

# 第2章 知识表示

1968 年，费根鲍姆研发出第一个专家系统 DENDRAL，人工智能也进入了第二个阶段——**知识工程期**

专家系统：采用人工智能中的**知识表示**和**知识推理**技术，具有专门知识和经验的计算机智能程序系统。

比如说在做探矿系统时，如果看到岩石渗出红色，那么里面很可能是铁矿。把知识总结出来，编入计算机系统，以解决实际问题。

一个系统之所以能展示高级的智能理解和行为，主要是因为在所从事的领域所表现出来的特定知识：概念、事实、表示、方法、比喻以及启发。



费根鲍姆( Feigenbaum)，图灵奖得主，被称为“知识工程之父”

— Feigenbaum

# 第2章 知识表示

按照符号主义的观点，知识是一切智能行为的基础，要使计算机具有智能，首先必须使它拥有知识。

- 2.1 知识与知识表示的概念
- 2.2 一阶谓词逻辑表示法
- 2.3 产生式表示法
- 2.4 语义网络表示法
- 2.5 框架表示法

# 2.1 知识与知识表示的概念

- 2.1.1 知识的概念
- 2.1.2 知识表示的概念

## 2.1.1 知识的概念

### 什么是知识

- 知识的一般概念

知识是人们在改造客观世界的实践中积累起来的认识和经验

- 认识：包括对事物现象、本质、属性、状态、关系、联系和运动等的认识
- 经验：包括解决问题的
  - 宏观方法：如战略、战术、计谋、策略等
  - 微观方法：如步骤、操作、规则、过程、技巧等

## 2.1.1 知识的概念

### 什么是知识

- 知识的有代表性的定义

I. **Feigenbaum:** 知识是经过剪裁、塑造、解释、选择和转换了的信息

II. **Bernstein:** 知识由特定领域的描述、关系和过程组成

III. **Heyes-Roth:** 知识=事实+信念+启发式

## 2.1.1 知识的概念

### 什么是知识

- 知识、数据、信息及其关系

- 数据是信息的载体，本身无确切含义，其关联构成信息（图像—像素矩阵）
- 信息赋予数据特定的含义，可理解为描述性知识
- 知识是对信息的关联，也可以是对已有知识的再认识
  - 常用的关联方式： if ..... then .....

## 2.1.1 知识的概念

### 知识的类型

- 按知识的性质

概念、命题、公理、定理、规则和方法

- 按知识的作用域

常识性知识：通用通识的知识。人们普遍知道的、  
适应所有领域的知识。

领域性知识：面向某个具体专业领域的知识。

例如：专家经验。

## 2.1.1 知识的概念

### 知识的类型

- 按知识的作用效果

叙述性知识：用于描述事物的概念、定义、属性等；  
或用于描述问题的状态、环境、条件等。

过程性知识：用于问题求解过程的操作、演算和行为的知识；用来指出如何使用那些与问题有关的事实性知识的知识；  
表示方式：产生式、谓词、语义网络等。

控制性知识：是关于如何使用过程性知识的知识；  
(元知识或超知识)  
例如：推理策略、搜索策略

## 2.1.1 知识的概念

### 知识的类型

- 按知识的层次

表层知识：描述客观事物的现象的知识。例如：感性、事实性知识

深层知识：描述客观事物本质、内涵等的知识。例如：理论知识

- 按知识的等级

零级知识：叙述性知识

一级知识：过程性知识

二级知识：控制性知识（元知识或超知识）

## 2.1.1 知识的概念

### 知识的类型

- 按知识的确定性

确定性知识：可以说明其真值为真或为假的知识

不确定性知识：包括不精确、模糊、不完备知识

不精确：知识本身有真假，但由于认识水平限制却不能肯定其真假。用可信度、概率等描述

模糊：知识本身的边界就是不清楚的

例如：大，小。用可能性、隶属度描述

不完备：解决问题时不具备解决该问题的全部知识。

例如：医生看病

## 2.1.2 知识表示的概念

### 知识表示的含义及要求

- 什么是知识表示

对知识进行描述，用一组符号把知识编码成计算机可以接受的某种结构。（如何最佳地捕捉智能行为的关键特征以供计算机使用）

- 知识表示的要求

- 表示能力：正确、有效地表示问题

1. 表示范围的广泛性
2. 领域知识表示的高效性
3. 对非确定性知识表示的支持程度

## 2.1.2 知识表示的概念

### 知识表示的含义及要求

- 可利用性：可利用这些知识进行有效推理。
  1. 对推理的适应性：  
推理是根据已知事实利用知识导出结果的过程
  2. 对高效算法的支持程度：  
知识表示要有较高的处理效率
- 可实现性：要便于计算机直接对其进行处理
- 可组织性：可以按某种方式把知识组织成某种知识结构
- 可维护性：便于对知识的增、删、改等操作
- 自然性：符合人们的日常习惯
- 可理解性：知识应易读、易懂、易获取等

## 2.1.2 知识表示的概念

### 知识表示的观点

- 知识表示的观点

- 陈述性观点：知识的存储与知识的使用相分离

- 优点：灵活、简洁，演绎过程完整、确定，

- 知识维护方便

- 缺点：推理效率低、推理过程不透明

- 过程性观点：知识寓于使用知识的过程中

- 优点：推理效率高、过程清晰

- 缺点：灵活性差、知识维护不便

## 2.1.2 知识表示的概念

### 知识表示的方法

- 知识表示的方法
  - 逻辑表示法：一阶谓词逻辑  
基于语法的策略，目标应用通用广泛
  - 产生式表示法：产生式规则  
大量使用特定问题域的经验
  - 结构表示法：语义网络，框架  
对象及其之间的关系（概念关联，层次分类）
  - 过程表示法：  
~~显示表示问题域的基础理论、功能及求解过程~~

# 第2章 知识表示

- 2.1 知识表示与知识表示的概念
- 2.2 一阶谓词逻辑表示法
- 2.3 产生式表示法
- 2.4 语义网络表示法
- 2.5 框架表示法

## 2.2 一阶谓词逻辑表示法

- 一阶谓词逻辑表示法: 基于数理逻辑的表示方法

□ 哲学家 + 数学家:

如何形式化地描述合理、完备的推理规则

□ 强 调:

合式表达式的“真值保持”(truth-preserving)运算

数理逻辑是一门研究推理的学科:

分为 {  
    ➤ 一阶经典逻辑: 一阶经典命题逻辑, 一阶经典谓词逻辑  
    ➤ 非一阶经典逻辑: 指除经典逻辑以外的那些逻辑

例如: 二阶逻辑, 多值逻辑, 模糊逻辑等。

## 2.2 一阶谓词逻辑表示法

主要讨论：

- 一阶谓词逻辑表示的逻辑学基础  
    命题和真值；论域和谓词；连词和量词；  
    项与合式公式；自由变元与约束变元
- 谓词逻辑表示方法
- 谓词逻辑表示的应用
- 谓词逻辑表示的特性

## 2.2.1一阶谓词逻辑表示的逻辑学基础

### 命题与真值

- 命题的定义:

断言: 定义2.1 一个陈述句称为一个断言.

命题: 具有真假意义的断言称为命题.

- 命题的真值:

T: 命题意义为真

F: 命题意义为假

- 命题真值的说明

一个命题不能同时既真又假

一个命题可在一定条件下为真, 而在另一条件下为假

## 2.2.1一阶谓词逻辑表示的逻辑学基础 论域和谓词(1/2)

- 论域  $D$ ：由所讨论对象的全体构成的集合
- 个体  $x$ ：论域中的元素,表示独立存在的事物或概念
- 谓词  $P$ : 命题是用形如 $P(x_1, x_2, \dots, x_n)$ 的谓词来表示的,表示个体间关系

定义2.2 设 $D$ 是论域,  $P: D^n \rightarrow \{T, F\}$ 是一个映射, 其中

$$D^n = \{(x_1, x_2, \dots, x_n) \mid x_1, x_2, \dots, x_n \in D\}$$

若 $P(x_1, x_2, \dots, x_n)$ , 则称 $P$ 是一个n元谓词,其中  $x_i$  为个体, 可以是个体常量、变元和函数。

例如: GREATER(x,6)                            x大于6

TEACHER(father(Wang Hong))    王宏的父亲是一位教师

## 2.2.1一阶谓词逻辑表示的逻辑学基础 论域和谓词(2/2)

- 函数:

定义2-3

设 $D$ 是个体域,  $f: D^n \rightarrow D$ 是一个映射, 其中

$$D^n = \{(x_1, x_2, \dots, x_n) \mid x_1, x_2, \dots, x_n \in D\}$$

则称 $f$ 是 $D$ 上的一个n元函数, 记作 $f(x_1, x_2, \dots, x_n)$

- 谓词与函数的区别:

谓词是 $D$ 到 $\{T, F\}$ 的映射, 函数是 $D$ 到 $D$ 的映射

谓词的真值是 $T$ 和 $F$ , 函数的值(无真值)是 $D$ 中的元素

谓词可独立存在, 函数只能作为谓词的个体

## 2.2.1一阶谓词逻辑表示的逻辑基础 连词

- 连词：

$\neg$ ：“非”，表示对其后面的命题的否定

$\vee$ ：“析取”，命题之间“或”关系

$\wedge$ ：“合取”，命题之间“与”关系。

$\rightarrow$ ：“蕴含”，“如果P，则Q”。

$\leftrightarrow$ ：称为“双条件”。表示“当且仅当”。

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

连词的优先级： $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

## 2.2.1一阶谓词逻辑表示的逻辑基础 量词

- 量词：
  - $\forall$ ：全称量词，意思是“任一个”、“所有的”  
命题( $\forall x$ )P(x)为真，当且仅当对论域中的所有x，都有P(x)为真  
命题( $\forall x$ )P(x)为假，当且仅当至少存在一个 $x_i \in D$ ，使得P( $x_i$ )为假
  - $\exists$ ：存在量词，意思是“至少有一个”、“存在有”  
命题( $\exists x$ )P(x)为真，当且仅当至少存在一个 $x_i \in D$ ，使得P( $x_i$ )为真  
命题( $\exists x$ )P(x)为假，当且仅当对论域中的所有x，都有P(x)为假

## 2.2.2 谓词逻辑表示方法

- 知识的表示步骤：

- (1)先根据要表示的知识定义谓词
- (2)再用连词、量词把这些谓词连接起来

例2.1 表示知识“所有教师都有自己的学生”。

1. 定义谓词：

$T(x)$ : 表示 $x$ 是教师。

$S(y)$ : 表示 $y$ 是学生。

$TS(x, y)$ : 表示 $x$ 是 $y$ 的老师。

2. 表示知识：

$$(\forall x)(\exists y)(T(x) \rightarrow TS(x, y) \wedge S(y))$$

可读作：对所有 $x$ ，如果 $x$ 是一个教师，那么一定存在一个个体 $y$ ， $y$ 的老师是 $x$ ，且 $y$ 是一个学生。

## 2.2.3 谓词逻辑表示的应用

### 机器人移盒子问题(1/5)

- 分别定义描述状态和动作的谓词

- 描述状态的谓词：

**TABLE(x):** x是桌子

**EMPTY(y):** y手中是空的

**AT(y, z):** y在z处

**HOLDS(y, w):** y拿着w

**ON(w, x):** w在x桌面上

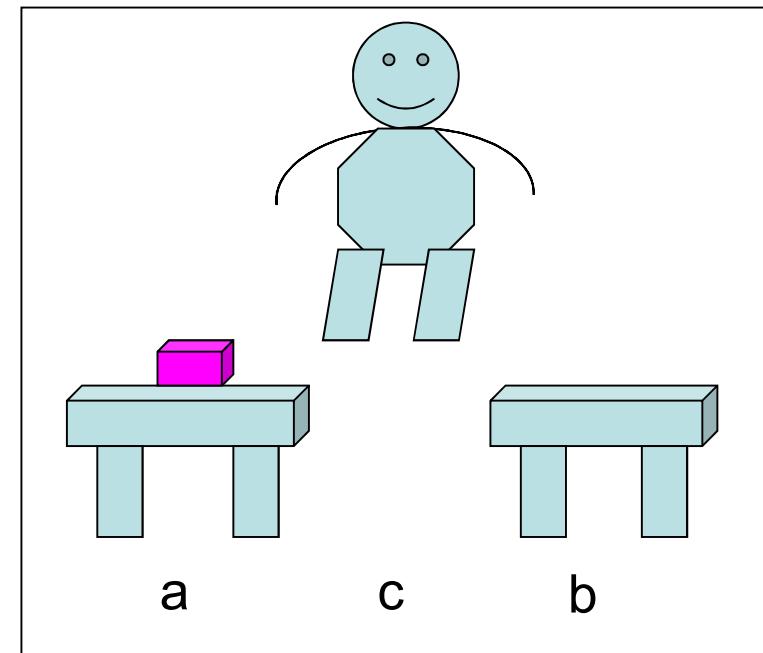
- 变元的论域：

x的论域是{a, b}

y的论域是{robot}

z的论域是{a, b, c}

w的论域是{box}



## 2.2.3 谓词逻辑表示的应用

### 机器人移盒子问题(2/5)

初始状态:

AT(robot, c)

EMPTY(robot)

ON(box, a)

TABLE(a)

TABLE(b)

目标状态:

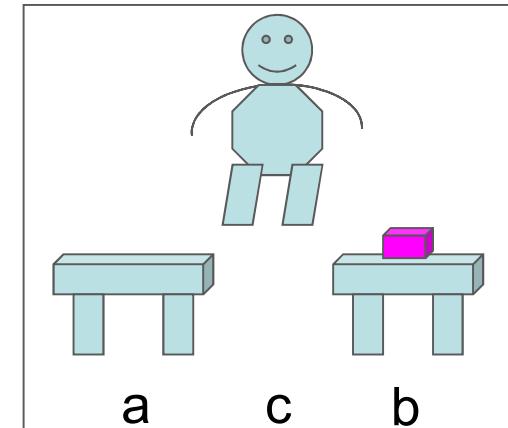
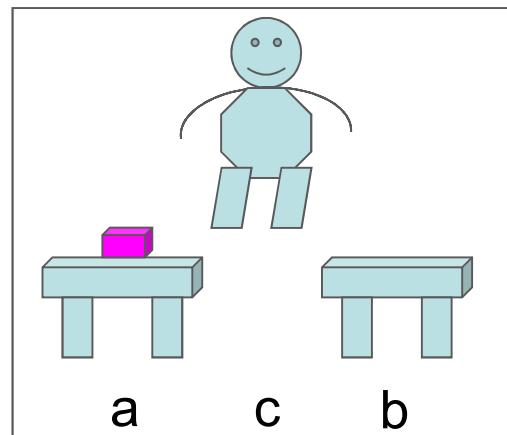
AT(robot, c)

