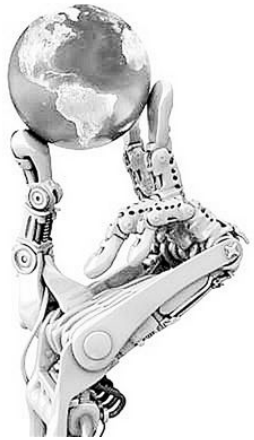
Knowledge Representation in Logic

一阶谓词逻辑



Knowledge Representation and Logic

- 逻辑推理是人类另一常用的问题求解技术
- 如何进行逻辑推理？
- 推理的过程怎样？
- 怎么实现自动推理？
- 所有这些问题的基础首先必须是怎样用适合逻辑推理的方式表示问题

推理示例—马普尔小姐探案

谁是马克谁是约尔？

马普尔小姐的推理过程

- 观察结果
  - 马克是右撇
  - 约尔是左撇
- 推理规则
  - 如果手表戴
  - 如果手表戴

则

结论

只是穿着[illegible]服的同一个人——约尔

人类的推理可以理解语义

机器如何进行这样类似的推理？

需要将推理的过程与理解分割开，将其形式化

推理的一般形式

已知：*事实*1，*事实*2，…

如果 *事实*1 那么 *结论*1

如果 *事实*2 那么 *结论*2

….

得到：*结论*1，*结论*2，…

- 将事实与规则等抽象出来，不涉及具体内容，借助一些符号来表示，推理过程可以被形式化

P：某已知事实

P→Q：如果 P 那么 Q

结论： Q

这个过程不需要直觉和解释

符号与形式语言

- 自然语言不适合计算机处理
  - 例：用红墨水写一个“蓝”字，请问，这个字是红字还是蓝字？
- 需要一种无歧义，方便存储和表达的形式化符号表征体系
  - 数理逻辑
    - 命题逻辑
    - 谓词逻辑

谓词逻辑

- 什么是谓词？
  - 原子命题中刻画个体的性质或个体间关系的成分
- 谓词逻辑是一种形式语言
- 是目前为止能够表达人类思维活动规律的一种最精确的语言
- 接近自然语言，又方便存入计算机处理
- 最早应用于人工智能中表示知识
- 适合于表示事物的状态、属性、概念等，也可用来表示事物间确定的因果关系

谓词逻辑

- Terms（项）
  a. 一个常量是项
  b. 一个变量是项
  c. 如果f是一个n元函数符号，$t_1, t_2, \ldots, t_n$是项，则$f(t_1, t_2, \ldots, t_n)$也是项
  d. 所有项都是由规则(a)(b)(c)产生的
- Atoms（原子公式 ）

  如果P是一个n元谓词符号，$t_1, t_2, \ldots, t_n$是项，则$P(t_1, t_2, \ldots, t_n)$ 是一个原子公式，其他任何表达式都不是原子公式
- WFF（合式公式）
  a. An atom is WFF
  b. 如果F和G是WFF，则~F，F∧G，F∨G，F→G，F≡G都是WFF
  c. 如果F是WFF，x是自由变量则$(\forall x)$F，$(\exists x)$F都是WFF
  d. WFF仅由有限次使用规则(a)(b)(c)产生。

Equivalence

否定之否定　$\sim(\sim P) \equiv P$

$P \vee Q \equiv \sim P \rightarrow Q$

摩根律　$\sim(P \wedge Q) \equiv \sim P \vee \sim Q$

$\sim(P \vee Q) \equiv \sim P \wedge \sim Q$

分配律　$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$

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

交换律　$P \wedge Q \equiv Q \wedge P$

$P \vee Q \equiv Q \vee P$

结合律　$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$

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

逆反律　$P \rightarrow Q \equiv \sim Q \rightarrow \sim P$

Quantifiers

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

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

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

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

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

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

[例]

man(smith)　　smith是人

between(albert, susan, david)

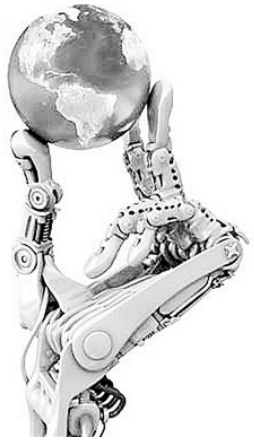
albert在susan与david之间

$(\forall x)(\text{man}(x)\rightarrow\text{mortal}(x))$　　人都会死

$(\exists x)(\text{man}(x)\wedge\text{clever}(x))$　　有的人聪明

Knowledge Representation in Production Rules

产生式规则



产生式

- 美国数学家E．Post 1943年提出
- Post根据替换规则提出了一种称为波斯特机的计算模型
- 模型中的每一条规则当时被称为一个产生式
- 后来，这一术语几经修改扩充，被用到许多领域
  - 例如，形式语言中的文法规则也称为产生式
- 产生式又称为产生式规则，或简称规则。

产生式的一般形式

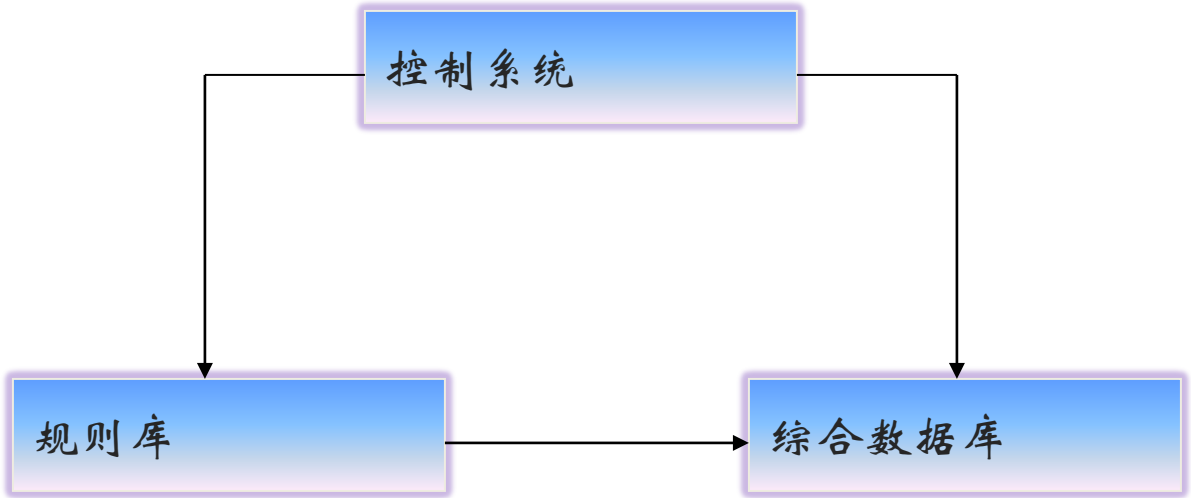
- 产生式通常用于表示具有因果关系的知识
- 一般形式
  - IF $A_1, A_2, \ldots, A_n$ THEN B
    - IF部分——前件：条件、前提
    - THEN部分——后件：动作、结论
- 语义：如果前提满足，则可得结论或者执行相应的动作
  - 即后件由前件来触发
  - 前件是规则的执行条件，后件是规则体
- 例：
  - IF 动物是哺乳动物 AND 吃肉 THEN 该动物是食肉动物

产生式与蕴含式

- 产生式与逻辑蕴含式在形式上非常相似，但二者不同
- 逻辑蕴含式只是产生式的一种特殊情况
  - 产生式除逻辑蕴含式外，还包括各种操作、规则、变换、算子、函数等等
  - 产生式描述了事物之间的一种对应关系(包括因果关系和蕴含关系)，其外延十分广泛
    - 状态转换规则和问题变换规则
    - 程序设计语言的文法规则、逻辑中的逻辑蕴含式和等价式、数学中的微分和积分公式、化学中分子结构式的分解变换规则
    - 体育比赛中的规则、国家的法律条文、单位的规章制度…

产生式系统

- 由Post在1943年提出
- 1954年，Markov提出产生式规则的控制策略
- 1965年设计出基于产生式系统结构的第一个专家系统DENDRAL
- 又称基于规则的系统
- 基本结构



