



Chương 2

Tiếp Nhận VÀ BIỂU DIỄN TRI THỨC

Knowledge Collection and Presentation

- **TS. Lê Đức Như**
- Khoa Công nghệ thông tin - Trường Đại học Hải Phòng
- Phone: (+84)987394900. Email: Nhuongld@dhhp.edu.vn
- Website: www.dhhp.edu.vn/~nhuongld



Nội dung Chương 2

1. Tri thức và phân loại tri thức

2. Đặc trưng của tri thức

3. Phương pháp tiếp nhận tri thức

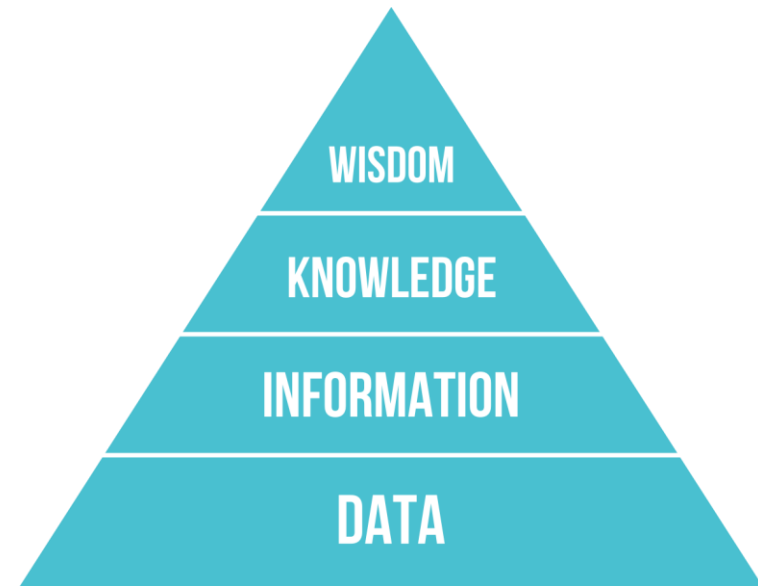
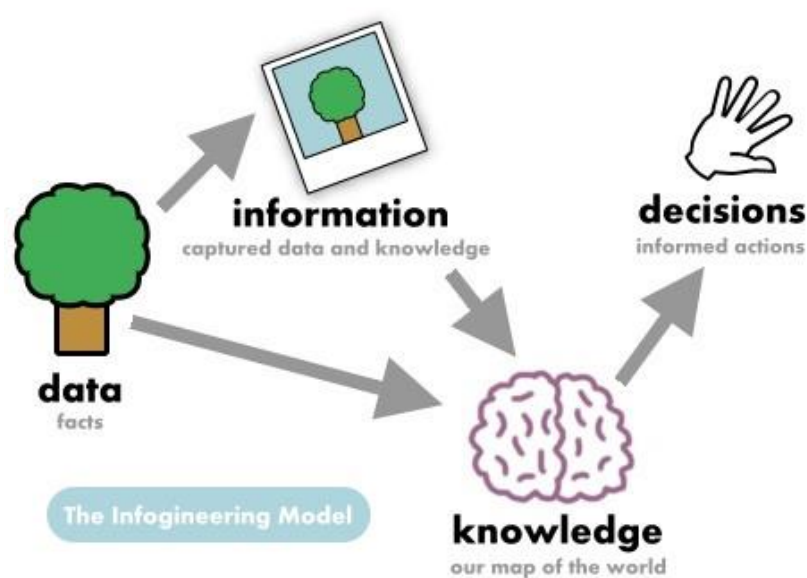
4. Phương pháp biểu diễn tri thức

5. Các vấn đề trong biểu diễn tri thức

1. Tri thức và phân loại tri thức

■ Tri thức (Knowledge)

- Tri thức là kết quả của quá trình nhận thức, học tập và lập luận.



- Máy tính: Dữ liệu (**Data**) là các con số, ký hiệu mà máy tính có thể lưu trữ, biểu diễn, xử lý. Bản thân dữ liệu không có ý nghĩa → Chỉ khi con người cảm nhận, tư duy thì dữ liệu mới có một ý nghĩa nhất định, đó chính là thông tin (**Information**)
- Tri thức (**Knowledge**) là kết tinh, cô đọng, chắt lọc của thông tin. Tri thức hình thành từ quá trình xử lý thông tin mang lại



1. Tri thức và phân loại tri thức

▪ 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.



1. Tri thức và phân loại tri thức

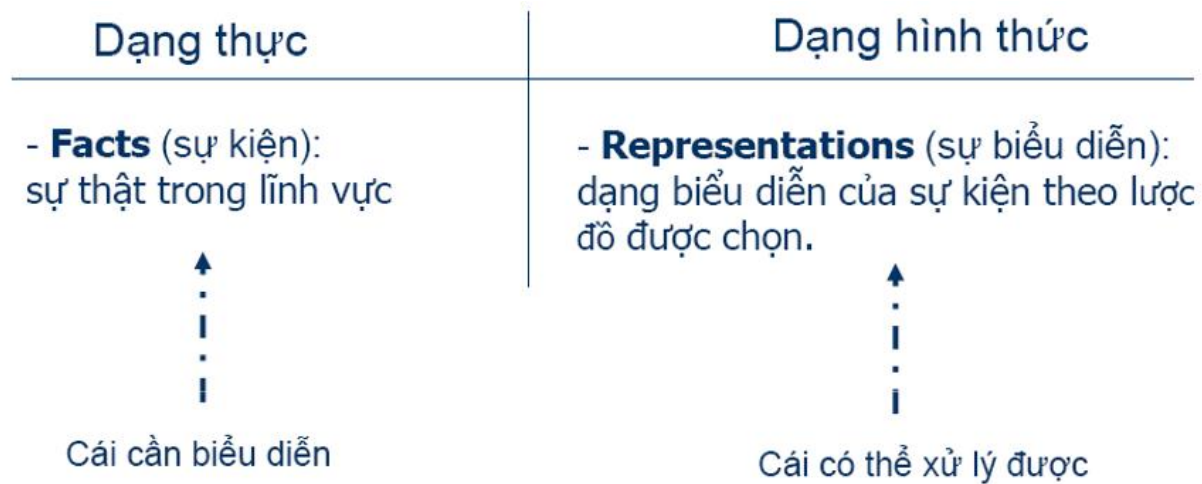
■ Phân loại tri thức

- **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, theo **quan điểm của chuyên gia** (Khái niệm, đối tượng, chức năng và mối liên hệ giữa dựa theo cấu trúc xác định).

2. Đặc trưng của tri thức

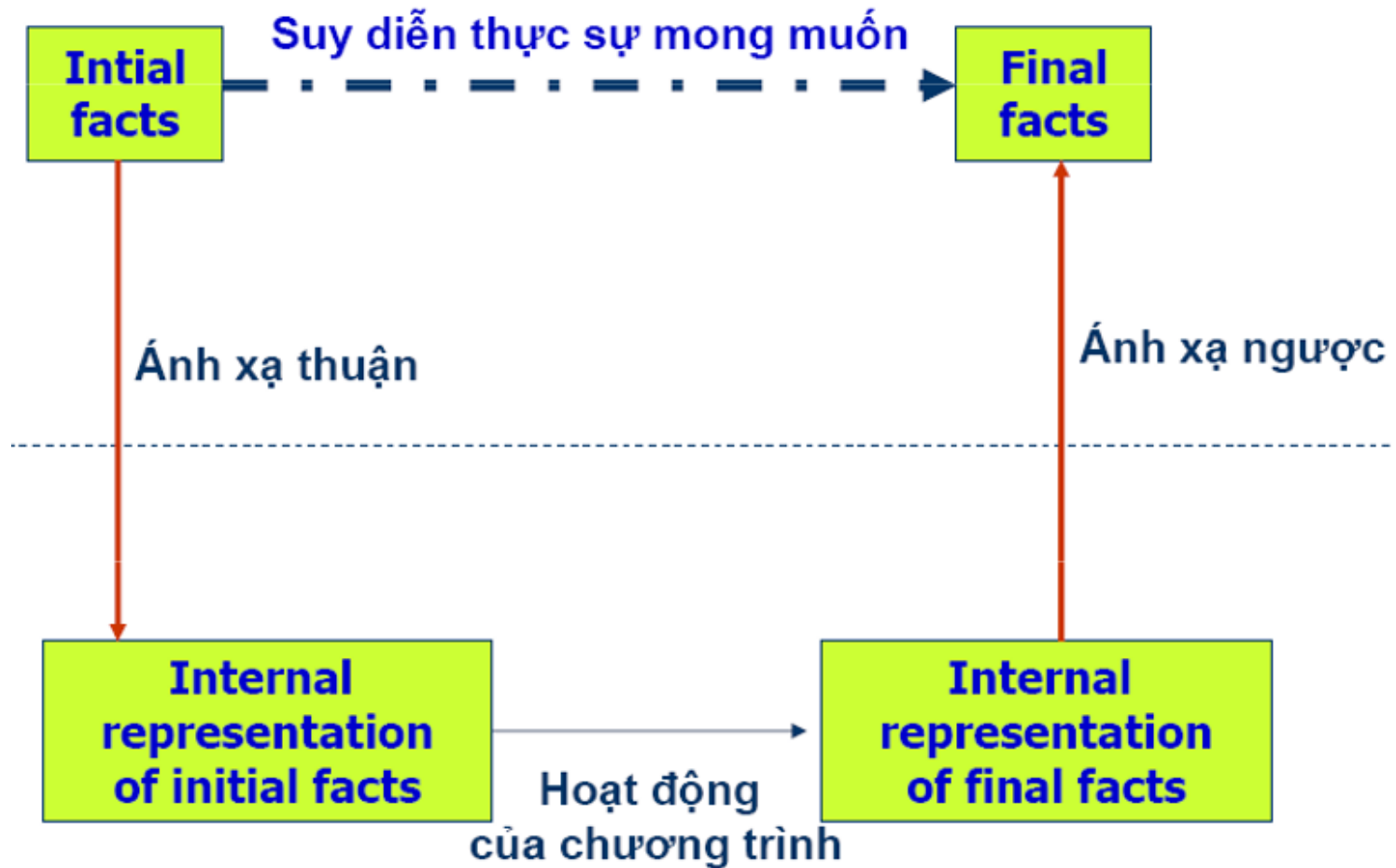
■ Hình thức hóa tri thức

- **Mức tri thức:** Mức mà các sự kiện, gồm cách hành xử của agent và mục tiêu hiện tại, được mô tả.
- **Mức ký hiệu:** Mức mà sự biểu diễn của các đối tượng đã được chọn trong mức tri thức được viết ra ở dạng ký hiệu có thể xử lý bằng chương trình.



