

# 第10章 知识表达与处理方法

人工智能的求解是以知识为基础的，一个程序具备的知识越多，它的求解问题的能力就越强，所以知识表示是人工智能研究的一个重要课题。

# 10.1 概述

## 10.1.1 知识和知识表示的含义

### 1. 知识的定义

知识是人们对于可重复信息之间联系的认识，它是信息经过加工整理、解释、挑选和改造而形成的。比较有代表性的定义有：

- Feigenbaum：知识是经过裁剪、塑造、解释、选择和转换了的信息。
- Bernstein：知识由特定领域的描述、关系和过程组成。
- Heyes-Roth：知识=事实+信念+启发式。

# 10.1 概述

## 10.1.1 知识和知识表示的含义

- 知识：把有关信息关联在一起所形成的信息结构。
- 知识反映了客观世界中事物之间的信息关联形式：“如果……，则……”不同关系形成了不同的知识。

如果大雁向南飞，则冬天就要来临了。

例如：

“雪是白色的” —— 事实

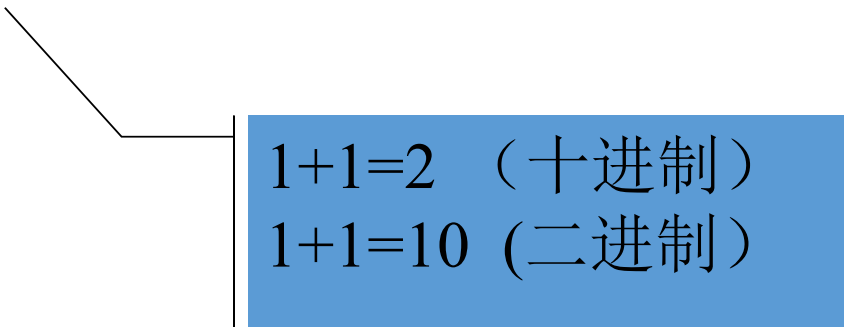
“如果头痛且流涕，则有可能患了感冒” —— 规则

# 10.1 概述

## 知识的特性

### (1) 相对正确性

任何知识都是在一定的条件及环境下产生的，在这种条件及环境下才是正确的。



$1+1=2$  （十进制）  
 $1+1=10$  （二进制）

## 10.1 概述

### 知识的特性

#### (2) 不确定性

知识状态:

确定性: “真”、“假”

不确定性: 真与假之间的中间状态

- ① 随机性引起的不确定性
- ② 模糊性引起的不确定性
- ③ 经验引起的不确定性
- ④ 不完全性引起的不确定性

# 10.1 概述

---

## 知识的特性

### (3) 可表示性与可利用性

- 知识的可表示性：知识可以用适当形式表示出来，如用语言、文字、图形、神经网络等。
- 知识的可利用性：知识可以被利用。

# 10.1 概述

## 10.1.1 知识和知识表示的含义

### 2. 知识的表示

知识表示是指将知识符号化并输入给计算机的过程和方法。

- (1) 用给定的结构，按一定的原则、组织方式表示知识。
- (2) 解释所表示知识的含义。

# 10.1 概述

## 10.1.1 知识和知识表示的含义

### 2. 知识的表示

在AI领域，研究知识表示方法的目的是用知识来改善程序的性能，具体表现为：

- 利用知识来帮助选择或限制程序搜索的范围；
- 利用知识来帮助程序识别、判断、规划与学习。

AI的研究集中在如何使程序拥有知识而具有智能行为。



# 10.1 概述

## 10.1.1 知识和知识表示的含义

### 3. 智能系统中的知识

要使计算机系统具有智能，一般来说至少应使系统拥有以下几个方面的知识：

- 关于对象（Object）的知识，如火山、人。知识表示应能表示各种知识对象以及对象的类型、性质等。
- 事件（Event），能表示事件的时序，因果关系等，如：天下雨，火山在一定条件下会爆发等。
- 行为（Performance），如何写文章，如何造句，如何证明定理等。
- 元知识（Meta knowledge），是关于知识的知识，即什么是知识，以及如何运用知识。

# 10.1 概述

## 10.1.2 AI中知识表示方法分类

AI中知识表示方法注重知识的运用，所以将知识表示方法粗略地可分为以下两大类。

### 1. 过程式知识表示（procedure）

- 过程性知识是一般是表示如何做的知识，是有关系统变化、问题求解过程的操作、演算和行为的知识。这种知识是隐含在程序之中的，机器无法从程序的编码中抽取这些知识。
- 过程式知识表示描述表示控制规则和控制结构的知识，给出一些客观规律，告诉怎么作。

# 10.1 概述

## 10.1.2 AI中知识表示方法分类

### 2. 陈述式知识表示 (declarative)

- 陈述式知识描述系统状态、环境和条件，以及问题的概念、定义和事实。
- 陈述式知识表示描述这种事实性知识，描述客观事物所涉及的对象是什么，有时有必要给出对象之间的联系。它的表示与知识运用（推理）是分开处理的。这种知识是显式表示的，如：

- $\text{isa}(\text{John}, \text{man})$

- $\text{isa}(\text{ABC}, \text{triangle}) \rightarrow \text{cat}(a, b) \wedge \text{cat}(b, c) \wedge \text{cat}(c, a)$

# 10.1 概述

## 10.1.3 AI对知识表示方法的要求

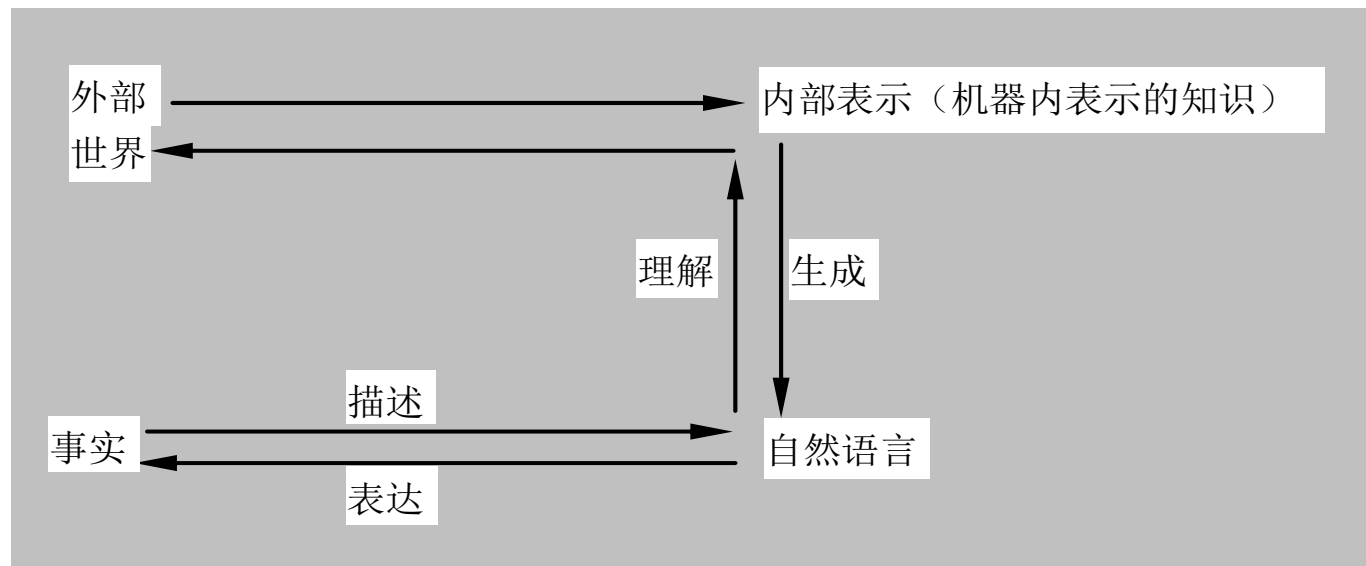
从知识利用上讲，衡量知识表示方法可以从以下三个方面考察：

- 便于获取，便于表示新知识，并以合适方式与已有知识相连接。
- 便于搜索，在求解问题时，能够较快地在知识库中找出有关知识。因此，知识库应具有较好的记忆组织结构。
- 便于推理，要能够从已有知识中推出需要的答案或结论。

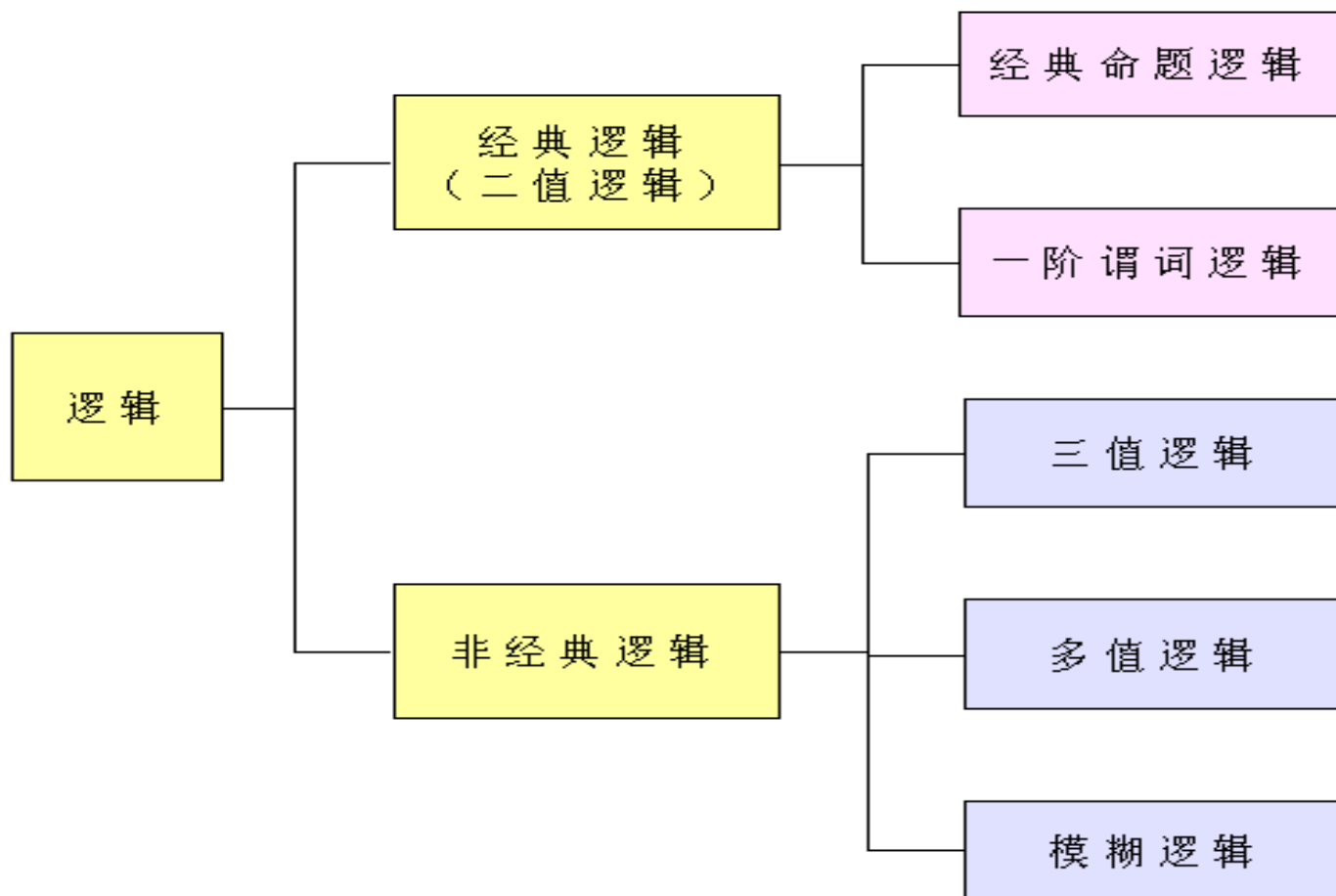
# 10.1 概述

## 10.1.4 知识表示要注意的问题

理想的内部表示要能真实的直接反映外部世界的事实，内部知识表示与自然语言和外部世界三者之间的关系可用下图描述。



## 10.2 逻辑表示法



# 命题

例如：  $3 < 5$

命题 (proposition)：一个非真即假的陈述句。

- 若命题的意义为真，称它为真命题。例如：太阳从西边升起

- 若命题的意义为假，称它为假命题。例：  $1 + 1 = 10$  记为  $F$ 。

- 一个命题可在一种条件下为真，在另一种条件下为假。

$P$ ：北京是中华人民共和国的首都

- 命题逻辑：研究命题及命题之间关系的符号逻辑系统。

- 命题逻辑表示法：无法把它所描述的事物的结构及逻辑特征反映出来，也不能把不同事物间的共同特征表述出来。

$P$ ：老李是小李的父亲

$P$ ：李白是诗人

$Q$ ：杜甫也是诗人

# 谓词

谓词的一般形式： $P(x_1, x_2, \dots, x_n)$

- 个体  $x_1, x_2, \dots, x_n$ ：某个独立存在的事物或者某个抽象的概念；
- 谓词名  $P$ ：刻画个体的性质、状态或个体间的关系。

(1) 个体是常量：一个或者一组指定的个体。

- “老张是一个教师”：一元谓词 *Teacher (Zhang)*
- “ $5 > 3$ ”：二元谓词 *Greater (5, 3)*
- “Smith作为一个工程师为IBM工作”：  
三元谓词 *Works (Smith, IBM, engineer)*



# 谓词

(2) 个体是变元（变量）：没有指定的一个或者一组个体。

“ $x < 5$ ” :  $Less(x, 5)$

(3) 个体是函数：一个个体到另一个个体的映射。

■ “小李的父亲是教师” :  $Teacher(father(Li))$

(4) 个体是谓词

■ “Smith作为一个工程师为IBM工作” :  
二阶谓词  $Works(engineer(Smith), IBM)$

# 谓词公式

## 1. 连接词（连词）

(1)  $\neg$ : “机器人不在2号房间” :  $\neg \text{Inroom}(\text{robot}, r2)$

(2)  $\vee$ : “李明打篮球或踢足球” :  
 $\text{Plays}(\text{Liming}, \text{basketball}) \vee \text{Plays}(\text{Liming}, \text{football})$

(3)  $\wedge$ : “合取”（conjunction）——与。

“我喜欢音乐和绘画” :  
 $\text{Like}(I, \text{music}) \wedge \text{Like}(I, \text{painting})$

# 谓词公式

## 1. 连接词（连词）

(4)  $\rightarrow$ : “蕴含” (implication) 或 “条件” (condition)。

$P \rightarrow Q$ :  $P$  称为条件的前件,  $Q$  称为条件的后件

“如果刘华跑得最快, 那么他取得冠军。” :

$RUNS(Liuhua, faster) \rightarrow WINS(Liuhua, champion)$

(5)  $\leftrightarrow$ : “等价” (equivalence) 或 “双条件”

(bicondition) 。

$P \leftrightarrow Q$ : “ $P$ 当且仅当 $Q$ ”。

# 谓词公式

## 1. 连接词（连词）

谓词逻辑真值表

P	Q	$\neg P$	$P \vee Q$	$P \wedge Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
T	T	F	T	T	T	T
T	F	F	T	F	F	F
F	T	T	T	F	T	F
F	F	T	F	F	T	T

# 谓词公式

## 2. 量词 (quantifier)

(1) 全称量词 (universal quantifier) ( $\forall x$ ) : “对个体域中的所有 (或任一个) 个体  $x$ ”。

“所有的机器人都是灰色的” :

$$(\forall x)[\text{ROBOT}(x) \rightarrow \text{COLOR}(x, \text{GRAY})]$$

(2) 存在量词 (existential quantifier) ( $\exists x$ ) : “在个体域中存在个体  $x$ ”。

“1号房间有个物体” :

$$(\exists x) \text{INROOM}(x, r1)$$

# 谓词公式

## 全称量词和存在量词举例：

- $(\forall x)(\exists y) F(x, y)$  表示对于个体域中的任何个体 $x$ 都存在个体 $y$ ， $x$ 与 $y$ 是朋友。
- $(\exists x)(\forall y) F(x, y)$  表示在个体域中存在个体 $x$ ，与个体域中的任何个体 $y$ 都是朋友。
- $(\exists x)(\exists y) F(x, y)$  表示在个体域中存在个体 $x$ 与个体 $y$ ， $x$ 与 $y$ 是朋友。
- $(\forall x)(\forall y) F(x, y)$  表示对于个体域中的任何两个个体 $x$ 和 $y$ ， $x$ 与 $y$ 都是朋友。

## 谓词公式

全称量词和存在量词出现的次序将影响命题的意思。

例如：

- $(\forall x)(\exists y)(Employee(x) \rightarrow Manager(y, x)) :$   
“每个雇员都有一个经理。”
- $(\exists y)(\forall x)(Employee(x) \rightarrow Manager(y, x)) :$   
“有一个人是所有雇员的经理。”

