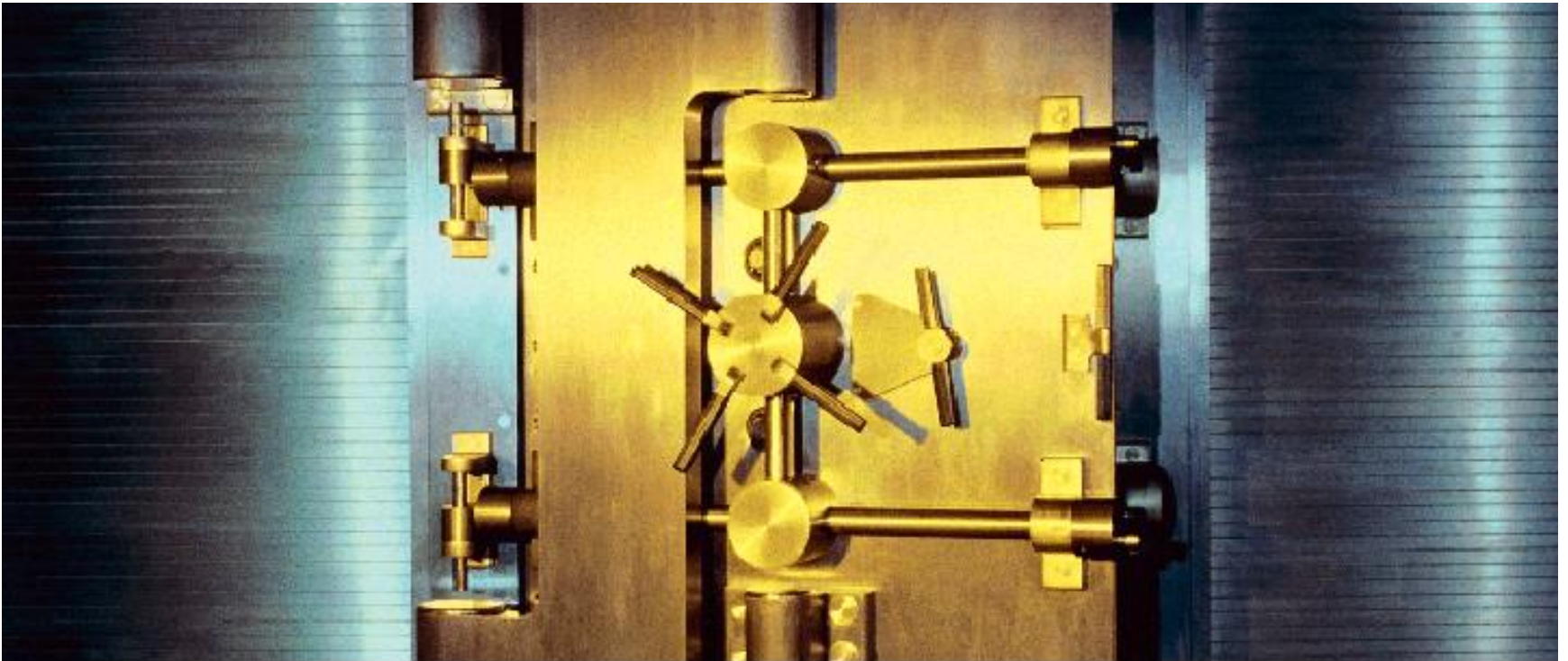


第2章：知识表示方法

Knowledge Representation



李磊

March 16, 2020

第2章 知识表示方法

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

2.6 知识的其它表示法

2.1 知识与知识表示的概念

知识就是力量！

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。

2.1.1 知识

1.什么是知识？

- 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**微观方法**：如步骤、操作、规则、过程、技巧等。**宏观方法**：如战略、战术、计谋、策略等。

知识的有代表性的定义

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

2.1.1 知识

- (4) <http://en.wikipedia.org/wiki/Knowledge>: Knowledge is a collection of facts, information, and/or skills acquired through experience or education or (more generally) the theoretical or practical understanding of a subject. It can be **implicit** (as with practical skill or expertise) or **explicit** (as with the theoretical understanding of a subject); and it can be more or less **formal** or **systematic**.

2.1.1 知识

2.数据、信息、知识及其关系

- 数据是信息的载体，本身无确切含义，其关联构成信息。
- 信息是数据的关联，赋予数据特定的含义，仅可理解为描述性知识。
- 知识可以是对信息的关联，也可以是对已有知识的再认识
 - 常用的关联方式：if then
 - 知识在信息的基础上增加了**上下文信息**，提供了更多的意义因此也就更加有用和有价值。
 - 知识是随着时间的变化而**动态变化**的，新的知识可以根据规则和已有的知识推导出来。
 - 因此可以认为知识是**经过加工**的信息，它包括事实、信念和启发式规则。
 - 关于知识的研究称为认识论（Epistemology），它涉及知识的本质、结构和起源。

2.1.1 知识

3.知识的类型

(1) 按知识的性质

- 概念、命题、公理、定理、规则和方法

(2) 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。

(3) 按知识的作用效果

- **事实性知识**：用于描述事物的概念、定义、属性等；
- **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；
例如：推理策略、搜索策略、不确定性的传播策略。

2.1.1 知识

(4) 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

(5) 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
表示：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：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

(6) 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

4.知识的特性

- 相对正确性
 - 在一定的条件及环境下，知识一般是正确的，可信任的
- 不确定性
 - 由随机性引起的不确定性
 - 由模糊性引起的不确定性
 - 由不完全性引起的不确定性
 - 由经验性引起的不确定性
- 可表示性和可利用性
 - 知识是可以表示出来的
 - 知识是可以利用的
- 相对稳定和持久性

2.1.1 知识

5.人工智能系统所关心的知识

- 一个Agent需要什么样的知识才可能具有智能呢？一般说来至少包括下面几个方面的知识
- 事实：
是关于对象和物体的知识。
- 规则：
是有关问题中与事物的行动、动作相联系的因果关系的知识，是动态的，常以“如果……那么……”形式出现。
- 元知识：
是有关知识的知识，是知识库中的高层知识。例如包括怎样使用规则、解释规则、校验规则、解释程序结构等知识。
- 常识性知识：
泛指普遍存在而普遍认识了客观事实一类知识，即指人们共有的知识。

2.1.2 知识表示的概念

1. 什么是知识表示

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。其表示方法不唯一。

- 知识表示的要求
 - **表示能力**：能否正确、有效地表示问题。包括：
 - 表示范围的广泛性
 - 领域知识表示的高效性
 - 对非确定性知识表示的支持程度
 - **可利用性**：可利用这些知识进行有效推理。包括：
 - 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
 - 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

2.知识表示的两种基本观点

- 叙述性的观点：

将知识的表示和知识的运用分开处理，在知识表示时不涉及如何运用知识的问题。

- 过程性的观点：

将知识的表示和知识的运用结合起来，知识包含于程序之中。

2.1.2 知识表示的概念

3.知识表示方法

(1)非结构化方法

- 谓词逻辑表示法
- 产生式系统表示法
- 过程表示法
- ...

(2)结构化方法

- 语义网络表示法
- 框架表示法
- ...

2.2 谓词逻辑表示法

一阶谓词逻辑表示法是一种基于数理逻辑的表示方法。

数理逻辑是一门研究推理的学科。可分为：

- **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。

本节主要讨论

- 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 一阶谓词的逻辑学基础

1. 命题 (proposition)

命题的定义：

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。

2. 命题的真值：

T：表示命题的意义为真

F：表示命题的意义为假

- 命题真值的说明

- 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
 - 薛定谔的猫

2.2.1—一阶谓词的逻辑学基础

3.论域：

由所讨论对象的全体构成的集合。亦称为个体域。（类似定义域）。

- 个体：论域中的元素。

4.谓词：

在谓词逻辑中命题是用形如 $P(x_1, x_2, \dots, x_n)$ 的谓词来表示，其中：

- **谓词名**：是命题的谓语，表示个体的性质、状态或个体之间的关系
- **个体**：是命题的主语，表示独立存在的事物或概念

2.2.1一阶谓词的逻辑学基础

- 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($x, 6$)

x 大于6

TEACHER(father(Wang Hong)) 王宏的父亲是一位教师

2.2.1—一阶谓词的逻辑学基础

5.函数：

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 一阶谓词的逻辑学基础

6. 谓词逻辑组成

- 谓词符号
- 变量符号
- 常量符号
- 函数符号
- 其他辅助符号——括号、标点符号等。

例：

- (1) 机器人(Robot)在1号房间(Room1)里。
INROOM(Robot, Room1) ——INROOM,谓词符号； Robot, Room1常量符号。
INROOM(x,y) ——x,y变量符号。
- (2) 李的父亲和母亲结婚。
MARRIED[father(Li), Mother(Li)]
MARRIED——谓词符号； father(Li), Mother(Li)——函数。

2.2.2 连接词

1.合取(与)

- 符号： \wedge
- 定义： $A \wedge B$ 为真，当且仅当A、B同时为真。
- 例子：

- ① 我喜欢音乐和绘画。

LIKE (I, Music) \wedge LIKE(I, Painting)——同时性

- ② 李住在一幢黄色的房子里。

LIVES(Li, House1) \wedge COLOR(House1, Yellow)——同时性

2.2.2 连接词

2.析取（或）

- 符号： \vee
- 定义： $A \vee B$ 为假，当且仅当 A 、 B 同时为假。
- 例子：李明打篮球或踢足球。

PLAYS (LiMing,Basketball) \vee PLAYS (LiMing,Football)

- 排斥或(异或)：一种可能发生，另一种可能性不可能发生的情形。例如：
 - 李现在在教室，或在实验室。
——李在教室，就不可能在实验室；反之亦然。

2.2.2 连接词

3.否定（非）

- 符号： \neg 、 \sim
- 定义： $\neg A$ 为真，当且仅当A为假。
- 例子：机器人不在2号房间里。
 - $\neg \text{INROOM}(\text{Robot}, \text{Room2})$
- 排斥或例子：李现在在教室，或在操场。

$[\text{IN}(\text{Li}, \text{Classroom}) \wedge \neg \text{IN}(\text{Li}, \text{Field})]$

\vee

$[\neg \text{IN}(\text{Li}, \text{Classroom}) \wedge \text{IN}(\text{Li}, \text{Field})]$

----李在教室，不在操场；李在操场，不在教室

2.2.2 连接词

4. 蕴涵

- 符号： \rightarrow ， \Rightarrow
- 定义： $A \rightarrow B$ 为假，当且仅当A为真，B为假。
- 等价式： $\neg A \vee B$
- 表达语义：如果A，则B。
- 例子：如果刘华跑得最快，那么他取得冠军。
 - $\text{RUNS}(\text{LiuHua}, \text{Fastest}) \rightarrow \text{WINS}(\text{LiuHua}, \text{Champion})$

连词的优先级（不加括号）

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

2.2.3 约束量词

约束量词，简称为**量词**，用来约束逻辑变元在其论域中的取值方式。

1. 全称量词

- 符号： \forall
- 含义：表达变量 x 在其论域内的所有可能取值。意思是“所有的”、“任一个”。记为： $(\forall x)$
 - 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
- 例子：
 - ① 对所有的 x ， $P(x)$ 为真。—— $(\forall x)P(x)$
 - ② 所有的狗都有尾巴。
 $(\forall x)[\text{DOG}(x) \rightarrow \text{HAVE}(x, \text{Tail})]$

2.2.3 约束量词

2.存在量词

- 符号： \exists
- 含义：表达变量 x 在其论域内，存在着、有一个、或至少有一个取值，使 $P(x)$ 为真。记为： $(\exists x)$
 - 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
- 例子：
 - ① 至少有一个 x ，是 $P(x)$ 为真。—— $(\exists x)P(x)$
 - ② 1号房间内有一个物体。
 $(\exists x)INROOM(x, Room1)$
 - ③ 有的狗是黄色的。（与全称量词比较）
 $(\exists x)[DOG(x) \wedge COLOR(x, Yellow)]$

2.2.4 谓词公式

1.项 (Item)