2. Đặc trưng của tri thức

▪ Hình thức hóa tri thức



2. Đặc trưng của tri thức

▪ Nhu cầu xử lý tri thức

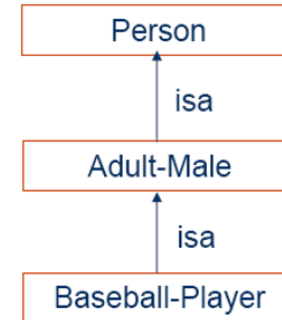
*Ngôn ngữ biểu diễn tri thức = **Cú pháp + Ngữ nghĩa + Cơ chế suy diễn***

- **Cú pháp** bao gồm các ký hiệu và các quy tắc liên kết các ký hiệu (các luật cú pháp) để tạo thành các câu (công thức) trong ngôn ngữ
- **Ngữ nghĩa** xác định ý nghĩa của các câu trong một miền nào đó của thế giới hiện thực
- **Cơ chế suy diễn** để từ các tri thức trong cơ sở tri thức và các sự kiện ta nhận được các tri thức mới

2. Đặc trưng của tri thức

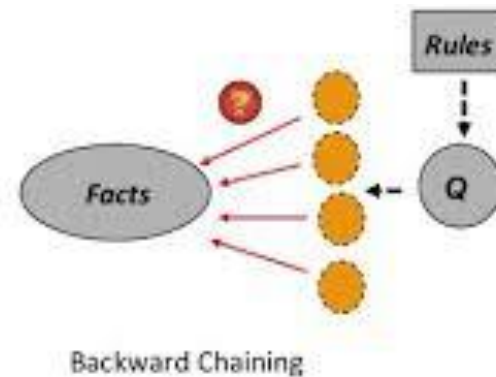
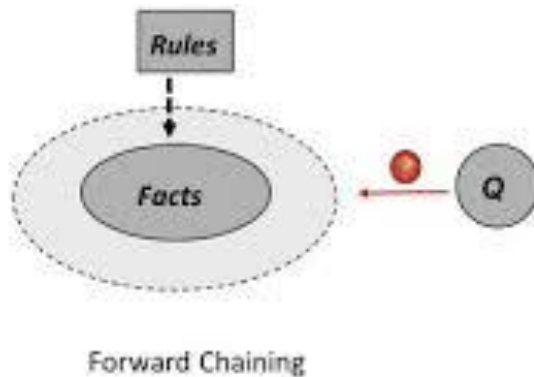
■ Tri thức có khả năng thừa kế

- Thừa kế thuộc tính, tổ chức thành các lớp



■ Tri thức có khả năng suy diễn

- Forward Chaining** (suy diễn tiến): Sự kiện → Kết luận
- Backward Chaining** (suy diễn lùi): Kết luận → Sự kiện đã cho



2. Đặc trưng của tri thức

▪ Forward Chaining (Suy diễn tiến)

▪ Ví dụ:

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

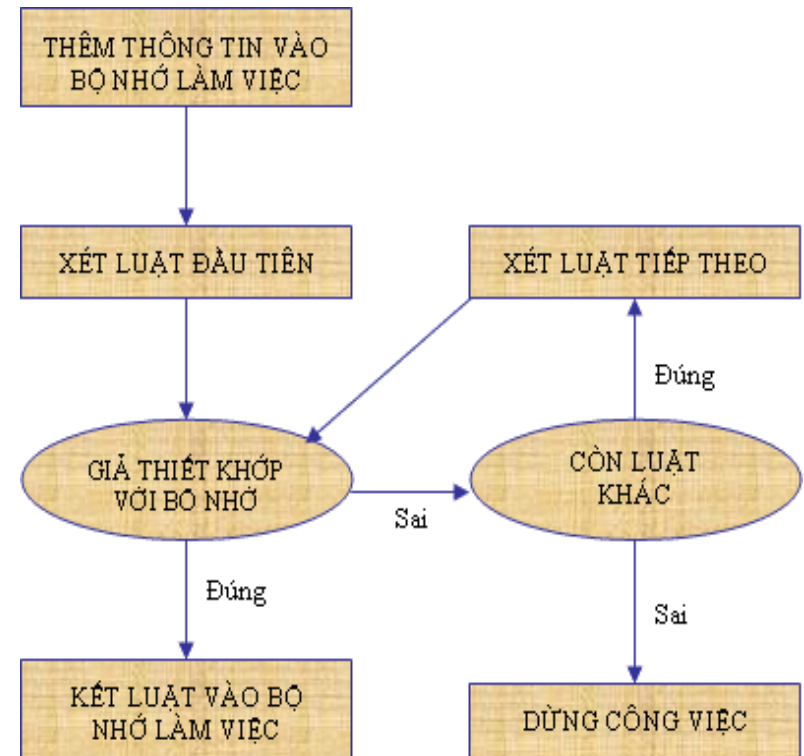
Luật 2. IF Nhiệt độ bệnh nhân quá 37 độ
THEN Bệnh nhân bị sốt

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.



2. Đặc trưng của tri thức

▪ Forward Chaining (Suy diễn tiến)

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);

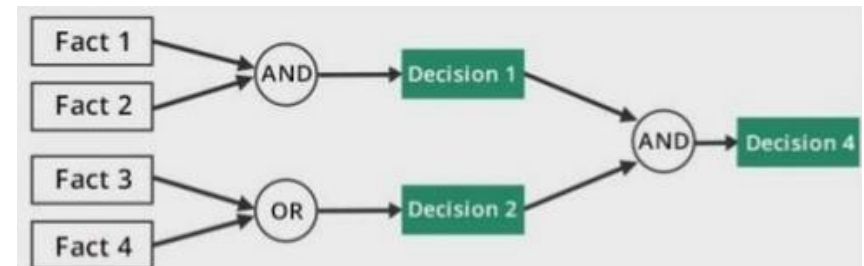
}

If KL \subseteq TD THEN Return “True”

else Return “False”

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

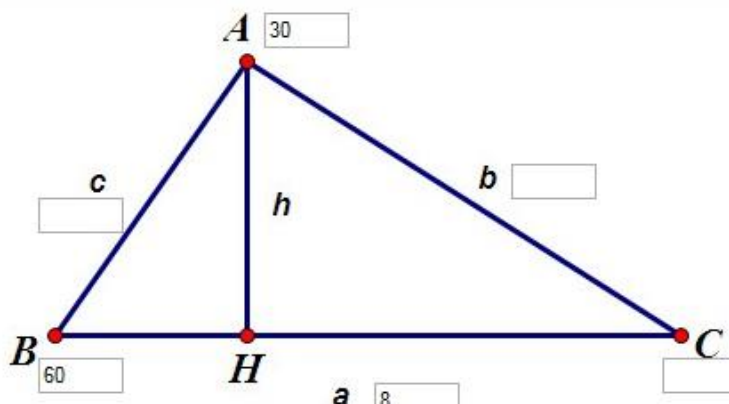
Hỏi $a \wedge b \rightarrow g$? $a \rightarrow g$?



2. Đặc trưng của tri thức

Forward Chaining (Suy diễn tiến)

BÀI TOÁN TAM GIÁC



Các thông số khác

ha	<input type="text"/>	hc	<input type="text"/>	S	<input type="text"/>
hb	<input type="text"/>	p	<input type="text"/>		

Kết luận

Thông số cần tính: S

☒ Suy diễn tiến
 ☐ Suy diễn lùi

Bài làm:

* Bước 1: Tính góc C
- Áp dụng công thức: $C = 180 - A - B$
Ta tính được: $C = 90$

* Bước 2: Tính độ dài cạnh b
- Áp dụng công thức: $b = a * \sin B / \sin A$
Ta tính được: $b = 13,86$

* Bước 3: Tính độ dài cạnh c
- Áp dụng công thức: $c = a * \sin C / \sin A$
Ta tính được: $c = 16$

* Bước 4: Tính độ dài nửa chu vi p
- Áp dụng công thức: $p = (a + b + c) / 2$
Ta tính được: $p = 18,93$

* Bước 5: Tính diện tích S
- Áp dụng công thức: $S = \sqrt{p * (p - a) * (p - b) * (p - c)}$
Ta tính được: $S = 55,44$

Kết luận: Vậy diện tích S của tam giác cần tính là 55,44.

Xử lý

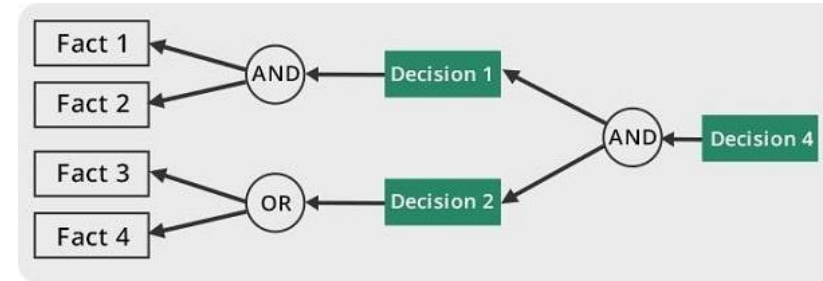
2. Đặc trưng của tri thức

Backward Chaining (suy diễn lùi)

```

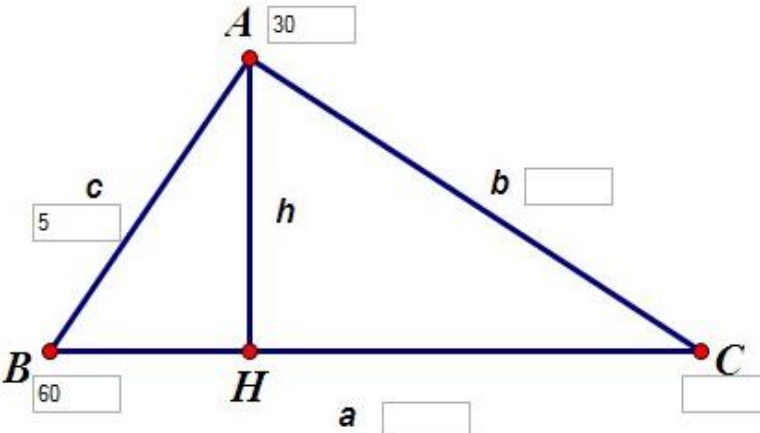
If  $KL \subseteq GT$  THEN Return "True"
Else {  $TĐích = \emptyset$ ;  $Vết = \emptyset$ ;  $First = 1$ ;  $Quaylui = False$ ; }
For Each  $q \in KL$  DO  $TĐích = TĐích \cup \{(q, 0)\}$ ;
Repeat
    first ++;     $((f, i) = Get(TĐích))$ ;
    If  $(f \notin GT)$  THEN {
         $j = Timluật(f, i, Rule)$ ; //  $r_j: Left_j \rightarrow f$ 
        If (Tìm có  $r_j$ ) THEN
            {  $Vet = Vet \cup \{(f, j)\}$ ;
              For Each  $t \in (Left_j \setminus GT)$  DO  $TĐích = TĐích \cup \{(t, 0)\}$ ;
            }
        else
            {  $Quaylui = True$ ;
              While  $(f \notin KL)$  AND  $Quaylui$  DO {
                  Repeat {  $(g, k) = Get(Vết)$ ;
                            $TĐích = TĐích \setminus Left_k$ ;
                           Until  $f \in Left_k$ ;
                            $l = Timluật(g, k, Rule)$ ;
                           If (Tìm có  $r_l$ ) THEN
                               {  $TĐích = TĐích \setminus Left_l$ ;
                                 For Each  $t \in (Left_l \setminus GT)$  DO  $TĐích = TĐích \cup \{(t, 0)\}$ ;
                                  $Vết = Vết \cup \{(g, l)\}$ ;  $Quaylui = False$ ; }
                               else  $f = g$ ; }
                          }
            }
    }
Until  $(TĐích = \emptyset)$  OR  $((f \in KL)$  and  $(First > 2))$ ;
If  $(f \in KL)$  then Return False else Return True;

```



2. Đặc trưng của tri thức

Backward Chaining (suy diễn lùi)



Các thông số khác

ha hc S

hb p

Kết luận

Thông số cần tính hc

☐ Suy diễn tiến ☒ Suy diễn lùi

Xử lý

File About

* Quá trình suy diễn lùi:

- Để tính được hc, ta dùng công thức:

$$hc = a * \sin B \quad (1)$$
- + Để suy ra được (1) ta cần đi tìm a:
- Để tính được a, ta dùng công thức:

$$a = c * \sin A / \sin C \quad (2)$$
- + Để suy ra được (2) ta cần đi tìm C:
- Để tính được C, ta dùng công thức:

$$C = 180 - A - B \quad (3)$$

--> Áp dụng (3) ta tìm được giá trị C cần tìm sau đó thay ngược trở lại để tìm được kết quả của bài toán.

+ Từ (3): $C = 90$

+ Từ (2): $a = 2,5$

+ Từ (1): $hc = 2,17$

Kết luận: Vậy độ dài của đường cao hc cần tính là 2,17.

