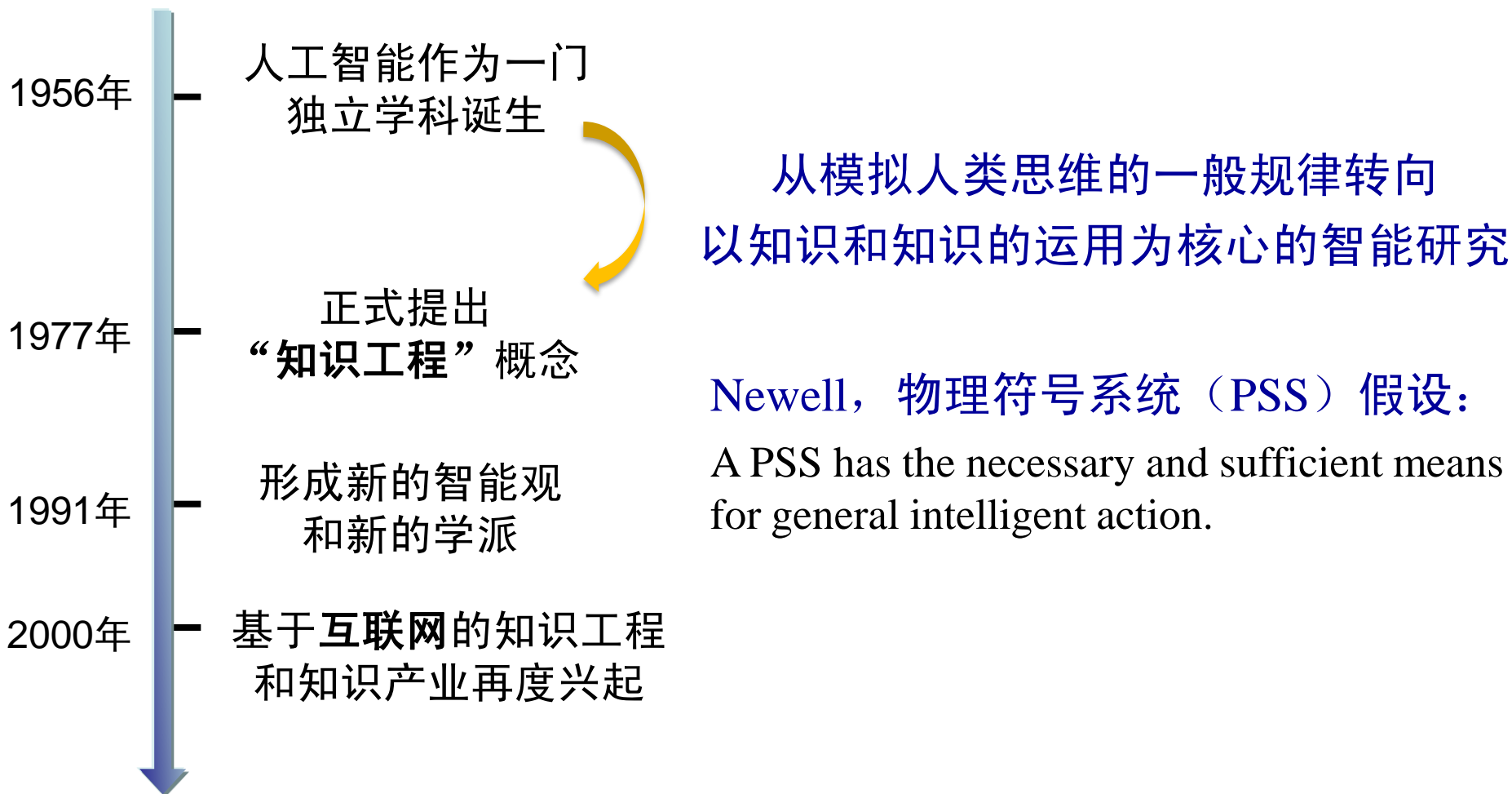


# AI 发展历程



# 知识表示、推理与专家系统

- 知识表示
- 基于知识的推理
- 专家系统及其应用



# 知识表示 (Knowledge Representation)

毛文吉

中国科学院自动化研究所

2016年11月

# 关于知识表示

- **什么是知识表示？**

知识表示是用计算机能够接受并进行处理的方式来表示人类的知识

- **知识表示的目的**

通过知识的有效表示，使人工智能程序能够利用这些知识求解问题



# 人工智能所涉及的知识

- **事实**

关于问题环境的一些客观事物的知识，通常形式：“…是…”

- **规则**

关于事物之间因果关系的知识，通常形式：“如果…那么…”

