

EXPERT SYSTEMS

BIỂU DIỄN TRI THỨC

PGS.TS. Hoàng Văn Dũng
dunghv@hcmute.edu.vn

Biểu diễn tri thức

Nguyên lý làm việc như sau:

Tri thức được thu nhận trong ngôn ngữ tự nhiên L

Tri thức được dịch ra trong ngôn ngữ P

Tìm giải pháp suy diễn trong P

Lời giải được dịch ra trong ngôn ngữ L

Biểu diễn tri thức

Tri thức của XPS có thể được biểu diễn theo nhiều cách khác nhau như:

- Biểu diễn tri thức bởi các luật sản xuất (LSX)

- Biểu diễn tri thức nhờ mệnh đề logic

- Biểu diễn tri thức nhờ mạng ngữ nghĩa

- Biểu diễn tri thức nhờ ngôn ngữ nhân tạo

Tri thức cũng có thể biểu diễn nhờ các sự kiện không chắc chắn, nhờ bộ ba:

- Đối tượng,

- Thuộc tính và giá trị (O-A-V: Object-Attribute-Value),

- Khung (frame), v.v...

Biểu diễn tri thức bởi các luật sản xuất

Hầu hết các hệ chuyên gia đều là các hệ thống dựa trên luật vì:

- ☀ *Bản chất đơn thể* (modular nature)
 - ❖ Có thể đóng gói tri thức
 - ❖ Có thể mở rộng XPS một cách dễ dàng
- ☀ *Khả năng diễn giải dễ dàng* (explanation facilities)
 - ❖ Vận dụng luật có thể đặc tả chính xác các yếu tố tri thức
 - ❖ Dễ dàng diễn giải vấn đề nhờ các tiền đề
 - ❖ Rút ra được kết quả mong muốn
- ☀ *Tương tự quá trình nhận thức của con người.*
 - ❖ Dựa trên các công trình của Newell và Simon, các luật được xây dựng từ cách con người giải quyết vấn đề
 - ❖ Cách biểu diễn luật nhờ IF THEN đơn giản cho phép giải thích dễ dàng cấu trúc tri thức cần trích lọc

Biểu diễn tri thức bởi các luật sản xuất

Các luật sản xuất thường được viết dưới 2 dạng IF THEN.

IF < điều kiện > THEN <hành động>

hoặc

IF < điều kiện > THEN <kết luận> DO <hành động>

Luật là một kiểu sản xuất dạng : LHS →
RHS

Phần trái LHS (Left Hand Side)

có nội dung được gọi theo nhiều tên khác nhau, như :

- ❖ tiền đề (antecedent)
- ❖ điều kiện (conditional part)
- ❖ mẫu so khớp (pattern part)

Phần sau luật RHS (Right Hand Side) là kết luận hay *hậu quả* (consequent)

Biểu diễn tri thức bởi các luật sản xuất

Ví dụ :

Rule: Đèn đỏ

IF Đèn đỏ sáng THEN Dừng

Rule: Đèn-xanh

IF Đèn xanh sáng THEN Đi

Rule: Điều trị sốt

IF

Bệnh nhân sốt

THEN

cho uống thuốc Aspirin

Biểu diễn tri thức bởi các luật sản xuất

Hệ thống chẩn đoán xe máy (OPS5)

IF Máy xe không nổ khi khởi động

THEN Dự đoán: Xe bị panne sức nén. Pittong, bạc xéc-măng và lòng xi lanh sai tiêu chuẩn, dễ tạo thành những khe hở nhỏ làm cho pittong không còn kín nên hoà khí không được nén lên đầy đủ.

Xử lý : nên điều chỉnh/thay mới pittong, bạc xéc-măng và lòng xi lanh cho đúng tiêu chuẩn

IF máy xe nổ không ổn định,
 OR máy xe nổ rồi lại tắt,
 AND bugi khô

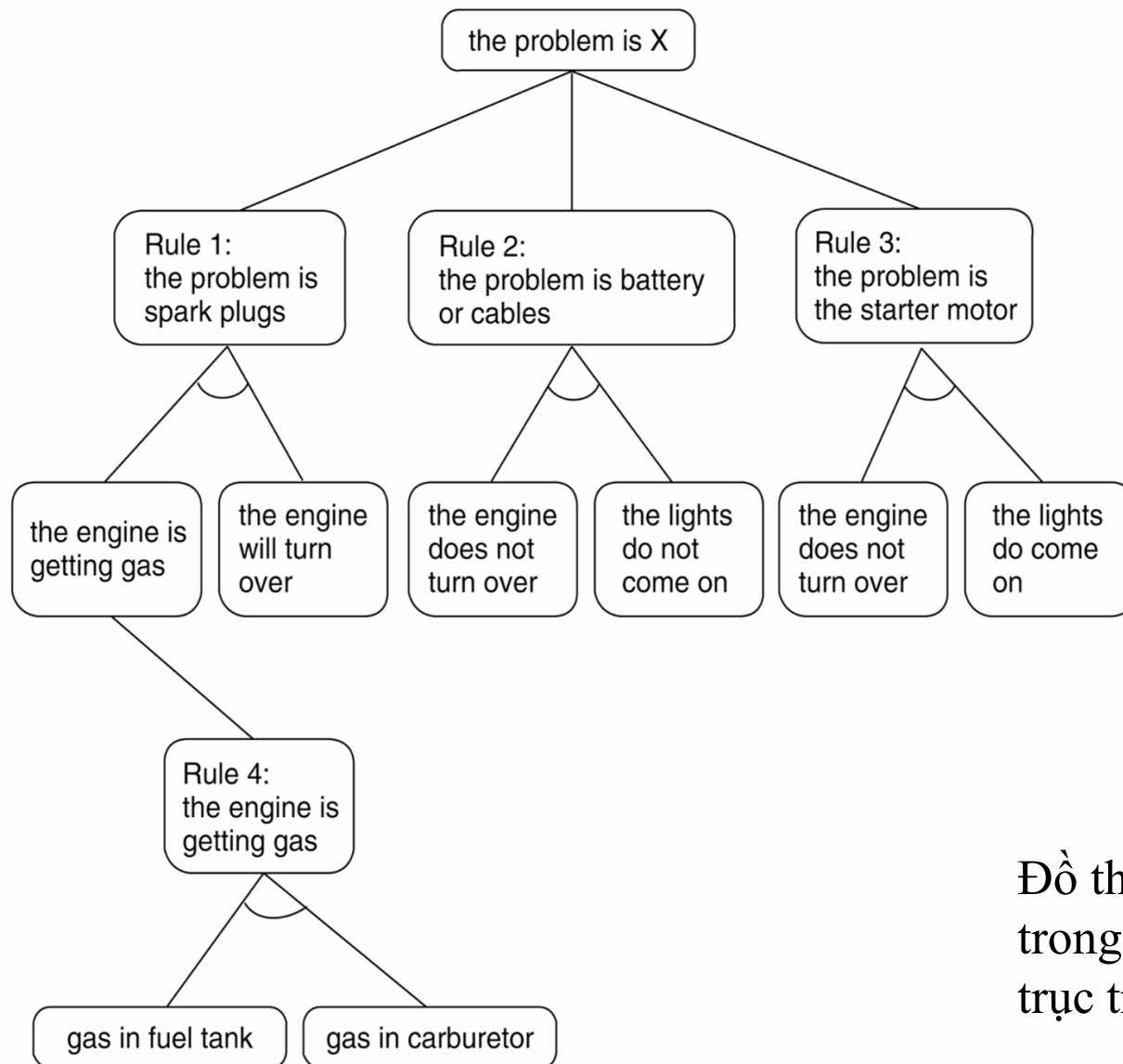
THEN Dự đoán : Xe đã bị nghẹt xăng.