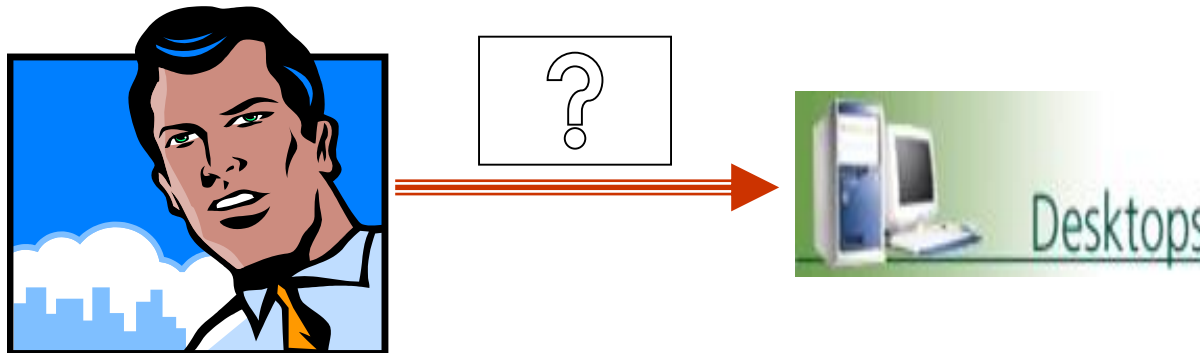
3. Phương pháp tiếp nhận tri thức

▪ Thụ động

- **Gián tiếp:** Những tri thức kinh điển (Định lý, định nghĩa,...)
- **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.





4. Phương pháp biểu diễn tri thức

- 4.1. Logic mệnh đề & logic vị từ
- 4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value OAV)
- 4.3 Tri thức luật dẫn
- 4.4 Mạng ngữ nghĩa
- 4.5 Frame
- 4.6 Đồ thị khái niệm
- 4.7 Script

4.1 Logic mệnh đề & logic vị từ

- Là dạng **biểu diễn tri thức cổ điển** nhất trong máy tính với 2 dạng là logic mệnh đề và logic vị từ.
- Dùng kí hiệu để biểu diễn tri thức và các toán tử áp lên các ký hiệu để suy luận logic. Là công cụ hình thức để biểu diễn và suy luận tri thức.

Phép toán	AND	OR	NOT	Kéo theo	Tương đương
Kí hiệu	$\wedge, \&, \cap$	$\vee, \cup, +$	\neg, \sim	\supset, \rightarrow	\equiv

Các phép toán logic và các ký hiệu sử dụng

4.1 Logic mệnh đề & logic vị từ

4.1.1 Logic mệnh đề

▪ Ví dụ 1:

IF Xe không khởi động được $\rightarrow A$

AND Khoảng cách từ nhà đến chỗ làm là xa $\rightarrow B$

THEN Sẽ trễ giờ làm $\rightarrow C$

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

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \equiv B$
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

Các phép toán quen thuộc trên các mệnh đề

4.1 Logic mệnh đề & logic vị từ

4.1.1 Logic vị từ

- **Mệnh đề:** Thì *không có cấu trúc* \rightarrow hạn chế nhiều thao tác suy luận \rightarrow đưa vào khái niệm vị từ và lượng từ (\forall - **với mọi**, \exists - **tồn tại**) để tăng cường tính cấu trúc của một mệnh đề.
- 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 \Rightarrow Vị (cam, ngọt)
Cam có màu xanh \Rightarrow Màu(cam, xanh)

4.1 Logic mệnh đề & logic vị từ

4.1.1 Logic vị từ

- Ví dụ 2: Tri thức “*A là bố của B nếu B là anh hoặc em của một người con của A*” được biểu diễn dưới dạng vị từ như thế nào?

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

- Khi đó: mệnh đề **Bố**(A,B) là một mệnh đề tổng quát và ta có các mệnh đề cơ sở là :
 - **a)** **Bố** (“An”, “Bình”) có giá trị đúng (An là bố của Bình)
 - **b)** **Anh**(“Tú”, “Bình”) có giá trị đúng (Tú là anh của Bình) thì mệnh đề
 - **c)** **Bố** (“An”, “Tú”) sẽ có giá trị là đúng (An là bố của Tú).

4.1 Logic mệnh đề & logic vị từ

4.1.1 Logic vị từ

- Ví dụ 3: Câu “*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 thế nào?

$$\text{LớnHơn}(x, y) = x > y \text{ và } \text{NhỏHơn}(x, y) = x < y$$

Kết quả: $\forall x, \exists y : \text{LớnHơn}(y, x)$ và $\forall x, \exists y : \text{NhỏHơn}(y, x)$

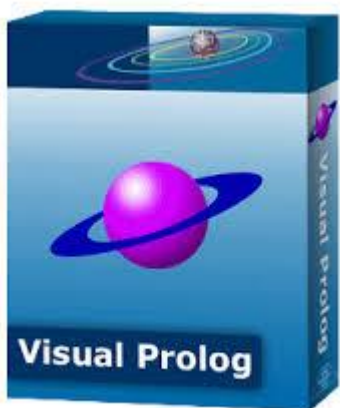
- Ví dụ 4: Câu châm ngôn “*Gần mực thì đen, gần đèn thì sáng*” được hiểu là “*chơi với bạn xấu nào thì ta cũng sẽ thành người xấu*” có thể được biểu diễn bằng vị từ như sau :

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

4.1 Logic mệnh đề & logic vị từ

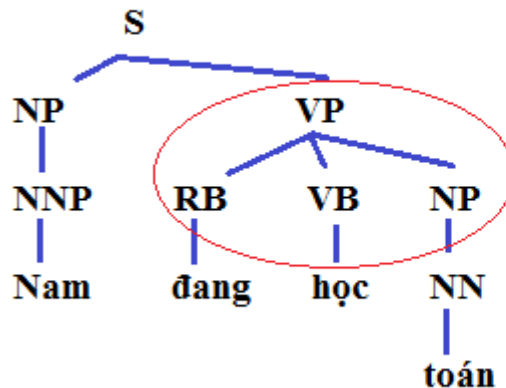
4.1.1 Logic vị từ

Ngôn ngữ PROLOG



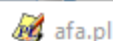
Ký hiệu	Ý nghĩa sử dụng
S	Câu
NP	Danh ngữ
VP	Ngữ động từ
AVDP	Trạng ngữ
ADJP	Tính ngữ (RB, JJ)
PP	Giới ngữ (IN, NN)
NN	Danh từ chung
NNP(Nproper)	Danh từ riêng
VB	Động từ (làm, có)
RB hoặc ADV	Trạng từ (lại, chỉ)
Adj hoặc JJ	Tính từ chỉ phẩm chất ("mới")
CD	Số lượng ("một")
UN hoặc Q	Danh từ chỉ đơn vị/ Lượng từ (cuốn, quyển, cái, con, căn, cả, những, chút)
PPR hoặc PRP	Đại từ nhân xưng / Đại danh từ
IN	Giới từ
C	Liên từ

Cây cú pháp:



CFG:

S → NP VP
NP → NNP
NP → NN
VP → RB VB NP
NNP → Nam
NN → toán
RB → đang
VB → học



```

File Edit Browse Compile
afa.pl
s--> np, vp.
np-->nnp.
np-->nn.
vp-->rb, vb, np.
nnp-->['Nam'].
nn-->[toan].
rb-->[dang].
vb-->[hoc].
    
```



SWI-Prolog -- c:/Users/phams/Desktop/afa.pl

```

File Edit Settings Run Debug Help
1 ?- s(['Nam',dang,hoc,toan],[ ]).
true.
2 ?- vp([dang,hoc,toan],[ ]).
true.
3 ?- nnp([Nam],[ ]).
Nam = 'Nam'.
4 ?- nnp([Nam],[ ]).
Nam = 'Nam'.
5 ?- nnp(['Nam'],[ ]).
true.
6 ?-
    
```


4.1 Logic mệnh đề & logic vị từ

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.

4.1 Logic mệnh đề & logic vị từ

4.1.3 Một số thuật giải liên quan đến logic mệnh đề:

Luật	Điều kiện	Kết luận
Modus Ponens	$\alpha \Rightarrow \beta, \alpha$	β
Modus Tollens	$\alpha \Rightarrow \beta, \neg\beta$	$\neg\alpha$
Bắc cầu	$\alpha \Rightarrow \beta, \beta \Rightarrow \gamma$	$\alpha \Rightarrow \gamma$
Loại bỏ hội	$\alpha_1 \wedge \dots \wedge \alpha_i \wedge \dots \wedge \alpha_m$	α_i
Đưa vào hội	$\alpha_1, \dots, \alpha_i, \dots, \alpha_m$	$\alpha_1 \wedge \dots \wedge \alpha_i \wedge \dots \wedge \alpha_m$
Đưa vào tuyển	α_i	$\alpha_1 \vee \dots \vee \alpha_i \vee \dots \vee \alpha_m$
Phân giải	$\alpha \vee \beta, \neg\beta \vee \gamma$	$\alpha \vee \gamma$



4.1 Logic mệnh đề & logic vị từ

4.1.3 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 ($a \rightarrow b$).
- Phương pháp lập bảng chân trị luôn cho được kết quả cuối cùng nhưng độ phức tạp quá lớn: $O(2^n)$ với n là số biến mệnh đề.
- Thuật giải Vương Hạo và Thuật giải Robinson có độ phức tạp $O(n)$

4.1 Logic mệnh đề & logic vị từ

Thuật giải Vương Hạo:

- **Bước 1:** Phát biểu lại giả thiết và kết luận của vấn đề theo dạng chuẩn sau :

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

Trong đó: GT_i và KL_i là các mệnh đề gồm các biến và 3 phép nối cơ bản: \wedge, \vee, \neg

- **Bước 2:** Chuyển về các GT_i và KL_i có dạng phủ định.
- **Bước 3:** Nếu ở GT_i có phép \wedge thì thay thế phép \wedge bằng dấu “,”
Nếu ở KL_i có phép \vee thì thay thế phép \vee bằng dấu “,”
- **Bước 4:** Nếu ở GT_i có chứa phép \vee thì tách thành hai dòng con.
Nếu ở KL_i có chứa phép \wedge thì tách thành hai dòng con.
- **Bước 5:** Một dòng được chứng minh nếu tồn tại chung một mệnh đề ở ở cả hai phía.
- **Bước 6:**
 - a) Nếu một dòng không còn phép nối \wedge hoặc \vee ở cả hai vế và ở 2 vế không có chung một biến mệnh đề thì dòng đó không được chứng minh.
 - b) Một vấn đề được chứng minh nếu tất cả dòng dẫn xuất từ dạng chuẩn ban đầu đều được chứng minh.

4.1 Logic mệnh đề & logic vị từ

Thuật giải Vương Hạo:

Ví dụ: Chứng minh: $p \rightarrow q, q \rightarrow r \Rightarrow p \rightarrow r$

Bước 1: $\neg p \vee q, \neg q \vee r \Rightarrow \neg p \vee r$

Bước 3: $\neg p \vee q, \neg q \vee r \Rightarrow \neg p, r$

$\neg p \vee q, \neg q \vee r, p \rightarrow r$

Bước 4: Tách mệnh đề đầu $\neg p, \neg q \vee r, p \rightarrow r$ (1) và $q, \neg q \vee r, p \rightarrow r$ (2)

Từ (1): $\neg q \vee r, p \rightarrow r, p$: được chứng minh

Từ (2): $q, \neg q \vee r, p \rightarrow r$ tách thành

$q, \neg q, p \rightarrow r$ (2.1)

$q, r, p \rightarrow r$ (2.2): được chứng minh

Từ (2.1): $q, p \rightarrow r, q$: được chứng minh

Kết luận: $p \rightarrow q, q \rightarrow r \Rightarrow p \rightarrow r$



4.1 Logic mệnh đề & logic vị từ

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.