- **控制知识**

关于做一件事的技巧性、策略性的知识

- **元知识**

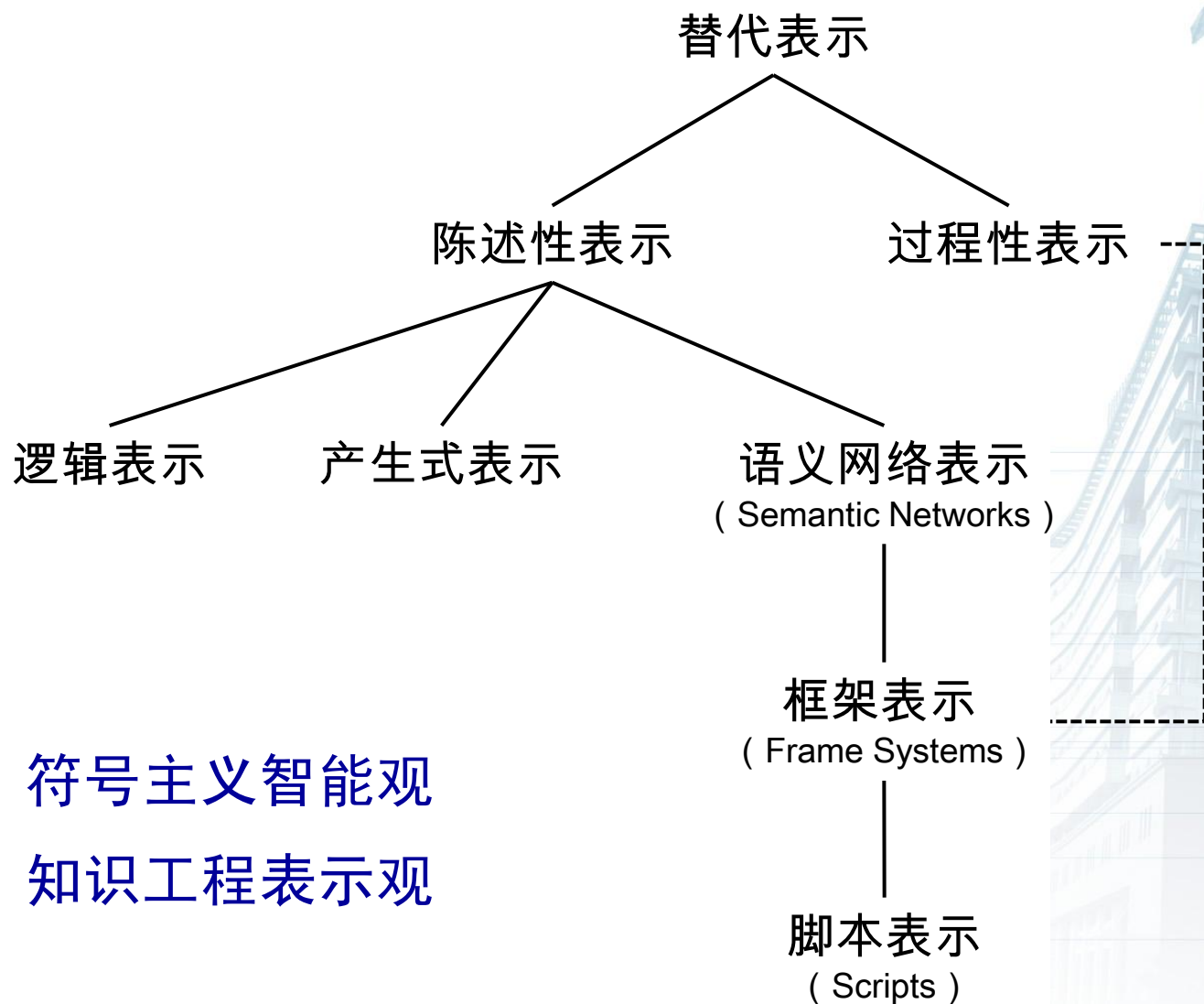
关于知识的知识，有两种：包括说明知识和运用知识的元知识

# 如何衡量知识表示方法

- 知识表示的范围和程度（广度、精度）
- 是否适于计算机处理
- 是否适合推理
- 是否易于用户使用

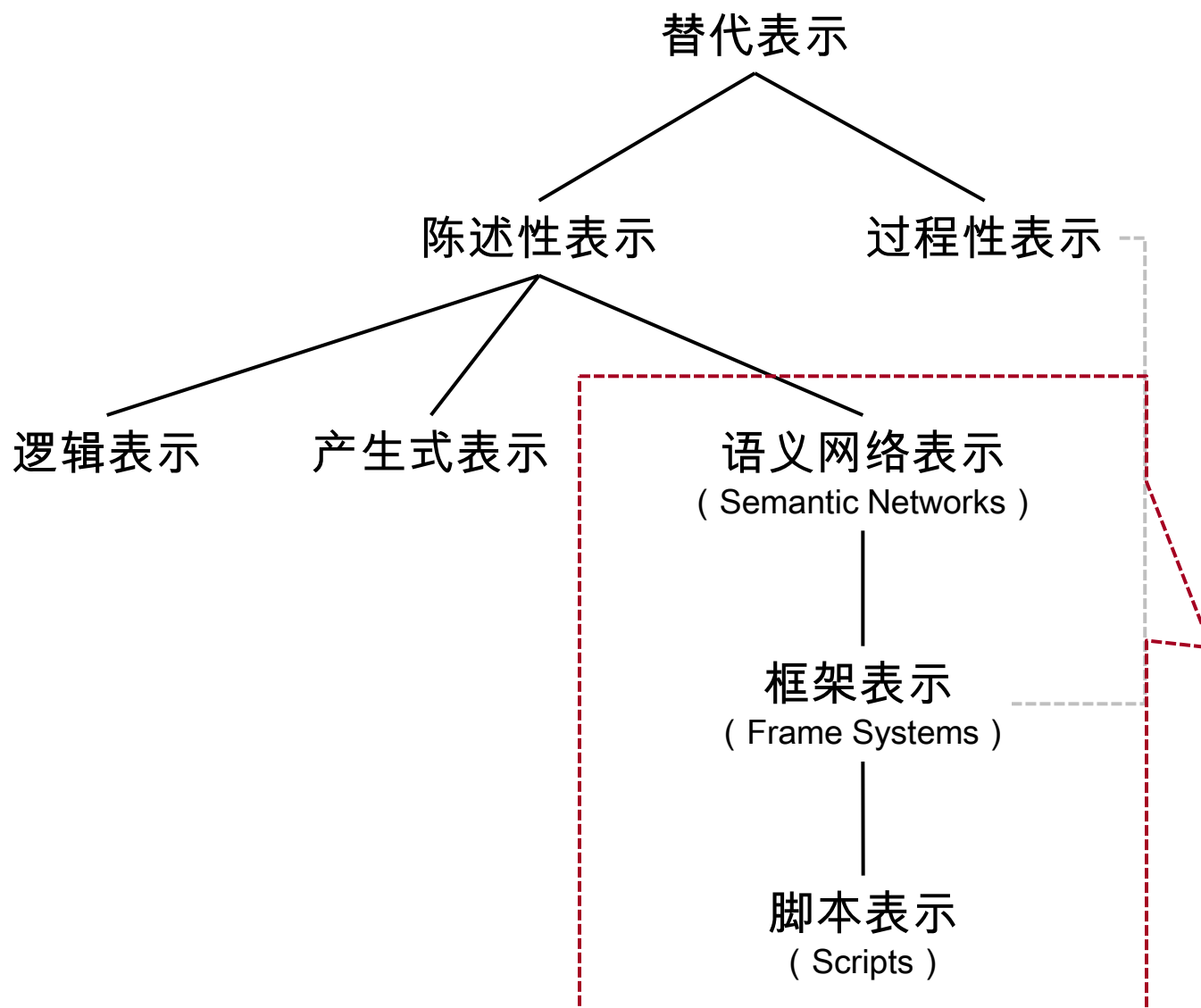
陈述性表示	过程性表示
知识的静态描述	知识的动态描述
表示与推理分开处理	表示与推理相结合
较严格，模块性较好	不够严格，知识间有交互
有时效率较低	求解效率高

# 知识表示方法体系



- 符号主义智能观
- 知识工程表示观

# 结构化知识表示方法





# 语义网络

- 结构

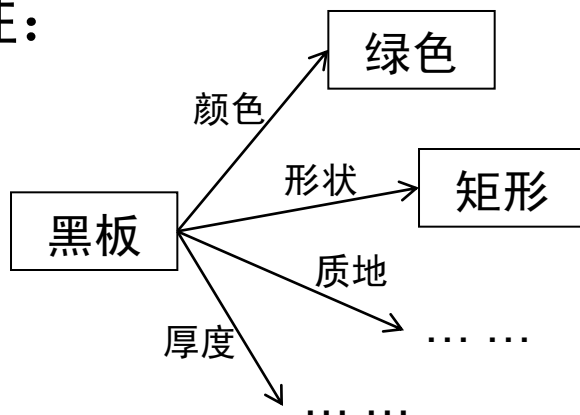
采用有向图表示知识：

结点：事物、概念、属性值、事件、情形等

弧：结点1的属性或结点之间的关系；方向表示主次

- 表示

(1) 结点的属性：

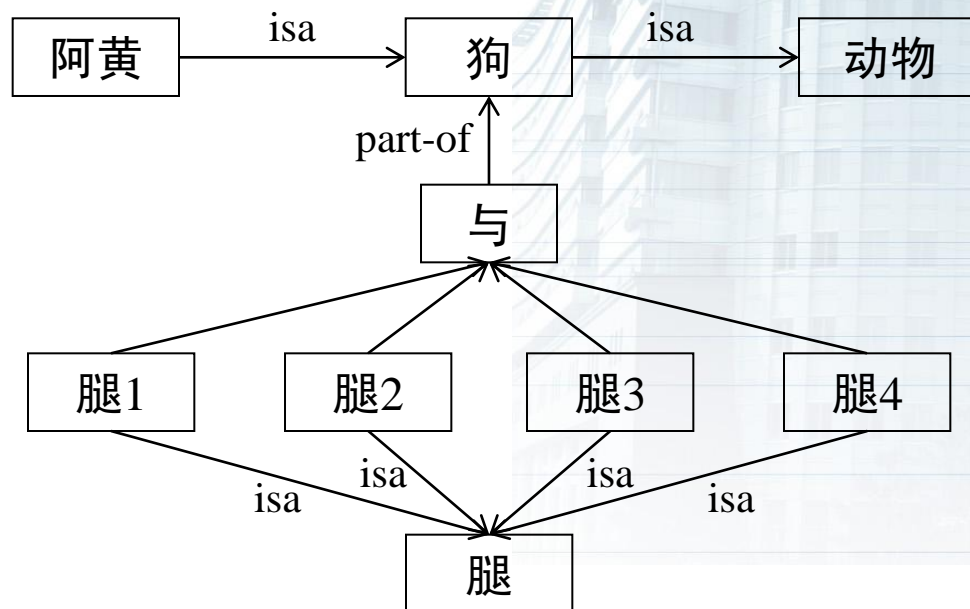
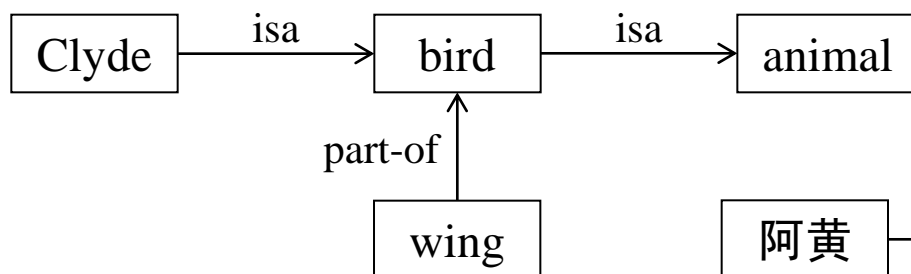