EMPTY(robot)

ON(box, b)

TABLE(a)

TABLE(b)



机器人行动的目标 :从初始状态转换为目标状态，而要实现  
问题状态的转换需要完成一系列的操作

## 2.2.3 谓词逻辑表示的应用

### 机器人移盒子问题(3/5)

- 各操作的条件和动作：

**Goto(x, y):** 从x处走到y处

条件: AT(robot, x)

动作: 删除表: AT(robot, x)

添加表: AT(robot, y)

**Pickup(x):** 在x处拿起盒子

条件: ON(box, x), TABLE(x), AT(robot, x), EMPTY(robot)

动作: 删除表: EMPTY(robot), ON(box, x)

添加表: HOLDS(robot, box)

**Setdown(x):** 在x处放下盒子

条件: AT(robot, x), TABLE(x), HOLDS(robot, box)

动作: 删除表: HOLDS(robot, box)

添加表: EMPTY(robot), ON(box, x)

执行操作前，要检查先决条件是否满足。如满足，执行相应的操作；否则再检查下一个操作。

## 2.2.3 谓词逻辑表示的应用

### 机器人移盒子问题(4/5)

- 这个机器人行动规划问题的求解过程如下：

#### 状态1(初始状态)

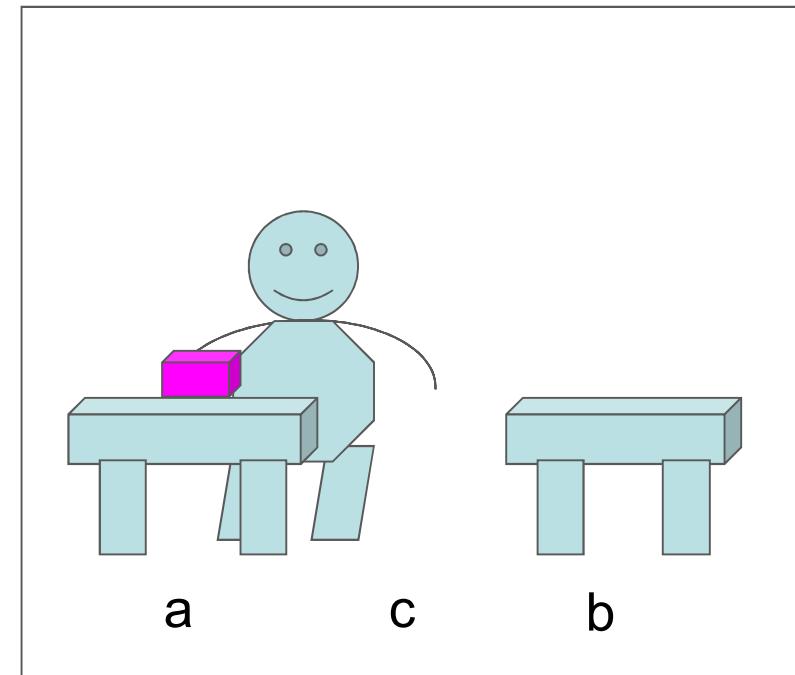
开始            AT(robot, c)  
=====>    EMPTY(robot)  
                ON(box, a)  
                TABLE(a)  
                TABLE(b)

#### 状态2

Goto(c, a)    AT(robot, a)  
=====>    EMPTY(robot)  
                ON(box, a)  
                TABLE(a)  
                TABLE(b)

#### 状态3

Pickup(a)    AT(robot, a)  
=====>    HOLDS(robot, box)  
                TABLE(a)  
                TABLE(b)



## 2.2.3 谓词逻辑表示的应用

### 机器人移盒子问题(5/5)

状态4

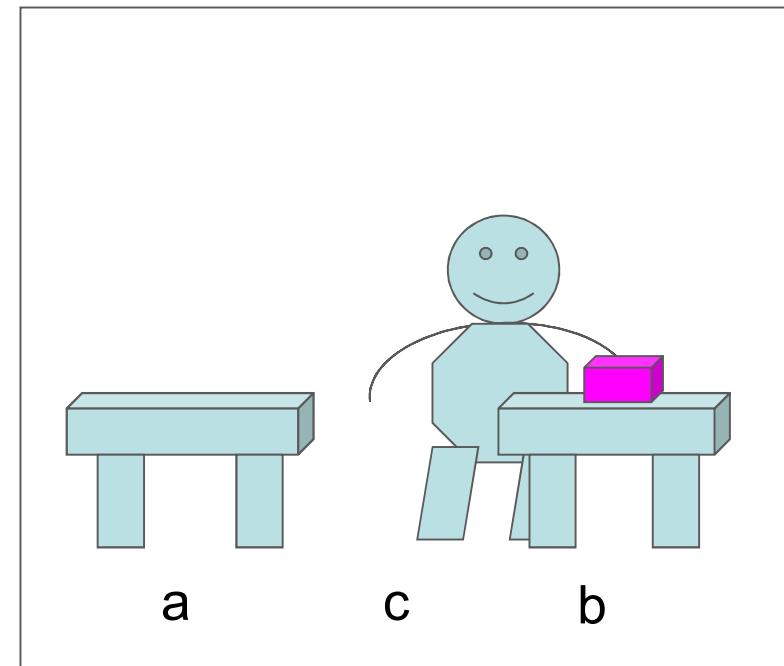
**Goto(a, b)**      AT(robot, b)  
=====>      HOLDS(robot, box)  
                  TABLE(a)  
                  TABLE(b)

状态5

**Setdown(b)**      AT(robot, b)  
=====>      EMPTY(robot)  
                  ON(box, b)  
                  TABLE(a)  
                  TABLE(b)

状态6(目标状态)

**Goto(b, c)**      AT(robot, c)  
=====>      EMPTY(robot)  
                  ON(box, b)  
                  TABLE(a)  
                  TABLE(b)



## 2.2.4 谓词逻辑表示的特征

- 优点

自然：一阶谓词逻辑接近于自然语言的形式，接近于人对问题的直观理解

明确：有一套标准的知识解释方法，这种方法表示的知识明确、易于理解

精确：谓词逻辑只有“真”与“假”，其表示、推理都是精确的

灵活：知识和处理知识的程序是分开的，无须考虑处理知识的细节

模块化：知识之间相对独立，添加、删除、修改知识比较容易

- 缺点

知识表示能力差：不能表示非确定性知识、过程性知识和启发式知识

知识库管理困难：缺乏知识的组织原则，知识库管理比较困难

存在组合爆炸：由于难以表示启发式知识，因此只能盲目地使用推理规则，  
这样当系统知识量较大时，容易发生组合爆炸

系统效率低：推理演算与知识含义截然分开，抛弃了语义信息，使推理过  
程冗长，降低了系统效率

# 第2章 知识表示

- 2.1 知识表示与知识表示的概念
- 2.2 一阶谓词逻辑表示法
- 2.3 产生式表示法
- 2.4 语义网络表示法
- 2.5 框架表示法

- 逻辑表示只强调真实性，忽略了前提与结论间的特定关系

“ $\vee$ ” = or

“ $\rightarrow$ ” = “if...then...”

“如果一只鸟是和平鸽，那么它就是白色的”

$\forall X(pigeon(X) \rightarrow white(X))$

等价于： $\forall X(\sim white(X) \rightarrow \sim pigeon(X))$

考虑事实：“这块木炭是黑色的，则它不是和平鸽”

用逻辑推理得出荒谬的推理结论

丢失了特定的关系，如相关性，因果性等如：类属关系，“类中成员拥有该类的属性”

- 产生式：心理学家 + 语言学家的观点

如何刻画人类理解的本质

不关心如何建立正确推理的规则

关心如何描述人类获取和使用周遭世界知识的实际方式

# 产生式系统的适用领域

产生式（Production）是目前人工智能中使用最多的一种知识表示方法

(1) 可表示由许多相对独立的知识元组成的领域知识，彼此之间关系不密切，不存在结构关系。如：化学反应方面的知识，

**DENDRAL**帮助判定物质的分子结构的专家系统

(2) 可表示经验性及不确定性的知识，而且相关领域中对这些知识没有严格、统一的理论。如：医疗诊断、故障诊断等方面的知识，

**MYCIN**感染菌诊断以及抗生素给药推荐系统

# 2.3 产生式表示法

## 2.3.1 产生式表示的基本方法

事实的表示

规则的表示

产生式与蕴含式的区别

产生式与条件语句的区别

## 2.3.2 产生式系统的**基本结构**

## 2.3.3 产生式系统的**基本过程**

## 2.3.4 产生式系统的**控制策略**

## 2.3.5 产生式系统的**类型**

## 2.3.6 产生式系统的**特性**

## 2.3.1 产生式表示的基本方法

### 事实的表示

- 事实的定义

断言一个语言变量的值或断言多个语言变量之间关系的陈述句

例如：“雪是白的”，“雪”——语言变量，“白的”——语言变量的值

“王峰酷爱篮球”，“王峰”和“篮球”是两个语言变量，“酷爱”是语言变量之间的关系

- 事实的表示

确定性知识，事实可用如下三元组表示：

(对象, 属性, 值) 或 (关系, 对象1, 对象2)



(雪, 颜色, 白)

(酷爱, 王峰, 篮球)

非确定性知识，事实可用如下四元组表示：

(对象, 属性, 值, 可信度因子)

其中，“可信度因子”是指该事实为真的相信程度。可用 [0,1] 之间的一个实数来表示。

## 2.3.1 产生式表示的基本方法

### 规则的表示

- 规则的作用

描述事物之间的因果关系。

规则的产生式表示形式称为产生式或规则。

- 产生式的基本形式

$P \rightarrow Q$  或者  $if \ P \ then \ Q$

$P$ 是产生式的前提，也称为前件

$Q$ 是一组结论或操作，也称为产生式的后件

例子：

$r_6: IF$  动物有犬齿 **AND** 有爪 **AND** 眼盯前方

**THEN** 该动物是食肉动物

其中， $r_6$ 是该产生式的编号；

“动物有犬齿 **AND** 有爪 **AND** 眼盯前方”是产生式的前提P；

“该动物是食肉动物”是产生式的结论Q。

## 2.3.1 产生式表示的基本方法

产生式与蕴涵式、条件语句的区别

- 与蕴涵式的主要区别：

(1) 蕴涵式表示的知识只能是精确的，产生式表示的知识可以是不确定的

原因是蕴涵式是一个逻辑表达式，其逻辑值只有真和假。

(2) 蕴含式的匹配一定要求是精确的，而产生式的匹配可以是不确定的

原因是产生式的前提条件和结论都可以是不确定的，因此其匹配也可以是不确定的。

## 2.3.1 产生式表示的基本方法 产生式与蕴涵式、条件语句的区别

例如，专家系统MYCIN中有如下产生式：

*IF* 本微生物的染色斑呈阴性

本微生物的形状是杆状

病人是中间宿主

*THEN* 该微生物是绿脓杆菌，置信度CF=0.6

CF表示知识的强度，在谓词逻辑中的蕴涵式则不可以。

## 2.3.1 产生式表示的基本方法

产生式与蕴涵式、条件语句的区别

- 与条件语句的主要区别：

### (1) 前件结构不同

产生式的前件可以是一个复杂的的结构

传统程序设计语言中的左部仅仅是一个布尔表达式

### (2) 控制流程不同

产生式中满足前提条件后，不一定被立即执行，能否执行将取决于冲突消解策略

传统程序设计语言中是严格地从一个条件语句向其下一个条件语句传递。

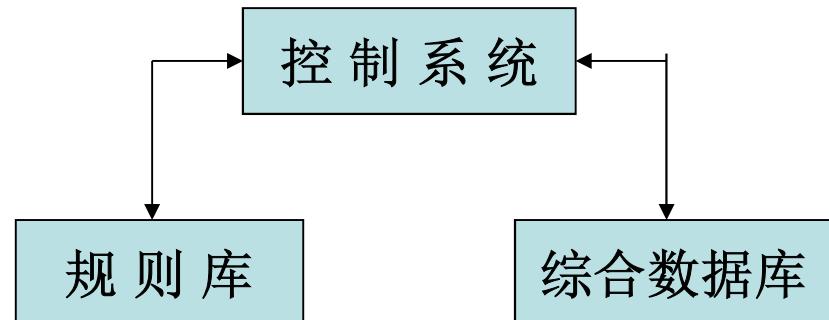
## 2.3.2 产生式系统的基本结构

### 系统结构及其说明(1/2)

- **综合数据库DB(Data Base)**

存放求解问题的各种当前信息  
如：问题的初始状态

输入的事实  
中间结论及最终结论等



- **规则库RB(Rule Base)**

也称知识库KB(Knowledge Base),用于存放与求解问题有关的所有规则的集合

作用：是产生式系统问题求解的基础

要求：具有完整性、一致性、准确性、灵活性和知识组织的合理性

推理过程中，当规则库中某条规则的前提可以和综合数据库的已知事实匹配时，该规则被激活，由它推出的结论将被作为新的事实放入综合数据库，成为后面推理的已知事实。

## 2.3.2 产生式系统的基本结构 系统结构及其说明(2/2)

- 控制系统(Control system)