# 谓词公式

## 3. 谓词公式

• 定义 可按下述规则得到谓词演算的谓词公式：

- (1) 单个谓词是谓词公式，称为原子谓词公式。
- (2) 若A是谓词公式，则  $\neg A$  也是谓词公式。
- (3) 若A, B都是谓词公式，则  $A \wedge B$ ,  $A \vee B$ ,  $A \rightarrow B$ ,  $A \leftrightarrow B$  也都是谓词公式。
- (4) 若A是谓词公式，则  $(\forall x) A$ ,  $(\exists x) A$  也是谓词公式。
- (5) 有限步应用 (1) — (4) 生成的公式也是谓词公式。

连接词的优先级别从高到低排列：

$\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$ ,  $\leftrightarrow$



# 谓词公式

## 4. 量词的辖域

- 量词的辖域：位于量词后面的单个谓词或者用括弧括起来的谓词公式。
- 约束变元与自由变元：辖域内与量词中同名的变元称为约束变元，不同名的变元称为自由变元。

■ 例如：

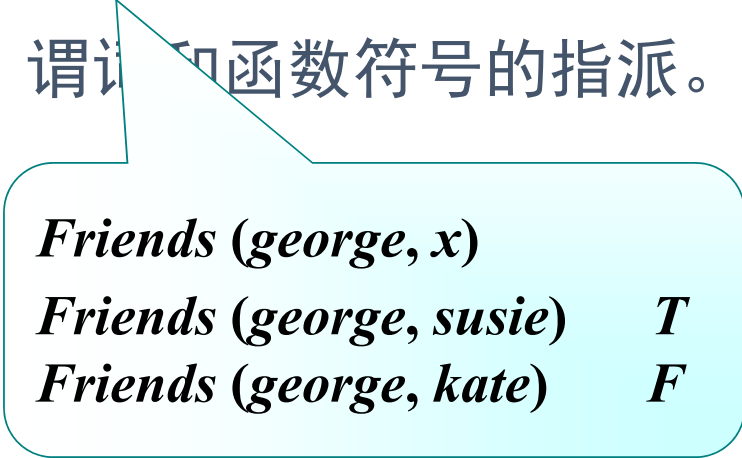
$$(\exists x)(P(x, y) \rightarrow Q(x, y)) \vee R(x, y)$$

- $(P(x, y) \rightarrow Q(x, y))$ ： $(\exists x)$ 的辖域，辖域内的变元 $x$ 是受 $(\exists x)$ 约束的变元， $R(x, y)$ 中的 $x$ 是自由变元。
- 公式中的所有 $y$ 都是自由变元。

# 谓词公式的性质

## 1. 谓词公式的解释

- 谓词公式在个体域上的解释：个体域中的实体对谓词演算表达式的每个常量、变量、谓词和函数符号的指派。



<i>Friends (george, x)</i>	
<i>Friends (george, susie)</i>	<i>T</i>
<i>Friends (george, kate)</i>	<i>F</i>

- 对于每一个解释，谓词公式都可求出一个真值（*T*或*F*）。

# 谓词公式的性质

## 2. 谓词公式的永真性、可满足性、不可满足性

- 定义 如果谓词公式 $P$ 对个体域 $D$ 上的任何一个解释都取得真值 $T$ ，则称 $P$ 在 $D$ 上是永真的；如果 $P$ 在每个非空个体域上均永真，则称 $P$ 永真。
- 定义 如果谓词公式 $P$ 对个体域 $D$ 上的任何一个解释都取得真值 $F$ ，则称 $P$ 在 $D$ 上是永假的；如果 $P$ 在每个非空个体域上均永假，则称 $P$ 永假。
- 定义 对于谓词公式 $P$ ，如果至少存在一个解释使得 $P$ 在此解释下的真值为 $T$ ，则称 $P$ 是可满足的，否则，则称 $P$ 是不可满足的。

# 谓词公式的性质

## 3. 谓词公式的等价性

- 定义2.6 设 $P$ 与 $Q$ 是两个谓词公式， $D$ 是它们共同的个体域，若对 $D$ 上的任何一个解释， $P$ 与 $Q$ 都有相同的真值，则称公式 $P$ 和 $Q$ 在 $D$ 上是等价的。如果 $D$ 是任意个体域，则称 $P$ 和 $Q$ 是等价的，记为 $P \Leftrightarrow Q$ 。

# 谓词公式的等价性

(1) 交换律

$$P \wedge Q \Leftrightarrow Q \wedge P$$

$$P \vee Q \Leftrightarrow Q \vee P$$

(2) 结合律

$$(P \wedge Q) \wedge R \Leftrightarrow P \wedge (Q \wedge R)$$

$$(P \vee Q) \vee R \Leftrightarrow P \vee (Q \vee R)$$

(3) 分配律

$$P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$$

$$P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$$

(4) 德·摩根定律

$$\neg (P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$$

$$\neg (P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$$

(5) 双重否定

$$\neg(\neg P) \Leftrightarrow P$$

(6) 吸收律

$$P \vee (P \wedge Q) \Leftrightarrow P$$

$$P \wedge (P \vee Q) \Leftrightarrow P$$

(7) 补余律（否定律）

$$P \vee \neg P \Leftrightarrow T$$

$$P \wedge \neg P \Leftrightarrow F$$

(8) 连接词化归律

$$P \rightarrow Q \Leftrightarrow \neg P \vee Q$$

(9) 逆否律

$$P \rightarrow Q \Leftrightarrow \neg Q \rightarrow \neg P$$

# 谓词公式的等价性

(10) 量词转换律

$$\neg(\exists x)P(x) \Leftrightarrow (\forall x)\neg P(x)$$

$$\neg(\forall x)P(x) \Leftrightarrow (\exists x)\neg P(x)$$

(11) 量词分配律

$$(\exists x)(A(x) \vee B(x)) \Leftrightarrow (\exists x)A(x) \vee (\exists x)B(x)$$

$$(\forall x)(A(x) \wedge B(x)) \Leftrightarrow (\forall x)A(x) \wedge (\forall x)B(x)$$

# 谓词公式的性质

## 4. 谓词公式的永真蕴含

■ 定义 对于谓词公式 $P$ 与 $Q$ ，如果 $P \rightarrow Q$ 永真，则称公式 $P$ 永真蕴含 $Q$ ，且称 $Q$ 为 $P$ 的逻辑结论，称 $P$ 为 $Q$ 的前提，记为 $P \Rightarrow Q$ 。

# 永真蕴涵

## (1) 假言推理

$$P, P \rightarrow Q \Rightarrow Q$$

即由P为真及 $P \rightarrow Q$ 为真，可推出Q为真。

## (2) 拒取式推理

$$\neg Q, P \rightarrow Q \Rightarrow \neg P$$

## (3) 假言三段论

$$P \rightarrow Q, Q \rightarrow R \Rightarrow P \rightarrow R$$



# 永真蕴涵

(4) 全称固化

$$(\forall x) P(x) \Rightarrow P(y)$$

(5) 存在固化

$$(\exists x) P(x) \Rightarrow P(y)$$

(6) 反证法

定理：Q为 $P_1, P_2, \dots, P_n$ 的逻辑结论，当且仅当  $(P_1 \wedge P_2 \wedge \dots \wedge P_n) \wedge \neg Q$ 是不可满足的。

# 一阶谓词逻辑知识表示方法

谓词公式表示知识的步骤：

- (1) 定义谓词及个体。
- (2) 变元赋值。
- (3) 用连接词连接各个谓词，形成谓词公式。

**例：**用一阶谓词逻辑表示“每个储蓄钱的人都得到利息”

解：定义谓词：save (x) 表示x储蓄钱，interest (x) 表示x获得利息。

则“每个储蓄钱的人都得到利息”可以表示为：

$$(\forall x) (\text{save}(x) \rightarrow \text{interest}(x))$$

# 一阶谓词逻辑知识表示方法

## ■ 用一阶谓词表示：

- 例如：用一阶谓词逻辑表示下列关系数据库。

住户	房间	电话号码	房间
Zhang	201	491	201
Li	201	492	201
Wang	202	451	202
Zhao	203	451	203

*Occupant*

*Telephone*

*Occupant* (Zhang , 201)

*Occupant* (Li, 201)

*Occupant* (Wang, 202)

*Occupant* (Zhao, 203)

*Telephone* (491, 201)

*Telephone* (492, 201)

*Telephone* (451, 202)

*Telephone* (451, 203)

# 一阶谓词逻辑知识表示方法

例10.1 已知命题公式集合s可以转化为如下2条子句。

起大风(5日) (1)

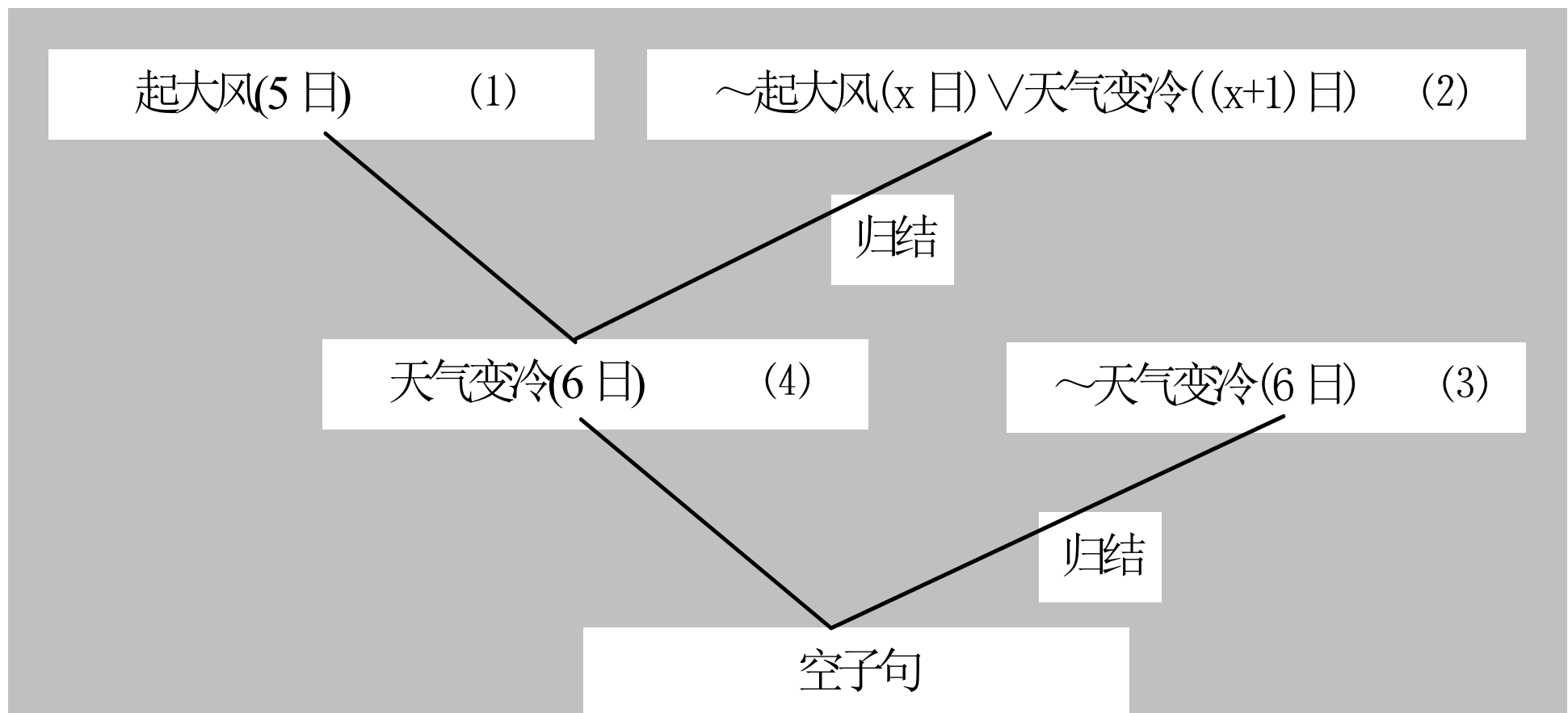
$\sim$ 起大风(x日)  $\vee$  天气变冷((x+1)日) (2)

根据(1)是否可以推断6日天气要变冷呢？可以用如下的反证法予以证明。

首先假设6日天气不会变冷，用公式表示为

$\sim$ 天气变冷(6日)

# 一阶谓词逻辑知识表示方法



# 一阶谓词逻辑表示法的特点

## 优点：

- ① 自然性
- ② 精确性
- ③ 严密性
- ④ 容易实现

## 局限性：

- ① 不能表示不确定的知识
- ② 组合爆炸
- ③ 效率低

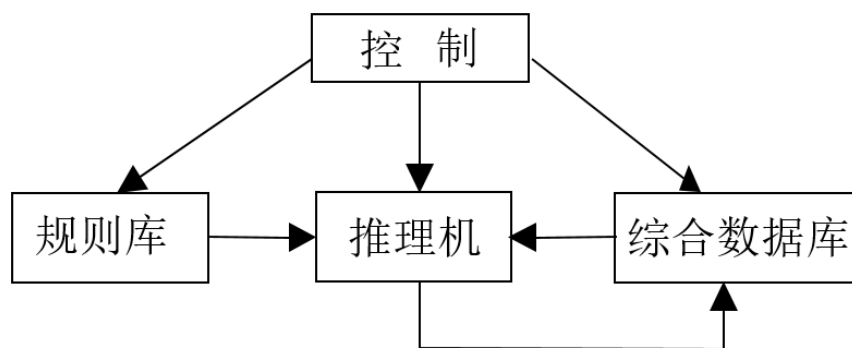
## □ 应用：

- (1) 自动问答系统（Green等人研制的QA3系统）
- (2) 机器人行动规划系统（Fikes等人研制的STRIPS系统）
- (3) 机器博弈系统（Filman等人研制的FOL系统）
- (4) 问题求解系统（Kowalski等设计的PS系统）

## 10.3产生式表示法

- “产生式”：1943年，美国数学家波斯特（E. Post）首先提出。
- 1972年，纽厄尔和西蒙在研究人类的认知模型中开发了基于规则的产生式系统。
- 产生式通常用于表示事实、规则以及它们的不确定性度量，适合于表示事实性知识和规则性知识。

# 产生式系统的基本结构



- **综合数据库**(事实库、上下文、黑板等): 一个用于存放问题求解过程中各种当前信息的数据结构。

- **控制系统**（推理机构）：由一组程序组成，负责整个产生式系统的运行，实现对问题的求解。

- 所完成的工作： 匹配 冲突消解 执行某一规则  
检查推理终止条件

- **规则库**: 用于描述相应领域内知识的产生式集合。

- 基本形式: IF P THEN Q

或者  $P \rightarrow Q$

例如: r4: IF 动物会飞 AND 会下蛋  
THEN 该动物是鸟



# 产生式系统的知识表示

产生式系统的知识表示方法，包括事实的表示和规则的表示。

## 1. 事实的表示

### (1) 孤立事实的表示

孤立事实通常用三元组（对象，属性，值）或（关系，对象 1，对象 2）表示，其中对象就是语言变量。

# 产生式系统的知识表示

**例：**老李年龄是40岁： (Li, age, 40)

若要表示老王、老张是朋友，则可表示成： (friend, Wang, Zhang)

