫu ban đầu thành n bảng con. Mỗi bảng con ứng với một giá trị của thuộc tính phân lớp của tập mẫu.
- (* thực hiện các bước 2 đến 8 cho mỗi bảng con*)
- *Bước 2:* Khởi tạo biến đếm kết hợp thuộc tính j , $j=1$.
- *Bước 3:* Với mỗi bảng con đang khảo sát, phân chia danh sách các thuộc tính theo các tổ hợp phân biệt, mỗi tổ hợp ứng với j thuộc tính phân biệt.
- *Bước 4:* Với mỗi tổ hợp các thuộc tính, tính số lượng các giá trị thuộc tính xuất hiện theo cùng tổ hợp thuộc tính trong các dòng chưa được đánh dấu của bảng con đang xét (mà đồng thời không xuất hiện với tổ hợp thuộc tính này trên các bảng còn lại). Gọi tổ hợp đầu tiên (trong bảng con) có số lần xuất hiện nhiều nhất là *tổ hợp lớn nhất*.

4.4 THUẬT GIẢI ILA(tiếp)

- *Bước 5:* Nếu tổ hợp lớn nhất bằng \square tăng j lên 1 và quay lại bước 3.
- *Bước 6:* Đánh dấu các dòng thoả tổ hợp lớn nhất của bảng con đang xử lý theo lớp.
- *Bước 7:* Thêm luật mới vào tập luật R, với vế trái là tập các giá trị của thuộc tính ứng với tổ hợp lớn nhất (kết hợp các thuộc tính bằng toán tử AND) và vế phải là giá trị thuộc tính quyết định tương ứng.
- *Bước 8:* Nếu tất cả các dòng đều đã được đánh dấu phân lớp, tiếp tục thực hiện từ bước 2 cho các bảng con còn lại. Ngược lại (nếu chưa đánh dấu hết các dòng) thì quay lại bước 4. Nếu tất cả các bảng con đã được xét thì kết thúc, kết quả thu được là tập luật cần tìm.

Minh họa thuật giải ILA

Mẫu số	Size	Color	Shape	Decision
1	medium	blue	brick	yes
2	small	red	wedge	no
3	small	red	sphere	yes
4	large	red	wedge	no
5	large	green	pillar	yes
6	large	red	pillar	no
7	large	green	sphere	yes

Bảng 4.1 Tập mẫu học cho bài toán phân lớp đối tượng

Minh họa thuật giải ILA(tiếp): Bước 1

<i>Bảng con 1</i>					
Mẫu số cũ, mới		Size	Color	Shape	Decision
1	1	medium	blue	brick	yes
3	2	small	red	sphere	yes
5	3	large	green	pillar	yes
7	4	large	green	sphere	yes
<i>Bảng con 2</i>					
Mẫu số cũ	mới	Size	Color	Shape	Decision
2	1	small	red	wedge	no
4	2	large	red	wedge	no
6	3	large	red	pillar	no

Bảng 4.2. Với $n=2$, Chia thành hai bảng con theo thuộc tính Decision

Minh họa thuật giải ILA(*tiếp*)

- Áp dụng bước 2 của thuật giải vào bảng con thứ nhất trong bảng trên. Với $j=1$, danh sách các tổ hợp thuộc tính gồm có {Size}, {Color}, và {Shape}.
- Với tổ hợp {Size}, giá trị thuộc tính "medium" xuất hiện trong bảng con thứ nhất nhưng không có trong bảng con thứ hai, do đó giá trị tổ hợp lớn nhất là "medium". Bởi vì các giá trị thuộc tính "small" và "large" xuất hiện trong cả hai bảng con, nên không được xét trong bước này. Với tổ hợp {Size}, giá trị thuộc tính "medium" chỉ bằng 1

Minh họa thuật giải ILA(*tiếp*)

- Xét tiếp cho tổ hợp {Color} thì giá trị tổ hợp lớn nhất là bằng 2, ứng với thuộc tính "green", còn thuộc tính "blue" là bằng 1.
- Tương tự, với tổ hợp {Shape}, ta có "brick" xuất hiện một lần, và "sphere" hai lần. Đến cuối bước 4, ta có tổ hợp {Color} với thuộc tính "green" và {Shape} với thuộc tính "sphere" đều có số lần xuất hiện lớn nhất là 2. Thuật toán mặc định chọn trường hợp thứ nhất để xác định luật tổ hợp lớn nhất. Dòng 3 và 4 được đánh dấu đã phân lớp, ta có luật dẫn như sau:

Rule 1: IF color IS green THEN decision IS yes

Minh họa thuật giải ILA(*tiếp*)

- Tiếp tục thực hiện bước 4 đến 8 cho các mẫu còn lại (chưa đánh dấu) trong bảng con này (tức dòng 1 và 2). Áp dụng tương tự như trên, ta thấy giá trị thuộc tính "medium" của {Size}, "blue" của "Color", "brick" và "sphere" của {Shape} đều xuất hiện một lần. Bởi vì số lần xuất hiện này giống nhau, thuật giải áp dụng luật mặc định chọn trường hợp đầu tiên. Ta có thêm luật sau:

Rule 2: IF size IS medium THEN decision IS yes

Đánh dấu cho dòng 1 trong bảng con thứ nhất. Tiếp tục áp dụng bước 4 đến 8 trên dòng còn lại (tức dòng 2). Giá trị thuộc tính "sphere" của {Shape} xuất hiện một lần, ta có luật thứ ba:

Rule 3: IF shape IS sphere THEN decision IS yes

Dòng 2 được đánh dấu. Như vậy, tất cả các dòng trong bảng con 1 đã được đánh dấu, ta chuyển qua xử lý tiếp bảng con 2.

Minh họa thuật giải ILA(*tiếp*)

• Thuộc tính "wedge" của {Shape} xuất hiện hai lần trong dòng 1 và 2 của bảng con này. Đánh dấu các dòng này với luật dẫn thứ tư sau:

Rule 4: IF shape IS wedge THEN decision IS no

• Với dòng còn lại (tức dòng 3) của bảng con 2, ta có thuộc tính {Size} với giá trị "large" có xuất hiện trong bảng con 1. Do đó, theo thuật giải, ta loại bỏ trường hợp này. Tương tự như vậy cho giá trị "red" của {Color} và "pillar" của {Shape}. Khi đó, ILA tăng j lên 1, và khởi tạo các tổ hợp 2 thuộc tính là {Size và Color}, {Size và Shape}, và {Color và Shape}. Các tổ hợp thứ nhất và thứ ba thoả mãn điều kiện không xuất hiện trong bảng con 1 với các cặp thuộc tính hiện có của dòng này. Theo luật mặc định, ta chọn luật theo trường hợp thứ nhất. Đánh dấu dòng này, ta có thêm luật dẫn thứ 5:

Rule 5: IF size IS large AND color IS red THEN decision IS no

Minh họa thuật giải ILA(*tiếp*)

Rule 1: IF color IS green THEN decision IS yes

Rule 2: IF size IS medium THEN decision IS yes

Rule 3: IF shape IS sphere THEN decision IS yes

Rule 4: IF shape IS wedge THEN decision IS no

Rule 5: IF size IS large AND color IS red THEN decision IS no

Đánh giá thuật giải:

- Số lượng các luật thu được xác định mức độ thành công của thuật giải. Đây chính là mục đích chính của các bài toán phân lớp thông qua một tập mẫu học. Ngoài ra, để đánh giá các hệ học quy nạp là khả năng hệ thống có thể phân lớp các mẫu được đưa vào sau này.
- Thuật giải ILA được đánh giá mạnh hơn hai thuật giải về phương pháp học quy nạp trước đây là ID3 và AQ).

Chương 5: Hệ thống mờ cho các biến liên tục

5.1 Các khái niệm

1. Tập rõ và hàm đặc trưng

- Ngôn ngữ tự nhiên và logic mờ.
- Tập rõ (crisp set): Gọi A là một tập hợp rõ, một phần tử x có thể có $x \in A$ hoặc $x \notin A$, Có thể sử dụng hàm μ_A để mô tả khái niệm thuộc về. Nếu $x \in A$, $\mu_A(x) = 1$, ngược lại nếu $x \notin A$, $\mu_A(x) = 0$. Hàm μ_A được gọi là hàm đặc trưng của tập hợp A.
- Tập mờ và hàm thành viên: Khác với tập rõ, khái niệm thuộc về được mở rộng nhằm phản ánh mức độ x là phần tử của tập mờ A. Một tập mờ (fuzzy set): A được đặc trưng bằng hàm thành viên μ_A và cho x là một phần tử, $\mu_A(x)$ phản ánh mức độ x thuộc về A.
- Một tập mờ A trong tập vũ trụ U được xác định bởi hàm: $\mu_A: U \rightarrow [0, 1]$

5.1 Các khái niệm về Logic mờ

- Ví dụ về tập mờ:
 - High
 - Young
 - Số gần 7
 - Tốc độ nhanh

Biểu diễn tập mờ:

1. Nếu tập vũ trụ U là rời rạc và hữu hạn thì tập mờ A trong U được biểu diễn:

$$A = \frac{\mu_A(x_1) + \mu_A(x_2) + \dots + \mu_A(x_n)}{x_1 + x_2 + \dots + x_n}$$

Ví dụ: Cho $U = \{a, b, c, d\}$. Ta có thể xác định một tập mờ A như sau:

$$A = \frac{0.3}{a} + \frac{0.5}{b} + \frac{0}{c} + \frac{0.7}{d} \quad \text{Hoặc } A = \{0.3, 0.5, 0.0, 0.7\}$$

5.1 Các khái niệm về Logic mờ (tiếp)

2. Nếu tập vũ trụ U là liên tục thì tập mờ A trong U được biểu diễn:

$$A = \int_x \mu_A(x) dx$$

Ví dụ: Tập mờ $A = \{\text{"Số gần 2"}\}$ có thể xác định hàm thuộc như sau:

$$\mu_A(x) = e^{-(x-2)^2}$$

$$A = \int_x \frac{e^{-(x-2)^2}}{x} dx$$

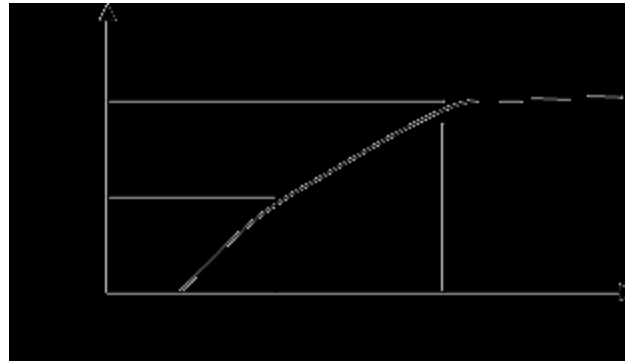
5.1 Các khái niệm về Logic mờ(tiếp)

Ghi chú: Đồ thị hàm thuộc cho tập mờ $A=\{\text{“Số gần 2”}\}$ có thể xác định cách khác như sau:

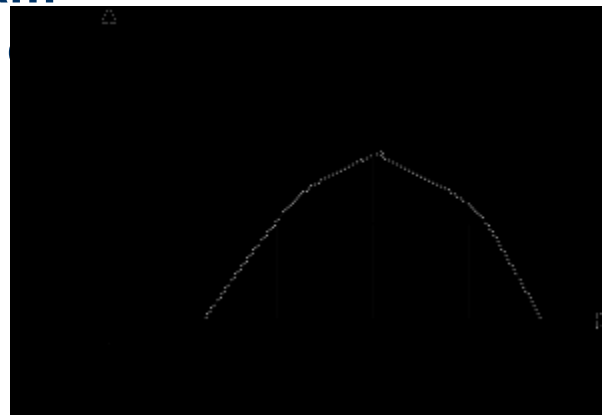
$$\mu_A(x) = \begin{cases} 0 & \text{nếu } x < 1 \\ x-1 & \text{nếu } 1 \leq x < 2 \\ 1 & \text{nếu } x = 2 \\ -x+3 & \text{nếu } 2 < x \leq 3 \\ 0 & \text{nếu } x > 3 \end{cases}$$

5.2 Các dạng của hàm thành viên

- Các hàm thành viên của tập mờ có 3 dạng cơ bản là: dạng tăng, dạng giảm và dạng chuông
- Dạng S tăng



- b) Dạng S giảm
- c) Dạng hình



Hình 4.3. Hàm dạng chuông

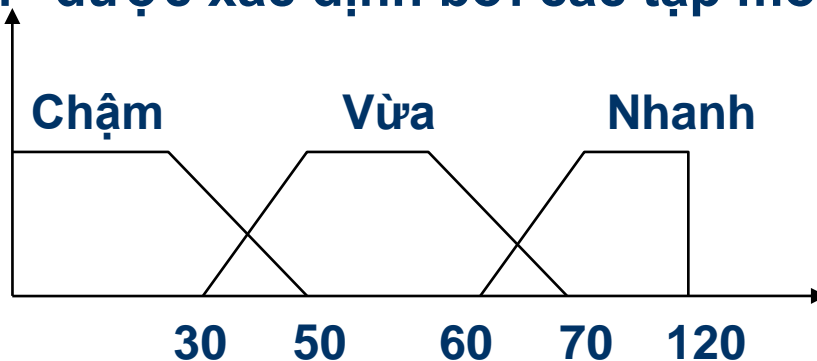
5.3 Biến ngôn ngữ (Linguistic Variable)

- Logic mờ liên quan đến lập luận trên các thuật ngữ mờ và mờ hồ trong ngôn ngữ tự nhiên của con người.
- Biến nhận các từ trong ngôn ngữ tự nhiên làm giá trị gọi là biến ngôn ngữ.
- Biến ngôn ngữ được xác định bởi bộ bốn (x, T, U, M) :
 - X là tên biến. Ví dụ: “nhiệt độ”, “tốc độ”, “áp suất”, ...
 - T là tập các từ (các giá trị ngôn ngữ) mà x có thể nhận. Ví dụ: x là “tốc độ” thì T có thể là $T = \{\text{chậm, vừa, nhanh}\}$
 - U là miền giá trị mà x có thể nhận. Ví dụ, nếu x là “tốc độ” của xe máy thì $U = [0 .. 120 \text{ km/h}]$
 - M là luật ngữ nghĩa, ứng với mỗi từ $t \in T$ với một tập mờ A .

5.4 Biến ngôn ngữ (tiếp)

Biến ngôn ngữ	Các giá trị điển hình
Nhiệt độ Độ cao Tốc độ	Nóng, lạnh Thấp, trung bình, cao Chậm, vừa, nhanh

Ví dụ: Cho x là tốc độ, $T=\{\text{chậm, vừa, nhanh}\}$, các từ “chậm”, “vừa”, “nhanh” được xác định bởi các tập mờ trong hình sau:



5.5 Gia tử

- Gia tử làm mờ hồ thêm các câu như: **rất, hơi, có vẻ, ...**

1. **Rất:** $\mu_{\text{rất}(A)}(x) = (\mu(x))^2$

Ví dụ: Tập mờ gồm những người rất cao

2. **Co giãn / một ít:** $\mu_{\text{co giãn}(A)}(x) = (\mu(x))^{0.5}$

Ví dụ: A là tập những người tầm thước thì Co giãn A là tập những người thiên về cao và thấp trong những người tầm thước.

3. **Nhấn mạnh/ thực sự là:**

$$\mu_{\text{nhấn mạnh}(A)}(x) = 2(\mu(x))^2 \quad \text{nếu } 0 \leq \mu(x) \leq 0.5$$

$$\mu_{\text{nhấn mạnh}(A)}(x) = 1 - 2(1 - \mu(x))^2 \quad \text{nếu } 0.5 \leq \mu(x) \leq 1$$

Ví dụ: Sau khi dùng phép toán này với tập mờ cao ta được tập những người thực sự cao

4. **Mạnh mẽ / rất rất:** $\mu_{\text{rất rất}(A)}(x) = (\mu(x))^n$

Ví dụ: Sau khi dùng phép toán này với tập mờ cao ta được tập những người thực sự cao

5.6 Các phép toán trên tập mờ

Cho ba tập mờ A, B, C với $\mu_A(x)$, $\mu_B(x)$, $\mu_C(x)$

- $C = A \cap B$: $\mu_C(x) = \min(\mu_A(x), \mu_B(x))$
- $C = A \cup B$: $\mu_C(x) = \max(\mu_A(x), \mu_B(x))$
- $C = \bar{A}$: $\mu_C(x) = 1 - \mu_A(x)$

Xét tập mờ “cao” và “thấp” về chiều cao của người:

$$\text{Cao} = 0.0/1.5 + 0.2/1.55 + 0.5/1.60 + 0.8/1.65 + 1.0/1.70$$

$$\text{Thấp} = 1.0/1.5 + 0.8/1.55 + 0.5/1.60 + 0.2/1.65 + 0.0/1.70$$

- $\mu_{\text{Cao} \cap \text{Thấp}}(x) = 0.0/1.5 + 0.2/1.55 + 0.5/1.60 + 0.2/1.65 + 0.0/1.70$
Dùng để chỉ những người tầm thước: giá trị cao nhất ở giữa tập, thấp nhất ở 2 bên
- $\mu_{\text{Cao} \cup \text{Thấp}}(x) = 1.0/1.5 + 0.8/1.55 + 0.5/1.60 + 0.8/1.65 + 1.0/1.70$
Dùng để chỉ những người không tầm thước
- $\mu_{\bar{\text{Cao}}}(x) = 1.0/1.5 + 0.8/1.55 + 0.5/1.60 + 0.2/1.65 + 0.0/1.70$
Dùng để chỉ những người không cao hay tầm thước hay thấp.

5.7 Suy diễn mờ

Mệnh đề mờ là mệnh đề khẳng định giá trị cho biến ngôn ngữ.

<Miền xác định X> is <tập mờ A>

Ví dụ: Chiều cao là tầm thước

- Logic mờ sử dụng tập mờ trong các mệnh đề mờ.

IF X is A THEN Y is B

Ví dụ: Nếu chiều cao là tầm thước thì trọng lượng là trung bình

- Nếu A và B là tập mờ thì Hệ chuyên gia lưu trữ liên kết (A,B) trong ma trận M (hay ký hiệu R).
- Có thể thể hiện cả A và B như các vector (A, B) thích hợp và đặt quan hệ này vào ma trận M.
 - Ma trận liên kết mờ M ánh xạ tập mờ A sang tập mờ B.
- Hai kỹ thuật suy diễn thông dụng là;
 - Suy diễn Max-Min
 - Suy diễn cực đại.

5.7.1 Nhân ma trận vector mờ

Cho $A = (a_1, a_2, \dots, a_n)$ với $a_i = \mu_{A_i}(x_i)$

Cho $B = (b_1, b_2, \dots, b_p)$ với $b_j = \mu_{B_j}(y_j)$

Ma trận $M_{n \times p}$ được xác định như sau: $A \circ M = B$

Trong đó $b_j = \max \{ \min(a_i, m_{ij}) \mid 1 \leq i \leq n \}$

Ví dụ: Cho $A = (0.2, 0.4, 0.6, 1)$

Cho

$$M = \begin{bmatrix} 0.1 & 0.6 & 0.8 \\ 0.6 & 0.8 & 0.6 \\ 0.8 & 0.6 & 0.5 \\ 0.0 & 0.5 & 0.5 \end{bmatrix}$$

$$b_1 = \max\{\min(0.2, 0.1), \min(0.4, 0.6), \min(0.6, 0.8), \min(1, 0.0)\} = 0.6$$

$$b_2 = \max\{\min(0.2, 0.6), \min(0.4, 0.8), \min(0.6, 0.6), \min(1, 0.5)\} = 0.6$$

Tương tự $b_2 = 0.6$ và $b_3 = 0.5$

Như vậy, $B = (0.6, 0.6, 0.5)$

5.7.2 Suy diễn Max-Min (Tìm ma trận M)

- Suy diễn Max-Min: $m_{ij} = \text{Min}(a_i, b_j)$

Ví dụ: X: nhiệt độ, tập mờ A trên X: “nhiệt độ bình thường”

Y: tốc độ, tập mờ B trên Y: “tốc độ vừa phải”

Giả sử có luật: IF nhiệt độ bình thường THEN tốc độ vừa phải”.

Giả sử các tập mờ thể hiện bằng các vector sau:

Nhiệt độ bình thường = (0/100, 0.5/125, 1/150, 0.5/175, 0/200)

Hoặc (0, 0.5, 1, 0.5, 0)

Tốc độ vừa phải = (0/10, 0.6/20, 1/30, 0.6/40, 0/50)

Hoặc (0, 0.6, 1, 0.6, 0)

Ma trận M được tạo ra như sau:

$$M = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.5 & 0.5 & 0.0 \\ 0.0 & 0.6 & 1.0 & 0.6 & 0.0 \\ 0.0 & 0.5 & 0.5 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Cho tập A' = (0/100, 0.5/125, 0/150, 0/175, 0/200)

Ta có $b_1 = \max\{\min(0,0), \min(0.5,0), \min(0,0), \min(0,0), \min(0,0)\}=0$

B' = (0/10, 0.5/20, 0.5/30, 0.5/40, 0/50)

5.7.3 Suy diễn Tích cực đại (Tìm ma trận M)

- Dùng phép nhân để tạo các thành phần của ma trận M

Suy diễn Tích cực đại: $m_{ij} = a_i * b_j$

- Sau đó dùng cách tính Max-Min để suy ra B' từ A'

Ví dụ: $A = (0, 0.5, 1, 0.5, 0)$

$B = (0, 0.6, 1, 0.6, 0)$

Ma trận M được tạo ra như sau:

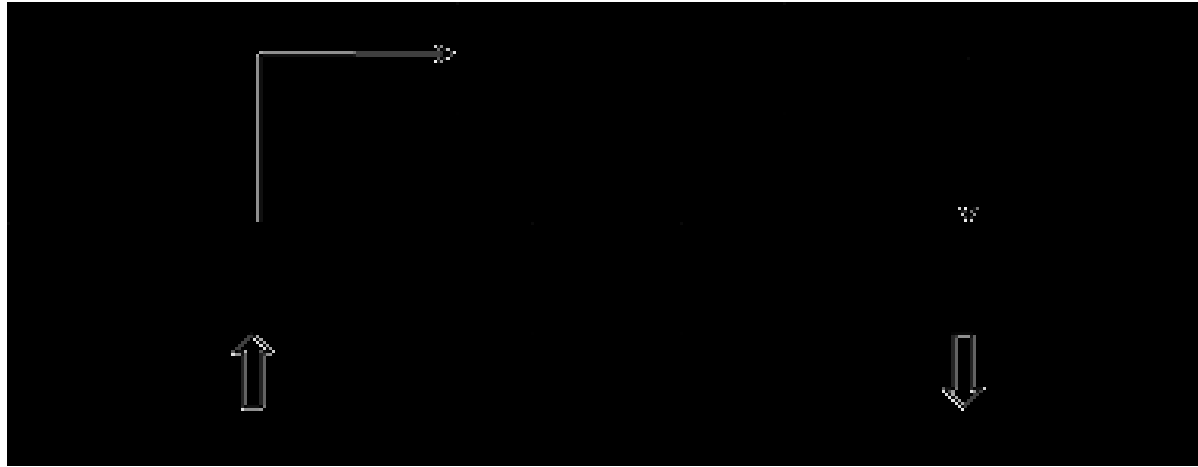
$$M = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.5 & 0.3 & 0.0 \\ 0.0 & 0.6 & 1.0 & 0.6 & 0.0 \\ 0.0 & 0.3 & 0.5 & 0.3 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Cho tập $A' = (0, 0.5, 0, 0, 0)$

Ta có $b_1 = \max\{\min(0,0), \min(0.5,0), \min(0,0), \min(0,0), \min(0,0)\}=0$

$B' = (0, 0.3, 0.5, 0.3, 0)$

5.8 NGUYÊN LÝ XỬ LÝ CÁC BÀI TOÁN MỜ

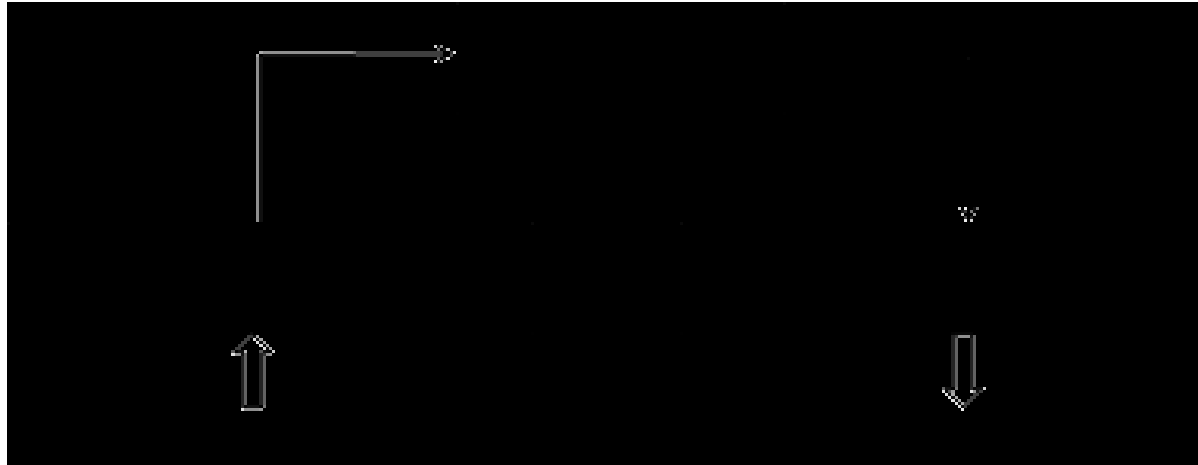


Hình 9.4. Hệ thống mờ

1. Mờ hóa: Chuyển đổi giá trị rõ đầu vào thành vector
2. Xác định các luật hợp thành và thuật toán xác định giá trị mờ
3. Giải mờ: Phương pháp điểm trọng tâm

$$x_0 = \frac{\int_0^x f(x) dx}{\int_0^x g(x) dx}$$

5.8 NGUYÊN LÝ XỬ LÝ BÀI TOÁN MỜ(tiếp)



Hình 9.4. Hệ thống mờ

Bài toán 1: Dữ liệu Input là các giá trị rõ.

Ví dụ: Xét bài toán mờ xác định bởi các luật sau:

Luật 1: if x is A1 and y is B1 Then z is C1

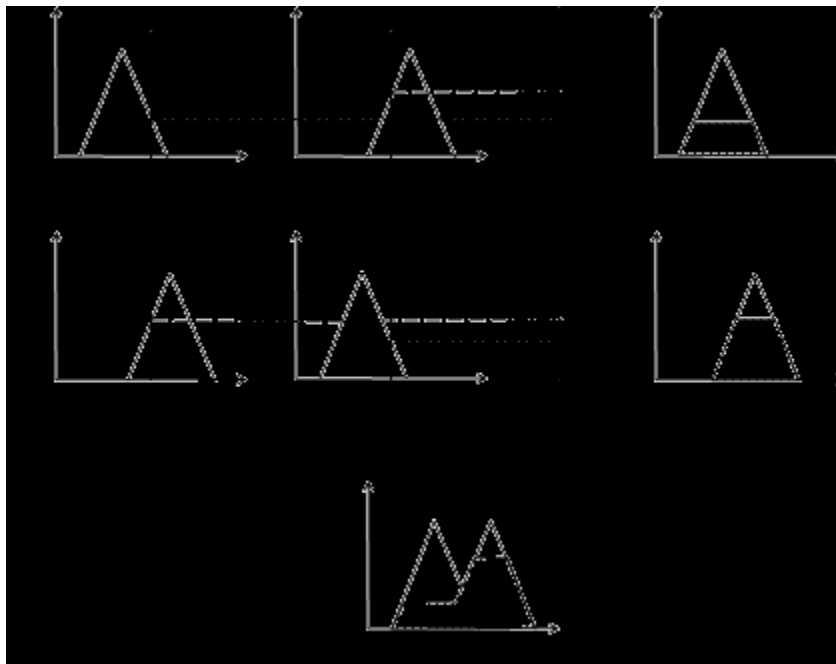
Luật 2: if x is A2 or y is B2 Then z is C2

Vào: trị x_0, y_0

Ra : trị z_0 tương ứng

5.8 NGUYÊN LÝ XỬ LÝ BÀI TOÁN MỜ (tiếp)

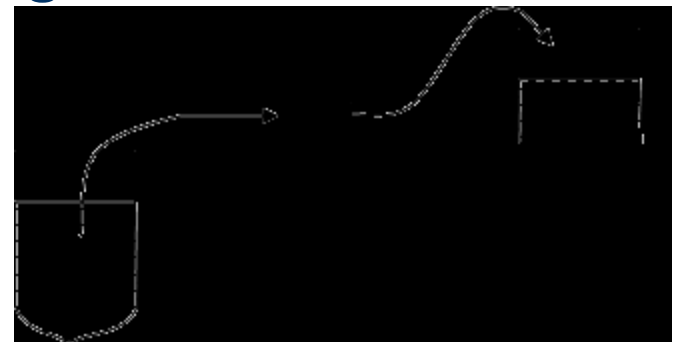
Ứng với tập mờ A_1 ta có hàm thành viên $\mu_{A_1}(x)$
Ứng với tập mờ A_2 ta có hàm thành viên $\mu_{A_2}(x)$
Ứng với tập mờ B_1 ta có hàm thành viên $\mu_{B_1}(y)$
Ứng với tập mờ B_2 ta có hàm thành viên $\mu_{B_2}(x)$
Ứng với tập mờ C_1 ta có hàm thành viên $\mu_{C_1}(x)$
Ứng với tập mờ C_2 ta có hàm thành viên $\mu_{C_2}(x)$



Ví dụ: Giải bài toán điều khiển tự động mờ cho hệ thống bơm nước lấy nước từ giếng.

Trong khi hồ hết nước và trong giếng có nước thì máy bơm tự động bơm.

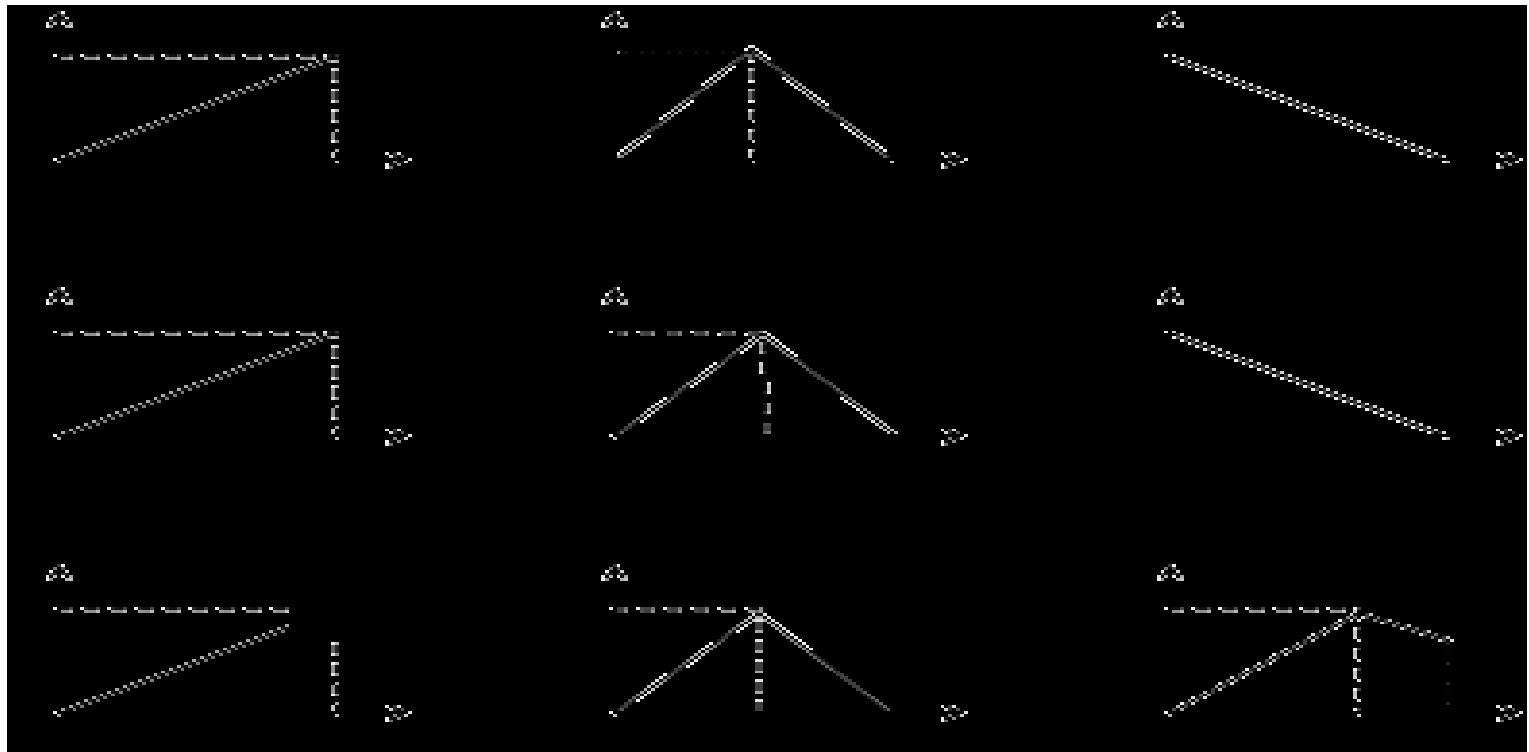
	H.Đầy	H.Lượng	H.Cạn
N.Cao	0	B.Vừa	B.Lâu
N.Vừa	0	B.Vừa	B.HơiLâu
N.Ít	0	0	0



- Với biến ngôn ngữ Hồ có các tập mờ hồ đầy (H.Đầy), hồ lượng (H.Lượng) và hồ cạn (H.Cạn).
- Với biến ngôn ngữ Giếng có các tập mờ nước cao (N.Cao), nước vừa (N.Vừa), nước ít (N.Ít).
- Với biến ngôn ngữ kết luận xác định thời gian bơm sẽ có các tập mờ bơm vừa (B.Vừa), bơm lâu (B.Lâu), bơm hơi lâu (B.HơiLâu).

Ví dụ (tiếp)

Trong đó x chỉ độ sâu của Hồ ($0 \leq x \leq 2$), y chỉ độ sâu của Giếng ($0 \leq y \leq 10$) và z chỉ thời gian bơm ($0 \leq z \leq 30$).



Ví dụ (tiếp)

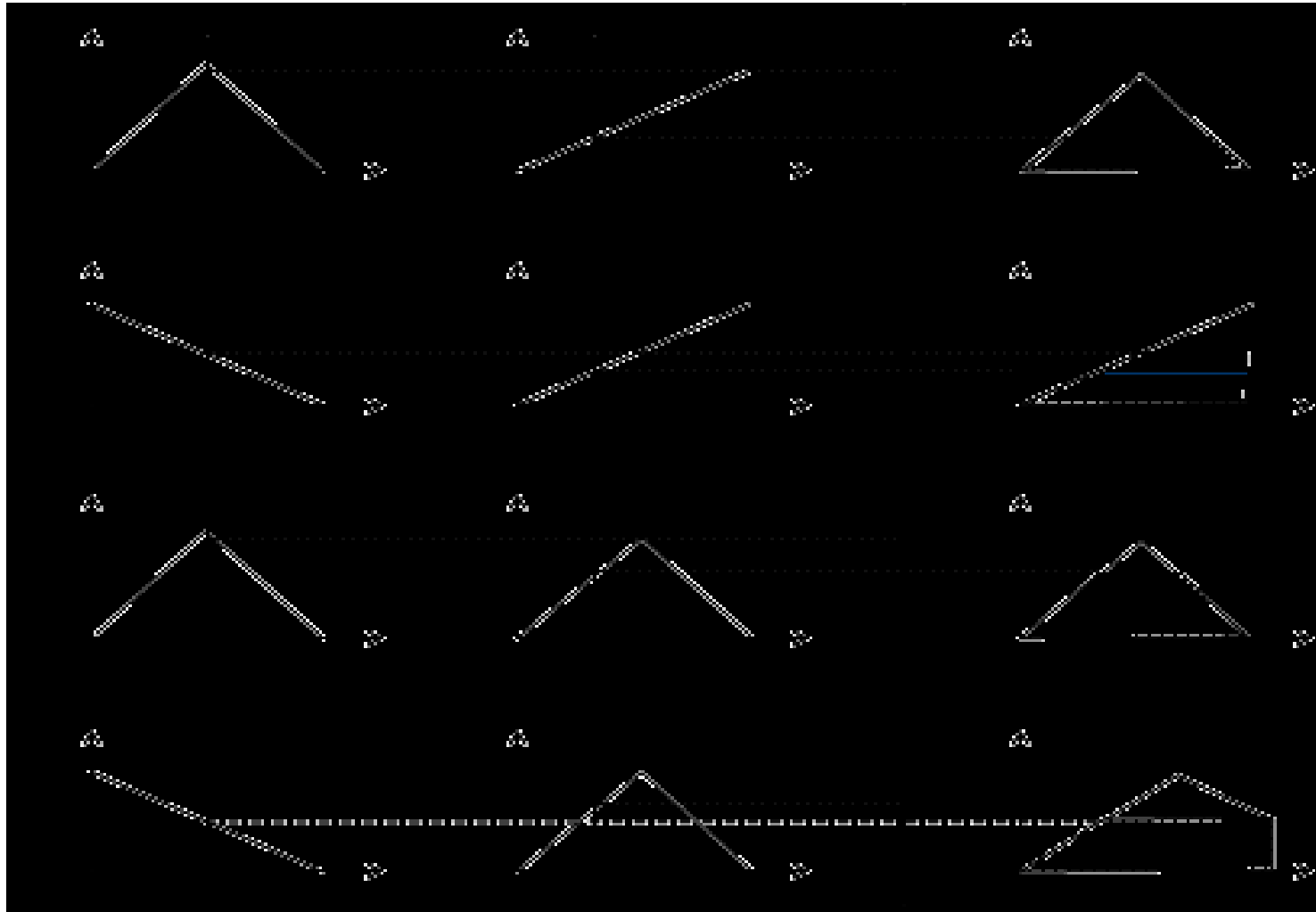
Từ bảng trên ta có các luật:

- **Luật 1:** if *x* is H.Lượng and *y* is N.Cao Then *z* is B.Vừa
- **Luật 2:** if *x* is H.Cạn and *y* is N.Cao Then *z* is B.Lâu
- **Luật 3:** if *x* is H.Lượng and *y* is N.Vừa Then *z* is B.Vừa
- **Luật 4:** if *x* is H.Cạn and *y* is N.Vừa Then *z* is B.Hơi lâu

Nếu nhập trị Input $x_0 = 1$ (Độ cao của nước trong hồ),
 $y_0 = 3$ (Độ cao của nước trong giếng)

$$\begin{array}{l} \mu_{H.Lượng}(x_0) = 1 \\ \mu_{N.Cao}(y_0) = 3/10 \end{array} \left. \vphantom{\begin{array}{l} \mu_{H.Lượng}(x_0) = 1 \\ \mu_{N.Cao}(y_0) = 3/10 \end{array}} \right\} \Rightarrow W_1 = \min(1, 3/10)$$
$$\begin{array}{l} \mu_{H.Cạn}(x_0) = 0.5 \\ \mu_{N.Vừa}(y_0) = 3/5 \end{array} \left. \vphantom{\begin{array}{l} \mu_{H.Cạn}(x_0) = 0.5 \\ \mu_{N.Vừa}(y_0) = 3/5 \end{array}} \right\} \Rightarrow W_2 = \min(0.5, 3/5) = 0.5$$
$$\begin{array}{l} \mu_{H.Lượng}(x_0) = 1 \\ \mu_{N.Vừa}(y_0) = 3/5 \end{array} \left. \vphantom{\begin{array}{l} \mu_{H.Lượng}(x_0) = 1 \\ \mu_{N.Vừa}(y_0) = 3/5 \end{array}} \right\} \Rightarrow W_3 = \min(1, 3/5) = 3/5$$
$$\begin{array}{l} \mu_{H.Cạn}(x_0) = 0.5 \\ \mu_{N.Vừa}(y_0) = 3/5 \end{array} \left. \vphantom{\begin{array}{l} \mu_{H.Cạn}(x_0) = 0.5 \\ \mu_{N.Vừa}(y_0) = 3/5 \end{array}} \right\} \Rightarrow W_4 = \min(0.5, 3/5) = 0.5$$

Ví dụ (tiếp)



Ví dụ (tiếp)

Các W_i gọi là các trọng số của luật thứ i

Theo lý thuyết hàm thành viên của kết luận cho bởi công thức:

$$\mu_c(z) = \sum_{i=1}^N W_i \mu_{1i}(z) \quad i = 1 \dots N$$

$$\mu_c(z) = W_1.B.Vừa(z) + W_2.B.Lâu(z) + W_3.B.Vừa(z) + W_4.B.Hơi Lâu(z)$$

$$\mu_c(z) = 3/10.B.Vừa(z) + 0.5.B.Lâu(z) + 3/5.B.Vừa(z) + 0.5.B.HơiLâu(z)$$

Bước tiếp theo là ta phải giải mờ từ hàm thành viên của kết luận bằng cách tính trọng tâm của hàm $\mu_c(z)$

Moment $\mu_c(z)$ là $\int_U z \mu_c(z) dz$

và $\int_U \mu_c(z) dz$

$$z_0 = \frac{\int_U z \mu_c(z) dz}{\int_U \mu_c(z) dz} = 8.15$$

Vậy Defuzzy(z) = 8.15

Do đó nếu mực nước trong hồ và giếng là 1m và 3m thì thời gian cần bơm là 8 phút và 15/100 giây.

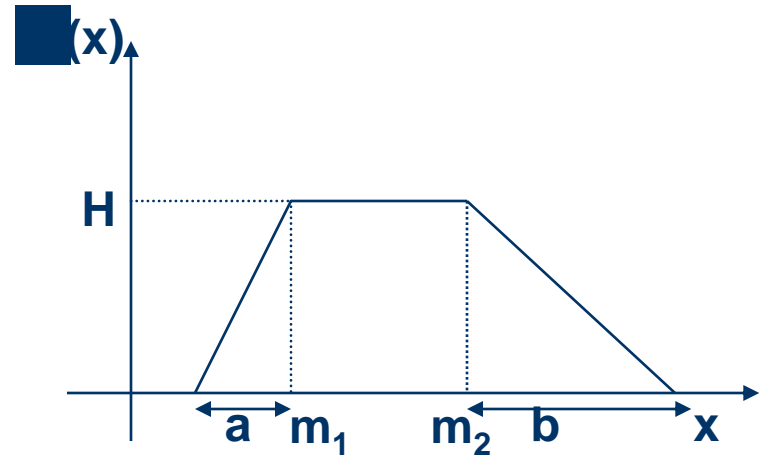
Khử mờ: Lấy điểm trọng tâm

$$\begin{array}{c}
 \int_U x \rho(x) dx \\
 \hline
 \int_U \rho(x) dx
 \end{array}
 \quad
 \begin{array}{c}
 \int_{U_i} x \rho_i(x) dx \\
 \hline
 \int_{U_i} \rho_i(x) dx
 \end{array}$$

$$\begin{array}{c}
 \int_{U_i} x \rho_i(x) dx \\
 \hline
 \int_{U_i} \rho_i(x) dx
 \end{array}$$

Tính các tích phân

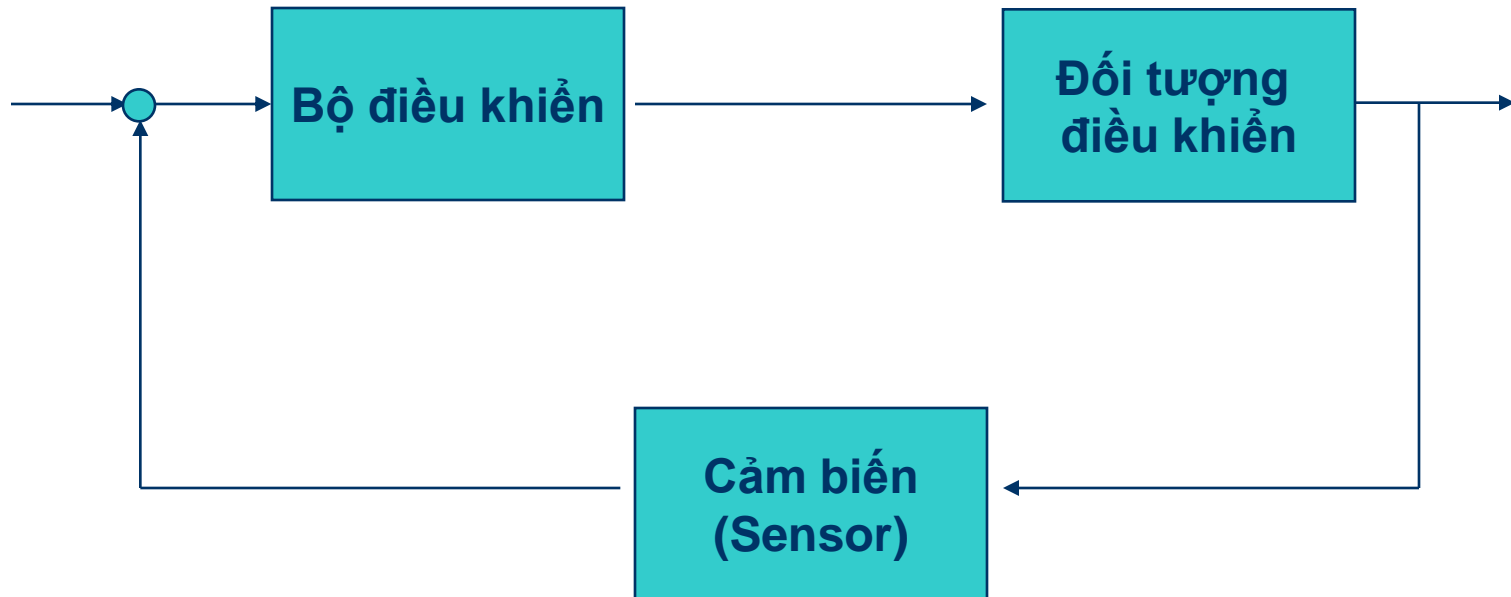
$$\int_{x_0}^x f(x) dx - \int_{x_0}^u f(x) dx$$



$$\int_{x_0}^x f(x) dx = \frac{H}{6} [m_2^2 - 3m_1^2 x - a^2 m_2 b - m_1 a]$$

$$\int_{x_0}^u f(x) dx = \frac{H}{2} [2m_2 - 2m_1 x]$$

5.8 NGUYÊN LÝ XỬ LÝ CÁC BÀI TOÁN MỜ (tiếp)



Bộ điều khiển mờ:

SISO: Single Input, Single Output

MIMO: Multi Input, Multi Output

SIMO: Single Input, Multi Output

MISO: Multi Input, Single Output

5.8 NGUYÊN LÝ XỬ LÝ CÁC BÀI TOÁN MỜ (tiếp)

Bộ điều khiển mờ:

SISO: If (A is A_1) Then (B is B_1)

...

If (A is A_n) Then (B is B_n)

MIMO: If (A_1 is A_{11}) and ... and (A_m is A_{m1}) Then (B_1 is B_{11}) and ... and (B_s is B_{1s})

...

If (A_1 is A_{n1}) and ... and (A_m is A_{n1}) Then (B_1 is B_{n1}) and ... and (B_s is B_{ns})

MISO: If (A_1 is A_{11}) and ... and (A_m is A_{m1}) Then (B is B_1)

...

If (A_1 is A_{n1}) and ... and (A_m is A_{n1}) Then (B is B_n)

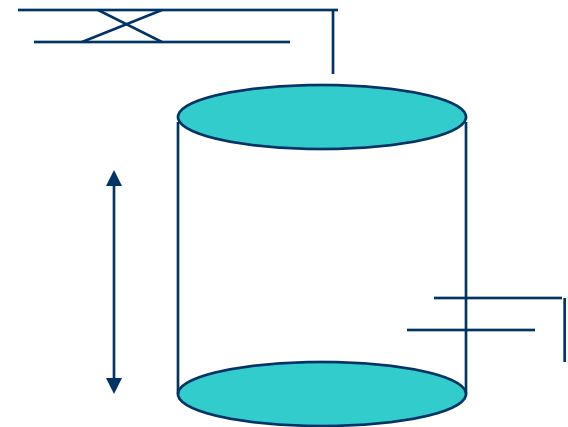
SIMO: If (A is A_1) Then (B_1 is B_{11}) and ... and (B_s is B_{1s})

...

If (A is A_n) Then (B_1 is B_{n1}) and ... and (B_s is B_{ns})

Bài tập: Điều khiển mực nước

Bài toán điều khiển mực nước: Không phụ thuộc vào lượng nước chảy ra khỏi bình, cần phải điều chỉnh van cho lượng nước chảy vào bình vừa đủ để sao cho mực nước h trong bình là luôn không đổi.



Giả sử bộ điều khiển là con người, sẽ có các nguyên tắc sau:

R1: Nếu mực nước là thấp nhiều thì van ở mức độ mở to

R2: Nếu mực nước là thấp ít thì van ở mức độ mở nhỏ

R3: Nếu mực nước là đủ thì van ở vị trí đóng.

R4: Nếu mực nước là cao thì van ở vị trí đóng.

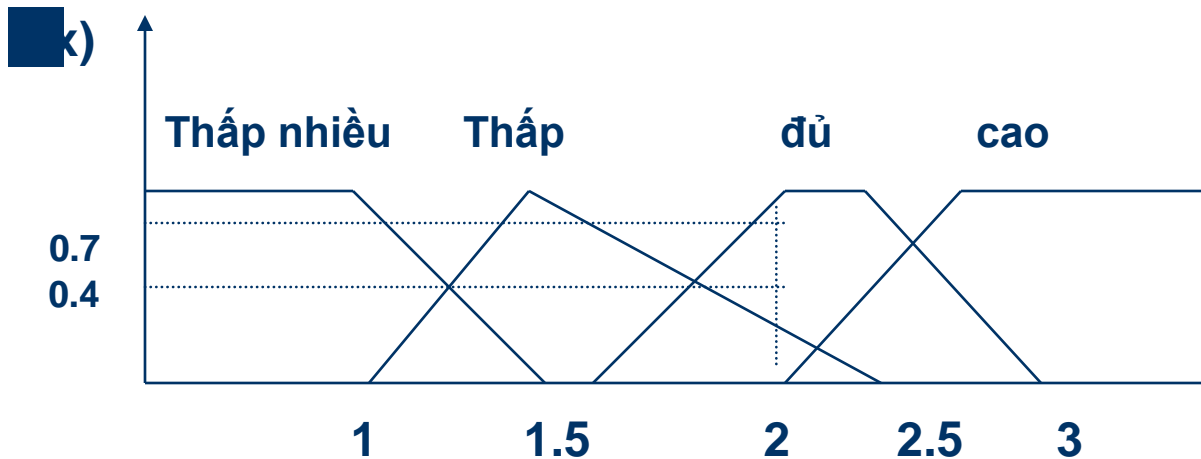
Bài tập: Điều khiển mực nước(tiếp)

Các biến ngôn ngữ:

+ x là mực nước: có 4 giá trị T={thấp nhiều, thấp ít, đủ, cao}

+ y van: có 3 giá trị T={to, nhỏ, đóng}

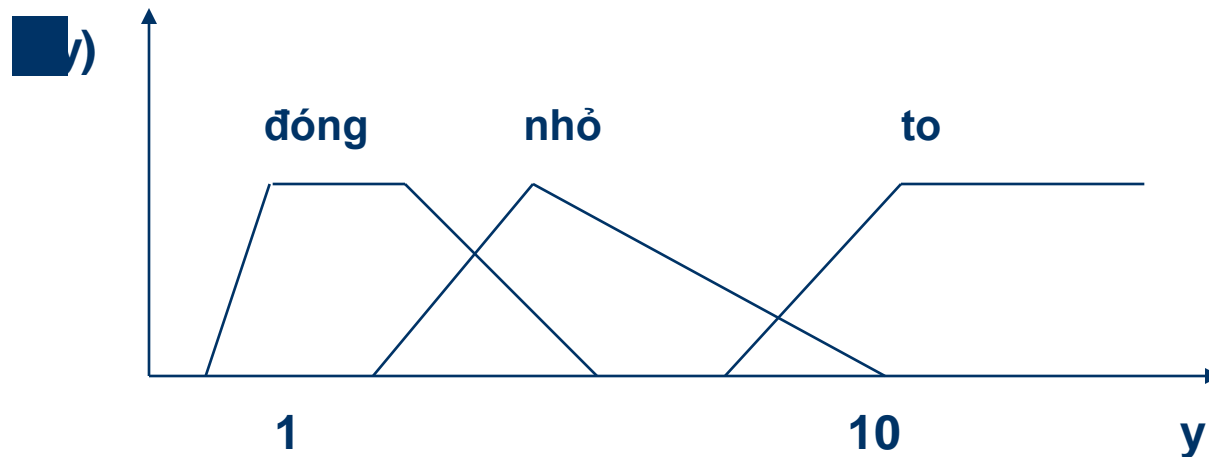
Mực nước X $\in [0 .. 3]$ (mét)



Bài tập: Điều khiển mực nước(tiếp)

Van Y $\in [0, 10]$ (cm)

Với y là độ mở của van, $T = \{\text{to, nhỏ, đóng}\}$



Cho mực nước cao $x_0 = 2$ m, hỏi độ mở của van y_0 là bao nhiêu

Chương 6: Công cụ tạo lập Hệ CSTT

Hệ CSTT gồm 3 thành phần sau:

1. Bộ tạo lập cơ sở tri thức: bảo đảm các tính chất:

- ❖ Nhất quán
- ❖ Không mâu thuẫn
- ❖ Không dư thừa dữ liệu
- ❖ Cập nhật dễ dàng

2. Bộ tạo động cơ suy diễn

- ❖ Với mỗi cách biểu diễn tri thức cần có chiến lược suy diễn tương ứng
- ❖ Tạo thuận tiện cho bộ phận giải thích

3. Bộ tạo các giao diện:

- ❖ Cung cấp khả năng tạo giao diện với người sử dụng.

Công cụ tạo lập Hệ CSTT

6.1 Soạn thảo tri thức

Các chức năng cơ bản của hệ soạn thảo tri thức:

- Cho phép soạn thảo các sự kiện, các luật.
- Đưa ra phạm vi tri thức cần thu nạp.
- Lên danh sách các giá trị cần thiết cho mỗi đối tượng.
- Kiểm tra sự đúng đắn của các luật đưa vào.
- Kiểm tra sự trùng lặp khi bổ sung tri thức.
- Cho phép sửa đổi, huỷ bỏ tri thức.
- Hệ soạn thảo tri thức làm việc trên cơ sở ngôn ngữ gốc:
 - Ngôn ngữ tự nhiên.
 - Ngôn ngữ hình thức.

6.1 Soạn thảo tri thức (tiếp)

1. Soạn thảo các luật bằng ngôn ngữ tự nhiên với cú pháp hạn chế.

Ví dụ:

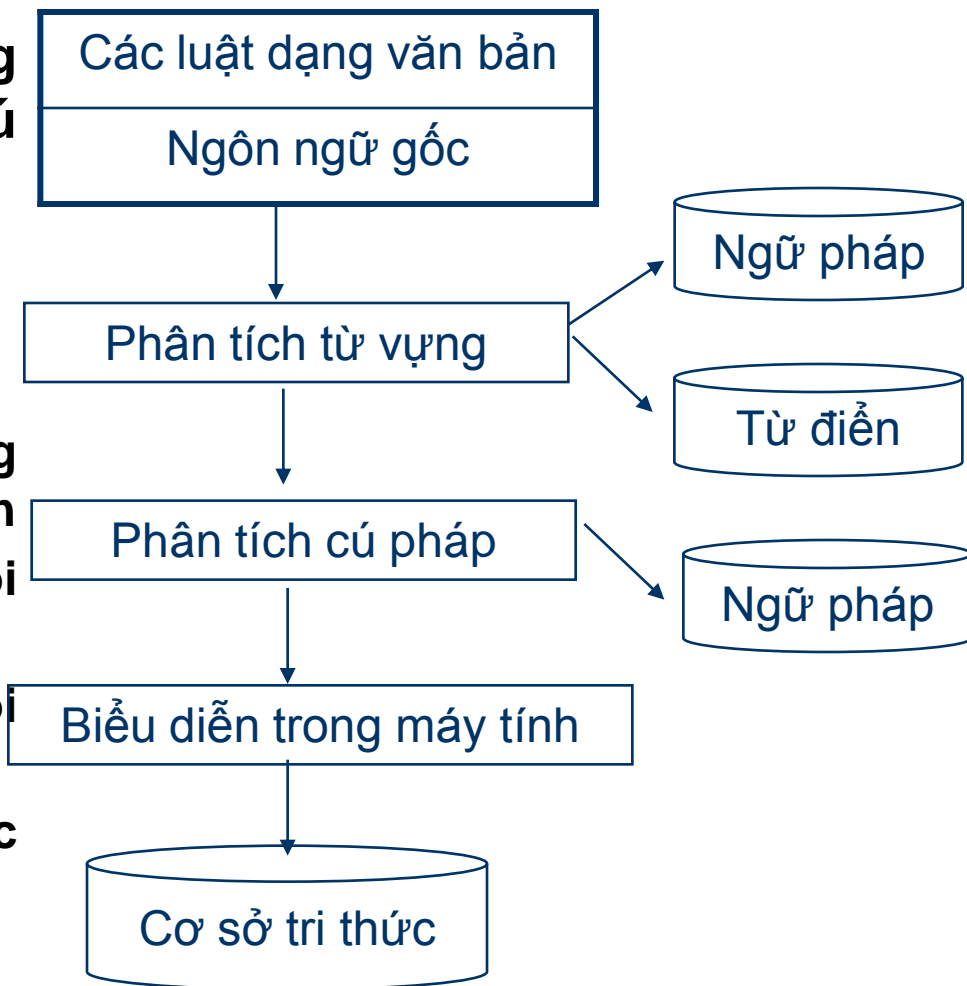
- IF chất lỏng có độ PH < 6
THEN đó là Axit

- IF tam giác có 2 góc bằng nhau
THEN đó là tam giác cân

Phân tích từ vựng: Kiểm tra lỗi chính tả

Phân tích cú pháp: Kiểm tra lỗi câu, đặt mệnh đề.

Biểu diễn trong máy tính: Gồm các sự kiện, các luật.



Bộ soạn thảo Cơ sở tri thức

6.1 Soạn thảo tri thức(tiếp)

2. Soạn thảo các luật theo cú pháp mệnh đề Horn

Hệ soạn thảo luật = Hệ soạn thảo văn bản + Hệ phân tích cú pháp.

Dạng tổng quát của luật:

IF <C₁> AND <C₂> AND ... AND <C_n> THEN <C>

Trong đó: C_i, C có dạng:

<Giá trị> <Phép toán> <Giá trị>

Cú pháp:

<Luật> := IF <điều kiện> THEN <đk đơn giản>

<Điều kiện> := <đk đơn giản> / <đk đơn giản> AND <đk>

<Đk đơn giản> := <danh từ> / <danh từ> <phép toán> <giá trị>

<Phép toán> := ■■■■■

Ví dụ: Soạn thảo các luật theo cú pháp mệnh đề Horn

Luật: IF tuổi 30 THEN trẻ

Biểu diễn bên trong:

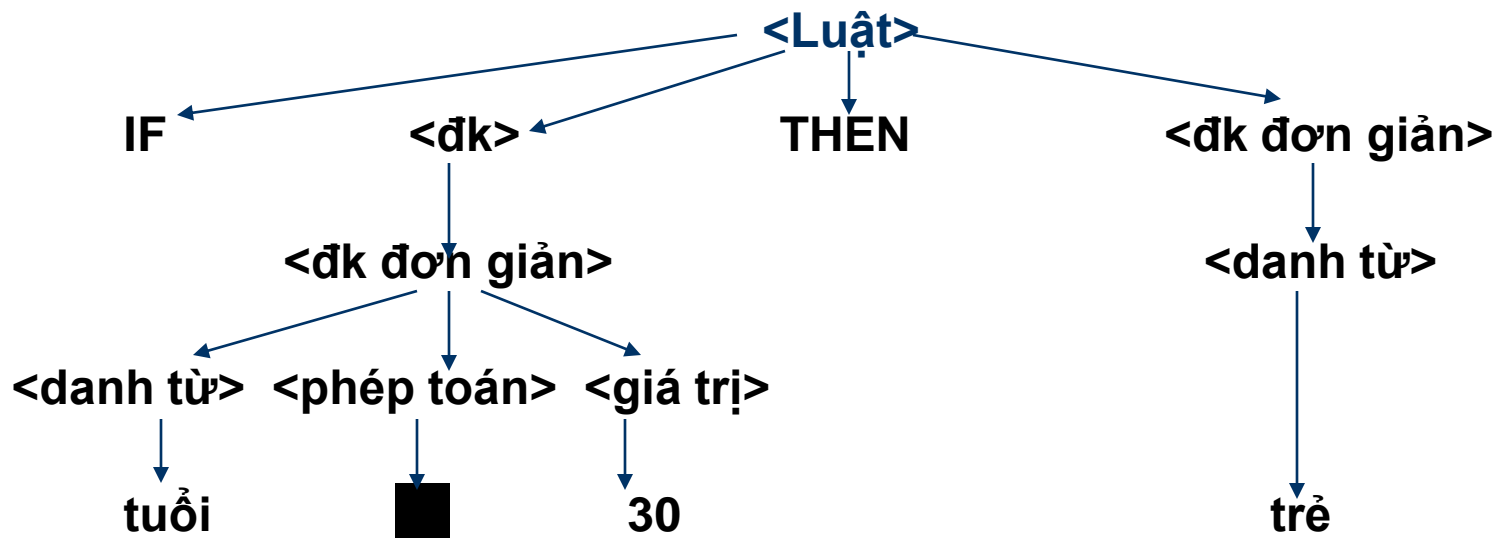
Luật

P ₁	Trẻ

Sự kiện

P ₁	Tuổi <input type="checkbox"/> 0

Biểu diễn bên ngoài:



6.1 Soạn thảo tri thức

3. Soạn thảo các luật dựa vào bộ ba liên hợp O-A-V Dạng tổng quát của luật:

IF $\langle C_1 \rangle$ AND $\langle C_2 \rangle$ AND ... AND $\langle C_n \rangle$ THEN $\langle C \rangle$

Trong đó mỗi C_i được biểu diễn dưới dạng bộ ba liên hợp Object-Attribute-Value (O-A-V)

Ngữ pháp: Như câu lệnh Horn, ngoại trừ điều kiện được thay bởi:

$\langle \text{điều kiện đơn giản} \rangle := \langle \text{tên đối tượng} \rangle \langle \text{tên thuộc tính} \rangle \langle \text{giá trị} \rangle$

Ghi chú:

Mối liên hệ giữa $\langle \text{đối tượng} \rangle \langle \text{thuộc tính} \rangle \langle \text{giá trị} \rangle$ được viết dưới dạng:

$\langle \text{đối tượng} \rangle . \langle \text{thuộc tính} \rangle = \langle \text{giá trị} \rangle$

6.2. Hệ chuyên gia dựa trên luật

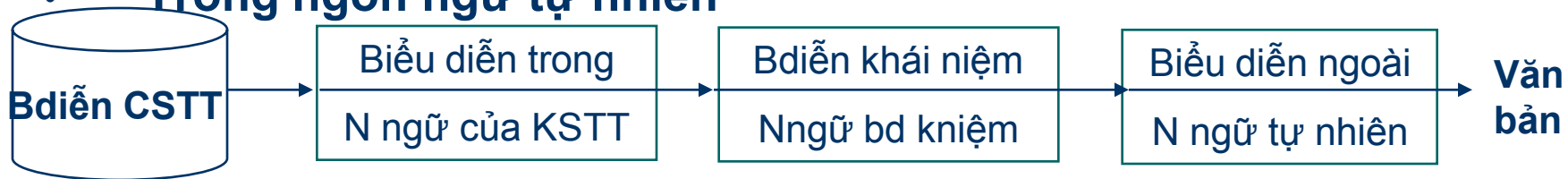
6.2.1 Hiện thị sự kiện và luật

1. Hiện thị sự kiện: Bằng ngôn ngữ tự nhiên hiển thị các dòng văn bản chỉ nội dung sự kiện

Ví dụ: Cao huyết áp
Viêm phổi cấp

2. Hiện thị các luật:

• Trong ngôn ngữ tự nhiên



• Dưới dạng đồ thị

- Biểu diễn khái niệm: p_1 ■ p_2 ■ .. ■ p_n ■

- Biểu diễn ngoài: p_1 → q
 p_2 → q
 p_n → q

6.2.2 Cập nhật tri thức

- 1. Cập nhật luật: Khi cập nhật luật cần kiểm tra các 2 điều kiện:**
 - Nếu đã tồn tại thì loại bỏ.
 - Nếu chưa tồn tại thì kiểm tra có mâu thuẫn hoặc tranh chấp với các luật sẵn có trong CSTT.
- 2. Cập nhật sự kiện: Kiểm tra cú pháp và duyệt trong CSTT, nếu trùng thì loại bỏ.**

6.2.2 Cập nhật tri thức(tiếp)

Tính mâu thuẫn trong cơ sở tri thức:

1. Sự dư thừa các luật, dư thừa sự kiện
2. Mâu thuẫn trong cơ sở tri thức
3. Sự nhất quán của các luật

1. Sự dư thừa các luật, dư thừa sự kiện

Cho tập $SC=(SF,SR)$

Trong đó: SC: cơ sở tri thức

SF: tập các sự kiện

$SR = \{r_1, r_2, \dots, r_m\}$: tập luật

r_i : left

Đn 5.1: Ta nói tập SR suy diễn luật r : left ký hiệu $SR \models r$, nếu thỏa mãn tất cả các luật trong SR thì cũng thỏa r .

Ví dụ: $SF=\{a, b, c\}$, $SR=\{r_1: a \quad r_2: b \quad \models \{a \quad$

Đn 5.2: Bao đóng của tập luật: $SR^+=\{r \mid SR \models r\}$

Đn 5.3: Bao đóng của tập sự kiện X là

$X^+=\{A \mid X \quad SR^+\}$

Đn 5.4: Hai tập luật SR_1 và SR_2 là tương đương nếu:

$SR_1^+ = SR_2^+$

Thuật toán 6.1

Input: $SR = \{r_1, r_2, \dots, r_m\}$

$X \text{ [redacted] } F$

Output: $X^+ = \{A \mid X \text{ [redacted] } SR^+\}$

Method:

$TG = X;$

$Sat = \text{Loc}(SR, TG);$

While (Sat [redacted] do

{ $r = \text{Get}(Sat);$ // r: left [redacted]

$TG = TG \text{ [redacted] } q};$

$SR = SR \setminus \{r\};$

$Sat = \text{Loc}(SR, TG);$

}

$X^+ = TG$

Return X^+

Sự dư thừa các luật, dư thừa sự kiện (tiếp)

Ví dụ: Cho SR gồm các luật:

a ■ b ■

c ■

b ■ c ■

a ■ b ■

c ■ e ■

d ■ g ■

b ■ e ■

c ■ g ■

e ■ g ■

c ■ e ■

Ta có: $\{b, d\}^+ = \{a, b, c, d, e, g\}$

Sự dư thừa các luật, dư thừa sự kiện (tiếp)

Đn 5.6: Một cơ sở luật SR là cực tiểu nếu thỏa:

1. Mỗi luật $r \in R$ có dạng Horn:

$$r: p_1 \wedge p_2 \wedge \dots \wedge p_k \rightarrow q$$

2. Không tồn tại luật $r \in R$, sao cho: $(SR \setminus \{r\})^+ = SR^+$

3. Không tồn tại luật $r: \text{left} \rightarrow \text{right}$ và tập con $F \subseteq \text{left}$, sao cho:

$$[(SR \setminus \{r\}) \cup F]^+ = SR^+$$

Ghi chú: Điều kiện 2 tránh tình trạng dư thừa luật.

Điều kiện 3 tránh dư thừa sự kiện.

Ví dụ 1: Cho cơ sở luật:

$$SR = \{r_1: a \rightarrow b, r_2: a \rightarrow c, r_3: b \rightarrow c\}$$

Như vậy, cơ sở luật tương đương tối thiểu là:

$$SR = \{r_1', r_2, r_3\} \text{ với } r_1': a \rightarrow b \text{ hoặc } r_1': b \rightarrow c$$

2. Mâu thuẫn trong cơ sở tri thức

Đn 5.7: Các luật r và r' gọi là xung đột với nhau nếu tồn tại sự liên hệ giữa chúng như sau:

r : left \blacksquare ; r' : left' \blacksquare ; left \blacksquare left' \blacksquare hoặc left' \blacksquare left; nhưng p và q là mâu thuẫn nhau.

Ví dụ: r : IF A THEN B

r' : IF A AND C THEN NOT B

Đn 5.8: Các luật có giả thiết vô dụng: là các luật có kết luận giống nhau và tập các phần của các giả thiết tạo nên hằng đề.

