

人的大脑依赖所学的知识进行思考、逻辑推理、理解语言...

知识是什么？如何用计算机储存知识？如何从已知的知识推导出未知的知识？



聪明的AI

神经网络

知识表示

有学识的AI

感知

识别

判断

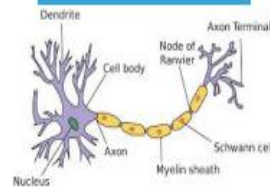
思考

语言

推理



深度学习



知识图谱



未来的AI

联接 + 符号

学习 + 推理

感知 + 认知

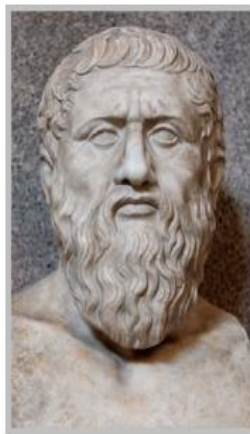
第 2 章 知识表示

柏拉图：知识即被证实的、真的和被相信的陈述。

Knowledge is justified true belief.

人类的自然语言、创作的绘画和音乐、数学语言、物理模型、化学公式等都是人类知识的表示形式和传承方式。具有获取、表示和处理知识的能力是人类心智区别于其它物种心智的重要特征。

AI的核心是研究怎样用计算机易于处理的方式表示、学习和处理各种各样的知识，广义的讲，神经网络也是一种知识表示形式。



柏拉图

Knowledge is a familiarity, awareness, or understanding of someone or something, such as facts, information, descriptions, or skills, which is acquired through experience or education by perceiving, discovering, or learning.

KG is just one form of Knowledge, text is also knowledge.

知识表示与自然语言理解 通用人工智能的必由之路

王咏刚 创新工场 AI 工程院副院长 2017.9

知识表示的历史沿革，内涵与外延

知识表示与今天主流的深度学习算法的关系

知识表示与通用人工智能（强人工智能）的关系

人工智能悖论

- 几乎所有成功的 AI 应用场景都较少依赖于知识的符号化表达
 - 每当我开除一名语言学家，语音识别系统的准确率就得到一次提升 —— Frederick Jelinek
- 几乎所有失败的 AI 应用场景都被归因于机器无法理解和表达人类知识
 - 客服系统做不好 —— 缺少领域知识
 - 机器人不够聪明 —— 缺少足够大的知识库
 - 图灵测试通不过 —— 缺少常识、世界知识和推理能力
- 人工智能真的需要知识表示和知识推理吗？
 - 这要看我们讨论的是弱人工智能还是强人工智能（通用人工智能）
 - 通用人工智能必然和知识表示密切联系在一起

- **领域性知识**：面向某个具体领域的知识，是专业性知识，如疾病诊断的知识。
- **常识性知识**：通用性知识，适用于所有领域；如一年有四个季节。

2013年，Google开源了一款用于词向量计算的工具word2vec:

可以在百万数量级的词典和上亿的数据集上进行高效地训练；得到的训练结果——词向量（word embedding），可以很好地度量词与词之间的相似性。

知识表示与自然语言理解

词向量或词嵌入会将每一个单词表示成一个相对较低的维度的向量（例如100维或200维）

- 用 word2vec 生成的 word embeddings 是不是一种知识表示？

- 完全基于字面统计信息 vs. 引入附加的语义信息（如 WordNet）

- 用语义信息辅助词向量生成 Semantic Word Embedding

1. 狗、犬的词向量距离接近

2. **king-man = queen-woman**

词向量隐含了表达性别概念

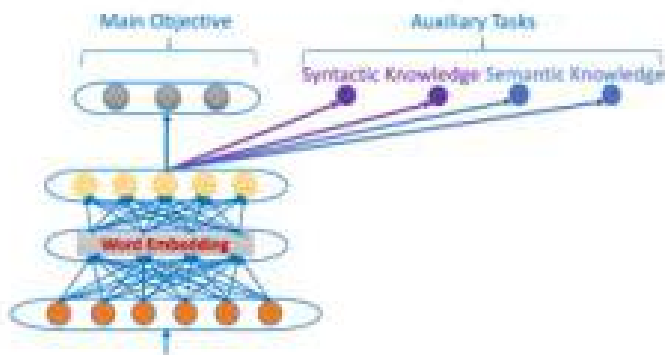


Fig. 4. Using syntactic and semantic knowledge as auxiliary objectives.

J Bian, B Gao, TY Liu, 2014

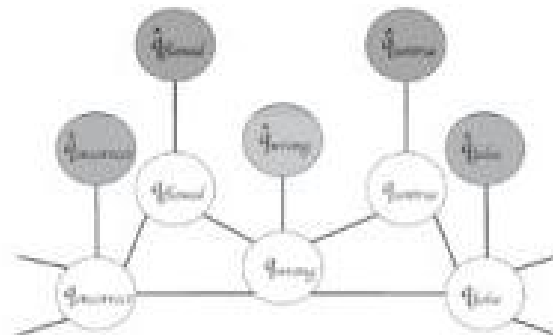
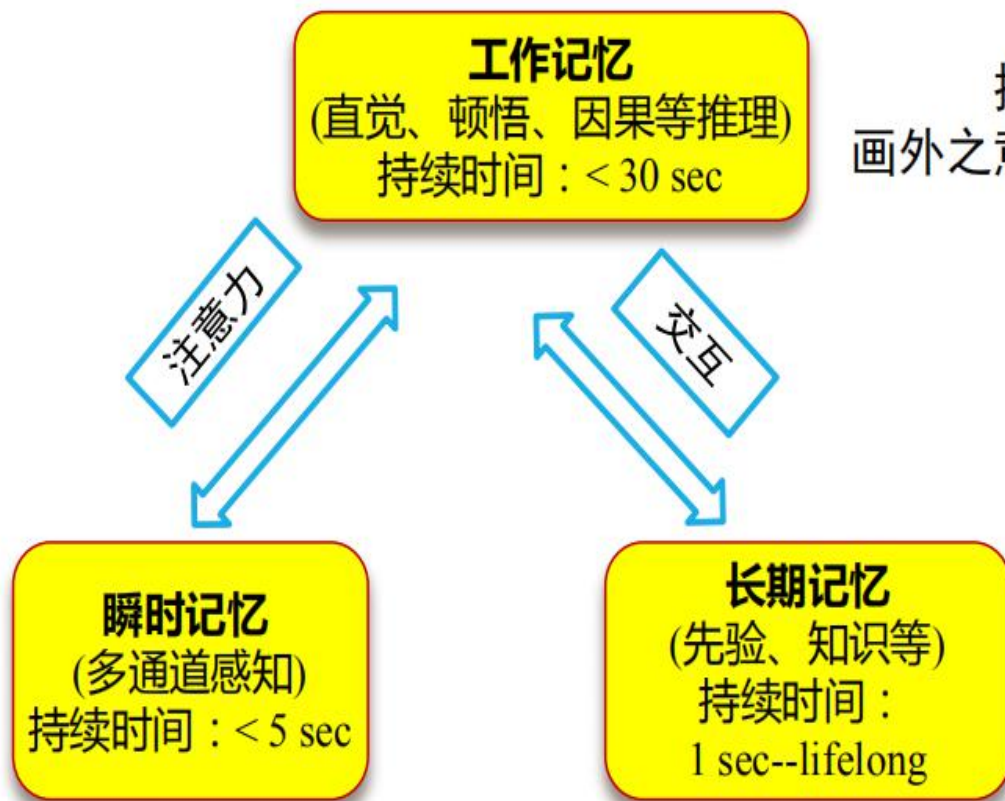


Figure 1: Word graph with edges between related words showing the observed (grey) and the inferred (white) word vector representations.

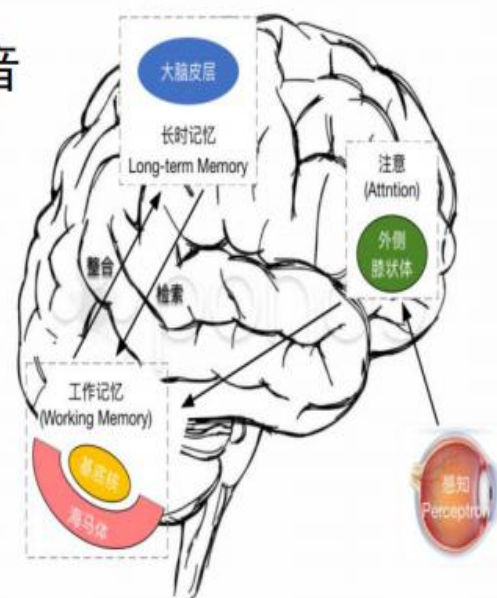
M Faruqui, J Dodge, SK Jauhar, et al., 2014

知识表示与今天主流的深度学习算法的关系

- 人脑对知识的加工与处理与记忆息息相关。记忆就是对信息的保存和再现能力。

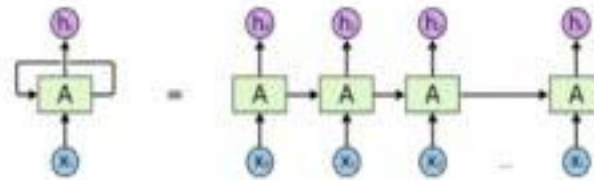


推理：
画外之意、弦外之音



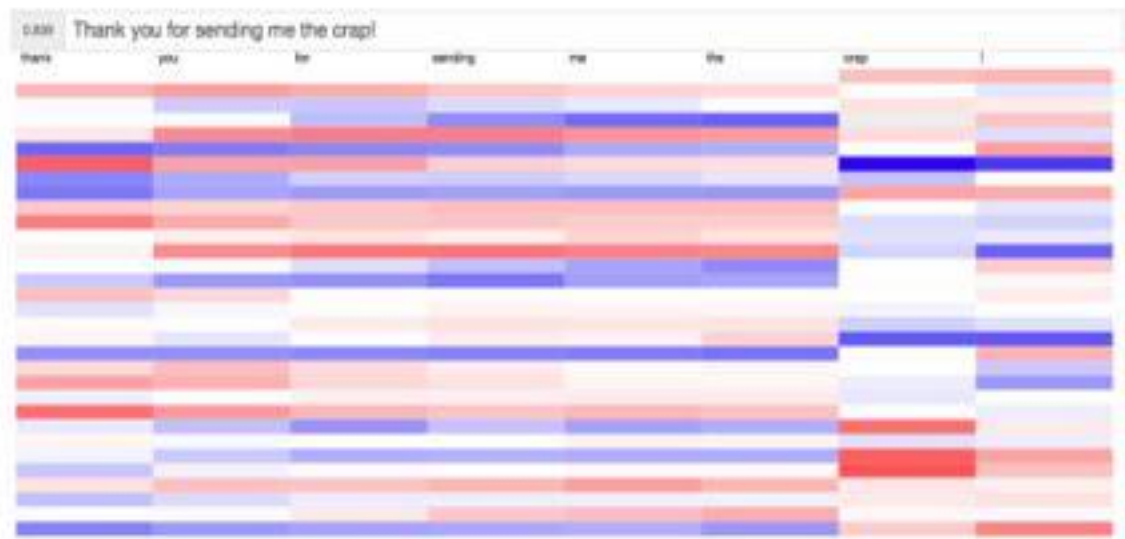
从知识 (Knowledge) 到记忆 (Memory)

- RNN / LSTM 网络中传递的信息到底是什么？
 - 是一种知识 (记忆) 表示吗？
- 大脑处理自然语言的时候，也有类似的信息传递吗？



An unrolled recurrent neural network.

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>



<https://medium.com/@plusepsilon/visualizations-of-recurrent-neural-networks-c18f07779d56>

RNN (循环神经网络):能够让信息在网络中再次循环的网络。在每个时间步长内重写自己的记忆。

LSTM (1997, 长短期记忆网络):应用专门的学习机制来记住、更新、聚焦于信息。可在更长的时期内跟踪信息。

例如，我们的视觉系统倾向于关注图像中辅助判断的部分信息，并忽略掉不相关的信息。同样，在涉及语言或视觉的问题中，输入的某些部分可能会比其他部分对决策更有帮助。

注意力 (Attention) 又是什么？

- 基于 Attention 的 seq2seq 是人类抽象能力的一种抽象吗？
 - 基于 Attention 的翻译算法，真的理解人类语言中的所谓“语义”吗？
 - 或者，在翻译任务中，“语义”其实是个伪命题？
- 翻译会是下一个 AlphaGo 吗？
 - N 年后：我们不再需要花生命的四分之一或更多的时间来学习外语.....

pork belly = delicious . || scallops? || I don't even like scallops, and these were a-m-a-z-i-n-g . || fun and tasty cocktails. || next time I in Phoenix, I will go back here. || Highly recommend.