如果增加不确定的度量，可增加一个因子表示两人友谊的可信度。如

(friendship, Wang, Zhang, 0.8) 可理解为王、张二人友谊的可信度为0.8。

**例：**MyCIN系统 (特性，对象，取值)

(GRAMSTAIN, ORGANISM, GRAM-POSITIVE) 细菌染色是革兰氏阳性

(DOSE, DRUG, 20-GRAMS) 用药的剂量是2.0克

(IDENT, ORGANISM-2, STREPTOCOCCUS, 0.7) 细菌-2是链球菌的置信度为0.7

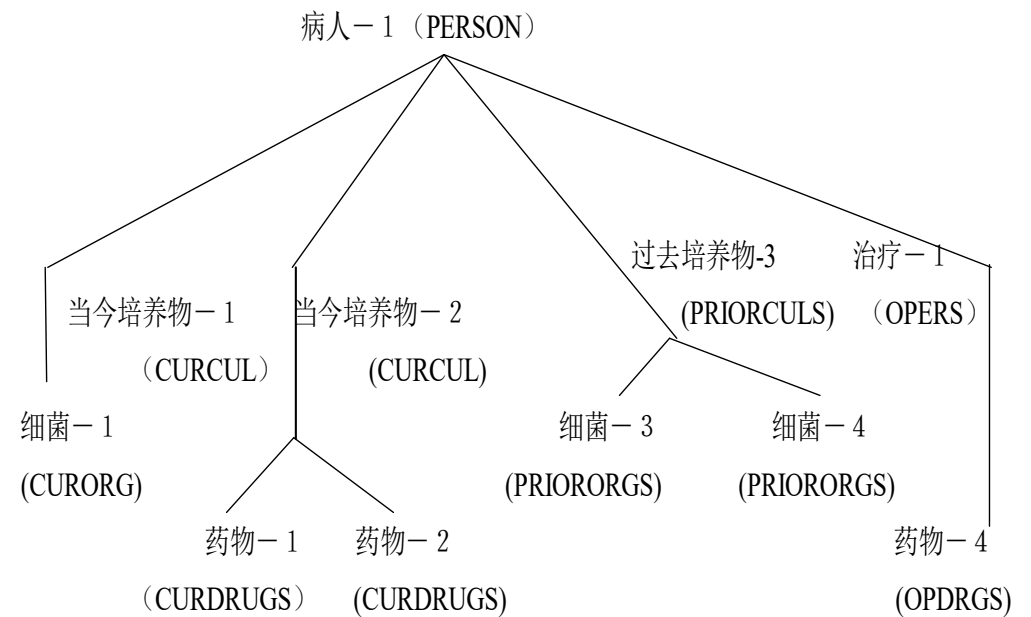
# 产生式系统的知识表示

## 1. 事实的表示

### (2) 有关联事实的表示

#### a. 树型结构

在MYCIN系统中表示事实用的是四元组，为了查找的方便，它把不同的对象（即上下文）按层次组成一种上下文树。



# 产生式系统的知识表示

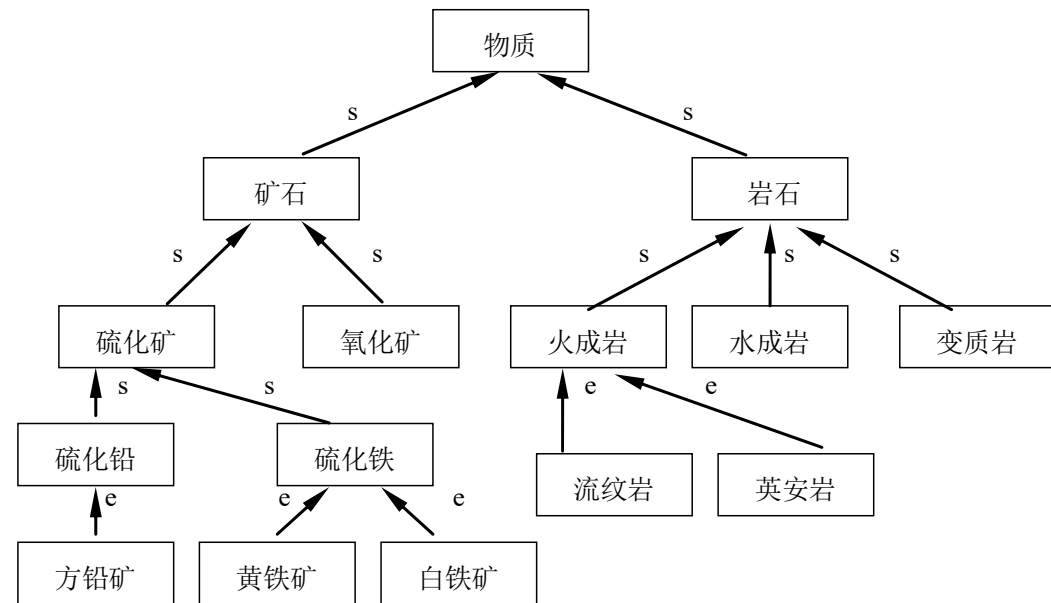
## 1. 事实的表示

### (2) 有关联事实的表示

#### b. 网状结构

在PROSPECTOR探矿系统中，整个静态知识以语义网络的结构把相关的知识连在一起，这样就使查找更加方便了。

其中s表示子集关系， $\langle \text{subset } x \ y \rangle$ 表示y是x的子集；e表示成员关系， $\langle \text{element } x \ y \rangle$ 表示y是x的成员。



# 产生式系统的知识表示

## 2. 规则的表示

### (1) 单个规则的表示

- **前项**由逻辑连接词组成各种不同的前提条件；
- **后项**表示前提条件为真时，应采取的行为或所得的结论。
- 如果考虑不精确推理，则可考虑附加**置信度量值**。

MYCIN系统中的规则定义为：

$\langle \text{rule} \rangle = (\text{IF } \langle \text{antecedent} \rangle \text{ THEN } \langle \text{action} \rangle \text{ ELSE } \langle \text{action} \rangle)$

其中各部分的定义分别为：

$\langle \text{antecedent} \rangle = (\text{AND } \{ \langle \text{condition} \rangle \})$

$\langle \text{condition} \rangle = (\text{OR } \{ \langle \text{condition} \rangle \} | (\langle \text{predicate} \rangle \langle \text{associative.triple} \rangle))$

$\langle \text{associative.triple} \rangle = (\langle \text{attribute} \rangle \langle \text{object} \rangle \langle \text{value} \rangle)$

$\langle \text{action} \rangle = \{ \langle \text{consequent} \rangle \} | \{ \langle \text{procedure} \rangle \}$

$\langle \text{consequent} \rangle = (\langle \text{associative.triple} \rangle \langle \text{certainty.factor} \rangle)$

MYCIN规则中，无论前项或后项，其基本部分是关联三元组（〈特性—对象—取值〉）或一个谓词加上三元组。

# 产生式系统的知识表示

具体的MYCIN规则以及它在机器内部用LISP 语言的表示。

MYCIN系统中有如下所示的一个典型规则：

前提条件：

细菌革氏染色阴性

形态杆状

生长需氧

结论：

该细菌是肠杆菌属，CF=0.8。

采用LISP表达式描述为：

PREMISE: (\$AND (SAME CNTXT GRAM  
GRAMNEG)

(SAME CNTXT MORPH ROD)

(SAME CNTXT AIR AEROBIC) )

ACTION: (CONCLUDE CNTXT CLASS  
ENTEROBACTERIACEAE  
TALLY 0.8)

# 产生式系统的知识表示

## 2. 规则的表达

### (2) 有关联规则间关系的表示

在知识库(规则库)中某些规则常按某种特征组织起来放在一起, 形成某种结构。这样既便于规则库的维护管理, 也方便于规则的使用。

#### a. 规则按参数分类

在MYCIN中每一项特性(临床参数)设有一种专门的特征表。

- IDENT: 〈属细菌属性PROP-ORG〉
- CONTAINED-IN: (RULE 030)
- EXPECT: (ONE OF (ORGANISMS) )
- LABDATA: T
- LOOKAHEAD: (RULE004, RULE054, ..., RULE168)
- PROMPT: (Enter the identity (genus) of \*)
- TRANS: (THE IDENTITY OF \*)
- UPDATED-BY: (RULE021, RULE003, ...,

# 产生式系统的知识表示

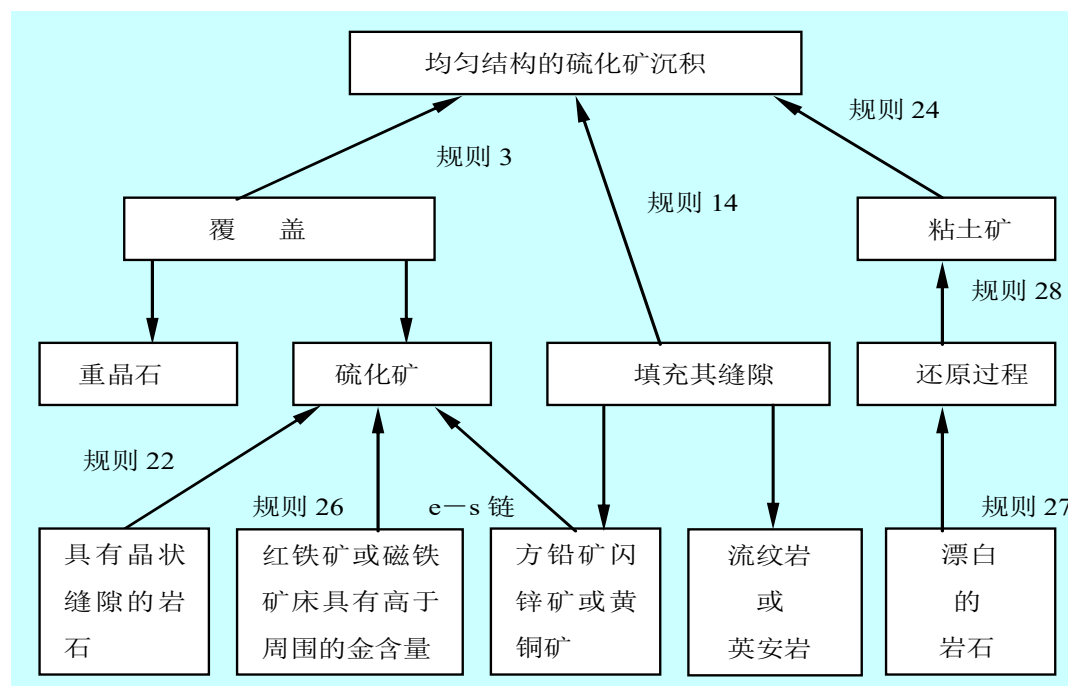
## 2. 规则的表示

### (2) 有关联规则间关系的表示

#### b. 规则的网状结构

规则之间可以以各种方式相互联系，当某一规则的结论正好是另一规则的前提或前提的一部分时，这两个规则就形成了一种“序关系”。如果用箭头表示这种序关系，在规则之间就形成了一种复杂的网状结构。

- 例10.2 PROSPECTOR系统中由不同规则所形成的部分推理网络图如图10-6所示，这是Kuroko-型均匀结构的硫化矿沉积的部分矿床模型。





# 产生式系统的知识表示

## 产生式与谓词逻辑中的蕴含式的区别

- (1) 除逻辑蕴含外，产生式还包括各种操作、规则、变换、算子、函数等。例如，“如果炉温超过上限，则立即关闭风门”是一个产生式，但不是蕴含式。
- (2) 蕴含式只能表示精确知识，而产生式不仅可以表示精确的知识，还可以表示不精确知识。蕴含式的匹配总要求是精确的。产生式匹配可以是精确的，也可以是不精确的，只要按某种算法求出的相似度落在预先指定的范围内就认为是可匹配的。

# 产生式系统的推理方式

## 1. 正向推理

正向推理是从已知事实出发，通过规则库求得结论，称为数据驱动方式，也称作自底向上的方式。推理过程是：

- (1) 规则集中规则的前件与数据库中的事实进行匹配，得到匹配的规则集合；
- (2) 从匹配规则集合中选择一条规则作为使用规则；
- (3) 执行使用规则，将该使用规则后件的执行结果送入数据库。
- (4) 重复这个过程直到达到目标。

# 产生式系统的推理方式

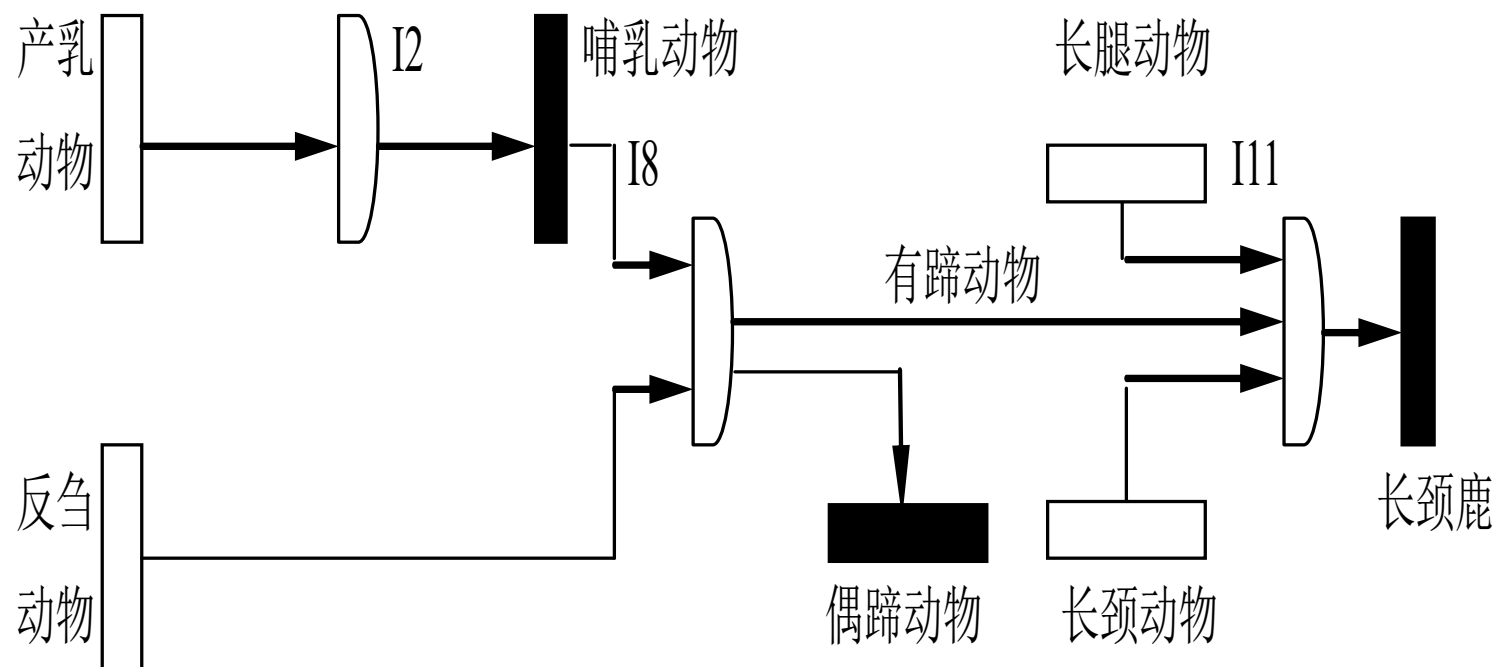
例10.3 动物识别系统IDENTIFIER包含有如下规则：

规则I2 如果 该动物能产乳，  
那么 它是哺乳动物。

规则I8 如果 该动物是哺乳动物，它反刍，  
那么 它是有蹄动物而且是偶蹄动物。

规则I11 如果 该动物是有蹄动物，它有长颈，它有长腿，它的颜色是黄褐色，  
它有深色斑点，  
那么 它是长颈鹿。

# 产生式系统的推理方式



# 产生式系统的推理方式

## 2. 逆向推理

逆向推理是从目标（作为假设）出发，逆向使用规则，找到已知事实。逆向推理过程如下：

- （1）规则集中的规则后件与假设的目标事实进行匹配，得到匹配的规则集合。
- （2）从匹配规则集合中选择一条规则作为使用规则。
- （3）将使用规则的前件作为新的假设子目标。
- （4）重复这个过程，直至各子目标均为已知事实后成功结束。

