

---

# 知识表示

---





20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023 September 2-8, 2023, Rhodes, Greece

[Download KR Whova App](#)

Welcome to the 20<sup>th</sup> International Conference on Principles of Knowledge Representation and Reasoning (KR2023), which will be held in Rhodes, Greece, in September 2-8, 2023.

**Knowledge Representation and Reasoning (KR)** is a well-established and lively field of research. In KR a fundamental assumption is that an agent's knowledge is explicitly represented in a declarative form, suitable for processing by dedicated reasoning engines. This assumption, that much of what an agent deals with is knowledge-based, is common in many modern intelligent systems. Consequently, KR has contributed to the theory and practice of various areas in AI, including automated planning and natural language understanding, and to fields beyond AI, including databases, verification, software engineering, and robotics. In recent years, KR has contributed also to new and emerging fields, including the semantic web, computational biology, cyber security, and the development of software agents.

The KR conference series is the leading forum for timely in-depth presentation of progress in the theory and principles underlying the representation and computational management of knowledge.

## Latest News

- The full Schedule is under [Schedule](#)
- The list of accepted papers is under [Accepted Papers](#)
- We are pleased to announce, as part of the Doctoral Consortium, an invited talk by Mary-Anne Williams, AAAI fellow and Michael J Crouch Chair for Innovation at the University of New South Wales.

## Important Dates

All dates are 'Anywhere on Earth', namely 23:59 UTC-12.

## Regular and Special Sessions

- **Submission of title and abstract:** March 3, 2023
- **Paper submission deadline:** March 14, 2023



## Principles of Knowledge Representation and Reasoning

[KR Inc Home](#)  
[Past Conferences](#)  
[Future Conferences](#)  
[Proceedings](#)  
[Prizes](#)  
[Publication Ethics](#)  
[Diversity & Inclusion](#)  
[KR Inc Organization](#)  
[Related Conferences](#)  
[KR Mailing Lists](#)  
[Resources](#)  
[Contact Information](#)

### Past KR Conferences

KR is the premier International Conference on Knowledge Representation and Reasoning.

The KR Conference Series provides more intimate setting than that of general AI conferences for researchers studying explicit representations of knowledge manipulated by inference algorithms, which provide an important foundation for much work in Artificial Intelligence, Computer Science, and Software Engineering. The conference emphasizes both the theoretical principles of knowledge representation and reasoning and the relationships between these principles and their embodiments in working systems.

The KR Conferences Series began in 1989:

- 2023, Rhodes, Greece
- 2022, Haifa, Israel
- 2021, Hanoi, Vietnam (virtual)
- 2020, Rhodes, Greece (virtual)
- 2018, Tempe, Arizona, USA
- 2016, Cape Town, South Africa
- 2014, Vienna, Austria
- 2012, Rome, Italy
- 2010, Toronto, Canada
- 2008, Sydney, Australia
- 2006, Lake District of the UK
- 2004, Whistler, Canada
- 2002, Toulouse, France
- 2000, Breckenridge, Colorado, USA
- 1998, Trento, Italy
- 1996, Cambridge, Massachusetts, USA
- 1994, Bonn, Germany
- 1992, Cambridge, Massachusetts, USA
- 1991, Cambridge, Massachusetts, USA
- 1989, Toronto, Canada

The proceedings for KR'89 - KR2002 are published by Morgan Kaufmann Publishers, San Francisco, California, USA, the Proceedings from KR2004 - KR2018 are published by AAAI, and the Proceedings of KR2020 - KR2021 are published by IJCAI.

---

# Outline

- ❖ **基本概念**
  - ❖ **一阶谓词表示法**
  - ❖ **产生式表示法**
  - ❖ **框架表示法**
  - ❖ **语义网络表示法 > 知识图谱**
  - ❖ **剧本（脚本）表示法**
  - ❖ **Petri网表示法**
  - ❖ **过程表示法**
-

---

# 知识表示的重要性

## ❖ 知识是智能的基础

- ◆ 获得知识
- ◆ 运用知识

## ❖ 符合计算机要求的知识模式

- ◆ 计算机能存储、处理的知识表示模式
  - ◆ 数据结构（List, Table, Tree, Graph, etc.）
-



---

# Importance of Good Notations

- ❖ “By relieving the brain of all unnecessary work, a good notation sets it free to concentrate on more advanced problems, and in effect increases the mental power of the race. Before the introduction of the Arabic notation, multiplication was difficult, and the division even of integers called into play the highest mathematical faculties...Civilization advances by extending the number of important operations which we can perform without thinking about them.”

Alfred North Whitehead, *Introduction to Mathematics*

---

---

# 一、基本概念

## 1. 数据、信息与知识

### ■ 数据 (Data)

- ◆ 信息的载体和表示
- ◆ 用一组符号及其组合表示信息

### ■ 信息 (Information)

- ◆ 数据的语义
- ◆ 数据在特定场合下的具体含义

### ■ 知识 (Knowledge)

- ◆ 信息关联后所形成的信息结构：事实 & 规则
  - ◆ 经加工、整理、解释、挑选、改造后的信息
-

---

## 2. 知识的特性

### ❖ 相对正确性

- ◆ 一定条件下
- ◆ 某种环境中
- ◆ .....

### ❖ 不确定性

- ◆ 存在“中间状态”
- ◆ “真”（“假”）程度
- ◆ 随机性
- ◆ 模糊性
- ◆ 经验性
- ◆ 不完全性
- ◆ .....

### ❖ 可表示性 & 可利用性

- ◆ 语言
  - ◆ 文字
  - ◆ 图形
  - ◆ 图像
  - ◆ 视频
  - ◆ 音频
  - ◆ 神经网络
  - ◆ 概率图模型
  - ◆ .....
-



---

### 3. 知识的分类

- ❖ 常识性知识、领域性知识

  - 作用范围

- ❖ 事实性知识、过程性知识、控制知识

  - 作用及表示

- ❖ 确定性知识、不确定性知识

  - 确定性

- ❖ 逻辑性知识、形象性知识

  - 结构及表现形式

- ❖ 零级知识、一级知识、二级知识

  - 抽象程度

---

---

## 4. 常用的知识表示方法

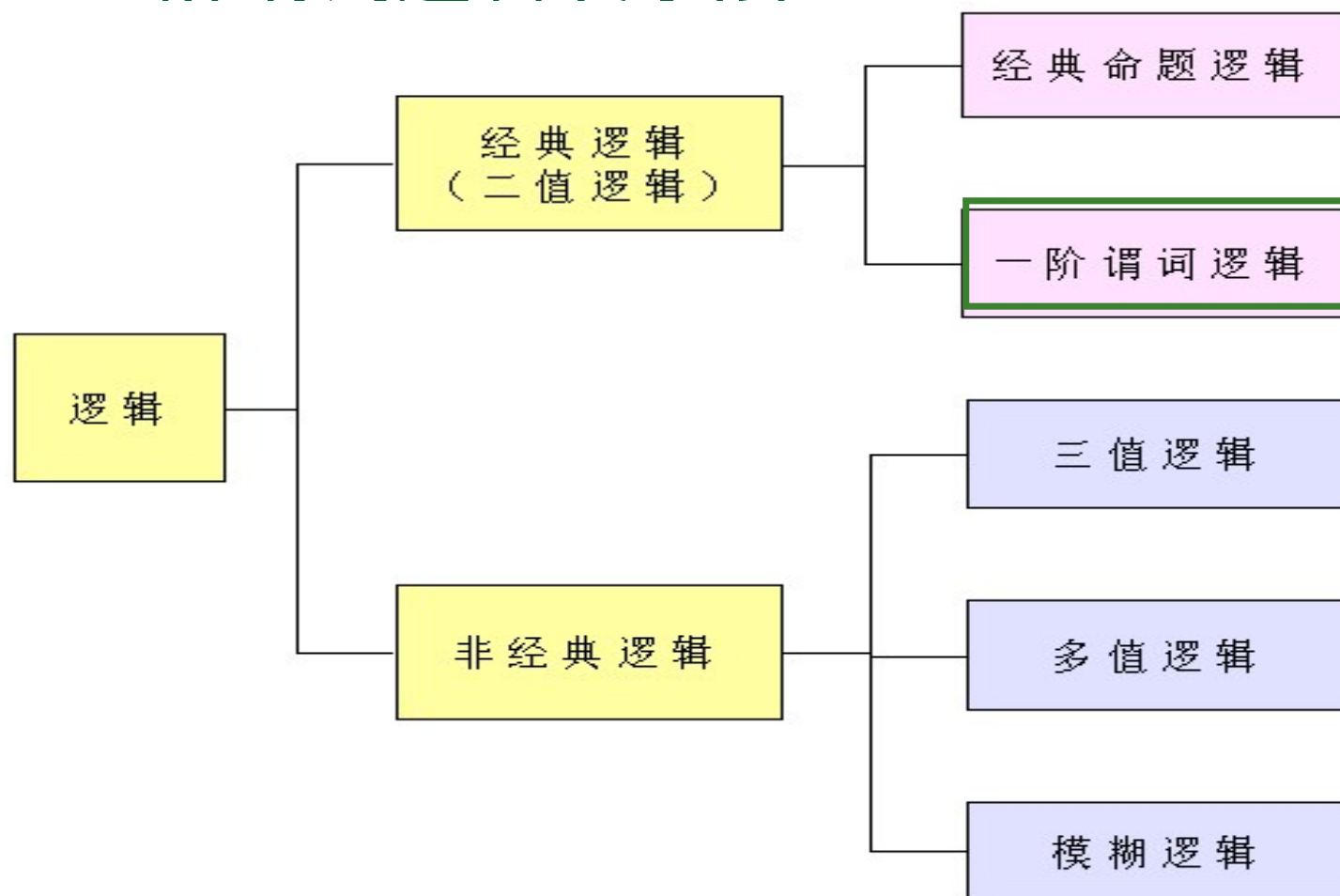
- ❖ **状态空间** (State Space)
  - ❖ **与或图** (And-Or Graph)
  - ❖ **一阶谓词** (First Order Predicate)
  - ❖ **产生式** (Production)
  - ❖ **框架** (Framework)
  - ❖ **语义网络** (Semantic Network)
  - ❖ **剧本** (Script)
  - ❖ **Petri网** (Petri Network)
  - ❖ **过程** (Procedure)
  - ❖ **面向对象** (Object-Oriented)
  - ❖ **信念网** (Belief Network)
  - ❖ **本体论** (Ontology)
  - ❖ .....
-