Fig.1 Yelp评论情感分类中的注意建模实例题

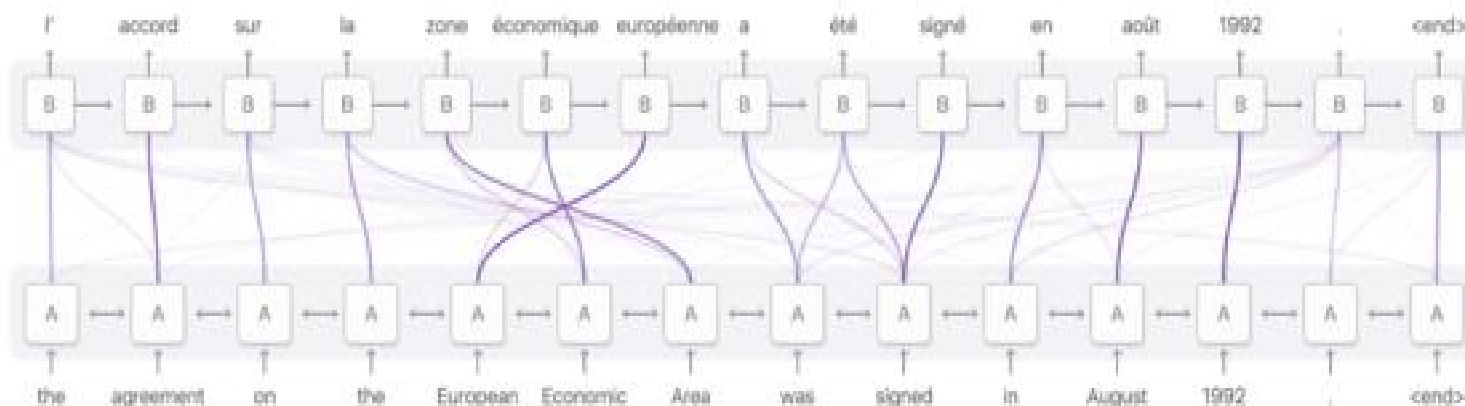


Diagram derived from Fig. 3 of Bahdanau, et al. 2014

知识表示与今天主流的深度学习算法的关系

我们距通用人工智能还有多远？

- 今天，有些问题甚至都没有准确的描述和定义
 - 常识：你有没有常识？
 - 情感：问世间，情为何物？直教人生死相许。
 - 自我意识：我是谁？我从哪里来？我要到哪里去？
- 乐观派：
 - 这些东西一样可以用神经网络来表示
- 悲观派：
 - 这些根本就不是数学模型或图灵机可以表示的东西



<http://www.kdnuggets.com/2017/04/cartoon-taxes-vs-ai.html>

通用人工智能是和知识表示密切联系在一起。

第2章 知识表示

✓ 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 框架表示法

■ 2.5 语义网络

- 了解知识及知识表示的基本概念
- 重点：一阶谓词逻辑表示法、产生式表示法、语义网络的具体表示形式
- 了解各种表示方法的优缺点、适宜的应用对象。

2.1 知识与知识表示的概念

● 2.1.1 知识的概念

● 2.1.2 知识的特性

● 2.1.3 知识的表示

◆ 知识的精确定义仍然没有定论。

◆ 知识的基本单位是概念。

◆ 知识本身也是一个概念。

人工智能的问题变成了如下3个问题：

- 1) 如何定义（或者表示）一个概念？
- 2) 如何学习一个概念？
- 3) 如何应用一个概念？

◆ 经典概念的例子——素数

- 概念的符号表示，即概念名称：素数/prime number
- 概念的内涵表示，由命题表示：只能被1和自身整除的自然数
- 概念的外延表示，由经典集合表示：{2, 3, 5, 7, 11, 13, 17,}

2.1.1 知识的概念

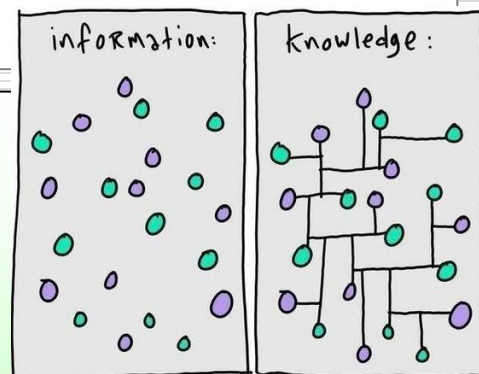
■ 信息、知识和智能

- **智能**：指人类在认识客观世界中，由思维过程和脑力活动所表现出的综合能力。
- **信息**：由数据表达的客观事实。
- **知识**：由智力对信息进行加工后所形成的对客观世界规律性的认识。

信息：形成知识的原料、智能的加工对象

知识：信息的关联、由智能加工后的产品

智能：信息到知识的一个加工器



知识 = 事实 + 规则 + 概念

事实：高老师今年60岁

规则：所有的老师都60岁退休

概念：老师、退休、年龄等

2.1.2 知识的特性

1. 相对正确性

$1+1=2$ (十进制)

$1+1=10$ (二进制)

任何知识都是在一定的条件及环境下产生的，在这种条件及环境下才是正确的。

2. 不确定性

知识状态：“真”、“假”、“真”与“假”之间的中间状态

① 随机性引起的不确定性

② 模糊性引起的不确定性

③ 经验性引起的不确定性

④ 不完全性引起的不确定性

今天阴天，可能下雨
(有可能为真)

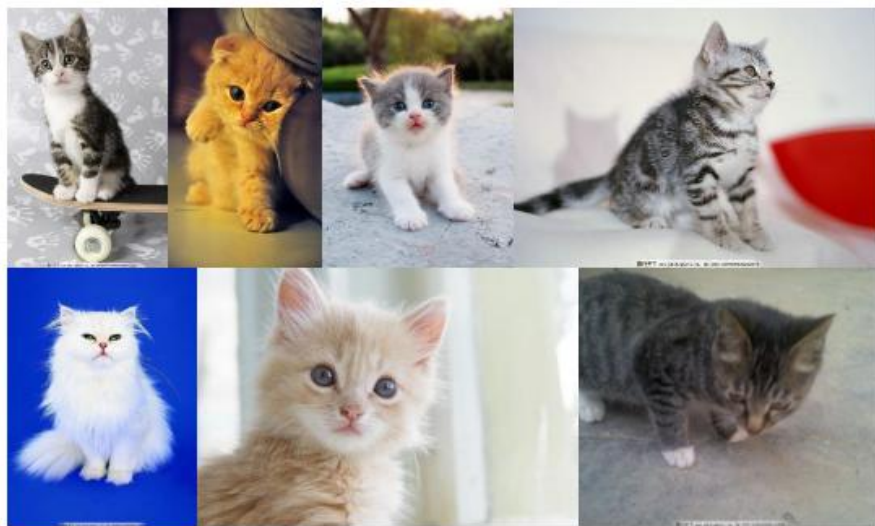
某人很高

2.1.2 知识的特性

3. 可表示性与可利用性

- 可表示性: 知识可以用适当形式表示出来, 如用文字、语言、图形、声音、神经网络等。
- 可利用性: 知识可以被利用。

1953年, 维特根斯坦
《哲学研究》: 不是任何概念都可以被精确定义, 例如猫、狗等。



2.1.3 知识的表示

■ **知识表示**：对知识的一种描述，或者一组约定，一种计算机可以接受的用于描述知识的数据结构。

简单而言，知识表示（KR）就是用易于计算机处理的方式来描述人脑的知识。

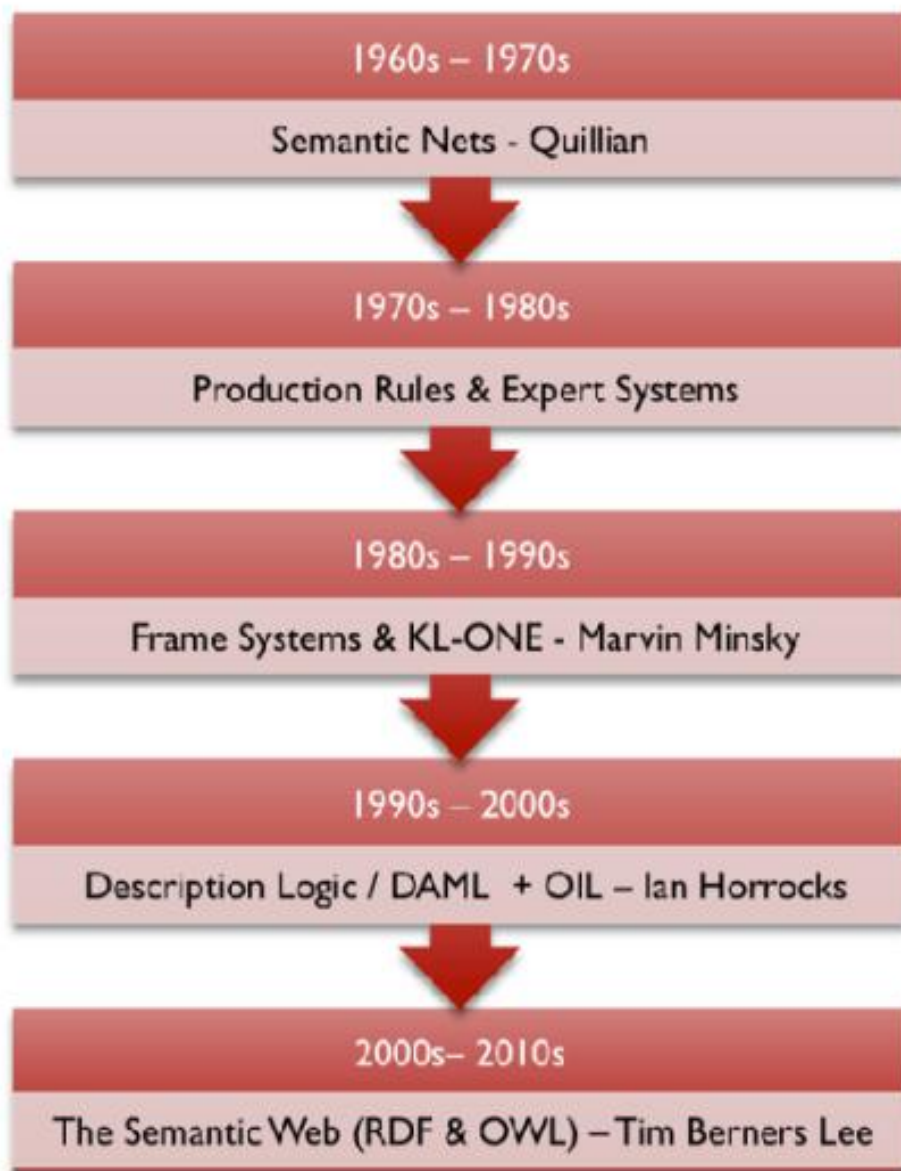
KR不是数据格式、不等同于数据结构、也不是编程语言，对于人工智能而言，数据与知识的区别在于KR支持推理。

■ **选择知识表示方法的原则：**

- ① 充分表示领域知识
- ② 有利于对知识的利用：支持推理、便于计算
- ③ 便于对知识的组织、维护与管理
- ④ 便于理解与实现：人可理解

.....