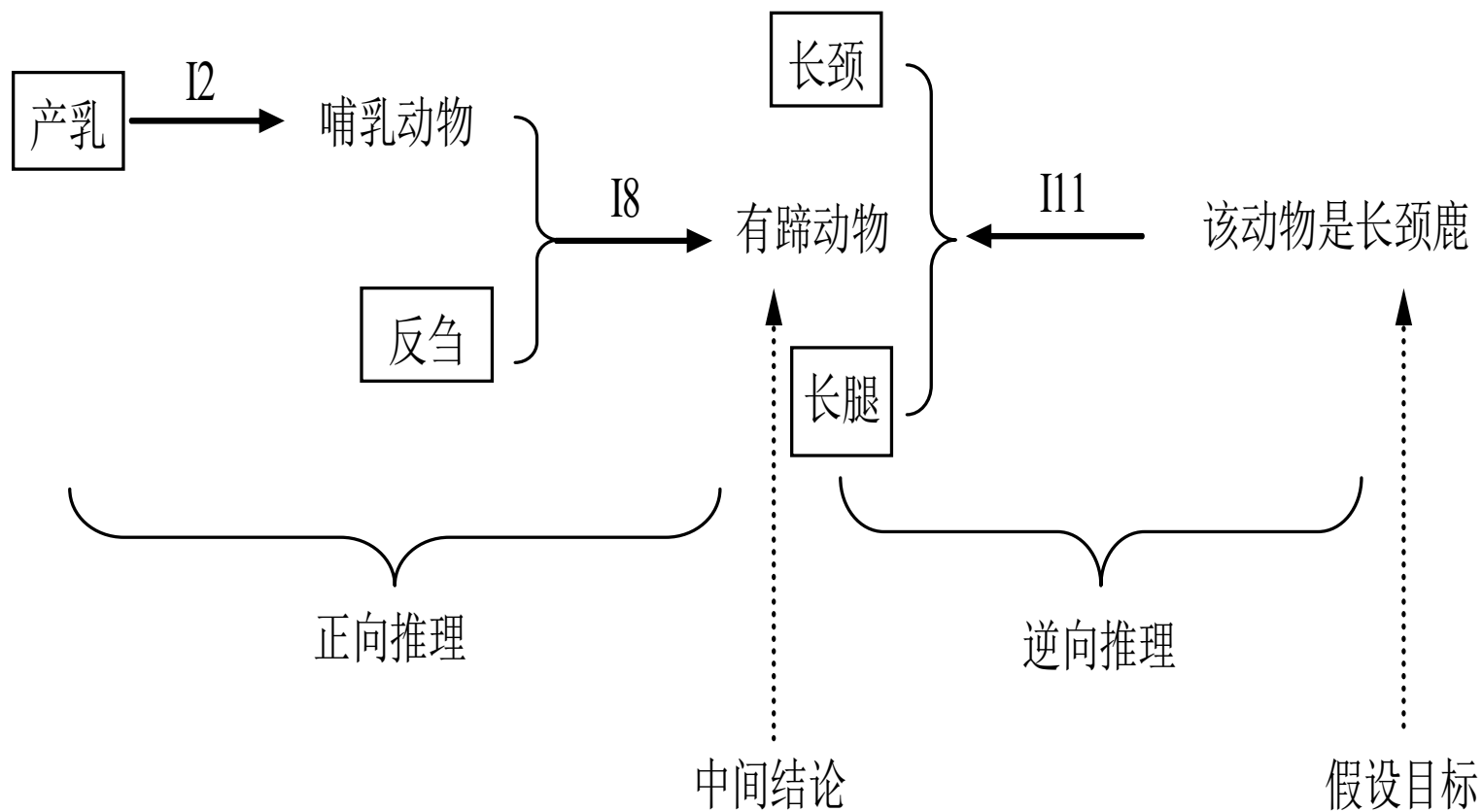
# 产生式系统的推理方式

## 3. 双向推理

双向推理，又叫混合推理，既自顶向下，又自底向上，从两个方向作推理，直至某个中间界面上两个方向的结果相符便成功结束。

例如，在动物识别系统中，已知某动物具有特征：长腿、长颈、反刍、产乳。为了识别一个动物，可以进行以下的双向推理。

# 产生式系统的推理方式



# 产生式规则的选择与匹配

---

通常搜索策略的主要任务是确定如何选取规则的方式和方法。

(1) 不考虑给定问题所具有的特定知识。

(2) 考虑问题领域可应用的知识。

选择规则的方法可以使用匹配。



# 产生式规则的选择与匹配

## 1. 规则的匹配

### (1) 用索引匹配

对全局数据库Gd(Global database)加索引，再通过映射函数找出相应的规则。

### (2) 变量匹配

例如，符号积分，使用规则： $\int u dv \rightarrow uv - \int v du$ ，而系统实际求积分时，要查找Gd中 $\int x dy$ 的形式，要求x与u，y与v匹配。

### (3) 近似匹配

在匹配中，有大部分条件符合或接近符合，则可认为规则匹配。

# 产生式规则的选择与匹配

## 2. 规则的选择

**(1) 专用与通用性排序**，如果某一规则的条件部分比另一规则的条件部分所规定的情况更为专门化，则更为专门化的规则优先使用。

例10.4 美式足球训练规则：

pr1: If 是第四次开始进攻（Fourth down）且进攻方前三次进攻中前进的距离少于10码  
then 可以在第四次进攻时踢悬空球

pr2: If 是第四次开始进攻（Fourth down）且进攻方前三次进攻中前进的距离少于10码）  
且进攻位置在对方球门线30码之内  
then 可以射门（field goal）

pr2比pr1更专门化，因为pr2条件限制更多一些。

# 产生式规则的选择与匹配

通常的判定方法是：

- 如果某一规则的前件集包含另一规则的所有前件，则前一规则较后一规则更为专门化。
- 如果某一规则中的变量处在第二规则中是常量，而其余相同，则后一规则比前一规则更专门化。

# 产生式规则的选择与匹配

## 2. 规则的选取

**(2) 规则排序**，通过对问题领域的了解，规则集本身就可划分优先次序。那些最适用的或使用频率最高的规则优先使用。例如，模仿心理疗法的行为的智能程序ELIZA，有规则： pr1: ...; pr2: ...; ... prn: ...;

其中，最后一个规则为：

prn: if nil then “tell me more about your family”

因为最后一个规则的前件是空的，无条件地可以使用，所以要安排在没有其他规则可使用时才可使用。

# 产生式规则的选择与匹配

## 2. 规则的选取

- (3) **数据排序**，将规则中的条件部分按某个优先次序排序。
- (4) **规模排序**，按条件部分的多少排序，条件多者优先。
- (5) **就近排序**，最近使用的规则排在优先位置，这样使用多的规则优先。
- (6) **按上下文限制将规则分组**，如在医学专家系统MYCIN中，不同上下文用不同组的规则进行诊断或开处方。

## 产生式系统的例子——动物识别系统

- 例如：动物识别系统——识别虎、金钱豹、斑马、长颈鹿、鸵鸟、企鹅、信天翁等七种动物的产生式系统。



## 产生式系统的例子——动物识别系统

- 规则库：
  - $r_1$ : IF 该动物有毛发 THEN 该动物是哺乳动物
  - $r_2$ : IF 该动物有奶 THEN 该动物是哺乳动物
  - $r_3$ : IF 该动物有羽毛 THEN 该动物是鸟
  - $r_4$ : IF 该动物会飞 AND 会下蛋 THEN 该动物是鸟
  - $r_5$ : IF 该动物吃肉 THEN 该动物是食肉动物
  - $r_6$ : IF 该动物有犬齿 AND 有爪 AND 眼盯前方  
THEN 该动物是食肉动物
  - $r_7$ : IF 该动物是哺乳动物 AND 有蹄  
THEN 该动物是有蹄类动物
  - $r_8$ : IF 该动物是哺乳动物 AND 是反刍动物  
THEN 该动物是有蹄类动物

## 产生式系统的例子——动物识别系统

- $r_9$ : IF 该动物是哺乳动物 AND 是食肉动物 AND 是黄褐色  
AND 身上有暗斑点 THEN 该动物是金钱豹
- $r_{10}$ : IF 该动物是哺乳动物 AND 是食肉动物 AND 是黄褐色  
AND 身上有黑色条纹 THEN 该动物是虎
- $r_{11}$ : IF 该动物是有蹄类动物 AND 有长脖子 AND 有长腿  
AND 身上有暗斑点 THEN 该动物是长颈鹿
- $r_{12}$ : IF 该动物有蹄类动物 AND 身上有黑色条纹  
THEN 该动物是斑马
- $r_{13}$ : IF 该动物是鸟 AND 有长脖子 AND 有长腿 AND 不会飞  
AND 有黑白二色 THEN 该动物是鸵鸟
- $r_{14}$ : IF 该动物是鸟 AND 会游泳 AND 不会飞  
AND 有黑白二色 THEN 该动物是企鹅
- $r_{15}$ : IF 该动物是鸟 AND 善飞 THEN 该动物是信天翁



# 产生式系统的例子——动物识别系统

- 设已知初始事实存放在综合数据库中：

该动物身上有：暗斑点，长脖子，长腿，奶，蹄

- 推理机构的工作过程：

(1) 从规则库中取出 $r_1$ ，检查其前提是否可与综合数据库中的已知事实匹配。匹配失败则 $r_1$ 不能被用于推理。然后取 $r_2$ 进行同样的工作。匹配成功则 $r_2$ 被执行。

- 综合数据库：

该动物身上有：暗斑点，长脖子，长腿，奶，蹄，哺乳动物

## 产生式系统的例子——动物识别系统

### ■ 推理机构的工作过程：

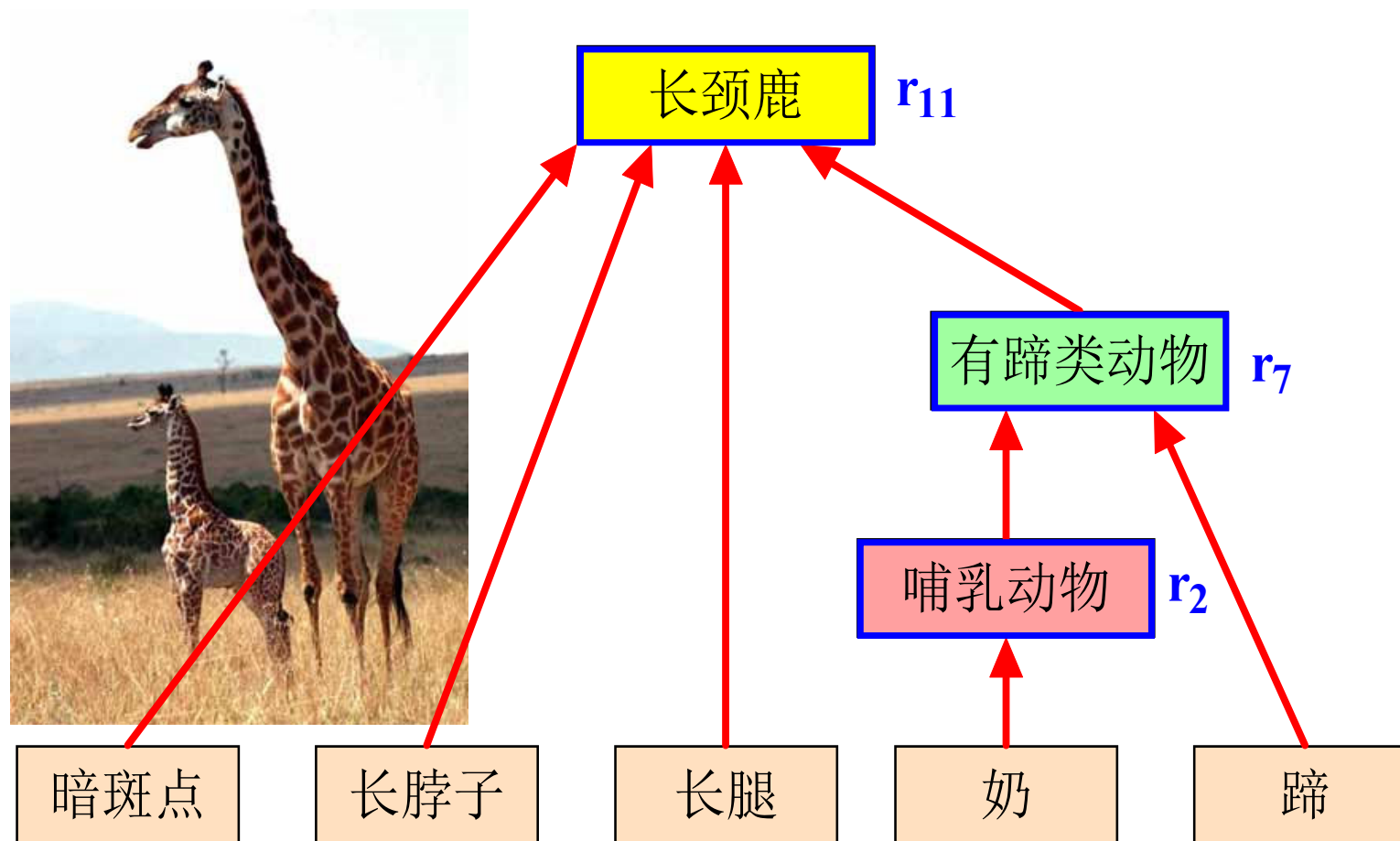
(2) 分别用 $r_3$ ,  $r_4$ ,  $r_5$ ,  $r_6$ 综合数据库中的已知事实进行匹配, 均不成功。 $r_7$ 匹配成功, 执行 $r_7$ 。

### ■ 综合数据库：

该动物身上有：暗斑点，长脖子，长腿，奶，蹄，哺乳动物，有蹄类动物

(3)  $r_{11}$ 匹配成功，并推出“该动物是长颈鹿”。

## 产生式系统的例子——动物识别系统



# 产生式表示法的特点

## 1. 产生式表示法的优点

- (1) 自然性
- (2) 模块性
- (3) 有效性
- (4) 清晰性

## 2. 产生式表示法的缺点

- (1) 效率不高
- (2) 不能表达结构性知识

## 3. 适合产生式表示的知识

- (1) 领域知识间关系不密切，不存在结构关系。
- (2) 经验性及不确定性的知识，且相关领域中对这些知识没有严格、统一的理论。
- (3) 领域问题的求解过程可被表示为一系列相对独立的操作，且每个操作可被表示为一条或多条产生式规则。

## 10.4 语义网络表示法

- 1968年Quillian的博士论文建议用一种语义网络来描述人对事物的认知，实际上是对人脑功能的模拟。
- 逻辑和产生式表示方法常用于表示有关领域中各个不同状态间的关系。然而用于表示一个事物同其各个部分间的分类知识就不方便了。
- 槽和填槽表示方法便于表示这种分类知识。这种表示方法包括语义网络、框架、概念从属和脚本。语义网络方法的特点就在于提出了槽和填槽的结构。
- 语义网络同一阶逻辑有相同的能力。多用于自然语言处理。

# 语义网络表示法

语义网络表示由下列4个相关部分组成：

- (1) **词法部分** 决定该表示方法词汇表中允许有哪些符号，它涉及各个结点和弧线。
- (2) **结构部分** 叙述符号排列的约束条件，指定各弧线连接的结点对。
- (3) **过程部分** 说明访问过程，这些过程能用来建立和修正概念的描述，以及回答相关问题。
- (4) **语义部分** 确定与描述相关的意义的方法，即确定有关结点和对应弧线的排列及其相互关系。

# 语义网络结构

三元组 (结点1, 弧, 结点2)

