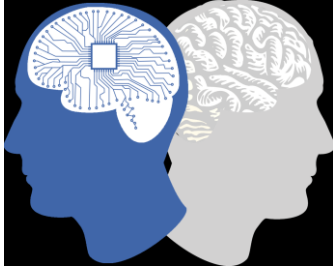


AI Artificial Intelligence

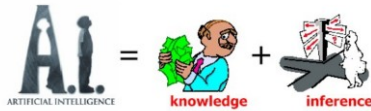


BIỂU DIỄN TRI THỨC

1. Dữ liệu – Thông Tin – Tri Thức VS AI VS Programming
2. Tri thức các đặc điểm và phân loại
3. Một số phương pháp biểu diễn tri thức
 - I. Lược đồ logic
 - II. Lược đồ thủ tục
 - III. Lược đồ cấu trúc
 - IV. Lược đồ mạng.
4. Biểu diễn tri thức
 - I. Logic mệnh đề (Thuật giải robinson và)
 - II. Mạng ngữ nghĩa.

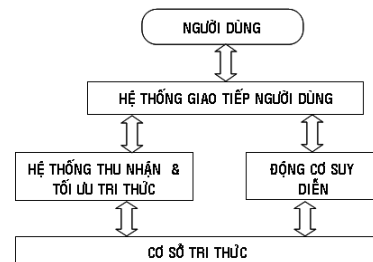
AI VS Programming

So với chương trình truyền thống (được cấu tạo từ hai "chất liệu" cơ bản là **dữ liệu và thuật toán**), chương trình trí tuệ nhân tạo được cấu tạo từ hai thành phần là **cơ sở tri thức (knowledge base)** và **động cơ suy diễn (inference engine)**.



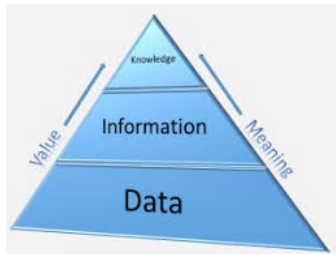
AI

- **Cơ sở tri thức** : là **tập hợp các tri thức liên quan** đến vấn đề mà chương trình quan tâm giải quyết.
- **Động cơ suy diễn** : là **phương pháp vận dụng tri thức** trong cơ sở tri thức để giải quyết vấn đề.



Cấu trúc của một chương trình trí tuệ nhân tạo.

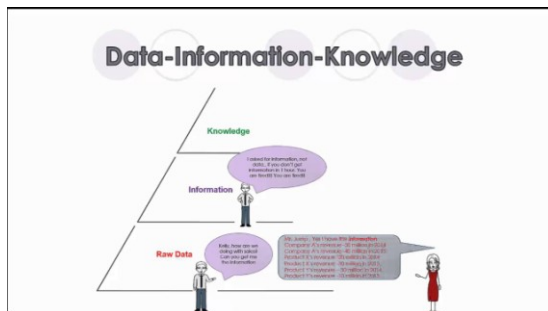
DỮ LIỆU – THÔNG TIN – TRI THỨC



DỮ LIỆU – THÔNG TIN – TRI THỨC

- **Dữ liệu (data)** thường được định nghĩa là các sự kiện (facts) hoặc các ký hiệu (symbols)
- **Thông tin (information)** thường được định nghĩa là dữ liệu đã được xử lý hoặc chuyển đổi thành những dạng hoặc cấu trúc phù hợp cho việc sử dụng của con người
- Thông tin có được sau (chứ không xuất hiện trước) dữ liệu
- **Tri thức (knowledge)** thường được định nghĩa là sự hiểu biết (nhận thức) về thông tin

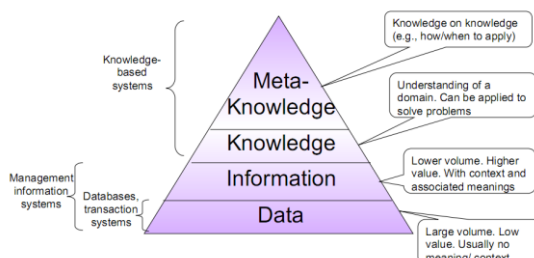
DỮ LIỆU – THÔNG TIN – TRI THỨC



DỮ LIỆU – THÔNG TIN – TRI THỨC

- Dữ liệu
 - Nhiệt độ ngoài trời là 5 độ C
 - Thông tin
 - Ngoài trời thời tiết lạnh
 - Tri thức
 - Nếu ngoài trời thời tiết lạnh thì bạn nên mặc áo choàng ấm (khi đi ra ngoài)
- Giá trị (sử dụng) của dữ liệu tăng lên khi nó được “chuyển đổi” thành tri thức
- Sử dụng tri thức sẽ cho phép đưa ra các quyết định phù hợp và hiệu quả

DỮ LIỆU – THÔNG TIN – TRI THỨC



(Adapted from "Knowledge Engineering course (CM3016), by K. Hui 2008-2009")

PHÂN LOẠI TRI THỨC

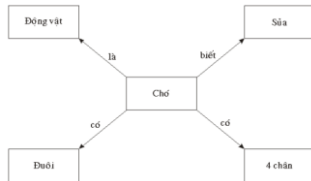
1. Tri thức thủ tục: mô tả cách thức giải quyết một vấn đề. Loại tri thức này đưa ra giải pháp để thực hiện một công việc nào đó. Các dạng tri thức thủ tục tiêu biểu thường là các luật, chiến lược, lịch trình và thủ tục.
2. Tri thức khai báo: cho biết một vấn đề được thấy như thế nào. Loại tri thức này 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.

PHÂN LOẠI TRI THỨC

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 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.

ĐẶC ĐIỂM CỦA TRI THỨC

- **Tính liên hệ:** Ngoài các quan hệ về cấu trúc của mỗi tri thức (khái niệm, quá trình, sự kiện, hiện tượng,...) *giữa các đơn vị tri thức còn có nhiều mối liên hệ khác (không gian, thời gian, nhân-quả, ...)*
- **Ví dụ:** Các khái niệm: chó, sủa, động vật, bốn chân, đuôi.



BIỂU DIỄN TRI THỨC

- Biểu diễn tri thức (Knowledge representation) là một lĩnh vực nghiên cứu quan trọng của Trí tuệ nhân tạo
 - Nhằm phát triển các phương pháp, cách thức biểu diễn tri thức và các công cụ hỗ trợ việc biểu diễn tri thức
- Một số cách biểu diễn:
 - Lược đồ logic
 - Lược đồ thủ tục
 - Mạng ngữ nghĩa
 - Ontology
 -

ĐẶC ĐIỂM CỦA TRI THỨC

Làm thế nào để **phân biệt thông tin vào máy tính là dữ liệu hoặc tri thức**.
Giữa tri thức và dữ liệu có một số đặc trưng khác nhau.