---

## 5. 如何选择合适的表示方法？

- ❖ 充分表示**领域知识**
  - ❖ 有利于**对知识的利用**
  - ❖ 便于**理解和实现**
  - ❖ 便于**对知识的组织、管理与维护**
-

## 二、一阶谓词逻辑表示法



---

# 谓词逻辑

- ❖ 命题逻辑与谓词逻辑是最先应用于人工智能的两种逻辑。
  - ❖ 对于知识的形式化表示，特别是定理的证明发挥了重要作用。
  - ❖ 谓词逻辑是在命题逻辑基础上发展起来的，命题逻辑可看作是谓词逻辑的一种特殊形式。
-

# 1. 命题

- ❖ 命题（**proposition**）：一个非真即假的陈述句。
  - 若命题的意义为真，称它的真值为真，记为  $T$ 。
  - 若命题的意义为假，称它的真值为假，记为  $F$ 。
  - 一个命题可在一种条件下为真，在另一种条件下为假。

# 命题逻辑的局限

❖ 令  $P$ ：李白是诗人。

$Q$ ：杜甫是诗人。

❖ 从符号  $P$ 、 $Q$  中不能归纳出李白和杜甫都是诗人的共性。

■ 命题逻辑：局限性大：无法把所描述的客观事物的结构及逻辑特征反映出来，也不能把不同事物间的共同特征表述出来。

➤ 命题逻辑不适合表示复杂的问题。



## 2. 谓词

❖ 谓词的一般形式:  $P(x_1, x_2, \dots, x_n)$

- 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; 个体可以是常量、变量、函数、谓词等
  - **Teacher (Zhang)**
  - **Less(x, 5)**
  - **Teacher (father (Li) )**
  - **Works (engineer (Smith), IBM)**
- 谓词名  $P$ : 刻画个体的性质、状态或个体间的关系

---

## 3. 谓词公式

### (1) 连接词（连词）

- 1)  $\neg$ : “否定”（**negation**）或“非”
  - 2)  $\vee$ : “析取”（**disjunction**）——或
  - 3)  $\wedge$ : “合取”（**conjunction**）——与
  - 4)  $\rightarrow$ : “蕴含”（**implication**）或“条件”（**condition**）
  - 5)  $\leftrightarrow$ : “等价”（**equivalence**）或“双条件”（**bicondition**）
-

谓词逻辑真值表

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

## (2) 量词 (quantifier)

- 1) 全称量词 (**universal quantifier**) ( $\forall x$ ) : “对个体域中的所有 (或任一个) 个体  $x$ ”。

$$(\forall x)[ROBOT(x) \rightarrow COLOR(x, GRAY)]$$

- 2) 存在量词 (**existential quantifier**) ( $\exists x$ ) : “在个体域中存在个体  $x$ ”。

$$(\exists x) INROOM(x, r1)$$

# 谓词公式

❖ 定义1 可按下述规则得到谓词演算的谓词公式:

- (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。

连接词的优先级别从高到低排列:

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

### (3) 量词的辖域

- ❖ 量词的辖域：位于量词后面的单个谓词或者用括弧括起来的谓词公式。
- ❖ 约束变元与自由变元：辖域内与量词中同名的变元称为约束变元，不同名的变元称自由变元。

■ 例：  $(\exists x)(P(x, y) \rightarrow Q(x, y)) \vee R(x, y)$

## 4. 谓词公式的性质

### (1) 谓词公式的解释

❖ 谓词公式在个体域上的解释：个体域中的实体对谓词演算表达式的每个常量、变量、谓词和函数符号的指派。

*Friends (george, x)*

*Friends (george, susie)      T*

*Friends (george, kate)      F*

■ 对于每一个解释，谓词公式都可求出一个真值（*T*或*F*）。



## (2) 谓词公式的永真性、可满足性、不可满足性

■ 定义2 如果谓词公式 $P$ 对个体域 $D$ 上的任何一个解释都取得真值 $T$ ，则称 $P$ 在 $D$ 上是永真的；如果 $P$ 在每个非空个体域上均永真，则称 $P$ 永真。

■ 定义3 如果谓词公式 $P$ 对个体域 $D$ 上的任何一个解释都取得真值 $F$ ，则称 $P$ 在 $D$ 上是永假的；如果 $P$ 在每个非空个体域上均永假，则称 $P$ 永假。

■ 定义4 对于谓词公式 $P$ ，如果至少存在一个解释使得 $P$ 在此解释下的真值为 $T$ ，则称 $P$ 是可满足的，否则，称 $P$ 是不可满足的。

### (3) 谓词公式的等价性

■ 定义5 设 $P$ 与 $Q$ 是两个谓词公式， $D$ 是它们共同的个体域，若对 $D$ 上的任何一个解释， $P$ 与 $Q$ 都有相同的真值，则称公式 $P$ 和 $Q$ 在 $D$ 上是等价的。如果 $D$ 是任意个体域，则称 $P$ 和 $Q$ 是等价的，记为 $P \Leftrightarrow Q$ 。

#### (4) 谓词公式的永真蕴含

■ **定义6** 对于谓词公式 $P$ 与 $Q$ ，如果 $P \rightarrow Q$ 永真，则称公式 $P$ 永真蕴含 $Q$ ，且称 $Q$ 为 $P$ 的逻辑结论，称 $P$ 为 $Q$ 的前提，记为 $P \Rightarrow Q$ 。

■ **假言推理**  $P, P \rightarrow Q \Rightarrow Q$

■ **拒取式推理**  $P \rightarrow Q, \sim Q \Rightarrow \sim P$

## ❖ 谓词逻辑的其他推理规则

- ① ***P***规则：在推理的任何步骤上都可引入前提。
- ② ***T***规则：在推理过程中，如果前面步骤中有一个或多个公式永真蕴含公式***S***，则可把***S***引入推理过程中。
- ③ ***CP***规则：如果能从任意引入的命题***R***和前提集合中推出***S***来，则可从前提集合推出 **$R \rightarrow S$** 来。

❖ 谓词逻辑的其他推理规则：

④ **反证法**： $P \Rightarrow Q$ ，当且仅当  $P \wedge \neg Q \Leftrightarrow F$ ，即  $Q$  为  $P$  的逻辑结论，当且仅当  $P \wedge \neg Q$  是不可满足的。

■ **定理**： $Q$  为  $P_1, P_2, \dots, P_n$  的逻辑结论，当且仅当  $(P_1 \wedge P_2 \wedge \dots \wedge P_n) \wedge \neg Q$  是不可满足的。

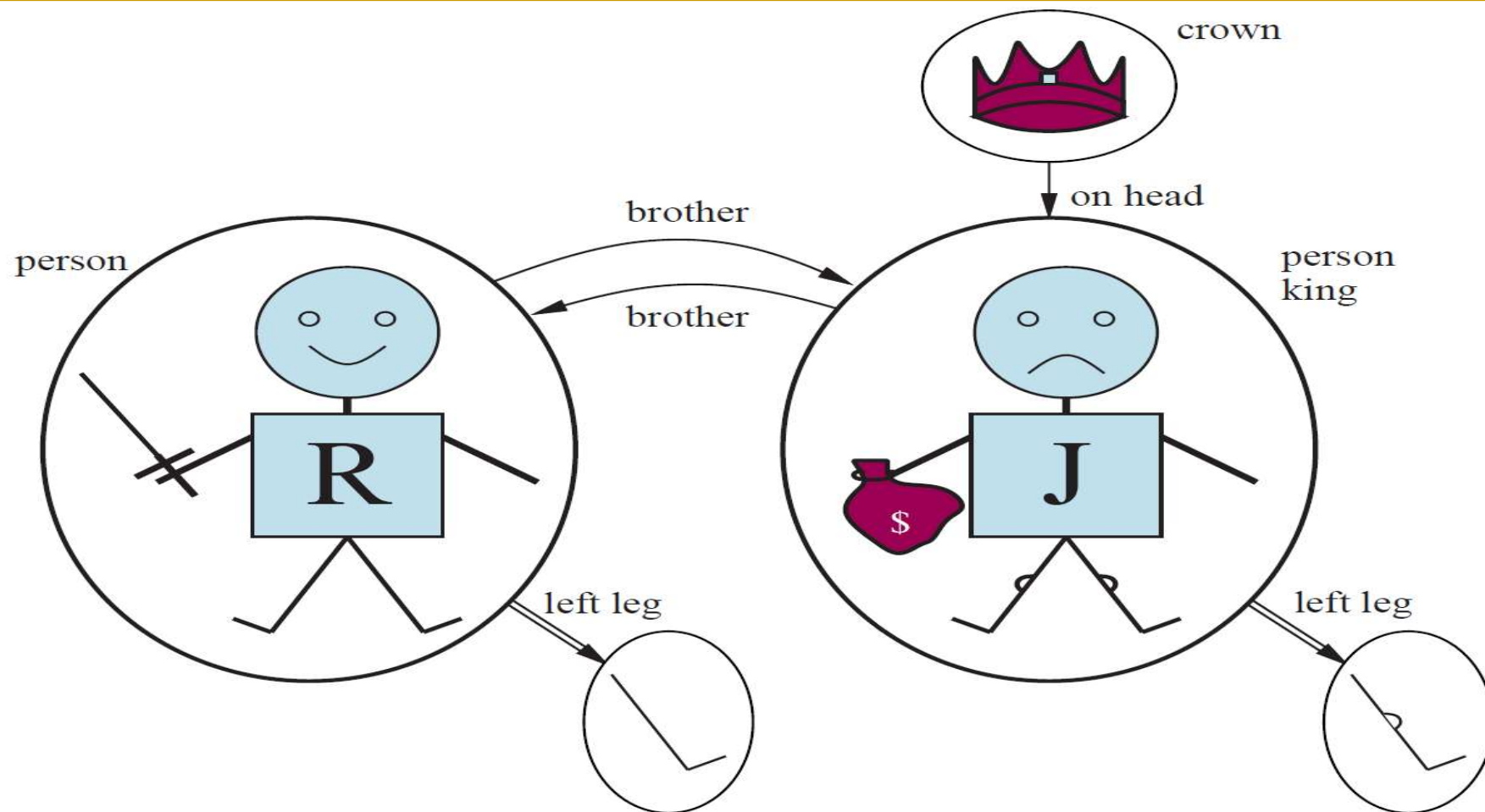
## 5. 一阶谓词逻辑知识表示方法

❖ 谓词公式表示知识的步骤：

(1) 定义谓词及个体。

(2) 变元赋值。

(3) 用连接词连接各个谓词，形成谓词公式。



- ❖ A model containing five objects, two binary relations (brother and on-head), three unary relations (person, king, and crown), and one unary function (left-leg).



