

第四章 知识表示及推理

4.1 基本概念

4.2 一阶谓词逻辑表示法

4.3 产生式表示方法及推理

4.4 语义网络表示法

4.5 框架表示及推理

4.1 基本概念

- * 人工智能研究中最基本的问题之一
 - * 在知识处理中总要问到：“如何表示知识？”，“知识是用什么来表示的？”。怎样使机器能懂，能对之进行处理，并能以一种人类能理解的方式将处理结果告诉人们。

知识及特性

* 知识:

- * 是有关信息关连在一起形成的信息结构
- * 信息：是数据的语言（含义）
- * 数据：是信息的载体和表示

* 知识的特性

- * 相对正确性
- * 不确定性
- * 模糊性
- * 可表示性与可利用性

知识的类型（作用效果）

- * **事实性知识**：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；
- * **控制性知识**：（元知识或超知识）
* 是关于如何使用过程性知识的知识；
* **如**：推理策略、搜索策略、不确定性的传播策略。

知识的类型（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

表示方法：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，小等

表示方法：用可能性、隶属度来描述

- * 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

知识表示的作用和评价

- * 知识表示的作用
 - * 建立智能系统的关键
- * 选择知识表示方法考虑的因素
 - * 充分表示领域知识
 - * 有利于对知识的利用
 - * 便于组织维护和管理
 - * 便于理解和实现（外部表示和内部表示）

基本的知识表示方法

- * 一阶谓词逻辑表示法
- * 产生式表示方法
- * 框架表示方法
- * 语义网络表示法

4.2 一阶谓词逻辑表示法

- * 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

1.谓词逻辑表示的逻辑学基础

命题、真值、论域

命题

断言：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。（定义2.1）

真值

T：表示命题的意义为真

F：表示命题的意义为假

说明：

一个命题不能同时既为真又为假

一个命题可在一定条件下为真，而在另一条件下为假

论域

由所讨论对象的全体构成的集合。也称为个体域

论域中的元素称为个体

1.谓词表示的逻辑学基础

谓词、函数

谓词

用来表示谓词逻辑中命题，形如 $P(x_1, x_2, \dots, x_n)$ 。其中
 P 是谓词名，即命题的谓语，表示个体的性质、状态或个体之间的关系；
 x_1, x_2, \dots, x_n 是个体，即命题的主语，表示独立存在的事物或概念。

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

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

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

例如：**GREATER**($x, 6$)，表示 x 大于6，

函数：

可作为谓词的个体

定义2.3: 设 D 是个体域， $f: D^n \rightarrow D$ 是一个映射，其中

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

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

1.谓词逻辑表示逻辑学基础

连词:

连词和量词

- \neg : “非”或者“否定”。表示对其后面的命题的否定
- \vee : “析取”。表示所连结的两个命题之间具有“或”的关系
- \wedge : “合取”。表示所连结的两个命题之间具有“与”的关系。
- \rightarrow : “条件”或“蕴含”。表示“若...则...”的语义。读作“如果P，则Q”
其中，P称为条件的前件，Q称为条件的后件。
- \leftrightarrow : 称为“双条件”。它表示“当且仅当”的语义。
例如，对命题P和Q， $P \leftrightarrow Q$ 表示“P当且仅当Q”，

量词:

- \forall : 全称量词。意思是“所有的”、“任一个”
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

1.谓词逻辑表示的逻辑学基础

自由变元和约束变元

辖域：指位于量词后面的单个谓词或者用括弧括起来的合式公式

约束变元：辖域内与量词中同名的变元称为约束变元

自由变元：不受约束的变元称为自由变元

例子： $(\forall x)(P(x, y) \rightarrow Q(x, y)) \vee R(x, y)$

其中， $(P(x, y) \rightarrow Q(x, y))$ 是 $(\forall x)$ 的辖域

辖域内的变元 x 是受 $(\forall x)$ 约束的变元

$R(x, y)$ 中的 x 和所有的 y 都是自由变元

变元的换名：

谓词公式中的变元可以换名。但需注意：

第一：对约束变元，必须把同名的约束变元都统一换成另外一个相同的名字，且不能与辖域内的自由变元同名。

例，对 $(\forall x)P(x, y)$ ，可把约束变元 x 换成 z ，得到公式 $(\forall z)P(z, y)$ 。

第二：对辖域内的自由变元，不能改成与约束变元相同的名字。

例，对 $(\forall x)P(x, y)$ ，可把 y 换成 t ，得到 $(\forall x)P(x, t)$ ，但不能换成 x 。

2.谓词逻辑表示方法

表示步骤及简例(1/2)

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

简例

例2.1 表示知识“所有教师都有自己的学生”。

解：先定义谓词：

$T(x)$ ：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。

然后将知识表示如下：

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

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 是学生，且 x 是 y 的老师。

2.谓词逻辑表示方法

表示步骤及简例(2/2)

例2.2 表示知识“所有的整数不是偶数就是奇数”。

解：先定义谓词：

$I(x)$: x 是整数, $E(x)$: x 是偶数, $O(x)$: x 是奇数

然后再将知识表示为：

$(\forall x)(I(x) \rightarrow E(x) \vee O(x))$

例2.3 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：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。

$L(x,y)$: 表示 x 喜欢 y 。

然后再将知识表示为：

$CS(\text{Wang hong})$

$CM(\text{Wanghong}, \text{Li ming})$

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

3.谓词逻辑表示的经典例子

机器人移盒子 (1/5)

例2.4 机器人移盒子

解：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

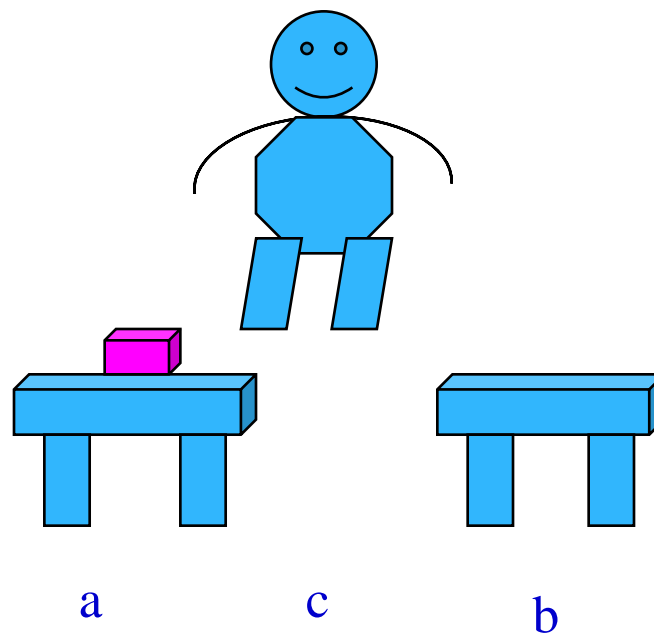
变元的个体域：

x的个体域是{a, b}

y的个体域是{robot}

z的个体域是{a, b, c}

w的个体域是{box}



3.谓词逻辑表示的经典例子

机器人移盒子 (2/5)