Ví dụ: r : IF A AND B THEN C

r' : IF A AND NOT B THEN C

Có thể thay thế như sau:

r'' : IF A THEN C

Đn 5.9: Luật nối vòng mâu thuẫn: Là dãy các luật mà trong đó các kết quả của luật trước là một phần của giả thiết tiếp theo và trong chúng có gây mâu thuẫn giữa các sự kiện.

Ví dụ: IF A AND B THEN C

IF C AND D THEN NOT A

3. Sự nhất quán của các luật

Giải quyết mâu thuẫn trong cơ sở tri thức:

Nếu có tồn tại hai luật r và r' mâu thuẫn, cần chỉ ra luật nào sẽ loại bỏ

Nguyên tắc 1: Dựa vào trọng số của luật

Luật r' được loại bỏ nếu $w(r') < w(r)$

Ví dụ: r' : IF có tiền THEN mua xe máy

r : IF có tiền THEN mua xe ô tô

Ta có $w(r') < w(r)$

Nguyên tắc 2: Dựa vào tần suất xuất hiện

Luật r' được loại bỏ nếu $f(r') < f(r)$

Ví dụ: r' : IF nhà ở xa THEN không làm việc ban đêm

r : IF nhà ở xa AND phải tăng ca THEN có thể làm đêm

3. Sự nhất quán của các luật (tiếp)

Nguyên tắc 3: Dựa vào lĩnh vực đang xét

r': IF luật r và r' mâu thuẫn nhau

AND luật r thuộc lĩnh vực A

AND luật r' thuộc lĩnh vực B

AND đang xét lĩnh vực A

THEN loại bỏ luật r'

Nguyên tắc 4: Trường hợp tranh chấp (chung- riêng)

IF luật r và r' mâu thuẫn nhau

r' biểu thị trường hợp chung r': left ████████

r biểu thị trường hợp riêng r : left' ████████ và left ████████ ft'

THEN Không áp dụng luật r'

6.3 Mô tở suy diễn

- Suy diễn tri thức
- Động cơ suy diễn
- Suy diễn với logic mệnh đề: Suy diễn tiến, Suy diễn lùi
- Suy diễn với logic vị từ: Suy diễn tiến, Suy diễn lùi

Cơ chế suy diễn

2. Suy diễn với logic vị từ:

- Suy diễn tiến
- Suy diễn lùi

Tương tự như trong logic mệnh đề, thêm vào phép hợp giải

6.4 Cơ chế điều khiển

6.4.1 Chọn hướng suy diễn

Gọi: $f_{GT} = \#GT$

$f_{KL} = \#KL$

$f_{Trước} = \text{Max} \#\{r \mid r \text{ có thể áp dụng cho tập con của GT}\}$

$f_{Sau} = \text{Max} \#\{r \mid r \text{ có cùng KL}\}$

Ví dụ: Rule = $\{r_1:a \blacksquare r_2:a \blacksquare r_3:a \blacksquare \blacksquare \blacksquare r_4:d \blacksquare \blacksquare$
 $r_5:b \blacksquare r_6:e \blacksquare$

Hỏi: a \blacksquare

Ta có: $f_{GT} = 1, f_{KL} = 1, f_{Trước} = 1, f_{Sau} = 4$

Heuristic chọn hướng suy diễn:

Luật 1: IF $f_{Trước} > f_{Sau}$ THEN Chọn Suy diễn tiến

Luật 2: IF $f_{Trước} < f_{Sau}$ THEN Chọn Suy diễn lùi

6.4 Cơ chế điều khiển(tiếp)

Luật 3: IF ($f_{\text{Trước}} = f_{\text{sau}}$) AND ($f_{\text{GT}} < f_{\text{KL}}$) THEN Chọn Suy diễn tiến

Luật 4: IF ($f_{\text{Trước}} = f_{\text{sau}}$) AND ($f_{\text{GT}} > f_{\text{KL}}$) THEN Chọn Suy diễn lùi

Luật 5: IF ($f_{\text{Trước}} = f_{\text{sau}}$) AND ($f_{\text{GT}} = f_{\text{KL}}$) THEN

IF Người thiết kế muốn Chọn Suy diễn tiến

THEN Chọn Suy diễn tiến

ELSE Chọn Suy diễn lùi

**Ghi chú: Thay vì dùng $f_{\text{Trước}}$ và f_{sau} ta có thể sử dụng $f'_{\text{Trước}} =$
Số luật trung bình sử dụng cho các sự kiện giả thiết và
 $f'_{\text{sau}} =$ Số luật trung bình sử dụng cho các sự kiện kết luận**

6.4.2 Giải quyết vấn đề cạnh tranh

1. Vấn đề cạnh tranh trong suy diễn tiến:

$r_1: \text{left}_1$ , $r_2: \text{left}_2$ , left_1 , left_2  và $\#lọc(F,R)$ 

Vấn đề: Làm thế nào để chọn 1 luật r trong số các luật có thể áp dụng được bởi $lọc(F,R)$?

Giải pháp 1: Tổ chức các luật có thể sử dụng được thành Queue

Giải pháp 2: Tổ chức các luật có thể sử dụng được thành Stack

Giải pháp 3: Dùng Heuristic: Dùng hàm ước lượng đánh giá liên hệ giữa KL và vế phải của các luật.

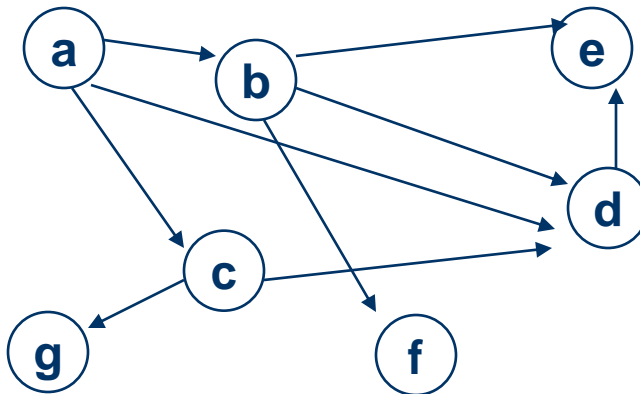
Cạnh tranh trong suy diễn tiến

Giải pháp 4: Dùng đồ thị FPG (Fact Precedence Graph): Sự kiện f_i là trước sự kiện f_j ($f_i \prec f_j$) nếu f_j là kết quả của luật r_k nào đó có chứa f_i ở vế trái, nghĩa là:

$$f_i \prec f_j \iff \exists k: f_i \in \text{Left}_k \wedge f_j \in \text{Right}_k$$

Chọn luật có chiều dài đường đi đến kết luận là ngắn

Ví dụ: Rule = { $r_1: a \rightarrow b$, $r_2: a \rightarrow c$, $r_3: a \rightarrow d$, $r_4: d \rightarrow e$, $r_5: b \rightarrow c$, $r_6: c \rightarrow f$ }



Hỏi: a \prec ta có $h(r_1, e)=1$, $h(r_2, e)=2$ Chọn r_1

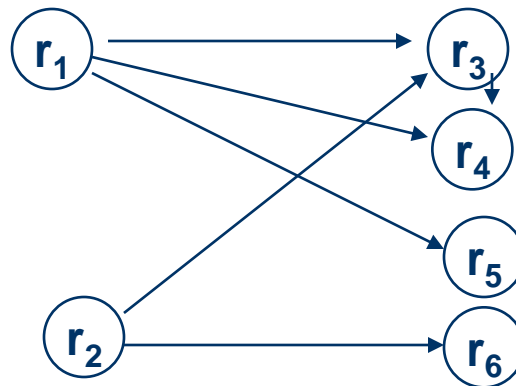
Cạnh tranh trong suy diễn tiến(tiếp)

Giải pháp 5: Dùng đồ thị RPG (Rule Precedence Graph): Luật r_i là trước luật r_j ($r_i \prec r_j$) nếu tồn tại sự kiện f sao cho:

r_i : left f và r_j : ... f ...

Hàm ước lượng chọn $h(r, r_{KL})$ là nhỏ nhất

Rule = { r_1 :a r_2 :a r_3 :a r_4 :d r_5 :b r_6 :c



Hỏi: a $h(r_1, r_4)=1, h(r_2, r_4)=2$ Chọn r_1

Cạnh tranh trong suy diễn lùi

Cạnh tranh trong suy diễn lùi xảy ra khi:

Với sự kiện f có tồn tại ít nhất 2 luật có f ở vế phải:

$r_1: \text{left}_1$ và $r_2: \text{left}_2$

Giải pháp 1: Nếu $i < j$ thì r_i được chọn.

Giải pháp 2: Dùng độ thì FPG.

Tương tự như trong suy diễn tiến

6.5 Bộ giải thích

6.5.1 Câu hỏi: Why?

Tại sao phải xác định sự tồn tại / giá trị (số, logic) của một sự kiện f nào đó.

Trả lời: Sự kiện f có lợi để xác định kết luận KL.

Kỹ thuật: Tồn tại đường đi từ f đến kết luận trong đồ thị FPG.

Cơ sở luật R  đồ thị FPG

 biểu diễn bằng ma trận $A=(a_{ij})$

6.5 Bộ giải thích (tiếp)

- **Câu hỏi: How?**

- Làm thế nào mà hệ thống lại kết luận sự kiện f là đúng?
- Trả lời: Hệ thống đã kết luận sự kiện f là đúng vì áp dụng tập Vet trong suy diễn tiến hoặc suy diễn lùi
- Kỹ thuật: Xác định tập Vết $T = \{r_1, r_2, \dots, r_k\}$ sao cho:

GT ████████

Hai cách giải thích:

- Giải thích trọn vẹn.
- Giải thích từng bước.

6.5 Bộ giải thích (tiếp)

- **Câu hỏi: Why not?**
 - Tại sao hệ thống lại không thể kết luận rằng f xuất phát từ các sự kiện trong GT?
 - Trả lời: Hệ thống đã thử các đường đi trong quá trình suy luận, nhưng không tìm ra được f .
- **Kỹ thuật: Hệ thống công bố quá trình suy luận (tiến hoặc lùi) để người sử dụng lần ra vết. Kỹ thuật đó là chỉ ra rằng $f \in GT^+$**

Các hệ cơ sở tri thức

KBS: Knowledge Based Systems

Trần Nguyên Hương

Hệ cơ sở tri thức

- Chương 1: Tổng quan về hệ cơ sở tri thức
- Chương 2: Biểu diễn và suy luận tri thức
- Chương 3: Hệ MYCIN
- Chương 4: Hệ học
- Chương 5: Hệ thống mờ cho các biến liên tục
-

Tài liệu tham khảo

1. GS.TSKH. Hoàng Kiếm. ***Giáo trình các hệ cơ sở tri thức***. NXB Đại học Quốc gia Thành phố Hồ Chí Minh – 2007
2. Đỗ Trung Tuấn. ***Hệ Chuyên gia***. NXB Giáo dục 1999
3. Robert I Levine. ***Knowledge Based Systems***. Wissenschafts Verlag, 1991.

Chương 1. Tổng quan về Hệ cơ sở tri thức

1.1. Khái niệm về Hệ cơ sở tri thức (CSTT)

- Hệ CSTT là chương trình máy tính được thiết kế để mô hình hoá khả năng giải quyết vấn đề của chuyên gia con người
- Hệ CSTT là hệ thống dựa trên tri thức, cho phép mô hình hoá các tri thức của chuyên gia, dùng tri thức này để giải quyết vấn đề phức tạp thuộc cùng lĩnh vực.
- Hai yếu tố quan trọng trong Hệ CSTT là: tri thức chuyên gia và lập luận, tương ứng với 2 khối chính là cơ sở tri thức và động cơ suy diễn.

1.1 Khái niệm về Hệ CSTT (Tiếp)

- Hệ Chuyên gia là một loại cơ sở tri thức được thiết kế cho một lĩnh vực ứng dụng cụ thể.

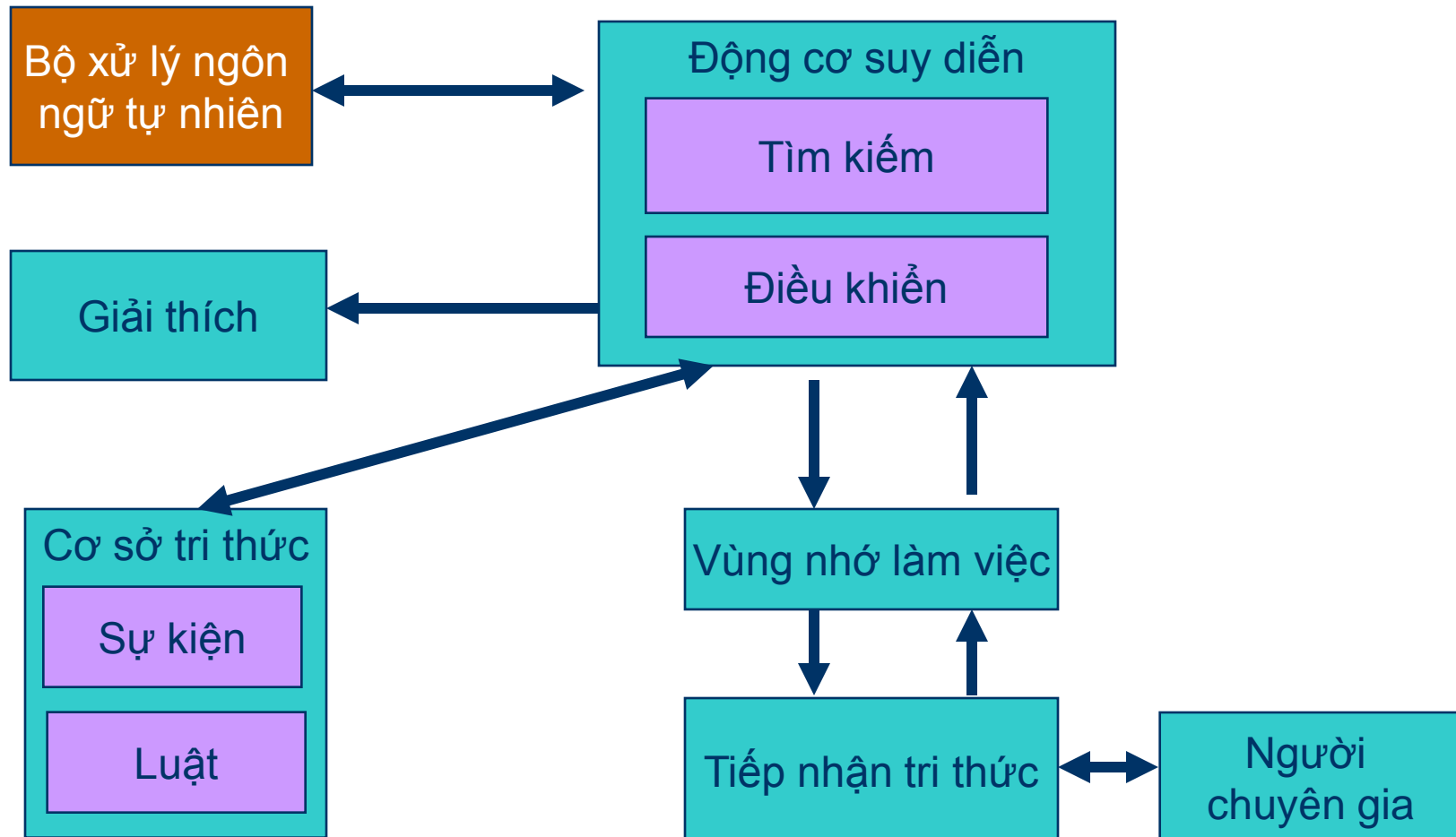
Ví dụ: Hệ Chuyên gia về chẩn đoán bệnh trong Y khoa, Hệ Chuyên gia chẩn đoán hồng học của đường dây điện thoại,...

- Hệ Chuyên gia làm việc như một chuyên gia thực thụ và cung cấp các ý kiến dựa trên kinh nghiệm của chuyên gia con người đã được đưa vào Hệ Chuyên gia.

1.1 Khái niệm về Hệ CSTT (Tiếp)

- Cơ sở tri thức: Chứa các tri thức chuyên sâu về lĩnh vực như chuyên gia. Cơ sở tri thức bao gồm: các sự kiện, các luật, các khái niệm và các quan hệ.
- Động cơ suy diễn: bộ xử lý tri thức theo mô hình hoá theo cách lập luận của chuyên gia. Động cơ hoạt động trên thông tin về vấn đề đang xét, so sánh với tri thức lưu trong cơ sở tri thức rồi rút ra kết luận.
- Kỹ sư tri thức (Knowledge Engineer): người thiết kế, xây dựng và thử nghiệm Hệ Chuyên gia.

1.2. Cấu trúc của Hệ chuyên gia



1.2 Cấu trúc của Hệ Chuyên gia(tiếp)

1. **Giao diện người máy (User Interface):** Thực hiện giao tiếp giữa Hệ Chuyên gia và User. Nhận các thông tin từ User (các câu hỏi, các yêu cầu về lĩnh vực) và đưa ra các câu trả lời, các lời khuyên, các giải thích về lĩnh vực đó. Giao diện người máy bao gồm: Menu, bộ xử lý ngôn ngữ tự nhiên và các hệ thống tương tác khác.
2. **Bộ giải thích (Explanation system):** Giải thích các hoạt động khi có yêu cầu của User.
3. **Động cơ suy diễn (Inference Engine):** Quá trình trong Hệ Chuyên gia cho phép khớp các sự kiện trong vùng nhớ làm việc với các tri thức về lĩnh vực trong cơ sở tri thức, để rút ra các kết luận về vấn đề đang giải quyết.

1.2 Cấu trúc của Hệ Chuyên gia(tiếp)

4. **Bộ tiếp nhận tri thức (Knowledge editor):** Làm nhiệm vụ thu nhận tri thức từ chuyên gia con người (human expert), từ kỹ sư tri thức và User thông qua các yêu cầu và lưu trữ vào cơ sở tri thức
5. **Cơ sở tri thức:** Lưu trữ, biểu diễn các tri thức mà hệ đảm nhận, làm cơ sở cho các hoạt động của hệ. Cơ sở tri thức bao gồm các sự kiện (facts) và các luật (rules).
6. **Vùng nhớ làm việc (working memory):** Một phần của Hệ Chuyên gia chứa các sự kiện của vấn đề đang xét.

1.3 Hệ hỗ trợ ra quyết định

DSS (Decision Support System)

- Chức năng: Hỗ trợ ra quyết định
- Hoạt động theo cách tương tác với người sử dụng

Các tính chất của DSS:

- Hướng đến các quyết định của người quản lý
- Uyển chuyển với hoàn cảnh
- Trả lời câu hỏi trong tình huống
- Do người sử dụng khởi động và kiểm soát

1.4. Hệ học

- Trong nhiều tình huống, sẽ không có sẵn tri thức như:
 - Kỹ sư tri thức cần thu nhận tri thức từ chuyên gia lĩnh vực.
 - Cần biết các luật mô tả lĩnh vực cụ thể
 - Bài toán không được biểu diễn tường minh theo luật, sự kiện hay quan hệ.
- Có 2 tiếp cận cho hệ thống học
 - **Học từ ký hiệu:** Bao gồm việc hình thức hoá, sửa chữa các luật tường minh, sự kiện và các quan hệ.
 - **Học từ dữ liệu số:** được áp dụng cho những hệ thống được mô hình dưới dạng số liên quan đến các kỹ thuật nhằm tối ưu các tham số. Học theo dạng số bao gồm: Mạng Nơron nhân tạo, thuật giải di truyền, bài toán tối ưu truyền thống. Các kỹ thuật học theo số không tạo ra CSTT tường minh.

1.5 Hệ điều khiển mờ

- Mờ hóa: Chuyển đổi giá trị rõ đầu vào thành các vector mờ
- Xác định các luật hợp thành và thuật toán xác định giá trị mờ
- Giải mờ: Phương pháp điểm trọng tâm

1.6 Ứng dụng của Hệ Cơ sở tri thức

1. Diễn giải (Interpretation): Mô tả tình huống các dữ liệu thu thập được
2. Dự báo (Prediction): đưa ra các tri thức về dự báo một tình huống: dự báo giá cả, ...
3. Thiết kế (Design): Lựa chọn cấu hình phù hợp, ví dụ: sắp xếp công việc.
4. Chẩn đoán (Diagnosis): Dựa vào các dữ liệu quan sát được, xác định các lỗi hỏng hóc.

1.6 Ứng dụng của Hệ Cơ sở tri thức(tiếp)

5. Vạch kế hoạch (Planing): tạo lập các phương án hành động.
6. Dẫn dắt (Monotoring): So sánh dữ liệu và các kết quả hoạt động.
7. Gỡ rối (Debugging): Mô tả các phương pháp khắc phục của hệ thống.
8. Giảng dạy (Instruction): Sửa chữa các lỗi của người học trong quá trình học tập.
9. Điều khiển (Control): dẫn dắt đáng điều tổng thể của hệ thống.

Chương 2. Biểu diễn và suy luận tri thức

Trần Nguyên Hương

Chương 2: Biểu diễn và suy luận tri thức

2.1. Mở đầu

- *Tri thức, lĩnh vực và biểu diễn tri thức.*

2.2. Các loại tri thức: được chia thành 5 loại

1. **Tri thức thủ tục:** mô tả cách giải quyết một vấn đề. Loại tri thức này đưa ra giải pháp để thực hiện một công việc nào đó. Các dạng tri thức thủ tục tiêu biểu thường là các luật, chiến lược, lịch trình và thủ tục.
2. **Tri thức khai báo:** cho biết một vấn đề được thấy như thế nào. Loại tri thức này bao gồm các phát biểu đơn giản, dưới dạng các khẳng định logic đúng hoặc sai. Tri thức khai báo cũng có thể là một danh sách các khẳng định nhằm mô tả đầy đủ hơn về đối tượng hay một khái niệm nào đó.

2.2. Các loại tri thức (tiếp)

3. **Siêu tri thức**: mô tả tri thức về tri thức. Loại tri thức này giúp lựa chọn tri thức thích hợp nhất trong số các tri thức khi giải quyết một vấn đề. Các chuyên gia sử dụng tri thức này để điều chỉnh hiệu quả giải quyết vấn đề bằng cách hướng các lập luận về miền tri thức có khả năng hơn cả.
4. **Tri thức heuristic**: Mô tả các “mẹo” để dẫn dắt tiến trình lập luận. Tri thức heuristic là tri thức không bảo đảm hoàn toàn 100% chính xác về kết quả giải quyết vấn đề. Các chuyên gia thường dùng các tri thức kho học như sự kiện, luật,... sau đó chuyển chúng thành các tri thức heuristic để thuận tiện hơn trong việc giải quyết một số bài toán.

2.2. Các loại tri thức (tiếp)

5. Tri thức có cấu trúc: mô tả tri thức theo cấu trúc. Loại tri thức này mô tả mô hình tổng quan hệ thống theo quan điểm của chuyên gia, bao gồm khái niệm, khái niệm con, và các đối tượng; diễn tả chức năng và mối liên hệ giữa các tri thức dựa theo cấu trúc xác định.

2.3. CÁC KỸ THUẬT BIỂU DIỄN TRI THỨC

2.3.1. Bộ ba: Đối tượng - Thuộc tính – Giá trị

2.3.2. Các luật dẫn

2.3.3. Mạng ngữ nghĩa

2.3.4. Frames

2.3.5. Logic

2.3.1. Bộ ba Đối tượng-Thuộc tính-Giá trị

- Một sự kiện có thể được dùng để xác nhận giá trị của một thuộc tính xác định của một vài đối tượng. Ví dụ, mệnh đề “quả bóng màu đỏ” xác nhận “đỏ” là giá trị thuộc tính “màu” của đối tượng “quả bóng”. Kiểu sự kiện này được gọi là bộ ba Đối tượng-Thuộc tính-Giá trị (O-A-V – Object – Attribute - Value)



2.3.1 Bộ ba Đối tượng-Thuộc tính-Giá trị (tiếp)

- Trong các sự kiện O-A-V, một đối tượng có thể có nhiều thuộc tính với các kiểu giá trị khác nhau. Hơn nữa một thuộc tính cũng có thể có một hay nhiều giá trị. Chúng được gọi là các sự kiện *đơn trị* (single-valued) hoặc *đa trị* (multi-valued). Điều này cho phép các hệ tri thức linh động trong việc biểu diễn các tri thức cần thiết.
- Các sự kiện không phải lúc nào cũng bảo đảm là đúng hay sai với độ chắc chắn hoàn toàn. Ví thế, khi xem xét các sự kiện, người ta còn sử dụng thêm một khái niệm là *độ tin cậy*. Phương pháp truyền thống để quản lý thông tin không chắc chắn là sử dụng nhân tố chắc chắn CF (certainly factor). Khái niệm này bắt đầu từ hệ thống MYCIN (khoảng năm 1975), dùng để trả lời cho các thông tin suy luận. Khi đó, trong sự kiện O-A-V sẽ có thêm một giá trị xác định độ tin cậy của nó là CF.

2.3.2 Các luật dẫn

- Luật là cấu trúc tri thức dùng để liên kết thông tin đã biết với các thông tin khác giúp đưa ra các suy luận, kết luận từ những thông tin đã biết.
- Trong hệ thống dựa trên các luật, người ta thu thập các tri thức lĩnh vực trong một tập và lưu chúng trong cơ sở tri thức của hệ thống. Hệ thống dùng các luật này cùng với các thông tin trong bộ nhớ để giải bài toán. Việc xử lý các luật trong hệ thống dựa trên các luật được quản lý bằng một module gọi là *bộ suy diễn*.

2.3.2 Các luật dẫn (tiếp)

Các dạng luật cơ bản: 7 dạng

1. Quan hệ:

IF Bình điện hỏng

THEN Xe sẽ không khởi động được

2. Lời khuyên:

IF Xe không khởi động được

THEN Đi bộ

3. Hướng dẫn

IF Xe không khởi động được AND Hệ thống nhiên liệu tốt

THEN Kiểm tra hệ thống điện

2.3.2 Các luật dẫn (tiếp)

4. Chiến lược

IF Xe không khởi động được

THEN Đầu tiên hãy kiểm tra hệ thống nhiên liệu, sau đó kiểm tra hệ thống điện

5. Diễn giải

IF Xe nổ AND tiếng giòn

THEN Động cơ hoạt động bình thường

6. Chẩn đoán

IF Sốt cao AND hay ho AND Họng đỏ

THEN Viêm họng

7. Thiết kế

IF Là nữ AND Da sáng

THEN Nên chọn Xe Spacy AND Chọn màu sáng

2.3.2 Các luật dẫn (tiếp)

Mở rộng cho các luật

Trong một số áp dụng cần thực hiện cùng một phép toán trên một tập hay các đối tượng giống nhau. Lúc đó cần các *luật có biến*.

Ví dụ: IF X là nhân viên AND Tuổi của X > 65
THEN X có thể nghỉ hưu

- Khi mệnh đề phát biểu về sự kiện, hay bản thân sự kiện có thể không chắc chắn, người ta dùng hệ số chắc chắn CF. Luật thiết lập quan hệ không chệch xác giữa các sự kiện giả thiết và kết luận được gọi là *luật không chắc chắn*.

Ví dụ: IF Lạm phát CAO THEN Hài như chắc chắn lãi suất sẽ CAO

Luật này được viết lại với giá trị CF có thể như sau:

IF Lạm phát cao THEN Lãi suất cao, CF = 0.8

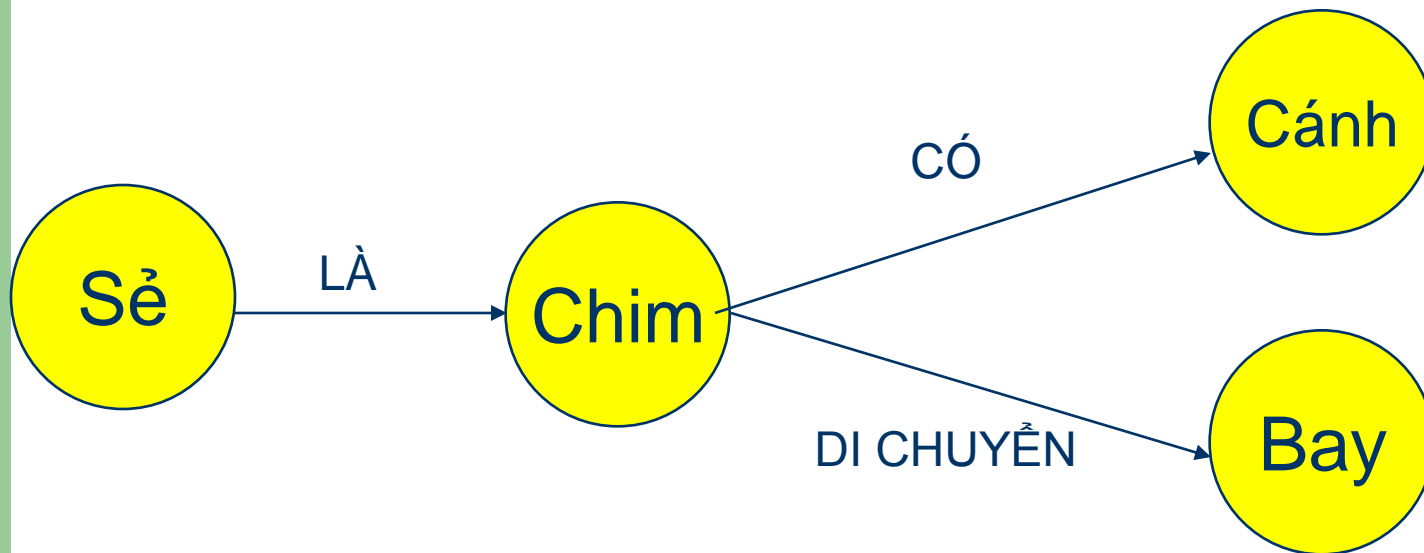
- Dạng luật tiếp theo là *siêu luật*:

Một luật với chức năng mô tả cách thức dùng các luật khác. Siêu luật sẽ đưa ra chiến lược sử dụng các luật theo lĩnh vực chuyên dụng, thay vì đưa ra thông tin mới.

Ví dụ: IF Xe không khởi động AND Hệ thống điện làm việc bình thường
THEN Có thể sử dụng các luật liên quan đến hệ thống điện

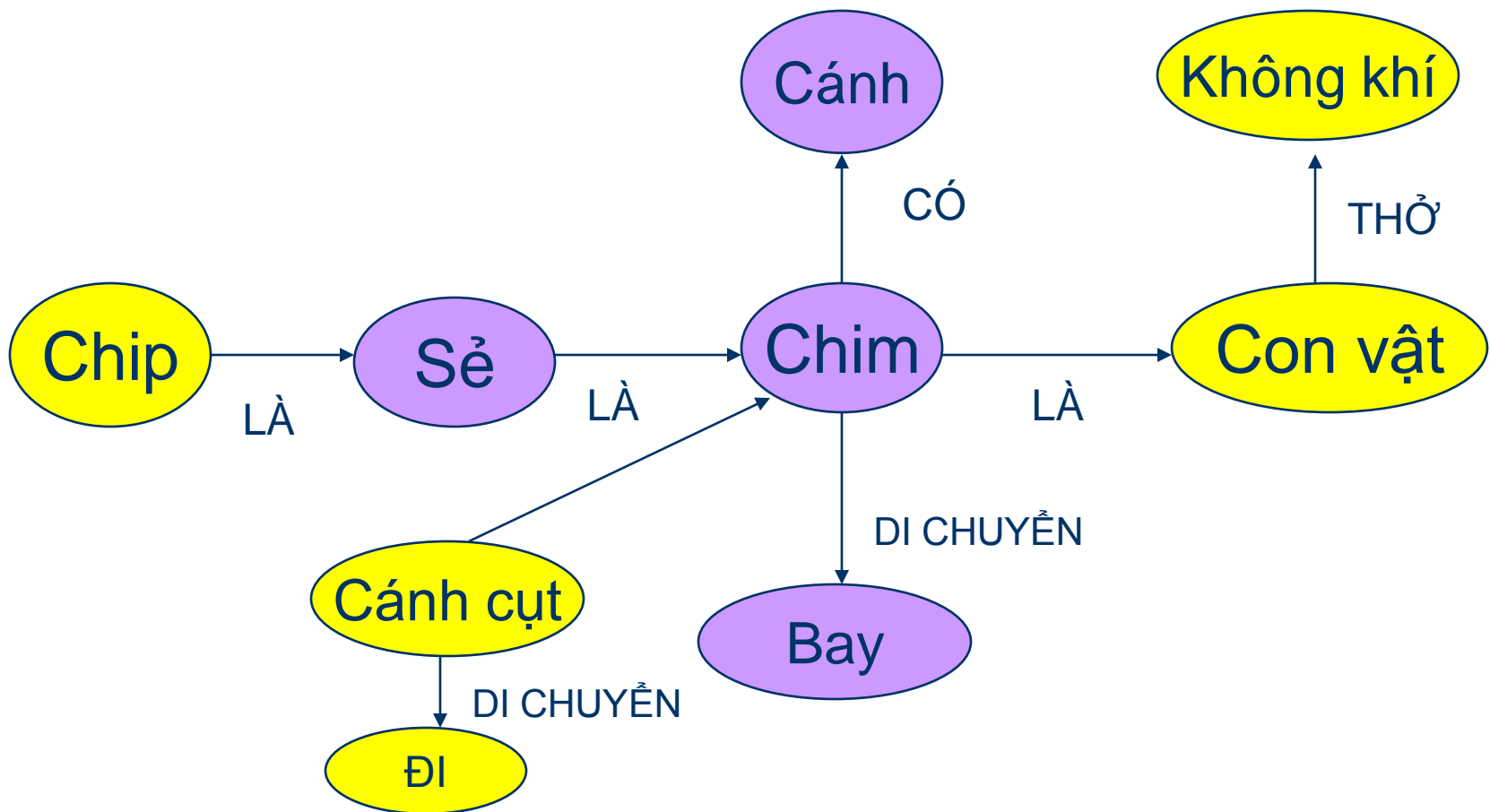
2.3.3. Mạng ngữ nghĩa

- Là một phương pháp biểu diễn tri thức dùng đồ thị trong đó nút biểu diễn đối tượng và cung biểu diễn quan hệ giữa các đối tượng.



Hình 2.3. “Sẻ là Chim” thể hiện trên mạng ngữ nghĩa

2.3.3. Mạng ngữ nghĩa (tiếp)



Hình 2.4. Phát triển mạng ngữ nghĩa

2.3.4. Frame

Frame là cấu trúc dữ liệu để thể hiện tri thức đa dạng về khái niệm hay đối tượng nào đó

PHIẾU ĐIỂM

Học tên:

Lớp:

Môn	Điểm
Toán	7
Văn	5
.....

Tên Frame

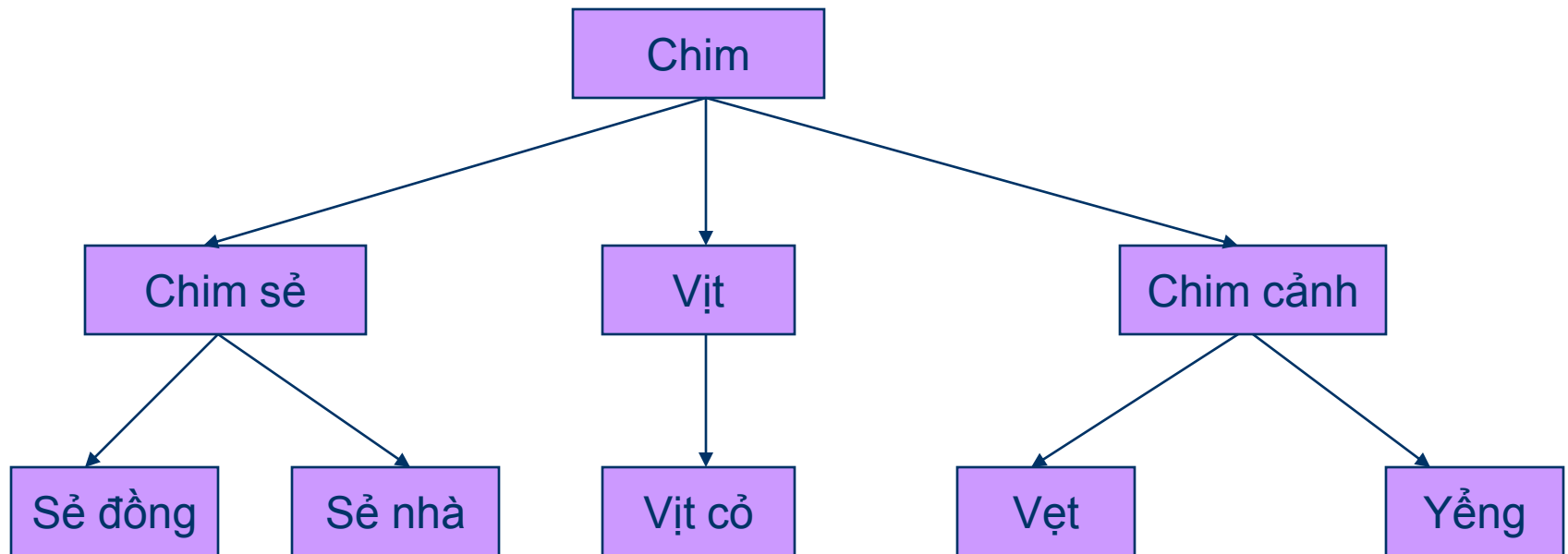
Lớp:

Thuộc tính:

Thuộc tính 1	Giá trị 1
Thuộc tính 2	Giá trị 2
.....

Hình 2.6. Cấu trúc Frame

2.3.4. Frame (tiếp)



Hình 2.7. Nhiều mức của Frame mô tả quan hệ phức tạp hơn

2.3.5 Logic

1. Logic mệnh đề

IF Xe không khởi động được (A)

AND Khoảng cách từ nhà đến chỗ làm là xa (B)

THEN Sẽ trễ giờ làm (C)

- Luật trên có thể biểu diễn lại như sau: $A \wedge B \Rightarrow C$

2. Logic vị từ

- Logic vị từ, cũng giống như logic mệnh đề, dùng các ký hiệu để thể hiện tri thức. Những ký hiệu này gồm **hằng số**, **vị từ**, **biến** và **hàm**.

2.4 SUY DIỄN DỮ LIỆU

1. Modus ponens

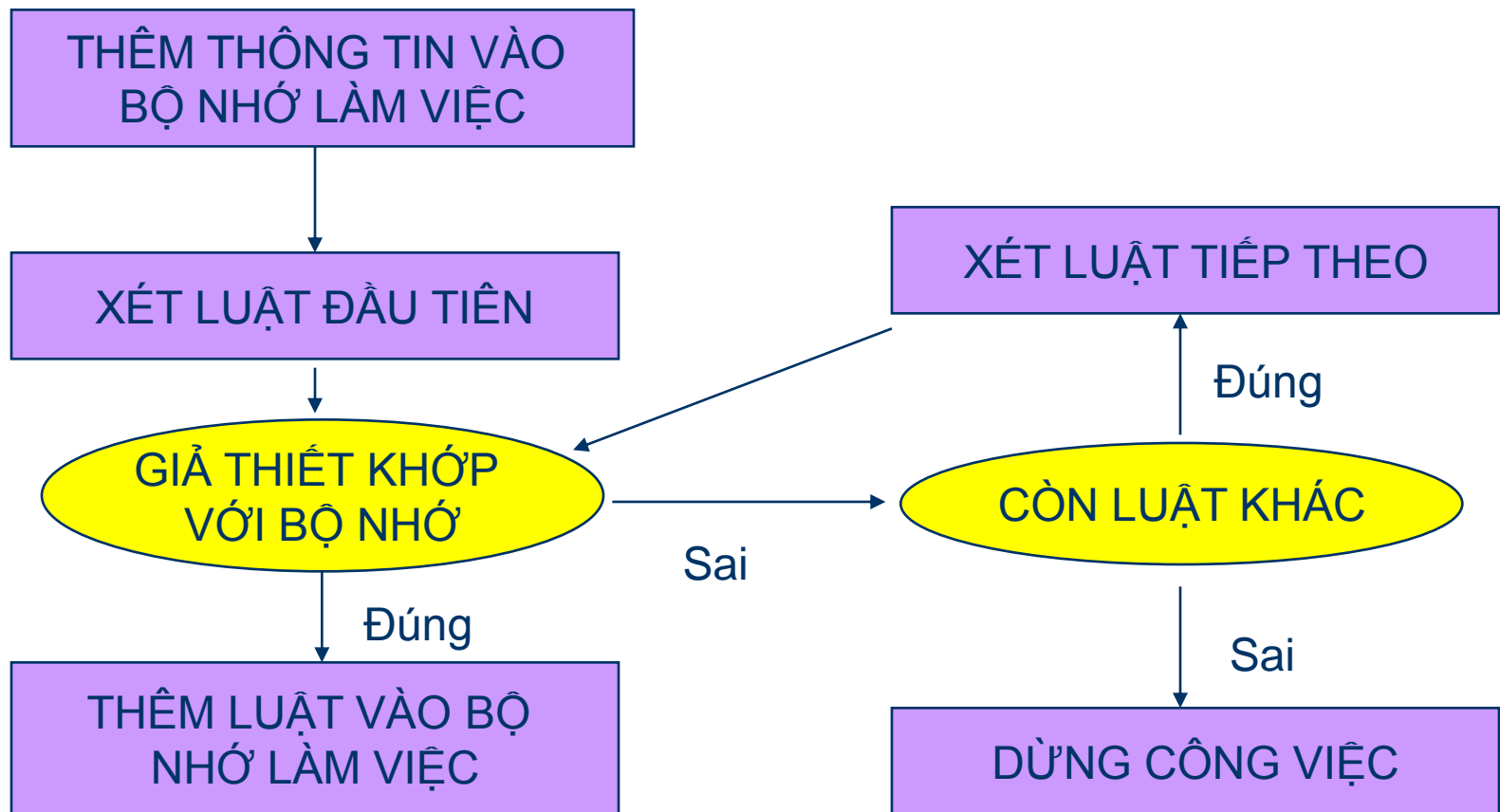
1. $E1$
2. $E1 \rightarrow E2$
3. $E2$

Nếu có tiên đề khác, có dạng $E2 \rightarrow E3$ thì $E3$ được đưa vào danh sách.

2. Modus tollens

1. $\neg E2$
2. $E1 \rightarrow E2$
3. $\neg E1$

2.4.2. Các hoạt động của Hệ thống Suy diễn tiến



Ví dụ về Suy diễn tiến

Luật 1. IF Bệnh nhân rất họng AND Nghi viêm nhiễm
THEN Tin rằng bệnh nhân viêm họng, đi chữa họng.

Luật 2. IF Nhiệt độ bệnh nhân quá 37 độ
THEN Bệnh nhân bị sốt

Luật 3. IF Bệnh nhân ốm trên 1 tuần AND Bệnh nhân sốt
THEN Nghi bệnh nhân viêm nhiễm.

Thông tin từ bệnh nhân là:

- Bệnh nhân có nhiệt độ 39 độ
- Bệnh nhân đã ốm hai tuần
- Bệnh nhân họng rất

Khi hệ thống thấy giả thiết của luật khớp với thông tin trong bộ nhớ, câu kết luận của luật được bổ sung vào bộ nhớ.

Minh họa

Ví dụ suy diễn lùi

Cơ chế suy diễn

Suy diễn với logic mệnh đề:

1. Thuật toán suy diễn tiến

Input: - Tập luật Rule = $\{r_1, r_2, \dots, r_m\}$
- GT, KL

Output: Thông báo "thành công" nếu $GT \rightarrow KL$
Ngược lại, thông báo "không thành công"

Method:

```
TD = GT;  
T = Loc(Rule, TD);  
While (KL  $\not\subseteq$  TD) AND (T  $\neq \emptyset$ ) Do  
{  
    r = Get(T);  
    TD = TD  $\cup$  {q}; // r: left  $\rightarrow$  q  
    Rule = Rule  $\setminus$  {r};  
    T = Loc(Rule, TD);  
}  
If KL  $\subseteq$  TD THEN Return "True"  
else Return "False"
```

Ví dụ: Rule = $\{r_1: a \rightarrow c, r_2: b \rightarrow d, r_3: a \rightarrow e, r_4: a \wedge d \rightarrow e, r_5: b \wedge c \rightarrow f, r_6: e \wedge f \rightarrow g\}$

Hỏi $a \wedge b \rightarrow g$?

Thuật toán suy diễn lùi

```
If KL  $\subseteq$  GT THEN Return "True"  
Else {TĐích= $\emptyset$ ; Vết =  $\emptyset$ ; First=1; Quaylui= False;}  
For Each q $\in$  KL DO TĐích=TĐích $\cup$ {(q,0)};  
Repeat  
    first ++;  
    ((f, i)=Get(TĐích));  
    If (f $\in$  GT) THEN  
    {  
        j = Timluật(f, i, Rule);           // rj: Leftj  $\rightarrow$  f  
        If (Tim có rj) THEN  
            { Vết = Vết  $\cup$ {(f, i)};  
              For Each t $\in$  (Leftj\GT) DO TĐích = TĐích $\cup$ {(t,0)};  
            }  
        else  
            { Quaylui=True;  
              While (f $\notin$  KL) AND Quaylui DO  
              {  
                  Repeat           { (g, k)=Get(Vết);  
                                     TĐích = TĐích \ Leftk; }  
                  Until f $\in$  Leftk;  
                  l=Timluật(g, k, Rule);  
              }  
            }  
    }  
}
```

Thuật toán suy diễn lùi

```
    If (Tìm có  $r_i$ ) THEN
      { TĐích = TĐích \ Left $_k$  ;
        For Each  $t \in$  (Left $_i$  \ GT) DO
          TĐích = TĐích  $\cup$  {( $t, 0$ )};
          Vết = Vết  $\cup$  {( $g, 1$ )};
          Quaylui = False;
        } //end if3
      else f=g;
    } //end while
  } //enf if2
} //end if1
```

Until (TĐích = \emptyset) OR (($f \in$ KL) and (First >2));

If ($f \in$ KL) then Return False else Return True;

Ví dụ: Rule = { $r_1: a \rightarrow c, r_2: b \rightarrow d, r_3: a \rightarrow e, r_4: a \wedge d \rightarrow e, r_5: b \wedge c \rightarrow f, r_6: e \wedge f \rightarrow g$ }

Hỏi $a \wedge b \rightarrow g$?

Ví dụ 2: Rule = { $r_1: a \wedge b \rightarrow c, r_2: a \wedge h \rightarrow d, r_3: b \wedge c \rightarrow e, r_4: a \wedge d \rightarrow m, r_5: a \wedge b \rightarrow p, r_6: p \wedge e \rightarrow m$ } Hỏi i) $a \wedge b \rightarrow m$? ii) $a \rightarrow m$?

2.4.3 Ưu điểm

* Suy diễn tiến

- Ưu điểm chính của suy diễn tiến là làm việc tốt khi bài toán về bản chất đi thu thập thông tin rồi thấy điều cần suy diễn.
- Suy diễn tiến cho ra khối lượng lớn các thông tin từ một số thông tin ban đầu. Nó sinh ra nhiều thông tin mới.
- Suy diễn tiến là tiếp cận lý tưởng đối với loại bài toán cần giải quyết các nhiệm vụ như lập kế hoạch, điều hành điều khiển và diễn dịch.

* Suy diễn lùi

- Một trong các ưu điểm chính của suy diễn lùi là phù hợp với bài toán đưa ra giả thuyết rồi xem hiệu quả giả thiết đó có đúng không.
- Suy diễn lùi tập trung vào đích đã cho. Nó tạo ra một loạt câu hỏi chỉ liên quan đến vấn đề đang xét, đến hoàn cảnh thuận tiện đối với người dùng.
- Khi suy diễn lùi muốn suy diễn cái gì đó từ các thông tin đã biết, nó chỉ tìm trên một phần của cơ sở tri thức thích đáng đối với bài toán đang xét.

2.4.4 Nhược điểm

* Suy diễn tiến

- Một nhược điểm chính của hệ thống suy diễn tiến là không cảm nhận được rằng chỉ một vài thông tin là quan trọng. Hệ thống hỏi các câu hỏi có thể hỏi mà không biết rằng chỉ một ít câu đã đi đến kết luận được.
- Hệ thống có thể hỏi cả câu không liên quan. Có thể các câu trả lời cũng quan trọng, nhưng làm người dùng lúng túng khi phải trả lời các câu không dính đến chủ đề.

* Suy diễn lùi

- Nhược điểm cơ bản của suy diễn này là nó thường tiếp theo dòng suy diễn, thay vì đúng ra phải đúng ở đó mà sang nhánh khác. Tuy nhiên có thể dùng nhân tố tin cậy và các luật meta để khắc phục.

Chương 3. Hệ MYCIN

Trần Nguyên Hương

Chương 3: Hệ MYCIN

3.1 Giới thiệu

MYCIN là một hệ lập luận trong y học được hoàn tất vào năm 1970 tại đại học Stanford, Hoa Kỳ. Đây là một hệ chuyên gia dựa trên luật và sự kiện, MYCIN sử dụng cơ chế lập luận gắn đng để xử lý các luật suy diễn dựa trên độ đo chắc chắn. Tiếp theo sau MYCIN, hệ EMYCIN ra đời. EMYCIN là một hệ chuyên gia tổng quát được tạo lập bằng cách loại phần cơ sở tri thức ra khỏi hệ MYCIN. EMYCIN cung cấp một cơ chế lập luận và tùy theo bài toán cụ thể sẽ bổ sung tri thức riêng của bài toán đó để tạo thành hệ chuyên gia.

Các đặc điểm chính:

- Sử dụng kỹ thuật suy diễn lùi.
- Có khả năng phân tích tri thức và điều khiển.
- Có tích hợp Meta-Rule.
- Có thể dùng khi thiếu thông tin hoặc thông tin không chắc chắn.
- Dễ sử dụng, giao tiếp bằng tiếng Anh.
- Cung cấp các chức giải thích: HOW, WHY.

Một phiên làm việc cụ thể với MYCIN:

?Thông tin về bệnh nhân

- MYCIN: Họ và tên?
- User: Jack Durkin.
- MYCIN: Tuổi?
- User: 24.
- MYCIN: Giới tính?
- User: Nam.
- MYCIN: Chủng tộc?
- User: CAUCASIAN.

?Xác định bản chất lây nhiễm

- MYCIN: Có bất kỳ phép cấy vi khuẩn cho Jack Durkin liên quan đến bệnh hiện tại?
- User: Có.
- MYCIN: Mẫu vật lấy từ đâu?
- User: CSF.
- MYCIN: Ngày giờ thực hiện cấy:
- User: 5-6-77 8:09 AM
-
- MYCIN: Gần đây Jack Durkin có các triệu
- chứng như: choáng váng, hôn mê không?
- User: Có.
-

CÁC THÀNH PHẦN CỦA HỆ MYCIN

1. Chương trình tư vấn: Cung cấp cho các Bác sĩ các lời khuyên để chọn phương pháp điều trị thích hợp bằng cách xác định rõ cách thức điều trị bởi các dữ liệu lấy ra từ các phòng thí nghiệm lâm sàng thông qua các câu trả lời của bác sĩ cho câu hỏi của máy tính.
2. Khả năng giải thích có tác động ngược lại: Cho phép chương trình tư vấn giải thích các kiến thức của nó về các phương pháp điều trị và chứng minh các chú thích về các phương pháp điều trị đặc biệt.
3. Thu nạp tri thức: cho phép các chuyên gia y học người trong lĩnh vực điều trị các căn bệnh truyền nhiễm dạy cho MYCIN các luật quyết định theo phương pháp điều trị mà họ tìm thấy trong thực tế lâm sàng.

PHẠM VI SỬ DỤNG CỦA HỆ MYCIN

1. Chẩn đoán nguyên nhân gây bệnh: đối với các bác sĩ điều trị, khi xét nghiệm cho bệnh nhân để có kết quả chẩn đoán chắc chắn mất 24-48 giờ. Nhiều trường hợp phải điều trị cả ngay khi chưa có kết luận hoàn chỉnh. MYCIN giúp chẩn đoán nguyên nhân gây bệnh nhanh hơn: khi gọi chương trình MYCIN, các bác sĩ trả lời các câu hỏi về tiền sử bệnh nhân, bệnh án, các kết quả xét nghiệm, các triệu chứng, ... từ đó MYCIN đưa ra chẩn đoán bệnh.
2. Tạo ra phương pháp điều trị: Sau khi nhận được các câu trả lời của bác sĩ về tình trạng bệnh nhân thông qua đối thoại. Trong trường hợp câu trả lời không biết hoặc biết không chắc chắn, thì MYCIN sẽ suy luận từ các thông tin không hoàn chỉnh.
3. Dự đoán diễn biến của bệnh: Bằng các câu hỏi "HOW, WHY", MYCIN sẽ giải thích các nguyên nhân và lý do cho các bác sĩ. Sau khi việc chẩn đoán bệnh và kê đơn hoàn tất, bác sĩ có thể theo dõi toàn bộ quá trình chẩn đoán bệnh của MYCIN và qua đó theo dõi diễn biến của bệnh.

NGUYÊN NHÂN THÀNH CÔNG CỦA MYCIN

1. Sự cần thiết của việc tư vấn dùng kháng sinh của các bác sĩ: vào thời điểm này việc lạm dụng kháng sinh đã đem lại không ít phản ứng phụ.
2. Cơ sở tri thức của MYCIN được thu nạp từ các chuyên gia xuất sắc nhất trong lĩnh vực.
3. MYCIN không bao giờ đi đến ngay kết luận để luôn có thêm các thông tin cốt yếu qua mỗi bước.
4. MYCIN được hình thành từ một chương trình trí tuệ nhân tạo đã được áp dụng thực tế (DENDRAL) và đã được thực hiện tại trung tâm y tế nổi tiếng với các tri thức mới nhất về bệnh học và dược học.

3.2 LÝ THUYẾT VỀ SỰ CHẮC CHẮN

MB (Measure of Belief in): độ đo sự tin cậy

MD (Measure of Disbelief in): độ đo sự không tin cậy

CF (Certainly Factor): Hệ số chắc chắn

Gọi: **MB**(H/E) là độ đo sự tin cậy của giả thuyết H khi có chứng cứ E.

MD(H/E) là độ đo sự không tin cậy của giả thuyết H khi có chứng cứ E.

Khi đó: $0 < \mathbf{MB}(H/E) < 1$ trong khi **MD**(H/E) = 0

$0 < \mathbf{MD}(H/E) < 1$ trong khi **MB**(H/E) = 0

Độ đo chắc chắn **CF**(H/E) được tính bằng công thức:

$$\mathbf{CF}(H/E) = \mathbf{MB}(H/E) - \mathbf{MD}(H/E)$$

3.2 LÝ THUYẾT VỀ SỰ CHẮC CHẮN *(tiếp)*

1. Luật đơn giản: **If(e) then (c)**

CF(e) là độ đo chắc chắn của chứng cứ.

CF(r) là độ đo chắc chắn của luật suy diễn.

Khi đó: CF(c) là độ đo chắc chắn của kết luận sẽ được tính bằng công thức:

$$CF(c) = CF(e) * CF(r)$$

2. Luật phức tạp:

- **If(e1 AND e2) then (c)**

$$CF(e1 \text{ AND } e2) = \text{MIN}(CF(e1), CF(e2))$$

- **if (e1 OR e2) then (c)**

$$CF(e1 \text{ OR } e2) = \text{MAX}(CF(e1), CF(e2))$$

3.2 LÝ THUYẾT VỀ SỰ CHẮC CHẮN *(tiếp)*

3. Với luật: **if ((e1 AND e2) OR e3) then (c)**

CF ((e1 AND e2) OR e3) = MAX(MIN(CF(e1), CF(e2)), CF(e3))

4. **CF(NOT e) = - CF(e)**

5. Kết hợp nhiều luật có cùng kết luận:

- Luật 1: **if(e1) then (c) với CF(r1): độ đo chắc chắn của luật 1**
- Luật 2: **if(e2) then (c) với CF(r2): độ đo chắc chắn của luật 2**

- Với **CF(t1), CF(t2)** là CF của kết luận của luật 1 và 2, khi **CF(t1)** và **CF(t2)** đều dương thì:

$$\text{Ctổng} = \text{CF}(t1) + \text{CF}(t2) - \text{CF}(t1) * \text{CF}(t2)$$

- Khi **CF(t1)** và **CF(t2)** đều âm thì:

$$\text{Ctổng} = \text{CF}(t1) + \text{CF}(t2) + \text{CF}(t1) * \text{CF}(t2)$$

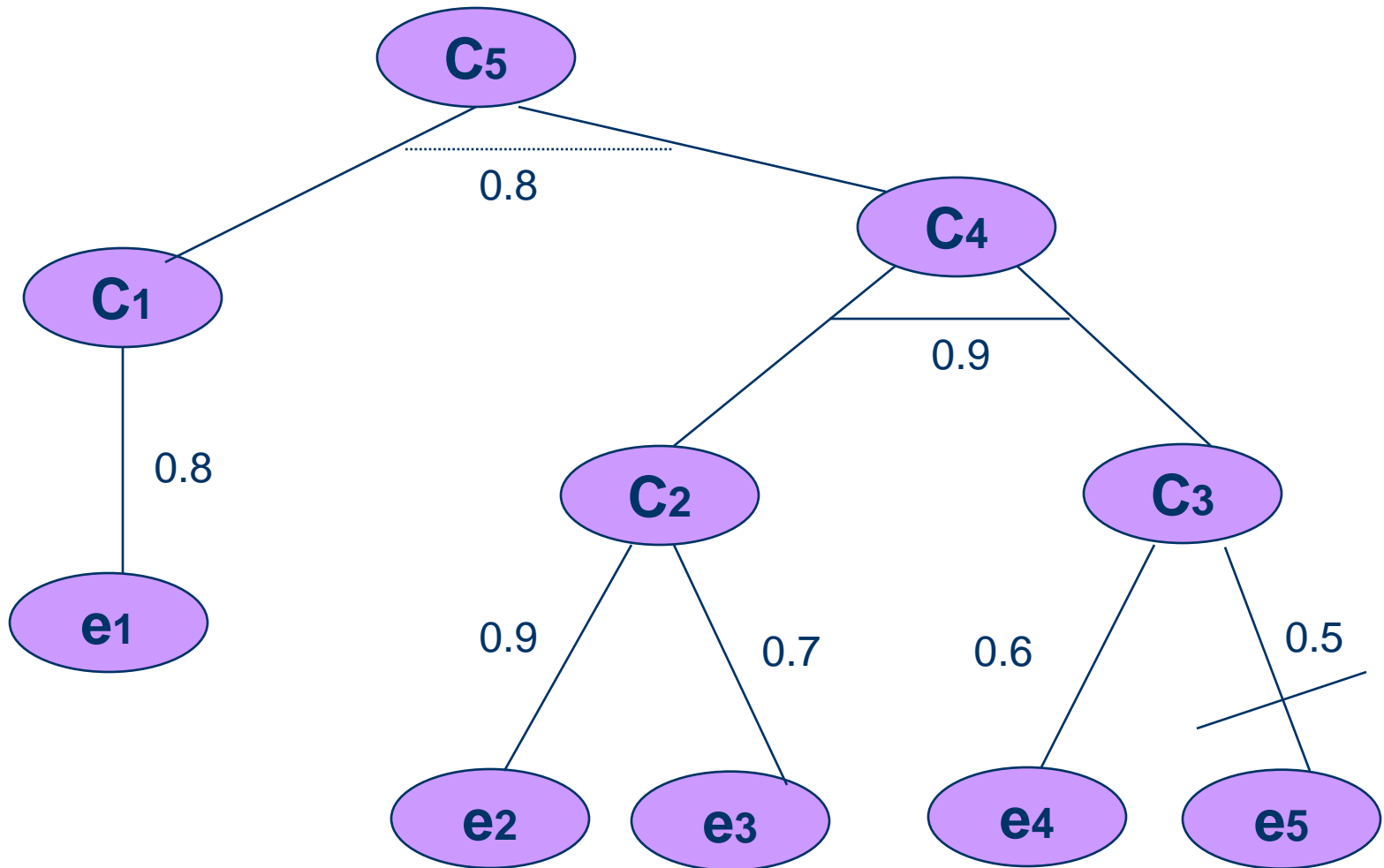
- Nếu **CF(t1)** khác dấu với **CF(t2)** thì:

$$\text{Ctổng} = (\text{CF}(t1) + \text{CF}(t2)) / (1 - \text{MIN}(\text{ABS}(\text{CF}(t1)), \text{ABS}(\text{CF}(t2))))$$

Ví dụ về lập luận trong Hệ MYCIN

- Ví dụ: Có 7 luật sau đây:
 - r1: **If**(e1) **Then** (c1) **CF**(r1) = 0,8
 - r2: **If** (e2) **Then** (c2) **CF**(r2) = 0,9
 - r3: **If** (e3) **Then** (c2) **CF**(r3) = 0,7
 - r4: **If** (e4) **Then** (c3) **CF**(r4) = 0,6
 - r5: **If** (**NOT** e5) **Then** (c3) **CF**(r5) = 0,5
 - r6: **If** (c2 **AND** c3) **Then** (c4) **CF**(r6) = 0,9
 - r7: **If** (c1 **OR** c4) **Then** (c5) **CF**(r7) = 0,8
- Bảng luật này tạo thành mạng suy diễn ở hình 3.1 với c5 là giả thuyết cần hướng đến.

Hình 3.1. Mạng suy diễn



Lập luận trên mạng suy diễn

- Giả sử các chứng cứ e_1, e_2, e_3, e_4, e_5 có độ đo chắc chắn như sau:

$$CF(e_1) = 0,9$$

$$CF(e_2) = 0,9$$

$$CF(e_3) = -0,3$$

$$CF(e_4) = 0,4$$

$$CF(e_5) = -0,3$$

Lập luận trên mạng suy diễn (tiếp)

- Chúng ta sẽ lập luận từ các **CF** của chứng cứ dẫn lên giả thuyết c5 như sau:
- Dựa vào luật r1 tính được **CF(c1)**:
- **CF(c1) = CF(e1) * CF(r1) = 0,8*0,9 = 0,72**
- Dựa vào luật r2, r3 tính được **CF(c2)**
- Với luật r2: **CF(c2) = CF(e2) * CF(r2) = 0,9 * 0,9 = 0,81**
- Với luật r3: **CF(c2) = CF(e3) * CF(r3) = -0,3 * 0,7 = -0,21**
- Do **CF(c2)** của r2 trái dấu với **CF(c2)** của r3, nên:
CF(c2) tổng = (0,81 + (-0,21)) / (1-MIN (0,81, 0,21)) = 0,74

Lập luận trên mạng suy diễn (tiếp)

- Dựa vào luật r4, r5 ta tính được CF(c3)
- Với luật r4:
- $CF(c3) = CF(e4) * CF(r4) = 0,4 * 0,6 = 0,24$
- Với luật r5:
- $CF(c3) = CF(\text{NOT } e5) * CF(r5) = -CF(e5) * CF(r5) = 0,3 * 0,5 = 0,15$
- Do CF(c3) của r4 và CF(c3) của r5 cùng dương nên
 $CF(c3)_{\text{tổng}} = 0,24 + 0,15 - 0,24 * 0,15 = 0,354$
- Dựa vào luật r6 ta tính được CF(c4):
- $CF(c4) = \text{MIN}(CF(c2), CF(c3)) * CF(r6) = \text{MIN}(0,74, 0,354) * 0,9 = 0,354 * 0,9 = 0,3186$
- Dựa vào luật r7 ta tính được CF(c5)
- $CF(c5) = \text{MAX}(CF(c1), CF(c4)) * CF(r7) = \text{MAX}(0,72, 0,3186) * 0,8 = 0,576$
- Như thế độ chắc chắn của giả thuyết c5 là 0,576.

Chương 4. HỆ HỌC

Trần Nguyên Hương

4.1. Mở đầu

- Các chương trước đã thảo luận về biểu diễn và suy luận tri thức. Trong trường hợp này giả định đã có sẵn tri thức và có thể biểu diễn tường minh tri thức.
- Tuy vậy, trong nhiều tình huống sẽ không có sẵn tri thức:
 - Cần biết các luật mô tả lĩnh vực cụ thể
 - Bài toán không được biểu diễn tường minh dưới dạng các luật sự kiện
 - ..

4.1. Mở đầu

- Một người lạc trên hoang đảo. Để sống, cần phải thử xem loại quả nào ăn được loại nào độc. Sau nhiều lần thử, sẽ lập được bảng thống kê sau

Identifying what's good to eat?

<i>Conclusion</i>	<i>Skin</i>	<i>Colour</i>	<i>Size</i>	<i>Flesh</i>
safe	Hairy	brown	large	Hard
safe	hairy	green	large	Hard
dangerous	smooth	red	large	Soft
safe	hairy	green	large	Soft
safe	hairy	red	small	Hard
safe	smooth	red	small	Hard
safe	smooth	brown	small	Hard
dangerous	hairy	green	small	Soft
dangerous	smooth	green	small	Hard
safe	hairy	red	large	Hard
safe	smooth	brown	large	Soft
dangerous	smooth	green	small	soft
safe	hairy	red	small	soft
dangerous	smooth	red	large	hard
safe	smooth	red	small	hard
dangerous	hairy	green	small	hard

4.2. Các hình thức học

1. Học vẹt
2. Học bằng cách chỉ dẫn
3. Học bằng quy nạp
4. Học bằng tương tự
5. Học dựa trên giải thích
6. Học dựa trên tình huống
7. Học không giám sát



Xem giáo trình
(chương 7)

Ví dụ về CÁC HÌNH THỨC HỌC

Ví dụ:

- **Hệ MYCIN**
- **Mạng Neural nhân tạo**
- **Thuật toán học Quinland**
- **Bài toán nhận dạng**
- **Máy chơi cờ carô, cờ tướng**

4.3. Bài toán

- Cho bảng nhiều cột, mỗi cột là một dấu hiệu (thuộc tính..),
- Một cột là kết luận chỉ 2 khả năng “có” | “không”.
- Mỗi dòng của bảng là một trường hợp (có được từ người chuyên gia, từ kinh nghiệm quá khứ ..).
- > Cây quyết định biểu diễn tri thức từ bảng này các nút = lựa chọn, rẽ thành nhiều nhánh, tùy theo giá trị của một dấu hiệu (thuộc tính ..).
- Nút lá là một phương án quyết định: có | không.

Ví dụ

- Một người lạc trên hoang đảo. Để sống, cần phải thử xem loại quả nào ăn được loại nào độc. Sau nhiều lần thử, sẽ lập được bảng thống kê sau

Identifying what's good to eat?

<i>Conclusion</i>	<i>Skin</i>	<i>Colour</i>	<i>Size</i>	<i>Flesh</i>
safe	Hairy	brown	large	Hard
safe	hairy	green	large	Hard
dangerous	smooth	red	large	Soft
safe	hairy	green	large	Soft
safe	hairy	red	small	Hard
safe	smooth	red	small	Hard
safe	smooth	brown	small	Hard
dangerous	hairy	green	small	Soft
dangerous	smooth	green	small	Hard
safe	hairy	red	large	Hard
safe	smooth	brown	large	Soft
dangerous	smooth	green	small	soft
safe	hairy	red	small	soft
dangerous	smooth	red	large	hard
safe	smooth	red	small	hard
dangerous	hairy	green	small	hard

Ví dụ

- Mỗi dòng trong bảng là một ví dụ học.
- Bảng là tập ví dụ học
- Mỗi dòng có thể coi là một luật dạng

IF skin = hairy
and colour = brown
and size = large
and flesh = hard
THEN conclusion = safe

Ví dụ

- Có thể tạo ra một cây quyết định để thay thế tập hợp các luật
- Một nút lựa chọn của cây quyết định có dạng tổng quát như sau

IF thuộc tính = giá trị1 then <subtree 1>
 else if thuộc tính = giá trị2 then <subtree 2>
 else if ...

.....

 else if thuộc tính = giá trịN then <subtree N>

- Trên cây quyết định, một đường đi từ gốc đến nút lá sẽ ứng với một luật.

4.4. Tạo cây quyết định - Thuật toán CLS

- Do Hunt đề xuất sử dụng trong hệ thống học khái niệm CLS – concept learning system, 1966.
- Là Thuật toán học quy nạp lần đầu tiên

4.4. Tạo cây quyết định - Thuật toán CLS

Xuất phát: cây rỗng. Bổ xung thêm dần các nút cho đến khi cây quyết định phân loại được đúng tất cả các ví dụ trong tập học C .

1- Nếu các ví dụ trong C đều “đúng” thì tạo nút lá “có” và kết thúc

Nếu các ví dụ trong C đều “sai” thì tạo nút lá “không” và kết thúc. Trái lại:

2- Chọn một thuộc tính A có các giá trị $V_1, V_2 \dots V_m$. Tạo nút quyết định m nhánh.

3- Chia tập ví dụ học thành m tập con $C_1, C_2 \dots C_m$ tùy theo giá trị của thuộc tính đã chọn

4- Quay lại từ bước 1.

4.4. Tạo cây quyết định - Thuật toán CLS

- Trong thuật toán CLS, việc chọn thuộc tính A ở bước 2 là ngẫu nhiên.
- Thuật toán Quinland sẽ cải tiến để tăng hiệu quả bằng cách chọn thuộc tính có *độ phân biệt cao nhất*.