控制系统的主要作用

用于控制整个产生式系统的运行，决定问题求解过程的推理线路。

控制系统的主要任务

选择匹配：从规则库中选择规则与综合数据库中进行匹配。匹配成功，该规则为可用；否则，称匹配失败，该规则不可用。

冲突消解：匹配成功后，按照某种策略从中选出一条规则执行。

执行操作：对所执行的规则，若其后件为一个或多个结论，则把这些结论加入综合数据库；若其后件为一个或多个操作时，执行这些操作。

终止推理：检查综合数据库中是否包含有目标，若有，则停止推理。

路径解释：在问题求解过程中，记住应用过的规则序列，以便最终能够给出问题的解的路径。

## 2.3.2 产生式系统的基本结构

### 产生式系统的例子

- 动物识别系统
- 该系统可以识别老虎、金钱豹、斑马、长颈鹿、企鹅、信天翁这6种动物。其规则库包含多条规则：

$r_1$  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

:

:

$r_{13}$  IF 该动物是哺乳动物 AND 有蹄 THEN 该动物是有蹄类动物

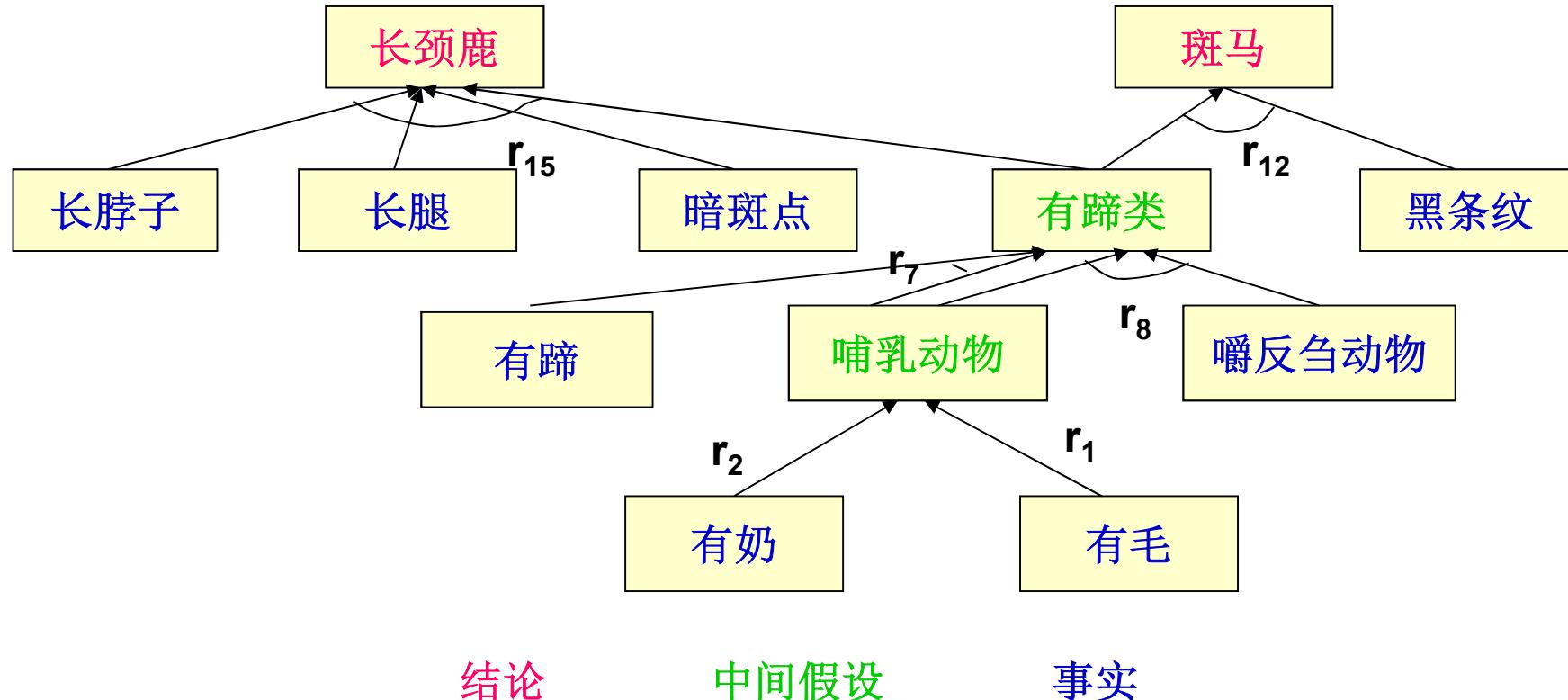
$r_{14}$  IF 该动物是哺乳动物 AND 是嚼反刍动物 THEN 该动物是有蹄类动物

$r_{15}$  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

## 2.3.2 产生式系统的基本结构

### 产生式系统的例子

部分推理网络如下：



初始综合数据库包含的事实有：

动物有暗斑点，有长脖子，有长腿，有奶，有蹄

检测同各规则是否匹配,发现与  $r_2, r_7, r_{15}$  匹配

## 2.3.4 产生式系统的控制策略

- 总体上可分为以下两种方式

### 1. 不可撤回方式

“一路往前走”不回头的方式

它即根据当前已知的局部知识选取一条规则作用于当前综合数据库，接着再根据新状态继续选取规则，如此进行下去，**不考虑撤回用过的规则**。

不理想规则的应用会降低效率，但不影响可解性。

优点是控制过程简单，缺点是当问题**有多个解时不一定能找到最优解**

### 2. 试探性方式

回溯方式

是一种碰壁回头的方式。即在问题求解过程中，允许先试一试某条规则，若发现这条规则不合适，则允许退回，再另选一条规则来试。

图搜索方式

用图或树把全部求解过程，由于它记录了已试过的所有路径，因此便于从中选取最优路径。

## 2.3.5 产生式系统的类型

### 按规则库的性质及结构(1/3)

- 可分解的产生式系统

它是把一个整体问题分解成若干个子问题，然后再通过对这些子问题的求解来得到整个问题解的一种产生式系统。

例2.7 设综合数据库的初始状态为{C, B, Z}，目标状态为{M, M, ..., M}，规则库中有如下重写规则：

$$r_1: C \rightarrow \{D, L\}$$

$$r_2: C \rightarrow \{B, M\}$$

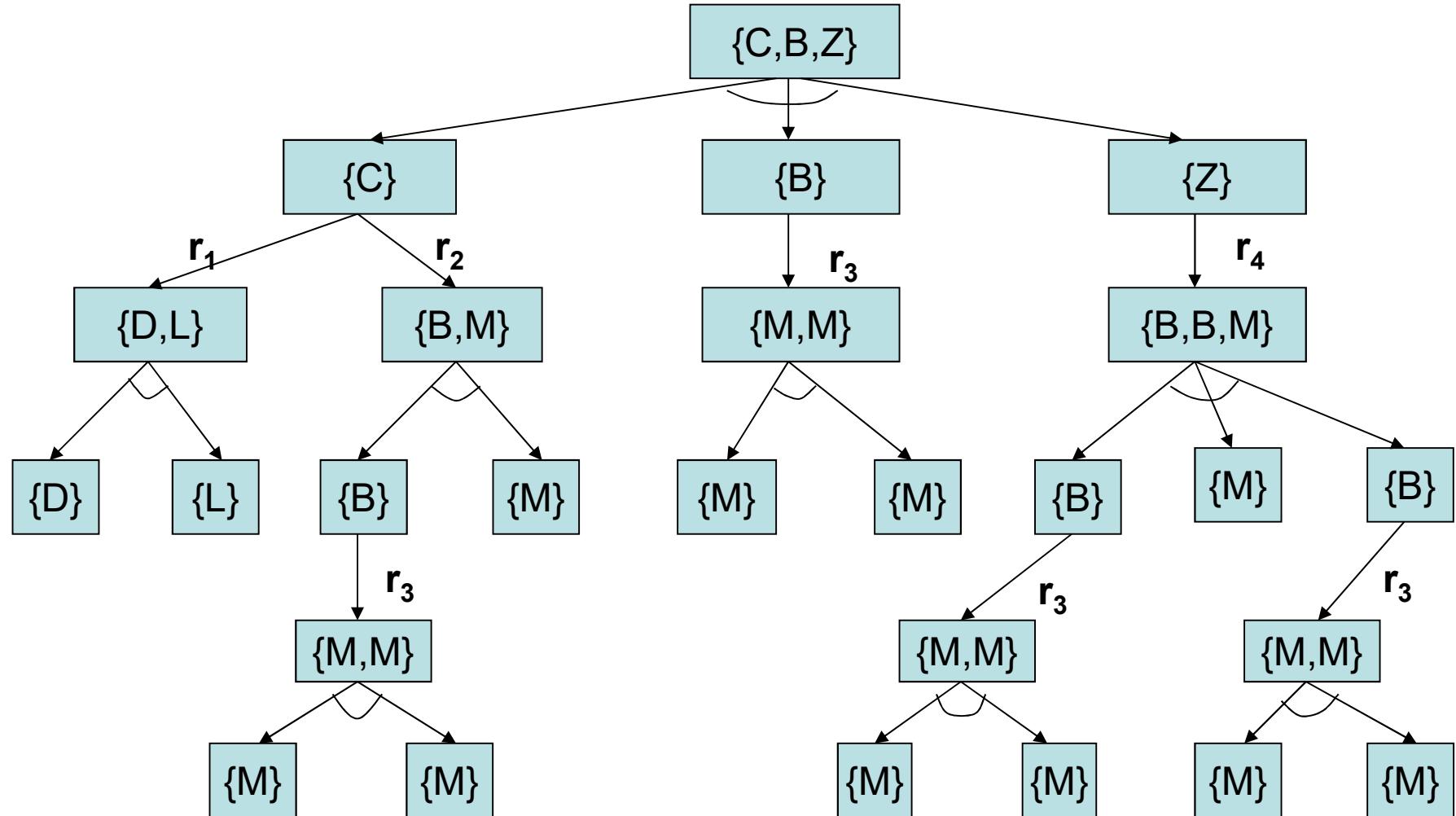
$$r_3: B \rightarrow \{M, M\}$$

$$r_4: Z \rightarrow \{B, B, M\}$$

解决该问题时，可先把初始综合数据库分为三个子库，然后对这三个子库分别应用规则库中的相应规则进行求解。其求解过程如下图所示。

## 2.3.5 产生式系统的类型

### 按规则库的性质及结构(2/3)



## 2.3.6 产生式系统的特点

- 主要优点

- 自然性：采用“如果……，则……”的形式，人类的判断性知识基本一致。
- 模块性：规则是规则库中最基本的知识单元，各规则之间只能通过综合数据库产生联系，不能互用，从而增加了规则的模块性。
- 有效性：产生式知识表示法既可以表示确定性知识，又可以表示不确定性知识，既有利于表示启发性知识，又有利 于表示过程性知识。
- 一致性：规则库中的所有规则都具有相同的格式，并且综合数据库可被所有规则访问，因此规则库中的规则可以统一处理。

## 2.3.6 产生式系统的特点

- 主要缺点

- 效率较低：各规则之间的联系必须以综合数据库为媒介。并且，其求解过程是一种反复进行的“匹配—冲突消解—执行”过程。这样的执行方式将导致执行的低效率。
- 不便于表示结构性知识：由于产生式表示中的知识具有一致格式，且规则之间不能相互调用，因此那种具有结构关系或层次关系的知识则很难以自然的方式来表示。

# 第2章 知识表示

- 2.1 知识表示与知识表示的概念
- 2.2 一阶谓词逻辑表示法
- 2.3 产生式表示法
- 2.4 语义网络表示法
- 2.5 框架表示法

## 2.4 语义网络表示法

- 1968年J.R.Quillian在其博士论文中最先提出语义网络，把它作为人类联想记忆的一个显式心理学模型，设计可教式语言理解器TLC（Teachable Language Comprehenden），用作知识表示方法。
  - 人类可以把信息放在最合适的分类层次上
  - “金丝雀是鸟吗？” “金丝雀会飞吗？” “金丝雀会唱歌吗？”

回答问题的时间长短有差别

记忆层次结构有多层分级，越一般的特征层次越高，回忆更快

1972年，西蒙在他的自然语言理解系统中也采用了语义网络表示法。

1975年，亨德里克(G.G.Hendrix)又对全称量词的表示提出了语义网络分区技术。

## 2.4 语义网络表示法

- 2.4.1 语义网络的基本概念
- 2.4.2 事物和概念的语义网络表示
- 2.4.3 情况和动作的语义网络表示
- 2.4.4 逻辑关系的语义网络表示
- 2.4.5 语义网络的求解过程
- 2.4.6 语义网络表示法的特征

## 2.4.1 语义网络的基本概念

### 什么是语义网络

- 什么是语义网络

语义网络是一种有向图,用实体及其语义关系来表达知识。

结点代表实体，表示各种事物、概念、属性、状态、事件、动作等；

弧代表语义关系，表示连结的实体间的语义联系，它必须带有标识。

- 语义基元

是语义网络中最基本的语义单元，可用三元组表示为：

(结点1, 弧, 结点2)

- 基本网元

与语义基元对应的有向图

例如：有语义基元 (A, R, B) ,对应的基本网元如下图所示：



其中， A、B : 结点， R : A与B之间的某种语义联系

## 2.4.1 语义网络的基本概念

### 基本的语义关系

- 实例关系： ISA  
含义为“是一个”



- 分类关系： AKO  
体现“子类与超类”的概念，含义为“是一种”



- 上述关系的主要特征  
最主要特征是属性的继承性，结点可以继承抽象层结点的所有属性。
- 属性关系  
指事物和其属性之间的关系。例如：“鸟有翅膀”



