



第4章 知识表示方法

4.1 知识表示概念及特征

4.2 知识表示常见方法

4.3 知识图谱概念及应用

4.4 资源描述框架RDF



4.1 知识表示概念及特征

- 知识的定义 https://blog.csdn.net/qq_44046091/article/details/108943934
 - Feigenbaum: 知识是经过削减、塑造、解释和转换的信息。简单地说，知识是经过加工的信息。
 - Bernstein: 知识是特定领域的描述、关系和过程组成。
 - Hayes-Roth: 知识是事实、信念和启发式规则。
 - 信息关联后所形成的信息结构：事实&规则
 - 知识是经过加工的信息，包括事实、信念和规则。事实是关于对象和物体的知识。规则是有关问题中与事务的行动、动作相联系的因果关系的知识。
- 相关概念
 - 信息数据所表达的客观事实；
 - 数据是信息的载体，是事务、概念或指令的形式化表示形式，与具体的介质和编码方法有关，便于人工或自然方式进行通信、解释或处理。



4.1 知识表示概念及特征

■ 知识的类型

- 按性质：概念、命题、公理、定理、规则和方法
- 按作用域：①常识性知识②领域性知识
- 按作用效果：①事实性知识②过程性(规则)知识③控制性知识
- 按层次：①表层知识（专家系统）②深层知识（数据挖掘）
- 按确定性：①确定性知识②不确定性知识
- 按等级：①零级知识（叙述性知识）②一级知识（过程性知识）③二级知识（元知识、超知识）④三级知识（元元知识）

4.1 知识表示概念及特征

■ 人工智能系统所关心的知识（按作用效果划分）

事实知识

- 有关问题环境的一些事物的知识，包括事物的分类、属性、事物间关系、科学事实、客观事实等
- 在知识库中属于低层的知识。
- 常以“...是...”的形式出现。
- 例：雪是白色的、鸟有翅膀、张三李四是好朋友

规则知识

- 有关问题中与事物的行动、动作相联系的因果关系知识
- 是动态的、变化的，属于专家提供的专门经验知识，这种知识虽无严格解释但很有用处
- 常以“如果...那么...”形式出现

控制知识

- 有关问题的求解步骤，技巧性知识，告诉怎么做一件事。
- 包括有关部门各种处理过程、策略和结构的知识

元知识

- 有关知识的知识，是知识库中的高层知识
- 包括怎样使用规则、解释规则、校验规则、解释程序结构等知识



4.1 知识表示概念及特征

■ 知识表示定义

- 是对知识的描述，即用一组符号把知识编码成计算机可以接受的某种结构

■ 人工智能对知识表示的要求

- ①表示能力：正确、有效地将各类知识表示出来
- ②可理解性：易懂、易读
- ③便于知识的获取：智能系统能够渐进地增加知识，逐步进化
- ④便于搜索：支持对知识库的高效搜索，使智能系统高效感知事务之间的关系和变化
- ⑤便于推理：能够从已有的知识中推出需要的答案和结论



第4章 知识表示方法

4.1 知识表示概念及特征

4.2 知识表示常见方法

4.2.1 一阶谓词表示

4.2.2 产生式表示

4.2.3 语义网络表示

4.2.4 框架表示

4.2.5 应用示例

4.3 知识图谱概念及应用

4.4 资源描述框架

4.2.1 一阶谓词表示

一、命题

■ 命题含义：

- 命题就是具有真假意义的陈述句
- 命题都应具有确定的真值

命题

- 今天下雨(T)
- 雪是黑的(F)

非命题

- 把门关上!
- 你到哪里去?

■ 命题类型

- 原子命题：不能分解成更简单的陈述句的命题
- 复合命题：由连接词、标点符号和原子命题等复合构成的命题

我是一位学生和他是一位工人。

■ 命题逻辑：

- 研究命题和命题之间关系的符号逻辑系统
- 通常用大写字母P、Q、R、S等表示

P：我是一位学生。

4.2.1 一阶谓词表示

- 命题常元：已知真假值的命题。
 - 可直接用{T、F} 表示。
- 命题变元：
 - 以真假为其变域之变元，或没有指定真值的命题，但它不是命题。
- 命题联结词
 - 否定词(\neg)
 - 合取词(\wedge)
 - 析取词(\vee)
 - 蕴含词(\rightarrow), 如果..., 则....
 - 等值词（双条件词）(\leftrightarrow)

P	Q	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1



4.2.1 一阶谓词表示

二、谓词

- 谓词：

- 对象的属性和对象之间的关系

- 谓词逻辑：

- 根据对象和对象上的谓词，使用连接词和量词来表示世界

- 谓词逻辑与命题逻辑区别：

- 命题逻辑中，每个表达式都是句子，表示事实；
- 谓词逻辑中，用句子、项表示对象

- 谓词符号：

- 若谓词符号 P 中包含的个体数目为 n ，称 P 为 n 元谓词符号

Father(x):一元谓词; Less(x,y):二元谓词



4.2.1 一阶谓词表示

■ 一阶谓词：

- 谓词 P 中的所有个体都是常量、变元或函数
- 否则， P 中某个个体本身又是一个一阶谓词，称二阶谓词
- 一阶逻辑是数理逻辑的基础部分

■ 例：

- 一阶谓词指刻画个体属性的谓词，如“红色”“大于”等谓词都只适用于个体概念
- 像“鲜艳”“传递性”等用来刻画“红色”“大于”这种谓词的谓词就是高阶谓词了，它们刻画的是属性的属性。



4.2.1 一阶谓词表示

■ 谓词公式（一阶谓词公式）

- 是命题公式的扩充和发展，其本质是命题公式，是把数学中的逻辑论证加以符号化

■ 谓词公式语法元素

- 常量符号
- 变量符号
- 函数符号
- 谓词符号：
 - 否定词(\neg)
 - 合取词(\wedge)
 - 析取词(\vee)
 - 蕴含词(\rightarrow), 如果..., 则....
 - 等值词（双条件词）(\leftrightarrow)

量词:

全称量词 $\forall X$

存在量词 $\exists X$

4.2.1 一阶谓词表示

■ 谓词逻辑表示知识

■ 事实性知识

张三是一名计算机系的学生，他喜欢编程。
 $\text{Computer}(\text{张三}) \wedge \text{Like}(\text{张三}, \text{Programming})$

■ 规则性知识

如果x，则y
 $x \rightarrow y$

■ 谓词公式表示知识的步骤

- 定义谓词及个体，确定每个谓词及个体的确切含义
- 变元赋值：根据要表达的事务或概念，为每个谓词变元赋值
- 连接词连接谓词，形成谓词公式：根据要表达的知识和语义



4.2.1 一阶谓词表示

- 例.用谓词逻辑表示下列知识:

- 武汉是一个美丽的城市，但它不是一个沿海城市。
- 如果马亮是男孩，张红是女孩，则马亮比张红长得高

- 解:

- 1.定义谓词:

