重庆邮电大学

计算机科学与技术学院

人工智能原理

知识表示与逻辑推理

知识是人工智能的基石！

人工智能
（功能模拟）
基本问题

知识工程的三大要素

- 知识表示
- 知识获取
- 知识应用

研究及应用热点——知识图谱、知识自动化

Fish 于2012 年出版的Knowledge Automation(知识自动化) 一书

知识及其表示

1. 什么是知识呢?
2. 数据、信息、知识的区别与联系?

* (Feigenbaum)知识:是经过削减、塑造、解释和转换的信息；即：知识是经过加工的信息。
* 知识:是由特定领域的描述、关系和过程组成的。
* 知识:是事实、信念和启发式规则

知识表示内涵

- 知识表示：是研究用机器表示知识的可行性、有效性的一般方法，是一种数据结构与控制结构的统一体，既考虑知识的存储又考虑知识的使用。知识表示可看作是一组描述事物的约定，即把人类知识表示成机器能够处理的数据结构。

知识表示= 知识的数据结构+处理机制

知识表示

- 状态空间法
- 问题归约法
- 谓词逻辑法
- 语义网络法
- ……

引入：传教士与野人过河问题

有3个野人和3个传教士在河的左岸，需要过河到右岸。现在有一艘小船，最多能容2人，在渡河时，无论是在左岸还是右岸，如果野人的数目超过传教士的数目，野人就会吃掉传教士。问怎样才能安全渡河?

状态空间法（State Space Representation）

* 问题求解技术主要是两个方面：
  * **问题的表示**
  * **求解的方法**

* 状态空间法
  * **状态**（state）
  * **算符**（operator）
  * **状态空间法**：基于解答空间的状态表示和求解方法。

问题描述

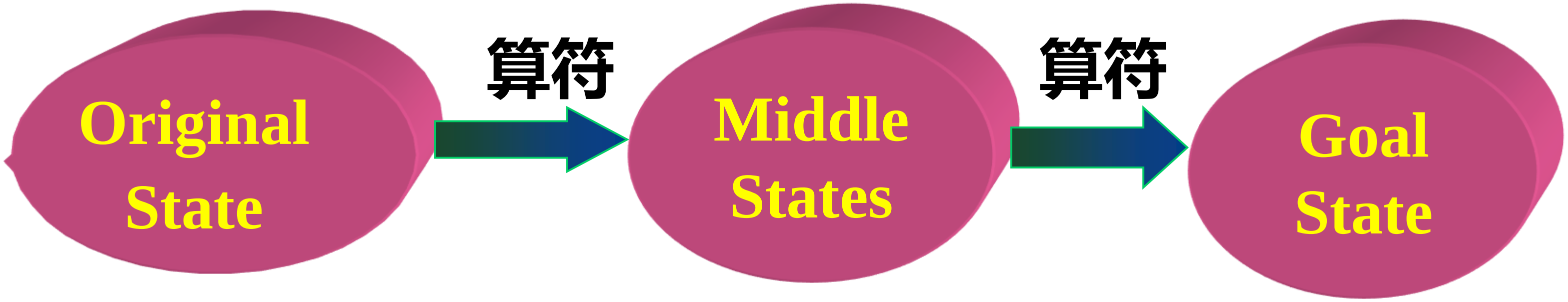
* 状态空间法的3个关键部分：
  * 状态：为描述某类问题中不同事物间的差别而引入的一组最少变量（状态变量）q_0，q_1，…，q_n的有序集合；为所有状态变量确定一组值，就得到一个具体状态。
  * 算符：使问题从一种状态变化为另一种状态的手段称为操作符或算符。如：走步、规则、过程、数学算子、运算符号和逻辑符号。
  * 问题的状态空间：是一个表示该问题全部可能状态及其关系的图（状态图），可以用一个三元组（S，F，G）表示：初始状态、算符集合、目标状态。

状态空间法

状态变量:

$(q_0, q_1, \ldots, q_n)$

算符



状态空间

状态空间法的问题求解方法

状态图示法

* 状态图：有向图、与或图
* 节点、边、路径
* (状态转移：边、路径)代价
* 图的显示说明、隐示说明

状态空间法举例1

传教士与野人过河问题

有3个野人和3个传教士在河的左岸，需要过河到右岸。现在有一艘小船，最多能容两人，在渡河时，无论是在左岸还是右岸，如果野人的数目超过传教士的数目，野人就会吃掉传教士。问怎样才能安全渡河?

状态空间法

(1)问题描述：

定义状态变量：（M，N，C）表示传教士、野人、船在左岸状态

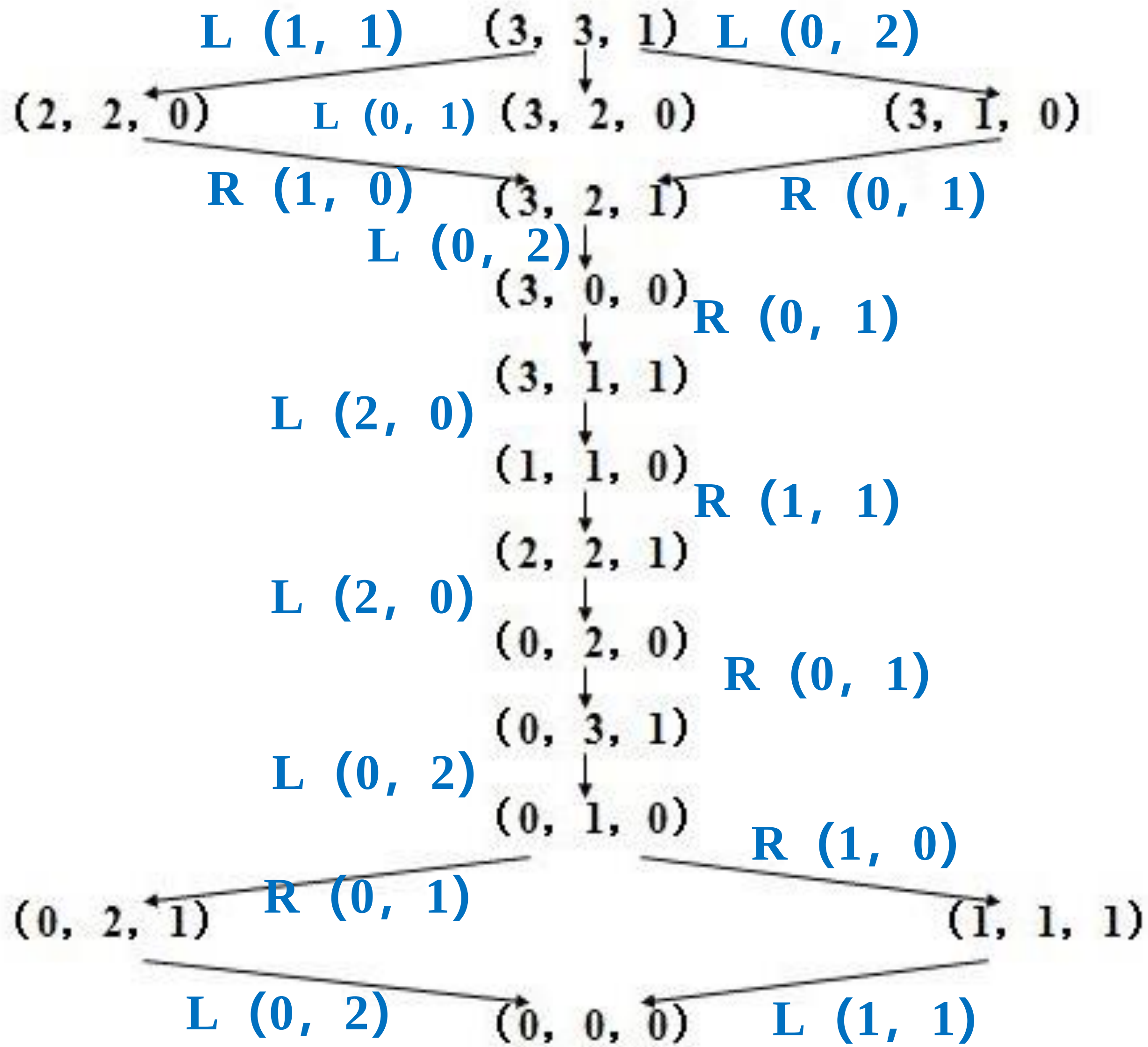
问题：起始状态（3，3，1）→ 目标状态（0，0，0）

(2) 定义算符：L（i，j）表示将i个传教士j个野人从左岸运送到右岸，L (1,0) , L(0,1),L(2,0),L(0,2),L(1,1)；

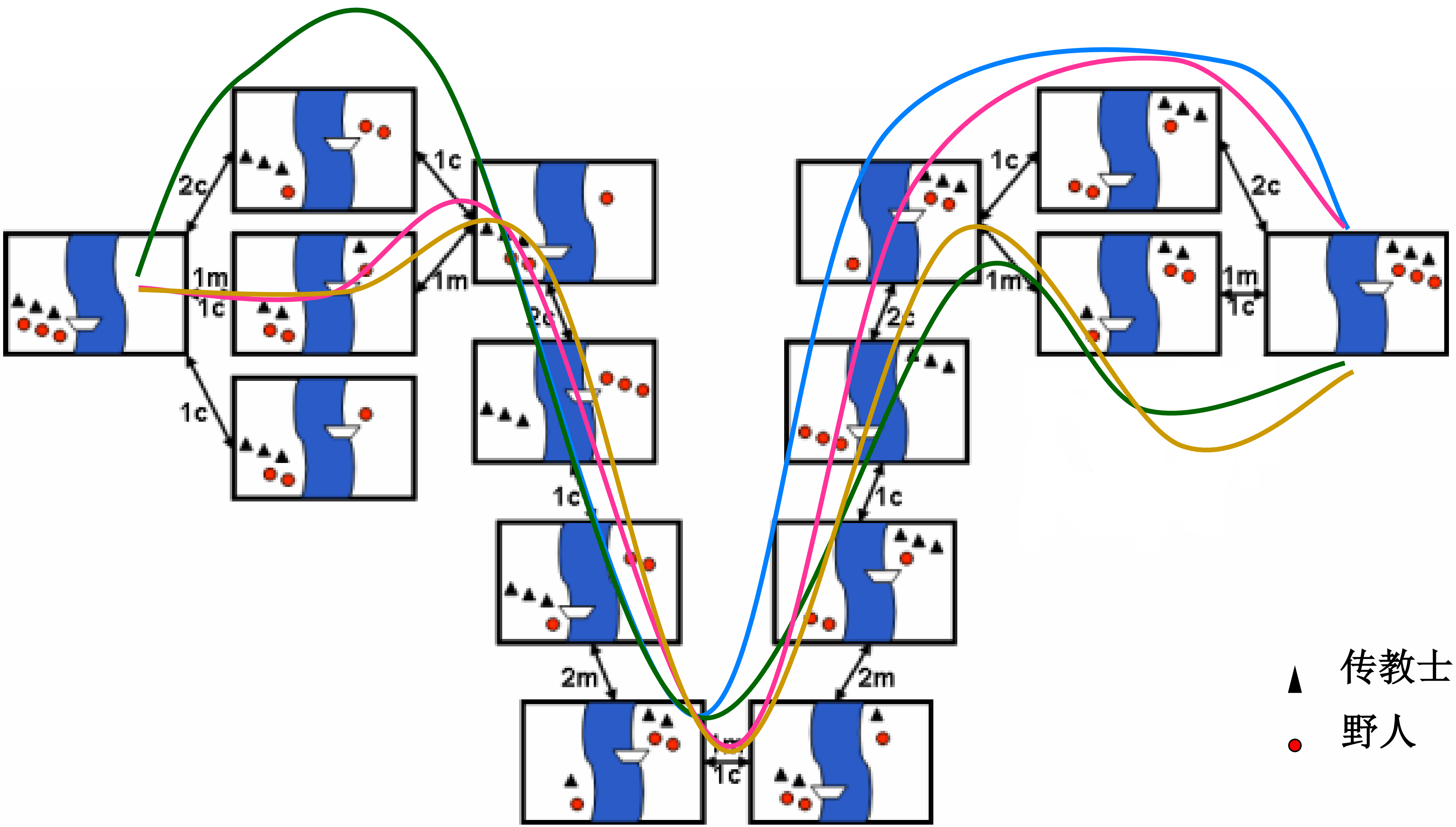
R（i，j）表示将i个传教士j个野人从右岸运送到左岸，R (1,0) , R(0,1),R(2,0),R(0,2),R(1,1)

传教士与野人过河问题

状态空间图



重庆邮电大学


2c
1m
1c
1c
1c
1m
2c
1c
2m
1m
1c
2m
1c
2c
1c
1m
2c
1m
1c
传教士
野人

状态空间法举例2

N×N阶数码问题(如3数码、8数码、15数码等)

8数码问题（3×3阶数码问题）：

将分别标有数字1，2，3，…，8的八块正方形数码牌任意地放在一块3×3的数码盘上。放牌时要求不能重叠。在3×3的数码盘上有一个空格。现在要求按照每次只能将与空格相邻的数码牌与空格交换的原则，将任意摆放的数码盘逐步摆成某种特殊的排列。