# 语义网络表示

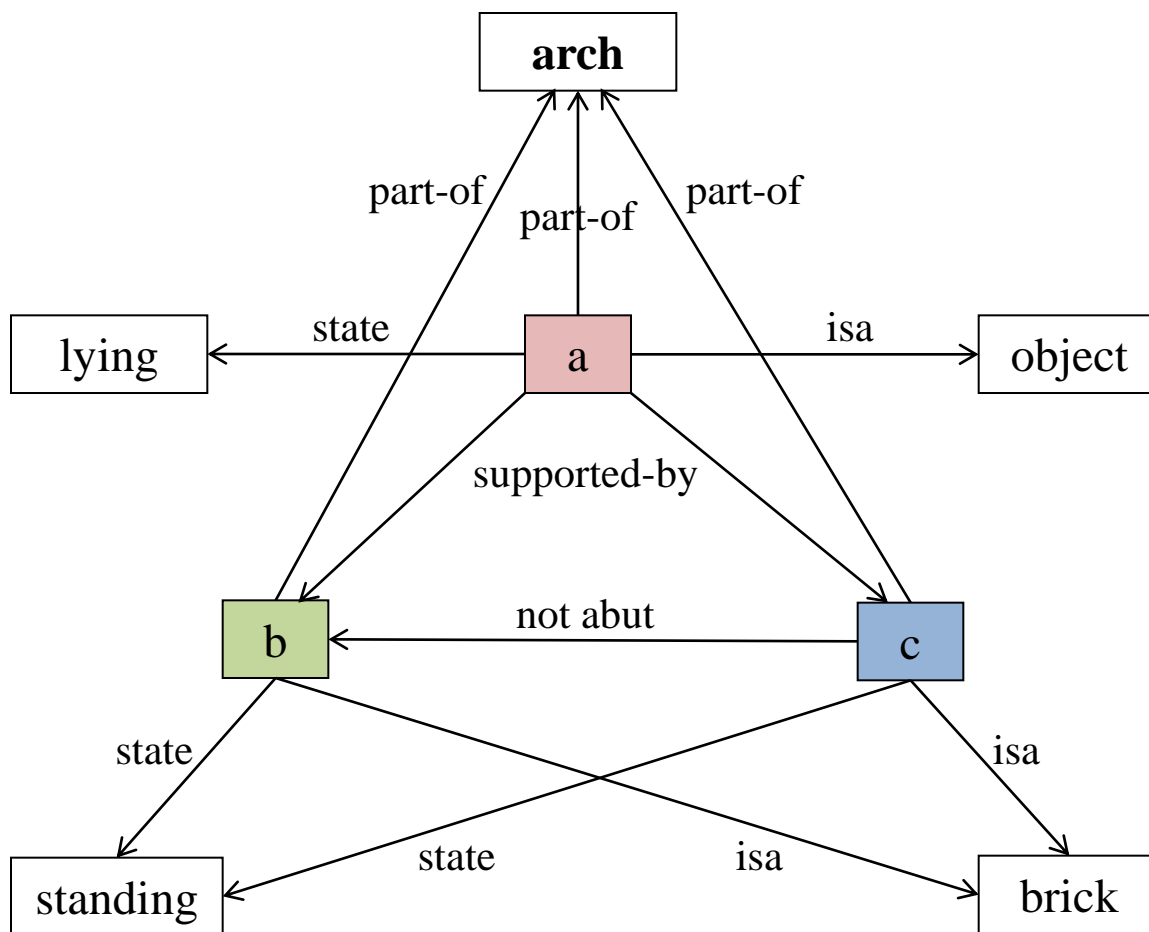
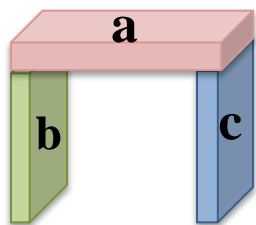
(2) 结点间典型关系:

**isa:** 个体对类、子类对类的隶属关系

**part-of:** 全体对部分的包含关系



Winston 学习系统中对“拱”的概念进行学习时的语义网络：



# 语义网络表示

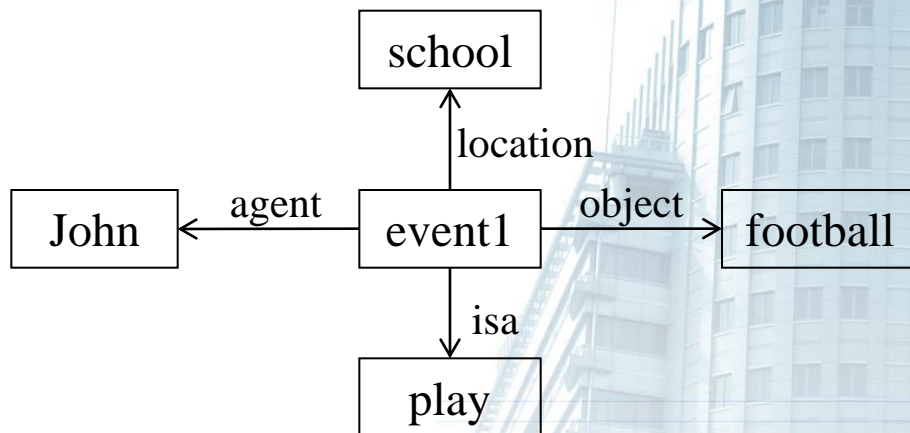
(3) event 结点：表示一个动作或事件

agent: 动作的施主

object: 动作的对象

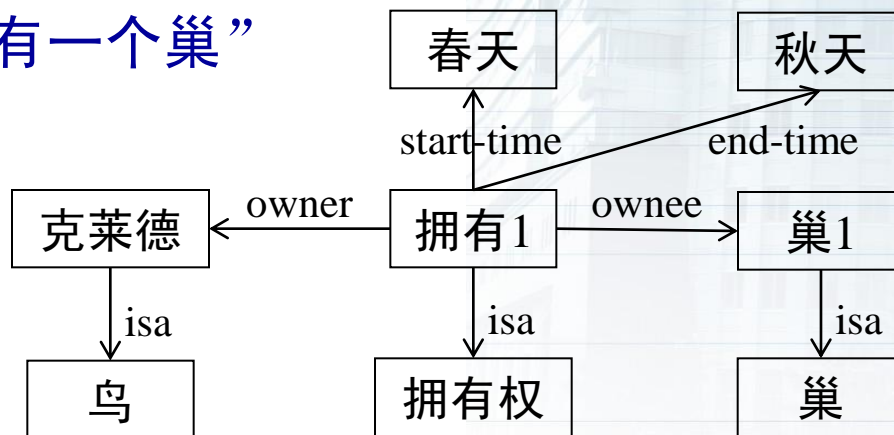
location: 动作的位置

time: 动作的时间等

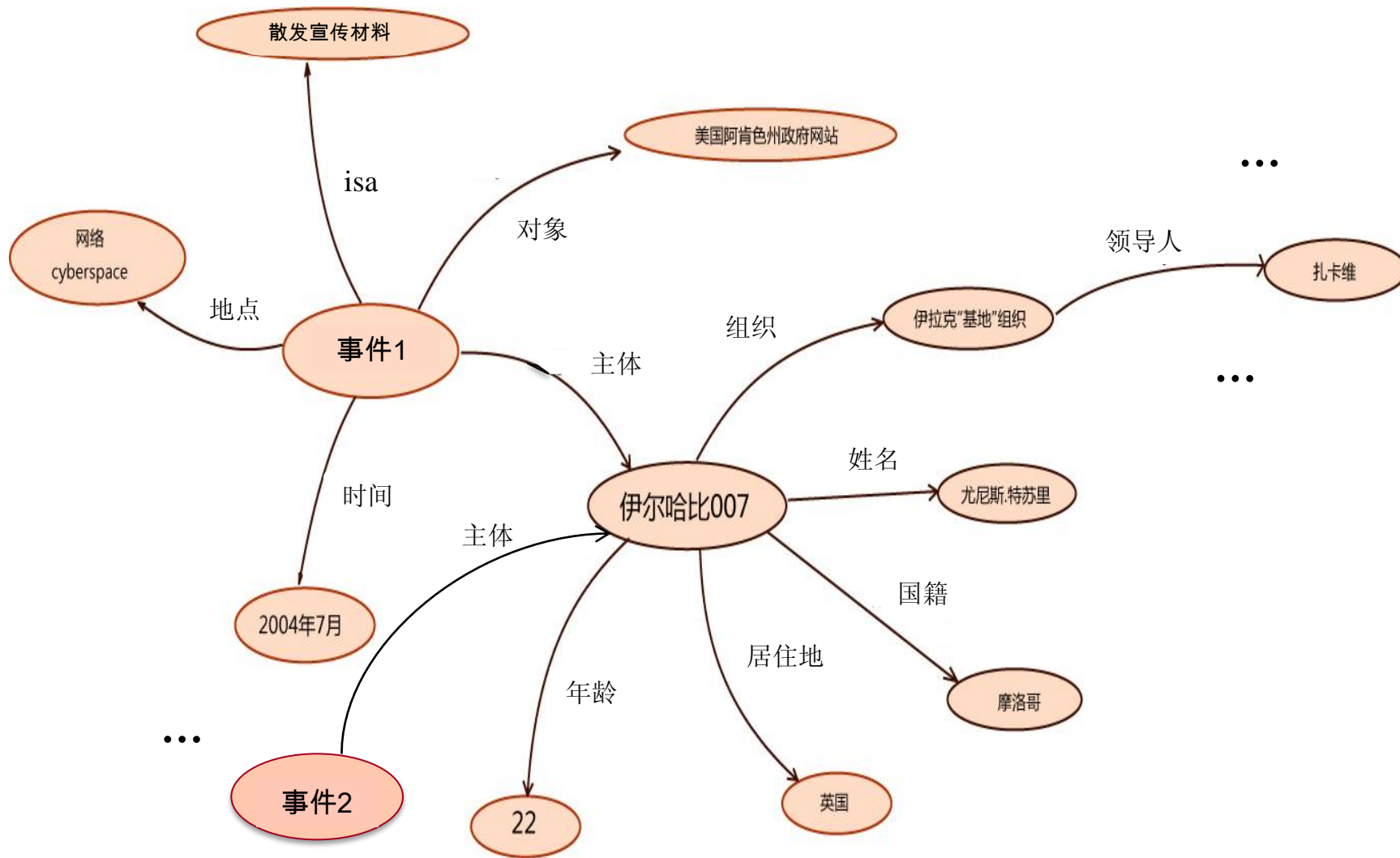


(4) situation 结点：表示一个情形或状况

“克莱德从春天到秋天拥有一个巢”



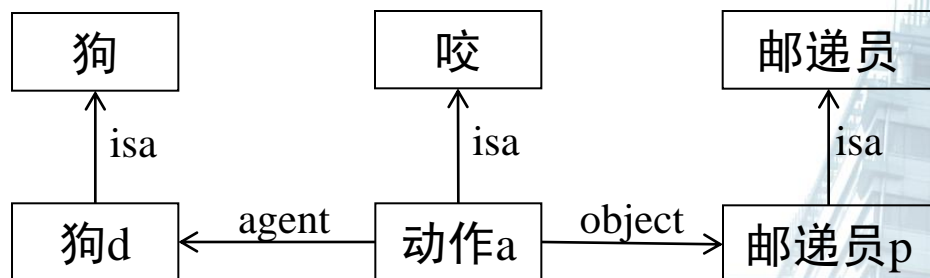
# 语义网络应用示例



# 语义网络表示

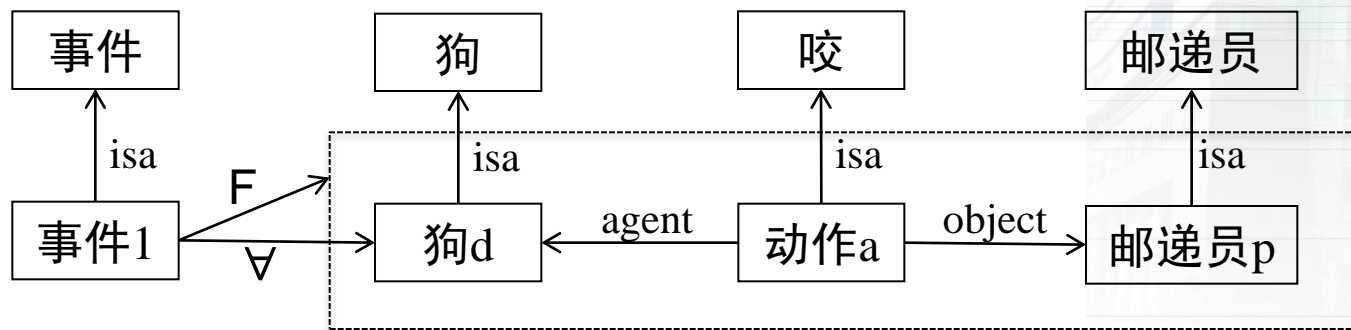
## (5) 逻辑联结词和量词的表示

“狗咬了邮递员”