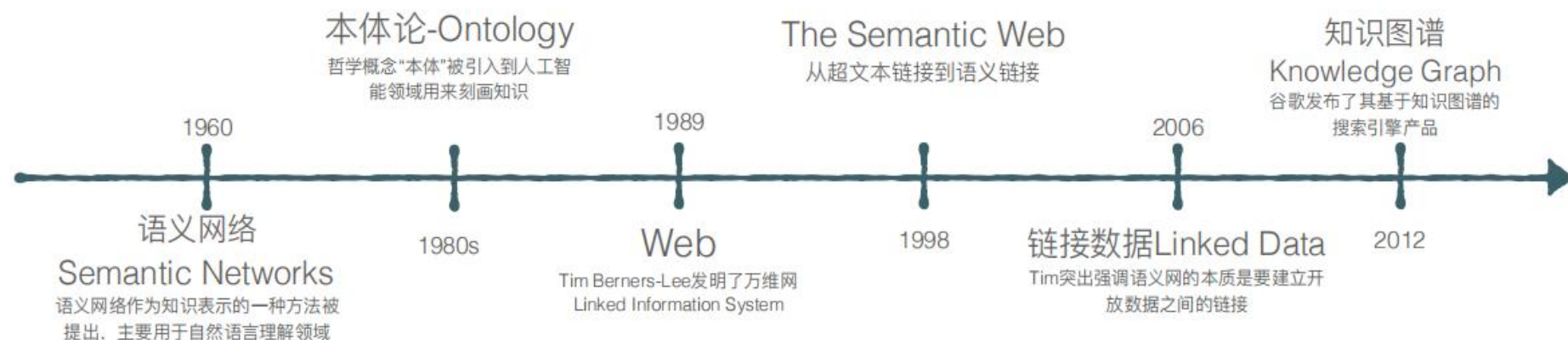
知识表示的研究



- 一阶谓词逻辑 (First-Order Logic)
 - Horn Logic
 - Description Logic
- 语义网络 (Semantic Net)
- 产生式规则 (Production Rule)
- 框架系统 (Framework)
- 描述逻辑 (Description Logic)
- 逻辑程序 (Logic Programming)

RDF (Resource Description Framework) 资源描述框架、**OWL (Web Ontology Language)** 网络本体语言都是面向Web设计实现的标准化的知识表示语言

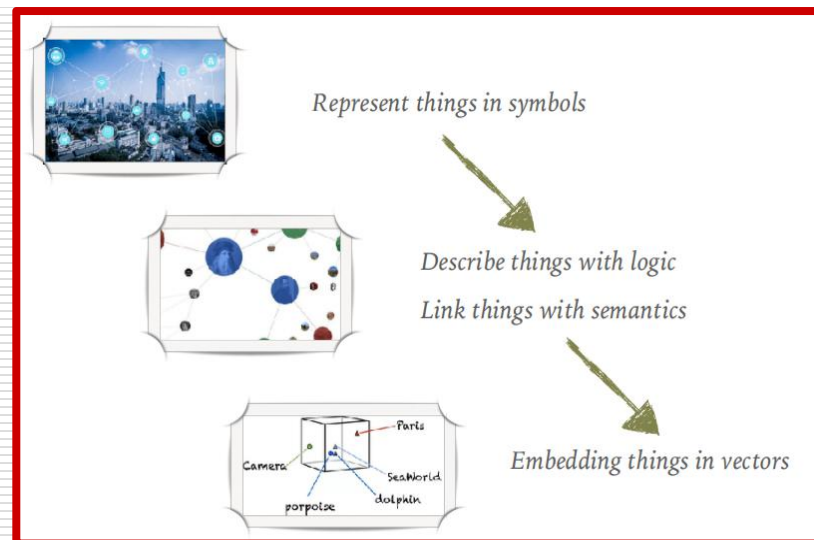
知识表示的发展



知识表示与知识库- Knowledge Representation / Knowledge Base

人工智能研究者陆续提出了大量知识表示的方法, 如框架系统、产生式规则、描述逻辑等。

RDF (Resource Description Framework) 资源描述框架、**OWL (Web Ontology Language)** 网络本体语言都是面向**Web**设计实现的标准化的知识表示语言



知识图谱得益于Web的发展 (更多的是数据层面), 有着来源于KR、NLP、Web、AI多个方面的基因。

知识图谱使传统知识表示和推理技术有了落脚点, 也为知识表示和推理带来了新的挑战。

第2章 知识表示

■ 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 框架表示法

■ 2.5 语义网络

以数理逻辑为基础,是到目前为止能够表达人类思维活动规律的一种最精确的形式语言。

2.2 一阶谓词逻辑表示法

2.2.1 命题

2.2.2 谓词

2.2.3 谓词公式

2.2.4 谓词公式的性质

2.2.5 一阶谓词逻辑知识表示

2.2.6 一阶谓词逻辑知识表示的

➤ 命题是一个**非真即假**的陈述句。

如下哪些语句不是命题？

1. 您去电影院吗？
2. 黑天鹅！
3. 哎呀，您.....
4. 比秃子多一根头发的人也是秃子。
5. $x=2$ 。
6. 两个奇数之和是奇数。
7. 欧拉常数是无理数。
8. 有缺点的战士毕竟是战士，完美的苍蝇毕竟是苍蝇。
9. 任何人都是会死的，苏格拉底是人，因此苏格拉底是会死的。
10. 如果下雨，则我打伞。
11. $x^2=0$ 当且仅当 $x=0$ 。
12. 李白要么擅长写诗，要么擅长喝酒。
13. 李白既不擅长写诗，又不擅长喝酒。
14. 李白是诗人，杜甫也是诗人。

命题符号化表示

6. 两个奇数之和是奇数。

令 p : 两个奇数之和是奇数。其真值为假。

7. 欧拉常数是无理数。

令 p : 欧拉常数是无理数。其真值确定，现在未知。

8. 有缺点的战士毕竟是战士，完美的苍蝇毕竟是苍蝇。

令 p : 有缺点的战士毕竟是战士。
 q : 完美的苍蝇毕竟是苍蝇，则原命题为 $p \wedge q$ ，其真值为真。

9. 任何人都是会死的，苏格拉底是人，因此苏格拉底是会死的。

令 p : 任何人都是会死的。
 q : 苏格拉底是人，
 r : 苏格拉底是会死的，则原命题为 $(p \wedge q) \rightarrow r$ 。

- 命题是一个非真即假的陈述句。
- 命题：简单命题、复合命题（简单命题+逻辑连接词）
- 命题逻辑无法反映所描述事物的结构及逻辑特征，也不能表述不同事物间的共同特征。

10. 如果下雨，则我打伞。

11. $x^2=0$ 当且仅当 $x=0$ 。

令 p : $x^2=0$, q : $x=0$. 则原命题为 $p \leftrightarrow q$ 。

12. 李白要么擅长写诗，要么擅长喝酒。

令 p : 李白擅长写诗, q : 李白擅长喝酒。则原命题为 $p \vee q$ 。

13. 李白既不擅长写诗，又不擅长喝酒。

14. 李白是诗人，杜甫也是诗人。

令 p : 李白是诗人, q : 杜甫是诗人. 则原命题为 $p \wedge q$ 。

2.2.2 谓词 (predicate)

■ 谓词的一般形式: $P(x_1, x_2, \dots, x_n)$

■ 谓词名 P : 刻划个体的性质、状态或个体间的关系;

■ 个体 x_1, x_2, \dots, x_n : 某个独立存在的事物或者某个抽象的概念。

■ 1) **Smith是一个工程师**: 一元谓词 **engineer (Smith)**

■ 2) **Bill和Richard是朋友**: 二元谓词 **friend(Bill, Richard)**

■ 3) **小李的父亲是一个工程师**: **engineer (father (Li))**

■ 4) **Smith是IBM公司的一个工程师**:

二阶谓词 **works(engineer (Smith), IBM)**

■ **一阶谓词**: 个体可以是常量、变元 (变量) 和**函数**。

2.2.2 谓词 (predicate)

- 谓词的一般形式: $P(x_1, x_2, \dots, x_n)$
- 谓词名 P : 刻划个体的性质、状态或个体间的关系;
- 个体 x_1, x_2, \dots, x_n : 某个独立存在的事物或者某个抽象的概念。
 - 一阶谓词: 个体可以是常量、变元 (变量) 和函数。
 - 个体域: 个体变元的取值范围, 一般用 D 表示; 可以是无限的, 如自然数集合, 也可以是有限的, 如 $\{1, 2, 3, 4\}$ 。
 - 注意谓词和函数的区别: 函数是把个体域中的个体映射到另一个个体, 如 $\text{father}(\text{Li})$ 函数值代表一个人, 函数没有真值, 而谓词有真值, 例如 $\text{engineer}(\text{Smith})$ 代表真或假

2.2.2 谓词 (predicate)

- 谓词的一般形式: $P(x_1, x_2, \dots, x_n)$
- 谓词名 P : 刻划个体的性质、状态或个体间的关系;
- 个体 x_1, x_2, \dots, x_n : 某个独立存在的事物或者某个抽象的概念。

(1)能阅读者是识字的;

(2)海豚不识字;

(3)有些海豚是聪明的;

归结原理: 一种基于一阶谓词
逻辑知识表示的演绎推理方法

请用归结原理证明: 有些聪明者并不能阅读。

已知, 谓词 $R(x)$ 表示 x 能阅读, $L(x)$ 表示识字,

$D(x)$ 表示 x 是海豚, $I(x)$ 表示聪明的

2.2.3 谓词公式

(1)能阅读者是识字的;(2)海豚不识字;(3)有些海豚是聪明的;请用归结原理证明:有些聪明者并不能阅读。定义谓词 $R(x)$ 表示 x 能阅读, $L(x)$ 表示识字, $D(x)$ 表示 x 是海豚, $I(x)$ 表示聪明的

谓词公式 (合式公式)

- (1) 单个谓词是谓词公式, 称为原子谓词公式;
- (2) 若 A 是谓词公式, 则 $\neg A$ 也是谓词公式;
- (3) 若 A, B 都是谓词公式, 则 $A \wedge B, A \vee B, A \rightarrow B, A \leftrightarrow B$ 也都是谓词公式。
- (4) 若 A 是谓词公式, 则 $(\forall x)A, (\exists x)A$ 也是谓词公式;
- (5) 有限步应用(1)-(4)生成的公式也是谓词公式。

连接词的优先级别: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

全称量词 \forall : $\forall X$ 读作“任取 X ”、“所有 X ”、“任一 X ”。

存在量词 \exists : $\exists X$ 读作“存在 X ”、“一些 X ”、“至少有一个 X ”。

2.2.3 谓词公式

全称量词 \forall : $\forall X$ 读作“任取 X ”、“所有 X ”、“任一 X ”。

存在量词 \exists : $\exists X$ 读作“存在 X ”、“一些 X ”、“至少有一个 X ”。

全称量词和存在量词出现的次序将影响命题的意思。

■ $(\forall x)(\exists y)(Employee(x) \rightarrow Manager(y, x))$:

“每个雇员都有一个经理。”

■ $(\exists y)(\forall x)(Employee(x) \rightarrow Manager(y, x))$:

“有一个人是所有雇员的经理。”

注意: $Manager(x, y)$: x 是 y 的经理

$Manager(y, x)$: y 是 x 的经理

2.2.3 谓词公式

- **量词的辖域**：量词后面的单个谓词或用括弧括起来的谓词公式。
- 辖域内与量词中同名的变元称为**约束变元**，不同名的变元称为**自由变元**。

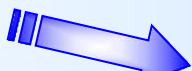
- 例如：1) $\forall x F(x) \rightarrow G(e)$
2) $(\exists x)(P(x, y) \rightarrow Q(x, y)) \vee R(x, y)$
3) $\forall x \exists y R(x, y)$

■ 注意： $(\forall x) P(x, y)$

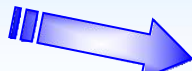
- 约束变元换名时，不能和自由变元同名；
- 自由变元换名时，也不能和约束变元同名



$(\forall z) P(z, t)$



$(\forall y) P(y, y)$ 错误！



$(\forall x) P(x, x)$ 错误！



2.2.4 谓词公式的性质

- 谓词公式在个体域上的解释：针对个体变量和函数在个体域中的取值，为谓词公式指派真值。
- 对于每一个解释，谓词公式都可求出一个真值（T或F）。

friend (mary, X), $D_x = \{ \text{john, kate, mike} \}$

friend (mary, john) T

friend (mary, kate) T

friend (mary, mike) F

2.2.5 一阶谓词逻辑知识表示方法

- 事实：谓词公式的与/或形（用 \wedge 、 \vee 连接的公式）表示。
- 规则：谓词公式的蕴含式。

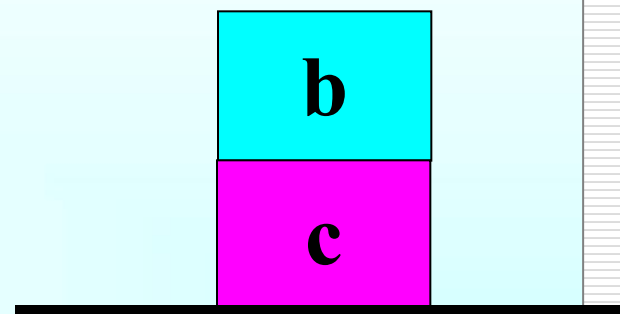
- 例，机械手摆积木问题，要处理的世界有一张桌子，桌上堆放若干相同的方积木块。设有2个积木块 b, c ，它们直接的位置关系可用下列谓词公式表示：

on (b,c): b 在 c 之上； **clear(b):** b 是无束缚的；

ontable (c): c 在桌子之上；

$\forall X(\neg \exists Y \text{ on}(Y,X) \rightarrow \text{clear}(X))$:

对于所有的 X ，如果不存在一个 Y 在 X 上，那么 X 是无束缚的。



2.2.5 一阶谓词逻辑知识表示方法

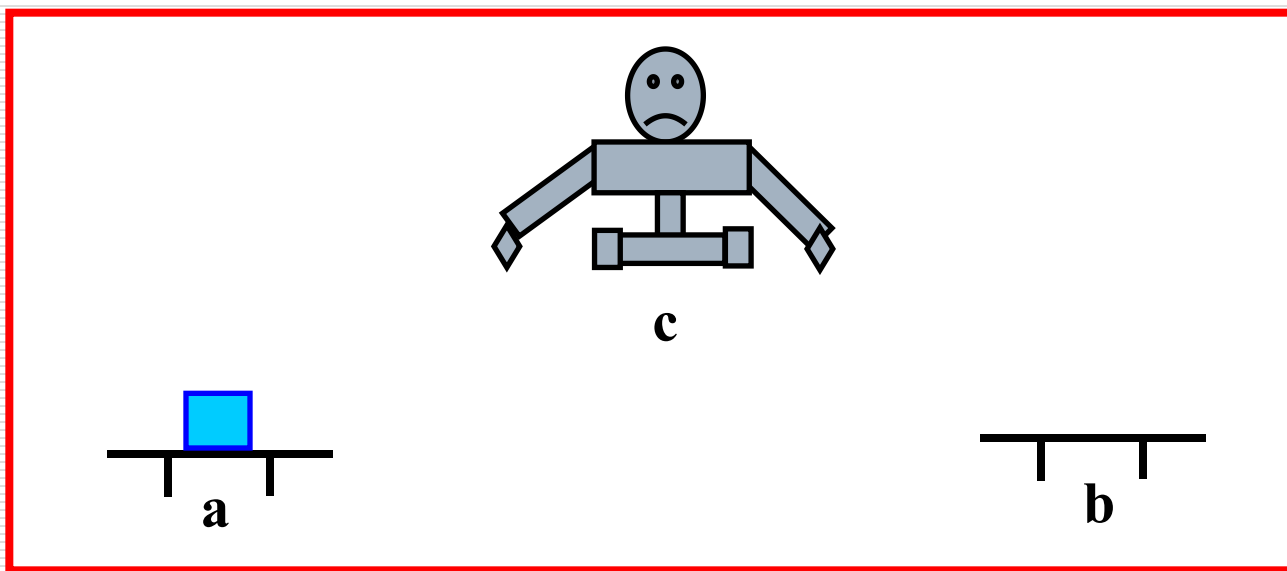
- 事实：谓词公式的与/或形（用 \wedge 、 \vee 连接的公式）表示。
- 规则：谓词公式的蕴含式。

■ 谓词公式表示知识的步骤：

- ① 定义谓词；
- ② 个体变元赋值；
- ③ 用连接词连接各个谓词，形成谓词公式。

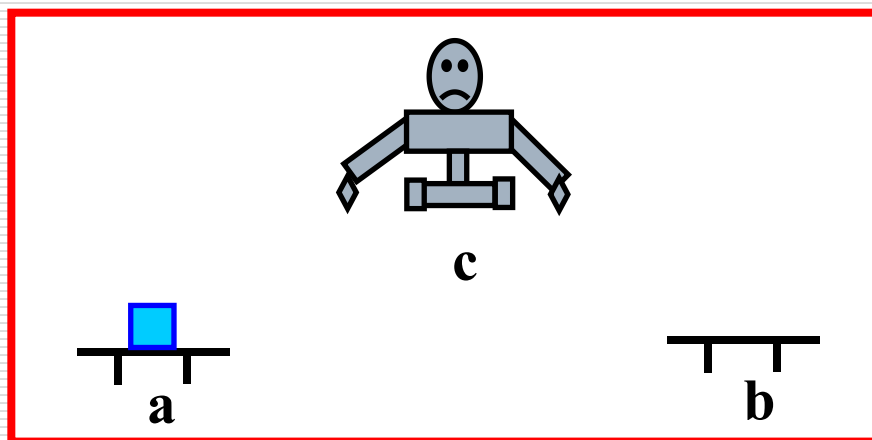
■ 例：机器人行动规划的谓词逻辑表达

- 设在房间c处有一个机器人(robot)，在a和b处各有一张桌子，a桌上有一个箱子(box)。让机器人从c处出发，把桌子a上的箱子移到桌子b上，然后回到c处，制定相应的行动规划。



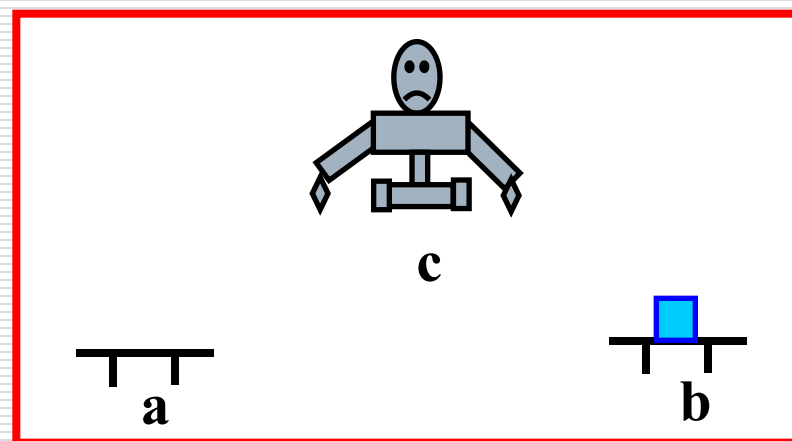
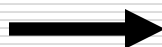
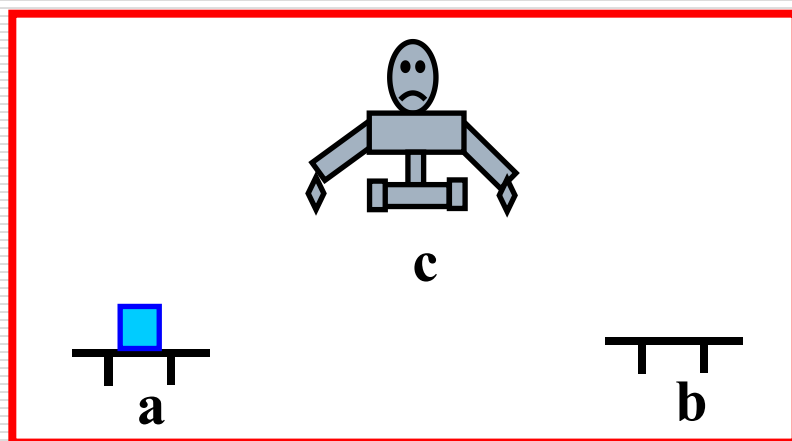
■ 定义描述事物状态、位置的谓词:

谓 词	变元的个体域
table (X): X 是桌子	DX = { a, b }
on (W, X): W 在 X 的上面	DY = { robot }
empty (Y): Y 手中是空的	DZ = { a, b, c }
at (Y, Z): Y 在Z 的附近	DW = { box }
holds (Y, W): Y 拿着 W	

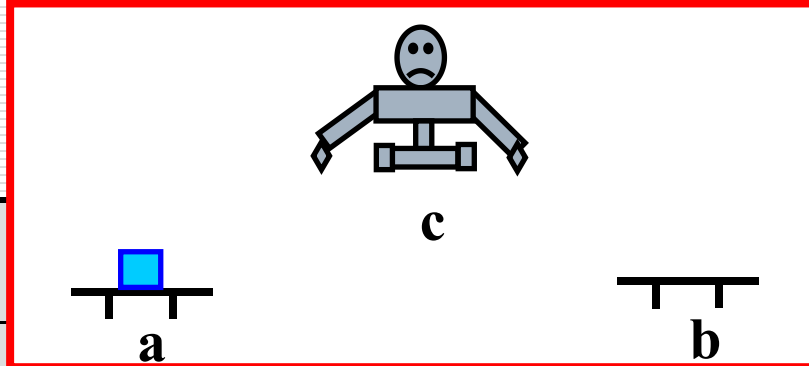


问题的谓词公式表示:

初 始 状 态	目 标 状 态
<p>at (robot, c)</p> <p>empty (robot)</p> <p>on (box, a)</p> <p>table (a)</p> <p>table (b)</p>	<p>at (robot, c)</p> <p>empty (robot)</p> <p>on (box, b)</p> <p>table (a)</p> <p>table (b)</p>

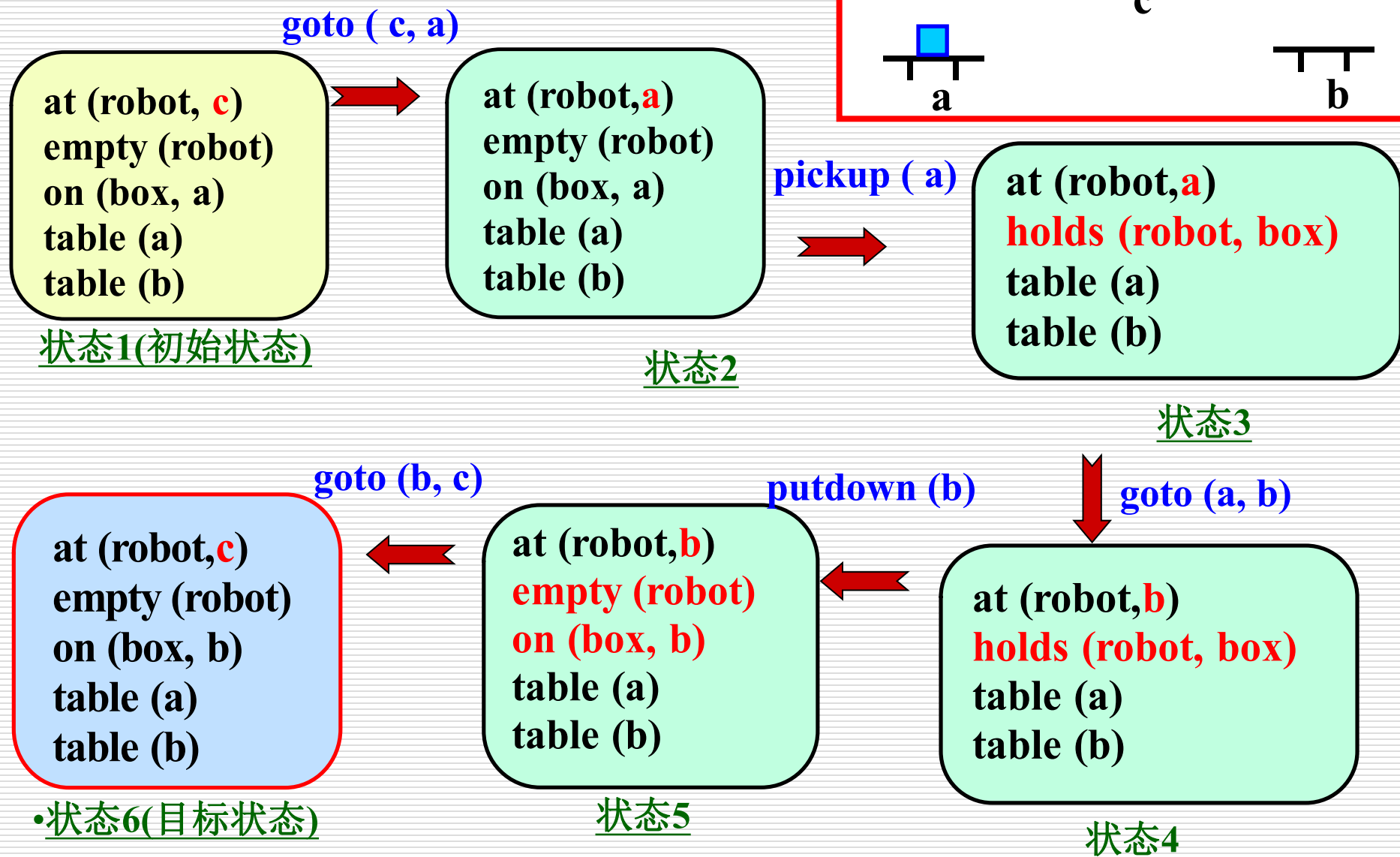
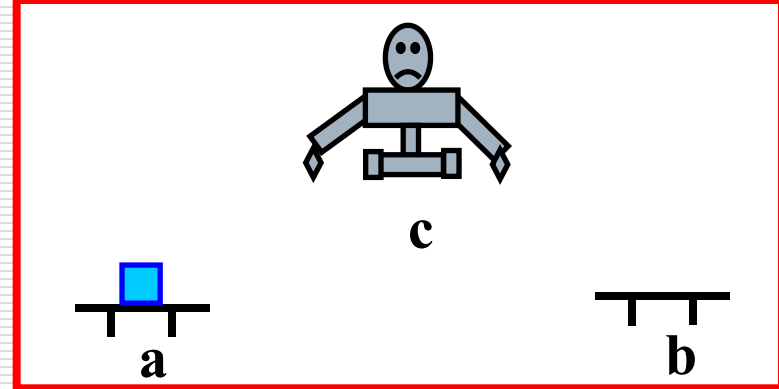