问题的初始状态:

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



问题的目标状态:

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

机器人行动的目标是把问题的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。

描述操作的谓词

条件部分: 用来说明执行该操作必须具备的先决条件，用谓词公式来表示。

动作部分: 给出了该操作对问题状态的改变情况，通过在执行该操作前的问题状态中删去和增加相应的谓词来实现。

这些操作包括:

Goto(x, y): 从x处走到y处。

Pickup(x): 在x处拿起盒子。

Setdown(y): 在x处放下盒子。

3.谓词逻辑表示的经典例子 机器人移盒子 (3/5)

各操作的条件和动作:

Goto(x, y)

条件: **AT(robot, x)**

动作: 删除表: **AT(robot, x)**

添加表: **AT(robot, y)**

Pickup(x)

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)

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)**

各操作的执行方法:

机器人每执行一操作前, 都要检查该操作的先决条件是否可以满足。如果满足, 就执行相应的操作; 否则再检查下一个操作。

3.谓词逻辑表示的经典例子

机器人移盒子 (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)

AT(robot, c)

开始 EMPTY(robot)

=====> ON(box, a)

TABLE(a)

TABLE(b)

状态2

AT(robot, a)

Goto(c, a) EMPTY(robot)

=====> ON(box, a)

TABLE(a)

TABLE(b)

状态3

AT(robot, a)

Pickup(a) HOLDS(robot, box)

=====> TABLE(a)

TABLE(b)

3.谓词逻辑表示的经典例子

机器人移盒子 (5/5)

状态4

AT(robot, b)

Goto(a, b) HOLDS(robot, box)

=====> TABLE(a)

TABLE(b)

状态5

AT(robot, b)

Setdown(b) EMPTY(robot)

=====> ON(box, b)

TABLE(a)

TABLE(b)

状态6(目标状态)

AT(robot, c)

Goto(b, c) EMPTY(robot)

=====> ON(box, b)

TABLE(a)

TABLE(b)

3.谓词逻辑表示的经典例子 猴子摘香蕉(1/4)

例2.5 猴子摘香蕉问题

解：先定义谓词

描述状态的谓词：

AT(x, y): x在y处

ONBOX: 猴子在箱子上

HB: 猴子得到香蕉

个体域：

x : {monkey, box, banana}

Y: {a, b, c}

问题的初始状态

AT(monkey, a)

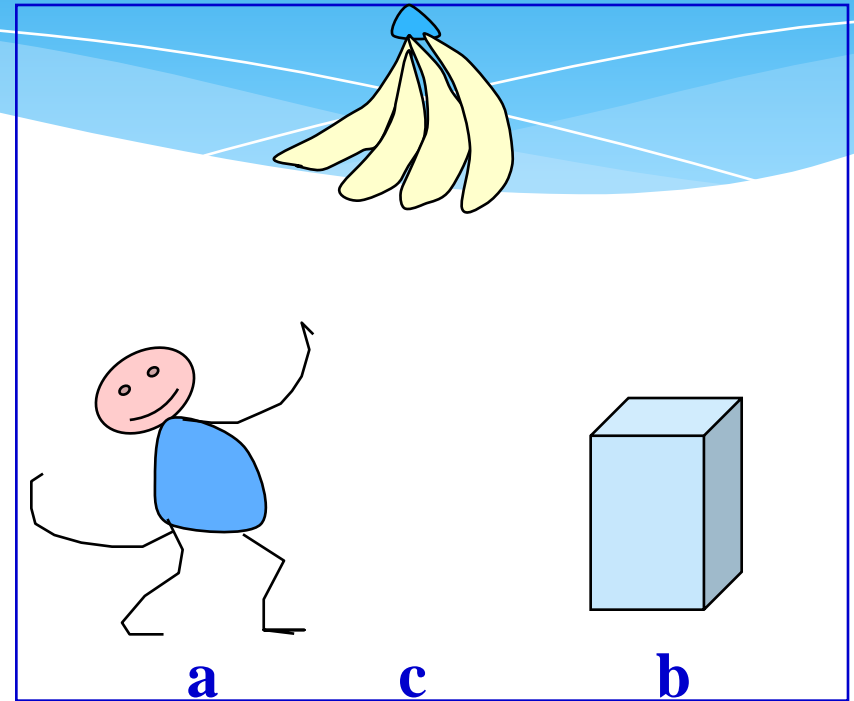
AT(box, b)

¬ONBOX , ¬HB

问题的目标状态

AT(monkey, c) , AT(box, c)

ONBOX , HB



3.谓词逻辑表示的经典例子

猴子摘香蕉 (2/4)

描述操作的谓词:

Goto(u, v): 猴子从u处走到v处

Pushbox(v, w): 猴子推着箱子从v处移到w处

Climbbox: 猴子爬上箱子

Grasp: 猴子摘取香蕉

各操作的条件和动作:

Goto(u, v)

条件: $\neg \text{ONBOX}$, $\text{AT}(\text{monkey}, u)$,

动作: 删除表: $\text{AT}(\text{monkey}, u)$ 添加表: $\text{AT}(\text{monkey}, v)$

Pushbox(v, w)

条件: $\neg \text{ONBOX}$, $\text{AT}(\text{monkey}, v)$, $\text{AT}(\text{box}, v)$

动作: 删除表: $\text{AT}(\text{monkey}, v)$, $\text{AT}(\text{box}, v)$

添加表: $\text{AT}(\text{monkey}, w)$, $\text{AT}(\text{box}, w)$

Climbbox

条件: $\neg \text{ONBOX}$, $\text{AT}(\text{monkey}, w)$, $\text{AT}(\text{box}, w)$