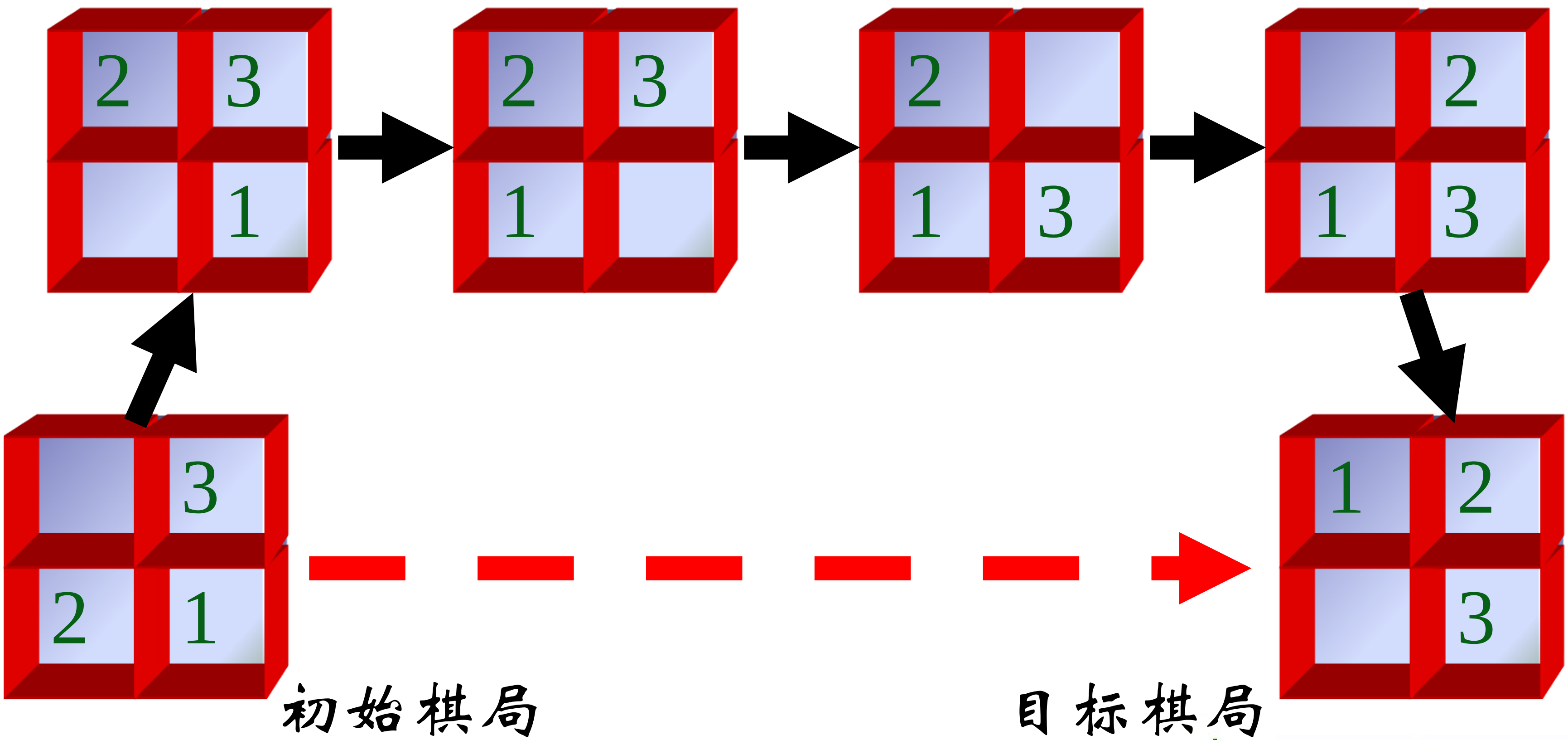
2	8	3
1		4
7	6	5

1	2	3
8		4
7	6	5

3数码难题（3 puzzle problem）

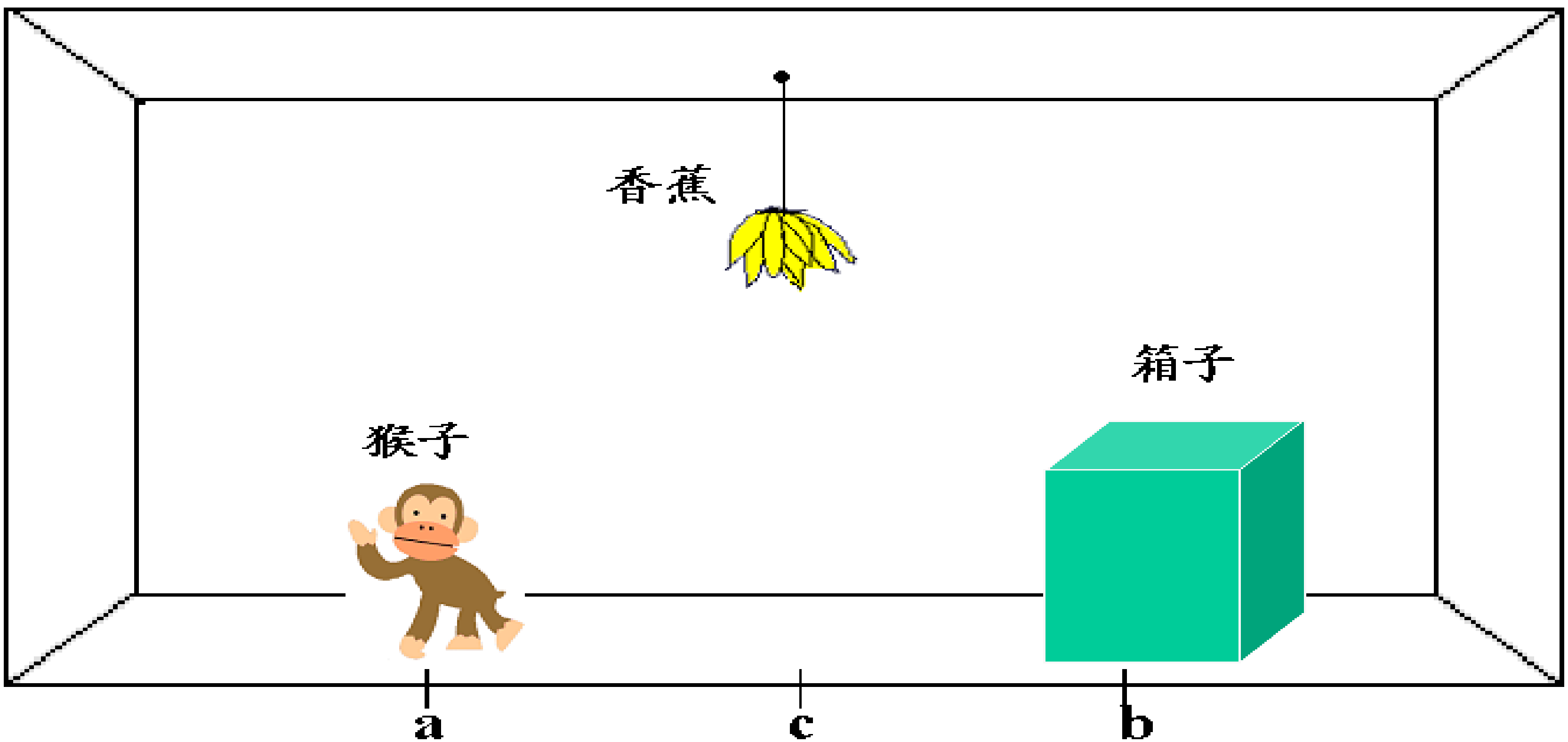
初始棋局

目标棋局


初始棋局
目标棋局

状态空间法举例3

* 例： 猴子摘香蕉问题



解题过程：

- 用一个四元表列（W，x，Y，z）来表示这个问题状态.
  - W 猴子的水平位置
  - x 当猴子在箱子顶上时取 $x = 1$；否则取 $x = 0$
  - Y 箱子的水平位置
  - z 当猴子摘到香蕉时取 z=1；否则取 z=0

- 这个问题的操作（算符）如下：
  - goto (U)：表示猴子走到水平位置U

$$(W, 0, Y, z) \xrightarrow{\text{goto}(U)} (U, 0, Y, z)$$

- pushbox (V)：表示猴子把箱子推到水平位置V

$$(W, 0, W, z) \xrightarrow{\text{pushbox}\ (V)} (V, 0, V, z)$$

•注意：要应用算符pushbox（V），就要求规则的左边，猴子与箱子必须在同一位置上，并且，猴子不是箱子顶上。这种强加于操作的适用性条件，叫做产生式规则的**先决条件。**

- climbbox：猴子爬上箱顶

$$(W, 0, W, z) \xrightarrow{\text{climbbox}} (W, 1, W, z)$$

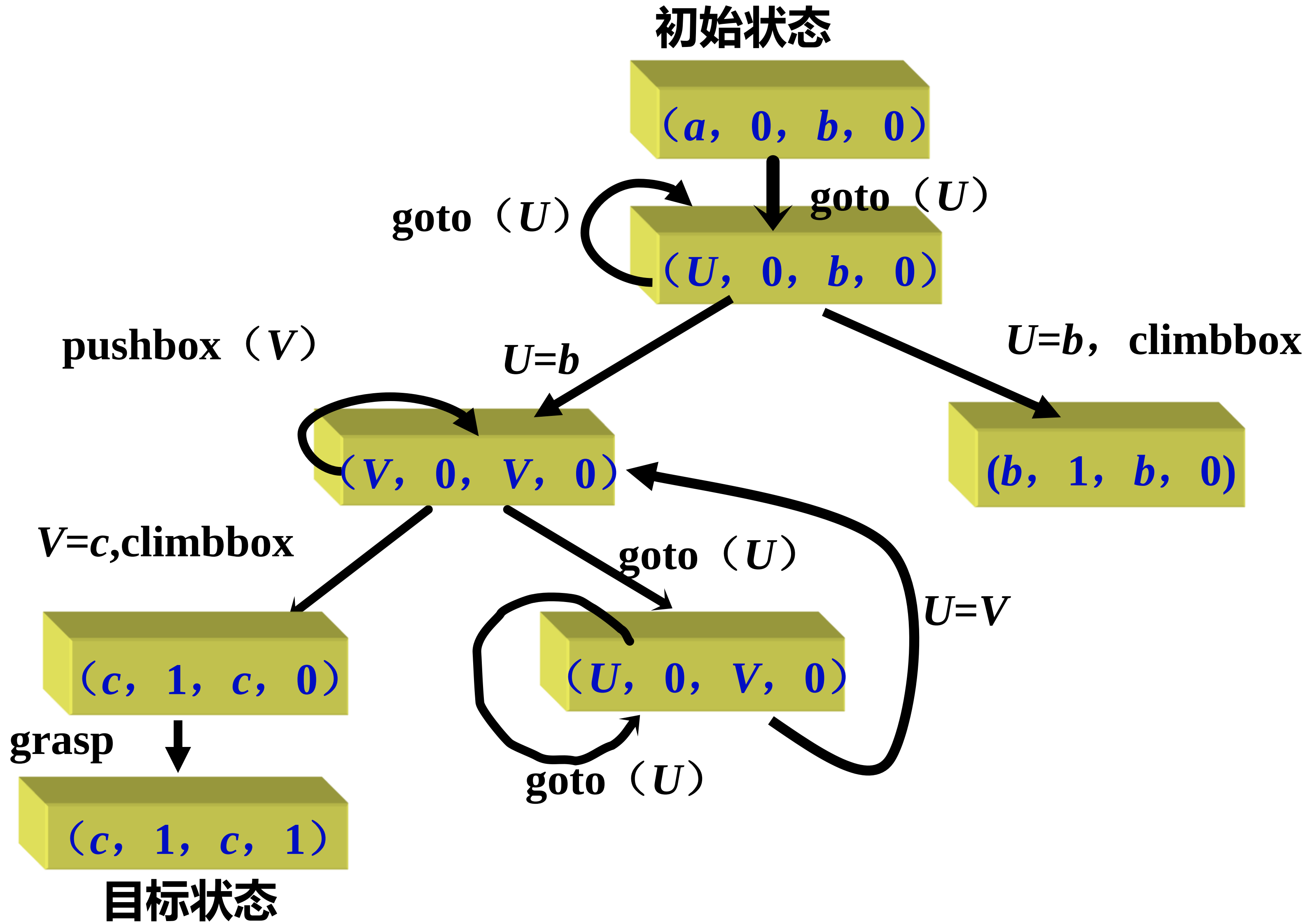
应用算符climbbox的先决条件是什么?