BCity(x):x是一个美丽的城市

Hcity(x):x是一个沿海城市

Boy(x):x是男孩

Girl(x):x是女孩

High(x,y):x比y长得高

- 这里涉及的个体有武汉(wuhan),马亮(mal),张红(zhangh)



4.2.1 一阶谓词表示

2.将个体代入谓词:

BCity(wuhan), Hcity(wuhan), Boy(mal), Girl(zhangh),
High(mal, zhangh)

3.根据语义连接, 得到谓词公式:

$\text{BCity(wuhan)} \wedge \neg \text{Hcity(wuhan)}$

$\text{Boy(mal)} \wedge \text{Girl(zhangh)} \rightarrow \text{High(mal, zhangh)}$

4.2.1 一阶谓词表示

用谓词逻辑表示如下知识：

王宏是计算机系的一名学生。

王宏和李明是同班同学。

凡是计算机系的学生都喜欢编程序。

解： 首先定义谓词： $CS(x)$ ：表示 x 是计算机系的学生。

$CM(x, y)$ ：表示 x 和 y 是同班同学。

$L(x, z)$ ：表示 x 喜欢 z 。

此时，可用谓词公式把上述知识表示为：

$CS(\text{Wang Hong})$

$CM(\text{Wang Hong}, \text{Li Ming})$

$(\forall x)(CS(x) \rightarrow L(x, \text{programing}))$

4.2.1 一阶谓词表示

三、谓词逻辑表示法的应用

<https://blog.csdn.net/starter/article/details/88638857>

用谓词表示法求解机器人摆积木问题。设机器人有一只机械手，要处理的世界有一张桌子，桌上可堆放若干相同的方积木块。机械手有 4 个操作积木的典型动作：从桌上拣起一块积木；将手中的积木放到桌之上；在积木上再摆上一块积木；从积木上面拣起一块积木。积木世界的布局如下图所示。

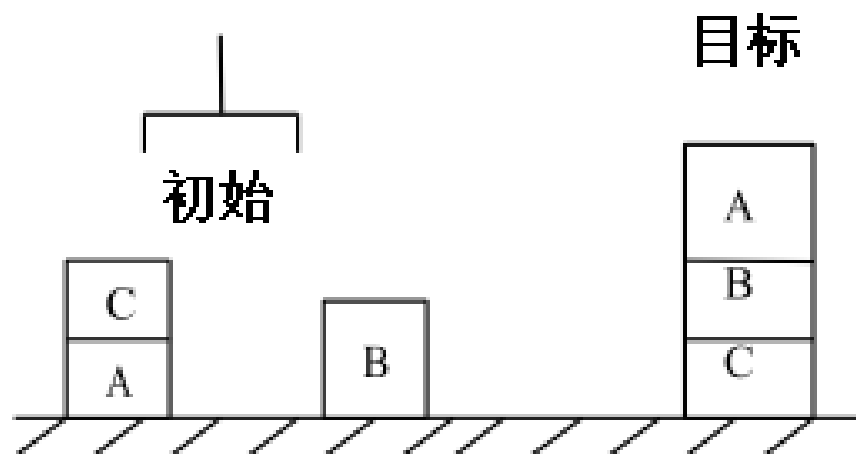


图 机器人摆积木问题

步骤一：定义描述状态的谓词如下：

EMPTY: 机械手中是空的

HOLD(x): 机械手中拿着积木x

ON(x,y): 积木x在积木y上面

CLEAR(x): 积木x的上面是空的

ONTABLE(x): 积木x在桌子上

其中，x和y的个体域为{A,B,C}

步骤二：问题的初始状态：

EMPTY

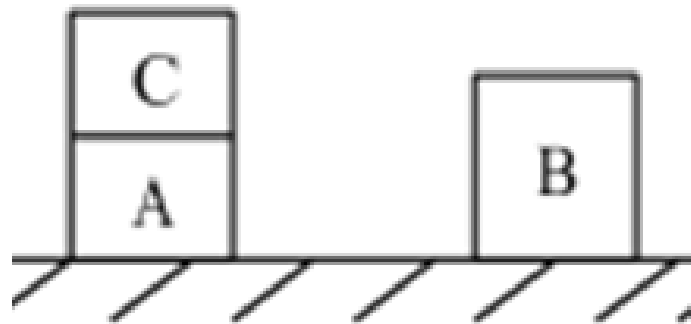
ONTABLE(A)

ONTABLE(B)

ON(C,A)

CLEAR(B)

CLEAR(C)



4.2.1 一阶谓词表示

步骤三：问题的目标状态：

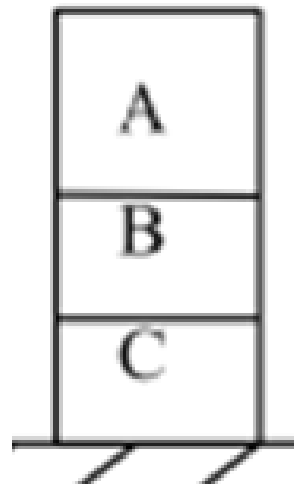
EMPTY

ONTABLE(C)

ON(B,C)

ON(A,B)

CLEAR(A)



步骤四：定义描述操作的谓词：

PICKUP(x): 从桌子上捡起积木x

PUTDOWN(x): 将手中的积木x放到桌子上

STACK(x,y): 在积木x上再摞上一块积木y

UNSTACK(x,y): 从积木x上面拣起一块积木y

其中，x和y的个体域为{A,B,C}

步骤四：定义描述操作的谓词

PICKUP(x)（从桌子上捡起积木x）

条件：EMPTY, ONTABLE(x), CLEAR(x)

动作：

删除：EMPTY, ONTABLE(x), CLEAR(x)

增加：HOLD(x)

PUTDOWN(x)：（将手中的积木x放到桌子上）

条件：HOLD(x)

动作：

删除：HOLD(x)

增加：EMPTY, ONTABLE(x), CLEAR(x)

STACK(x,y)：（在积木x上再摞上一块积木y）

条件：HOLD(y), CLEAR(x)

动作：

删除：HOLD(y), CLEAR(x)

增加：EMPTY, ON(x,y), CLEAR(y)

UNSTACK(x,y)：（从积木x上面拣起一块积木y）

条件：EMPTY, ON(x,y), CLEAR(y)