- (1) 个体常量、个体变量是项；
- (2) 若 t_1, t_2, \dots, t_n 是项， f 是 n 元函数，则 $f(t_1, t_2, \dots, t_n)$ 是项；
- (3) 由(1)、(2)生成的表达式是项。
- 个体常量、个体变量和函数统称为项。

2.原子谓词公式 (原子公式)

- 若 t_1, t_2, \dots, t_n 是项， P 是谓词，则称 $P(t_1, t_2, \dots, t_n)$ 为原子谓词公式。

2.2.4 谓词公式

3.谓词公式（合式）

- (1) 原子公式是谓词公式；
- (2) 若A是谓词公式，则 $\neg A$ 也是谓词公式；
- (3) 若A,B是谓词公式，则 $A \vee B$ ， $A \wedge B$ ， $A \rightarrow B$ 也都是谓词公式；
- (4) 若A是谓词公式，x是项，则 $(\forall x)A(x)$ 和 $(\exists x)A(x)$ 都是谓词公式。
- (5) 按（1）—（4）规则构造的表达式是谓词公式。
- 例如：
 $\neg P(x,y) \vee Q(y)$ ， $(\forall x)(A(x) \rightarrow B(x))$ ，都是谓词公式。

2.2.4 谓词公式

4.量词的辖域

- 量词的约束范围，即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。

- 例：

$$(\forall x)\{P(x) \rightarrow \{(\forall y)[P(y) \rightarrow P(f(x,y))] \wedge \neg(\exists y)[Q(x,y) \rightarrow P(y)]\}\}$$

- $(\forall x)$ 约束整个公式，即公式中 x 都受全称量词约束；
- $(\forall y)$ 和 $(\exists y)$ 只能约束相应中括号内的 y 变元。

2.2.4 谓词公式

5.约束变元:

- 受到量词约束的变元, 即辖域内与量词中同名的变元称为约束变元

6.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

2.2.4 谓词公式

7.变元的换名：

谓词公式中的变元可以换名。要保持变量的论域不变。此外还需注意：

- 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 y 不能换成 x 。
- 存在量词约束的变量换名同上。

2.2.4 谓词公式

8.谓词公式真值表

取出公式中所有单个谓词，按所有可能的取值组合，再按连接词和量词的定义给出合适公式的真值。

• 例：

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

2.2.5 谓词公式性质

谓词公式的等价公式

- 两个谓词公式，无论如何解释，若两者的真值表相同，则两公式为等价谓词公式。

1. 否定之否定(=肯定)

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

2. 逆否律

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

3. 德.摩根定律(De.Morgen's Law)

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

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

4. 分配率

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

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

2.2.5 谓词公式性质

5. 交换率

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

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

6. 结合率

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

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

7. 逆否率

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

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

2.2.5 谓词公式性质

8. 否定移过量词

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

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

验证：假定 x 的论域为非空有限集合，

即 $x=\{a_1, a_2, \dots, a_n\}$

$$\neg(\forall x)P(x) \Leftrightarrow \neg\{P(a_1) \wedge P(a_2) \wedge \dots \wedge P(a_n)\}$$

$$\Leftrightarrow \neg P(a_1) \vee \neg P(a_2) \vee \dots \vee \neg P(a_n)$$

$$\Leftrightarrow (\exists x)[\neg P(x)]$$

假定否定符号无论移过多少个约束量词，每个都要相应改变。即：全称变为存在，存在变为全称。例：

- $\neg(\forall x)(\exists y)(\forall z)P(x,y,z) \Leftrightarrow (\exists x)(\forall y)(\exists z)\neg P(x,y,z)$

2.2.5 谓词公式性质

9. 约束量词移进

$$(\forall x)[P(x) \wedge Q(x)] \Leftrightarrow (\forall x)P(x) \wedge (\forall x)Q(x)$$

$$(\exists x)[P(x) \vee Q(x)] \Leftrightarrow (\exists x)P(x) \vee (\exists x)Q(x)$$

10. 变量换名（论域不变、名字不冲突）

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

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

2.2.6 谓词逻辑表示方法

谓词逻辑表示步骤：

- (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2.2.1 表示知识“所有教师都有自己的学生”。

- 解：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。

- 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，那么一定存在一个体y，y的老师是x，且y是一个学生。

2.2.6 谓词逻辑表示方法

例2.2.2 表示知识“所有的整数不是偶数就是奇数”。

- 解：定义谓词：I(x)：x是整数，E(x)：x是偶数，O(x)：x是奇数
- 表示知识：(排斥或、异或)

$$(\forall x) \{ I(x) \rightarrow [E(x) \wedge \neg O(x)] \vee [\neg E(x) \wedge O(x)] \}$$

- 类似问题：任何整数或为正数，或为负数。

2.2.6 谓词逻辑表示方法

例2.2.3 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：定义谓词：

COMPUTER(x)：表示x是计算机系的学生。

CLASSMATE(x,y)：表示x和y是同班同学。

LIKE(x,y)：表示x喜欢y。

- 表示知识：

COMPUTER(Wang Hong)

CLASSMATE(Wang Hong, Li Ming)

$(\forall x)[\text{COMPUTER}(x) \rightarrow \text{LIKE}(x, \text{programming})]$

2.2.6 谓词逻辑表示方法

例2.2.4 用谓词公式表示：For every set x, there is a set y, such that the cardinality of y is greater than the cardinality of x.

- 解：定义谓词：

SET(x) — x is a set.

CARD(x,y)— y is the cardinality of x, and x is a set.

GREAT(x,y) — x is greater than y.

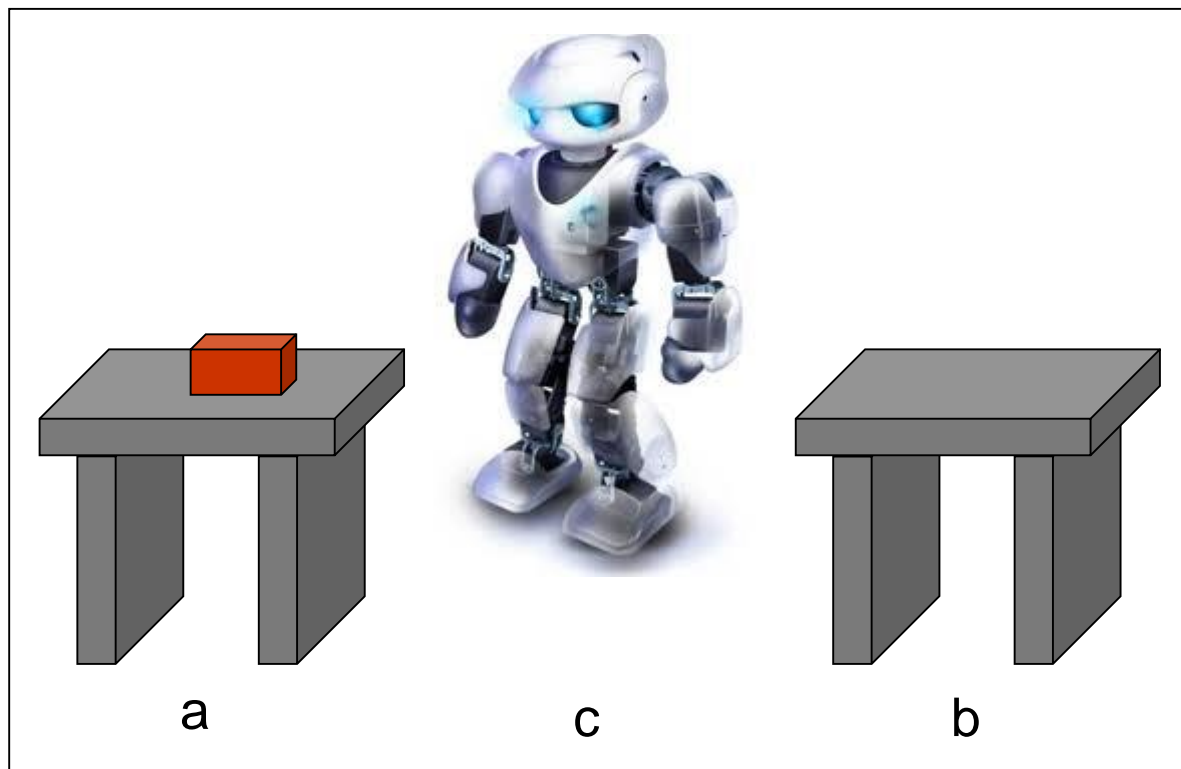
- 表示知识：

$(\forall x)\{SET(x) \rightarrow (\exists y)(\exists u)(\exists v)[SET(y) \wedge CARD(x,u) \wedge CARD(y,v) \wedge GREAT(v,u)]\}$

2.2.7 谓词逻辑表示的应用

1. 机器人搬箱子

- 机器人robot，箱子box，2个桌子a和b。初始状态机器人位于c处，空手，box在a上。目标：机器人把位于a上的box搬运到b上。
- 解：



分别定义描述状态和动作的谓词

定义描述状态的谓词：

- $\text{TABLE}(x)$: x 是桌子
- $\text{EMPTY}(y)$: y 手中是空的
- $\text{AT}(y, z)$: y 在 z 处
- $\text{HOLDS}(y, w)$: y 拿着 w
- $\text{ON}(w, x)$: w 在 x 桌面上
- 变元的个体域（论域）：
 - x 的个体域是 $\{a, b\}$
 - y 的个体域是 $\{\text{robot}\}$
 - z 的个体域是 $\{a, b, c\}$
 - w 的个体域是 $\{\text{box}\}$

- 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

定义描述操作的谓词：

- 条件部分：用来说明执行该操作必须具备的先决条件，可用谓词公式来表示
- 动作部分：给出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
- 需要定义的操作：
 - Goto(x, y)：从x处走到y处。
 - Pickup(x)：在x处拿起盒子。
 - Setdown(y)：在y处放下盒子。

各操作的条件和动作：

- Goto(x, y)
 - 条件：AT(robot, x)
 - 动作：删除：AT(robot, x)
添加：AT(robot, y)

- Pickup(x)
 - 条件: ON(box, x), TABLE(x), AT(robot, x), EMPTY(robot)
 - 动作: 删除: EMPTY(robot), ON(box, x)
添加: HOLDS(robot, box)
- Setdown(x)
 - 条件: AT(robot, x), TABLE(x), HOLDS(robot, box)
 - 动作: 删除: HOLDS(robot, box)
添加: EMPTY(robot), ON(box, x)
- 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。

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)

- 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)

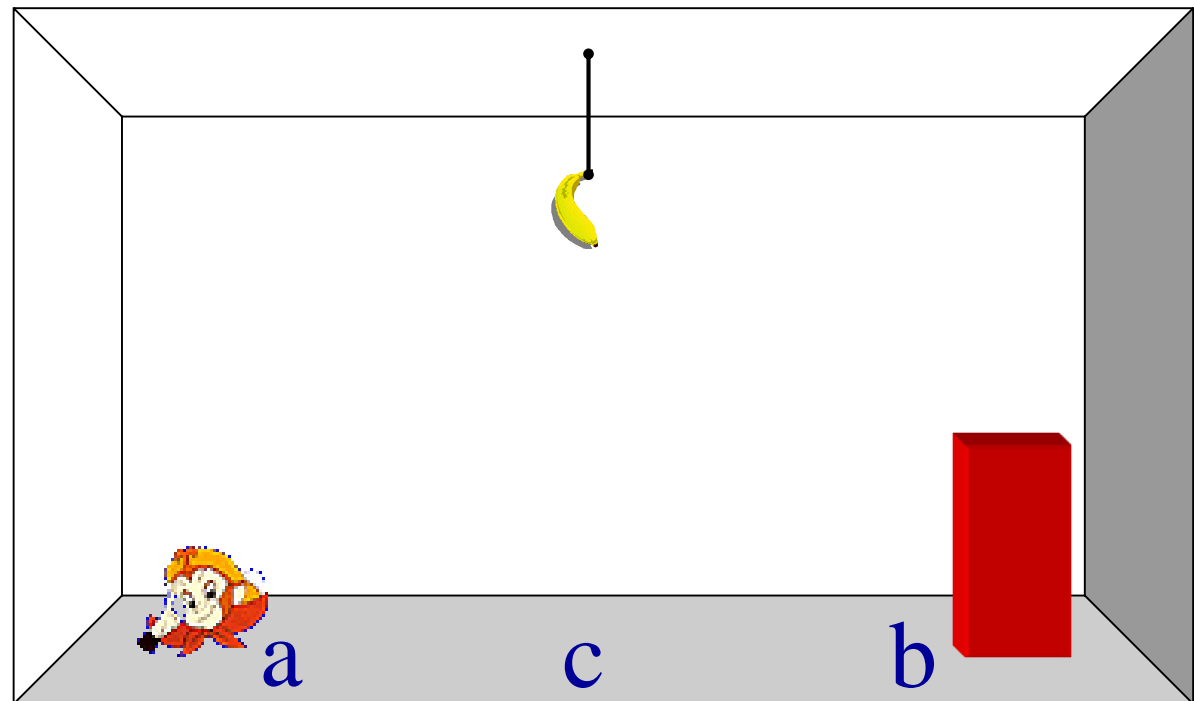
Goto(b, c)