- grasp： 表示猴子摘到香蕉

$$(c，1，c，z) \xrightarrow{\text{grasp}} (c，1，c，1)$$

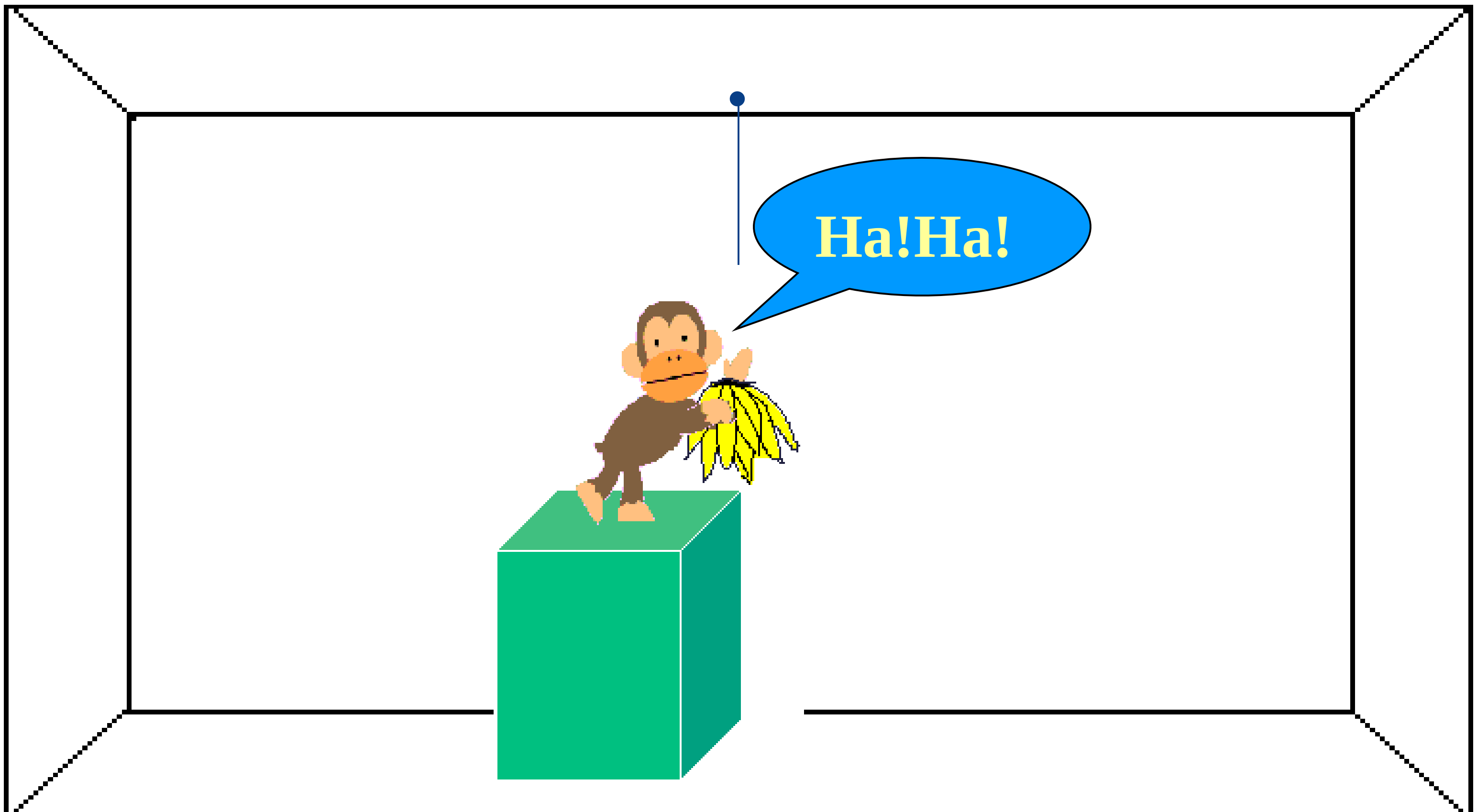
•令初始状态为(a，0，b，0)。这时，goto(U)是唯一适用的操作，并导致下一状态(U，0，b，0)。现在有3个适用的操作，即goto(U)，pushbox(V)和climbbox(若U=b)。把所有适用的操作继续应用于每个状态，就能够得到状态空间图，如下图所示。从图不难看出，把该**初始状态变换为目标状态的操作序列为：**

{goto(b), pushbox(c), climbbox, grasp}



猴子和香蕉问题的状态空间图

猴子和香蕉问题自动演示：



* 练习题：
* 1.画8数码的状态空间图

初始状态：

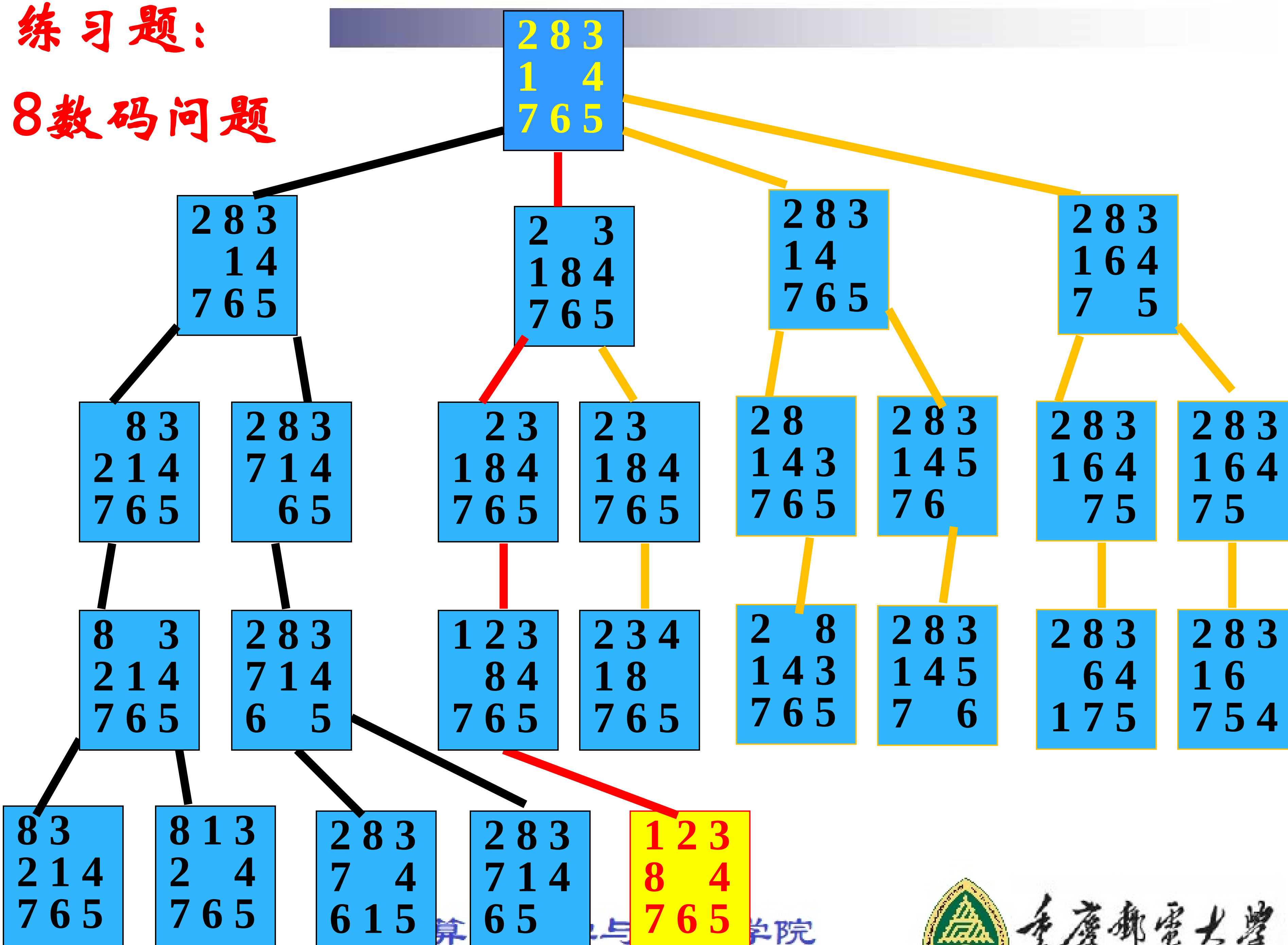
2	**8**	**3**
1		4
7	6	5

目标状态：

1	2	3
8		4
7	6	5

练习题：

8数码问题



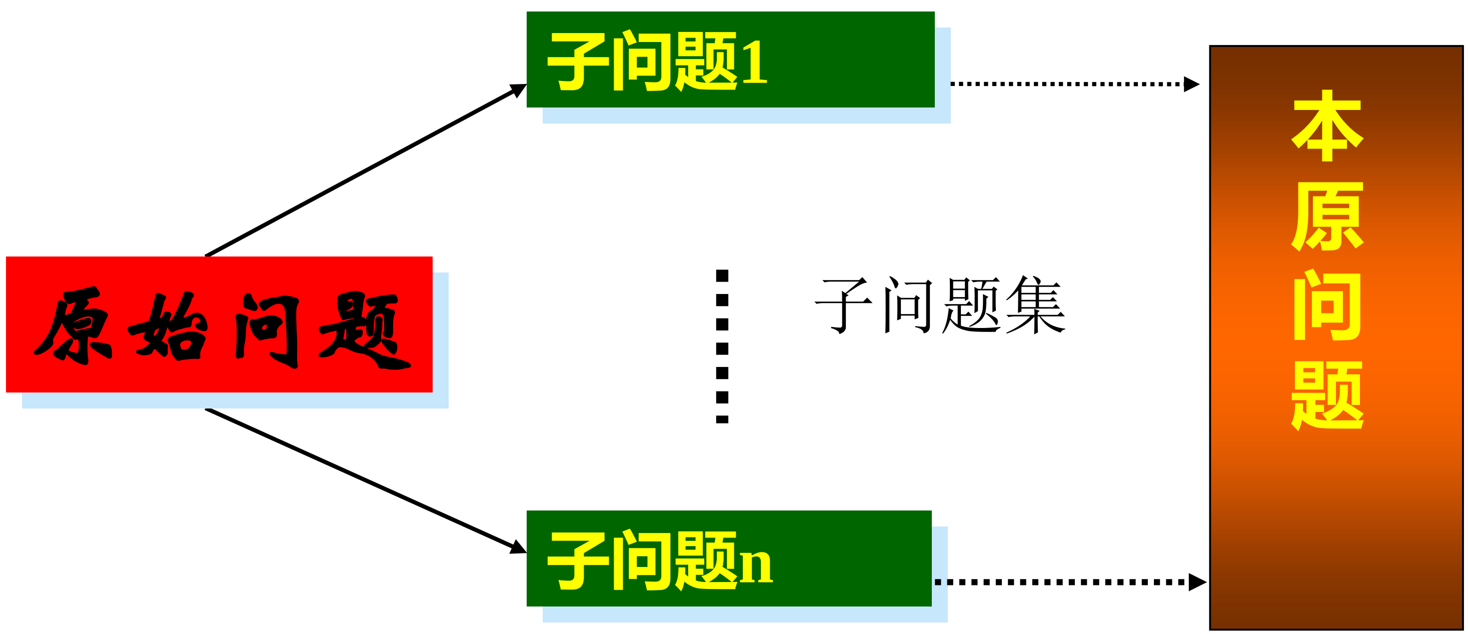
2 8 3 / 1 4 / 7 6 5
2 8 3 / 1 4 / 7 6 5
2 3 / 1 8 4 / 7 6 5
2 8 3 / 1 4 / 7 6 5
2 8 3 / 1 6 4 / 7 5
8 3 / 2 1 4 / 7 6 5
2 8 3 / 7 1 4 / 6 5
2 3 / 1 8 4 / 7 6 5
2 3 / 1 8 4 / 7 6 5
2 8 / 1 4 3 / 7 6 5
2 8 3 / 1 4 5 / 7 6
2 8 3 / 1 6 4 / 7 5
2 8 3 / 1 6 4 / 7 5
8 3 / 2 1 4 / 7 6 5
2 8 3 / 7 1 4 / 6 5
1 2 3 / 8 4 / 7 6 5
2 3 4 / 1 8 / 7 6 5
2 8 / 1 4 3 / 7 6 5
2 8 3 / 1 4 5 / 7 6
2 8 3 / 6 4 / 1 7 5
2 8 3 / 1 6 / 7 5 4
8 3 / 2 1 4 / 7 6 5
8 1 3 / 2 4 / 7 6 5
2 8 3 / 7 4 / 6 1 5
2 8 3 / 7 1 4 / 6 5
1 2 3 / 8 4 / 7 6 5

知识表示

- 状态空间法
- 问题归约法
- 谓词逻辑法
- 语义网络法
- ……

知识表示：问题归约法

问题归约法（Problem Reduction Representation）



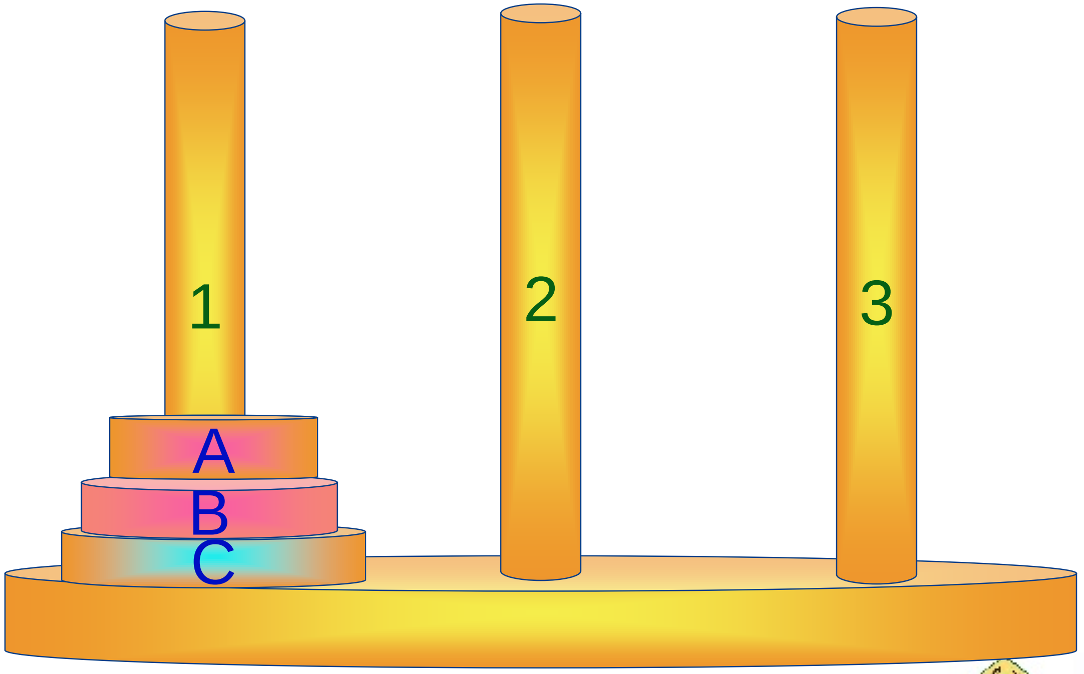
问题归约法

- 问题归约法的3个组成部分：
  - 一个初始问题描述；
  - 一套把问题变换为子问题的**操作符**；
  - 一套本原问题描述。
- 问题归约的实质：
  - 从目标(要解决的问题)出发逆向推理，把问题分解为子问题以及子问题的子问题，直至最后把初始问题归约为一个基础的本原问题集合。