Minh hoạ CLS

<i>Conclusion</i>	<i>Skin</i>	<i>Colour</i>	<i>Size</i>	<i>Flesh</i>
safe	Hairy	brown	large	Hard
safe	hairy	green	large	Hard
dangerous	smooth	red	large	Soft
safe	hairy	green	large	Soft
safe	hairy	red	small	Hard
safe	smooth	red	small	Hard
safe	smooth	brown	small	Hard
dangerous	hairy	green	small	Soft
dangerous	smooth	green	small	Hard
safe	hairy	red	large	Hard
safe	smooth	brown	large	Soft
dangerous	smooth	green	small	soft
safe	hairy	red	small	soft
dangerous	smooth	red	large	hard
safe	smooth	red	small	hard
dangerous	hairy	green	small	hard

Minh hoạ CLS

Skin="Hairy"

<i>Conclusion</i>	<i>Skin</i>	<i>Colour</i>	<i>Size</i>	<i>Flesh</i>
safe	Hairy	brown	large	Hard
safe	hairy	green	large	Hard
safe	hairy	green	large	Soft
safe	hairy	red	small	Hard
dangerous	hairy	green	small	Soft
safe	hairy	red	large	Hard
safe	hairy	red	small	soft
dangerous	hairy	green	small	hard

Minh hoạ CLS

Skin="Smooth"

<i>Conclusion</i>	<i>Skin</i>	<i>Colour</i>	<i>Size</i>	<i>Flesh</i>
dangerous	smooth	red	large	Soft
safe	smooth	red	small	Hard
safe	smooth	brown	small	Hard
dangerous	smooth	green	small	Hard
safe	smooth	brown	large	Soft
dangerous	smooth	green	small	soft
dangerous	smooth	red	large	hard
safe	smooth	red	small	hard

Minh hoạ CLS

Skin="Hairy" and Size = "large"

<i>Conclusion</i>	<i>Skin</i>	<i>Colour</i>	<i>Size</i>	<i>Flesh</i>
safe	Hairy	brown	large	Hard
safe	hairy	green	large	Hard
safe	hairy	green	large	Soft
safe	hairy	red	large	Hard

R1:

If Skin="Hairy" and Size = "large"
Then Safe

Minh hoạ CLS

Skin="Hairy" and Size = "Small"

<i>Conclusion</i>	<i>Skin</i>	<i>Colour</i>	<i>Size</i>	<i>Flesh</i>
safe	hairy	red	small	Hard
dangerous	hairy	green	small	Soft
safe	hairy	red	small	soft
dangerous	hairy	green	small	hard

R2:

If Skin="Hairy" and Size = "Small" and Colour="Green"
then Dangerous

R3:

If Skin="Hairy" and Size = "Small" and Colour="Red"
then Safe

Skin="Smooth" and Size = "Small"

<i>Conclusion</i>	<i>Skin</i>	<i>Colour</i>	<i>Size</i>	<i>Flesh</i>
safe	smooth	red	small	Hard
safe	smooth	brown	small	Hard
dangerous	smooth	green	small	Hard
safe	smooth	red	small	hard

R4:

If Skin="Smooth" and Size = "Small" and Colour="Green"
then Dangerous

R5:

If Skin="Smooth" and Size = "Small" and Colour="Red"
then Safe

R6: If Skin="Smooth" and Size = "Small" and Colour="Brown"
then Safe

Skin="Smooth" and Size = "Large"

<i>Conclusion</i>	<i>Skin</i>	<i>Colour</i>	<i>Size</i>	<i>Flesh</i>
dangerous	smooth	red	large	Soft
safe	smooth	brown	large	Soft
dangerous	smooth	red	large	hard

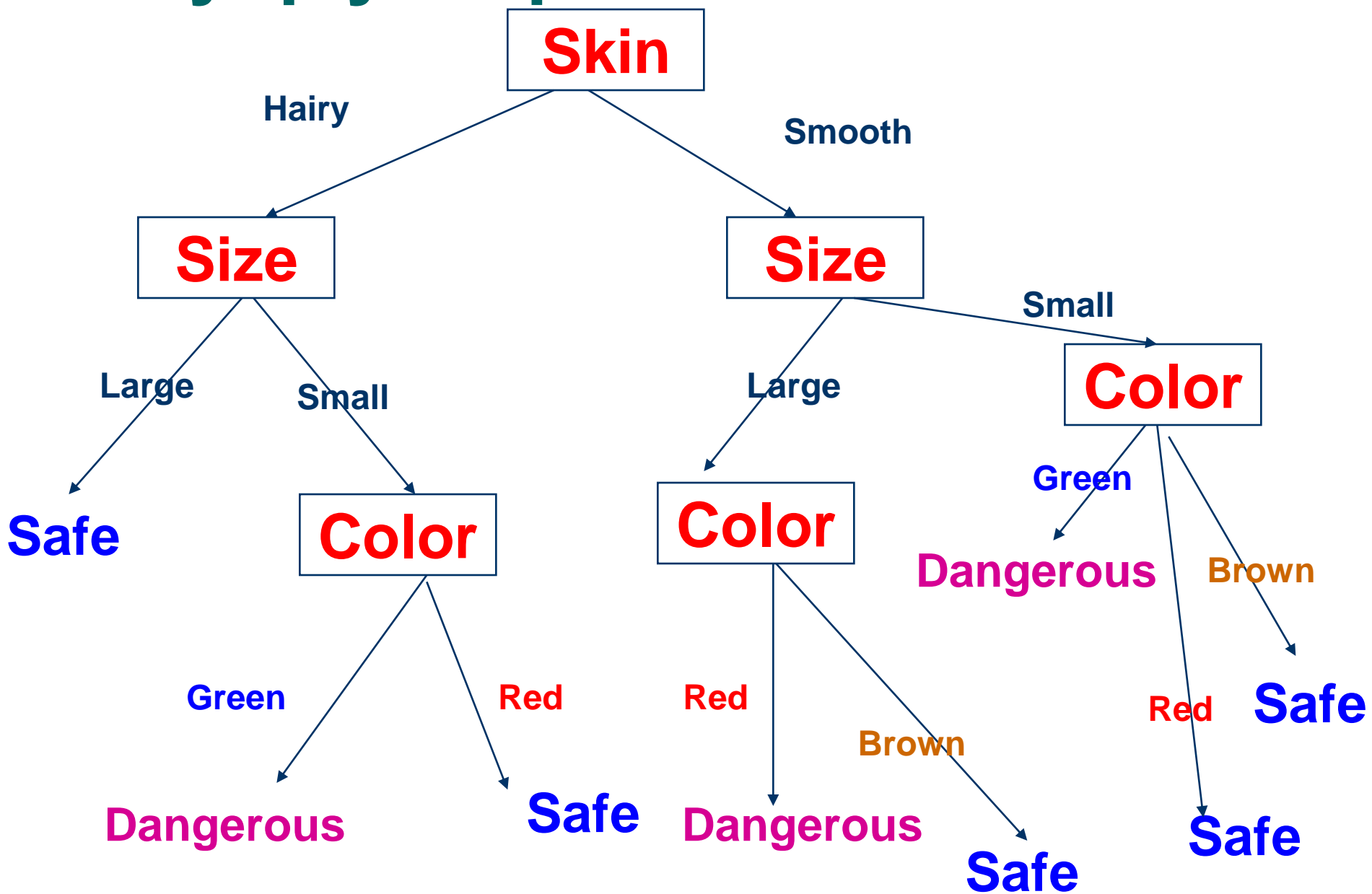
R7:

If Skin="Smooth" and Size = "Large" and Colour="Red"
then Dangerous

R8:

If Skin="Smooth" and Size = "Small" and Colour="Brown"
then Safe

Cây quyết định



4.5. Entropy và mức độ phân biệt của một thuộc tính

- Thế nào là độ phân biệt của một thuộc tính?
- Lí thuyết thông tin cho phép lượng hóa thông tin.
- Một cách tổng quát, giả sử kết luận C có thể nhận một trong n giá trị c_1, c_2, \dots, c_n . Trong ví dụ trên, C nhận 2 giá trị “ăn được”, “độc”.

4.5. Entropy và mức độ phân biệt của một thuộc tính (tiếp)

- Giả sử thuộc tính A có thể nhận m giá trị a_1, a_2, \dots, a_m .
- Kí hiệu xác suất điều kiện $P(C = c_i | A = a_j)$ hay gọn hơn $P(c_i | a_j)$.
- Ví dụ $P(C = \text{safe} | \text{Skin} = \text{hairy}) = 6/8 = 3/4$,
(8 dòng với $\text{Skin} = \text{hairy}$,
(trong đó 6 dòng kết luận $C = \text{safe}$).

4.5. Entropy và mức độ phân biệt của một thuộc tính (tiếp)

$$\text{Entropy}(C) = - \sum_{i=1, 2, \dots, n} P(C = c_i) \log_2 P(C = c_i)$$

4.5. Entropy và mức độ phân biệt của một thuộc tính (tiếp)

- Entropy của thông tin $A = a_j$ đối với kết luận C

Biểu thức $-\log_2 P(c_i | a_j)$ là lượng tin mà $A = a_j$ mang lại cho kết luận $C = c_i$.

Tổng theo $i = 1 \dots n$ là entropy của thông tin $A = a_j$ đối với kết luận C : $Entropy(a_j) = - \sum P(c_i | a_j) \log_2 P(c_i | a_j)$

Entropy của thuộc tính A đối với C được định nghĩa là tổng

$$\begin{aligned} Entropy(A) &= - \sum P(A = a_j) * Entropy(a_j) \\ &= - \sum P(A = a_j) \sum P(c_i | a_j) \log_2 P(c_i | a_j) \end{aligned}$$

4.5. Entropy và mức độ phân biệt của một thuộc tính (tiếp)

- Entropy là một số biến thiên trong đoạn $[0, 1]$. Entropy là độ đo mức nghi ngờ, mức ngẫu nhiên của kết luận.
 - Nó càng cao thì nghi ngờ về kết luận C càng lớn.
 - Entropy càng thấp thì a_j càng mang nhiều thông tin về kết luận C .
 - Entropy = 0 nghĩa là tất cả các ví dụ thuộc cùng 1 lớp, có cùng một kết luận.
 - Entropy = 1 nghĩa là hoàn toàn ngẫu nhiên

4.5. Entropy và mức độ phân biệt của một thuộc tính (tiếp)

- Ví dụ

Xét thuộc tính *Size* trong ví dụ trên. Từ bảng dữ liệu ta có

- $P(\text{safe} \mid \text{large}) = 5/7$
- $P(\text{dangerous} \mid \text{large}) = 2/7$
- $P(\text{large}) = 7/16$
- $P(\text{safe} \mid \text{small}) = 5/9$
- $P(\text{dangerous} \mid \text{small}) = 4/9$
- $P(\text{small}) = 9/16$

4.5. Entropy và mức độ phân biệt của một thuộc tính (tiếp)

- Như vậy entropy của thuộc tính *Size* đối với kết luận “safe | dangerous” là

$$\begin{aligned} & 7/16 * \{5/7 * \log_2 5/7 + 2/7 * \log_2 2/7\} + \\ & 9/16 * \{5/9 * \log_2 5/9 + 4/9 * \log_2 4/9\} = 0.9350955 \end{aligned}$$

4.5. Entropy và mức độ phân biệt của một thuộc tính (tiếp)

- Định nghĩa lượng tin mà thuộc tính A mang lại đối với tập ví dụ

$$\text{Gain}(C, A) = \text{Entropy}(C) - \text{Entropy}(A)$$

ở đây:

Thuộc tính có entropy thấp nhất chính là có độ phân biệt cao nhất (cho kết luận C).

4.6. Thuật giải Quinlan

- Là thuật toán học theo quy nạp dùng luật, đa mục tiêu
- Do Quinlan đưa ra năm 1979. Cải tiến thuật toán CLS.
- Còn gọi là thuật toán ID3 (ID là viết tắt của 'iterative dichotomiser = chia đôi nhiều lần)
- Ý tưởng:
 - 1- CLS làm việc với toàn bộ tập thí dụ học có sẵn từ đầu. ID3 giảm số lượng thí dụ học, dùng một tập con xuất phát.
 - 2- ở mỗi bước ID3 chọn thuộc tính có mức phân biệt cao nhất để phân nhánh.

4.6. Thuật giải Quinlan

- **Các bước của thuật toán ID3**
 1. Chọn ngẫu nhiên một tập con W của tập các ví dụ học, gọi là cửa sổ
 2. Áp dụng thuật toán CLS tạo cây (hay luật) quyết định cho W
 3. Duyệt toàn bộ các ví dụ còn lại (trừ W) trên cây để phát hiện các ngoại lệ
 4. Nếu có ví dụ là ngoại lệ, thêm vào W và lặp lại từ bước 2. Trái lại, kết thúc cho kết quả là cây nhận được

4.6. Thuật giải Quinlan

Chi tiết bước 2

1. Tính entropy của tất cả các thuộc tính (đối với kết luận cần quyết định);
2. Chọn thuộc tính (ví dụ A) có entropy thấp nhất
3. Chia tập ví dụ thành các tập con tùy theo giá trị của thuộc tính A. A nhận cùng một giá trị trên mỗi tập con.
4. Xây dựng cây phân nhánh theo giá trị của A:
if $A=a_1$ then ... (subtree1)
if $A=a_2$ then ... (subtree2)
...etc...
5. Lặp lại từ bước 1 với mỗi cây con.
6. Mỗi lần lặp xét được 1 thuộc tính. Quá trình dừng khi đã xét hết các thuộc tính, hoặc không cần phân nhánh nữa (vì mọi ví dụ trong một cây con đã có cùng kết luận).

Minh họa Thuật giải Quinlan

- Tập C gồm 14 ví dụ như trong bảng dưới. Cột kết luận là có chơi bóng hay không.
- Các thuộc tính: outlook, temperature, humidity, wind speed.
- Các giá trị có thể:
 - Outlook (thời tiết) = { sunny, overcast – u ám , rain }
 - temperature (nhiệt độ) = { hot, mild, cool }
 - Humidity (độ ẩm) = { high, normal }
 - Wind (gió) = { weak, strong }

Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Minh họa Thuật giải Quinlan

- Xác định thuộc tính làm nút gốc cây.
- Tính Gain của tất cả các thuộc tính
- Ví dụ $\text{Gain}(C, \text{Wind}) = ?$

$$\text{Gain}(C, \text{Wind}) = \text{Entropy}(C) - (8/14) * \text{Entropy}(\text{weak}) - (6/14) * \text{Entropy}(\text{strong})$$

$$\text{Entropy}(\text{weak}) = - (6/8) * \log_2(6/8) - (2/8) * \log_2(2/8) = 0.811$$

$$\text{Entropy}(\text{Strong}) = - (3/6) * \log_2(3/6) - (3/6) * \log_2(3/6) = 1.00$$

Vậy:

$$\begin{aligned} \text{Gain}(C, \text{Wind}) &= 0.940 - (8/14) * 0.811 - (6/14) * 1.00 \\ &= 0.048 \end{aligned}$$

Minh họa Thuật giải Quinlan

Kết quả:

$$\text{Gain}(C, \text{Outlook}) = 0.246$$

$$\text{Gain}(C, \text{Temperature}) = 0.029$$

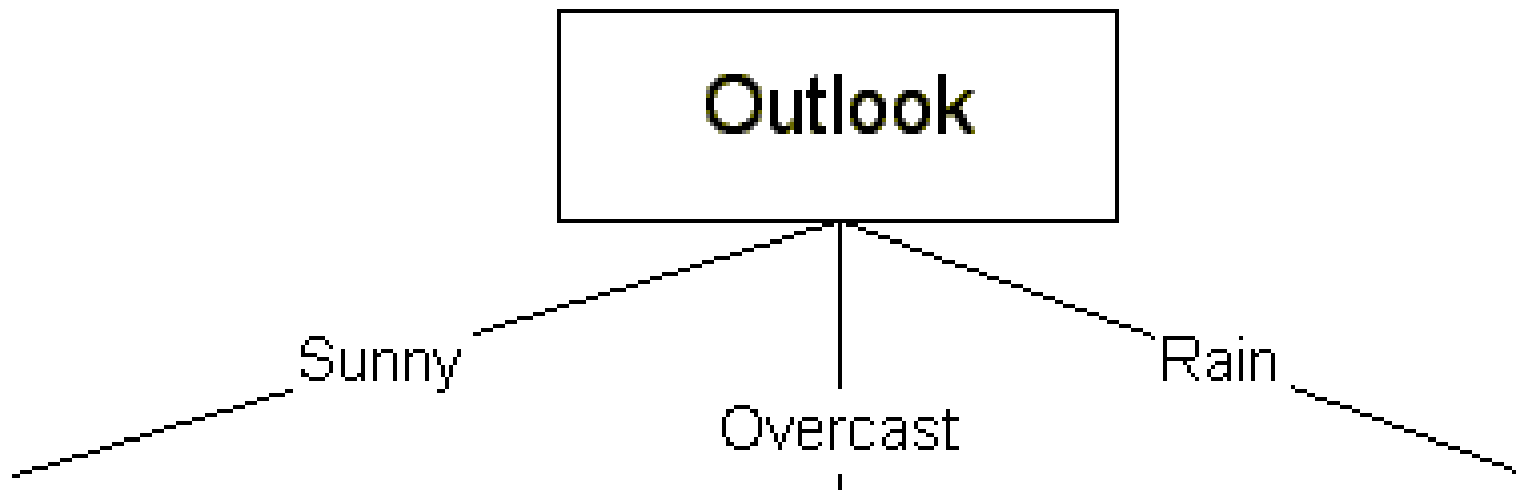
$$\text{Gain}(C, \text{Humidity}) = 0.151$$

$$\text{Gain}(C, \text{Wind}) = 0.048$$

Thuộc tính Outlook có Gain cao nhất. Do đó, nó được dùng làm nút gốc.

Minh họa Thuật giải Quinlan

Outlook có 3 giá trị, cần phân 3 nhánh *sunny*, *overcast*, *rain*.

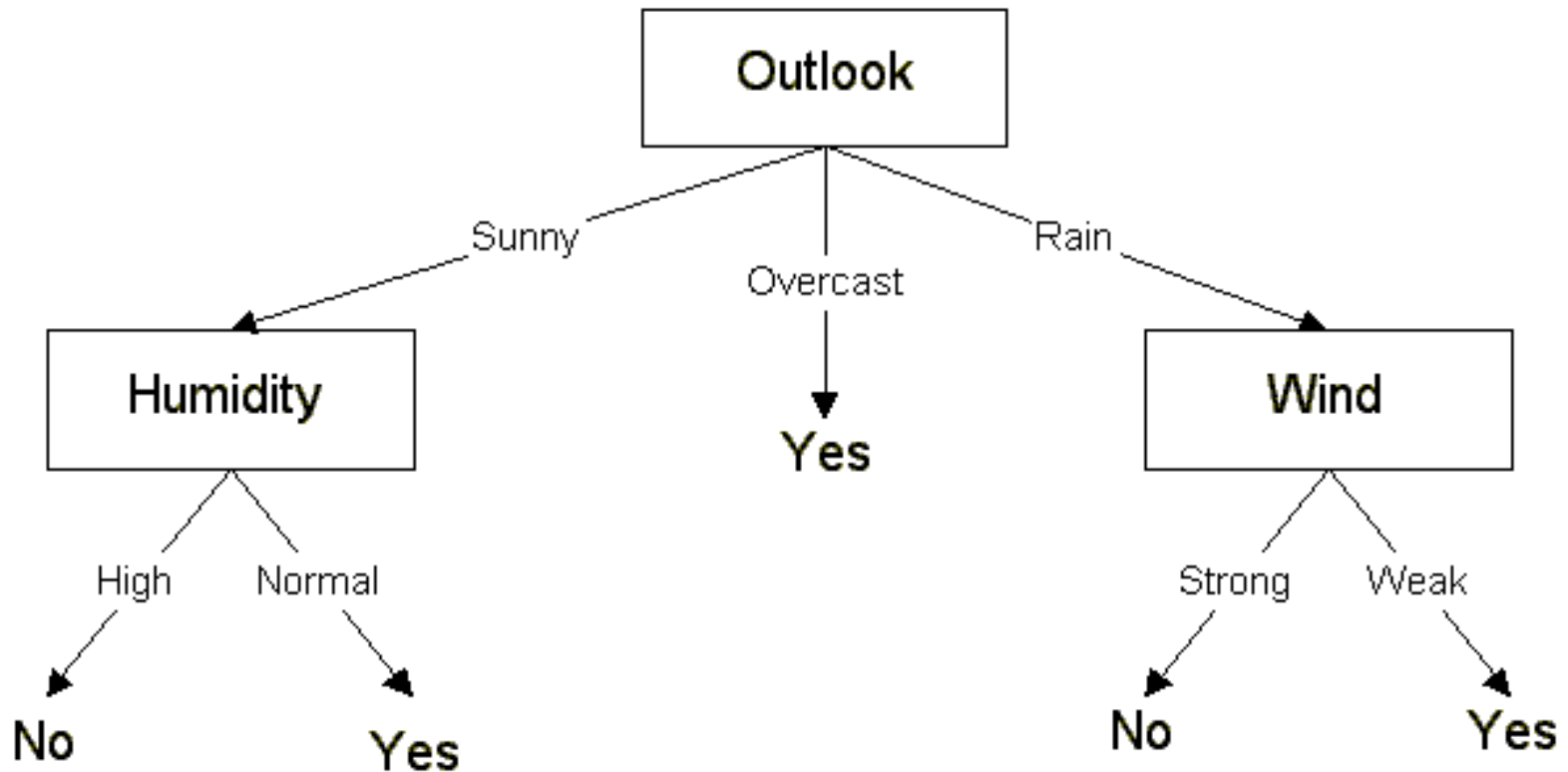


Minh họa Thuật giải Quinlan

- Tiếp tục phân nhánh. Xét nút Sunny. Chỉ còn các thuộc tính Humidity, Temperature, Wind.
- $C_{\text{sunny}} = \{D1, D2, D8, D9, D11\} = 5$ trường hợp với outlook = sunny
 - $\text{Gain}(C_{\text{sunny}}, \text{Humidity}) = 0.970$
 - $\text{Gain}(C_{\text{sunny}}, \text{Temperature}) = 0.570$
 - $\text{Gain}(C_{\text{sunny}}, \text{Wind}) = 0.019$

Minh hoạ Thuật giải Quinlan

- Thuộc tính Humidity có Gain cao nhất. Lấy nó làm nút phân nhánh tiếp theo. Quá trình lặp lại



Minh hoạ Thuật giải Quinlan

Cây quyết định thể hiện các luật, là tri thức rút ra từ bảng trên.

1. IF outlook = sunny AND humidity = high THEN
playball = no
2. IF outlook = rain AND humidity = high THEN
playball = no
3. IF outlook = rain AND wind = strong THEN
playball = yes
4. IF outlook = overcast THEN playball = yes
5. IF outlook = rain AND wind = weak THEN
playball = yes

Ưu điểm của giải thuật Quinlan

1. Dùng cửa sổ hay phương pháp lọc các ngoại lệ. Giảm số ví dụ cần xử lí, tập trung vào các ví dụ tốt
2. Chọn thuộc tính có độ phân biệt cao nhất ở mỗi bước, là một heuristic cho phép tăng hiệu quả của hệ thống.

Nhược điểm của giải thuật Quinlan

1. Thuật toán không biết phát hiện thuộc tính không liên quan đến kết luận.
 - Ví dụ nếu trong bảng trên có cả cột chứa thông tin về “ăn quả vào ngày thứ mấy trong tuần” thì thuật toán vẫn xử lí mối tương quan giữa việc “ăn quả vào ngày thứ mấy trong tuần” và kết luận “ăn được” / “độc” một cách giả tạo.

Nhược điểm của giải thuật Quinlan

- Thuật toán chỉ xét mỗi lần một thuộc tính. Nếu có 2 thuộc tính có entropy trùng khớp nhau thì nó vẫn xét riêng mặc dù đúng ra nên xem xét chúng cùng với nhau.

Thuật toán ID3 sẽ không thể phát hiện ra luật đơn giản là :

if $X = Y$ then outcome = yes
else outcome = no

<i>Outcome</i>	<i>X</i>	<i>Y</i>
yes	3	3
no	2	1
yes	4	4
no	2	4
no	1	3
yes	1	1
yes	2	2
no	2	3

Nhược điểm của giải thuật Quinlan

3. Khi quy nạp rút ra luật từ một tập ví dụ với nhiều khả năng kết luận khác nhau (n lớn chứ không phải là 2) thì thuật toán quá nhạy cảm với những thay đổi tầm thường trong tập ví dụ.
4. Thuật toán không xử lí các luật không chắc chắn, dữ liệu không chắc chắn.
5. Nhiều thí dụ tương tự cũng chỉ như một thí dụ.
6. Không xử lí được các thí dụ mâu thuẫn nhau

Bài tập 1

- Một người lạc trên hoang đảo. Để sống, cần phải thử xem loại quả nào ăn được loại nào độc. Sau nhiều lần thử, sẽ lập được bảng thống kê sau

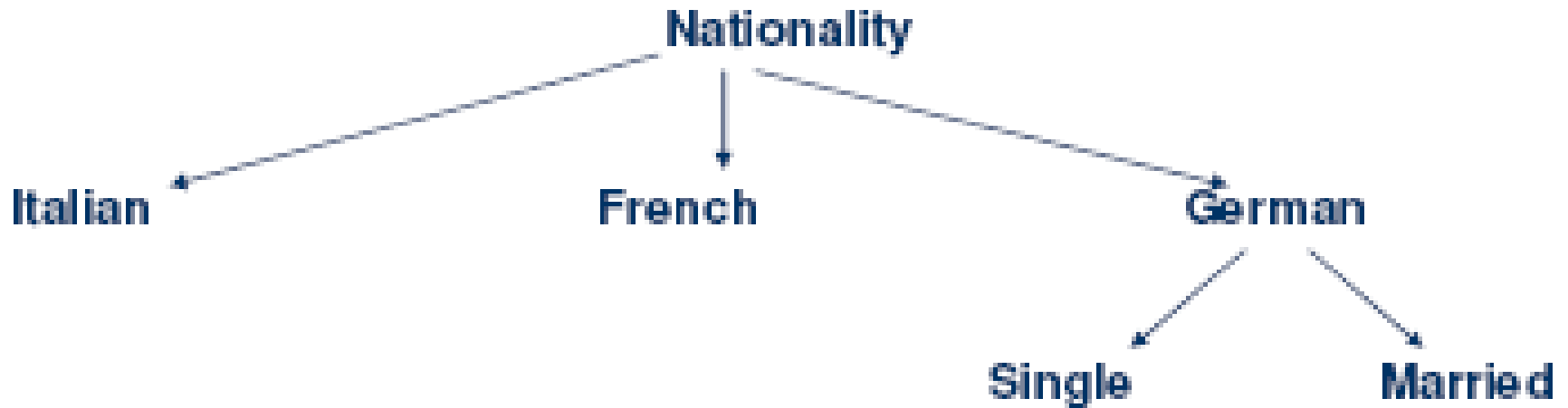
Xây dựng các tập luật dùng thuật toán Quinlan

<i>Conclusion</i>	<i>Skin</i>	<i>Colour</i>	<i>Size</i>	<i>Flesh</i>
safe	Hairy	brown	large	Hard
safe	hairy	green	large	Hard
dangerous	smooth	red	large	Soft
safe	hairy	green	large	Soft
safe	hairy	red	small	Hard
safe	smooth	red	small	Hard
safe	smooth	brown	small	Hard
dangerous	hairy	green	small	Soft
dangerous	smooth	green	small	Hard
safe	hairy	red	large	Hard
safe	smooth	brown	large	Soft
dangerous	smooth	green	small	soft
safe	hairy	red	small	soft
dangerous	smooth	red	large	hard
safe	smooth	red	small	hard
dangerous	hairy	green	small	hard

Bài tập 2. Xây dựng tập luật để suy diễn ra Conclusion sử dụng thuật toán Quinlan

STT	Size	Nationality	Family	Conclusion
1	Small	German	Single	A
2	Large	French	Single	A
3	Large	German	Single	A
4	Small	Italian	Single	B
5	Large	German	Married	B
6	Large	Italian	Single	B
7	Large	Italian	Married	B
8	Small	German	Married	B

Bài tập 2. Kết quả



Rule 1: If (Nationality IS Italian) then (Conclusion IS B)

Rule 2: If (Nationality IS French) then (Conclusion IS A)

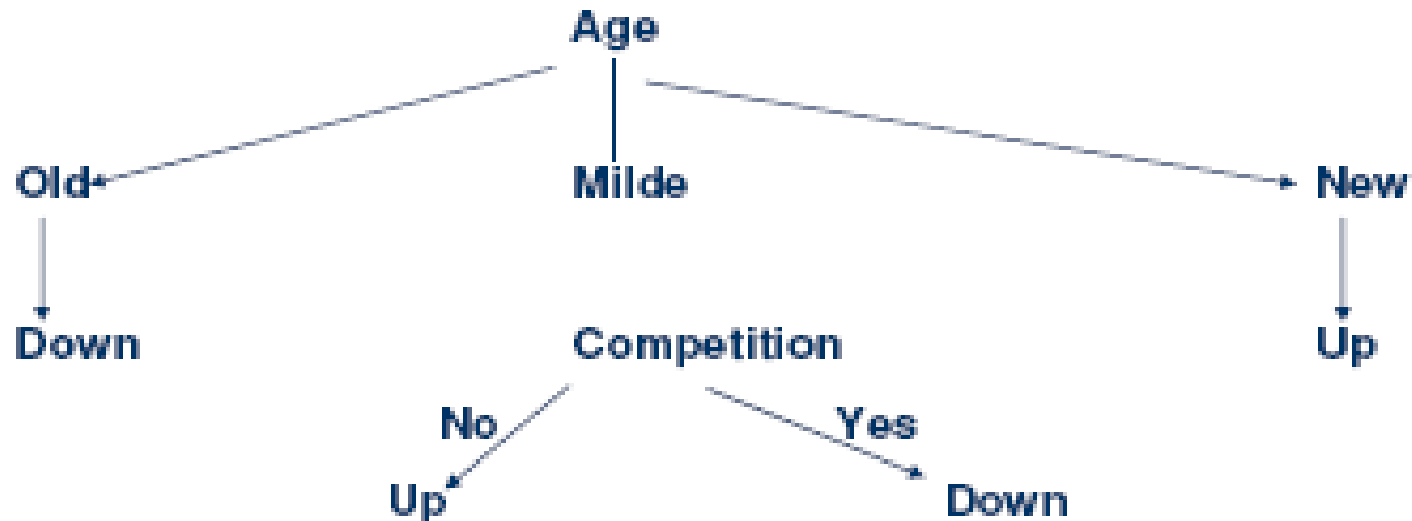
**Rule 3: If (Nationality IS German) AND (Family IS Single)
then (Conclusion IS A)**

**Rule 4: If (Nationality IS German) AND (Family IS Married)
then (Conclusion IS B)**

Bài tập 3. Xây dựng tập luật suy diễn ra Profit sử dụng thuật toán Quinlan

Stt	Age	Competition	Type	Profit
1	Old	No	Software	Down
2	Midle	Yes	Software	Down
3	Midle	No	Hardware	Up
4	Old	No	Hardware	Down
5	New	No	Hardware	Up
6	New	No	Software	Up
7	Midle	No	Software	Up

Kết quả



Rule 1: If (Age IS Old) then (Profit IS Down)

Rule 2: If (Age IS New) then (Profit IS Up)

**Rule 3: If (Age IS Milde) And (Competition IS No)
then (Profit IS Up)**

**Rule 4: If (Age IS Milde) And (Competition IS Yes)
then (Profit IS Down)**

4.7. Thuật giải ILA

(Inductive Learning Algorithm)

Xác định dữ liệu

1. Tập mẫu được liệt kê trong một bảng, với mỗi dòng tương ứng một mẫu, và mỗi cột thể hiện một thuộc tính trong mẫu.
2. Tập mẫu có m mẫu, mỗi mẫu gồm k thuộc tính và trong đó có một thuộc tính quyết định. Tổng số n các giá trị của thuộc tính quyết định chính là số lớp của tập mẫu.
3. Tập luật R có giá trị khởi tạo là \emptyset
4. Tất cả các dòng trong bảng ban đầu chưa được đánh dấu (kiểm tra).

4.7. Thuật giải ILA

- *Bước 1:* Chia bảng m mẫu ban đầu thành n bảng con. Mỗi bảng con ứng với một giá trị của thuộc tính phân lớp của tập mẫu.
(* thực hiện các bước 2 đến 8 cho mỗi bảng con*)
- *Bước 2:* Khởi tạo biến đếm kết hợp thuộc tính j , $j=1$.
- *Bước 3:* Với mỗi bảng con đang khảo sát, phân chia danh sách các thuộc tính theo các tổ hợp phân biệt, mỗi tổ hợp ứng với j thuộc tính phân biệt.

4.7. Thuật giải ILA

- *Bước 4:* Với mỗi tổ hợp các thuộc tính, tính số lượng các giá trị thuộc tính xuất hiện theo cùng tổ hợp thuộc tính trong các dòng chưa được đánh dấu của bảng con đang xét (mà đồng thời không xuất hiện với tổ hợp thuộc tính này trên các bảng còn lại). Gọi tổ hợp đầu tiên (trong bảng con) có số lần xuất hiện nhiều nhất là *tổ hợp lớn nhất*.
- *Bước 5:* Nếu tổ hợp lớn nhất bằng \emptyset , tăng j lên 1 và quay lại bước 3.
- *Bước 6:* Đánh dấu các dòng thoả tổ hợp lớn nhất của bảng con đang xử lý theo lớp.

4.7. Thuật giải ILA

- *Bước 7:* Thêm luật mới vào tập luật R , với vế trái là tập các giá trị của thuộc tính ứng với tổ hợp lớn nhất (kết hợp các thuộc tính bằng toán tử AND) và vế phải là giá trị thuộc tính quyết định tương ứng.
- *Bước 8:* Nếu tất cả các dòng đều đã được đánh dấu phân lớp, tiếp tục thực hiện từ bước 2 cho các bảng con còn lại. Ngược lại (nếu chưa đánh dấu hết các dòng) thì quay lại bước 4. Nếu tất cả các bảng con đã được xét thì kết thúc, kết quả thu được là tập luật cần tìm.

Minh họa thuật giải ILA

Mẫu số	Size	Color	Shape	Decision
1	medium	blue	brick	yes
2	small	red	wedge	no
3	small	red	sphere	yes
4	large	red	wedge	no
5	large	green	pillar	yes
6	large	red	pillar	no
7	large	green	sphere	yes

Minh họa thuật giải ILA (tiếp): Bước 1

<i>Bảng con 1</i>					
Mẫu số cũ, mới		Size	Color	Shape	Decision
1	1	medium	blue	brick	yes
3	2	small	red	sphere	yes
5	3	large	green	pillar	yes
7	4	large	green	sphere	yes
<i>Bảng con 2</i>					
Mẫu số cũ, mới		Size	Color	Shape	Decision
2	1	small	red	wedge	no
4	2	large	red	wedge	no
6	3	large	red	pillar	no

Bảng 4.2. Với $n=2$, Chia thành hai bảng con theo thuộc tính Decision

Minh họa thuật giải ILA (tiếp)

- Áp dụng bước 2 của thuật giải vào bảng con thứ nhất trong bảng trên. Với $j=1$, danh sách các tổ hợp thuộc tính gồm có {Size}, {Color}, và {Shape}.
- Với tổ hợp {Size}, giá trị thuộc tính "medium" xuất hiện trong bảng con thứ nhất nhưng không có trong bảng con thứ hai, do đó giá trị tổ hợp lớn nhất là "medium". Bởi vì các giá trị thuộc tính "small" và "large" xuất hiện trong cả hai bảng con, nên không được xét trong bước này. Với tổ hợp {Size}, giá trị thuộc tính "medium" chỉ bằng 1

Minh họa thuật giải ILA (tiếp)

- Xét tiếp cho tổ hợp {Color} thì giá trị tổ hợp lớn nhất là bằng 2, ứng với thuộc tính "green", còn thuộc tính "blue" là bằng 1.
- Tương tự, với tổ hợp {Shape}, ta có "brick" xuất hiện một lần, và "sphere" hai lần. Đến cuối bước 4, ta có tổ hợp {Color} với thuộc tính "green" và {Shape} với thuộc tính "sphere" đều có số lần xuất hiện lớn nhất là 2. Thuật toán mặc định chọn trường hợp thứ nhất để xác định luật tổ hợp lớn nhất. Dòng 3 và 4 được đánh dấu đã phân lớp, ta có luật dẫn như sau:

Rule 1: IF color IS green THEN decision IS yes

Minh họa thuật giải ILA (tiếp)

- Tiếp tục thực hiện bước 4 đến 8 cho các mẫu còn lại (chưa đánh dấu) trong bảng con này (tức dòng 1 và 2). Áp dụng tương tự như trên, ta thấy giá trị thuộc tính "medium" của {Size}, "blue" của "Color", "brick" và "sphere" của {Shape} đều xuất hiện một lần. Bởi vì số lần xuất hiện này giống nhau, thuật giải áp dụng luật mặc định chọn trường hợp đầu tiên. Ta có thêm luật sau:

Rule 2: IF size IS medium THEN decision IS yes

Đánh dấu cho dòng 1 trong bảng con thứ nhất. Tiếp tục áp dụng bước 4 đến 8 trên dòng còn lại (tức dòng 2). Giá trị thuộc tính "sphere" của {Shape} xuất hiện một lần, ta có luật thứ ba:

:Rule 3: IF shape IS sphere THEN decision IS yes

Dòng 2 được đánh dấu. Như vậy, tất cả các dòng trong bảng con 1 đã được đánh dấu, ta chuyển qua xử lý tiếp bảng con 2.

Minh họa thuật giải ILA (tiếp)

• Thuộc tính "wedge" của {Shape} xuất hiện hai lần trong dòng 1 và 2 của bảng con này. Đánh dấu các dòng này với luật dẫn thứ tư sau:

Rule 4: IF shape IS wedge THEN decision IS no

• Với dòng còn lại (tức dòng 3) của bảng con 2, ta có thuộc tính {Size} với giá trị "large" có xuất hiện trong bảng con 1. Do đó, theo thuật giải, ta loại bỏ trường hợp này. Tương tự như vậy cho giá trị "red" của {Color} và "pillar" của {Shape}. Khi đó, ILA tăng j lên 1, và khởi tạo các tổ hợp 2 thuộc tính là {Size và Color}, {Size và Shape}, và {Color và Shape}. Các tổ hợp thứ nhất và thứ ba thoả mãn điều kiện không xuất hiện trong bảng con 1 với các cặp thuộc tính hiện có của dòng này. Theo luật mặc định, ta chọn luật theo trường hợp thứ nhất. Đánh dấu dòng này, ta có thêm luật dẫn thứ 5:

Rule 5: IF size IS large AND color IS red THEN decision IS no

Minh họa thuật giải ILA (*tiếp*)

Rule 1: IF color IS green THEN decision IS yes

Rule 2: IF size IS medium THEN decision IS yes

Rule 3: IF shape IS sphere THEN decision IS yes

Rule 4: IF shape IS wedge THEN decision IS no

Rule 5: IF size IS large AND color IS red THEN decision IS no

Đánh giá thuật giải:

- Số lượng các luật thu được xác định mức độ thành công của thuật giải. Đây chính là mục đích chính của các bài toán phân lớp thông qua một tập mẫu học. Ngoài ra, để đánh giá các hệ học quy nạp là khả năng hệ thống có thể phân lớp các mẫu được đưa vào sau này.
- Thuật giải ILA được đánh giá mạnh hơn hai thuật giải về phương pháp học quy nạp trước đây là ID3 và AQ).

Bài tập:
**Xây dựng tập luật cho kết luận
sử dụng thuật toán ILA**

(Bảng 7.1. trang 129 giáo trình)

STT	Tên người	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
1	Hoa	Đen	Tầm thước	Nhẹ	Không	Bị rám
2	Lan	Đen	Cao	Vừa phải	Có	Không
3	Xuân	Râm	Thấp	Vừa phải	Có	Không
4	Hạ	Đen	Thấp	Vừa phải	Không	Bị rám
5	Thu	Bạc	Tầm thước	Nặng	Không	Bị rám
6	Đông	Râm	Cao	Nặng	Không	Không
7	Mơ	Râm	Tầm thước	Nặng	Không	Không
8	Đào	Đen	Thấp	Nhẹ	Có	Không

Chương 5. HỆ THỐNG MỜ CHO CÁC BIẾN LIÊN TỤC

Trần Nguyên Hương

Chương 5: Hệ thống mờ cho các biến liên tục

5.1 Các khái niệm

1. Tập rõ và hàm đặc trưng

- Ngôn ngữ tự nhiên và logic mờ.
- Tập rõ (crisp set): Gọi A là một tập hợp rõ, một phần tử x có thể có $x \in A$ hoặc $x \notin A$. Có thể sử dụng hàm χ để mô tả khái niệm thuộc về. Nếu $x \in A$, $\chi(x) = 1$, ngược lại nếu $x \notin A$, $\chi(x) = 0$. Hàm χ được gọi là hàm đặc trưng của tập hợp A .
- Tập mờ và hàm thành viên: Khác với tập rõ, khái niệm thuộc về được mở rộng nhằm phản ánh mức độ x là phần tử của tập mờ A . Một tập mờ (fuzzy set): A được đặc trưng bằng hàm thành viên μ và cho x là một phần tử, $\mu_A(x)$ phản ánh mức độ x thuộc về A .
- Một tập mờ A trong tập vũ trụ U được xác định bởi hàm:

$$\mu_A: U \rightarrow [0, 1]$$

5.1 Các khái niệm về Logic mờ

- Ví dụ về tập mờ:
 - High
 - Young
 - Số gần 7
 - Tốc độ nhanh

Biểu diễn tập mờ:

1. Nếu tập vũ trụ U là rời rạc và hữu hạn thì tập mờ A trong U được biểu diễn:

$$A = \sum_{x \in U} \frac{\mu_A(x)}{x}$$

Ví dụ: Cho $U = \{a, b, c, d\}$. Ta có thể xác định một tập mờ A như sau:

$$A = \frac{0.3}{a} + \frac{0.5}{b} + \frac{0}{c} + \frac{0.7}{d}$$

5.1 Các khái niệm về Logic mờ (tiếp)

2. Nếu tập vũ trụ U là liên tục thì tập mờ A trong U được biểu diễn:

$$A = \int_{x \in U} \frac{\mu_A(x)}{x} dx$$

Ví dụ: Tập mờ $A = \{\text{"Số gần 2"}\}$ có thể xác định hàm thuộc như sau:

$$\mu_A(x) = e^{-\alpha \cdot 2^2}$$

$$A = \int_{x \in U} \frac{e^{-\alpha \cdot 2^2}}{x} dx$$

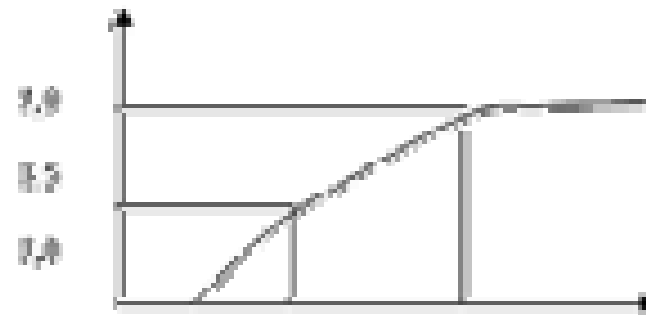
5.1 Các khái niệm về Logic mờ (tiếp)

Ghi chú: Đồ thị hàm thuộc cho tập mờ $A = \{\text{"Số gần 2"}\}$ có thể xác định cách khác như sau:

$$\mu_A(x) = \begin{cases} 0 & \text{nếu } x < 1 \\ x-1 & \text{nếu } 1 \leq x < 2 \\ 1 & \text{nếu } x = 2 \\ -x+3 & \text{nếu } 2 < x \leq 3 \\ 0 & \text{nếu } x > 3 \end{cases}$$

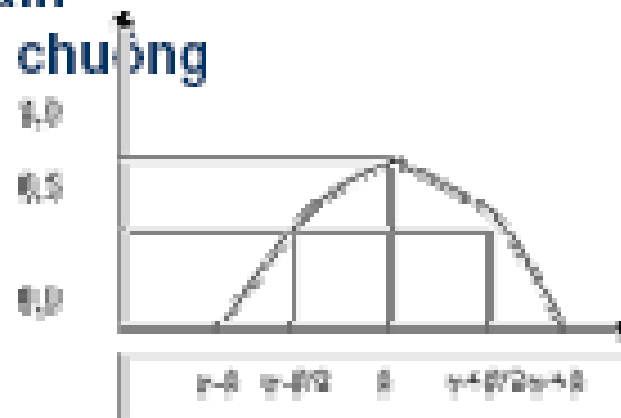
5.2 Các dạng của hàm thành viên

- Các hàm thành viên của tập mờ có 3 dạng cơ bản là: dạng tăng, dạng giảm và dạng chuông
- Dạng S tăng



Hình 4.2. Hàm S tăng

- b) Dạng S giảm
- c) Dạng hình chuông



Hình 4.3. Hàm dạng chuông

5.3 Biến ngôn ngữ (Linguistic Variable)

- Logic mờ liên quan đến lập luận trên các thuật ngữ mờ và mờ hồ trong ngôn ngữ tự nhiên của con người.
- Biến nhận các từ trong ngôn ngữ tự nhiên làm giá trị gọi là biến ngôn ngữ.
- Biến ngôn ngữ được xác định bởi bộ bốn (x, T, U, M) :
 - X là tên biến. Ví dụ: “nhiệt độ”, “tốc độ”, “áp suất”, ...
 - T là tập các từ (các giá trị ngôn ngữ) mà x có thể nhận. Ví dụ: x là “tốc độ” thì T có thể là $T = \{\text{chậm, vừa, nhanh}\}$
 - U là miền giá trị mà x có thể nhận. Ví dụ, nếu x là “tốc độ” của xe máy thì $U = [0 .. 120 \text{ km/h}]$
 - M là luật ngữ nghĩa, ứng với mỗi từ $t \in T$ với một tập mờ A .

5.4 Biến ngôn ngữ (tiếp)

Biến ngôn ngữ	Các giá trị điển hình
Nhiệt độ Độ cao Tốc độ	Nóng, lạnh Thấp, trung bình, cao Chậm, vừa, nhanh

Ví dụ: Cho x là tốc độ, $T = \{\text{chậm, vừa, nhanh}\}$, các từ “chậm”, “vừa”, “nhanh” được xác định bởi các tập mờ trong hình sau:



5.5 Gia tử

- Gia tử làm mờ hồ thêm các câu như: *rất, hơi, có vẻ, ...*

1. **Rất:** $\mu_{\text{Rất}(A)}(x) = (\mu_A(x))^2$

Ví dụ: Tập mờ gồm những người rất cao

2. **Có gian / một ít:** $\mu_{\text{Có gian}(A)}(x) = (\mu_A(x))^{0.5}$

Ví dụ: A là tập những người tầm thước thì Có gian A là tập những người thiên về cao và thấp trong những người tầm thước.

3. **Nhấn mạnh/ thực sự là:**

$$\mu_{\text{Nhấn mạnh}(A)}(x) = 2(\mu_A(x))^2 \quad \text{nếu } 0 \leq \mu_A(x) \leq 0.5$$

$$\mu_{\text{Nhấn mạnh}(A)}(x) = 1 - 2(1 - \mu_A(x))^2 \quad \text{nếu } 0.5 \leq \mu_A(x) \leq 1$$

Ví dụ: Sau khi dùng phép toán này với tập mờ cao ta được tập những người thực sự cao

4. **Mạnh mẽ / rất rất:** $\mu_{\text{Rất rất}(A)}(x) = (\mu_A(x))^n$

Ví dụ: Sau khi dùng phép toán này với tập mờ cao ta được tập những người thực sự cao

5.6 Các phép toán trên tập mờ

Cho ba tập mờ A, B, C với $\mu_A(x), \mu_B(x), \mu_C(x)$

- $C=A \cap B: \mu_C(x) = \min(\mu_A(x), \mu_B(x))$
- $C=A \cup B: \mu_C(x) = \max(\mu_A(x), \mu_B(x))$
- $C=\neg A: \mu_C(x) = 1 - \mu_A(x)$

Xét tập mờ "cao" và "thấp" về chiều cao của người:

$$\text{Cao} = 0.0/1.5 + 0.2/1.55 + 0.5/1.60 + 0.8/1.65 + 1.0/1.70$$

$$\text{Thấp} = 1.0/1.5 + 0.8/1.55 + 0.5/1.60 + 0.2/1.65 + 0.0/1.70$$

- $\mu_{\text{Cao} \cap \text{Thấp}}(x) = 0.0/1.5 + 0.2/1.55 + 0.5/1.60 + 0.2/1.65 + 0.0/1.70$
Dùng để chỉ những người tầm thước: giá trị cao nhất ở giữa tập, thấp nhất ở 2 bên
- $\mu_{\text{Cao} \cup \text{Thấp}}(x) = 1.0/1.5 + 0.8/1.55 + 0.5/1.60 + 0.8/1.65 + 1.0/1.70$
Dùng để chỉ những người không tầm thước
- $\mu_{\neg \text{Cao}}(x) = 1.0/1.5 + 0.8/1.55 + 0.5/1.60 + 0.2/1.65 + 0.0/1.70$
Dùng để chỉ những người không cao hay tầm thước hay thấp.

5.7 Suy diễn mờ

Mệnh đề mờ là mệnh đề khẳng định giá trị cho biến ngôn ngữ.

<Miền xác định X> is <tập mờ A>

Ví dụ: Chiều cao là tầm thước

- Logic mờ sử dụng tập mờ trong các mệnh đề mờ.

IF X is A THEN Y is B

Ví dụ: Nếu chiều cao là tầm thước thì trọng lượng là trung bình

- Nếu A và B là tập mờ thì Hệ chuyên gia lưu trữ liên kết (A,B) trong ma trận M (hay ký hiệu R).
- Có thể thể hiện cả A và B như các vector (A, B) thích hợp và đặt quan hệ này vào ma trận M.
 - Ma trận liên kết mờ M ánh xạ tập mờ A sang tập mờ B.
- Hai kỹ thuật suy diễn thông dụng là;
 - Suy diễn Max-Min
 - Suy diễn cực đại.

5.7.1 Nhân ma trận vector mờ

Cho $A = (a_1, a_2, \dots, a_n)$ với $a_i = \mu_A(x_i)$

Cho $B = (b_1, b_2, \dots, b_p)$ với $b_j = \mu_B(y_j)$

Ma trận $M_{n \times p}$ được xác định như sau: $A \circ M = B$

Trong đó $b_j = \text{Max} \{ \text{Min} (a_i, m_{ij}) \} \quad 1 \leq i \leq n$

Ví dụ: Cho $A = (0.2, 0.4, 0.6, 1)$

Cho

$$M = \begin{bmatrix} 0.1 & 0.6 & 0.8 \\ 0.6 & 0.8 & 0.6 \\ 0.8 & 0.6 & 0.5 \\ 0.0 & 0.5 & 0.5 \end{bmatrix}$$

$b_1 = \text{Max} \{ \text{Min}(0.2, 0.1), \text{Min}(0.4, 0.6), \text{Min}(0.6, 0.8), \text{Min}(1, 0.0) \} = 0.6$

Tương tự $b_2 = 0.6$ và $b_3 = 0.5$

Như vậy, $B = (0.6, 0.6, 0.5)$

5.7.2 Suy diễn Max-Min

- Khi biết ma trận A và B hãy tạo ma trận M

- Cho A' hãy tính B'

- **Suy diễn Max-Min:** $m_{ij} = \text{Min}(a_i, b_j)$

Ví dụ: X: nhiệt độ, tập mờ A trên X: "nhiệt độ bình thường"

Y: tốc độ, tập mờ B trên Y: "tốc độ vừa phải"

Giả sử có luật: IF nhiệt độ bình thường THEN tốc độ vừa phải.

Giả sử các tập mờ thể hiện bằng các vector sau:

Nhiệt độ bình thường = (0/100, 0.5/125, 1/150, 0.5/175, 0/200)

Tốc độ vừa phải = (0/10, 0.6/20, 1/30, 0.6/40, 0/50)

Ma trận M được tạo ra như sau:

$$M = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.5 & 0.5 & 0.0 \\ 0.0 & 0.6 & 1.0 & 0.6 & 0.0 \\ 0.0 & 0.5 & 0.5 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Cho tập A' = (0/100, 0.5/125, 0/150, 0/175, 0/200)

Ta có $b_j = \max\{\min(0,0), \min(0.5,0), \min(0,0), \min(0,0), \min(0,0)\} = 0$

B' = (0/10, 0.5/20, 0.5/30, 0.5/40, 0/50)

5.7.3 Suy diễn Tích cực đại

- Dùng phép nhân để tạo các thành phần của ma trận M

Suy diễn Tích cực đại: $m_{ij} = a_i * b_j$

- Sau đó dùng cách tính Max-Min để suy ra B' từ A'

Ví dụ: $A = (0, 0.5, 1, 0.5, 0)$

$B = (0, 0.6, 1, 0.6, 0)$

Ma trận M được tạo ra như sau:

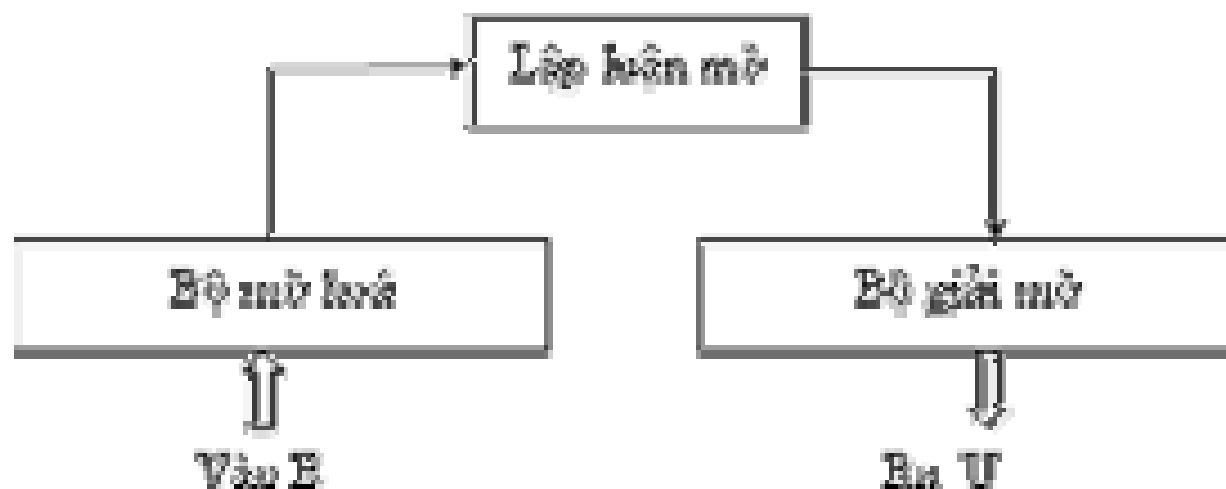
$$M = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.5 & 0.3 & 0.0 \\ 0.0 & 0.6 & 1.0 & 0.6 & 0.0 \\ 0.0 & 0.3 & 0.5 & 0.3 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Cho tập $A' = (0, 0.5, 0, 0, 0)$

Ta có $b_j = \max\{\min(0,0), \min(0.5,0), \min(0,0), \min(0,0), \min(0,0)\} = 0$

$B' = (0, 0.3, 0.5, 0.3, 0)$

5.8 NGUYÊN LÝ XỬ LÝ CÁC BÀI TOÁN MỜ

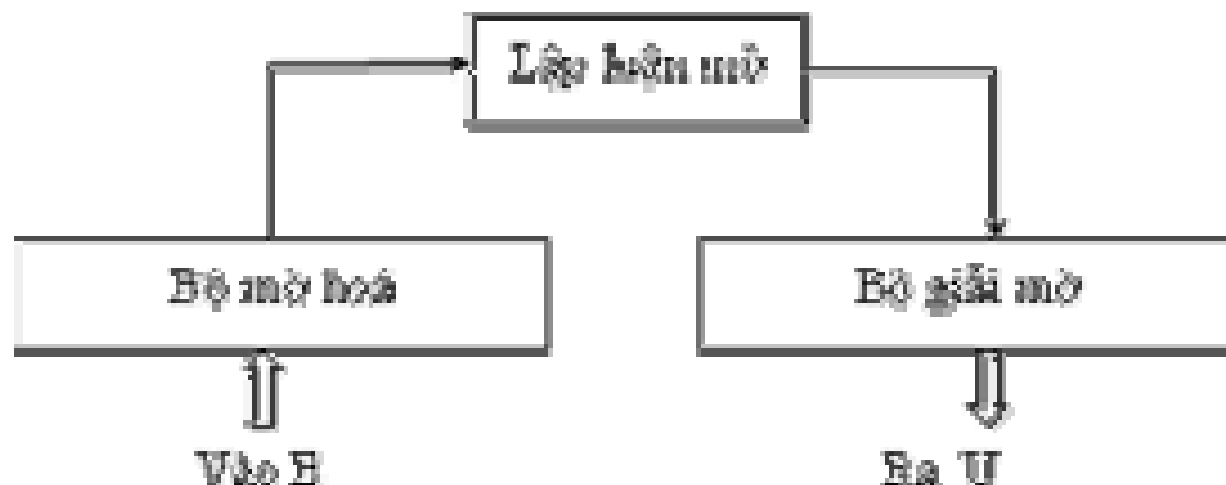


Hình 9.4. Hệ thống mờ

1. Mờ hóa: Chuyển đổi giá trị rõ đầu vào thành vector μ_i
2. Xác định các luật hợp thành và thuật toán xác định giá trị mờ
3. Giải mờ: Phương pháp điểm trọng tâm

$$x_s = \frac{\int x \mu_s(x) dx}{\int \mu_s(x) dx}$$

5.8 NGUYÊN LÝ XỬ LÝ BÀI TOÁN MỜ(tiếp)



Hình 9.4. Hệ thống mờ

Bài toán 1: Dữ liệu Input là các giá trị rõ.

Ví dụ: Xét bài toán mờ xác định bởi các luật sau:

Luật 1: if x is A1 and y is B1 Then z is C1

Luật 2: if x is A2 or y is B2 Then z is C2

Vào: trị x_0, y_0

Ra : trị z_0 tương ứng

5.8 NGUYÊN LÝ XỬ LÝ BÀI TOÁN MỜ(tiếp)

Ứng với tập mờ A_1 ta có hàm thành viên $\mu_{A_1}(x)$

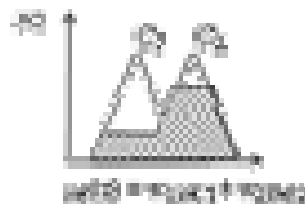
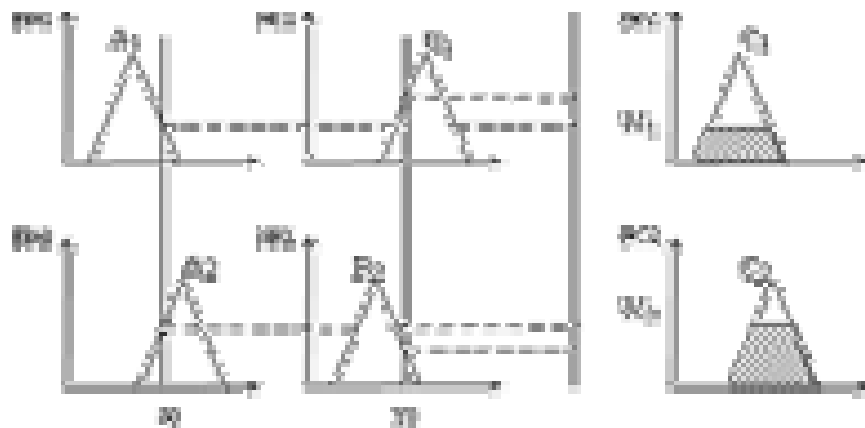
Ứng với tập mờ A_2 ta có hàm thành viên $\mu_{A_2}(x)$

Ứng với tập mờ B_1 ta có hàm thành viên $\mu_{B_1}(y)$

Ứng với tập mờ B_2 ta có hàm thành viên $\mu_{B_2}(x)$

Ứng với tập mờ C_1 ta có hàm thành viên $\mu_{C_1}(x)$

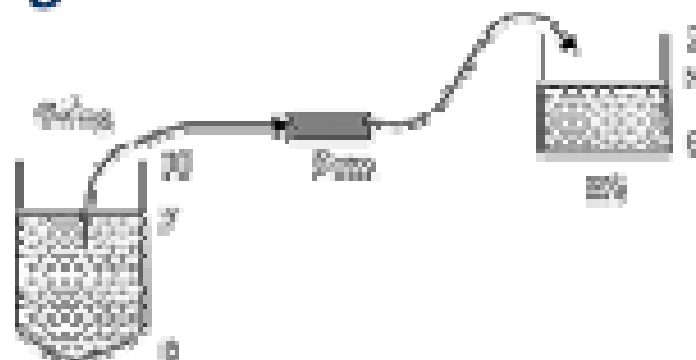
Ứng với tập mờ C_2 ta có hàm thành viên $\mu_{C_2}(x)$



Ví dụ: Giải bài toán điều khiển tự động mờ cho hệ thống bơm nước lấy nước từ giếng.

Trong khi hồ hết nước và trong giếng có nước thì máy bơm tự động bơm.

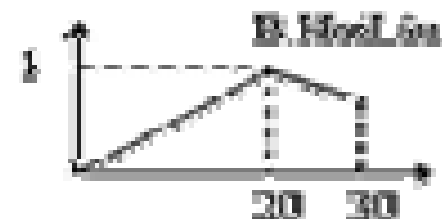
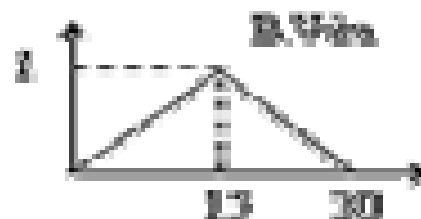
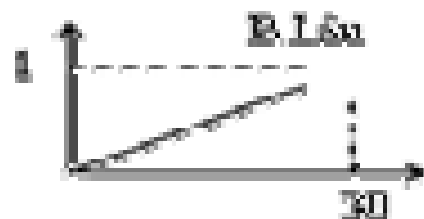
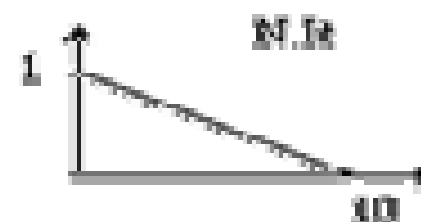
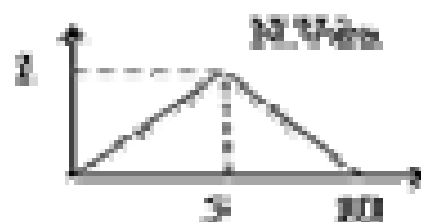
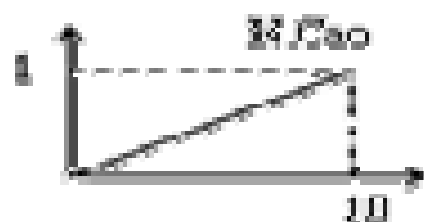
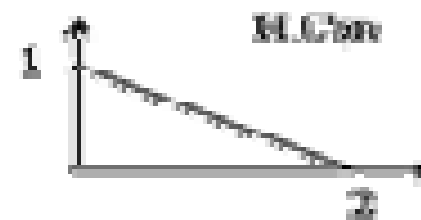
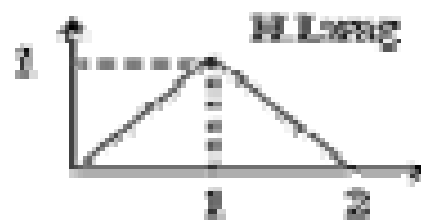
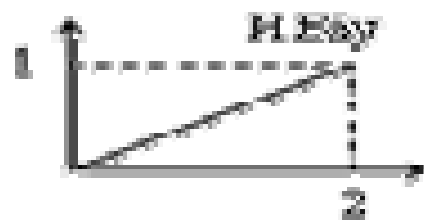
	H.Đầy	H.Lung	H.Cạn
N.Cao	0	B.Vừa	B.Lâu
N.Vừa	0	B.Vừa	B.HoiLâu
N.Ít	0	0	0



- Với biến ngôn ngữ Hồ có các tập mờ hồ đầy (H.Đầy), hồ lung (H.Lung) và hồ cạn (H.Cạn).
- Với biến ngôn ngữ Giếng có các tập mờ nước cao (N.Cao), nước vừa (N.Vừa), nước ít (N.Ít).
- Với biến ngôn ngữ kết luận xác định thời gian bơm sẽ có các tập mờ bơm vừa (B.Vừa), bơm lâu (B.Lâu), bơm hơi lâu (B.HoiLâu).

Ví dụ (tiếp)

Trong đó x chỉ độ sâu của Hồ ($0 \leq x \leq 2$), y chỉ độ sâu của Giếng ($0 \leq y \leq 10$) và z chỉ thời gian bơm ($0 \leq z \leq 30$).