=====>

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

2.猴子和香蕉问题 (Monkey and Banana Problem)

- 在房间的 a 处有一只猴子，b 处有一个箱子，c 处挂着一串香蕉，猴子欲想拿取香蕉，他必须首先走到 b 处，推动箱子到 c 处，然后爬上箱子才可以办到。假设猴子和箱子都只做一维运动。用谓词逻辑表示的状态空间求解此问题。
- 解：



定义描述状态的谓词:

- $AT(x, y)$: x 在 y 处
- ONBOX: 猴子在箱子上
- GB: 猴子得到香蕉
- 个体域:
 - x : {monkey, box, banana}
 - Y : {a, b, c}
- 问题的初始状态
 - $AT(monkey, a)$
 - $AT(box, b)$
 - $\neg ONBOX, \neg GB$
- 问题的目标状态
 - $AT(monkey, c), AT(box, c)$
 - ONBOX, GB

定义描述操作的谓词

- 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: $\neg \text{ONBOX}$, AT(monkey, u),
 - 动作: 删除: AT(monkey, u)
添加: AT(monkey, v)
- PushBox(v, w)
 - 条件: $\neg \text{ONBOX}$, AT(monkey, v), AT(box, v)
 - 动作: 删除: AT(monkey, v), AT(box, v)
添加: AT(monkey, w), AT(box, w)

- ClimbBox()
 - 条件: $\neg \text{ONBOX}, \text{AT}(\text{monkey}, w), \text{AT}(\text{box}, w)$
 - 动作: 删除: $\neg \text{ONBOX}$
添加: ONBOX
- Grasp()
 - 条件: $\text{ONBOX}, \text{AT}(\text{box}, c)$
 - 动作: 删除: $\neg \text{GB}$
添加: GB

逻辑表示是推理的基础，后面推理部分还将介绍更多的内容。

2.2 谓词逻辑表示法小结

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“T”与“F”，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



The Apple 1 which was sold as a do-it-yourself kit (1976, without the lovely case seen here)

March 16, 2020

2.3 产生式表示法

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。本节主要内容：

- 产生式规则
- 产生式与蕴含式的异同
- 产生式与条件语句的区别

1943年，逻辑学家 Post 首提，用于计算形式体系中的符号串替换运算。认为，任何数学或逻辑系统都可以简化为一些列规则，在规则中制定如何把一个符号串变成另一个符号串，即给定一个输入符号串通过产生式规则可产生另一符号串，而不管符号串的物理意义。

2.3 产生式表示法

1950年，Simon, Newell首次将其应用于智能系统。

1954年，Markov在字符串替换研究中进行了改进，提出了产生式的控制策略，根据规则的优先级确定其执行顺序。

1957年，Chomsky在自然语言结构的研究中提出了文法分层的概念，利用产生式规则来描述每层文法的语言生成规则，即重写规则。

1972年，Simon, Newell重新改进产生式系统，作为人类心理活动中的信息加工过程研究的基础，并用它来建立人类问题求解行为的模型。

现在，产生式系统已经发展成为基本的知识表示和问题求解方式之一。在专家系统以及人工智能其他方面得到了广泛的应用。产生式表示法具有和图灵机相同的运算能力。

2.3 产生式表示法

1. 产生式规则 (Production Rule)

形如: $P \rightarrow Q$ 或者

IF P THEN Q 或者

如果 P, 那么 Q

的语句叫做一条产生式规则。

- 其中:

- P — 叫做: 前件、前项、左边(LHS-left hand side)
它给出了该产生式可否使用的**前提条件**, 由事实的逻辑组合来构成;
- Q — 叫做: 后件、后项、右边(RHS-right hand side)
它指出当前提P满足时, 应该推导出的**结论**或应该执行的**动作**。

2.3 产生式表示法

2.产生式的含义和作用

含义：如果前提P满足，则可推出结论Q或执行Q所规定的操作。

作用：产生式规则用来表示问题领域的一般知识。

【例2.3.1】

- R_1 IF x 动物是哺乳动物，且吃肉 THEN x 动物被称为食肉动物。
- R_2 IF 是红灯 THEN 停车等待。
- R_3 IF 发烧、头痛且流鼻涕 THEN 患感冒。
 - R_1 、 R_2 和 R_3 是规则的编号。 R_1 和 R_3 的后项为结论； R_2 的后项为动作。

2.3 产生式表示法

3.产生式与逻辑蕴涵（逻辑规则）的异同：

- 逻辑规则的两边只能是命题或谓词，产生式可以是命题和谓词，也可以是其他符号串；
- 逻辑规则要满足真值表，即前后件最终只能为T或F。产生式前后件都可以是任意类型的值，产生式规则右边可能是动作，甚至是复杂的过程语句；
- 产生式可以处理不确定性，逻辑规则只能用于确定推理；
- 逻辑规则可以视为产生式规则，反之不然；
- 形式相同，产生式应用范围更广。

2.3 产生式表示法

4.产生式与条件语句的主要区别：

(1) 前项结构不同

- 产生式的前项可以是一个复杂的的结构，其值不一定为逻辑类型；
- 传统程序设计语言中if的左部仅仅是一个布尔表达式

(2) 控制流程不同

- 产生式系统中满足前提条件的规则被激活后，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 产生式表示法

5. 产生式表示实例

- 【例2.3.2】 重排九宫问题(8 number puzzle)

- 解:

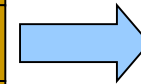
- 用二维数组 $S[i,j]$ 表示,
- 其中 $0 \leq i,j \leq 2$; 空格用0表示。
- 则: $S[i,j] \in \{0,1,2,3,4,5,6,7,8\}$
- 用 i_0 和 j_0 表示空格的下标。

- 产生式规则:

- 左移 R_1 if($j_0 \geq 1$) then $\{S[i_0,j_0]=S[i_0,j_0-1]; S[i_0,j_0-1]=0;\}$
- 右移 R_2 if($j_0 \leq 1$) then $\{S[i_0,j_0]=S[i_0,j_0+1]; S[i_0,j_0+1]=0;\}$
- 上移 R_3 if($i_0 \geq 1$) then $\{S[i_0,j_0]=S[i_0-1,j_0]; S[i_0-1,j_0]=0;\}$
- 下移 R_4 if($i_0 \leq 1$) then $\{S[i_0,j_0]=S[i_0+1,j_0]; S[i_0+1,j_0]=0;\}$

2	8	3
1	6	4
7		5

初始状态

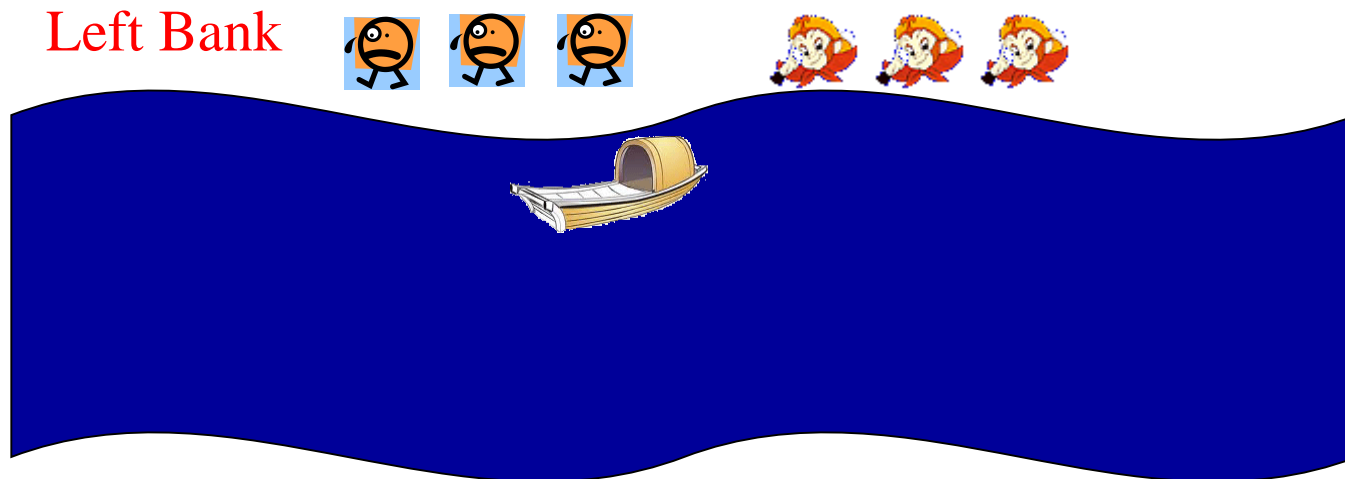


1	2	3
8		4
7	6	5

目标状态

2.3 产生式表示法

【例2.3.3】 设有3个传教士和3个野人来到一条河的左岸，打算乘一只船从左岸渡到右岸去。该船的负载能力为2人。在任何时候，如果野人人数超过传教士人数，那么野人就会把传教士吃掉。野人绝对服从传教士的指挥和调度。



2.3 产生式表示法

解：

- 考察左岸，定义三元组(M,W,B)表示左岸状态，其中：
 - M—左岸传教士人数
 - W—左岸野人数
 - B—布尔变量，船在左岸B=1，否则B=0
- 则：
 - 初始状态：(3,3,1)
 - 目标状态：(0,0,0)

2.3 产生式表示法

- 左岸往右岸运载产生式
 - L_{10} if(M,W,1) then (M-1,W,0) (左往右运1传教士)
 - L_{01} if(M,W,1) then (M,W-1,0) (左往右运1野人)
 - L_{11} if(M,W,1) then (M-1,W-1,0) (左往右运1传教士和1野人)
 - L_{20} if(M,W,1) then (M-2,W,0) (左往右运2传教士)
 - L_{02} if(M,W,1) then (M,W-2,0) (左往右运2野人)
- 右岸往左岸运载产生式
 - R_{10} if(M,W,0) then (M+1,W,1) (右往左运1传教士)
 - R_{01} if(M,W,0) then (M,W+1,1) (右往左运1野人)
 - R_{11} if(M,W,0) then (M+1,W+1,1) (右往左运1传教士和1野人)
 - R_{20} if(M,W,0) then (M+2,W,1) (右往左运2传教士)
 - R_{02} if(M,W,0) then (M,W+2,1) (右往左运2野人)

2.3 产生式表示法

5. 产生式系统特点

- 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。
 - **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。
- 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。



ANALOG COMPUTERS

2.4 语义网络表示法

Semantic Network

1968年，奎廉(J.R.Quillian)，在研究人类联想记忆时提出的一种心理学模型，认为记忆是由概念间的联系实现的。随后，奎廉又把它用作知识表示。

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。

语义网络是知识的一种结构化有向图表示方法。
具有与产生式相同的表示能力。

2.4.1 语义网络基本概念

1.什么是语义网络

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（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；
- **弧（有向边）**：关系

2.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）

3.基本网络单元

指一个语义基元对应的有向图。

- 例如：若有语义基元 (A, R, B) ，其中，A、B分别表示两个结点，R表示A与B之间的某种语义联系，则它所对应的基本网元如下图所示：



【例2.4.1】用语义网络表示“鸵鸟是一种鸟”