产生式系统的分类

- 产生式系统从不同的角度出发，有不同的分类
- 按推理方向
  - 前向
  - 后向
  - 双向产生式系统
- 按产生式规则库和全局数据库的性质及结构特征
  - 可交换
  - 可分解
  - 可恢复产生式系统

产生式系统的表示例

- MYCIN系统中从专家那里获得的规则是
  - IF (1) the stain of organism is gramnegative（细菌染色为革兰氏阴性）
  - AND (2) the morphology of the organism is rod,（细菌的形态为杆状）
  - AND (3) the aerobicity of the organism is anaerobic（细菌厌氧）
  - THEN there is suggestive evidence(0.6) that the identity of the organism is bacteroides（细菌是杆菌的可信度为0.6）
- 用Lisp实现的机器内部表示为

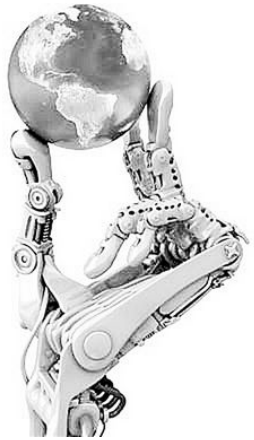
```
PROMISE: ($AND (SAME CNTXT GRAM GRAMNEG)
                                (SAME CNTXT MORPH ROD)
                                (SAME CNTXT AIR AEROBIC))
ACTION: (CONCLUDE CNTXT CLASS BACTERIACEAE TALLY 0.6)
```

产生式表示法的特点

- 优点：
  - –产生式表示格式固定，形式单一，数据库的建立较为容易
  - –推理方式单纯，知识库与推理机分离
  - –既可以表示确定性知识又可以表示不确定性知识，既便于表示启发性知识又便于表达过程性知识
  - –经常作为建造专家系统的首选知识表示方法
- 缺点：
  - –求解过程是一个重复的“匹配—冲突消解—执行”的过程，匹配耗时，系统工作效率低
  - –在求解复杂问题时易引起组合爆炸
  - –适合表达具有因果关系的过程性知识，无法表示结构关系的知识
  - –更适合于表示那些相关性不强、不存在结构关系的领域性知识

Semantic Network & Frame

语义网络与框架

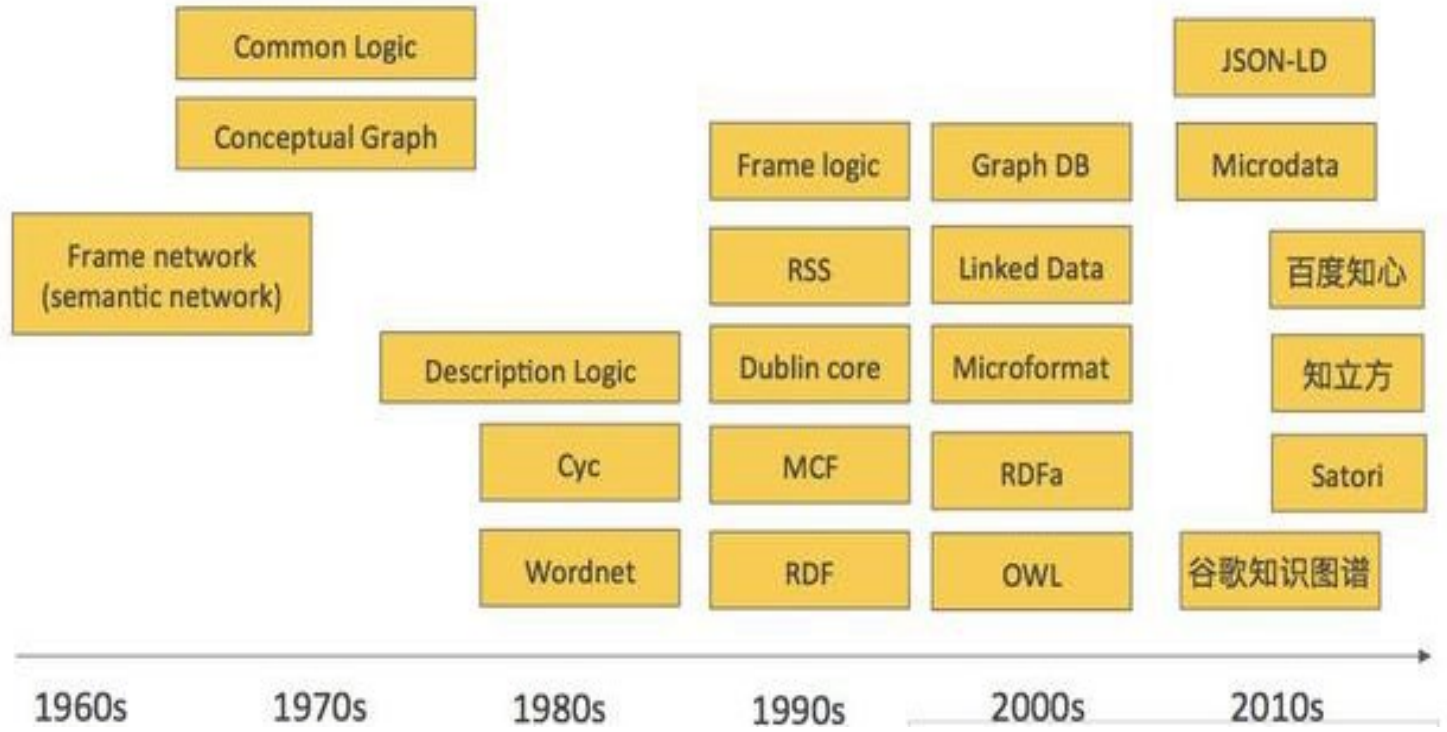


语义网络法（Semantic Network）

- 1968年J.R.Quillian在其博士论文中作为人类联想记忆的一个显式心理学模型最先提出
- 1972年，Simmons首先将语义网络表示法用于自然语言理解系统
- 是通过概念及其语义关系来表达知识的一种网络图
- 带标志的有向图
- 语义网络是知识的一种结构化表示方法
- 由节点(node)与有向弧(arc)组成
- 节点表示概念、事物、事件、情况等
- 弧表示语义关系