- 结点表示概念、事物、事件、情况等；

类结点（概念结点）：汽车

实例结点：我的汽车



- 有向弧体现结点之间的语义联系；

弧的方向体现主次关系，结点1为主，结点2为辅；

弧上的标注表示结点1的属性或结点1和结点2之间的关系。

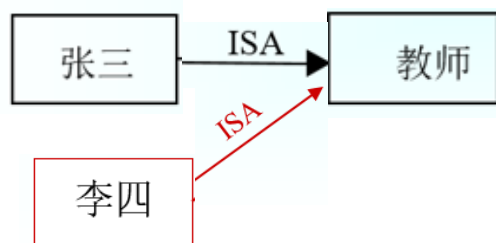
# 典型的语义联系

## 1.以个体为中心组织知识的语义联系

结点一般是名词性个体或概念

**(1) 实例联系：**用于表示类结点和实例结点之间的联系。通常用ISA来标识。

表示“张三是教师”这一事实，可建立两个结点：张三 教师。两节点以ISA（表示“是一个”）链相连，如下图所示：



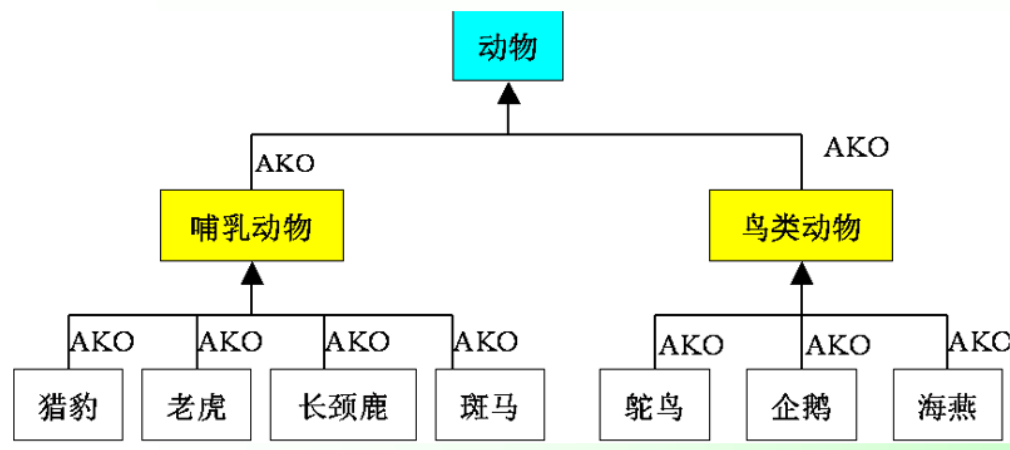


# 典型的语义联系

## 1.以个体为中心组织知识的语义联系

结点一般是名词性个体或概念

(2) **泛化联系**：用于表示类结点和抽象层次更高的类结点之间的联系。通常用AKO (A Kind Of) 标识。



# 典型的语义联系

## 1.以个体为中心组织知识的语义联系

结点一般是名词性个体或概念

(3) **聚集联系**：用于表示与其组成成分之间的联系。通常用Has-part表示。

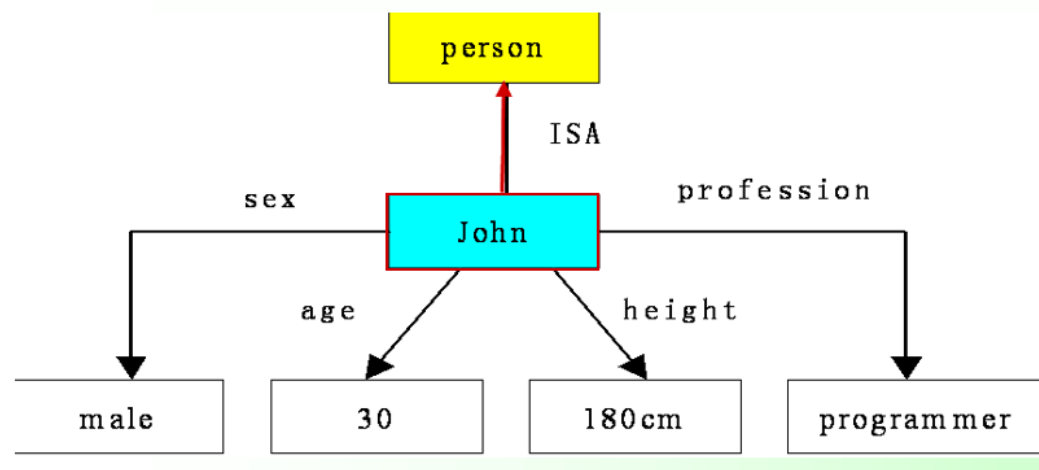


# 典型的语义联系

## 1.以个体为中心组织知识的语义联系

结点一般是名词性个体或概念

(4) **属性联系**：用于表示个体，属性及其取值之间的联系。通常有向弧表示属性，用弧所指向的结点表示属性的值。



# 典型的语义联系

## 1.以个体为中心组织知识的语义联系

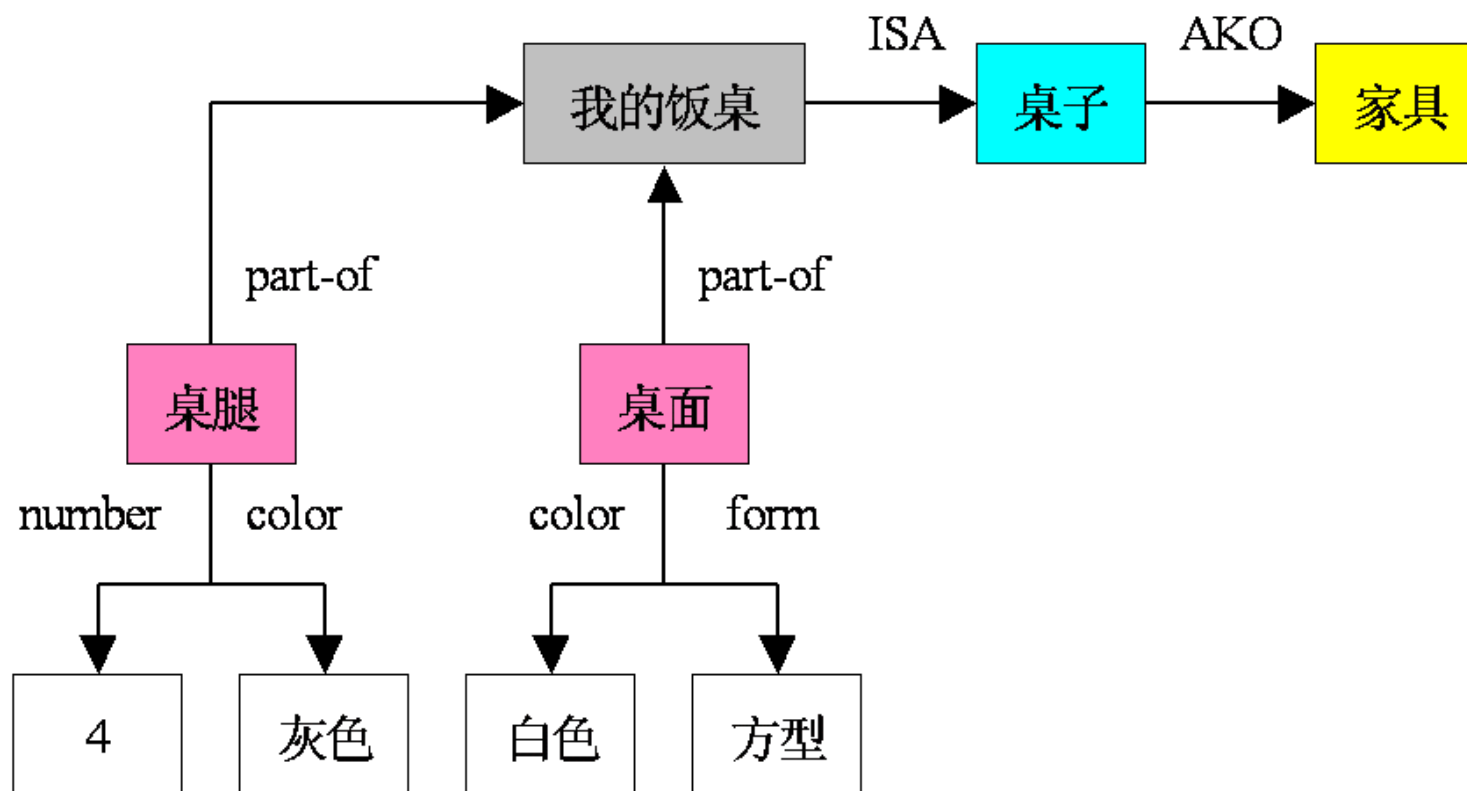
如果增添新的事实，只需在语义网络中增加新的结点和弧线就可以了。如果在上图所示的语义网络中，要增添事实：

“CLYDE是一个知更鸟，并且有一个叫做NEST-1的巢。”



# 典型的语义联系

## 1.以个体为中心组织知识的语义联系



# 典型的语义联系

## 2.以谓词或关系为中心

关系结点可以分为类结点和实例结点两种。实例关系结点和类关系结点之间用ISA标识。

多元关系 $R(x1, x2, \dots, xn)$ 可以转成

$$R(x11, x12) \wedge R(x21, x22) \wedge \dots \wedge R(xn1, xn2,).$$

例如，TRIANGLE (a, b, c) 表示一个三角形由三条边a ,b ,c构成，可表述成：

$$\text{cat}(a, b) \wedge \text{cat}(b, c) \wedge \text{cat}(c, a)$$

# 典型的语义联系

## 2. 以谓词或关系为中心

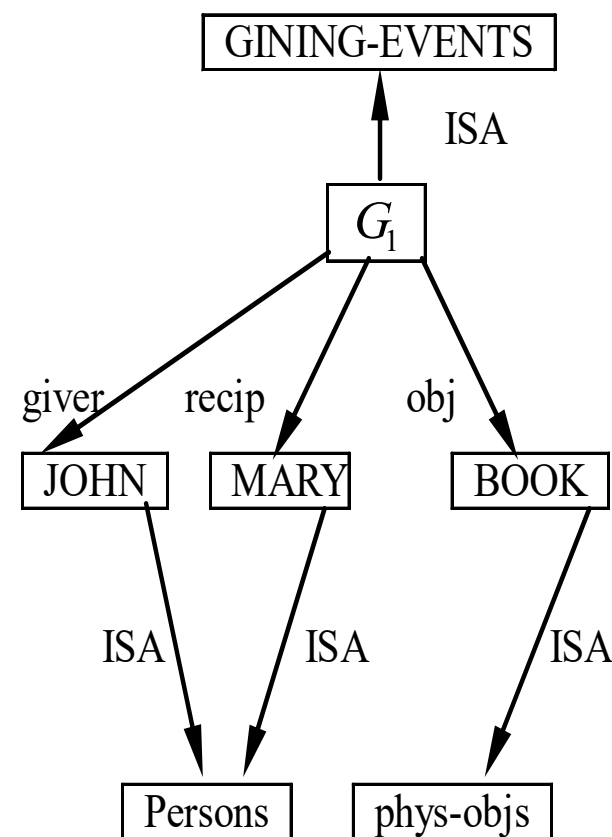
“John gave Mary a book.”这一事实，用谓词可表示为：

GIVE(JOHN, MARY, BOOK)

这是一个多元关系，用语义网络表示如图所示。

其中G1是增加的一个节点，用来表示一个特定

**GIVING-EVENTS事件。**



# 连接词和量词的表示

从逻辑上看，一个基本的语义网络相当于一组二元谓词。

三元组 (结点1, 弧, 结点2)

P (个体1, 个体2)。

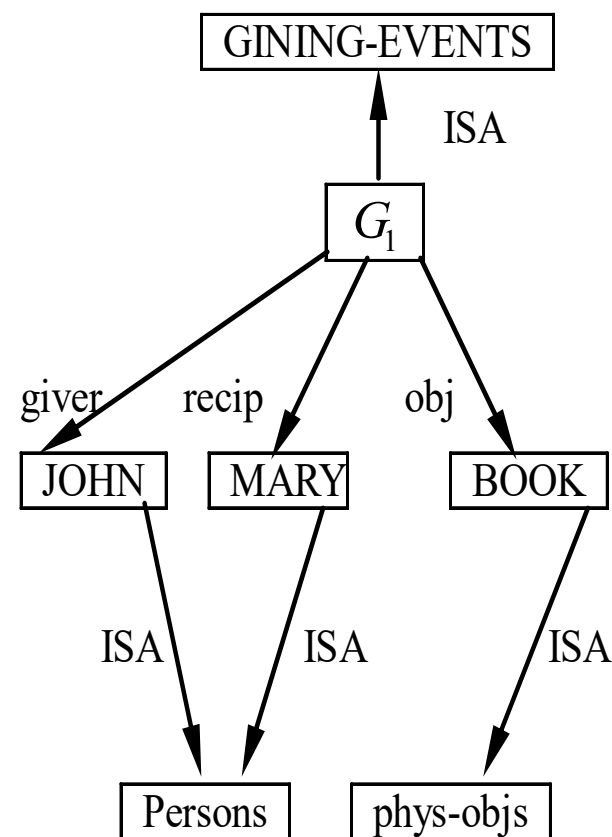
语义网络可以作为一种“粒度”较大的、表达信息更丰富知识单元。在这种知识单元中也存在与谓词逻辑中类似的各种连接词及量词。



# 连接词和量词的表示

## 1. 合取

在上例中与节点 $G_1$ 相连的链GIVER, OBJ以RECIP之间是合取关系。因为只有给予者是JOHN, 接受者是MARY, 给予物是BOOK, 这三个关系同时成立时, 才构成事件 $G_1$ 。在语义网络的表示中约定: 如果不另外增加标志, 就意味着弧与弧之间的关系就是合取。



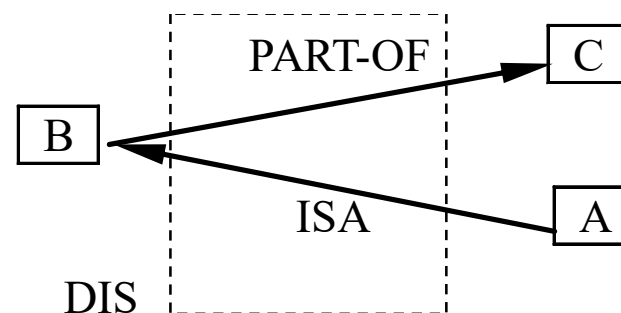
# 连接词和量词的表示

## 2. 析取

在语义网络中，为了表示“或”的关系，一种最常用的方法是将“或”关系的弧用一条封闭虚线包围起来，并标记**DIS**。

下图表示的就是具有“或”关系的语义网络，其含义用谓词公式表示出来就是：

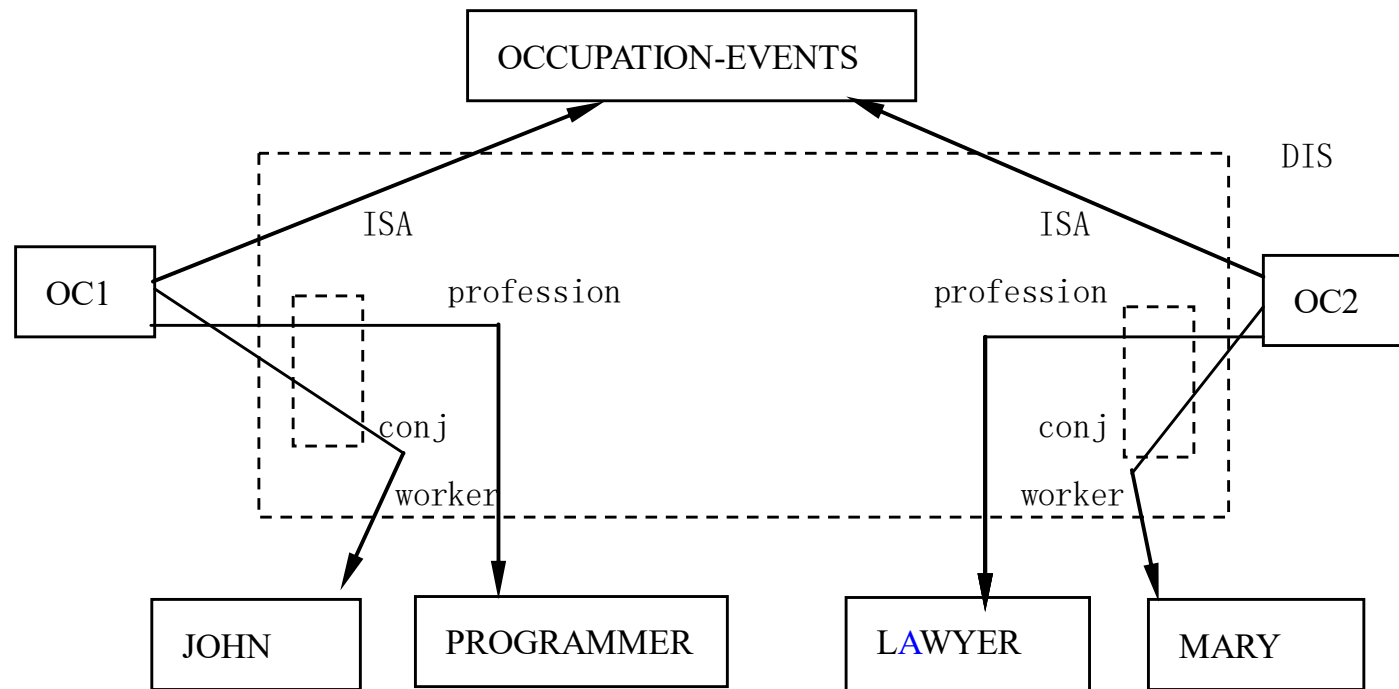
$ISA(A,B) \vee PART-OF(B,C)$



# 连接词和量词的表示

如果“与”关系是嵌套在“或”关系内的，则这些具有“与”关系的弧用标记为 CONJ 的封闭虚线包围起来。

JOHN是一个程序员或  
者MARY是一个律师



# 连接词和量词的表示

## 3. 否定

可以采用 $\sim$ ISA和 $\sim$ PART-OF 关系或标注出NEG界限



其中,  $\boxed{A} \xrightarrow{\sim ISA} \boxed{B}$  表示  $\sim(A \text{ ISA } B)$  ,

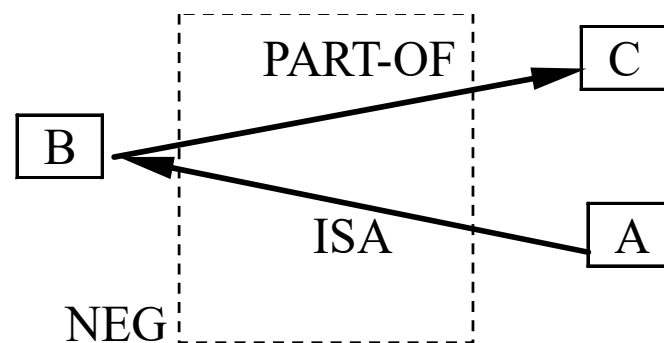
$\boxed{B} \xrightarrow{\sim PART-OF} \boxed{C}$  表示  $\sim(B \text{ PART-OF } C)$