■ 定义描述动作的谓词:



操 作	条件与动作
goto (Z, U): 从 Z 处走到 U 处	条件: $\text{at}(\text{robot}, Z)$ 动作: { 删除 $\text{at}(\text{robot}, Z)$ 增加 $\text{at}(\text{robot}, U)$
pickup (Z): 在 Z 处拿起盒子	条件: $\text{on}(\text{box}, Z) \wedge \text{table}(Z) \wedge \text{at}(\text{robot}, Z) \wedge \text{empty}(\text{robot})$ 动作: { 删除 $\text{empty}(\text{robot}) \wedge \text{on}(\text{box}, Z)$ 增加 $\text{holds}(\text{robot}, \text{box})$
putdown (Z) 在 Z 处放下盒子	条件: $\text{at}(\text{robot}, Z) \wedge \text{table}(Z) \wedge \text{holds}(\text{robot}, \text{box})$ 动作: { 删除 $\text{holds}(\text{robot}, \text{box})$ 增加 $\text{empty}(\text{robot}) \wedge \text{on}(\text{box}, Z)$

机器人的行动规划过程:



2.2.6 一阶谓词逻辑表示法的特点

□ 应用领域:

- ◆ 编程语言: **Prolog**、**Datalog**、**pyDatalog**、**SQL**语言等
- ◆ 机器定理证明
- ◆ 问题求解
- ◆ 本体知识库构建: 描述逻辑
- ◆ 知识图谱补全或扩充: 归纳逻辑程序设计
- ◆ 专家系统: 医疗诊断等

一阶谓词逻辑的典型例子：Horn Logic

- Horn Logic是一阶谓词逻辑的子集，主要特点是表达形式简单，复杂度低。著名的Prolog语言就是基于Horn逻辑设计实现的。
- 核心概念：
 - 原子Atoms
 - $p(t_1, t_2, \dots, t_n)$, p 是谓词, n 是目, t_i 是项 (变量或者常量)
 - 例子: has_child(Helen, Jack)
 - 规则Rules
 - 由原子构建: $H:- B_1, B_2, \dots, B_m$.
 - H 与 B_1, B_2, \dots, B_m 是原子
 - H 是头部原子, B_1, B_2, \dots, B_m 是体部原子
 - 例子: has_child(X, Y) :- has_son(X, Y)
 - 事实Facts:
 - 没有体部且没有变量的规则
 - 例子: has_child(Helen, Jack) :-

一阶谓词逻辑的子集举例：Description Logic

- 描述逻辑是一阶谓词逻辑的可判定子集，主要用于描述本体概念和属性，对于本体知识库的构建提供了便捷的表达形式，是与知识图谱最密切相关的知识表示方法之一。
- 核心概念：
 - 概念Concepts——解释为一个领域的子集
 - 例如：学生，已婚者： $\{x \mid \text{Student}(x)\}$, $\{x \mid \text{Married}(x)\}$
 - 关系Relations——解释为指该领域上的二元关系
 - 例如：示例：朋友，爱人： $\{ \langle x,y \rangle \mid \text{friend}(x,y) \}$, $\{ \langle x,y \rangle \mid \text{loves}(x,y) \}$
 - 个体Individuals——一个领域内的实例
 - 例如：小明，小红： $\{\text{Ming, Hong}\}$

一阶谓词逻辑的子集举例：Description Logic

- 描述逻辑的知识库 $O := \langle T, A \rangle$ ，T即Tbox，A即Abox
- TBox包含内涵知识，描述概念的一般性质
 - 定义：引入概念以及关系的名称
 - 例如：Mother, Person, has_child
 - 包含：声明包含关系的公理
 - 例如：Mother $\sqsubseteq \exists$ has_child.Person
- ABox包含外延知识 (又称断言知识)，描述论域中的特定个体。
 - 概念断言——表示一个对象是否属于某个概念
 - 例如：Mother(Helen), Person(Jack)
 - 关系断言——表示两个对象是否满足一定的关系
 - 例如：has_child(Helen, Jack)

2.2.6 一阶谓词逻辑表示法的特点

优点	缺点
自然性	无法表示不确定性知识
接近自然语言，容易接受	难以表示启发性知识及元知识
精确性	组合爆炸，经常出现事实、规则等的组合爆炸
用于表示精确知识	效率低，推理复杂度通常较高
严密性	是以人的逻辑为主导的表示方式，可能并不适合机器，机器可能有自己的逻辑表示。
有严格的形式定义和推理规则	<ul style="list-style-type: none">➤ 启发式知识：加快问题求解的知识➤ 元知识：关于如何运用已有的知识进行问题求解的知识，也称为关于知识的知识。
易实现性	

第2章 知识表示

■ 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 框架表示法

■ 2.5 语义网络



1. 中国大学MOOC：第2讲单元测试
2. 网络教学平台：W1-2作业



2.3 产生式表示法

- 2.3.1 产生式 (Production)
- 2.3.2 产生式系统 (Production System)
- 2.3.3 产生式系统的例子——动物识别系统
- 2.3.4 产生式表示法的特点

- 什么是产生式？
- 如何用产生式表示确定性知识以及不确定性知识？
- 如何用产生式系统求解问题？
- 产生式表示法的优缺点是什么？
- 产生式表示法适合描述什么样的知识？

2.3.1 产生式

- **产生式 (Production)**：1943年，美国数学家Post 首先提出，用于计算形式体系中的**符号串替换**运算。
- 1954, Markov 提出了**产生式系统的控制策略**，根据规则的优先级来确定其执行的顺序。
- 1972年，Newell和Simon在研究人类的认知模型中开发了**基于规则的产生式系统**。
- **产生式系统**是一种更广泛意义的规则系统，早期的专家系统多数是基于**产生式系统**：
 - Feigenbaum研制的化学分子结构专家系统**DENDRAL**
 - Shortliffe研制的诊断感染性疾病的专家系统**MYCIN**

1. 事实性知识的产生式表示

(1) 确定性事实性知识

- 三元组表示：（对象，属性，值）
或（关系，对象1，对象2）
- john的年龄是20岁：（john, age, 20）
- john和mary是朋友：（friend, john, mary）

(2) 不确定性事实性知识

也称确定性因子（CF，
Certainty Factor）

- 四元组表示：（对象，属性，值，置信度）
或（关系，对象1，对象2，置信度）
- john的年龄很可能是20岁：（john, age, 20, 0.8）
- john和mary不太可能是朋友：（friend, john, mary, 0.1）

2.3.1 产生式

2. 规则知识的产生式表示

(1) 确定性规则知识

■ 基本形式: IF P THEN Q

或 $P \rightarrow Q$

■ P : 产生式的前提, Q : 一组结论或操作

产生式除了逻辑蕴含式, 还包括各种操作、变换、算子、函数等。

例如, “如果炉温超过上限, 则立即关闭风门” 是一个产生式, 但不是蕴含式。

(2) 不确定性规则知识

■ 基本形式: IF P THEN Q (置信度)

或 $P \rightarrow Q$ (置信度)

IF 本微生物的染色斑是革兰氏阴性

本微生物的形状呈杆状

病人是中间宿主

THEN 该微生物是绿脓杆菌, 置信度为 $CF=0.6$

MYCIN专家系统:



2.3.1 产生式

■ 产生式规则例子：

- 图搜索中的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等
- 体育比赛中的规则、国家的法律条文、单位的规章制度等

■ 下列自然语言表达可归结为产生式的知识表达形式：

- 天下雨，地上湿； “原因→结果”
- 如果把水加热到0°以上，冰就会溶化为水； “条件→结论”
- 夜来风雨声，花落知多少； “事实→进展”
- 若能找到一根合适的杠杆，就能撬起那座大山； “前提→操作”
- 才饮长沙水，又食武昌鱼； “事实→进展”
- 刚才开机了，意味着发出了捕获目标图像的信号。 “情况→行为”

2.3.2 产生式系统

- **规则库（知识库）**：存储有关问题的状态转移、性质变化等领域规则。

规则库

推理机

综合数据库

- **推理机(控制系统)**：负责问题求解。

产生式系统的组成

- **综合数据库(动态数据库、事实库、工作存储器、上下文、黑板等)**：存储初始事实数据、中间结果和最后结果等。

2.3.2 产生式系统

■ 基于产生式规则的一般推理模式

利用产生式规则实现逻辑推理的方法是当有事实能与某规则的前提匹配(即规则的前提成立)时,就得到该规则后部的结论(即结论也成立)。

基于产生式规则的逻辑推理模式: 逻辑上所说的假言推理(对常量规则而言)和三段论推理(对变量规则而言)

$$A \rightarrow B$$

$$A$$

$$B$$

大前提: 一个产生式规则

小前提: 证据事实

产生式系统中的推理是更广义的推理

2.3.2 产生式系统

□ 推理机（控制系统）的工作：

- 1) **匹配**：按一定的策略从规则库中选择规则，并把规则的前提条件与综合数据库中的已知事实进行比较。
- 2) **冲突消解**：调用**冲突解决策略（3.1.4节）**进行消解，从匹配成功的多条规则中选出一条执行。

R1 IF 第四次进攻
and 前进的距离小于10码
THEN 踢悬空球

R2 IF 第四次进攻
and 前进的距离小于10码
and 进攻位置在离对方球门30码内
THEN 射门



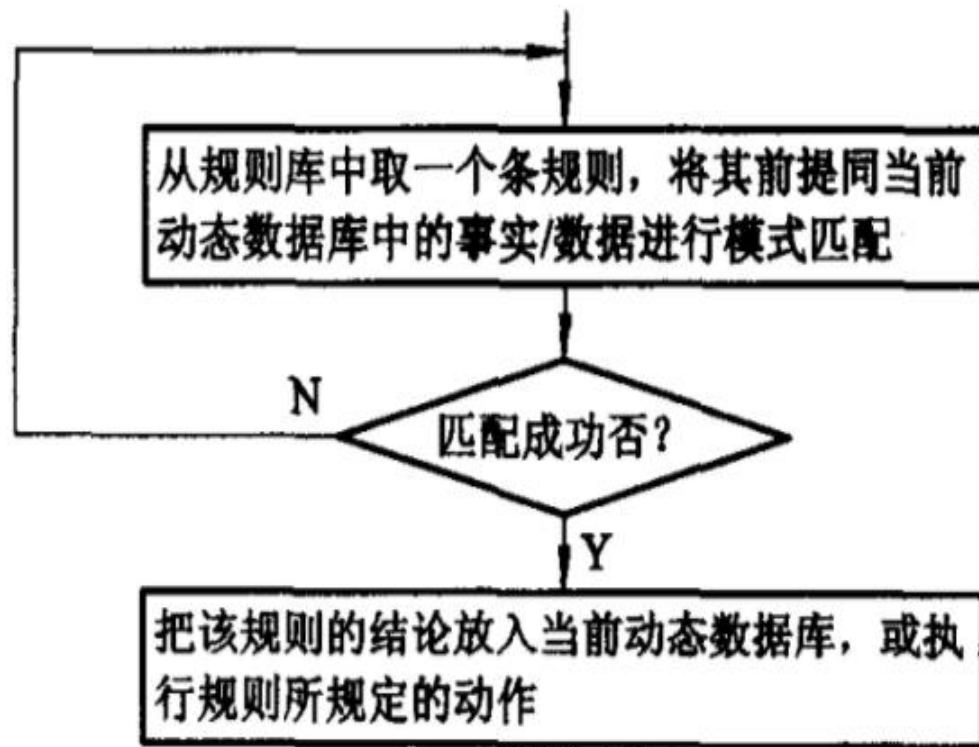
规则排序
专一性排序
就近排序
规模排序
.....