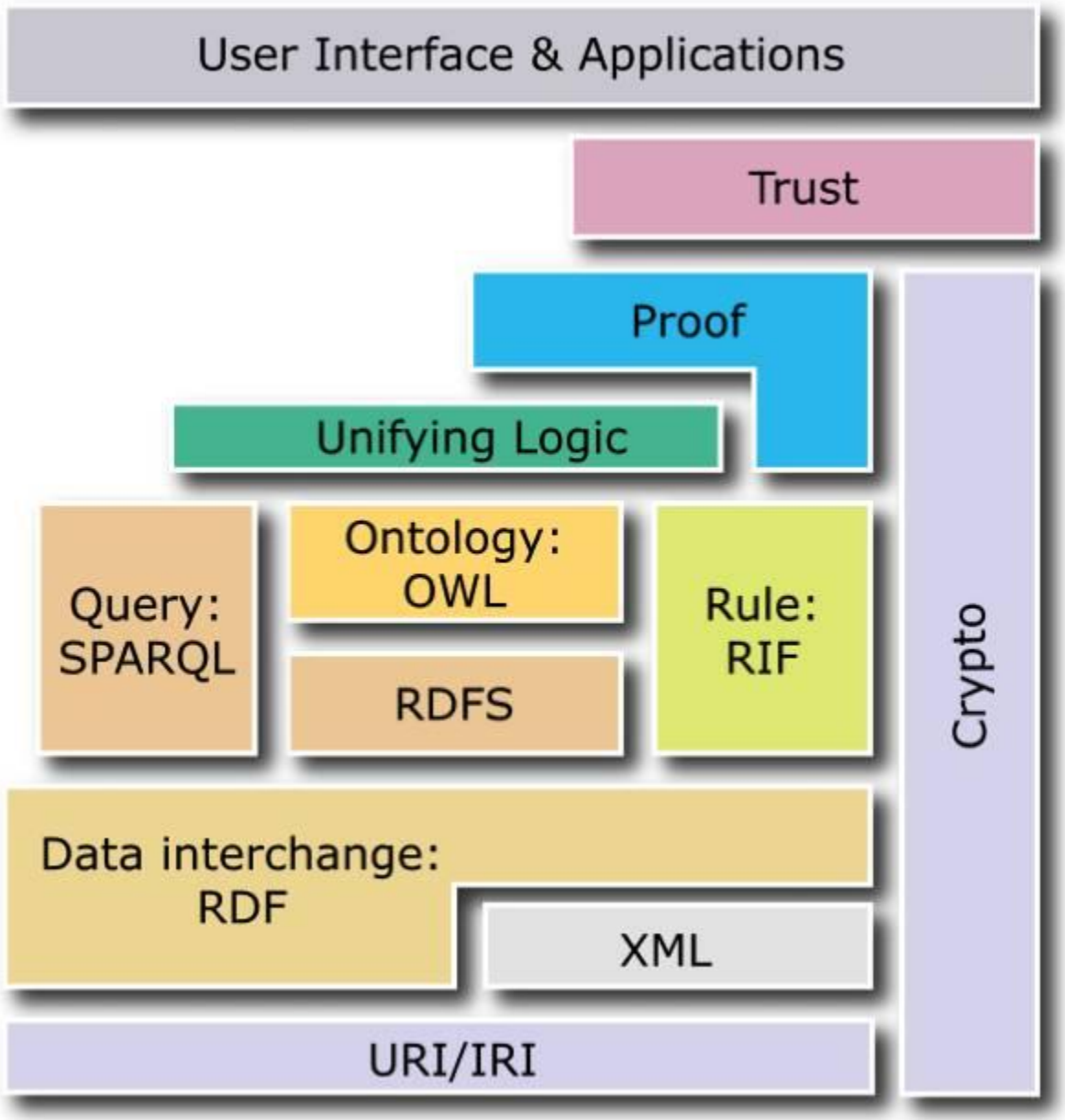
从语义网络到知识图谱

- 知识图谱旨在描述真实世界中存在的各种实体或概念
  - 其中，每个实体或概念用一个全局唯一确定的ID来标识，称为它们的标识符
  - 每个属性-值对用来刻画实体的内在特性，而关系用来连接两个实体，刻画它们之间的关联



从语义网络到知识图谱

- 到2006年，语义堆栈变得越来越复杂
  - 从开始的层次蛋糕——七层协议
  - 不能被大多数人理解
- 随着Linked Data的提出（Tim Berners-Lee，2006），很多数据被公开，并通过RDF结构化
  - 有了DBpedia（一个基于维基百科的结构化数据库，然后才有了IBM Waston
- 2013年，Google开始推Microdata
- 2006年之前的时候，弱语义转向强语义，导致复杂性增加，没有实用性
- 2006年后，向工程妥协

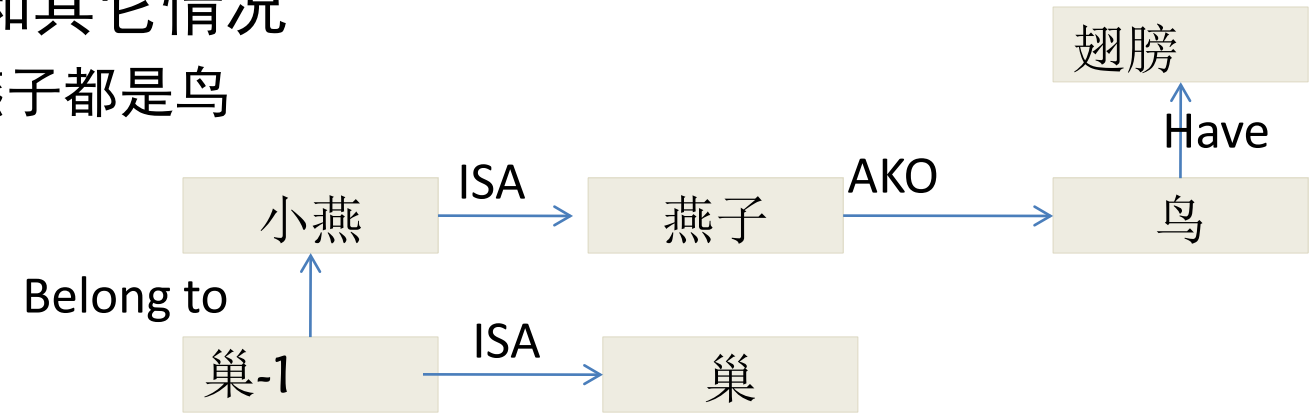


Semantic Network Representation

- 语义网络的结构
  - –组成部分
    - 词法
      - –词汇表中允许有哪些符号，它涉及各个节点和弧
    - 结构
      - –符号排列的约束条件，指定各弧线连接的节点对
    - 过程
      - –访问过程，这些过程能用来建立和修正描述，以及回答相关问题
    - 语义
      - –确定有关节点的排列及其占有物和对应弧线

二元语义网络的表示

- 表示占有关系和其它情况
  - 例1：所有的燕子都是鸟



- 小燕是一只燕子，燕子是鸟；巢-1是小燕的巢，巢-1是巢中的一个。

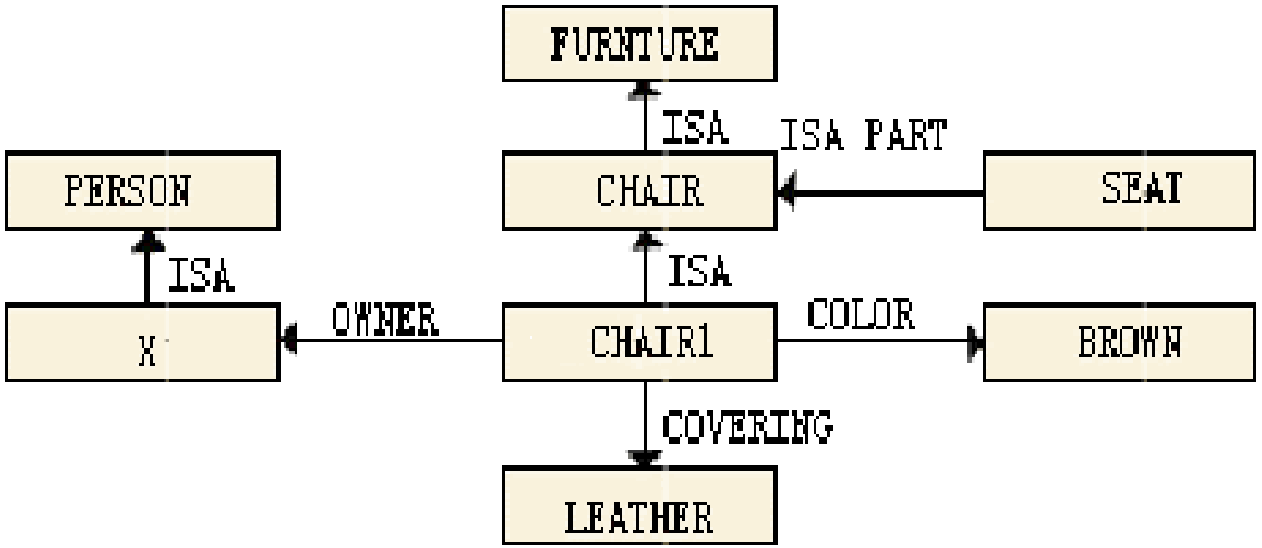
- 选择语义基元
  - 试图用一组基元来表示知识，以便简化表示，并可用简单的知识来表示更复杂的知识。

常用语义联系

- 表示事物的性质、属性：
  - ISA（Is-A）、AKO（A-Kind-Of）、HAVE、AMO（A-Member-Of）
- 构成关系：
  - Composed-of
- 时间关系：
  - Before、After、At
- 位置关系：
  - Located-on、Located-at、Located-under
- 相似或接近关系：
  - Similar-to、Near-to表示