Ví dụ (tiếp)

Từ bảng trên ta có các luật:

- Luật 1: if x is H.Lưng and y is N.Cao Then z is B.Vừa
- Luật 2: if x is H.Chặt and y is N.Cao Then z is B.Lâu
- Luật 3: if x is H.Lưng and y is N.Vừa Then z is B.Vừa
- Luật 4: if x is H.Chặt and y is N.Vừa Then z is B.Hơi lâu

Nếu nhập trị Input $x_0 = 1$ (Độ cao của nước trong hồ),
 $y_0 = 3$ (Độ cao của nước trong giếng)

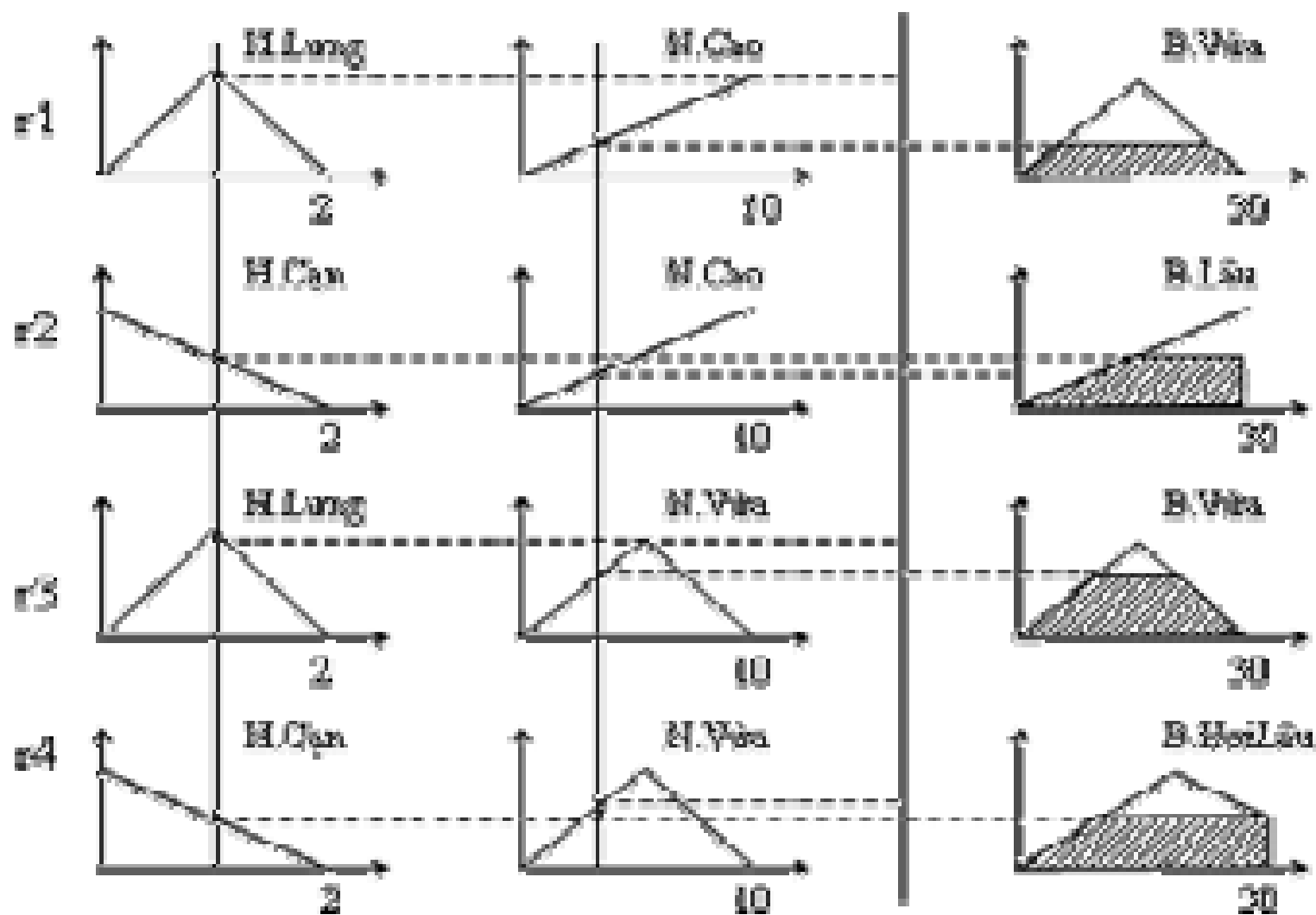
$$\left. \begin{array}{l} \forall x_1 \text{ (H.Lưng)} = 1 \\ \forall x_2 \text{ (N.Cao)} = 3 \end{array} \right\} \Rightarrow \forall z_1 = \min \{1, 3 \wedge 1\}$$

$$\left. \begin{array}{l} \forall x_1 \text{ (H.Chặt)} = 0.3 \\ \forall x_2 \text{ (N.Cao)} = 3 \end{array} \right\} \Rightarrow \forall z_2 = \min \{0.3, 3 \wedge 3\} = 0.3$$

$$\left. \begin{array}{l} \forall x_1 \text{ (H.Lưng)} = 1 \\ \forall x_2 \text{ (N.Vừa)} = 3 \end{array} \right\} \Rightarrow \forall z_3 = \min \{1, 3 \wedge 3\} = 3$$

$$\left. \begin{array}{l} \forall x_1 \text{ (H.Chặt)} = 0.3 \\ \forall x_2 \text{ (N.Vừa)} = 3 \end{array} \right\} \Rightarrow \forall z_4 = \min \{0.3, 3 \wedge 3\} = 0.3$$

Ví dụ (tiếp)



Ví dụ (tiếp)

Các W_i gọi là các trọng số của luật thứ i

Theo lý thuyết hàm thành viên của kết luận cho bởi công thức:

$$\mu_C(z) = \sum_{i=1}^N W_i \mu_{K_i}(z)$$

$$\mu_C(z) = W_1 \cdot B.Vừa(z) + W_2 \cdot B.Lâu(z) + W_3 \cdot B.Vừa(z) + W_4 \cdot B.Hơi Lâu(z)$$

$$\mu_C(z) = 3/10 \cdot B.Vừa(z) + 0.5 \cdot B.Lâu(z) + 3/5 \cdot B.Vừa(z) + 0.5 \cdot B.Hơi Lâu(z)$$

Bước tiếp theo là ta phải giải mờ từ hàm thành viên của kết luận bằng cách tính trọng tâm của hàm $\mu_C(z)$

Moment $\mu_C(z)$ là $\int_V \mu_C(z) dz$

$$\text{và } z_0 = \frac{\int_V z \mu_C(z) dz}{\int_V \mu_C(z) dz} = 16.06$$

Vậy Defuzzy(z) = 16.06

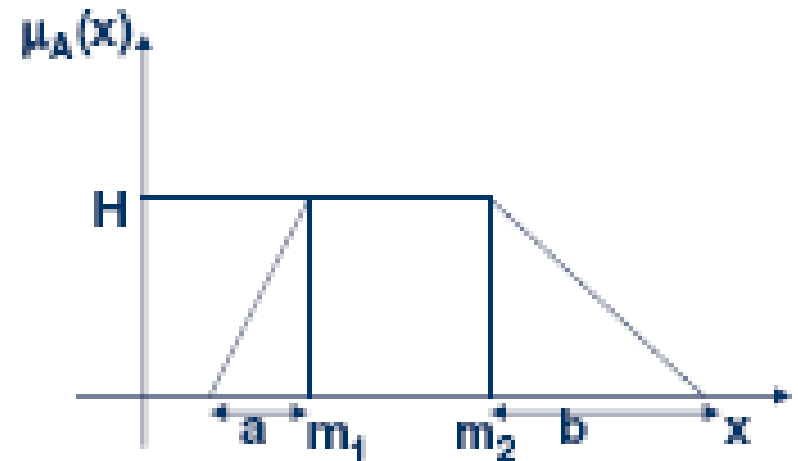
Do đó nếu mực nước trong hồ và giếng là 1m và 3m thì thời gian cần bơm là 16 phút và 06 giây.

Khử mờ: Lấy điểm trọng tâm

$$\begin{aligned}x_0 &= \frac{\int x \mu_A(x) dx}{\int \mu_A(x) dx} = \frac{\int x \sum_{i=1}^n \mu_{v_i}(x) dx}{\int \sum_{i=1}^n \mu_{v_i}(x) dx} \\ &= \frac{\sum_{i=1}^n \int x \mu_{v_i}(x) dx}{\sum_{i=1}^n \int \mu_{v_i}(x) dx}\end{aligned}$$

Tính các tích phân

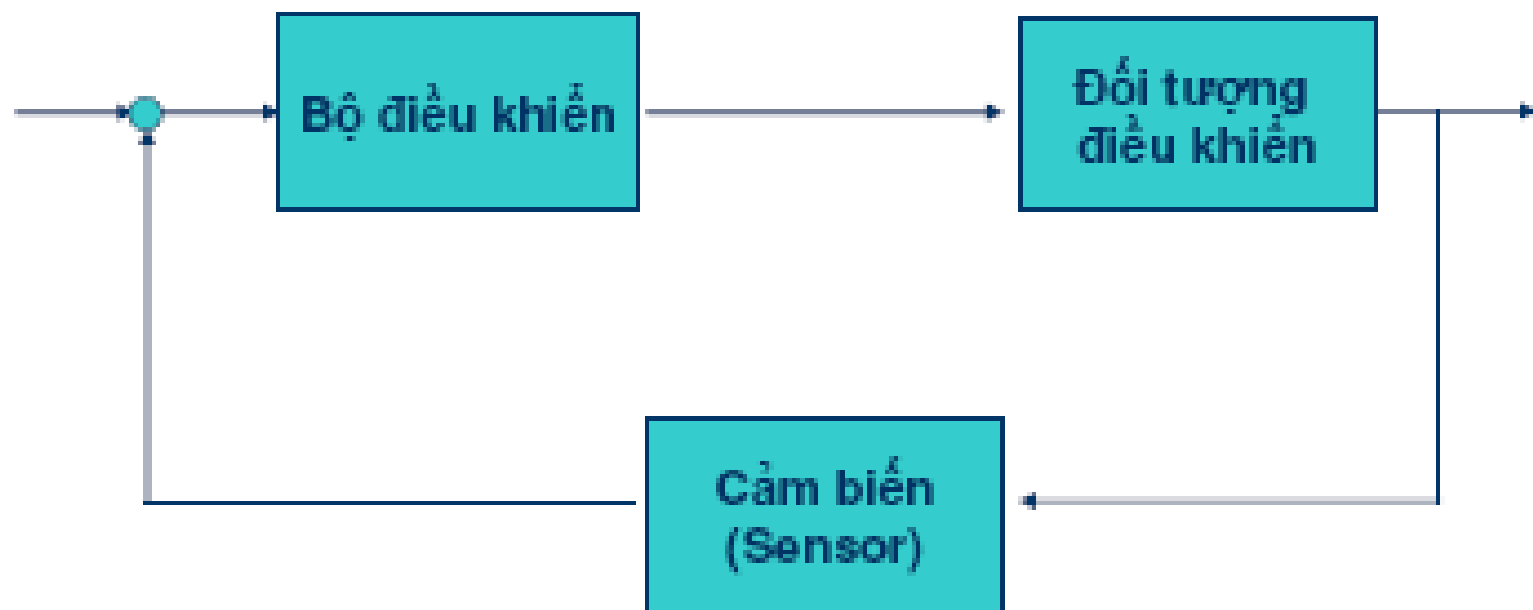
$$x_0 = \frac{\int_v x \mu_A(x) dx}{\int_v \mu_A(x) dx}$$



$$\int_v x \mu_A(x) dx = \frac{H}{6} (3m_2^2 - 3m_1^2 + b^2 - a^2 + 3m_2b + 3m_1a)$$

$$\int_v \mu_A(x) dx = \frac{H}{2} (2m_2 - 2m_1 + a + b)$$

4.6.8 NGUYÊN LÝ XỬ LÝ CÁC BÀI TOÁN MỜ (tiếp)



Bộ điều khiển mờ:

SISO: Single Input, Single Output

MIMO: Multi Input, Multi Output

SIMO: Single Input, Multi Output

MISO: Multi Input, Single Output

4.6.8 NGUYÊN LÝ XỬ LÝ CÁC BÀI TOÁN MỜ (tiếp)

Bộ điều khiển mờ:

SISO: If (A is A_1) Then (B is B_1)

...

If (A is A_n) Then (B is B_n)

MIMO: If (A_1 is A_{11}) and ... and (A_m is A_{m1}) Then (B_1 is B_{11}) and ... and (B_s is B_{1s})

...

If (A_1 is A_{n1}) and ... and (A_m is A_{n1}) Then (B_1 is B_{n1}) and ... and (B_s is B_{ns})

MISO: If (A_1 is A_{11}) and ... and (A_m is A_{m1}) Then (B is B_1)

...

If (A_1 is A_{n1}) and ... and (A_m is A_{n1}) Then (B is B_n)

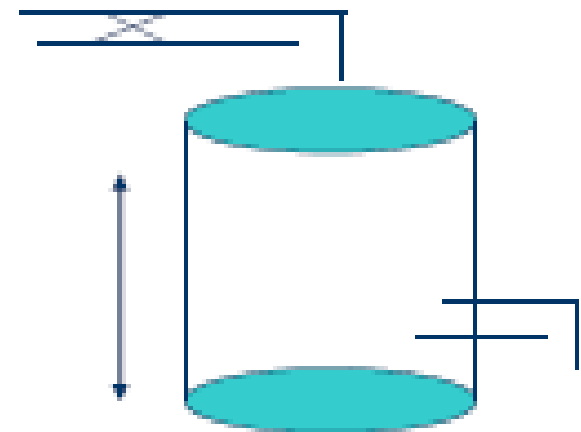
SIMO: If (A is A_1) Then (B_1 is B_{11}) and ... and (B_s is B_{1s})

...

If (A is A_n) Then (B_1 is B_{n1}) and ... and (B_s is B_{ns})

Bài tập: Điều khiển mực nước

Bài toán điều khiển mực nước: Không phụ thuộc vào lượng nước chảy ra khỏi bình, cần phải điều chỉnh van cho lượng nước chảy vào bình vừa đủ để sao cho mực nước h trong bình là luôn không đổi.



Giả sử bộ điều khiển là con người, sẽ có các nguyên tắc sau:

R1: Nếu mực nước là thấp nhiều thì van ở mức độ mở to

R2: Nếu mực nước là thấp ít thì van ở mức độ mở nhỏ

R3: Nếu mực nước là đủ thì van ở vị trí đóng.

R4: Nếu mực nước là cao thì van ở vị trí đóng.

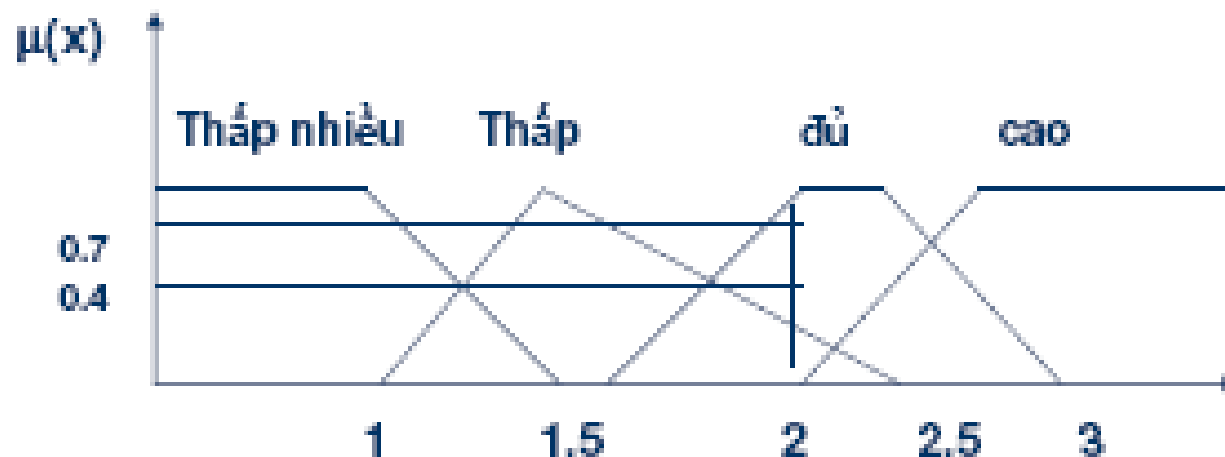
Bài tập: Điều khiển mực nước(tiếp)

Các biến ngôn ngữ:

+ x là mực nước: có 4 giá trị T={thấp nhiều, thấp ít, đủ, cao}

+ y van: có 3 giá trị T={to, nhỏ, đóng}

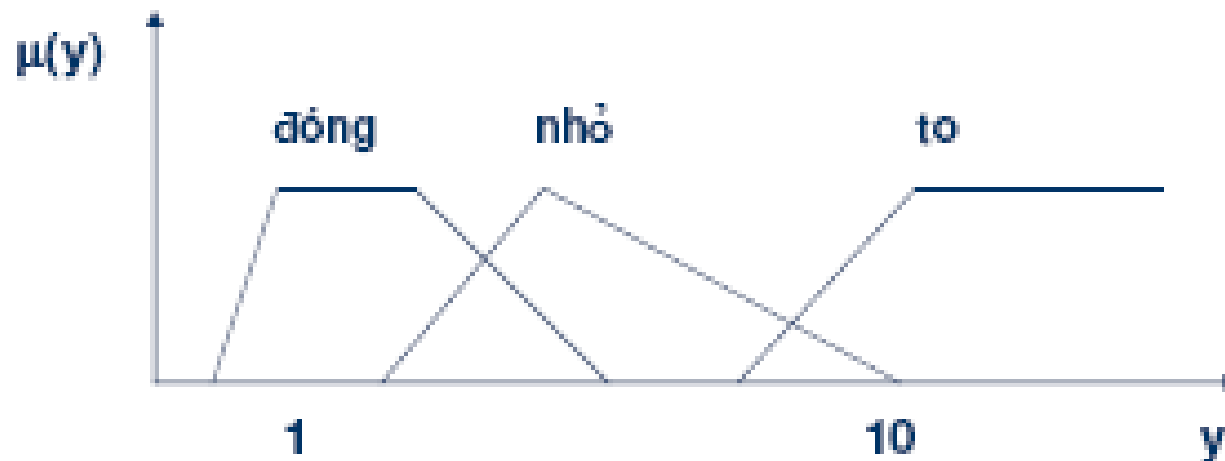
Mực nước $X \in [0 .. 3]$ (mét)



Bài tập: Điều khiển mực nước(tiếp)

Van $Y \in [0 .. 10]$ (cm)

Với y là độ mở của van, $T = \{to, nhỏ, đóng\}$



Cho mực nước cao $x_0 = 2$ m, hỏi độ mở của van y_0 là bao nhiêu

Đại Học Quốc gia TP HCM

Công nghệ tri thức và ứng dụng

GS.TSKH. Hoàng Kiếm

Nội dung môn học

Mở đầu: Giới thiệu tổng quan

Phần I: Quản lý tri thức (knowledge management)

Chương 1: Tiếp nhận, biểu diễn tri thức

Chương 2: Tối ưu hóa CSTT

Phần II: Các hệ CSTT (knowledge-based systems)

Chương 3: Bên trong một hệ CSTT

Chương 4: Phân loại các hệ CSTT

Chương 5: Một số hệ điển hình

Phần III: Khai mở dữ liệu và khám phá tri thức (Data mining and Knowledge Discovery)

Chương 6: Máy học & khám phá tri thức.

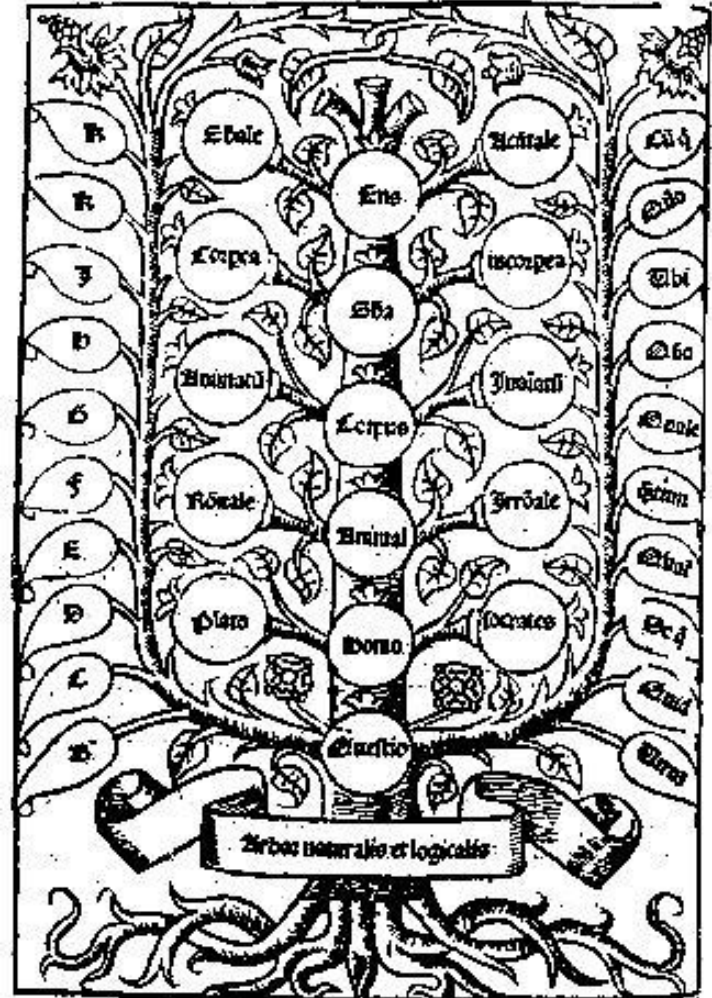
Chương 7: Khai mở dữ liệu.

Tổng kết: Tóm tắt, giới thiệu một số công trình nổi bật

Mở đầu: Giới thiệu tổng quan

Công nghệ tri thức là gì ?

- ❖ Công nghệ tri thức (Knowledge Engineering): có thể xem là một nhánh nghiên cứu của trí tuệ nhân tạo, phân tích tri thức lĩnh vực và chuyển nó thành những mô hình tính toán đưa vào máy tính để phục vụ những nhu cầu cần thiết. (*John F.Sowa. Knowledge representation: Logical, philosophical, and Computational Foundations. Copyright ©2000 by Brooks/Cole. A division of Thomson Learning*)



Công nghệ tri thức là gì ? (tt)

❖ **Công nghệ tri thức (Knowledge Engineering):** là các phương pháp, kỹ thuật được những kỹ sư tri thức (knowledge engineers) dùng để xây dựng những hệ thống thông minh như: hệ chuyên gia, hệ cơ sở tri thức, hệ hỗ trợ quyết định, etc. (Dr Dickson Lukose. Department of Mathematics, Statistics and Computer Science - The University of New England. Dr Rob Kremer Department of Computer Science The University of Calgary Calgary, Alberta, T2N 1N4 Canada. Courses: KNOWLEDGE ENGINEERING, PART A: Knowledge Representation. July 1996)

- ❖ **Công nghệ tri thức là những phương pháp, kỹ thuật dùng để:**
- ✓ **Tiếp nhận, biểu diễn tri thức.**
 - ✓ **Xây dựng các hệ cơ sở tri thức**
 - ✓ **Khám phá tri thức**

Khoa học tri thức (knowledge science)

JAIST: National graduate institution for advancement of the frontiers of science and technology

Khoa học về sáng tạo, quản lý, khai thác, sử dụng tri thức

Dựa trên sự kết hợp của Khoa học thông tin (TTNT) + Khoa học hệ thống + Khoa học xã hội (kinh tế)

- school of information science (1992)
- school of materials science (1993)
- school of **knowledge science** (1998)



Vai trò của công nghệ tri thức

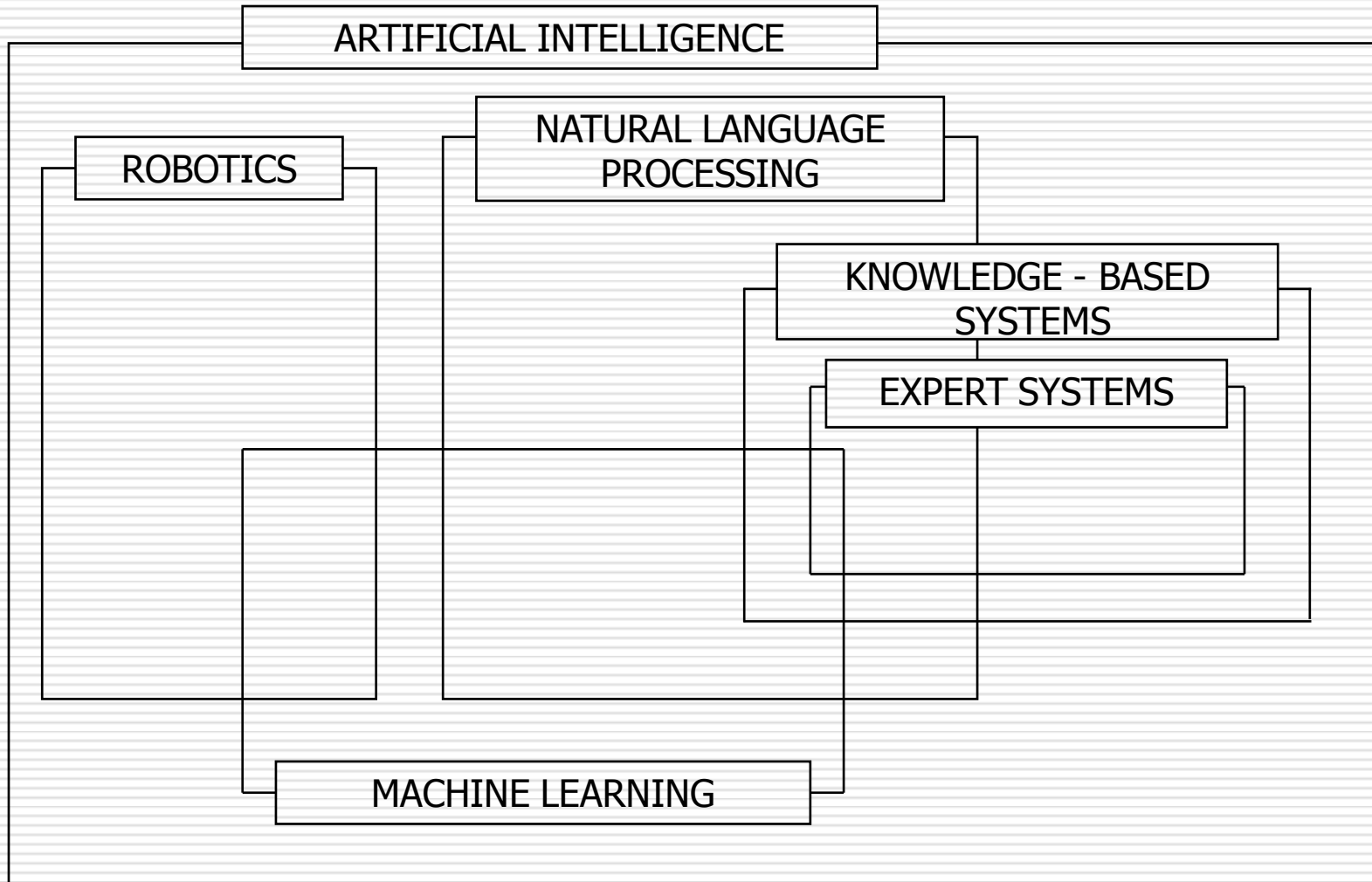
❖ Cùng với sự phát triển nhanh chóng, vượt bậc của ngành công nghiệp máy tính, nhu cầu của người dùng đối với máy tính ngày một cao hơn: không chỉ giải quyết những công việc lưu trữ, tính toán bình thường, người dùng còn mong đợi máy tính có khả năng thông minh hơn, có thể giải quyết vấn đề như con người. Và từ đó trí tuệ nhân tạo nói chung và đặc biệt là công nghệ tri thức ra đời và phát triển

❖ Công nghệ tri thức đóng vai trò hết sức quan trọng trong việc phát triển Công nghệ thông tin, nâng cao sự hữu dụng của máy tính, giúp con người gần gũi với máy tính hơn.

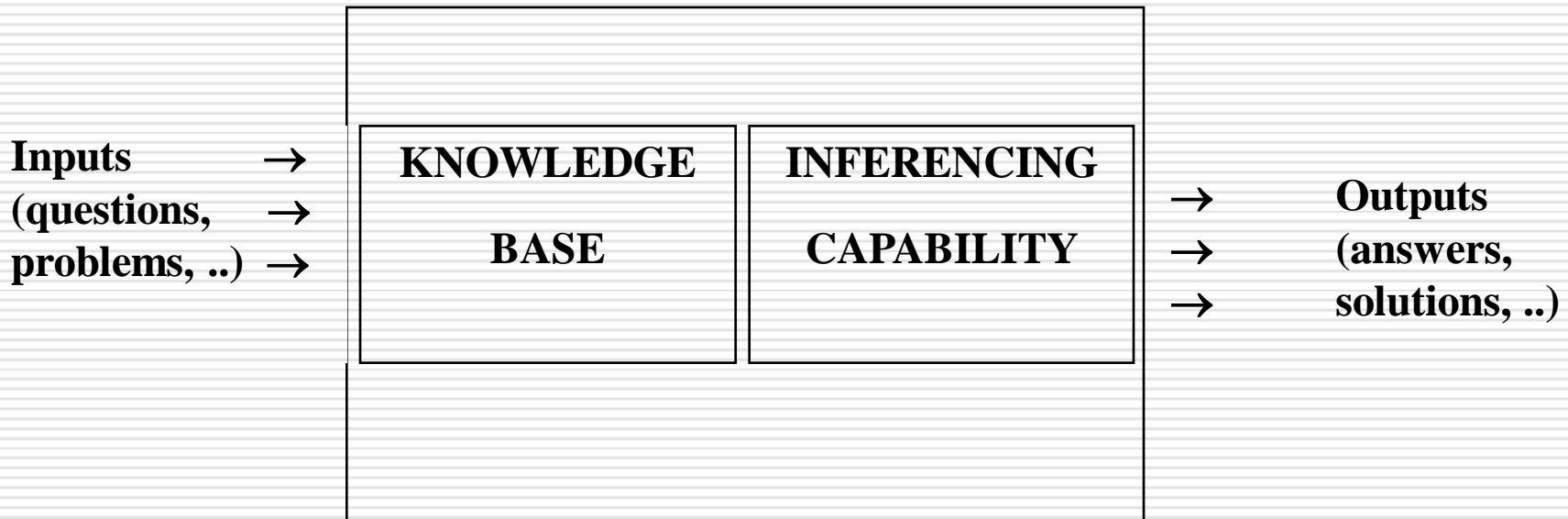
❖ Công nghệ tri thức còn góp phần thúc đẩy nhiều ngành khoa học khác phát triển, khả năng phát triển khoa học dựa trên tri thức liên ngành

❖ ...

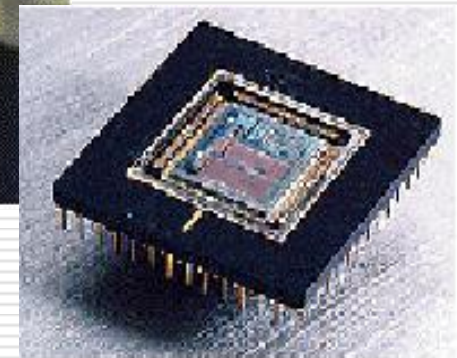
Các lĩnh vực trong thông minh nhân tạo (AI)



Áp dụng các khái niệm của AI vào máy tính

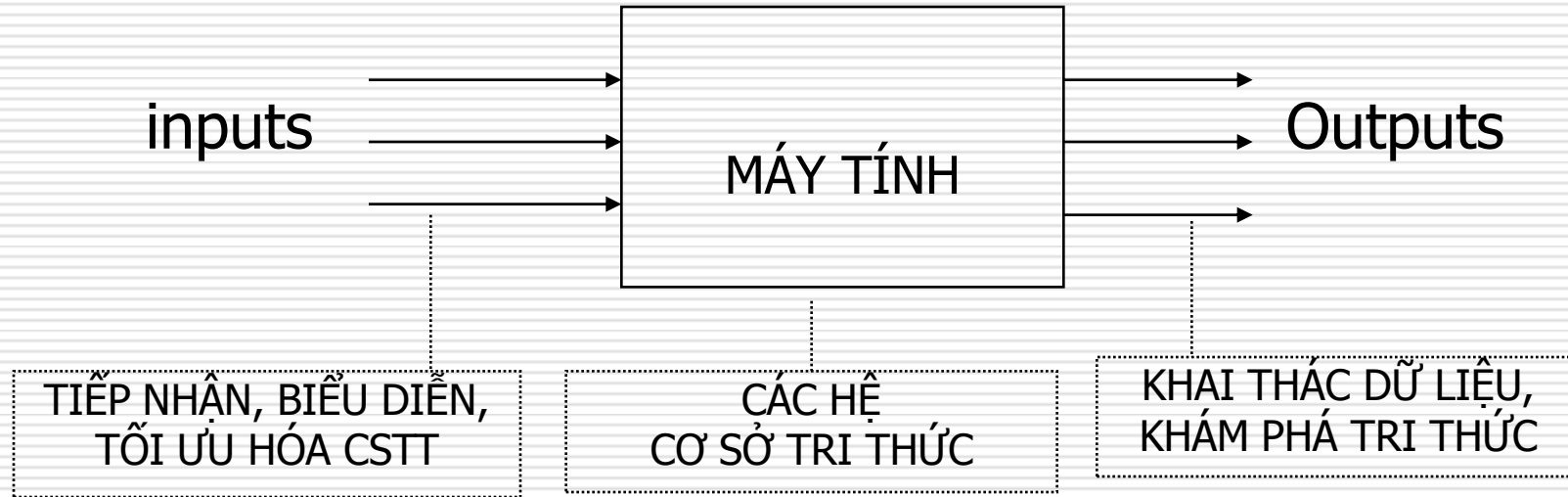


❖ Một trong những mục tiêu quan trọng của lĩnh vực nghiên cứu này là làm cho máy tính có khả năng tiếp nhận, giải quyết vấn đề giống như con người, thậm chí hơn cả con người (máy tính IBM Deep Blue đã chiến thắng vua cờ Kasparov).



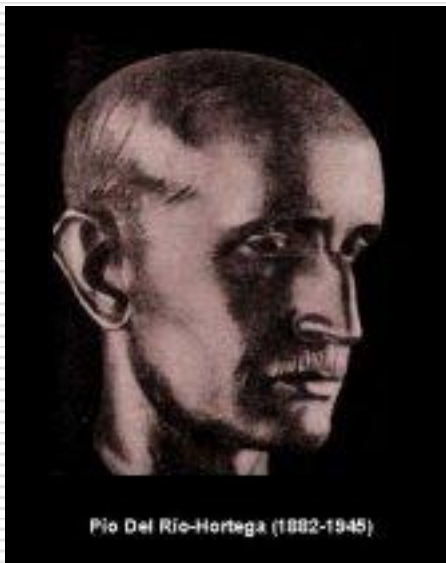
Deep Blue chip

Hướng nghiên cứu, phát triển công nghệ tri thức



- ❖ **Quản lý tri thức (knowledge management):** bao gồm tiếp nhận, biểu diễn và tối ưu hóa cơ sở tri thức...
- ❖ **Các hệ cơ sở tri thức (knowledge-based systems):** tìm hiểu cấu trúc bên trong của một hệ cơ sở tri thức, phân loại các hệ cơ sở tri thức, và một số hệ cơ sở tri thức điển hình.
- ❖ **khai mở dữ liệu, khám phá tri thức (Data mining, knowledge discovery):** nghiên cứu về phương pháp, kỹ thuật để khai mở dữ liệu và khám phá tri thức.

Đưa tri thức vào máy tính



- Nhận thức \Leftrightarrow Tiếp nhận, biểu diễn và tối ưu hóa cơ sở tri thức
- Suy luận \Leftrightarrow Động cơ suy diễn
- Phản ứng \Leftrightarrow Phản ứng, trả lời
- Tình cảm \Leftrightarrow “Bộ xử lý tình cảm” ?

Quản lý tri thức: Tiếp nhận tri thức

Có thể chia thành 2 cách để tiếp nhận tri thức như sau:

❖ **Thụ động**

- **Gián tiếp:** những tri thức kinh điển.
- **Trực tiếp:** những tri thức kinh nghiệm (không kinh điển) do “chuyên gia lĩnh vực” đưa ra.

❖ **Chủ động**

- **Đối với những tri thức tiềm ẩn, không rõ ràng hệ thống phải tự phân tích, suy diễn, khám phá để có thêm tri thức mới**

Quản lý tri thức: Tiếp nhận tri thức

❖ Giao tiếp người-máy

In: Keyboard, Mouse, sensors, touch-pad, touchable screen, speech-recognition, ...

Out: text, graphics, voice, ...



Handheld
Acern10



Con người



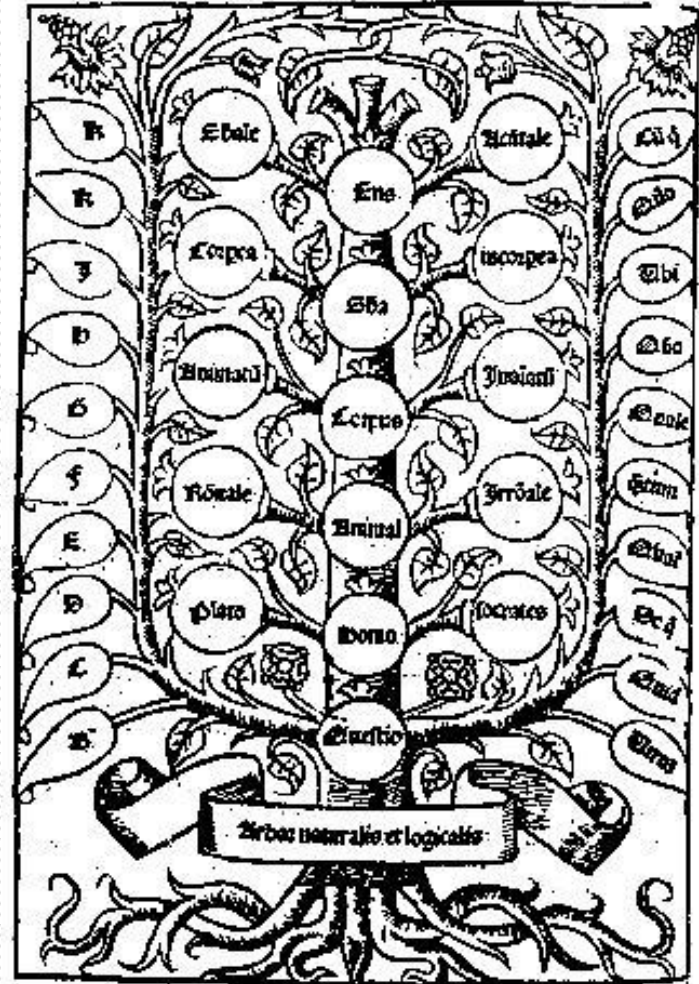
Bộ xử lý ngôn ngữ tự nhiên



Quản lý tri thức: Biểu diễn tri thức

❖ Phương pháp biểu diễn tri thức

- Logic mệnh đề & logic vị từ
- Hệ luật dẫn
- Đối tượng-thuộc tính-giá trị
- Mạng ngữ nghĩa
- Frame
- Script
- ...



Quản lý tri thức: Tối ưu cơ sở tri thức

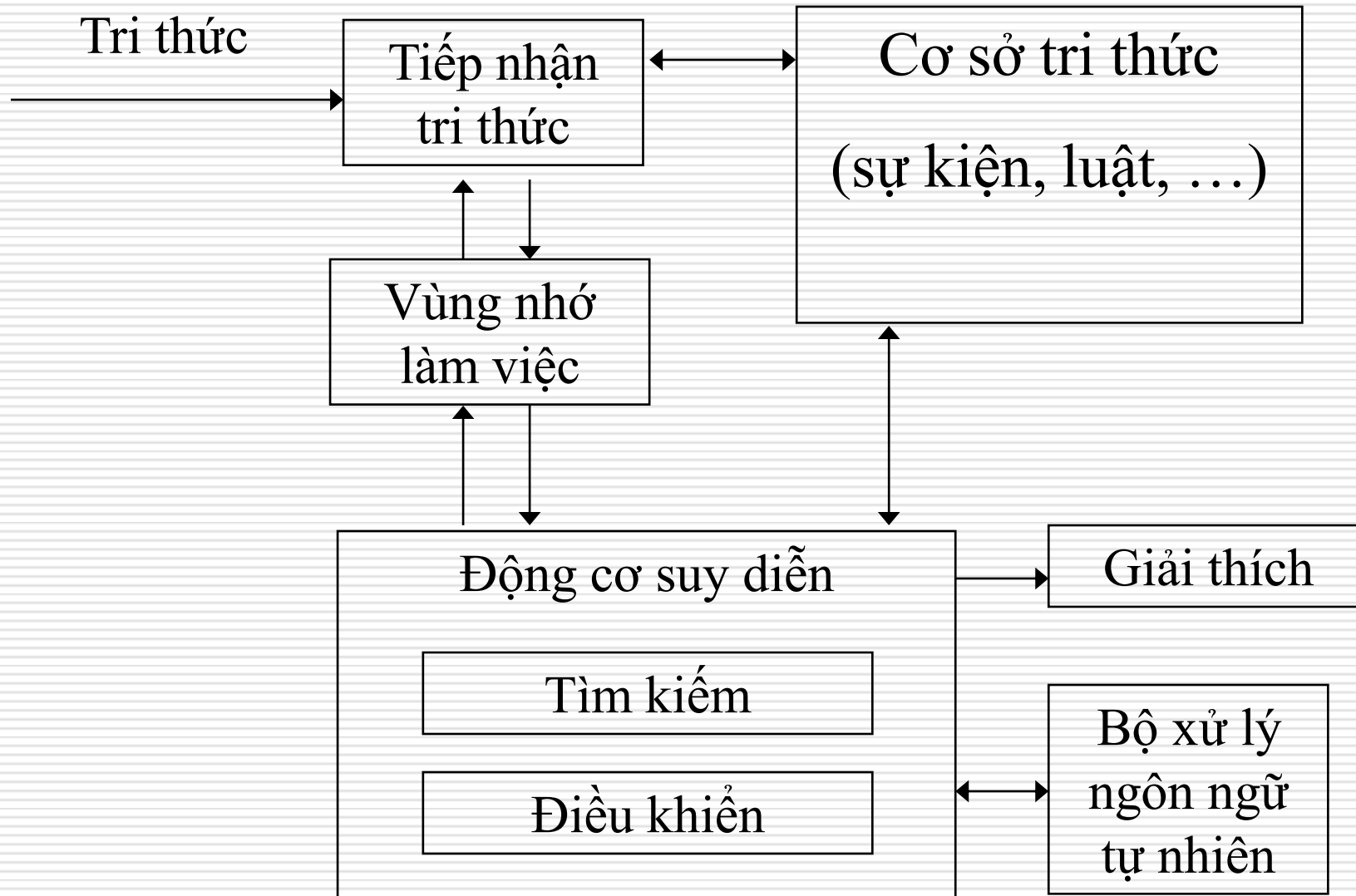
Tại sao tối ưu cơ sở tri thức ?

❖ Vấn đề mâu thuẫn, trùng lặp, dư thừa nảy sinh khi tri thức được tiếp nhận và biểu diễn trong cơ sở tri thức. Vì vậy đòi hỏi chúng ta phải có phương pháp để tối ưu cơ sở tri thức.

❖ Tùy thuộc vào cách biểu diễn tri thức, chúng ta sẽ có phương pháp thích hợp để tối ưu cơ sở tri thức.

Ví dụ: điển hình cho vấn đề này là bài toán loại bỏ luật thừa trong cơ sở tri thức luật.

Tổng quan hệ cơ sở tri thức



Các hệ cơ sở tri thức: đóng, mở, kết hợp

❖ **Hệ cơ sở tri thức đóng:** là những hệ cơ sở tri thức được xây dựng với một số “tri thức lõi vực” ban đầu, và chỉ những tri thức đó mà thôi trong suốt quá trình hoạt động hay suốt thời gian sống của nó.

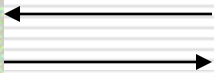
Ví dụ: những hệ cơ sở tri thức về kinh dịch, những hệ giải toán, thường là những hệ cơ sở tri thức giải quyết vấn đề...



Các hệ cơ sở tri thức: đóng, mở, kết hợp (tt)

Hệ cơ sở tri thức mở: là những hệ cơ sở tri thức tiên tiến hơn, nó có khả năng bổ sung tri thức trong quá trình hoạt động, khám phá.

Ví dụ: Những hệ giải toán cho phép bổ sung tri thức trong quá trình suy luận (tri thức ban đầu là những tiên đề và một số định lý, tri thức bổ sung là những định lý mới, những tri thức heurictis, ...); những hệ cơ sở tri thức chẩn đoán, dự báo chẳng hạn: hệ chẩn đoán y khoa MYCIN và EMYCIN, những hệ dự báo thời tiết, khí hậu, động đất



Các hệ cơ sở tri thức: đóng, mở, kết hợp (tt)

Hệ cơ sở tri thức kết hợp: bao gồm sự kết hợp giữa hệ đóng và hệ mở, hệ kết hợp giữa CSTT và CSDL, hệ kết hợp giữa hệ CSTT này với một hệ CSTT khác, ... Những hệ cơ sở tri thức kết hợp thường phát triển mạnh dựa trên tri thức liên ngành.

Ví dụ: những hệ hỗ trợ ra quyết định trong đời sống, kinh tế và khoa học; (kinh dịch, tử vi áp dụng với đời sống; kinh dịch, tử vi áp dụng với y học; ...); những hệ chẩn đoán, dự báo đòi hỏi tri thức liên ngành; ...

Các hệ cơ sở tri thức: phân loại theo phương pháp biểu diễn tri thức

Tùy thuộc vào phương pháp biểu diễn tri thức mà chúng ta có thể phân loại các hệ cơ sở tri thức

Hệ cơ sở tri thức dựa trên logic mệnh đề và logic vị từ

Hệ cơ sở tri thức dựa trên luật dẫn

Hệ cơ sở tri thức dựa trên đối tượng

Hệ cơ sở tri thức dựa trên Frame

Hệ cơ sở tri thức dựa trên mạng ngữ nghĩa

...

Hệ CSTT kết hợp một số phương pháp biểu diễn đã nêu trên.

Các hệ cơ sở tri thức: phân loại theo ứng dụng

Hệ giải quyết vấn đề: thường là hệ có tính chất đóng, nhưng đôi khi cũng có hệ mang tính mở.

Ví dụ: Những hệ giải toán, thuật giải Vương Hạo, thuật giải Robinson, ...

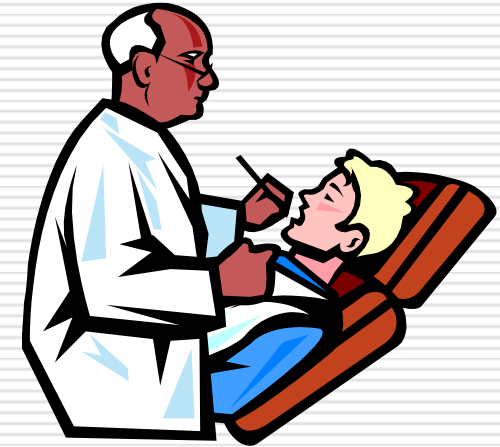
Hệ hỗ trợ quyết định: thường là các hệ mang tính kết hợp (CSDL + tri thức ngành + hàm toán học + ..), đối tượng sử dụng là các nhà lãnh đạo.

Ví dụ: những hệ thống đánh giá doanh nghiệp (tình hình tài chính, kết quả kinh doanh, qui trình nghiệp vụ, qui trình sản xuất, tính chuyên nghiệp trong quản lý, ...), những hệ thống lập kế hoạch (planning), ...

Các hệ cơ sở tri thức: phân loại theo ứng dụng (tt)

Hệ dự báo, chẩn đoán: thường cũng giống như những hệ hỗ trợ ra quyết định với tính ngoại suy cao hơn.

Ví dụ: Bài toán chẩn đoán hồng học xe, chẩn đoán y khoa, dự báo thị trường chứng khoán, thời tiết ...



Hệ điều khiển: là những hệ điều khiển có gắn với CSTT. Những hệ thống này thường ứng dụng trong công nghiệp, trong điều khiển tự động hóa, thường là những hệ thống thời gian thực (real-time systems). Một số hệ thống này có sử dụng kết hợp lý thuyết mờ để xử lý.



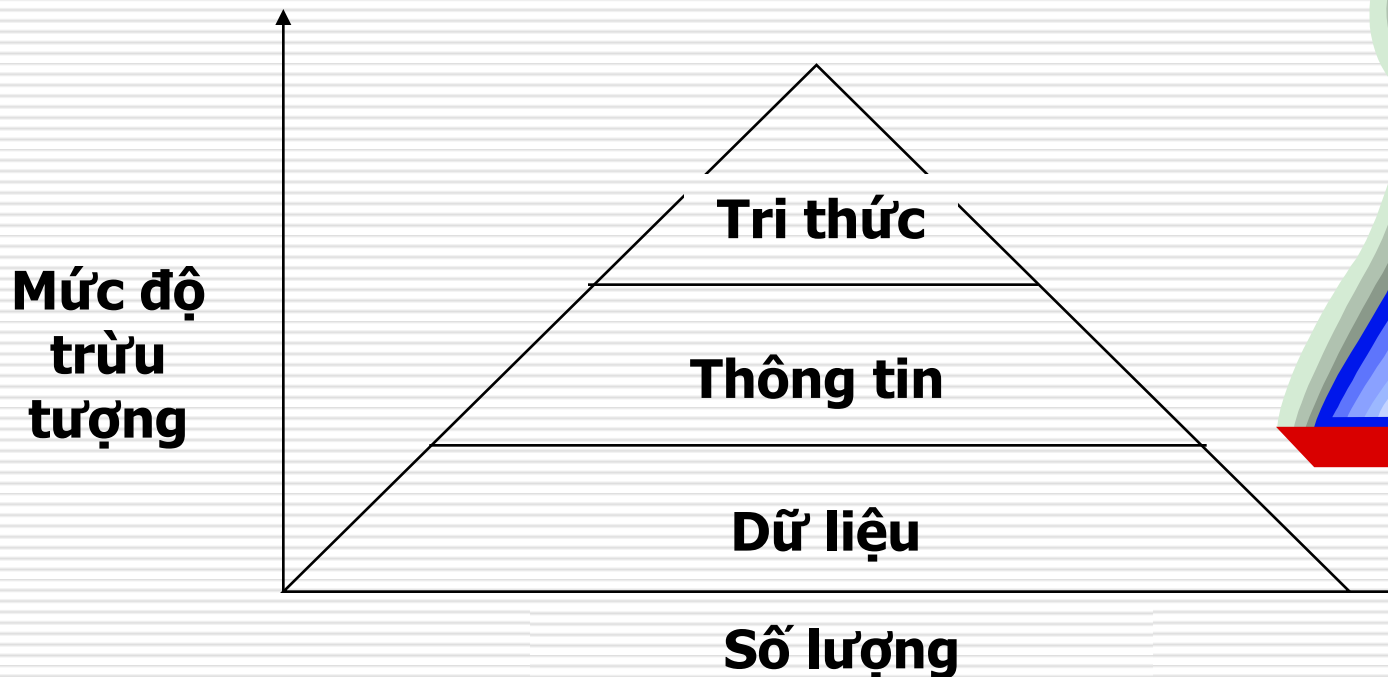
Ví dụ: Máy giặt, Máy bơm nước với bộ điều khiển mờ, ...

Máy học và khám phá tri thức

Thế nào là khám phá tri thức (knowledge discovery) ?

❖ Khám phá tri thức là tìm ra những tri thức tiềm ẩn, những tri thức mới (không phải là những tri thức kinh điển, kinh nghiệm, ...)

Thừa dữ liệu, thông tin nhưng thiếu tri thức.



Máy học và khám phá tri thức (tt)

Ví dụ: Trong toán học

❖ Dữ liệu: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Mối liên hệ này có thể được biểu diễn bằng công thức sau: $U_n = U_{n-1} + U_{n-2}$

⇒ Công thức tìm ra ở trên chính là tri thức

Ví dụ: Trong vật lý

Công thức: $U = I \times R$ là tri thức rút ra từ thực nghiệm

Ví dụ: Chuồn chuồn bay thấp thì mưa, bay cao thì nắng, bay vừa thì râm

⇒ Lời nhận xét trên là tri thức rút ra từ kinh nghiệm đời sống.

Máy học và khám phá tri thức (tt)

Thế nào là máy học (Learning Machine) ?

❖ Máy tính hay chương trình máy tính có khả năng tự hoàn thiện từ “kinh nghiệm”.

❖ Máy học còn có nghĩa là việc mô hình hóa môi trường xung quanh hay khả năng một chương trình máy tính sinh ra một cấu trúc dữ liệu mới khác với cấu trúc hiện có. Chẳng hạn việc tìm ra những luật If...then... từ tập dữ liệu đầu vào.

(Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski. Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, 1998)

Máy học và khám phá tri thức (tt)

Phân loại các phương pháp máy học: có nhiều quan điểm phân loại khác nhau

Phân loại thô:

- ❖ Học giám sát (supervised learning)
- ❖ Học không giám sát (unsupervised learning)

Phân loại theo 2 tiêu chuẩn cùng lúc: “cấp độ học” & “cách tiếp cận”

Cấp độ học:

- ❖ Học vẹt (Rote learning)
- ❖ Học theo giải thích (by explanation)
- ❖ Học theo ví dụ, trường hợp (by examples, cases)
- ❖ Học khám phá (by discovering)

Máy học và khám phá tri thức (tt)

Cách tiếp cận:

- ❖ Tiếp cận thống kê
- ❖ Tiếp cận toán tử logic
- ❖ Tiếp cận hình học (phân hoạch không gian, xây dựng cây định danh, ...)
- ❖ Tiếp cận mạng Neural
- ❖ Tiếp cận khai mở dữ liệu
- ❖ ...

Nhà kho dữ liệu và khai mỏ dữ liệu

Tạo DATA WAREHOUSE = Biến đổi dữ liệu thành tri thức yểm trợ tiến trình ra quyết định.

Dữ liệu

Khoa học

Giáo dục

Công nghệ

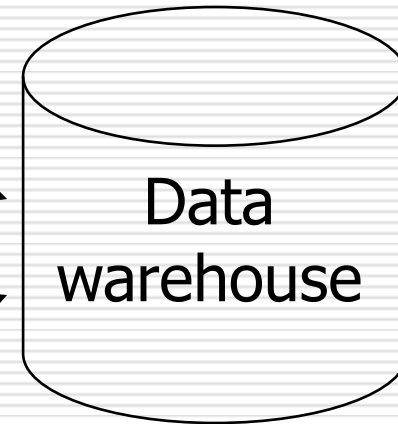
Kinh doanh

Thị trường

Thời tiết

...

Tạo Data Warehouse



Tri thức yểm trợ ra quyết định

Nhà kho dữ liệu và khai mỏ dữ liệu (tt)

DatawareHouse = Business Information + Decision Making

(IBM BPEC '96 Conference, San Diego, USA)

Sự bùng nổ của các CSDL lớn vượt quá khả năng diễn dịch và lĩnh hội của con người, phát sinh yêu cầu sáng tạo các công cụ kỹ thuật mới để phân tích dữ liệu một cách thông minh và tự động nhằm tạo ra tri thức hữu dụng hỗ trợ tốt cho tiến trình ra quyết định.

(Usama, Data Mining and Knowledge Discovery, 1995)

Nhà kho dữ liệu và khai mỏ dữ liệu (tt)

Một số bài toán điển hình về data mining

Bài toán khám phá luật kết hợp

Bài toán nhận dạng mẫu

Bài toán phân loại dữ liệu

Bài toán gom nhóm dữ liệu

Bài toán lập mô hình

Bài toán dự báo

...



Tài liệu tham khảo

- [1] GS.TSKH Hoàng Kiếm. Bài giảng cao học môn học cơ sở tri thức và ứng dụng. ĐHKHTN-TPHCM.
- [2] GS.TSKH Hoàng Kiếm. Thư viện những báo cáo khoa học, bài thu hoạch môn cơ sở tri thức và ứng dụng. Các lớp cao học thuộc khoa CNTT- ĐHKHTN. TPHCM.
- [3] GS.TSKH Hoàng Kiếm, TS. Đỗ Văn Nhơn, Th.sĩ Đỗ Phúc. Giáo trình Các hệ cơ sở tri thức. Đại Học Quốc Gia TPHCM – 2002.
- [4] GS.TSKH Hoàng Kiếm, Th.sĩ Đinh Nguyễn Anh Dũng. Giáo trình Trí tuệ nhân tạo. Đại Học Quốc Gia TPHCM – 2002.
- [5] John F.Sowa. Knowledge representation: Logical, Philosophical, and Computational Foundations. Copyright @ 2000 by Brooks/Cole. A division of Thomson Learning.
- [6] Adrian A.Hopgood. Knowledge-based systems for Engineers and Scientists. The Open University – CRC Press. Boca-Raton Ann-Arbor London Tokyo 1998.
- [7] Sharon Wood. Planning and decision making in dynamic domains. Ellis Horwood Series in Artificial Intelligence - 1998.
- [8] Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski. Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, 1998
- [9] Citeseer - Scientific Literature Digital Library. Artificial Intelligence-
<http://citeseer.nj.nec.com/ArtificialIntelligence/> - 2003

Một số nghiên cứu đang được quan tâm về AI
<http://citeseer.nj.nec.com/ArtificialIntelligence/>

- Expert Systems :

<http://citeseer.nj.nec.com/ArtificialIntelligence/ExpertSystems/>

- Knowledge Representation:

<http://citeseer.nj.nec.com/ArtificialIntelligence/KnowledgeRepresentation/>

- Natural Language Processing:

<http://citeseer.nj.nec.com/ArtificialIntelligence/NaturalLanguageProcessing/>

- Optimization:

<http://citeseer.nj.nec.com/ArtificialIntelligence/Optimization/>

- Planning:

<http://citeseer.nj.nec.com/ArtificialIntelligence/Planning/>

- Robotics:

<http://citeseer.nj.nec.com/ArtificialIntelligence/Robotics/>



Phần I:

Quản lý tri thức

Tham khảo thêm:

[1] GS.TSKH Hoàng Kiếm, TS. Đỗ Văn Nhơn, Th.sĩ Đỗ Phúc. Giáo trình Các hệ cơ sở tri thức. Đại Học Quốc Gia TP HCM – 2002.

[2] GS.TSKH Hoàng Kiếm, Th.sĩ Đinh Nguyễn Anh Dũng. Giáo trình Trí tuệ nhân tạo. Đại Học Quốc Gia TP HCM – 2002.

[3] John F.Sowa. Knowledge representation: Logical, Philosophical, and Computational Foundations. Copyright @ 2000 by Brooks/Cole. A division of Thomson Learning.

Chương 1: Tiếp nhận và biểu diễn tri thức

I. Tri thức & Các loại tri thức

Tri thức (knowledge) ?

Knowledge: the psychological result of perception and learning and reasoning (English – English Dictionary)

Tri thức là kết quả của quá trình nhận thức, học tập và lập luận.

Phân loại tri thức

Tri thức thủ tục: mô tả cách thức giải quyết một vấn đề. Loại tri thức này đưa ra giải pháp để thực hiện một công việc nào đó.

Tri thức khai báo: cho biết một vấn đề được thấy như thế nào. Loại tri thức này bao gồm các phát biểu đơn giản, dưới dạng các khẳng định logic đúng hoặc sai.

I. Tri thức & Các loại tri thức (tt)

Siêu tri thức: mô tả *tri thức về tri thức*. Loại tri thức này giúp lựa chọn tri thức thích hợp nhất trong số các tri thức khi giải quyết một vấn đề.

Tri thức heuristic: mô tả các "*mẹo*" để dẫn dắt tiến trình lập luận. Tri thức heuristic còn được gọi là *tri thức nông cạn* do không đảm bảo hoàn toàn chính xác về kết quả giải quyết vấn đề.

Tri thức có cấu trúc: mô tả tri thức theo cấu trúc. Loại tri thức này mô tả mô hình tổng quan hệ thống theo quan điểm của chuyên gia, bao gồm khái niệm, khái niệm con, và các đối tượng; diễn tả chức năng và mối liên hệ giữa các tri thức dựa theo cấu trúc xác định

II. Phương pháp tiếp nhận tri thức

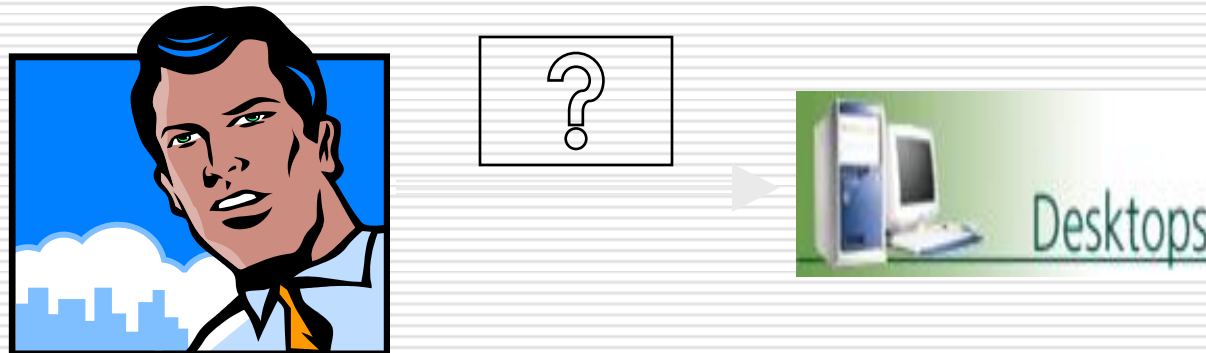
Có thể chia thành 2 cách để tiếp nhận tri thức như sau:

❖ Thụ động

- Gián tiếp: những tri thức kinh điển.
- Trực tiếp: những tri thức kinh nghiệm (không kinh điển) do “chuyên gia lĩnh vực” đưa ra.

❖ Chủ động

- Đối với những tri thức tiềm ẩn, không rõ ràng hệ thống phải tự phân tích, suy diễn, khám phá để có thêm tri thức mới



III. Phương pháp biểu diễn tri thức

1. Logic mệnh đề & logic vị từ: Dạng biểu diễn tri thức cổ điển nhất trong máy tính là logic, với 2 dạng phổ biến là logic mệnh đề và logic vị từ. Cả 2 dạng này đều dùng kí hiệu để biểu diễn tri thức và các toán tử áp lên các ký hiệu để suy luận logic. Logic đã cung cấp cho các nhà nghiên cứu những công cụ hình thức để biểu diễn và suy luận tri thức.

Các phép toán logic và các ký hiệu sử dụng

Phép toán	AND	OR	NOT	Kéo theo	Tương đương
Kí hiệu					

III. Phương pháp biểu diễn tri thức (tt)

1.1 Logic mệnh đề

Ví dụ 1:

IF Xe không khởi động được \blacksquare
AND Khoảng cách từ nhà đến chỗ làm là xa \blacksquare
THEN Sẽ trễ giờ làm \blacksquare

Luật trên có thể biểu diễn lại như sau: $A \blacksquare B \blacksquare$.

Các phép toán quen thuộc trên các mệnh đề trong bảng sau:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
T	T	F	T	T	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
F	F	T	F	F	T	T

III. Phương pháp biểu diễn tri thức (tt)

1.2 Logic vị từ

Mệnh đề: thì *không có cấu trúc* [] an chế nhiều thao tác suy luận []ưa vào khái niệm vị từ và lượng từ ([] với mọi , [] tồn tại) để tăng cường tính cấu trúc của một mệnh đề.

Trong logic vị từ, một mệnh đề được cấu tạo bởi 2 thành phần là các đối tượng tri thức và mối liên hệ giữa chúng (gọi là vị từ)

Biểu diễn: Vị từ(<đối tượng 1>, <đối tượng 2>, ..., <đối tượng n>)

Ví dụ 1: Cam có vị ngọt [] (cam, ngọt)

Cam có màu xanh [] àu(cam, xanh)

...

III. Phương pháp biểu diễn tri thức (tt)

Ví dụ 2: Tri thức “*A là bố của B nếu B là anh hoặc em của một người con của A*” có thể được biểu diễn dưới dạng vị từ như sau :

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

Trong trường hợp này, mệnh đề $\text{Bố}(A,B)$ là một mệnh đề tổng quát

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

a) $\text{Bố}(\text{“An”}, \text{“Bình”})$ có giá trị đúng (An là bố của Bình)

b) $\text{Anh}(\text{“Tú”}, \text{“Bình”})$ có giá trị đúng (Tú là anh của Bình)

thì mệnh đề **c)** $\text{Bố}(\text{“An”}, \text{“Tú”})$ sẽ có giá trị là đúng.

(An là bố của Tú).

III. Phương pháp biểu diễn tri thức (tt)

Ví dụ 3: Câu cách ngôn “*Không có vật gì là lớn nhất và không có vật gì là bé nhất!*” có thể được biểu diễn dưới dạng vị từ như sau :

$$\text{LớnHơn}(x,y) = x > y$$

$$\text{NhỏHơn}(x,y) = x < y$$

■ ■ : $\text{LớnHơn}(y,x)$ và ■ ■ : $\text{NhỏHơn}(y,x)$

Ví dụ 4: Câu châm ngôn “Gần mực thì đen, gần đèn thì sáng” được hiểu là “chơi với bạn xấu nào thì ta cũng sẽ thành người xấu” có thể được biểu diễn bằng vị từ như sau :

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

Công cụ vị từ đã được nghiên cứu và phát triển thành một ngôn ngữ lập trình đặc trưng cho trí tuệ nhân tạo. Đó là ngôn ngữ PROLOG.

III. Phương pháp biểu diễn tri thức (tt)

Nhận xét:

- ❖ Kiểu biểu diễn tri thức vị từ giống như hàm trong các ngôn ngữ lập trình, đối tượng tri thức là tham số của hàm, giá trị mệnh đề chính là kết quả của hàm (kiểu Boolean).
- ❖ Biểu diễn tri thức bằng mệnh đề gặp khó khăn là không thể can thiệp vào cấu trúc của một mệnh đề đưa ra khái niệm lượng từ, vị từ.
- ❖ Với vị từ có thể biểu diễn tri thức dưới dạng các mệnh đề tổng quát tổng quát.

III. Phương pháp biểu diễn tri thức (tt)

1.3 Một số thuật giải liên quan đến logic mệnh đề:

- ❖ Một trong những vấn đề khá quan trọng của logic mệnh đề là chứng minh tính đúng đắn của phép suy diễn (a ██████).
- ❖ Với công cụ máy tính, bạn có thể cho rằng ta sẽ dễ dàng chứng minh được mọi bài toán bằng một phương pháp “thô bạo” là lập bảng chân trị. Tuy về lý thuyết, phương pháp lập bảng chân trị luôn cho được kết quả cuối cùng nhưng độ phức tạp của phương pháp này là quá lớn, $O(2^n)$ với n là số biến mệnh đề. Sau đây chúng ta sẽ nghiên cứu hai phương pháp chứng minh mệnh đề với độ phức tạp chỉ có $O(n)$. Thuật giải Vương Hạo và thuật giải Robinson.

III. Phương pháp biểu diễn tri thức (tt)

Thuật giải Vương Hạo:

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

$$GT_1, GT_2, \dots, GT_n \quad \blacksquare \quad KL_1, KL_2, \dots, KL_m$$

Trong đó các GT_i và KL_i là các mệnh đề được xây dựng từ các biến mệnh đề và 3 phép nối cơ bản : \blacksquare

B2 : Chuyển vế các GT_i và KL_i có dạng phủ định.

Ví dụ :

$$p \blacksquare q, \blacksquare r \blacksquare s), \blacksquare, p \blacksquare \blacksquare \blacksquare$$
$$\blacksquare \blacksquare p \blacksquare q, p \blacksquare, p \blacksquare \blacksquare s), g, s$$

III. Phương pháp biểu diễn tri thức (tt)

Thuật giải Vương Hạo: (tt)

B3 : Nếu ở GT_i có phép \square thì thay thế phép \square bằng dấu “,”

Nếu ở KL_i có phép \square thì thay thế phép \square bằng dấu “,”

Ví dụ : $p \square, r \square \square) \square \square$
 $\square p, q, r, \square \square \square \square$

B4 : Nếu ở GT_i có chứa phép \square thì tách thành hai dòng con.

Nếu ở KL_i có chứa phép \square thì tách thành hai dòng con.

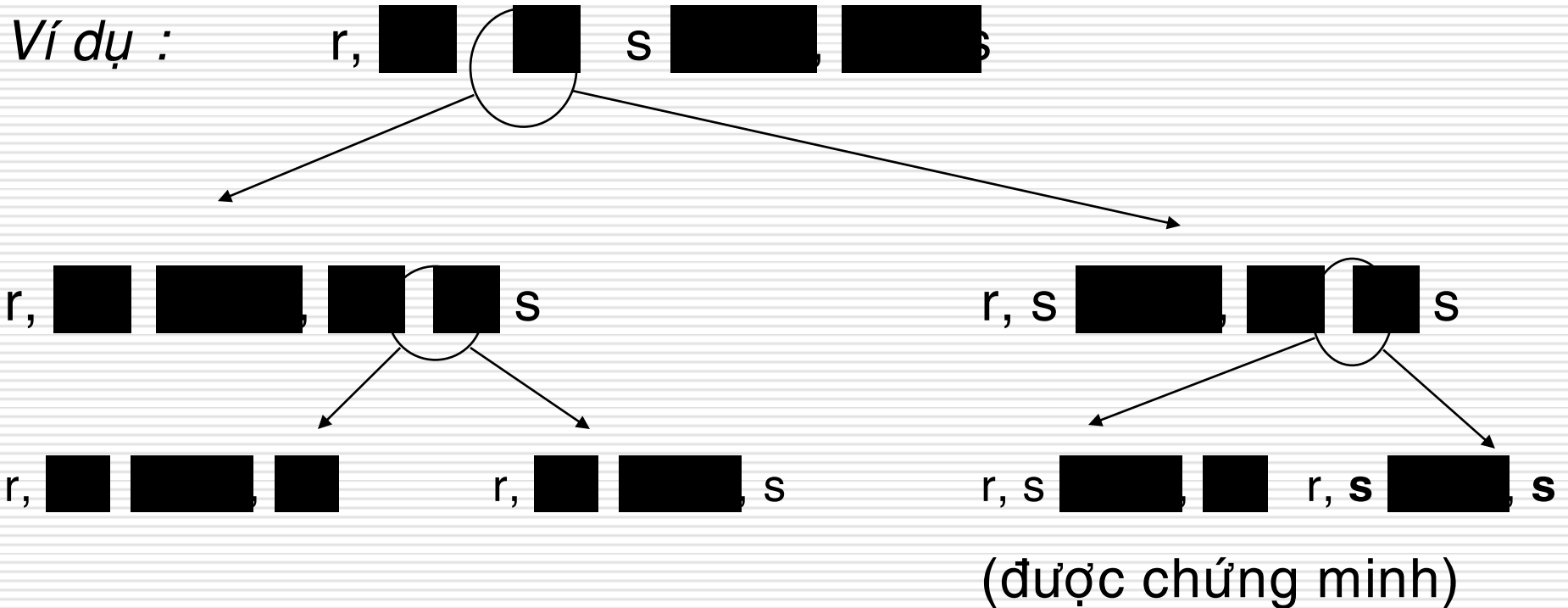
Ví dụ :

$p, \square \square q \square$

$p, \square \square$ $p, q \square$

III. Phương pháp biểu diễn tri thức (tt)

Thuật giải Vương Hạo: (tt)



hư vậy biểu thức ban đầu không được chứng minh.

III. Phương pháp biểu diễn tri thức (tt)

Thuật giải Robinson:

- Thuật giải này hoạt động dựa trên phương pháp chứng minh phản chứng.

Chứng minh phép suy luận (a \Rightarrow b) là đúng (với a là giả thiết, b là kết luận).

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

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

III. Phương pháp biểu diễn tri thức (tt)

Thuật giải Robinson: (tt)

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

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

Trong đó : GT_i và KL_i được xây dựng từ các biến mệnh đề và các phép toán :

B2 : Nếu ở GT_i có phép thì thay thế phép bằng dấu “,”

Nếu ở KL_i có phép thì thay thế phép bằng dấu “,”

B3 : Biến đổi dòng chuẩn ở B1 về thành danh sách mệnh đề như sau :

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

III. Phương pháp biểu diễn tri thức (tt)

Thuật giải Robinson: (tt)

B4 : Nếu trong danh sách mệnh đề ở bước 2 có 2 mệnh đề đối ngẫu nhau thì bài toán được chứng minh. Ngược lại thì chuyển sang B4. (a và ■ gọi là hai mệnh đề đối ngẫu nhau)

B5 : Xây dựng một mệnh đề mới bằng cách tuyển một cặp mệnh đề trong danh sách mệnh đề ở bước 2. Nếu mệnh đề mới có các biến mệnh đề đối ngẫu nhau thì các biến đó được loại bỏ.