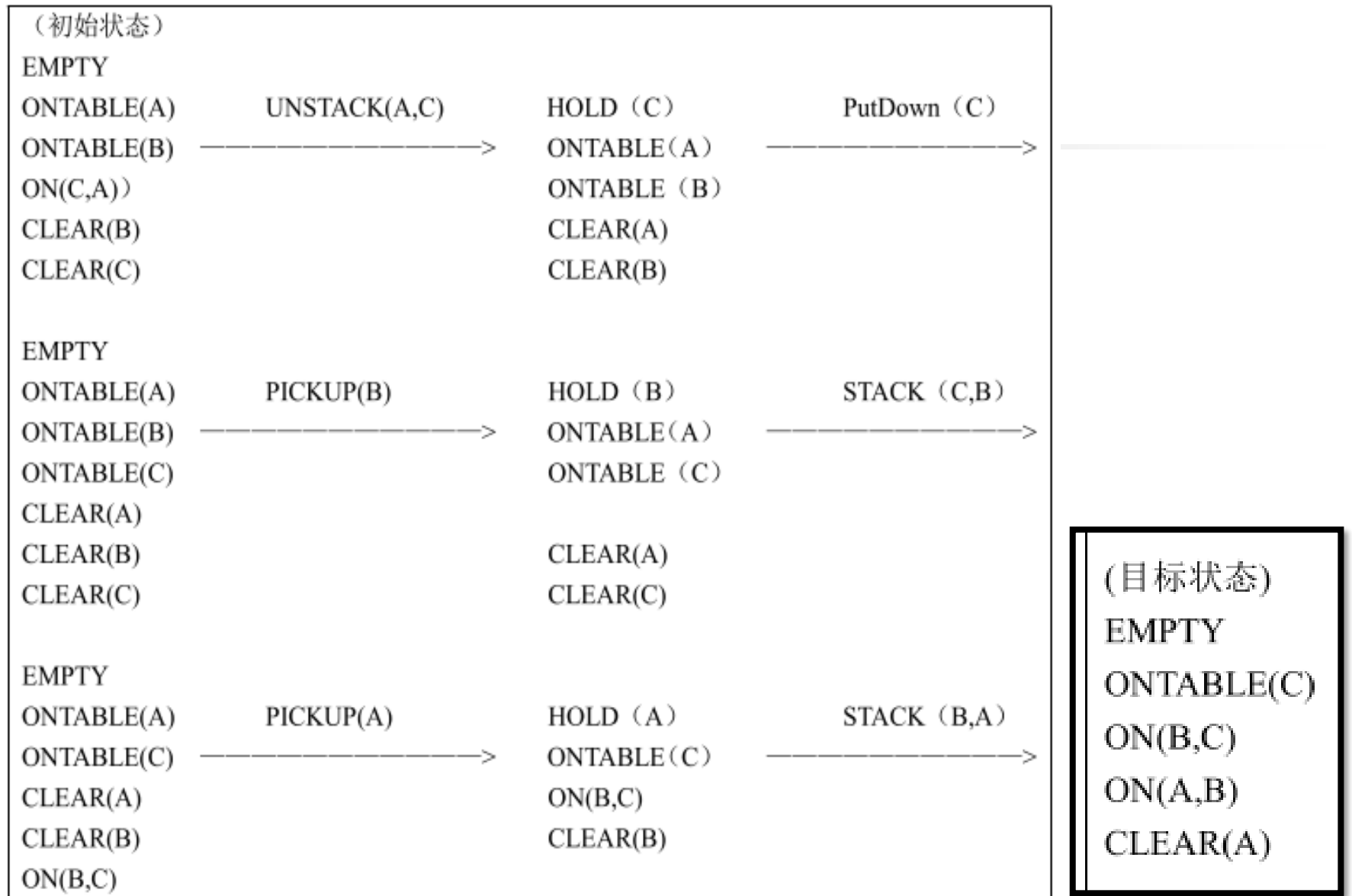
动作：

删除：EMPTY, ON(x,y), CLEAR(y)

增加：HOLD(y), CLEAR(x)

其中，x和y的个体域为{A,B,C}

■ 步骤六：问题的求解过程：





4.2.1 一阶谓词表示

■ 一阶谓词优缺点

■ 优点：

- 具有自然性、接近自然语言、容易接受、严密性、易于转化为计算机内部形式

■ 缺点：

- 无法表示不确定性知识、难以表示启发性知识及元知识、组合爆炸、效率低等缺点



第4章 知识表示方法

4.1 知识表示概念及特征

4.2 知识表示常见方法

4.2.1 一阶谓词表示

4.2.2 产生式表示

4.2.3 语义网络表示

4.2.4 框架表示

4.2.5 应用示例

4.3 知识图谱概念及应用

4.4 资源描述框架



4.2.2 产生式表示

产生式是表示因果之间关系的知识，其基本形式是：

$P \rightarrow Q$

或者

IF P THEN Q

P是前提(前件，条件，前提条件)

Q产生式的结论，操作亦可称后件

@

4.2.2 产生式表示

- 产生式表示法适合表示事实性知识和规则性知识：

- 确定性规则知识 **IF 动物会飞 AND 会下蛋 THEN 该动物是鸟**

$P \rightarrow Q$ IF P THEN Q

- 不确定性规则知识 **IF 发烧 THEN 感冒 (0.6)**

$P \rightarrow Q(\text{可信度})$ IF P THEN Q(可信度)

- 确定性事实知识
 - 三元组表示

事实“老李年龄是35岁”，(Lee,age,35)

事实“老李、老张是朋友”，(friend, Lee, Zhang)

(对象, 属性, 值) (关系, 对象1, 对象2)

- 不确定性事实知识

(对象, 属性, 值, 可信度值) (关系, 对象1, 对象2, 可信度值)

例：老李年龄很可能是40岁：(*Li*, *age*, 40, 0.8)
老李和老王不大可能是朋友：(*friend*, *Li*, *Wang*, 0.1)

产生式系统



- 一组协同作用的产生式。
- 一个产生式生成的结论可以供另一个产生式作为已知事实使用
- 早期的专家系统多数是基于产生式系统的。
- 由规则库、推理机构和综合数据库三个基本部分组成
 - 规则库
 - 存放与求解问题有关的某个领域知识的规则集合及其交换规则
 - 规则是以产生式表示的。规则集蕴涵着将问题从初始状态转换解状态的那些变换规则
 - 规则库是专家系统的核心。
 - 推理机
 - 由一组程序组成,用来控制产生式系统的运行,决定问题求解过程的推理线路,实现对问题的求解。
 - 产生式系统的控制策略随搜索方式的不同可分为可撤回策略、回溯策略、图搜索策略等。
 - 综合数据库
 - 存放着初始事实、外部数据库输入的事实、中间结果事实和最后结果事实。