4.语义网络与产生式对应的表示能力

(1)事实的表示

- 例：“雪的颜色是白的”



(2)规则的表示

- 例：规则 R 的含义是 “if A then B ”



2.4.2 基本语义关系

1.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（类与对象）。例：



2.分类关系：AKO

亦称泛化关系，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：



3.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：



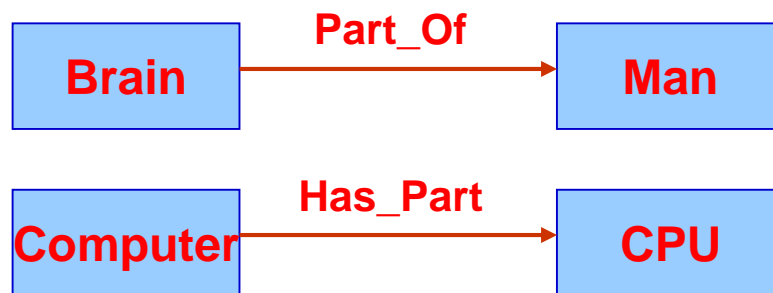
- 上述关系的主要特征—属性继承
 - 处在具体层的结点可以继承抽象层结点的所有属性。

2.4.2 基本语义关系

4. 聚集(聚类、包含)关系

表达整体与其组成部分的关系。

- 常用聚集关系——Part_Of、Has_Part
- 例：



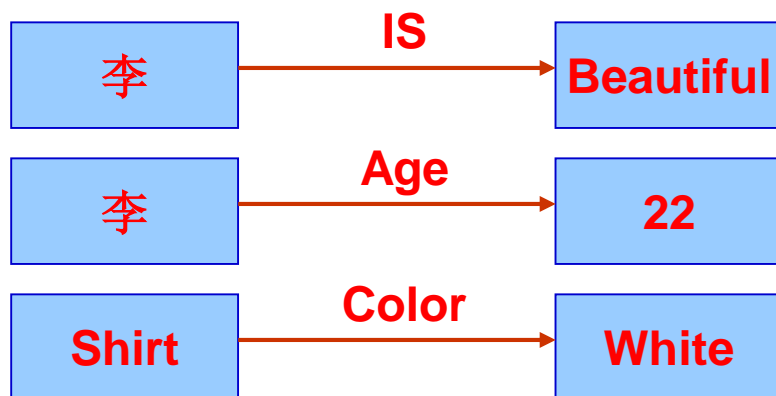
- 聚类关系与实例、分类、成员关系的主要区别：
聚类关系一般不具备属性的继承性。

2.4.2 基本语义关系

5. 属性关系

指事物和其属性之间的关系,一个结点是另一个结点的属性值。

- 常用的属性关系有: Is, Can等; 或具体的属性名词。
- 例:



2.4.2 基本语义关系

6.拥有关系——Have, Own



7.时间关系——Before、After、At等

指不同事件在其发生时间方面的先后次序关系。



8.位置关系——Located_on (_at, _inside, _outside等)



2.4.2 基本语义关系

9.相似、相近关系——Similar-to, Near-to



10.推论关系——Infer

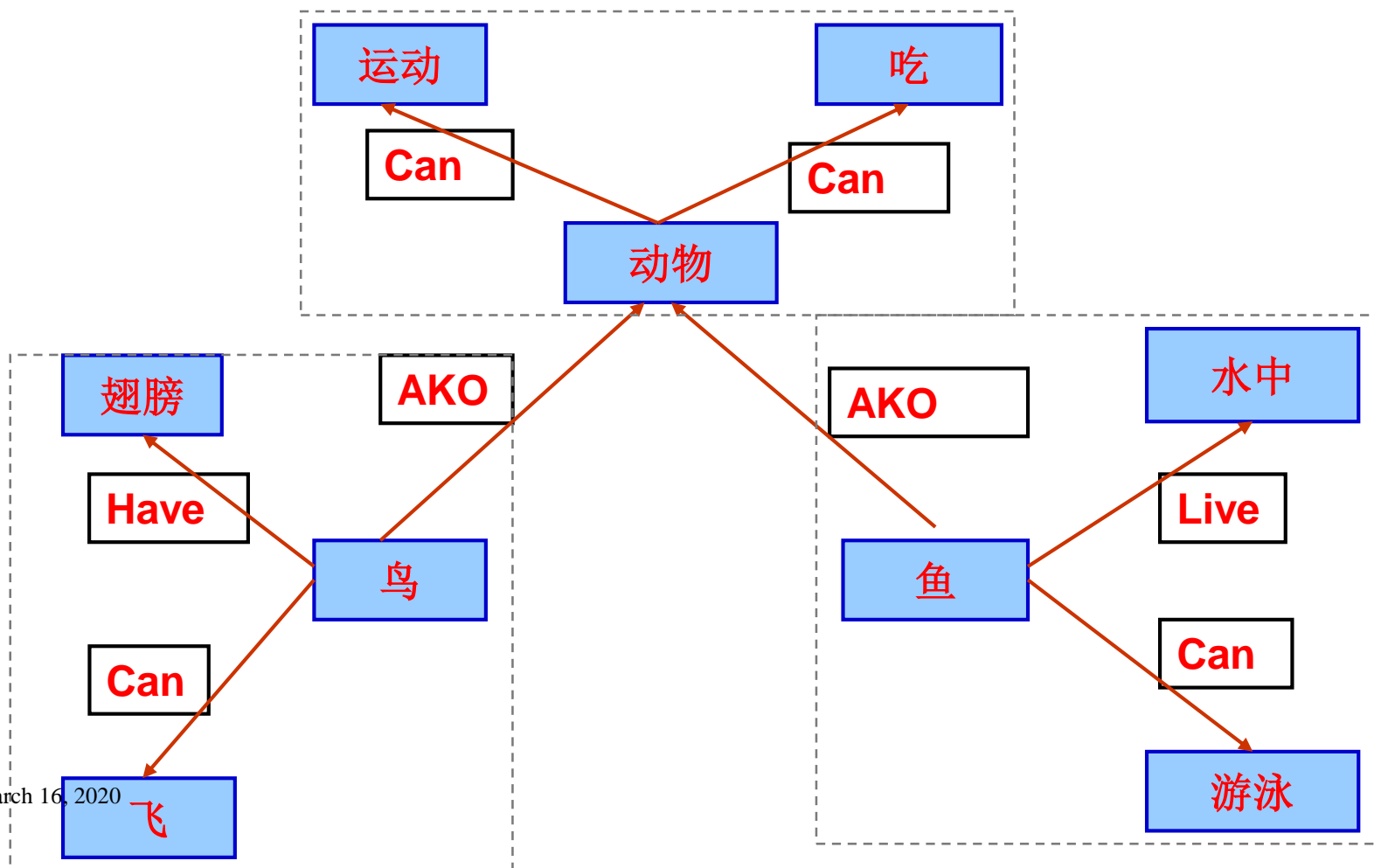


其它具体语义关系...

2.4.2 基本语义关系

【例2.4.2】用语义网络表示：

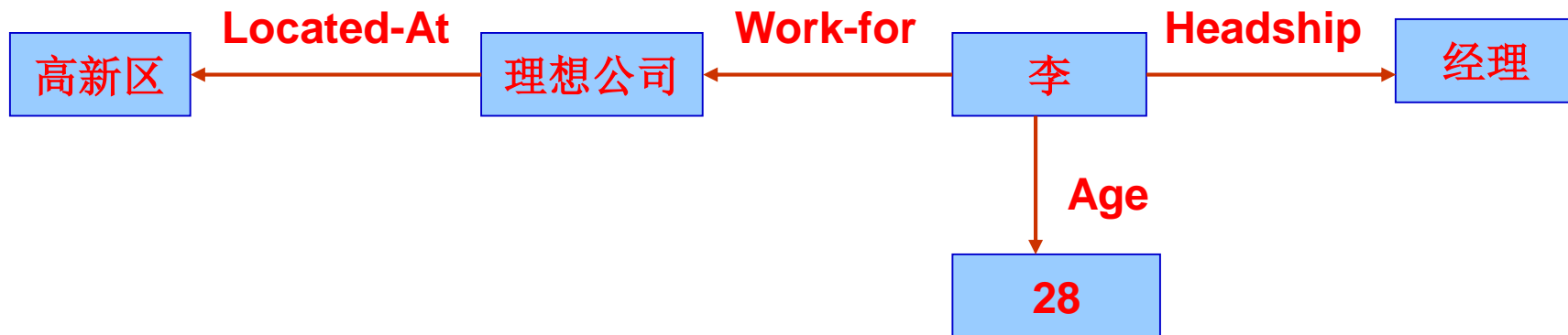
- 动物能运动、会吃。鸟是一种动物，鸟有翅膀、会飞。鱼是一种动物，鱼生活在水中、会游泳。