Xử lý : nên xúc rửa bình xăng và bộ khoá xăng của xe.

Ví dụ một XPS dựa trên luật

| | |
|--------|---|
| Luật 1 | IF động cơ nhận được xăng AND động cơ khởi động được THEN trục trặc là do bugi. |
| Luật 2 | IF động cơ không khởi động được AND đèn không sáng THEN trục trặc là do ắc quy hoặc dây cáp |
| Luật 3 | IF động cơ không khởi động được AND đèn sáng THEN trục trặc là do mô-tơ khởi động |
| Luật 4 | IF còn xăng trong bình chứa nhiên liệu AND còn xăng trong bộ chế hòa khí THEN động cơ nhận được xăng |

Ví dụ một XPS dựa trên luật



Đồ thị Và/Hoặc tìm được
trong ví dụ chẩn đoán
trục trặc xe hơi

Khả năng giải thích và tính trong sáng

còn xăng trong bình chứa nhiên liệu?

yes

còn xăng trong bộ chế hòa khí?

yes

động cơ khởi động được không?

why

Nó đã suy luận được:

1. động cơ nhận được xăng

Vì vậy nếu

2. động cơ khởi động được

Thì có thể kết luận: trục trặc là do bugi

động cơ nhận được xăng như thế nào?

Điều này có từ luật 4

IF còn xăng trong bình chứa nhiên liệu

AND còn xăng trong bộ chế hòa khí

THEN động cơ nhận được xăng

Người dùng cho biết: còn xăng trong bình chứa nhiên liệu

Người dùng cho biết: còn xăng trong bộ chế hòa khí

Giới hạn của XPS dựa trên luật

Tri thức bề nổi
Các luật heuristic

Tri thức sâu
Lý thuyết chuyên ngành
+
Tri thức giải quyết vấn đề

Bộ sinh của XPS

Bộ sinh của hệ chuyên gia (expert-system generator) là hợp của :

- một máy suy diễn,

- một ngôn ngữ thể hiện tri thức (bên ngoài)

- và một tập hợp các cấu trúc và các quy ước thể hiện các tri thức (bên trong)

Nhờ các tri thức chuyên môn để định nghĩa một hệ chuyên gia, người ta đã tạo ra bộ sinh để làm đầy cơ sở tri thức

Bộ sinh của XPS

Nhờ bộ sinh, mỗi hệ chuyên gia có thể chứa từ hàng trăm đến hàng ngàn luật.

Bảng thống kê số luật của một số hệ chuyên gia :

| Hệ chuyên gia | Lĩnh vực | Năm xuất hiện | Số luật |
|---------------|----------|---------------|---------|
| MYCIN | Y học | 1974 | 500 |
| PROSPECTOR | Địa chất | 1979 | 1600 |
| R1/XCON | Tin học | 1980 | > 7 000 |
| LITHO | Địa chất | 1982 | 500 |
| SPHINX | Y học | 1984 | 400 |
| TOM | Nông học | 1984 | 200 |

R1/XCON

Mục đích: Tạo cấu hình hệ thống VAX-11/780 của công ty DEC

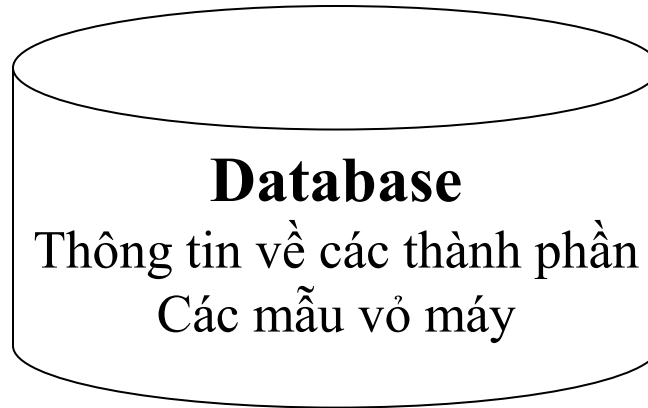
Đầu vào: Đơn đặt hàng = danh sách các thành phần cấu hình nên hệ thống.

Đầu ra: Sơ đồ cấu hình

Kết quả: cấu hình 97% các đơn đặt hàng của DEC

Công việc tạo cấu hình của R1/XCON có thể được xem như là một hệ thống phân cấp các công việc nhỏ hơn với *sự phụ thuộc thời gian (temporal dependency)* rất mạnh.

Kiến trúc của XCON



Cơ sở các luật

Các luật ‘điều hành’
Các luật chuyển đổi theo tình huống

Bộ nhớ làm việc

Các ký hiệu thành phần
Các cấu hình chưa hoàn chỉnh
Các ký hiệu tình huống

OPS5

Động cơ suy diễn

Đặc biệt: chọn luật có điều kiện trùng khớp với yếu tố mới nhất
trong bộ nhớ làm việc

«Soạn thảo kết hợp» các luật

Tuỳ theo XPS mà những quy ước để tạo ra luật cũng khác nhau. Sự giống nhau cơ bản giữa các hệ chuyên gia về mặt ngôn ngữ là cách soạn thảo kết hợp (associative writing) các luật.

«Soạn thảo kết hợp» các luật

Soạn thảo kết hợp các luật gồm những quy ước

Mỗi luật do chuyên gia cung cấp phải định nghĩa được các điều kiện khởi động (tác nhân/tiền đề) của luật, và kết quả của luật, để luật này có thể áp dụng.

Trong luật, không bao giờ người ta chỉ định một luật khác bởi tên riêng.

Ví dụ : Luật R tuân thủ hai đặc trưng :

IF bệnh nhân sốt AND tốc độ lắng huyết cầu trong máu tăng lên
THEN bệnh nhân nhiễm bệnh virus

«Soạn thảo kết hợp» các luật

Từ nội dung luật R, người ta có thể vận dụng như sau :

Khi xảy ra tình huống “bệnh nhân sốt” và “tốc độ lắng huyết cầu trong máu tăng lên” là những điều kiện để khởi động luật.

Kết quả của luật là “bệnh nhân nhiễm bệnh virus”.

➤ Như vậy, việc áp dụng luật sẽ dẫn đến một sự kiện mới được thiết lập từ đây trở đi : “bệnh nhân nhiễm bệnh virus”.

Khi muốn tạo sự kiện với điều kiện khởi động luật là “bệnh nhân nhiễm bệnh virus”.