动物识别规则库

R1: 动物有毛 → 哺乳类

R2: 动物产奶 → 哺乳类

R3: 动物有羽毛 → 鸟类

R4: 动物会飞 ∧ 会下蛋 → 鸟类

R5: 哺乳类 ∧ 动物吃肉 → 食肉类

R6: 动物有犬齿 ∧ 有爪 ∧ 眼盯前方 → 肉食类

R7: 哺乳类 ∧ 有蹄 → 蹄类动物

R8: 哺乳类 ∧ 反刍 → 蹄类

R9: 食肉类 ∧ 黄褐色 ∧ 有斑点 → 金钱豹

R10: 食肉类 ∧ 黄褐色 ∧ 有黑色条纹 → 虎

R11: 蹄类 ∧ 长脖 ∧ 长腿 ∧ 有斑点 → 长颈鹿

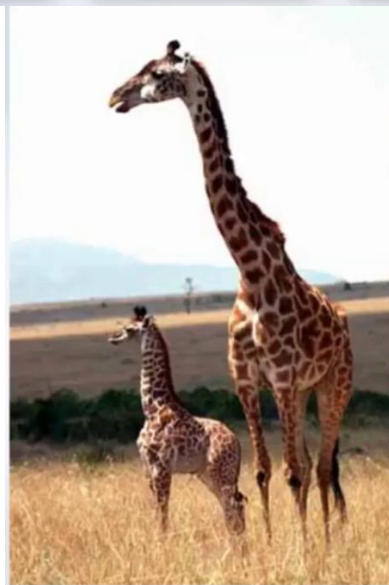
R12: 蹄类 ∧ 有黑色条纹 → 斑马

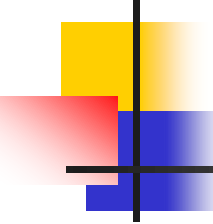
R13: 鸟类 ∧ 长脖 ∧ 长腿 ∧ 不会飞 → 鸵鸟

R14: 鸟类 ∧ 会游泳 ∧ 黑白二色 ∧ 不会飞 → 企鹅

R15: 鸟类 ∧ 善飞 → 信天翁

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





4.2.2 产生式表示

■ 产生式系统的推理方式

- 正向推理：从已知事实出发，通过规则库求得结论，或称数据驱动方式
- 反向推理：从目标（作为假设）出发，反向使用规则，求得已知事实，或称目标驱动方式
- 双向推理：同时使用正向推理又使用反向推理。

■ 产生式规则表示法优缺点

- 优点：自然性好，易于模块化管理、表示清晰等
- 缺点：效率不高、不能表达具有结构性的知识等

■ 经常将它与其它知识表示方法(如框架表示法、语义网络表示法)相结合



第4章 知识表示方法

4.1 知识表示概念及特征

4.2 知识表示常见方法

4.2.1 一阶谓词表示

4.2.2 产生式表示

4.2.3 语义网络表示

4.2.4 框架表示

4.2.5 应用示例

4.3 知识图谱概念及应用

4.4 资源描述框架

4.2.3 语义网络表示

■ 语义网络：

- 是一种通过概念及其语义关系来表达知识的有向图

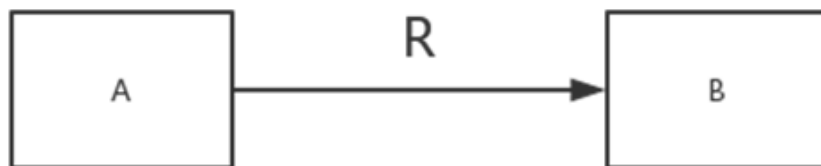
- 包含：

- 节点：代表实体，表示各种事物、概念、情况、属性、状态、事件、动作等。节点也可以是语义子网
- 弧：代表语义关系，表示它所连接的两个实体之间的语义联系。弧有方向、标注，方向表示节点间的主次关系，且方向不能随意调换；标注表示语义联系
- 在语义网络表示中，每一个节点和弧都必须有标志，用来说明它所代表的实体或语义。

■ 语义基元：在语义网络表示中最基本的语义单元

- 三元组描述：（节点1，弧，节点2）

■ 基本网元：一个语义基元所对应的那部分网络结构



4.2.3 语义网络表示

■ 常用的基本语义关系（<https://blog.csdn.net/hjc256/article/details/88823327>）

1.类属关系：指具有共同属性的不同事务间的分类关系、成员关系或示例关系，包括：

实例关系（ISA: is a）:

- 一个事物是另一个事物的具体例子。例如“我是一个人”，含义为“是一个”



分类关系（泛化关系“AKO”，a kind of）:

- 一个事物是另一个事物的一个成员，体现的是子类与父类的关系



成员关系(A-Member-of):

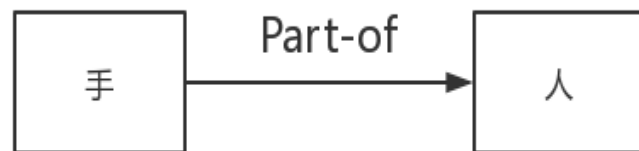
- 体现个体与集体的关系，表示一个事物是另一个事物的成员型



4.2.3 语义网络表示

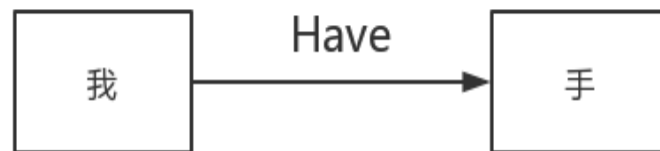
2.包含关系（Part-of）：

- 具有组织或结构特征的“部分与整体”之间的关系。
- 跟分类关系最主要区别在于包含关系一般不具备属性的继承性。



3.属性关系：事物与其行为、能力、状态、特征等属性之间的关系，常用多种，例如：

- Have, 含义为“有”，例如“我有手”
- Can, 含义为“可以、会”，例如“狗会跑”
- Age, 含义为年龄，例如“我今年22岁”



4.时间关系：表示时间上的先后次序关系。常用的时间关系有：

- Before: 表示一个事件在另一个事件之前发生
- After: 表示一个事件在另一个事件之后发生
- 例如“深圳大运会在广州亚运会之后举行”



4.2.3 语义网络表示

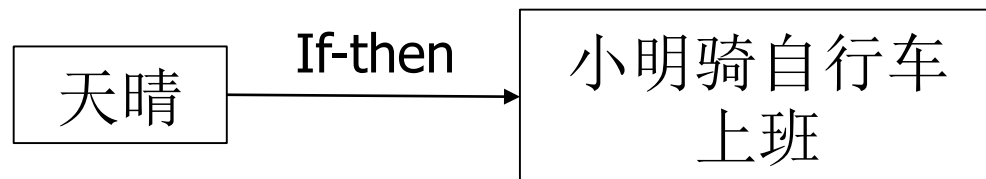
5.位置关系：是指不同的事物在位置方面的关系，常用的有：

- Located-on：表示某一物体在另一物体上面
- Located-at：表示某一物体所处的位置
- Located-under：表示某一物体在另一物体下方
- Located-inside：表示某一物体在另一物体内部
- Located-outside：表示某一物体在另一物体外部
- 例如“书在桌上”