---

# 使用一阶谓词逻辑——断言与查询

- ❖ 通过**TELL**将语句添加到知识库，这种语句称为断言（**assertion**）  
例如：
    - ◆ **TELL**(KB, King(John))
    - ◆ **TELL**(KB, Person(Richard))
    - ◆ **TELL**(KB,  $\forall x \text{ King}(x) \Rightarrow \text{Person}(x)$ )
  - ❖ 使用**ASK**对知识库进行提问。使用**ASK**提出的问题称为查询或目标。  
例如：
    - ◆ **ASK**(KB, King(John)), 返回True。
    - ◆ 量化问题的查询: **ASKVARS**(KB, Person(x)), 返回答案: {x/John} and {x/Richard}
-

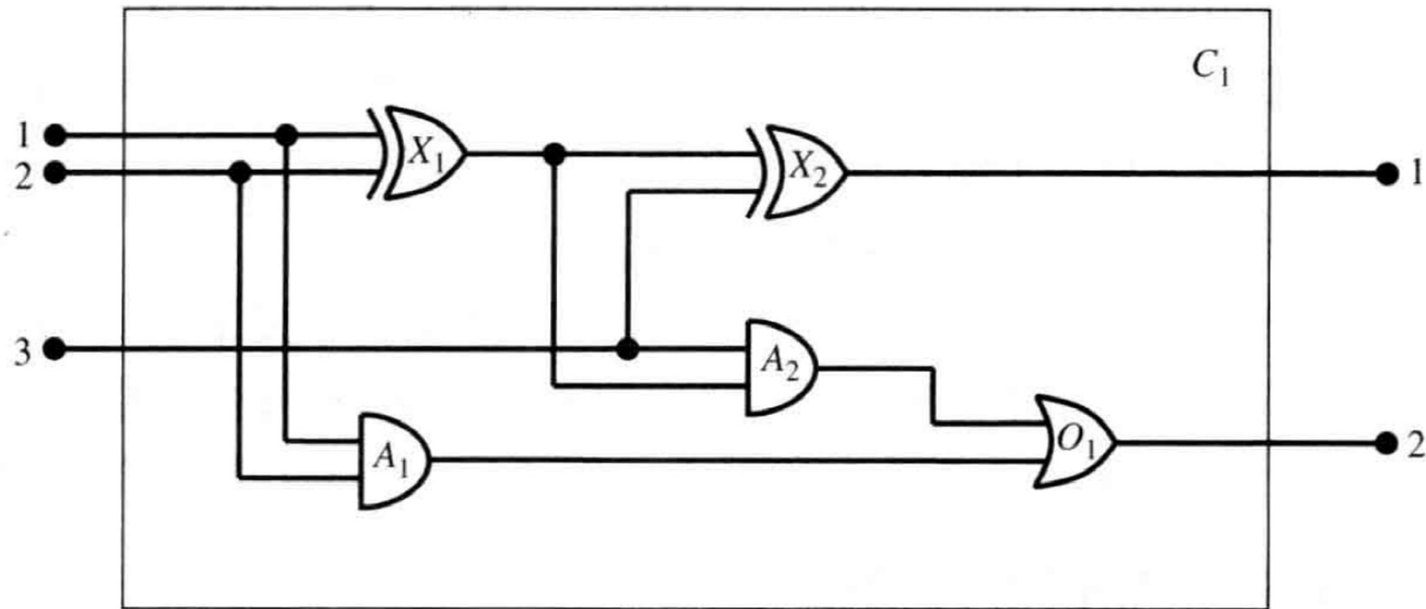
---

# 一阶谓词逻辑中的知识工程

- ❖ 知识工程——知识库构造的一般过程。
  - ❖ 方法：知识工程师对特定领域进行调研，总结出在该领域的重要概念，构建该领域的对象和关系的形式化表示。
    - ◆ 确定任务：刻画知识库支持哪些问题查询，以及对于每个特定的问题可以采用哪些种类的事实。
    - ◆ 搜集相关知识：由任务确定知识库范围，并了解该领域的工作模式
    - ◆ 确定词汇表：包括谓词、函数和常量。把重要的领域概念转换为逻辑名称
    - ◆ 对领域通用知识编码：应方便专家对内容进行检查
    - ◆ 对特定问题实例描述编码
    - ◆ 把查询提交给推理过程并获取答案
    - ◆ 知识库调试
-

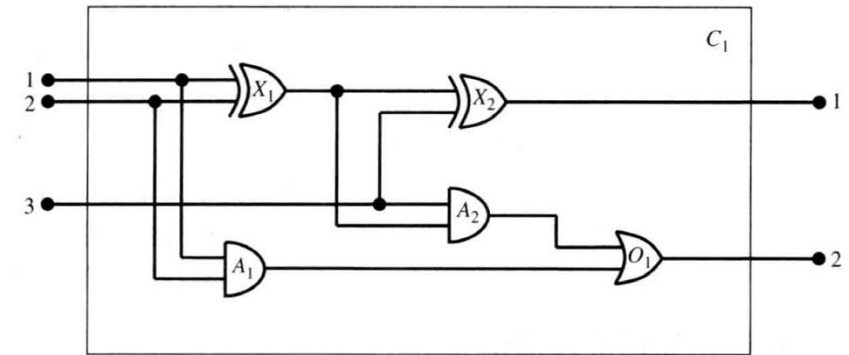
# 数字电路的本体论和知识库构建

- ❖ 数字电路 **C**，实现一位全加器。最初的两个输入是加法的两个操作数，第三个输入是进位。第一个输出是本位和，第二个输出是向上一位的进位。电路包括两个异或门，两个与门和一个或门。



# 知识库构建过程

- ❖ 1) 确定任务：分析电路图及其功能？调试电路？能否正确实现、性能如何等等？
- ❖ 2) 收集相关知识：门、导线、延时、面积、成本.....
- ❖ 3) 确定词汇表：选择函数、谓词和常量来表示电路、端、信号和门
  - ◆ 谓词Circuit(C)、Terminal(x)、Gate(X)、函数Type(X)= XOR等
  - ◆ Connected(Out(1,X),In(1,X))
  - ◆ In(1,X1): the first input terminal for circuit
  - ◆ Out(n,c): 输出端
  - ◆ On(t): 端口t的信号值为On
  - ◆ Signal(t): t的信号值



#### 4) 对领域通用知识编码:

(1) 如果两个端是连通的, 那么它们信号相同:

$$\forall t_1, t_2 \text{ Terminal}(t_1) \wedge \text{Terminal}(t_2) \wedge \text{Connected}(t_1, t_2) \Rightarrow \text{Signal}(t_1) = \text{Signal}(t_2)$$

(2) 每个端的信号不是 1 就是 0:

$$\forall t \text{ Terminal}(t) \Rightarrow \text{Signal}(t) = 1 \vee \text{Signal}(t) = 0$$

(3) Connected 是对称的:

$$\forall t_1, t_2 \text{ Connected}(t_1, t_2) \Leftrightarrow \text{Connected}(t_2, t_1)$$

(4) 存在有四种类型的门:

$$\forall g \text{ Gate}(g) \wedge k = \text{Type}(g) \Rightarrow k = \text{AND} \vee k = \text{OR} \vee k = \text{XOR} \vee k = \text{NOT}$$

(5) 与门的输出为 0, 当且仅当它的任一输入为 0:

$$\forall g \text{ Gate}(g) \wedge \text{Type}(g) = \text{AND} \Rightarrow \text{Signal}(\text{Out}(1, g)) = 0 \Leftrightarrow \exists n \text{ Signal}(\text{In}(n, g)) = 0$$

(6) 或门的输出为 1, 当且仅当它的任一输入为 1:

$$\forall g \text{ Gate}(g) \wedge \text{Type}(g) = \text{OR} \Rightarrow \text{Signal}(\text{Out}(1, g)) = 1 \Leftrightarrow \exists n \text{ Signal}(\text{In}(n, g)) = 1$$

(7) 异或门的输出为 1, 当且仅当它的输入不相等:

$$\forall g \text{ Gate}(g) \wedge \text{Type}(g) = \text{XOR} \Rightarrow \text{Signal}(\text{Out}(1, g)) = 1 \Leftrightarrow \text{Signal}(\text{In}(1, g)) \neq \text{Signal}(\text{In}(2, g))$$

(8) 非门的输出与它的输入相反:

$$\forall g \text{ Gate}(g) \wedge (\text{Type}(g) = \text{NOT}) \Rightarrow \text{Signal}(\text{Out}(1, g)) \neq \text{Signal}(\text{In}(1, g))$$

(9) 门（除了非门）有两个输入和一个输出：

$$\forall g \text{ Gate}(g) \wedge \text{Type}(g) = \text{NOT} \Rightarrow \text{Arity}(g, 1, 1)$$

$$\forall g \text{ Gate}(g) \wedge k = \text{Type}(g) \wedge (k = \text{AND} \vee k = \text{OR} \vee k = \text{XOR}) \Rightarrow \text{Arity}(g, 2, 1)$$