2.3.2 产生式系统

□ 推理机（控制系统）的工作：

3) **执行**：如果规则的右部是一个或多个结论，则把结论加入到综合数据库；如果是一个或多个操作，则执行操作。

对于不确定性知识，在执行每一条规则时还要按一定的算法**计算结论的不确定性**。



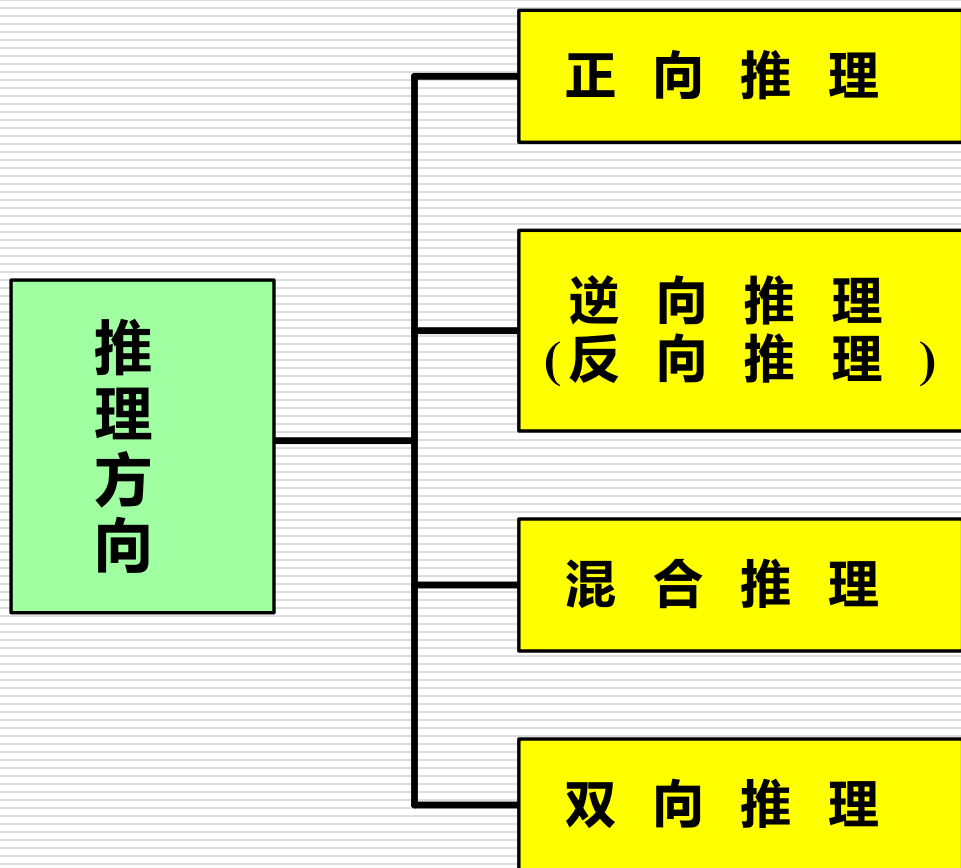
4) 检查综合数据库中是否包含了最终结论，决定是否停止系统的运行。

2.3.3 产生式系统的例子——动物识别系统

- 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。



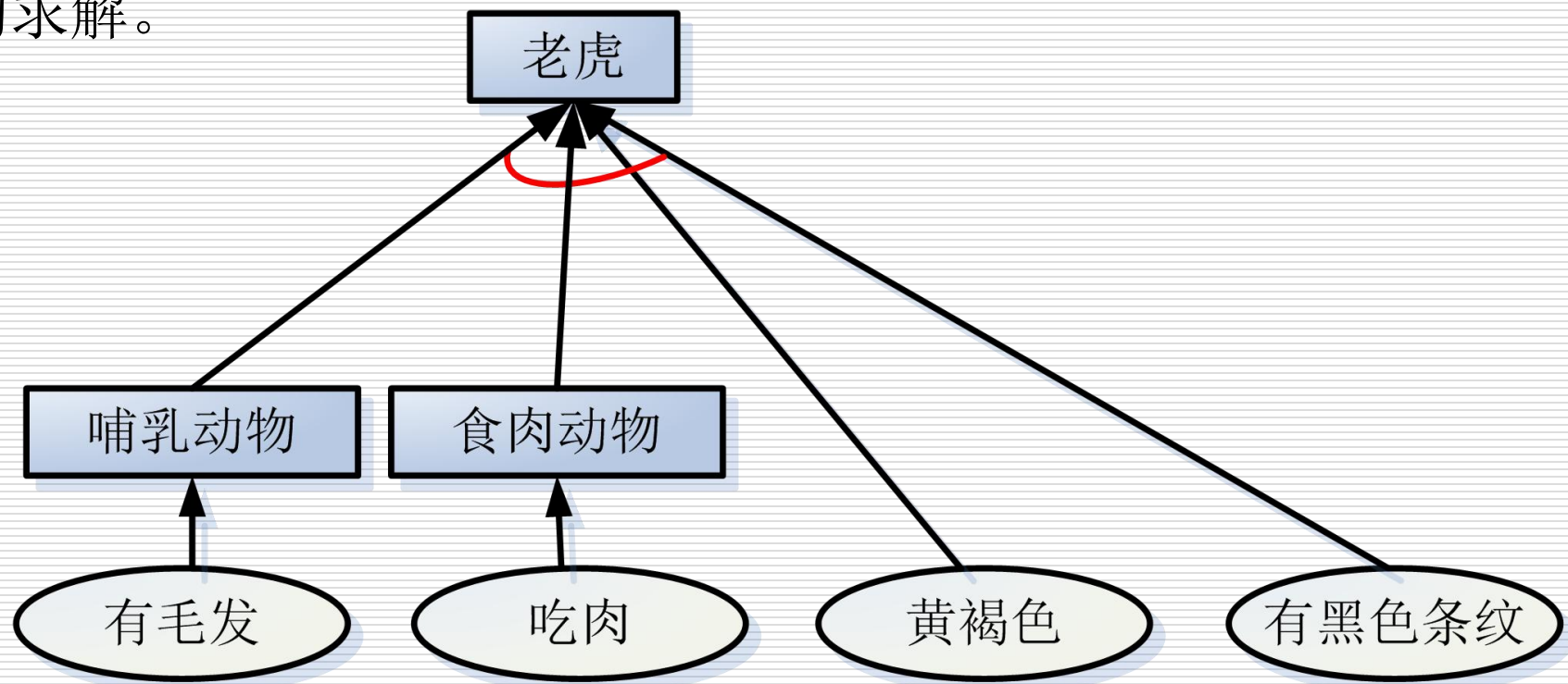
推理方向（3.1.3）



推理方向（3.1.3）

1.正向推理

- 正向推理简单，易实现，但目的性不强，效率低。
- 正向推理主要用于已知初始数据，而无法提供推理目标，或解空间很大的一类问题，如监控、预测、规划、设计等问题的求解。

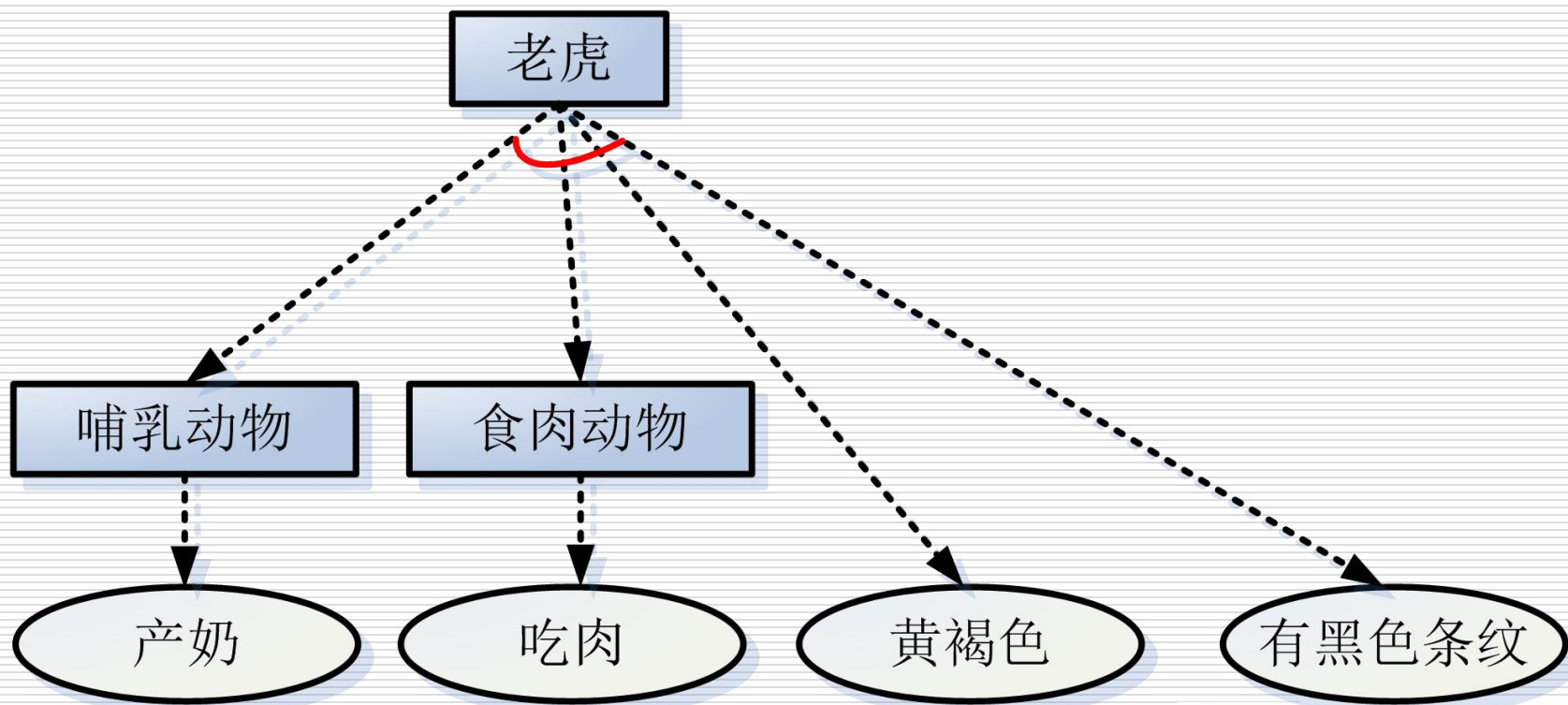


关于“老虎”的正向推理树

推理方向（3.1.3）

2. 逆向推理

- 逆向推理：目的性强，利于向用户提供解释，但选择初始目标时具有盲目性，比正向推理复杂。
- 反向推理主要用于结论单一或已知目标结论而要求证实的系统，如选择、分类、故障诊断等问题的求解。



关于“老虎”的反向推理树

推理方向 (3.1.3)

3. 混合推理

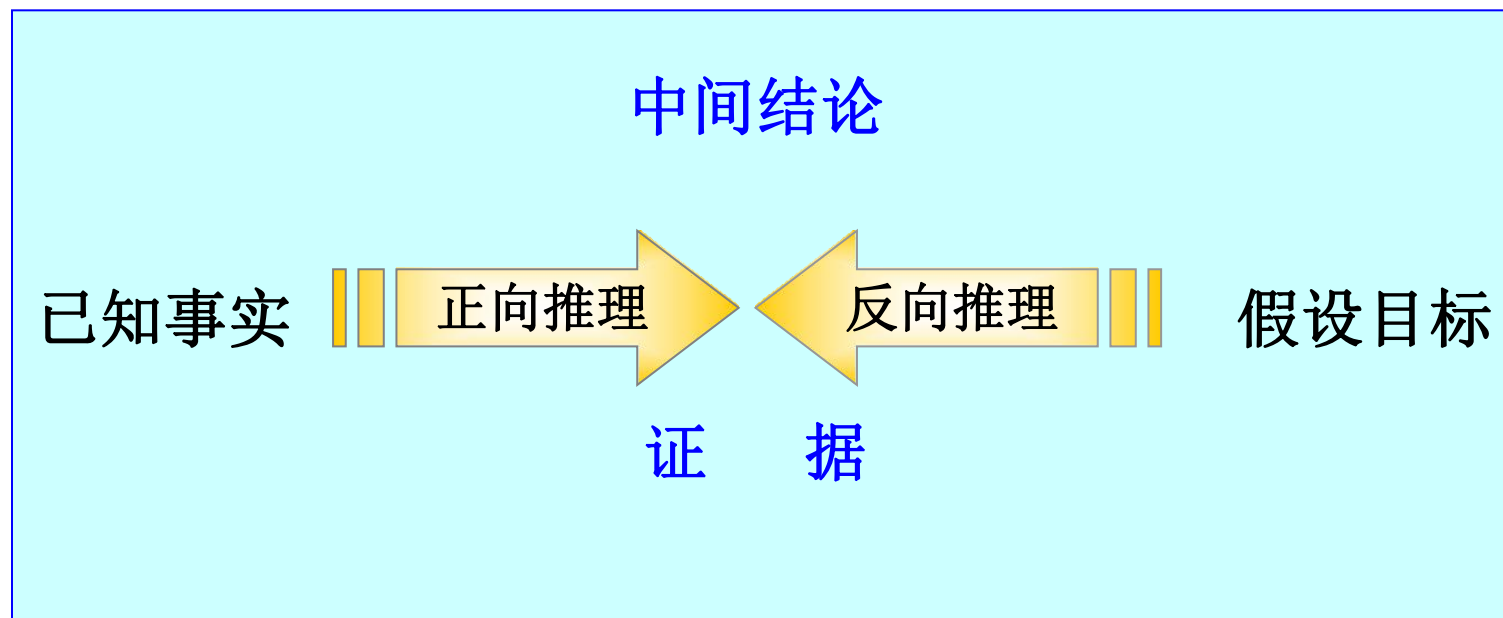
- 正向推理：盲目、效率低
- 逆向推理：若提出的假设目标不符合实际，会降低效率。
- 正反向混合推理：
 - 1) 先正向后逆向：先进行正向推理，帮助选择某个目标，即从已知事实演绎出部分结果，然后再用逆向推理证实该目标或提高其可信度；
 - 2) 先逆向后正向：先假设一个目标进行逆向推理，然后再利用逆向推理中得到的信息进行正向推理，以推出更多的结论。