- **Tự giải thích nội dung:** *Tri thức tự giải thích nội dung còn dữ liệu không tự giải thích được.* Chỉ có người lập trình mới hiểu được nội dung ý nghĩa các dữ liệu.
- **Ví dụ:**
 - Dữ liệu là số 7.
 - Tri thức là số 7: là số lẻ, là số nguyên tố, là số dương,...
- **Tính cấu trúc:** Một trong những đặc trưng cơ bản của hoạt động nhận thức con người đối với thế giới xung quanh là khả năng **phân tích cấu trúc các đối tượng**. Ở mức đơn giản nhất là cấu trúc: là một bộ phận của toàn thể, là một giống của một loài nào đó, là phần tử của lớp nào đó.

Tri thức đưa vào máy cũng cần có khả năng tạo được phân cấp giữa các khái niệm và quan hệ giữa chúng.

ĐẶC ĐIỂM CỦA TRI THỨC

- Có tính chủ động:

Dữ liệu hoàn toàn bị động do con người khai thác, còn **tri thức thì có tính chủ động**. Khi hoạt động bất kỳ ở đâu trong lĩnh vực nào, con người cũng bị điều khiển bởi tri thức của mình. Các tri thức biểu diễn trong máy tính cũng vậy, chúng **chủ động hướng người dùng biết cách khai thác dữ liệu**.

BIỂU DIỄN TRI THỨC

1. LƯỢC ĐỒ LOGIC

Logic mệnh đề:

-Định nghĩa: Mệnh đề là một khẳng định có thể nhận giá trị đúng hoặc sai.

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:

$$A \wedge B \Rightarrow C$$

BIỂU DIỄN TRI THỨC

1. LƯỢC ĐỒ LOGIC

Logic vị từ :

- **Định nghĩa:** là sự mở rộng của logic mệnh đề bằng cách đưa vào các khái niệm vị từ và các lượng từ phổ thông dụng (\forall, \exists).

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

• Vị từ (\langle đối tượng 1 \rangle , \langle đối tượng 2 \rangle , ..., \langle đối tượng n \rangle)

Ví dụ:

- Từ đó câu: "Spot has a tail", có thể thu được qua các bước:

Từ 2, $X = \text{"Spot"}: \text{dog}(\text{Spot}) \rightarrow \text{hastail}(\text{Spot})$.

Từ 1, 3: $\text{hastail}(\text{Spot})$.

- Ánh xạ ngược \rightarrow "Spot has a tail".

Ví dụ:

- Câu tiếng Anh:

"Spot is a dog"

"Every dog has a tail"

- Dạng logic:

1. $\text{dog}(\text{Spot})$.

2. X

3. $(\text{dog}(X) \rightarrow \text{hastail}(X))$.

Ví dụ:

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 :

$\text{LớnHơn}(x,y) = x > y$

$\text{NhỏHơn}(x,y) = x < y$

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

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 :

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

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**.

BIỂU DIỄN TRI THỨC

2. LƯỢC ĐỒ THU TỤC

Khác với khai báo, lược đồ này **biểu diễn tri thức như tập các chỉ thị lệnh để giải quyết vấn đề**.

Các chỉ thị lệnh trong lược đồ thủ tục chỉ ra bằng cách nào giải quyết vấn đề

Ví dụ: **hệ luật sinh** điển hình cho loại lược đồ này

Hệ luật sinh

- 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**
- 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 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**

Hệ luật sinh – 7 dạng cơ bản

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. **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

BIỂU DIỄN TRI THỨC

3. LƯỢC ĐỒ MẠNG

- **Biểu diễn tri thức như là đồ thị**, các **đỉnh** như là các **đối tượng** hoặc **khái niệm**, các **cung** như là **quan hệ** giữa chúng
- Ví dụ: **mạng ngữ nghĩa (semantic network)**, lược đồ **quan hệ phụ thuộc** khái niệm đồ thị khái niệm

BIỂU DIỄN TRI THỨC

4. LƯỢC CẤU TRÚC

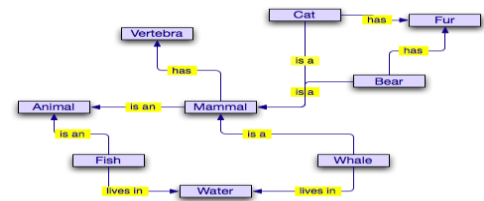
- Là một **mở rộng của lược đồ mạng**; bằng cách cho phép các **nút có thể là một CTDL phức tạp gồm các khe (slot) có tên và trị** hay một **thủ tục**.
- Ví dụ: kịch bản (**script**), khung (**frame**), đối tượng (**object**)

Hệ luật sinh – 7 dạng cơ bản

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

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**



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

Frame – Biểu diễn ở dạng khái niệm hay đối tượng

- Mỗi frame mô tả một *đối tượng (object)*
- 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, ...*

Frame – Biểu diễn ở dạng khái niệm hay đối tượng

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

Frame

- Frame thường được dùng để biểu diễn những tri thức "**chuẩn**"
- hoặc những **tri thức được xây dựng dựa trên những kinh nghiệm hoặc các đặc điểm đã được hiểu biết cận kề**

Đối tượng - Thuộc tính – Giá trị

- **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*"

Các vấn đề trong biểu diễn tri thức

- Có những **thuộc tính cơ bản** nào của đối tượng mà chúng xuất hiện trong mọi lĩnh vực không? Nếu có: những thuộc tính nào?
- Có chắc chắn là chúng sẽ được xử lý thích hợp trong từng cơ chế được đề nghị không?
- Có **quan hệ quan trọng nào tồn tại cùng với thuộc tính không?**
- Các đối tượng và các quan hệ **có thể biểu diễn cho cái gì trong lĩnh vực?**

Ví dụ: để biểu diễn cho ý "Nam cao 1 mét 70", có thể dùng: **chieucao(nam, 170)**. Để diễn tả "An cao hơn Nam"?

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**: các hình thức: kế thừa, ngoại lệ, trị mặc định, ngoại lệ, đa thừa kế phải đặc tả như thế nào

Các vấn đề trong biểu diễn tri thức

- 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**?

Đặc điểm của hệ thống biểu diễn tri thức

- Khả năng **biểu diễn tất cả** các tri thức cần thiết cho lĩnh vực đó
- Khả năng **xử lý các cấu trúc sẵn có** để sinh ra các **cấu trúc mới tương ứng** với tri thức mới được sinh ra từ tri thức cũ
- Khả năng **thêm vào cấu trúc những tri thức thông tin bổ sung** mà nó có thể được dùng để hướng dẫn cơ chế suy luận theo hướng có nhiều triển vọng nhất

Đặc điểm của hệ thống biểu diễn tri thức

- Khả năng **thu được thông tin mới dễ dàng**
- Trường hợp đơn giản nhất là **chèn trực tiếp tri thức mới (do con người) vào cơ sở tri thức**. Lý tưởng nhất là chương trình có thể kiểm soát việc thu được tri thức

Biểu diễn tri thức bằng logic mệnh đề

- AI: Phát triển các chương trình có khả năng suy luận
- Suy luận giúp chương trình AI biết được tính đúng/sai của một vấn đề nào đó
- Phép toán vị từ \rightarrow cung cấp một khả năng triển khai các quá trình suy diễn trên máy tính
- Phép toán vị từ được hiện thực bằng ngôn ngữ lập trình trên máy tính **PROLOG**