其他：包含关系，时间关系，位置关系，相近关系等

## 2.4.2 事物和概念的表示

### 表示一元关系

- 一元关系  $P(x)$

描述的是一些最简单、最直观的事物或概念，

常用：“是”、“有”、“会”、“能”等语义关系来说明例如，“李刚是一个人”为一元关系，其语义网络如前所示。

- 二元关系  $P(x,y)$

$x, y$ 为实体， $P$ 为实体之间的关系。

单个二元关系可直接用一个基本网元来表示

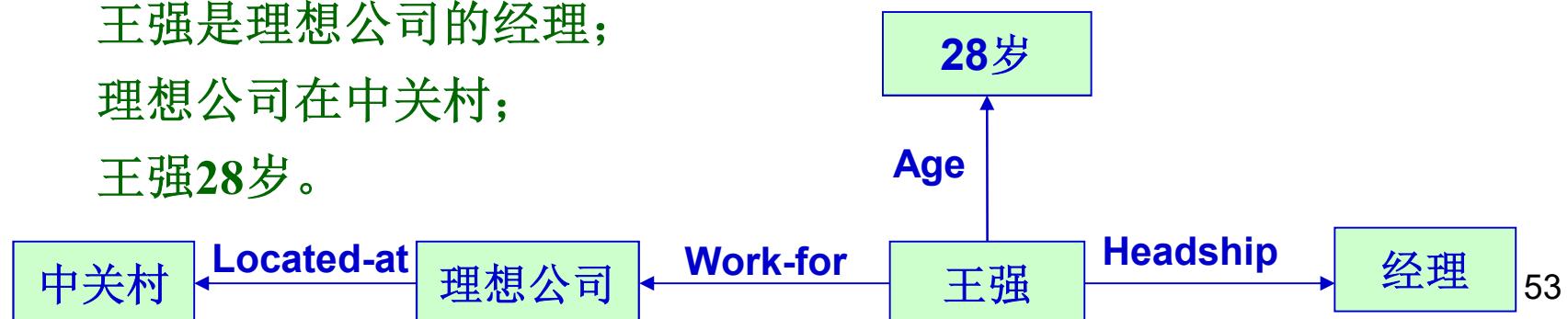
对复杂关系，可通过相对独立的二元或一元关系的组合来实现。

- 例2-10 用语义网络表示：

王强是理想公司的经理；

理想公司在中关村；

王强28岁。



## 2.4.4 逻辑关系的表示

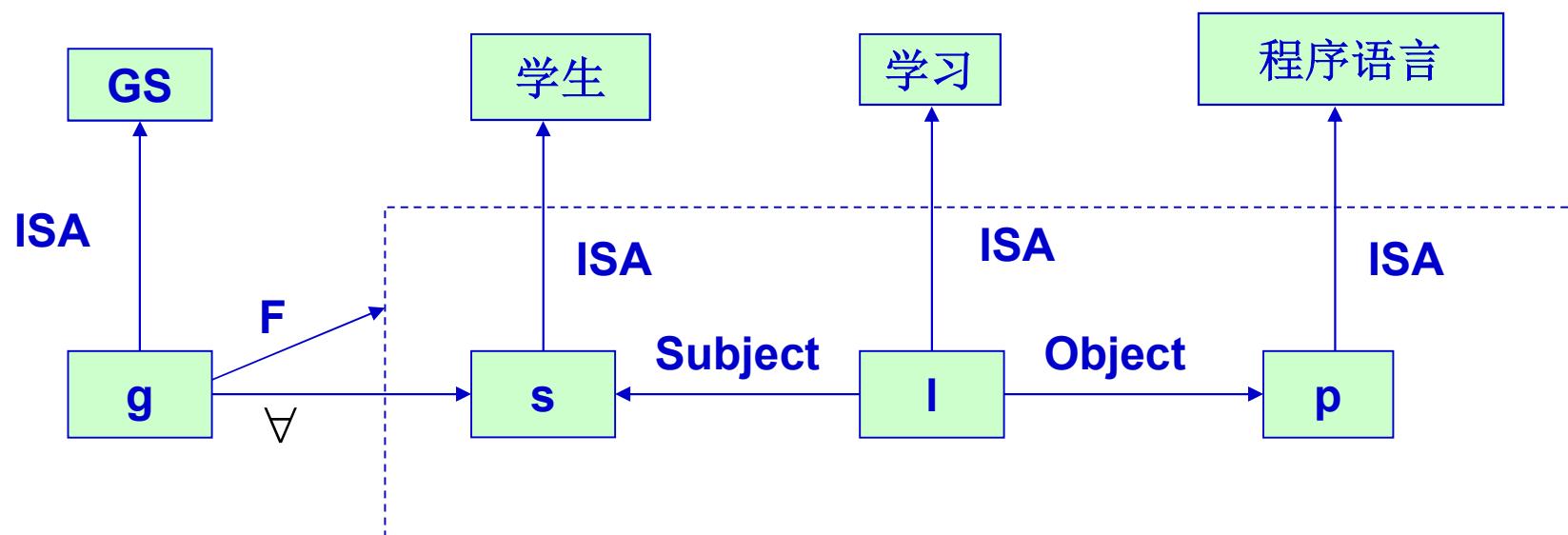
### 存在和全称量词的表示(1/4)

- 存在量词：可直接用“ISA”、“AKO”等这样的语义关系来表示
- 全称量词：可采用亨德里克提出的网络分区技术
  - 基本思想：把一个复杂命题划分为若干个子命题，每个子命题用一个较简单的语义网络表示，称为一个子空间，多个子空间构成一个大空间。每个子空间看作是大空间中的一个结点，称作超结点。空间可逐层嵌套，子空间之间用弧互相连结。
- 例2-19 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”
  - 其语义网络如下图。在该图中：
    - GS是一个概念结点，它表示具有全称量化的一般事件。
    - g是一个实例结点，代表GS中的一个具体例子，如上所提到的事实。
    - 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。
    - 这样，s、l、p之间的语义联系就构成一个子空间，它表示对每一个学生s，都存在一个学习事件l和一门程序设计语言p。

## 2.4.4 逻辑关系的表示

### 存在和全称量词的表示(2/4)

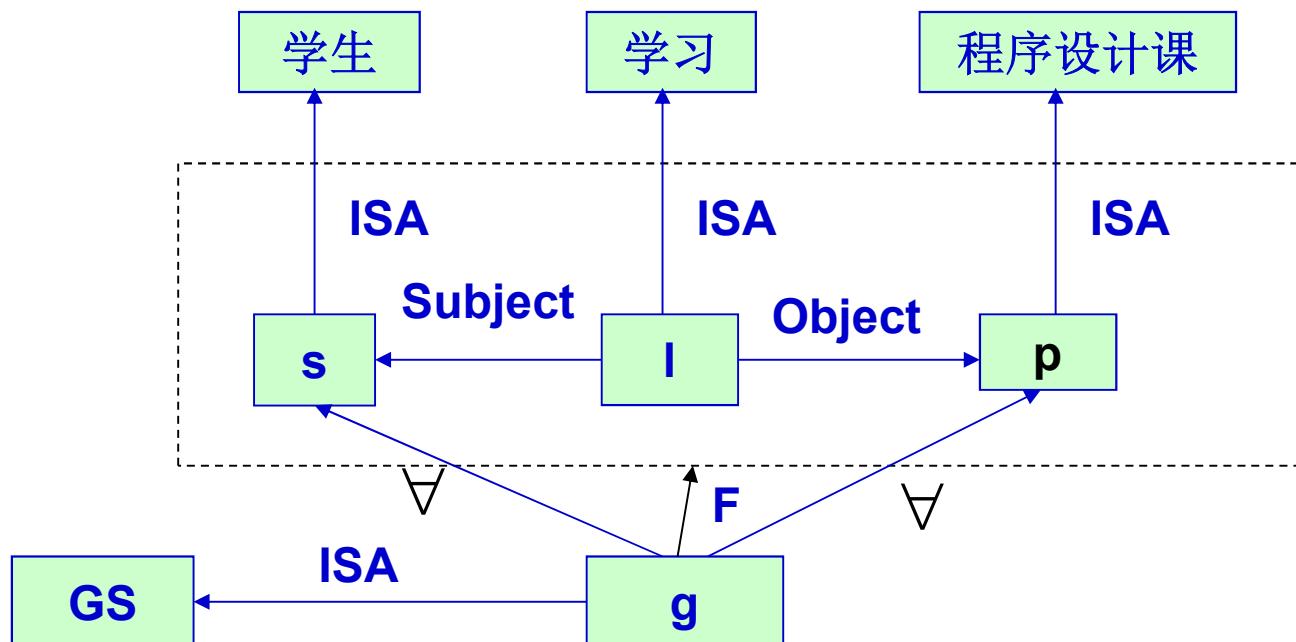
- 在从结点g引出的三条弧中，弧“ISA”说明结点g是GS中一个实例；弧“F”说明它所代表的子空间及其具体形式；弧“ $\forall$ ”说明它所代表的全称量词。



## 2.4.4 逻辑关系的表示

### 存在和全称量词的表示(3/4)

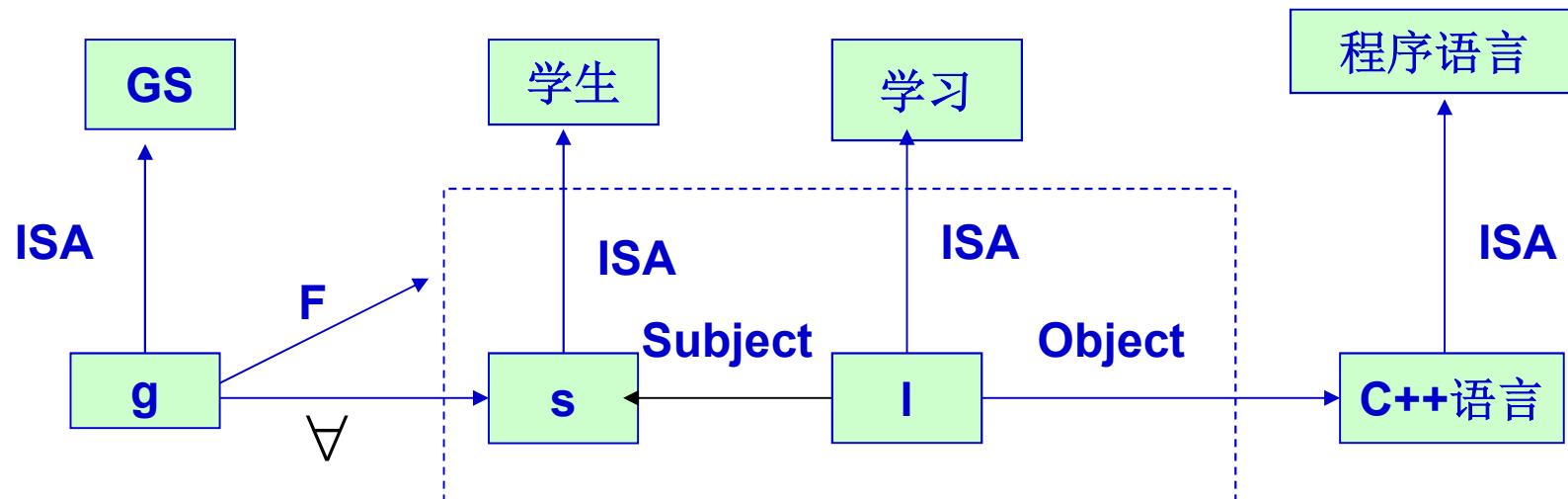
- 每一个全称量词都需要一条这样的弧，子空间中有多少个全称量词，就需要有多少条这样的弧。
- 例2-19 用语义网络表示事实：
- “每个学生都学习了所有的程序设计课程”
- 其语义网络如下图所示。其中，结点g有两条指向全称变量的弧。



## 2.4.4 逻辑关系的表示

### 存在和全称量词的表示(4/4)

- 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全称变量结点的函数(存在量词约束)，否则应放在子空间的外面。
- 例2-21：用语义网络表示事实：
- “每个学生都学习了C++语言”
- 其语义网络如下图所示。结点“C++语言”代表一门具体的程序设计语言，是结点“程序语言”的一个实例，故被放到F所指的子空间的外边



## 2.4.6 语义网络表示法的特征

- 主要优点：

- 结构性 显式地表示出事物的属性及事物间的语义联系，是一种结构化的知识表示方法。
- 联想性 本来是作为人类联想记忆模型提出来的，它着重强调事物间的语义联系，体现了人类的联想思维过程。
- 自索引性 把各接点之间的联系以明确、简洁的方式表示出来，通过与某一结点连结的弧可以很容易的找出与该结点有关的信息，而不必查找整个知识库。能有效的避免搜索时所遇到的组合爆炸问题。
- 自然性 带有标识的有向图，可比较直观地把知识表示出来，符合人们表达事物关系的习惯，并且同自然语言的语义网络的转换也比较容易实现。

## 2.4.6 语义网络表示法的特征

- 主要缺点：

- 非严格性 没有象谓词那样严格的形式表示体系，一个给定语义网络的含义完全依赖于处理程序对它所进行的解释，通过语义网络所实现的推理不能保证其正确性。
- 复杂性 语义网络表示知识的手段多种多样的，带来了灵活性，但同时也由于表示形式的不一致，使得它的处理增加了复杂性。



## CYC: LOGICAL REASONING WITH THE WORLD'S LARGEST KNOWLEDGE BASE

Left  
brain

Right  
brain

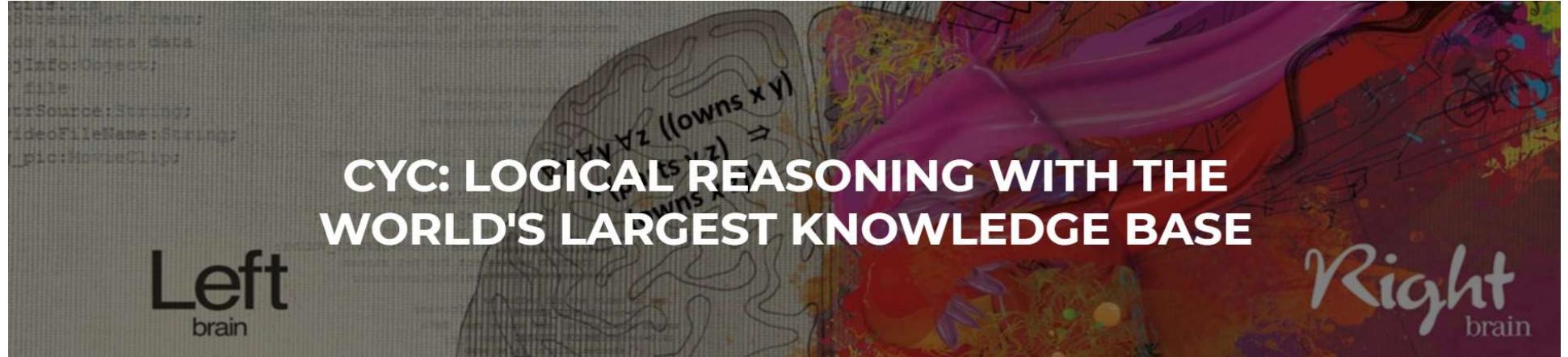
人们愚蠢的质问电脑为什么不能真正的思考。答案是我们还不能给他们编写出正确的程序。他们根本没有任何常识。仅有的一个大型项目在做这方面的工作，这就是著名的CYC项目。“