(10) 门电路有多个端，输入端和输出端都不能超出它的维数：

$$\begin{aligned} \forall c, i, j \text{ Circuit}(c) \wedge \text{Arity}(c, i, j) \Rightarrow & \forall n (n \leq i \Rightarrow \text{Terminal}(\text{In}(c, n))) \wedge (n > i \Rightarrow \text{In}(c, n) = \text{Nothing}) \wedge \\ & \forall n (n \leq j \Rightarrow \text{Terminal}(\text{Out}(c, n))) \wedge (n > j \Rightarrow \text{Out}(c, n) = \text{Nothing}) \end{aligned}$$

(11) 门、端、信号、门的类型和空是互不相同的。

$$\forall g, t \text{ Gate}(g) \wedge \text{Terminal}(t) \Rightarrow g \neq t \neq 1 \neq 0 \neq \text{OR} \neq \text{AND} \neq \text{XOR} \neq \text{NOT} \neq \text{Nothing}$$

(12) 门是电路。

$$\forall g \text{ Gate}(g) \Rightarrow \text{Circuit}(g)$$

## 5) 对特定问题实例描述编码

(1) 对电路和组成它的门加以分类  $Circuit(C_1) \wedge Arity(C_1, 3, 2)$

$Gate(X_1) \wedge Type(X_1)=XOR$

$Gate(X_2) \wedge Type(X_2)=XOR$

$Gate(A_1) \wedge Type(A_1)=AND$

$Gate(A_2) \wedge Type(A_2)=AND$

$Gate(O_1) \wedge Type(O_1)=OR$

(2) 说明它们之间的连接

$Connected(Out(1, X_1), In(1, X_2))$

$Connected(Out(1, X_1), In(2, A_2))$

$Connected(Out(1, A_2), In(1, O_1))$

$Connected(Out(1, A_1), In(2, O_1))$

$Connected(Out(1, X_2), Out(1, C_1))$

$Connected(Out(1, O_1), Out(2, C_1))$

$Connected(In(1, C_1), In(1, X_1))$

$Connected(In(1, C_1), In(1, A_1))$

$Connected(In(2, C_1), In(2, X_1))$

$Connected(In(2, C_1), In(2, A_1))$

$Connected(In(3, C_1), In(2, X_2))$

$Connected(In(3, C_1), In(1, A_2))$



## 6) 向推理过程提交查询

哪种输入组合可以使得 C 的第一个输出(和)为0而第二个输出(进位)为1?

$$\begin{aligned} \exists i_1, i_2, i_3 \quad & \text{Signal(In(1, } C_1)) = i_1 \wedge \text{Signal(In(2, } C_1)) = i_2 \wedge \text{Signal(In(3, } C_1)) = i_3 \\ & \wedge \text{Signal(Out(1, } C_1)) = 0 \wedge \text{Signal(Out(2, } C_1)) = 1 \end{aligned}$$

回答是变量*i*<sub>1</sub>、*i*<sub>2</sub>和*i*<sub>3</sub>的置换，其结果语句为知识库所蕴涵，这样的置换有三个：

$$\{i_1/1, i_2/1, i_3/0\} \quad \{i_1/1, i_2/0, i_3/1\} \quad \{i_1/0, i_2/1, i_3/1\}$$

加法器电路所有端的可能值的集合是什么?

$$\begin{aligned} \exists i_1, i_2, i_3, o_1, o_2 \quad & \text{Signal(In(1, } C_1)) = i_1 \wedge \text{Signal(In(2, } C_1)) = i_2 \wedge \text{Signal(In(3, } C_1)) = i_3 \\ & \wedge \text{Signal(Out(1, } C_1)) = o_1 \wedge \text{Signal(Out(2, } C_1)) = o_2 \end{aligned}$$

查询将返回一个完整的输入输出真值表，可以用于检验该加法器是否正确地对其输入进行了加法运算。

---

## 7) 调试知识库

可以用很多方法来干扰知识库以便了解知识库可能会出现什么错误行为。  
可以通过检查每个门的输出来查明问题。

---

## 6. 一阶谓词逻辑表示法的特点

### ❖ 优点:

- ◆ 自然性
- ◆ 精确性
- ◆ 严密性
- ◆ 容易实现

### ❖ 局限性:

- ◆ 不能表示不确定的知识
- ◆ 组合爆炸
- ◆ 效率低

❖ 应用：

- ◆ 自动问答系统（**Green**等人研制的**QA3**系统）
- ◆ 机器人行动规划系统（**Fikes**等人研制的**STRIPS**系统）
- ◆ 机器博弈系统（**Filman**等人研制的**FOL**系统）
- ◆ 问题求解系统（**Kowalski**等设计的**PS**系统）

。 。 。

---

## 三、产生式表示法

- ❖ 1943年E. Post第一次提出
  - ◆ 称为“Post机”的计算模型（《计算理论》）
  - ◆ 一种描述形式语言的语法

# 1. 产生式的基本形式

❖ 产生式的基本形式:

$$P \rightarrow Q$$

或     If P then Q

- ◆ **P: 前提**, 指出该产生式是否可用的条件。又称为前件、前提条件、条件、左部等。
- ◆ **Q: 结论或操作**, 指出当前题P所指示的条件被满足时, 应该得出的结论和应该执行的操作。又称为后件、右部等。

---

## ❖ 产生式与蕴涵式的区别

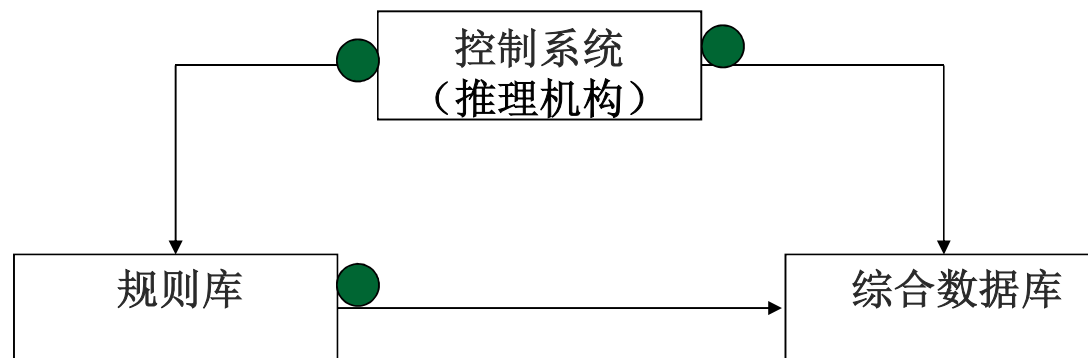
- ◆ 知识的精确性
- ◆ 匹配的精确性

## ❖ 产生式与条件语句的区别

- ◆ 前件结构
  - ◆ 控制流程
-

## 2. 产生式系统的组成

❖ 产生式系统一般由三个基本部分组成：





---

## (1) 规则库

用于描述相应领域内知识的产生式集合。

建立规则库时应注意的问题：

- 1) 有效地表达领域内的过程性知识
  - 2) 对知识进行合理的组织与管理：提高问题求解效率。
-

## (2) 综合数据库

又称事实库、上下文、黑板等。

- ❖ 一个用于存放问题求解过程中各种当前信息（初始状态、原始证据、中间结论、最终结论等）的数据结构。
- ❖ 综合数据库的内容是在不断变化的，是动态的。
- ❖ 综合数据库中的已知事实通常用字符串、向量、集合、矩阵、表等数据结构表示。

例：在疾病诊断专家系统(MYCIN)中，事实用四元组表示：

（特性 对象 值 可信度因子）

如：（age zhangshan 25 0.8）

### (3) 控制系统

又称**推理机构**，负责整个产生式系统的运行，实现对问题的求解。

(1) 从规则库中**选择规则**，并与综合数据库中的已知事实**进行匹配**。

(2) 发生**冲突**时调用相应策略予以消解

(3) 执行某条规则时，若该规则的右部是一个或多个结论，则**把这些结论加到综合数据库中**；若规则的右部是一个或多个操作，则**执行这些操作**。

(4) 对于**不确定性知识**，按一定的算法计算结论的不确定性

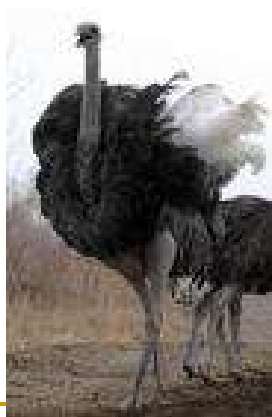
(5) 在适当的时候停止系统的运行

## (4) 产生式系统求解问题的一般步骤

- (1) 初始化综合数据库；
- (2) 检查规则库中是否有未使用过的规则，若无转 (7)；
- (3) 检查未使用规则中是否有其前提可与综合数据库中已知事实相匹配的规则，若有，形成当前可用规则集；否则转(6)；
- (4) 按冲突消解策略选择一个规则执行，并对该规则作上标记。把执行该规则后所得到的结论作为新的事实放入综合数据库。如果该规则的结论是一些操作，则执行这些操作；
- (5) 检查综合数据库中是否包含了该问题的解，若已包含，则结束问题求解过程；否则，转(2)；
- (6) 当规则库中还有未使用规则，但均不能与综合数据库中的已有事实相匹配时，要求用户进一步提供关于该问题的已知事实，若能提供，则转(2)；否则，执行下一步；
- (7) 若知识库中不再有未使用规则，则问题无解，终止求解。

### 3. 产生式系统的例子——动物识别系统

- 例：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:

- $r_1$ : 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

- $r_9$ : 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

❖ 设已知初始事实存放在综合数据库中：

该动物身上有：暗斑点，长脖子，长腿，奶，蹄

❖ 推理机构的工作过程：

(1) 从规则库中取出 $r_1$ ，检查其前提是否可与综合数据库中的已知事实匹配。匹配失败则 $r_1$ 不能被用于推理。然后取 $r_2$ 进行同样的工作。匹配成功则 $r_2$ 被执行。