Kết quả của luật sẽ là “bệnh nhân sốt” và “tốc độ lắng huyết cầu trong máu tăng lên”.

➤ Như vậy, luật sẽ khởi động các sự kiện mới vừa được thiết lập “bệnh nhân sốt” và “tốc độ lắng huyết cầu trong máu tăng lên”.

«Soạn thảo kết hợp» các luật

Cách biểu diễn các điều kiện khởi động trong luật phù hợp với cách tư duy tự nhiên của các chuyên gia.

Do vậy, người ta dễ dàng thể hiện cũng như sửa đổi các tri thức tiếp nhận. Không nhất thiết phải đặt tên cho luật để có thể gọi đến khi cần, mà có thể khai thác thông tin từ các điều kiện khởi động của luật.

Nếu tìm được các luật có khả năng thiết lập sự kiện “*bệnh nhân nhiễm bệnh virus*”, người ta sẽ để ý đến phần **THEN** của chúng như là các điều kiện khởi động.

Luật R là một trong các luật có điều kiện khởi động tương ứng với lời gọi “*bệnh nhân nhiễm bệnh virus*”.

Nếu tìm được các luật có khả năng đưa ra sự kiện “*bệnh nhân sốt*”, chỉ cần để ý đến phần **IF** của chúng như là các điều kiện khởi động.

Luật R là một trong các luật có điều kiện khởi động tương ứng với lời gọi “*bệnh nhân sốt*”.

Biểu diễn tri thức nhờ mệnh đề logic

Lôgic mệnh đề

Lôgic vị từ bậc một

Biểu diễn tri thức nhờ logic vị từ bậc một

Dùng logic vị từ biểu diễn tri thức

Trong logic vị từ:

Các vị từ thường có chứa hằng, biến hay hàm

Người ta gọi các vị từ không chứa biến (hoặc hằng) là các *mệnh đề* (preposition)

Mỗi vị từ có thể là một *sự kiện* (fact) hay một luật

Luật là vị từ gồm hai vế trái và phải được nối nhau bởi một dấu mũi tên (\rightarrow)

Người ta sử dụng các ký hiệu và các phép toán logic tác động lên các ký hiệu để thể hiện tri thức và suy luận logic

Dùng logic mệnh đề biểu diễn tri thức

Ví dụ :

- ☀ $\text{MAN}(X), \text{FATHER}(X, Y)$ là các sự kiện
- ☀ $\text{MAN}(X) \rightarrow \text{MORTAL}(X)$ là một luật

Giải thích :

- ☀ $\text{MAN}(X)$: «X là một người»
- ☀ $\text{MORTAL}(X)$: «X chết»

| <i>Phát biểu</i> | <i>Vị từ</i> |
|---------------------------|--|
| Tom là một người | $\text{MAN}(\text{tom})$ |
| Tom là cha của Mary | $\text{FATHER}(\text{tom}, \text{mary})$ |
| Tất cả mọi người đều chết | $\text{MAN}(X) \rightarrow \text{MORTAL}(X)$ |

Ví dụ dùng logic vị từ

Từ các tri thức sau :

- ☀ Marc có tóc vàng hoe
- ☀ Jean có tóc màu nâu
- ☀ Pierre là cha của Jean
- ☀ Marc là cha của Pierre
- ☀ Jean là cha của René
- ☀ Marc là con của Georges.
- ☀ Giả sử X, Y và Z là những người nào đó
- ☀ Nếu Y là con của X, thì X là cha của Y
- ☀ Nếu X là cha của Z và Z là cha của Y, thì X là ông của Y

Ta có thể biểu diễn thành các sự kiện và các luật như sau :

BLOND (marc) BROWN

(jean) FATHER (pierre, jean)

FATHER (marc, pierre)

FATHER (jean, rené)

SON (marc, georges)

FATHER (X, Y) \leftarrow SON (Y, X)

GRANDFATHER (X, Y) \leftarrow
FATHER (X, Z),
FATHER (Z, Y)

Biểu diễn tri thức nhờ mạng ngữ nghĩa

Mạng ngữ nghĩa là một đồ thị :

- ☀ Các **nút** (node) dùng để thể hiện :
 - ❖ Các đối tượng
 - ❖ Thuộc tính của đối tượng
 - ❖ Giá trị của thuộc tính
- ☀ Các **cung** (arc) nối các nút để thể hiện mối quan hệ giữa các đối tượng

Các nút và các cung đều được gắn nhãn

Nghĩa sử dụng của một tri thức :

- ☀ Một đường đi trong đồ thị gồm các nút nối các cung

Semantic Nets

Semantic net is a knowledge presentation method based on a network structure

It consists of

- ✱ points called nodes connected by
- ✱ links called arcs

Nodes - object, concepts, events

Arcs - relationships between nodes

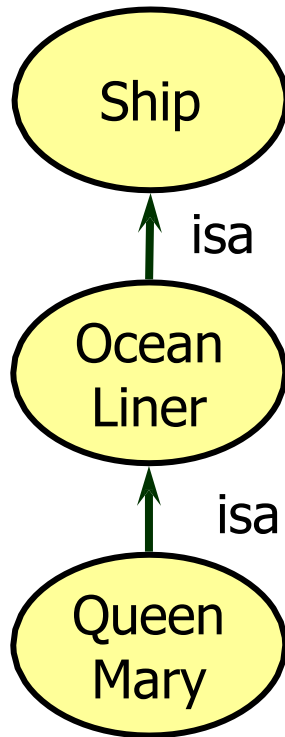
Common arcs used for representing hierarchies include

isa and *has-part*

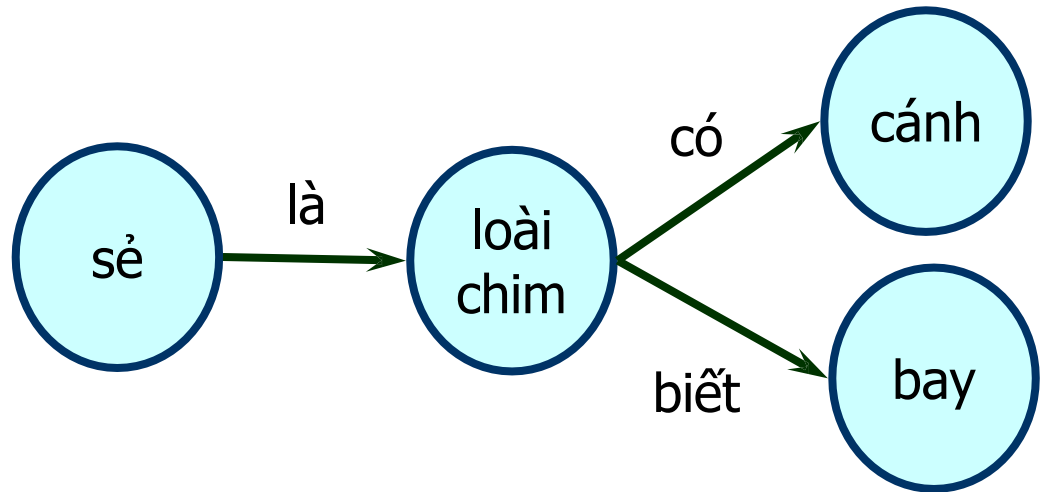
Knowledge represented as a network or *graph*