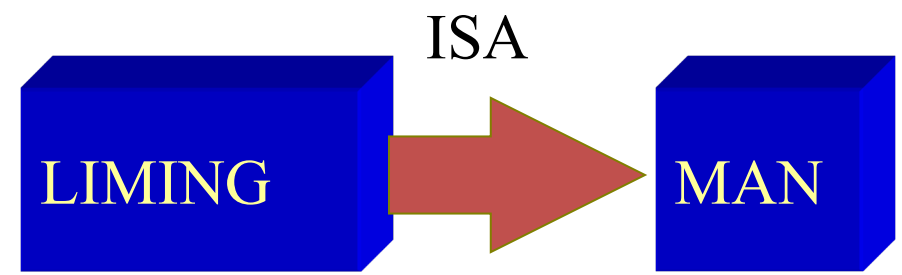
例

- 有一个椅子的颜色是咖啡色的；椅子包套是皮革；椅子是一种家具；椅子是座位的一部分；椅子的是属于X先生的



多元语义网络的表示

- 谓词逻辑与语义网络等效

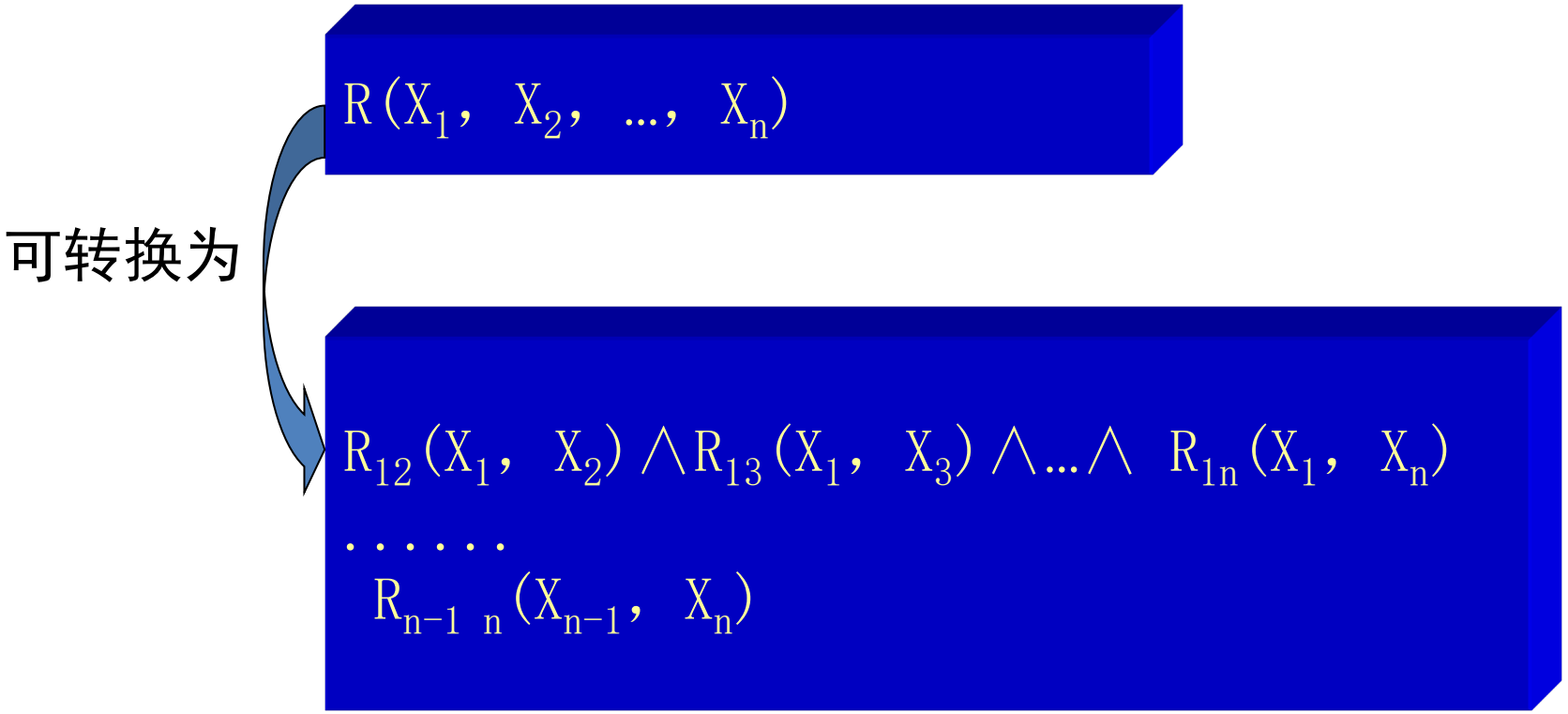


（语义网络）

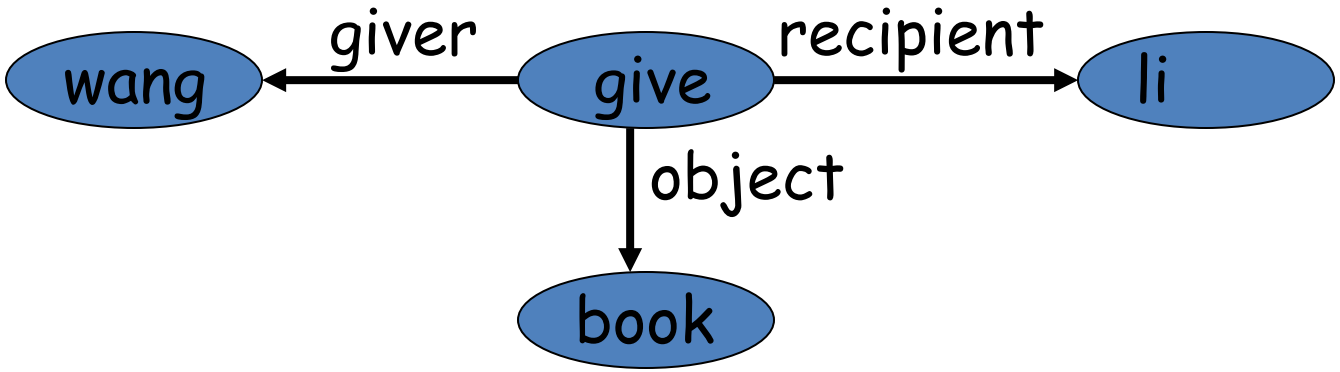
ISA（LIMING，MAN）或 MAN（LIMING）

（谓词逻辑）

- 多元语义网络表示的实质
  - 把多元关系转化为一组二元关系的组合，或二元关系的合取。



例：　　小王给小李一本书



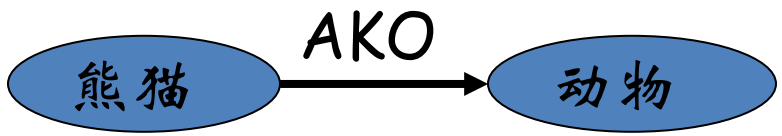
通过增加附加节点将多元关系转化为二元关系

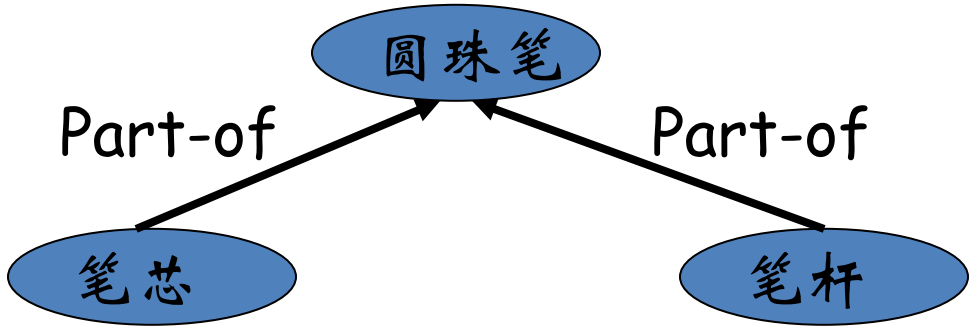
- node
  - –概念节点

    用以表示基本概念
  - –实例节点

    用以表示具体事物或属性

- 弧
  - –以个体为中心
    节点一般是名词性个体或概念
    - ISA （实例联系）
    - AKO（A Kind Of，泛化联系，子类）
      - Part-of（聚集联系）