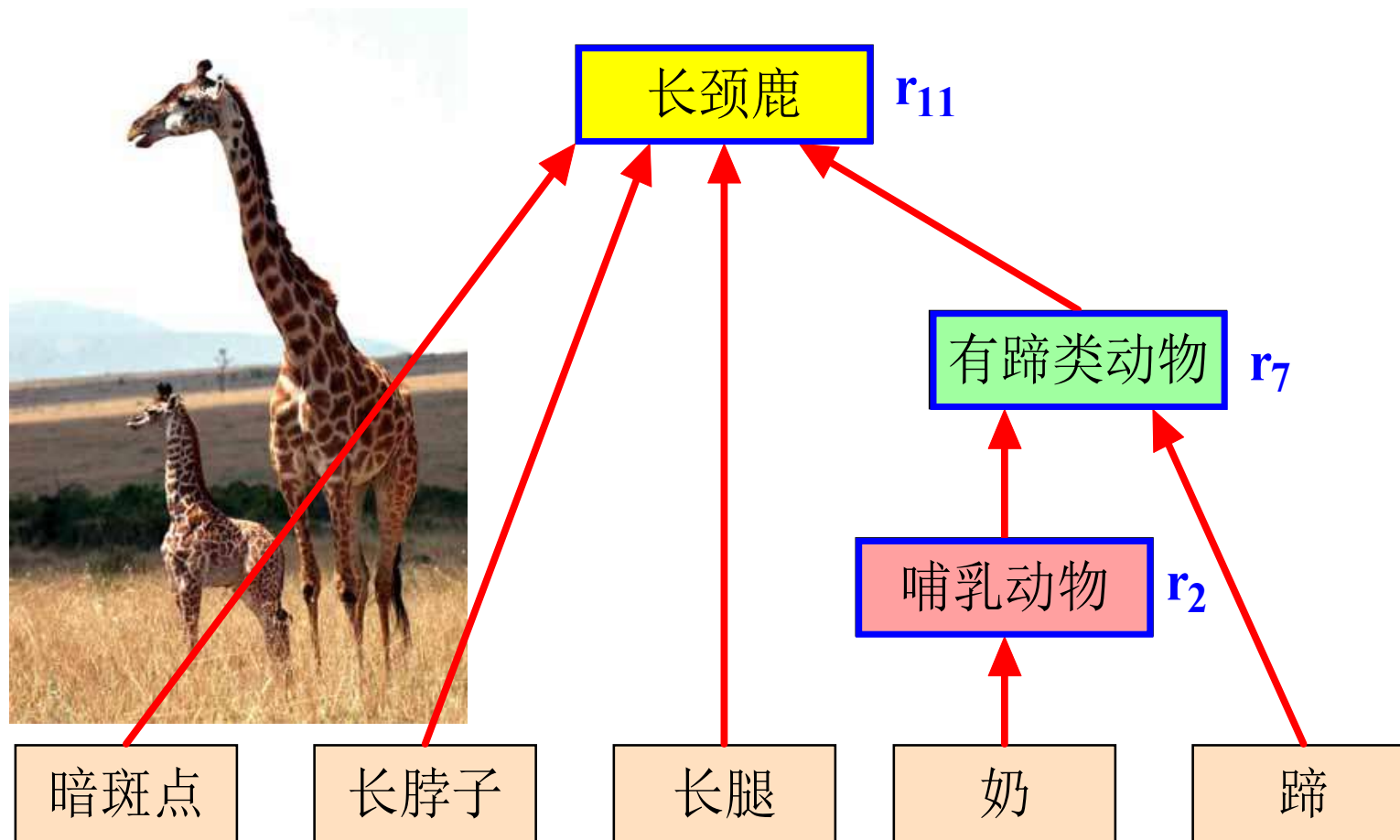
□ 综合数据库：该动物身上有：暗斑点，长脖子，长腿，奶，蹄，哺乳动物

(2) 分别用 $r_3$ ， $r_4$ ， $r_5$ ， $r_6$ 综合数据库中的已知事实进行匹配，均不成功。 $r_7$ 匹配成功，执行 $r_7$ 。

□ 综合数据库：该动物身上有：暗斑点，长脖子，长腿，奶，蹄，哺乳动物，有蹄类动物

(3)  $r_{11}$ 匹配成功，推出“该动物是长颈鹿”。





---

## 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
  - 2) 不能表达结构性知识
-

### (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，且相关领域中对这些知识没有严格、统一的理论。
- 3) 领域问题的求解过程可被表示为一系列相对独立的操作，且每个操作可被表示为一条或多条产生式规则。

### (4) 产生式表示法的应用：AI中应用最多的知识方法之一

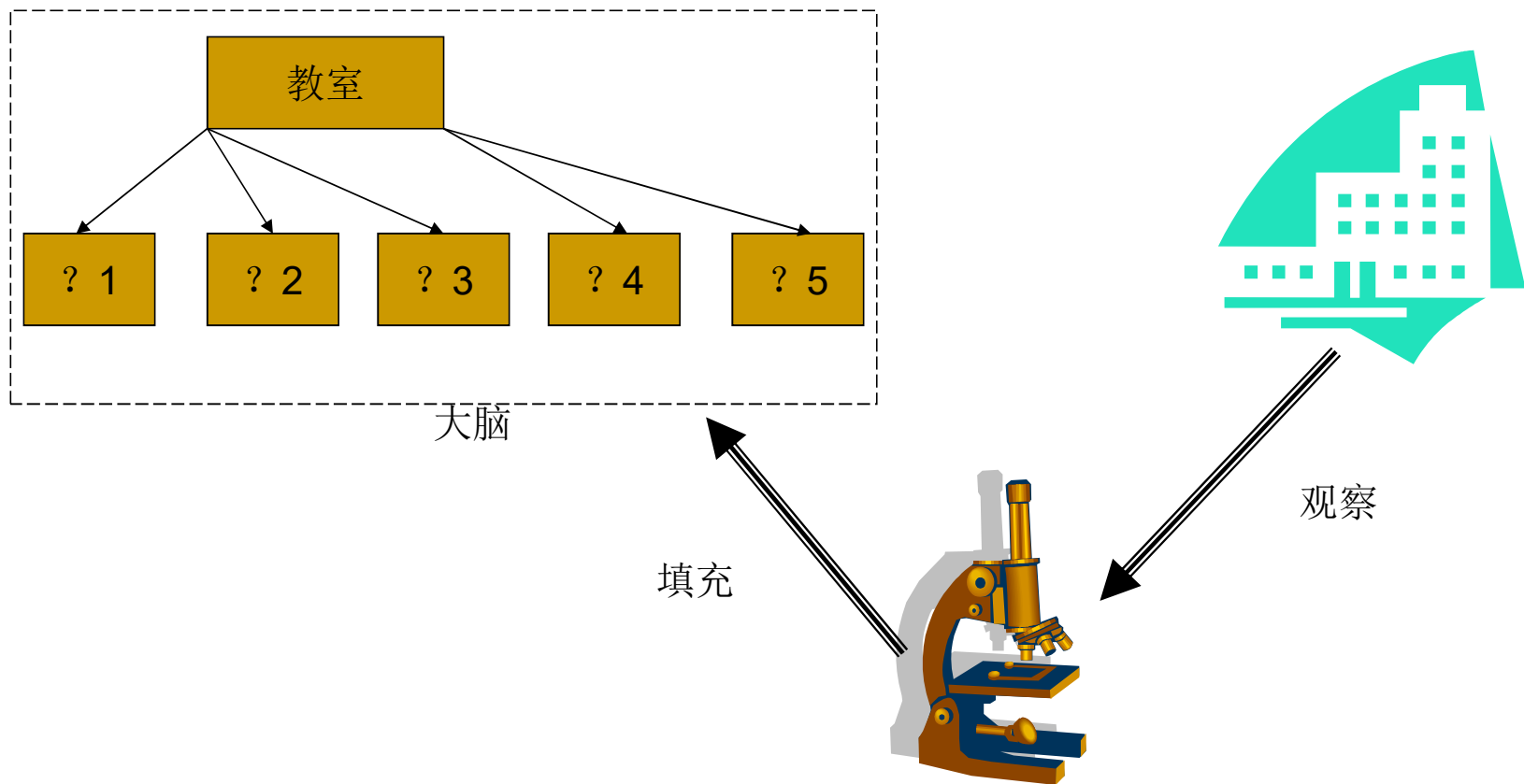
- ◆ 1972年A.Newell和H.A.Simon在研究人类的认识模型中提出了 **Rule-Based** 产生式方法及规则表示模式
- ◆ Feigenbaum研制的化学分子结构专家系统**DENDRAL**
- ◆ Shortliffe研制的诊断感染性疾病的专家系统**MYCIN**
- ◆ .....