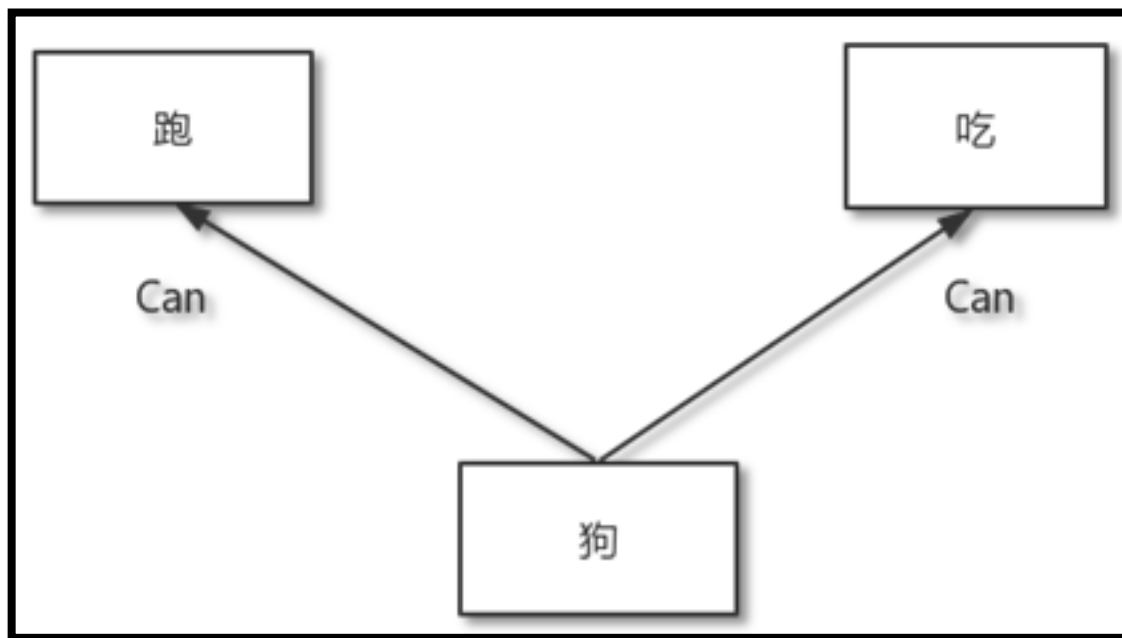
6.其它：

- 相近关系: Similar-to, Near-to
- 因果关系: If-then
- 组成关系: Composed-of
- 等



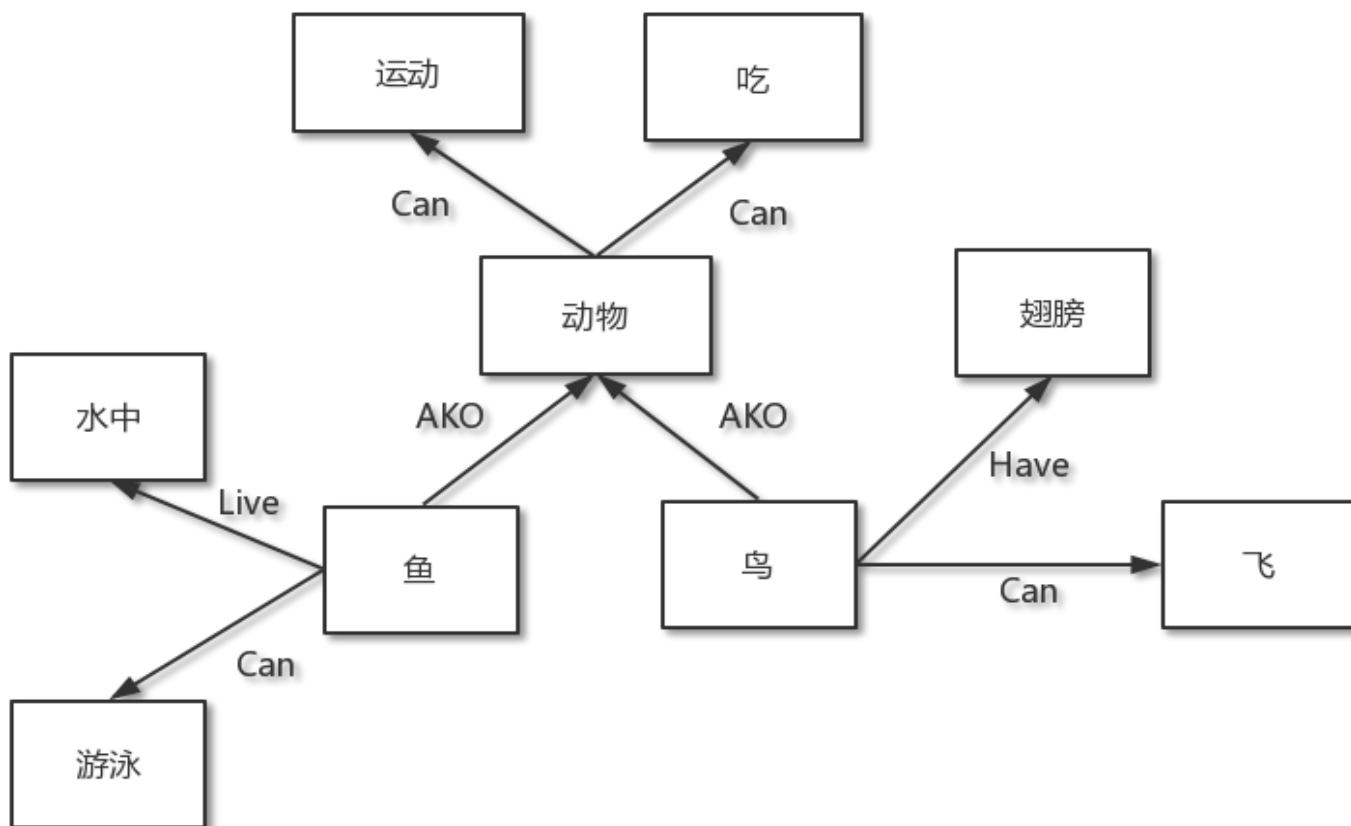
4.2.3 语义网络表示

- 语义网络表示事物与概念
 - “狗能吃，会跑”



较复杂关系的表示方法

1. 动物能吃、能运动
2. 鸟是一种动物，鸟有翅膀、会飞
3. 鱼是一种动物，鱼生活在水中、会游泳

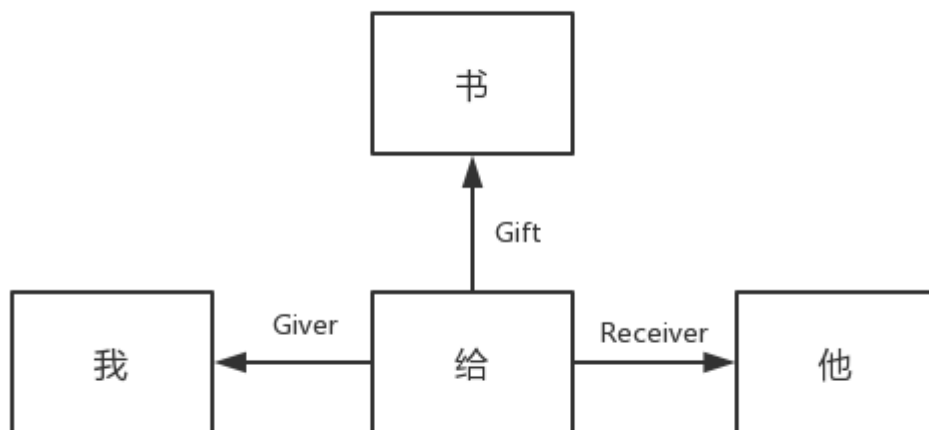


4.2.3 语义网络表示

- 事件和动作的表示：

- 用语义网络表示事件或动作时需要设立一个事件节点。
- 事件节点有一些向外引出的弧，表示动作的主体和客体。

- 例如，“我给他一本书”