Semantic Nets

Example: The queen mary is an ocean liner
Every ocean liner is a ship



Sẻ là một loài chim có cánh và biết bay



Tính thừa kế của mạng ngữ nghĩa

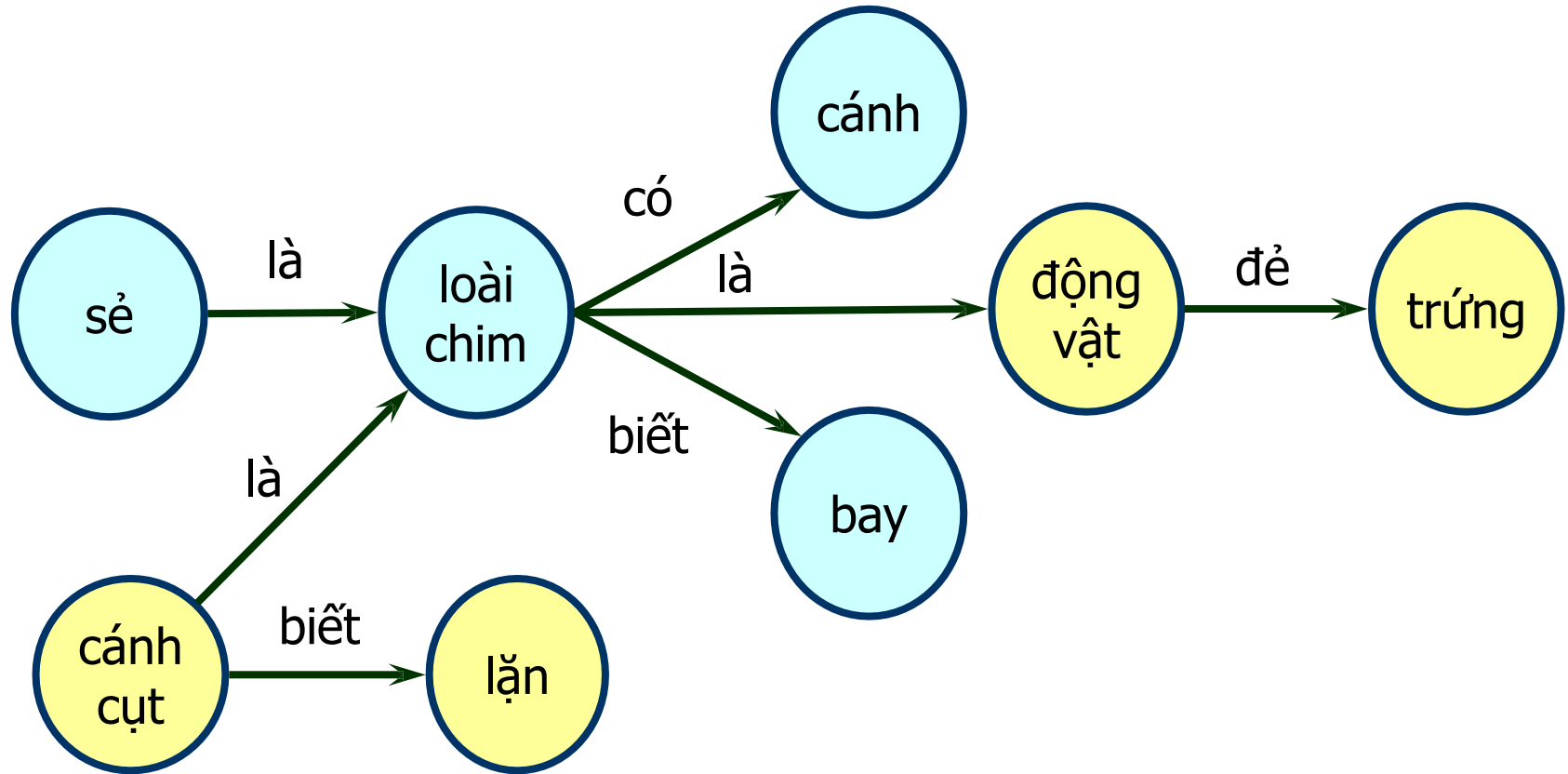
Một trong những tính chất quan trọng của mạng ngữ nghĩa là tính thừa kế

Bằng cách thêm vào đồ thị các nút mới và các cung mới, người ta có thể mở rộng một mạng ngữ nghĩa

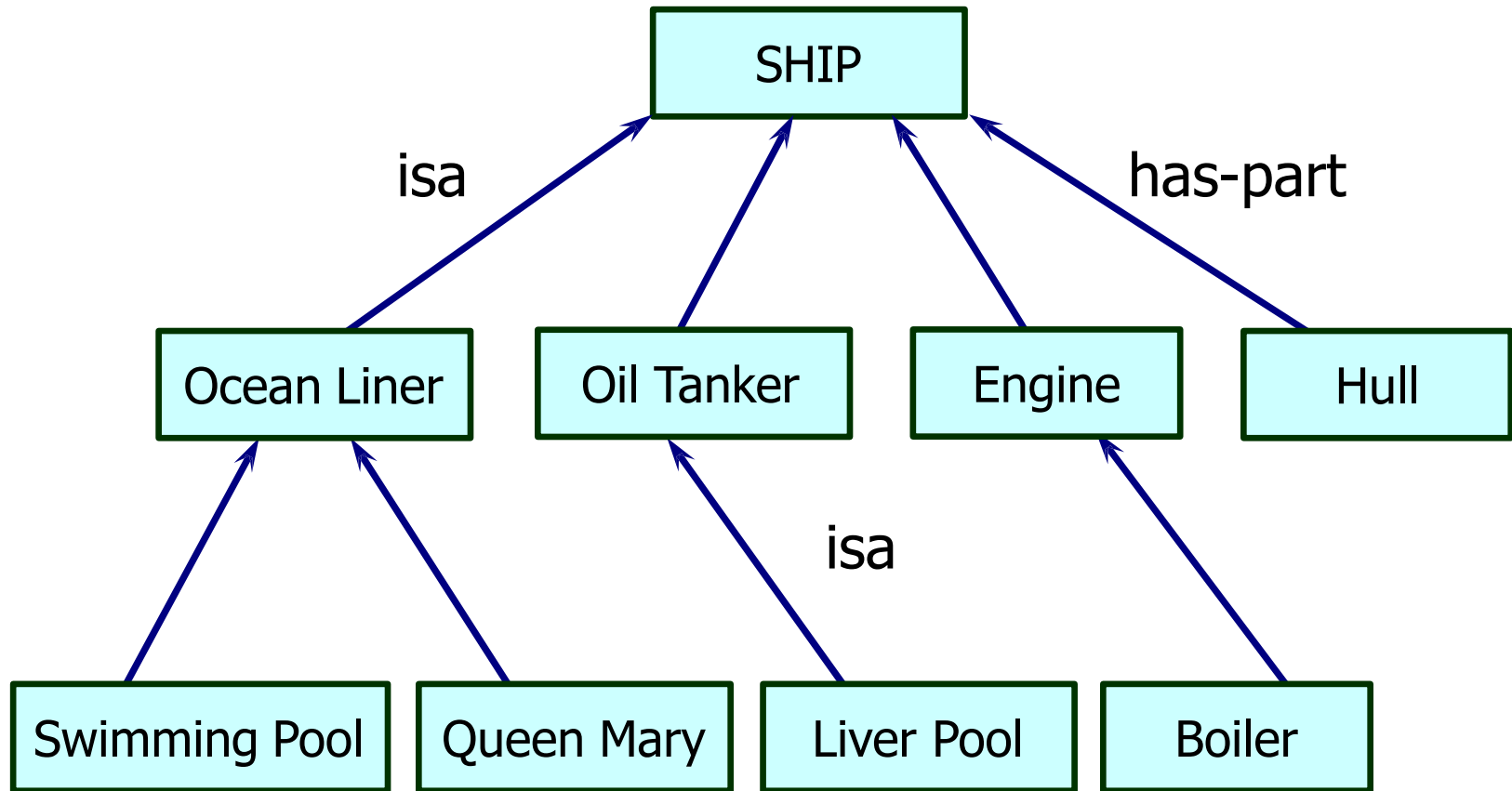
Các nút mới được thêm thể hiện các đối tượng tương tự (với các nút đã có trong đồ thị), hoặc tổng quát hơn