Ví dụ : p ■ ■ ■ § ■ q

Hai mệnh đề ■, q là đối ngẫu nên sẽ được loại bỏ

■ ■ §

III. Phương pháp biểu diễn tri thức (tt)

Thuật giải Robinson: (tt)

B6 : Thay thế hai mệnh đề vừa tuyển trong danh sách mệnh đề bằng mệnh đề mới.

Ví dụ :

{ p ■■■■ , ■■■■ s ■■■■ q , w ■■■■ , s ■■■■ q }
■■■■ p ■■■■ s ■■■■ , w ■■■■ , s ■■■■ q }

B7 : Nếu không xây dựng được thêm một mệnh đề mới nào và trong danh sách mệnh đề không có 2 mệnh đề nào đối ngẫu nhau thì vấn đề không được chứng minh.

III. Phương pháp biểu diễn tri thức (tt)

Ví dụ về thuật giải Robinson: Chứng minh rằng

□, □, □, □, □, □

B3: { □, □, □, □, □, □ p, u }

B4 : Có tất cả 6 mệnh đề nhưng chưa có mệnh đề nào đối ngẫu nhau.

B5 : Chọn một cặp mệnh đề (chọn hai mệnh đề có biến đối ngẫu). Chọn hai mệnh đề đầu :

□ □ □ □ □ □

Danh sách mệnh đề thành : { □, □, □, □, □, □ p, u }

Vẫn chưa có mệnh đề đối ngẫu.

III. Phương pháp biểu diễn tri thức (tt)

Ví dụ về thuật giải Robinson: (tt)

✓ Tuyển hai cặp mệnh đề đầu tiên: [] [] s [] [] s

Danh sách mệnh đề thành { [] [] s, [] [] p, u }

Vẫn chưa có hai mệnh đề đối ngẫu

✓ Tuyển hai cặp mệnh đề đầu tiên: [] [] s [] [] [] []

Danh sách mệnh đề thành : { [] [] p, u }

Vẫn chưa có hai mệnh đề đối ngẫu

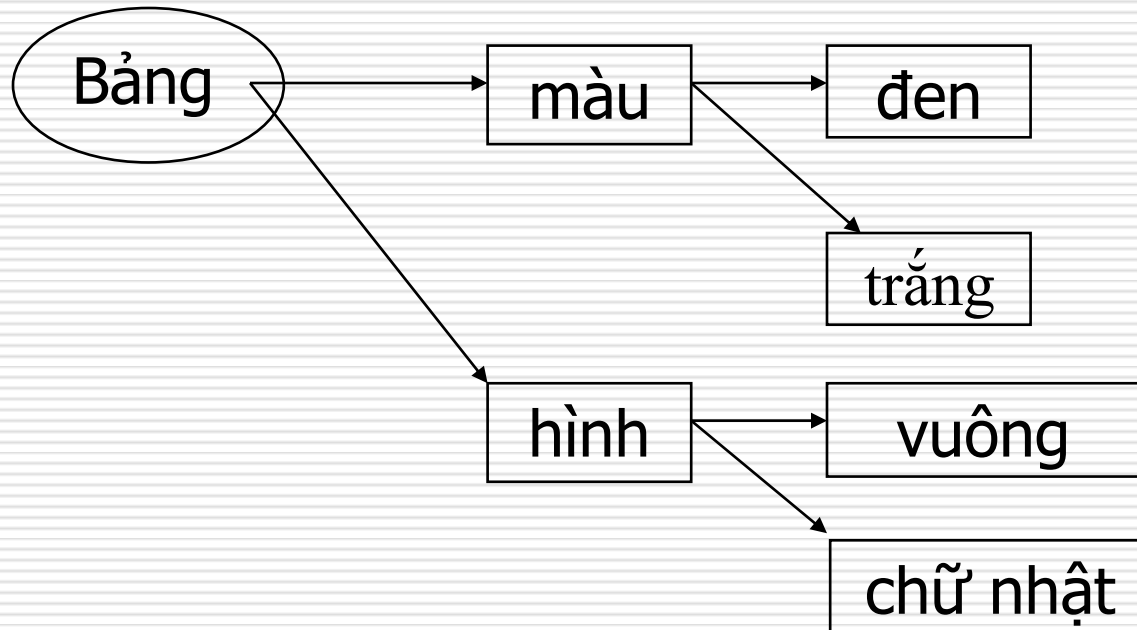
✓ Tuyển hai cặp mệnh đề : [] [] [] []

Danh sách mệnh đề trở thành : { [] p }

Có hai mệnh đề đối ngẫu nên biểu thức ban đầu đã được chứng minh.

III. Phương pháp biểu diễn tri thức (tt)

2. Đối tượng-thuộc tính-giá trị (object-attribute-value)



III. Phương pháp biểu diễn tri thức (tt)

3. Tri thức luật dẫn

Phương pháp biểu diễn tri thức bằng luật sinh được phát minh bởi Newell và Simon trong lúc hai ông đang cố gắng xây dựng một hệ giải bài toán tổng quát (các hệ GPS). Đây là một kiểu biểu diễn tri thức có cấu trúc. Ý tưởng cơ bản là tri thức có thể được cấu trúc bằng một cặp **điều kiện – hành động**

Ví dụ 1: Bài toán đổ nước, chúng ta có 2 bình có dung tích là 4 lít và 3 lít, hỏi làm thế nào để đong được chính xác 2 lít nước.



III. Phương pháp biểu diễn tri thức (tt)

Bài toán đổ nước (tt)

Bài toán trên được biểu diễn dưới dạng không gian trạng thái bằng luật như sau:

(x, y : lần lượt là số lít nước hiện có trong bình 4 lít và 3 lít)

1. Nếu ($x < 4$)

(x, y)  $(4, y)$ Đổ đầy bình 4 lít

2. Nếu ($y < 3$)

(x, y)  $(x, 3)$ Đổ đầy bình 3 lít

3. Nếu ($x > 0$)

(x, y)  $(x-d, y)$ Đổ d lít ra khỏi bình 4 lít

III. Phương pháp biểu diễn tri thức (tt)

Bài toán đổ nước (tt)

4. Nếu $(y > 0)$

(x, y)  $(x, y-d)$ Đổ d lít ra khỏi bình 3 lít

5. Nếu $(x > 0)$

(x, y)  $(0, y)$ Đổ hết nước ra khỏi bình 4 lít

6. Nếu $(y > 0)$

(x, y)  $(x, 0)$ Đổ hết nước ra khỏi bình 3 lít

7. Nếu $(x+y < 4)$ và $(y > 0)$

(x, y)  $(4, y-(4-x))$ Đổ nước từ bình 3 lít vào bình 4 lít đến khi bình 4 lít đầy

III. Phương pháp biểu diễn tri thức (tt)

Bài toán đổ nước (tt)

8. Nếu $(x+y \leq 3)$ và $(x > 0)$

$$(x, y) \rightarrow (x-(3-y), 3)$$

Đổ nước từ bình 4 lít vào bình 3 lít cho đến khi bình 3 lít đầy.

9. Nếu $(x+y \leq 4)$ và $(y > 0)$

$$(x, y) \rightarrow (x+y, 0)$$

Đổ hết nước bình 3 lít vào bình 4 lít.

10. Nếu $(x+y \leq 3)$ và $(x > 0)$

$$(x, y) \rightarrow (0, x+y)$$

Đổ hết nước bình 4 lít vào bình 3 lít.

11. $(0, 2) \rightarrow (2, 0)$

Đổ 2 lít từ bình 3 lít vào bình 4 lít.

12. $(2, y) \rightarrow (0, y)$

Đổ 2 lít ra khỏi bình 4 lít.

III. Phương pháp biểu diễn tri thức (tt)

Bài toán đổ nước (tt)

❖ Hiện nay để giải bài toán đổ nước như thế này người ta đã rút gọn lại chỉ còn 3 luật như sau:

(L1): Nếu bình 3 lít đầy thì đổ hết nước trong bình 3 lít đi.

(L2): Nếu bình 4 lít rỗng thì đổ đầy nước vào bình 4 lít.

(L3): Nếu bình 3 lít không đầy và bình 4 lít không rỗng thì đổ nước từ bình 4 lít sang bình 3 lít (cho tới khi bình 3 lít đầy hoặc bình 4 lít hết nước).

III. Phương pháp biểu diễn tri thức (tt)

Ưu điểm và nhược điểm của tri thức luật dẫn

Ưu điểm:

- ❖ Các luật rất dễ hiểu nên có thể dễ dàng dùng để trao đổi với người dùng (vì nó là một trong những dạng tự nhiên của ngôn ngữ).
- ❖ Có thể dễ dàng xây dựng được cơ chế suy luận và giải thích từ các luật.
- ❖ Việc hiệu chỉnh và bảo trì hệ thống là tương đối dễ dàng.
- ❖ Có thể cải tiến dễ dàng để tích hợp các luật mờ.
- ❖ Các luật thường ít phụ thuộc vào nhau.

III. Phương pháp biểu diễn tri thức (tt)

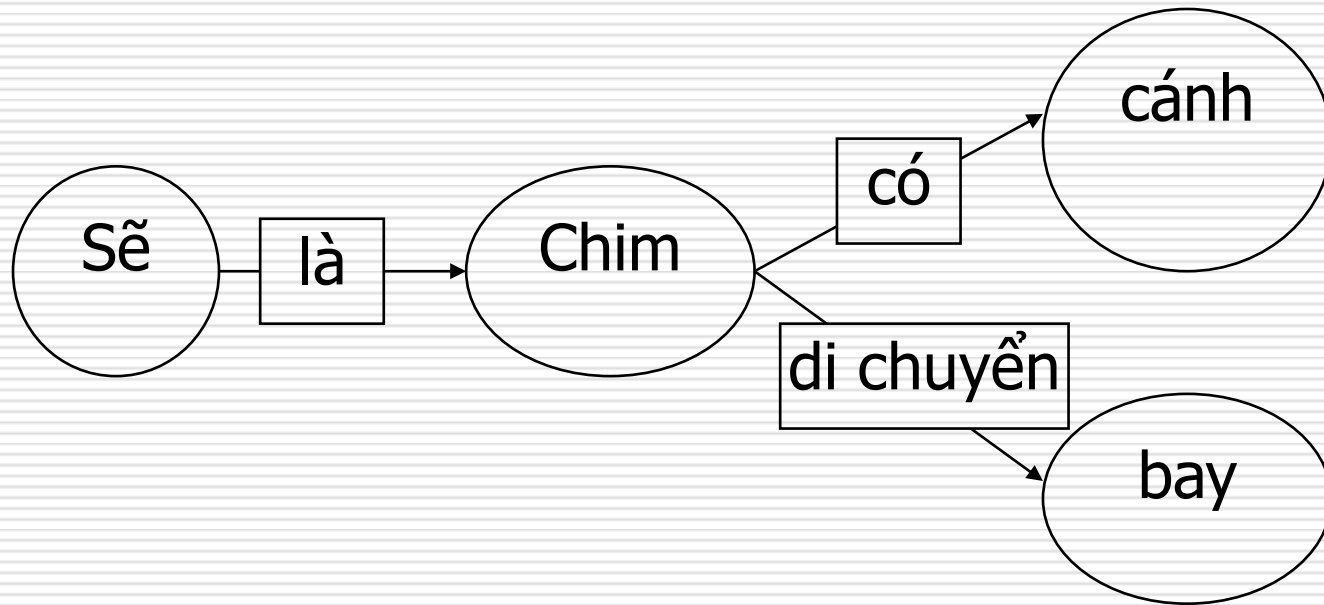
Ưu điểm và nhược điểm của tri thức luật dẫn

Nhược điểm:

- ❖ Các tri thức phức tạp đôi lúc đòi hỏi quá nhiều (hàng ngàn) luật dẫn. Điều này sẽ làm nảy sinh nhiều vấn đề liên quan đến tốc độ lần quản trị hệ thống.
- ❖ Người xây dựng hệ thống thích sử dụng luật dẫn hơn tất cả phương pháp khác, nên họ thường tìm mọi cách để biểu diễn tri thức bằng luật cho dù có phương pháp khác thích hợp hơn! Đây là nhược điểm mang tính chủ quan của con người.
- ❖ Cơ sở tri thức luật dẫn lớn sẽ làm giới hạn khả năng tìm kiếm của chương trình điều khiển. Nhiều hệ thống gặp khó khăn trong việc đánh giá các hệ dựa trên luật cũng như gặp khó khăn khi suy luận trên luật.

III. Phương pháp biểu diễn tri thức (tt)

4. Mạng ngữ nghĩa: là một phương pháp biểu diễn tri thức dùng đồ thị. Trong đó nút biểu diễn đối tượng, và cung biểu diễn quan hệ giữa các đối tượng.



Một số tri thức về loài "chim sẽ" được biểu diễn trên mạng ngữ nghĩa

III. Phương pháp biểu diễn tri thức (tt)

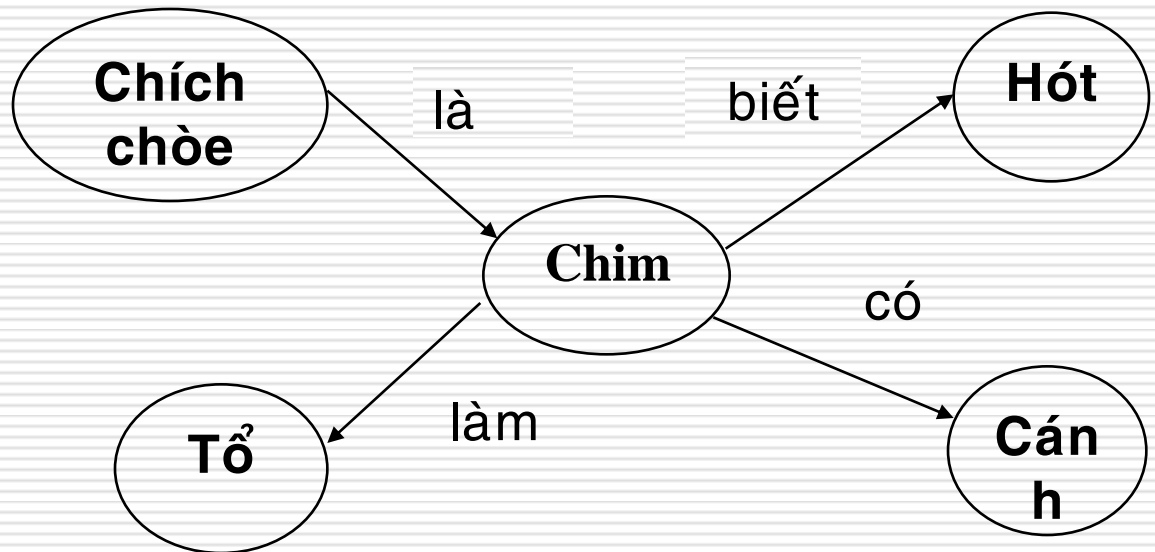
Ví dụ 1: giữa các khái niệm *chích chòe*, *chim*, *hót*, *cánh*, *tổ* có một số mối quan hệ như sau :

Chích chòe là một loài chim.

Chim biết hót

Chim có cánh

Chim sống trong tổ



Các mối quan hệ này sẽ được biểu diễn trực quan bằng một đồ thị bên cạnh

❖ Xem thêm ví dụ về giải bài toán tam giác tổng quát (trong tài liệu tham khảo)

III. Phương pháp biểu diễn tri thức (tt)

Ví dụ 2: Bài toán tam giác tổng quát

❖ Một số bài toán thông thường về tam giác như: "*Cho 3 cạnh của một tam giác, tính chiều dài các đường cao*", "*cho góc a, b và cạnh AC , tính chiều dài các đường trung tuyến*", ...

Tồn tại hay không một chương trình tổng quát có thể giải được tất cả những bài toán tam giác dạng này? Câu trả lời là có.

❖ Bài toán sẽ giải bằng mạng ngữ nghĩa:

Có 22 yếu tố liên quan đến cạnh và góc của tam giác. Để xác định hay để xây dựng một tam giác ta cần 3 yếu tố trong đó có yếu tố cạnh

Sử dụng khoảng 200 đỉnh để chứa công thức + 22 đỉnh để chứa các yếu tố của tam giác.

III. Phương pháp biểu diễn tri thức (tt)

Mạng ngữ nghĩa cho bài toán có cấu trúc như sau

Đỉnh của đồ thị bao gồm 2 loại:

Đỉnh chứa công thức (ký hiệu bằng hình chữ nhật)

Đỉnh chứa yếu tố tam giác (ký hiệu bằng hình tròn)

Cung: chỉ nối từ đỉnh hình tròn đến đỉnh hình chữ nhật cho biết yếu tố tam giác xuất hiện trong công thức nào

Lưu ý: Trong một công thức liên hệ giữa n yếu tố của tam giác, ta giả định rằng nếu đã biết giá trị của $n-1$ yếu tố thì sẽ tính được giá trị của yếu tố còn lại

III. Phương pháp biểu diễn tri thức (tt)

Cơ chế suy diễn thực hiện theo thuật toán “loang” đơn giản sau:

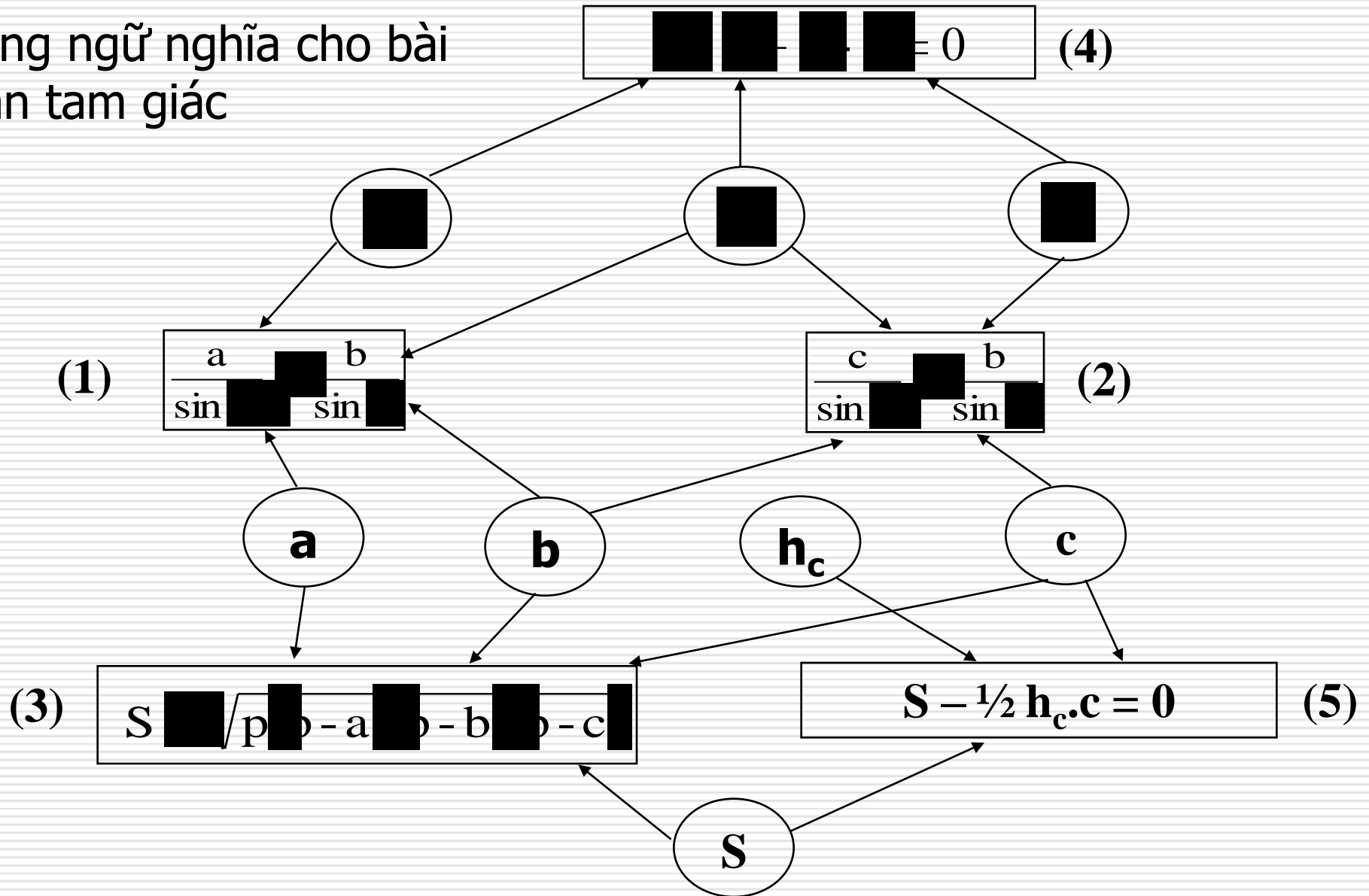
B1: Kích hoạt những đỉnh hình tròn đã cho ban đầu (những yếu tố đã có giá trị)

B2: Lặp lại bước sau cho đến khi kích hoạt được tất cả những đỉnh ứng với những yếu tố cần tính hoặc không thể kích hoạt được bất kỳ đỉnh nào nữa

Nếu một đỉnh hình chữ nhật có cung nối với n đỉnh hình tròn mà $n-1$ đỉnh hình tròn đã được kích hoạt thì kích hoạt đỉnh hình tròn còn lại (và tính giá trị đỉnh còn lại này thông qua công thức ở đỉnh hình chữ nhật).

III. Phương pháp biểu diễn tri thức (tt)

Mạng ngữ nghĩa cho bài toán tam giác



III. Phương pháp biểu diễn tri thức (tt)

- ❖ Ví dụ: Cho hai góc a, b và chiều dài cạnh a của tam giác. Tính chiều dài đường cao h_c . Với mạng ngữ nghĩa đã cho trong hình trên. Các bước thi hành của thuật toán như sau:
 - ✓ Bắt đầu: đỉnh a, b, a được được kích hoạt
 - ✓ Công thức (1) được kích hoạt. Từ (1) tính cạnh b , đỉnh b được kích hoạt.
 - ✓ Công thức (4) được kích hoạt. Từ (4) tính được góc ■ đỉnh ■ được kích hoạt.
 - ✓ Công thức (2) được kích hoạt. Từ (2) tính được cạnh c , kích hoạt đỉnh c .
 - ✓ Công thức (3) được kích hoạt. Từ (3) tính được S , kích hoạt đỉnh S .
 - ✓ Công thức (5) được kích hoạt. Từ (5) tính được h_c , kích hoạt đỉnh h_c
 - ✓ Giá trị h_c được tính, thuật toán kết thúc.

III. Phương pháp biểu diễn tri thức (tt)

Ưu điểm và nhược điểm của mạng ngữ nghĩa

Ưu điểm

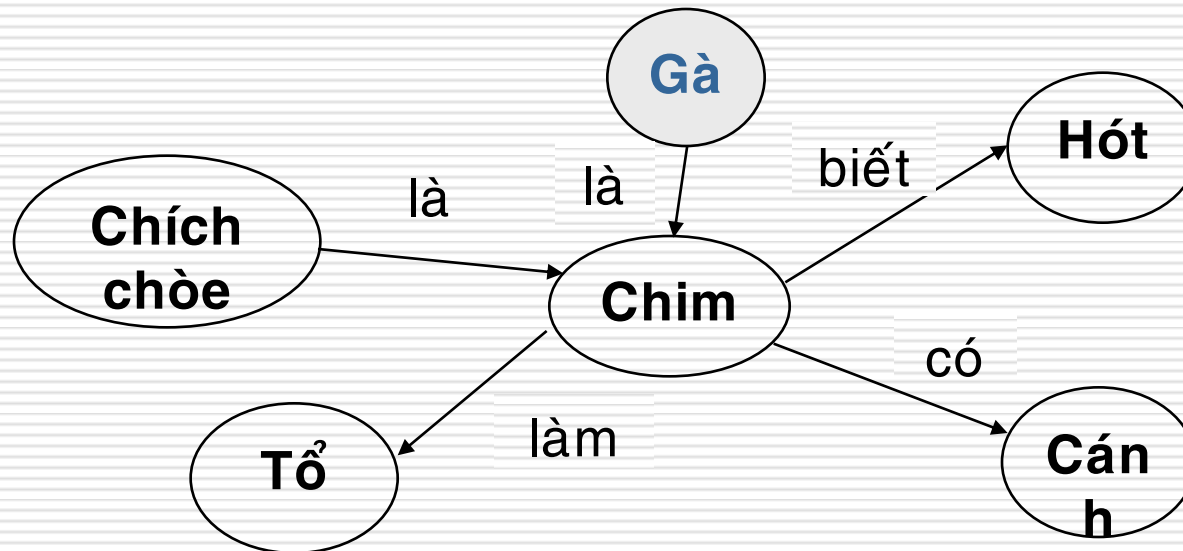
- ✓ Mạng ngữ nghĩa rất linh động, ta có thể dễ dàng thêm vào mạng các đỉnh hoặc cung mới để bổ sung các tri thức cần thiết.
- ✓ Mạng ngữ nghĩa có tính trực quan cao nên rất dễ hiểu.
- ✓ Mạng ngữ nghĩa cho phép các đỉnh có thể thừa kế các tính chất từ các đỉnh khác thông qua các cung loại “là”, từ đó, có thể tạo ra các liên kết “ngầm” giữa những đỉnh không có liên kết trực tiếp với nhau.
- ✓ Mạng ngữ nghĩa hoạt động khá tự nhiên theo cách thức con người ghi nhận thông tin.

III. Phương pháp biểu diễn tri thức (tt)

Ưu điểm và nhược điểm của mạng ngữ nghĩa (tt)

Nhược điểm:

- ✓ Cho đến nay, vẫn chưa có một chuẩn nào quy định các giới hạn cho các đỉnh và cung của mạng. Nghĩa là bạn có thể gán ghép bất kỳ khái niệm nào cho đỉnh hoặc cung!
- ✓ Tính thừa kế (vốn là một ưu điểm) trên mạng sẽ có thể dẫn đến nguy cơ mâu thuẫn trong tri thức.



III. Phương pháp biểu diễn tri thức (tt)

5. Frame:

- ❖ Frame là một cấu trúc dữ liệu chứa đựng tất cả những tri thức liên quan đến một đối tượng cụ thể nào đó.
- ❖ Frames có liên hệ chặt chẽ đến khái niệm hướng đối tượng nên nó thường được sử dụng trong các ngôn ngữ lập trình hướng đối tượng phục vụ cho trí tuệ nhân tạo và các hệ chuyên gia.

III. Phương pháp biểu diễn tri thức (tt)

Frame : XE HƠI

✓ **Thuộc lớp:** phương tiện vận chuyển.

✓ **Tên nhà sản xuất:** Audi

✓ **Quốc gia của nhà sản xuất:** Đức

✓ **Model:** 5000 Turbo

✓ **Loại xe:** Sedan

✓ **Trọng lượng:** 3300lb

✓ **Số lượng cửa:** 4 (default)

✓ **Hộp số:** 3 số tự động

✓ **Số lượng bánh:** 4 (default)

✓ **Máy (tham chiếu frame Máy)**

Kiểu: In-line, overhead cam

Số xy-lanh : 5

✓ **Khả năng tăng tốc**

0-60: 10.4 giây

¼ dặm: 17.1 giây, 85 mph.

III. Phương pháp biểu diễn tri thức (tt)

Frame **MÁY**

Bán kính xy-lanh : 3.19 inch

Tỷ lệ nén : 3.4 inche

Hệ thống xăng : TurboCharger

Mã lực : 140 hp

Torque : 160ft/LB

III. Phương pháp biểu diễn tri thức (tt)

Một ví dụ biểu diễn các đối tượng hình học bằng frame

Các kiểu dữ liệu cơ bản :

Area : numeric; // diện tích

Height : numeric; //chiều cao

Perimeter : numeric; //chu vi

Side : numeric; //cạnh

Diagonal : numeric; //đường chéo

Radius : numeric; //bán kính

Angle : numeric; //góc

Diameter : numeric; //đường kính

pi : (val:numeric = 3.14159)

III. Phương pháp biểu diễn tri thức (tt)

Frame : **CIRCLE** (hình tròn)

r : radius;

s : area;

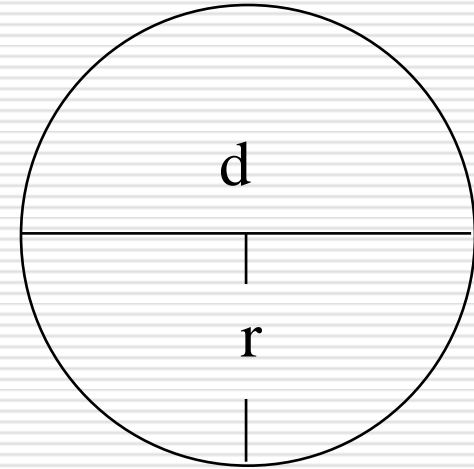
p : perimeter;

d : diameter;

$$d = 2r;$$

$$s = \pi r^2;$$

$$p = 2\pi r;$$



III. Phương pháp biểu diễn tri thức (tt)

Frame **RECTANGLE** (hình chữ nhật)

b_1 : side;

b_2 : side;

s : area;

p : perimeter;

$$s = b_1 \cdot b_2;$$

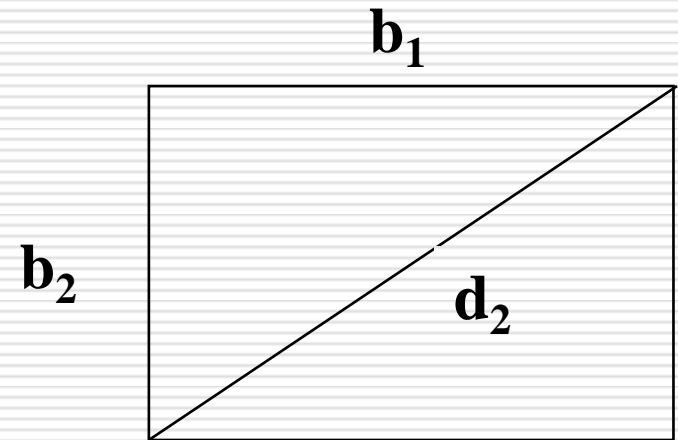
$$p = 2 \cdot (b_1 + b_2);$$

$$d^2 = b_1^2 + b_2^2;$$

Frame **SQUARE** (hình vuông)

Là : **RECTANGLE**

$$b_1 = b_2;$$



III. Phương pháp biểu diễn tri thức (tt)

Frame

RHOMBUS

(hình thoi)

b : side;

d_1 : diagonal;

d_2 : diagonal;

s : area;

p : perimeter;

α_1 : angle;

α_2 : angle;

h : height;

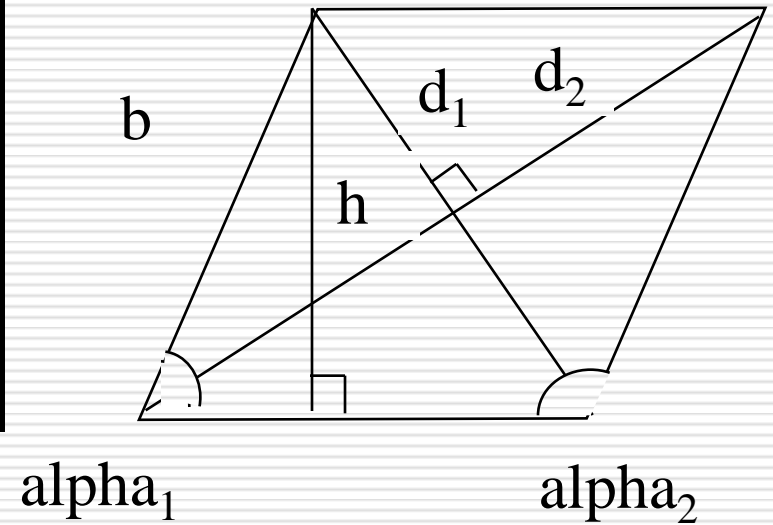
$$\cos(\alpha_2/2) \cdot d_1 = h;$$

$$s = d_1 \cdot d_2 / 2;$$

$$p = 4 \cdot b;$$

$$s = b \cdot h;$$

$$\cos(\alpha_2/2) / (2 \cdot \sin(\alpha_2/2)) = d_2;$$



III. Phương pháp biểu diễn tri thức (tt)

Frame TRIANGLE

a : side;

b : side;

c : side;

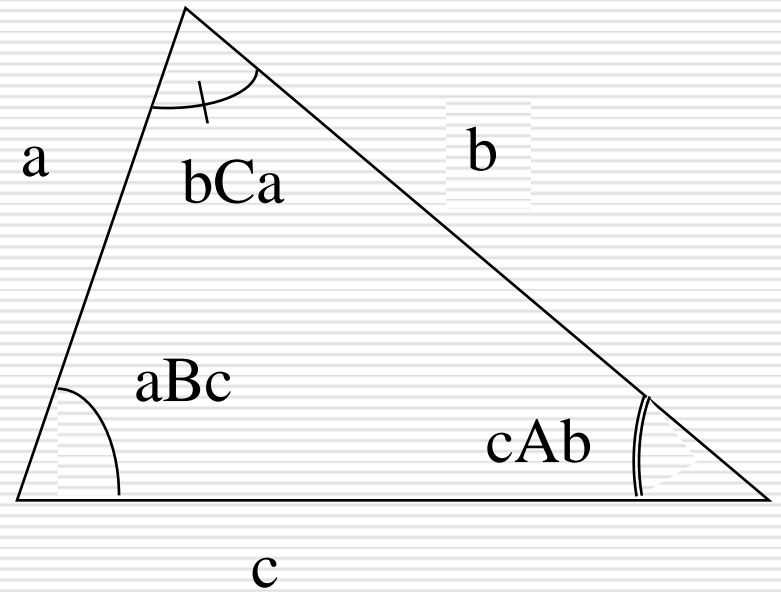
aBc : angle; //góc đối diện cạnh B.

cAb : angle; //góc đối diện cạnh C.

bCa : angle; //góc đối diện cạnh A.

$$s = \sqrt{p(p-a)(p-b)(p-c)} \quad (p = (a+b+c)/2)$$

$$\frac{a}{\sin(cAb)} = \frac{b}{\sin(aBc)} = \frac{c}{\sin(bCa)}$$



III. Phương pháp biểu diễn tri thức (tt)

Frame ARC

Là : **CIRCLE**

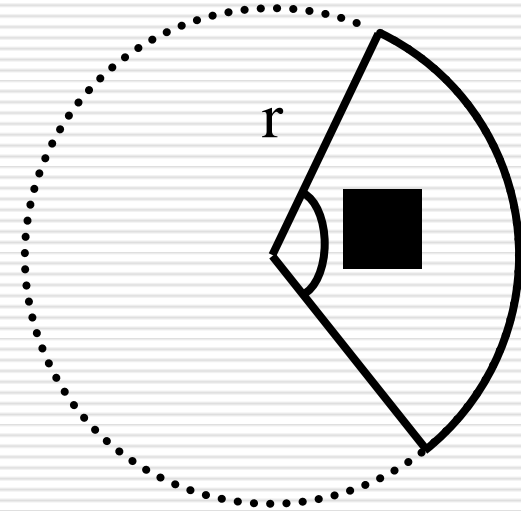
■ angle;

$s = (\text{■} (2 \text{■} \pi)) \text{■} \text{CIRCLE}.s ;$

$p = (\text{■} (2 \text{■} \pi)) \text{■} \text{CIRCLE}.p;$

***CIRCLE.s** là diện tích của vòng tròn chứa cung.*

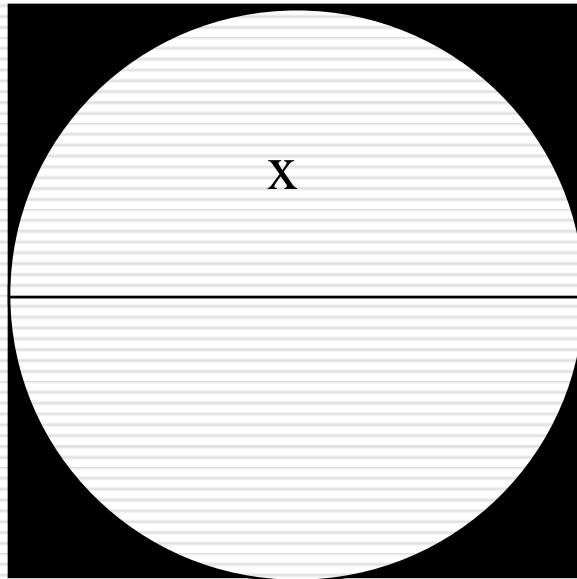
***CIRCLE.p** là chu vi của vòng tròn chứa cung.*



III. Phương pháp biểu diễn tri thức (tt)

Vận dụng để giải các bài toán hình học, chẳng hạn bài toán tính diện tích.

Ví dụ: cho hình vuông **k** và vòng tròn nội tiếp **c**, biết cạnh hình vuông có chiều dài là **x**, hãy viết chương trình để tính diện tích phần tô đen.



III. Phương pháp biểu diễn tri thức (tt)

VAR

x, s : numeric; k : square; c : circle;

BEGIN

<Nhập x>;

k.b₁ := x; { thiết "*hình vuông có cạnh với chiều dài x*" }

c.d := x; { đặc tả giả thiết "*hình tròn nội tiếp*" }

s := k.s – c.s; { tính diện tích bằng cách lấy diện tích hình
vuông trừ cho diện tích hình tròn }

END.

Tóm tắt chương 1: Tiếp nhận, biểu diễn tri thức

- ❖ Tiếp nhận tri thức & phương pháp
- ❖ Biểu diễn tri thức & phương pháp

Chương 3:

Bên trong một hệ

Cơ sở tri thức

I. Hệ cơ sở tri thức (knowledge-based systems) ?

Hệ cơ sở tri thức = Cơ sở tri thức + Động cơ suy diễn



Hệ giải toán = Tiên đề, định lý + Lập luận logic (toán học)



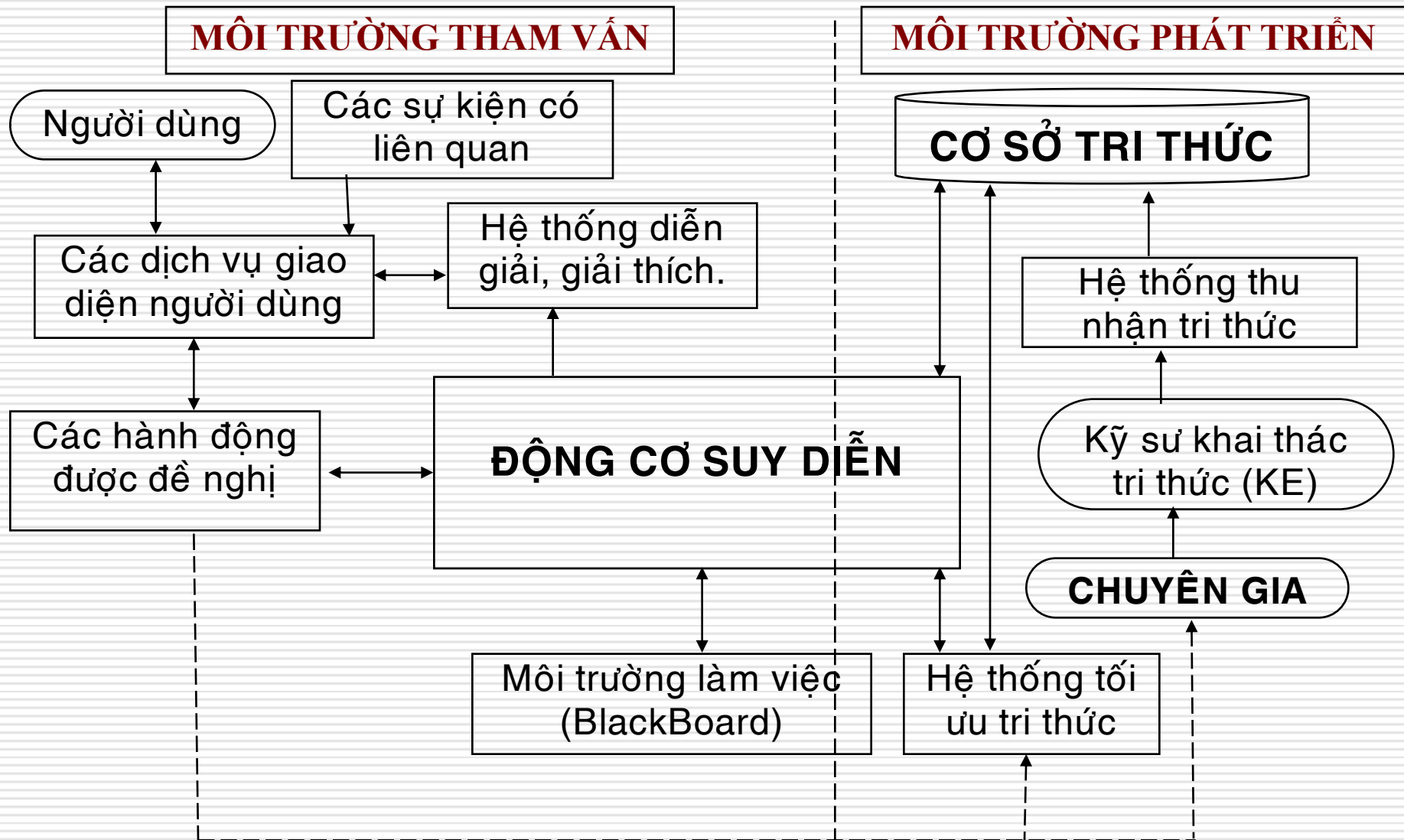
=



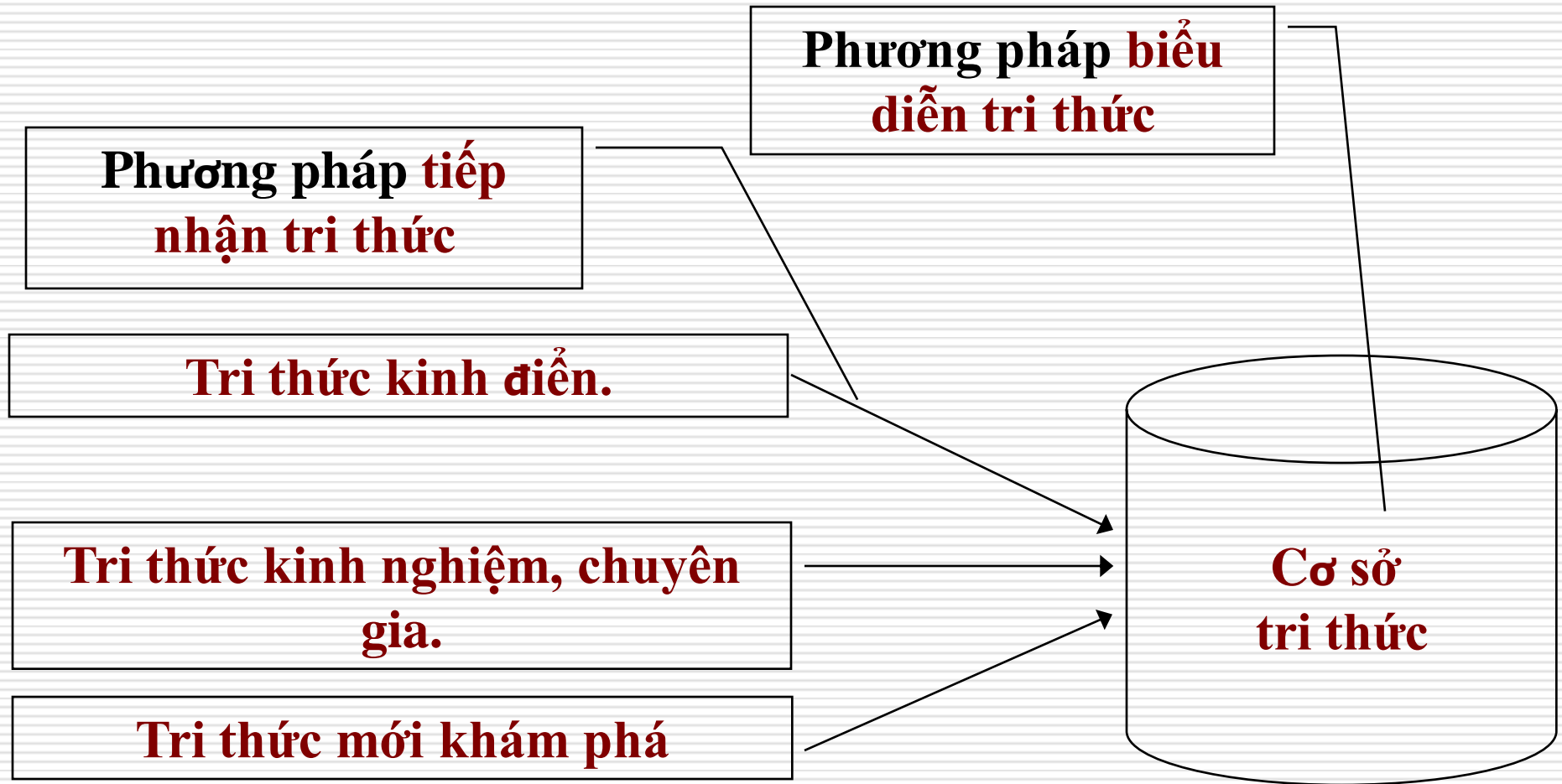
+



II. Cấu trúc chung của một hệ CSTT



III. Cơ sở tri thức



IV. Phương pháp suy diễn

1. Mô hình tổng quát của suy diễn

FACT: Tập sự kiện

HYPO: Tập giả thuyết

Operator MATCH(X, Y) = $\left\{ \begin{array}{l} T \text{ if } X \text{ được lượng giá } T \text{ trong } Y \\ F \text{ if } X \text{ được lượng giá } F \text{ trong } Y \\ ? \text{ If } X \text{ không thể lượng giá trong } Y \end{array} \right.$

- a. Dẫn ra sự kiện mới
- b. Tạo ra giả thuyết mới
- c. Khẳng định hay phủ định giả thuyết
- d. Tiếp nhận FACT mới từ bên ngoài

IV. Phương pháp suy diễn(tt)

a. Dẫn ra sự kiện mới

- (1) **If** MATCH(LHS, FACT) = T **THEN ADD** RHS TO FACT
- (2) **If NOT** MATCH(RHS, FACT) = F **THEN ADD NOT**(LHS) TO FACT

b. Tạo giả thuyết mới

- (3) **If** MATCH(LHS, FACT) = F **THEN ADD NOT**(RHS) TO HYPO
- (4) **If** MATCH(LHS, HYPO) = T **THEN ADD** RHS TO HYPO
- (5) **If** MATCH(LHS, HYPO) = F **THEN ADD NOT**(RHS) TO HYPO
- (6) **If** MATCH(RHS, FACT) = T **THEN ADD** LHS TO HYPO
- (7) **If** MATCH(RHS, HYPO) = T **THEN ADD** LHS TO HYPO
- (8) **If** MATCH(LHS, HYPO) = F **THEN ADD NOT**(LHS) TO HYPO

IV. Phương pháp suy diễn(tt)

c. Khẳng định hay phủ định giả thuyết

(9) **If MATCH (hypo.FACT) = T THEN ADD hypo TO HYPO**

(10) **If MATCH (hypo.FACT) = F THEN DELETE hypo TOHYPO**

d. Tiếp nhận FACT mới từ bên ngoài

GET (FACT)

[] : Lặp lại nhiều lần

{ } : Tùy chọn

Lập luận tiến: [(1)]

Lập luận lùi: (6) + [(7)] + {d} + (9) + [(1)]

Lập luận phản chứng: [(4)] + {d} + (10) + [(2)]

IV. Phương pháp suy diễn(tt)

2. Suy diễn tiến : là quá trình suy luận xuất phát từ một số sự kiện ban đầu, xác định các sự kiện có thể được “sinh” ra từ sự kiện này.

Ví dụ : Cho 1 cơ sở tri thức được xác định như sau :

Các sự kiện : A, B, C, D, E, F, G, H, K

Tập các quy tắc hay luật sinh (rule)

{ R1 : A \rightarrow E; R2 : B \rightarrow D;
 R3 : H \rightarrow A; R4 : E \wedge G \rightarrow C;
 R5 : E \wedge K \rightarrow B; R6 : D \wedge E \wedge K \rightarrow C;
 R7 : G \wedge K \wedge F \rightarrow A;
}

IV. Phương pháp suy diễn(tt)

Ví dụ: (tt) (suy diễn tiến)

Sự kiện ban đầu : H, K

$$\mathbf{R3 : H \rightarrow A \quad \{A, H, K\}}$$

$$\mathbf{R1 : A \rightarrow E \quad \{A, E, H, K\}}$$

$$\mathbf{R5 : E \wedge K \rightarrow B \quad \{A, B, E, H, K\}}$$

$$\mathbf{R2 : B \rightarrow D \quad \{A, B, D, E, H, K\}}$$

$$\mathbf{R6 : D \wedge E \wedge K \rightarrow C \quad \{A, B, C, D, E, H, K\}}$$

Tập hợp $\{A, B, C, D, E, H, K\}$ được gọi là bao đóng của tập $\{H, K\}$ trên tập luật R (gồm 7 luật như trên).

IV. Phương pháp suy diễn(tt)

3. Suy diễn lùi: là quá trình suy luận ngược xuất phát từ một số sự kiện ban đầu, ta tìm kiếm các sự kiện đã “sinh” ra sự kiện này. Một ví dụ thường gặp trong thực tế là xuất phát từ các tình trạng của máy tính, chẩn đoán xem máy tính đã bị hỏng hóc ở đâu.

Ví dụ: Tập các sự kiện :

Ổ cứng là “hỏng” hay “hoạt động bình thường”

Hỏng màn hình.

Lỏng cáp màn hình.

Tình trạng đèn ổ cứng là “tắt” hoặc “sáng”

Có âm thanh đọc ổ cứng.

Tình trạng đèn màn hình “xanh” hoặc “chớp đỏ”

Điện vào máy tính “có” hay “không”

IV. Phương pháp suy diễn(tt)

Ví dụ: (tt) (chẩn đoán hỏng máy tính)

Một số luật suy diễn :

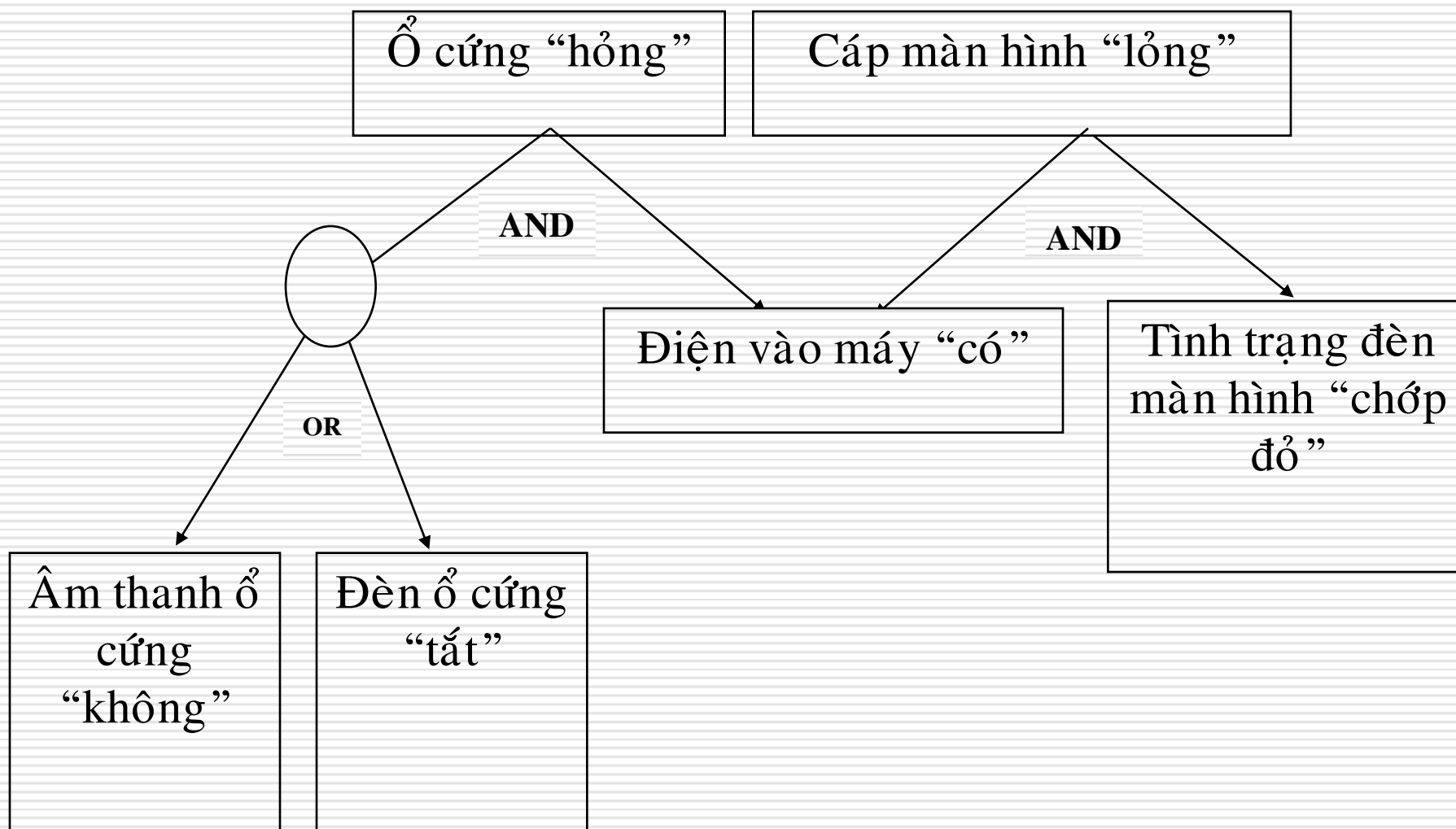
R1. Nếu (điện vào máy là “có”) và (âm thanh đọc ổ cứng là “không”) thì (ổ cứng “hỏng”).

R2. Nếu (điện vào máy là “có”) và (tình trạng đèn ổ cứng là “tắt”) thì (ổ cứng “hỏng”).

R3. Nếu (điện vào máy là “có”) và (tình trạng đèn màn hình là “chớp đỏ”) thì (cáp màn hình “lỏng”).

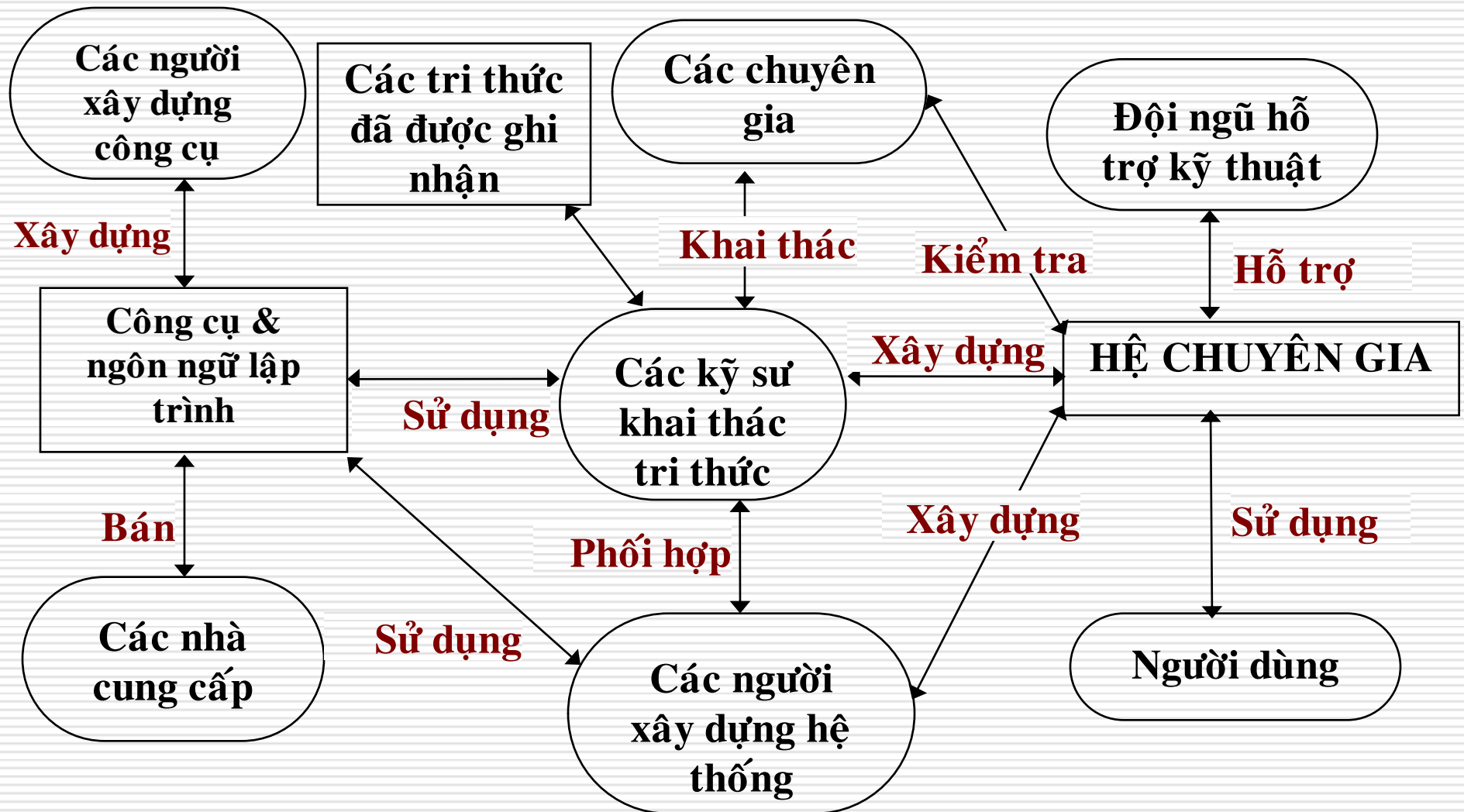
Để xác định được các nguyên nhân gây ra sự kiện “không sử dụng được máy tính”, ta phải xây dựng một cấu trúc đồ thị gọi là đồ thị AND/OR như sau :

IV. Phương pháp suy diễn(tt)



V. Xây dựng hệ CSTT

1. Tổng quan quá trình xây dựng hệ CSTT



V. Xây dựng hệ CSTT (tt)

2. Một số bước cơ bản để xây dựng hệ cơ sở tri thức

- ❖ Tiếp cận chuyên gia

- ❖ Tổ chức thu thập tri thức

- ❖ Chọn lựa công cụ phát triển hệ cơ sở tri thức
 - ✓ Chọn ngôn ngữ lập trình trí tuệ nhân tạo (LISP, PROLOG, ...)
 - ✓ Các ngôn ngữ lập trình thông dụng
 - ✓ Các hệ cơ sở tri thức rỗng (shell): là một công cụ lai giữa hai loại trên

- ❖ Cài đặt hệ CSTT

VI. Cài đặt hệ CSTT

1. Vài nét về PROLOG

Prolog (PROgramming in Logic) là một ngôn ngữ lập trình dạng khai báo

1.1 Mô tả các vị từ: Cơ sở tri thức của Prolog bao gồm các vị từ, có thể mô tả các khái niệm sau:

Sự kiện:

Cú pháp: <thuộc tính>(<đối tượng>)

Quả chanh có màu xanh → Xanh(Chanh)

Mối liên hệ giữa các đối tượng

Cú pháp: <quan hệ> (<đối tượng 1>, ..., <đối tượng n>)

An yêu Bình → Yêu(An, Bình)

VI. Cài đặt hệ CSTT (tt)

Cấu trúc giữa các đối tượng

Cú pháp: <đối tượng> (<thành phần 1>, ..., <thành phần n>)

Xe máy hiệu Dream, 110 phân khối, màu nâu, 4 số, giá 30 triệu.

→ Xe máy(Dream, 110, nâu, 4, 30)

Các luật

Cú pháp: <luật>(<đối tượng 1>, ..., <đối tượng n>) :- <điều kiện 1>, ..., <điều kiện n>

A là chim nếu A có cánh và A biết bay

→ Chim(A) :- CóCánh(A), BiếtBay(A).

Dùng dấu phẩy (,) để biểu diễn toán tử AND, dấu chấm phẩy (;) để biểu diễn toán tử OR và toán tử không bằng là \=

VI. Cài đặt hệ CSTT (tt)

Ví dụ:

A là tổ tiên của B nếu:

A là cha mẹ của B (phần kết thúc)

A là cha mẹ của C và C là tổ tiên của B.

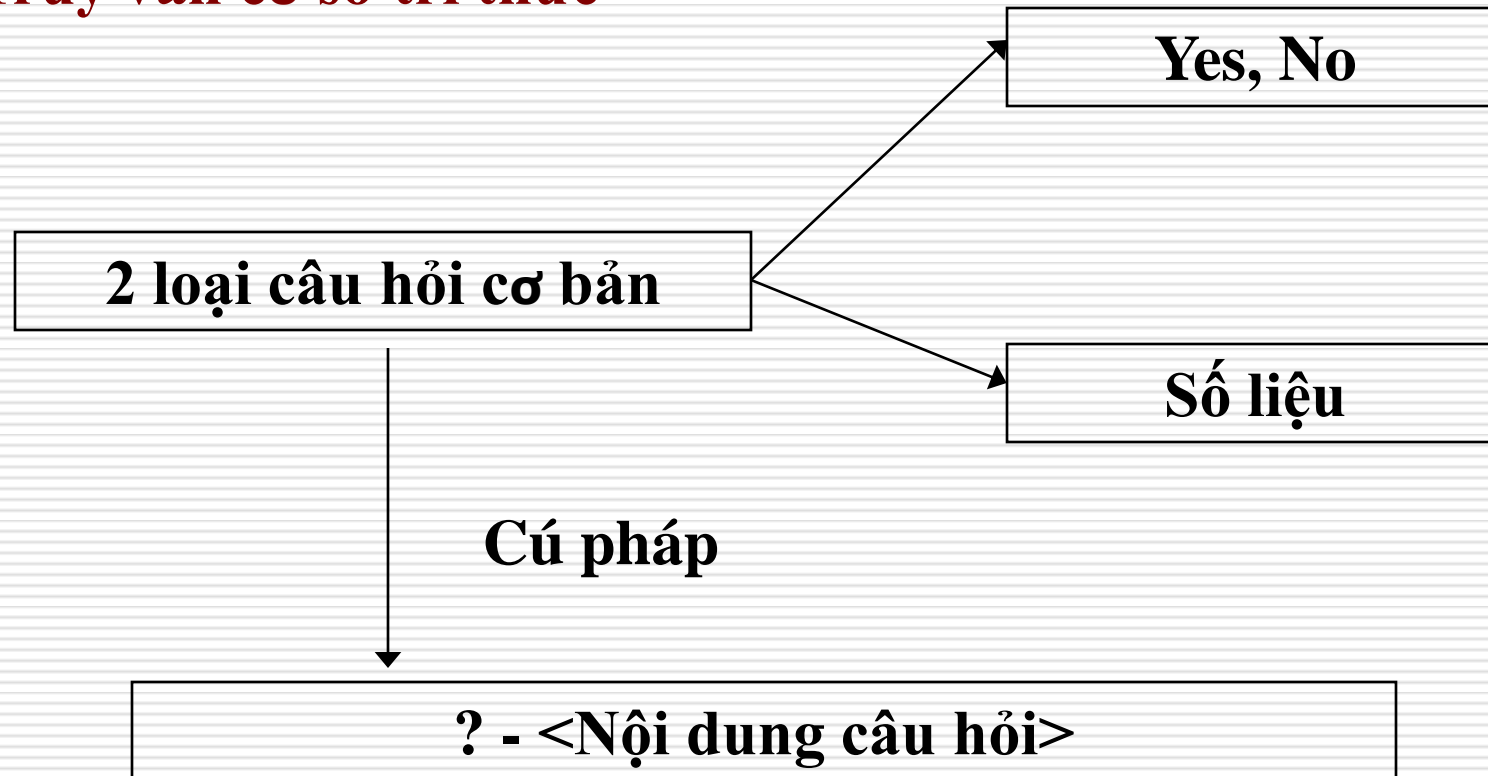
Ta định nghĩa luật như sau :

ToTien(A,B) :- ChaMe(A,B).

ToTien(A,B) :- ChaMe(A,C), ToTien(C,B).

VI. Cài đặt hệ CSTT (tt)

1.2 Truy vấn cơ sở tri thức



VI. Cài đặt hệ CSTT (tt)

Ví dụ : Quả chanh có màu xanh là đúng hay sai ?

→ ?- Xanh(Chanh)

Ví dụ : Nếu ta có khai báo hai vị từ là : $\text{Yeu}(\text{An}, \text{Binh})$, $\text{Yeu}(\text{An}, \text{Chau})$

An yêu ai ? → $\text{Yeu}(\text{An}, \text{X})$

Hệ thống sẽ trả lời là :

$\text{X} \rightarrow \text{Binh}$

$\text{X} \rightarrow \text{Chau}$

2 Solution(s)

VI. Cài đặt hệ CSTT (tt)

2. Cài đặt một hệ CSTT về tình trạng gia đình bằng ngôn ngữ Prolog

2.1 Mô tả các sự kiện trong quan hệ gia đình

married(philip, elizabeth).

married(mark, anne).

married(charles, diana).

married(tim, anne).

divorced(mark, anne).

parents(philip, elizabeth, charles).

parents(philip, elizabeth, anne).

parents(philip, elizabeth, andrew).

parents(philip, elizabeth, edward).

parents(mark, anne, zara).

parents(mark, anne, peter).

parents(charles, diana, william).

parents(charles, diana, harry).

parents(andrew, sarah, eugene).

parents(andrew, sarah, beatrice).

VI. Cài đặt hệ CSTT (tt)

gender(philip, male).

gender(elizabeth, female).

gender(charles, male).

gender(andrew, male).

gender(edward, male).

gender(mark, male).

gender(tim, male).

gender(diana, female).

gender(sarah, female).

gender(zarah, female).

gender(peter, male).

gender(william, male).

gender(harry, male).

gender(eugene, female).

gender(beatrice, female).

VI. Cài đặt hệ CSTT (tt)

2.2 Định nghĩa các quan hệ gia đình khác dựa trên các sự kiện đã nêu

parent_of(Parent, Child) :- parents(Parent, _, Child).

parent_of(Parent, Child) :- parents(_, Parent, Child).

sibling_of(S, Person) :- parents(F, M, S), parents(F, M, Person), S \= Person.

father_of(Father, Child) :- parents(Father, _, Child).

sister_of(S, Person) :- gender(S, female), sibling_of(S, Person).

grand_parent_of(GP, GC) :- parent_of(GP, Temp), parent_of(Temp, GC).

ancesstor_of(Ancesor, Person) :- parent_of(Ancesor, Person).

ancesstor_of(Ancesor, Person) :-

ancesstor_of(Ancesor, Temp), parent_of(Temp, Person).

cousin_of(A, B) :- parent_of(P1, A), parent_of(P2, B), sibling_of(P1, P2).

VI. Cài đặt hệ CSTT (tt)

2.3 Suy luận

Chẳng hạn khi muốn đặt ra câu hỏi "Ai là chị của andrew" và bật chức năng TRACE cho phép dò theo quá trình suy luận của PROLOG ta sẽ được hiển thị các thông tin sau :

?- sister_of(S, andrew)

Đầu tiên, hệ thống sẽ tìm giá trị S thỏa điều kiện `gender(S, female)`. Quá trình tìm kiếm sẽ dừng lại ở sự kiện `gender(elizabeth, female)`.

CALL gender(S, female) ... succeeds; S → elizabeth

Do đó, sự kiện `sibling_of(elizabeth, andrew)` được đánh giá là sai.

FAIL sibling_of(elizabeth, andrew)

VI. Cài đặt hệ CSTT (tt)

Hệ thống tìm một giá trị S khác thỏa điều kiện $\text{gender}(S, \text{female})$.
Quá trình tìm kiếm sẽ dừng lại ở sự kiện $\text{gender}(\text{anne}, \text{female})$.

REDO $\text{gender}(S, \text{female})$... succeeds; $S \rightarrow \text{anne}$

Hệ thống tìm A, B thỏa điều kiện tiếp theo là $\text{parents}(A, B, \text{elizabeth})$. Quá trình tìm kiếm sẽ dừng lại ở sự kiện $\text{parents}(\text{philip}, \text{elizabeth}, \text{anne})$

CALL $\text{sibling_of}(\text{anne}, \text{andrew})$

CALL $\text{parents}(A, B, \text{anne})$

... succeeds; $A \rightarrow \text{philip}, B \rightarrow \text{elizabeth}$

Hệ thống kiểm tra điều kiện cuối cùng $S \neq \text{andrew}$

CALL $\text{anne} \neq \text{andrew}$... succeeds

VI. Cài đặt hệ CSTT (tt)

Như vậy là vị từ `subling_of(anne, andrew)` có giá trị đúng.