—— 马文·明斯基，麻省理工学院，2001年5月

Cyc知识库是一个经过形式化的海量的**人类基本知识库**。包括：事实，经验规则，日常生活对象与事件的启发式推理。**Cyc**致力把各个领域的**本体及常识知识**综合地集成在一起，并在此基础上实现类似于人类知识推理的人工智能项目。

包含：

- 320 万条人类定义的断言
- 涉及 30 万个概念
- 15000 个谓词



## CYC: LOGICAL REASONING WITH THE WORLD'S LARGEST KNOWLEDGE BASE

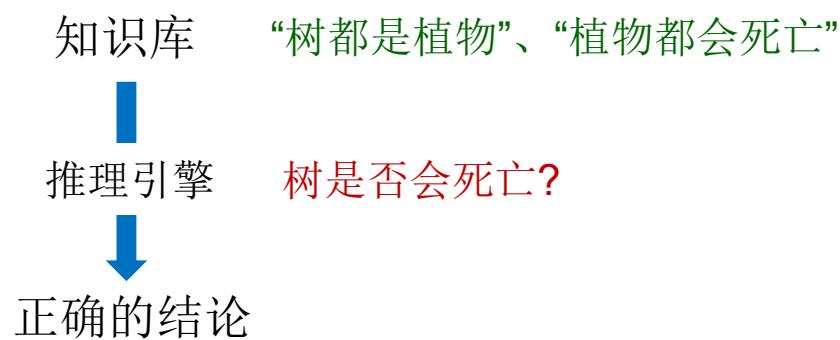
Cyc 类常识的知识库是实现人类水平智能的前提条件。

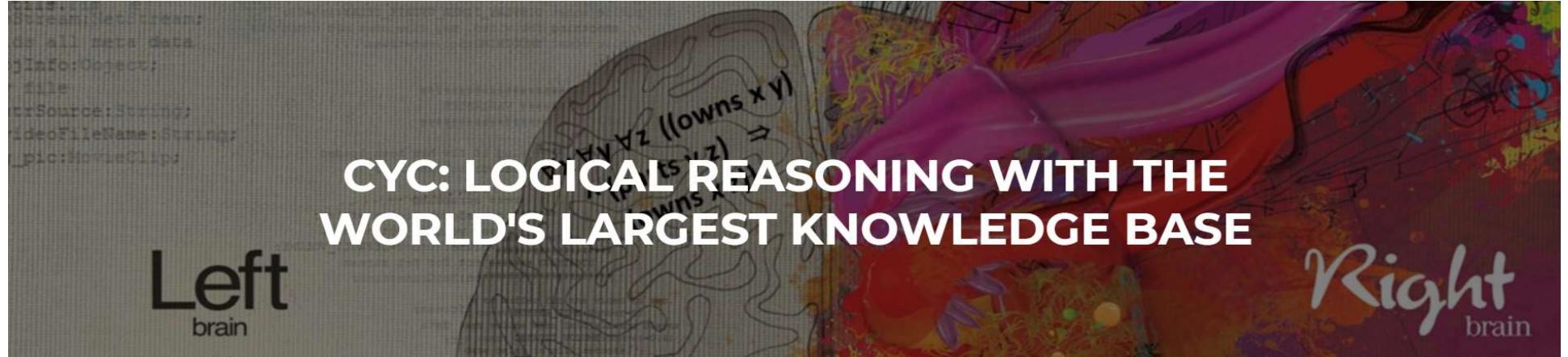
以下一对句子为例：

1. Fred saw the **plane flying** over Zurich.
2. Fred saw the **mountains flying** over Zurich.

传统的自然语言处理系统  
很难处理此类句法歧义。

由于 CYC 系统独有的自然常识辨别能力，系统会知道飞机可以飞行，而山脉不可以飞翔，所以第二句中所说的多半是彼得在飞翔了。





## □ CYC 表述语言(CYCL)

用来表述 CYC 知识库的一种形式化语言,它的语法来源于一阶谓词运算(形式逻辑语言)。为了表达现实世界中各种各样的专业知识。

### ◆ Cyc 中的概念被称为“常量(constants)”，以 “#” 开头

个体，即*individuals*: 例如 #\$BillClinton

集合，即*Collections*: 例如 #\$Tree-ThePlant (包含所有的树)

真值函数，即*Truth Functions*: 应用于一个或多个概念，并返回真或假

函数，即*Functions*: 用于以现有术语为基础产生新的术语。以 “ Fn” 结尾

#\$isa : 表示某个对象是某个集合的个体 (instance)

#\$genls: 表示某个集合是另外一个集合的子集合。

(#\$isa #\$BillClinton #\$UnitedStatesPresident), 表示“ Bill Clinton 属于美国总统集合”

(#\$genls #\$Tree-ThePlant #\$Plant), 表示“所有的树都是植物”

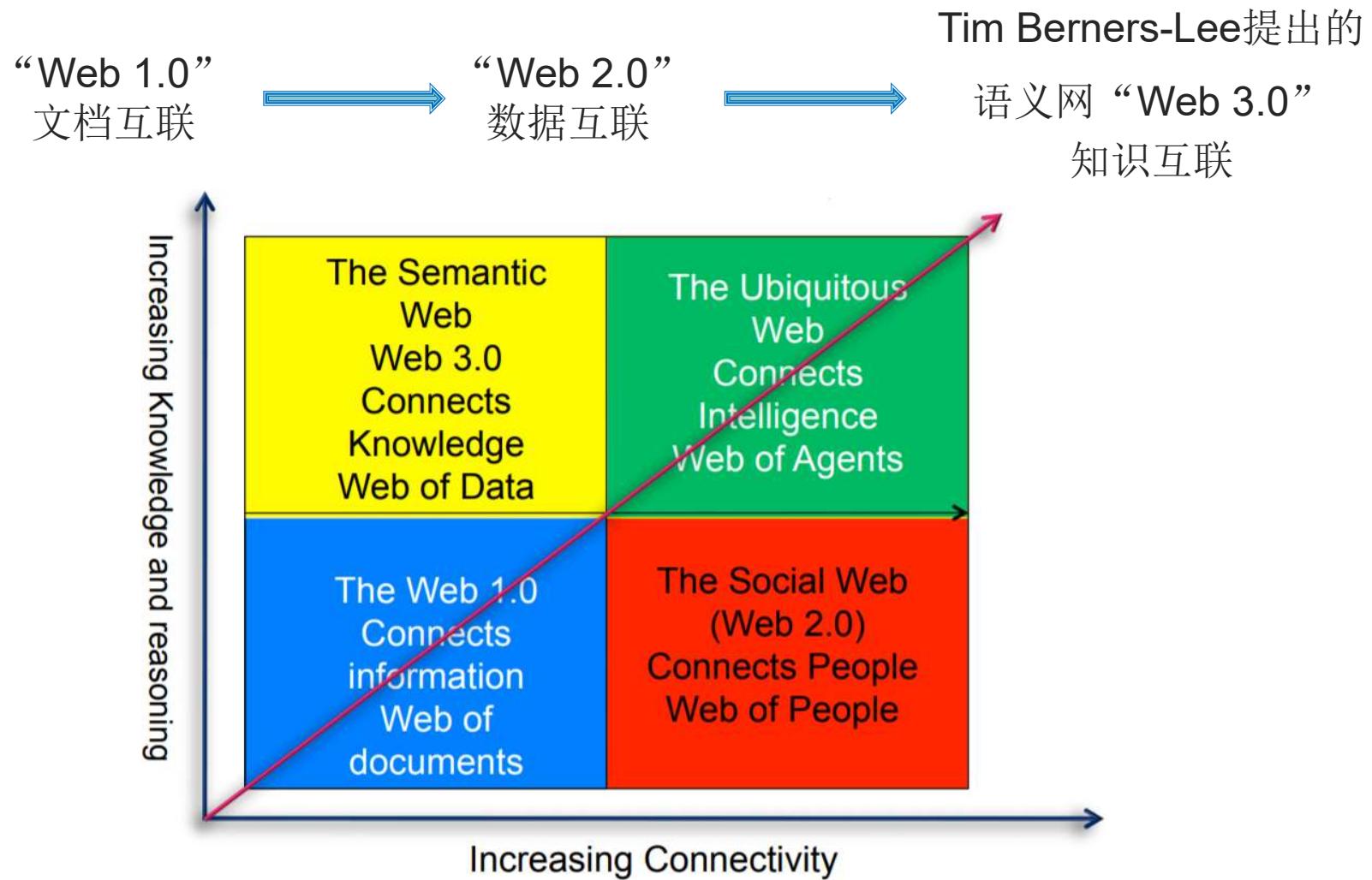
## Semantic Networks VS. Semantic Web

- 知识表达形式的语义网络 VS. 互联网中的语义网络

互联网之父Tim Berners-Lee在1998年提出，与Web 2.0, Web3.0概念相关

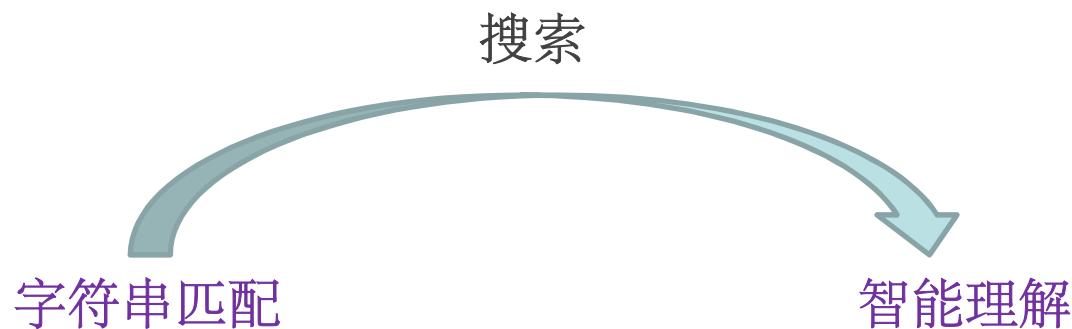
- 现有的Web向结构化数据的演进
- 近几年，语义网络分成学院派和实践派
- 学院派逐渐向知识表述和推理的方向突进
- 而实践派仍然致力于Web内容的结构化上，将自由文本内容结构化，这样计算机程序就能自动处理Web内容，而不是仅仅给人阅读

## □ Web逐步向语义网演变



将来的网络会像人一样认知、学习以及推理

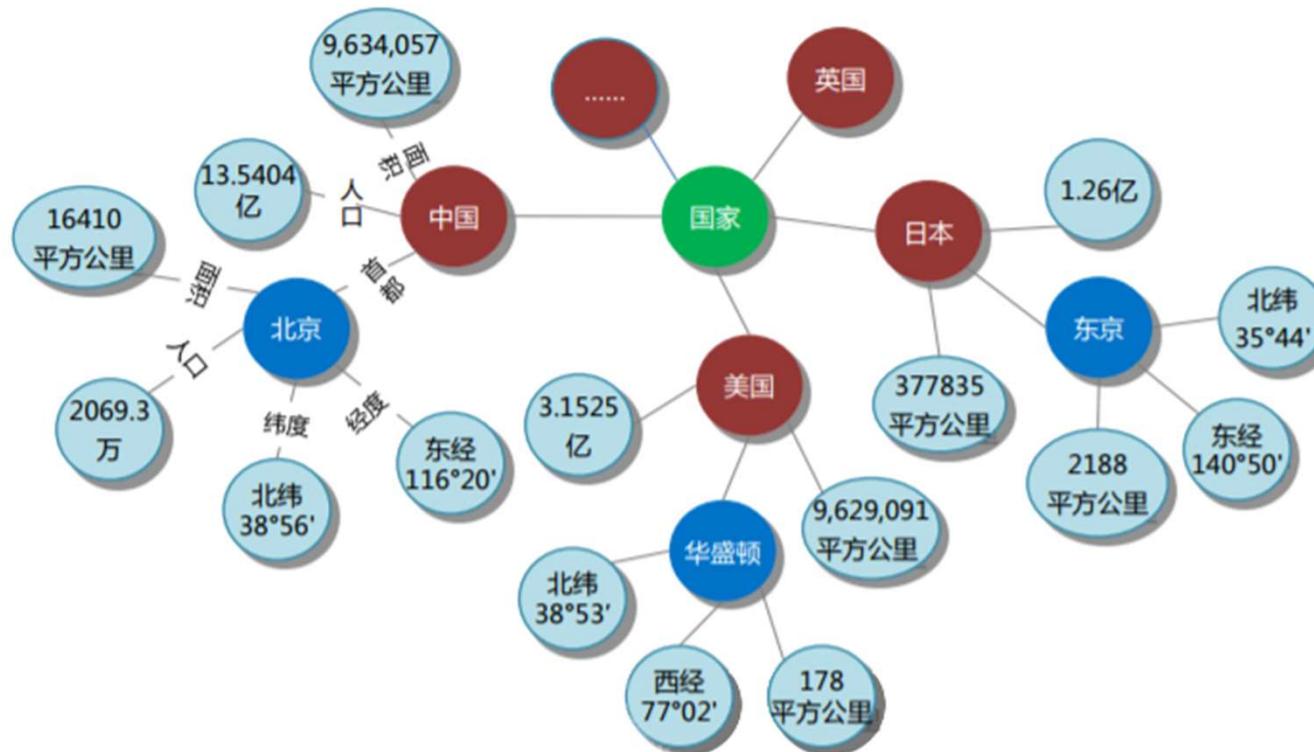
## □ 知识图谱: 智能语义检索的基础和桥梁



Wikipeia: **The Knowledge Graph** is a knowledge base used by Google to enhance its search engine's search results with semantic-search information gathered from a wide variety of sources.