---

## 四、框架表示法

1975年美国著名AI学者Minsky在其论文“A framework for representing knowledge”中提出了框架理论，并把它作为理解视觉、自然语言对话及其它复杂行为的基础。

---



人利用框架的思考过程

---

# 1. 框架的定义及表示形式

## (1) 定义

**框架**：是一种描述对象（事物、事件或概念等）属性的数据结构。在框架理论中，框架是知识表示的基本单位。

- ❖ 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**”。
  - ◆ **槽**：描述对象某一方面的属性；
  - ◆ **侧面**：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**。
  - ◆ 槽值或侧面值可以是数值、字符串、布尔值、在满足某个给定条件时要执行的动作、过程，或另一**框架的名字**
-

## (2) 框架的一般表示形式

<框架名>

槽名1:	侧面名 <sub>11</sub>	侧面值 <sub>111</sub> , ... , 侧面值 <sub>11P1</sub>
	⋮	
	侧面名 <sub>1m</sub>	侧面值 <sub>1m1</sub> , ... , 侧面值 <sub>1mPm</sub>
槽名n:	侧面名 <sub>n1</sub>	侧面值 <sub>n11</sub> , ... , 侧面值 <sub>n1P1</sub>
	⋮	
	侧面名 <sub>nm</sub>	侧面值 <sub>nm1</sub> , ... , 侧面值 <sub>nmPm</sub>
约束:	约束条件 <sub>1</sub>	
	⋮	
	约束条件 <sub>n</sub>	

---

## 2. 框架网络

❖ 框架网络：具有横向联系及纵向联系的一组框架

◆ 横向联系：

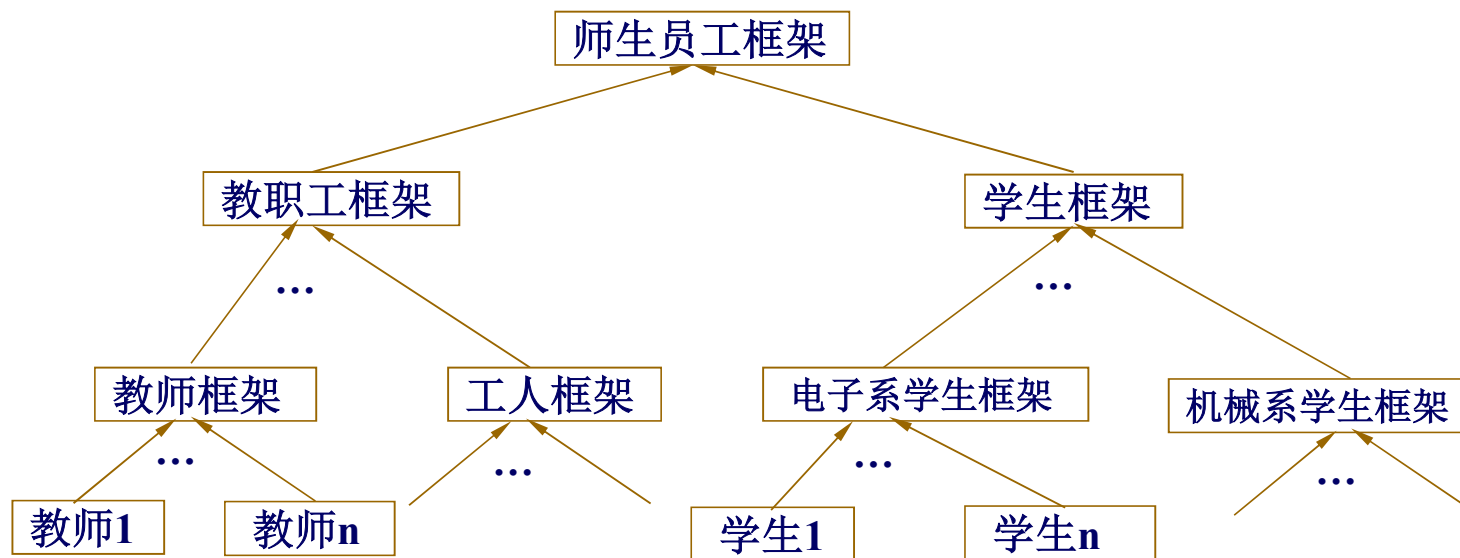
- 框架的槽值或侧面值可以是另一个框架的名字

◆ 纵向联系：

- 抽取共同属性构成一个上层框架，对专有属性分别构成下层框架，并在下层设立专用的槽（一般称“继承”槽）
-



# 框架网络例



---

### 3. 框架系统求解问题的基本过程

- ❖ 用一个框架表示问题
  - ❖ 与知识库中已有的框架匹配，找出一个或几个可匹配的预选框架作为初步假设，并在此初步假设的引导下收集进一步的信息
  - ❖ 用某种评价方法对预选框架进行评价，以便决定是否接受它
-

---

## 4. 框架表示法的特点

### ❖ 优点

- ◆ 结构性：便于表达结构性知识
- ◆ 继承性：下层框架可继承上层框架的槽值，也可进行补充和修改
- ◆ 自然性：与人在观察事物时的思维活动一致

### ❖ 缺点

- ◆ 主要缺点是不善于表达过程性的知识
-

---

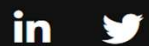
## 五、语义网络表示法

- ❖ 语义网络是奎廉(**J.R.Quillian**) 于**1968**年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(**Hendrix**)又对全称量词的表示提出了语义网络分区技术。
  - ❖ 目前，语义网络已成为人工智能中应用较多的一种知识表示方法，尤其是在自然语言处理方面。
-

---

# 知识图谱

- ❖ 谷歌在2012年首先提出知识图谱的概念
  - ❖ 本质上是语义网络，旨在描述客观世界的概念实体事件及其之间的关系。
  - ❖ 特点：一个非常大规模的语义网络
    - ◆ 目前实体的数据已超过数十亿级，有达到千亿级的实例关系，规模非常巨大
-



[Home](#) • Knowledge Graph Search

# Knowledge Graph Search

Search the entire Google “Knowledge Graph” database for a person, company, product, place, or other entity.

If you found this helpful please come say hello [Follow @carlhendy](#)

**Search**

Any Entity



Any Language



Tweet



Share



Post



Email



## 知识图谱

实体

100

## 描述

- 基于图的数据结构
- 关系网络
- 大规模语义网络
- 显示知识发展进程与结构关系的一系列各种不同的图形
- 推动人工智能发展的核心驱动力之一
- 语义网络的知识库

## 属性 仅显示部分热门信息 · 更多请点击或直接搜索问题

提出	2012年	组成	节点
前身	语义网	技术委员会主席	葛浩
应用	智能搜索	基本组成单位	实体-关系-实体
核心	模式层	初衷	搜索体验
发起人	肖仰华	作用	衔接企业的数据

## 标签 仅显示部分来源充分的信息 · 点击可显示详情

技术 产品 知识库 AI技术 概念 人工智能技术 工具 系统 语义网络  
服务 应用 公司 平台 数据结构 引擎

## 近义项

KG 知识域可视化 知识领域映射地图 KGs Knowledge Graph

## 主要学习来源

转型AI产品经理需要掌握的硬知识二：AI常见概念和算法梳理 - 简书  
www.jianshu.com · 2017年12月20日

## 知识图谱是什么？ - 推酷

www.tuicool.com · 2019年7月26日

知识图谱在互联网金融行业的应用\_平台事件\_互金知识\_网贷之家  
www.wdzj.com · 2018年5月3日

知识图谱（一）-基本概念 原 - 云+社区 - 腾讯云  
cloud.tencent.com · 2019年8月7日

AI产品经理必修课：知识图谱的入门与应用 - 推酷  
www.tuicool.com · 2019年9月4日

智能SOA：勾画企业“演进图谱”\_CNKI学问  
xuewen.cnki.net · 2007年12月12日

揭秘AI诈骗的矛与盾：AI反网络诈骗白皮书出炉|电信网络\_新浪财经...  
finance.sina.com.cn · 2020年1月5日

知识图谱赋能金融科技，看这一篇就够了-亿欧  
www.iyiou.com · 2019年10月27日

领域专家走进平安科技（PA Tech），共议知识图谱为医疗AI赋能路...  
science.china.com.cn · 2018年4月25日

历史的看，微信公号的红旗还能抗多久？-钛媒体官方网站  
www.tmtpost.com · 2019年7月14日



“基于图的数据结构” “关系网络” “大规模语义网络”  
“显示知识发展进程与结构关系的一系列各种不同的图形”  
“推动人工智能发展的核心驱动力之一” “语义网络的知识库”

属性 仅显示部分热门信息·更多请点击或直接搜索问题

提出	2012年	组成	节点
前身	语义网	技术委员会主席	葛浩
应用	智能搜索	基本组成单位	实体-关系-实体
核心	模式层	初衷	搜索体验
发起人	肖仰华	作用	衔接企业的数据

标签 仅显示部分来源充分的信息·点击可显示详情

技术 产品 知识库 AI技术 概念 人工智能技术 工具 系统 语义网络  
服务 应用 公司 平台 数据结构 引擎

近义项

科学知识图谱 KG 知识域可视化 知识领域映射地图 KGs Knowledge Graph

知识图谱

集合

展开内容 80

知识图谱是什么? - 推酷

www.tuicool.com · 2019年7月26日

知识图谱在互联网金融行业的应用\_平台事件\_互金知识\_网贷之家

www.wdzt.com · 2018年5月3日

知识图谱（一）-基本概念 原 - 云+社区 - 腾讯云

cloud.tencent.com · 2019年8月7日

AI产品经理必修课：知识图谱的入门与应用 - 推酷

www.tuicool.com · 2019年9月4日

智能SOA：勾画企业“演进图谱”\_CNKI学问

xuewen.cnki.net · 2007年12月12日

揭秘AI诈骗的矛与盾：AI反网络诈骗白皮书出炉|电信网络\_新浪财经...

finance.sina.com.cn · 2020年1月5日

知识图谱赋能金融科技，看这一篇就够了-亿欧

www.iyiou.com · 2019年10月27日

领域专家走进平安科技（PA Tech），共议知识图谱为医疗AI赋能路...

science.china.com.cn · 2018年4月25日

历史的看，微信公号的红旗还能抗多久？-钛媒体官方网站

www.tmtpost.com · 2019年7月14日



---

❖ 在问答研究中，自然语言问题的理解或者语义表示是一个难题。

❖ 同样的语义，表示方式可以不同

◆ How many people are there in Shanghai ?

◆ What is the population of Shanghai ?

若把上海换成北京、深圳.....