问题归约法

问题归约描述（Problem Reduction Description）

* 梵塔难题



根据问题归约的思想，我们可把三阶梵塔问题分解为下面的三个子问题：

(1) 把*A*、*B*盘从1号杆移到2号杆。

(2) 把*C*盘从1号杆移到3号杆。

(3) 把*A*、*B*盘从2号杆移到3号杆。

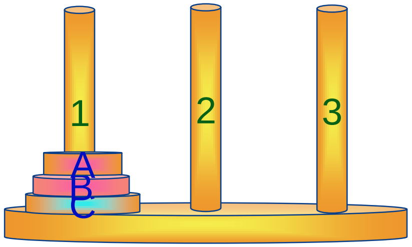
其中子问题(1)、(3)又分别可继续分解。

如果用三元组

$$(i,\ j,\ k)$$

i代表金盘C所在的杆号；j代表金盘B所在的杆号，k代表金盘A所在的杆号。初始状态(1, 1, 1)表示3个盘子都在1号杆。

三阶Hanoi塔的图（树）表示



(i,j,k)

i: C盘所在杆号，
j: B盘所在杆号，
k: A盘所在杆号。

(1,1,1)⇒(3,3,3)

(1,1,1)⇒(1,2,2)

(1,2,2)⇒(3,2,2)

(3,2,2)⇒(3,3,3)

(1,1,1) ⇒ (1,1,3)

(1,1,3) ⇒ (1,2,3)

(1,2,3) ⇒ (1,2,2)

(3,2,2) ⇒ (3,2,1)

(3,2,1) ⇒ (3,3,1)

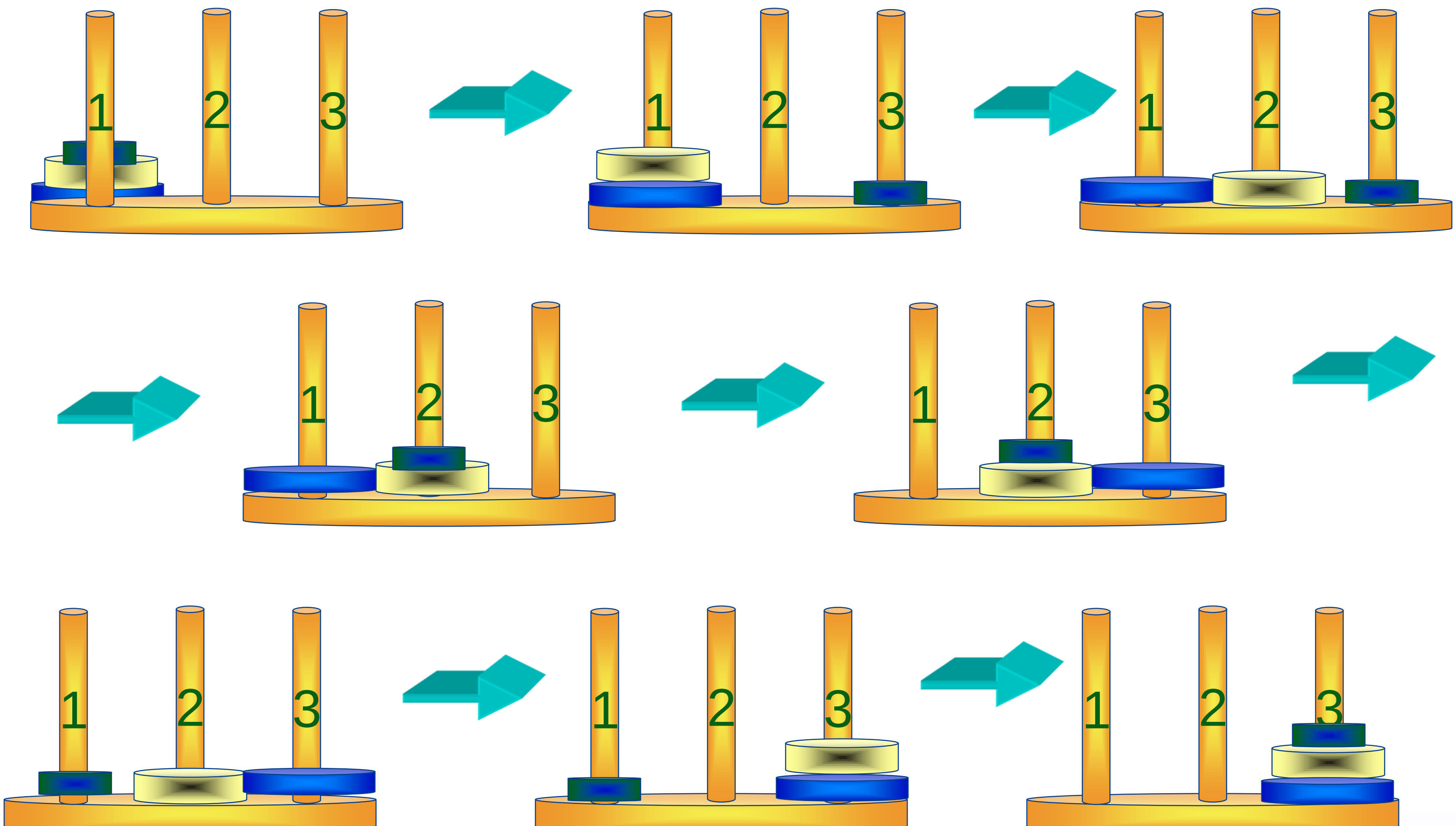
(3,3,1) ⇒ (3,3,3)

7个本原问题(终止节点)，将他们的解按从左至右的顺序排列，就得到原始问题解。

解题过程（3个圆盘问 题）

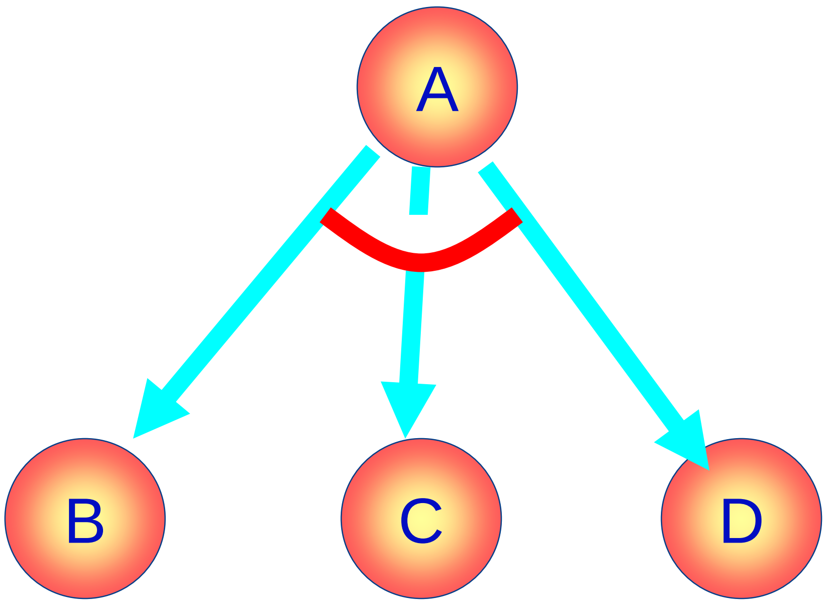
(1,1,1)⇒(3,3,3)