Cú pháp

- Cú pháp của logic mệnh đề gồm tập **các ký hiệu và tập các luật** xây dựng công thức
- **Các ký hiệu**
 - Hai hằng logic: **True** và **False**
 - Ký hiệu mệnh đề (biến mệnh đề): P, Q, \dots
 - Các phép kết nối logic: \wedge (*hội*), \vee (*tuyển*), \neg (*phủ định*), \Rightarrow (*kéo theo/ suy ra*), \Leftrightarrow (*tương đương*)
 - Cặp dấu ngoặc tròn: ()

\rightarrow Cách đánh giá giá trị của phép toán: **Bảng chân trị**

Ví dụ

Mệnh đề thực tế	Mệnh đề logic
▪ “Nếu trời mưa thì bầu trời có mây”	▪ $P = \text{“Trời mưa”}$
▪ Trời đang mưa	▪ $Q = \text{“Bầu trời có mây”}$
Vậy \rightarrow Bầu trời có mây	Ta có hai phát biểu sau đúng: ▪ $P \rightarrow Q$ ▪ P
	Theo luật suy diễn $\rightarrow Q$ là đúng.
	Nghĩa là: “Bầu trời có mây”

Ví dụ

Mệnh đề thực tế	Mệnh đề logic
▪ “Nếu NAM có nhiều tiền thì NAM đi mua sắm”	▪ $P = \text{“Nam có nhiều tiền”}$
▪ “Nam KHÔNG đi mua sắm”	▪ $Q = \text{“Nam đi mua sắm”}$
Vậy \rightarrow Nam KHÔNG có nhiều tiền	Ta có hai phát biểu sau đúng: ▪ $P \rightarrow Q$ ▪ $\neg Q$
	Vậy theo luật suy diễn $\rightarrow \neg P$ là đúng. Nghĩa là: “Nam KHÔNG có nhiều tiền”

Mệnh đề

Mệnh đề:

- Mệnh đề là một phát biểu khai báo
- Mệnh đề chỉ nhận một trong hai giá trị:
 - ĐÚNG (*True*)
 - hoặc SAI (*False*)

Ví dụ:

- Ngày 01 tháng giêng là ngày tết cổ truyền
- Môn bạn đang học là AI

Bảng chân trị

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

Qui tắc xây dựng công thức

- Các công thức là các ký hiệu mệnh đề sẽ được gọi là các *câu đơn* hoặc *câu phân tử*
- Các công thức không phải là câu đơn sẽ được gọi là câu phức hợp
- Nếu P là ký hiệu mệnh đề thì P và $\neg P$ được gọi là *literal*, P là *literal dương*, còn $\neg P$ là *literal âm*

Ngữ nghĩa

- Xác định** ý nghĩa của các công thức trong thế giới thực bằng cách kết hợp mỗi ký hiệu mệnh đề với sự kiện nào đó trong thế giới hiện thực
- Ký hiệu mệnh đề có thể ứng với sự kiện nào đó
- Sự kết hợp các ký hiệu mệnh đề với các sự kiện trong thế giới thực được gọi là một *minh họa* (interpretation)
- Một sự kiện chỉ có thể đúng hoặc sai. VD sự kiện “Paris là thủ đô nước Pháp” là đúng

Ví dụ

$P = \text{“Nam học giỏi”}$; $Q = \text{“Nam thông minh”}$; $R = \text{“Nam đẹp trai”}$

Mệnh đề thực tế	Biểu thức mệnh đề
▪ “Nam học giỏi, thông minh, đẹp trai”	▪ $P \wedge Q \wedge R$
▪ “Nam học giỏi hoặc thông minh”	▪ $P \vee Q$
▪ “Nam hoặc học giỏi, hoặc đẹp trai”	▪ $(P \wedge \neg R) \vee (\neg P \wedge R)$
▪ “Nam thông minh thì học giỏi”	▪ $Q \Rightarrow P$

Dạng chuẩn tắc

- Chuẩn hóa các công thức
- Đưa các công thức về dạng **thuận lợi cho việc lập luận, suy diễn**.
- Sử dụng các phép biến đổi tương đương \rightarrow có thể đưa một công thức bất kỳ về dạng chuẩn tắc

Sự tương đương các công thức

- $A \Rightarrow B \equiv \neg A \vee B$
- $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$
- $\neg(\neg A) \equiv A$

Luật De Morgan

- $\neg(A \vee B) \equiv \neg A \wedge \neg B$
- $\neg(A \wedge B) \equiv \neg A \vee \neg B$

Sự tương đương các công thức

Luật giao hoán

- $A \vee B \equiv B \vee A$
- $A \wedge B \equiv B \wedge A$

Luật kết hợp

- $(A \vee B) \vee C \equiv A \vee (B \vee C)$
- $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$

Luật phân phối

- $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
- $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

Chuyển thành dạng chuẩn tắc

- Để dễ dàng viết các chương trình máy tính \rightarrow đưa chúng về dạng biểu diễn **dạng chuẩn hội**
- Một công thức ở dạng chuẩn hội nếu nó là hội của các câu tuyển
- **Cách thực hiện**
 1. Bỏ các dấu kéo theo (\Rightarrow) bằng cách thay $(A \Rightarrow B)$ bởi $(\neg A \vee B)$
 2. Chuyển các dấu phủ định (\neg) vào sát các ký hiệu mệnh đề bằng cách áp dụng luật De Morgan và thay $\neg(\neg A)$ bởi A
 3. Áp dụng luật phân phối, thay các công thức có dạng $A \vee (B \wedge C)$ bởi $(A \vee B) \wedge (A \vee C)$

Ví dụ chuẩn hóa $(P \Rightarrow Q) \vee \neg(R \vee \neg S)$

- $(P \Rightarrow Q) \vee \neg(R \vee \neg S)$
 - $(\neg P \vee Q) \vee \neg(R \vee \neg S)$: Loại bỏ dấu \Rightarrow
 - $(\neg P \vee Q) \vee (\neg R \wedge S)$: Luật De Morgan
 - $((\neg P \vee Q) \vee \neg R) \wedge ((\neg P \vee Q) \vee S)$: Luật phân phối
 - $(\neg P \vee Q \vee \neg R) \wedge (\neg P \vee Q \vee S)$
- \rightarrow Tập hợp các câu tuyển

Câu Horn

- Mọi công thức đều có thể đưa về dạng chuẩn hội, có dạng:
 - $\neg P_1 \vee \dots \vee \neg P_m \vee Q_1 \vee \dots \vee Q_n$
 - tương đương với: $P_1 \wedge \dots \wedge P_m \Rightarrow Q_1 \vee \dots \vee Q_n$ gọi là **câu Kowalski** (do nhà logic Kowalski đưa ra năm 1971)
- Khi $n \leq 1$, tức là câu tuyển chỉ chứa nhiều nhất một literal dương \rightarrow gọi là **câu Horn** (mang tên nhà logic Alfred Horn, năm 1951)
- Nếu $m > 0$, $n=1$, câu **Horn** có dạng: $P_1 \wedge \dots \wedge P_m \Rightarrow Q$
- Có thể biểu diễn thành các luật **if-then**:

If P_1 and ... and P_m then Q

Luật suy diễn

- H được xem là **hệ quả logic** (logical consequence) của một tập công thức $G = \{G_1, \dots, G_m\}$ nếu trong bất kỳ minh họa nào mà $\{G_1, \dots, G_m\}$ đúng thì H cũng đúng
- Dùng các tri thức trong cơ sở để suy ra tri thức mới mà nó là hệ quả logic của các công thức trong cơ sở tri thức: sử dụng **các luật suy diễn** (rule of inference)
- Một luật suy diễn gồm hai phần: **một tập các điều kiện** và **một kết luận**

Một số luật suy diễn quan trọng

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	$\alpha \vee \beta, \neg\beta \vee \gamma$	$\alpha \vee \gamma$

Một luật suy diễn được xem là tin cậy nếu bất kỳ một mô hình nào của giả thiết của luật cũng là mô hình kết luận của luật

Tiên đề, định lý, chứng minh

- Các luật suy diễn cho phép suy ra các công thức mới từ các công thức đã có bằng một dãy áp dụng các luật suy diễn
- Các công thức đã cho: các **tiên đề**
- Các công thức được suy ra: các **định lý**
- Dãy các luật được áp dụng để dẫn tới định lý: **chứng minh** của định lý
- Nếu các luật suy diễn là tin cậy, thì các định lý là hệ quả logic của các tiên đề

Ví dụ - Cần chứng minh công thức GVH

Giả sử có các công thức:

- $Q \wedge S \Rightarrow G \vee H$ (1)
- $P \Rightarrow Q$ (2)
- $R \Rightarrow S$ (3)
- P (4)
- R (5)

Áp dụng luật Modus Ponens: Từ (2) và (4) suy ra Q.
Từ (3) và (5) suy ra S

• Từ Q, S ta suy ra $Q \wedge S$ (luật đưa vào hội)

• Từ (1) và $Q \wedge S$ ta suy ra GVH

→ Công thức GVH đã được chứng minh

Vấn đề chứng minh phép suy diễn

- Tính đúng đắn của phép suy diễn: $a \rightarrow b$?
- Hai phép suy luận cơ bản của logic mệnh đề (Modus Ponens, Modus Tollens) cộng với các phép biến đổi hình thức → có thể chứng minh được phép suy diễn
- Tuy nhiên, thao tác biến đổi hình thức là rất khó cài đặt trên máy tính

Vấn đề chứng minh phép suy diễn

- Công cụ máy tính có thể chứng minh bằng cách lập bảng chân trị → **Độ phức tạp quá lớn: $O(2^n)$ với n là số biến mệnh đề**
- Phương pháp chứng minh mệnh đề với độ phức tạp $O(n)$: **Thuật giải Vương Hạo và Robinson**

Thuật giải Vương Hạo

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

GT1, GT2, ..., GTn \rightarrow KL1, KL2, ..., KLm

- Trong đó các GTi và KLi là các mệnh đề được xây dựng từ các biến mệnh đề và 3 phép: \wedge, \vee, \neg
- B2 : Chuyển về các GTi và KLi có dạng phủ định

Thuật giải Vương Hạo

- Ví dụ:
 $p \vee q, \neg (r \wedge s), \neg g, p \vee r \rightarrow s, \neg p$
 chuyển thành:
 $p \vee q, p \vee r, p \rightarrow (r \wedge s), g, s$

Thuật giải Vương Hạo

- B3 : Nếu GTi có phép \wedge thì thay thế phép \wedge bằng dấu ",". Nếu KLi có phép \vee thì thay thế phép \vee bằng dấu ","

Ví dụ:

$p \wedge q, r \wedge (\neg p \vee s) \rightarrow \neg q, \neg s$

chuyển thành:

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

- B4 : Nếu GTi có phép \vee thì tách thành hai dòng con. Nếu ở KLi có phép \wedge thì tách thành hai dòng con

Thuật giải Vương Hạo

- Ví dụ :
 $p, \neg p \vee q \rightarrow q$
 tách thành:
 $p, \neg p \rightarrow q$
 và $p, q \rightarrow q$

Thuật giải Vương Hạo

- B5 : Một dòng được chứng minh nếu tồn tại chung một mệnh đề ở cả hai phía

Ví dụ: $p, q \rightarrow q$ được chứng minh

- B6 :
 - a) Nếu một dòng không còn phép nối \wedge và phép nối \vee ở cả hai vế và 2 vế không có chung 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

Thuật giải Vương Hạo – Ví dụ

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

B1: $\neg p \vee q, \neg q \vee r \rightarrow \neg p \vee r$

B3: $\neg p \vee q, \neg q \vee r \rightarrow \neg p, r$

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

B4: Tách mệnh đề đầu

$\neg p, \neg q \vee r, p \rightarrow r$ (1)

$q, \neg q \vee r, p \rightarrow r$ (2)

Thuật giải Vương Hạo – Ví dụ

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

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: $a \rightarrow b$

Thuật giải Robinson

- Thuật giải này hoạt động dựa **trên phương pháp chứng minh phản chứng và phép hợp giải Robinson**.
- 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.

Thuật giải Robinson

- Phép hợp giải Robinson:
 - $p \wedge (\neg p \vee q) \rightarrow q$
 - $(p \vee q) \wedge (\neg p \vee r) \rightarrow q \vee r$
- Bài toán được chứng minh nếu:

a đúng và $\neg b$ đúng sinh ra một mâu thuẫn

Thuật giải Robinson

- B1 : 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
- B2 : Nếu GT_i có phép \wedge thì thay bằng dấu " , ". Nếu KL_i có phép \vee thì thay bằng dấu " , "
- B3 : Biến đổi dòng chuẩn ở B1 về thành danh sách mệnh đề như sau:

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

Thuật giải Robinson

- B4 : Nếu trong danh sách mệnh đề có 2 mệnh đề đối ngẫu nhau thì bài toán được chứng minh. Ngược lại thì chuyển sang B5. (a và $\neg a$ gọi là hai mệnh đề đối ngẫu)
- B5 : Xây dựng mệnh đề mới bằng cách tuyển một cặp trong mệnh đề. Nếu mệnh đề mới có các biến mệnh đề đối ngẫu thì các biến đó được loại bỏ.
- B6 : Áp dụng phép hợp giải
 - $p \wedge (\neg p \vee q) \rightarrow q$
 - $(p \vee q) \wedge (\neg p \vee r) \rightarrow q \vee r$

Thuật giải Robinson

- Nếu không xây dựng được thêm một mệnh đề mới
- Trong danh sách mệnh đề không có 2 mệnh đề nào đối ngẫu nhau thì vấn đề không được chứng minh

Thuật giải Robinson – Ví dụ

Chứng minh:

$$(\neg p \vee q) \wedge (\neg q \vee r) \wedge (\neg r \vee s) \wedge (\neg u \vee \neg s) \rightarrow \neg p \vee \neg u$$

$$B2: \neg p \vee q, \neg q \vee r, \neg r \vee s, \neg u \vee \neg s \rightarrow \neg p, \neg u$$