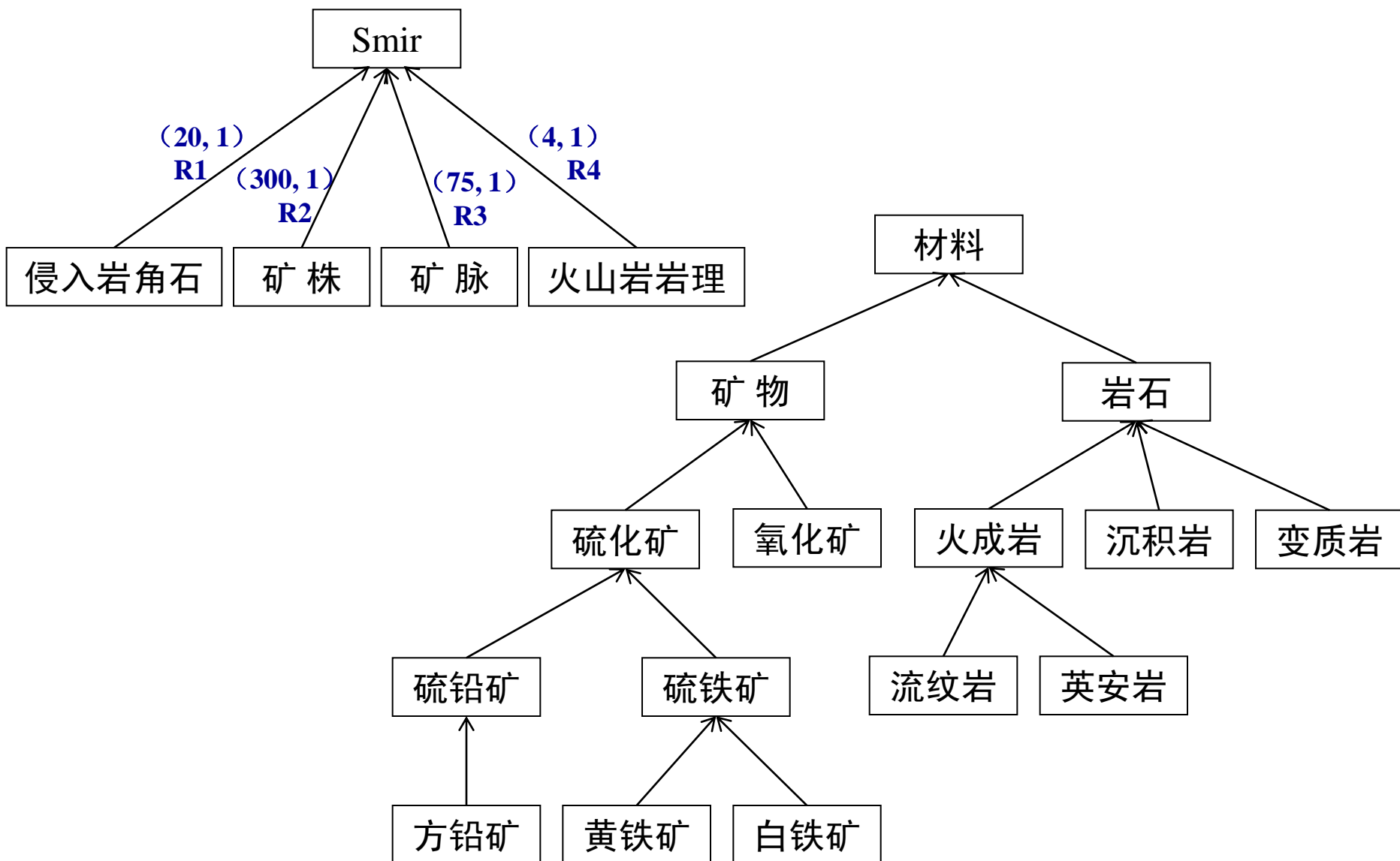


网络分块技术 (Hendrix) :

“所有的狗都咬过邮递员”



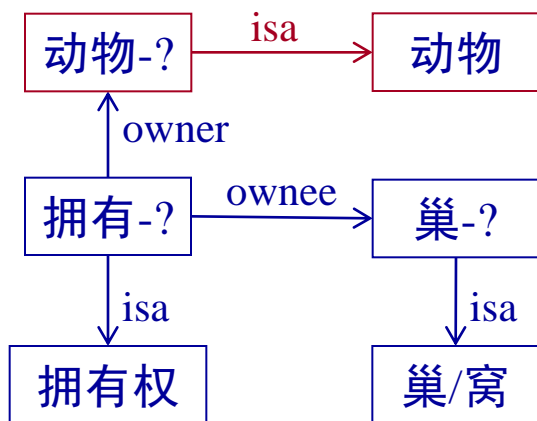
Prospector 系统中，用语义网络表示规则和分类知识：



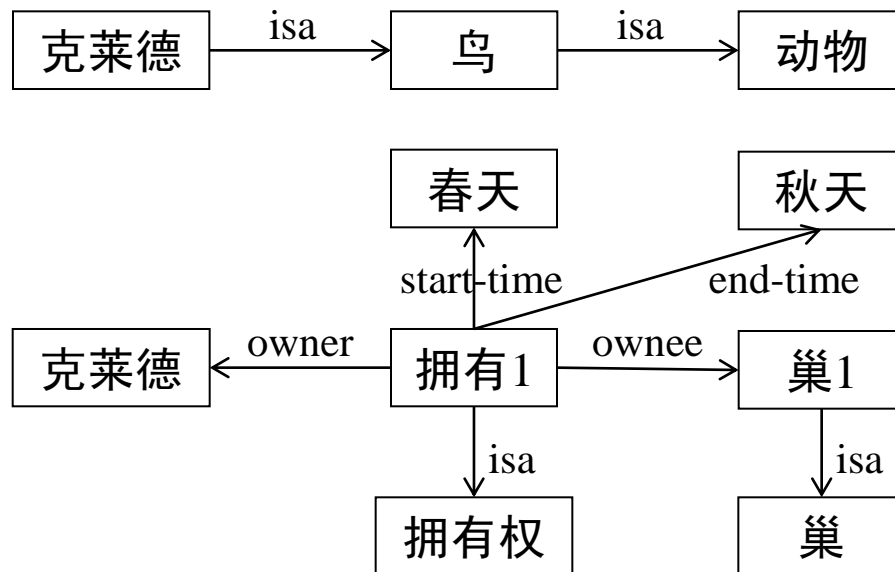
# 语义网络的推理

- 匹配：语义网络知识库与待求解问题的语义网络模式

问题：“是否有一只动物，它拥有一个窝？”



语义网络知识库：



- 继承

隶属关系下的属性继承、直接继承等



# 语义网络表示法的特点

- 优点

- 自然、直观，结构化
- 联想性
- 推理效率较高

- 主要弱点

- 缺乏严格的理论基础

缺乏严格的形式定义；对于网络结构，缺乏明确的语义解释

- 继承推理的非有效性

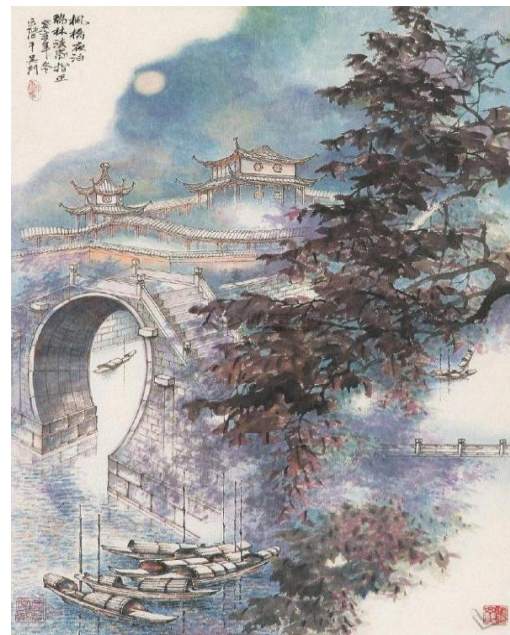
复杂情形下继承推理的冲突；非单调性、常识推理研究



# 作业·练习

用语义网络表示：

1. 孤舟蓑笠翁，独钓寒江雪。
2. 慈母手中线，游子身上衣。
3. 两个黄鹂鸣翠柳，一行白鹭上青天。
4. 姑苏城外寒山寺，夜半钟声到客船。



# 框架表示法

- **框架结构：** 由框架名和一些槽组成，每个槽有单值或多值

〈框架名〉

〈槽名<sub>1</sub>〉 〈侧面<sub>11</sub>〉 (值<sub>111</sub>, 值<sub>112</sub>, ...)

〈侧面<sub>12</sub>〉 (值<sub>121</sub>, 值<sub>122</sub>, ...)

...

〈槽名<sub>2</sub>〉 〈侧面<sub>21</sub>〉 (值<sub>211</sub>, 值<sub>212</sub>, ...)