推理方向（3.1.3）

4. 双向推理

- **双向推理**：正向推理与逆向推理同时进行，且在推理过程中的某一步骤上“碰头”的一种推理。



2.3.3 产生式系统的例子——动物识别系统

■ 产生式系统求解问题的一般步骤（正向推理为例）：

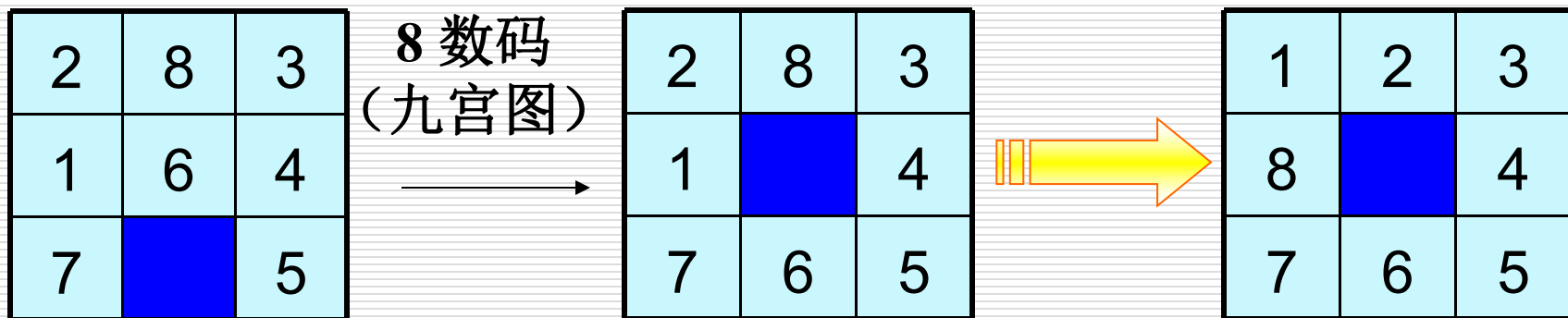
- 1) 初始化综合数据库。
- 2) 检查规则库中是否有未使用规则。
- 3) 若存在未用规则前提与已知事实相匹配则转4) 否则转6)。
- 4) 若存在多条匹配规则，则冲突消解，然后执行所选择的规则，更新综合数据库，并标记所用规则。
- 5) 综合数据库是否包含解。若是，则终止求解，否则转2)。
- 6) 要求提供更多的关于问题的信息，若不能，则求解失败，否则更新综合数据库并转2)。
- 7) 若规则库中不再有未使用规则，则终止问题求解。

问题：匹配执行的规则
可能会阻挠或延迟找到
最终解

产生式系统的控制策略（补充）

■ 基本产生式系统的控制策略：

- **不可撤回策略**：搜索过程一直往前进行，不允许撤回已经选择的规则。——“**一直往前走**”不回头
- **回溯策略（“碰壁回头”）**：在从冲突集选择一条规则执行时，建立回溯点，若以后发现此规则会阻挠或延迟到达目标，则沿原路径返回，重新选择另一规则，继续搜索。
- **图搜索策略（第5章）**：用**状态空间图**表示问题求解过程，搜索过程可看作从初始问题图中求出含有解路径的子图。

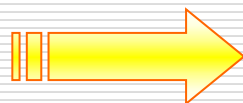


- 综合数据库（选择一种数据结构来表示将牌的布局）：
 - 通常可用来表示综合数据库的数据结构有字符串、向量、集合、数组、树、表格、文件等。
- 规则库：

用产生式规则表示八数码中空格的移动：

- r1: if 空格不在左侧的边上 than 向左移动空格
- r2: if 空格不在靠上的边上 than 向上移动空格
- r3: if 空格不在右侧的边上 than 向右移动空格
- r4: if 空格不在靠下的边上 than 向下移动空格

2	8	3
1	6	4
7		5

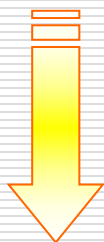


8 数码
(九宫图)

1	2	3
8		4
7	6	5

8 数码（九宫图）

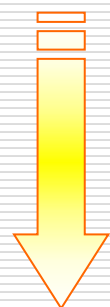
2	8	3
1	6	4
7		5



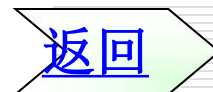
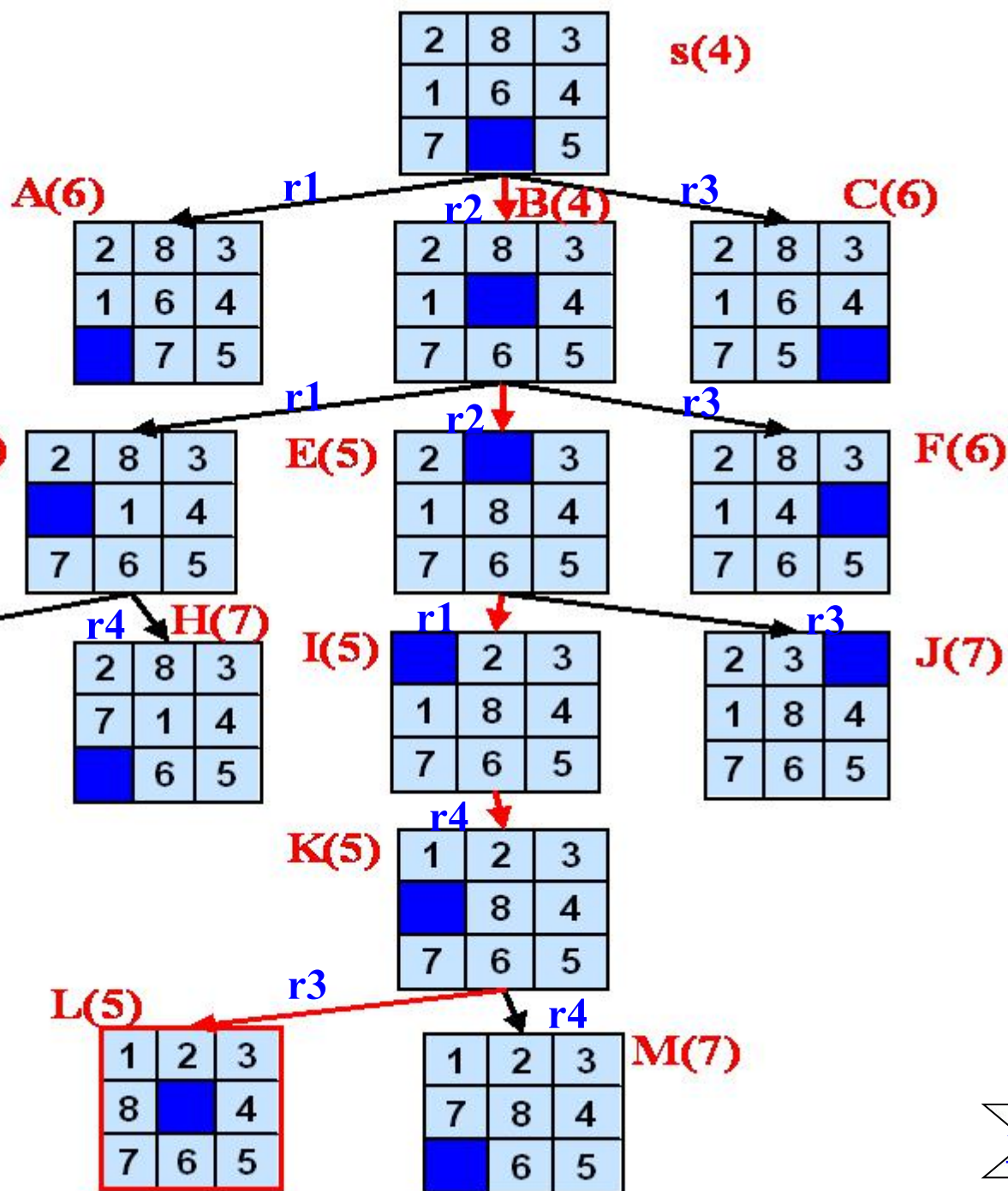
1	2	3
8		4
7	6	5

产生式系统	图搜索
初始事实数据	初始节点
目标条件	目标节点
产生式规则	状态转换规则 问题变换规则
规则库	操作集
动态数据库	节点(状态/问题) CLOSED表、OPEN表
控制策略	搜索策略

2	8	3
1	6	4
7		5



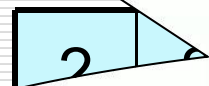
1	2	3
8		4
7	6	5



产生式系统的控制策略（补充）

■ 基本产生式系统的控制策略：

- **不可撤回策略**：搜索过程一直往前进行，不允许撤回已经选择的规则。——“**一直往前走**”不回头
- **回溯策略（“碰壁回头”）**：在从冲突集选择一条规则执行时，建立回溯点，若以后发现此规则会阻挠或延迟到达目标，则沿原路径返回，重新选择另一规则，继续搜索。
- **图搜索策略（第5章）**：用**状态空间图**表示问题求解过程，搜索过程可看作从初始问题图中求出含有解路径的子图。



回溯策略：抹去了所有引起失败的试探路径，
图搜索策略：记住了已试过的所有路径。



（九宫图）



2.3.4 产生式表示法的特点

1. 产生式表示法的优点

- (1) 自然性
- (2) 模块性
- (3) 有效性
- (4) 清晰性

2. 产生式表示法的缺点

- (1) 效率不高

快速模式匹配算法，如**RETE**，1979年由**Charles Forgy (CMU)**提出，**Rete** 匹配速度与规则数目无直接关系，其核心思想是**以空间换时间**。

工具名称	实现语言	支持编程语言	是否开源	算法
Drools	Java	Java	✓	RETEOO/PHREAK
Jena	Java	Java	✓	RETE
RDF4J	Java	Java/PHP/Python	✓	RETE

Drools是通用规则管理系统，而另外2个均是图数据库

2.3.4 产生式表示法的特点

1. 产生式表示法的优点

- (1) 自然性
- (2) 模块性
- (3) 有效性
- (4) 清晰性

2. 产生式表示法的缺点

- (1) 效率不高
快速模式匹配算法（如RETE）
- (2) 不易表达结构性知识

3. 适合产生式表示的知识

- 1) 领域知识间关系不密切，不存在结构关系。
- 2) 经验性及不确定性的知识，且对这些知识没有严格、统一的理论。
- 3) 问题求解过程可表示为一系列相对独立的操作，且每个操作可表示为一条或多条产生式。

第2章 知识表示

■ 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（推理的方向、冲突消解策略）

■ 2.4 框架表示法

- 1. 中国大学MOOC：第3讲单元测试
- 2. 网络教学平台：W2-1作业

■ 2.5 语义网络

框架表示法：以框架理论为基础发展起来的一种结构化的知识表示方法。

2.4 框架表示法

- 2.4.1 框架的一般结构
- 2.4.2 用框架表示知识的例子
- 2.4.3 框架表示法的特点

- 什么是框架？框架的一般结构？
- 如何用框架表示知识？框架的预定义槽（**default**、**if-needed**、**if-added**、**AKO**、**ISA**等）含义？
- 如何用框架系统求解问题？框架之间的横向联系、纵向联系？
- 框架表示法的优缺点是什么？
- 框架表示法适合描述什么样的知识？

2.4 框架表示法

- **框架理论**（1975年，美国明斯基）的基本思想：
 - 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**面临**一个新事物时，就从记忆中找出一个合适的框架，并根据实际情况对其细节加以修改、补充，从而形成对当前事物的认识。
- **框架（Frame）**：一种描述所论对象（事物、事件或概念）属性的数据结构。
- **事例框架/实例框架**：框架的一个具体事例/实例。
- 框架系统在很多自然语言处理任务，如Dialogue系统中都有**广泛的应用**，例如**阿里小蜜——任务型对话系统**。

2.4.1 框架的一般结构

<框架名>

槽名1: 侧面名₁₁ 值₁₁₁, ..., 值_{11P1}

| |

侧面名_{1m} 值_{1m1}, ..., 值_{1mPm}

槽名n: 侧面名_{n1} 值_{n11}, ..., 值_{n1P1}

|

侧面名_{nm} 值_{nm1}, ..., 值_{nmPm}

约束: 约束条件1

|

约束条件n

- 一个框架由若干个“槽”(Slot)结构组成, 每个槽又可分为若干个“侧面”。
- 一个槽: 用于描述所论对象某一方面的属性;
- 一个侧面: 用于描述相应属性的一个方面。
- 槽和侧面所具有的属性值分别称为槽值和侧面值。