- B3: Danh sách mệnh đề:

$$\{\neg p \vee q, \neg q \vee r, \neg r \vee s, \neg u \vee \neg s, p, u\}$$

- B4: Có 6 mệnh đề nhưng không đối ngẫu sang B5

Thuật giải Robinson – Ví dụ

$$\{\neg p \vee q, \neg q \vee r, \neg r \vee s, \neg u \vee \neg s, p, u\}$$

- B5: Chọn cặp mệnh đề $\neg p \vee q \vee \neg q \vee r \Rightarrow \neg p \vee r$

- B6: Danh sách mệnh đề mới:

$$\{\neg p \vee r, \neg r \vee s, \neg u \vee \neg s, p, u\}$$

chưa có cặp mệnh đề đối ngẫu

- Chọn cặp mệnh đề $\neg p \vee r \vee \neg r \vee s \Rightarrow \neg p \vee s$

- Danh sách mệnh đề mới: $\{\neg p \vee s, \neg u \vee \neg s, p, u\}$ chưa có cặp mệnh đề đối ngẫu

Thuật giải Robinson – Ví dụ

$$\{\neg p \vee s, \neg u \vee \neg s, p, u\}$$

- Chọn cặp mệnh đề

$$\neg p \vee s \vee \neg u \vee \neg s \Rightarrow \neg p \vee \neg u$$

- Danh sách mệnh đề mới: $\{\neg p \vee \neg u, p, u\}$ chưa có cặp mệnh đề đối ngẫu

- Chọn cặp mệnh đề $\neg p \vee \neg u \vee p \Rightarrow \neg u$

- Danh sách mệnh đề mới: $\{\neg u, u\}$ có cặp mệnh đề đối ngẫu \rightarrow Biểu thức ban đầu được chứng minh

Bài tập

- Chứng minh:

$$\{p \rightarrow q, q \rightarrow r, r \rightarrow s, p\} \Rightarrow p \wedge s$$

Logic vị từ

- Ký hiệu hằng:

- Hằng của ngôn ngữ: true, false

- Hằng do người sử dụng đặt cho tên đối tượng cụ thể: An, Binh, ..., a, b, c, ... (đối tượng là các chủ ngữ hoặc tân ngữ trong mệnh đề)

- Ký hiệu biến (thường là biến đối tượng, đại diện cho chủ ngữ hoặc tân ngữ): x, y, z, t, u, ...

- Ký hiệu vị từ: P, Q, ... hoặc Sinhvien, Yeu, father, ... (mỗi ký hiệu tương ứng vị từ trong mệnh đề). Mỗi ký hiệu vị từ là câu đơn trong logic cấp một và có ngữ nghĩa true hay false.

Logic vị từ

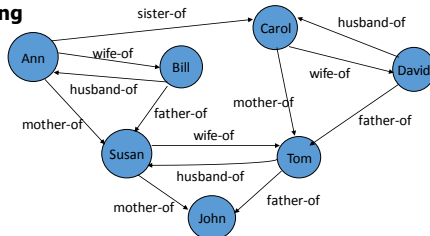
- Ký hiệu hàm: sin, cos, log, father, ...
- Ký hiệu kết nối logic: \neg , \wedge , \vee , \Rightarrow , \Leftrightarrow
- Ký hiệu lượng từ: \forall , \exists
- Các ký hiệu "(", ")", " " và " , "

Logic vị từ - Ví dụ

- "An là sinh viên"
 $\text{Sinhvien}(\text{An})$
- "Nam là cha của Hoàn"
 $\text{Cha}(\text{Nam}, \text{Hoàn})$
- "Mọi sinh viên đều học giỏi"
 $\forall x \text{Sinhvien}(x) \Rightarrow \text{Hocgioi}(x)$
- "Trong sinh viên có bạn học giỏi"
 $\exists x \text{Sinhvien}(x) \wedge \text{Hocgioi}(x)$

Mạng ngữ nghĩa – Semantic Network

Mạng ngữ nghĩa là một đồ thị có hướng được gán nhãn, trong đó **mỗi nút thể hiện một đối tượng (một mệnh đề)**, và mỗi **liên kết thể hiện mối quan hệ giữa 2 đối tượng**



Mạng ngữ nghĩa

- **"IS-A"** (là một thể hiện của). Đề cập đến một thành viên của một class, trong đó một class là một nhóm đối tượng với một hay nhiều thuộc tính (properties). Ví dụ, "Tom IS-A bird".
- **"A-KIND-OF"**. Đề cập một class với một class khác. Ví dụ, "Birds are A-KIND-OF animals".
- **"HAS-A"**. Thuộc tính liên quan đến đối tượng. Ví dụ, "Mary HAS-A cat".
- **"CAUSE"**. Thể hiện quan hệ nhân quả. Ví dụ, "Fire CAUSES smoke".

Đặc điểm của mạng ngữ nghĩa

- Ưu điểm:
 - Mô hình thực thi rất đơn giản
 - Thể hiện rất dễ đọc giúp nó dễ dàng thể hiện các bước suy luận
- Mở rộng mạng ngữ nghĩa:
 - Chỉ cần thêm các đỉnh và các cung quan hệ với các đỉnh có sẵn. Các đỉnh được thêm vào mạng hoặc là biểu diễn đối tượng hoặc là biểu diễn thuộc tính
- Tính thừa kế:
 - Quan hệ đặc biệt "IS-A" để chỉ ra sự thừa kế
- Tính ngoại lệ:
 - Định nghĩa một cung quan hệ mới đến một đỉnh có trị khác

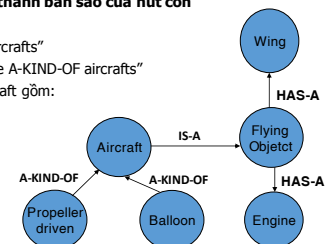
Suy luận trong mạng ngữ nghĩa

Thủ tục suy luận trong mạng ngữ nghĩa được gọi là thừa kế, và nó cho phép đặc điểm của một nút thành bản sao của nút con

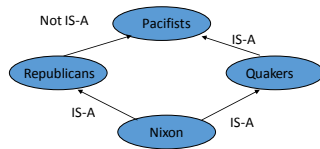
Ví dụ

"Balloons are A-KIND-OF aircrafts"
 "Propeller-driven objects are A-KIND-OF aircrafts"
 Giả sử các thuộc tính của aircraft gồm:
 - "Aircraft IS-A flying object"
 - "Aircraft HAS-A wings"
 - "Aircraft HAS-A engines"

Tất cả các thuộc tính của lớp cha (superclass): "aircraft" sẽ được kế thừa bởi các lớp con



Đa thừa kế có thể dẫn đến suy luận trái chiều nhau

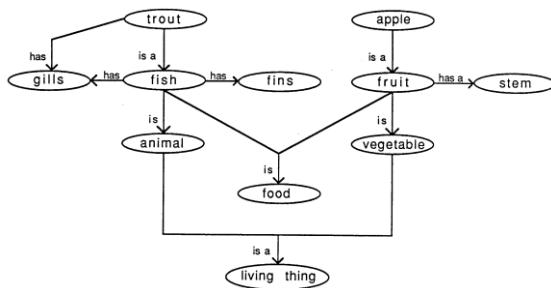


<https://www.nodebox.net/perception>

Ví dụ - Xây dựng mạng ngữ nghĩa

- (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.