EXIT `subling_of(anne, andrew)`

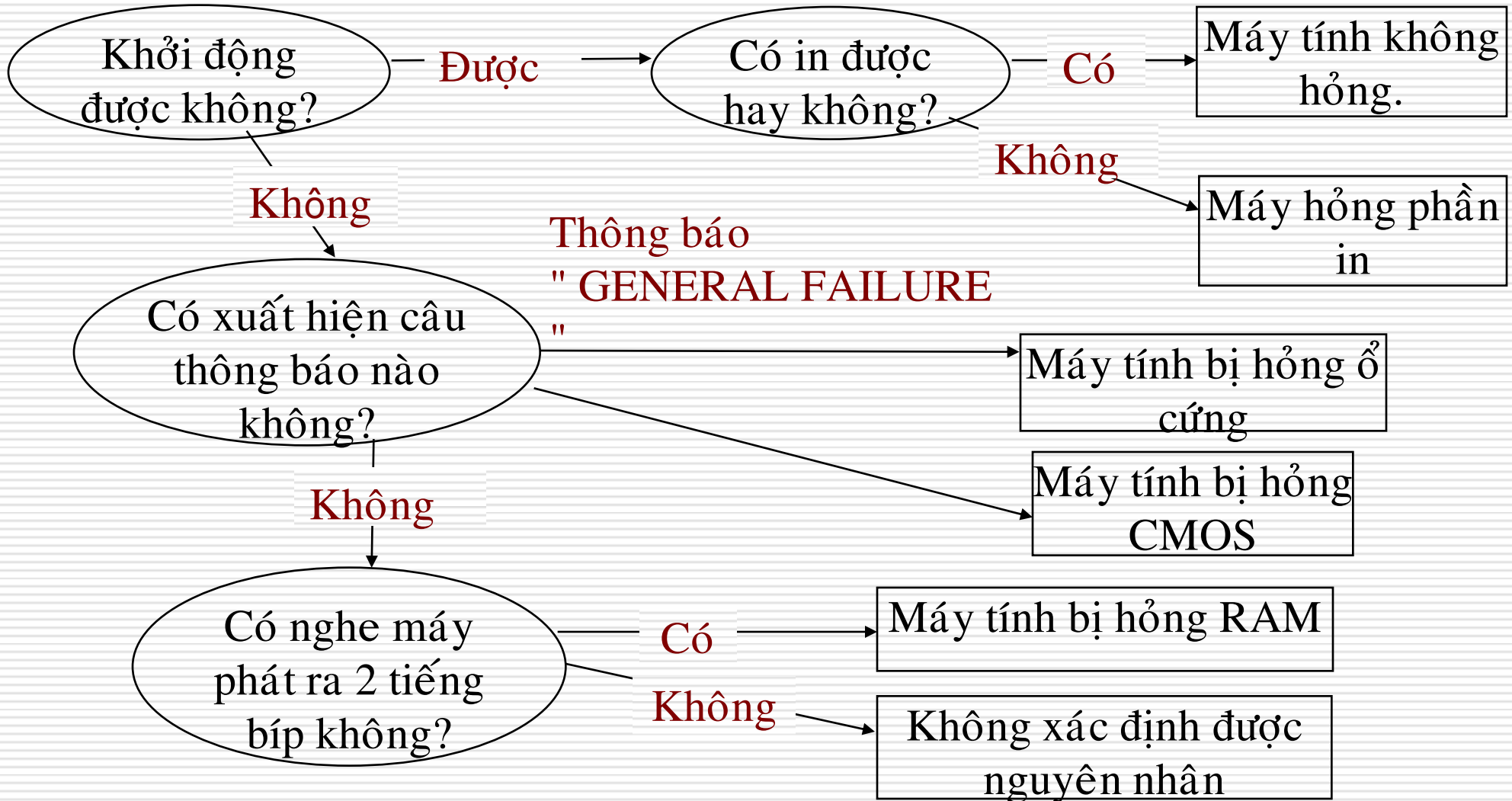
Kết luận là `anne` là chị của `andrew`.

EXIT `sister_of(anne, andrew)`

VI. Cài đặt hệ CSTT (tt)

3. Cài đặt hệ CSTT bằng ngôn ngữ lập trình thông thường

❖ Giả sử hệ CSTT của chúng ta hoạt động theo cây quyết định sau:



VI. Cài đặt hệ CSTT (tt)

3.1 Biểu diễn tri thức dưới dạng luật dẫn

Tên biến	Miền giá trị
KHOIDONG	DUOC, KHONG
IN	DUOC, KHONG
THONGBAO	HDD, GENERAL, KHONG
AMTHANH	CO, KHONG
HONG	KHONG, IN, HDD, CMOS, RAM, UNKNOWN

VI. Cài đặt hệ CSTT (tt)

Tập luật dẫn ban đầu có được từ cây quyết định trên sẽ như sau :

1. **IF** (KHOIDONG = DUOC) **AND** (IN = DUOC) **THEN** HONG = KHONG.
2. **IF** (KHOIDONG = DUOC) **AND** (IN = KHONG) **THEN** HONG = IN
3. **IF** (KHOIDONG = KHONG) **AND** (THONGBAO = HDD)
THEN HONG = HDD
4. **IF** (KHOIDONG = KHONG) **AND** (THONGBAO = GENERAL)
THEN HONG = CMOS
5. **IF** (KHOIDONG = KHONG) **AND** (THONGBAO = KHONG) **AND** (AMTHANH
= CO) **THEN** HONG = RAM
6. **IF** (KHOIDONG = KHONG) **AND** (THONGBAO = KHONG) **AND** (AMTHANH
= KHONG) **THEN** HONG = UNKNOWN

VI. Cài đặt hệ CSTT (tt)

Tập luật có thể viết lại như sau :

(không khởi động và không thông báo → KH_KDTB)

- **IF (KHOIDONG = DUOC) AND (IN = DUOC) THEN HONG = KHONG.**
- **IF (KHOIDONG = DUOC) AND (IN = KHONG) THEN HONG = IN**
- **IF (KHOIDONG = KHONG) AND (THONGBAO = HDD) THEN HONG = HDD**
- **IF (KHOIDONG = KHONG) AND (THONGBAO = GENERAL) THEN HONG = CMOS**
- **IF (KHOIDONG = KHONG) AND (THONGBAO=KHONG) THEN KH_KDTB = DUNG**
- **IF (KH_KDTB = DUNG) AND (AMTHANH = CO) THEN HONG = RAM**
- **IF (KH_KDTB = DUNG) AND (AMTHANH = KHONG)**
- **THEN HONG = UNKNOWN**

VI. Cài đặt hệ CSTT (tt)

3.2 Lưu trữ và phân loại biến

- ❖ **Biến nhập** : là các biến chỉ xuất hiện ở vế trái của các luật
- ❖ **Biến trung gian** : là các biến xuất hiện ở cả vế trái lẫn vế phải ở các luật
- ❖ **Biến xuất**: các biến chỉ xuất hiện ở vế phải ở các luật

VI. Cài đặt hệ CSTT (tt)

Tên biến	Khởi tạo	Giá trị	Loại	Câu thông báo
KHOIDONG	FALSE		INPUT	Máy tính có khởi động được không?
IN	FALSE		INPUT	Máy tính có in được không?
THONGBAO	FALSE		INPUT	Máy tính có thông báo gì không?
AMTHANH	FALSE		INPUT	Máy tính có phát ra âm thanh gì không?
HONG	FALSE		OUTPUT	Máy bị hỏng ở phần
KH_KDTB	FALSE		TEMP	

VI. Cài đặt hệ CSTT (tt)

3.3 Lưu trữ luật

- ❖ Để lưu trữ một luật, ta cần lưu trữ các biến tham gia vào vế trái cùng với giá trị của các biến đó (để kích hoạt luật).
- ❖ Vế phải của luật chỉ bao gồm một biến nên khá đơn giản ta chỉ việc thêm một cột tên biến và giá trị của biến sẽ được đặt khi luật cháy gọi là giá trị cháy vào bảng Vế Phải sau:

Luật	Biến	Giá trị cháy
1	HONG	KHONG
2	HONG	IN
3	HONG	HDD
4	HONG	CMOS
5	KH_KDTB	DUNG
6	HONG	RAM
7	HONG	UNKNOWN

VI. Cài đặt hệ CSTT (tt)

❖ Để mô tả về trái của luật, ta dùng bản Vế Trái với 3 cột như sau:

Luật	BIẾN	Giá Trị Cháy
1	KHOIDONG	DUOC
1	IN	DUOC
2	KHOIDONG	DUOC
2	IN	KHONG
3	KHOIDONG	KHONG
3	THONGBAO	HDD
4	KHOIDONG	KHONG
4	THONGBAO	GENERAL
5	KHOIDONG	KHONG
5	THONGBAO	KHONG
6	KH_TDTB	DUNG
6	AMTHANH	CO
7	KH_TDTB	DUNG
7	AMTHANH	KHONG

VI. Cài đặt hệ CSTT (tt)

❖ Với các cấu trúc trên, tại mọi thời điểm, ta đều có thể truy xuất đến mọi thuộc tính của các luật.

❖ Sau đây là các ký hiệu :

<luật>.Chay : cho biết luật có cháy hay chưa.

<luật>.VePhai.Bien : biến ở vế phải của luật.

<luật>.VePhai.GiaTriChay : giá trị cháy ứng với biến ở vế phải của luật.

<luật>.VeTrai.SoBien : số lượng biến trong vế trái của luật.

<luật>.VeTrai.Bien[i] : biến thứ i ở vế trái của luật.

<luật>.VeTrai.GiaTriChay[i] : giá trị cháy ứng với biến thứ i ở vế trái của luật.

VI. Cài đặt hệ CSTT (tt)

3.4 Hàm kích hoạt luật

```
FUNCTION KichHoatLuat(L : Luat) : BOOLEAN
```

```
BEGIN
```

```
    IF L.Chay = TRUE THEN
```

```
        RETURN FALSE; { Luật đã cháy rồi, không kích hoạt được }
```

```
    Fire = TRUE;
```

```
    FOR i = 1 TO L.VeTrai.SoBien BEGIN
```

```
        v = L.VeTrai.Bien[i];
```

```
        { có một biến không thỏa điều kiện cháy }
```

```
        IF (v.KhoiTao = FALSE) OR
```

```
            (v.GiaTri  $\neq$  L.VeTrai.GiaTriChay[i]) THEN BEGIN
```

```
                Fire = FALSE; EXIT FOR;
```

```
            END;
```

```
    END;
```

```
    If Fire = TRUE THEN
```

```
        L.VePhai.Bien.ThuocTinh.GiaTri = L.VePhai.Bien.GiaTriChay;
```

```
    RETRUN Fire;
```

```
END;
```

VI. Cài đặt hệ CSTT (tt)

3.5 Cài đặt thuật toán suy diễn lùi

FUNCTION TinhGiaTriBien(V : Bien, L : Luat)

{ Tính giá trị của biến V trong trái của luật L }

BEGIN

IF (V.KhoiTao = TRUE) THEN RETURN;

ELSE BEGIN

IF V.Loai = INPUT THEN BEGIN

<Hỏi người dùng giá trị biến V>;

RETURN;

END;

ELSE BEGIN

FOR EACH LT IN TapLuat DO

IF (LT.VePhai = V) THEN BEGIN

FOR i = 1 TO LT.VeTrai.SoBien DO BEGIN

TinhGiaTriBien(LT.VeTrai.Bien[i], LT);

END;

IF KichHoatLuat(LT) THEN RETURN;

END;

END;

END;

END

VI. Cài đặt hệ CSTT (tt)

Trong ví dụ của chúng ta, để biết giá trị biến HONG, ta có thể thực hiện như sau :

{ Khởi động trạng thái ban đầu cho tập biến và tập luật. }

FOR EACH $v \in \text{TapBien}$ $v.\text{KhoiTao} = \text{FALSE}$;

FOR EACH $LT \in \text{TapLuot}$ $LT.\text{Chay} = \text{FALSE}$

$V = \text{HONG}$;

{ Luật 0 là một luật rỗng, dùng để "đệm" cho lần đệ quy đầu tiên, luôn cháy }

TinhGiaTriBien($V, 0$);

IF $V.\text{KhoiTao} = \text{FALSE}$ THEN

<Không tính được giá trị biến>;

ELSE

<Hiển thị giá trị V >;

VI. Cài đặt hệ CSTT (tt)

3.6 Cài đặt thuật toán suy diễn tiến

Thuật toán suy diễn tiến rất đơn giản. Chẳng hạn, xuất phát từ ba trạng thái là KHOIDONG = KHONG, THONGBAO = KHONG, AMTHANH = CO ta có thể kết luận được điều gì ?

<Cập nhật các trạng thái khởi tạo là TRUE và giá trị của các biến đã biết>;

CapNhat = TRUE;

LSET = TapLuot;

WHILE CapNhat DO BEGIN

 CapNhap = FALSE;

 FOR EACH LT \in LSET DO

 IF KichHoatLuot(LT) = TRUE THEN BEGIN

 {Bỏ những luật đã cháy ra khỏi tập luật.}

 LSET = LSET \ LT;

 CapNhap = TRUE;

 END;

END

VI. Cài đặt hệ CSTT (tt)

❖ Mục đích thường gặp của quá trình suy diễn tiến là xác định giá trị của tất cả biến xuất. Do đó, ta có thể xem đây là một trường hợp đặc biệt của suy diễn lùi.

Như vậy, chỉ cần bỏ dòng <Hỏi người dùng giá trị biến V>;

❖ Trong cài đặt ở phần suy diễn lùi là ta có thể dùng lại hàm `TinhGiaTriBien` để cài đặt thuật toán suy diễn tiến một cách vô cùng đơn giản (!) như sau :

<Cập nhật các trạng thái khởi tạo là TRUE và giá trị của các biến nhập đã biết>;

FOR EACH $LT \in \text{TapLuot DO}$

`TinhGiaTriBien(LT.VePhai.Bien,LT);`

Phần II: Các hệ Cơ sở tri thức (knowledge-based systems)

Chương 4:

Phân loại Các hệ cơ sở tri thức

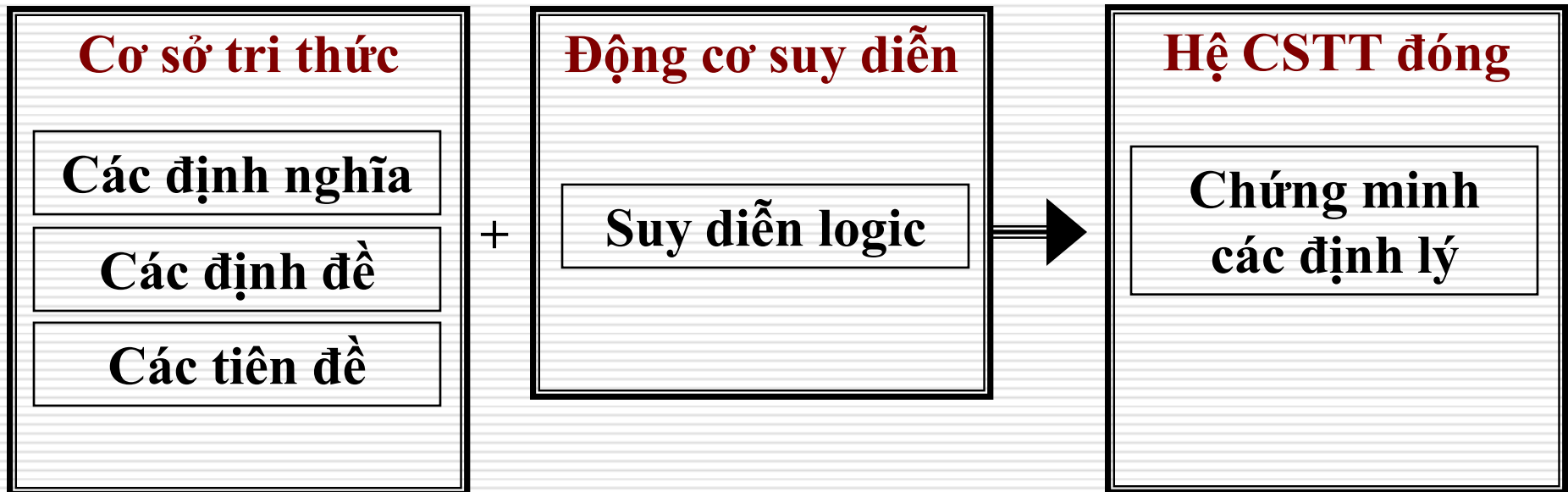
I. Một số tiêu chuẩn phân loại các hệ CSTT

- ❖ Tính đóng, mở, kết hợp
- ❖ Phương pháp biểu diễn tri thức
- ❖ Lĩnh vực ứng dụng

II. Hệ CSTT đóng

❖ **Hệ cơ sở tri thức đóng:** được xây dựng với một số “tri thức lĩnh vực” ban đầu và chỉ với những tri thức đó mà thôi trong suốt quá trình hoạt động hay suốt thời gian sống của nó.

Ví dụ: Các định nghĩa và các tiên đề trong tác phẩm của Oclit.



II. Hệ CSTT đóng (tt)

1. Các định nghĩa và các tiên đề trong tác phẩm của Oclit

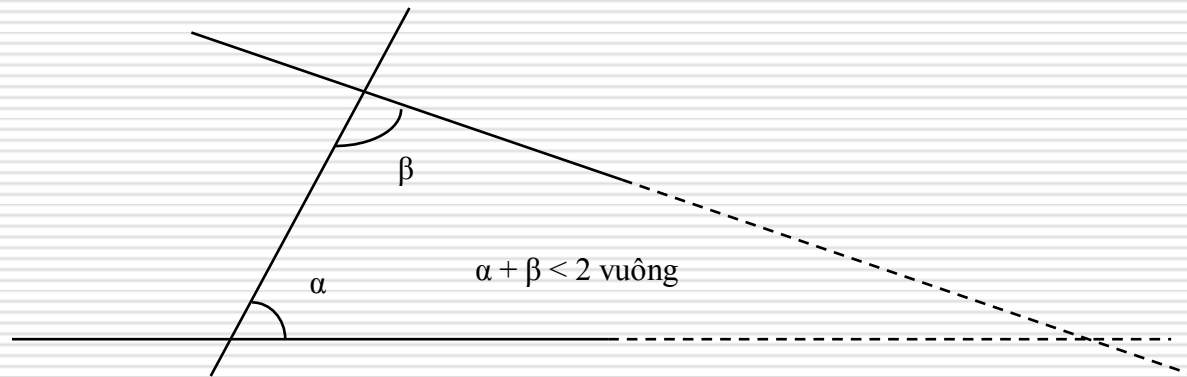
Điểm là cái gì không có bộ phận

- ❖ Đường có bề dài và không có bề rộng
- ❖ Các đầu mút của một đường là những điểm
- ❖ Đường thẳng là đường có sự sắp đặt vị trí như nhau đối với mọi điểm của nó
- ❖ Mặt là cái chỉ có bề dài và bề rộng
- ❖ Các biên của một mặt là những đường
- ❖ Mặt phẳng là mặt có sự sắp đặt vị trí như nhau đối với mọi đường thẳng của nó
- ❖ ...

II. Hệ CSTT đóng (tt)

Các định đề

- ❖ Từ một điểm bất kỳ này đến một điểm bất kỳ khác có thể vẽ một đường thẳng.
- ❖ Một đường thẳng có thể kéo dài ra vô hạn.
- ❖ Từ một điểm bất kỳ làm tâm, và với một bán kính tùy ý, có thể vẽ một đường tròn.
- ❖ Tất cả các góc vuông đều bằng nhau.
- ❖ Nếu một đường thẳng cắt hai đường thẳng khác tạo nên hai góc trong cùng phía có tổng nhỏ hơn hai vuông thì hai đường thẳng đó phải cắt nhau về phía có hai góc nói trên đối với đường thẳng cắt.



II. Hệ CSTT đóng (tt)

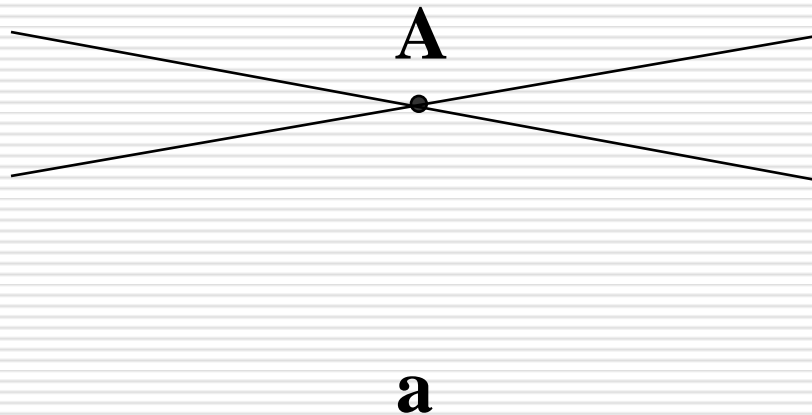
Các tiên đề

- ❖ Hai cái cùng bằng cái thứ ba thì bằng nhau.
- ❖ Thêm những cái bằng nhau vào những cái bằng nhau thì được những cái bằng nhau.
- ❖ Bớt những cái bằng nhau từ những cái bằng nhau thì được những cái bằng nhau.
- ❖ Các hình chồng khít lên nhau thì bằng nhau.
- ❖ Toàn thể lớn hơn một phần

II. Hệ CSTT đóng (tt)

2. Tiên đề Lobasepxki (tiên đề V')

- Trong mặt phẳng xác định bởi đường thẳng a và một điểm A không thuộc đường thẳng đó có ít ra là 2 đường thẳng đi qua A và không cắt a .



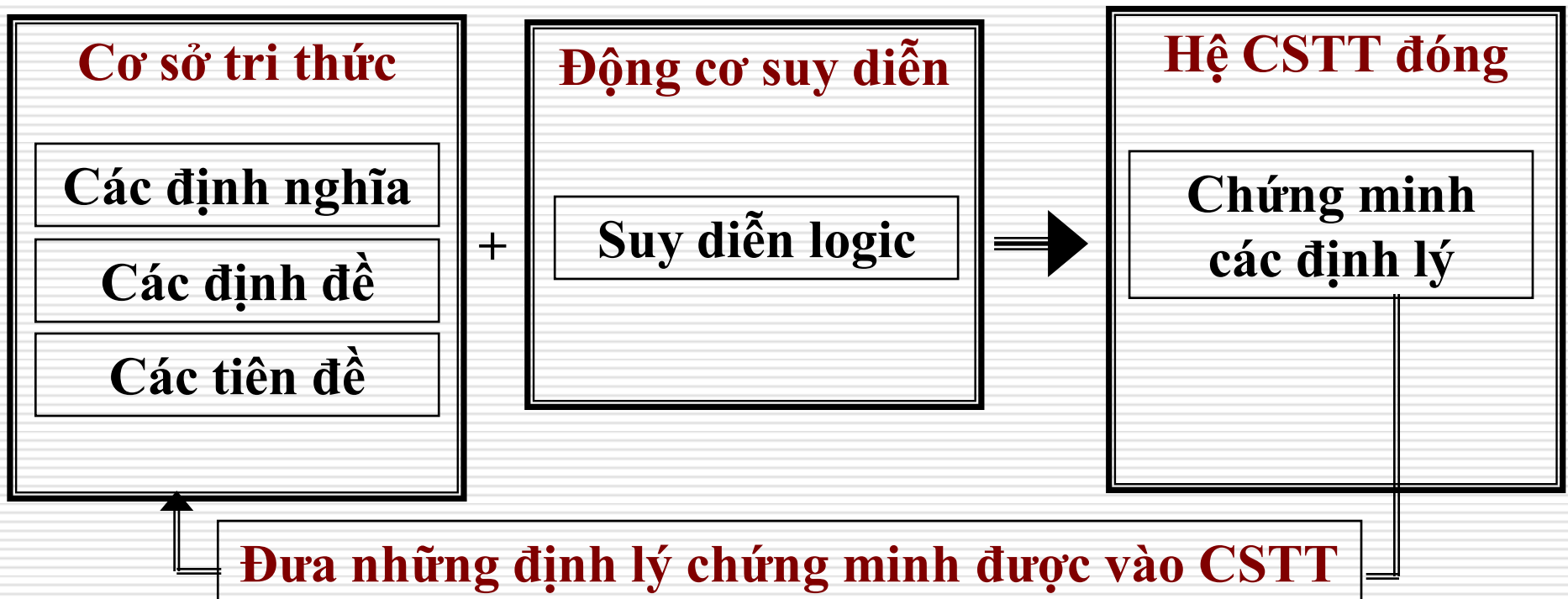
❖ Tiên đề V' phủ định tất cả các mệnh đề tương đương với tiên đề V của Oclit

III. Hệ CSTT mở

❖ **Hệ cơ sở tri thức mở:** là những hệ cơ sở tri thức tiên tiến hơn, nó có khả năng bổ sung tri thức trong quá trình hoạt động, khám phá.

Ví dụ 1: Những hệ giải toán cho phép bổ sung tri thức trong quá trình suy luận (tri thức ban đầu là những tiên đề và một số định lý, tri thức bổ sung là những định lý mới, những tri thức heurictis, ...),

...



III. Hệ CSTT mở (tt)

Ví dụ 2: Hệ chẩn đoán hỏng hóc xe dựa trên tri thức luật dẫn

Tập các luật liên quan đến việc chẩn đoán hỏng hóc xe



IV. Hệ CSTT kết hợp

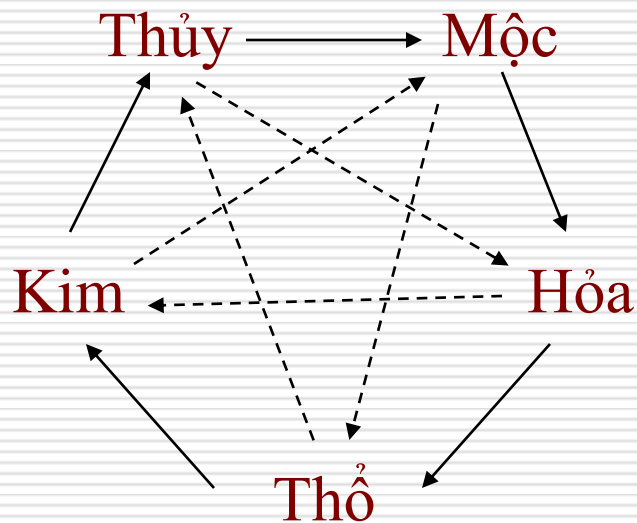
❖ **Hệ cơ sở tri thức kết hợp:** bao gồm sự kết hợp giữa hệ đóng và hệ mở, hệ kết hợp giữa CSTT và CSDL, hệ kết hợp giữa CSTT này với CSTT khác, ... Những hệ cơ sở tri thức kết hợp thường phát triển mạnh dựa trên tri thức liên ngành.

Ví dụ: kinh dịch, tử vi áp dụng với đời sống; kinh dịch, tử vi áp dụng với y học; ...); những hệ chẩn đoán, dự báo đòi hỏi tri thức liên ngành; ...



IV. Hệ CSTT kết hợp (tt)

Sơ đồ ngũ hành sinh khắc



————→ Sinh

-----> Khắc

❖ Tương sinh, tương khắc giống như âm dương, là 2 mặt không thể tách rời của sự vật

Tương sinh: Kim sinh Thủy

Thủy sinh Mộc

Mộc sinh Hỏa

Hỏa sinh Thổ

Thổ sinh Kim

Tương khắc: Kim khắc Mộc

Thủy khắc Hỏa

Mộc khắc Thổ

Hỏa khắc Kim

Thổ khắc Thủy

IV. Hệ CSTT kết hợp (tt)

Âm dương ngũ hành kết hợp với đời sống

Mọi sự vật và hiện tượng trong vũ trụ đều có thể gán với một “nhân ngũ hành”. Một ví dụ điển hình về việc xem tuổi hợp hay khắc

STT	Địa chi	Âm dương	Ngũ hành	Phương vị
1	Tý	Dương	Thủy	Bắc
2	Sửu	Âm	Thổ	4 phương
3	Dần	Dương	Mộc	Đông
4	Mão	Âm	Mộc	Đông
5	Thìn	Dương	Thổ	4 phương
6	Tỵ	Âm	Hỏa	Nam
7	Ngọ	Dương	Hỏa	Nam
8	Mùi	Âm	Thổ	4 phương
9	Thân	Dương	Kim	Tây

STT	Địa chi	Âm dương	Ngũ hành	Phương vị
10	Dậu	Âm	Kim	Tây
11	Tuất	Dương	Thổ	4 phương
12	Hợi	Âm	Thủy	Bắc

Mộc sinh Hỏa ⇒ Mão không khắc Tỵ

Thủy khắc Hỏa ⇒ Tý khắc Ngọ

...

V. Hệ thống mờ

1. Các khái niệm cơ bản

1.1 Tập rõ và hàm thành viên

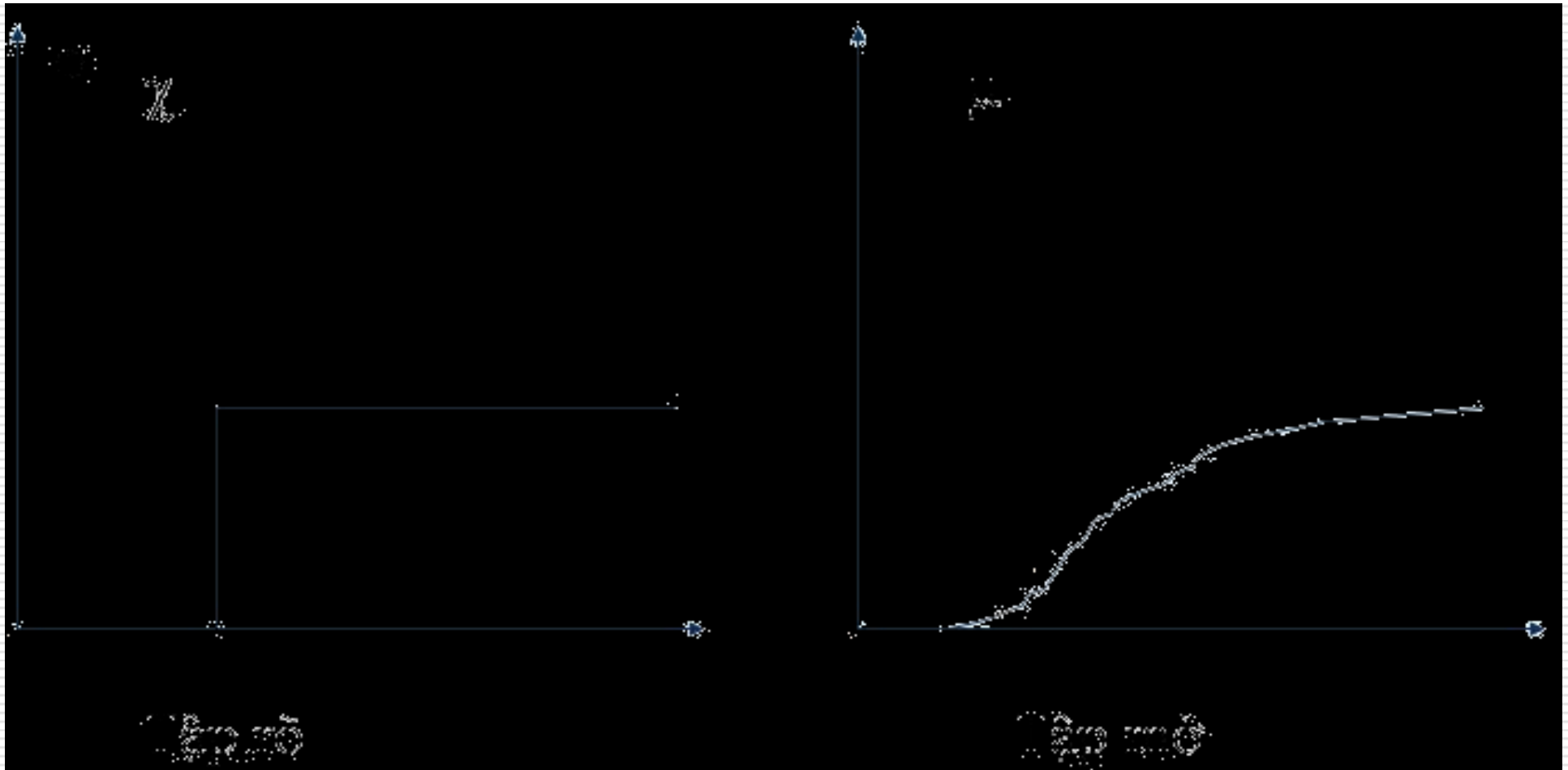
Tập rõ (crisp set) là tập hợp truyền thống theo quan điểm của Cantor (crisp set). Gọi A là một tập hợp rõ, một phần tử x có thể có $x \in A$ hoặc $x \notin A$, Có thể sử dụng hàm c để mô tả khái niệm thuộc về. Nếu $x \in A$, $c(x) = 1$, ngược lại nếu $x \notin A$, $c(x) = 0$. Hàm c được gọi là hàm đặc trưng của tập hợp A

1.2 Tập mờ và hàm thành viên

Khác với tập rõ, khái niệm thuộc về được mở rộng nhằm phản ánh mức độ x là phần tử của tập mờ A . Một tập mờ (fuzzy set): A được đặc trưng bằng hàm thành viên μ và cho x là một phần tử $\mu(x)$ phản ánh mức độ x thuộc về A .

Ví dụ: Cho tập mờ High Lan cao 1.5m, $\mu(\text{Lan})=0.3$ Hùng cao 2.0m, $\mu(\text{Hùng})=0.9$

V. Hệ thống mờ (tt)

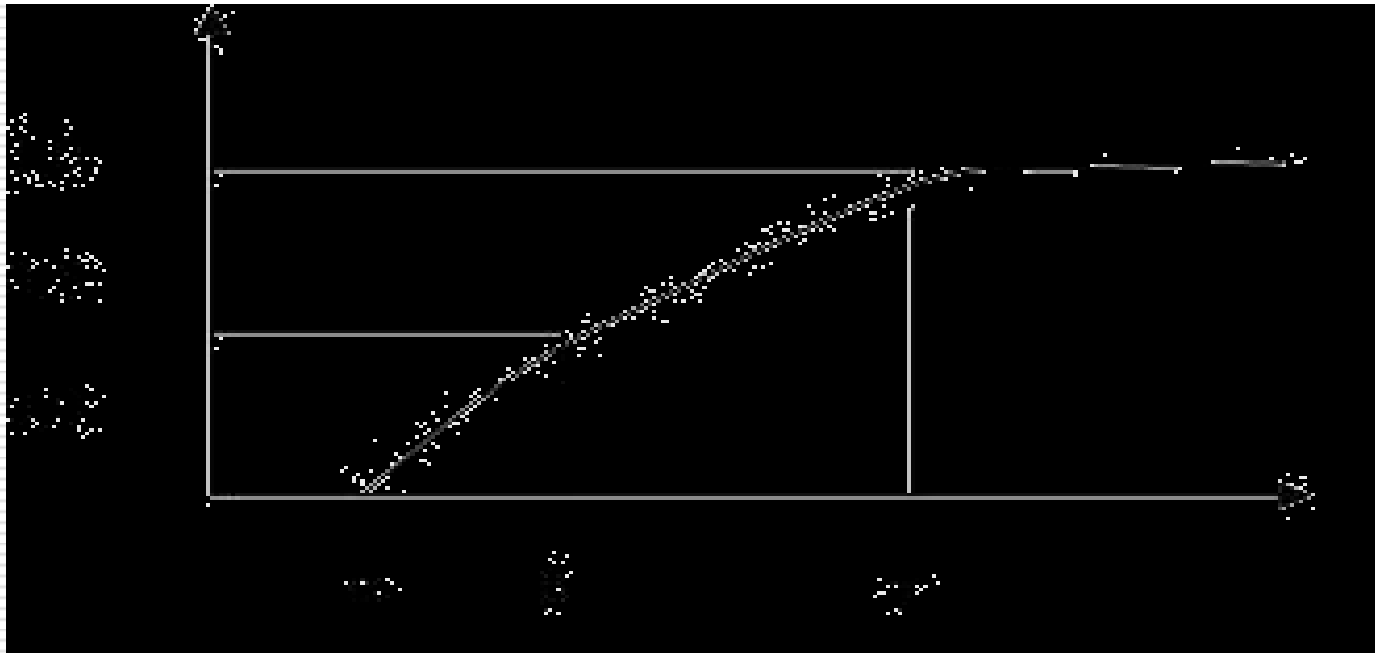


V. Hệ thống mờ (tt)

1.3 Các dạng của hàm thành viên

a) Dạng S tăng

$$m(x) = S(x, a, b, g) = \begin{cases} 0 & \text{nếu } x \leq \alpha \\ 2(x - \alpha) / (\gamma - \alpha) & \text{nếu } \alpha < x \leq \beta \\ 1 - [2(x - \alpha) / (\gamma - \alpha)] & \text{nếu } \beta < x < \gamma \\ 1 & \text{nếu } x \geq \gamma \end{cases}$$



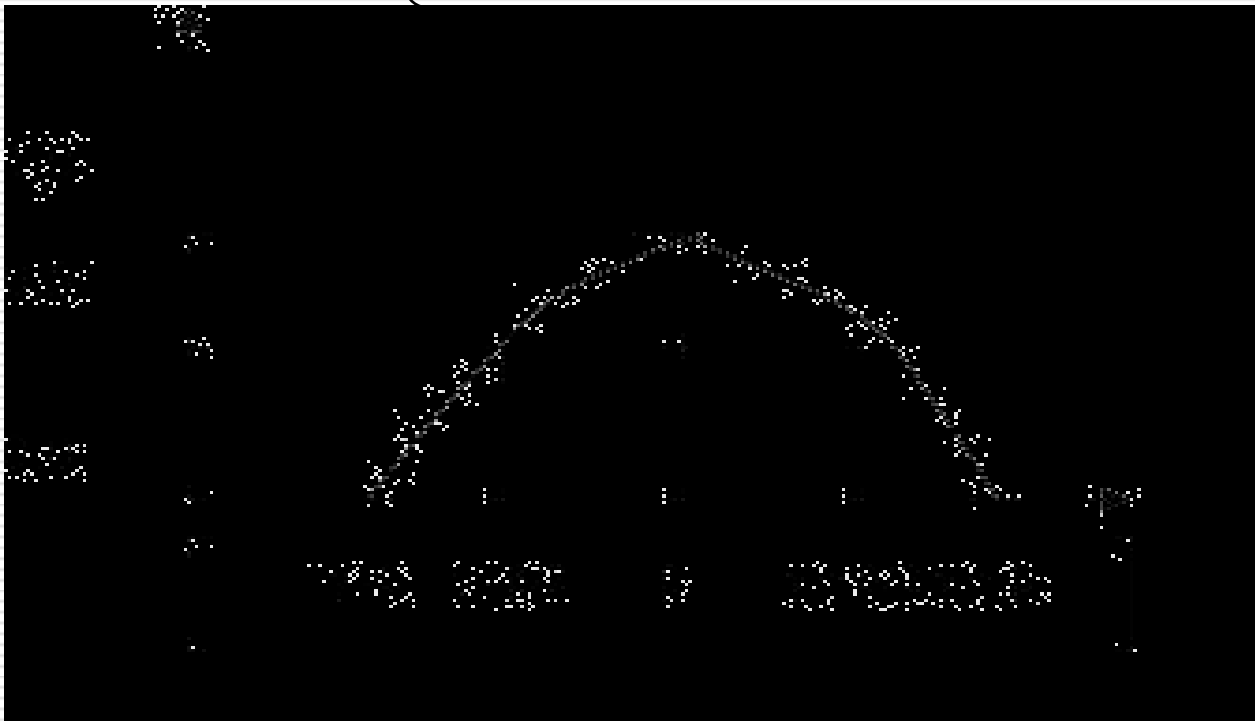
V. Hệ thống mờ (tt)

b) Dạng S giảm

$$m(x) = 1 - S(x, a, b, g)$$

c) Dạng hình chuông

$$P(x; g, b) = \begin{cases} S(x; g - b, g - b/2; g) & \text{if } x \leq g \\ S(x; g, g + b/2; g + b) & \text{if } x > g \end{cases}$$



V. Hệ thống mờ (tt)

1.4 Các phép toán trên tập mờ

Cho ba tập mờ A, B, C với $m A(x), m B(x), m C(x)$

$$C = A \underset{\zeta}{\cap} B : m C(x) = \min(m A(x), m B(x))$$

$$C = A \overset{\zeta}{\cup} B : m C(x) = \max(m A(x), m B(x))$$

$$C = \emptyset A : m C(x) = 1 - m A(x)$$

V. Hệ thống mờ (tt)

2. Các hệ thống mờ

2.1 Hàm thành viên cho các biến rời rạc

Cho tập vũ trụ $E = \text{Tốc độ} = \{ 20, 50, 80, 100 \}$ đơn vị là Km/g.

a. Xét tập mờ $F = \text{Nhanh}$ xác định bởi hàm membership

$$m_{\text{nhanh}}: E \rightarrow [0, 1]$$

$$x \mapsto m_{\text{nhanh}}(x)$$

Khi ta gán $m_{\text{nhanh}}(20) = 0$ nghĩa là tốc độ 20 Km/g được xem như là không nhanh



V. Hệ thống mờ (tt)

b. Xét tập mờ trung_bình với hàm thành viên xác định như sau



Tập Trung_bình = { 0.3,1,0.5,0 }

V. Hệ thống mờ (tt)

2.2 Hàm thành viên trong không gian các biến liên tục

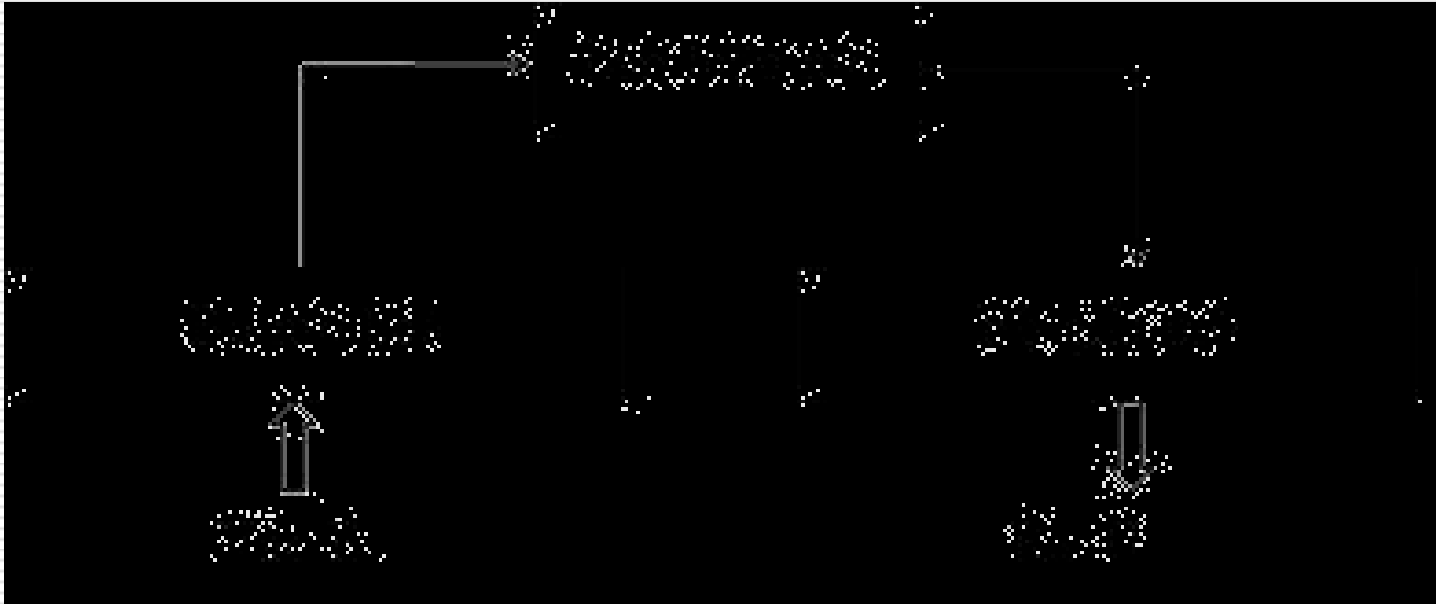
Chẳng hạn như các tập mờ Nhanh và Trung bình ở trên có thể định nghĩa như là các hàm

$$m_{\text{nhanh}}(x) = (x/100)^2$$

$$m_{\text{trung-bình}}(x) = \begin{cases} 0 & \text{if } x \leq 20 \\ (x-20)/30 & \text{if } 20 \leq x \leq 50 \\ (100-x)/50 & \text{if } 50 \leq x \leq 100 \end{cases}$$

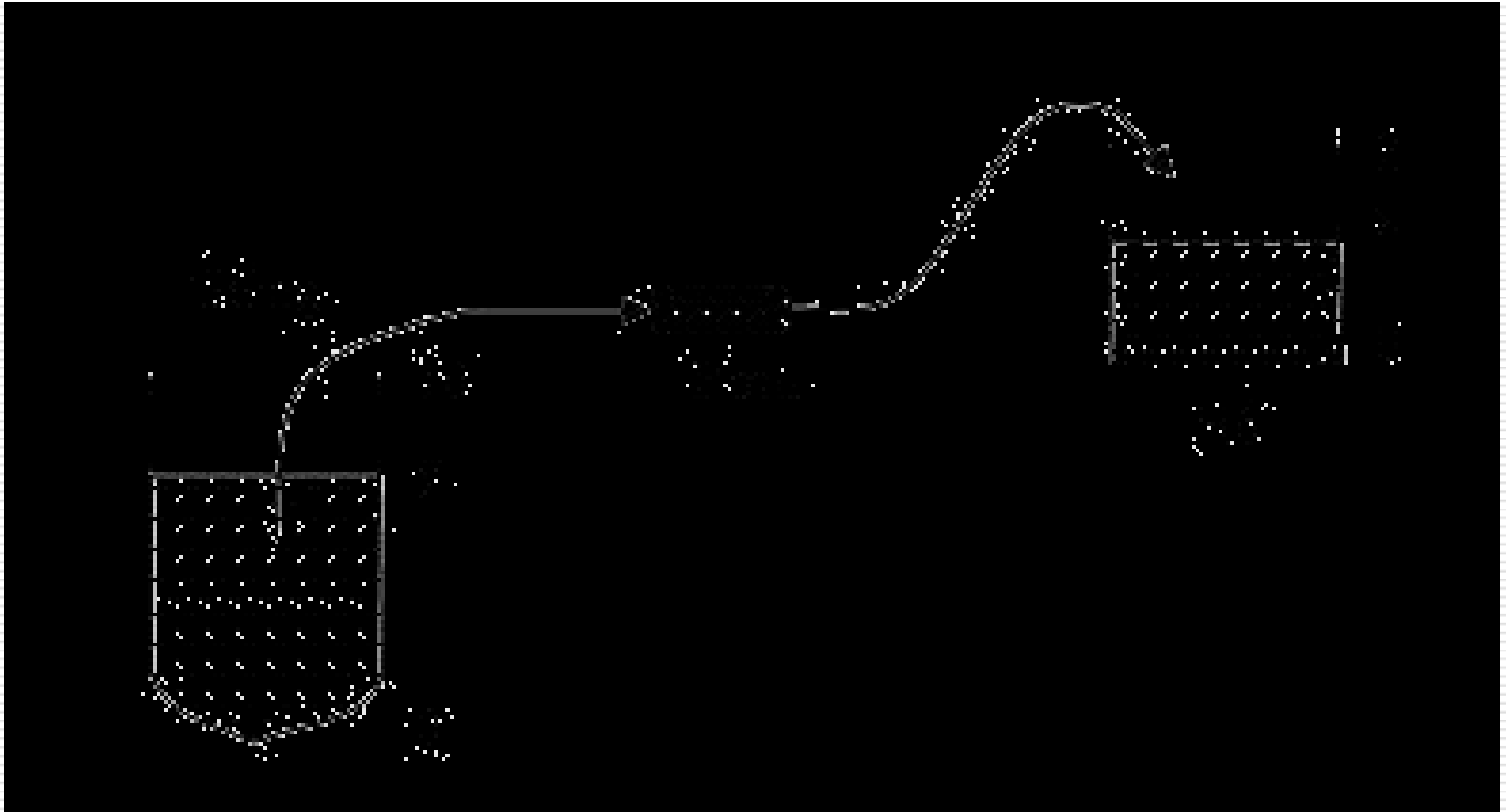
V. Hệ thống mờ (tt)

3. Xử lý bài toán mờ



V. Hệ thống mờ (tt)

Ví dụ: Giải bài toán điều khiển tự động mờ cho hệ thống bơm nước lấy nước từ giếng. Trong khi hồ hết nước và trong giếng có nước thì máy bơm tự động bơm



V. Hệ thống mờ (tt)

- ❖ Với biến ngôn ngữ Hồ có các tập mờ hồ đầy (H.Đầy), hồ lưng (H.Lưng) và hồ cạn (H.Cạn).
- ❖ Với biến ngôn ngữ Giếng có các tập mờ nước cao (G.Cao), nước vừa (G.Vừa), nước ít (G.Ít).
- ❖ Với biến ngôn ngữ kết luận xác định thời gian bơm sẽ có các tập mờ bơm vừa (B.Vừa), bơm lâu (B.Lâu), bơm hơi lâu (B.HơiLâu).

	H.Đầy	H.Lưng	H.Cạn
G.Cao	0	B.Vừa	B.Lâu
G.Vừa	0	B.Vừa	B.HơiLâu
G.Ít	0	0	0

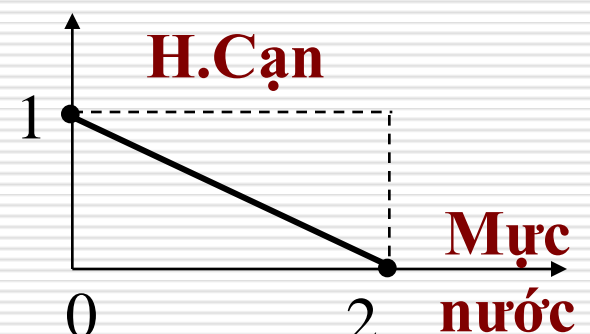
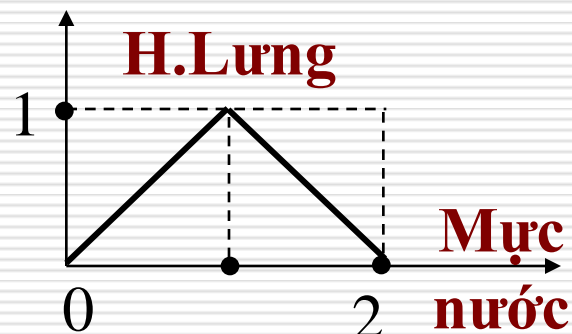
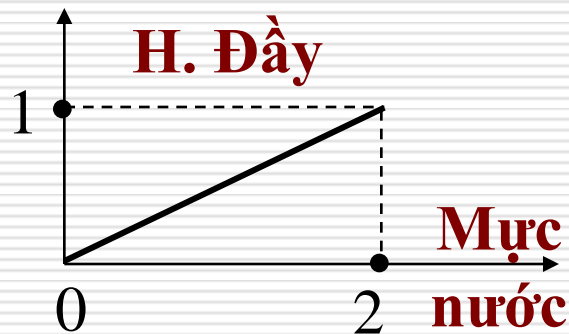
V. Hệ thống mờ (tt)

Hàm thành viên của Hồ nước:

❖ H.Đầy(x) = $x/2$ nếu $0 \leq x \leq 2$

❖ H.Lưng(x) = { x nếu $0 \leq x \leq 1$
 $(2-x)$ nếu $1 \leq x \leq 2$ }

❖ H.Cạn(x) = $(1-x/2)$ nếu $0 \leq x \leq 2$



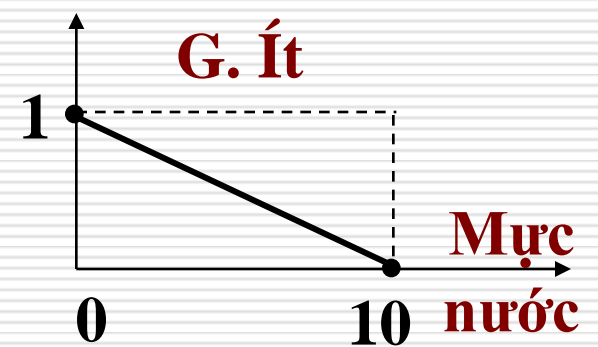
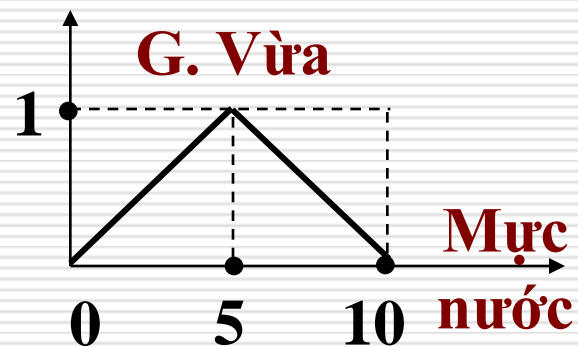
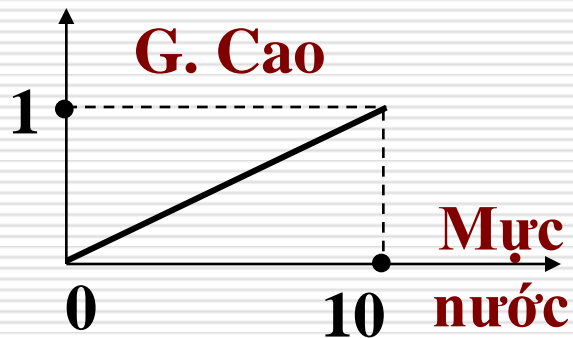
V. Hệ thống mờ (tt)

Hàm thành viên cho giếng:

❖ $G.Cao(y) = y/10$ nếu $0 \leq y \leq 10$

❖ $G.Vừa(y) = \begin{cases} y/5 & \text{nếu } 0 \leq y \leq 5 \\ (10-y)/5 & \text{nếu } 5 \leq y \leq 10 \end{cases}$

❖ $G.Ít(y) = (1-y/10)$ nếu $0 \leq y \leq 10$



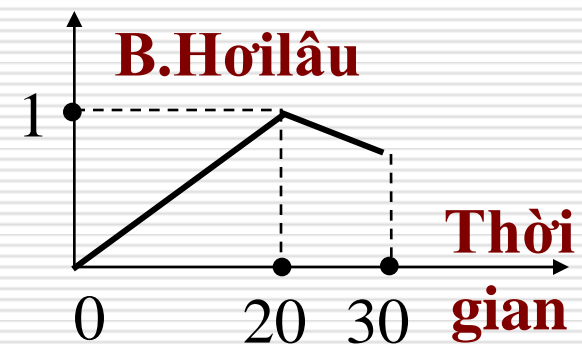
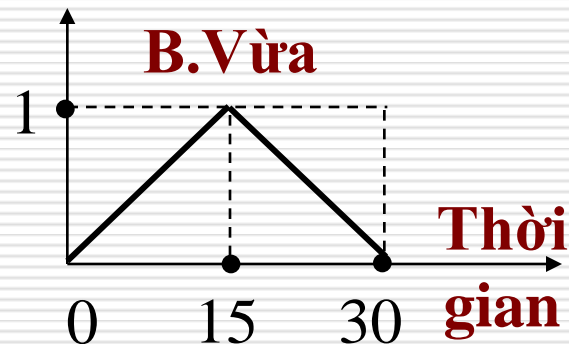
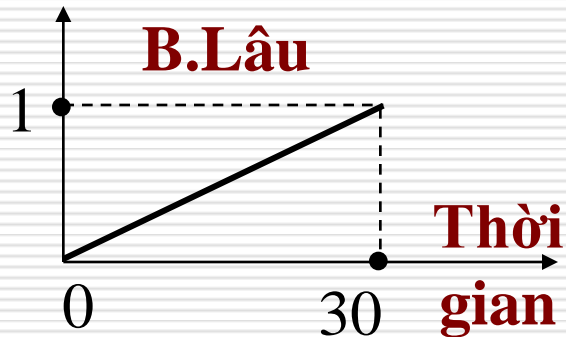
V. Hệ thống mờ (tt)

Hàm thành viên của Kết luận cho từng luật:

$$\text{B.Vừa}(z) = \left\{ \begin{array}{ll} z/15 & \text{nếu } 0 \leq z \leq 15 \\ (30-z)/15 & \text{nếu } 15 \leq z \leq 30 \end{array} \right\}$$

$$\text{B.lâu}(z) = z/30 \quad \text{nếu } 0 \leq z \leq 30$$

$$\text{B.Hơi lâu}(z) = \left\{ \begin{array}{ll} z/20 & \text{nếu } 0 \leq z \leq 20 \\ 1-0.05(z-20) & \text{nếu } 20 \leq z \leq 30 \end{array} \right\}$$



V. Hệ thống mờ (tt)

Chỉ xét 4 luật trong trường hợp máy bơm hoạt động (tất cả 9 luật)

- Luật 1: **if** x is H.Lung **and** y is N.Cao **Then** z is B.Vừa
- Luật 2: **if** x is H.Cạn **and** y is N.Cao **Then** z is B.Lâu
- Luật 3: **if** x is H.Lung **and** y is N.Vừa **Then** z is B.Vừa
- Luật 4: **if** x is H.Cạn **and** y is N.Vừa **Then** z is B.Hơi lâu

Bây giờ nếu ta nhập trị Input:

$x_0 = 1$ (Độ cao của nước trong hồ)

$y_0 = 3$ (Độ cao của nước trong giếng).

Hỏi bơm bao lâu?

V. Hệ thống mờ (tt)

Luật 1: $\mu_{H.Lưng}(x_0) = 1$

$\mu_{N.Cao}(y_0) = 3/10$

$W_1 (B.Vừa) = \min(1; 3/10) = 3/10$

Luật 2: $\mu_{H.Cạn}(x_0) = 0.5$

$\mu_{G.Cao}(y_0) = 3/10$

$W_2 (B.Lâu) = \min(0.5; 3/10) = 3/10$

Luật 3: $\mu_{H.Lưng}(x_0) = 1$

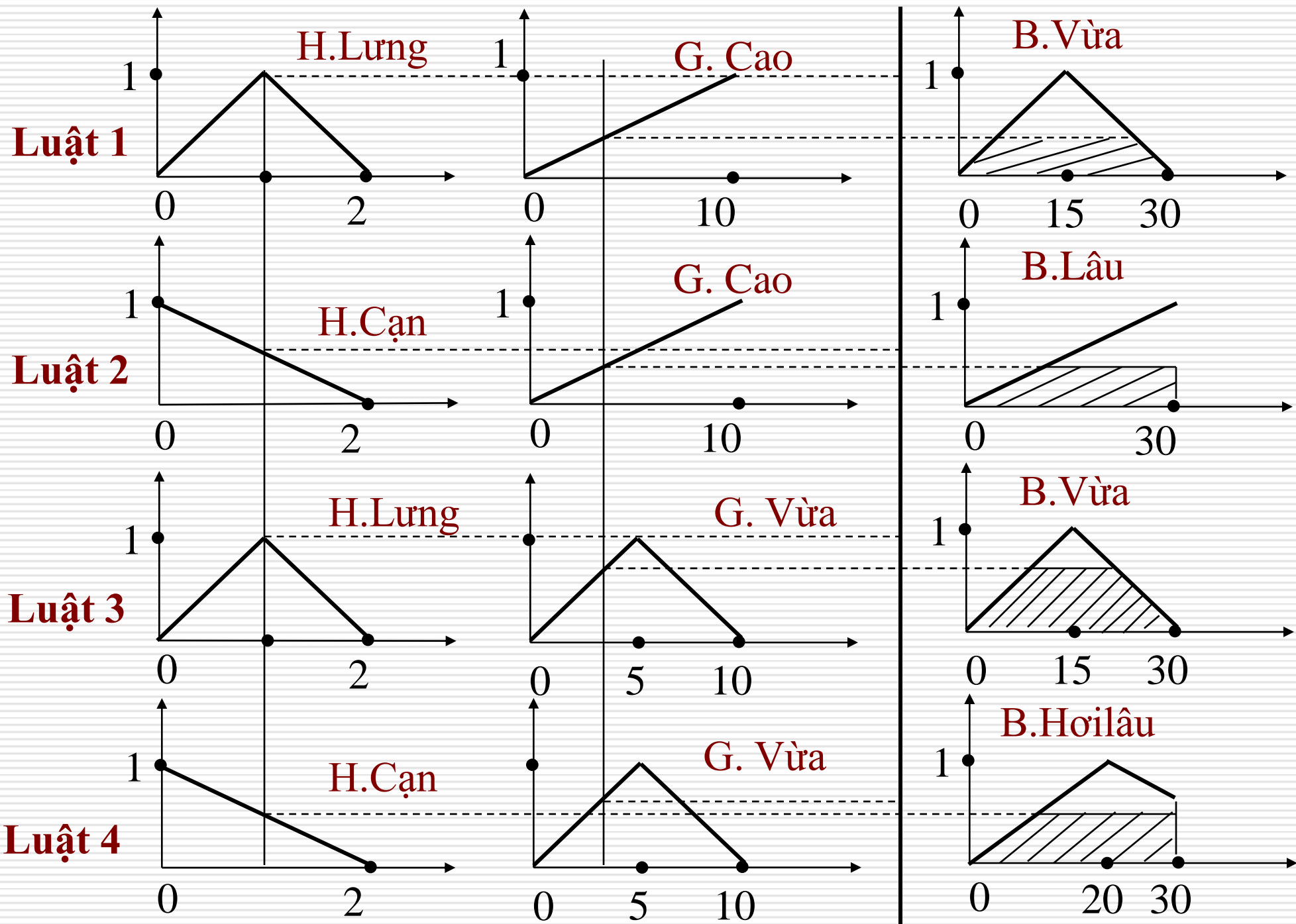
$\mu_{G.Vừa}(y_0) = 3/5$

$W_3 (B.Vừa) = \min(1; 3/5) = 3/5$

Luật 4: $\mu_{H.Cạn}(x_0) = 0.5$

$\mu_{G.Vừa}(y_0) = 3/5$

$W_4 (B.Hoailâu) = \min(0.5; 3/5) = 0.5$



V. Hệ thống mờ (tt)

❖ Các W_i gọi là các trọng số của luật thứ i

$$\mu_C(z) = \sum W_i m_{KLi}(Z) \quad i = 1 \dots N$$

$$m_C(z) = W_1 \cdot B.Vừa(z) + W_2 \cdot B.Lâu(z) + W_3 \cdot B.Vừa(z) + \\ + W_4 \cdot B.Hơi Lâu(z)$$

$$\mu_C(z) = 3/10 \cdot B.Vừa(z) + 3/10 \cdot B.Lâu(z) + 3/5 \cdot B.Vừa(z) + \\ + 0.5 \cdot B.HơiLâu(z)$$

❖ Bước tiếp theo là ta phải giải mờ từ hàm thành viên của kết luận bằng cách tính trọng tâm của hàm $m_C(z)$

V. Hệ thống mờ (tt)

Moment $m_c(z)$ là



và



Vậy $\text{Defuzzy}(z) = 17.12 / 2.3 = 8.15$

❖ Do đó nếu mực nước trong hồ và giếng là 1m và 3m thì thời gian cần bơm là 8 phút và 15 giây.

Chương 5:

Một số hệ CSTT điển hình

Tham khảo thêm:

- [1] GS.TSKH Hoàng Kiếm, TS. Đỗ Văn Nhơn, Th.sĩ Đỗ Phúc. Giáo trình Các hệ cơ sở tri thức. Đại Học Quốc Gia TP HCM – 2002.
- [2] GS.TSKH Hoàng Kiếm, Th.sĩ Đinh Nguyễn Anh Dũng. Giáo trình Trí tuệ nhân tạo. Đại Học Quốc Gia TP HCM – 2002.
- [3] Adrian A.Hopgood. Knowledge-based systems for Engineers and Scientists. The Open University – CRC Press. Boca-Raton Ann-Arbor London Tokyo 1998.

I. Hệ giải toán

Giới thiệu: Mục đích của nhóm tác giả là xây dựng một chương trình thông minh giúp sinh viên và học sinh học toán.

Chức năng chính:

Khả năng giải toán tự động.

Tìm kiếm tri thức: khả năng giúp người dùng tìm kiếm những tri thức cần thiết một cách nhanh chóng.

Phương pháp biểu diễn tri thức: Tiếp cận hướng đối tượng (Object - Oriented) để biểu diễn tri thức.

Tham khảo thêm trong tài liệu: Hoàng Kiếm, Đỗ Văn Nhơn. A Program for Studying and Solving Problems in Plane Geometry, 2002.



II. Hệ chẩn đoán y khoa MYCIN

Giới thiệu: MYCIN là một hệ lập luận trong y học được hoàn tất vào năm 1970 tại đại học Stanford, Hoa Kỳ. Đây là một hệ chuyên gia dựa trên luật và sự kiện.

MYCIN sử dụng cơ chế lập luận gần đúng để xử lý các luật suy diễn dựa trên độ đo chắc chắn. Tiếp sau hệ MYCIN là hệ EMYCIN ra đời.

EMYCIN cung cấp cơ chế lập luận và tùy vào bài toán cụ thể sẽ bổ sung tri thức riêng cho bài toán đó để trở thành hệ chuyên gia.

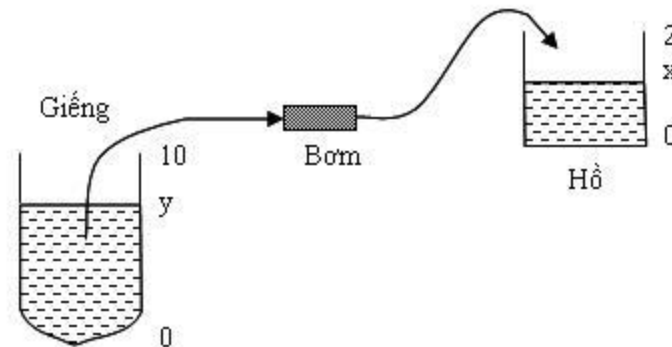
Tham khảo thêm tài liệu: GS.TSKH Hoàng Kiếm, TS. Đỗ Văn Nhơn, Th.sĩ Đỗ Phúc. Giáo trình Các hệ cơ sở tri thức. Đại Học Quốc Gia TPHCM – 2002.



III. Hệ điều khiển tự động

Giới thiệu: là những hệ điều khiển gắn với CSTT. Những hệ thống này thường ứng dụng trong công nghiệp, trong điều khiển tự động hóa, thường là những hệ thống thời gian thực. Một trong số các hệ thống này có kết hợp với lý thuyết mờ.

Ví dụ: máy giặt, máy bơm nước, ... (GS.TSKH Hoàng Kiếm, TS. Đỗ Văn Nhơn, Th.sĩ Đỗ Phúc. Giáo trình Các hệ cơ sở tri thức. Đại Học Quốc Gia TPHCM – 2002.)



III. Hệ điều khiển tự động (tt)

Ví dụ: LINKman là hệ thống điều khiển mờ ứng dụng điều khiển lò luyện kim, và những tiến trình sản xuất khác. (Taunton, J. C. and Haspel, D. W., “The application of expert system techniques in on-line process control”, in Expert systems in engineering, Pham, D. T. (ed.), IFS Publications/Springer-Verlag (1988)).

Ví dụ: RESCU là một hệ CSTT thời gian thực phục vụ cho điều khiển (Leitch, R., Kraft, R. and Luntz, R., “RESCU: a real-time knowledge based system for process control”, IEE Proceedings-D, 138, p127 (1991)).

Tham khảo thêm tài liệu: Adrian A.Hopgood. Knowledge-based systems for Engineers and Scientists. The Open University – CRC Press. Boca-Raton Ann-Arbor London Tokyo 1998.