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)

2.4.1 框架的一般结构

■ 槽(slot)值或侧面(faced)值:

- 数值型、字符串、布尔值
- 另一个框架
- 满足某个给定条件时执行的动作或过程

- “默认”值（**default**）：为相应槽提供缺省值。
- “如果需要”值（**if-needed**）：提供一个为相应槽赋值的过程。
- “如果加入”值（**if-added**）：提供一个因相应槽值变化而引起的后继处理过程。

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)

上述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。

2.4.3 用框架表示知识的例子

■ 框架名：〈 饭店 〉

地址：

内容：〈地址框架〉

if-needed : (查看选择单)

名字：

if-needed : (查看选择单)

食物类型：

范围：(中国菜、法国菜、韩国菜、日本菜)

默认(default)：中国菜

if-added : (更新可选饭店)

付费类型：

范围：(现金、信用卡、支付宝、微信)

可选饭店：

内容：相同食物类型的所有饭店

if-needed : (查找所有相同食物类型的饭店)

翻页

2.4.3 用框架表示知识的例子

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; HomeUnit(Number)
    MobileUnit(Number)
    If-Needed; Ask-Telephone
```

框架之间
横向联系

新的硕士框架

```
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
```

框架之间
纵向联系

- 学生框架描述所有学生的共性，新的硕士框架描述硕士生的个性，并继承学生框架的所有属性。
- 在新的硕士框架中使用了一个系统预定义槽名**AKO**(“是一种”)。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。

2.4.3 用框架表示知识的例子

新的硕士框架

```
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
```

实例框架

```
Frame <Master-1>
  ISA: <Master>
  Name: Yang Ye
  Sex: female
  Major: Computer
  Field: Web-Intelligence
  Advisor: Lin Hai
  Project: Provincial
```

- 当把一个学生的具体情况填入硕士框架之后，就可得到一个实例框架。
- 在这个实例框架中，用到了一个系统预定义槽名**ISA**(“是一个”)，表示这个实例框架是硕士框架的实例。

2.4.3 用框架表示知识的例子

□ 框架系统



- 框架之间的**纵向联系**是通过定义槽名**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。

2.4.3 用框架表示知识的例子

■ 例2.4，用框架表示下列一则地震消息：

- “某年某月某日，某地发生6.0级地震，若以膨胀注水孕震模式为标准，则三项地震前兆中的波速比为0.45，水氦含量为0.43，地形改变为0.60。”

框架名：〈地震〉

地 点：某地

日 期：某年某月某日

震 级：6.0

孕震模式：膨胀注水

波 速 比：0.45

水氦含量：0.43

地形改变：0.60

2008年5月12日14
时28分04秒，四川
汶川、北川发生8
级强震，……

翻页

2.4.3 用框架表示知识的例子

■ 例如，设有如下知识：

如果咳嗽、发烧且流涕，则八成是患了感冒，
需服用“感冒清”，
一日三次，每次2~3粒，
多喝开水。

框架名：〈**诊断规则**〉

症状1：咳嗽

症状2：发烧

症状3：流涕

可信度：0.8

框架名：〈**结论**〉

Possible-reason:〈**诊断规则**〉

病名：感冒

治疗方法：服用“感冒清”，
一日三次，每次2~3粒

注意事项：多喝开水

预后：良好

*框架系统求解问题的基本过程

■ 框架系统中问题的求解主要是通过**匹配**与**填槽**实现的。

■ 框架系统求解问题的基本过程

- 把**问题**用一个**框架**表示出来；

- 与知识库中已有的框架进行**匹配**，找出一个或几个**可匹配的预选框架**，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；

- 对预选框架进行评价，决定是否接受。

通过对相应槽的**槽名**和**槽值**逐个进行比较，并利用**继承关系**来实现

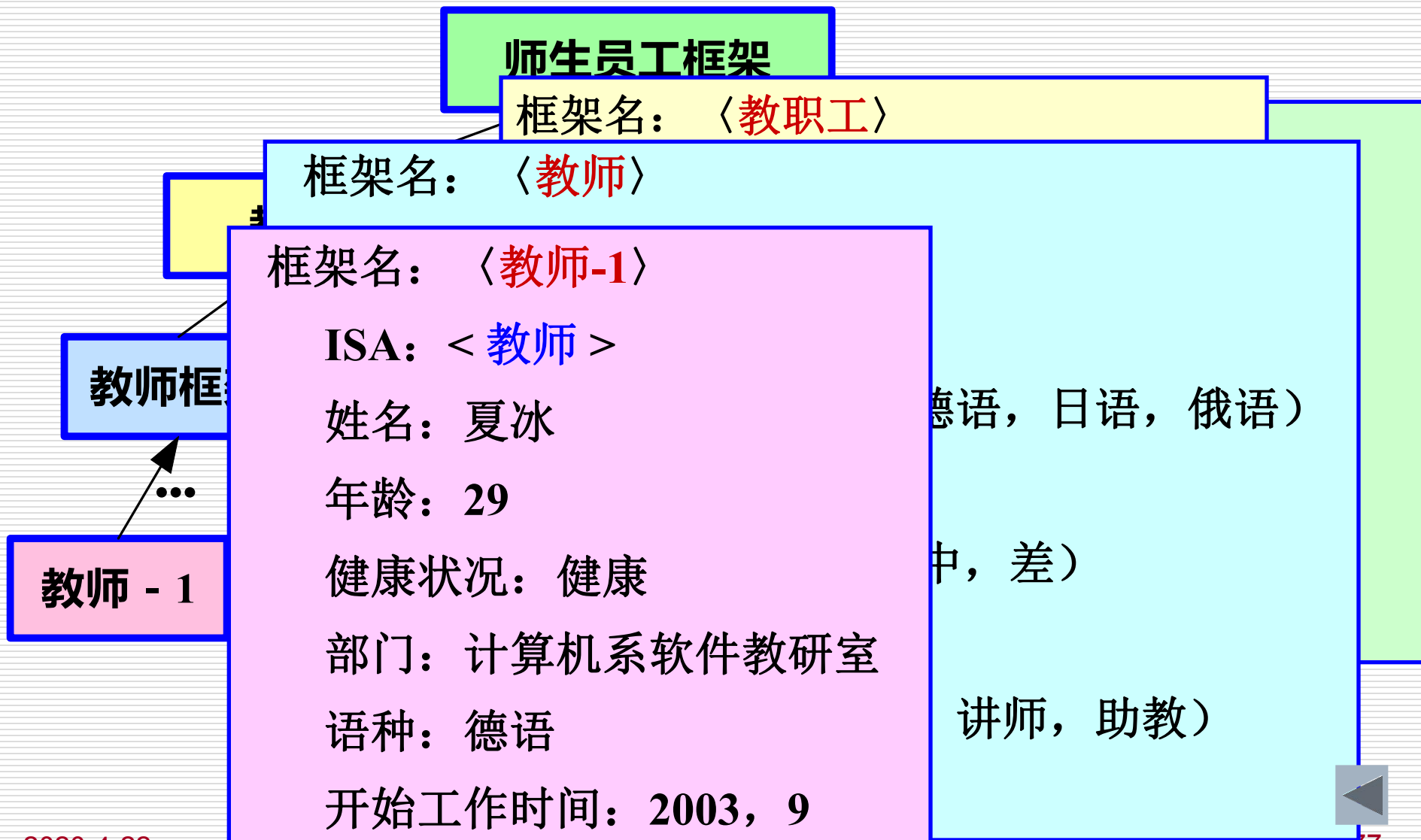
框架系统中求解问题的基本过程

- 例，关于师生员工的框架网络已建立在知识库中，现要从知识库中找出一个满足如下条件的教师：

男性，年龄在30岁以下，身体健康，职称为讲师

框架系统中求解问题的基本过程

■ 例，关于师生员工的框架网络



框架系统中求解问题的基本过程

- 例，关于师生员工的框架网络已建立在知识库中，现要从知识库中找出一个满足如下条件的教师：

男性，年龄在30岁以下，身体健康，职称为讲师

- 初始问题框架：

框架名： 〈 教师-x 〉

姓名：

年龄： < 30

性别： 男

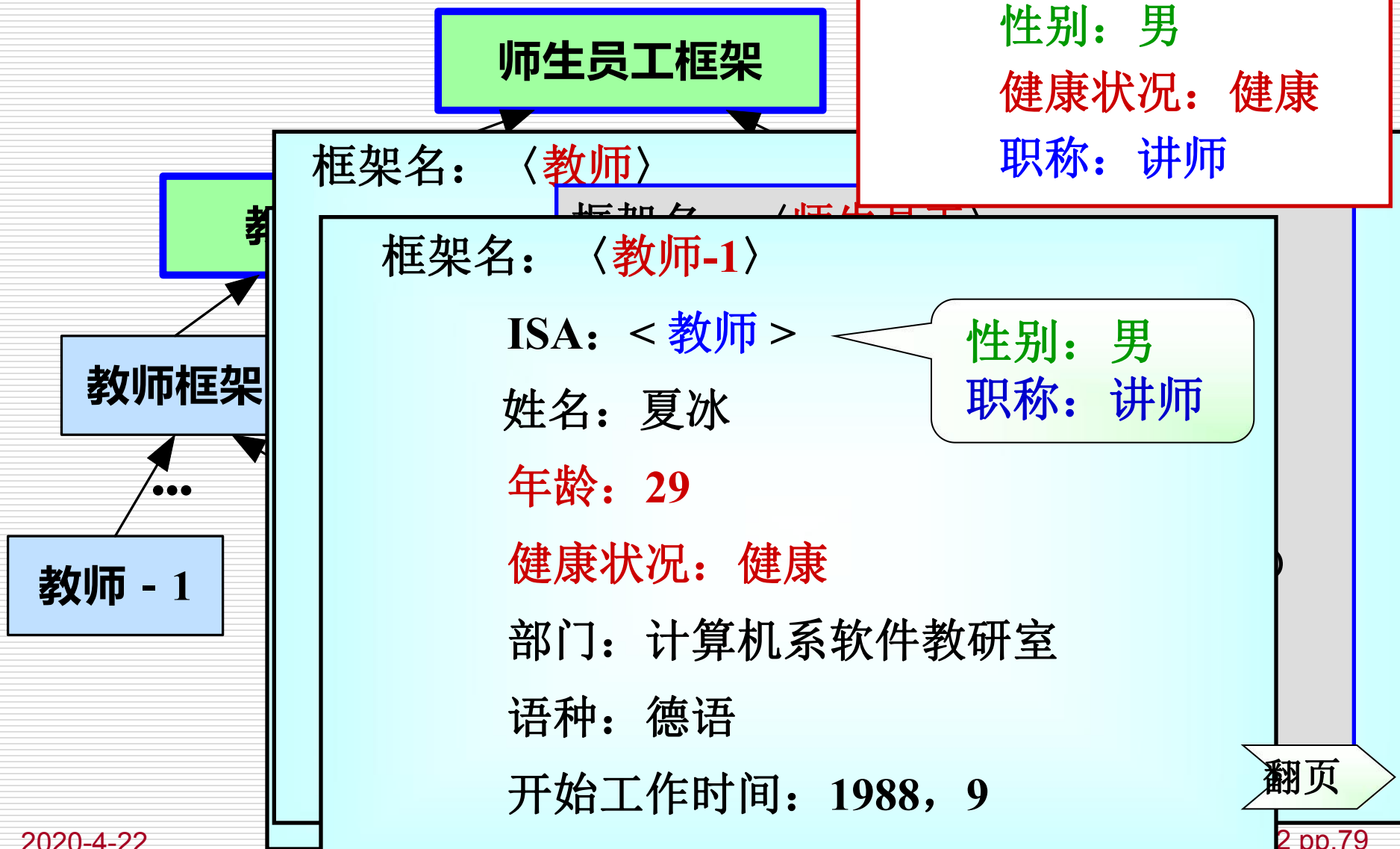
健康状况： 健康

职称： 讲师

翻页

框架系统中求解问题

■ 框架网络



2.4.5 框架表示法的特点

优点

1. 框架对于知识的描述**非常**完整和全**面**（结构性、继承性、自然性）
2. 基于框架的知识库质量**非常**高；
3. 且框架允许数值计算，这一点优于其它知识表示语言

缺点

1. 框架的构建成本**非常**高，对知识库的质量要求**非常**高；
2. 框架的表达形式不灵活，很难同其它形式的数据集相互关联使用。
3. 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很难表达。