❖ 表示形式相近，但表达的语义完全不同

◆ 他说你了吗？

◆ 你说他了吗？

---

---

# 1. 语义网络的概念

- ❖ **语义网络**：用实体及其语义关系来表达知识的一种网络图
  - ❖ 从图论的观点看，它是一个“带标识的有向图”。
    - ◆ **节点：实体**，表示各种事物、概念、情况、属性、状态、事件、动作等；
    - ◆ **弧：语义**，表示它所连结的两个实体之间的语义联系；
    - 节点和弧都必须带标识：区分对象以及对象间的语义联系。
  - ❖ 最简单的语义网络：  

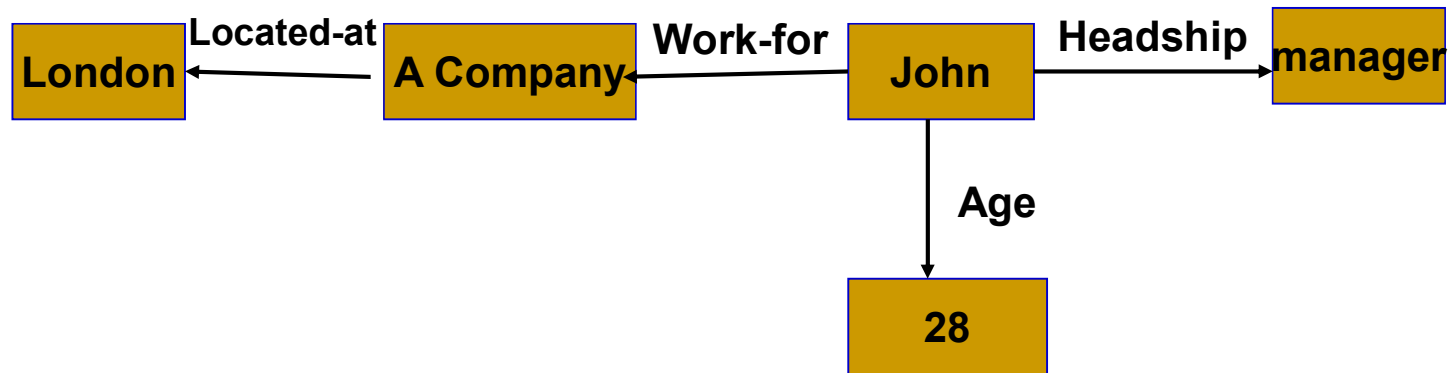
（节点1，弧，节点2）
-

---

## 2. 基本的语义关系

- ❖ 实例关系 **ISA** : 表示一个事物是另一个事物的一个实例
  - ❖ 分类关系 **AKO** : 表示一个事物是另一个事物的一种类型
  - ❖ 成员关系 **A-Member-of**: 表示一个事物是另一个事物的一个成员
  - ❖ 属性关系:指事物和其属性之间的关系, 如**Have**、**Can**、**Age**
  - ❖ 包含关系: 如**Part\_of**
  - ❖ 时间关系: 不同事件在其发生时间方面的先后次序关系。如**before**、**after**
  - ❖ 位置关系: 指不同事物在位置方面的关系。如**Located-on**等
  - ❖ 相近关系: 指不同事物在形状、内容等方面相似或接近。如**Similar-to**、**Near-to**
-

例



---

### 3. 语义网络系统求解问题的基本过程

- ❖ 语义网络系统指用语义网络表示知识的问题求解系统。
  - ❖ 语义网络系统两大组成部分:
    - ◆ 由语义网络构成的知识库
    - ◆ 语义网络推理机
-

---

❖ 在语义网络系统中，问题的求解一般通过匹配实现：

- ◆ 根据待求解问题的要求构造一个网络片断，其中有些节点或弧的标识是空的，反映待求解的问题；
  - ◆ 依网络片断到知识库中寻找可匹配的网络，以找出所需要的信息。
    - ▣ 这种匹配一般不是完全的，具有不确定性，因此需要解决不确定性匹配问题
  - ◆ 当问题的语义网络片断与知识库中的某语义网络片断匹配时，则与询问处匹配的事实即为问题的解。
-

---

## 4. 语义网络表示法的特点

### ❖ 优点

- ◆ 结构性
- ◆ 联想性
- ◆ 自然性

### ❖ 缺点

- ◆ 非严格性
- ◆ 处理上的复杂性

### ❖ 应用

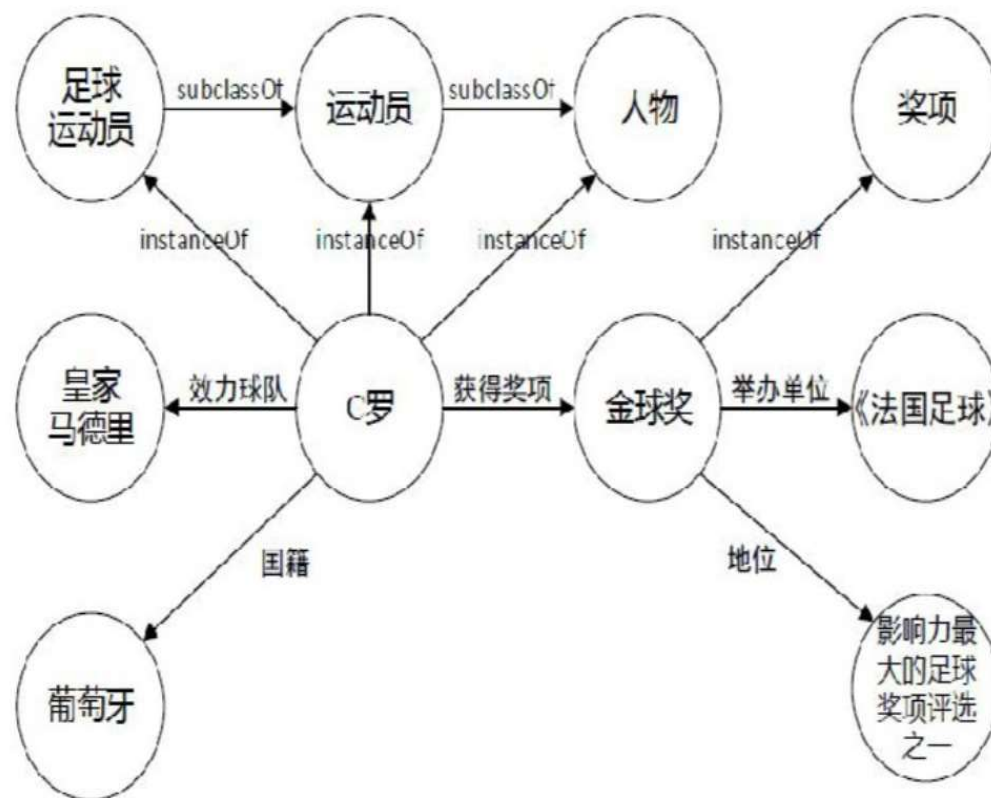
- ◆ **Walker** 研制的自然语言理解系统
- ◆ **Garbonell** 研制的回答地理问题的教学系统
- ◆ **Mytopoulous** 研制的自然语言理解系统
- ◆ **Simmon** 研制的自然语言理解系统
- ◆ **Hays**研制的描写概念的系统

。 。 。

# 基于语义网络/知识图谱的问题回答

❖ 知识图谱的简单推理：为什么说C罗很牛？

- ◆ C罗获得过金球奖（C罗跟金球奖之间的关系是获得奖项的一个关系）
- ◆ 金球奖具有非常高的地位（金球奖跟影响力最大的足球评选奖项是一个地位的关系）
- C罗很牛





---

## 六、剧本（脚本）表示法

剧本表示法是1975年R.C. Schank依据他的概念依赖理论而提出的一种知识表示方法。脚本与框架类似，由一组槽组成，用来表示特定领域内一些事件的发生序列。

---

---

## 一) 概念依赖理论

【**难点**】在人类的各種知識中，**常識性知識**數量最多、涉及面最寬、關係最複雜，很難形式化表示。

概念依赖理论的基本思想：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。

---

## Schank对11种动作（ACT）的原子化

- （1）PROPEL:应用物理力量（推、拉、打等）于一对象。
- （2）GRASP: 一个演员抓起一个物理对象。
- （3）MOVE: 演员身体的一部分变换空间位置，如抬手等。
- （4）PTRANS: 物理对象变换位置，如走进、跑出等。
- （5）ATRANS:抽象关系的改变，如传递、赠送、革命等。
- （6）ATTEND:用某个感官获取信息，如用目光搜索等。
- （7）INGEST:演员把某个东西吸入体内，如吃、喝等。
- （8）EXPEL:演员把某个东西送出体外，如呕吐、落泪等。
- （9）SPEAK:演员产生一种声音，如唱歌、尖叫等。
- （10）MTRANS:信息的传递，如读报、看信、看电视等。
- （11）MBUILD:由旧信息形成新信息，如“眉头一皱，计上心来”。

---

## 基于概念依赖关系的故事情节理解

(1) Schank定义上述11中动作的**主要意图**：上述原子概念主要不是用于表示动作本身，而是表示动作的结果，并且是本质的结果，因此也可以认为是这些概念的推理。

(2) 基于概念依赖关系的**故事情节理解办法**：事先编好许多剧本，每个剧本代表日常生活中发生的一种事件，它把这种事件的典型情节规范化，编成一种程式。

(3) Schank实现了SAM(Script Applier Mechanism)系统。其功能是：接受一个故事后，首先做**语法分解**工作，按照概念依赖关系的模式**化成内部表示**，然后从库中取出相应的剧本进行**匹配**，根据事先确定的剧本情节来理解故事。

---

---

## 二） 剧本（脚本）及其实例

1、**剧本**：描述特定范围内原型事件的结构。

### 2、剧本的组成

（1）**进入条件**：指出剧本所描述的事件可能发生的先决条件，即事件发生的**前提条件**。

（2）**角色**：描述事件中可能出现的人物。

（3）**道具**：描述事件中可能出现的有关物体。

（4）**场景**：描述事件序列，可以有多个场景。

（5）**结局**：给出剧本所描述的事件发生以后必须满足的条件。

---

---

### 3、Schank餐馆剧本

**剧本：**就餐

**进入条件：**顾客饿了，品尝佳肴，招待亲友，顾客有钱

**角色：**顾客、服务员、厨师、经理、老板

**道具：**食品、桌子、菜单、钱

**场景：**

#### **第一幕：进入餐馆**

PTRANS：步入餐馆

ATTEND：用目光寻找空桌

MBUILD：选定桌子

MOVE：坐下

---

---

## 第二幕：定菜

ATRANS：服务员送菜单

MTRANS：读菜单

MBUILD：选定所要的菜

MTRANS：告诉服务员

ATRANS：付钱

## 第三幕：吃饭

ATRANS：服务员上菜

INGEST：吃饭

## 第四幕：离开

MOVE：站起身来

PTRANS：步出餐馆

**结局：** 顾客吃了饭；顾客花了钱；老板挣了钱；餐厅食品减少了。

---

---

### 三）利用剧本进行故事理解时应注意的问题

每个剧本应有适应临时出现的新情况的能力，包括：

1、子剧本调用。往往由意外情况引起。

2、排除故障。剧本的“进入条件”往往是隐含的，有时可能因某个条件不成而ACT无法进行，这时要求剧本备有变通动作来排除这种障碍。

3、调用剧本以外的知识

应该有许多备用知识（即额外知识）存在库中供剧本调用。

额外知识可以用来排除障碍和调用子框架。

4、提炼、忘却和想象

（1）故事情节收缩以理解故事的核心；

（2）能通过想象、子剧本调用、插入障碍、设置变通等方式使故事情节膨胀

。

---



---

## 四）关于剧本表示法的说明

**1、剧本表示法的局限性：**比较呆板，能力也有限。

**2、剧本表示的其它可能用途。**用于自然语言理解方面。可以设想**将剧本与计算机动画进行结合**，即根据一段故事情节，利用剧本理解其情节和含义，然后，通过图形学、动画等技术，绘制出该故事的场景、角色、道具，并根据理解的故事发展过程，进行动画情节的演进。

