Chapter 2

Knowledge Representation

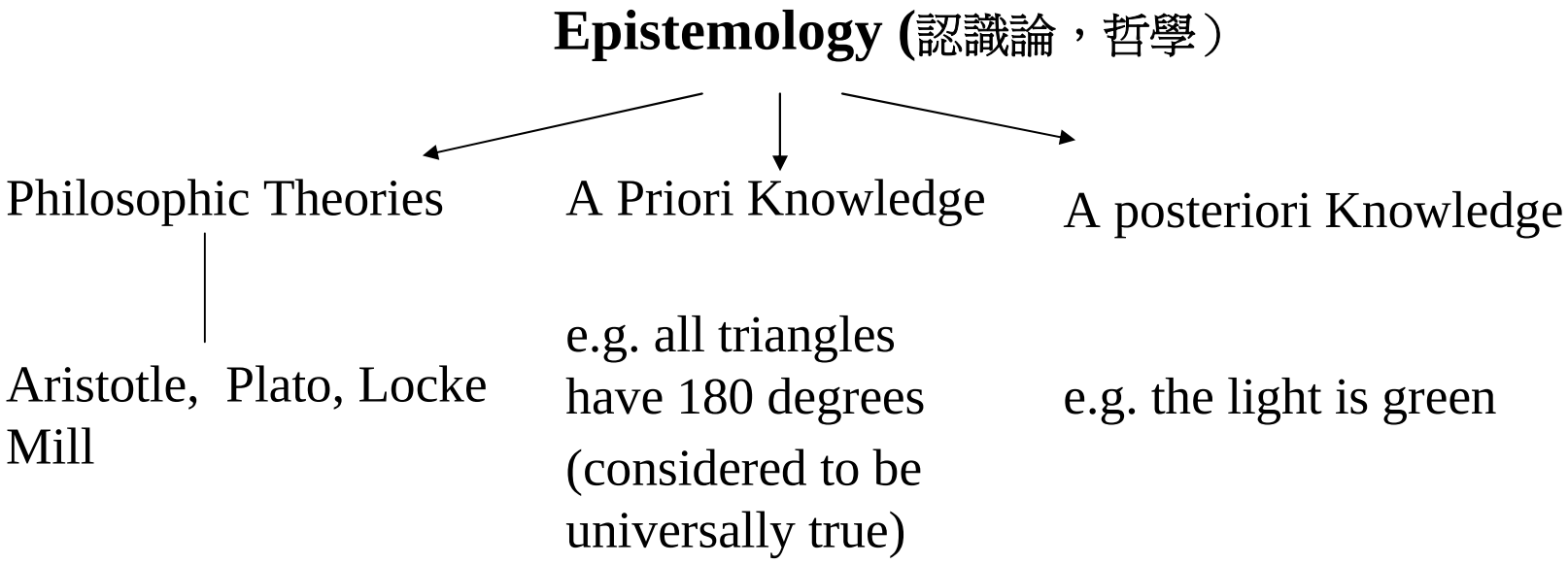
知識表示法

2.1 知識的定義

- Knowledge（知識） ＋ Inference （推論） ＝ Expert Systems（專家系統）
- “Knowledge representation” affects the development, efficiency, speed, and maintenance of expert systems

Epistemology

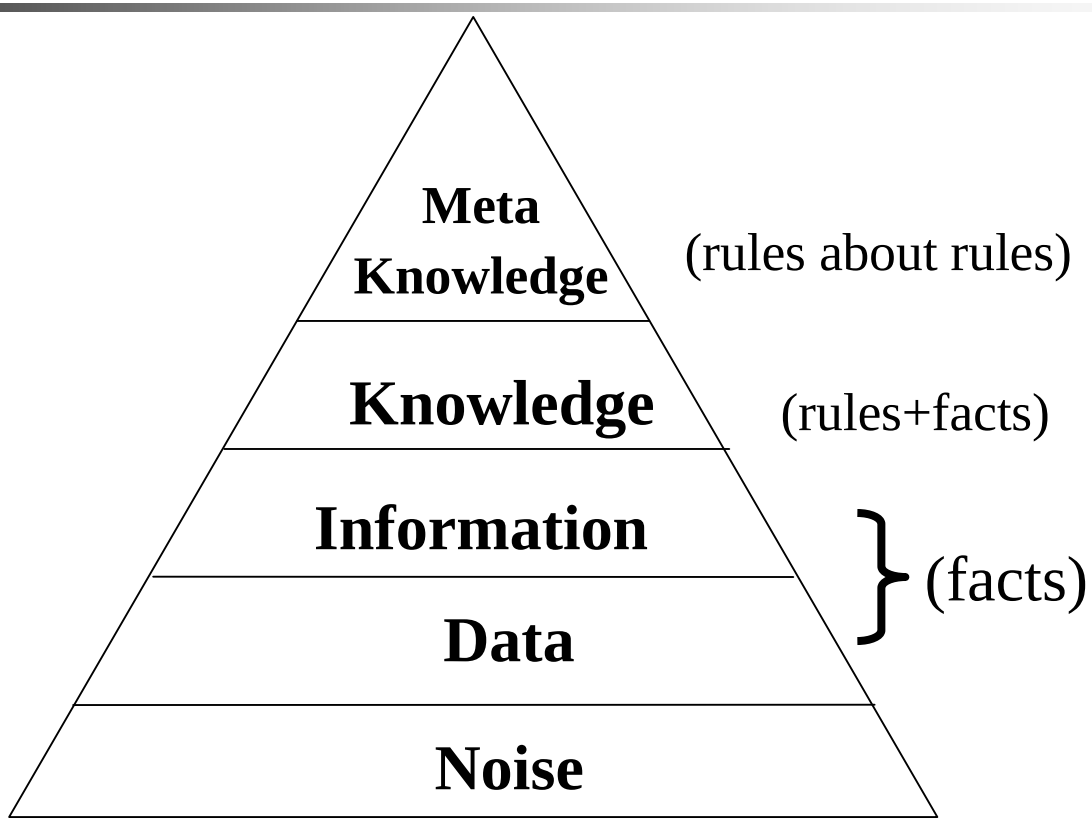
Epistemology: concerned with the nature, structure, and origin of knowledge



知識的種類 (Classifications of knowledge)

- **Procedural Knowledge（程序性知識）**
  How to do something
- **Declarative Knowledge（陳述性知識）**
  The truth of something
  “Don’t put your fingers in a pot of boiling water”
- **Tacit Knowledge（隱含知識）**
  (Unconscious Knowledge)
  Cannot be expressed explicitly
  —An example is how to move your hand
  —Walking or riding a bicycle
  —ANS is related to tacit knowledge

知識階層(Knowledge Levels)



The sequence of 12 numbers：137178007124
Without knowledge. This entire sequence may appear to be noise.