动作: 删除表: $\neg \text{ONBOX}$ 添加表: ONBOX

Grasp

条件: ONBOX , $\text{AT}(\text{box}, c)$

动作: 删除表: $\neg \text{HB}$

3.谓词逻辑表示的经典例子 猴子摘香蕉 (3/4)

猴子摘香蕉问题的求解过程如下：

状态1（初始状态）

AT(monkey, a)

开始

AT(box, b)

=====> \neg ONBOX

\neg HB

状态2

AT(monkey, b)

Goto(a, b)

AT(box, b)

=====> \neg ONBOX

\neg HB

状态3

AT(monkey, c)

Pushbox(b, c) AT(box, c)

=====> \neg ONBOX

\neg HB

3.谓词逻辑表示的经典例子

猴子摘香蕉 (4/4)

状态4

AT(monkey, c)

Climbbox

AT(box, c)

=====> ONBOX

¬HB

状态5(目标状态)

AT(monkey, c)

Grasp

AT(box, c)

=====> ONBOX

HB

4.谓词逻辑表示的特征

主要优点

自然：一阶谓词逻辑是一种接近于自然语言的形式语言系统，谓词逻辑表示法接近于人们对问题的直观理解

明确：有一种标准的知识解释方法，因此用这种方法表示的知识明确、易于理解

精确：谓词逻辑的真值只有“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，这种模块性使得添加、删除、修改知识比较容易进行

主要缺点

知识表示能力差：只能表示确定性知识，而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：它把推理演算与知识含义截然分开，抛弃了表达内容中所含有的语义信息，往往使推理过程冗长，降低了系统效率

4.3 产生式表示法及推理

- 产生式表示法又称为产生式规则表示法。有心理学家认为，人脑对知识的存储就是产生式形式。
- 产生式最早由P. Post于1943年提出，用于构造Post机计算模型；1972 A. Newell和H. A. Simon在研究人类的认识模型中提出了 Rule-Based 产生式系统方法以及规则表示模式。
- 目前，产生式表示法已经成为人工智能中应用最多的一种知识表示法，许多成功的专家系统都用它来表示知识。

1.产生式表示的基本方法

规则的表达

产生式也叫产生式规则，或简称规则。

规则的基本形式

IF P THEN Q 或者 $P \rightarrow Q$

其中，P是前提，也称或前件，给出了该产生式可否使用的先决条件。Q是结论或操作，也称后件，给出当P满足时，应该推出的结论或执行的动作。

形式化描述

<规则> : : = <前提> \rightarrow <结论>

<前提> : : = <简单条件> | <复合条件>

<结论> : : = <事实> | <动作>

<复合条件> : : = <简单条件> And <简单条件> [(And <简单条件> ...)]
| <简单条件> Or <简单条件> [(OR<简单条件> ...)]

<动作> : : = <动作名> | [(<变元>, ...)]

2.产生式表示简例

下面给出一个简化的动物识别例子（完整例子后面讨论），仅包括动物识别系统中的两条规则：

r_3 : IF 动物有羽毛 THEN 动物是鸟

r_{15} : IF 动物是鸟 AND 动物善飞 THEN 动物是信天翁

其中， r_3 和 r_{15} 是上述两条规则在动物识别系统中的规则编号，一般称为规则号。

r_3 :

前提条件是“动物有羽毛”

结论是“动物是鸟”

r_{15} :

前提条件是一个复合条件“动物是鸟 AND 动物善飞”，它是两个子条件的合取。

结论是“动物是信天翁”

3.产生式表示的特性

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发生联系，而不能相互调用，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。

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。

产生式系统的基本结构

- * 规则库：描述领域知识的产生式规则集
- * 综合数据库：记录求解（或推理）过程中各种信息的数据结构
- * 推理机构：负责匹配到得出结论的整个问题求解过程

产生式系统应用举例

- 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。



* 规则库（知识库）：

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

* 规则库（知识库）：

- 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

产生式系统的工作过程

- * 1.将观察到的事实放入综合数据库中
- * 2.用事实与规则库中的前提进行匹配
- * 3.将匹配成功规则的结论作为新的事实加到综合数据库
- * 4.重复上述2,3步直到满足结束条件

2.4.1 产生式系统简例

基于规则的动物识别系统

系统的推理过程

(1) 先从规则库中取出第一条规则 r_1 ，检查其前提是否可与综合数据库中的已知事实相匹配。 r_1 的前提是“动物有毛发”，但事实库中无此事实，故匹配失败。然后取 r_2 ，该前提可与已知事实“动物有奶”相匹配， r_2 被执行，并将其结论“动物是哺乳动物”作为新的事实加入到综合数据库中。此时，综合数据库的内容为：

动物有暗斑，动物有长脖子，动物有长腿，动物有奶，动物有蹄
动物是哺乳动物

(2) 再从规则库中取 r_3 ， r_4 ， r_5 ， r_6 进行匹配，均失败。接着取 r_7 ，该前提与已知事实“动物是哺乳动物”相匹配， r_7 被执行，并将其结论“动物是有蹄类动物”作为新的事实加入到综合数据库中。此时，综合数据库的内容变为：

动物有暗斑，动物有长脖子，动物有长腿，动物有奶，动物有蹄
动物是哺乳动物，动物是有蹄类动物

(3) 此后， r_8 ， r_9 ， r_{10} 均匹配失败。接着取 r_{11} ，该前提“动物是有蹄类动物 AND 动物有长脖子 AND 动物有长腿 AND 动物身上有暗斑”与已知事实相匹配， r_{11} 被执行，并推出“动物是长颈鹿”。由于“长颈鹿”已是目标集合中的一个具体动物，即已推出最终结果，故问题求解过程结束。

正向推理举例

初始综合数据库 (有长脖子, 有长腿 身上有暗斑点, 有奶, 有蹄)

* 匹配 R2, 结论 “哺乳动物”

综合数据库: (有长脖子, 有长腿 身上有暗斑点, 有奶, 有蹄, “哺乳动物”, R2)