〈侧面<sub>22</sub>〉 (值<sub>221</sub>, 值<sub>222</sub>, ...)

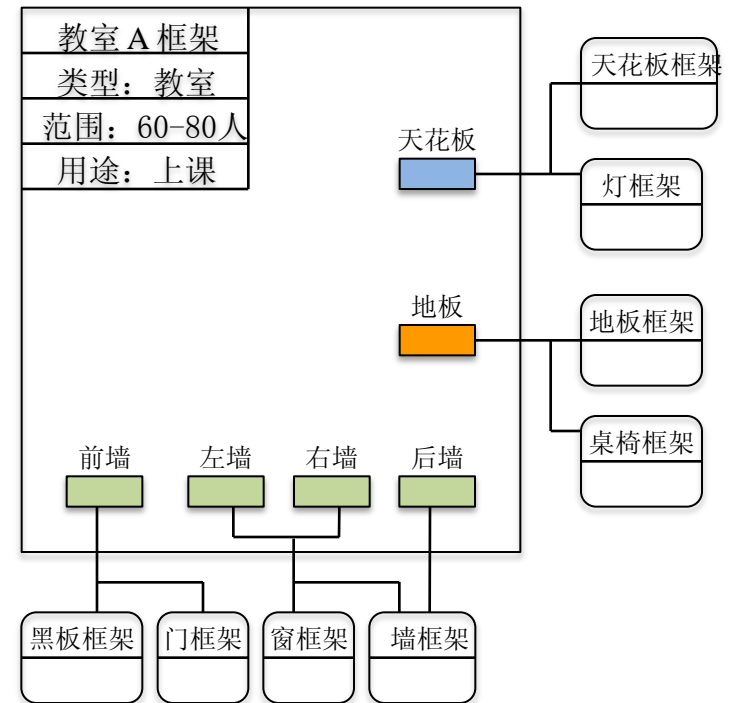
...

if needed (过程<sub>1</sub>)

...

if added (过程<sub>2</sub>)

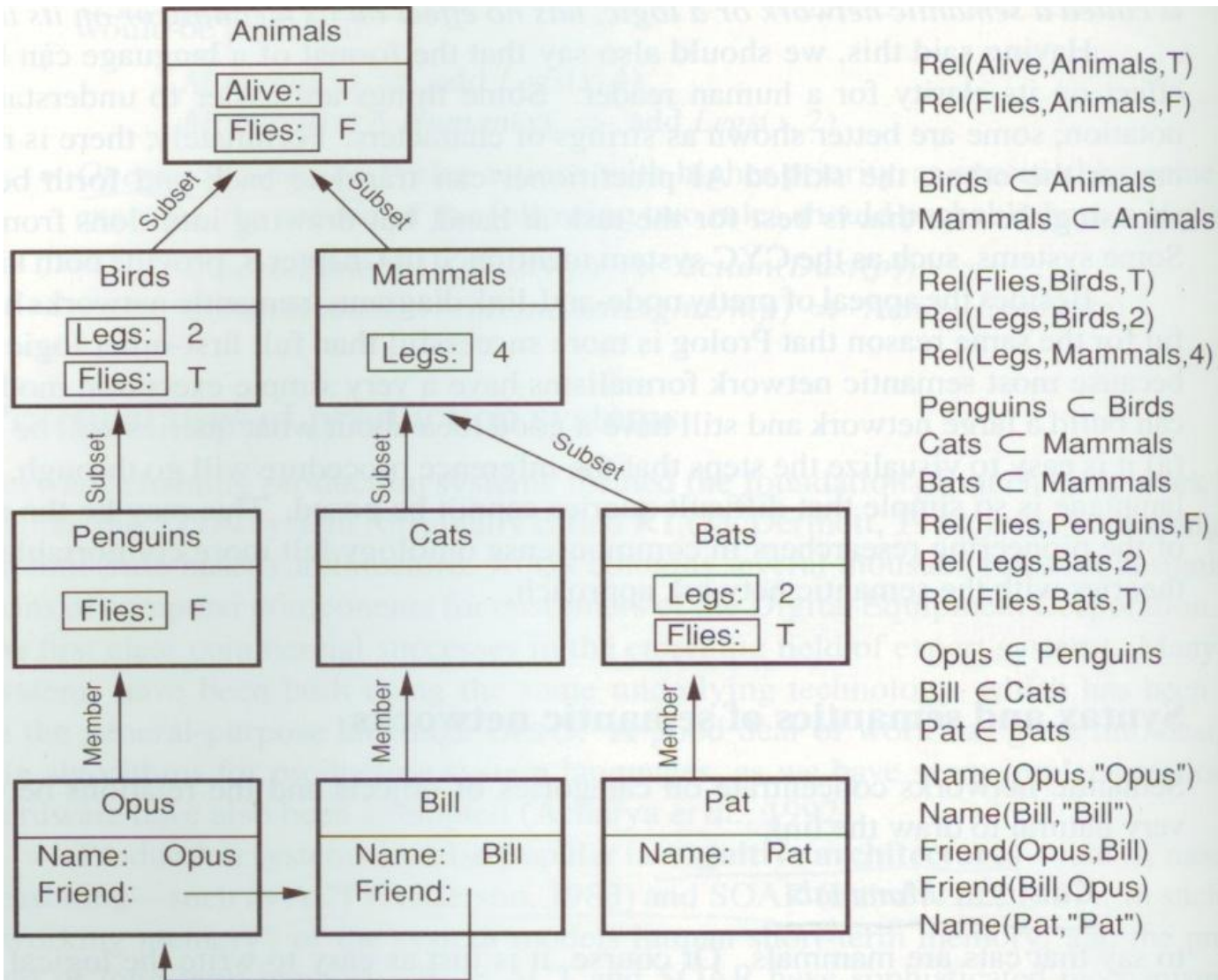
...



槽/侧面取值：逻辑值、数值、字符、默认值、子框架

if needed、if added 等槽的槽值是附加过程

# 框架系统示例



# 框架表示下的推理

- 匹配和继承

填槽方式:

- (1) 继承: 下层框架继承上层框架的属性
- (2) 默认: 如果没有相反的证据, 槽值取默认值
- (3) 附加过程:
  - if needed 槽: 用于说明如何计算得到槽值
  - if added 槽: 用于说明填槽值时需要完成哪些操作