问题归约法

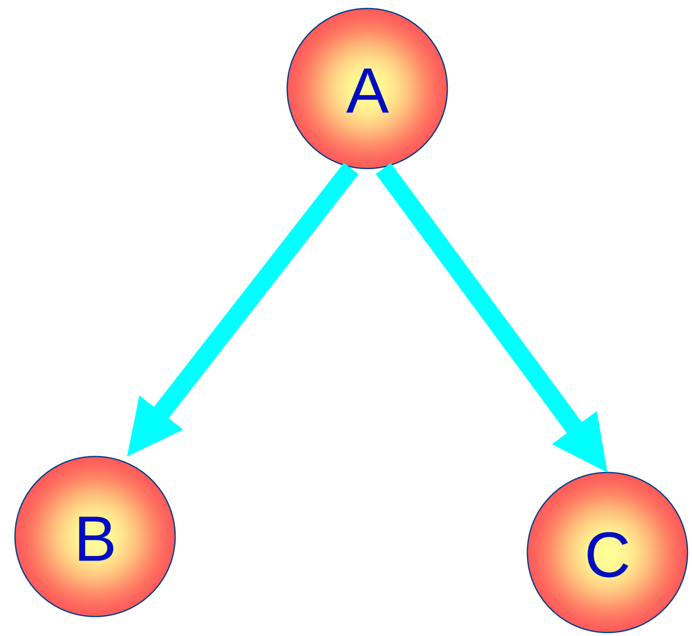


与或图

* 1.与图、或图、与或图



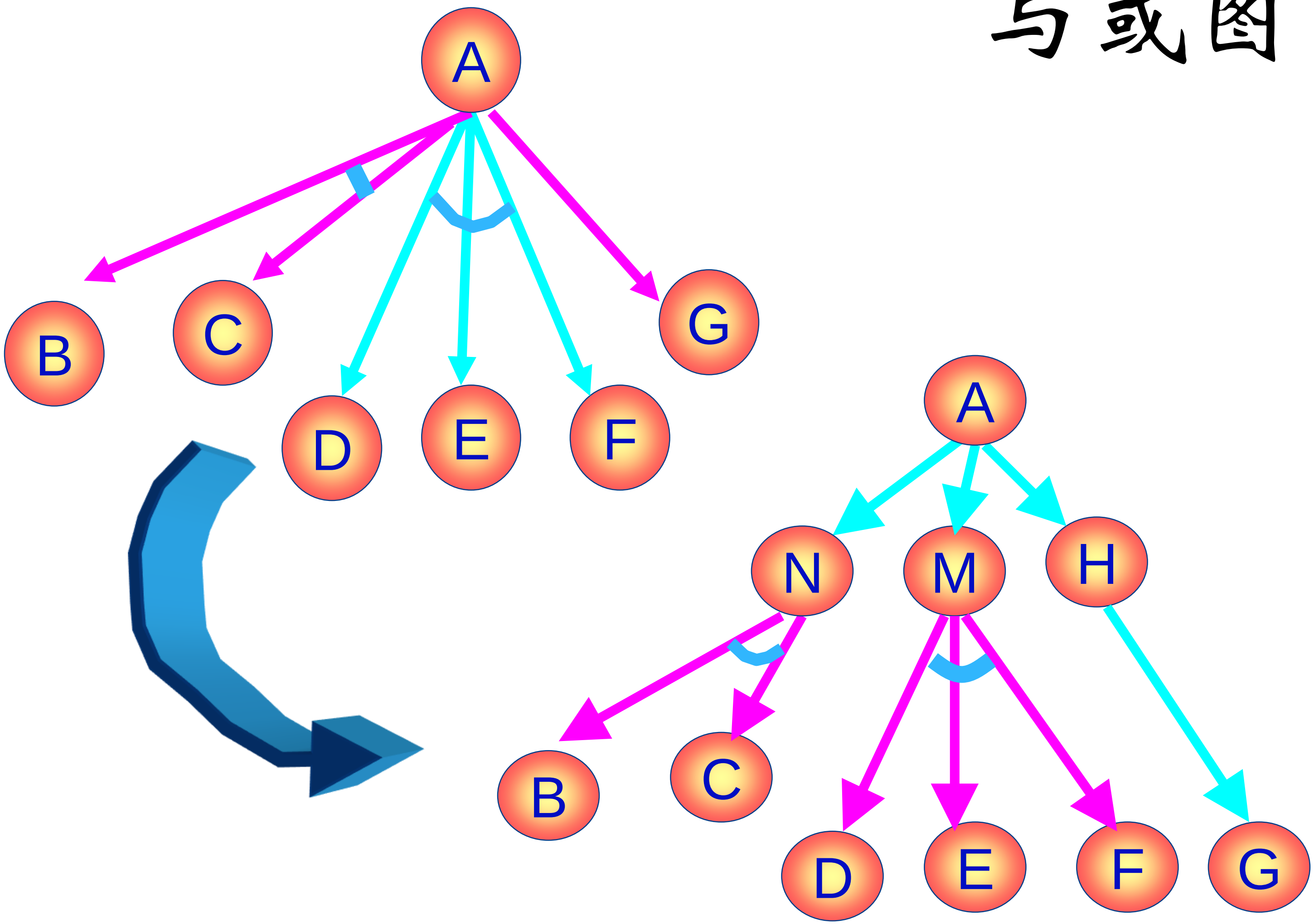
与图



或图

问题归约法

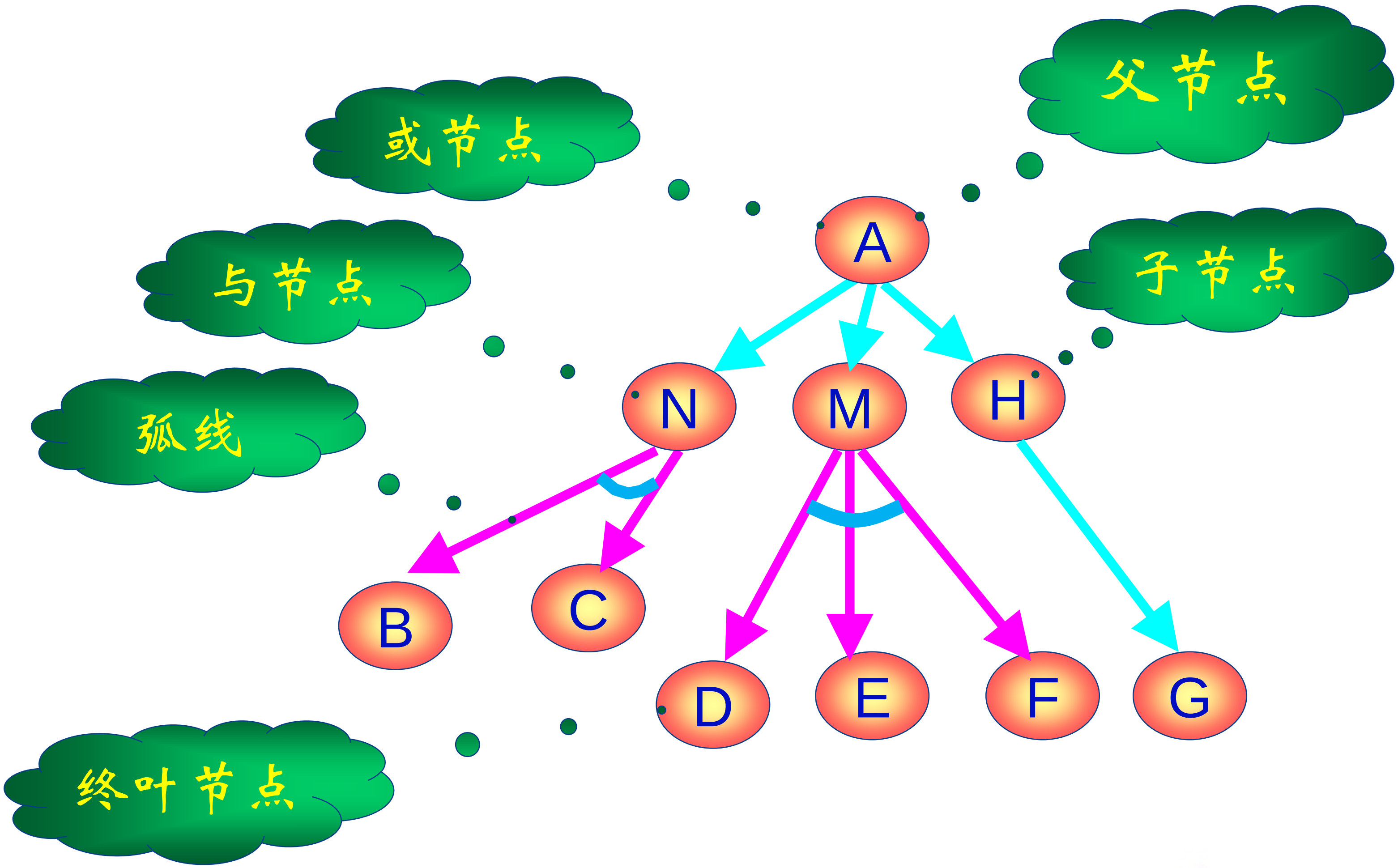
与或图



问题归约法

一些关于与或图的术语

与或图

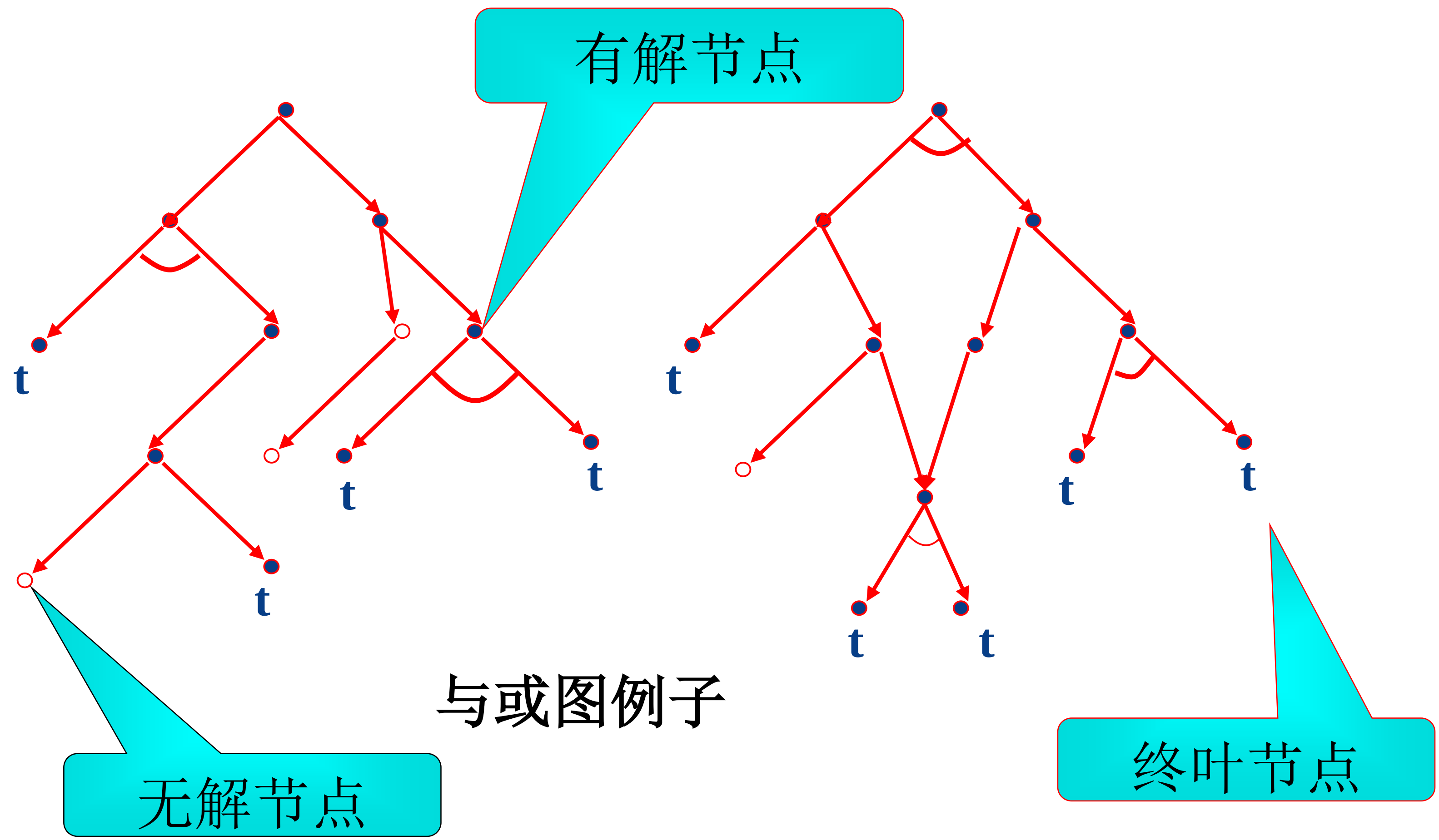


终叶节点对应本原问题（有明显解答）

一些关于与或图的术语

* 父节点、子（后继）节点、弧线
* 起始节点：对于于原始问题描述的节点
* 终叶节点：对应于本原问题的节点
* 或节点：只要解决某个问题就可解决其父辈问题的节点集合，如（M，N，H）。
* 与节点：只有解决所有子问题，才能解决其父辈问题的节点集合，如（B，C）和（D，E，F）。各个节点之间用一段小圆弧连接标记。
* 与或图：由与节点及或节点组成的结构图。

与或图



与或图例子

与或图

(1) 一个节点可解，则节点须满足下列条件之一：

① **终叶节点**是可解节点。

② 一个**与节点**可解，当且仅当其子节点全都可解。

③ 一个**或节点**可解，只要其子节点至少有一个可解。

(2) 一个节点不可解，则节点须满足下列条件之一：

① 没有后裔的非终叶节点（端节点）是不可解节点。

② 一个**与节点**不可解，只要其子节点至少有一个不可解。

③ 一个**或节点**不可解，当且仅当其子节点全都不可解。

* 与或图构成规则
  * 与或图中的每个节点代表一个要解决的单一问题或问题集合。起始节点对应于原始问题。
  * 对应于本原问题的节点，叫做终叶节点。
  * 对于把算符应用于问题A的每种可能情况，都把问题变换为一个子问题集合；有向弧线自A指向后继节点，表示所求得的子问题集合，这些子问题节点叫做或节点。
  * 一般对于代表两个或两个以上子问题集合的每个节点，有向弧线从此节点指向此子问题集合中的各个节点,这些子问题节点叫做与节点。

状态空间与问题归约两种表示方法有何关系？

小结

1. 掌握该类方法有利于培养解决复杂问题的能力。
2. 状态空间法是问题归约法的一种特例。在问题归约法的与或图中，包含有与节点和或节点，而在状态空间法中只含有或节点。
3. 两种方法对复杂问题可能发生组合爆炸。

知识表示

- 状态空间法
- 问题归约法
- 谓词逻辑法
- 语义网络法
- ……

谓词逻辑法

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，特别是定理的自动证明发挥了重要作用，在人工智能的发展史中占有重要地位。

命题逻辑

定义 1 命题（Proposition）：命题是具有真假意义的语句。

原子命题：如果一个命题不能进一步分解为更简单的命题，则此命题叫原子命题（或本原命题、原始命题）。

复合命题：使用适当的联结词，可以把原子命题组合成复合命题或分子命题。

定义 2 命题联接词：在数理逻辑中也有类似的严格定义过的联结词，叫命题联接词。常用的命题联接词有五个：**合取、析取、否定、蕴含、双条件（等价）**。

下面给出了五个陈述

p: 北京是中国的首都

q : 13能被6整除

r : x<8

s: 存在最大的素数

t: $m^2 \geq 0$

- 任何一个命题或为真、或为假。
- p和t是真命题，q和s是假命题。
- r的真假依赖于x的取值，无法判断r的真假，因此r 不是命题。

- **可通过命题联结词(connectives)对已有命题进行组合，得到新命题。这些通过命题联结词得到的命题被称为复合命题(compound proposition)。假设存在命题p和q，下面介绍五种主要的命题联结词：**

命题连接符号	表示形式	意义
与(and)	$p \wedge q$	命题合取(conjunction)，即“p且q”
或(or)	$p \vee q$	命题析取(disjunction)，即“p或 q”
非（not）	$\neg p$	命题否定(negation)，即“非 p ”
条件(conditional)	$p \rightarrow q$	命题蕴含(implication)，即“如果p 则 q”
双向条件(bi-conditional)	$p \leftrightarrow q$	命题双向蕴含(bi-implication)，即“p 当且仅当 q”

命题逻辑

命题连接词逻辑真值表

P　Q	~P	P∧Q	P∨Q	P→Q	P ↔ Q
F　F	T	F	F	T	T
F　T	T	F	T	T	F
T　F	F	F	T	F	F
T　T	F	T	T	T	T

逻辑等价：给定命题p和命题q，如果p和q在所有情况下都具有同样真假结果，那么p和q在逻辑上等价，一般用≡来表示，即$p \equiv q$。

逻辑等价为命题进行形式转换带来了可能，基于这些转换不再需要逐一列出p和q的真值表来判断两者是否在逻辑上等价，而是可直接根据已有逻辑等价公式来判断p和q在逻辑上是否等价。

逻辑等价的例子

$\alpha \wedge \beta \equiv \beta \wedge \alpha$ ($\wedge$的交互律)	$(\alpha \Rightarrow \beta) \equiv \neg\alpha \vee \beta$ **(蕴涵消除)**
$\alpha \vee \beta \equiv \beta \vee \alpha$ ($\vee$的交互律)	$(\alpha \Leftrightarrow \beta) \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$(双向消除)
$(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ ($\wedge$的结合律)	$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ (De Morgan)
$(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$ ($\vee$的结合律)	$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ (De Morgan)
$\neg(\neg\alpha) \equiv \alpha$ (双重否定)	$(\alpha \wedge (\beta \vee \gamma)) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ ($\wedge$对 $\vee$的分配律)
$(\alpha \Rightarrow \beta) \equiv \neg\beta \Rightarrow \neg\alpha$(逆否命题)	$(\alpha \vee (\beta \wedge \gamma)) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ ($\vee$对 $\wedge$的分配律)

命题逻辑中的推理规则

假言推理 (Modus Ponens)	$\dfrac{\alpha \Rightarrow \beta,\ \ \alpha}{\beta}$
与消解 (And-Elimination)	$\dfrac{\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n}{\alpha_i (1 \le i \le n)}$
与导入 (And-Introduction)	$\dfrac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n}$

命题逻辑中的推理规则

双重否定 (Double-Negation Elimination)	$\dfrac{\neg\neg\alpha}{\alpha}$
单项消解或单项归结 (Unit Resolution)	$\dfrac{\alpha \vee \beta, \neg\beta}{\alpha}$
消解或归结 (Resolution)	$\dfrac{\alpha\vee\beta, \neg\beta\vee\gamma}{\alpha\vee\gamma}, \dfrac{\alpha_1\vee\alpha_2\vee\cdots\vee\alpha_m, \neg\beta}{\alpha_1\vee\alpha_2\vee\cdots\vee\alpha_{k-1}\vee\alpha_{k+1}\vee\cdots\vee\alpha_m}(\neg\alpha_k = \neg\beta)$