4.1 Logic mệnh đề & logic vị từ

Thuật giải Robinson:

- **Bước 1:** Phát biểu lại giả thiết và kết luận của vấn đề dưới dạng chuẩn như sau :

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

Trong đó: GT_i và KL_j được xây dựng từ các biến mệnh đề và các phép toán : \wedge, \vee, \neg

- **Bước 2:** Nếu ở GT_i có phép \wedge thì thay thế phép \wedge bằng dấu “,”
Nếu ở KL_j có phép \vee thì thay thế phép \vee bằng dấu “,”
- **Bước 3:** Biến đổi dòng chuẩn ở **Bước 1** về thành danh sách mệnh đề như sau :
$$\{ GT_1, GT_2, \dots, GT_n, \neg KL_1, \neg KL_2, \dots, \neg KL_m \}$$
- **Bước 4:** Nếu trong danh sách mệnh đề ở Bước 2 có 2 mệnh đề đối ngẫu nhau (a và $\neg a$) thì bài toán được chứng minh. Ngược lại thì chuyển sang Bước 4.

4.1 Logic mệnh đề & logic vị từ

Thuật giải Robinson:

- **Bước 5:** 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.
- **Bước 6:** Thay thế hai mệnh đề vừa tuyển trong danh sách mệnh đề bằng mệnh đề mới.
- **Bước 7:** 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.

4.1 Logic mệnh đề & logic vị từ

Thuật giải Robinson:

- Ví dụ: Chứng minh $(\neg p \vee q) \wedge (\neg q \vee r) \wedge (\neg r \vee s) \wedge (\neg u \vee \neg s) \rightarrow \neg p \vee \neg u$
- **Bước 3:** $\{ \neg p \vee q, \neg q \vee r, \neg r \vee s, \neg u \vee \neg s, \mathbf{p}, \mathbf{u} \}$
- **Bước 4 :** Có tất cả 6 mệnh đề nhưng chưa có mệnh đề nào đối ngẫu nhau.
- **Bước 5 :**
 - Tuyển một cặp mệnh đề (chọn hai mệnh đề có biến đối ngẫu).
 - Chọn hai mệnh đề đầu : $\neg p \vee \mathbf{q} \vee \neg \mathbf{q} \vee r \Rightarrow \neg p \vee r$. Danh sách mệnh đề thành : $\{ \neg p \vee r, \neg r \vee s, \neg u \vee \neg s, p, u \} \rightarrow$ Vẫn chưa có mệnh đề đối ngẫu.
 - Tuyển hai cặp mệnh đề đầu tiên: $\neg p \vee \mathbf{r} \vee \neg \mathbf{r} \vee s \Rightarrow \neg p \vee s$. Danh sách mệnh đề thành $\{ \neg p \vee s, \neg u \vee \neg s, p, u \}$ vẫn chưa có hai mệnh đề đối ngẫu.
 - Tuyển hai cặp mệnh đề đầu tiên: $\neg p \vee \mathbf{s} \vee \neg u \vee \neg \mathbf{s} \Rightarrow \neg p \vee \neg u$. Danh sách mệnh đề thành: $\{ \neg p \vee \neg u, p, u \}$ vẫn chưa có hai mệnh đề đối ngẫu
 - Tuyển hai cặp mệnh đề : $\neg p \vee \neg u \vee u \Rightarrow \neg p$. Danh sách mệnh đề trở thành : $\{ \neg p, p \}$. Có hai mệnh đề đối ngẫu nên biểu thức ban đầu đã được chứng minh.



4.1 Logic mệnh đề & logic vị từ

Bài tập:

1. $p \wedge (\neg p \vee q) \rightarrow q$
2. $(p \vee q) \wedge (\neg p \vee r) \rightarrow q \vee r$
3. $(a \wedge b) \rightarrow c, (b \wedge c) \rightarrow d, \neg d$. Chứng minh: $a \rightarrow b$
4. $\{p \rightarrow q, q \rightarrow r, r \rightarrow s, p\} \Rightarrow p \wedge s$

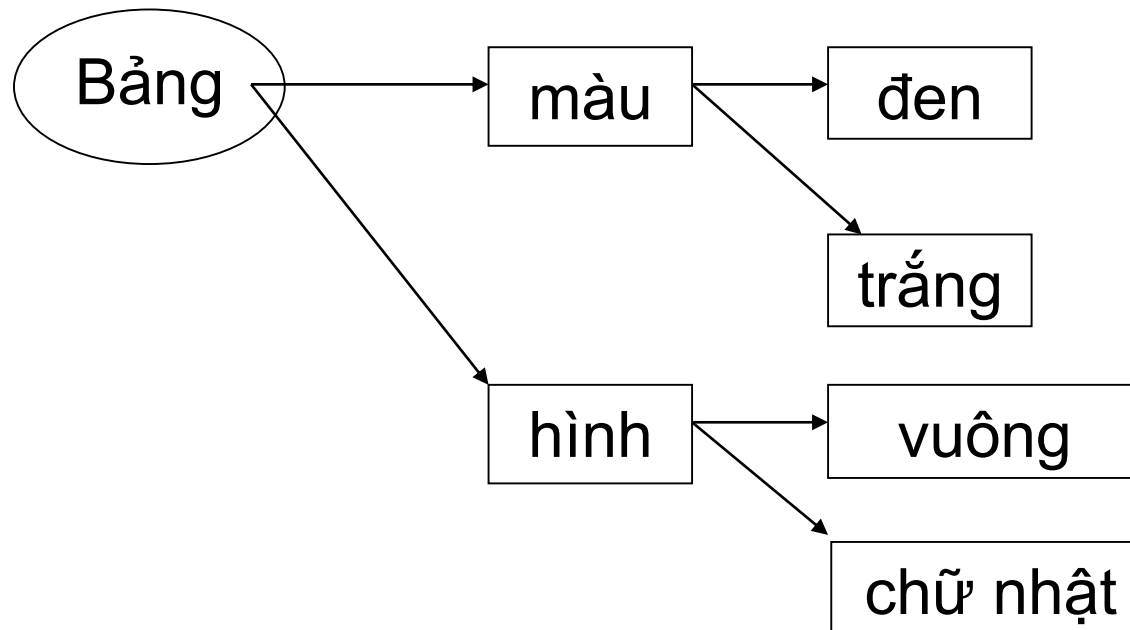


4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value)

- Sự kiện gồm **Object-Attribute-Value** được dùng để xác nhận giá trị của một thuộc tính xác định của một vài đối tượng.
 - Ví dụ: "quả bóng màu đỏ" xác nhận "đỏ" là giá trị thuộc tính "màu" của đối tượng "quả bóng".
- Một đối tượng có thể có nhiều thuộc tính với các kiểu giá trị khác nhau. Một thuộc tính cũng có thể có một (**single-valued**) hay nhiều giá trị (**multi-valued**) → **Linh động trong 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. Khi đó, trong sự kiện O-A-V sẽ có thêm một giá trị xác định **độ tin cậy của nó là CF (Certainly Factor)**.

4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value)

■ Ví dụ:





4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value)

- Thiết lập một tập các phần tử cơ bản để có thể biểu diễn cấu trúc ngữ nghĩa của các biểu thức ở ngôn ngữ tự nhiên theo một cách đồng nhất
- Bốn khái niệm cơ bản trong lược đồ phụ thuộc khái niệm:
 - **ACT** (**Action**): Các hành động (các động từ trong câu)
 - **PP** (**Picture Producers**): Các đối tượng (các chủ từ, tân ngữ,...)
 - **AA** (**Action Adder**): Bổ nghĩa cho hành động (trạng từ)
 - **PA** (**Picture Adder**): Bổ nghĩa cho đối tượng (tính từ)

4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value)

- Các hành động có thể được mô tả bằng cách phân rã về một/ nhiều hành động (ACT):
 - 1. **ATRANS**: chuyển đổi một quan hệ
 - VD: động từ: cho, biểu,...
 - 2. **PTRANS**: chuyển đổi vị trí vật lý
 - VD: đi, chạy, di chuyển,..
 - 3. **PROPEL**: tác động một lực vật lý lên đối tượng
 - VD: đẩy, chài,...
 - 4. **MOVE**: di chuyển một phần thân thể bởi đối tượng
 - VD: đá..
 - 5. **GRASP**: nắm lấy đối tượng khác.
 - VD: cầm, nắm, giữ,...
 - 6. **INGEST**: ăn vào bụng một đối tượng bởi đt khác
 - VD: ăn, nuốt,..
 - 7. **EXPEL**: tống ra từ thân thể của một đối tượng
 - VD: khóc,..
 - 8. **MTRANS**: chuyển đổi thông tin tinh thần
 - VD: nói, tiết lộ,..
 - 9. **MBUILD**: tạo ra một thông tin tinh thần mới
 - VD: quyết định, ...
 - 10. **CONC**: nghĩ về một ý kiến
 - VD: suy nghĩ, hình dung,...
 - 11. **SPEAK**: tạo ra âm thanh
 - VD: nói, phát biểu,...
 - 12. **ATTEND**: tập trung giác quan
 - VD: lắng nghe, nhìn,...

4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value)

▪ Lưu đồ về quan hệ phụ thuộc khái niệm:

PP \longleftrightarrow **ACT** Đối tượng PP
thực hiện hành động ACT

PP \longleftrightarrow **PA** Đối tượng PP có thuộc tính PA

ACT \xleftarrow{O} **PP** Hành động ACT tác động lên PP

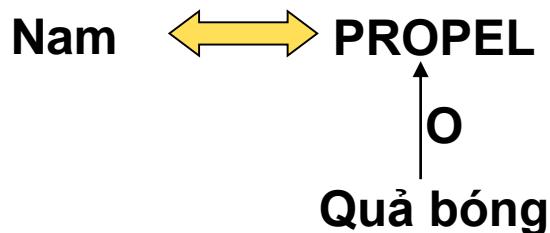
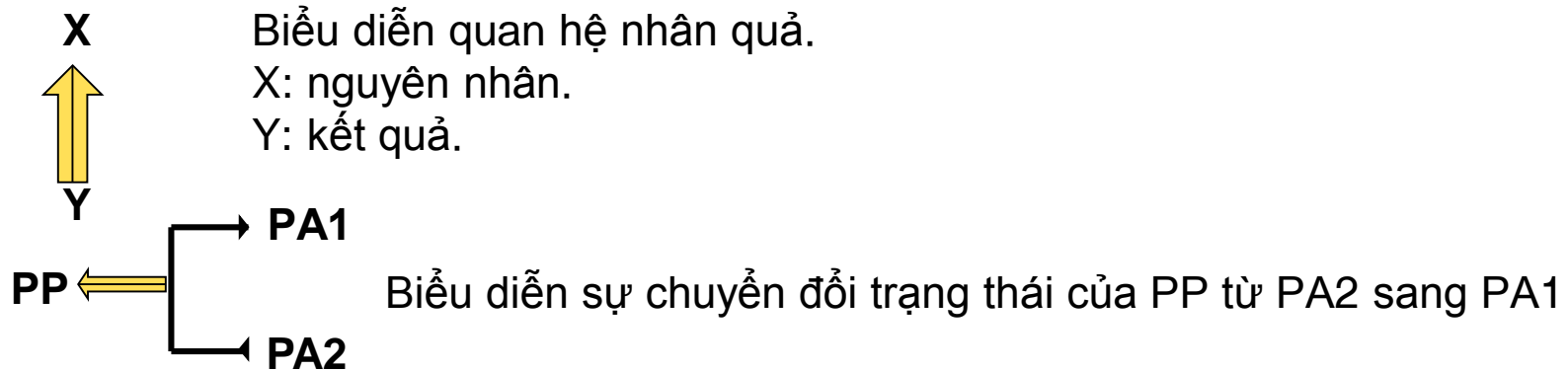
ACT \xleftarrow{R} $\begin{cases} \text{PP} \\ \text{PP} \end{cases}$ Đối tượng nhận và cho
trong hành động ACT
R: đối tượng nhận (recipient)

ACT \xleftarrow{D} $\begin{cases} \text{PP} \\ \text{PP} \end{cases}$ Hướng của đối tượng
trong hành động ACT
D: Hướng(Direction)

ACT \xleftarrow{I} \updownarrow Quan hệ giữa hành động và
thiết bị phục vụ cho hành động

4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value)

▪ Lưu đồ về quan hệ phụ thuộc khái niệm:



Ý nghĩa: “Nam tác dụng một lực vào quả bóng”



4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value)

▪ Lưu đồ về quan hệ phụ thuộc khái niệm:

f : tương lai.

t : chuyển tiếp.

ts : bắt đầu chuyển tiếp.

tf : kết thúc chuyển tiếp.

k : đang diễn ra.