後設知識(Meta Knowledge)

Meta Rule 1：

If Goal is “blood pressure”

THEH Try Rule Class 1

Meta Rule 2：

If Goal is “blood type”

THEH Try Rule Class 2

Meta knowledge is knowledge about knowledge and expertise.

－would specify which knowledge base was applicable or useful to solve the current problem.

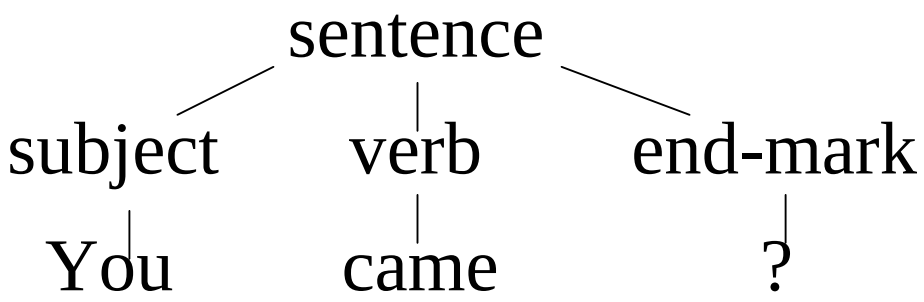
知識表示法 (Knowledge Representation)

- **Backus - Naur form**
- **Ontology（本體論）**
- **Semantic Network（語意網路）**
- **Frames-based Knowledge（框架式知識）**
- **Case-based Knowledge（案例式知識）**
- **Rule-based Knowledge（規則式知識）**
- **Knowledge Object（知識物件）**
- **Logic（邏輯）**

Backus - Naur form (BNF)

- **A meta language for defining the syntax of a language**
- **<sentence>::= <subject><verb><end-mark>**
  <subject>::= I | You | We
  <verb> ::= left | came
  <end-mark> ::= . | ? | !

- Parse Tree (derivation tree)



Ontology（本體論）

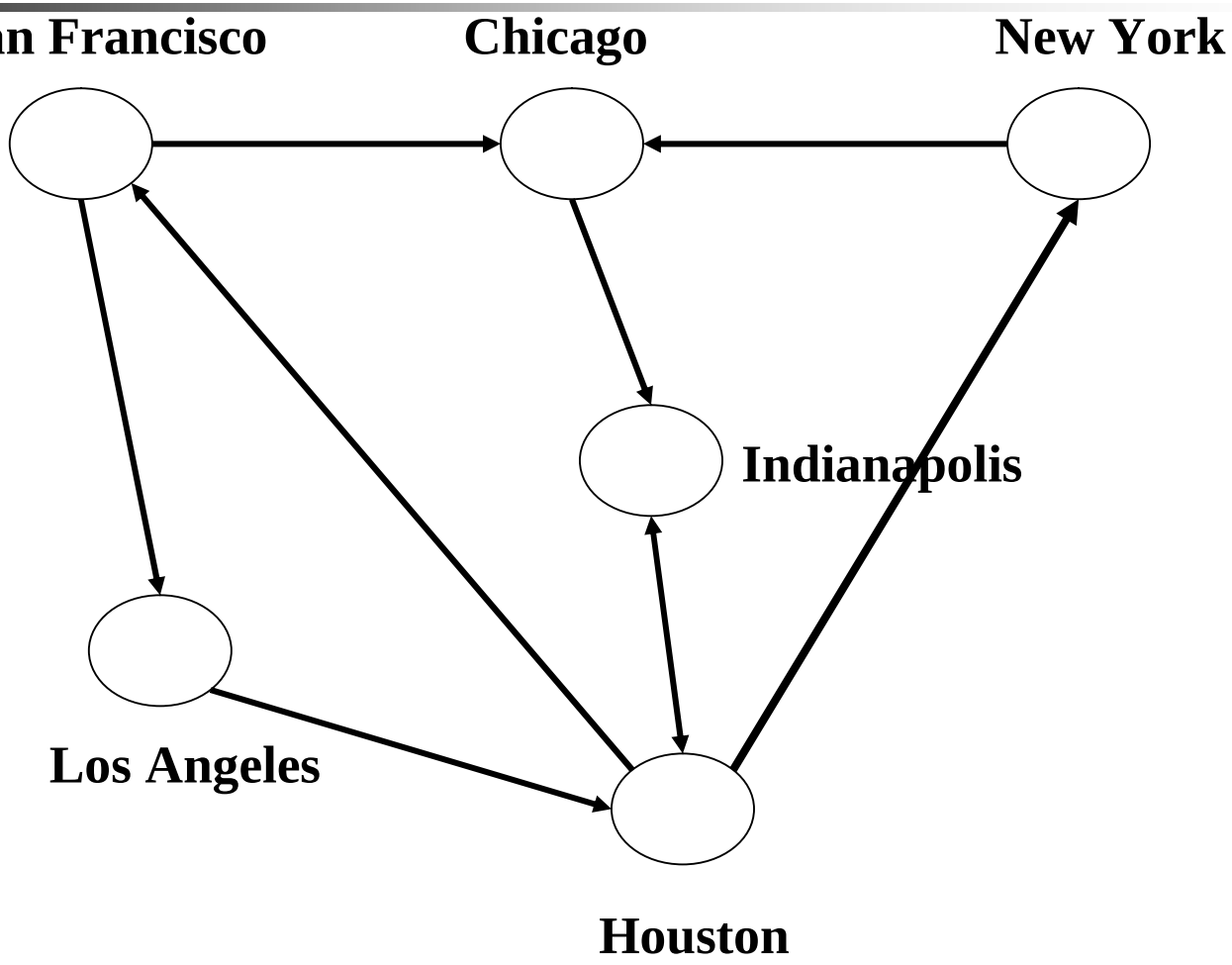
- **Ontology**一詞在**90**年代開始用在人工智慧領域，描述知識的知識構成要素之間的關係。
- **Ontology**的研究大分爲兩個方向：
  - 針對特定的問題領域建立大量的**Ontology**
    - 例如：建立某些領域詞彙的**Ontology**
  - 研究**Ontology**的建構方法與表示方法
    - 例如：利用**XML**（可延伸標記語言）或是**RDF**（資源描述格式）