Ví dụ - Xây dựng 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 .
- (12) A DOG IS A PET .
- (13) A DOG BARKS .
- (8) A FLAG IS AN ARTIFACT .
- (9) A CAT IS A MAMMAL .
- (10) A MAMMAL IS AN ANIMAL .
- (11) A DOG IS A MAMMAL .

Bộ ba Object-Attribute-Value

Một vấn đề với lưới ngữ nghĩa là **không có định nghĩa chuẩn của tên liên kết**. Để tránh sự **nhập nhằng** này, chúng ta có thể hạn chế các hình thức này thành một loại đơn giản của mạng ngữ nghĩa, chỉ có hai loại liên kết

"HAS-A" và "IS-A"

Hình thức như vậy được gọi là bộ ba Object-Attribute-Value (OAV) và là phương thức được sử dụng rộng rãi của biểu diễn tri thức (đặc biệt là đại diện cho tri thức khai báo)

Đối tượng - Thuộc tính – Giá trị

- 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).

Bộ ba Object-Attribute-Value

Ví dụ: Xét đối tượng “airplane”. Một vài thuộc tính là:

- Số động cơ
- Loại động cơ
- Loại thiết kế cánh quạt

Những giá trị có thể có của các thuộc tính này là:

- Số động cơ: 2, 3, 4.
- Loại động cơ: jet, propeller-driven.
- Loại thiết kế cánh quạt: conventional, swept-back.

Vấn đề của mạng ngữ nghĩa và bộ ba OAV

• **Không có định nghĩa chuẩn của liên kết và nút tên.** Điều này khó khăn để hiểu được mạng, và có hay không được thiết kế một cách hợp lý.

• Thừa kế là một tìm kiếm bùng nổ tổ hợp, đặc biệt là nếu các phản hồi cho truy vấn là phủ định. Thêm vào đó, cơ chế suy luận chỉ được xây dựng trong mạng ngữ nghĩa, có thể không đủ cho một số ứng dụng.

Vấn đề của mạng ngữ nghĩa và bộ ba OAV

• Mạng ngữ nghĩa có tính logic và heuristic rất yếu. Những phát biểu như “Một số sách thú vị nhiều hơn so với những sách khác”, “Không có cuốn sách về chủ đề này”, “Nếu một cuốn sách viễn tưởng được yêu cầu, không quan tâm các cuốn sách về lịch sử, y tế và toán học” không thể hiện được trong mạng ngữ nghĩa.

Lưu đồ về quan hệ phụ thuộc khái niệm

- 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:
 - 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)

Lưu đồ về quan hệ phụ thuộc khái niệm.

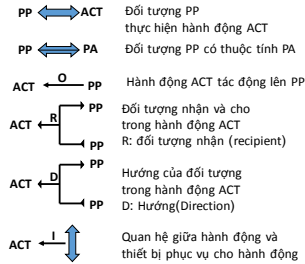
• Các hành động có thể được mô tả bằng cách phân rẽ về một/ nhiều hành động (ACT):

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 trng giác quan – VD: *lắng nghe, nhìn,...*

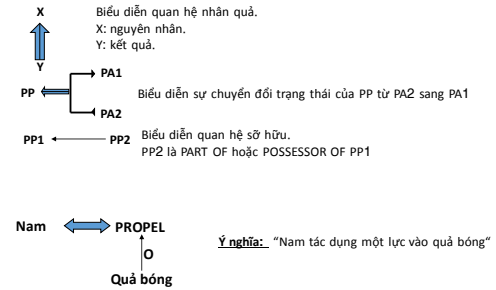
Lưu đồ về quan hệ phụ thuộc khái niệm

- Quan hệ phụ thuộc khái niệm bao gồm một **tập các luật cú pháp cho khái niệm, hình thành nên văn phạm về quan hệ ngữ nghĩa.**
- Các quan hệ này sẽ được dùng vào việc biểu diễn bên trong cho câu trong ngôn ngữ tự nhiên

Lưu đồ về quan hệ phụ thuộc khái niệm



Lưu đồ về quan hệ phụ thuộc khái niệm



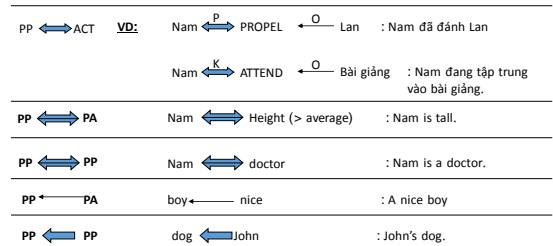
Lưu đồ về quan hệ phụ thuộc khái niệm

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)

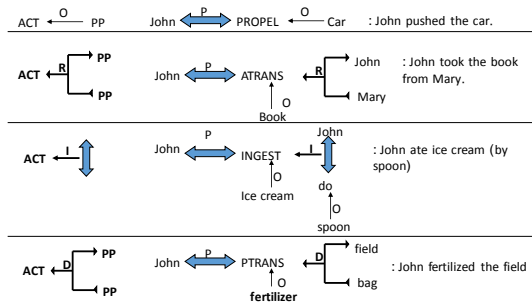
$Nam \xleftrightarrow{P} PROPEL \xleftarrow{O} \text{Cái bàn}$

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

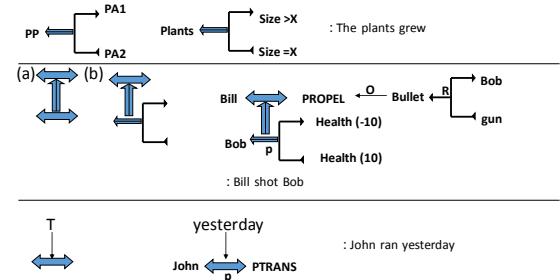
Một số ví dụ



Một số ví dụ

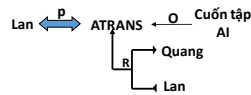


Một số ví dụ

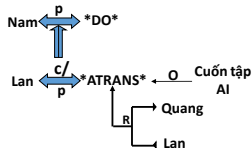


Một số ví dụ

Lan gửi cuốn tập cho Quang



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



Lưu đồ về quan hệ phụ thuộc khái niệm

Nhược điểm

- Khó khăn trong *việc phát triển chương trình để tự động thu giảm biểu diễn của câu bất kỳ về dạng quy tắc chuẩn*
- Chi phí phân rã về các thành phần cơ bản: ACT, PP, ...
- Các thành phần cơ bản không thích hợp để miêu tả những khái niệm tinh tế của ngôn ngữ tự nhiên, như các từ có nghĩa định tính: cao, đẹp, ...