IV. Hệ dự báo thời tiết

Giới thiệu: Báo cáo thu hoạch môn học Các hệ Cơ sở tri thức dùng suy diễn dựa trên tình huống (CBR – Case Based Reasoning) kết hợp với lý thuyết mờ (Fuzzy logic) để xây dựng nên một hệ thống dự báo thời tiết



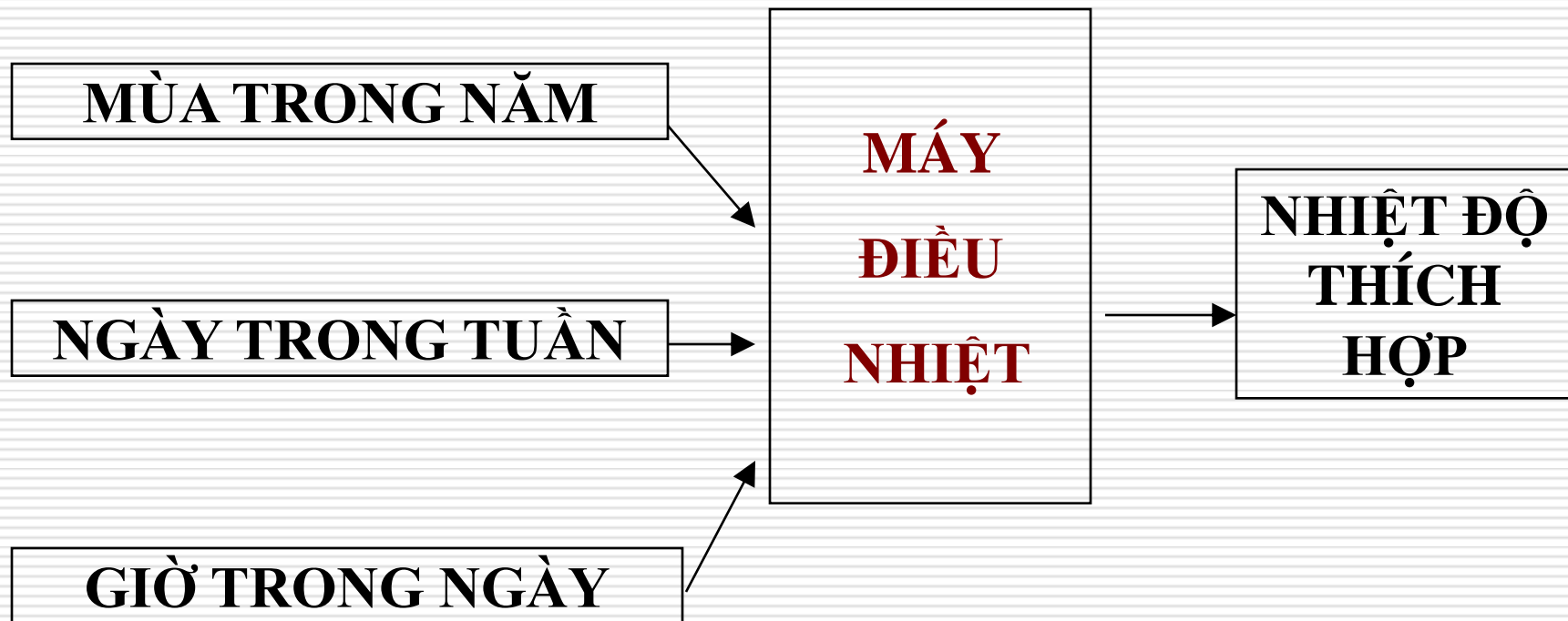
Tham khảo thêm tài liệu: Dự báo thời tiết dùng suy diễn dựa trên tình huống và lý thuyết tập mờ

Hướng dẫn: GSTSKH Hoàng Kiếm

Sinh viên thực hiện: Lê trọng Ngọc – Bùi Thùy Trang

Phần II: Các hệ Cơ sở tri thức (knowledge-based systems)

MÁY ĐIỀU NHIỆT



CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 1:

Nếu Ngày là Thứ 2

Hoặc Ngày là Thứ 3

Hoặc Ngày là Thứ 4

Hoặc Ngày là Thứ 5

Hoặc Ngày là Thứ 6

Thì Hôm nay là Ngày làm việc

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 2:

Nếu Ngày là Thứ 7

Hoặc Ngày là Chủ nhật

Thì Hôm nay là Cuối tuần

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 3:

Nếu Hôm nay là Ngày làm việc

Và Thời gian giữa 9 giờ sáng và 5 giờ chiều

Thì Sự khởi động là ‘trong thời gian làm việc’

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 4:

Nếu Hôm nay là Ngày làm việc

Và Thời gian trước 9 giờ sáng

Thì Sự khởi động là ‘ngoài thời gian làm việc’

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 5:

Nếu Hôm nay là Ngày làm việc

Và Thời gian là sau 5 giờ chiều

Thì Sự khởi động là ‘ngoài thời gian làm việc’

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 6:

Nếu Hôm nay là **Cuối tuần**

Thì **Sự khởi động là ‘ngoài thời gian làm việc’**

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 7:

Nếu Tháng là Tháng 1

Hoặc Tháng là Tháng 2

Hoặc Tháng là Tháng 12

Thì Mùa là Mùa Hè

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 8:

Nếu Tháng là Tháng 3

Hoặc Tháng là Tháng 4

Hoặc Tháng là Tháng 5

Thì Mùa là Mùa Thu

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 9:

Nếu Tháng là Tháng 6

Hoặc Tháng là Tháng 7

Hoặc Tháng là Tháng 8

Thì Mùa là Mùa Đông

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 10:

Nếu Tháng là Tháng 9

Hoặc Tháng là Tháng 10

Hoặc Tháng là Tháng 11

Thì Mùa là Mùa Xuân

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 11:

Nếu Mùa là Mùa Xuân

Và Sự khởi động là ‘trong thời gian làm việc’

Thì Cấu hình điều nhiệt là ‘20 độ’

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 12:

Nếu Mùa là Mùa Xuân

Và Sự khởi động là ‘ngoài thời gian làm việc’

Thì Cấu hình điều nhiệt là ‘15 độ’

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 13:

Nếu Mùa là Mùa Hè

Và Sự khởi động là ‘trong thời gian làm việc’

Thì Cấu hình điều nhiệt là ‘24 độ’

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 14:

Nếu Mùa là Mùa Hè

Và Sự khởi động là ‘ngoài thời gian làm việc’

Thì Cấu hình điều nhiệt là ‘27 độ’

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 15:

Nếu Mùa là Mùa Thu

Và Sự khởi động là ‘trong thời gian làm việc’

Thì Cấu hình điều nhiệt là ‘20 độ’

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 16:

Nếu Mùa là Mùa Thu

Và Sự khởi động là ‘ngoài thời gian làm việc’

Thì Cấu hình điều nhiệt là ‘16 độ’

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 17:

Nếu Mùa là Mùa Đông

Và Sự khởi động là ‘trong thời gian làm việc’

Thì Cấu hình điều nhiệt là ‘18 độ’

CƠ SỞ TRI THỨC LUẬT DẪN (18 LUẬT)

Luật dẫn 18:

Nếu Mùa là Mùa Đông

Và Sự khởi động là ‘ngoài thời gian làm việc’

Thì Cấu hình điều nhiệt là ‘14 độ’

CÁC ĐỐI TƯỢNG

Đối tượng	Giá trị
Tháng	Tháng 1, ..., Tháng 12
Mùa	Xuân, hè, thu, đông
Ngày	Thứ 2, ..., Thứ 7, Chủ nhật
Thời gian	Trước 9, Sau 17, Từ 9 -> 17
Sự khởi động	Trong giờ làm việc, ngoài giờ làm việc

GIẢI PHÁP ĐIỀU NHIỆT

7 giải pháp điều nhiệt

- Đặt máy điều nhiệt là “14° C”

- Đặt máy điều nhiệt là “15° C”

- Đặt máy điều nhiệt là “16° C”

- Đặt máy điều nhiệt là “18° C”

- Đặt máy điều nhiệt là “20° C”

- Đặt máy điều nhiệt là “24° C”

- Đặt máy điều nhiệt là “27° C”

MINH HỌA

Tháng ?

8

Thứ ?

Thứ 6

Giờ ?

$9 < \text{Giờ} < 17$

Luật 9

Mùa =

Mùa Đông

Luật 1

Hôm nay =

ngày làm việc

Luật 3

Sự hoạt động =

trong giờ làm việc

Máy điều nhiệt = 18° C

Chương 6

Máy học và khám phá tri thức

Tham khảo thêm:

[1] GS.TSKH Hoàng Kiếm. Bài giảng cao học môn học cơ sở tri thức và ứng dụng. ĐHKHTN-TPHCM.

[2] Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski. Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, 1998

I. Khái niệm máy học

Thế nào là máy học (Learning Machine) ?

❖ Máy tính hay chương trình máy tính có khả năng tự hoàn thiện từ “kinh nghiệm”.

❖ Máy học còn có nghĩa là việc mô hình hóa môi trường xung quanh hay khả năng một chương trình máy tính sinh ra một cấu trúc dữ liệu mới khác với cấu trúc hiện có. Chẳng hạn việc tìm ra những luật If...then... từ tập dữ liệu đầu vào.

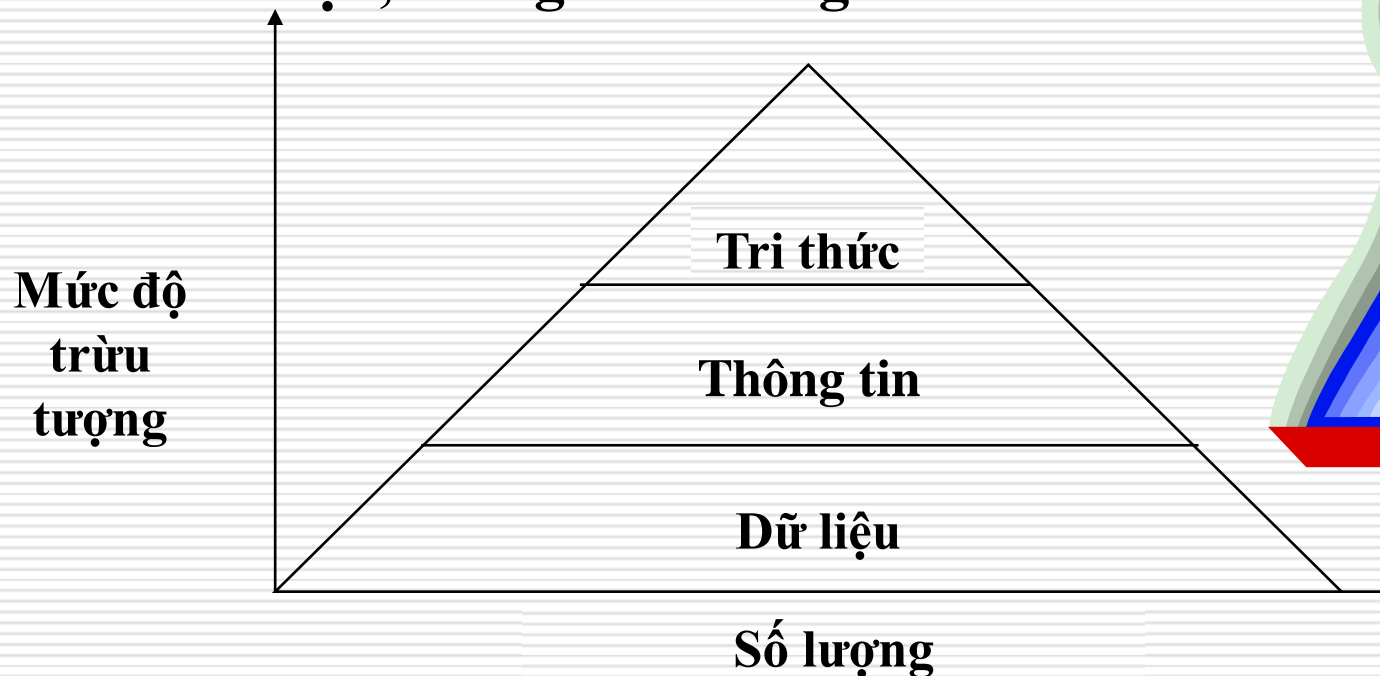
(Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski. Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, 1998)

II. Khám phá tri thức

Thế nào là khám phá tri thức (knowledge discovery) ?

❖ Khám phá tri thức là tìm ra những tri thức tiềm ẩn, những tri thức mới (không phải là những tri thức kinh điển, kinh nghiệm, ...)

Thừa dữ liệu, thông tin nhưng thiếu tri thức.



III. Phân loại máy học

Phân loại thô:

- ❖ Học giám sát (supervised learning)
- ❖ Học không giám sát (unsupervised learning)

Phân loại theo 2 tiêu chuẩn cùng lúc: “cấp độ học” & “cách tiếp cận”

Cấp độ học:

- ❖ Học vẹt (Rote learning)
- ❖ Học theo giải thích (by explanation)
- ❖ Học theo ví dụ, trường hợp (by examples, cases)
- ❖ Học khám phá (by discovering)

III. Phân loại máy học (tt)

Cách tiếp cận:

❖ Tiếp cận thống kê

❖ Tiếp cận toán tử logic

❖ Tiếp cận hình học

(phân hoạch không gian, xây dựng cây định danh, ...)

❖ Tiếp cận mạng Neural

❖ Tiếp cận khai mở dữ liệu

❖ ...

III.1 Tiếp cận thống kê

Ví dụ: Chương trình đoán ý nghĩ con người. Máy sẽ đoán người chơi nghĩ số 0 hay 1 trong đầu, người chơi sẽ phải trả lời cho máy biết là máy đã đoán đúng hay sai. Để từ đó máy tính sẽ học qui luật suy nghĩ của người chơi.



III.1 Tiếp cận thống kê (tt)

Ý tưởng cài đặt: hết sức đơn giản

- Lưu trữ toàn bộ dãy số 0, 1 mà người chơi đã nghĩ ra.
- Lấy 7 con số trước đó (người chơi đưa ra), tính xác suất xuất hiện của số 1 và số 0 sau dãy 7 con số này. Máy sẽ đoán số có xác suất xuất hiện cao hơn.

Giả sử ở lần đoán thứ i , dãy số mà người dùng đã đoán như sau:

... 1 1 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 ?

Từ dữ liệu lưu trữ ở những lần đoán trước, giả sử số lần xuất hiện của 1 sau dãy 0 0 0 0 1 0 0 là 28 và số lần xuất hiện của số 0 là 90

Xác suất xuất hiện của số 1 sau dãy này là: $28/(28+90) = 23.7\%$

Xác suất xuất hiện của số 0 sau dãy này là: $90/(28+90) = 76.3\%$

⇒ Máy sẽ đoán số 0

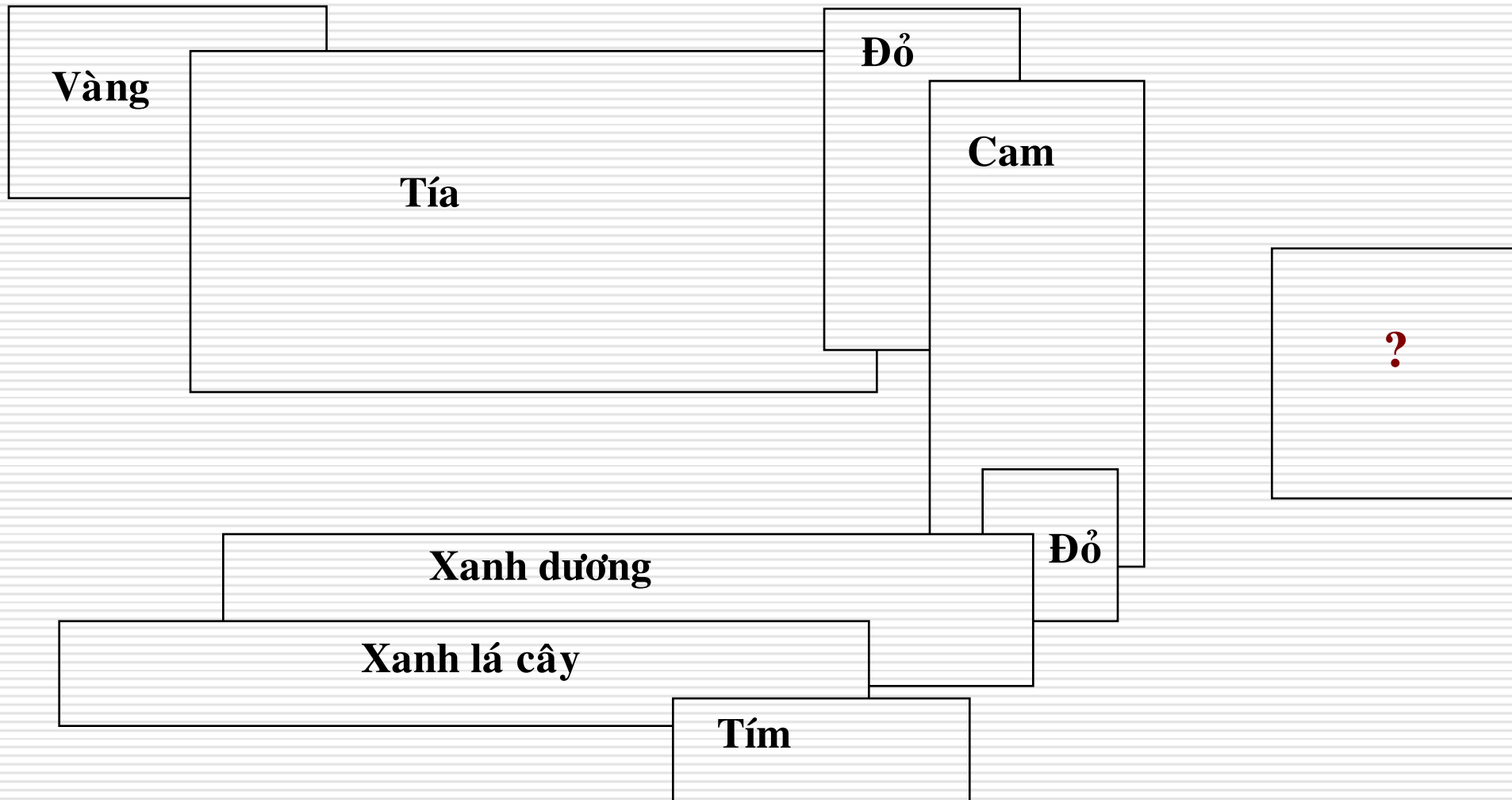
III.1 Tiếp cận thống kê (tt)

Nhận xét ví dụ:

- ❖ Ví dụ đã đưa ra là thuộc cấp độ học vẹt sử dụng cách tiếp cận thống kê.
- ❖ Máy không thể đoán đúng ngay được, nhưng càng về sau (vài trăm lần đoán) máy càng trở nên chính xác một cách kinh ngạc (trung bình có thể lên đến 90%).
- ❖ Trên thực tế khi cài đặt chương trình này tác giả không chỉ đoán qui luật từ dãy số của người chơi, máy còn sử dụng cả dãy số mà máy đã đoán

III.2 Tiếp cận hình học

Xét bài toán: cho tập các hình chữ nhật với kích thước (ngang & rộng) và màu sắc khác nhau (hình vẽ). Cho biết hình bên phải có màu gì?



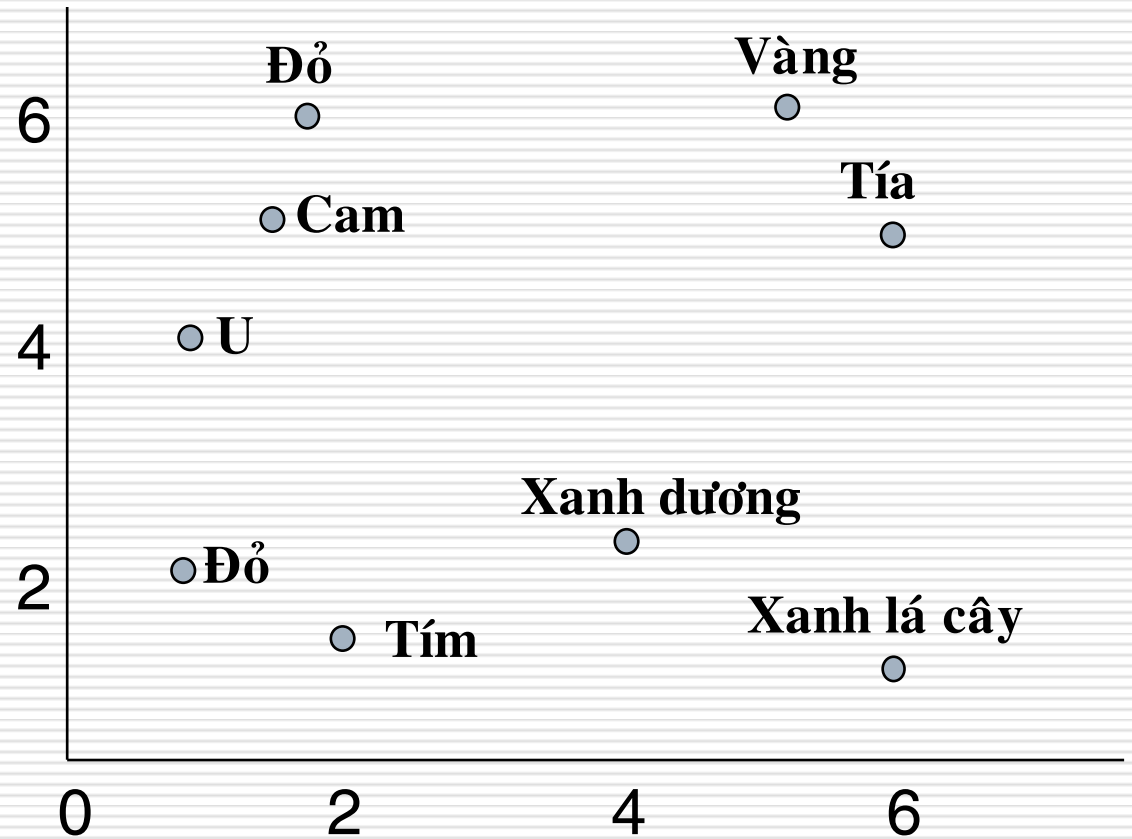
III.2 Tiếp cận hình học (tt)

Giải quyết bài toán:

❖ Phản ứng tự nhiên của con người: tìm khối có sẵn gần giống để đoán màu cho khối chưa biết. Như thế nào là gần giống ?

❖ Biểu diễn 2 thuộc tính chiều rộng & chiều cao dưới dạng 1 điểm trên mặt phẳng 2 chiều.

❖ Tính khoảng cách từ khối cần tìm đến tất cả các khối còn lại. (bài toán người láng giềng gần nhất với độ phức tạp $O(n)$).

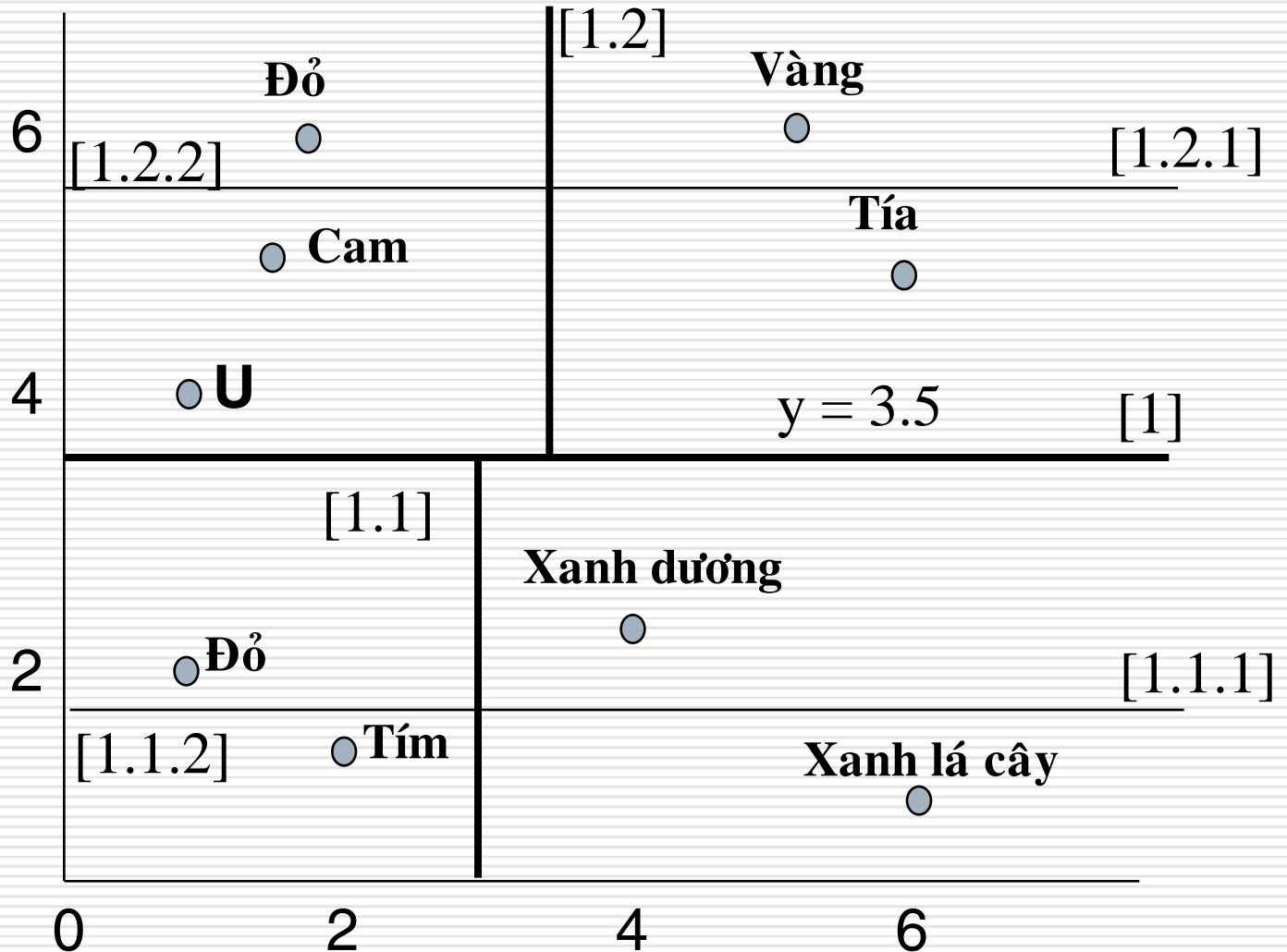


III.2 Tiếp cận hình học (tt)

❖ Cách làm hiệu quả hơn là tìm cách chia không gian các hình chữ nhật mẫu thành các khu vực riêng biệt theo kiểu phân cấp không gian.

❖ 8 không gian riêng biệt ứng với 8 hình chữ nhật đã cho ban đầu.

❖ Lần lượt xác định vị trí tương đối của U so với các đường chia. Cuối cùng U xếp cùng không gian với hình chữ nhật có màu cam \Rightarrow U có màu cam

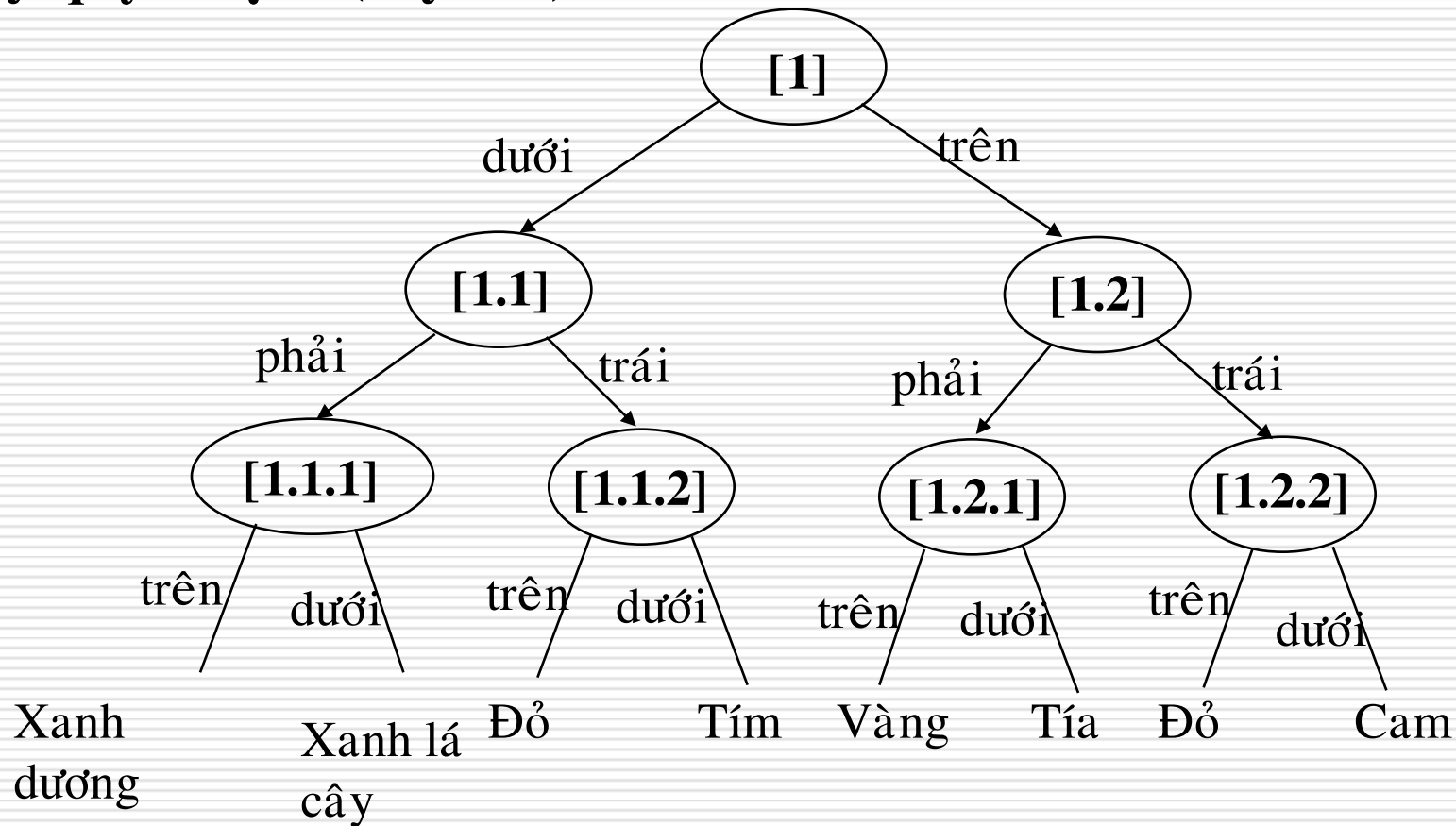


III.2 Tiếp cận hình học (tt)

Nhận xét bài toán:

❖ Về mặt thuật toán, phân chia không gian theo cách làm trên là phân chia theo cây k-d.

❖ Cây quyết định (cây k-2) của bài toán có thể biểu diễn như sau:



III.3 Tiếp cận logic (tt)

Nhận xét ví dụ 1:

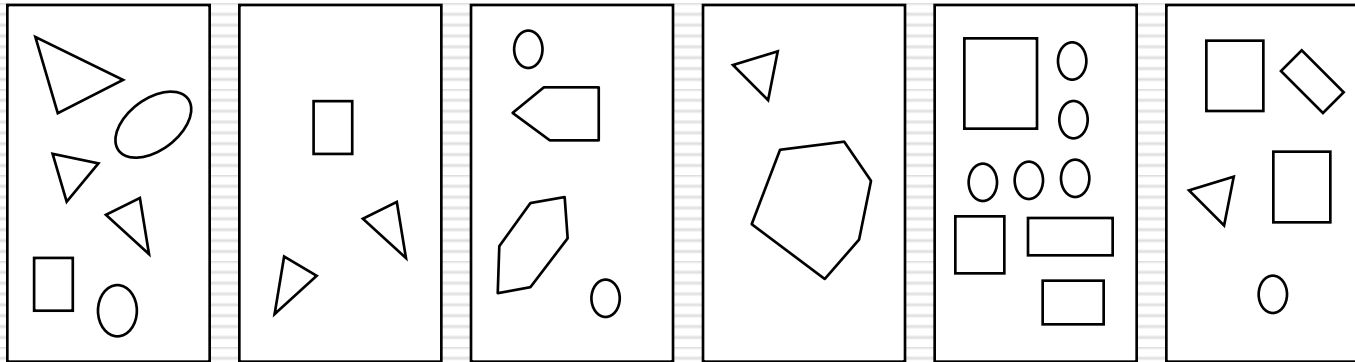
❖ Nếu tinh ý bạn sẽ nhận thấy các điểm trắng trong nhóm A luôn thẳng hàng.

❖ Thật khó để phát hiện ra đặc tính vừa nêu trên (ngay cả đối với con người) nhất là đối với các đối tượng hình học.

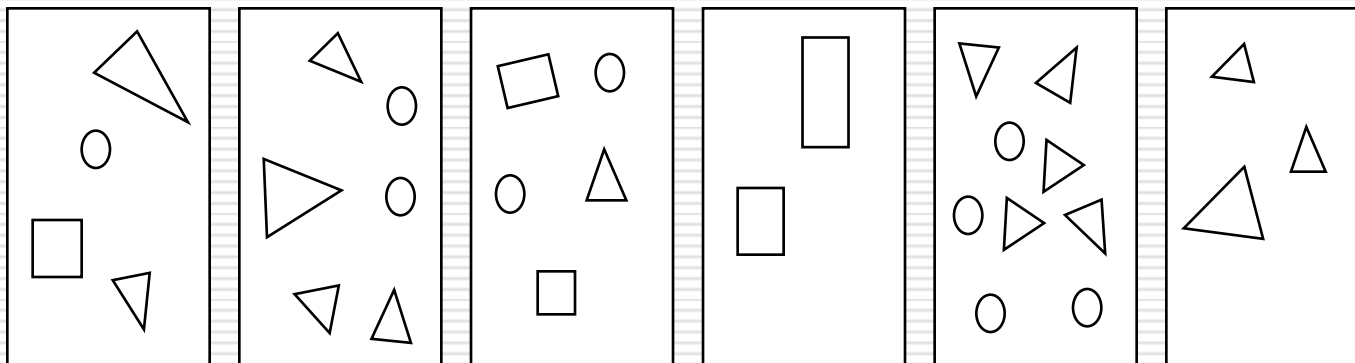
❖ Nhà bác học Bongard đã đề ra một phương án xác định mối liên hệ bằng cách xây dựng các mệnh đề logic. (xem ví dụ 2)

III.3 Tiếp cận logic (tt)

Ví dụ 2: Xác định đặc điểm của các nhóm hình A, B



Nhóm A



Nhóm B

III.3 Tiếp cận logic (tt)

Nhận xét ví dụ 2:

❖ Nhóm A : Tổng số đỉnh trừ tổng số đối tượng = 7. (Chẳng hạn như hình 2 trong nhóm A có 3 hình gồm 2 tam giác và một hình chữ nhật, tổng cộng có 10 đỉnh).

❖ Nhóm B : Tổng số đỉnh trừ tổng số đối tượng = 6.

❖ Hình ellipse và hình tròn được xem là không có đỉnh nào

❖ Không được gợi ý thì quan hệ trên là một loại quan hệ rất khó được phát hiện.

❖ Với phương án của Bongard, ta vẫn có thể tìm ra được mối liên hệ đủ để phân biệt hai nhóm hình này.

III.3 Tiếp cận logic (tt)

Định ra một số các mệnh đề logic đơn giản như:

❖ P_1 : “tồn tại tam giác”

❖ P_2 : “tồn tại vòng tròn”

❖ P_3 : “tồn tại hình oval”

❖ P_4 : “tồn tại hình chữ nhật”

❖ P_5 : “tồn tại hình đa giác nhiều hơn 4 cạnh”.

III.3 Tiếp cận logic (tt)

Hình	Tam giác P_1	Vòng tròn P_2	Oval P_3	Chữ nhật P_4	Đa giác P_5	Nhóm
1	1	1	1	1	0	A
2	1	0	0	1	0	A
3	0	1	0	0	1	A
4	1	0	0	0	1	A
5	0	1	0	1	0	A
6	1	1	0	1	0	A
7	1	1	0	0	0	B
8	1	1	0	1	0	B
9	0	0	0	1	0	B
10	1	0	1	0	0	B
11	1	1	0	0	0	B
12	1	0	0	0	0	B

III.3 Tiếp cận logic (tt)

⇒ Sử dụng các mệnh đề logic khá đơn giản, ta đã xây dựng được một liên hệ “đặc trưng” cho nhóm hình A như sau:

$$\begin{aligned} \varphi = & P_1 P_2 P_3 P_4 \neg P_5 \vee P_1 \neg P_2 \neg P_3 P_4 \neg P_5 \vee \neg P_1 P_2 \neg P_3 \neg P_4 P_5 \vee \\ & \neg P_1 P_2 \neg P_3 \neg P_4 \neg P_5 \vee \neg P_1 P_2 \neg P_3 P_4 P_5 \vee P_1 P_2 \neg P_3 P_4 \neg P_5 \end{aligned}$$

⇒ Bằng các phép biến đổi logic toán học, ta có thể thu gọn mệnh đề trên thành :

$$\varphi = \neg P_1 P_2 \vee P_1 (P_2 P_3 \vee \neg P_2 \neg P_3)$$

Như vậy 1 hình x nào đó để được xếp vào nhóm hình A thì giá trị các mệnh đề P_1 đến P_6 của hình x phải thỏa mệnh đề φ ở trên.

III.3 Tiếp cận logic (tt)

Nhận xét:

❖ Trong trường hợp tổng quát, phải chọn các mệnh đề cơ sở (như các mệnh đề P_1, P_2, \dots, P_6 trong ví dụ trên) như thế nào để mệnh đề đặc trưng của tất cả các hình trong tập mẫu là khác nhau và mệnh đề đặc trưng của nhóm hình cũng phải khác nhau.

❖ Làm sao xây dựng thủ tục để kiểm tra giá trị các mệnh đề cơ sở. Mắt người có thể dễ dàng nhận biết sự tồn tại một hình tròn, hình tam giác, ... trong một hình ảnh có nhiều đối tượng khác nhau nhưng làm điều bằng chương trình máy tính hoàn toàn không đơn giản.

❖ Chính vì lý do đó, phương pháp học này rất cần đến sự hỗ trợ của con người trong việc đưa ra quyết định tính đúng đắn của các mệnh đề thành viên trong mệnh đề đặc trưng.

III.4 Học dựa trên cây định danh

❖ Dựa trên ý tưởng của tiếp cận hình học là phân chia không gian bài toán tạo thành một cây quyết định, người ta đã xây dựng các phương pháp học dựa trên việc xây dựng cây định danh.

❖ Xây dựng cây định danh bằng cách tìm các qui luật của dữ liệu.

Ví dụ: Xây dựng các quy luật để kết luận một người như thế nào khi đi tắm biển thì bị cháy nắng. Ta gọi tính chất cháy nắng hay không cháy nắng là thuộc tính quan tâm (thuộc tính mục tiêu).

$R = \{ \text{“cháy nắng”}, \text{“bình thường”} \}$

$P =$ tập hợp 8 người quan sát được

4 thuộc tính : chiều cao (cao, trung bình, thấp), màu tóc (vàng, nâu, đỏ) cân nặng (nhẹ, TB, nặng), dùng kem (có, không)

III.4 Học dựa trên cây định danh (tt)

Tên	Tóc	Ch.Cao	Cân Nặng	Dùng kem?	Kết quả
Sarah	Vàng	T.Bình	Nhẹ	Không	Cháy
Dana	Vàng	Cao	T.Bình	Có	Không
Alex	Nâu	Thấp	T.Bình	Có	Không
Annie	Vàng	Thấp	T.Bình	Không	Cháy
Emilie	Đỏ	T.Bình	Nặng	Không	Cháy
Peter	Nâu	Cao	Nặng	Không	Không
John	Nâu	T.Bình	Nặng	Không	Không
Kartie	Vàng	Thấp	Nhẹ	Có	Không

III.4 Học dựa trên cây định danh (tt)

❖ Tìm cách phân hoạch tập P ban đầu thành các tập P_i sao cho tất cả các phần tử trong tất cả các tập P_i đều có chung thuộc tính mục tiêu r_i ($r_i \in R$).

$$P = P_1 \cup P_2 \cup \dots \cup P_n \text{ và } \forall (i,j) \ i \neq j : \text{thì } (P_i \cap P_j = \emptyset) \text{ và}$$

$$\forall i, \forall k,l : p_k \in P_i \text{ và } p_l \in P_j \text{ thì } f(p_k) = f(p_l)$$

❖ Ứng với mỗi phân hoạch P_i ta xây dựng luật $L_i : GT_i \rightarrow r_i$ (GT_i là kết hợp các thuộc tính dẫn xuất)

❖ Có hai cách phân hoạch hiển nhiên: Cách đầu tiên là cho mỗi người vào một phân hoạch riêng ($P_1 = \{\text{Sarah}\}$, $P_2 = \{\text{Dana}\}$, ... Cách thứ hai là phân hoạch thành hai tập, một tập gồm tất cả những người cháy nắng và tập còn lại bao gồm tất cả những người không cháy nắng. **Tuy đơn giản nhưng phân hoạch theo kiểu này thì chúng ta chẳng giải quyết được gì !!**

III.4 Học dựa trên cây định danh (tt)

⇒ Đề xuất phương pháp giải quyết

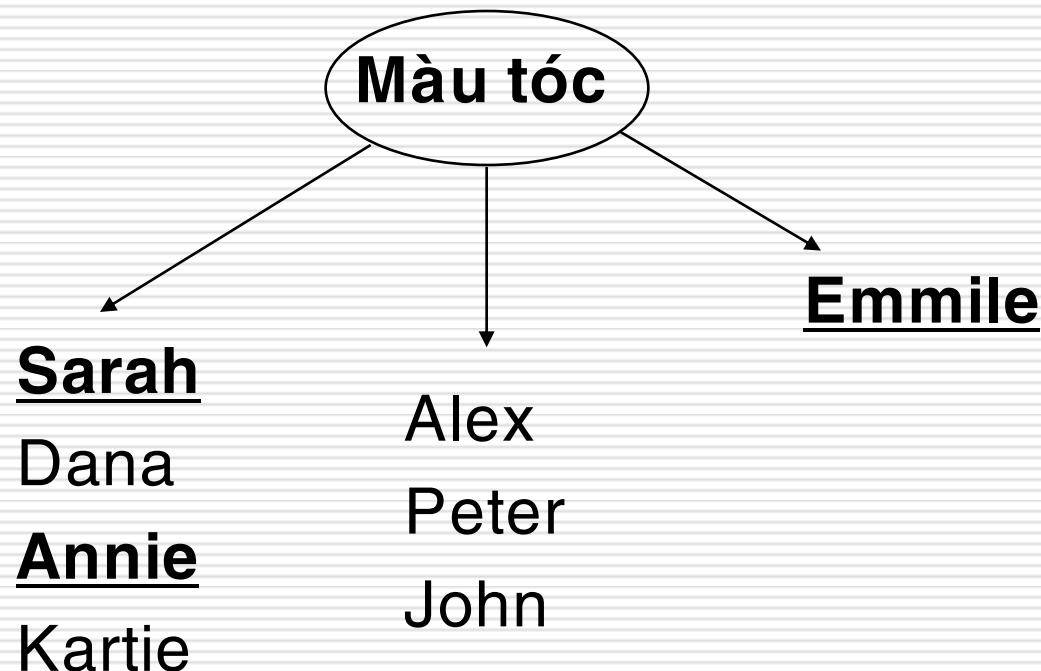
a. Đâm chôi

$P_{\text{vàng}} = \{ \underline{\text{Sarah}}, \text{Dana}, \underline{\text{Annie}}, \text{Kartie} \}$

$P_{\text{nâu}} = \{ \text{Alex}, \text{Peter}, \text{John} \}$

$P_{\text{đỏ}} = \{ \underline{\text{Emmille}} \}$

❖ Các người bị cháy nắng được gạch dưới và in đậm.



❖ $P_{\text{vàng}}$ là còn lẫn lộn người cháy nắng và không cháy nắng.

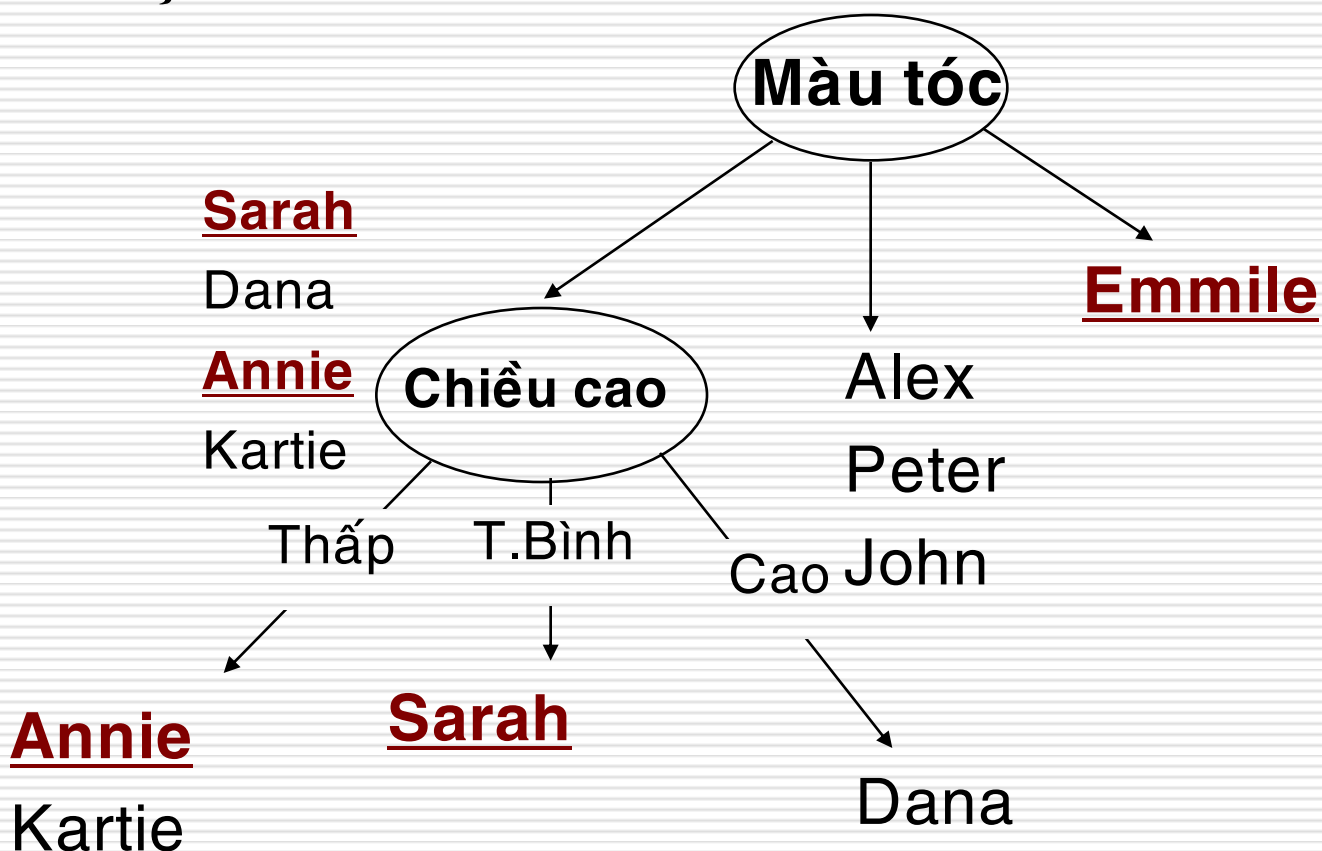
III.4 Học dựa trên cây định danh (tt)

❖ Quan sát thuộc tính chiều cao. Thuộc tính này giúp phân hoạch tập $P_{\text{vàng}}$ thành 3 tập con :

$P_{\text{Vàng, Thấp}} = \{\text{Annie, Kartie}\}$

$P_{\text{Vàng, T.Bình}} = \{\text{Sarah}\}$

$P_{\text{Vàng, Cao}} = \{\text{Dana}\}$



III.4 Học dựa trên cây định danh (tt)

Nhận xét:

- ❖ Quá trình này cứ thế tiếp tục cho đến khi tất cả các nút lá của cây không còn lẫn lộn giữa cháy nắng và không cháy nắng nữa.
- ❖ Qua mỗi bước phân hoạch cây phân hoạch ngày càng “phình” ra. Cây mà chúng ta đang xây dựng được gọi là cây định danh.

Vấn đề:

- ❖ Nếu như ban đầu ta không chọn thuộc tính màu tóc để phân hoạch mà chọn thuộc tính khác như chiều cao chẳng hạn để phân hoạch thì sao? Cuối cùng thì cách phân hoạch nào sẽ tốt hơn?

III.4 Học dựa trên cây định danh (tt)

b. Chọn thuộc tính phân hoạch

Quinlan: Với mỗi thuộc tính dẫn xuất A còn có thể sử dụng để phân hoạch, tính :

$$V_A(j) = (T(j, r_1), T(j, r_2), \dots, T(j, r_n))$$

$T(j, r_i) =$ (tổng số phần tử trong phân hoạch có giá trị thuộc tính dẫn xuất A là j và có giá trị thuộc tính mục tiêu là r_i) / (tổng số phần tử trong phân hoạch có giá trị thuộc tính dẫn xuất A là j)

trong đó r_1, r_2, \dots, r_n là các giá trị của thuộc tính mục tiêu

$$\sum_i T(j, r_i) = 1$$

Như vậy nếu một thuộc tính A có thể nhận một trong 5 giá trị khác nhau thì nó sẽ có 5 vector đặc trưng.

III.4 Học dựa trên cây định danh (tt)

- ❖ Một vector $V(A_j)$ được gọi là vector đơn vị nếu nó chỉ có duy nhất một thành phần có giá trị 1 và những thành phần khác có giá trị 0.

- ❖ Thuộc tính được chọn để phân hoạch là thuộc tính có nhiều vector đơn vị nhất.

III.4 Học dựa trên cây định danh (tt)

❖ Trở lại ví dụ của chúng ta, lúc ban đầu (chưa phân hoạch)

$$V_{\text{Tóc}}(\text{vàng}) = (T(\text{vàng, cháy nắng}), T(\text{vàng, không cháy nắng}))$$

Số người tóc vàng là : 4

Số người tóc vàng và cháy nắng là : 2

Số người tóc vàng và không cháy nắng là : 2

Do đó: $V_{\text{Tóc}}(\text{vàng}) = (2/4, 2/4) = (0.5, 0.5)$

III.4 Học dựa trên cây định danh (tt)

Tương tự

$$V_{\text{Tóc}}(\text{nâu}) = (0/3, 3/3) = (0,1) \text{ (vector đơn vị)}$$

Số người tóc nâu là : 3

Số người tóc nâu và cháy nắng là : 0

Số người tóc nâu và không cháy nắng là : 3

$$V_{\text{Tóc}}(\text{đỏ}) = (1/1, 0/1) = (1,0) \text{ (vector đơn vị)}$$

Tổng số vector đơn vị của thuộc tính tóc vàng là 2

III.4 Học dựa trên cây định danh (tt)

❖ Các thuộc tính khác được tính tương tự, kết quả như sau :

$$V_{C.Cao}(Cao) = (0/2, 2/2) = (0, 1)$$

$$V_{C.Cao}(T.B) = (2/3, 1/3)$$

$$V_{C.Cao}(Thấp) = (1/3, 2/3)$$

$$V_{C.Nặng}(Nhe) = (1/2, 1/2)$$

$$V_{C.Nặng}(T.B) = (1/3, 2/3)$$

$$V_{C.Nặng}(Nặng) = (1/3, 2/3)$$

$$V_{Kem}(Có) = (3/3, 0/3) = (1, 0)$$

$$V_{Kem}(Không) = (3/5, 2/5)$$

III.4 Học dựa trên cây định danh (tt)

❖ Như vậy thuộc tính màu tóc có số vector đơn vị nhiều nhất nên sẽ được chọn để phân hoạch.

❖ Phân hoạch theo tóc vàng ($P_{\text{vàng}}$) là còn chứa những người cháy nắng và không cháy nắng. Tiếp tục phân hoạch tập này. Tính vector đặc trưng tương tự đối với các thuộc tính còn lại (*chiều cao, cân nặng, dùng kem*). Trong phân hoạch $P_{\text{vàng}}$, tập dữ liệu của chúng ta còn lại là :

Tên	Ch.Cao	Cân Nặng	Dùng kem?	Kết quả
Sarah	T.Bình	Nhẹ	Không	Cháy
Dana	Cao	T.Bình	Có	Không
Annie	Thấp	T.Bình	Không	Cháy
Kartie	Thấp	Nhẹ	Có	Không

III.4 Học dựa trên cây định danh (tt)

$$V_{C.Cao}(Cao) = (0/1, 1/1) = \mathbf{(0,1)}$$

$$V_{C.Cao}(T.B) = (1/1, 0/1) = \mathbf{(1,0)}$$

$$V_{C.Cao}(Thấp) = (1/2, 1/2)$$

$$V_{C.Nặng}(Nhẹ) = (1/2, 1/2)$$

$$V_{C.Nặng}(T.B) = (1/2, 1/2)$$

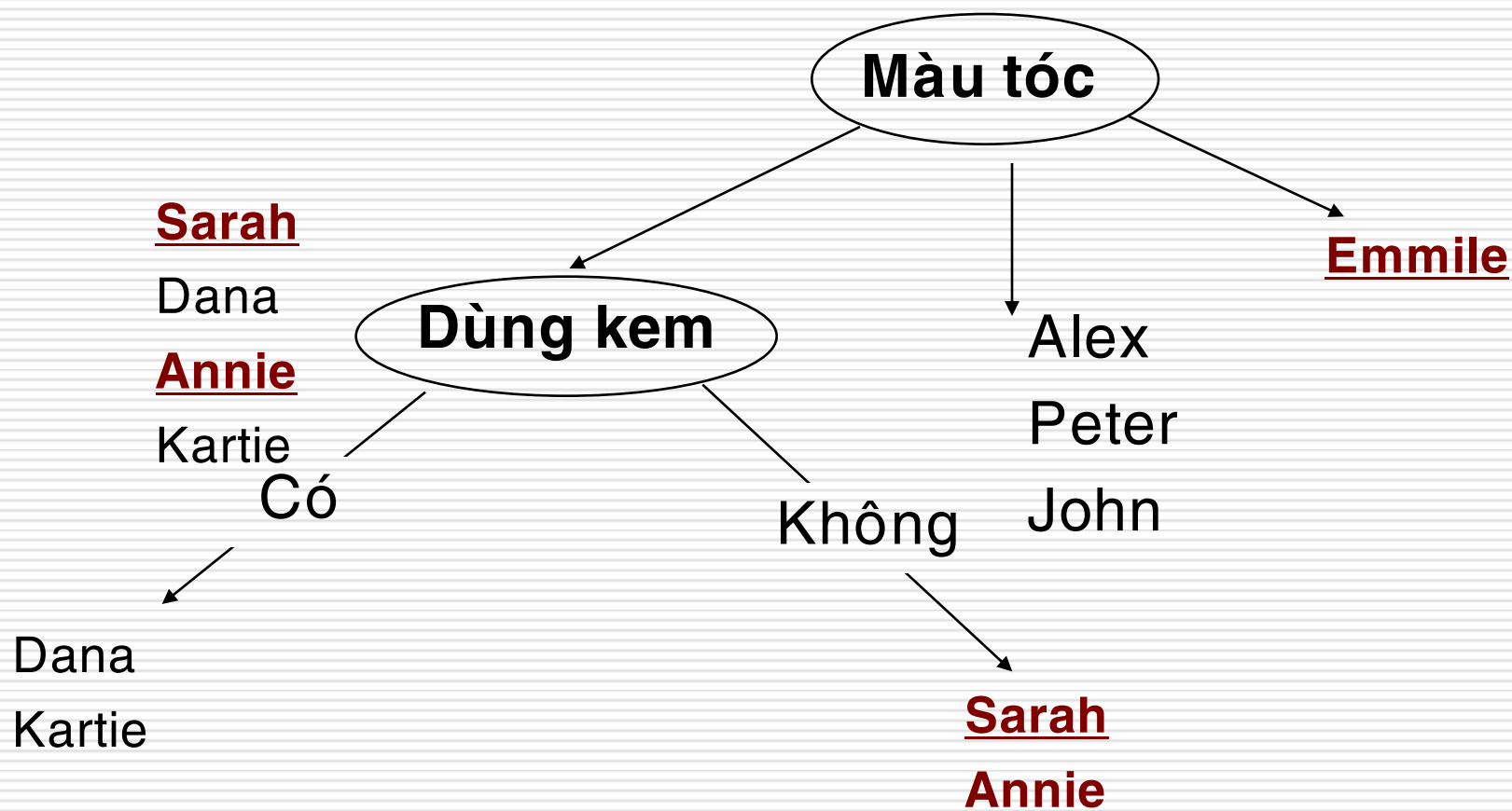
$$V_{C.Nặng}(Nặng) = (0,0)$$

$$V_{Kem}(Có) = (0/2, 2/2) = \mathbf{(0,1)}$$

$$V_{Kem}(Không) = (2/2, 0/2) = \mathbf{(1,0)}$$

III.4 Học dựa trên cây định danh (tt)

❖ 2 thuộc tính dùng kem và chiều cao đều có 2 vector đơn vị. Tuy nhiên, số phân hoạch của thuộc tính dùng kem là ít hơn nên ta chọn phân hoạch theo thuộc tính dùng kem. Cây định danh cuối cùng:



III.4 Học dựa trên cây định danh (tt)

Độ □ o hỗn loạn: Thay vì phải xây dựng các vector đặc trưng như phương pháp của Quinlan, ứng với mỗi thuộc tính dẫn xuất ta chỉ cần tính ra độ đo hỗn loạn và lựa chọn thuộc tính nào có độ đo hỗn loạn là thấp nhất. Công thức tính như sau:

$$T_A = \sum_j \left(\frac{b_j}{b_t} \times \sum_i \left(-\frac{b_{ri}}{b_j} \times \log_2 \left(-\frac{b_{ri}}{b_j} \right) \right) \right)$$

b_t là tổng số phần tử có trong phân hoạch

b_j là tổng số phần tử có thuộc tính dẫn xuất A có giá trị j.

b_{ri} : tổng số phần tử có thuộc tính dẫn xuất A có giá trị j và thuộc tính mục tiêu có giá trị i.

III.4 Học dựa trên cây định danh (tt)

c. Phát sinh tập luật

(Màu tóc vàng) và (có dùng kem) → không cháy nắng

(Màu tóc vàng) và (không dùng kem) → cháy nắng

(Màu tóc nâu) → không cháy nắng

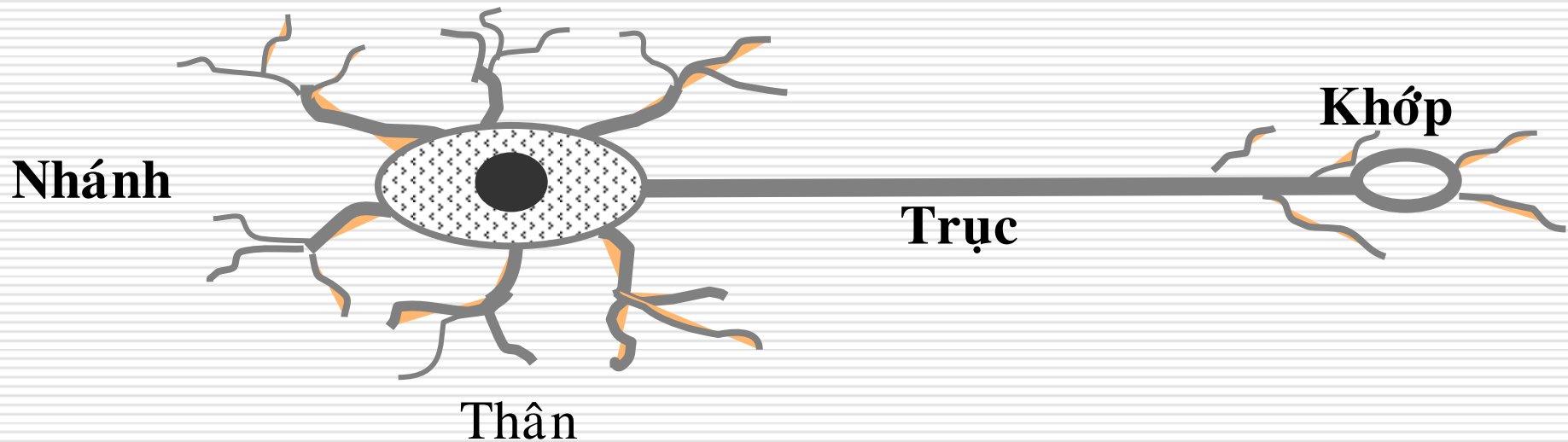
(Màu tóc đỏ) → cháy nắng

Vấn đề tiếp theo là tối ưu tập luật (tham khảo phần tối ưu hóa CSTT)

III.6 Tiếp cận mạng Neural

❖ Mạng neural là thuật ngữ nói đến một phương pháp giải quyết vấn đề – bài toán trên máy tính mô phỏng theo hoạt động của các tế bào thần kinh trong não bộ.

❖ Mạng neural nhân tạo là sự mô phỏng cấu trúc của mạng neural sinh học. Mạng neural nhân tạo được tạo thành bởi sự nối kết giữa rất nhiều đơn vị **thần kinh** gọi là perceptron.