Khi sử dụng mạng ngữ nghĩa để biểu diễn tri thức, ta phải xây dựng các phép toán tương ứng

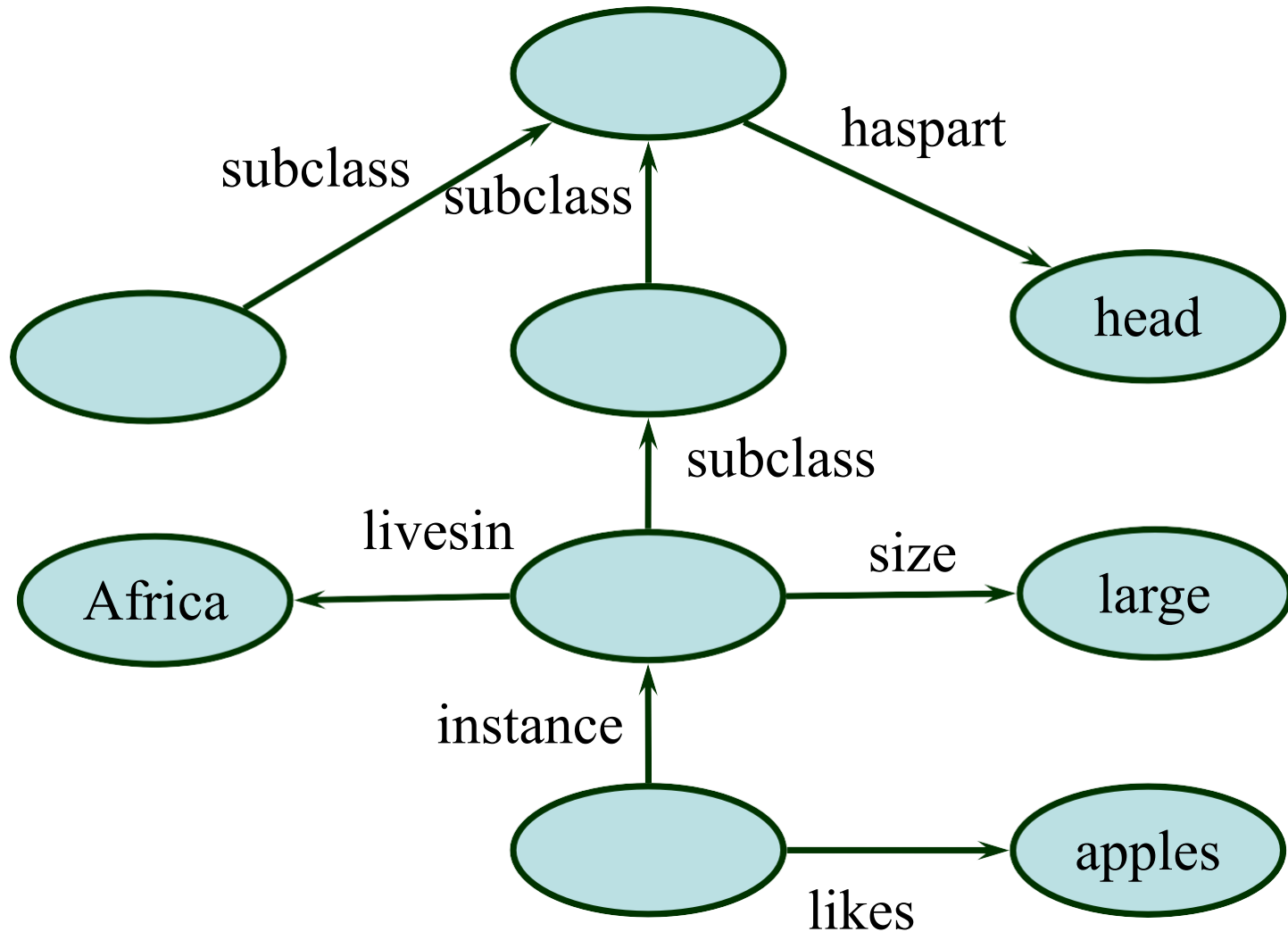
Mở rộng một mạng ngữ nghĩa



Mở rộng một mạng ngữ nghĩa



Mở rộng một mạng ngữ nghĩa



Semantic Networks

Arity of Relations

- ☀ Unary relations

Ex: Person(Jim): IS-A link

- ☀ Binary relations

Ex: Age(Jim, 27 years): Age link

- ☀ N-ary relations

Ex: Disease(Jim, Mumps, 5 days): By creating a reified

- ☀ disease-relation object with several cases :
(patient, diagnosis, duration)