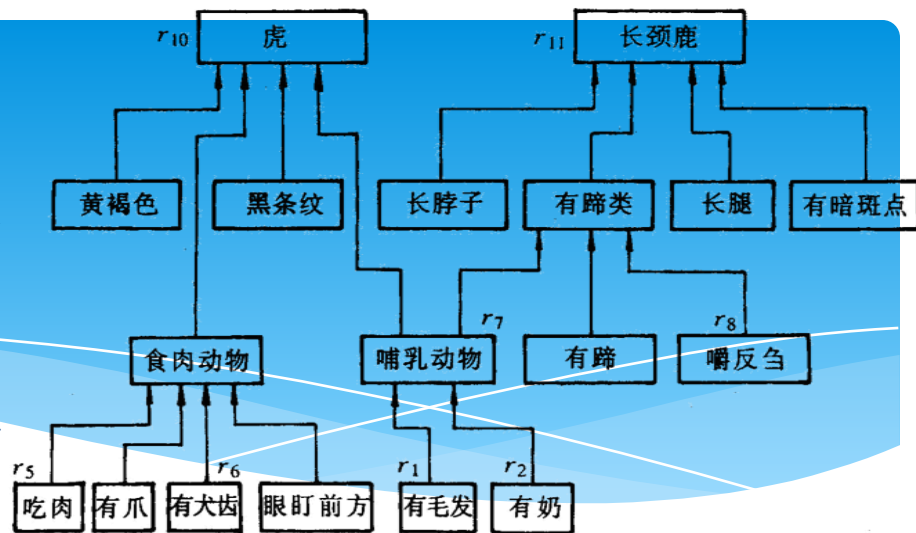
* 匹配 R7, 结论 “有蹄动物”

综合数据库: (有长脖子, 有长腿 身上有暗斑点, 有奶, 有蹄, “哺乳动物”, 有蹄动物, R2, R7)

* 匹配 R11, 结论 “长颈鹿”

综合数据库 (有长脖子, 有长腿 身上有暗斑点, 有奶, 有蹄, “哺乳动物”, 有蹄动物, **长颈鹿**, R2, R7, R11)

* 在结论表中有 “长颈鹿”, 找到目标, 推理结束。



4.4 语义网络表示

- * 语义网络的结构
- * 基本的语义关系
- * 语义网络的推理
- * 语义网络特点

1.语义网络概述

概念

语义网络是一种用实体及其语义关系来表达知识的有向图。

结点：代表实体，表示事物、概念、情况、属性、状态、事件、动作等

弧：代表语义关系，表示所连两个实体之间的语义联系，必须带有标识

语义基元

语义网络中最基本的语义单元称为语义基元，可用三元组表示为：

(结点1, 弧, 结点2)

基本网元

指一个语义基元对应的有向图，是语义网络中最基本的结构单元

例如：语义基元 (A, R, B) 所对应的基本网元，如图2-3所示。



图2-3

例2.6 用语义基元表示“鸵鸟是一种鸟”这一事实。

解：如图2-4所示。



图2-4

说明：弧的方向不可随意调换。

1.语义网络概述

基本语义关系(1/3)

实例关系：ISA

体现的是“具体与抽象”的概念，含义为“是一个”，表示一个事物是另一个事物的一个实例。例“李刚是一个人”，如图2-6。



图2-6 实例关系

分类关系：AKO

也称泛化关系，体现的是“子类与超类”的概念，含义为“是一种”，表示一个事物是另一个事物的一种类型。例“机器人是一种机器”，如图2-7。



图2-7 分类关系

成员关系：A-Member-of

体现的是“个体与集体”的关系，含义为“是一员”，表示一个事物是另一个事物的一个成员。例“张强是共青团员”，如图2-8。



图2-8 成员关系

上述关系的主要特征

属性的继承性，即处在具体层的结点可以继承抽象层结点的所有属性。

1.语义网络概述

基本语义关系(2/3)

属性关系

指事物和其属性之间的关系。常用的有：

Have: 含义为“有”，表示一个结点具有另一个结点所描述的属性

Can: 含义为“能”、“会”，表示一个结点能做另一个结点的事情

例如：“鸟有翅膀”，如图2-9

包含关系（聚类关系）

指具有组织或结构特征的“部分与整体”之间的关系。常用的包含关系是：

Part-of: 含义为“是一部分”，表示一个事物是另一个事物的一部分。

例如，“大脑是人体的一部分”，如图2-11

再如，“黑板是墙体的一部分”，如图2-12

聚类关系与实例、分类、成员关系的主要区别

聚类关系一般不具备属性的继承性。

如上例，大脑不一定具有人的各种属性

黑板也不具有墙的各种属性。



图2-9 属性关系



图2-11 包含关系一



图2-12 包含关系二

1.语义网络概述

基本语义关系(3/3)

时间关系

指不同事件在其发生时间方面的先后次序关系。

常用的时间关系有：

Before: 含义为“在前”

After: 含义为“在后”

如：“伦敦奥运会在北京奥运会之后”，图2-13



图2-13 时间关系

位置关系

指不同事物在位置方面的关系。常用的有：

Located-on: 含义为“在...上面”

Located-under: 含义为“在...下面”

Located-at: 含义为“在...”

如，“书在桌子上”，图2-14



图2-14 位置关系

相近关系

指不同事物在形状、内容等方面相似或接近。常用的相近关系有：

Similar-to: 含义为“相似”

Near-to: 含义为“接近”

如，“猫似虎”，图2-15



图2-15 相似关系

2.事物和概念的表示

表示一元关系

一元关系是指可以用一元谓词 $P(x)$ 表示的关系。谓词 P 说明实体的性质、属性等。常用：“是”、“有”、“会”、“能”等语义关系来说明。如，“雪是白的”。

一元关系的描述

一个一元关系就是一个语义基元，可用一个基本网元来表示。其中，结点1表示实体，结点2表示实体的性质或属性等，弧表示语义关系。

例如，“李刚是一个人”为一元关系，其语义网络如前所示。

例2.7 用语义网络表示“动物能运动、会吃”。

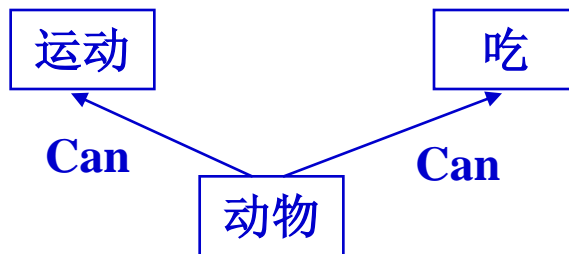


图2-16 动物的属性

2.事实 and 概念的表示

表示二元关系(1/3)