应用归结法进行证明 (1)

1	$\alpha \vee \beta$
2	$\alpha \Rightarrow \gamma$
3	$\beta \Rightarrow \gamma$

已知如上命题成立，请证明命题γ是成立的

1	$\alpha \vee \beta$	已知
2	$\neg\alpha \vee \gamma$	2进行蕴涵消除
3	$\neg\beta \vee \gamma$	3进行蕴涵消除
4	$\neg\gamma$	假设命题γ不成立
5	$\beta \vee \gamma$	1和2进行归结
6	$\neg\alpha$	2和4进行归结
7	$\neg\beta$	3和4进行归结
8	γ	5和7进行归结
9	假设不成立，命题γ成立	

应用归结法进行证明 (2)

1	$\alpha \vee \beta$
2	$\neg\alpha \vee \beta$
3	$\alpha \vee \neg\beta$
4	$\neg\alpha \vee \neg\beta$

证明如上命题集是不可满足的

1	$\alpha \vee \beta$	已知
2	$\neg\alpha \vee \beta$	已知
3	$\alpha \vee \neg\beta$	已知
4	$\neg\alpha \vee \neg\beta$	已知
5	β	1和2进行归结
6	$\neg\beta$	3和4进行归结
因此，从该命题集中同时推出命题β和命题$\neg\beta$，因此原命题集是不可满足的		

命题逻辑的局限性

命题逻辑的这种表示法有较大的局限性，它无法把它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。

假设要表示句子：“李明是工人。”用命题逻辑表示时可写为：LIWORKER。

如果要表示：“王华也是工人。”就要写成：WANGWORKER。

谓词逻辑法

- **引入个体、量词和谓词来分析逻辑命题与推理。**
- **个体**：表示某一个物体或元素，
- **量词**：表示数量
- **谓词**：在谓词逻辑中，原子命题分解成个体词和谓词. 个体词是可以独立存在的客体，它可以是具体事物或抽象的概念。**谓词是用来刻划个体词的性质或事物之间关系的词。**
- 例如：数x是正的　　　　　　P（x）

1 谓词公式

* 原子公式的的定义：
  * 用$P(x_1, x_2, \ldots, x_n)$表示一个n元谓词公式，其中P为n元谓词，$x_1,x_2,\ldots, x_n$为客体变量或变元。通常把$P(x_1,x_2,\ldots,x_n)$叫做谓词演算的原子公式，或原子谓词公式。
* 复合谓词公式
  * 可以用连词和量词把原子谓词公式组成复合谓词公式。

* 连词和量词(Connective &Quantifiers)
  * 连词
    * 与、合取（conjunction)：∧
    * 或、析取（disjunction）：∨
    * 蕴涵（Implication）：⇒或→
    * 等价：↔
    * 非（Not）：~或¬
  * 量词
    * 全称量词（Universal Quantifiers)：∀
    * 存在量词(Existential Quantifiers)：∃

谓词逻辑：量词

- 全称量词(universal quantifier, ∀)
  - 全称量词用符号∀表示，表示一切的、凡是的、所有的、每一个等。$\forall x$表示定义域中的所有个体， $\forall x P(x)$表示定义域中的所有个体具有性质P
- 存在量词(existential quantifier, ∃)
  - 存在量词用符号∃表示，表示存在、有一个、某些等。 $\exists x$表示定义域中存在一个或若干个个体，$\exists x P(x)$表示定义域中存在一个个体或若干个体具有性质P
- 全称量词和存在量词统称为量词。

- 全称量词与存在量词之间的组合
  - $\forall x \neg P(x) \equiv \neg \exists x P(x)$
  - $\neg \forall x P(x) \equiv \exists x \neg P(x)$
  - $\forall x P(x) \equiv \neg \exists x \neg P(x)$
  - $\exists x P(x) \equiv \neg \forall x \neg P(x)$

谓词公式-合式公式

合法的表达式称为合式公式

1）原子公式是合式公式。

2）若A,B是谓词公式，则

$$\neg A, A \wedge B, A \vee B, A \rightarrow B, A \leftrightarrow B, \forall x A, \exists x A$$

也是合式公式。

3）只有有限步应用1）2）生成的公式才是是合式公式。

•谓词逻辑表示知识的步骤：

- (1) 定义个体及谓词
- (2) 引入连接词及量词表示事实和问题

练 习

- 用谓词演算公式表示下列问题
  - 凡是清洁的东西人就喜欢；
  - 人们都不喜欢苍蝇。
- 苍蝇是不清洁的。

练 习

* 用谓词演算公式表示下列问题
* 前提：
  * 凡是清洁的东西人就喜欢；
  * 人们都不喜欢苍蝇。
* 证明：苍蝇是不清洁的。

引入谓词：

clear(x):表示清洁的东西；

like(x,y): 表示人 x 喜欢 y

于是有：

$(\forall x)(\forall y)(clear(x) \rightarrow like(y,x))$

$(\forall x)(\sim like(x, fly))$

$\sim clear(fly)$

若干谓词逻辑的推理规则

- **全称量词消去(Universal Instantiation, UI):** $(\forall x)A(x) \to A(y)$
- **全称量词引入(Universal Generalization, UG):** $A(y) \to (\forall x)A(x)$
- **存在量词消去(Existential Instantiation, EI):** $(\exists x)A(x) \to A(c)$
- **存在量词引入(Existential Generalization, EG):** $A(\mathrm{c}) \to (\exists x)A(x)$

谓词逻辑的推理例子

已知：

- $(\forall x)(P(x) \rightarrow Q(x))$
- $(\forall x)(Q(x) \rightarrow R(x))$

试证明： $(\forall x)(P(x) \rightarrow R(x))$

证明过程：

- $(\forall x)(P(x) \rightarrow Q(x))$
- $P(x) \rightarrow Q(x)$ （消去全称量词）
- $(\forall x)(Q(x) \rightarrow R(x))$
- $Q(x) \rightarrow R(x)$ （消去全称量词）
- $P(x) \rightarrow R(x)$
- $(\forall x)(P(x) \rightarrow R(x))$ （引入x）

谓词逻辑的推理例子

已知：

- $(\forall x)(F(x) \to (G(x) \land H(x)))$
- $(\exists x)(F(x) \land P(x))$

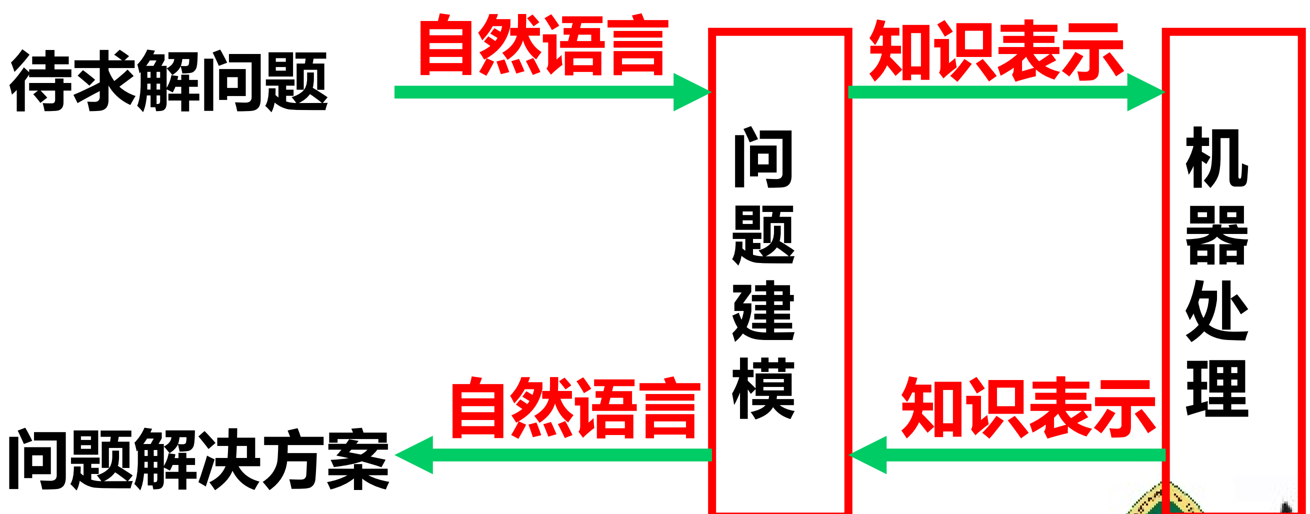
试证明：$(\exists x)\ (P(x) \land H(x))$

证明过程：

1. $(\forall x)(F(x) \to (G(x) \land H(x)))$ **(已知)**
2. $(\exists x)(F(x) \land P(x))$ **(已知)**
3. $F(a) \land P(a)$ **(2的EI)**
4. $F(a) \to (G(a) \land H(a))$ **(1的UI)**
5. $F(a)$ **(由3知)**
6. $G(a) \land H(a)$ **(4和5的假言推理)**
7. $P(a)$ **(由3知)**
8. $H(a)$ **(由6知)**
9. $P(a) \land H(a)$**(7和8的合取)**
10. $(\exists x)\ (P(x) \land H(x))$ **(9的EG)**