从框架库中寻找适合当前问题的最佳候选框架。

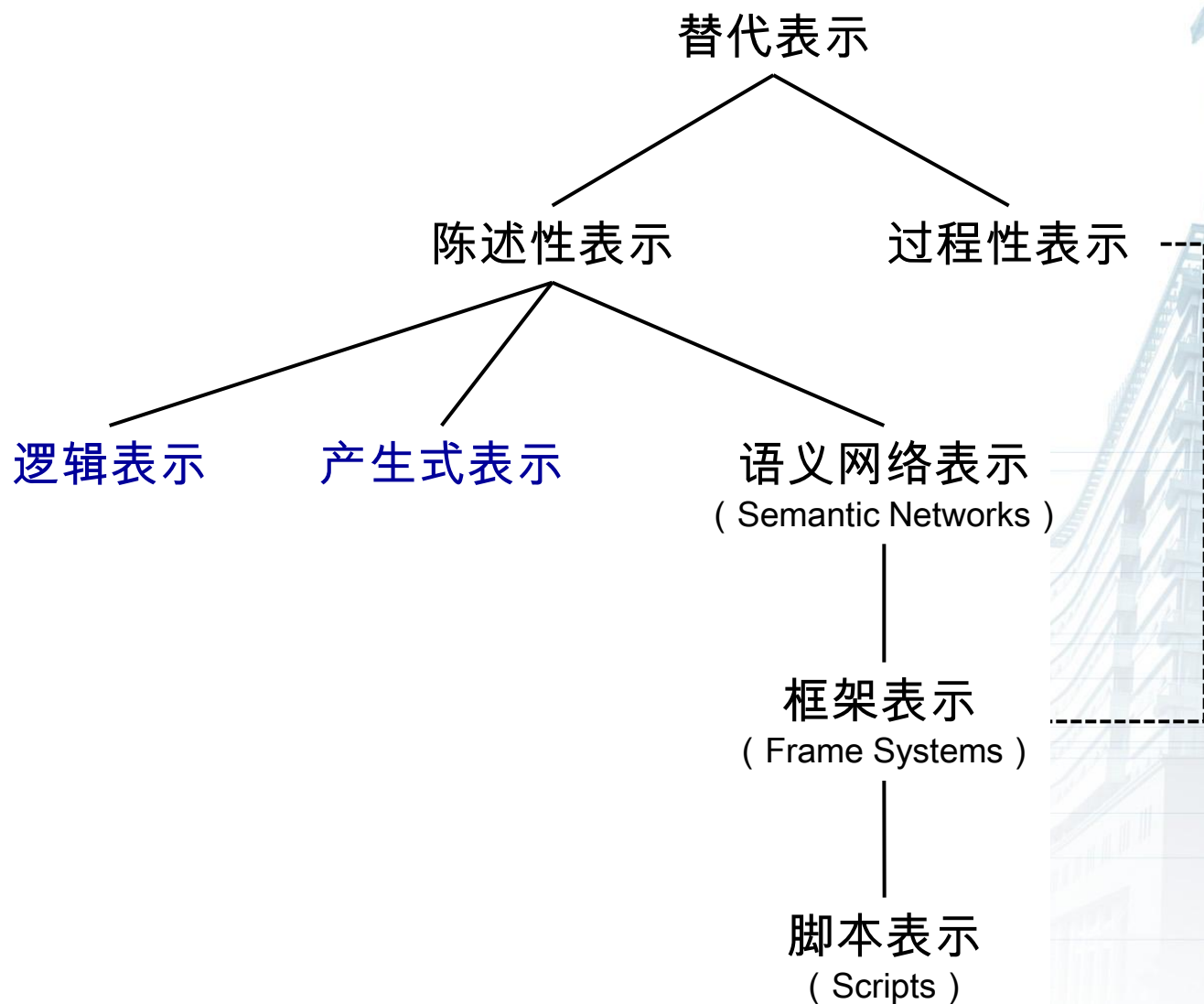


# 框架表示法的特点

- 优点：结构化知识、推理效率较高、陈述性与过程性表示结合
- 弱点：
  - 严格性不足、继承推理的非有效性
  - 许多实际情况与原型不符，对新的情况不易适应

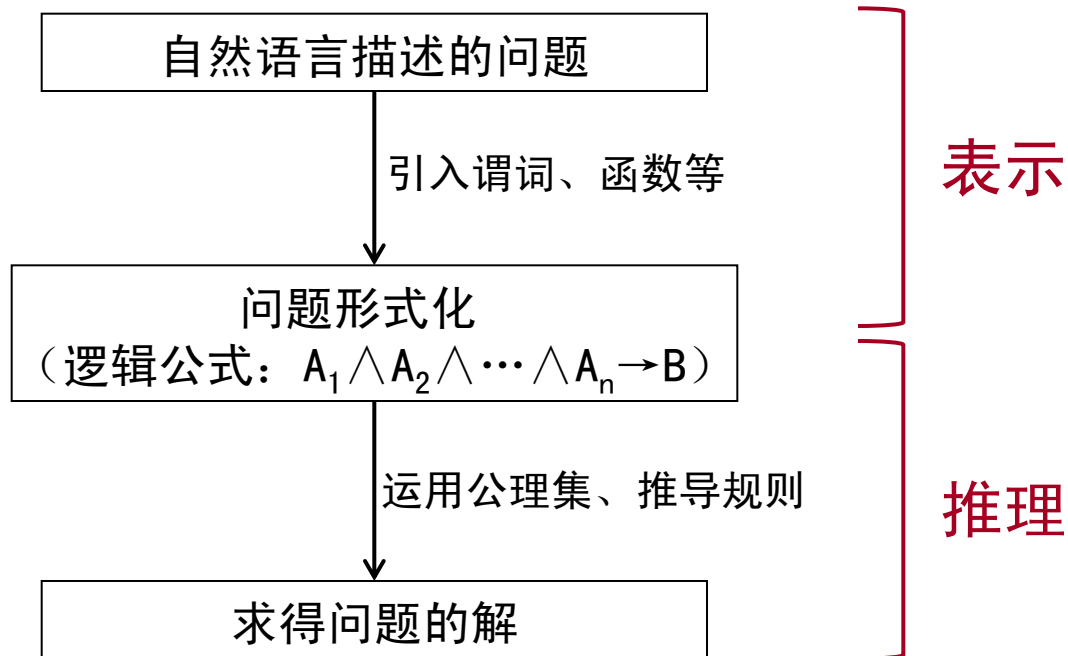
	产生式系统	框架系统
知识表示单位	相互独立的规则	框架结构
推理机制	固定、与知识库分离	可变、与知识库成一体
知识库构建	较容易	困难
系统通用性	较低	高
应用问题领域	简单问题	复杂问题
用户程度	初学者	专家

# 知识表示方法体系



# 逻辑表示法

- 逻辑法求解问题的一般步骤:



- 优点: 自然性、严格性、模块性
- 弱点: 难以表达结构性、多层次关联知识; 推理低效性



# 产生式表示法

- 事实

使用三元组（对象, 属性, 值）或（关系, 对象1, 对象2）表示

↑  
单个语言变量

↖ ↗  
多个语言变量

- 规则

采用“if <前件> then <后件>”的形式

↑  
条件部分, 事实 $A_i$ 的合取

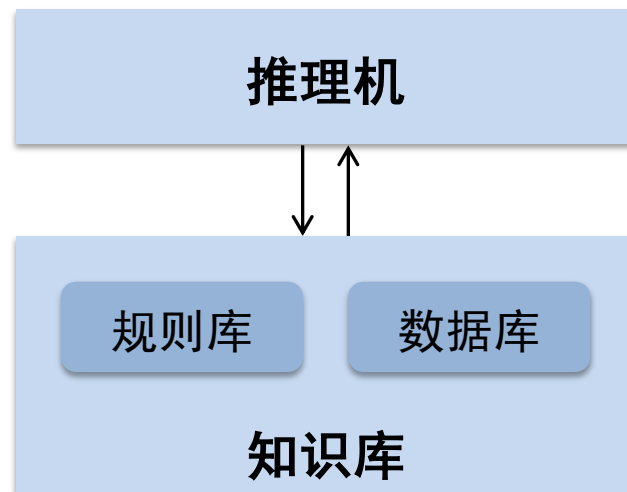
↑  
结论部分, 某一事实B

# 产生式系统的结构

- **知识库**

规则库：以产生式规则形式存放领域知识的存储区

数据库：存放初始事实、中间结果及最后结果的工作区



- **推理机：** 包含推理方式和控制策略，工作过程：

- (1) 数据库中的数据与规则库中的规则相匹配，得匹配规则集
  - (2) 从匹配规则集中选择一条规则作为使用规则
  - (3) 根据规则，执行相应的操作
- 重复这个过程，直到满足结束条件。

# 产生式系统的推理

- 正向推理（数据驱动、自底向上方式）

（1）规则前件与已知数据匹配

... ..