Lưu đồ về quan hệ phụ thuộc khái niệm

Ưu điểm

- Biểu diễn hình thức cho ngữ nghĩa của ngôn ngữ tự nhiên theo dạng có quy tắc → giảm sự nhập nhằng
- Tính **đồng nghĩa** tương ứng là sự **đồng nhất về cú pháp** của lược đồ biểu diễn → chứng minh tính đồng nghĩa bằng cách so trùng hai đồ thị biểu diễn

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 là khái niệm**
- Đỉnh quan hệ**: dùng để chỉ ra quan hệ giữa các khái niệm có nối đến nó

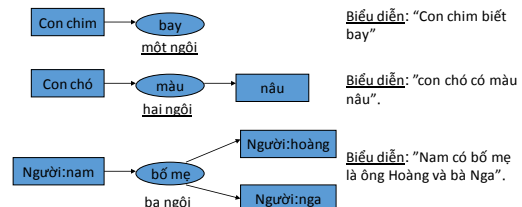
Đồ thị khái niệm

Trong đồ thị khái niệm: **chỉ có các đỉnh khác loại mới nối được với nhau**. Chính vì vậy các cung không cần phải được gán nhãn nữa

- Mỗi đồ thị khái niệm biểu diễn một mệnh đề đơn
- Cơ sở tri thức: chứa nhiều đồ thị khái niệm

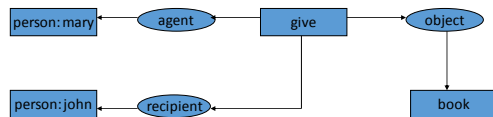
Đồ thị khái niệm

Một số ví dụ:



* Một đỉnh quan hệ có thể là một hay nhiều ngôi

Đồ thị khái niệm



“Mary give John the book”

Đồ thị khái niệm

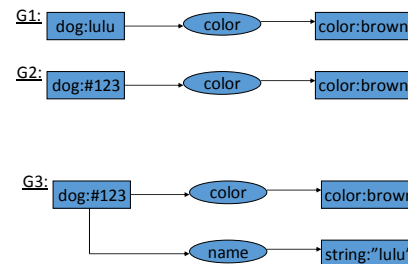
Trong đồ thị khái niệm, **mỗi đỉnh quan hệ biểu diễn cho một cá thể đơn lẻ thuộc một loại nào đó**. Để nói lên quan hệ giữa “loại-cá thể”, mỗi đỉnh khái niệm được quy định cách gán nhãn:

“loại:tên_cá_thể”

Đồ thị khái niệm – Tên cá thể

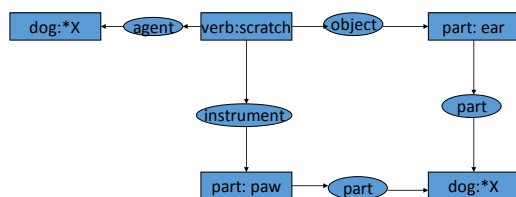
- Một tên nào đó
sinhviên: nam → một sinh viên có tên là Nam
- Một khoá để phân biệt, được viết theo cú pháp **#khóa**
sinhviên: #59701234 → một sinh viên có khóa là: 59701234
- Có thể dùng dấu sao (*) để chỉ ra một cá thể chưa xác định (*khái niệm tổng quát*)
sinhviên:* → chỉ ra một sinh viên bất kỳ
sinhviên:*X → sinh viên bất kỳ, tên sinh viên đã được lấy qua biến X
sinhviên:ng* → sinh viên có tên bắt đầu bởi “ng”

Đồ thị khái niệm



Đồ thị khái niệm

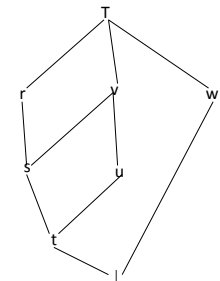
- Biến có thể được dùng khi cần chỉ ra nhiều đỉnh khái niệm đồng nhất nhau trong một đồ thị



“The dog scratches its ear with its paw”

Đồ thị khái niệm – Phân cấp

- s: được gọi là **common-subtype** của r và v.
- v: được gọi là **common-supertype** của s và u.
- T: supertype của mọi type
- ⊥: subtype của mọi type



Đồ thị khái niệm – Các phép toán

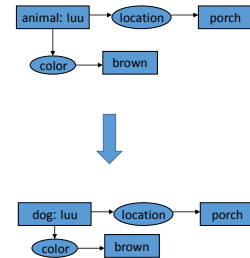
[1] Phép copy (nhân bản): nhân bản một đồ thị

[2] Phép Restriction (giới hạn): tạo ra đồ thị mới bằng cách: từ một đồ thị đã có, thay thế một đỉnh khái niệm bởi một đỉnh khác cụ thể hơn, như hai trường hợp:

→ Một biến *, được thay thế bởi một khoá, hay một tên của cá thể. VD: *dog:** → *dog:#123* hay *dog:luu*

→ Một type được thay thế bởi subtype của nó. VD:
người: nam → *sinhviên: nam*

Đồ thị khái niệm – Các phép toán

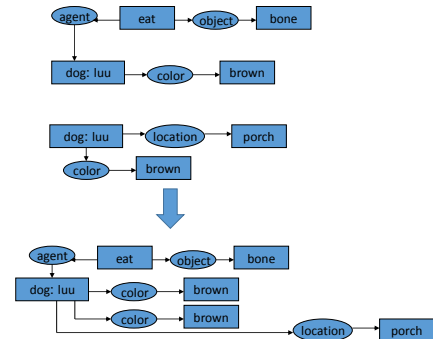


Đồ thị khái niệm – Các phép toán

[3] Phép Join (nối): Nối hai đồ thị để được một đồ thị khác.=

Nếu có đỉnh khái niệm C xuất hiện trên cả hai đồ thị X và Y, thì chúng ta có thể nối hai đồ thị trên đỉnh chung C nói trên

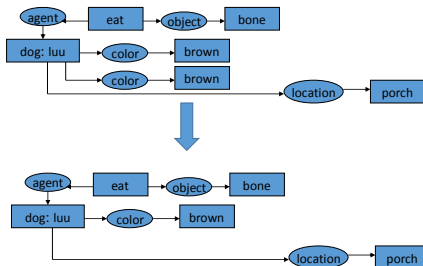
Đồ thị khái niệm



Đồ thị khái niệm – Các phép toán

[4] Phép simplify (rút gọn)

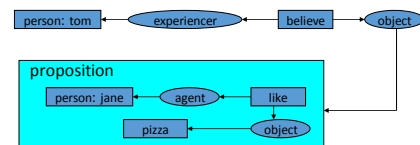
Nếu trên một đồ thị có hai đồ thị con giống nhau hoàn toàn thì chúng ta có thể bỏ đi một để tạo ra một đồ thị mới có khả năng biểu diễn không thay đổi



Đồ thị khái niệm – Định mệnh đề

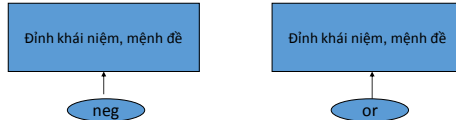
• Định khái niệm có thể chứa một mệnh đề → định mệnh đề