知识表示

- 状态空间法
- 问题归约法
- 谓词逻辑法
- 语义网络法→知识图谱
- ……

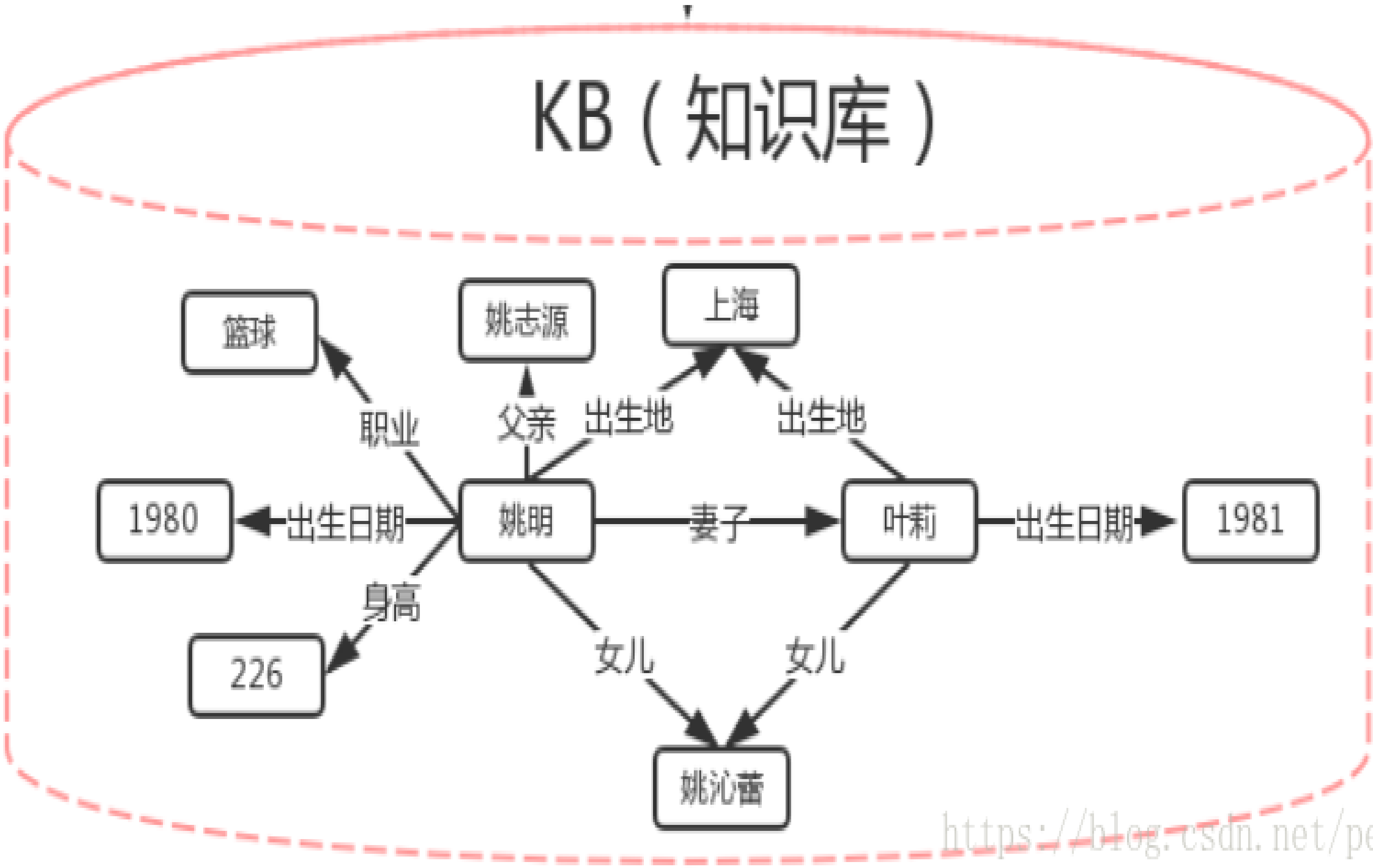


语义网络表示法
（知识图谱 Knowledge Graph）

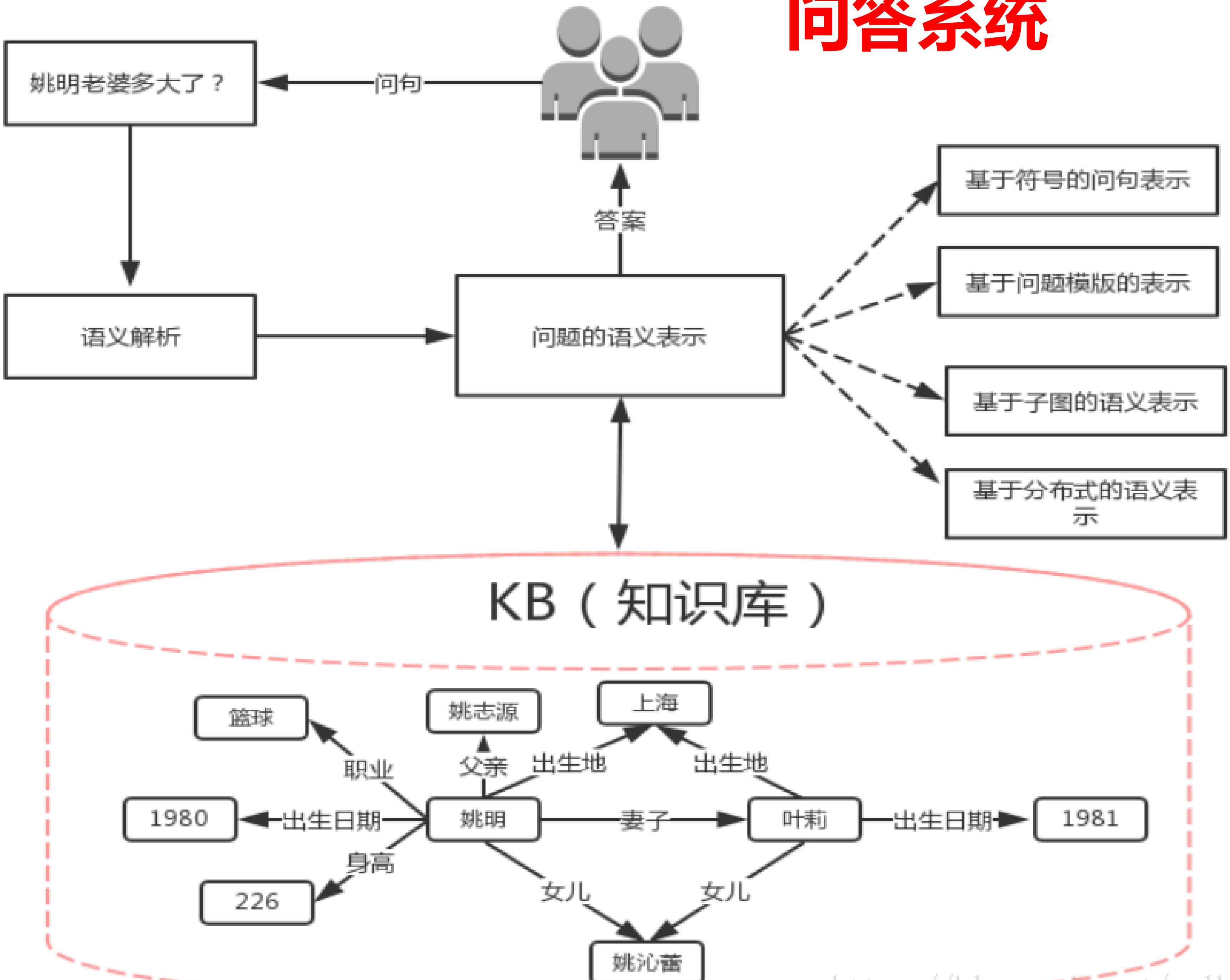
语义网络是1968年奎利恩(Quillian)在研究人类联想记忆时提出的心理学模型，认为记忆是由概念间的联系实现的。1972年西蒙斯(Simmous)首先将语义网络表示法用于**自然语言理解系统**。

- **语义网络法：**知识的一种结构化图解表示，它由节点和弧线组成。

知识的语义网络表示法（知识图谱 Knowledge Graph）



问答系统



姚明老婆多大了？
问句
语义解析
问题的语义表示
答案
基于符号的问句表示
基于问题模版的表示
基于子图的语义表示
基于分布式的语义表示
KB（知识库）
篮球
姚志源
上海
职业
父亲
出生地
出生地
1980
出生日期
姚明
妻子
叶莉
出生日期
1981
身高
226
女儿
女儿
姚沁蕾

二元语义网络的表示

* 表示实体之间关系、动作，和实体相关的事实、实体的属性/特征
  * 例： 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基本知识，以便简化表示，并可用简单的知识来表示更复杂的知识。

语义网络的基本概念

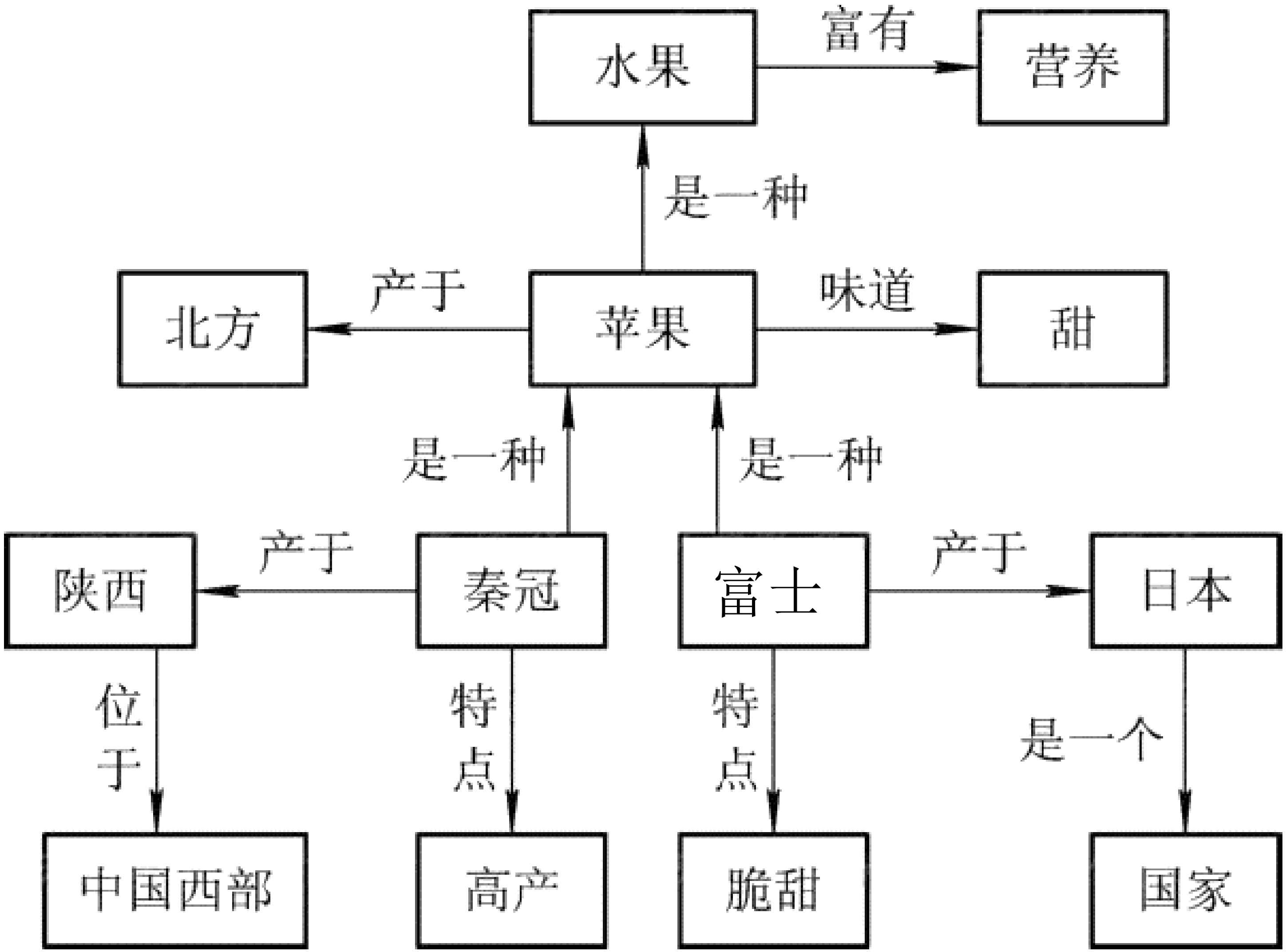
- 语义网络是对知识的有向图表示方法。一个语义网络是由一些有向图表示的三元组
- **(结点1，孤，结点2)**

- 结点表示概念、事物、事件、情况等。
- 弧是有方向的有标注的。**方向体现主次，结点1为主，结点2为副**。弧上标注表示结点1的属性或结点1和结点2之间的关系。

语义网络的例子

- 例如：雪是白色的，可表示成

语义网络


水果
富有
营养
是一种
北方
产于
苹果
味道
甜
是一种
是一种
陕西
产于
秦冠
富士
产于
日本
位于
特点
特点
是一个
中国西部
高产
脆甜
国家

语义网络的表达能力

语义网络不仅可以表示事物的属性、状态、行为等，而且适合表示事物之间的关系和联系。

而表示一个事物的层次、状态、行为，也可以看作是该事物与其属性、状态或行为的一种关系。

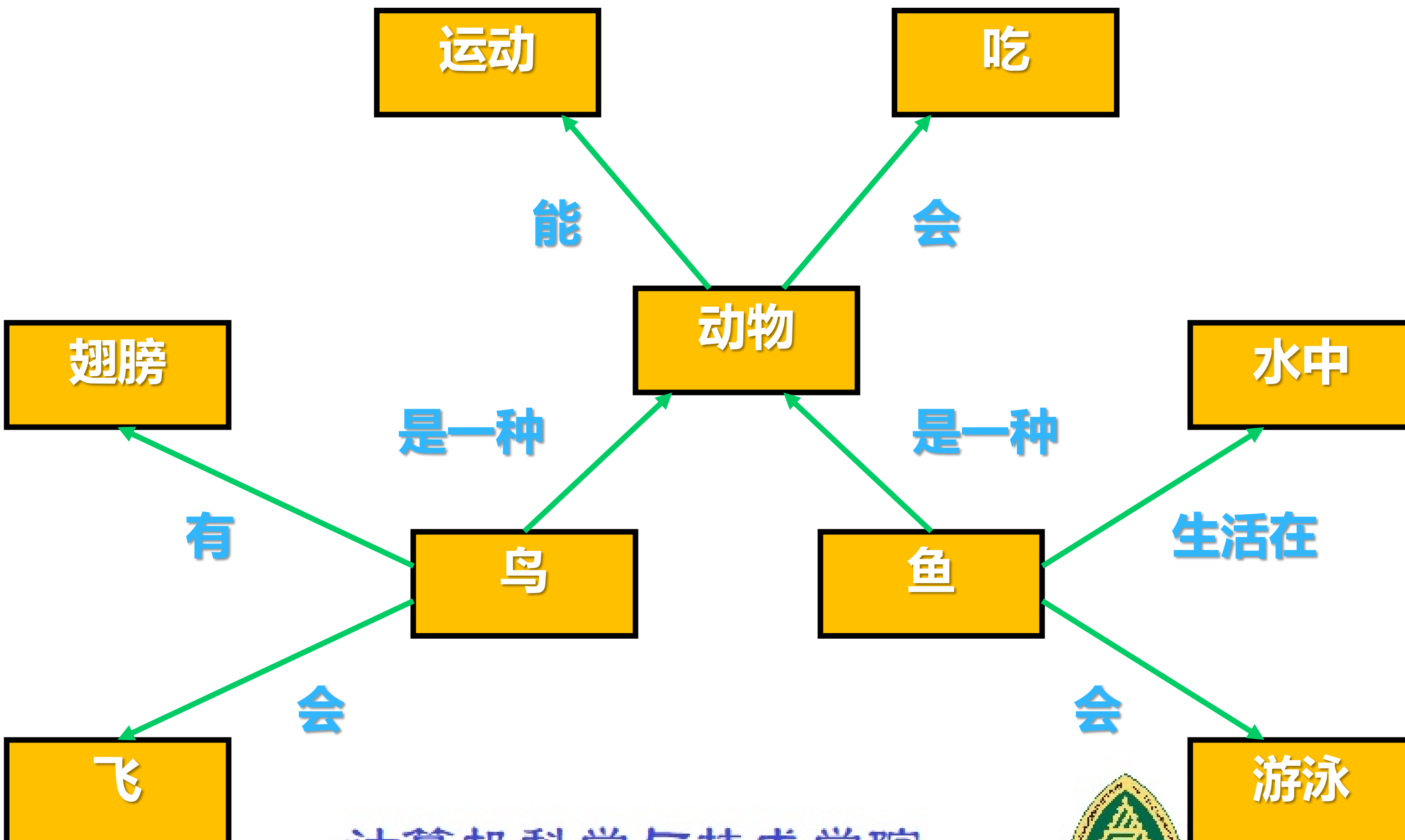
因此，**关系（或联系）型的知识、和能够化为关系型的知识都可以用语义网络来表示。**