二元关系是指可用二元谓词 $P(x,y)$ 表示的关系。其中， 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。

复杂关系，可通过一些相对独立的二元或一元关系的组合来实现。

例2-8 用语义网络表示：

动物能运动、会吃。

鸟是一种动物，鸟有翅膀、会飞。

鱼是一种动物，鱼生活在水中、会游泳。

解：其语义网络表示如图2-17所示

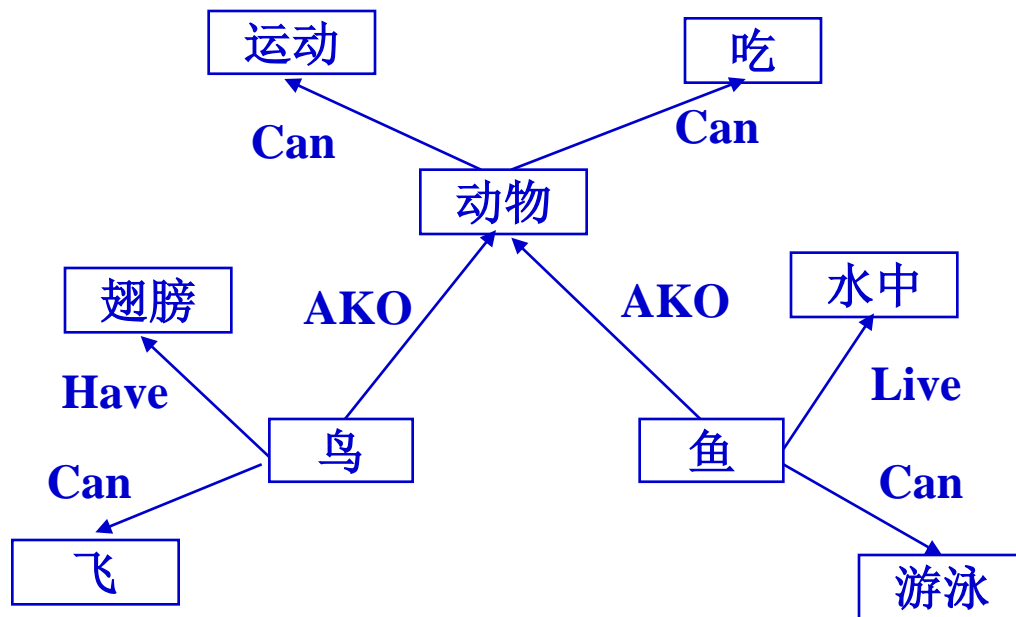


图2-17 动物分类的语义网络

2.事实和概念概念的表示

表示二元关系(2/3)

例2-9 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。

解: 其表示如图2.18所示



图2-18 经理王强的语义网络

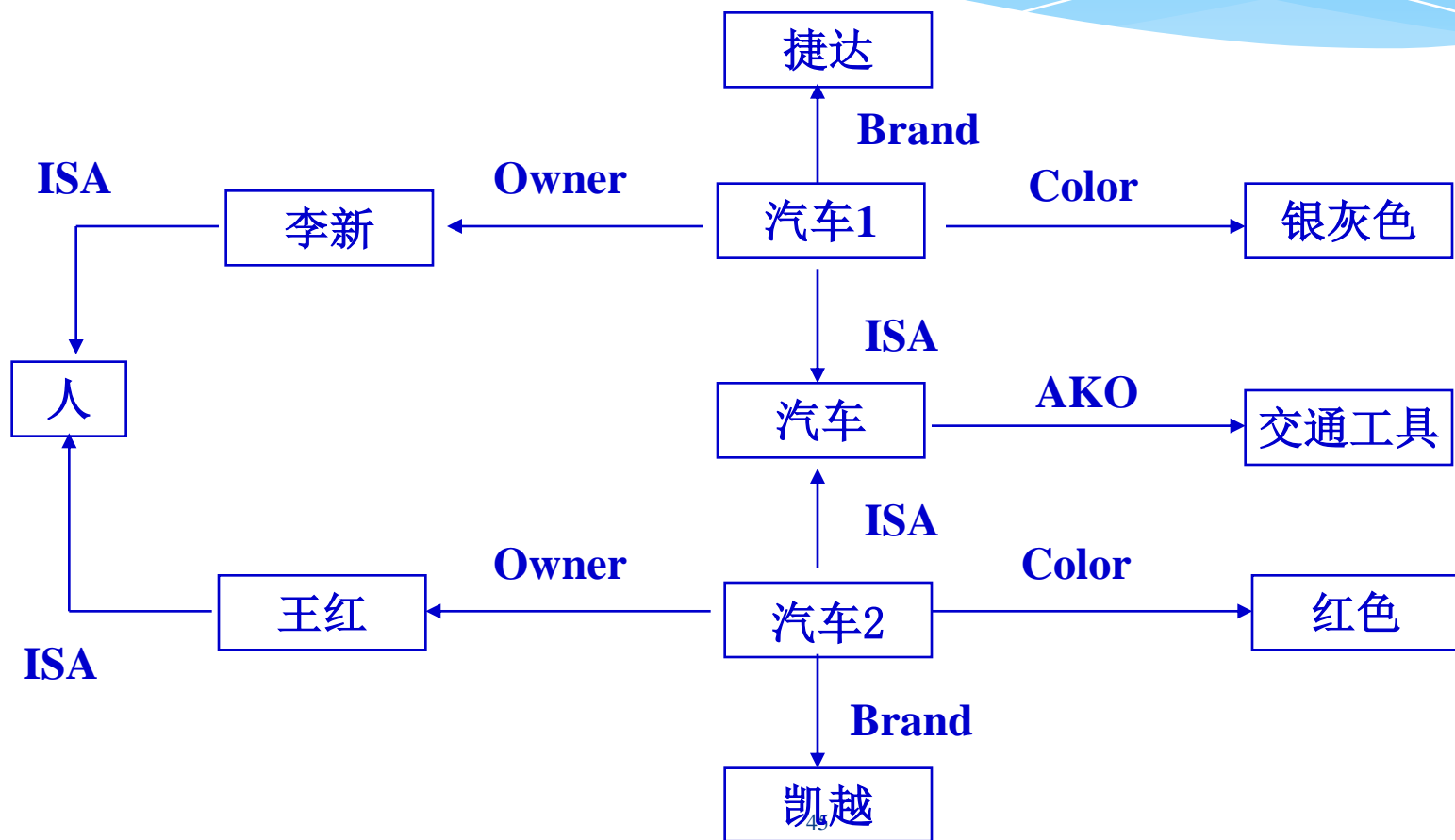
2.事物和概念的表达

表示二元关系(3/3)