"Tom believes that Jane likes pizza"



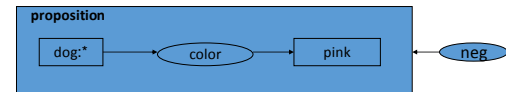
Đồ thị khái niệm - Logic

- Phép hội (and) của nhiều khái niệm, mệnh đề chúng ta có thể thực hiện bởi phép toán join
- Phép phủ định (not) và phép tuyển (or) giữa các khái niệm hay mệnh đề được thể hiện bằng các đỉnh quan hệ: neg (phủ định), or (tuyển)



Đồ thị khái niệm - Logic

"There are no pink dogs"



$$\neg \exists X \exists Y (\text{dog}(X) \wedge \text{color}(X, Y) \wedge \text{pink}(Y))$$

$$= \forall X \forall Y (\neg (\text{dog}(X) \wedge \text{color}(X, Y) \wedge \text{pink}(Y)))$$

Đồ thị khái niệm - Giải thuật

1. Gán một biến riêng biệt (X1, X2,...) cho mỗi khái niệm tổng quát.
2. Gán một hằng cho mỗi khái niệm cá thể trong đồ thị. Hằng này có thể là tên cá thể hay khoá của nó
3. Biểu diễn một đỉnh khái niệm bởi một vị từ một ngôi; có tên là tên loại (type), đối số là biến hay hằng vừa gán trên
4. Biểu diễn mỗi đỉnh quan hệ bởi một vị từ n ngôi; có tên là tên của đỉnh quan hệ, các thông số là biến hay hằng được gán cho các đỉnh khái niệm nối đến nó
5. Hội của tất cả các câu trong bước 3 và 4. Tất cả các biến trong biểu thức thu được đều đính kèm lượng từ tồn tại

Đồ thị khái niệm - Giải thuật



$$\exists X1 (\text{dog}(\text{lulu}) \wedge \text{color}(X1) \wedge \text{brown}(X1))$$

BIỂU DIỄN TRI THỨC BẰNG 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.
- Frame là nguồn gốc của lập trình hướng đối tượng.
- Một frame bao hàm trong nó một khối lượng tương đối lớn tri thức về một đối tượng, sự kiện, vị trí, tình huống hoặc những yếu tố khác

CẤU TRÚC CỦA FRAME

Mỗi một frame mô tả một *đối tượng (object)*. Một frame bao gồm 2 thành phần cơ bản là **slot** và **facet**. Một **slot** là một thuộc tính đặc tả đối tượng được biểu diễn bởi frame.

Ví dụ : trong frame mô tả xe hơi, có hai slot là *trọng lượng* và *loại*

Ví dụ Frame

Một số facet thường gặp.

Value: giá trị. Cho biết giá trị của thuộc tính đó

Default: giá trị mặc định

Range: miền giá trị

If added : mô tả một hành động sẽ được thi hành khi một giá trị trong slot được thêm vào (hoặc được hiệu chỉnh). Thủ tục thường được viết dưới dạng một script.

If needed : được sử dụng khi slot không có giá trị nào. Facet mô tả một hàm để tính ra giá trị của slot.

Frame MÁY
Xy-lanh : 3.19 inch
Tỷ lệ nén : 3.4 inche
Xăng : TurboCharger
Mã lực : 140 hp

BIỂU DIỄN TRI THỨC BẰNG SCRIPT

Script là một cách biểu diễn tri thức tương tự như frame

nhưng thay vì đặc tả một đối tượng, nó mô tả *một chuỗi các*

sự kiện. Để mô tả chuỗi sự kiện, script sử dụng một dãy các

slot chứa thông tin về các con người, đối tượng và hành

động liên quan đến sự kiện đó.

Các thành phần của Script

Điều kiện vào (entry condition): mô tả những tình huống hoặc điều kiện cần được thỏa mãn trước khi các sự kiện trong script có thể diễn ra.

Role (diễn viên): là những con người có liên quan trong script.

Prop (tác tố): là tất cả những đối tượng được sử dụng trong các chuỗi sự kiện sẽ diễn ra.

Scene(Tình huống) : là chuỗi sự kiện thực sự diễn ra.

Result (Kết quả) : trạng thái của các Role sau khi script đã thi hành xong.

Track (phiên bản) : mô tả một biến thể (hoặc trường hợp đặc biệt) có thể xảy ra trong đoạn script.

Phiên bản : Nhà hàng bán thức ăn nhanh.

Diễn viên : Khách hàng, Người phục vụ.

Tác tố : Bàn phục vụ. Chỗ ngồi. Khay đựng thức ăn ...

Điều kiện vào : Khách hàng đóiKhách hàng có đủ tiền để trả.

Tình huống 1 : Vào nhà hàng

Tình huống 2 : Kêu món ăn.

Tình huống 3 : Khách hàng dùng món ăn

Tình huống 4 : Ra về

Kết quả : Khách hàng không còn đói.

Khách hàng còn ít tiền hơn ban đầu.

Khách hàng vui vẻ *

Khách hàng bức mình *

Khách hàng quá no

PHỐI HỢP NHIỀU CÁCH BIỂU DIỄN TRI THỨC

P.Pháp	Ưu điểm	Nhược điểm
Luật sinh	Cú pháp đơn giản, dễ hiểu, diễn dịch đơn giản, tính đơn thể cao, linh động (dễ điều chỉnh).	Rất khó theo dõi sự phân cấp, không hiệu quả trong những hệ thống lớn, không thể biểu diễn được mọi loại tri thức, rất yếu trong việc biểu diễn các tri thức dạng mô tả, có cấu trúc.
Mạng ngữ nghĩa	Dễ theo dõi sự phân cấp, sẽ dò theo các mối liên hệ, linh động	Ngữ nghĩa gắn liền với mỗi đỉnh có thể nhập nhằng, khó xử lý các ngoại lệ, khó lập trình.

PHỐI HỢP NHIỀU CÁCH BIỂU DIỄN TRI THỨC(tt)

Frame	Có sức mạnh diễn đạt tốt, dễ cài đặt các thuộc tính cho các slot cũng như các mối liên hệ, dễ dàng tạo ra các thủ tục chuyên biệt hóa, dễ đưa vào các thông tin mặc định và dễ thực hiện các thao tác phát hiện các giá trị bị thiếu sót.	Khó lập trình, khó suy diễn, thiếu phần mềm hỗ trợ.
Logic hình thức	Cơ chế suy luận chính xác (được chứng minh bởi toán học).	Tách rời việc biểu diễn và xử lý, không hiệu quả với lượng dữ liệu lớn, quá chậm khi cơ sở dữ liệu lớn

Tài liệu tham khảo

Nội dung slide có sử dụng các hình ảnh lấy từ Google và một số slide AI của các course và các Thầy/ Cô

[1] <https://www.udacity.com/course/intro-to-artificial-intelligence--cs271>

[2] Slide môn TTNT Thầy Minh Thái

[3] Slide môn TTNT Thầy Trương Hải Bằng



134