? : nghi vấn.

/ : phủ định.

C : điều kiện.

Nil: hiện tại. (không ghi chú gì)



Ý nghĩa: Nam **đã** tác dụng một lực (đẩy) vào cái bàn

4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value)

▪ Lưu đồ về quan hệ phụ thuộc khái niệm:

PP \longleftrightarrow ACT VD: Nam \xrightarrow{P} PROPEL \xleftarrow{O} Lan : Nam đã đánh Lan

Nam \xrightarrow{K} ATTEND \xleftarrow{O} Bài giảng : Nam đang tập trung vào bài giảng.

PP \longleftrightarrow PA Nam \longleftrightarrow Height (> average) : Nam is tall.

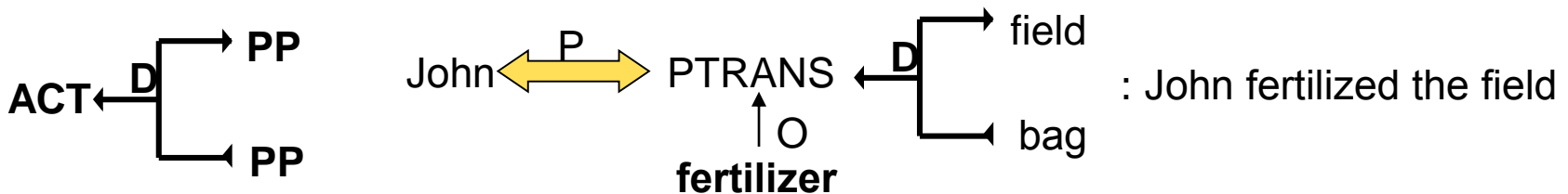
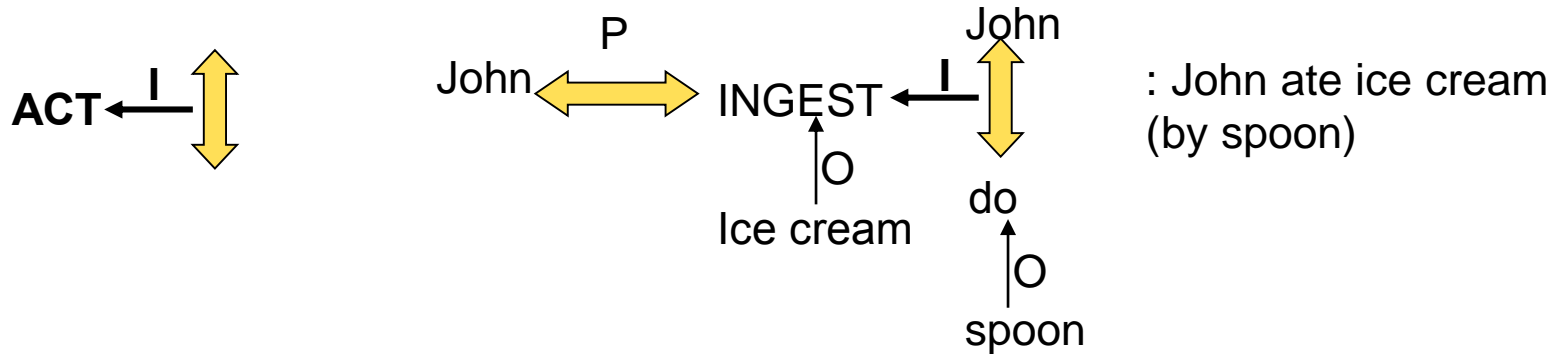
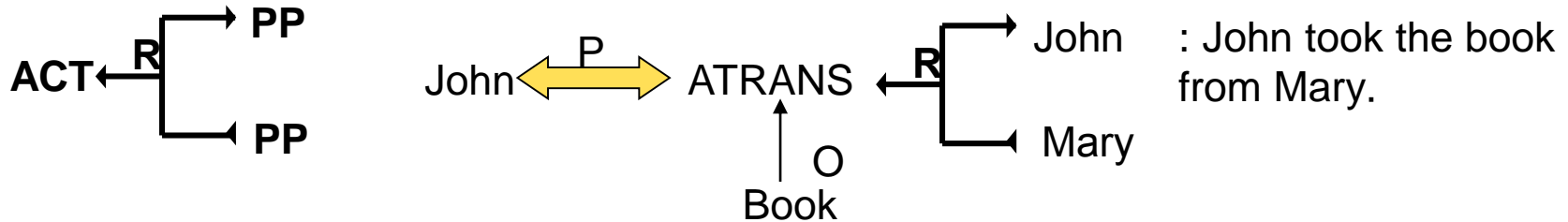
PP \longleftrightarrow PP Nam \longleftrightarrow doctor : Nam is a doctor.

PP $\xleftarrow{\quad}$ PA boy $\xleftarrow{\quad}$ nice : A nice boy

PP $\xleftarrow{\quad}$ PP dog $\xleftarrow{\quad}$ John : John's dog.

Một số ví dụ

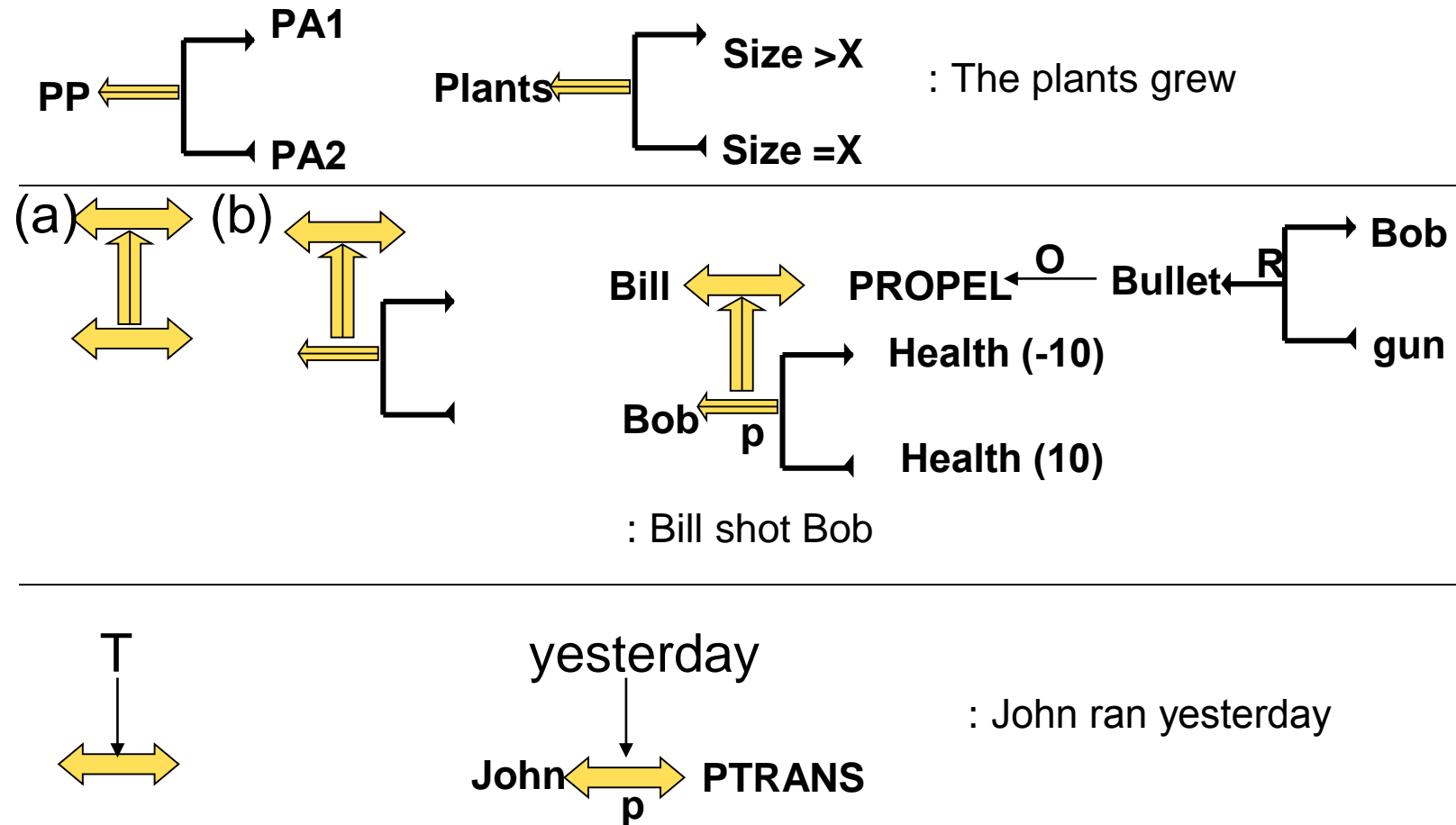
- Lưu đồ về quan hệ phụ thuộc khái niệm:



Một số ví dụ

4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value)

- Lưu đồ về quan hệ phụ thuộc khái niệm:

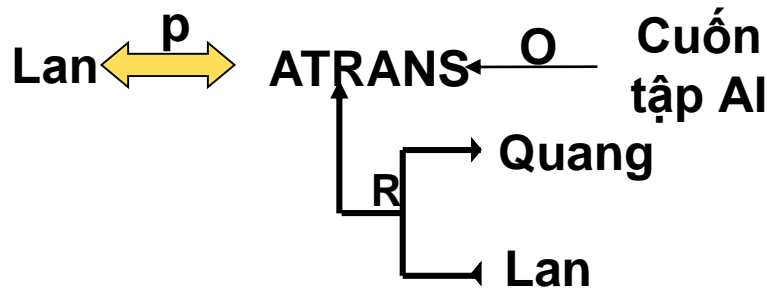


Một số ví dụ

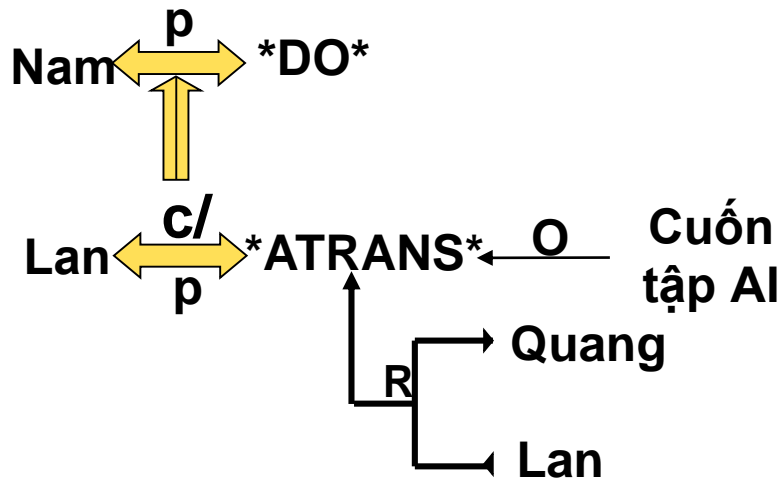
4.2 Đối tượng-thuộc tính-giá trị (object-attribute-value)

- Lưu đồ về quan hệ phụ thuộc khái niệm:

Lan gởi cuốn tập cho Quang



“Nam đã cấm Lan gởi cuốn tập AI cho Quang”





4.3 Tri thức luật dẫn

- Newell và Simon đề xuất khi xây dựng một hệ giải bài toán tổng quát. 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



4.3 Tri thức luật dẫn

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

4.3 Tri thức luật dẫn

Các dạng luật cơ bản: 7 dạng

▪ 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

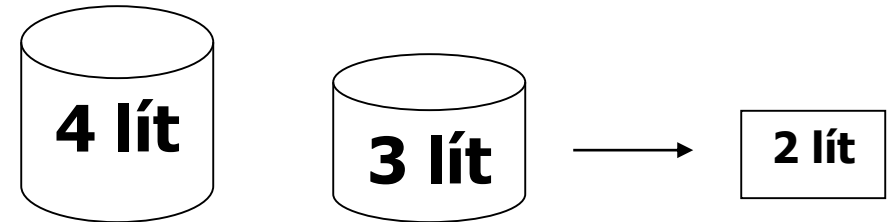
THEN Nên chọn Xe Spacy **AND** Chọn màu sáng

4.3 Tri thức luật dẫn

- **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 đo được chính xác 2 lít nước?