- **Ontology**的發展主要是用來使知識分享和再用更爲容易。
- 不同的研究對於**Ontology**的表示與描述有不同的方法，目前還未看到較一般化、通用的表示法 。

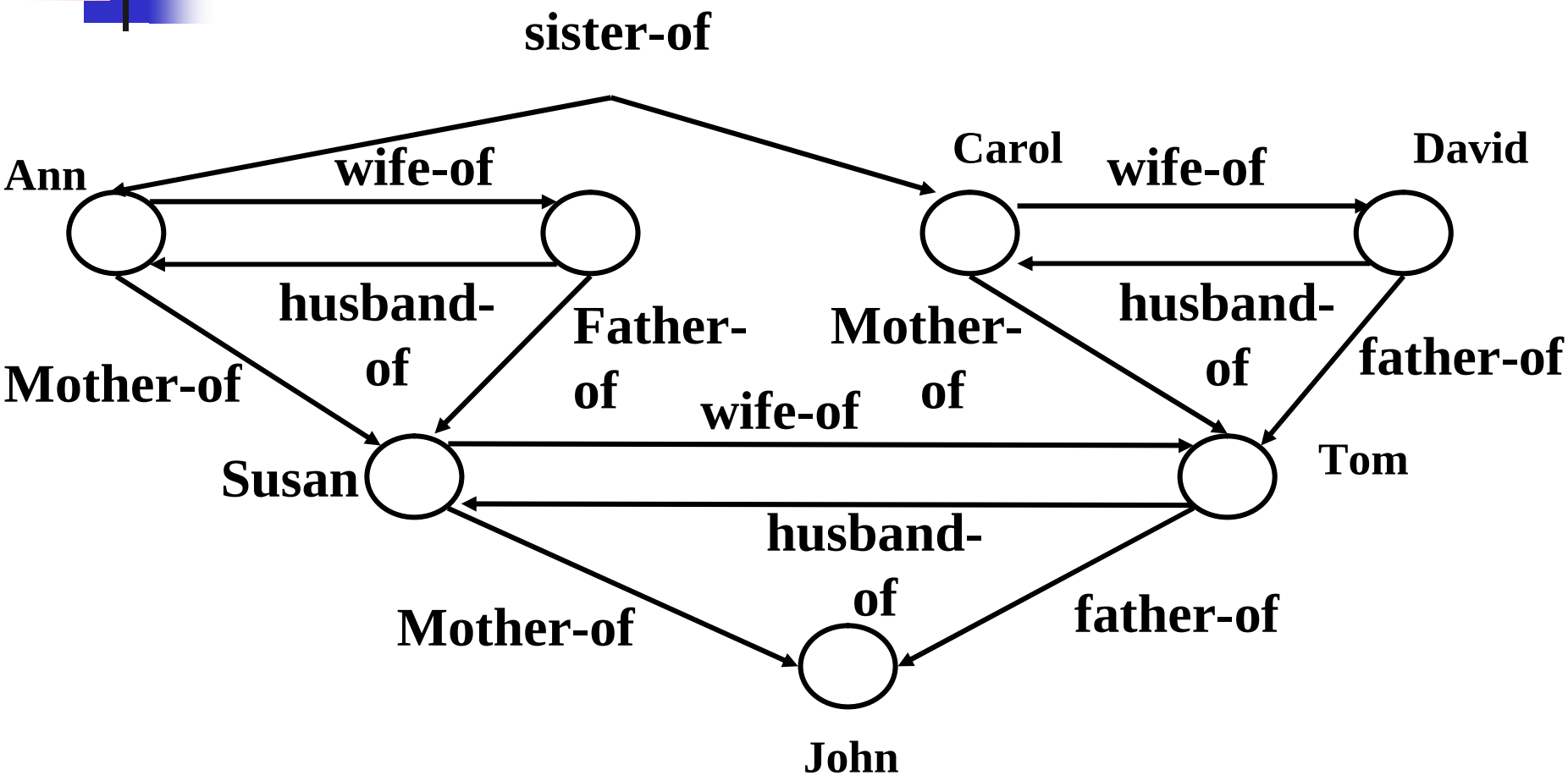
Semantic Network（語意網路）(Quillian 67 & 68)

- **A classic AI representation technique used for propositional information (sometimes called Propositional Net)**
- **A proposition is a statement that is either true or false**
- **A directed graph（有向圖形）**
  - **Node（點）: 知識的組成元素或是種類**
  - **Arc（有向線段）: 知識組成元素間的關**