陆汝钐等人在20世纪90年代研发了一套基于故事情节理解的动画——《天鹅》。

---

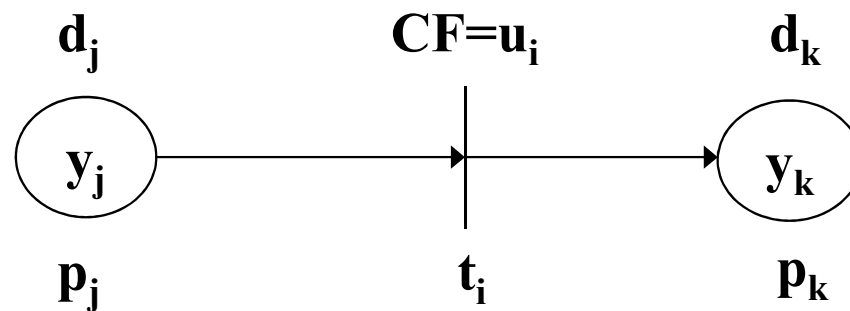
## 七、Petri网表示法

德国学者Cah Abam Petri于1962年首先提出Petri网的概念，用于构造系统模型及进行动态特性分析，后来逐渐被用作知识表示方法。

# 一) Petri网原理及其表示知识的方法

## 1、Petri网的基本原理

Petri网的三个基本元素：位置、转换、标记



## 2、基于Petri网的知识表示方法

**Pertri网常用一个八元组来表示知识间的因果关系**

具体形式：

**$(P, T, D, I, O, f, \alpha, \beta)$**

P——位置的有限集，记为 $P = \{p_1, p_2, \dots, p_n\}$ ；

T——转换的有限集，记为 $t = \{t_1, t_2, \dots, t_n\}$ ；

D——命题的有限集，记为 $D = \{d_1, d_2, \dots, d_n\}$ ；

I——输入函数，表示从位置到转换的映射；

O——输出函数，表示从转换到位置的映射；

f——相关函数，表示规则强度，取值范围 $[0, 1]$ ；

$\alpha$ ——相关函数，表示位置对应命题的可信度 $[0, 1]$ ；

$\beta$ ——相关函数，表示从位置到命题的映射。

### 3、基于Petri网的知识表示方法举例

设有产生式规则集：

$r_1$ : IF  $d_1$  THEN  $d_2$  (CF = 0.85)

$r_2$ : IF  $d_2$  THEN  $d_3$  (CF = 0.80)

$r_3$ : IF  $d_2$  THEN  $d_4$  (CF = 0.80)

$r_4$ : IF  $d_4$  THEN  $d_5$  (CF = 0.90)

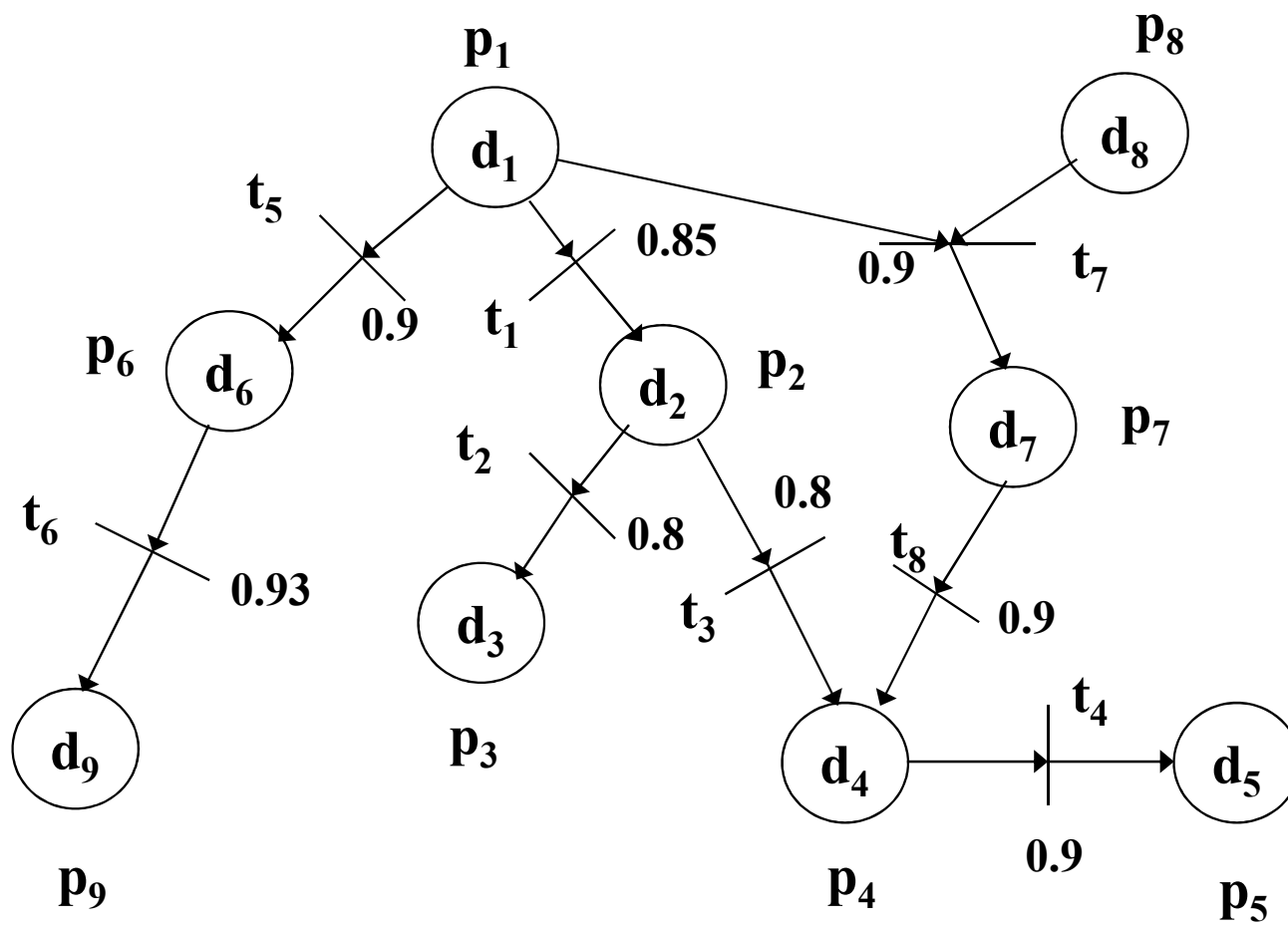
$r_5$ : IF  $d_1$  THEN  $d_6$  (CF = 0.90)

$r_6$ : IF  $d_6$  THEN  $d_9$  (CF = 0.93)

$r_7$ : IF  $d_1$  AND  $d_8$  THEN  $d_7$  (CF = 0.90)

$r_8$ : IF  $d_7$  THEN  $d_4$  (CF = 0.90)

试画出上述规则集的Petri网。



上述8条产生式规则集的Petri网

---

## 二) Petri网表示法的特点

- ❖ 便于描述系统状态的变化及对系统特性进行分析
  - ❖ 可在不同层次上变换描述，而不必注意细节及相应的物理表示，这样就可把注意力集中到某个层次的研究上。
-

---

# 知识表示方法

- ❖ 状态空间 (State Space)
  - ❖ 与或图 (And-Or Graph)
  - ❖ 一阶谓词 (First Order Predicate)
  - ❖ 产生式 (Production)
  - ❖ 框架 (Framework)
  - ❖ 语义网络 (Semantic Network)
  - ❖ 剧本 (Script)
  - ❖ Petri网 (Petri Network)
  - ❖ 过程 (Procedure)
  - ❖ 面向对象 (Object-Oriented)
  - ❖ 信念网 (Belief Network)
  - ❖ 本体论 (Ontology)
  - ❖ .....
-



---

Q & A

---

---

课程部分材料来自他人  
和网络，仅限教学使用  
，请勿传播，谢谢！

---

---

## 八、过程表示法

AI界关于知识表示方法的两种不同观点：

**第一种观点：**认为知识主要是**陈述性的**，其表示方法应着重将其静态特性（即事物的属性及事物间的关系）表示出来。人们称以这种观点表示知识的方法为**陈述式**或**说明性表示方法**。

**第二种观点：**认为知识主要是**过程性的**，其表示方法应将知识及如何使用这些知识的控制策略均表示为求解问题的过程，称以这种观点表示知识的方法为**过程性表示方法**，或**过程表示法**。

---

---

## 过程表示法的基本思想

侧重于对知识的利用，把与问题有关的**知识**以及如何运用这些知识求解问题的**控制策略**都表述为一个或多个**求解问题的过程**，每个过程就是一段程序，用于完成对一个具体事件或情况的处理。

在问题求解过程中，当需要使用某个过程时就调用相应的程序并执行之。

---