Bài tập

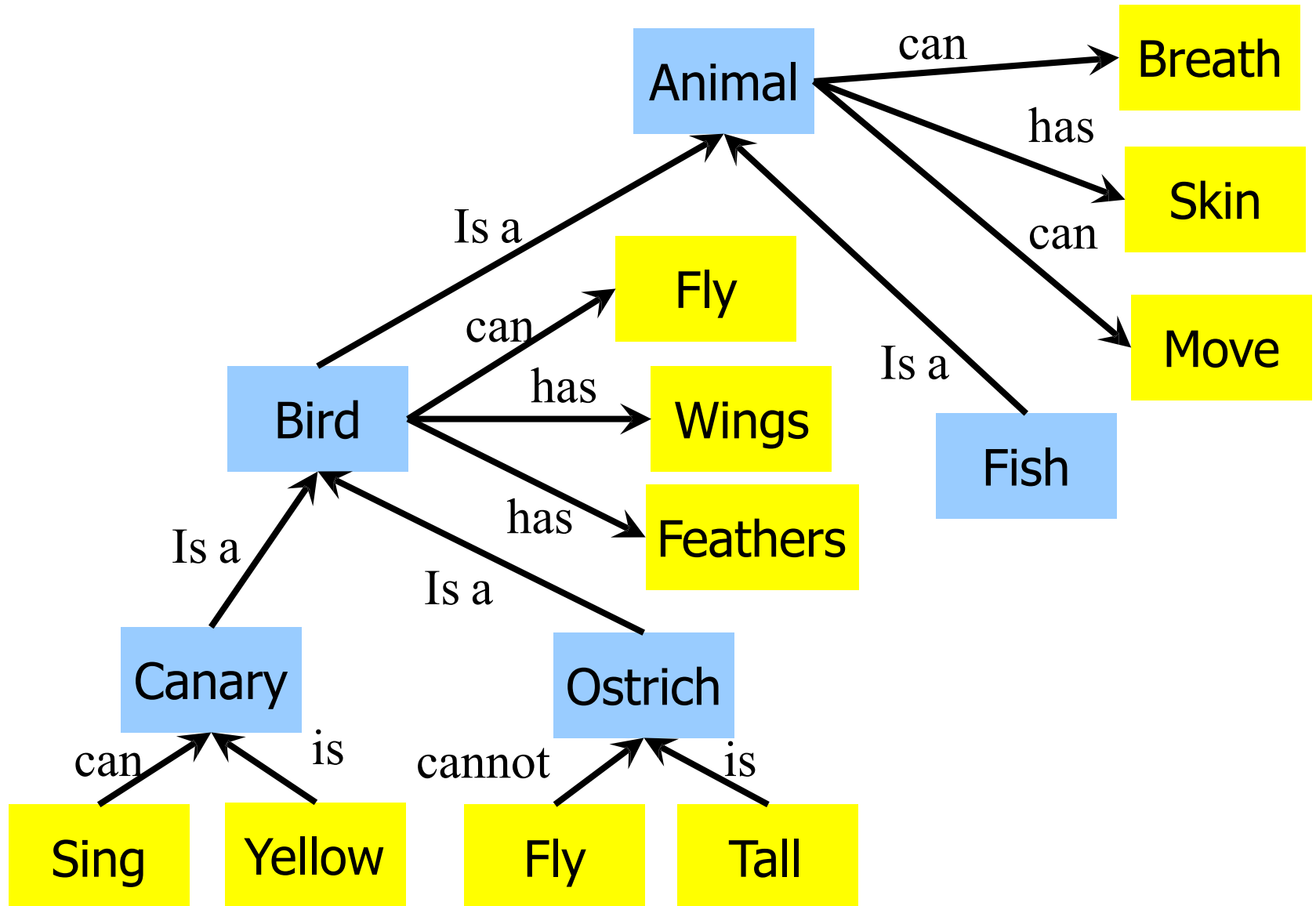
Vẽ mạng ngữ nghĩa tương ứng:

☀ Các đối tượng :

Animal, Bird, Breath, Skin, Move, Fish, Fly,
Wings, Feathers, Ostrich, Tall, Yellow,
Canary, Sing

☀ Các quan hệ: Can, Cannot, Has, Is, Is-a

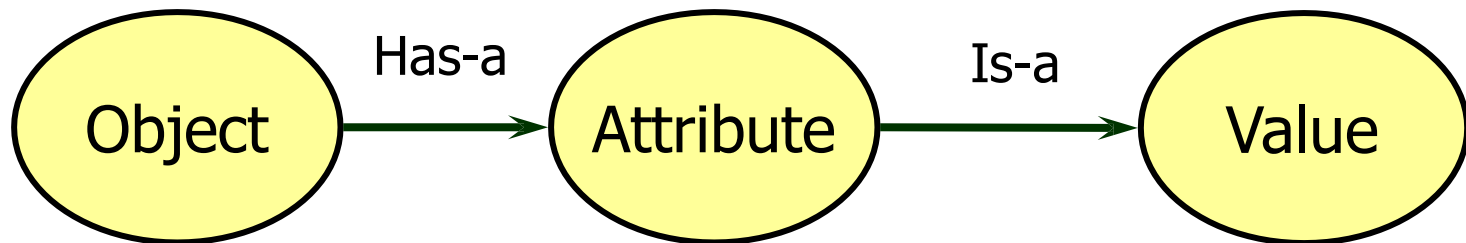
Semantic Network Representation



Biểu diễn tri thức nhờ bộ ba O-A-V

Bộ ba O-A-V (Object-Attribute-Value) :
đối tượng, thuộc tính và giá trị
cách mô tả các mạng ngữ nghĩa

- ✦ Can be used to characterize the knowledge in a semantic net
- ✦ Quickly leads to huge tables



Ví dụ : OAV Table

| Object | Attribute | Value |
|--------------|------------|------------|
| Beluga Whale | Dorsal Fin | No |
| Beluga Whale | Tail Fin | No |
| Blue Whale | Tail Fin | Yes |
| Blue Whale | Dorsal Fin | Yes |
| Blue Whale | Size | Very Large |

| Object | Attribute | Value |
|-----------|------------|-------------|
| Astérix | profession | warrior |
| Obélix | size | extra large |
| Idéfix | size | petite |
| Panoramix | wisdom | infinite |

Biểu diễn tri thức nhờ **khung**

Khi phạm vi bài toán được mở rộng ra :

- ☀ Số nút và số cung trong mạng tăng lên
- ☀ Mạng ngữ nghĩa trở nên phức tạp hơn

Semantic nets → Frame :

- ☀ Trong trường hợp phức tạp như vậy, người ta sử dụng khung frame

Giới thiệu khung (frame) :

Nhóm các thuộc tính, giá trị của các đối tượng

Hệ thống khung :

- ☀ Nhóm các khung có quan hệ với nhau.
- ☀ Quan hệ có thể là giá trị của một thuộc tính trong frame này là giá trị của một frame khác

Người ta xây dựng các thủ tục suy diễn sử dụng khung

Biểu diễn tri thức nhờ Frames

Frames were the next development, allowing more convenient “packaging” of facts about an object

Frames look much like modern classes, without the methods:

```
mammal:  
  subclass: animal  
  
elephant:  
  subclass: mammal  
  size: large  
  haspart: trunk  
  
Nellie:  
  instance: elephant  
  likes: apples
```

We use the terms “slots” and “slot values” (filler)

Frames

- Frames often allowed you to say which things were just typical of a class, and which were definitional, so couldn't be overridden
- Using an asterix to denote typical values:

Elephant:
subclass: mammal
haspart: trunk
* colour: grey
* size: large

- Frames also allowed multiple inheritance (Nellie is an Elephant and is a circus animal)
- Introduces problems in inheritance

Simple Frame Example

| <i>Slot Name</i> | <i>Filler</i> |
|------------------|-----------------|
| name | Astérix |
| height | small |
| weight | low |
| profession | warrior |
| armor | helmet |
| intelligence | very high |
| marital status | presumed single |

Thuộc tính

Giá trị

Representing Knowledge in Frames

Frame Architecture

- ✱ A record-like data structure for representing stereotypical knowledge about some concept or object (or a **class** of objects)
- ✱ A frame name represents a stereotypical situation/object/process
- ✱ Attributes or properties of the object also called **slot**
- ✱ Values for attributes called **fillers**,
facets provide additional control over fillers