# 连接词和量词的表示

## 3. 否定

如果我们要用语义网络表示  $\sim[IS\ A(A, B) \wedge PART-OF\ (B, C)]$

可利用 $\sim ISA$ 和 $\sim PART-OF$ 来表示这个事实。如果不希望改变这个表达式的形式，那么可以利用NEG界限，如下图所示。

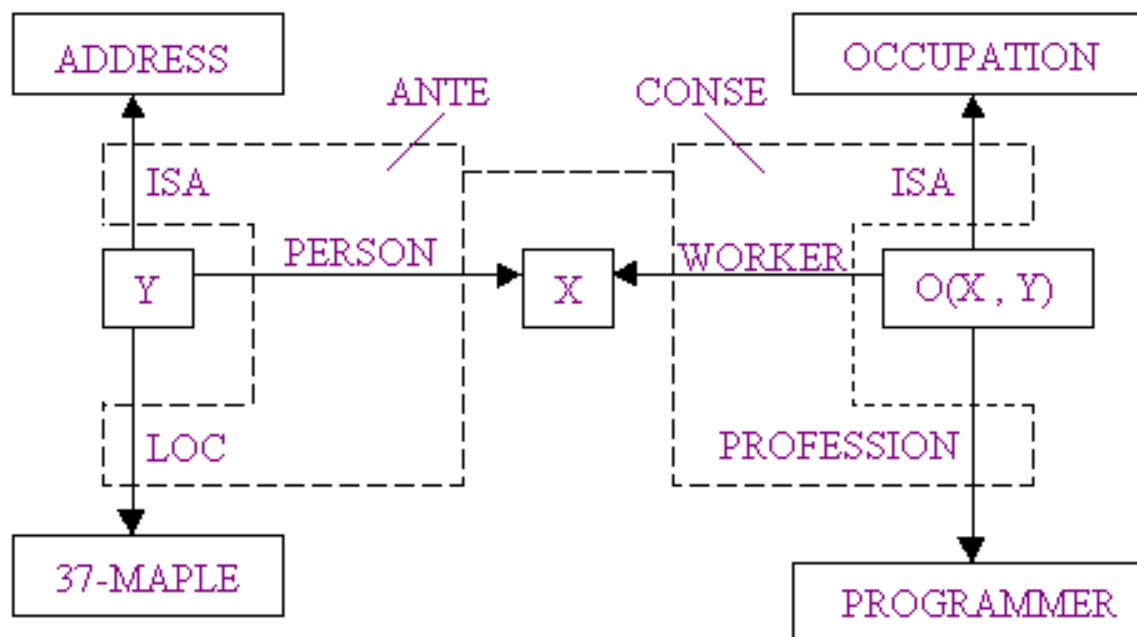


# 连接词和量词的表示

## 4. 蕴含

在语义网络中可用标注ANTE和CONSE界限来表示蕴涵关系。

“Every one who lives  
at 37 Maple Street is a  
programmer.”。

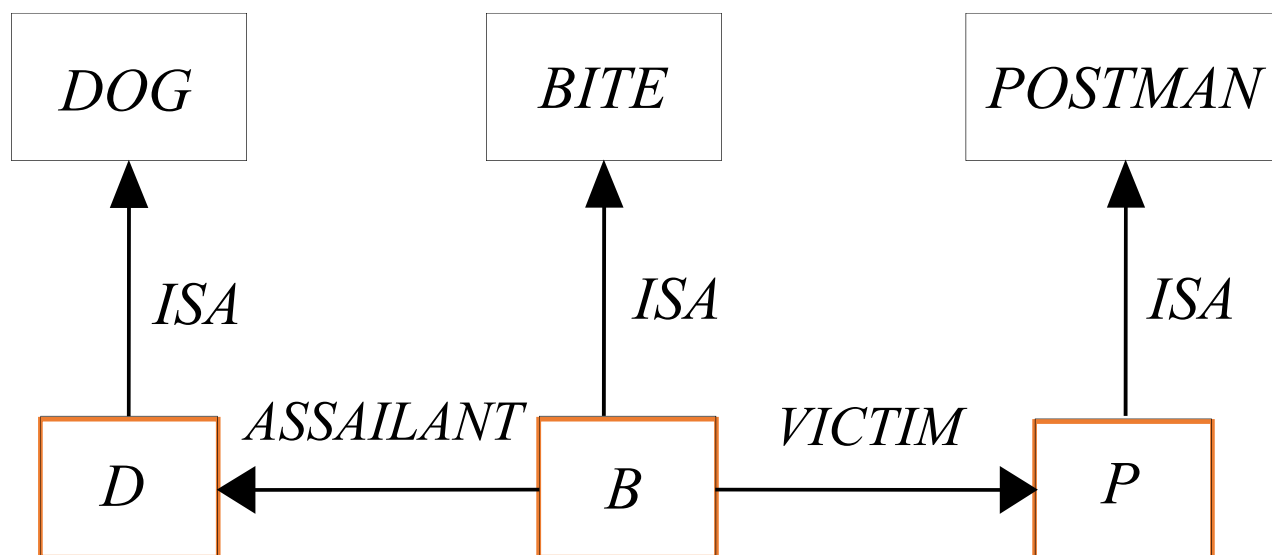


# 连接词和量词的表示

## 5. 量化

(1) 存在量词的量化: 存在量词在语义网络中可直接用ISA链来表示。

The dog bites the postman. 这句话意味着所涉及的是存在量化。



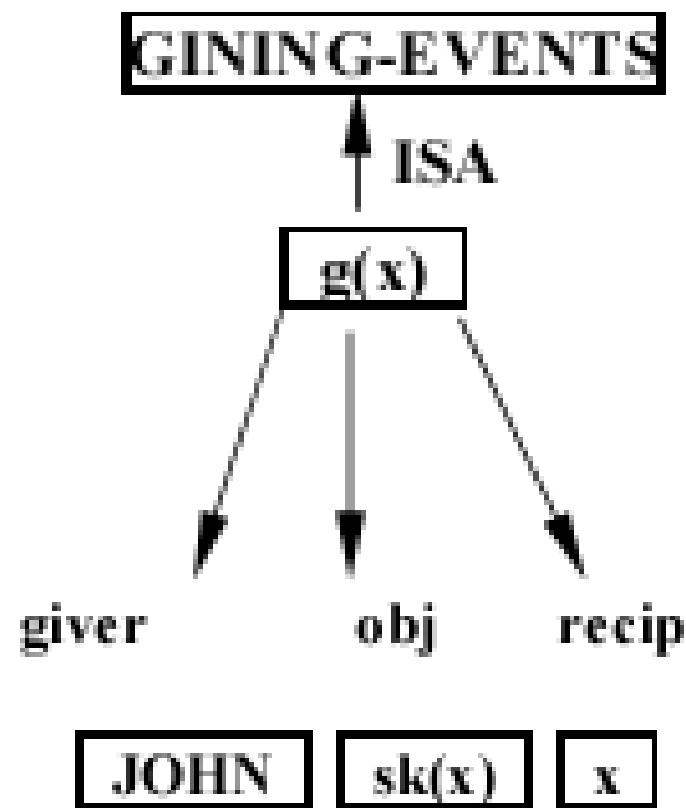
# 连接词和量词的表示

## 5. 量化

### (2) 全称量词的量化

全称量词的在语义网中的量化，其量词的辖域可以是整个语义网络，也可以是把语义网络分割后的某一个范围。

“JOHN给了所有人一件东西”





# 连接词和量词的表示

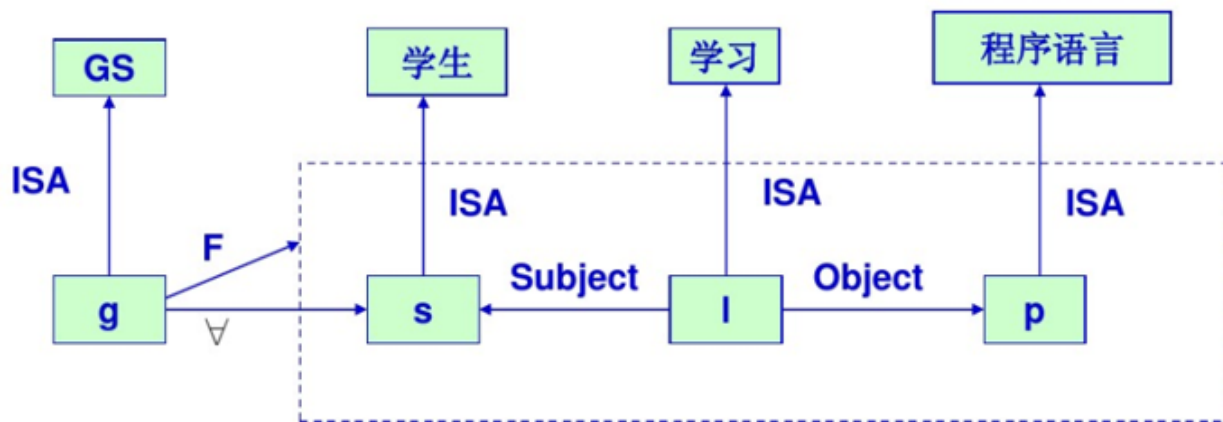
## 5. 量化

### (2) 全称量词的量化

把语义网络分割成若干空间分层集合，每一个空间对应于一个或几个变量辖域范围。

例：用语义网络表示如下事实：“每个学生都学习了一门程序设计语言”

GS是一个概念结点，它表示具有全称量化的一般事件。



S是一个全称变量，表示任意一个学生。

l是一个存在变量。表示某一次学习。

P是一个存在变量，表示某一门程序设计语言。

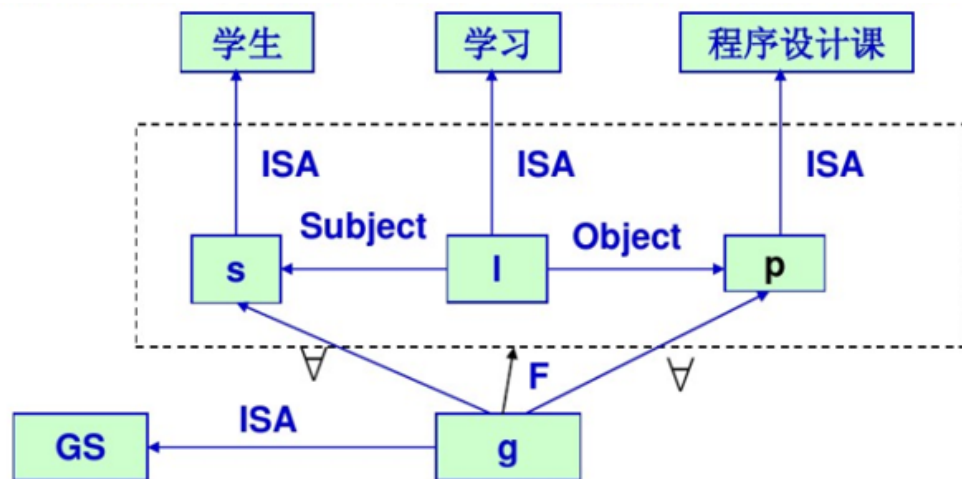
# 连接词和量词的表示

## 5. 量化

### (2) 全称量词的量化

把语义网络分割成若干空间分层集合，每一个空间对应于一个或几个变量辖域范围。

例：用语义网络表示如下事实：“每个学生都学习了所有的程序设计课程”



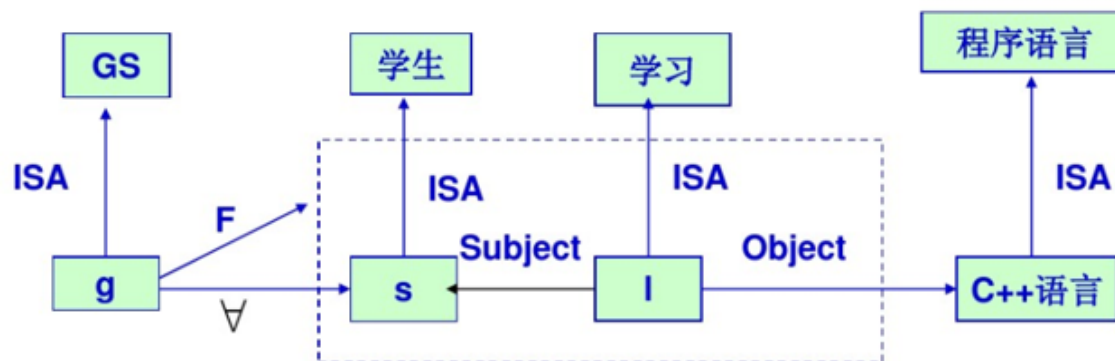
# 连接词和量词的表示

## 5. 量化

### (2) 全称量词的量化

另外，在网络分区技术中，要求F指向的子空间中的所有非全称变量结点都应该是存在约束的变量，否则应放在子空间的外边。

“每个学生都学习了C++语言”



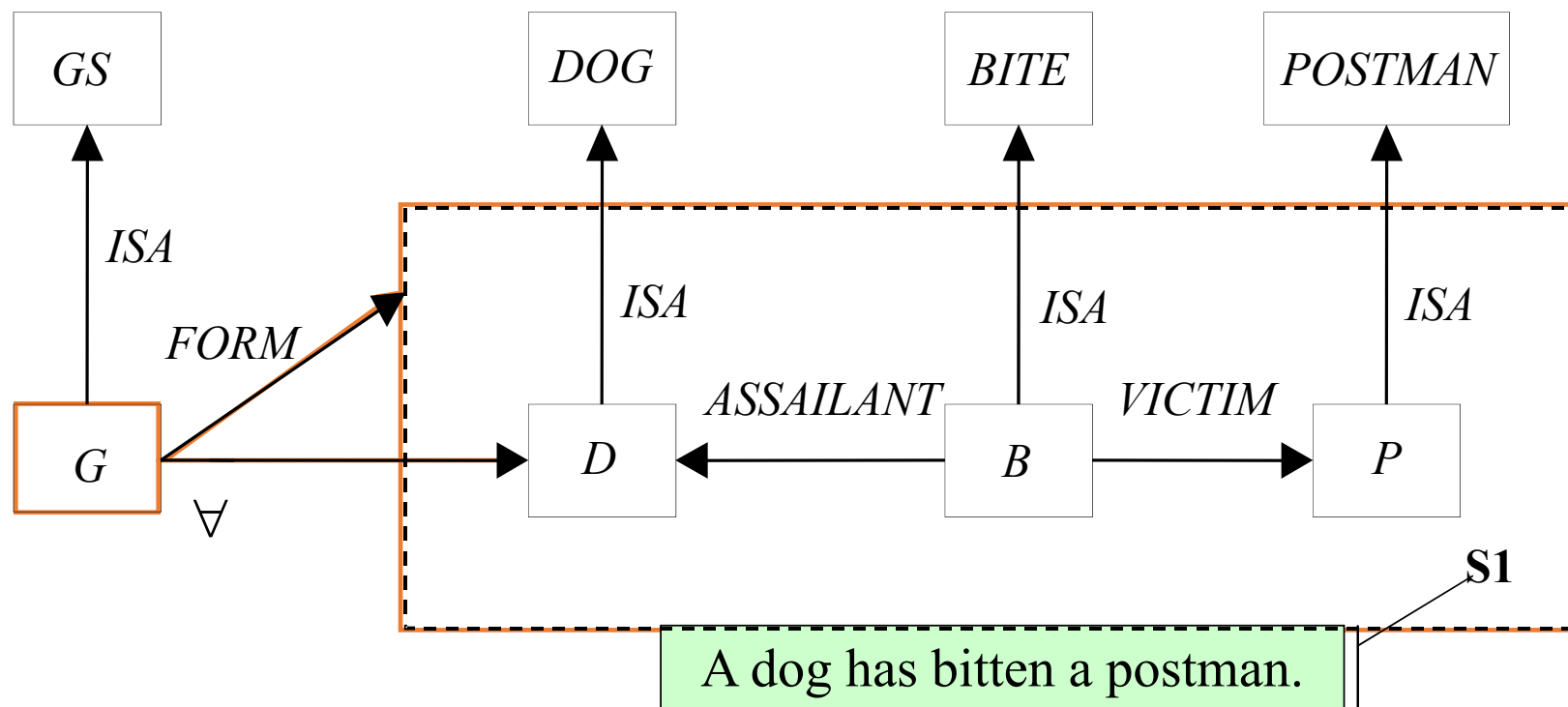
# 连接词和量词的表示

## 5. 量化

Every dog has bitten a postman.