例2-10：李新的汽车是“捷达”、银灰色。

王红的汽车是“凯越”、红色。

解：李新和王红的汽车均属于具体概念,可增加“汽车”这个抽象概念。



2.事物和概念的表示

表示多元关系

可用多元谓词 $P(x_1, x_2, \dots)$ 表示的关系。其中，个体 x_1, x_2, \dots 为实体，谓词 P 说明这些实体之间的关系。

多元关系的表示法

用语义网络表示多元关系时，可把它转化为一个或多个二员关系的组合，

3.情况和动作的表示

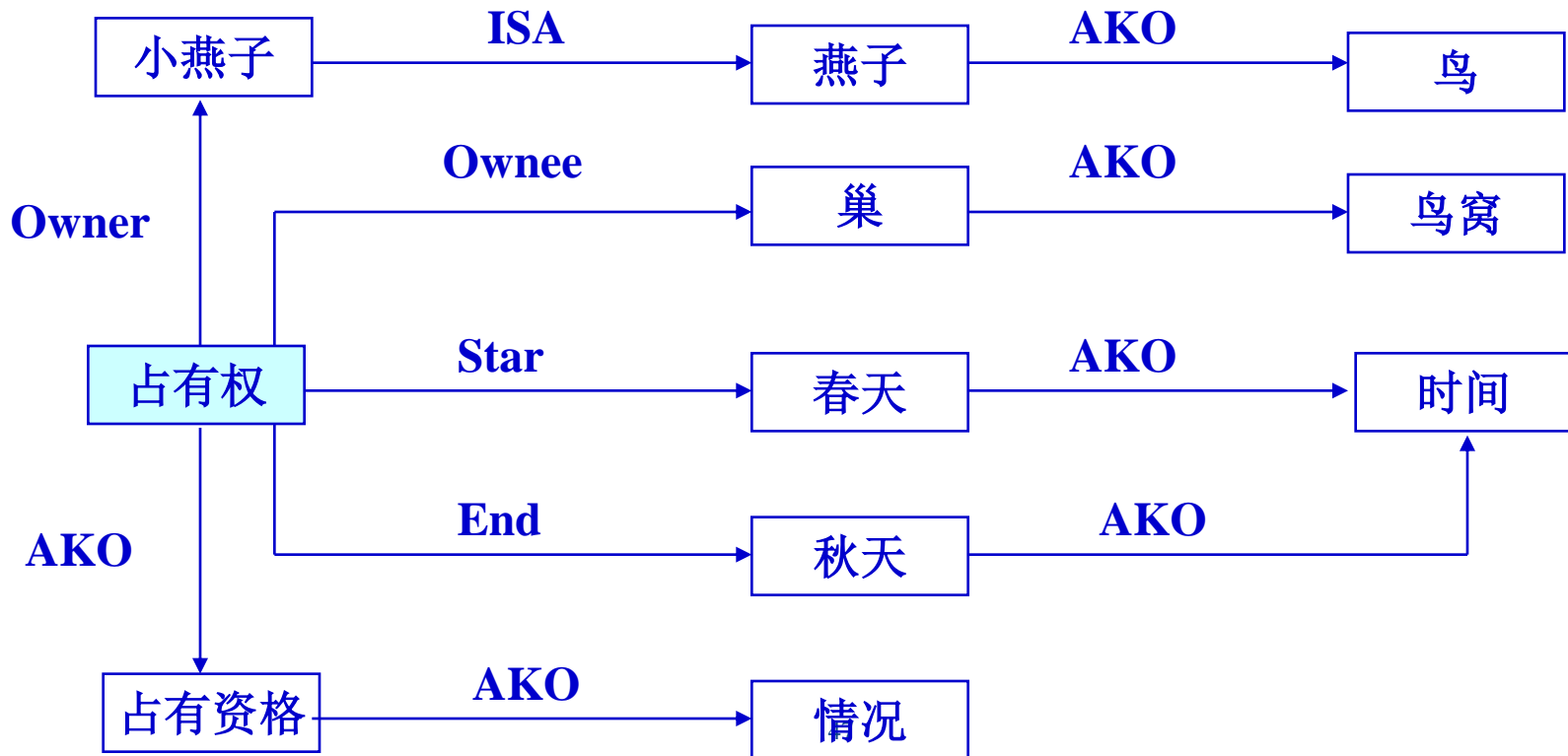
情况的表示(1/2)

表示方法：西蒙提出了增加情况和动作结点的描述方法。

例2.12：用语义网络表示：

“小燕子这只燕子从春天到秋天占有一个巢”

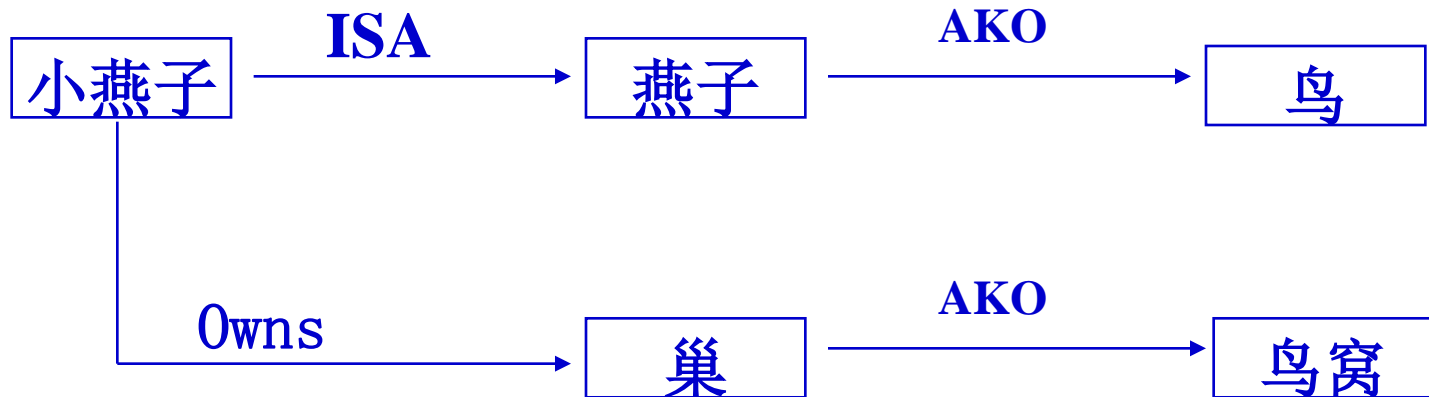
解：需要设立一个占有权结点，表示占有物和占有时间等。



3. 情况和动作的表示

情况的表示(2/2)