General Net


San Francisco
Chicago
New York
Indianapolis
Los Angeles
Houston

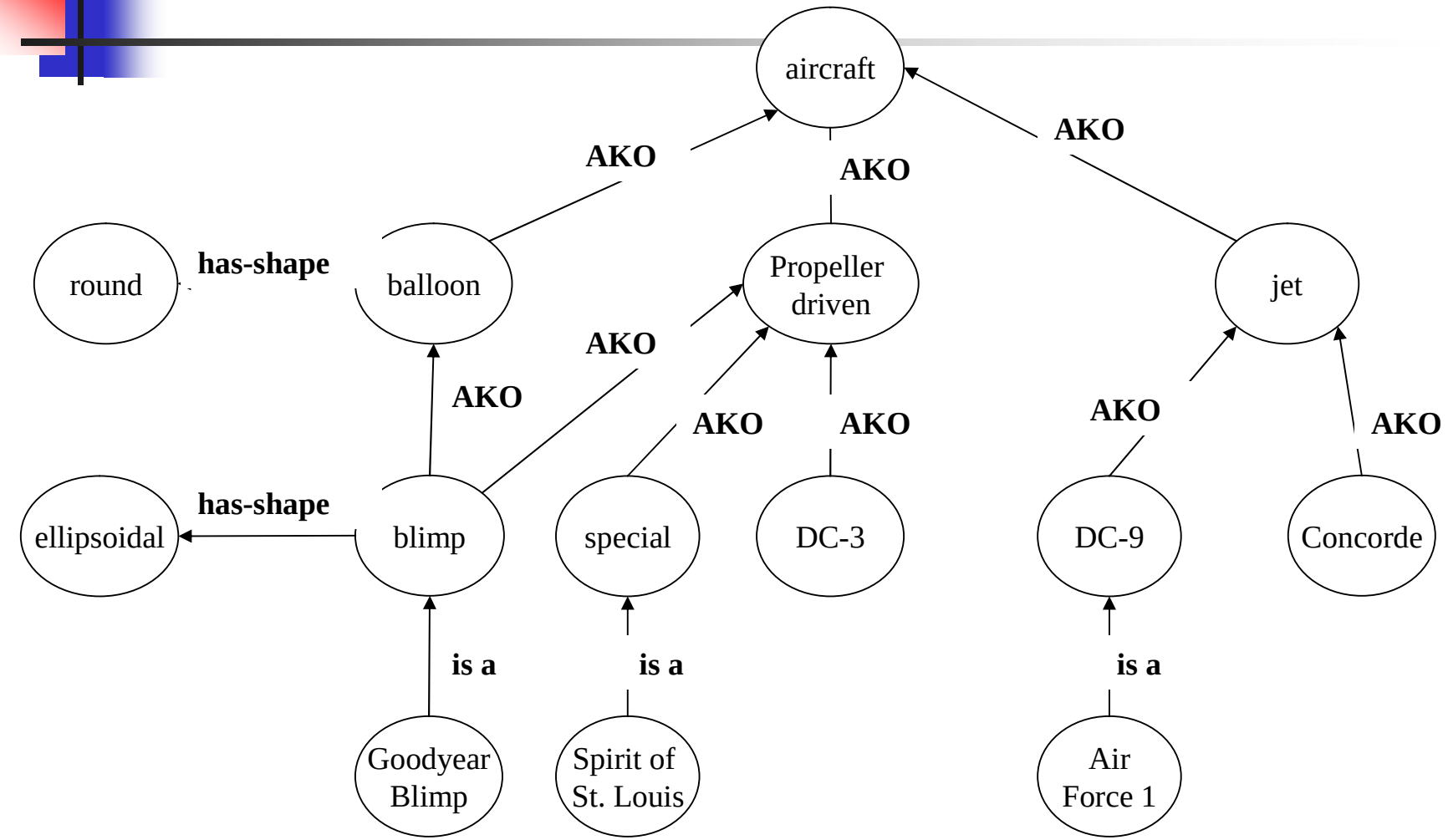
Semantic Net（語意網路）



其他關於Semantic Net 的特性

- 「**Is a**」 **:**表示的知識物件屬於知識類別中的一個例子。
- 「**Is a kind of**」 **:**表示的知識爲子類別。
  - **Superclass**（父類別）
  - **Subclass**（子類別）
- **Attribute, Value, Property(OAV)**
- **Inheritance**（繼承）

A Semantic Network with 「is a」 and 「a kind of」(AKO) Links



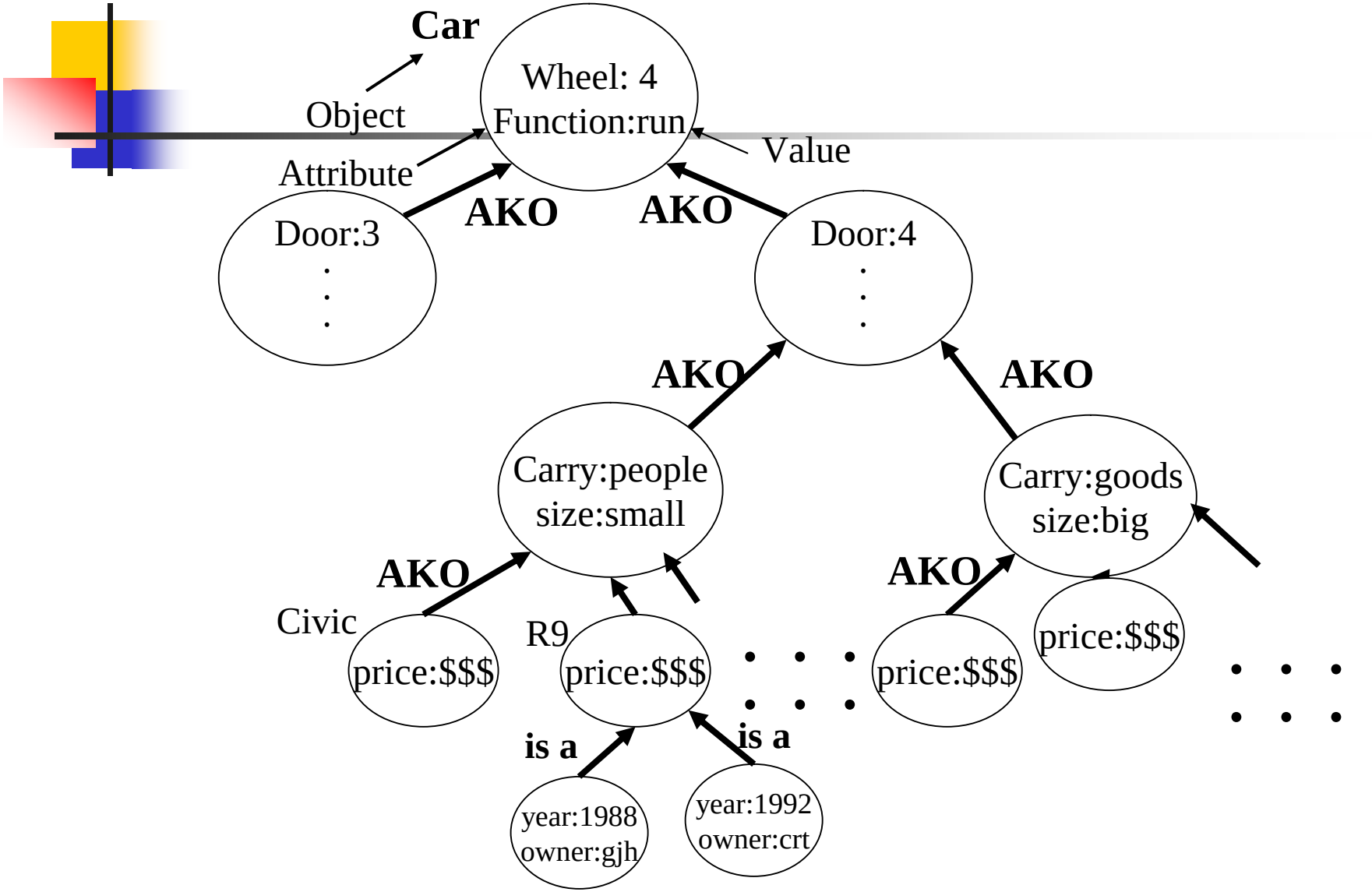
Object-Attribute-Value Triples (OAV)（物件-屬性-屬性值法）