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;  
END;  
...
```



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

4.3 Tri thức luật dẫn

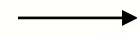


- Bài toán được biểu diễn ở dạng không gian trạng thái bằng luật:

(x, y : lần lượt là số lít nước hiện có trong bình 4 lít và 3 lít)

1. Nếu ($x < 4$) thì (x, y) \rightarrow ($4, y$) Đổ đầy bình 4 lít
2. Nếu ($y < 3$) thì (x, y) \rightarrow ($x, 3$) Đổ đầy bình 3 lít
3. Nếu ($x > 0$) thì (x, y) \rightarrow ($x-d, y$) Đổ d lít ra khỏi bình 4 lít
4. Nếu ($y > 0$) thì (x, y) \rightarrow ($x, y-d$) Đổ d lít ra khỏi bình 3 lít
5. Nếu ($x > 0$) thì (x, y) \rightarrow ($0, y$) Đổ hết nước ra khỏi bình 4 lít
6. Nếu ($y > 0$) thì (x, y) \rightarrow ($x, 0$) Đổ hết nước ra khỏi bình 3 lít

4.3 Tri thức luật dẫn



- Bài toán được biểu diễn ở dạng không gian trạng thái bằng luật:

7. Nếu $(x+y \geq 4)$ và $(y > 0)$ thì $(x, y) \rightarrow (4, y-(4-x))$

Đổ nước từ bình 3 lít vào bình 4 lít đến khi bình 4 lít đầy

8. Nếu $(x+y \geq 3)$ và $(x > 0)$ thì $(x, y) \rightarrow (x-(3-y), 3)$

Đổ nước từ bình 4 lít vào bình 3 lít cho đến khi bình 3 lít đầy.

9. Nếu $(x+y \leq 4)$ và $(y > 0)$ thì $(x, y) \rightarrow (x+y, 0)$

Đổ hết nước bình 3 lít vào bình 4 lít.

10. Nếu $(x+y \leq 3)$ và $(x > 0)$ thì $(x, y) \rightarrow (0, x+y)$

Đổ hết nước bình 4 lít vào bình 3 lít.

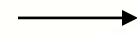
11. $(0, 2) \rightarrow (2, 0)$

Đổ 2 lít từ bình 3 lít vào bình 4 lít.

12. $(2, y) \rightarrow (0, y)$

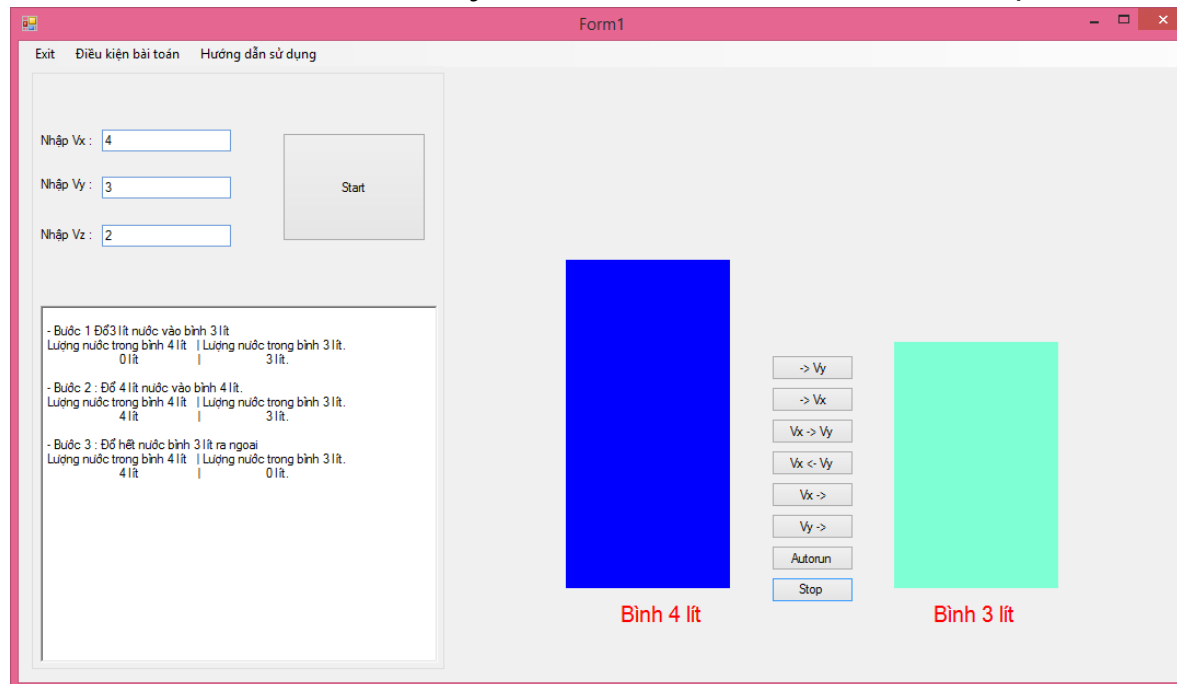
Đổ 2 lít ra khỏi bình 4 lít.

4.3 Tri thức luật dẫn



■ Chúng ta có thể rút gọn 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).



The screenshot shows a Java Swing application window titled "Form1". It contains a menu bar with "Exit", "Điều kiện bài toán", and "Hướng dẫn sử dụng". Below the menu bar, there are three input fields labeled "Nhập Vx:", "Nhập Vy:", and "Nhập Vz:" with values 4, 3, and 2 respectively. A "Start" button is to the right of these fields. Below the input fields, there is a text area containing the following text:

- Bước 1: Đổ 3 lít nước vào bình 3 lít.
Lượng nước trong bình 4 lít | Lượng nước trong bình 3 lít.
0 lít | 3 lít.

- Bước 2: Đổ 4 lít nước vào bình 4 lít.
Lượng nước trong bình 4 lít | Lượng nước trong bình 3 lít.
4 lít | 3 lít.

- Bước 3: Đổ hết nước bình 3 lít ra ngoài.
Lượng nước trong bình 4 lít | Lượng nước trong bình 3 lít.
4 lít | 0 lít.

To the right of the text area, there are two large colored rectangles representing the jugs: a blue rectangle labeled "Bình 4 lít" and a green rectangle labeled "Bình 3 lít". Between the jugs, there is a vertical stack of buttons: "> Vy", "> Vx", "Vx > Vy", "Vx < Vy", "Vy >", "Vx <", "Autonun", and "Stop".



4.3 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.

4.3 Tri thức luật dẫn

▪ Nhược điểm

- Các tri thức phức tạp đôi lúc đòi hỏi **quá nhiều luật dẫn** → 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 → 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.

4.4 Mạng ngữ nghĩa

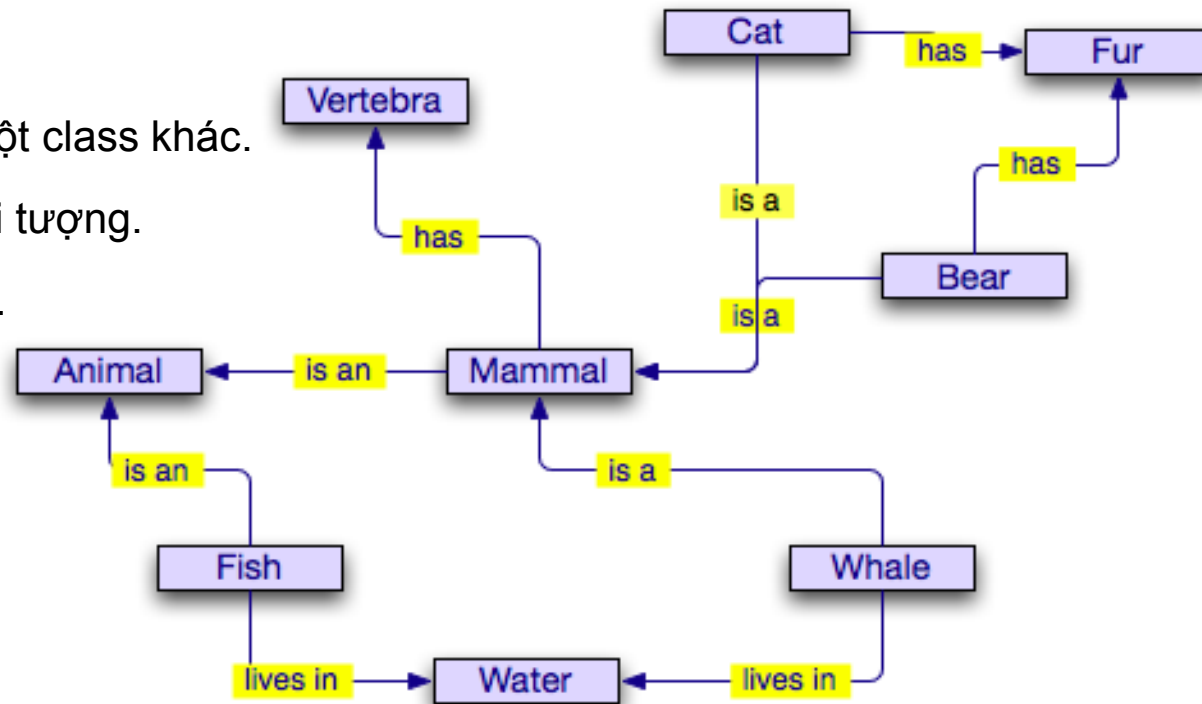
- Là một phương pháp biểu diễn tri thức dùng đồ thị. Trong đó nút biểu diễn đối tượng, và cung biểu diễn quan hệ giữa các đối tượng.

“**IS-A**” (là một thể hiện của)

“**A-KIND-OF**”. Đề cập một class với một class khác.

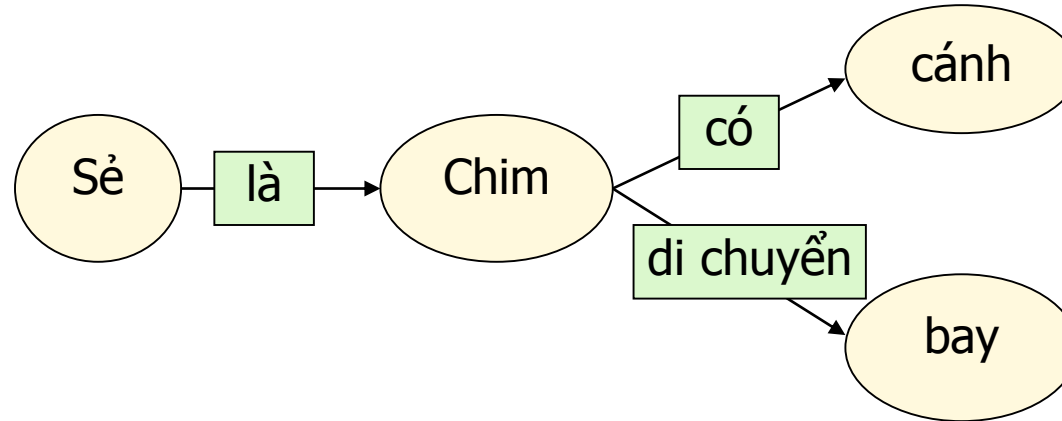
“**HAS-A**”. Thuộc tính liên quan đến đối tượng.

“**CAUSE**”. Thể hiện quan hệ nhân quả.



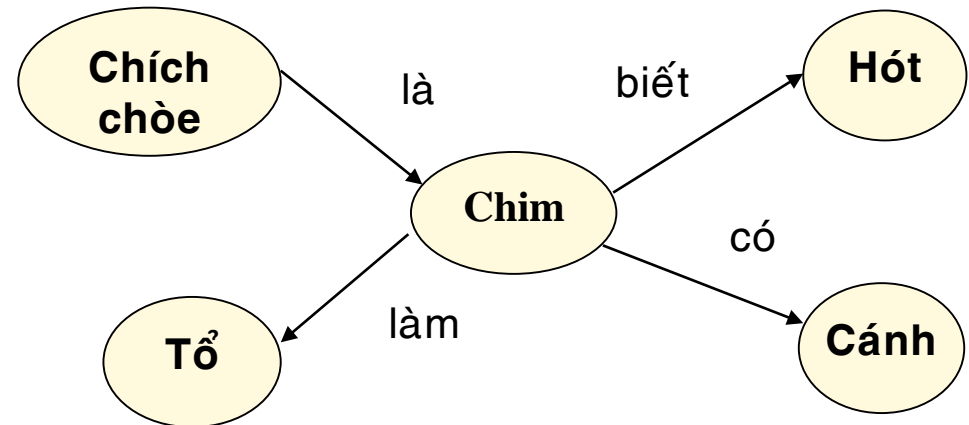
4.4 Mạng ngữ nghĩa

- Ví dụ 1: Một số tri thức về loài “chim sẻ” được biểu diễn trên mạng ngữ nghĩa



- Ví dụ 2: Giữa các khái niệm *chích chòe*, *chim*, *hót*, *cánh*, *tổ* có mối quan hệ sau

Chích chòe là một loài chim.
 Chim biết hót
 Chim có cánh
 Chim sống trong tổ



4.4 Mạng ngữ nghĩa

- Ví dụ 3: “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 các bài toán dạng này?
- Bài toán sẽ giải được 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 \rightarrow Sử dụng khoảng 200 đỉnh để chứa công thức + 22 đỉnh để chứa các yếu tố của tam giác.
 - 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?



4.4 Mạng ngữ nghĩa

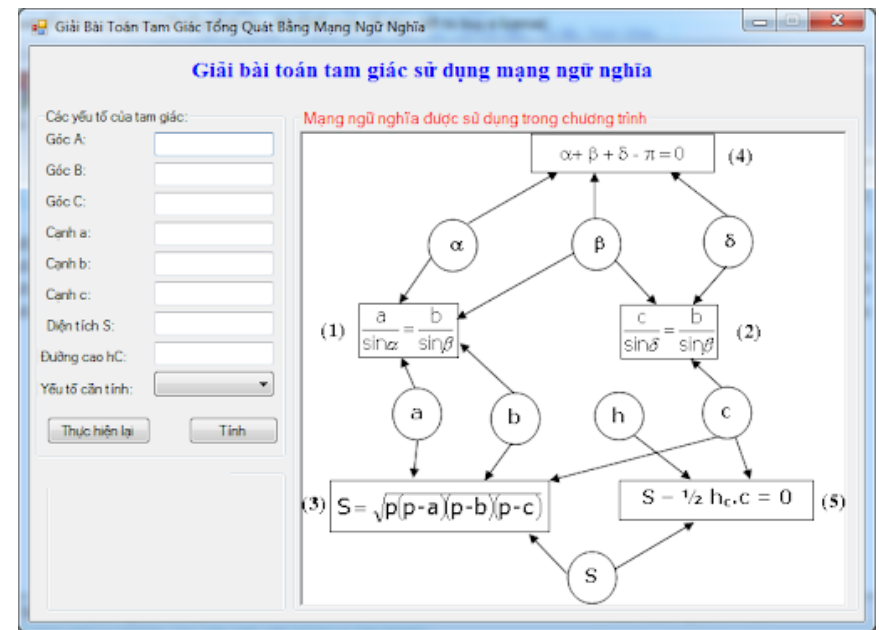
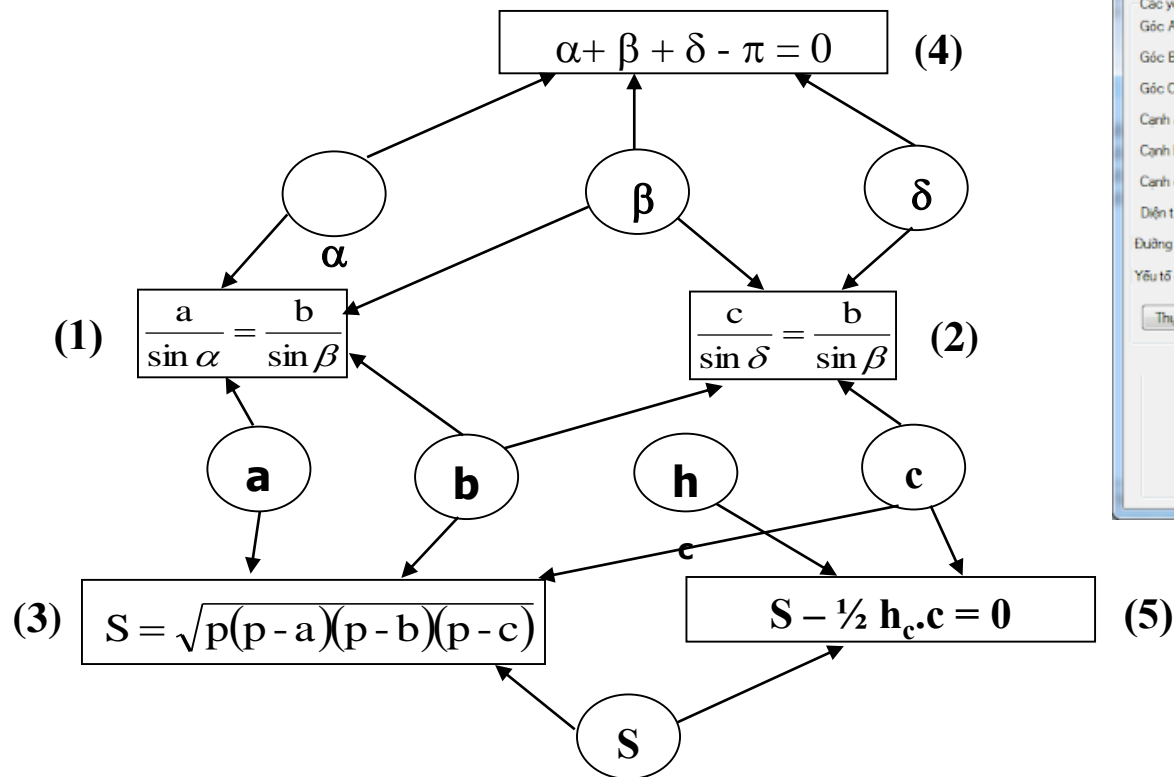
Cơ chế suy diễn thực hiện theo thuật toán “loang”:

- **Bước 1:** Kích hoạt những đỉnh hình tròn đã cho ban đầu (những yếu tố đã có giá trị)
- **Bước 2:** 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).

4.4 Mạng ngữ nghĩa

Cơ chế suy diễn thực hiện theo thuật toán “loang”:

Mạng ngữ nghĩa cho bài toán tam giác



4.4 Mạng ngữ nghĩa

- **Ví dụ 4:** 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 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 δ , đỉnh δ 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.



4.4 Mạng ngữ nghĩa

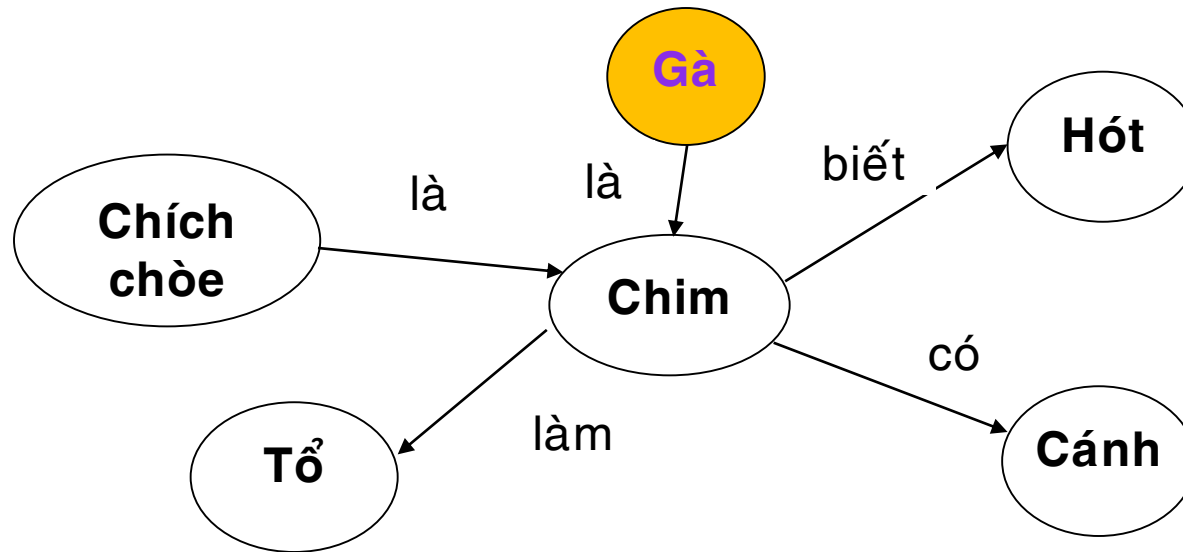
Ưu điểm

- Mạng ngữ nghĩa rất **linh động** → 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.
- 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.
- Hoạt động khá tự nhiên theo cách con người ghi nhận thông tin.

4.4 Mạng ngữ nghĩa

Nhược điểm

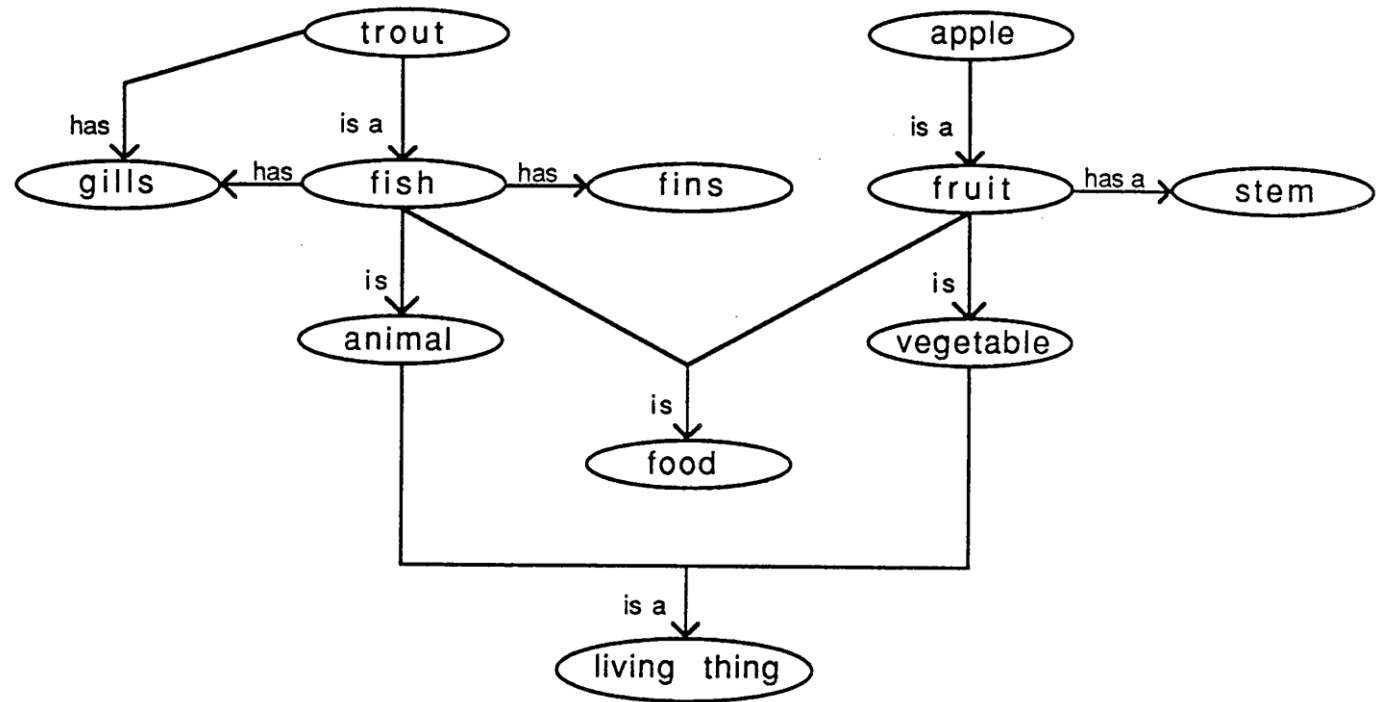
- Chưa có một chuẩn nào quy định các giới hạn cho các đỉnh và cung của mạng. Nghĩa là bạn có thể gán ghép bất kỳ khái niệm nào cho đỉnh hoặc cung!
- Tính thừa kế (vốn là một ưu điểm) trên mạng sẽ có thể dẫn đến **nguy cơ mâu thuẫn trong tri thức**.



4.4 Mạng ngữ nghĩa

Bài tập:

- (1) A trout is a fish.
- (2) A fish has gills.
- (3) A fish has fins.
- (4) Fish is food.
- (5) Fish is animal.
- (6) An apple is a fruit
- (7) Fruit has a stem.
- (8) Fruit is food.
- (9) Fruit is vegetable.
- (10) An animal is a living thing.
- (11) A vegetable is a living thing.





4.4 Mạng ngữ nghĩa

Bài tập:

- (1) A TIGER IS A CAT .
- (2) A TIGER HAS-STRIPES .
- (3) A TIGER IS WILD .
- (4) A TABBIE IS A CAT .
- (5) A TABBIE HAS-STRIPES .
- (6) A TABBIE IS A PET .
- (7) A FLAG HAS-STRIPES .
- (8) A FLAG IS AN ARTIFACT .
- (9) A CAT IS A MAMMAL .
- (10) A MAMMAL IS AN ANIMAL .
- (11) A DOG IS A MAMMAL .
- (12) A DOG IS A PET .
- (13) A DOG BARKS .

4.5 Frame

- Frame là một **cấu trúc dữ liệu** chứa đựng tất cả những tri thức liên quan đến một đối tượng (**Object**) cụ thể.
- Frames được sử dụng trong các ngôn ngữ lập trình hướng đối tượng phục vụ cho trí tuệ nhân tạo và các hệ chuyên gia.
- Một frame bao gồm 2 thành phần cơ bản là **slot** và **facet**
 - Slot là một thuộc tính đặc tả đối tượng được biểu diễn bởi frame. Mỗi slot có thể chứa một hoặc nhiều facet
 - Các facet (đôi lúc được gọi là **slot "con"**) đặc tả một số thông tin hoặc thủ tục liên quan đến thuộc tính được mô tả bởi slot. Facet có nhiều loại khác nhau, sau đây là một số facet thường gặp: **value**, **default value**, **range**, ...

4.5 Frame

- Frame biểu diễn ở dạng khái niệm hay đối tượng

Frame name:	Object1	
Class:	Object2	
Properties:	Property 1	Value1
	Property 2	Value2

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.

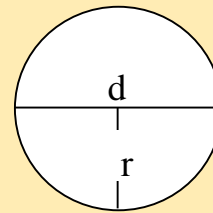
4.5 Frame

■ 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)

Frame : **CIRCLE** (hình tròn)

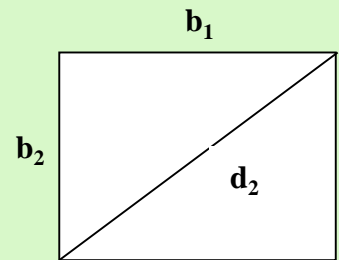
r : radius;
s : area;
p : perimeter;
d : diameter;
 $d = 2 \times r$;
 $s = \pi \times r^2$;
 $p = 2 \times \pi \times r$;



Frame **RECTANGLE**

(hình chữ nhật)

b_1 : side;
 b_2 : side;
s : area;
p : perimeter;
 $s = b_1 \times b_2$;
 $p = 2 \times (b_1 + b_2)$;
 $d^2 = b_1^2 + b_2^2$;



Frame **SQUARE** (hình vuông)

Là : **RECTANGLE**

$b_1 = b_2$;

4.5 Frame

- Một ví dụ biểu diễn các đối tượng hình học bằng frame

Frame **RHOMBUS** (hình thoi)

b : side;

d_1 : diagonal;

d_2 : diagonal;

s : area;

p : perimeter;

α_1 : angle;

α_2 : angle;

h : height;

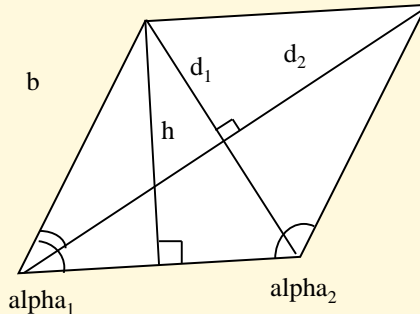
$$\cos(\alpha_2/2) \times d_1 = h;$$

$$s = d_1 \times d_2 / 2;$$

$$p = 4 \times b;$$

$$s = b \times h;$$

$$\cos(\alpha_2/2)/(2 \times b) = d_2/2;$$



Frame **TRIANGLE**

a : side;

b : side;

c : side;

aBc : angle; //góc đối diện cạnh B.

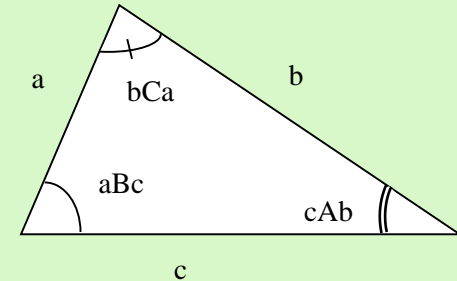
cAb : angle; //góc đối diện cạnh C.

bCa : angle; //góc đối diện cạnh A.

$$s = \sqrt{p(p-a)(p-b)(p-c)}$$

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

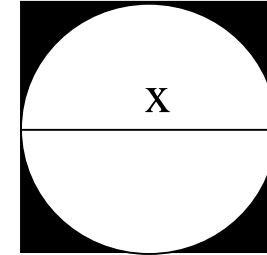
$$\frac{a}{\sin(cAb)} = \frac{b}{\sin(aBc)} = \frac{c}{\sin(bCa)}$$



4.5 Frame

- Vận dụng để giải các bài toán hình học, chẳng hạn bài toán tính diện tích

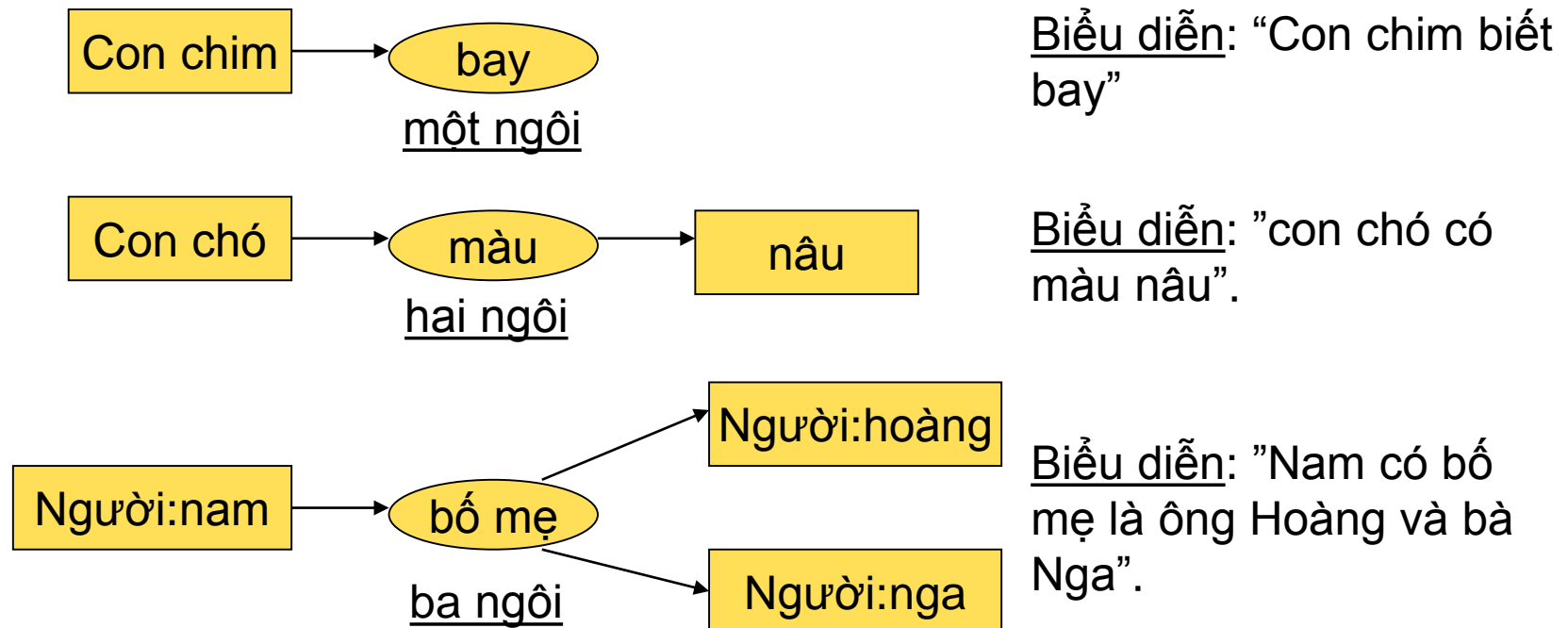
Cho hình vuông **k** và vòng tròn nội tiếp **c**, biết cạnh hình vuông có chiều dài là **x**, hãy viết chương trình để tính diện tích phần tô đen.