（3）使用规则的后件 → 数据库

... .. 结束条件：数据库中包含目标

**S**（包含目标）

⋮

↑  
**R**<sub>3</sub>

**S**<sub>2</sub>

↑  
**R**<sub>2</sub>

**S**<sub>1</sub>

↑  
**R**<sub>1</sub>

**S**<sub>0</sub>（初始事实）

# 产生式系统的推理

- 反向推理（目标驱动、自顶向下方式）

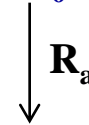
（1）规则后件与假设目标匹配

... ..

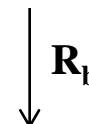
（3）使用规则的前件 → 子目标

... .. 结束条件：各子目标均为已知事实

$T_0$ （初始目标）



$T_1$



$T_2$



$\vdots$

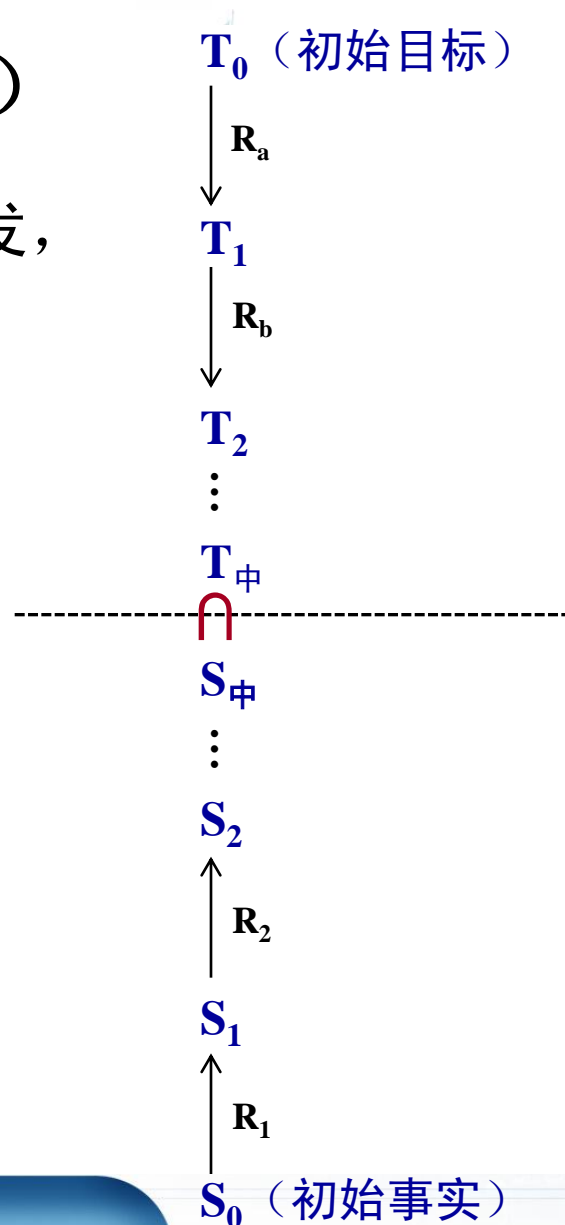
$T$ （已知事实）

# 产生式系统的推理

- **双向推理**（既自底向上又自顶向下）

既从已知事实出发，又从假设的目标出发，  
达到某个中间界面

结束条件：各子目标都在当前数据库中



# 产生式系统示例

- 一个识别食物的简单产生式系统

规则库中有5条规则：

R1: if 绿色 then 农产品

R2: if 装在小容器里 then 精美食品

R3: if 冷冻食品 or 农产品 then 易腐烂食品

R4: if 非精美食品 and 不易腐烂食品 then 大众食品

R5: if 重10斤 and 农产品 then 西瓜

推理方式：正向推理

控制策略：① 不产生冗余数据，② 按规则排序

结束条件：无可规则

数据库：（绿色，重10斤）

# 产生式系统示例

第1次循环：初始事实（绿色，重10斤）

1. 匹配规则集 {R1}
2. 选择 R1 作为使用规则
3. 数据库：（绿色，重10斤，农产品）

第2次循环：1. 匹配规则集 {R1, R3, R5}

2. 选择 R3 作为使用规则
3. 数据库：（绿色，重10斤，农产品，**易腐烂食品**）

第3次循环：1. 匹配规则集 {R1, R3, R5}

2. 选择 R5 作为使用规则
3. 数据库：（绿色，重10斤，农产品，**易腐烂食品**，西瓜）

第4次循环：无可规则，结束。

# 产生式系统示例

- 一个识别食物的简单产生式系统

规则库中有5条规则：

R1: if 绿色 then 农产品

R2: if 装在小容器里 then 精美食品

R3: if 冷冻食品 or 农产品 then 易腐烂食品

R4: if 非精美食品 and 不易腐烂食品 then 大众食品

R5: if 重10斤 and 农产品 then 西瓜

**R6: if 重10斤 and 易腐烂食品 then 鸡肉**

控制策略：① 不产生冗余数据，② 按规则排序

结束条件：无可规则

数据库：（绿色，重10斤）



# 产生式系统示例

第1次循环：初始事实（绿色，重10斤）

1. 匹配规则集 {R1}
2. 选择 R1 作为使用规则
3. 数据库：（绿色，重10斤，农产品）

第2次循环：1. 匹配规则集 {R1, R3, R5}

2. 选择 R3 作为使用规则
3. 数据库：（绿色，重10斤，农产品，易腐烂食品）

第3次循环：1. 匹配规则集 {R1, R3, R5, R6}

2. 选择 R5 作为使用规则
3. 数据库：（绿色，重10斤，农产品，易腐烂食品，西瓜）

第4次循环：使用规则 R6 → 鸡肉！

# 产生式表示法的特点

- 产生式表示格式固定，形式单一，规则间相互较为独立，使知识库的建立较为容易；同时知识库与推理机分离，易于对知识库进行增删、修改；推理方式单纯，对推理也容易作出解释；因此常作为建造知识系统的首选知识表示方法，尤其在一些较简单的知识系统中使用得最多
- 优点：自然性、模块性、有效性
- 弱点：
  - 推理过程的低效率，容易引起组合爆炸
  - 不能表达结构性的知识以及知识间的联系

# 知识表示小结

- 知识表示观

	认识论表示观	本体论表示观	知识工程表示观
核心观点	表示是对自然世界的描述	表示是对自然世界的近似描述	表示是对自然世界描述的计算机模型
	表示的唯一作用就是携带知识	承认表示的携带知识作用, 但认为不唯一, 强调一般知识	承认表示的携带知识作用, 但认为不唯一, 强调领域知识
	表示研究与启发式研究无关	启发式研究是表示研究的一部分	启发式研究至关重要
主要任务	对智能现象抽象与简洁的刻画, 常识的形式化和非单调推理	强调“聚焦”和本体论约定, 不依赖具体领域的一般性知识库	研究知识的存储与对其有效地使用(推理与搜索)
面临困难	推理过于复杂, 问题求解的非有效性	本体约定的相对性: 站在不同角度将导致不同的本体论约定	缺乏常识和一般知识, 脆弱性

# 知识表示小结

- 知识表示的粒度（Granularity）

知识表示：客观事物到推理内部的一种映射  
应与问题领域的知识特性相吻合，以最大程度地减少失真

- 知识库的组织

为提高对其知识进行检索和匹配的效率，将知识库按某种原则组织起来。例如：MYCIN 系统对每个临床参数建一个特性表：

---

染色斑

... ..

LOOKAHEAD: 指明哪些规则的前提涉及该参数

UPDATED-BY: 指出从哪些规则的结论部分可得到该参数

... ..

---

# 知识表示小结

- 知识表示学习

知识图谱本质上是一种语义网络

- 概念层次关系学习
- 实体识别与链接
- 事实知识学习
- 事件知识学习

- 知识表示与推理

基于知识的推理→演绎推理：确定、严格、一致  
知识不确定、不一致、不完全情形下的推理



谢谢大家！

Thank  
You!!