对上述问题，也可以把占有作为一种关系，并用一条弧来表示，但在这种表示方法下，占有关系就无法表示了



3.情况和动作的表示

事件和动作的表示

用这种方法表示事件或动作时，需要设立一个事件节点或动作结点。其中，事件节点由一些向外引出的弧来指出事件行为及发出者与接受者。动作结点由一些向外引出的弧来指出动作的主体与客体。

例2.13 用于语义网络表示：

“常河给江涛一个优盘”

解：用事件节点表示如图2.20所示。

用动作结点节点表示如图2.21所示

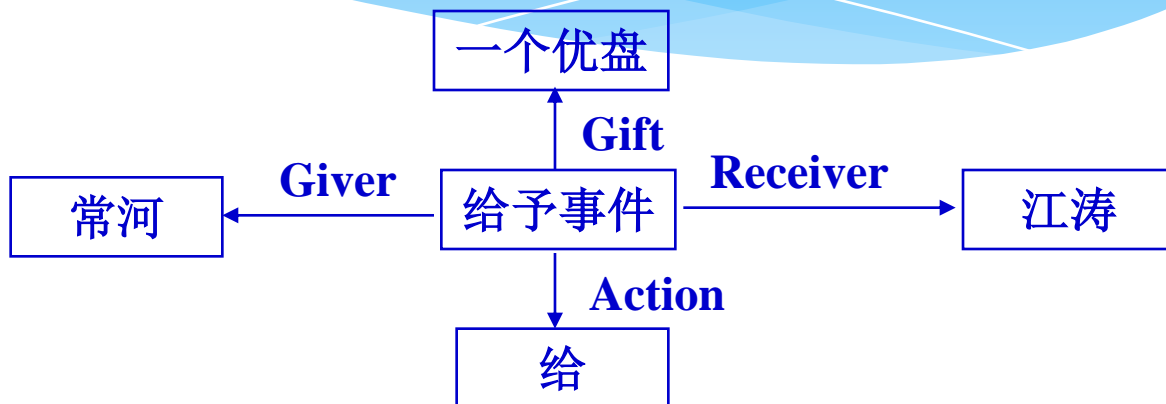


图2.20 带有事件节点的语义网络

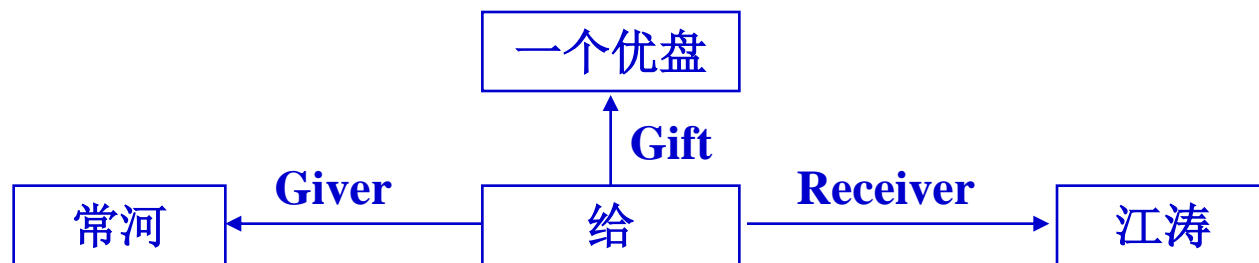


图2.21 带有动作节点的语义网络

5. 语义网络表示的特征

主要优点：

结构性 把事物的属性以及事物间的各种语义联系显式地表示出来，是一种结构化的知识表示方法。在这种方法中，下层结点可以继承、新增、变异上层结点的属性。

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。

自然性 语义网络可以比较直观把知识表示出来，符合人们表达事物间关系的习惯。

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是多种多样的，这虽然对其表示带来了灵活性，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。

1. 框架理论

框架理论是明斯基于1975年作为理解视觉、自然语言对话及其它复杂行为的一种基础提出来的。

它认为人们对现实世界中各种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，从而形成对这个新事物的认识。例如，对饭店、教室等的认识。

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，把一组有关的框架连结起来便可形成一个框架系统。

框架系统推理：由框架之间的协调来完成。

2. 框架结构和框架表示

框架的基本结构(1/2)

<框架名>

槽名1:	侧面名 1_1	值 1_{11} , 值 1_{12} , ...
	侧面名 1_2	值 1_{21} , 值 1_{22} , ...
	:	
槽名2:	侧面名 2_1	值 2_{11} , 值 2_{12} , ...
	侧面名 2_2	值 2_{21} , 值 2_{22} , ...
	:	
:		
:		
槽名n:	侧面名 n_1	值 n_{11} , 值 n_{12} , ...
	侧面名 n_2	值 n_{21} , 值 n_{22} , ...
	:	
	侧面名 n_m	值 n_{m1} , 值 n_{m2} , ...