Cấu trúc của một tế bào thần kinh sinh học

III.6 Tiếp cận mạng Neural

❖ Cấu tạo một đơn vị thần kinh nhân tạo (như hình vẽ)

~~Giá trị đầu ra y của một perceptron được tính bằng công thức sau:~~

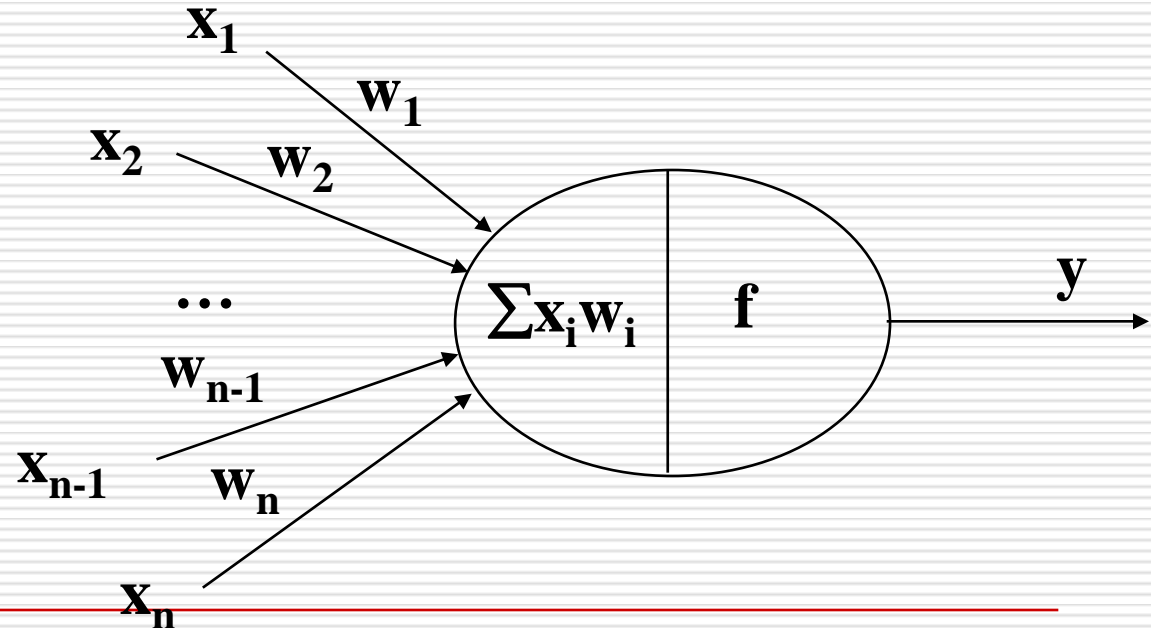
$$y = f((x_n w_n + x_{n-1} w_{n-1} + \dots + w_2 n_2 + w_1 n_1 + w_0) - \phi)$$

(ϕ được gọi là ngưỡng kích hoạt của neural)

Hàm f được gọi là hàm truyền. Một hàm truyền cần phải có tính chất sau

:

- bị chặn
- đơn điệu tăng
- hàm liên tục tăng



III.6 Tiếp cận mạng Neural (tt)

Các hàm truyền thường được sử dụng:

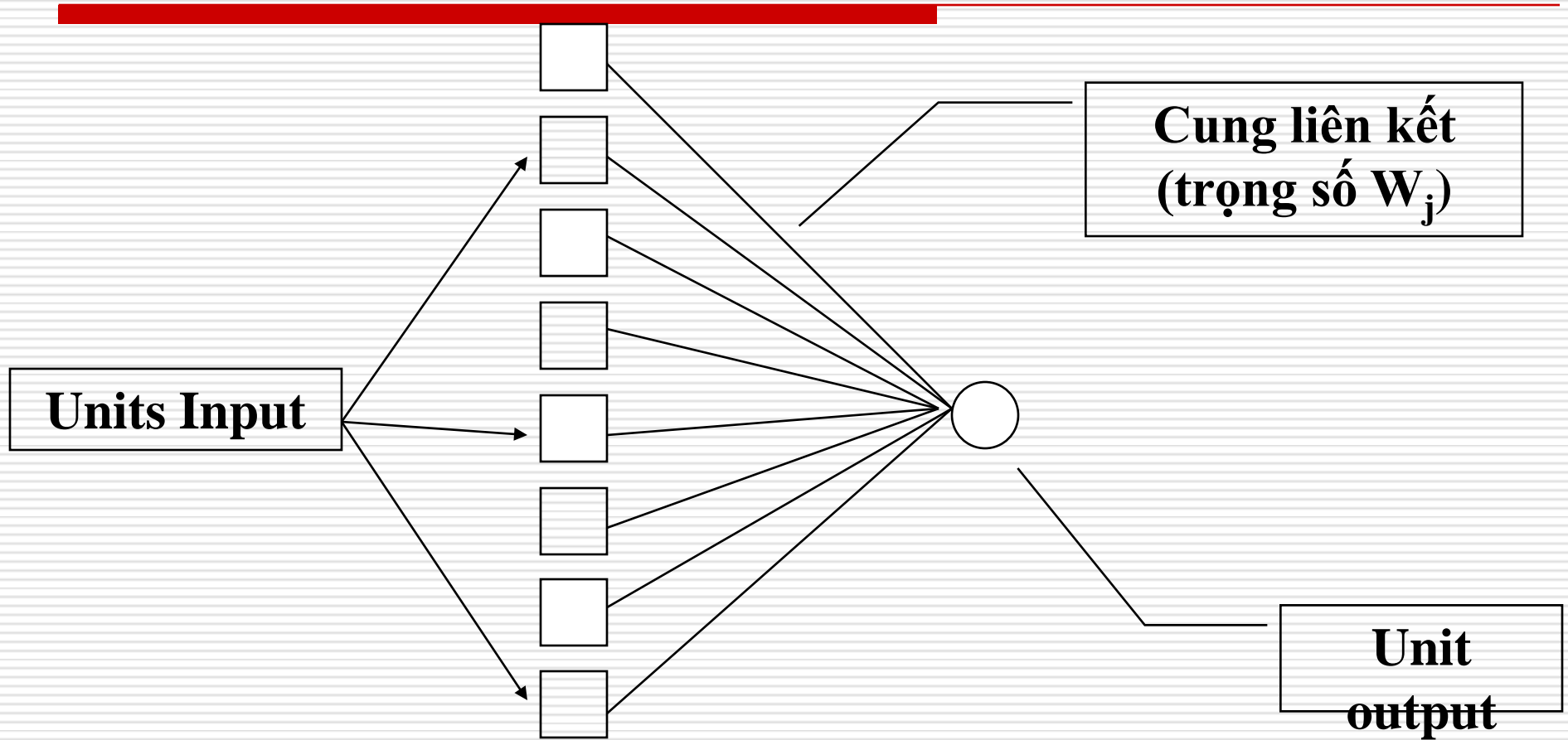
Hàm logistic (hay còn gọi là hàm Sigma) $f(x) = \frac{1}{1 + e^{-x}}$

Hàm hyperbol $h(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$

Hàm tang-hyperbol $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

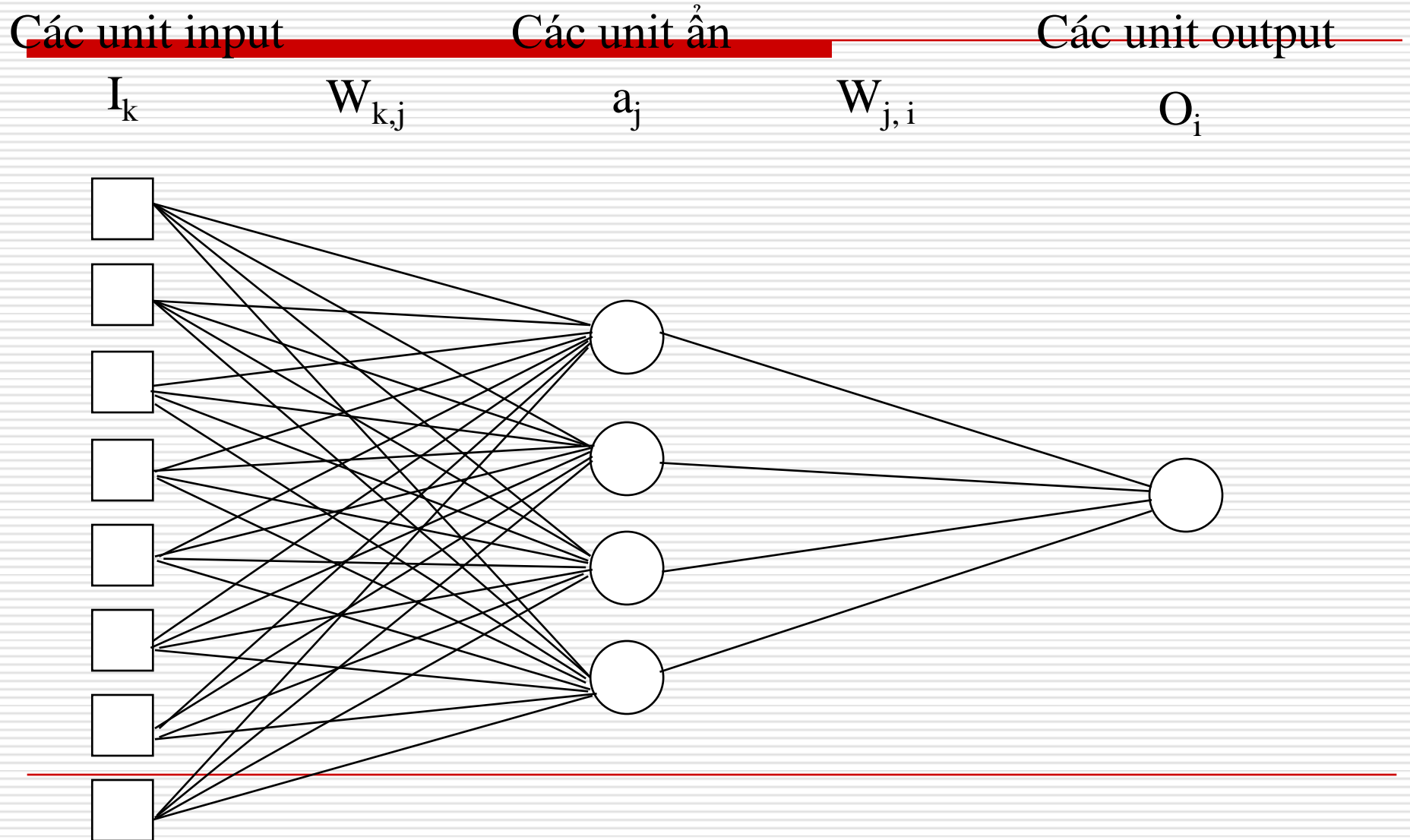
III.6 Tiếp cận mạng Neural (tt)

Mô hình minh họa mạng neural 1 lớp



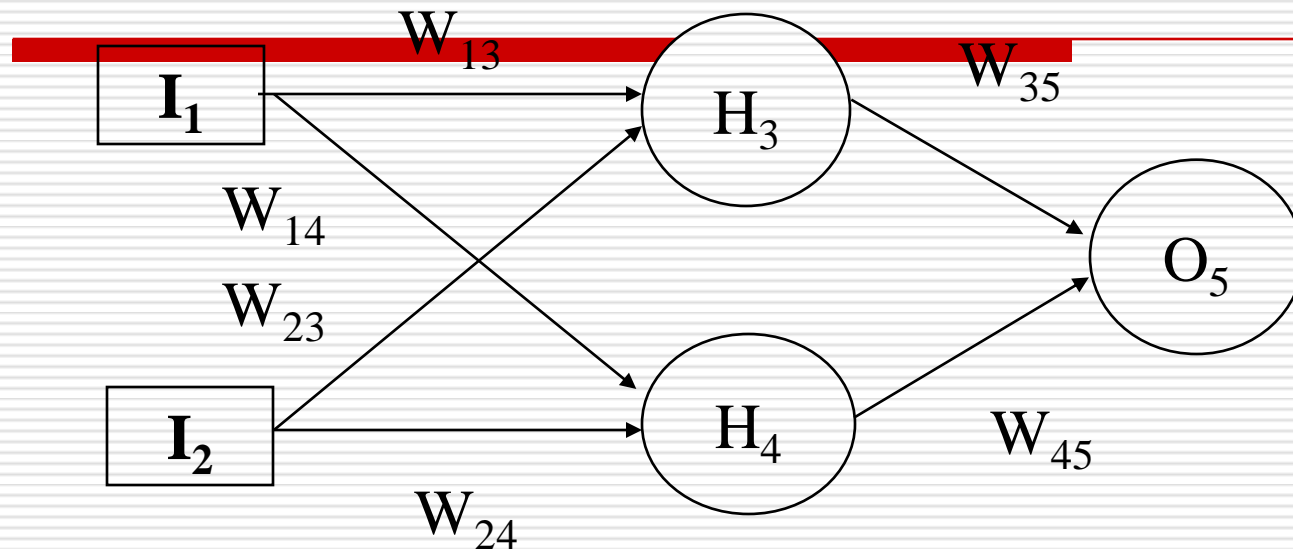
III.6 Tiếp cận mạng Neural (tt)

Mô hình minh họa mạng neural tổng quát



III.6 Tiếp cận mạng Neural (tt)

1. Mạng lan truyền (Feed Forward)



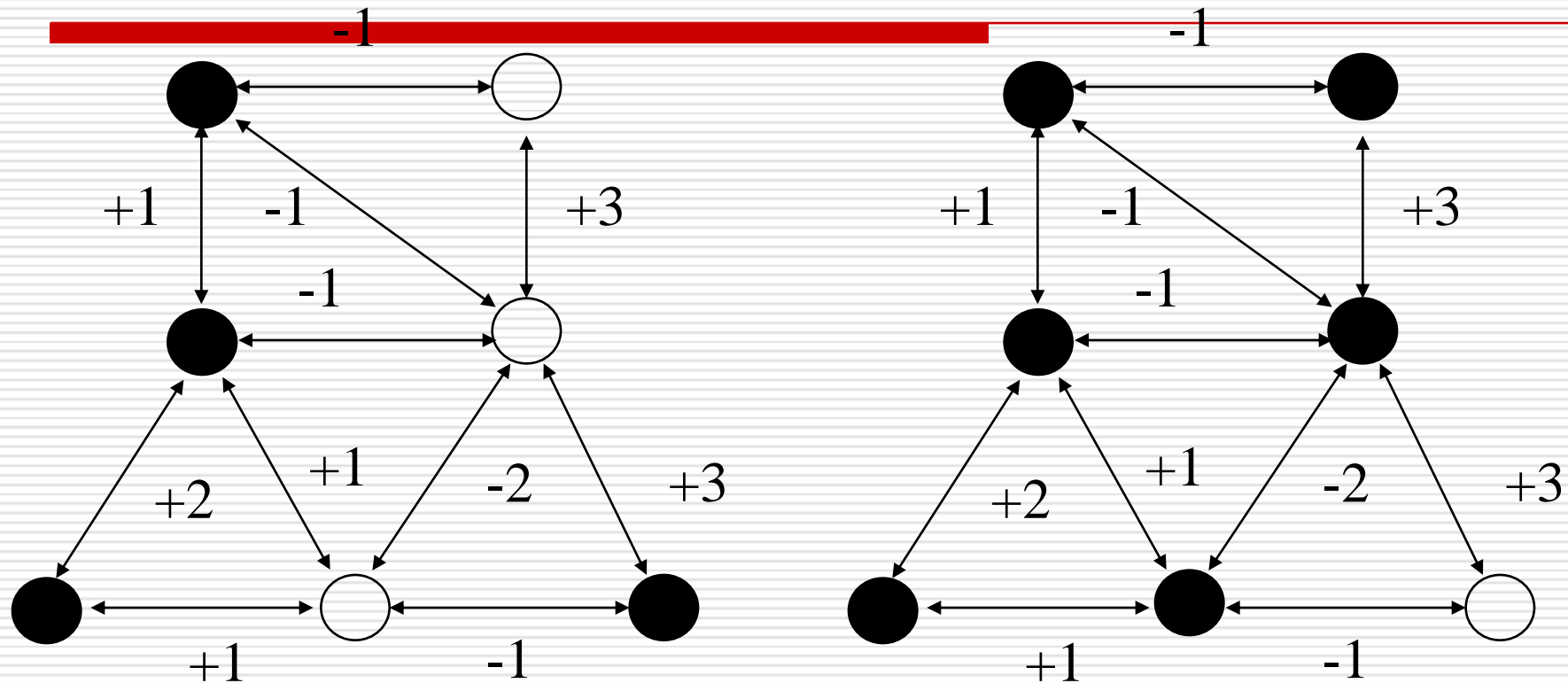
Nguyên tắc xác định giá trị Output của node 5:

$$a_5 = f(W_{3,5}a_3 + W_{4,5}a_4)$$

$$= f(W_{3,5}f(W_{1,3}a_1 + W_{2,3}a_2) + W_{4,5}f(W_{1,4}a_1 + W_{2,4}a_2))$$

III.6 Tiếp cận mạng Neural (tt)

2. Mạng Hopfield



Mạng Hopfield hoạt động tương tự hoạt động động một bộ nhớ kết hợp có:

$$W_{i,j} = W_{j,i}$$

III.5 Tiếp cận mạng Neural (tt)

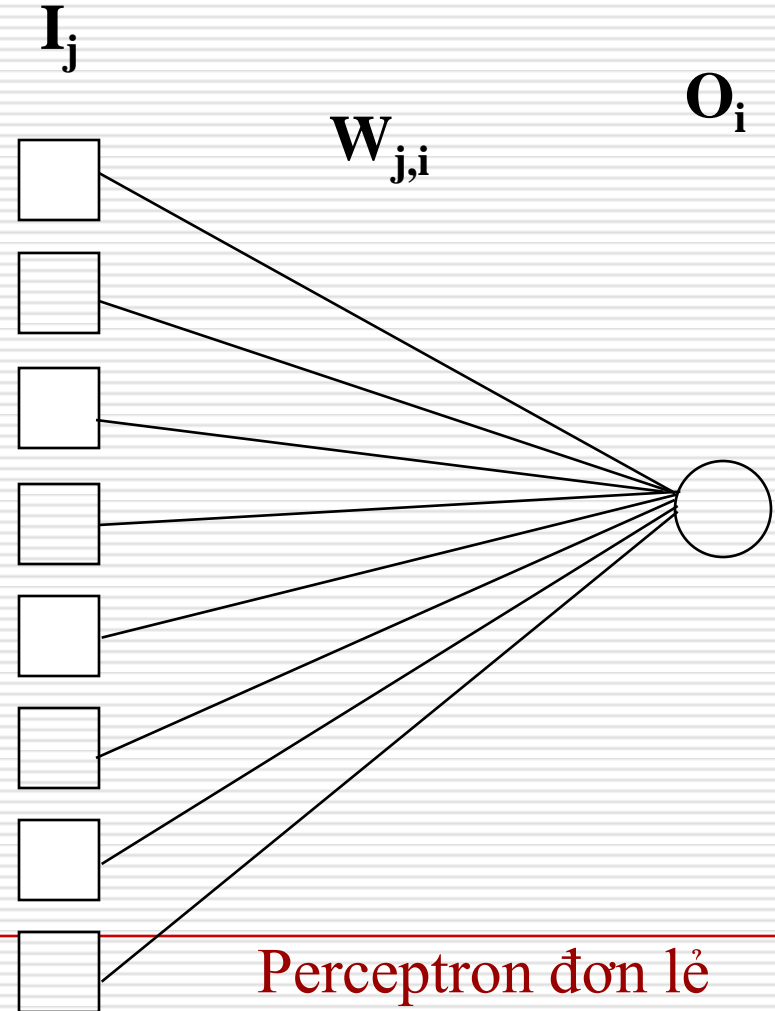
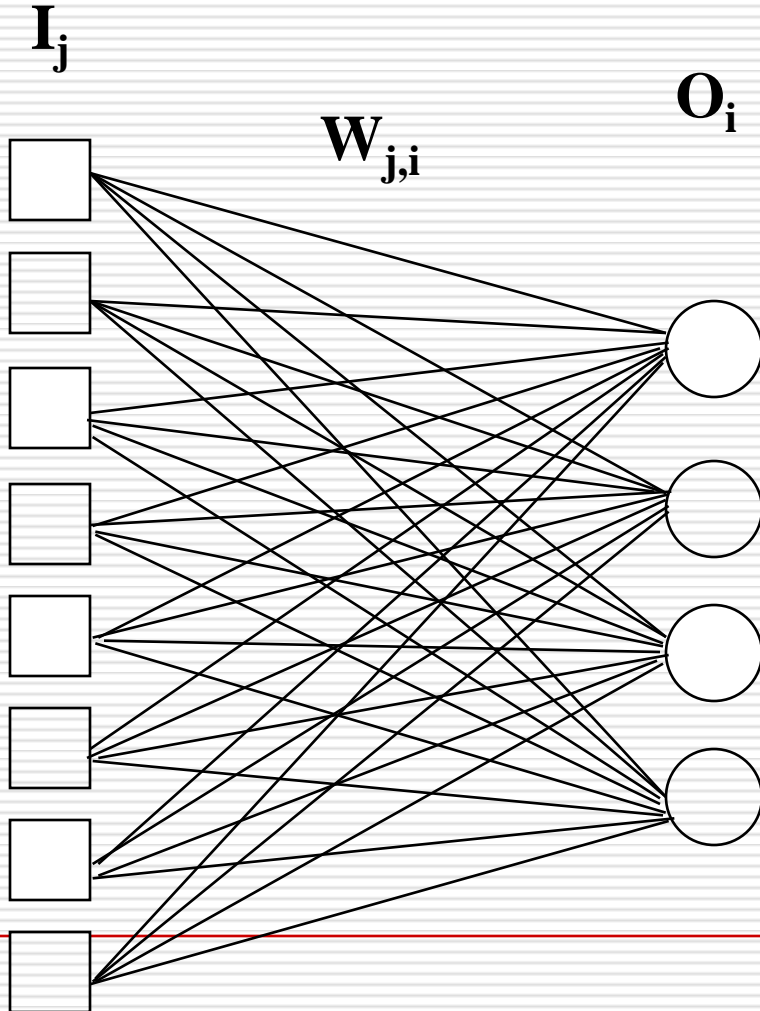
3. Mạng Perceptron

Các unit input

Các unit output

Các unit input

Các unit output



Perceptron đơn lẻ

III.5 Tiếp cận mạng Neural (tt)

Mạng Perceptron

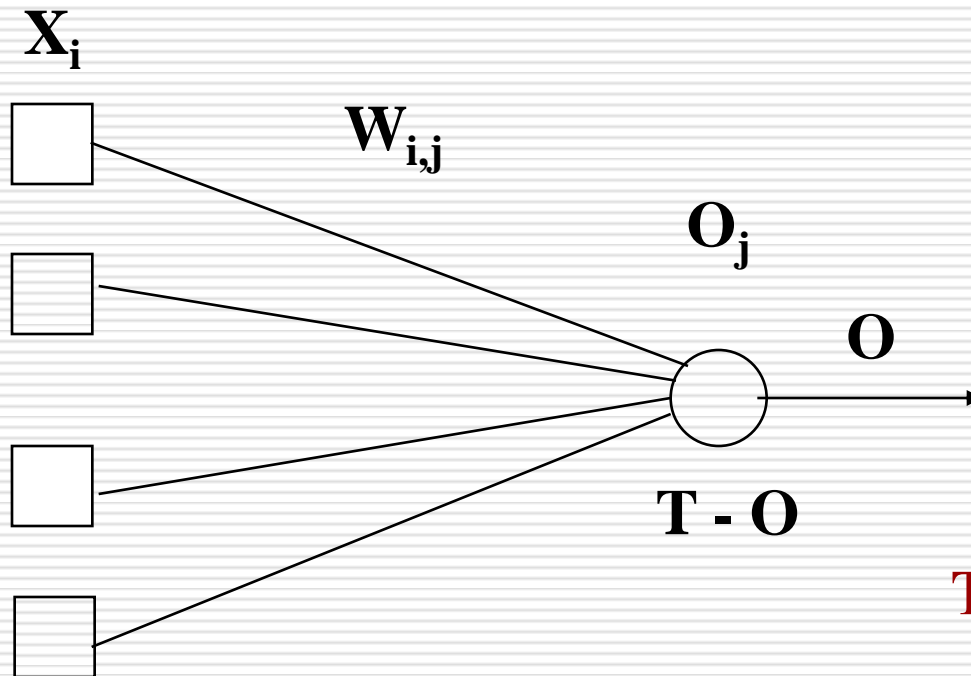
input

output

$$W_i = W_i + \alpha * X_i$$

$$W_i = W_i - \alpha * X_i$$

α : hệ số học



Perceptron đơn

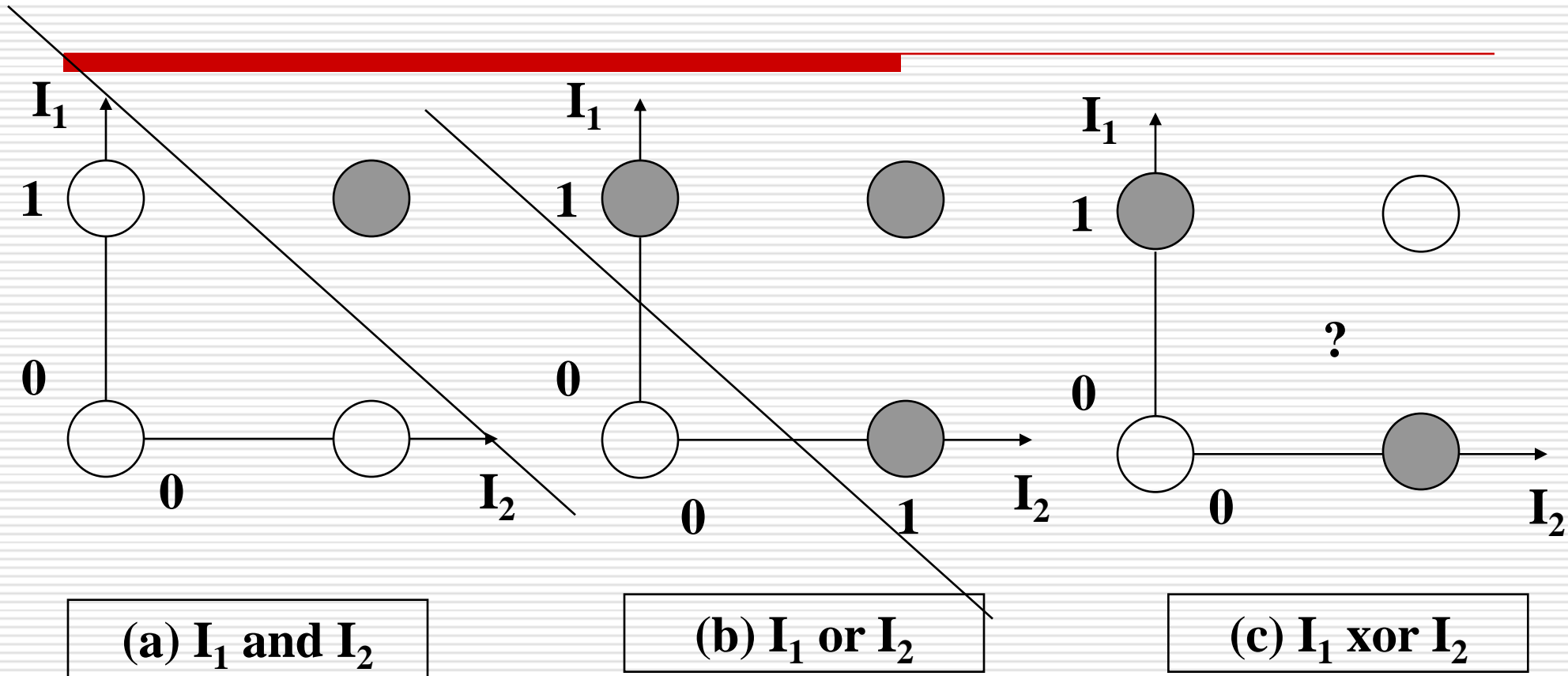
T : đầu ra mong muốn

O : đầu ra của mạng

lẽ

III.6 Tiếp cận mạng Neural (tt)

Khả năng phân chia tuyến tính trong mạng Perceptron



$$Y(I_1, I_2) = \begin{cases} 1 & \text{nếu } W_1 I_1 + W_2 I_2 > \theta \\ 0 & \text{ngược lại} \end{cases}$$

III.6 Tiếp cận mạng Neural (tt)

Hạn chế của mạng Perceptron

- Năm 1969, Minsky và Papert đã phân tích và chứng minh sự thất bại của mạng Perceptron với bài toán XOR.

I_1	I_2	$y(I_1, I_2) = I_1 \text{ XOR } I_2$
1	1	0
1	0	1
0	1	1
0	0	0

III.6 Tiếp cận mạng Neural (tt)

Xét từng trường hợp:

Nếu $[I_1, I_2]=[0,0]$ thì $y(I_1, I_2)=0 \Rightarrow W_1*0 + W_2*0 < \theta \Rightarrow \theta > 0$

Nếu $[I_1, I_2]=[0,1]$ thì $y(I_1, I_2)=1 \Rightarrow W_1*0 + W_2*1 \geq \theta \Rightarrow \theta \leq W_2$

Nếu $[I_1, I_2]=[1,0]$ thì $y(I_1, I_2)=1 \Rightarrow W_1*1 + W_2*0 \geq \theta \Rightarrow \theta \leq W_1$

Nếu $[I_1, I_2]=[1,1]$ thì $y(I_1, I_2)=0 \Rightarrow W_1*1 + W_2*1 < \theta \Rightarrow \theta > W_1 + W_2$

\Rightarrow Rõ ràng không thể tìm được W_1, W_2, θ thỏa mãn 4 bất đẳng thức trên.

\Rightarrow Mô hình Perceptron không thể tìm ra lời giải trong trường hợp này vì ~~hàm XOR không phải là hàm ngưỡng tuyến tính.~~

III.6 Tiếp cận mạng Neural (tt)

Ví dụ minh họa việc học của mạng Perceptron:

	Cột thêm vào	Độ dài đài hoa	Độ rộng đài hoa	Độ dài cánh hoa	Độ rộng cánh hoa	Loài
Cây Iris A	1	4.7	3.2	1.3	0.2	-1
Cây Iris B	1	6.1	2.8	4.7	1.2	1
Cây Iris C	1	5.6	3.0	4.1	1.3	1
Cây Iris D	1	5.8	2.7	5.1	1.9	-1
Cây Iris E	1	6.5	3.2	5.1	2.0	-1
Cây Iris Q	1	5.8	2.7	3.9	1.2	???

Khởi tạo trọng số: $w_1=1$; $w_2=0$; $w_3=0$; $w_4=0$; $w_5=1$

III.6 Tiếp cận mạng Neural (tt)

❖ Dự đoán lần thứ nhất đối với mẫu “Cây Iris A”

$$\sum_{i=1}^5 w_i x_i = 1 * 1 + 0 * 4.7 + 0 * 3.2 + 0 * 1.3 + 1 * 0.2 = 1.2$$

Perceptron dự đoán “Loài” có kết quả là 1 so với kết quả đúng là -1

❖ Cập nhật trọng số mạng

$$w_i \leftarrow w_i - \alpha * x_i \quad \text{Nếu kết quả “Loài” là 1 so với kết quả đúng là -1}$$

(α : Tỷ lệ học nên chọn nhỏ, ở đây ta chọn $\alpha = 0.05$)

III.6 Tiếp cận mạng Neural (tt)

$$w_1 \leftarrow w_1 - 0.05 * x_1 = 1 - 0.05 * 1 = 0.95$$

~~$$w_2 \leftarrow w_2 - 0.05 * x_2 = 0 - 0.05 * 4.7 = -0.24$$~~

$$w_3 \leftarrow w_3 - 0.05 * x_3 = 0 - 0.05 * 3.2 = -0.16$$

$$w_4 \leftarrow w_4 - 0.05 * x_4 = 0 - 0.05 * 1.3 = -0.07$$

$$w_5 \leftarrow w_5 - 0.05 * x_5 = 1 - 0.05 * 0.2 = 0.99$$

Các trọng số này dùng để luyện cho những mẫu kế tiếp

❖ **Điều kiện dừng:**

Tất cả các mẫu đã được dự đoán đúng.

Dừng sau một số bước lặp do người dùng quyết định.

III.6 Tiếp cận mạng Neural (tt)

Nhận xét: Cập nhật trọng số sao cho tối thiểu sai lệch.

~~(1) $w_i \leftarrow w_i - \alpha * x_i$ Nếu kết quả “Loài” là 1 so với kết quả đúng là -1~~

(2) $w_i \leftarrow w_i + \alpha * x_i$ Nếu kết quả “Loài” là -1 so với kết quả đúng 1

(1) $w_i x_i$ góp phần làm cho $\sum_{i=1}^n w_i x_i$

- Trọng số mới là $w_i \leftarrow w_i - \alpha * x_i$. Vì thế bước kế tiếp cho cùng mẫu dữ liệu ta được $(w_i - \alpha * x_i) * x_i = w_i x_i - \alpha * x_i^2 < w_i x_i$

- Như vậy $\sum_{i=1}^n w_i x_i$ sau khi thay đổi trọng số nhỏ hơn trước đó.

- ~~• Vì vậy kết quả dự đoán sẽ gần về kết quả đúng là -1.~~

III.6 Tiếp cận mạng Neural (tt)

Nguyên lý hoạt động của mạng Perceptron

Bước 1: Xác định giá trị của Err

$$Err = T - O \quad (T : \text{đầu ra mong muốn})$$

if $Err > 0$ then tăng O

else giảm O

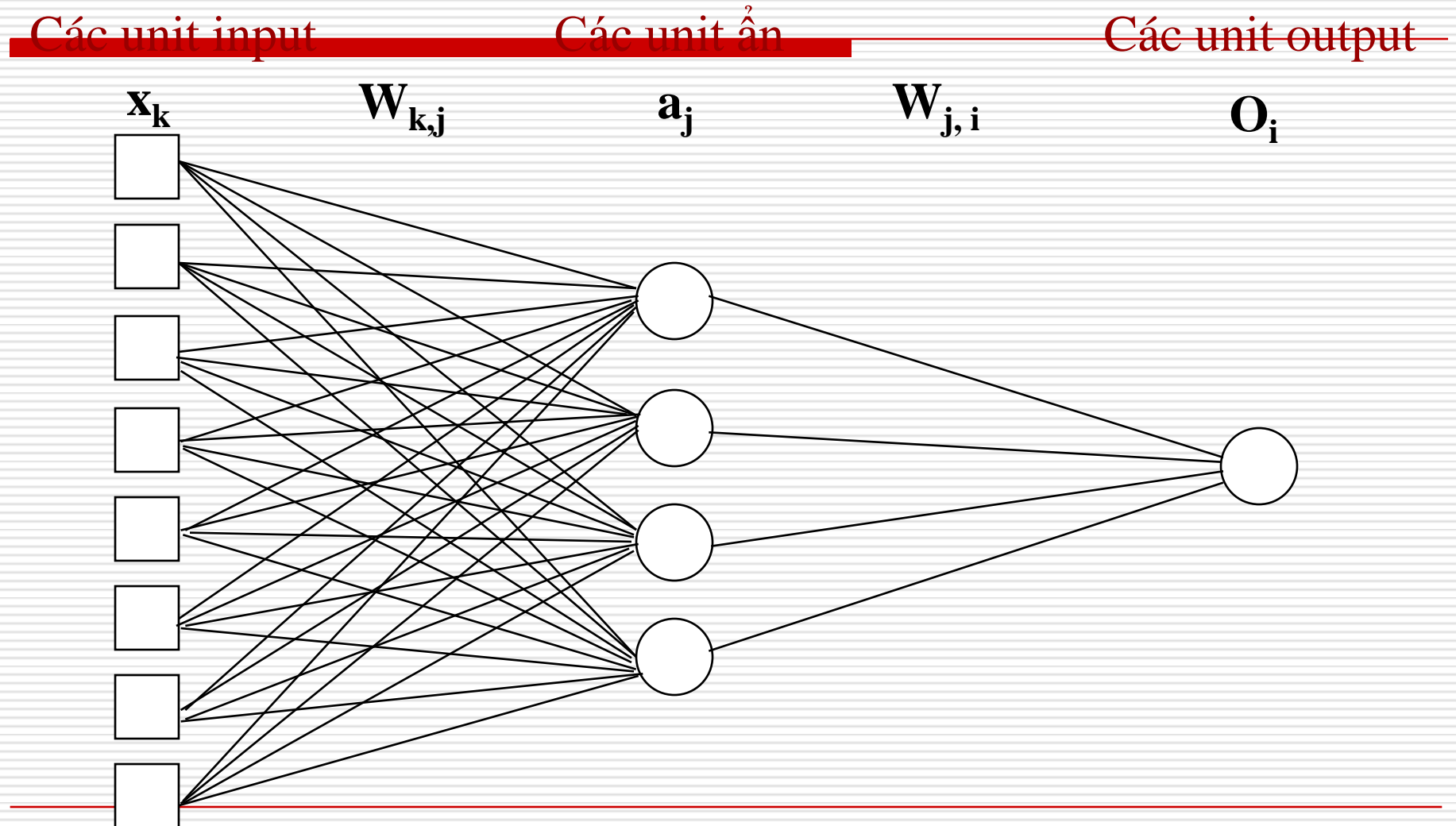
Bước 2: Cập nhật giá trị ma trận trọng số

$$W_j = W_j + \alpha * x_j * Err. \quad (\alpha \text{ là hệ số học})$$

Bước 3: Lặp lại bước 1 và 2 cho đến khi $Err < \epsilon$ thì dừng.

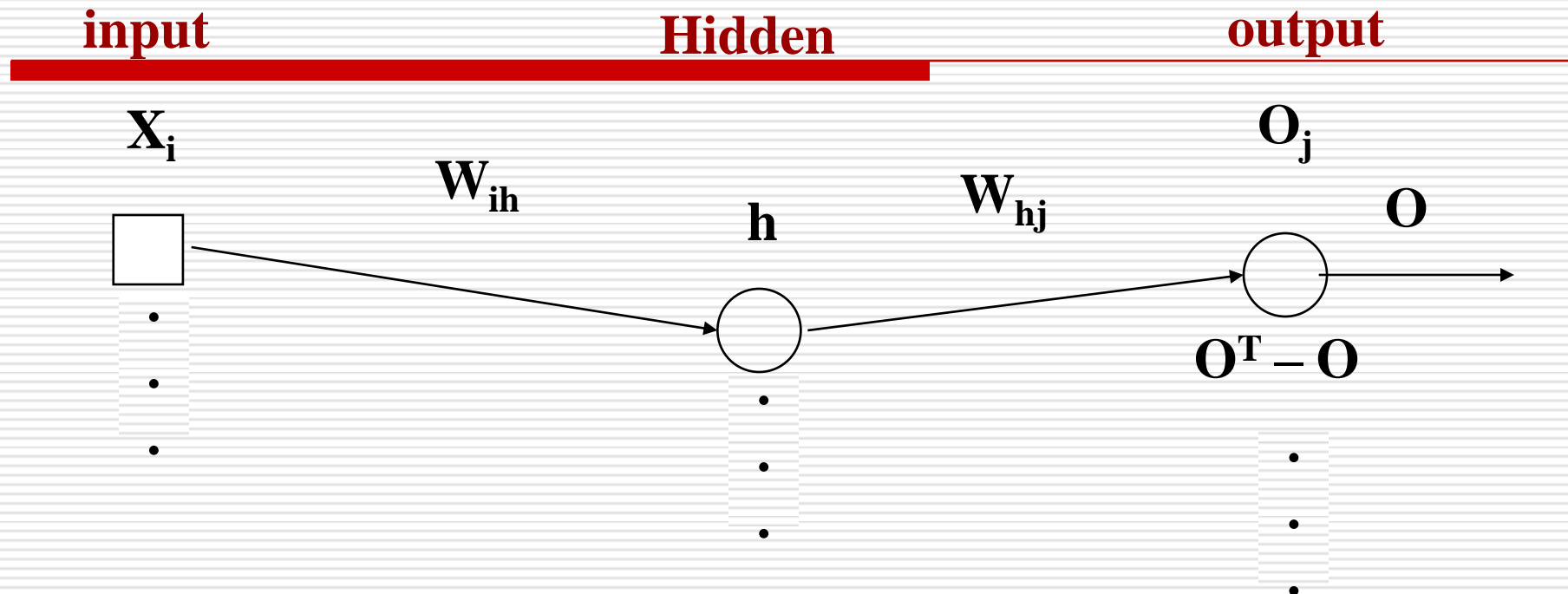
III.6 Tiếp cận mạng Neural (tt)

4. Mạng lan truyền ngược (back propagation)



III.5 Tiếp cận mạng Neural (tt)

Mạng lan truyền ngược (back propagation)



$$\delta_o = o(1-o)(T-o)$$

$$W_{hj} = W_{hj} + \alpha * \delta_{oj} * o_j$$

$$W_{ih} = W_{ih} + \alpha * \delta_o * W_h * o_h * (1-o_h) * X_i$$

III.6 Tiếp cận mạng Neural (tt)

Tập mẫu cho quá trình học

Mẫu	Đầu vào	Đầu ra mong muốn
1	$\mathbf{x}^1 = (x_1^1, x_2^1, \dots, x_n^1)$	y^1
2	$\mathbf{x}^2 = (x_1^2, x_2^2, \dots, x_n^2)$	y^2
...
M	$\mathbf{x}^m = (x_1^m, x_2^m, \dots, x_n^m)$	y^m

❖ Cho mẫu \mathbf{x}^k vào luyện \rightarrow Cho ra O^k sai lệch với kết quả mong muốn y^k . Sai lệch của mẫu thứ k là E_k :

$$E_k = 1/2 * (y^k - O^k)^2 = 1/2 (y^k - f(\mathbf{w}^T \mathbf{x}))^2$$

III.6 Tiếp cận mạng Neural (tt)

❖ Sai lệch của tất cả các mẫu:

$$E = \sum_{k=1}^m E_k = E_1 + E_2 + \dots + E_m$$

❖ Cập nhật trọng số: theo hướng cực tiểu hóa sai lệch

$$w = w - \alpha * E'_k(w)$$

α : là hệ số học

$$E'_k(w) = \frac{d}{dw} \left(\frac{1}{2} (y^k - f(w^T x))^2 \right) = -(y^k - f(w^T x)) f'(w^T x)$$

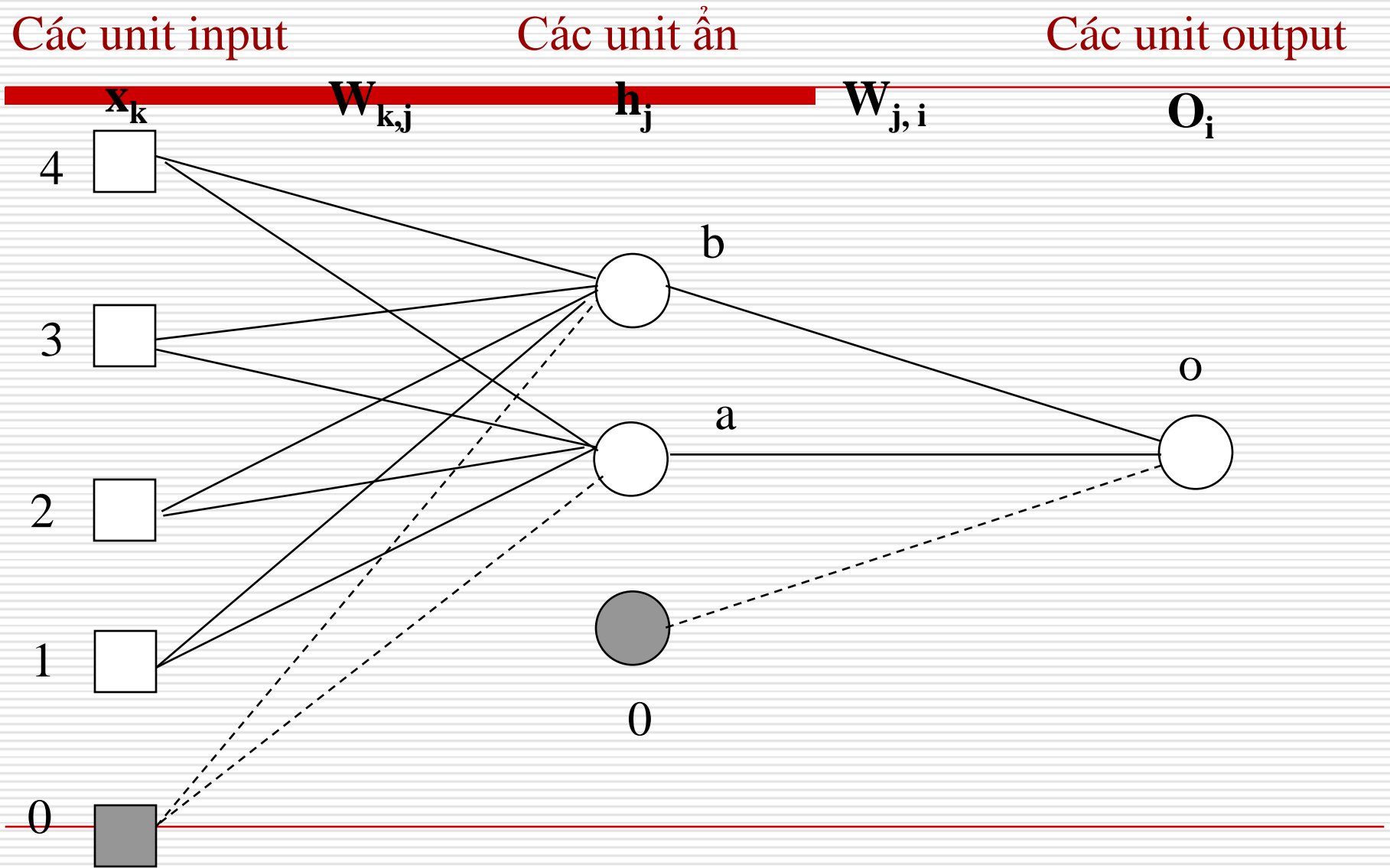
III.6 Tiếp cận mạng Neural (tt)

Ví dụ minh họa cho phương pháp lan truyền ngược

	Cột thêm vào	Độ dài đài hoa	Độ rộng đài hoa	Độ dài cánh hoa	Độ rộng cánh hoa	Loài
Cây Iris A	1	4.7	3.2	1.3	0.2	0
Cây Iris B	1	6.1	2.8	4.7	1.2	1
Cây Iris C	1	5.6	3.0	4.1	1.3	1
Cây Iris D	1	5.8	2.7	5.1	1.9	0
Cây Iris E	1	6.5	3.2	5.1	2.0	0
Cây Iris Q	1	5.8	2.7	3.9	1.2	???

Chọn hàm truyền: $f(x) = \frac{1}{1 + e^{-x}}$

III.6 Tiếp cận mạng Neural (tt)



III.6 Tiếp cận mạng Neural (tt)

Khởi tạo các trọng số có giá trị ngẫu nhiên nhỏ

$W_{0a} = 0.0$ Trọng số cột giá trị -1 trong đơn vị ẩn a

$W_{1a} = 0.1$ Trọng số trong đơn vị ẩn a từ giá trị nhập thứ 1

$W_{2a} = -0.1$ Trọng số trong đơn vị ẩn a từ giá trị nhập thứ 2

$W_{3a} = -0.1$ Trọng số trong đơn vị ẩn a từ giá trị nhập thứ 3

$W_{4a} = 0.0$ Trọng số trong đơn vị ẩn a từ giá trị nhập thứ 4

III.6 Tiếp cận mạng Neural (tt)

$W_{0b} = 0.0$	Trọng số cột giá trị -1 trong đơn vị ẩn b
$W_{1b} = -0.1$	Trọng số trong đơn vị ẩn b từ giá trị nhập thứ 1
$W_{2b} = 0.2$	Trọng số trong đơn vị ẩn b từ giá trị nhập thứ 2
$W_{3b} = 0.1$	Trọng số trong đơn vị ẩn b từ giá trị nhập thứ 3
$W_{4b} = -0.1$	Trọng số trong đơn vị ẩn b từ giá trị nhập thứ 4
$W_{0o} = 0.1$	Trọng số cột giá trị -1 trong đơn vị xuất o
$W_{ao} = 0.2$	Trọng số cột trong đơn vị ẩn b từ đơn vị ẩn a
$W_{bo} = -0.1$	Trọng số cột trong đơn vị ẩn b từ đơn vị ẩn b

III.6 Tiếp cận mạng Neural (tt)

Bước 1: Dự đoán

➤ Đối với “cây Iris A” đầu tiên ta tính giá trị xuất của nút ẩn a

$$\begin{aligned}o_a &= f(w_{0a} + w_{1a}x_1 + w_{2a}x_2 + w_{3a}x_3 + w_{4a}x_4) \\ &= f(0.0 + 0.1*4.7 + (-0.1)*3.2 + (-0.1)*1.3 + 0.0*0.2) \\ &= f(0.02) = 1/(1 + e^{-0.02}) = 0.5050\end{aligned}$$

➤ Tương tự với nút ẩn b

$$\begin{aligned}o_b &= f(w_{0b} + w_{1b}x_1 + w_{2b}x_2 + w_{3b}x_3 + w_{4b}x_4) \\ &= f(0.0 + (-0.1)*4.7 + 0.2*3.2 + 0.1*1.3 + (-0.1)*0.2) \\ &= f(0.28) = 1/(1 + e^{-0.28}) = 0.5695\end{aligned}$$

III.6 Tiếp cận mạng Neural (tt)

➤ **Cuối cùng ta tính giá trị xuất cho nút xuất o**

$$\begin{aligned}o_o &= f(w_{0o} + w_{ao}o_a + w_{bo}o_b) \\ &= f(0.1 + 0.2*0.5050 + (-0.1)*0.5695) \\ &= f(0.1440) = 1/(1 + e^{-0.1440}) = 0.5359\end{aligned}$$

⇒ **Giá trị dự đoán cho “cây Iris A” là 0.5359**

III.6 Tiếp cận mạng Neural (tt)

Bước 2: Xác định sai lệch

➤ Dùng kỹ thuật lan truyền ngược (back propagation)

Với t_o : giá trị xuất mong muốn

o_o : giá trị xuất thật

Sai lệch cần tính cho nút xuất o : $\delta_o = o_o(1 - o_o)(t_o - o_o)$

Sai lệch cần tính cho nút ẩn h từ o là: $o_h(1 - o_h)w_{ho} \delta_o$

➤ Trong ví dụ đang xét “cây Iris A” có $t_o = 0$.

Sai lệch của nút xuất o :

$$\delta_o = o_o(1 - o_o)(t_o - o_o) = 0.5359 * (1 - 0.5359) * (0 - 0.5359) = -0.1333$$

III.6 Tiếp cận mạng Neural (tt)

Sai lệch của **nút ẩn a**:

$$\begin{aligned}\delta_a &= o_a(1-o_a)w_{ao} \delta_o = 0.5050*(1-0.5050)*0.2*(-0.1333) \\ &= -0.0067\end{aligned}$$

Sai lệch của **nút ẩn b**:

$$\begin{aligned}\delta_b &= o_b(1-o_b)w_{bo} \delta_o = 0.5695*(1-0.5695)*(-0.1)*(-0.1333) \\ &= 0.0032\end{aligned}$$

III.6 Tiếp cận mạng Neural (tt)

Bước 3: Bước cập nhật trọng số

Chọn tỉ lệ học $\alpha = 0.1$

$$w_{0a} = w_{0a} + \alpha * \delta_a * 1 = 0.0 + 0.1 * (-0.0067) * 1 = -0.0007$$

$$w_{1a} = w_{1a} + \alpha * \delta_a * x_1 = 0.1 + 0.1 * (-0.0067) * 4.7 = 0.0969$$

$$w_{2a} = w_{2a} + \alpha * \delta_a * x_2 = -0.1 + 0.1 * (-0.0067) * 3.2 = -0.1021$$

$$w_{3a} = w_{3a} + \alpha * \delta_a * x_3 = -0.1 + 0.1 * (-0.0067) * 1.2 = -0.1009$$

$$w_{4a} = w_{4a} + \alpha * \delta_a * x_4 = 0.0 + 0.1 * (-0.0067) * 0.2 = -0.0001$$

III.6 Tiếp cận mạng Neural (tt)

$$W_{0b} = W_{0b} + \alpha * \delta_b * 1 = 0.0 + 0.1 * (0.0032) * 1 = 0.0003$$

$$W_{1b} = W_{1b} + \alpha * \delta_b * x_1 = -0.1 + 0.1 * (0.0032) * 4.7 = -0.0985$$

$$W_{2b} = W_{2b} + \alpha * \delta_b * x_2 = 0.2 + 0.1 * (0.0032) * 3.2 = 0.2010$$

$$W_{3b} = W_{3b} + \alpha * \delta_b * x_3 = 0.1 + 0.1 * (0.0032) * 1.2 = 0.1004$$

$$W_{4b} = W_{4b} + \alpha * \delta_b * x_4 = -0.1 + 0.1 * (0.0032) * 0.2 = -0.0999$$

$$W_{0o} = W_{0o} + \alpha * \delta_o * 1 = 0.1 + 0.1 * (-0.1333) * 1 = 0.0867$$

$$W_{ao} = W_{ao} + \alpha * \delta_o * o_a = 0.2 + 0.1 * (-0.1333) * 0.5050 = 0.1933$$

$$W_{bo} = W_{bo} + \alpha * \delta_o * o_b = -0.1 + 0.1 * (-0.1333) * 0.5695 = -0.1076$$

~~Tất cả các trọng số này sẽ dùng cho các mẫu kế tiếp.~~

III.6 Tiếp cận mạng Neural (tt)

➤ Thử tính cho “cây Iris A” thêm một bước kế tiếp, có kết quả:

$$O_a = f (-0.0007 + 0.0969*4.7 + (-0.1021)*3.2 + (-0.1009)*1.3 + (-0.0001)*0.2) = f (-0.0032) = 0.4992$$

$$O_b = f (0.0003 + -(0.0985)*4.7 + 0.2010*3.2 + 0.1004*1.3 + (-0.0999)*0.2) = f (0.2911) = 0.5724$$

$$O_o = f (0.0867 + 0.1933*0.4992 + (-0.1076)*0.5724) = f (0.1440) = 0.5304$$

➤ Kết quả, ta thấy 0.5304 gần về 0 (kết quả mong muốn) so với kết quả của bước trước đó là 0.5359.

III.6 Tiếp cận mạng Neural (tt)

Tóm tắt giải thuật

Cho trước K mẫu dữ liệu :

$$\left\{ \left(x^1, y^1 \right), \dots, \left(x^k, y^k \right) \right\}$$

với $\mathbf{x}^k = (x_1^k, x_2^k, \dots, x_n^k)$ và $y^k \in \mathbb{R}$, $k = 1, \dots, K$.

Bước 1: chọn trước giá trị $\alpha > 0$ và $E_{\max} > 0$.

Bước 2: khởi tạo ngẫu nhiên w , bắt đầu với mẫu thứ nhất $k = 1$ và gán sai lệch $E = 0$.

Bước 3: bắt đầu quá trình học, gán $\mathbf{x} = \mathbf{x}^k$, $y = y^k$. Đầu ra của mạng neuron tính theo:

$$O^k = \frac{1}{1 + \exp(-W^T o^k)}$$

III.6 Tiếp cận mạng Neural (tt)

trong đó o^k là giá trị ra của lớp ẩn, được tính bởi:

$$o_l^k = \frac{1}{1 + \exp(-w_l^T x_k)}$$

Bước 4: cập nhật trọng số neuron đầu ra: $W = W + \alpha \delta_o$

với $\delta = (y - O)O(1 - O)$

Bước 5: cập nhật trọng số của neuron ẩn

$$w_l = w_l + \alpha \delta W_l o_l (1 - o_l) x \quad (\text{cho } l = 1, \dots, L)$$

III.6 Tiếp cận mạng Neural (tt)

Bước 6: tính sai lệch bằng cách cộng thêm sai lệch hiện tại

$$E = E + \frac{1}{2} (y - O)^2$$

Bước 7: nếu $k < K$ thì $k = k + 1$ và trở lại Bước 3. Nếu không thì qua Bước 8.

Bước 8: Nếu $E \leq E_{\max}$ thì kết thúc quá trình học. Còn nếu $E > E_{\max}$ gán $E = 0$ và bắt đầu một chu kỳ học mới bằng cách trở lại Bước 3.

Chương 7:

Khai mở dữ liệu

Tham khảo thêm:

[1] GS.TSKH Hoàng Kiếm. Bài giảng cao học môn học cơ sở tri thức và ứng dụng. ĐHKHTN-TPHCM.

[2] Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski. Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, 1998

[3] Citeseer - Scientific Literature Digital Library. Artificial Intelligence-<http://citeseer.nj.nec.com/ArtificialIntelligence/> - 2003

I. Một số bài toán điển hình về data mining

- ❖ Bài toán khám phá luật kết hợp
- ❖ Bài toán phân loại dữ liệu
- ❖ Bài toán gom nhóm dữ liệu
- ❖ Bài toán lập mô hình
- ❖ Bài toán dự báo
- ❖ ...



I. Một số bài toán điển hình về data mining (tt)

a. Phát hiện luật kết hợp (association rules)

- ❖ Tìm ra những mối liên hệ giữa các trường mô tả đối tượng trong CSDL và xây dựng thành các luật cụ thể.
- ❖ Luật kết hợp là tri thức quan trọng nhất tiềm ẩn trong CSDL.

Ví dụ: Sau khi phân tích một CSDL bán hàng ta tìm ra được các luật

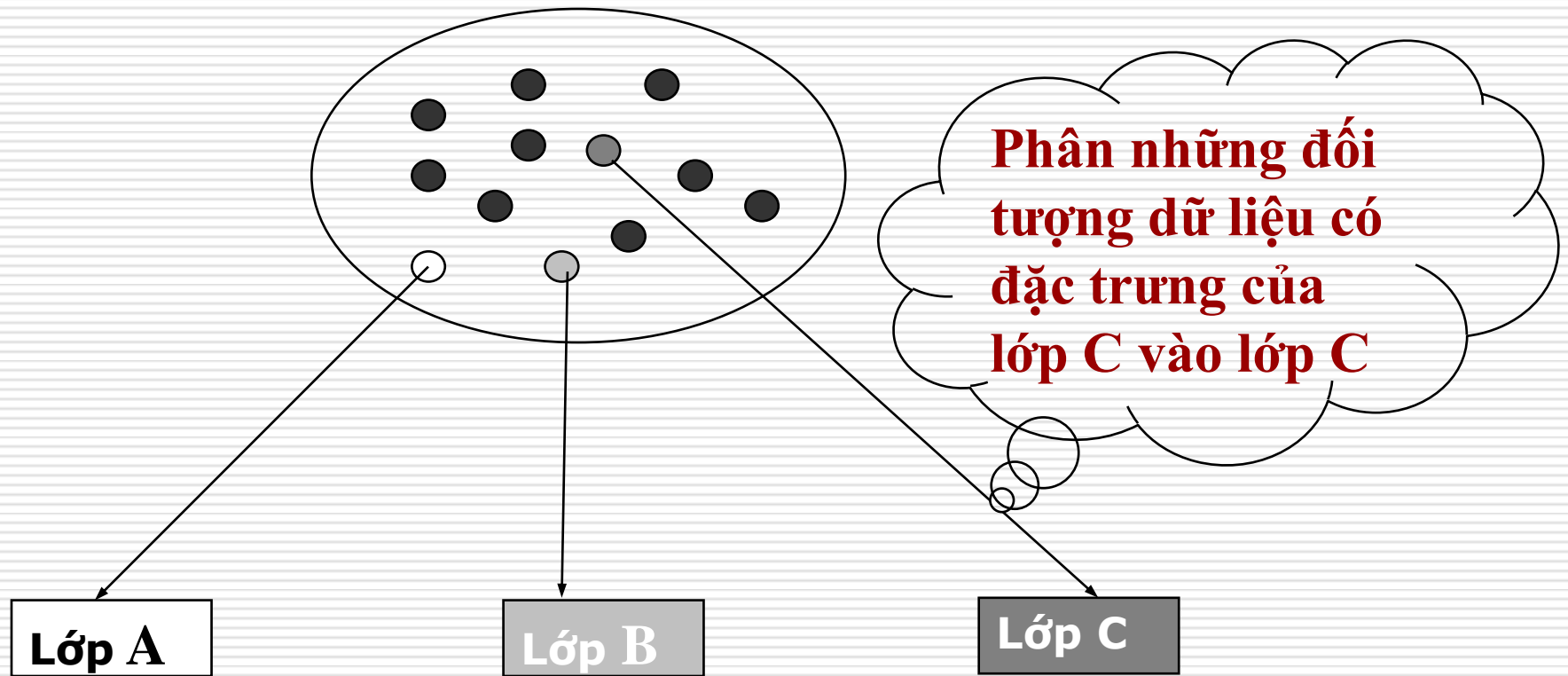
(1): Nếu khách hàng mua món A cũng sẽ mua món B.

(2): Nếu khách hàng mua món C thì tháng sau sẽ mua món D.

...

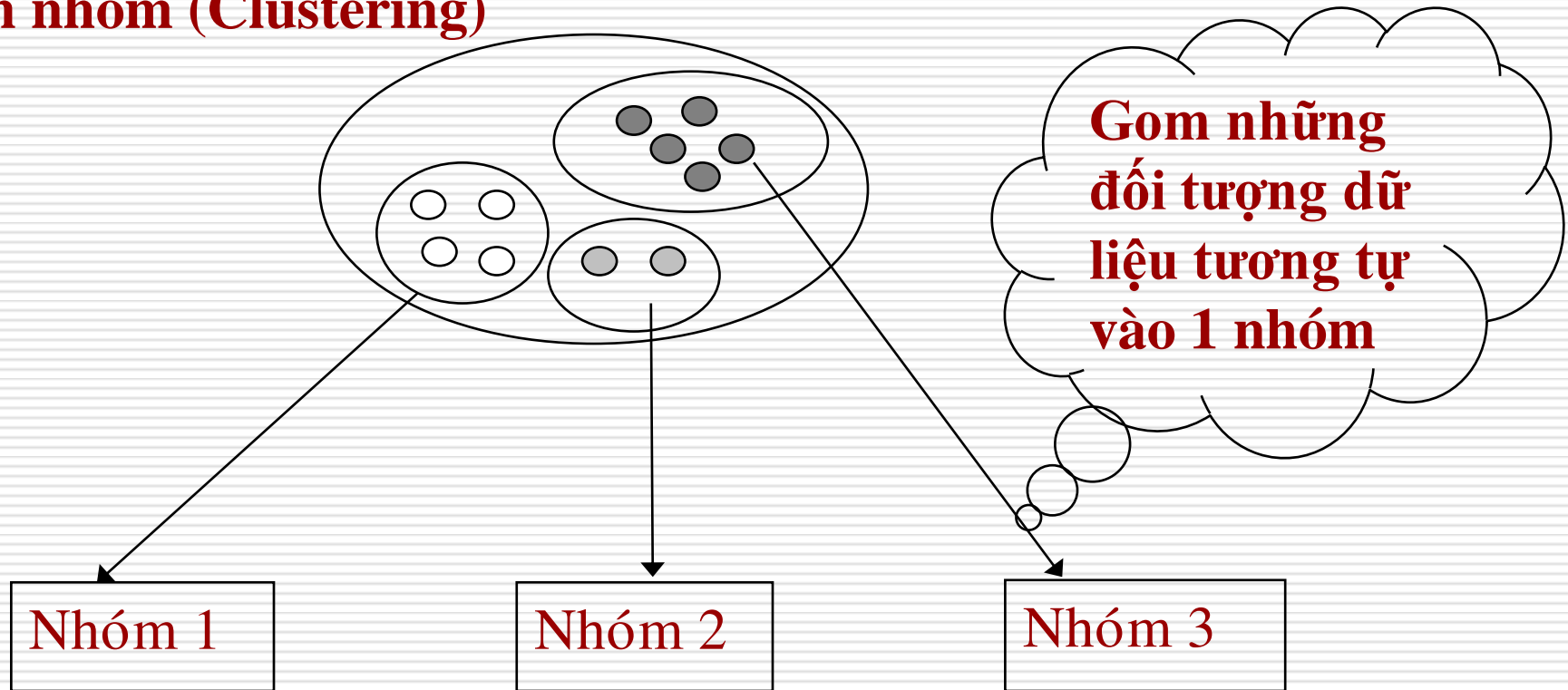
I. Một số bài toán điển hình về data mining (tt)

b. Phân lớp (classification)



I. Một số bài toán điển hình về data mining (tt)

c. Gom nhóm (Clustering)



Tham khảo thêm:

[1] Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski. Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, 1998

[2] Citeseer - Scientific Literature Digital Library. Artificial Intelligence-
<http://citeseer.nj.nec.com/ArtificialIntelligence/> - 2003

II. Luật kết hợp

1. Một số khái niệm:

- ❖ Cho $I = \{i_1, i_2, i_3, \dots, i_n\}$ là tập hợp các trường gọi là items
- ❖ D : tập các giao tác có các giao tác T_i mà $T_i \subseteq I$
- ❖ T chứa X nếu $X \subseteq T$ (X là tập có các phần tử $\subseteq I$).
- ❖ Mỗi giao tác T_i có chỉ danh là TID.
- ❖ Luật kết hợp là một mối liên hệ điều kiện giữa hai tập các hạng mục dữ liệu X và Y theo dạng sau: Nếu X thì Y , và ký hiệu là $X \Rightarrow Y$. Chúng ta có luật kết hợp $X \Rightarrow Y$, nếu $X \subset I$, $Y \subset I$ và $X \cap Y = \emptyset$
- ❖ Luật $X \Rightarrow Y$ có độ support là s nếu có $s\%$ số giao tác trong D có chứa $X \cup Y$. Hay là :

$$\text{support}(X \Rightarrow Y) = s\% = \text{Card}(X \cup Y) / \text{Card}(D) \%$$

II. Luật kết hợp (tt)

❖ Luật $X \Rightarrow Y$ có độ tin cậy là c (confidence) nếu có $c\%$ số giao tác trong D chứa $X \cup Y$ so với số giao tác trong D chứa X , khi đó ta có :

$$c = \text{Card}(X \cup Y) / \text{Card}(X) \%$$

❖ Tập các hạng mục dữ liệu gọi là ItemSet có độ support lớn hơn hay bằng giá trị ngưỡng nhỏ nhất (gọi là minsupp) được gọi là Large ItemSet. Các ItemSet còn lại được gọi là các Small ItemSet

❖ Với mỗi một Large ItemSet - L , và A là một tập con khác rỗng của L , nếu tỉ lệ phần trăm giữa support của L so với support của A lớn hơn hay bằng độ tin cậy nhỏ nhất. (gọi là minconf) thì ta có luật kết hợp $A \Rightarrow (L \setminus A)$.

II. Luật kết hợp (tt)

Ví dụ: (minh họa 2 đại lượng *minsupp* và *minconf*)

TID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	Yes	2

Người ta đưa ra $\text{minsupp} = 40\%$ và $\text{minconf} = 50\%$.

Tìm ra được 2 luật kết hợp thỏa mãn *minsupp* và *minconf*

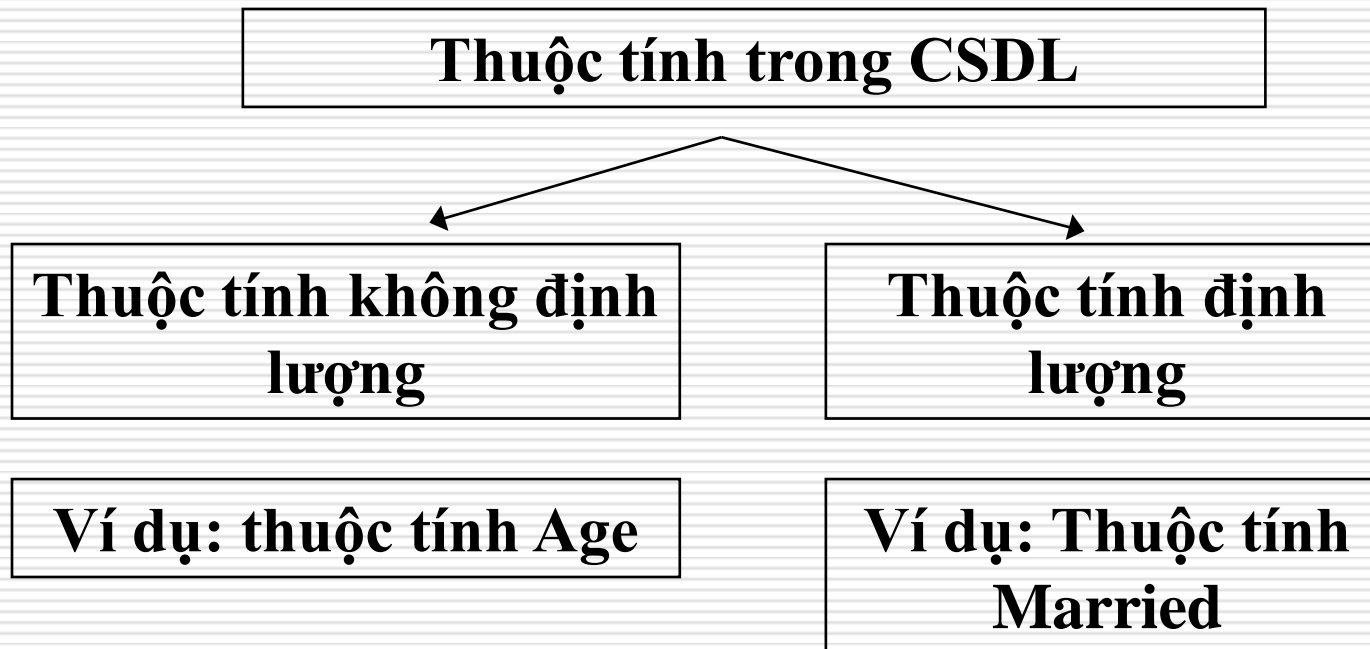
(1): Age (30..39) and (Married: Yes) \Rightarrow NumCars = 2 (s = 40%, c = 100%)

(2): NumCars(0..1) \Rightarrow Married = No (s = 40%, c = 66,6%).

II. Luật kết hợp (tt)

2. Biến đổi CSDL

❖ Phân chia giá trị của thuộc tính thành những khoảng và ứng với mỗi khoảng liên kết nó với một giá trị nguyên dương để dễ dàng thao tác trên các thuộc tính.



II. Luật kết hợp (tt)

Ví dụ: CSDL có thuộc tính Age nhận giá trị từ 20 → 50. Ta có thể chia 20 → 50 thành 4 khoảng: 10..19; 20..29; 30..39; 40..49. Xem mỗi miền này như là một thuộc tính riêng lần lượt là: 1, 2, 3, 4.

TID	Age
100	32
200	48
300	21
400	34
500	15

biến đổi thành



TID	Thuộc tính
100	3
200	4
300	2
400	3
500	1

II. Luật kết hợp (tt)

3. Tìm luật kết hợp

Để rút ra được luật trong CSDL cần tiến hành 5 bước sau:

- ❖ B1: Xác định khoảng phân chia của mỗi thuộc tính khi cần phân tích.
- ❖ B2: Kết hợp mỗi khoảng thuộc tính đã phân chia ở bước B1 với một số nguyên để thực hiện các thuật toán được nhanh, dễ dàng.
- ❖ B3: So sánh các support của các item với minsupp, tạo tập Largeitemset.
- ❖ B4: ABCD và AB là Large itemset ta rút ra được luật
 $AB \Rightarrow CD$ khi $\text{support}(ABCD)/\text{support}(AB) \geq \text{minconf}$
- ❖ B5: Xác định chọn những luật phù hợp

II. Luật kết hợp (tt)

Ví dụ: Dùng ví dụ về hồ sơ nhân sự ở trên chia khoảng trên thuộc tính AGE (giả sử chia thành 4 khoảng).

Minsupp = 40% = 2 records

Minconf = 50%

TID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	Yes	2

II. Luật kết hợp (tt)

Các khoảng chia Age

Interval
20 .. 24
25 .. 29
30 .. 34
35 .. 39



TID	Age	Married	NumCars
100	20 .. 24	No	1
200	25 .. 29	Yes	1
300	25 .. 29	No	0
400	30 .. 34	Yes	2
500	35 .. 39	Yes	2

Kết hợp thuộc tính Age và Married với một số nguyên

Interval	Integer
20 .. 24	1
25 .. 29	2
30 .. 34	3
35 .. 39	4

Value	Integer
Yes	1
No	2

II. Luật kết hợp (tt)

Bảng kết quả sau khi biến đổi

TID	Age	Married	NumCars
100	1	2	1
200	2	1	1
300	2	2	0
400	3	1	2
500	4	1	2

II. Luật kết hợp (tt)

Tập Large itemset tìm được như sau:

Itemset	Support
{(Age: 20 .. 29)}	3
{(Age: 30 .. 39)}	2
{(Married: Yes)}	3
{(Married: No)}	2
{(Numcars: 0 .. 1)}	3
{(Age: 30 .. 39), (Married: Yes)}	2

Rút ra được các luật sau:

Rule	S	C
(Age: 30..39) and (Married:Yes) \Rightarrow (Numcars: 2)	40%	100%
(Age: 20..29) \Rightarrow (Numcars: 0..1)	60%	66,6%

II. Luật kết hợp (tt)

Ví dụ: Bài toán tìm luật kết hợp

Cho CSDL sau: Tìm các luật kết hợp nếu cho
 $\text{minsupp} = 0.5(50\%)$ và $\text{minconf} = 1(100\%)$

Hóa đơn	Các mặt hàng
1	Bánh mì, nước ngọt, sữa
2	Bia, bánh mì
3	Bia, nước ngọt, khăn giấy, sữa
4	Bia, bánh mì, khăn giấy, sữa
5	Nước ngọt, khăn giấy, sữa

II. Luật kết hợp (tt)

➤ **Ta có:**

$sp(\text{“bánh mì”}) = 3/5; sp(\text{“bia”}) = 3/5; sp(\text{“nước ngọt”}) = 3/5; sp(\text{“sữa”}) = 4/5; sp(\text{“khăn giấy”}) = 3/5;$

$\Rightarrow F1 = \{\text{“bánh mì”, “bia”, “nước ngọt”, “sữa”, “khăn giấy”}\}$

$\Rightarrow C2 = \{ \{\text{“bánh mì”, “bia”}\}, \{\text{“bánh mì”, “nước ngọt”}\}, \{\text{“bánh mì”, “sữa”}\}, \{\text{“bánh mì”, “khăn giấy”}\}, \{\text{“bia”, “nước ngọt”}\}, \{\text{“bia”, “sữa”}\}, \{\text{“bia”, “khăn giấy”}\}, \{\text{“nước ngọt”, “sữa”}\}, \{\text{“nước ngọt”, “khăn giấy”}\}, \{\text{“sữa”, “khăn giấy”}\} \}$

II. Luật kết hợp (tt)

➤ Tìm F2 từ C2:

$$sp(\{\text{“bánh mì”}, \text{“bia”}\}) = 2/5 \text{ (loại)}$$

$$sp(\{\text{“bánh mì”}, \text{“nước ngọt”}\}) = 1/5 \text{ (loại)}$$

$$sp(\{\text{“bánh mì”}, \text{“sữa”}\}) = 2/5 \text{ (loại)}$$

...

$$sp(\{\text{“nước ngọt”}, \text{“sữa”}\}) = 3/5$$

...

$$sp(\{\text{“sữa”}, \text{“khăn giấy”}\}) = 3/5$$

$$\Rightarrow F2 = \{\{\text{“nước ngọt”}, \text{“sữa”}\}, \{\text{“sữa”}, \text{“khăn giấy”}\}\}$$

$$\Rightarrow C3 = \{\{\text{“nước ngọt”}, \text{“sữa”}, \text{“khăn giấy”}\}\}$$

II. Luật kết hợp (tt)

➤ **Tìm F3 từ C3:**

$sp(\{\text{“nước ngọt”}, \text{“sữa”}, \text{“khăn giấy”}\}) = 2/5$ (loại)

$\Rightarrow F3 = \{\}$

$\Rightarrow C4 = \{\}$

Vậy tập phổ biến là $\{\{\text{“nước ngọt”}, \text{“sữa”}\}\}$

➤ **Ta xây dựng 2 luật**

(R1) “nước ngọt” \rightarrow “sữa”; (R2) “sữa” \rightarrow “nước ngọt”

$conf(R1) = sp(R1)/sp(\text{“nước ngọt”}) = 3/5 : 3/5 = 1$ (100%)

$conf(R2) = sp(R1)/sp(\text{sữa}) = 3/5 : 4/5 = 3/4$ (75%) (loại)

II. Luật kết hợp (tt)

➤ Vậy tìm được 1 luật: “nước ngọt” → “sữa”

với $\text{minsupp} = 50\%$ $\text{minconf} = 100\%$

Khách hàng mua “nước ngọt” thì cũng sẽ mua “sữa”

4. Thuật toán tìm luật kết hợp

Bước 1 : Liệt kê tất cả các tập con P của I sao cho $|P| > 1$.

Bước 2 : Với mỗi tập con P, liệt kê tất cả các tập con X khác trống của P. Luật R được hình thành bởi :

$$R : X \rightarrow P \setminus X$$

❖ Thuật toán APRIORITID

(Tham khảo thêm bài giảng cao học môn học cơ sở tri thức và ứng dụng. ĐHKHTN-TPHCM của GS.TSKH Hoàng Kiếm)