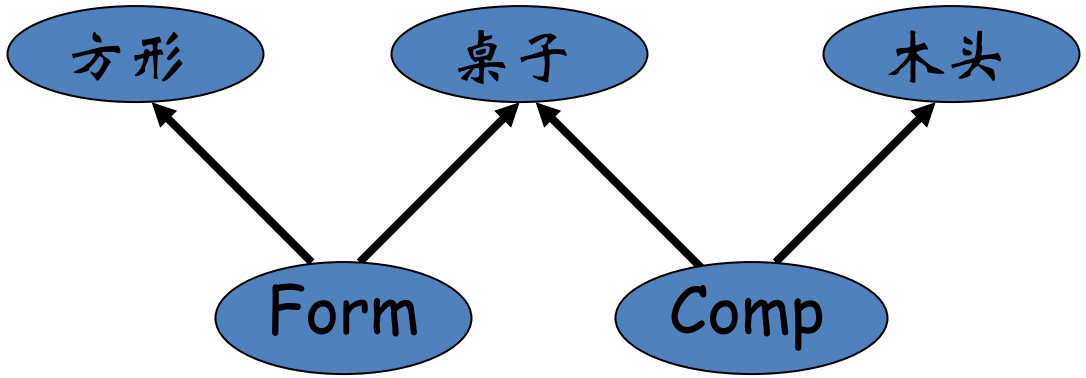


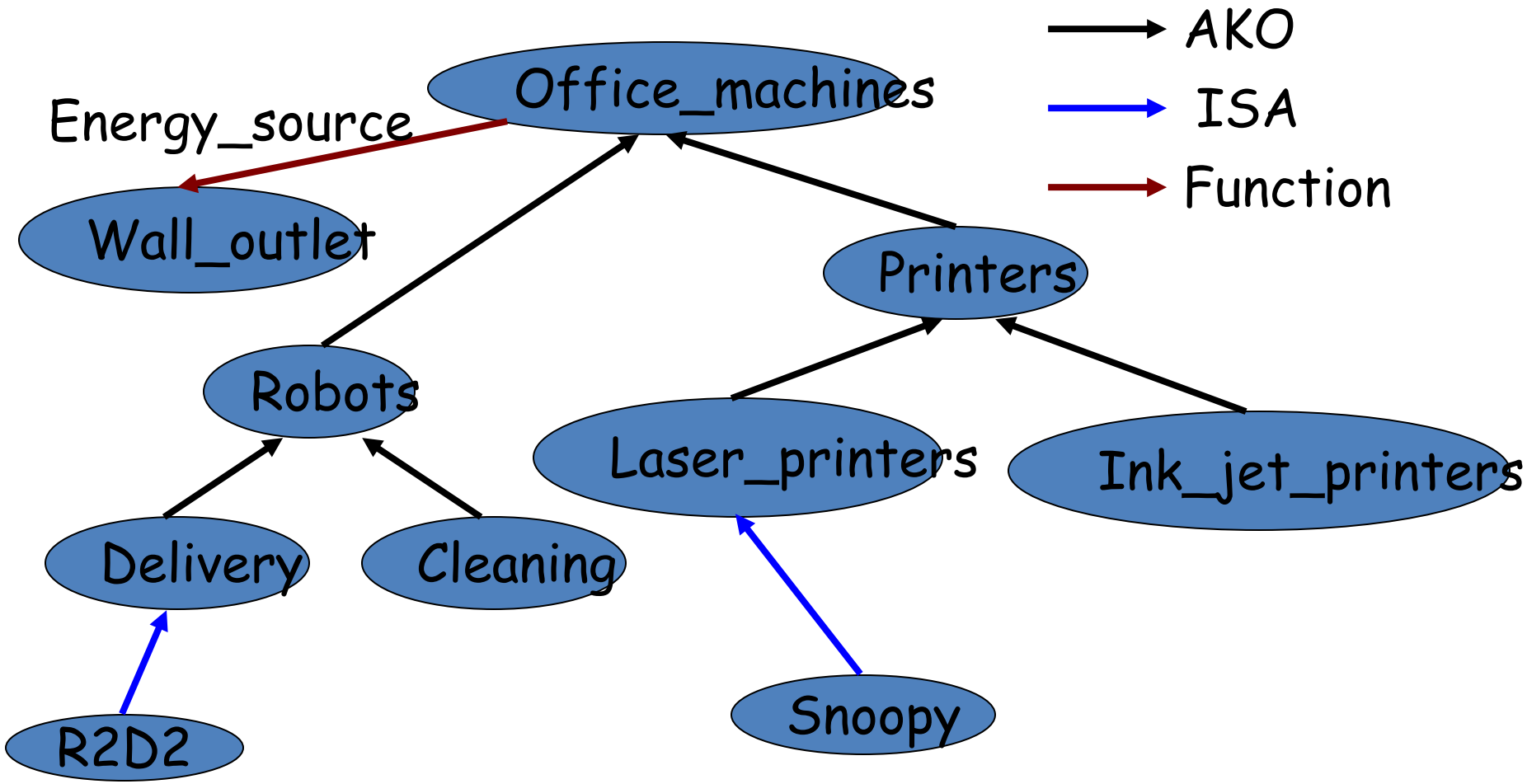


– 以谓词或关系为中心

例：有一张木头做的方桌

Form(桌子,方形) Comp(桌子,木头)

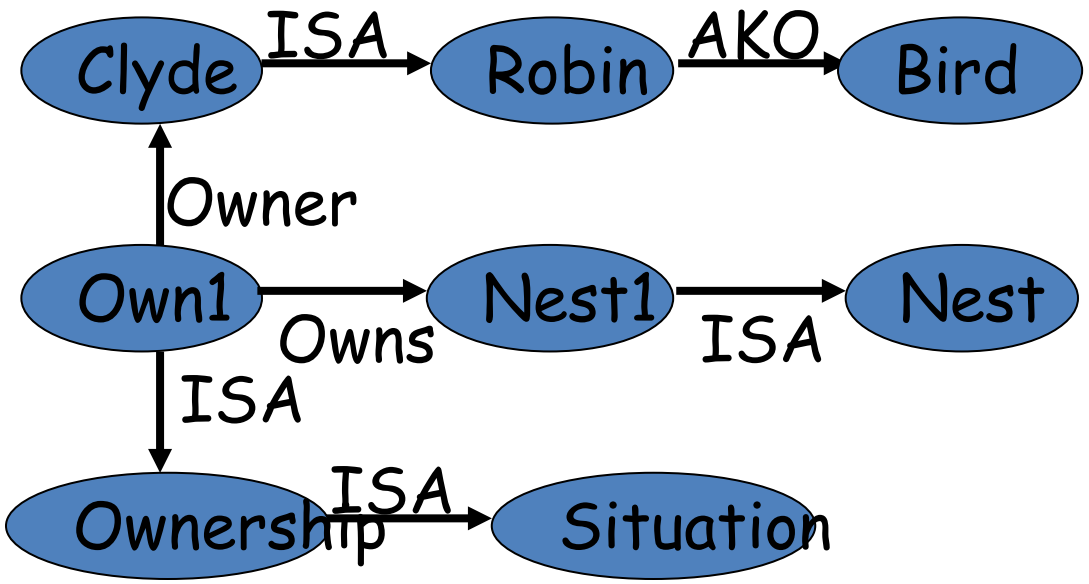



AKO
ISA
Function
Office_machines
Energy_source
Wall_outlet
Printers
Robots
Laser_printers
Ink_jet_printers
Delivery
Cleaning
R2D2
Snoopy

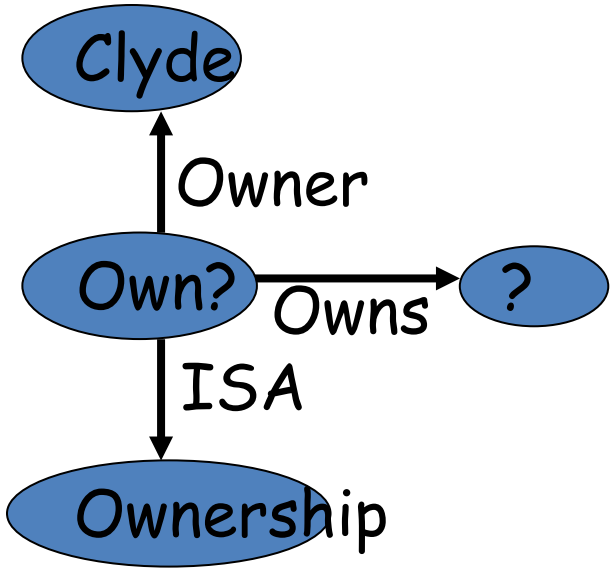
语义网络的推理

两种推理机制：匹配、继承

- 匹配(包括节点和弧的匹配)
  - 构造目标网络块
  - 在事实网络中寻找匹配



What does Clyde own?