- **OAV can be used to characterizes all the knowledge（知識） in a semantic net（語意網路） and was used in MYCIN for diagnosing infections diseases**

Object	Attribute	Value
apple	color	red
apple	type	mcintosh
apple	quantity	100
grapes	color	red
grapes	type	seedless
grapes	quantity	500

- **Especially useful for representing facts（事實）**

Object-Attribute-Value Triples
物件-屬性-屬性值法）



PROLOG and Semantic Nets

- **Essentials of PROLOG**

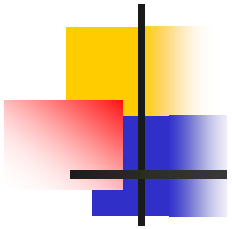
  Each of the statements above is a PROLOG predicate expression, or simply a predicate.

  Color(red). ; red is a color

  father_of(Tom,John). ; Tom is the father of John

  mother_of(Susan,John). ; Susan is the mother of John

  parents(Tom,Susan,John). ; Tom and Susan are parents of John



Predicates can also be expressed with relations such as the IS-A and HAS-A.

is_a (red,color).

has_a (John,father).

has_a (John,mother).

has_a (John,parents).

Some additional predicates

is_a (Tom,father).

is_a (Susan,mother).

is_a (Tom,parent).

is_a (Susan,parent).

Programs in PROLOG consist of facts and rules in the general form of goals.

$$p :- p_1, p_2 \cdots p_n.$$

In which p is the rule's head and the pi are the subgoals.

The symbol,:-, is interpreted as an IF.

parent (x,y) : - father (x,y).

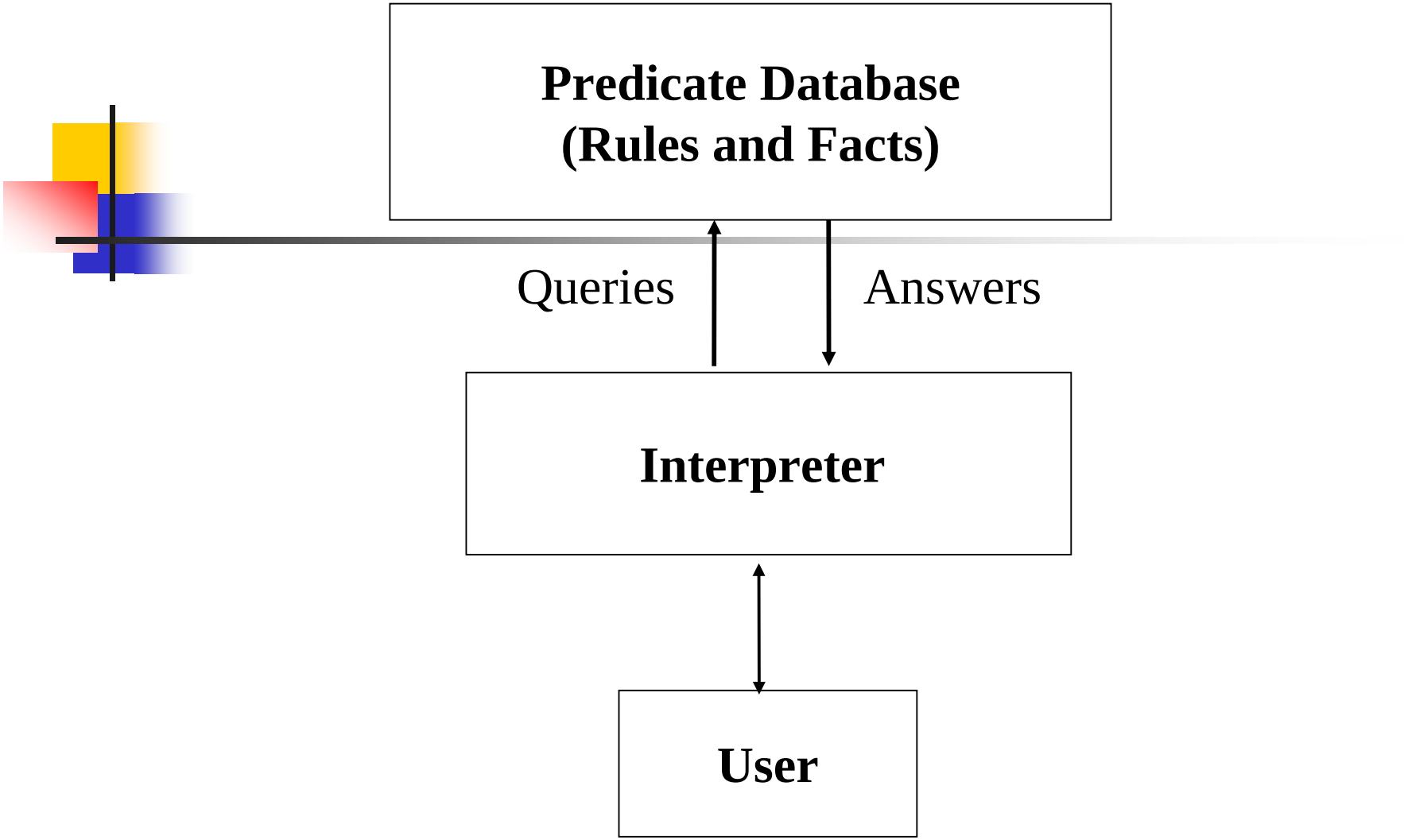
parent (x,y) : - mother(x,y).