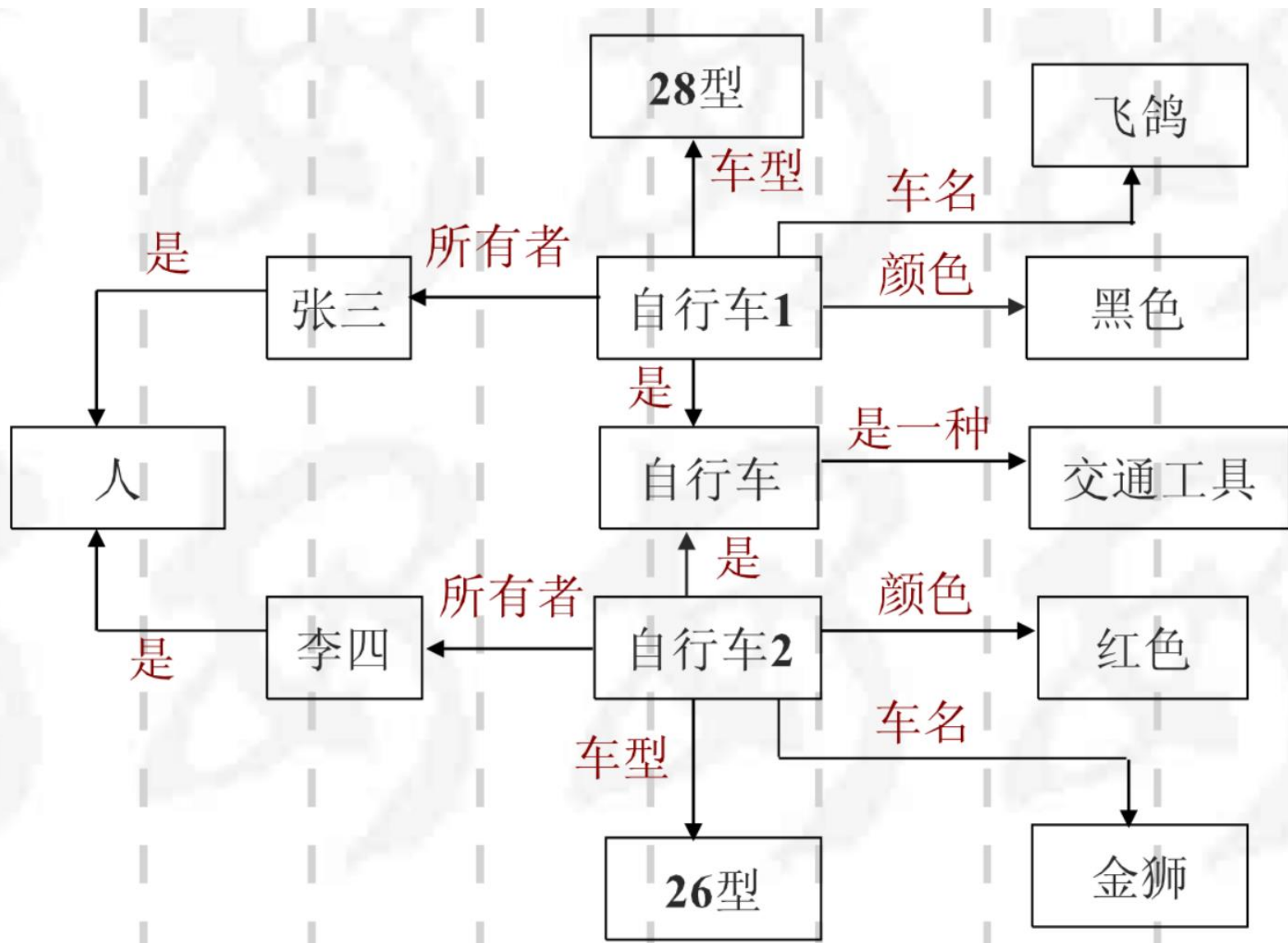
用语义网络表示比较复杂的知识

设有如下两个事实：张三的自行车是飞鸽牌，黑色，28型

李四的自行车是金狮牌，红色，26型

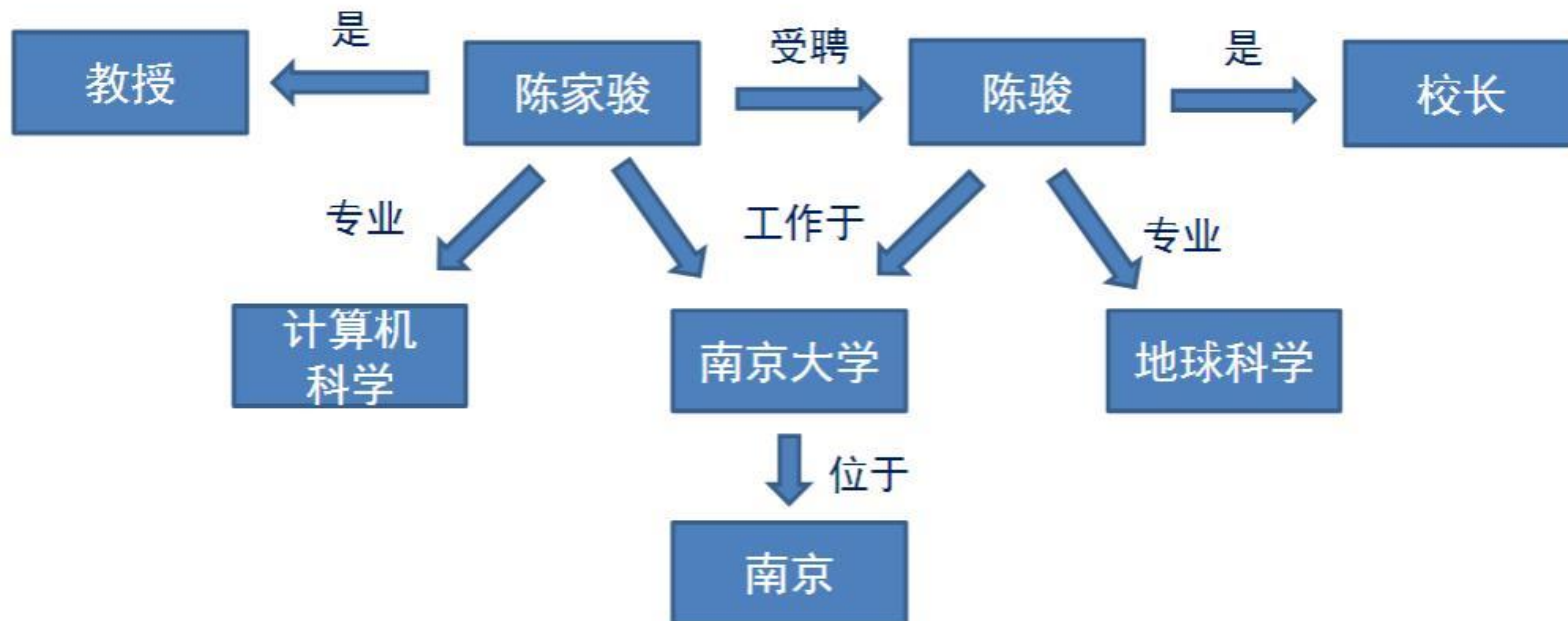
将其用语义网络描述出来。

- 如写成两个网络，很容易，但对知识的利用带来不便，如何写成一个呢？
- 分析事实发现，它们都是关于自行车的，因此只要把自行车作为一个通用概念用一个节点表示，而把张三李四的自行车作为他们的实例。这样，就很容易用一个语义网络把它们表示出来，当要寻找有关自行车的信息时，只要首先找到自行车这个节点就可以



4.2.3 语义网络表示

- 语义网例：陈骏是南京大学的校长；南京大学在南京；陈骏专业是地球科学。陈家骏是陈骏聘用的教授，陈家骏专业是计算机科学。





4.2.3 语义网络表示

■ 语义网络下的推理

- 语义网络表示法是依继承和匹配来进行推理的。
- 1. 继承。把对事物的描述从抽象节点传递到具体节点，通常沿着类属关系ISA, AKO等具有继承关系的边进行。
- 2. 匹配。把待求解问题构造为网络片段，其中某些节点或边的标识是空的，称为询问点。将网络片段与知识库中的某个语义网络片段进行匹配，则与询问点相匹配的事实就是该问题的解

■ 优点：

- 结构性、联想性、自索引性、自然语言的转换性
- 善于处理结构性的知识

■ 缺点：

- 不严格性、处理复杂
- 本质和谓词演算等价



第4章 知识表示方法

4.1 知识表示概念及特征

4.2 知识表示常见方法

4.2.1 一阶谓词表示

4.2.2 产生式表示

4.2.3 语义网络表示

4.2.4 框架表示

4.2.5 应用示例

4.3 知识图谱概念及应用

4.4 资源描述框架



4.2.4 框架表示

<https://blog.csdn.net/hjc256/article/details/88858261>

■ 框架理论思想

- 框架理论认为，我们对世间事物的认识都是以类似框架的结构存储在记忆中的。
- 当遇到一个新事物就从记忆中找出一个合适的框架，并根据新的情况对齐细节加以修改、补充，从而形成对这个新事物的认识。
- 用以往经验来认识新事物是人们常用的方法，但是我们又不可能把过往发生过的所有经验都存在脑海中，而只能用一种通用的数据结构把它们存储起来，当新情况发生的时候只需把新的数据添加到已有的通用数据结构中便可以实例化一个具体的实体。这种通用数据结构就是框架。

4.2.4 框架表示

- 框架是一种描述所讨论对象（事物、事件、概念等）属性和行为的数据结构
- 框架的基本结构

一个框架中可以包含各种信息：

- 描述事物的信息
 - 如何使用框架的信息
 - 关于下一步将发生什么情况的期望及如果期望的事件没有发生应该怎么办等信息等
- 这些信息包含在框架的各个槽或侧面中。

<框架名>

槽名1: 侧面名11 侧面值111, 侧面值112, ...

.....

侧面名1m 侧面值1m1, 侧面值1m2, ...

.....

槽名k: 侧面名k1 侧面值k11, 侧面值k12, ..., 侧面值k1n

.....

侧面名ks 侧面值ks1, 侧面值ks2, ..., 侧面值ksn

约束: 约束条件1

.....

约束条件n

< 大学教师 >

AKO: <教师>