- 继承

  把对事物的描述从概念节点或类节点传递到实例节点

三类继承性

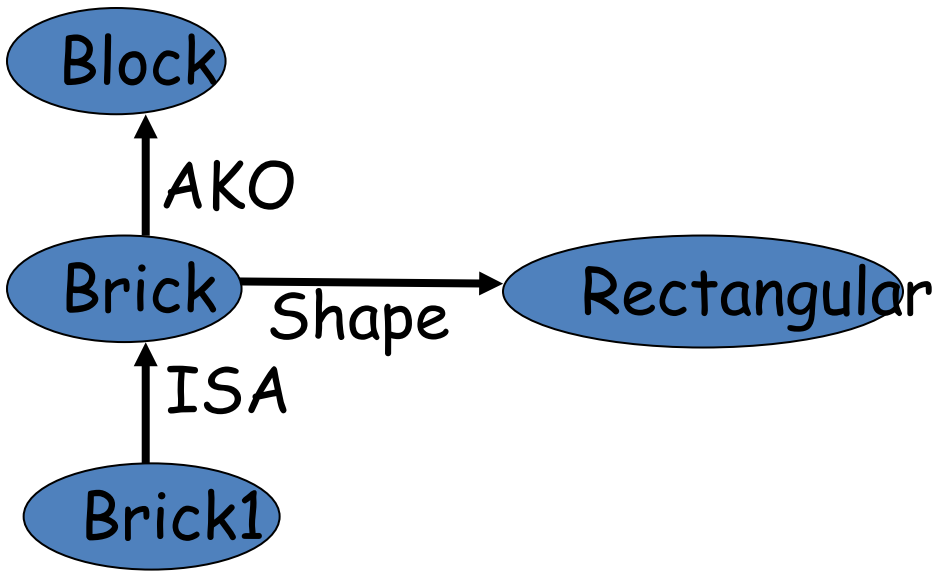
- 直接传递(pass)——-子节点直接继承父节点的属性
- 附加传递(add)————子节点把父节点的属性和自己的属性相综合
- 排斥传递(exclude)—子节点与父节点的属性不相容，抑止传递

三种继承方式

– 值继承

is-a, AKO(a kind of)

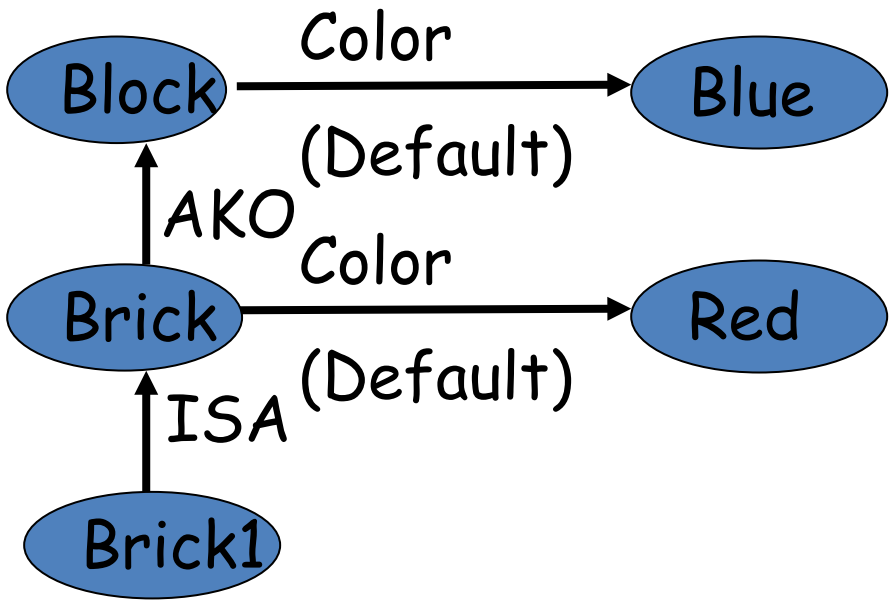
求Brick1的形状



—Default(默认继承)