2.4.2 基本语义关系

【例2.4.3】 用语义网络表示：

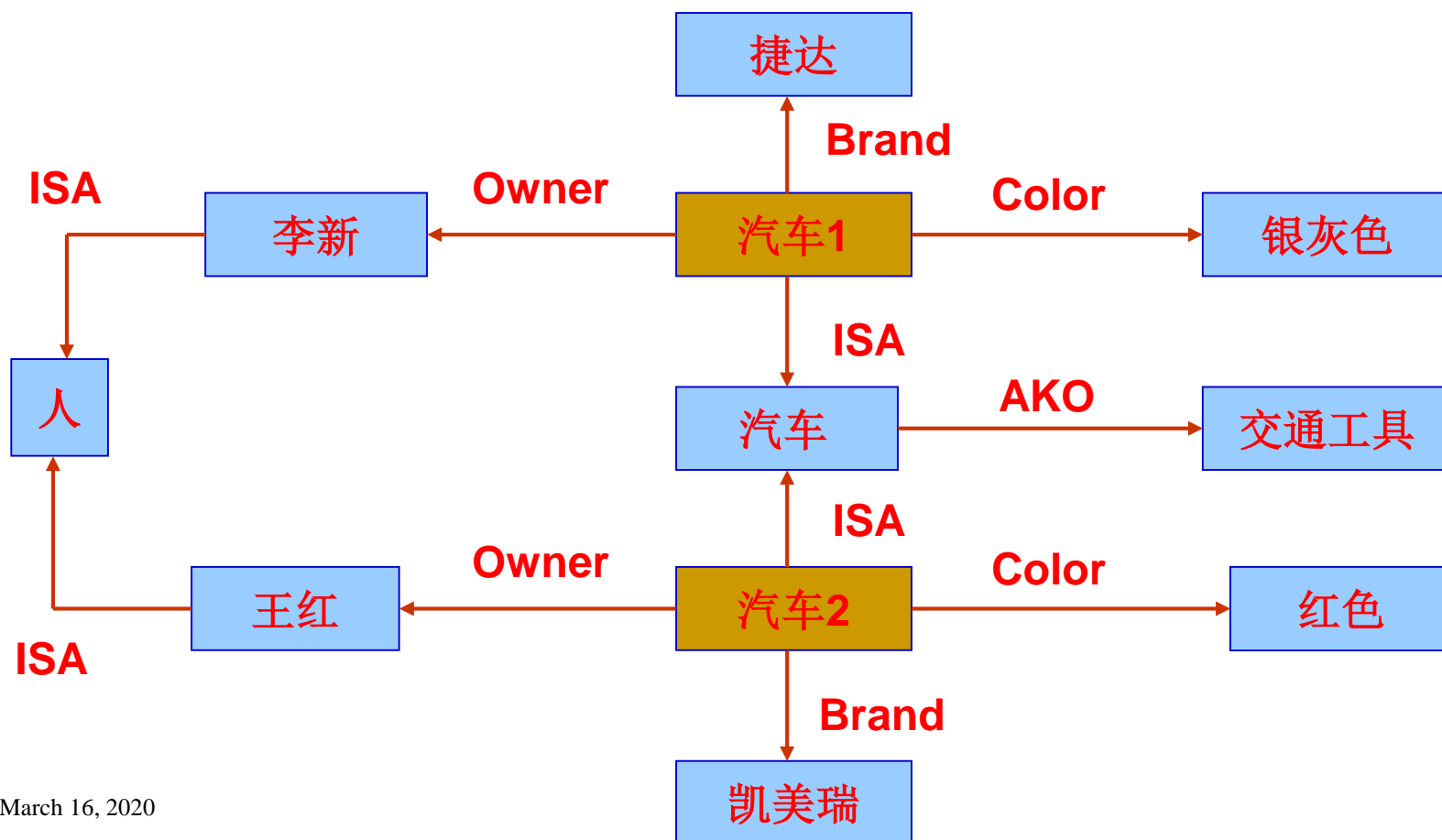
- 李是理想公司的经理；理想公司在高新区；李28岁。



2.4.2 基本语义关系

【例2.4.4】用语义网络表示：

- 李新的汽车的款式是“捷达”、银灰色。王红的汽车的款式是“凯美瑞”、红色。



2.4.3 多元语义网络

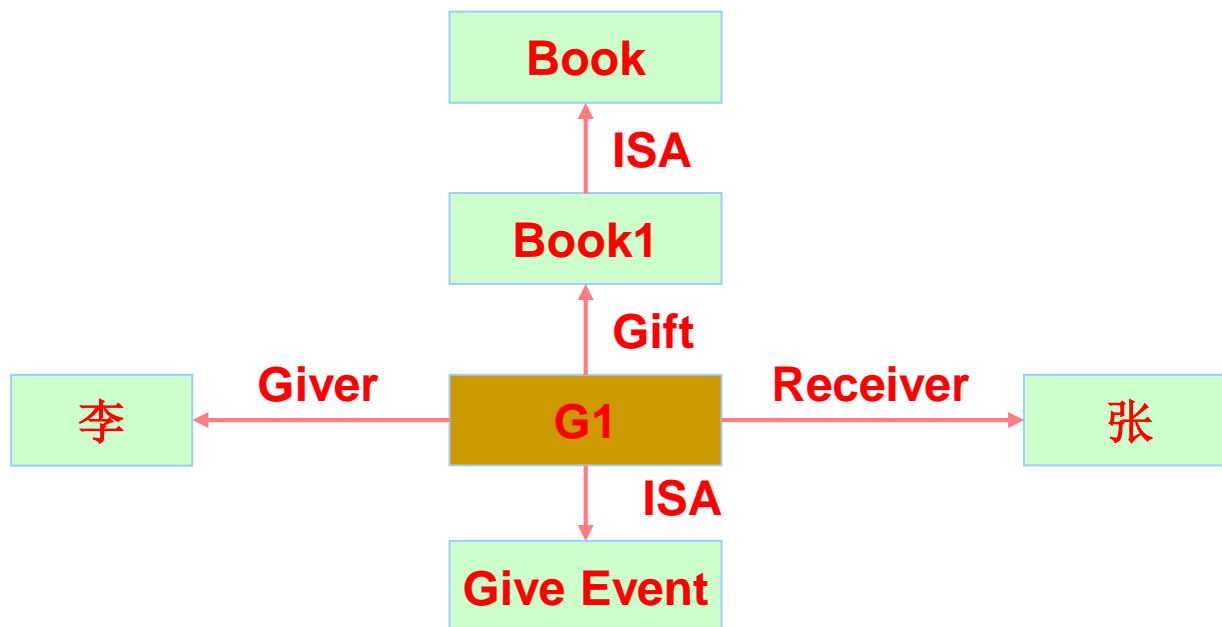
基本网络单元只能表示一元或二元语义关系，如何处理多元语义呢？

- 引入情形、事件、动作、概念等虚拟结点，将复杂的多元关系转换为多个二元关系，以表达复杂的知识。
- 此虚拟结点代表整个场景。

2.4.3 多元语义网络

【例2.4.5】 李给张一本书做为礼物。

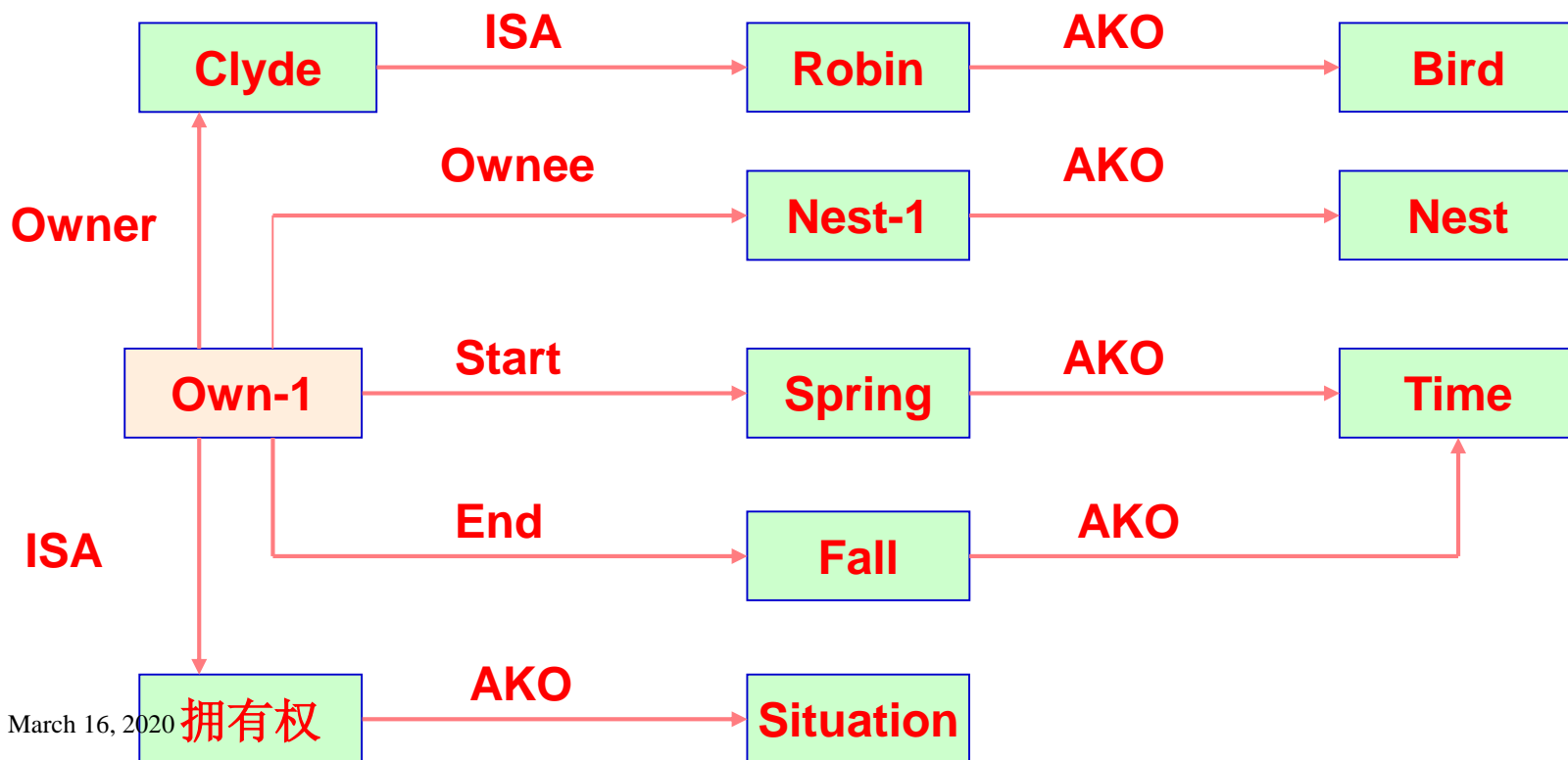
- 解：定义虚拟结点“G1”表示整个事件，将多元关系转换为对虚拟结点的二元关系。



2.4.3 多元语义网络

【例2.4.5】知更鸟 Clyde 从春天到秋天拥有一个巢。

- 解：定义虚拟结点“Own-1”表示整个情形，将多元关系转换为对虚拟结点的二元关系。



2.4.4 语义网络表示逻辑关系

对语义网络进行分区，标记网络区域的分界线、用文字标注类型情况，必要时引入情形、事件、概念节点。

用以表示逻辑中的合取、析取、否定、蕴含连接词等；以及全称和存在连词。

（详细内容略）

2.4.5 语义网络的推理

通过继承和匹配进行推理；推理能力较弱。

1. 继承

- 把事物的描述从概念节点，或类节点传递到实例节点。
- 通过继承可以得到所需结点的一些属性值；
- 继承通过ISA、AKO弧，寻找祖先结点完成。
- 继承可分为值继承、如果需要继承和缺省继承等类型。