grandparent(x,y) :- parent(x,z), parent(z,y).

and an ancestor can be defined as:

(1) ancestor(x,y) :- parent(x,y).

(2) ancestor(x,y) :- ancestor(x,z),ancestor(z,y).



General Organization of a PROLOG System

(3) parent (Ann,Mary).

(4) parent (Ann,Susan).

(5) parent (Mary,Bob).

(6) parent (Susan,John).

As another example, suppose the query is

:-ancestor(Ann,John).

The first ancestor rule(1) matches and X is set to Ann and Y is set to John. PROLOG now tries to match the body of (1), parent (Ann,John) with every parent statement.

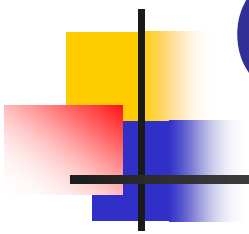
Rule (1) is not true since its IF part cannot be true.

Because Rule (1) cannot be true, PROLOG then tries the second ancestor statement.

For Rule (2), X is set to Ann and Y is set to John.

Control structure of PROLOG is of the Markov algorithm type, in which searching for pattern matching is normally determined by the order in which the Horn clauses are entered.

Schema
(plural schemas or schematas)



- **A semantic net（語意網路） is an example of a shallow knowledge（淺層知識） structure.**
- **A general term to describe a complex knowledge structure**
- **Focus on only relevant knowledge**
- **For examples: FRAME, SCRIPT**

Frames-based Knowledge
（框架式知識）

- **Suitable for related knowledge about a narrow subject with much default knowledge**
- **script－a time-ordered sequence of frames**

Slots	Slots value
manufacturer	General Motors
model	Chevrolet Cqprice
year	1979
transmission	automatic
engine	gasoline
tires	4
color	blue

Slot value

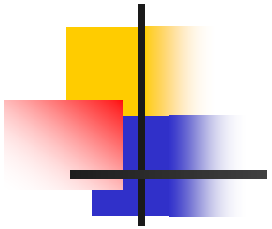
- Some frame-based tools (e.g., KEE) allow a wide range of items to be stored in slots
  - an assigned value
  - a default value
  - rules
  - graphics
  - comments
  - debugging information
  - questions for users
  - function
  - procedural attachment
  - link to other frame

Procedural Attachment

- If – needed, if-added, if-removed
- Examples：Human Property

Slots	Slots 值
name	property
specialization_of	a_kind_of object
types	(car, boat, house) if-added：Procedure ADD_PROPERTY
owner	default：government if-needed：Procedure FIND_OWNER
location	(home, work, mobile)
status	(missing, poor, good)
under_warranty	(yes, no)