2.5.2 框架结构和框架表示

框架的基本结构(2/2)

例2.14 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. 框架结构和框架表示

框架表示(1/3)

当知识结构比较复杂时，往往需要用多个相互联系的框架来表示。例如，对前面例2.14的硕士生框架“MASTER”可分为：

“Student”框架，描述所有学生的共性，上层框架

“Master”框架，描述硕士生的个性，子框架，继承“Student”框架的属性
学生框架

Frame <Student>

Name: Unit (Last-name, First-name)

Sex: Area (male, female)

Default: male

//缺省

Age: Unit (Years)

If-Needed: Ask-Age

//询问赋值

Address: < S-Address>

Telephone: Home Unit (Number)

Mobile Unit (Number)

If-Needed: Ask-Telephone

//询问赋值

2. 框架结构和框架表示

框架表示(2/3)

硕士生框架

Frame <Master>

AKO: <Student>

//预定义槽名

Major: Unit (Major)

//专业

If-Needed: Ask- Major

//询问赋值

If-Added: Check-Major

//后继处理

Field: Unit (Direction-Name)

//方向

If- Needed: Ask – Field

//询问赋值

Advisor: Unit (Last-name, First-name)

//导师

If- Needed: Ask -Visor

//询问赋值

Project : Area (National, Provincial, Other)

//项目

Default: National

//缺省

Paper: Area (SCI, EI, Core, General)

//论文

Default: Core

//缺省

这里，用到了一个系统**预定义槽名AKO**，其含义为“是一种”。

当AKO作为下层框架的槽名时，其槽值为上层框架的框架名，表示该下层框架所描述的事物比其上层框架更具体。并且，由AKO所联系的框架之间具有属性的继承关系。

2. 框架结构和框架表示

框架表示(3/3)

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。

2. 框架结构和框架表示

实例框架

例如，有杨叶和柳青2个硕士生，将他们的情况分别添入Master，会可得到：
硕士生-1框架：

Frame <Master-1>

ISA: <Master>

//是一个

Name: Yang Ye

Sex: female

Major: Computer

Field: Web-Intelligence

//方向 Web智能

Advisor: Lin Hai

//导师 林海

Project : Provincial

//项目 省部级

硕士生-2框架：

Frame <Master-2>

ISA: <Master>

Name: Liu Qing

Age: 22

Major: Computer

Advisor: Lin Hai

Paper: EI

//论文 EI收录

其中用到了系统预定以槽名ISA，即Master-1和Master-2是2个具体的Master。

3. 框架系统

框架之间的联系

框架系统由框架之间的横向或纵向联系构成。

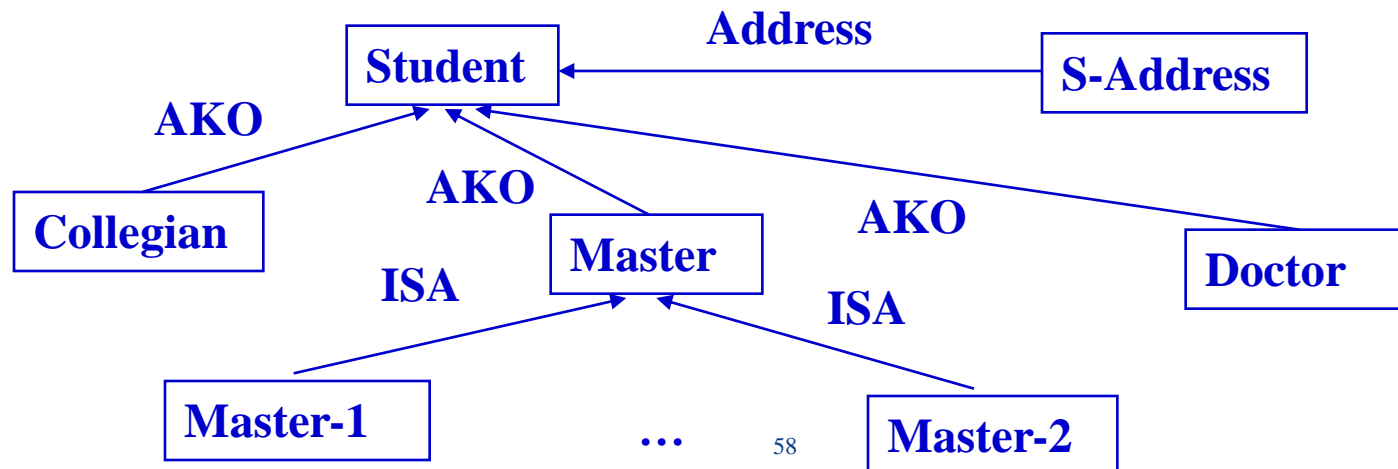
纵向联系

是指那种具有继承关系的上下层框架之间的联系。如下图，学生可按照接受教育的层次分为本生、硕和博。每类学生又可按照所学专业的不同划分。

纵向联系通过预定以槽名AKO和ISA等来实现。

横向联系

是指那种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。如下图中Student框架与S-Addre框架之间就是一种横向联系。



4. 框架系统问题求解的基本过程

特性继承(1/2)

特性继承过程

通过ISA、AKO链来实现。

当需要查询某一事物的某个属性，且描述该事物的框架未提供其属性值时，系统就沿ISA和AKO链追溯到具有相同槽的类或超类框架。

如果该槽提供有Default侧面值，就继承该默认值作为查询结果返回。

如果该槽提供有If-Needed侧面供继承，则执行If-Needed操作，去产生一个值作为查询结果。

如果对某个事物的某一属性进行了赋值或修改操作，则系统会自动沿ISA和AKO链追溯到具有相应的类或超类框架，去执行If-Added操作，作相应的后继处理。

If-Needed与If-Added过程的区别

它们的主要区别在于激活时机和操作目的不同。

If- Needed操作是在系统试图查询某个事物框架中未记载的属性值时激活，并根据查询需求，被动地即时产生所需要的属性值；

If-Added操作是在系统对某个事务框架的属性作赋值或修改工作后激活，目的在于通过这些后继处理，主动做好配套操作，以消除可能存在的_{不一致}。

4. 框架系统问题求解的基本过程

特性继承(2/2)

特性继承的例

如前面的学生框架

若要查询Master-1的Sex，则可直接回答；但要查询Master-2的Sex，则需要沿ISA链和AKO链到Student框架取其默认值male。

若要查询Master-2的Field，需要沿ISA链到Master框架，执行Field槽If-Needed侧面的Ask-Field操作，即时产生一个值，假设产生的值是Data-Mining，则表示Master-2的研究方向为数据挖掘。

如果要修改Master-2的Major，需要沿ISA链到Master框架，执行Major槽If-Added侧面的Check-Major操作，对Field、Advisor进行修改，以保持知识的一致性。

知识表示方法比较

表1 几种知识表示方法的比较

	逻辑	产生式	框架	语义网络
关系的描述	良好	一般	差	一般
阶层性	良好	差	良好	良好
概念和实体的 分离性	差	差	良好	差

本章小结

- * 知识表示的作用
- * 选择适当的表示方法
- * 知识的外部表示和内部表示
- * 知识表示与知识利用的关系
- * 参考章节：第2章