院系:

方向:

职称: 范围(教授、副教授、讲师、助教)

缺省: 讲师

工资: < 工资框架 >

4.2.4 框架表示



框架名

每个框架都有一个框架名，**唯一标识一个框架。**

槽 (Slot) 、槽名、槽值

一个框架由若干个槽构成，每个槽都有槽名；

一个槽用于说明框架某一方面的属性；

属性的值即为槽值。

侧面 (Facet) 、侧面名、侧面值

一个槽有可能划分为若干个侧面，具有相应的侧面名；

一个槽可能含有若干细分属性，一个侧面用来说明其中的一个属性；

属性的值即为侧面值。

约束条件

用来约束、限制槽值、侧面值的填写。一般不单独列出，而包含在值的填写约束中。



4.2.4 框架表示

- 框架系统中定义一些公用、常用且标准的槽名，如：
 - ISA槽：对象间抽象概念的类属关系，“是一个”、“是一种”、“是一只”
 - AKO槽：对象间的类属关系，“是一种”
 - Instance槽：AKO槽的逆关系
 - Part-of槽：部分与全体的关系

4.2.4 框架表示

<https://blog.csdn.net/hjc256/article/details/88858261>

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; HomeUnit(Number)

MobileUnit(Number)

- 槽 侧面
- 该框架中的每个槽或侧面都给出了相应的说明信息，这些说明信息用来指出填写槽值或侧面值时的一些格式限制。

■ 说明：

- 性别槽的第二个侧面是默认值（当相应槽没填入槽值时，默认值作为槽值）
- 单位（unit）：指出填写槽值或侧面值时的书写格式，例如姓名槽应先写姓后写名
- 范围（Area）：指出所填的槽值仅能在指定的范围内选择
- 尖括号' <> '：括起来的是框架名



4.2.4 框架表示

描述教授有关情况的框架

框架名：<教授>

姓名：单位（姓，名）

性别：范围（男，女）

默认（男）

年龄：单位（岁）

研究方向：单位（方向名）

主持课题：<课题>

任职时间：单位（年，月）

框架名：<教授>

姓名：高阳

性别：男

年龄：44

研究方向：人工智能

主持课题：<国家级课题>

任职时间：2008年12月

- 当知识的结构比较复杂时，往往需要多个相互联系的框架来表示。

- 学生框架：

- 描述所有学生的共性，硕士框架描述硕士生的个性，并继承学生框架的所有属性。

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; HomeUnit(Num

MobileUnit(Numb

If-Needed; Ask-Te

硕士生实例框架

Frame <Master-1>

ISA; <Master>

Name; Yang Ye

Sex; female

Major; Computer

Field; Web-Intelligence

Advisor; Lin Hai

Project; Provincial

- 新的硕士框架：

- 学生框架的子框架

Frame <Master>

AKO; <Student>

Major; Unit(Major)

If-Needed; Ask-Major

If-Added; Check-Major

Field; Unit(Field)