Google知识图谱，是Google的一个知识库，其使用语义检索从多种来源收集信息，以提高搜索的质量。

知识图谱亦可被看作是一张巨大的图，**节点**表示**实体或概念**，**边**则由**属性或关系**构成



本质上，知识图谱是一种揭示实体之间关系的语义网络，对现实世界的事物及其相互关系进行形式化地描述。

## □ 研究价值：将零散的信息变为知识

在 Web 网页之上建立概念间的链接关系，将互联网中积累的信息组织起来，成为可以被利用的知识。

## □ 应用价值：改变现有的信息检索方式

1. 通过推理，实现概念检索（搜索请求不再局限于简单的关键词匹配）
2. 图形化方式展示结构化知识，从人工过滤网页寻找答案的模式中解脱出来

刘德华的年龄 - Google 搜索

https://www.google.com.hk/webhp?sourceid=chrome-instant&rlz=1C1CHWL\_zh-CN644CN645&ion=1&espv=2&ie=UTF-8#q=%E5%88%A9%E6%9D%83%EF%BC%88%EF%BC%89%EF%BC%88%EF%BC%89

刘德华的年龄 用户搜索

By:Eastmount CSDN

刘德华 / 年龄

54岁  
1961年9月27日

准确答案

周润发 60岁  
梁朝伟 53岁  
周星驰 53岁

刘德华 知识卡片

刘德华，MH, JP；香港著名演员兼歌手，1990年代获封香港乐坛“四大天王”之一，也是吉尼斯世界纪录大全中获奖最多的香港歌手；电影方面他获得三次香港电影金像奖最佳男主角和两次金马奖最佳男主角，至今参演超过140部电影。刘德华是天幕公司和映艺集团的创建者，作为投资人已参与制作了20多部华语电影。[维基百科](#)

生于：1961年9月27日（54岁），大埔  
配偶：朱丽倩（结婚时间：2008年）  
专辑：忘情水，Shi Ting Zhi Wang，来生缘，天意，谢谢你的爱，等等  
所获奖项：香港电影金像奖最佳男主角，香港电影金像奖最佳男配角，香港电影金像奖最佳电影，香港电影金像奖最佳原创电影歌曲  
兄弟姐妹：刘德华

刘德华年龄多大？ - 爱问知识人

iask.sina.com.cn, 娱乐, 明星, 华人明星

姓名: [刘德华](#) 族名: 刘福荣 花名: 华仔 华弟 华英雄 荣仔 英文名: Andy Lau 血型: ab型 身高: 1.74cm 体重: 64kg 出生年月日: 1961-9-27 出生地: 香港新界大埔泰亨村 爵贯: 广东新

郭富城,黎明,刘德华的真实年龄多大?\_华人明星\_音乐\_娱乐\_...  
wendu.tianya.cn/question/5007ee93e4281260  
2009年1月23日 - 张学友 生日: 1961年07月10日 刘德华 生日: 1961

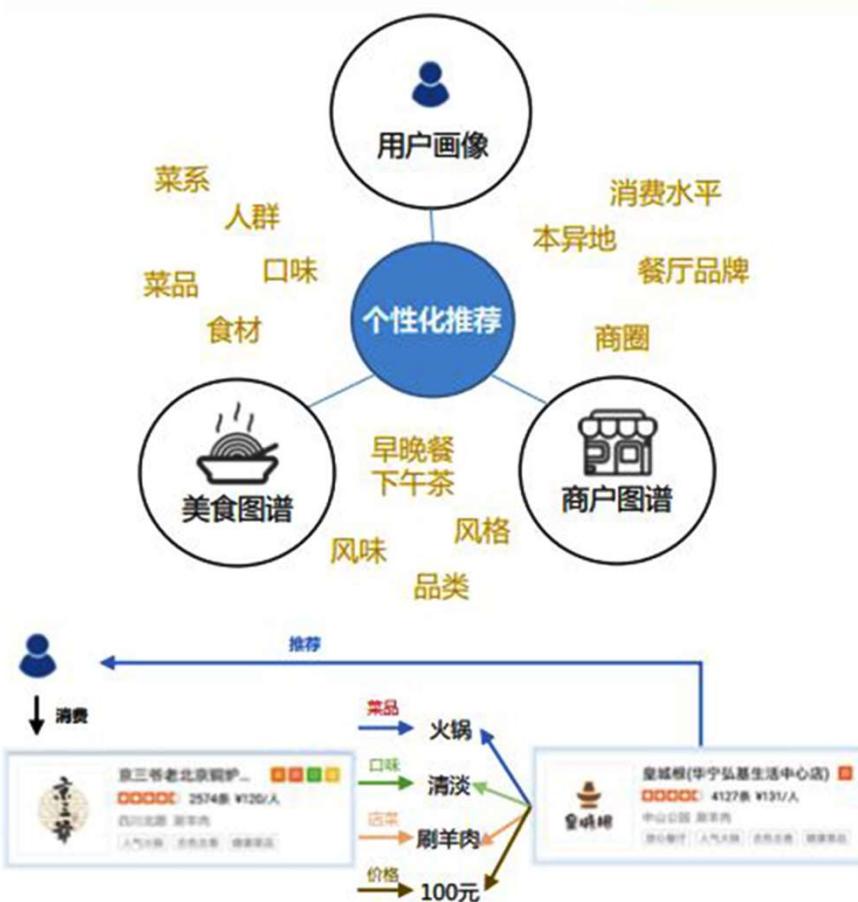
网页文档列表

电影

风再起时  
无间道  
头文字D  
新少林寺  
桃姐

# 口知识图谱应用:美图大脑

## 美团大脑在实际业务中的应用



## 国际上流行的知识库

名称	规模	
<b>Yago</b>	1千万实体，35万类别， 1.8亿事实，100种属性， 100语言	
<b>Dbpedia</b>	4千万实体，250类别， 5亿事实，6000种属性，	
<b>Freebase</b>	2千5百万实体，2000主题， 1亿事实，4000种属性	
<b>谷歌知识图谱</b>	5亿实体名字， 35亿条事实	
<b>NELL</b>	3百万实体名字，300类别 500属性，100万事实 1千5百万学习规则	

除谷歌外，其他互联网公司推出知识库产品以改进信息服务质量，如：  
百度知心、搜狗知立方、微软Bing Satori等

## □ 知识图谱的组织形式

- 组织形式

- 节点代表实体（人名、地名、概念等）
- 连边代表关系

- 事实可用三元组进行表示

- 表示方式

$$G = (E, R, S)$$

$E$ : 实体集合

$R$ : 关系集合

$S \subseteq E \times R \times E$  三元组集合

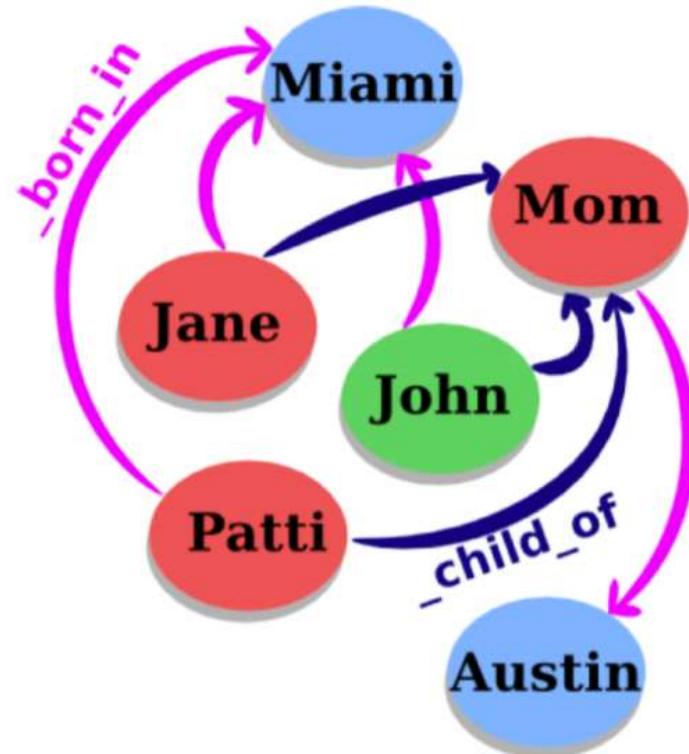
- 基本形式

(实体1, 关系, 实体2)

如: (乔布斯, 创始人, 苹果公司)

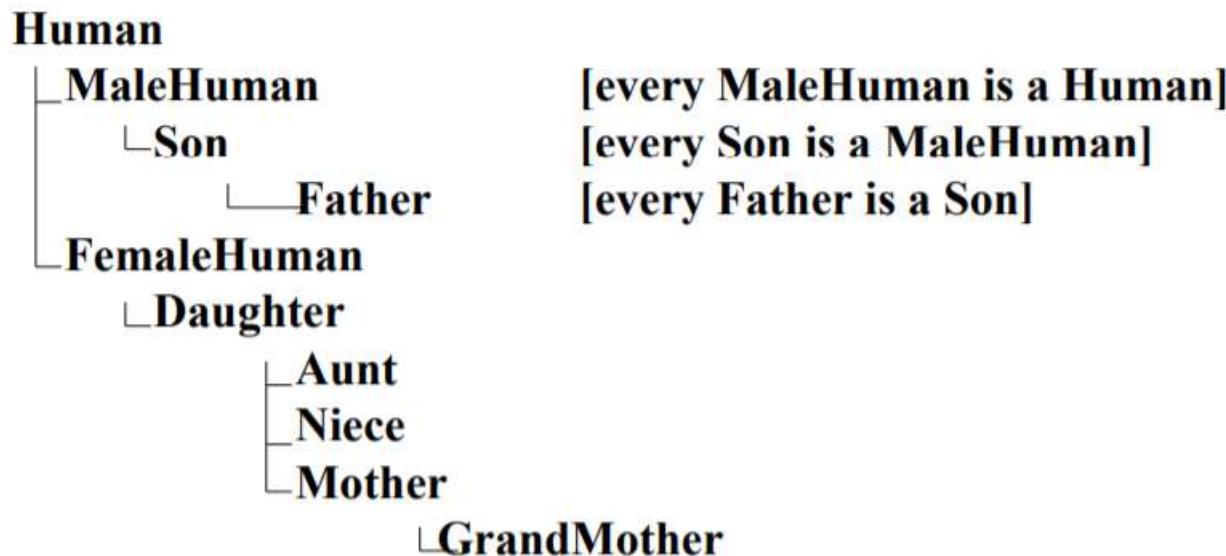
(概念、属性、属性值)

如: (王俊凯, 生日, 1999-09-21)



# 口本体(Ontology)

“**本体**”术语来自于哲学，研究世界上的各种实体以及他们是如何关联的科学。



对于 web，本体则关于对 web 信息及 web 信息之间的关系的精确描述。

# 口本体描述语言（Ontology Language）

- ◎ RDF (Resource Description Framework)

指定数据之间的关系

- ◎ RDFS (Resource Description Framework Schema)

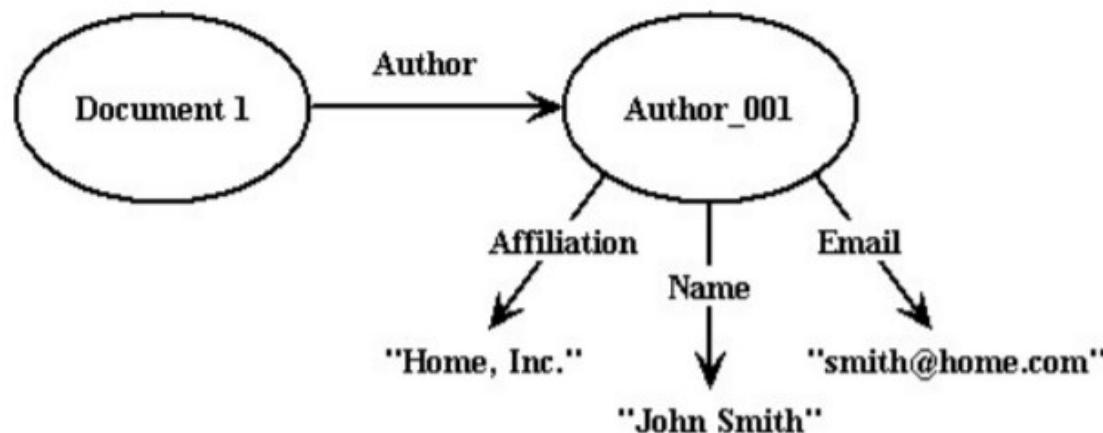
指定模式之间的关系

- ◎ OWL (Web Ontology Language)

基于描述逻辑，可指定模式间更复杂关系

## □ RDF 灵活的语义 Web 数据模型

用来描述网络资源的 W3C 标准。比如网页的标题、作者、修改日期、内容以及版权信息。



### 资源 Resource

- 使用URI唯一标示一个资源
- 一个资源通常表示一个事物(Thing)