Frame Architecture

Frame Name:

Object 1

Class:

Object 2

Properties:

| | |
|------------|---------|
| Property 1 | Value 1 |
| Property 2 | Value 2 |
| ... | ... |
| ... | ... |

Types of Frames

Class Frame

- ✦ Represents general characteristics of common objects
- ✦ Define properties that are common to all objects within class
- ✦ Static & dynamic property

Static:

- ✦ Describes an object feature whose value does not change

Dynamic:

- ✦ Feature whose value is likely to change during operation
- ✦

Mô hình kiểu của khung

Frame Name:

Bird

Class:

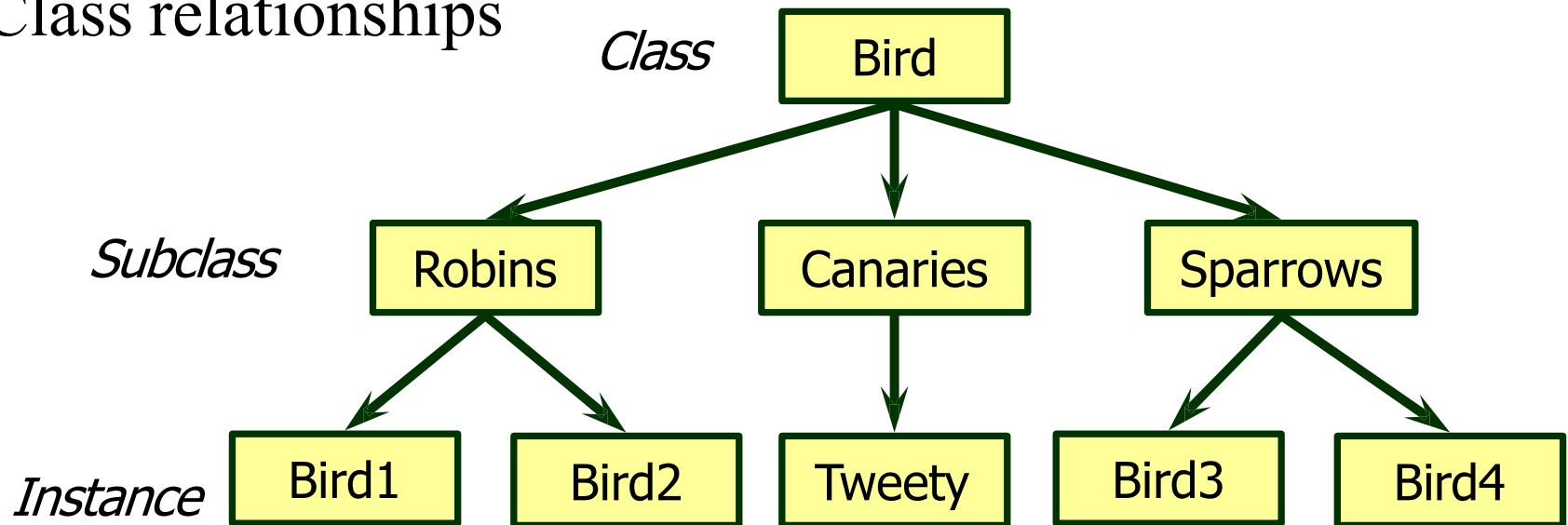
Animal

Properties:

| | |
|-----------|---------|
| Color | Unknown |
| Eats | Worms |
| No._Wings | 2 |
| Flies | True |
| Hungry | Unknown |
| Activity | Unknown |

Subclass Frame

- Represents subsets of higher level classes or categories
- Creates complex frame structures
- Class relationships



Instance Frame

Represents specific instance of a class frame

Inherits properties & values from the class

Able to change values of properties & add new properties

Frame Name:

Tweety

Class:

Bird

Properties:

| | |
|-----------|---------|
| Color | Yellow |
| Eats | Worms |
| No._Wings | 1 |
| Flies | False |
| Hungry | Unknown |
| Activity | Unknown |
| Lives | Cage |