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

表示这个实例框架是
硕士框架的实例

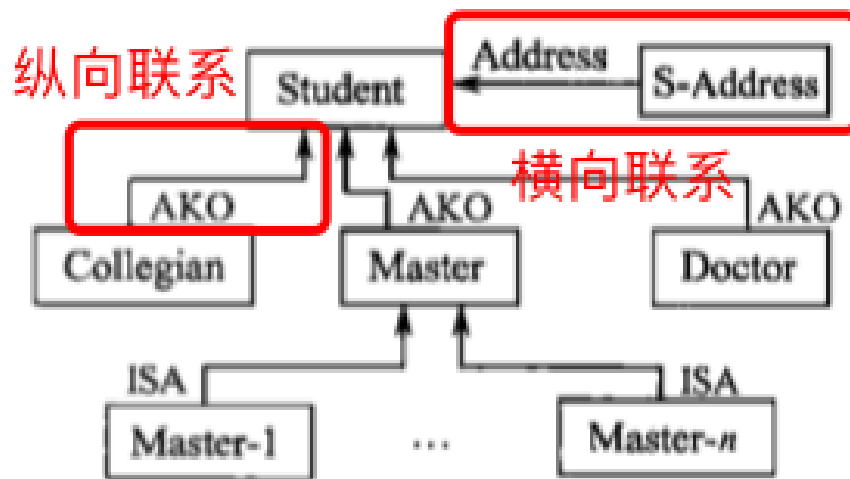
4.2.4 框架表示

- 框架的继承
- 框架中设置的三个侧面：Default Added来组合实现：

- **Default:** 槽统一的默认值
- **If-needed:** 当某个槽不能提供统一的默认值时，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
- **If-added:** 当某个槽值变化会影响到其他槽时，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后继处理。

```
Frame <Master>
  AKO; <Student>
  Major; Unit(Major)
    If-Needed; Ask-Major
    If-Added; Check-Major
  Field; Unit(Field)
    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
```


4.2.4 框架表



■ 框架系统

- 把一组有关的框架连接起来便可以形成一个框架系统

■ 框架之间的纵向联系

- 通过定义槽名**AKO**、**ISA**来实现的。

■ 框架之间的横向联系

- 一个框架的槽值或侧面值可以是另外一个框架的名字，这就在框架之间建立了横向联系。



4.2.4 框架表示

- 框架表示下的推理
- 框架表示法没有固定的推理机理。但框架系统的推理和语义网络一样遵循匹配和继承的原则，而且框架中如if-needed、if-added等槽的槽值是附加过程，在推理过程中起重要作用。
- 优点：
 - 结构性（不同于语义网络的结构性）、继承性、自然性
 - 是语义网络的重要扩展
 - 面对对象语言OO的产生
- 缺点：
 - 缺乏过程性知识表示



第4章 知识表示方法

4.1 知识表示概念及特征

4.2 知识表示常见方法

4.2.1 一阶谓词表示

4.2.2 产生式表示

4.2.3 语义网络表示

4.2.4 框架表示

4.2.5 应用示例

4.3 知识图谱概念及应用

4.4 资源描述框架

4.2.5 应用示例

- 例.动物识别系统（第四章\代码\animal.py）
- Python实现4.2.3产生式系统

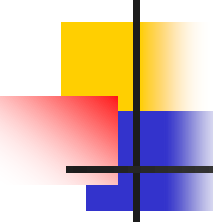
```
def match(key, features):  
    for x in key:  
        if x not in features:  
            return False  
    return True
```

```
feature_dict = {  
    '1': '有毛发', '2': '产奶', '3': '有羽毛', '4': '不会飞', '5': '会下蛋', '6': '吃肉', '7': '有犬齿', '8': '有爪', '9': '眼盯前方',  
    '10': '有蹄', '11': '反刍', '12': '黄褐色', '13': '有斑点', '14': '有黑色条纹', '15': '长脖', '16': '长腿', '17': '???', '18': '会游泳',  
    '19': '黑白二色', '20': '善飞', '21': '哺乳类', '22': '鸟类', '23': '食肉类', '24': '蹄类', '25': '金钱豹', '26': '虎', '27': '长颈鹿',  
    '28': '斑马', '29': '鸵鸟', '30': '企鹅', '31': '信天翁'}  
}
```

```
hop1_reason_dict = {  
    ('1'):'21',  
    ('2'):'21',  
    ('3'):'22',  
    ('4','5'):'22',  
    ('7','8','9'):'23',  
}
```

```
hop2_reason_dict = {  
    ('6','21'):'23',  
    ('10','21'):'24',  
    ('11','21'):'24',  
}
```


```
hop3_reason_dict = {  
    ('23','12','21','13'):'25',  
    ('23','12','21','14'):'26',  
    ('24','13','15','16'):'27',  
    ('14','24'):'28',  
    ('22','4','15','16'):'29',  
    ('22','4','18','19'):'30',  
    ('22','20'):'31',  
}
```



```
print("*"*60)
for i in range(24):
    s = '*' + str(i+1) + ':' + feature_dict[str(i+1)]
    print(s, end=' ')
    if (i+1) % 5 == 0:
        print()
print()
print("*"*60)

features = []
while True:
    tmp = input('请输入: ')
    if tmp == '0':
        break
    features.append(tmp)

print("前提条件: ")
for i in features:
    print(feature_dict[i], end=' ')
print()
```



```
print("推理过程如下：")
for key in hop1_reason_dict:
    value = hop1_reason_dict[key]
    if match(key, features) and value not in features:
        features.append(value)
        for x in key:
            print('%s'%feature_dict[x], end=', ')
        print('--> %s'%feature_dict[value])

for key in hop2_reason_dict:
    value = hop2_reason_dict[key]
    if match(key, features) and value not in features:
        features.append(value)
        for x in key:
            print('%s'%feature_dict[x], end=', ')
        print('--> %s'%feature_dict[value])

for key in hop3_reason_dict:
    value = hop3_reason_dict[key]
    if match(key, features) and value not in features:
        features.append(value)
        for x in key:
            print('%s'%feature_dict[x], end=', ')
        print('--> %s'%feature_dict[value])
```

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64-bit] on win32

Type "help", "copyright", "credits" or "license()" for more information
>>>

== RESTART: C:\Users\fcx\Desktop\2021关注\信息表达与智能处理2021-笔记本
章\代码\animal.py =

*1:有毛发	*2:产奶	*3:有羽毛	*4:不会飞	*5:会下蛋
*6:吃肉	*7:有犬齿	*8:有爪	*9:眼盯前方	*10:有蹄
*11:反刍	*12:黄褐色	*13:有斑点	*14:有黑色条纹	*15:长脖
*16:长腿	*17:???	*18:会游泳	*19:黑白二色	*20:善飞
*21:哺乳类	*22:鸟类	*23:食肉类	*24:蹄类	

请输入: 3

请输入: 5

请输入: 20

请输入: 0

前提条件:

有羽毛 会下蛋 善飞

推理过程如下:

有羽毛, --> 鸟类

鸟类, 善飞, --> 信天翁

>>> |