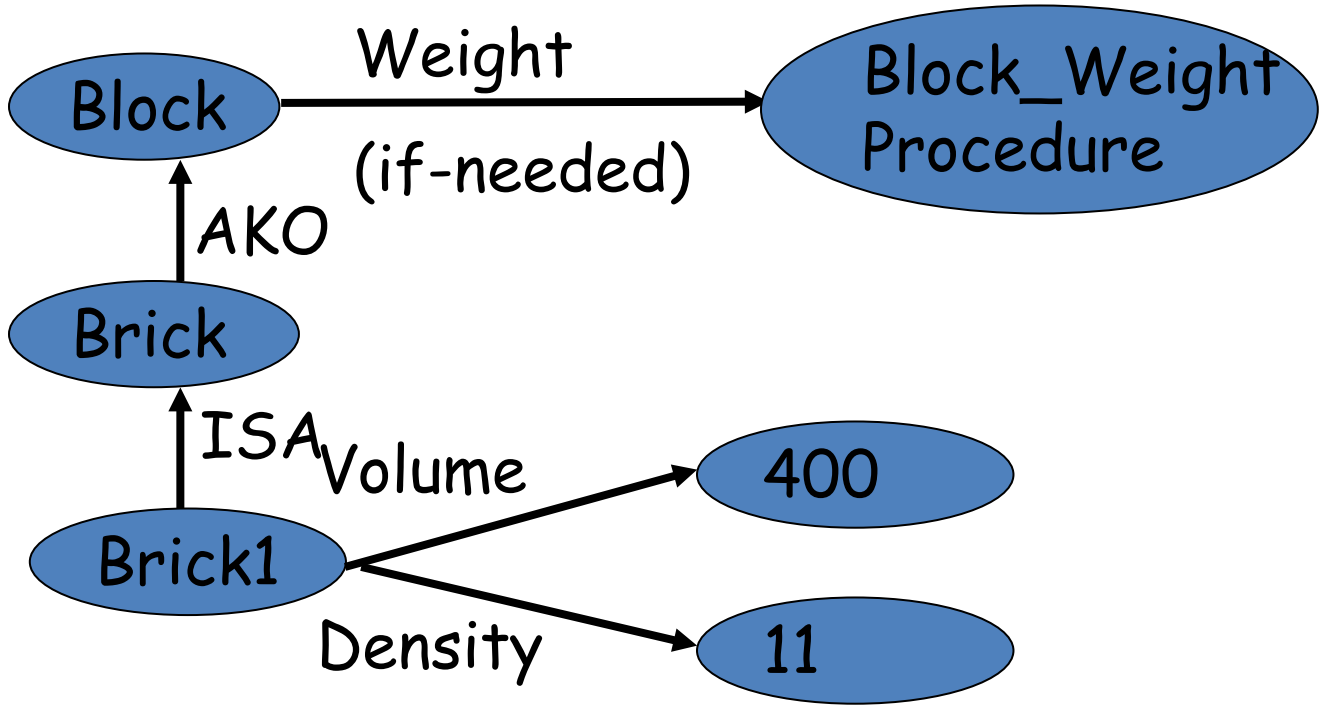
默认值——具有相当程度的真实性但不能十分肯定的值

如：鸟会飞；头疼可能是感冒



– if-needed（“如果需要”继承，“附加过程”继承）

某些情况下，对事物的描述不能直接从概念节点或类节点继承得到，但可利用已知信息来计算 ，这种计算程序称为if-needed程序



语义网络表示的特点与不足

- 具有结构性

适于表示分类学型知识和事物特性的知识

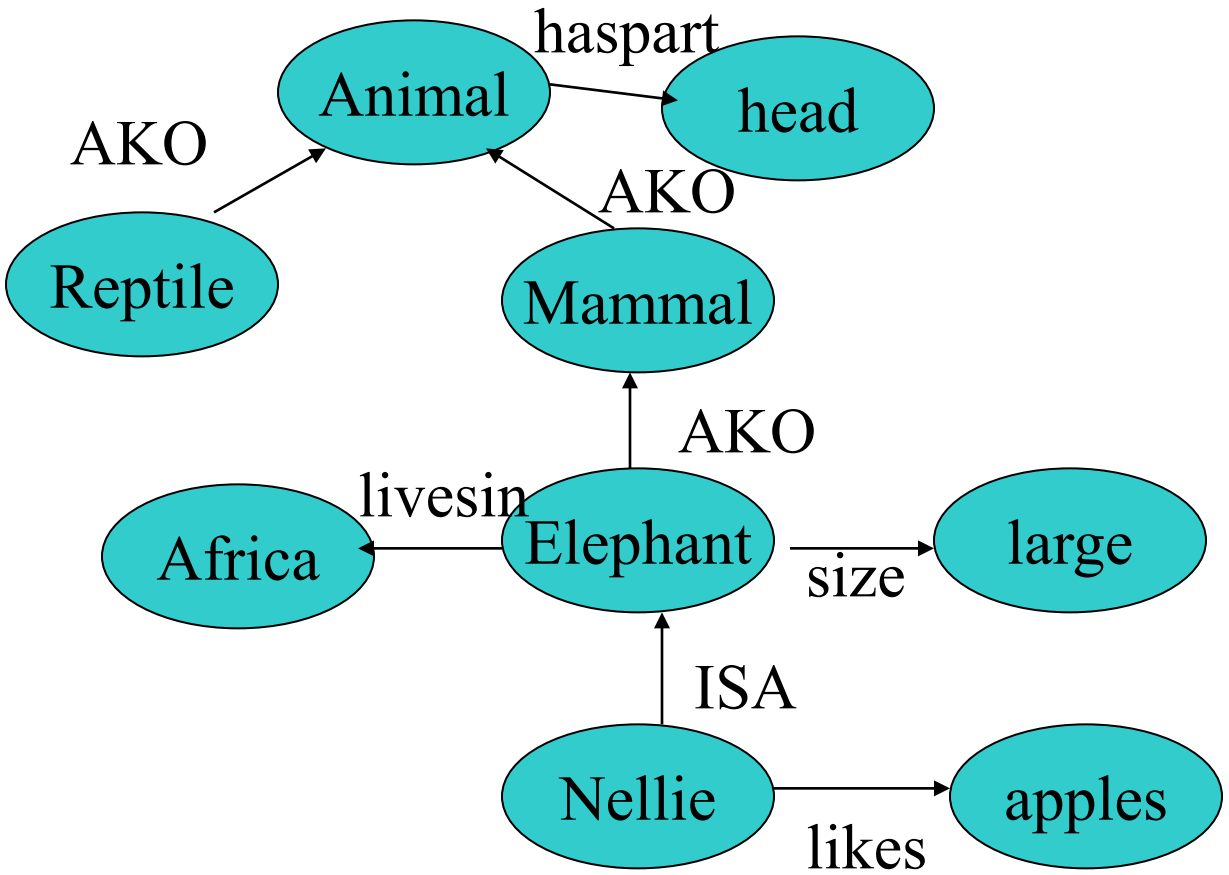
- 灵活性

可以任意定义新的节点与弧

- 继承性，表达能力强
- 语义网络上的继承推理具有非单调性
- 处理上的复杂性
- 非严格性
- 多重继承冲突问题是重要的研究课题

思考题

- Use semantic networks to represent:
  - Nellie is an elephant, he likes apples. Elephants are a kind of mammals, they live in Africa, and they are big animals. Mammals and reptiles are both animals, all animals have head.


haspart
Animal
head
AKO
AKO
Reptile
Mammal
AKO
livesin
Africa
Elephant
large
size
ISA
Nellie
apples
likes

Semantic Network Formalisms

- Used a lot for natural language understanding
  - Represent two sentences by graphs
    - Sentences with same meaning have exactly same graphs
- Conceptual Dependency Theory
  - Roger Schank's brainchild
  - Concepts are nodes, relationships are edges
  - Narrow down labels for edges
  - Problem:
    - Not clear whether reduction to graphs can be automated for all sentences in a natural language

Frame （框架）

- 75年由美国人工智能学者明斯基提出
- 认为人们对现实世界的认识都是以一种类似于框架的结构存储在记忆中
- Information retrieval when facing a new situation
  - The information is stored in frames with slots(槽)
  - Some of the slots trigger actions, causing new situations
- 框架是一种描述所论对象属性的数据结构
  - Frames are templates which are to be filled-in in a situation
- 在框架理论中，框架是知识表示的一个基本单位
- 也是一种语义网络
- 继承性是其重要特性
    - Also very similar to objects in OOP

框架的结构

- 框架描述一类物体，由框架名与一些描述物体各个方面的槽(slot)组成
- 每个槽可分为多个侧面(facet)
- 每个侧面可有一个或多个值

<框架名>

<槽名1>：<侧面11>(值111，值112，...)

<侧面12>(值121，值122，...)

<槽名2>：<侧面21>(值211，值212，...)

<侧面22>(值221，值222，...)

...

约束：　约束条件1，约束条件2...

例：描述“大学教师”的框架

框架名：<大学教师>
类属：<教师>
学位：（学士，硕士，博士）
专业：<学科专业>
职称：（助教，讲师，副教授，教授）
外语：语种：范围：（英，法，日，俄，德，...）
缺省：英
水平：（优，良，中，差）
缺省：良

- 槽（侧面）类型
  - 值
  - Default
  - 继承
  - if-needed

<CHAIR>
Specialization-of:FURNITURE
Number-of-Legs:DEFAULT 4
Style-of-Back:Straight,Cushioned
Number-of-Arms:0,1 or 2

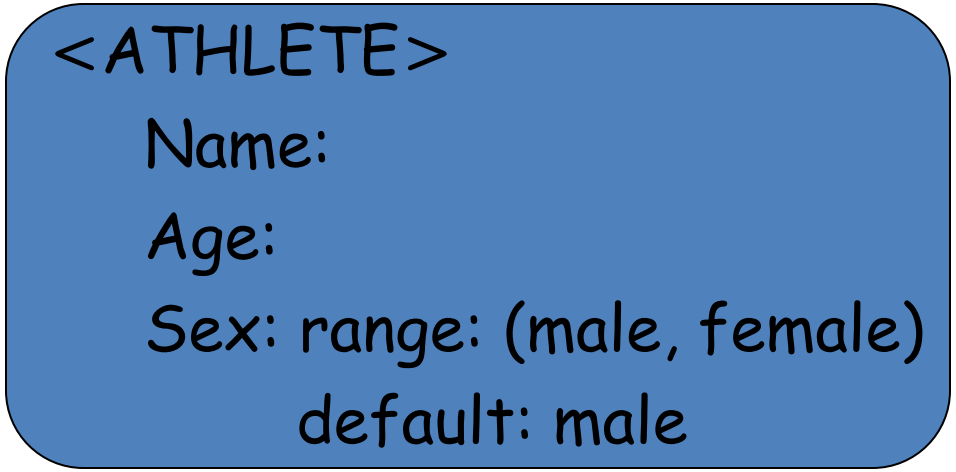
<JOHN'S-CHAIR>
Specialization-of:CHAIR
Style-of-Back:Cushioned
Number-of-Arms:0