```
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.
```

4.6 Đồ thị khái niệm

- Đồ thị khái niệm là một đồ thị hữu hạn, liên thông, các đỉnh được chia làm hai loại: đỉnh khái niệm và đỉnh quan hệ
 - **Đỉnh khái niệm**: dùng để biểu diễn các khái niệm cụ thể (cái, điện thoại, ...) hay trừu tượng (tình yêu, đẹp, văn hoá,...). Đỉnh khái niệm được biểu diễn bởi hình chữ nhật có gán nhãn
 - **Đỉnh quan hệ**: dùng để chỉ ra quan hệ giữa các khái niệm có nối đến nó





4.7 Script

Script: RESTAURANT

Track: Coffee Shop

Entry conditions:

S is hungry

S has money

Results:

S has less money

O has more money

S is not hungry

S is pleased (optional)

Props: Tables

Menu

Food (F)

Check

Money

Roles: Customer (S)

Waiter(W)

Cook(C)

Cashier(M)

Owner(O)

Scene 1: (Entering)

S PTRANS S into restaurant.

S ATTEND eyes to tables

S MBUILD where to sit

S PTRANS S to table

S MOVE S to sitting position

Scene 2: (Ordering)

(Menu on table)

S PTRANS menu to S

(S ask for menu)

S MTRANS signal to W

W PTRANS W to table

S MTRANS 'need menu' to W

W PTRANS W to menu

5. Các vấn đề trong biểu diễn tri thức

- Tri thức được biểu diễn đến **mức chi tiết nào?**
- Bằng cách nào **thể hiện được meta-knowledge?**
- Bằng cách nào thể hiện tính **phân cấp của tri thức** (kế thừa, ngoại lệ, giá trị mặc định, đa thừa kế)?
- Khi mô tả đối tượng, bằng cách nào có thể tích hợp một tri thức thủ tục vào bản thân mô tả, khi nào thủ tục được thực hiện
- Với số lượng lớn tri thức được chứa trong CSDL, bằng cách nào truy xuất những thành phần cần thiết?