### (2) 全称量词的量化

$$(\forall x) \text{DOG}(x) \rightarrow (\exists y)[\text{POSTMAN}(y) \wedge \text{BITE}(x, y)]$$

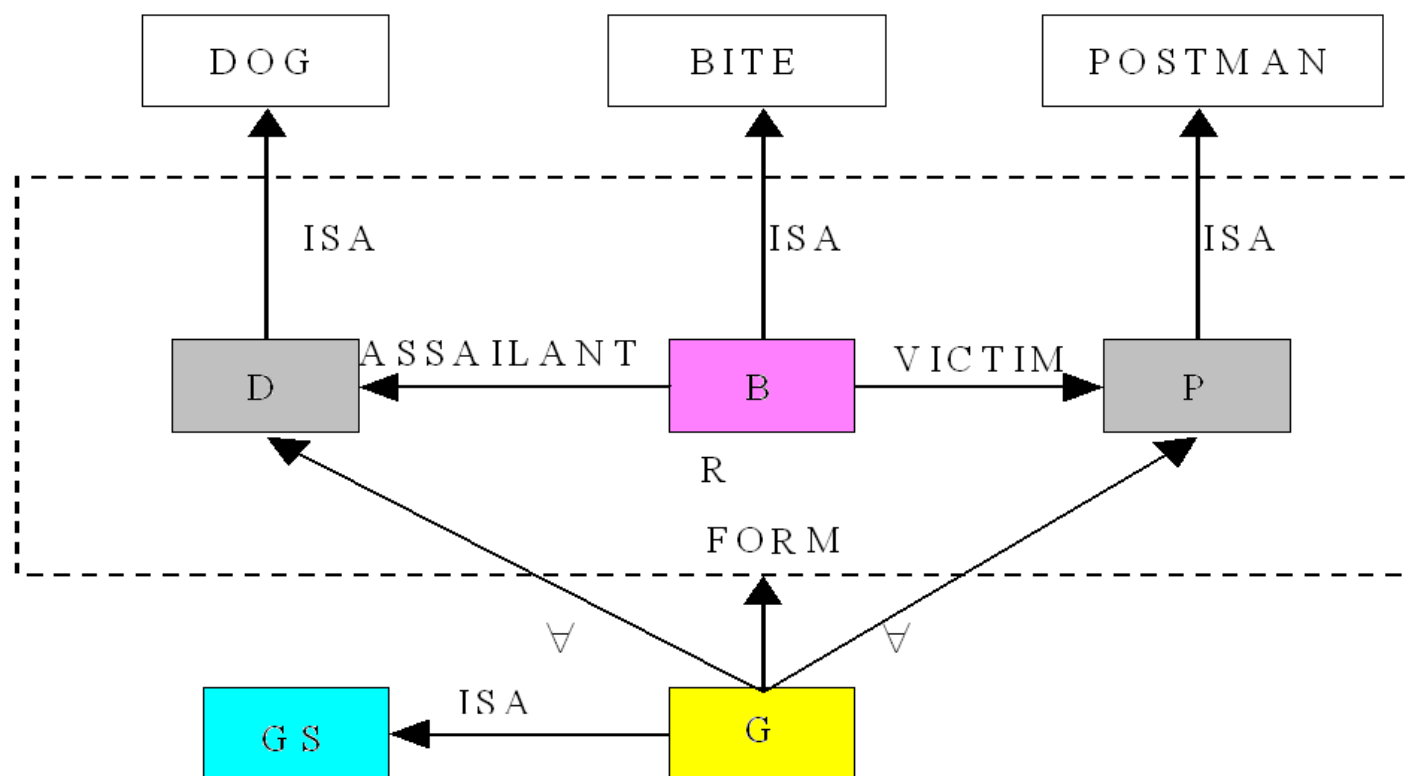


# 连接词和量词的表示

## 5. 量化

Every dog has bitten every postman.  $\forall x \forall y \text{ dog}(x) \wedge \text{postman}(y) \rightarrow \text{bit}(x, y)$

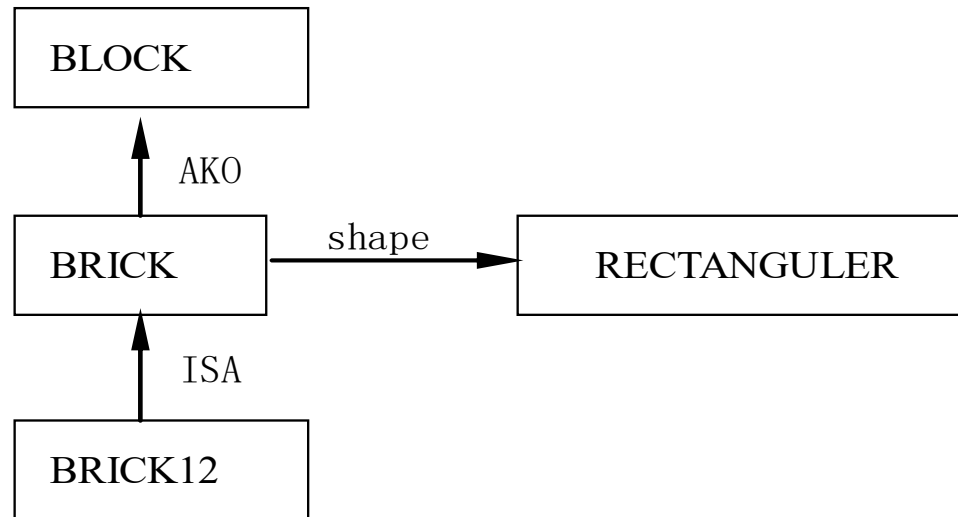
### (2) 全称量词的量化



# 语义网络的推理过程

## 1. 继承

把对事物的描述从概念结点或类结点传递到实例结点中去。



虽然BRICK12没有SHAPE槽，但可以从这个语义网络推理出BRICK12的外形是矩形的。

# 语义网络的推理过程

## 1. 继承

(1) 值继承：最简单的值继承是ISA关系下的直接继承。



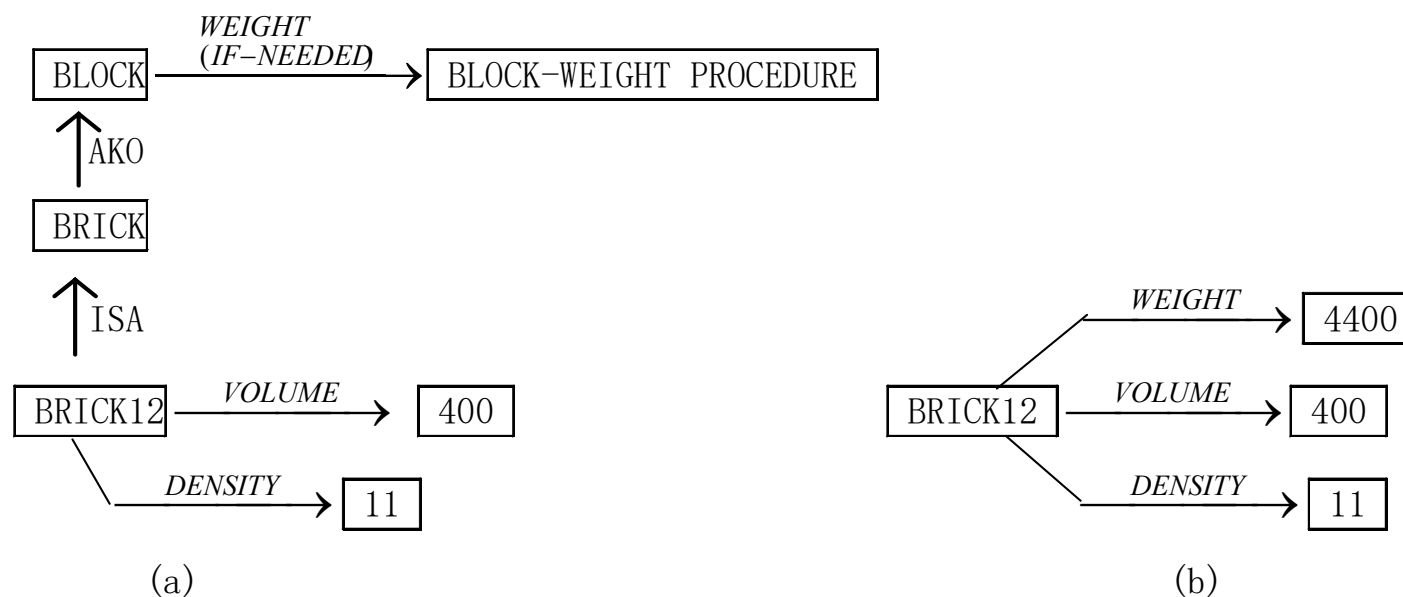
另外，还有一种AKO(A-KIND-OF)弧也用于语义网络中的描述特性的继承。

ISA和AKO弧都可以直接地表示类的成员关系以及子类和类之间的关系，提供了一种把知识从某一层传递到另一层的途径。

# 语义网络的推理过程

## 1. 继承

(2) “如果需要” 继承。在某些情况下，当我们不知道槽值但又需要这个槽值时，可以利用if-needed继承，进行这种计算的程序称为if-needed程序。这种槽有if-needed侧面，if-needed程序就有放在这侧面中。

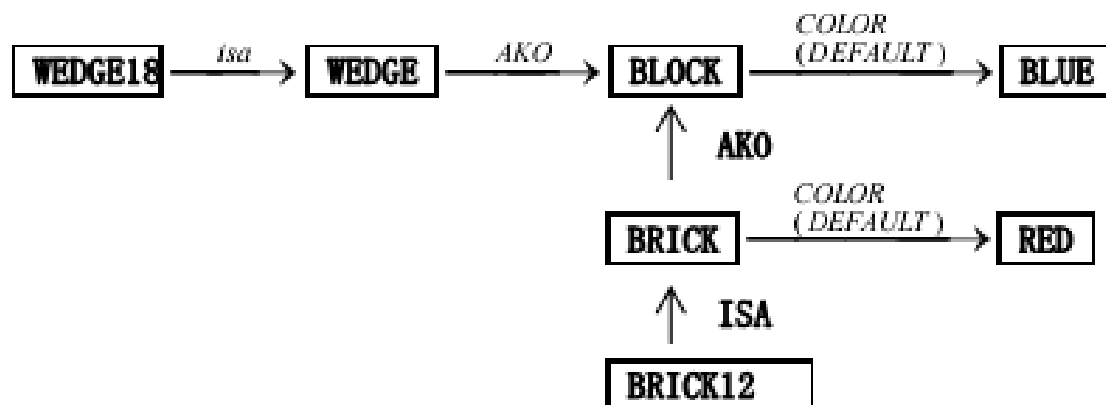




# 语义网络的推理过程

## 1. 继承

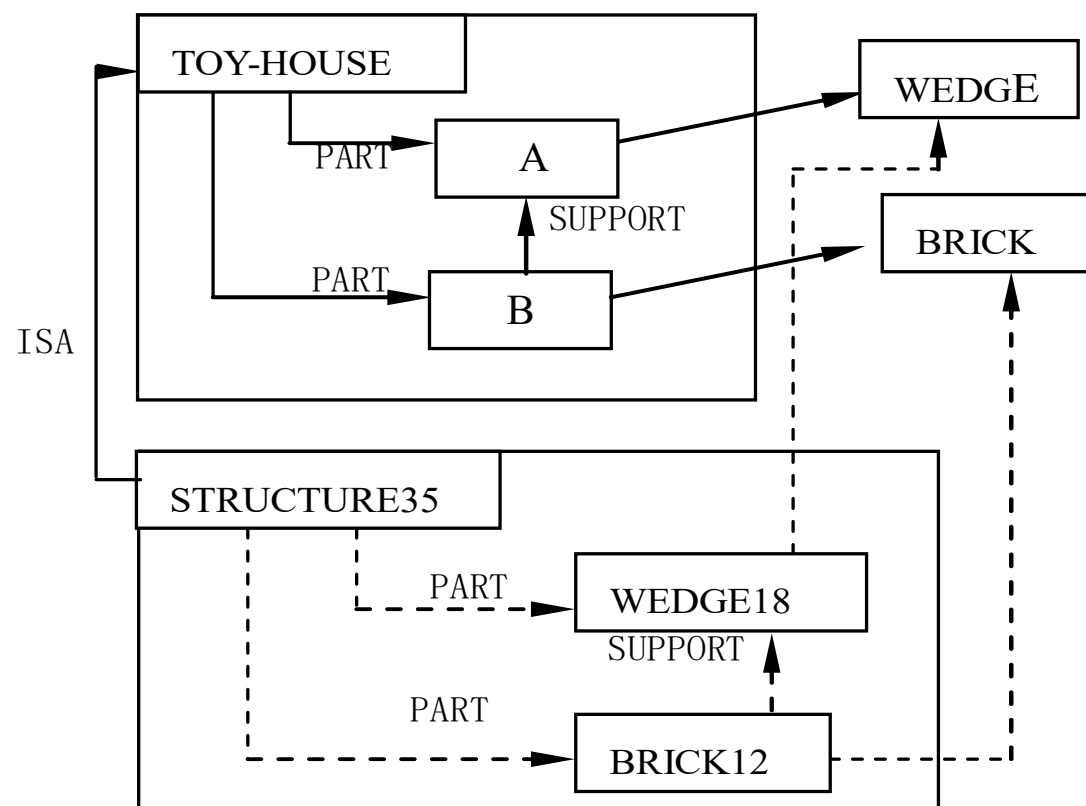
(3) **默认继承**。某个弧值具有相当程度的真实性，但又不能十分肯定的，因此设定为默认值，放在这个结点中，并标明这个弧为DEFAULT（默认）弧。只要不与现有事实相冲突，就默认这个值为这个结点的值。



# 语义网络的推理过程

## 2. 匹配

语义网络中推理方法主要是依靠匹配。进行匹配时，根据提出的问题可构成局部网络，这个网络中有的结点或弧的标记是空的，表示有待求解的。依据这个局部网络到知识库中寻找匹配的网络，以便求得问题的解答。



# 语义网络的推理过程

## 3. 语义网络上的推理

带蕴涵结点的语义网络又称为推理网络。

语义网络的演绎就是在推理网络上的搜索过程。此时的语义网络实际上演化为基于网络的规则系统。

推理网络上的搜索也有正向推理、逆向推理和双向推理。

# 语义网络的特点

- (1) 语义网络能把实体的结构、属性与实体间的因果关系显示地和简明地表达出来，这样可以联想方式实现对系统的解释。
- (2) 由于与概念相关的属性和联系被组织在一个相应的结点中，因而语义网络使概念易于访问和学习。
- (3) 语义网络表现问题更加直观，更易于理解，适合于知识工程师与领域专家沟通。

# 语义网络的特点

---

- (4) 语义网络结构的语义解释依赖于该结构的推理过程而没有固定结构的约定，因而得到的推理不能保证像谓词逻辑法那样有效。
- (5) 语义网络结点间的联系可能是线状、树状或网状的，甚至是递归状的结构，使相应的知识存储和检索需要比较复杂的过程。

## 10.5 框架表示法

### 框架理论

- 1975年Minsky在他的论文“A Framework for representing knowledge”中提出了框架理论，后来被逐步成为一种被广泛使用的知识表示方法。
- 框架理论提出是基于这样的心理学研究成果：在人类日常的思维及理解活动中已存储了大量的典型情景，当分析和理解所遇到的新情况时，人们从记忆中选择（即匹配）某个轮廓的基本知识结构（即框架）与当前的现实情况进行某种程度的匹配。

# 框架理论

- **框架：**是人们认识事物的一种通用的数据结构形式。即当新情况发生时，人们只要把新的数据加入到该通用数据结构中便可形成一个具体的实体（类），这样的通用数据结构就称为框架。
- **实例框架：**对于一个框架，当人们把观察或认识到的具体细节填入后，就得到了该框架的一个具体实例，框架的这种具体实例被称为实例框架。
- **框架系统：**在框架理论中，框架是知识的基本单位，把一组有关的框架联结起来便可形成一个框架系统。
- **框架系统推理：**由框架之间的协调完成。

# 框架结构

**框架**是基于概念的抽象程度表现出自上而下的分层结构，它的最顶层是固定的一类事物。框架由**框架名**和描述事物各个方面的**槽**组成。每个槽有可以拥有若干**侧面**，而每个侧面可以拥有若干个值。这些内容可以根据具体问题的具体需要来取舍。

一个框架的一般结构如下：

```
<框架名>  
  <槽1> <侧面11> <值111>...  
          <侧面12> <值121>...  
  <槽2> <侧面21> <值211>...  
  ...  
  <槽n> <侧面n1> <值n11>...
```



# 框架结构

例如，一个人可以用其职业、身高和体重等项信息来描述，因而可以用这些项目组成框架的槽。当描述一个具体的人时，再用这些项目的具体值填入到相应的槽中。下面给出的是描述JOHN这个人的一个框架。