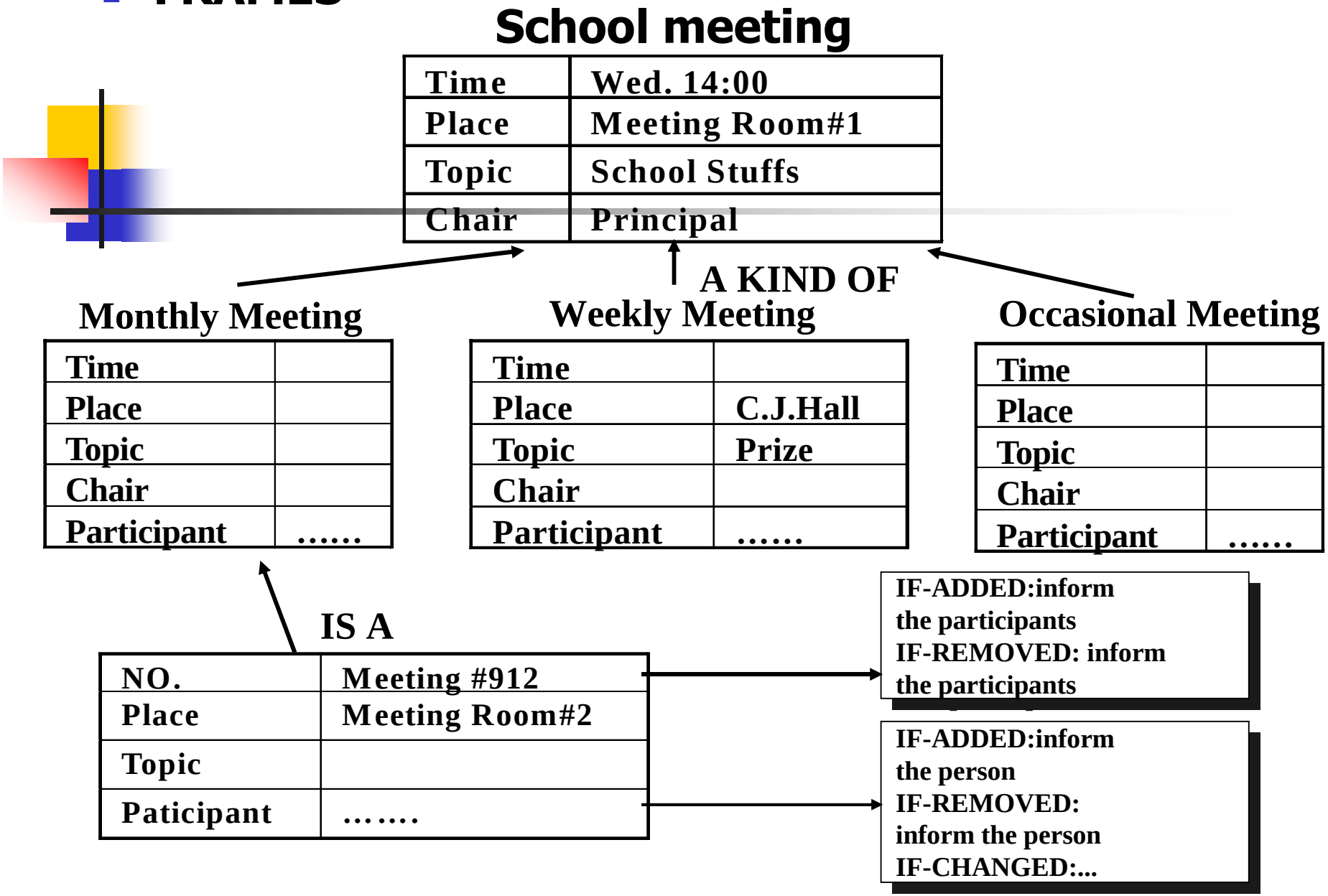
Frame Hierarchy



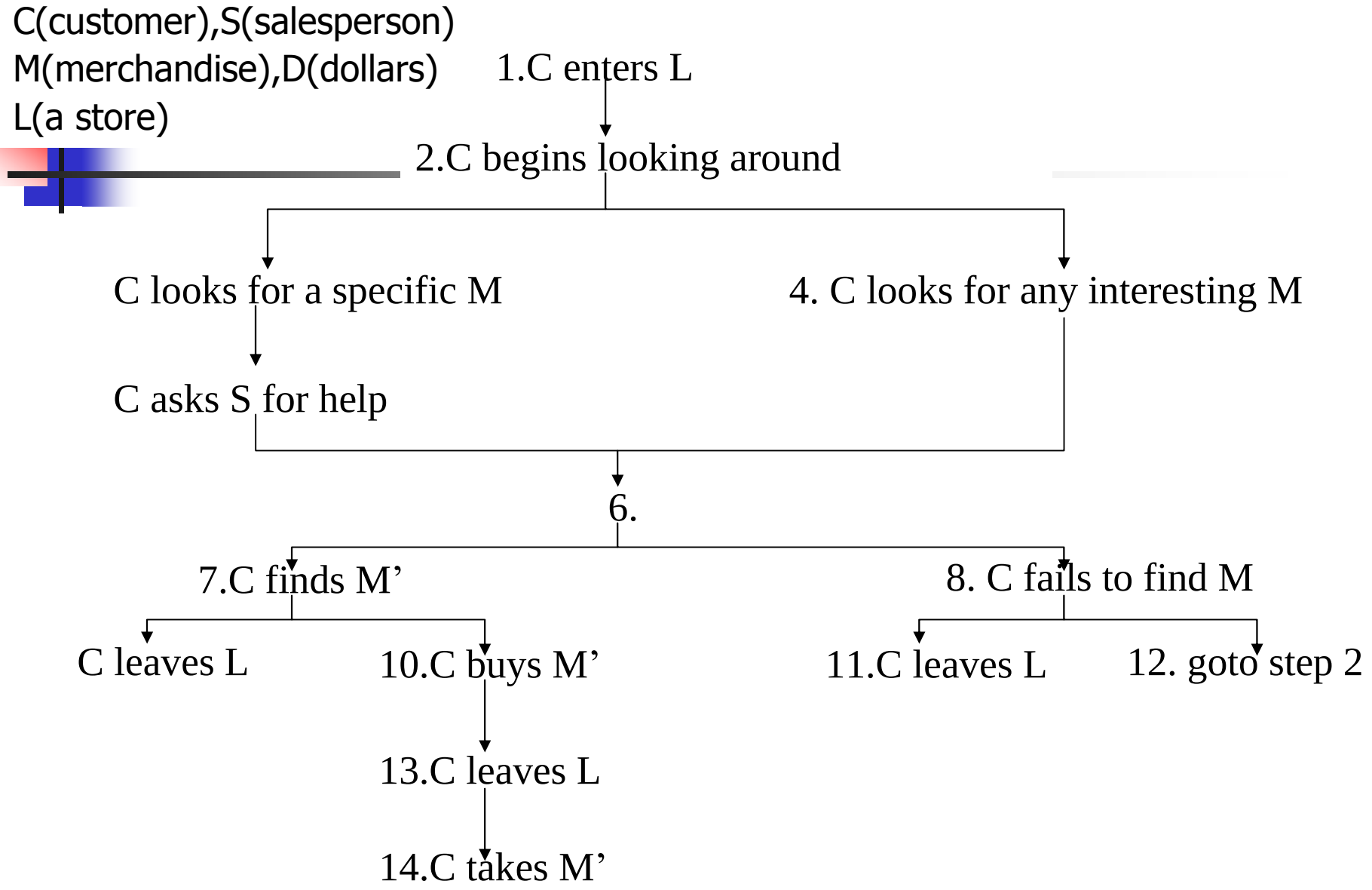
Slots	Slots 値
name	car
specialization_of	a-kind-of property
types	(sedan,sport,convertible)
manufacturer	(GM,Ford,Chrysler)
location	mobile
wheels	4
transmission	(manual, automatic)
engine	(gasoline, diesel)

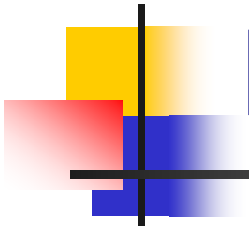
Slots	Slots 値
name	John's car
specialization_of	is_a car
manufacturer	GM
owner	John Doe
transmission	automatic
engine	gasoline
status	good
under_warranty	yes

- **FRAMES**


School meeting
Time
Wed. 14:00
Place
Meeting Room#1
Topic
School Stuffs
Chair
Principal
A KIND OF
Monthly Meeting
Time
Place
Topic
Chair
Participant
……
Weekly Meeting
Time
Place
C.J.Hall
Topic
Prize
Chair
Participant
……
Occasional Meeting
Time
Place
Topic
Chair
Participant
……
IS A
NO.
Meeting #912
Place
Meeting Room#2
Topic
Paticipant
…….
IF-ADDED:inform
the participants
IF-REMOVED: inform
the participants
IF-ADDED:inform
the person
IF-REMOVED:
inform the person
IF-CHANGED:...