- 四种侧面填写方式
  - 由已知情况或物体属性提供
  - 通过默认隐含
  - 由继承获得
  - 对附加过程侧面通过执行附加过程实现
- 框架中的槽与侧面可任意定义

槽与侧面也可以是另一框架，形成框架网络

框架网络

- 框架间的联系由槽名指明
- 常用槽
  - ISA槽



– Subclass槽

– AKO槽

– Instance槽

是AKO槽的逆关系

<ATHLETE>
Instance: <Chess player>,
<football player>,
<basketball player>
Name:
Age:
Sex: range: (male, female)
default: male

– Part-of槽

– Infer槽

指出两个框架间的逻辑推理关系

可表示相应的产生式规则

– Possible-Reason槽

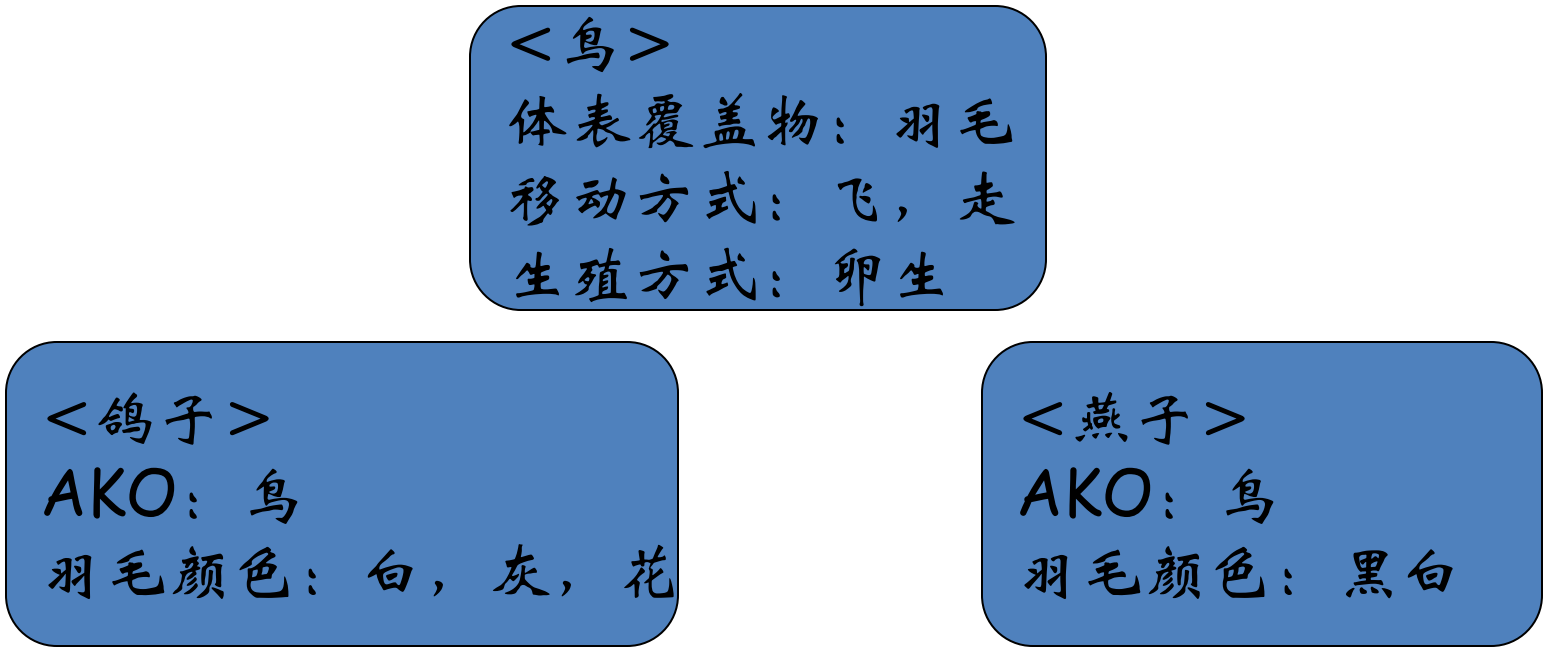
与Infer槽相反

<诊断规则>
症状1：咳嗽
症状2：发烧
症状3：流涕
Infer：<结论>
可信度：0.8

<结论>
病名：感冒
治疗方法：服用感冒胶囊，
一日3次，每次2-3粒
注意事项：多喝开水
Possible-Reason：<诊断规则>

- 对框架及侧面进行合理组织
  - 减少重复性信息
  - 尽量将不同框架描述的相同属性取出构成上层框架

  如：用框架描述鸽子、啄木鸟、布谷鸟、燕子及鹦鹉五种动物



框架推理

两种推理活动：匹配、填槽

- 匹配
  - 将待解问题用框架表示
  - 匹配通过对相应槽的槽名和值逐个比较实现
  - 比较往往牵涉到其它框架
  - 问题的随机性也使匹配复杂化
- 填槽

  四种填槽方式
  - 查询：中间结果或用户输入
  - 默认
  - 继承
  - 附加过程计算

例：在关于师生员工的框架网络种找满足如下条件的教师：
男性，30岁以下，身体健康，讲师

<师生员工>
姓名：
年龄：
性别：
range:男,女
default:男
健康状况：
range:健康,一般,差
default:一般
住址：<住址>

<教职工>
AKO：<师生员工>
工作类别：
range:教师,干部,工人
default:教师
开始工作时间：
截止工作时间：
default:现在
离退休状况：
range:离休,退休
default:退休

<教师>
AKO：<教职工>
部门：单位(系,教研室)
语种：
range:英,法,德,日,俄
default:英语
外语水平：
range:优,良,中,差
default:良
职称：
range:教授,副教授,讲师,助教
default:讲师
研究方向：

<教师1>
AKO：<教师>
姓名：孙林
年龄：28
健康状况：健康
部门：
计算机系软件教研室
语种：德语
开始工作时间：
1985.9
...

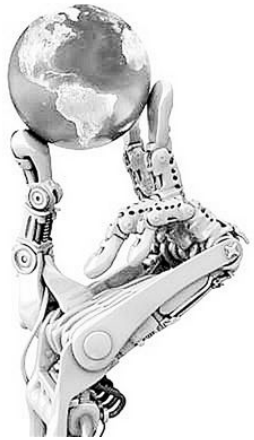
<教师x>
AKO：<教师>
姓名：
年龄：<30
性别：男
健康状况：健康
职称：讲师

框架表示的特点与不足

- 结构性
- 继承性
- 自然性
- 模块性
- 没有形成完整的理论体系
- 缺乏清晰的语义
- 多重继承的冲突

Script

脚本表示法



Script（脚本表示法）

- 夏克（R. C. Schank）1975年依据概念依赖理论提出
  –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
  –夏克在其研制的SAM（Script Applier Mechanism）中对动作一类的概念进行了原子化，抽取了11种原子动作，并用其来表示槽的行为
- 表示特定领域内事件的发生序列
- 脚本的组成
  – 进入条件
  – 角色
  – 道具
  – 场景
  – 结局

脚本：餐厅

进入条件：顾客饿了，需要进餐；顾客有钱

角色：顾客、服务员、厨师、老板

道具：食品、桌子、菜单、钱

场景：

第一场：进入餐厅

顾客走入餐厅

顾客寻找桌子

在桌旁坐下

第二场：点菜

服务员给顾客菜单

顾客点菜

顾客把菜单给服务员

服务员告诉厨师所要食品

厨师做菜

第三场：上菜进餐

厨师把菜给服务员

服务员把菜送给顾客

顾客吃菜

第四场：顾客离开

顾客告诉服务员结帐

服务员拿来帐单

顾客付钱

顾客离开餐厅

结局：顾客吃了饭；

顾客花了钱；

老板挣了钱；

餐厅食品少了

- 在脚本适用于给定事件时，可通过它预测没有明显提及的事实

小结（Summary）

- 本章所讨论的知识表示问题是人工智能研究的核心问题之一。
- 知识表示方法很多，本章介绍了其中的7种
  - 图示法
  - 公式法
  - 陈述式表示
  - 过程式表示…