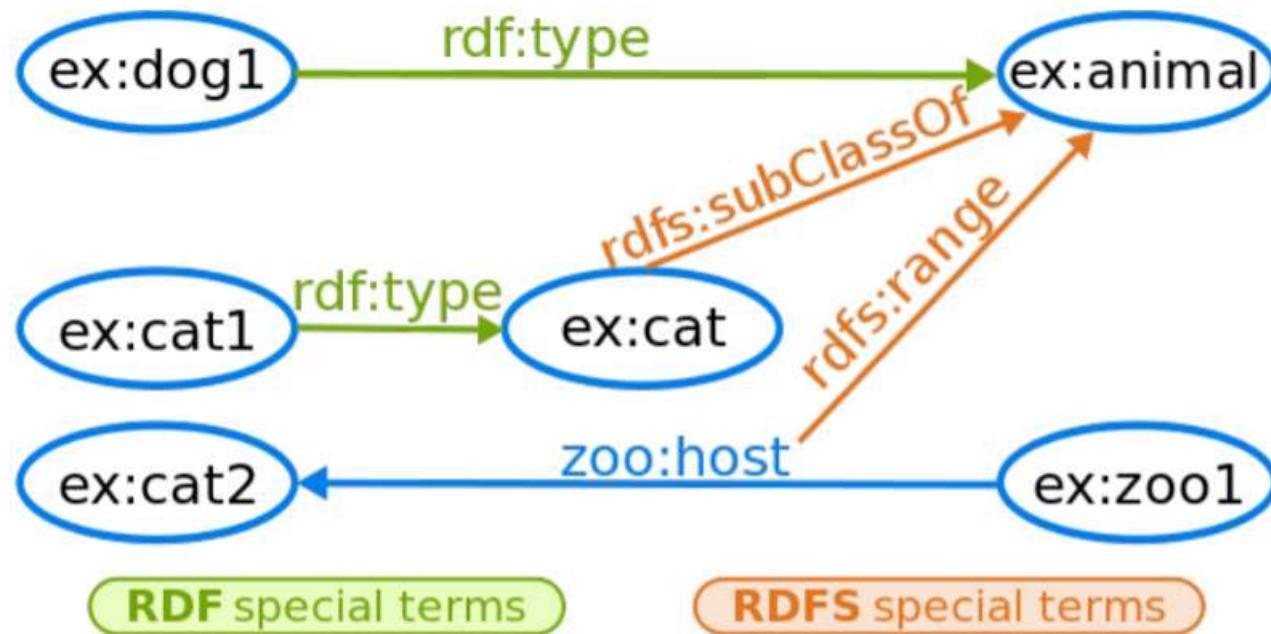
### 属性 Property

- 一种特殊类型的资源，用以描述资源与资源间的关系

### 语句 Statement

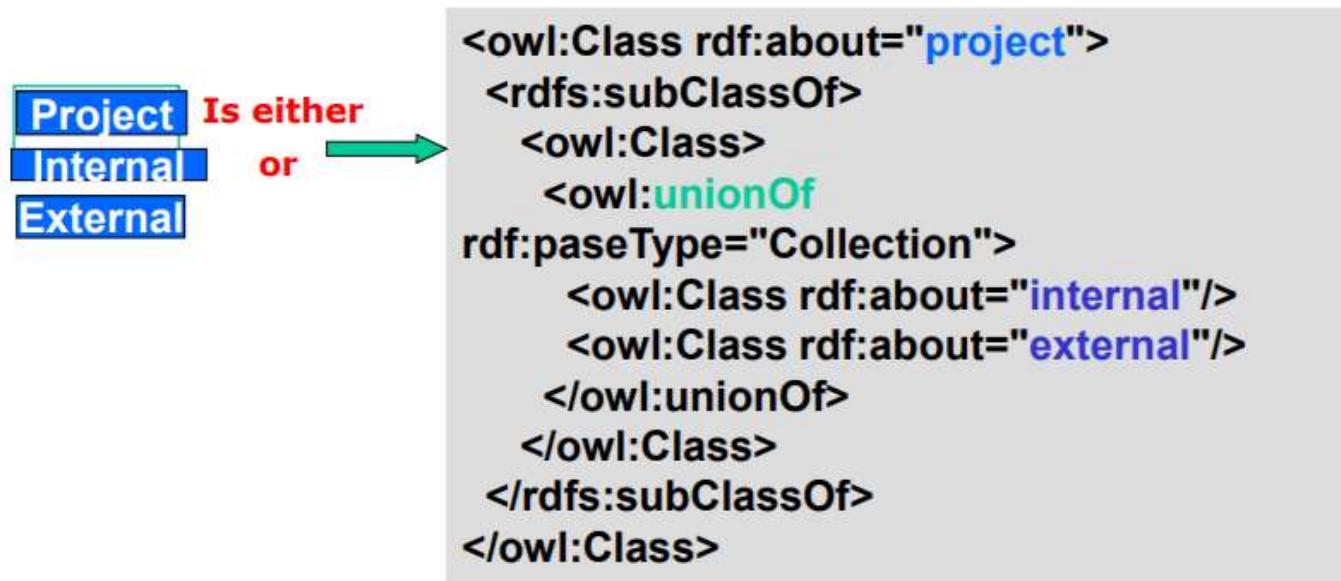
- 由3种资源组成的三元组(Triple)
- 主语rdf:subject，谓语rdf:predicate以及宾语rdf:object

## □ RDFS 提供简单的推理功能



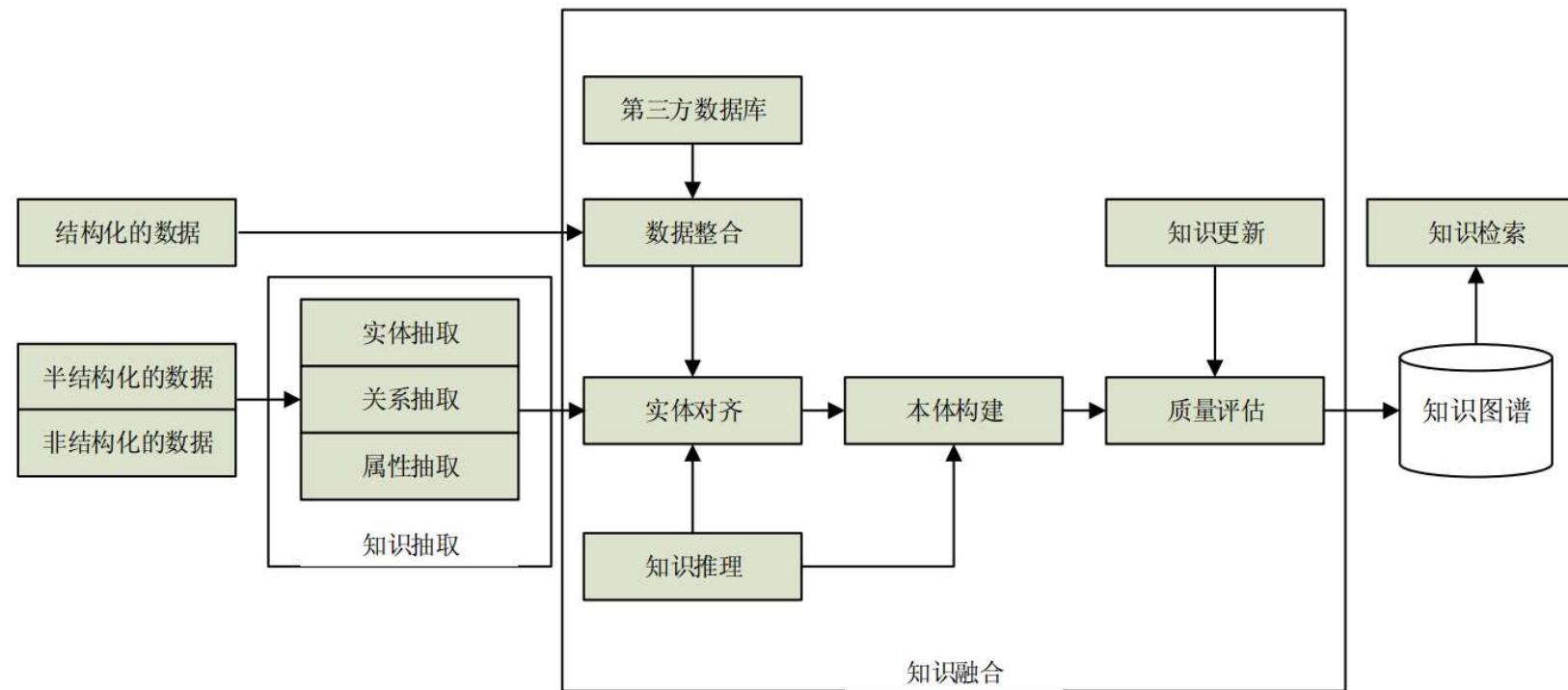
RDF只提供了概念和关系的基本描述能力，并没有推理的能力。  
RDFS则提供了简单的推理“schema”

## □ OWL 添加更多用于描述属性和类的词汇



OWL 允许更多的表达知识，比 XML、RDF 和 RDF-S 有更多的表达手段，因此在 Web 上表达机器可理解内容的能力更强

## 口知识图谱的构建过程



自底向上法：从一些开放链接数据中提取出实体，将置信度较高的加入到知识库，再构建顶层的本体模式

# □ 知识构建过程

## □ 知识抽取

面向开放的链接数据，构成三元组，形成高质量的事实表达

## □ 知识表示

三元组的表示方法

## □ 知识融合

解决知识良莠不齐、重复以及关联不明确等问题

## □ 知识推理

在已有的知识库基础上进一步挖掘隐含的知识

## □ 知识抽取

抽取实体、关系以及属性3个知识要素

- 实体抽取：从原始语料中自动识别出命名实体
  - ✓ 基于规则与词典的方法
  - ✓ 基于统计机器学习的方法
  - ✓ 面向开放域的抽取方法
- 关系抽取：识别实体关系，解决实体间语义链接的问题
  - ✓ 开放式实体关系抽取
  - ✓ 基于联合推理的实体关系抽取
- 属性抽取：针对实体而言的，通过属性可形成对实体的完整勾画
  - ✓ 基于规则与启发式算法方法
  - ✓ 从开放域数据集上抽取属性

## 口知识表示

三元组的表示形式，在计算效率、数据稀疏性上存在诸多问题。

### ➤ 以深度学习为代表的表示学习技术

可以将实体的语义信息表示为稠密低维实值向量，计算实体、关系间的复杂语义关联。

优势：

- ✓ 语义相似度计算。刻画实体之间的语义关联程度
- ✓ 链接预测。预测图谱中任意两个实体之间的关系
- ✓ 映射至低维空间。计算更加高效

### ➤ 代表模型：

距离模型、双线性模型、神经张量模型、矩阵分解模型、翻译模型等

## □ 知识融合

来自不同知识源的知识在同一框架规范下进行异构数据整合、消歧、加工、推理验证、更新

➤ 实体对齐：消除异构数据中实体冲突、指向不明

- 1) 将待对齐数据进行分区索引，以降低计算的复杂度；
- 2) 利用相似度函数或相似性算法查找匹配实例；
- 3) 使用实体对齐算法进行实例融合

➤ 知识加工：本体构建+质量评估

在层次上形成一个大规模的知识体系，统一对知识进行管理。

➤ 知识更新：本体更新+数据更新

需要随时间不断地迭代更新，扩展现有的知识，增加新的知识。

## □ 知识推理

在已有的知识库基础上进一步挖掘隐含的知识，从而丰富、扩展知识库。

例如：

(乾隆, 父亲, 雍正) + (雍正, 父亲, 康熙) → (乾隆, 祖父, 康熙)

### ➤ 基于逻辑的推理：

- ✓ 一阶谓词逻辑 (First order logic)
- ✓ 描述逻辑 (Description logic)

### ➤ 基于图的推理

利用关系路径中的蕴涵信息，通过图中两个实体间的多步路径来预测它们之间的语义关系。

一些较为复杂的推理规则往往是手动总结。对于推理规则的挖掘，主要还是依赖于实体以及关系间的丰富同现情况。

# 第2章 知识表示

- 2.1 知识表示与知识表示的概念
- 2.2 一阶谓词逻辑表示法
- 2.3 产生式表示法
- 2.4 语义网络表示法
- 2.5 框架表示法

## 2.5 框架表示法

在框架理论的基础上发展起来的一种结构化知识表示方法。

- 2.5.1 框架理论
- 2.5.2 框架和实例框架
- 2.5.3 框架系统
- 2.5.4 框架系统的问题求解过程
- 2.5.5 框架表示法的特征

## 2.5.1 框架理论

- 明斯基(Minsky)于1975年,在论文*A Framework for representing knowledge*中提出

- 基于心理学研究:

人们对现实世界事物的认识都是以一种类似于框架的结构存储在记忆中的，当遇到一个新事物时，就从记忆中找出一个合适的框架，并根据新的情况对其细节加以修改、补充，形成对这个新事物的认识。

- 是一种基于心理学的模型:

作为理解视觉、自然语言对话及其它复杂行为的一种基础。

## 2.5.1 框架理论

### □概述

- 框架: 框架是描述对象(事物, 事件或概念)属性的一种数据结构。
- 框架系统: 一组有关的框架通过属性之间的关系建立起联系, 形成一个框架系统。

### □基本概念

**槽:**描述对象的某一方面的属性

**侧面:**描述相应属性的一个方面

槽和侧面具有的属性值分别称为槽值和侧面值.

## 2.5.2 框架结构和框架表示

### 框架的基本结构

<框架名>

槽名1:

侧面名11:侧面值11

侧面名12:侧面值12

.....

侧面名1n:侧面值1n

槽名k:

侧面名k1:侧面值k1

.....