Script


C(customer),S(salesperson)
M(merchandise),D(dollars)
L(a store)
1.C enters L
2.C begins looking around
C looks for a specific M
4. C looks for any interesting M
C asks S for help
6.
7.C finds M'
8. C fails to find M
C leaves L
10.C buys M'
11.C leaves L
12. goto step 2
13.C leaves L
14.C takes M'



Did Mary buy anything?

Mary went shopping for a new coat. She found a new one. She really liked When she got it home, and discovered that it went perfectly with her favorite dress .

Question：Did Mary buy anything?

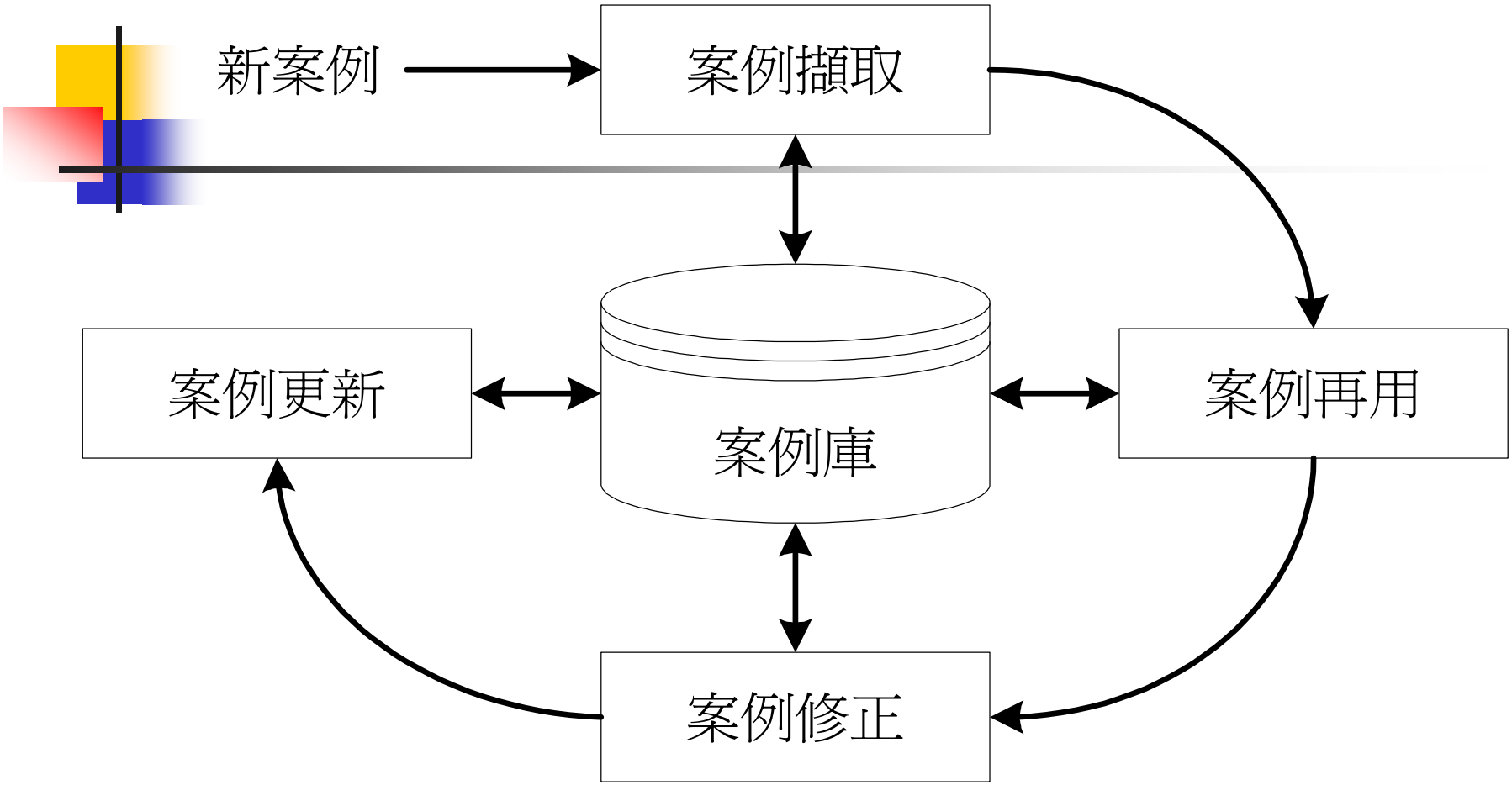
Case-based Knowledge
（案例式知識）

- 通常是用來描述屬於經驗的知識
- 從過去的經驗中，判定是何種相似的**case**（案例），並且依據過去解決此問題的方法，來解決此次問題
- **Case**（案例）：
  - 案例名稱
  - 屬性
  - 屬性值

案例名稱		
	屬性**1**	屬性值
	屬性**2**	屬性值
	……	……
	屬性**N**	屬性值

利用Case-based Knowledge（案例式知識）建構Expert System（專家系統）

- **Case Retrieve**（案例擷取）
- **Case Reuse**（案例再用）
- **Case Revise**（案例修正）
- **Case Retain**（案例更新）



案例推論循環

Rule-based Knowledge（規則式知識）

- 知識領域具備需要推論的特性
  - 例如：醫生依據其所學的醫學知識及病人所呈現的症狀去判別所罹患的疾病
- 最基本的**Rules**（規則）形式

  如果「狀態」則「結論」

  IF (condition) THEN (conclusion)
- **Inference Chaining**（推論鏈）
  - **Forward Inference**（前向推論）
  - **Backward Inference**（後向推論）