**JOHN**

**isa : PERSON**

**Profession : PROGRAMMER**

**Height : 1.8m**

**Weight : 79kg**

# 框架结构

下面是一个描述“教师”的框架。

框架名：〈教师〉  
姓名：单位（姓、名）  
年龄：单位（岁）  
性别：范围（男、女）  
缺省：男  
职称：范围（教授，副教授，讲师，助教）  
缺省：讲师  
部门：单位（系，教研室）  
住址：〈住址框架〉  
工资：〈工资框架〉  
开始工作时间：单位（年、月）  
截止时间：单位（年、月）  
缺省：现在

下面是描述一个具体教师的框架。

框架名：〈教师-1〉  
姓名：夏冰  
年龄：36  
性别：女  
职称：副教授  
部门：计算机软件教研室  
住址：〈adr-1〉  
工资：〈sal-1〉  
开始工作时间：1988, 9  
截止时间：1996, 7

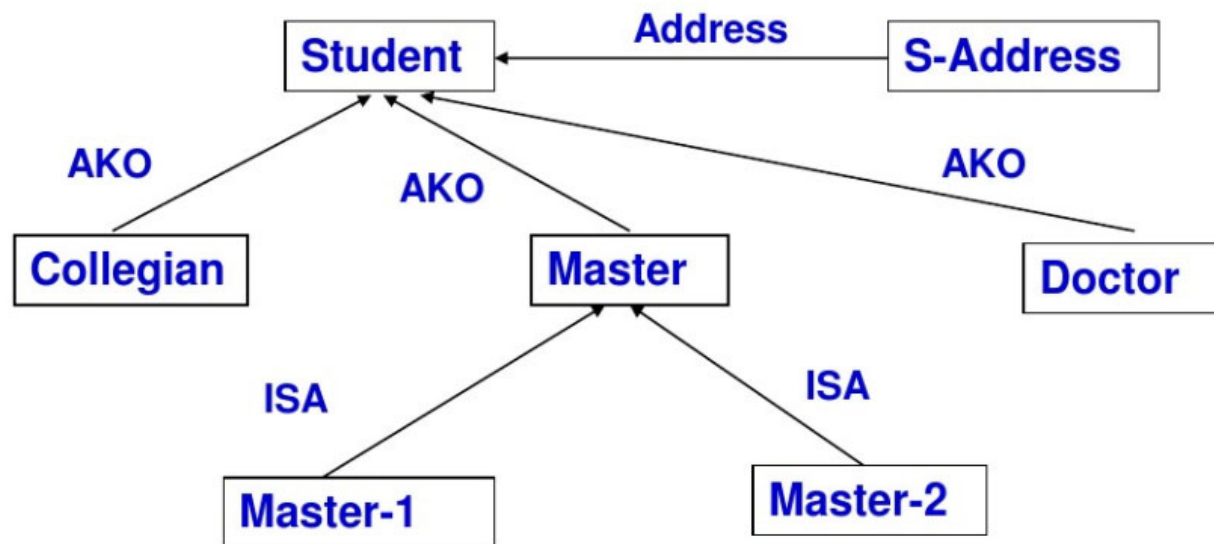
# 框架网络

连接起来组成的框架系统称为**框架网络**。

- **框架之间的纵向联系**：网络中的结点是框架，利用结点之间的关系可由某些框架推出另一些框架；
- **框架之间的横向联系**：网络中的结点即可代表框架，也可代表框架中的槽，每条弧的一头联着某个框架的一个槽，另一头联着另一个框架。

# 框架网络

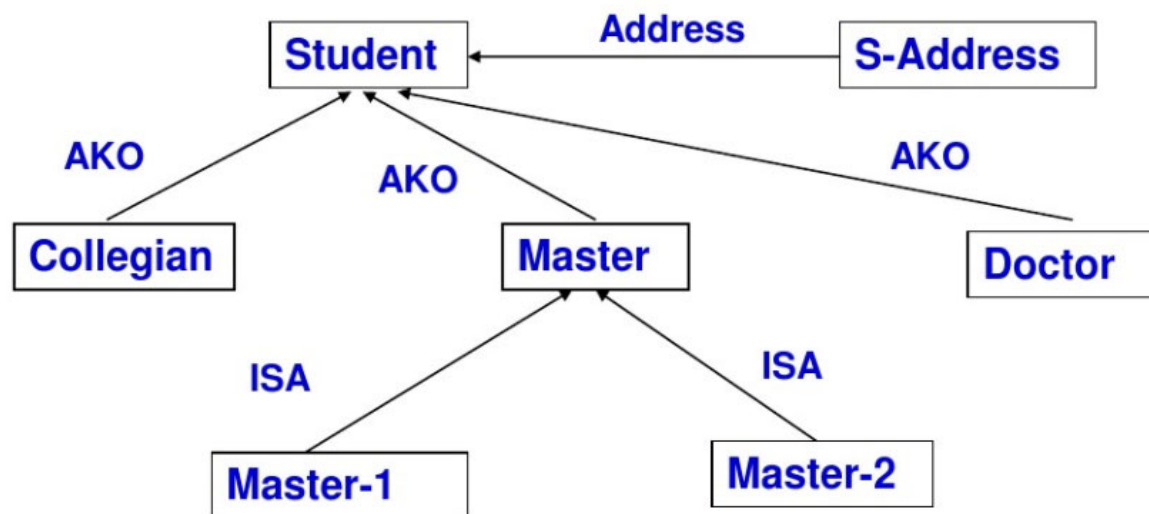
- **框架之间的纵向联系** 指具有继承关系的上下层框架之间的联系。



- AKO实现了student框架与Master框架之间的纵向联系
- ISA实现了Master框架与Master-1实例框架之间的联系

# 框架网络

- **框架之间的横向联系** 指以另外一个框架名作为一个槽的槽值或侧面值所建立起来的框架之间的联系。



# 框架结构

---

框架的一个重要属性是其继承性。为此，一个框架系统常被表示成一种树形结构，树的每一个结点是一个框架结构，子结点与父结点之间用ISA或AKO槽连接，从而有实例框架、类框架、超类框架，等等。所谓框架的继承性，就是当子结点的某些槽值或侧面值没有被直接记录时，可以从其父结点继承这些值，这种继承性和语义网是类似的。

# 框架结构

框架系统具有树状结构。每个结点具有如下框架结构形式：

框架名

框架名用类名表示

AKO VALUE<值>

AKO是一个槽，VALUE是它的侧面

PROP DEFAULT<表1>

SF IF-NEEDED<算术表达式> 附加过程

CONFLICT ADD<表2>

其PROP槽用来记录该结点所具有的特性，其侧面DEFAULT表示该槽的内容是可以进行默认继承的，即当<表1>为非NIL时，PROP的槽值为<表1>，当<表1>为NIL时，PROP的槽值用其父结点的PROP槽值来代替。

# 框架表示下的推理

## 1. 匹配

- 框架的匹配，就是根据已知事实寻找合适的框架。其过程可描述为：根据已知事实，与知识库中预先存储的框架进行匹配——逐槽比较，从中找出一个或几个与该事实所提供情况最适合的预选框架，形成初步假设；然后对所有预选框架进行评估，以决定最适合的预选框架。
- 由于类框架、超类框架等是对一类事物的一般性描述，当应用于某个具体事物时，具体事物往往存在偏离该类事物的某些特殊性，因此，框架的匹配只能做到不完全匹配。



# 框架表示下的推理

## 2. 填槽

填槽就是进行槽值的计算，主要有两种计算槽值的方法：继承和附加过程。

### (1) 继承

是指下层框架可以共享上层框架（直至顶层框架）中定义的有关属性和属性值，又分为：值继承、属性继承和限制继承。

“鸵鸟小白”框架可以继承“鸵鸟框架”中“脚”槽中“数量”侧面的值，这是值继承；“鸵鸟小黑框架”可以继承“鸵鸟框架”的“超类”槽，这是属性继承；“鸵鸟框架”不能继承“鸟框架”中“会飞”槽，这就是限制继承。

# 框架表示下的推理

## 2. 填槽

### (2) 附加过程

附加过程又叫幽灵（DEMON）程序，是附加在数据结构上，并在询问或修改数据结构所存放的值时被激活的过程。

**if-needed:** 按需求值。本程序所属槽的值将被使用而该槽又暂时无值时，自动启动本程序。

**if-added:** 一旦所属槽被赋值则启动本程序。

**if-removed:** 删除本程序所属槽的时候启动本程序。

**if-modified:** 本程序所属槽值被修改时启动本程序。

# 框架表示下的推理

例如，确定一个人的年龄，在已知知识库中要匹配的框架为：

框名：

年龄 NIL

if needed ASK

if added CHECK

这时便自动启动if need 槽的附加过程ASK。例如，当用户输入“25”后，便将年龄槽设定为25，进而启动if added槽执行附加过程CHECK程序，用来检查该年龄值是否合适。如果这个框架有默认槽：default 20，那么当用户没有输入年龄时，就默认年龄为20。

## 框架表示法的特征

---

**优点：** 结构性

深层性

继承性

自然性

**缺点：** 缺乏框架的形式理论

缺乏过程性知识表示

清晰性难以保证

## 10.6 过程式知识表示

- 过程式知识表示就是将有关某一问题领域的知识，连同如何使用这些知识的方法，隐式地表达为一个求解问题的过程。
- 过程表示法的知识描述形式就是程序，所有的信息均隐含在程序中。因而从程序求解问题效率上来说，过程式表达要比陈述式表达要高得多，但缺点是过程式表示难以添加新知识和扩充新的功能。
- 过程式知识表示依赖于具体的问题领域，所以它没有固定的表示形式。

# 知识表示例子

## 产生式系统的例子

例1 已知症状如下：

E1：打喷嚏，E2：咳嗽

诊断结果：

H1：可能感冒

H2：可能花粉过敏

**IF E1 THEN H2 (.50)**

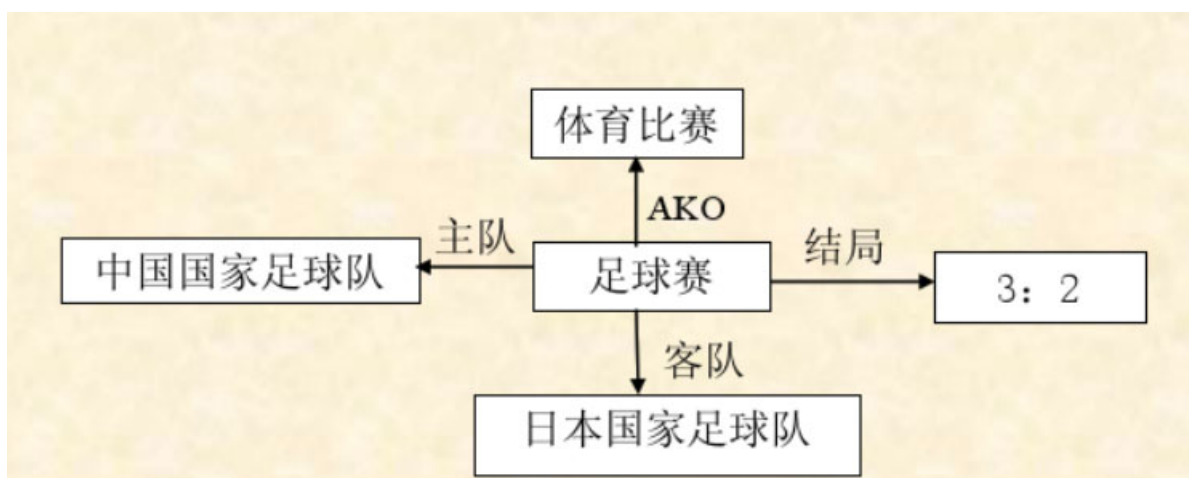
**IF E1,E2 THEN H2(0.90)**

例2 IF 动物会飞 AND 会下蛋 THEN 该动物可能是鸟。

$(\text{FLY}, X, \text{TRUE}) \wedge (\text{EGG}, X, \text{TRUE}) \rightarrow$   
 $(\text{BIRD}, X, \text{TRUE}) (0.8)$

# 知识表示例子

**语义网络的例子：**用语义网络表示知识“中国队与日本队两国的国家足球队在中国进行一场比赛，结局的比分是3:2”

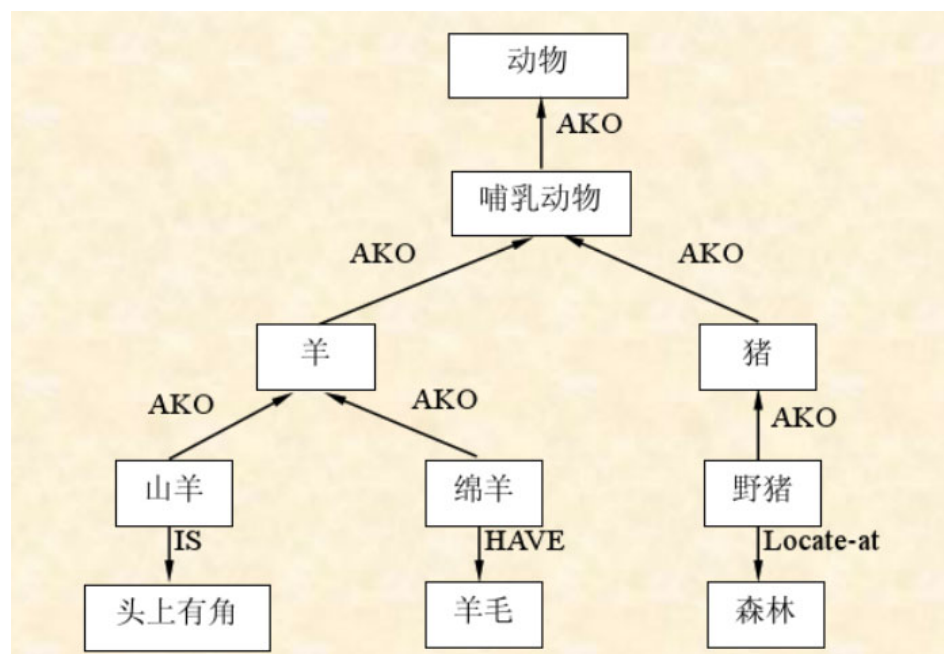


# 知识表示例子

## 语义网络的例子

把下面的命题用语义网络表示：

- (1) 猪和羊都是动物；
- (2) 猪和羊都是哺乳动物；
- (3) 野猪是猪，但生长在森林中；
- (4) 山羊是羊，头上长着角；
- (5) 绵羊是一种羊，它能生产羊毛

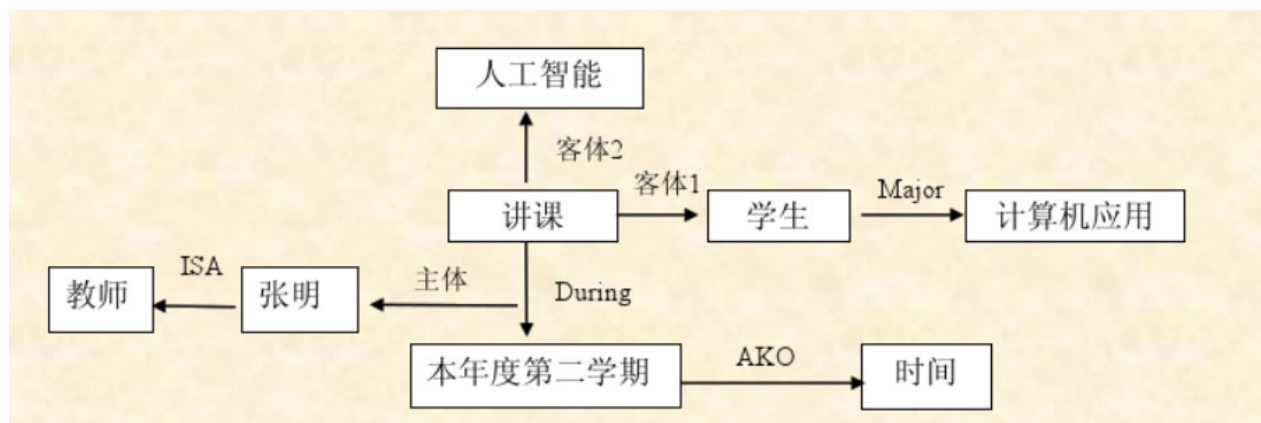




# 知识表示例子

## 语义网络的例子

教师张明在本年度第二学期给计算机应用专业的学生讲授“人工智能”这一门课程。



# 知识表示例子

将下列一则地震消息用**框架**表示：“某年某月某日，某地发生6.0级地震，若以膨胀注水孕震模式为标准，则三项地震前兆中的波速比为0.45，水氡含量为0.43，地形改变为0.60。”

解：地震消息用框架如下图所示。

框架名：〈地震〉

地 点：某地

日 期：某年某月某日

震 级：6.0

波 速 比：0.45

水氡含量：0.43

地形改变：0.60

# 知识表示例子