(1) 值继承

- 从祖先结点继承属性值。

(2) If_Needed 继承（如果需要）

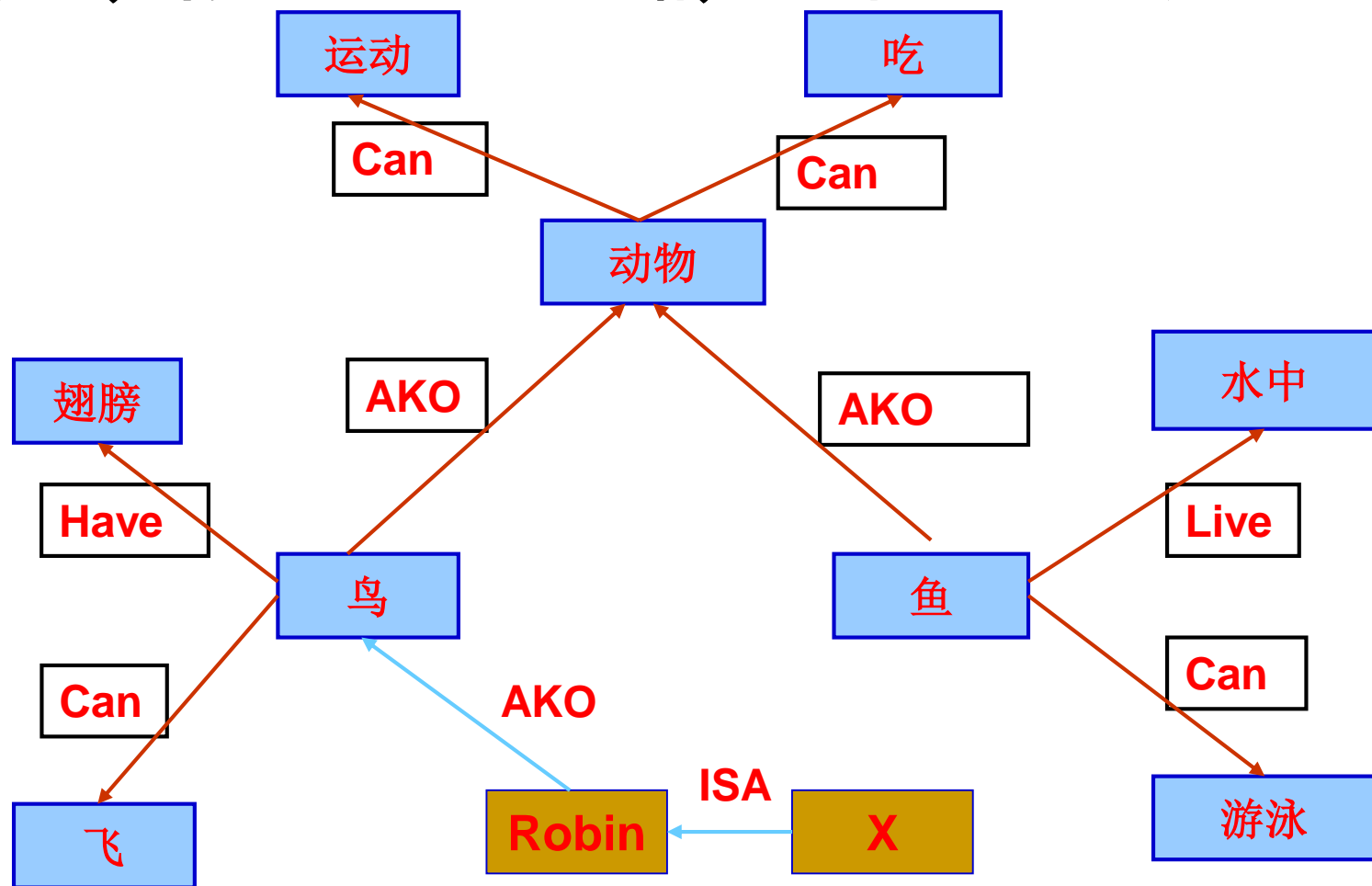
- 如果需要从祖先结点继承来调用外部过程。

(3) 缺省（默认）继承

- 从祖先结点取得默认值。与值继承类似，只是继承的值具有Default标记，作为节点的缺省、默认值，在没有反对意见的前提下，作为节点的可能取值。

2.4.5 语义网络的推理

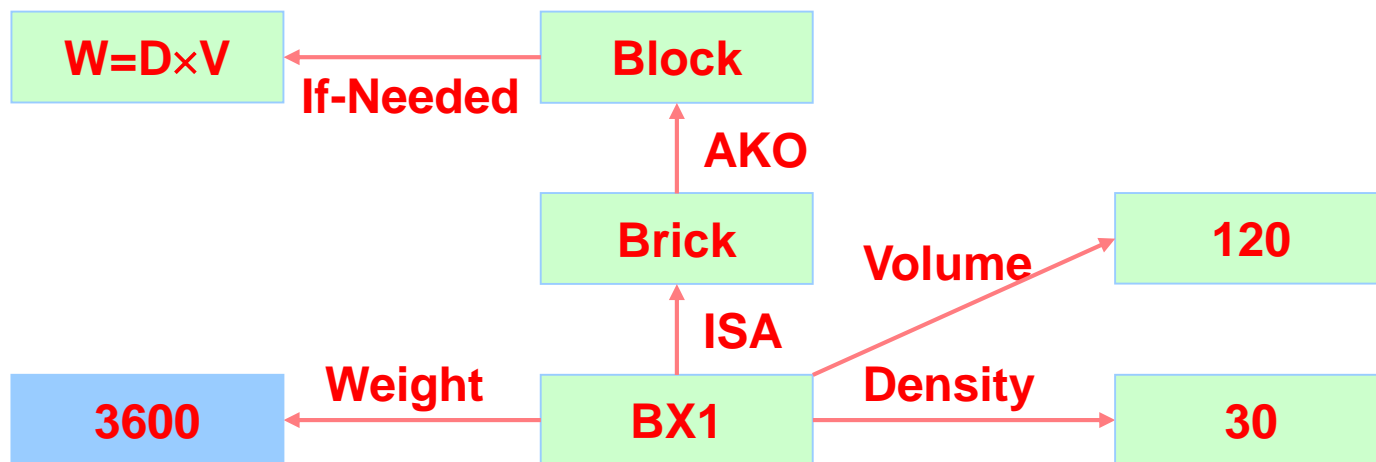
【例2.4.6】 现有一个未知物X，只知X是一只知更鸟，其它一无所知，利用例2.4.2的网络，求出X的相关属性。



通过继承可得：X有翅膀；会飞；能运动；会吃。83

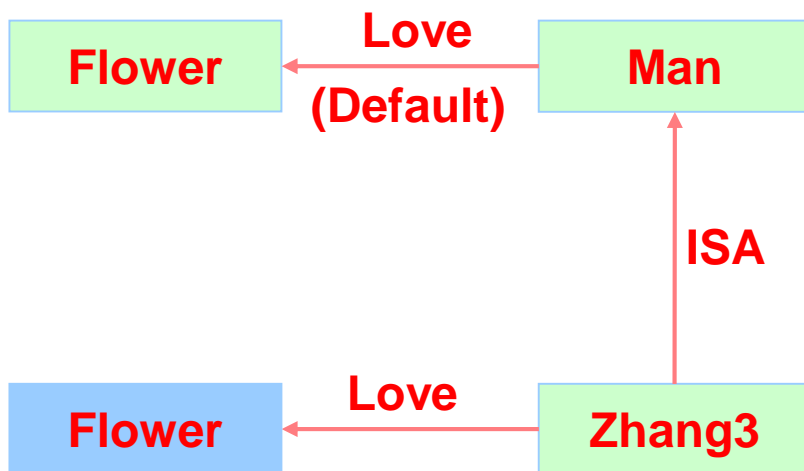
2.4.5 语义网络的推理

【例2.4.7】 通过If-Needed继承计算BX1的重量。



2.4.5 语义网络的推理

【例2.4.8】一般人都喜欢花，考虑张三这个具体的人，他可能就喜欢花，除非有明显的反对意见，如花粉过敏。



2.4.5 语义网络的推理

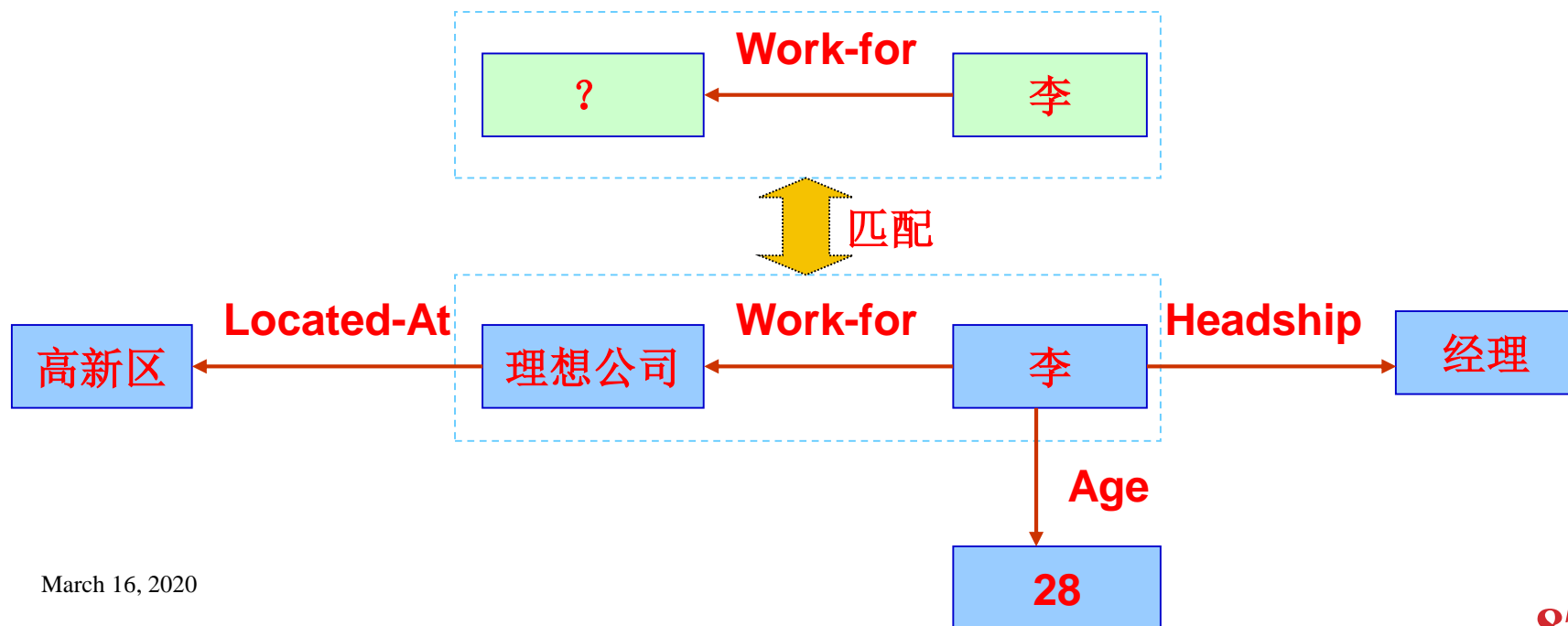
2.匹配

- 是指在知识库的语义网络中寻找与待求解问题相符的语义网络模式。
- - (1) 根据待求解问题的要求，构造一个网络片断，该网络片断中有些结点或弧的标识是空的，称为询问处，它反映的是待求解的问题。
 - (2) 根据该语义片断到知识库中去寻找所需要的信息。
 - (3) 当待求解问题的网络片断与知识库中的某语义网络片断相匹配时，则与询问处相匹配的事实就是问题的解。

2.4.5 语义网络的推理

【例2.4.9】 假设系统知识库中有例2.4.3的语义网络。问李在哪个公司工作。

- 解：根据这个问题的要求，可构造如下语义网络片断。到知识库中匹配，找到下图的匹配片段，可知李为“理想公司”工作。



2.4.5 语义网络的推理

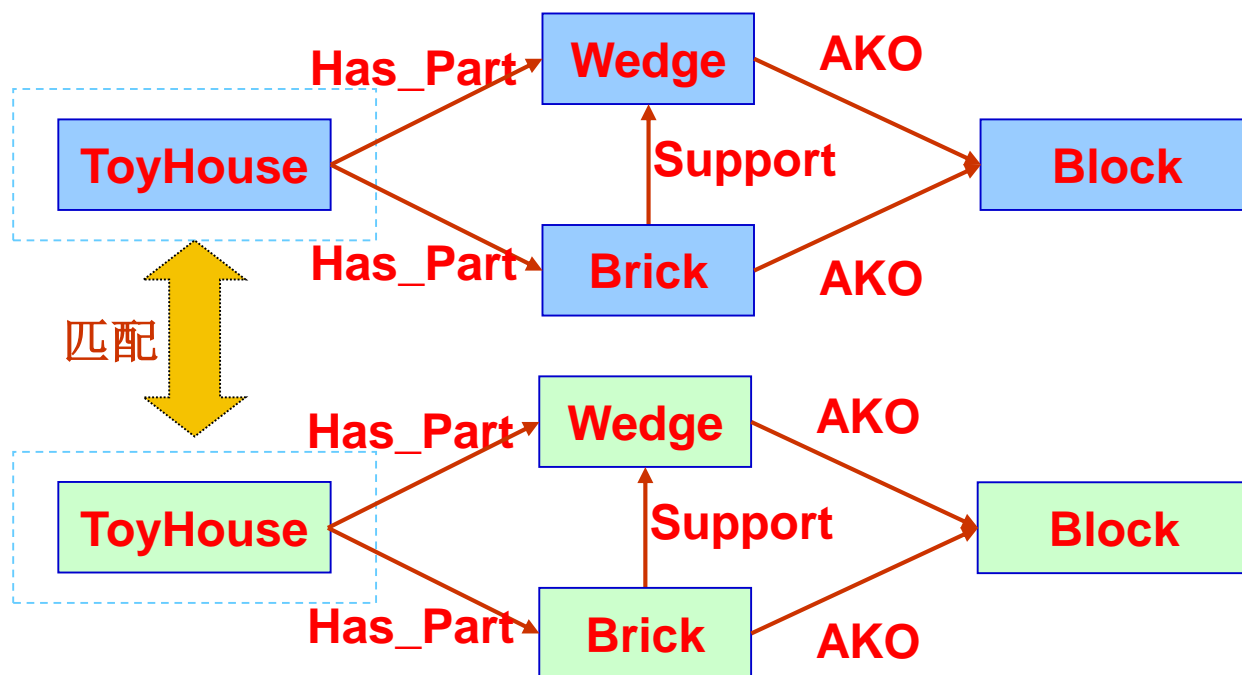
【例2.4.10】设知识库中有关于ToyHouse结构的网络，现在手头有一个ToyHouse，要猜想其结构。