一个框架由若干个“槽”组成,槽又可根据实际需要分为若干个“侧面”

槽值和侧面值,既可以是数字、字符串、布尔值,也可以是给定的操作

- 例子 描述“教师”的框架

框架名: <教师>

工 作 : 范围: (科研、教学)

缺省: 教学

类 属 : <知识分子>

性 别 : (男、女)

学 历 : (本科、研究生)

类 别 : (<小学教师>、<中学教师>、<大学教师>)

在教师这个框架中，含有 4个槽。槽名分别是“类属”、“性别”、“学历”和“类别”。这些槽名后面就是其槽值，而槽值“<知识分子>”又是一个框架名。“范围”、“缺省”是槽“工作”的两个不同的侧面，其后是侧面值。

- 例子 描述“大学教师”的框架

框架名：<大学教师>

类 属：<教师>

学 位：范围：（学士，硕士，博士）

缺省：硕士

专 业：<学科专业>

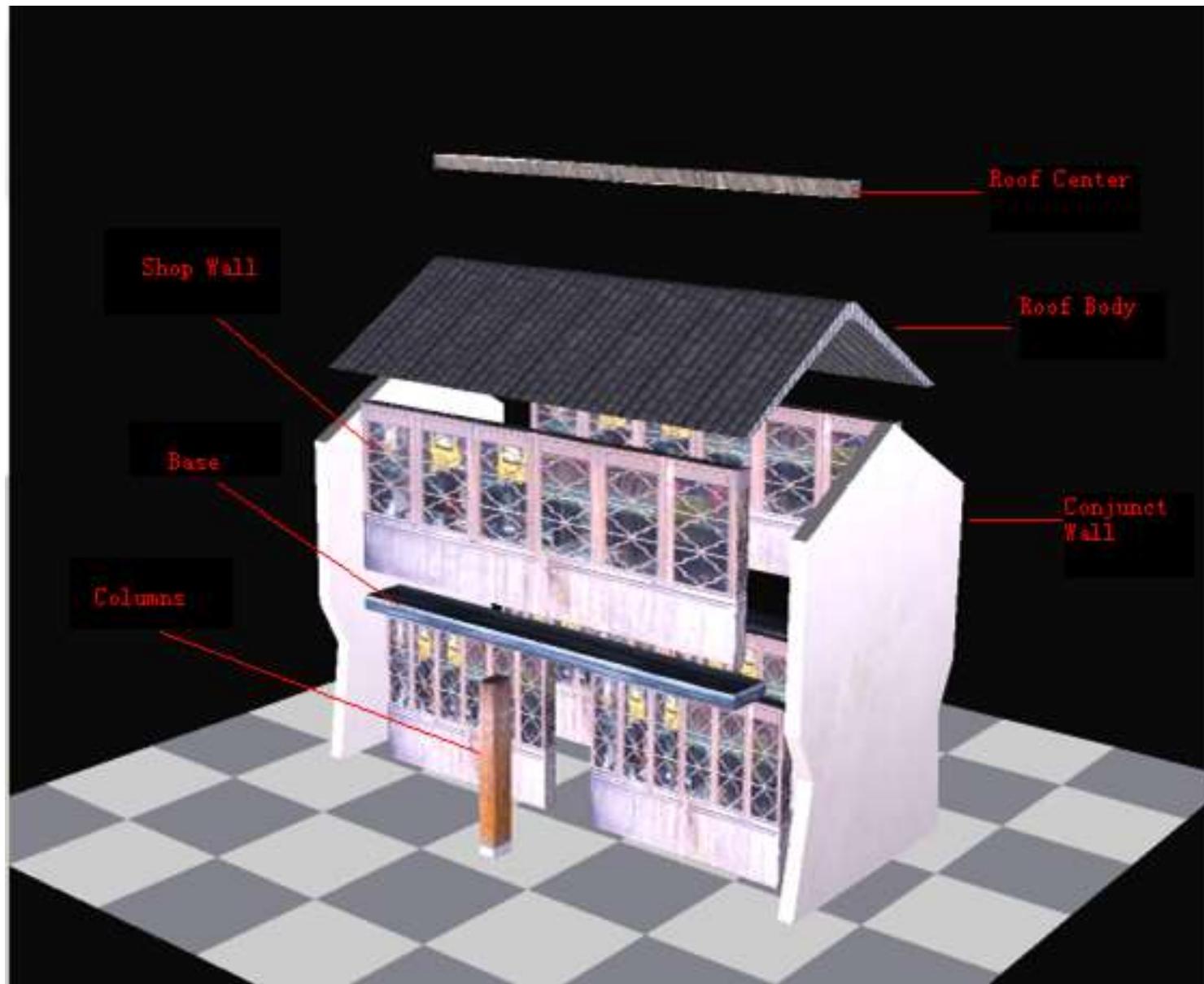
职 称：范围：（助教，讲师，副教授，教授）

缺省：讲师

水 平：范围：（优，良，中，差）

缺省：良

“教师”和“大学教师”框架之间存在层次关系，称前者为上层框架（或父框架），后者为下层框架（或子框架）。下层框架可从上层框架继承某些属性或值，减少信息冗余，及不必要的重复存储。





## 2.5.2 框架结构和框架表示

### 框架的基本结构

- 例：一个直接描述硕士生有关情况的框架
- Frame <MASTER>
- Name: Unit (Last-name, First-name)
- Sex: Area (male, female)
- Default: male
- Age: Unit (Years)
- Major: Unit (Major)
- Field: Unit (Field)
- Advisor: Unit (Last-name, First-name)
- Project : Area (National, Provincial, Other)
  - Default: National
- Paper: Area (SCI, EI, Core, General)
  - Default: Core
- Address: < S-Address>
- Telephone: Home Unit (Number)
  - Mobile Unit (Number)

## 2.5.2 框架结构和框架表示

### 框架表示

- 学生框架
- Frame <Student>
- Name: Unit (Last-name, First-name)
- Sex: Area (male, female)
- Default: male
- Age: Unit (Years)
- If-Needed: Ask-Age
- Address: < S-Address>
- Telephone: Home Unit (Number)  
Mobile Unit (Number)
- If-Needed: Ask-Telephone

## 2.5.2 框架结构和框架表示

### 框架表示

- 硕士生框架
- Frame <Master>
- AKO: Student
- Major: Unit (Major)
  - If-Needed: Ask- Major
  - If-Added: Check-Major
- Field: Unit (Direction-Name)
  - If- Needed: Ask - Field
- Advisor: Unit (Last-name, First-name)
  - If- Needed: Ask -Visor
- Project : Area (National, Provincial, Other)
  - Default: National
- Paper: Area (SCI, EI, Core, General)
  - Default: Core
- 在Master框架中，用到了一个系统预定义槽名AKO。

所谓系统预定义槽名，是指框架表示法中事先定义好的可公用的一些标准槽名。

## 2.5.2 框架结构和框架表示

### 框架表示

#### □ 系统预定义槽名

通常在框架系统中定义一些公用的、标准的槽名，来表示对象间关系，常见的槽名有:ISA,AKO和Instance等

##### ➤ ISA槽

其直观意义是“是一个”，“是一只”。指出对象间的类属关系ISA槽指出的联系都具有继承性。指当下层框架中的某些槽值或侧面值没有被直接给定时，可以从其上层框架中继承这些值或属性。

##### ➤ AKO槽

其直观意义是“是一种”，明确地指出了该下层框架所描述的事物是其上层框架所描述事物中的一种，下层框架可继承上层框架中值或属性。

##### ➤ Instance槽

用来表示AKO槽的逆关系。当作为某上层框架的槽时，可在该槽中指出它所联系的下层框架，具有继承性。

##### ➤ Part-of槽

用于指出部分和全体的关系。例如，“两条腿”是“人体”的一部分。可以定义“两条腿”为下层框架，“人体”为其上层框架。在“两条腿”的框架中设置一个Part-of槽，槽值填入<人体>这个框架名。

## 2.5.2 框架结构和框架表示

### 框架表示

- 框架的继承技术，通常由框架中设置的3个侧面：**Default**、**If-Needed**、**If-Added**所提供的缺省推理功能来组合实现。
- **Default**: 该侧面的作用是为相应槽提供缺省值。当其所在槽没有填入槽值时，系统就以此侧面值作为该槽的默认值。例如，**Paper**槽的默认值为**Core**。
- **If- Needed**: 该侧面的作用是提供一个为相应槽赋值的过程。当某个槽不能提供统一的缺省值时，可在该槽增加一个**If-Needed**侧面，系统通过调用该侧面提供的过程，产生相应的属性值。例如，**Age**槽、**Telephone**槽等。
- **If-Added**: 该侧面的作用是提供一个因相应槽值变化而引起的后继处理过程。当某个槽的槽值变化会影响到一些相关槽时，需要在该槽增加一个**If-Added**侧面，系统通过调用该侧面提供的过程去完成对其相关槽的后继处理。例如，**Major**槽，由于专业的变化，可能会引起**Field**和**Advisor**的变化，因此需要调用**If-Added**侧面提供的**Check-Major**过程进行后继处理。

# 框架的程序语言实现

- FRL(Frame Representation Language)
- PROLOG

- 例:
- frame(name("教师"),
- kind\_of("<知识分子>"),
- work(scope("教学", "科研"), default("教学")),
- sex("男", "女"),
- reco\_of\_f\_s("中师", "高师"),
- type("<小学教师>", "<中学教师>", "<大学教师>").

- frame(name("教师"),
- body( [st("类属", [st("<知识分子>", []])],  
• [st("工作", [st("范围", [st("教学",  
[]), st("科研", [])], [])), st("缺省", [st("教学", [])], [])],  
• [st("性别", [st("男", []), st("女", [])]),  
• [st("学历", [st("中师", []), st("高师", [])]),  
• [st("类型", [st("<小学教师>", []), st("<中学教师>", []), st("<大学教师>", [])]]]))

## 2.5.5 框架表示法的特征

- 框架表示法的优点

结构性：善于表示结构性知识，它能够把知识的内部结构关系以及知识间的特殊联系表示出来。

深层性：可以从多个方面、多重属性表示知识，而且还可以通过ISA、AKO等槽以嵌套结构分层地对知识进行表示，因此能用来表达事物间复杂的深层联系。

继承性：下层框架可以继承上层框架的槽值，也可以进行补充和修改。减少知识冗余，还较好地保证了知识的一致性。

自然性：高度模拟了人脑对实体多方面、多层次的存储结构，直观自然，易于理解。

- 框架表示法的不足

缺乏过程性知识表示：不善于表示过程性知识，缺乏如何使用框架中知识的描述能力。因此常与产生式表示法结合。

# 剧本（脚本）表示法

- 描述特定场合下事件序列的框架型表示方式
- 1975年R.C. Schank依据他的概念依赖理论而提出
  - 在自然语言处理系统中受到广泛重视
  - 【难点】在人类的各种知识中，常识性知识是数量最多、涉及面最宽、关系最复杂的知识，很难把它们形式化地表示出来交给计算机处理。
  - 概念依赖理论的基本思想：把人类生活中各类故事情节的基本概念抽取出来，构成一组原子概念，确定这些原子概念间的相互依赖关系，然后把所有故事情节都用这组原子概念及其依赖关系表示出来。
- 脚本与框架类似，由一组槽组成，用来表示特定领域内一些事件的发生序列

# Schank对11种动作（ACT）的原子化

- (1) **PROPEL**:应用物理力量（推、拉、打等）于一对象。
- (2) **GRASP**: 一个演员抓起一个物理对象。
- (3) **MOVE**: 演员身体的一部分变换空间位置，如抬手等。
- (4) **PTRANS**: 物理对象变换位置，如走进、跑出等。
- (5) **ATRANS**:抽象关系的改变，如传递、赠送、革命等。
- (6) **ATTEND**:用某个感官获取信息，如用目光搜索等。
- (7) **INGEST**:演员把某个东西吸入体内，如吃、喝等。
- (8) **EXPTEL**:演员把某个东西送出体外，如呕吐、落泪等。
- (9) **SPEAK**:演员产生一种声音，如唱歌、尖叫等。
- (10) **MTRANS**:信息的传递，如读报、看信、看电视等。
- (11) **MBUILD**:由旧信息形成新信息，如“眉头一皱，计上心来”。

# 基于概念依赖关系的故事情节理解

- (1) Schank定义上述11中动作的主要意图：上述原子概念主要不是用于表示动作本身，而是表示动作的结果，并且是本质的结果，因此也可以认为是这些概念的推理。
- (2) 基于概念依赖关系的故事情节理解办法：事先编好许多剧本，每个剧本代表日常生活中发生的一种事件，它把这种事件的典型情节规范化，编成一种程式。
- (3) Schank实现了SAM( **Script Applier Mechanism**)系统。其功能是：接受一个故事后，首先做语法分解工作，按照概念依赖关系的模式化成内部表示，然后从库中取出相应的剧本进行匹配，根据事先确定的剧本情节来理解故事。

- 概念依赖理论的意义及影响
  - 完整的自然语言语义模型，具有广泛适用性
  - 为自然语言语义提供了一套完整的形式理论，大大减少了模棱两可的问题
  - 提供了捕捉自然语言的大部分语义的正规表示，简化了对其进行理解所需的推理
    - 相同或相似的概念依赖图表示同一件事，避免在不同结构的图上进行大量运算
- 缺陷
  - 无法肯定能写出将语句简化为上述正规表示的程序
  - 无明确证据表明人类是按任何类似的范式来存储知识的

# 剧本（脚本）及其实例

- 1、剧本：描述特定范围内原型事件的结构。
- 2、剧本的组成
- (1) 进入条件：指出剧本所描述的事件可能发生的先决条件，即事件发生的前提条件。
- (2) 角色：描述事件中可能出现的人物。
- (3) 道具：描述事件中可能出现的有关物体。
- (4) 场景：描述事件序列，可以有多个场景。
- (5) 结局：给出剧本所描述的事件发生以后必须满足的条件。

# Schank餐馆剧本

- **剧本:** 就餐
- **进入条件:** 顾客饿了，品尝佳肴，招待亲友，顾客有钱
- **角色:** 顾客、服务员、厨师、经理、老板
- **道具:** 食品、桌子、菜单、钱
- **场景:**
  - **第一幕: 进入餐馆**
    - **PTRANS:** 步入餐馆
    - **ATTEND:** 用目光寻找空桌
    - **MBUILD:** 选定桌子
    - **MOVE:** 坐下

## 第二幕：定菜

ATRANS: 服务员送菜单

MTRANS: 读菜单

MBUILD: 选定所要的菜

MTRANS: 告诉服务员

ATRANS: 付钱

## 第三幕：吃饭

ATRANS: 服务员上菜

INGEST: 吃饭

## 第四幕：离开

MOVE: 站起身来

PTRANS: 步出餐馆

结局：顾客吃了饭；顾客花了钱；老板挣了钱；餐厅食品减少了。

由上述剧本实例可知，剧本就像电影剧本一样，一场一场比赛地表示一些特定事件的序列。

利用上述剧本可理解如下“老王吃烤鸭”故事：

“老王来到全聚德烤鸭店。他冲进去抢到了一个位置，服务员拿来菜单。老王要了两只烤鸭，八个菜。菜很快上齐了。老王又吃又喝，两个小时后他醉醺醺地离开了烤鸭店。”

利用Schank餐馆剧本可对上述故事进行提问和回答。详细过程请参见陆汝钤《人工智能》(上册) pp164-165。

# 小结

- 知识和知识表示的基本概念
  - 符号主义的观点
- 主要的知识表示方法
  - 一阶谓词逻辑表示
  - 产生式表示
  - 语义网络表示
  - 框架表示