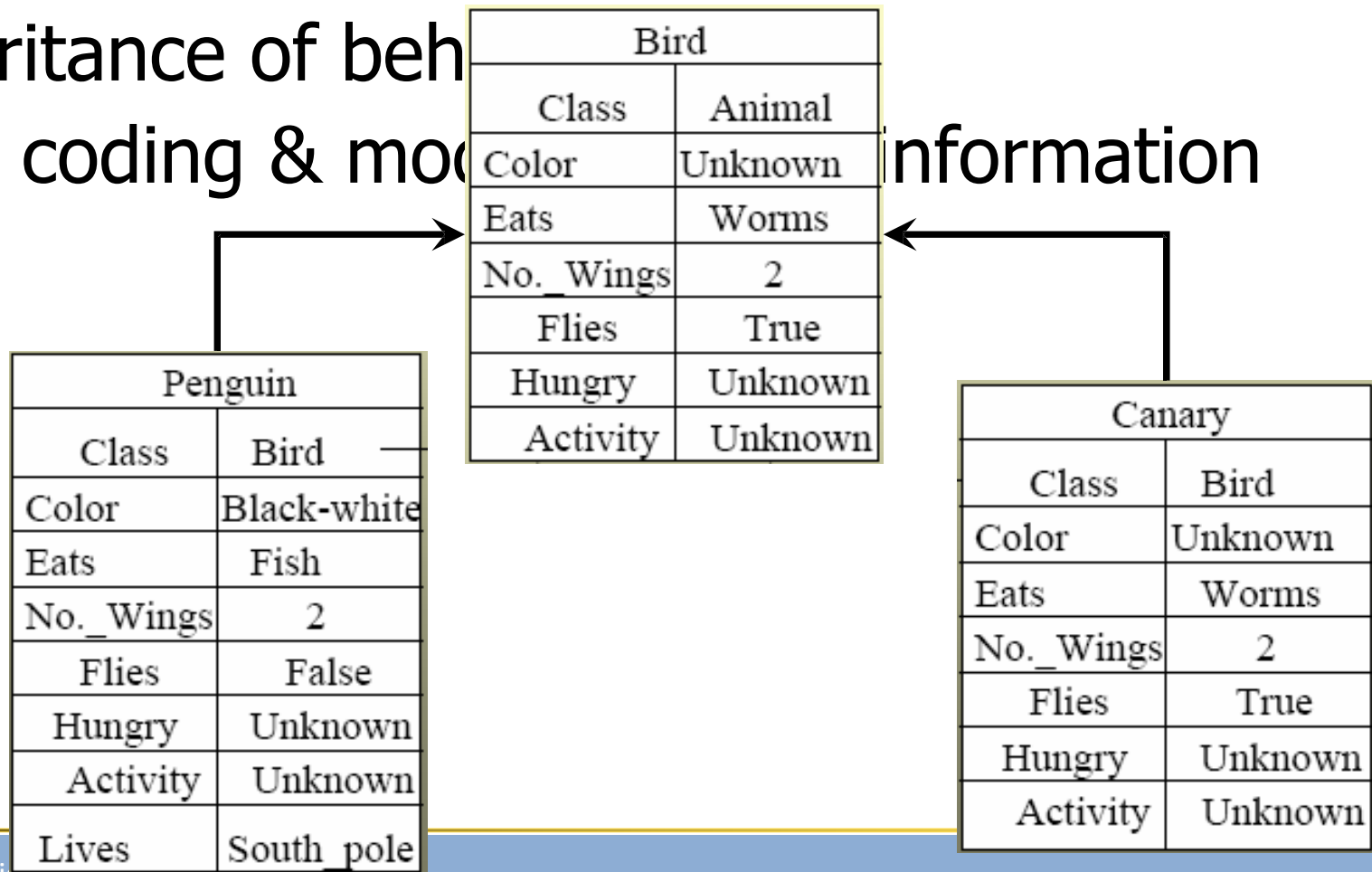
Frame Inheritance

Instance frame inherits information from its subclass frame and also its class

Inheritance of beh

Ease coding & mod

information



Frames and procedures

Frames often allowed slots to contain procedures

-So.. Size slot could contain code to run to calculate the size of an animal from other data

Sometimes divided into “if-needed” procedures, run when value needed, and “if-added” procedures, run when a value is added (to update rest of data, or inform user)

So.. Similar, but not quite like modern object-oriented languages

Overview of Frame Structure

Two basic elements: *slots* and *facets* (fillers, values, etc.);
Typically have parent and offspring slots

- ✱ Used to establish a property inheritance hierarchy (e.G., Specialization-of)

Descriptive slots

- ✱ Contain declarative information or data (static knowledge)

Procedural attachments

- ✱ Contain functions which can direct the reasoning process (dynamic knowledge)
(e.G., "Activate a certain rule if a value exceeds a given level")

Data-driven, event-driven (bottom-up reasoning)

Expectation-drive or top-down reasoning

Pointers to related frames/scripts - can be used to transfer control to a more appropriate frame

Usage of Frames

Filling slots in frames

- ✱ Can inherit the value directly
- ✱ Can get a default value
- ✱ These two are relatively inexpensive
- ✱ Can derive information through the attached procedures (or methods) that also take advantage of current context (slot- specific heuristics)
- ✱ Filling in slots also confirms that frame or script is appropriate for this particular situation

Problems with Frames

Negation cannot be represented

- ☀ Jim does not have pneumonia

Disjunction cannot be represented naturally

- ☀ Jim has Mumps or Rubella

Qualification is not a part of the language

- ☀ All of Jim's diseases are infectious

=> Thus, procedural attachments are often added

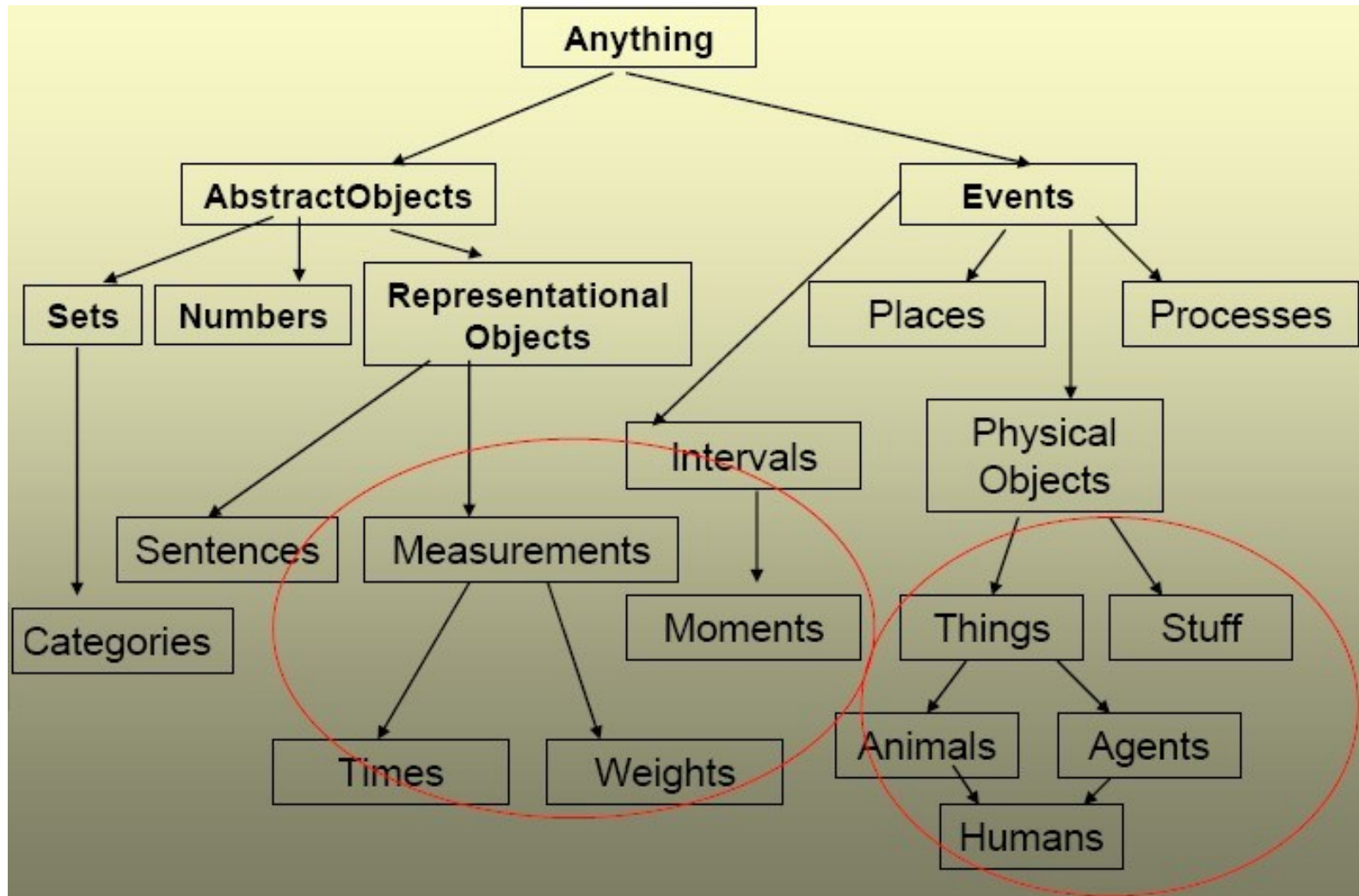
Biểu diễn tri thức ngôn ngữ nhân tạo

Thực tế, ngôn ngữ tự nhiên :

- ☀ Là phương cách thuận tiện nhất để giao tiếp với một XPS
- ☀ Không những đối với người quản trị hệ thống (tư cách chuyên gia)
- ☀ Mà còn đối với NSD cuối

Hiện nay đã có những XPS có khả năng đối thoại trên ngôn ngữ tự nhiên (thông thường là tiếng Anh) nhưng chỉ hạn chế trong một số lĩnh vực chuyên môn

A Quick Ontological View



Q&A