2.4.5 语义网络的推理

【例2.4.10】设知识库中有关于ToyHouse结构的网络，现在手头有一个ToyHouse，要猜想其结构。

- 解：构造网络片段，到知识库中进行匹配，由已知网络推断出玩具屋的结构组成如下图。



2.4.6 语义网络表示法的特点

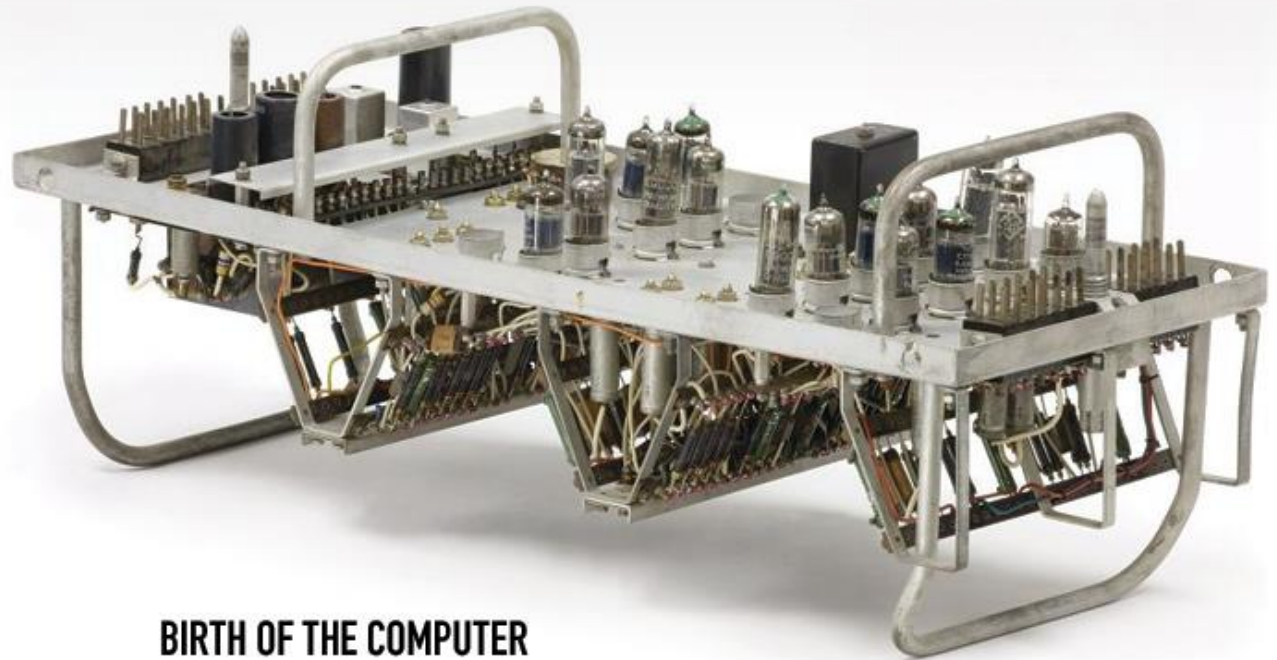
1. 主要优点:

- 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。
- 自索引性
 - 把各节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。这种自索引能力有效的避免搜索时所遇到的组合爆炸问题。
- 自然性
 - 这种带有标识的有向图，可比较直观地把知识表示出来，符合人们表达事物间关系的习惯，并且与自然语言语义网络之间的转换也比较容易实现。

2.4.6语义网络表示法的特点

2.主要缺点:

- 推理能力弱
- 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。



BIRTH OF THE COMPUTER

2.5 框架表示法

1975年，Minsky 在一篇论文 “A Framework for Representing Knowledge” 首先提出，认为人类面对的知识信息量太大，无法对所有细节都进行存储，只能以一种**结构一致的、抽象概括的数据结构**——**框架**--来存储。

- 当人们见到一个事物时，首先搜索记忆中同类、或近似事物的框架，如果先前已经遇见过，则调出记忆，给出事物的大概信息，再根据实际情况进行修正；如果是新的事物则抽象成框架结构存储在记忆中，以备后用。比如你见到**教室**，甚至是提到，你的大脑会立即浮现黑板、课桌、椅子、讲台等情景，然后修正；而不是见到一个教室就记忆一个教室的细节。

可用于表示自然语言理解、视觉理解等方面。

2.5.1 框架的结构

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。由以下部分构成：

1. 框架名

- 每个框架都有一个框架名子，唯一标识一个框架。

2. 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。

3. 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。

4. 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。

March 16, 2020

框架结构示意图

<框架名>

<槽名1><槽值1> | <侧面名11><侧面值111, 侧面值112, ...>
| <侧面名12><侧面值121, 侧面值122, ...>

<槽名2><槽值2> | <侧面名21><侧面值211, 侧面值212, ...>
| <侧面名22><侧面值221, 侧面值222, ...>

...

...

...

<槽名k><槽值n> | <侧面名n1><侧面值n11, 侧面值n12, ...>
| <侧面名n2><侧面值n21, 侧面值n22, ...>

【例2.5.1】 用框架表示“假冒伪劣商品”

- 本例中框架有5个槽，其中处罚槽有4个侧面，处罚时间侧面填写的约束条件：按年、月、日顺序。

< 假冒伪劣商品 >

类属（ISA, AKO）：

商品名称：

生产者：

销售者：

处 罚：

处理方式：

处罚依据：

处罚时间：单位（年、月、日）

经办机构：

2.5.2 框架的基本概念

1. 框架与框架实例

- 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（类）中便可形成一个具体的实体(对象)，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

< 大学教师 >

AKO:<教师>

姓名：单位（姓、名）

年龄：单位（岁）

性别：单位（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 >

开始时间：单位（年、月）

终止时间：单位（年、月）

缺省：当前时间

※ 框架实例

< 教师-1 >

ISA：<大学教师>

姓名：夏冰

年龄：30

性别：女

职称：副教授

部门：计算机软件研究所

住址：< addr-1 >

工资：< sala-1 >

开始时间：1986.9

终止时间：2000.9

2.框架的继承

- 通过AKO、ISA槽从上层框架往下层框架，或从框架向框架实例进行继承。

< 教师 >

AKO:<知识分子>

姓名：单位（姓、名）

年龄：单位（岁）

性别：单位（男、女）

缺省：男

学历：范围（博士、硕士、学士、其它）

缺省：学士

专业：

学校：

住址：< 住址框架 >

< 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 >

< 中学教师 >

AKO:<教师>

教授课程：

职称：范围（特级、一级、二级、其它）

缺省：二级

工资：< 工资框架 >

“教师-1”和 “教师-2”为2个框架实例

“教师-1”从 “大学教师” 和 “教师” 框架继承

“教师-2”从 “中学教师” 和 “教师” 框架继承

< 教师-1 >

ISA:<大学教师>

姓名：李四

年龄：45

性别：男

学历：博士

专业：计算机科学与技术

学校：合肥工业大学

住址：< Frm_addr1 >

院系：计算机学院

方向：人工智能

职称：教授

工资：< Frm_salary1 >

< 教师-2 >

ISA:<中学教师>

姓名：王丽

年龄：28

性别：女

学历：硕士

专业：数学

学校：合肥一中

住址：< Frm_addr2 >

教授课程：数学

职称：二级

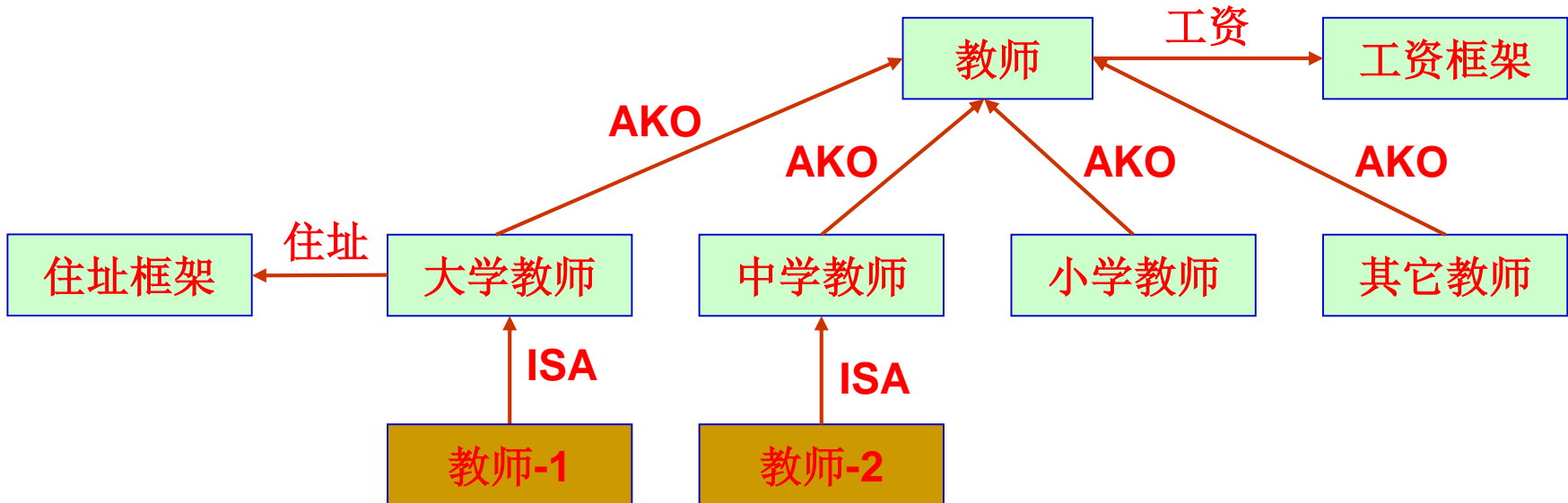
工资：< Frm_salary2 >

3.槽值和侧面值的填写

- ① 通过**已知**情况、或事物的属性值；
 - 见上例
- ② 通过**缺省、默认值**；
 - 上例中“教师”框架的“性别”和“学历”槽都有缺省定义。
- ③ 通过**继承**；
 - 上例中的“教师-1”和“教师-2”框架实例的所有槽
- ④ 通过**调用**其它框架、或附加过程。
 - 上例中“教师-1”和“教师-2”的“住址”和“工资”槽

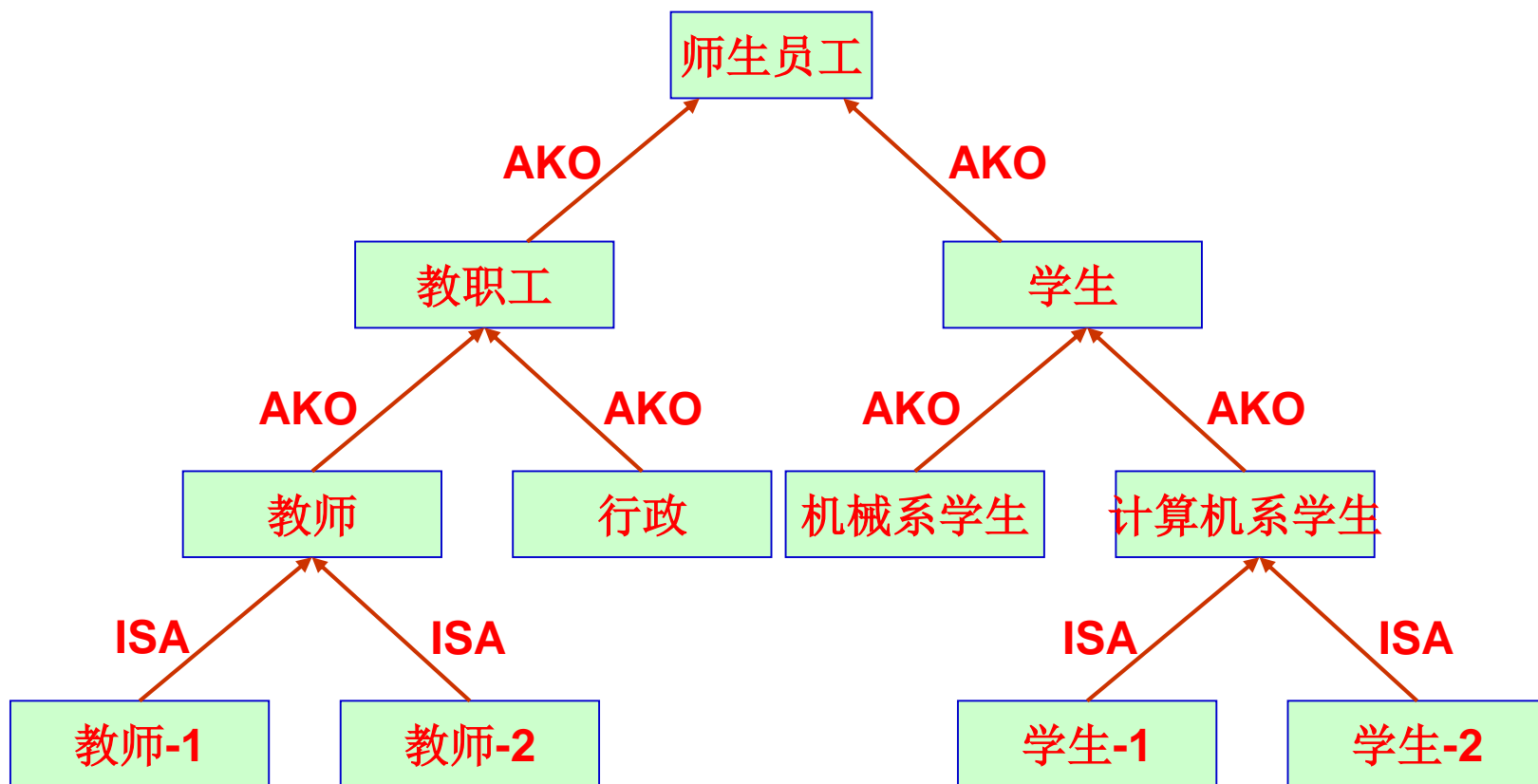
4. 框架网络（框架系统）

- 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网络（框架系统）。
- 连接手段
 - 继承 — 框架的**纵向联系**
 - 用另一框架填写槽值或侧面值 — 框架的**横向联系**



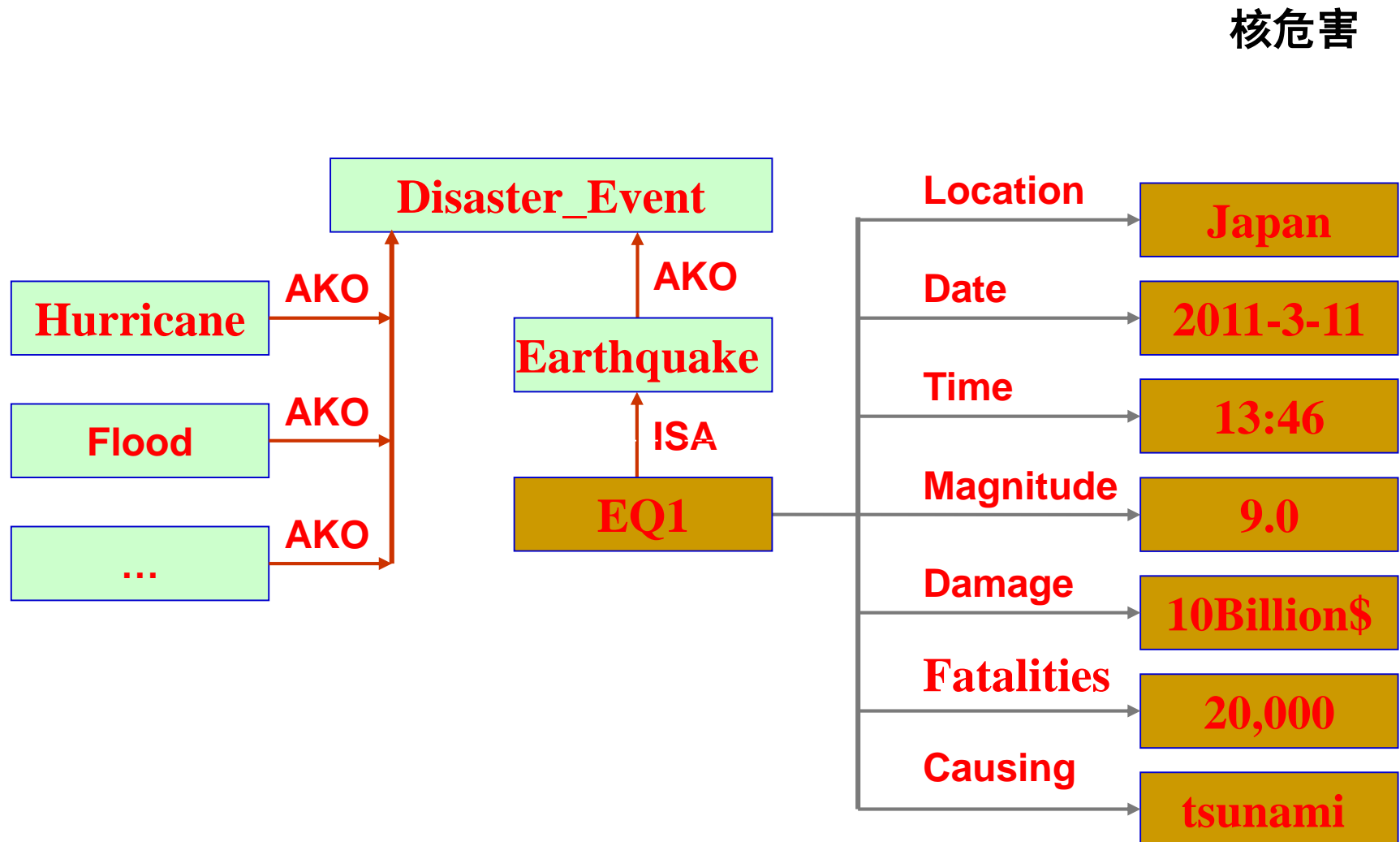
2.5.3 框架表示实例

【例2.5.2】 师生员工框架网络



2.5.3 框架表示实例

例2.5.3 自然灾害框架网络



【例2.5.4】 房间框架网络

框架名:<房间>

墙数 x_1 :缺省: $x_1 = 4$

条件: $x_1 > 0$

窗数 x_2 :缺省: $x_2 = 2$

条件: $x_2 \geq 0$

门数 x_3 :缺省: $x_3 = 1$

条件: $x_3 > 0$

前墙:<墙框架(w_1 , d_1)>

后墙:<墙框架(w_2 , d_2)>

左墙:<墙框架(w_3 , d_3)>

右墙:<墙框架(w_4 , d_4)>

天花板:<天花板框架>

地板:<地板框架>

门:<门框架>

窗:<窗框架>

条件: $w_1 + w_2 + w_3 + w_4 = x_2$

$d_1 + d_2 + d_3 + d_4 = x_3$

类型:(<办公室>, <教室>, <卧室>, <厨房>, <仓库>, ...)

2.5.3 框架表示实例

【例2.5.5】产生式的框架表示。设有以下产生式：

- IF 发烧、头痛且流鼻涕 THEN 患感冒。

用框架表示可为：

< 诊断-1 >
ISA:<诊断>
前提：条件1：发烧
条件2：头痛
条件3：流鼻涕
结论：患感冒

2.5.3 框架表示实例

【例2.5.6】 人与人之间纠纷的框架描述。

< 打人-1 >

ISA:<打人>

动作：打

动作发出者：张3

动作接受者：李4

后果：(<打人-2> OR <哭泣-2>)

< 哭泣-1 >

ISA:<哭泣>

动作：哭

动作发出者：张3

后果：心理平衡

< 打人-2 >

ISA:<打人>

动作：打

动作发出者：李4

动作接受者：张3

后果：(<打人-1> OR <哭泣-1>)

< 哭泣-2 >

ISA:<哭泣>

动作：哭

动作发出者：李4

后果：(<得意> OR <懊悔>)

March 16, 2020

2.5.4 框架推理

框架推理即基于框架的问题求解。在框架系统中，问题求解主要是通过通过对框架的**继承、匹配与填槽**来实现的。

问题求解（推理）手段：**继承、匹配与填槽**

问题求解过程：

- ① 首先要把该问题用框架表示出来。
- ② 然后利用框架之间的继承关系，把它与知识库中的已有框架进行匹配，找出一个或多个候选框架，并在这些候选框架引导下进一步获取附加信息，填充尽量多的槽值，以建立一个描述当前情况的实例。
- ③ 最后再用某种评价方法对候选框架进行评价，以决定是否接收该框架。

2.5.4 框架表示法特点

框架表示法的优点

- **结构性：**最突出特点是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：**框架表示法不仅可以从多个方面、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：**在框架系统中，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，这样既减少知识冗余，又较好地保证了知识的一致性。
- **自然性：**框架能把与某个实体或实体集相关特性都集中在一起，从而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。

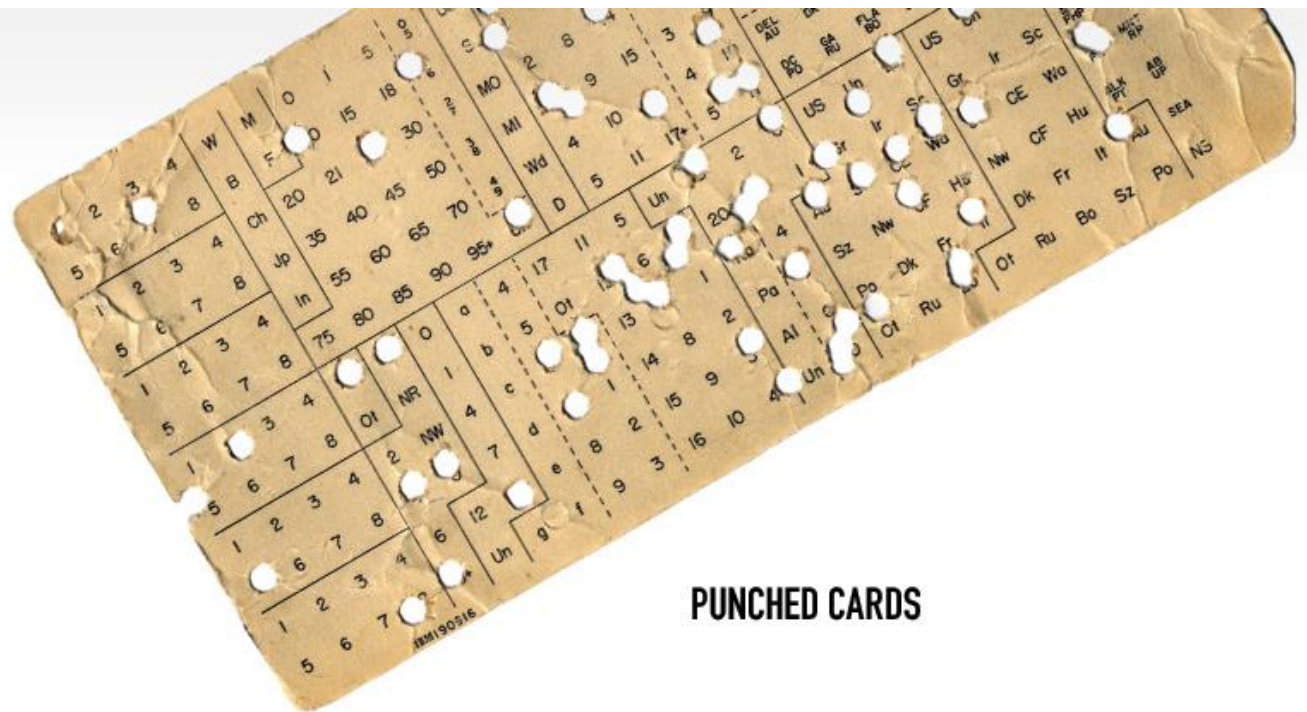
2.5.4 框架表示法特点

框架表示法的不足

- **缺乏框架的形式理论**：至今，还没有建立框架的形式理论，其推理和一致性检查机制并非基于良好定义的语义。
- **缺乏过程性知识表示**：框架系统不便于表示过程性知识，缺乏如何使用框架中知识的描述能力。框架推理过程需要用到一些与领域无关的推理规则，而这些规则在框架系统中又很难表达。
- **清晰性难以保证**：由于各框架本身的数据结构不一定相同，从而框架系统的清晰性很难保证。

框架表示法与产生式方法的比较

	产生式系统	框架系统
知识单元	规则	框架
推理机理	固定,与知识库独立	可变, 与知识库一体
建立知识库	容易	困难
通用性	低	高
应用	简单问题	复杂问题
用户	初学者	专家



PUNCHED CARDS

2.6 知识的其它表示方法

1.剧本表示法

2.过程表示法

3.面向对象表示法

4.Petri网表示法

...

Thank you !