语义网知识表示方法举例

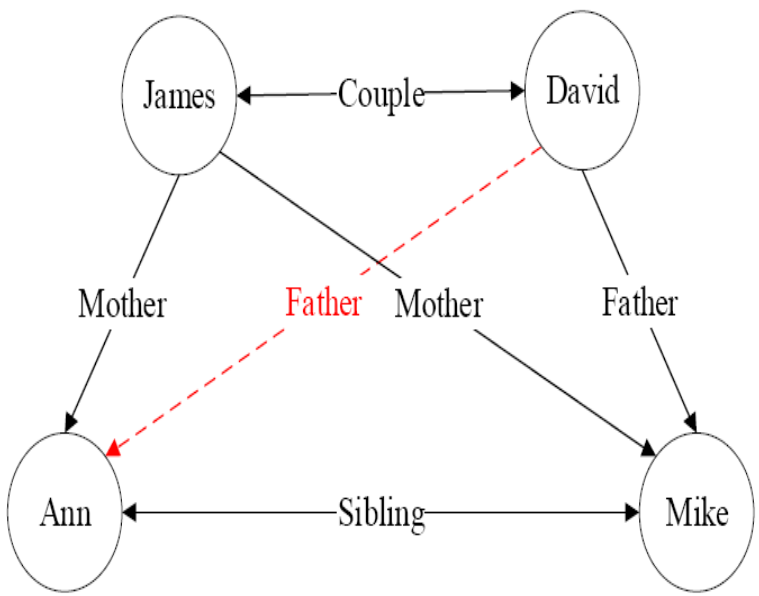
动物能运动、会吃。鸟是一种动物，鸟有翅膀、会飞。鱼是一种动物，鱼生活在水中，会游泳。

语义网络知识表示方法举例

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。



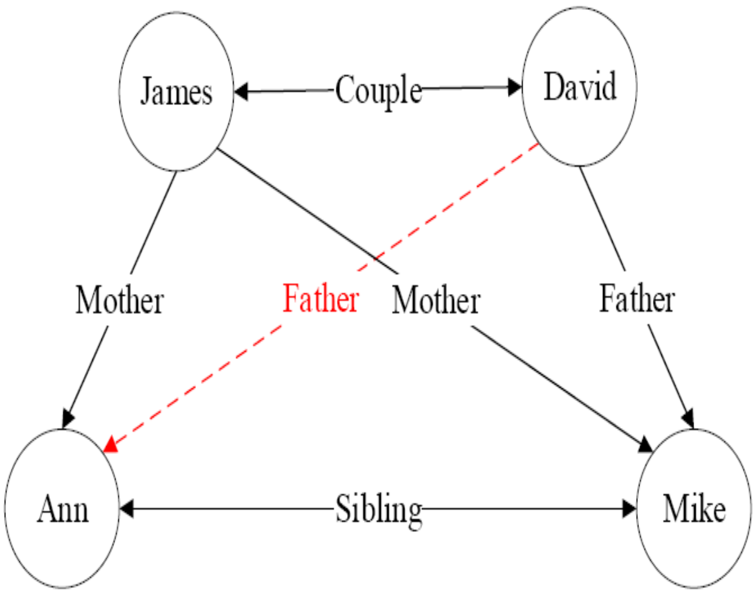
知识图谱推理



一个简单的家庭关系知识图谱

- 可利用一阶谓词来表达刻画知识图谱中节点之间存在的关系，如图中形如<**James**,*Couple*,**David**>的关系可用一阶逻辑的形式来描述，即*Couple*(**James, David**)。
- $Couple(x, y)$是一阶谓词，$Couple$是图中实体之间具有的关系，x和y是谓词变量
- 从图中已有关系可推知David和Ann具有父女关系，但这一关系在图中初始图(无红线)中并不存在，是需要推理的目标。

知识图谱推理



一个简单的家庭关系知识图谱

- 问题：如何从知识图谱中推理得到

$\underline{father(David, Ann)}$

$$(\forall x)(\forall y)(\forall z)\big(Mother(z,y) \wedge Couple(x,z) \rightarrow Father(x,y)\big)$$

如果能够学习得到这条规则，该有多好？

（从具体例子中学习，这是归纳推理的范畴）

知识图谱推理：归纳学习

归纳逻辑程序设计 (inductive logic programming，ILP)算法

归纳逻辑程序设计（ILP）是机器学习和逻辑程序设计交叉领域的研究内容。

ILP使用一阶谓词逻辑进行知识表示，通过修改和扩充逻辑表达式对现有知识归纳，完成推理任务。

作为ILP的代表性方法，FOIL（First Order Inductive Learner）通过序贯覆盖实现规则推理。

知识图谱推理：FOIL (First Order Inductive Learner)

$$(\forall x)(\forall y)(\forall z)\big(\underline{Mother(z,y) \wedge Couple(x,z)} \rightarrow \underline{Father(x,y)}\big)$$

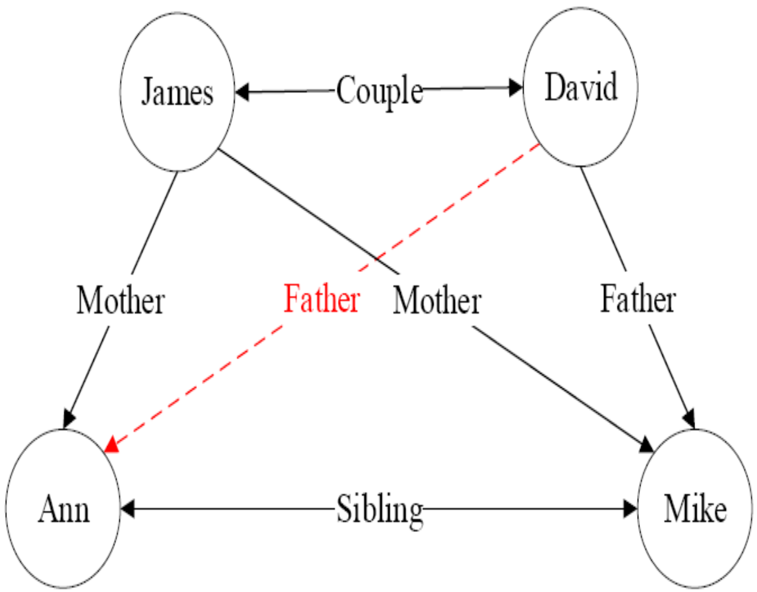
$\Uparrow$ 前提约束谓词（学习得到）

$\Uparrow$ 目标谓词（已知）

推理手段： ***positive examples*** + ***negative examples*** + ***background knowledge examples*** $\Longrightarrow$ **hypothesis**

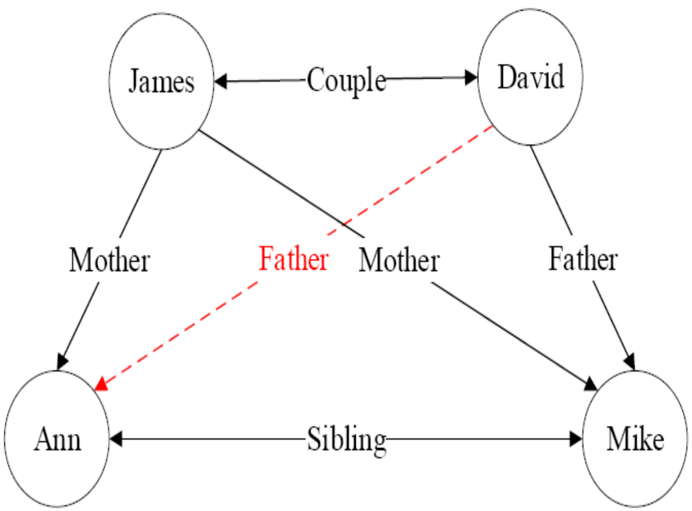
知识图谱推理：FOIL (First Order Inductive Learner)

- **目标谓词：** $Father(x, y)$
- 目标谓词只有一个正例$Father$**(David, Mike)**。
- 反例在知识图谱中一般不会显式给出，但可从知识图谱中构造出来。如从知识图谱中已经知道$Couple$**(David, James)**成立，则$Father$**(David, James)**可作为目标谓词P的一个反例，记为$\neg Father$ **(David, James)**。
- 只能在已知两个实体的关系且确定其关系与目标谓词相悖时，才能将这两个实体用于构建目标谓词的反例，而不能在不知两个实体是否满足目标谓词前提下将它们来构造目标谓词的反例。



一个简单的家庭关系知识图谱

知识图谱推理：FOIL (First Order Inductive Learner)



一个简单的家庭关系知识图谱

- **目标谓词：**$Father(x, y)$
- **背景知识：知识图谱中目标谓词以外的其他谓词实例化结果，如** $Sibling$**(Ann, Mike)**

知识图谱推理：FOIL (First Order Inductive Learner)

$$(\forall x)(\forall y)(\forall z)\big(Mother(z,y) \wedge Couple(x,z) \rightarrow \boxed{Father(x,y)}\big)$$

前提约束谓词
（学习得到）

目标谓词
（已知）

推理思路：从一般到特殊，逐步给目标谓词添加前提约束谓词，直到所构成的推理规则不覆盖任何反例。

从一般到特殊：对目标谓词或前提约束谓词中的变量赋予具体值，如将$(\forall x)(\forall y)(\forall z)\big(Mother(z,y) \wedge Couple(x,z) \rightarrow Father(x,y)\big)$这一推理规则所包含的目标谓词$Father(x,y)$中$x$和$y$分别赋值为David和Ann，进而进行推理。

知识图谱推理：FOIL (First Order Inductive Learner)

$$(\forall x)(\forall y)(\forall z)\big(Mother(z,y) \wedge Couple(x,z) \rightarrow \boxed{Father(x,y)}\big)$$

前提约束谓词
（学习得到）

目标谓词
（已知）

哪些谓词好呢？可以作为目标谓词的前提约束谓词？

FOIL信息增益值计算方法如下：

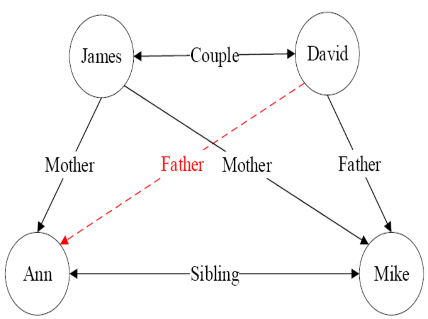
$$FOIL\_Gain = \widehat{m_+} \cdot \left(\log_2 \frac{\widehat{m_+}}{\widehat{m_+} + \widehat{m_-}} - \log_2 \frac{m_+}{m_+ + m_-}\right)$$

其中，$\widehat{m_+}$和$\widehat{m_-}$是增加前提约束谓词后所得新推理规则覆盖的正例和反例的数量，m_+和m_-是原推理规则所覆盖的正例和反例数量。

FOIL中信息增益值 (information gain)

知识图谱推理：FOIL (First Order Inductive Learner)

$$(\forall x)(\forall y)(\forall z)\big(Mother(z,y) \wedge Couple(x,z) \rightarrow \boxed{Father(x,y)}\big)$$



前提约束谓词
（学习得到）

目标谓词
（已知）

- $Mother(\cdot,\cdot)$
- $Sibling(\cdot,\cdot)$
- $Couple(\cdot,\cdot)$

依次将谓词加入到推理规则中作为前提约束谓词，并计算所得到新推理规则的FOIL增益值。

基于计算所得FOIL增益值来选择最佳前提约束谓词。

知识图谱推理：FOIL (First Order Inductive Learner)

$$(\forall x)(\forall y)(\forall z)\left(Mother(z,y) \wedge Couple(x,z) \rightarrow \boxed{Father(x,y)}\right)$$

前提约束谓词
（学习得到）

目标谓词
（已知）

背景知识样例集合	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)	目标谓词训练样例集合	Father(David, Mike) ¬Father(David, James) ¬Father(James, Ann) ¬Father(James, Mike) ¬Father(Ann, Mike)

知识图谱推理：FOIL (First Order Inductive Learner)

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x,y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x,y) \leftarrow$	$Mother(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 3$	0.32
	$Sibling(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 2$	0.74
	$Couple(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$\boldsymbol{Couple(x,z)}$	$\boldsymbol{\widehat{m_+} = 1}$	$\boldsymbol{\widehat{m_-} = 1}$	1.32
	$Couple(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Couple(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Couple(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Couple(z,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA

给定目标谓词，此时推理规则只有目标谓词，因此推理规则所覆盖的正例和反例的样本数分别是训练样本中正例和反例的数量，即1和4，因此，$m_+ = 1$，$m_- = 4$。

知识图谱推理：FOIL (First Order Inductive Learner)

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x,y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x,y) \leftarrow$	$Mother(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 3$	0.32
	$Sibling(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 2$	0.74
	$Couple(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$\boldsymbol{Couple(x,z)}$	$\boldsymbol{\widehat{m_+} = 1}$	$\boldsymbol{\widehat{m_-} = 1}$	1.32
	$Couple(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} =0$	NA
	$Couple(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Couple(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Couple(z,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA

- 将$Mother(x,y)$作为前提约束谓词加入，可得到推理规则 $Mother(x,y) \rightarrow Father(x,y)$

- 在背景知识中，$Mother(x,y)$有两个实例 $Mother$(James, Ann) $Mother$(James, Mike)

- 对于$Mother$(James, Ann)这一实例， x = James， y = Ann，将x和y代入$Father(x,y)$得到$Father$(James, Ann)，可知在训练样本中$Father$(James, Ann)是一个反例

知识图谱推理：FOIL (First Order Inductive Learner)

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x,y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x,y) \leftarrow$	$Mother(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 3$	0.32
	$Sibling(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 2$	0.74
	$Couple(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$\boldsymbol{Couple(x,z)}$	$\boldsymbol{\widehat{m_+} = 1}$	$\boldsymbol{\widehat{m_-} = 1}$	1.32
	$Couple(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Couple(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Couple(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Couple(z,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA

将$Mother(x,y)$作为前提约束谓词加入，可得到推理规则 $Mother(x,y) \rightarrow Father(x,y)$

在背景知识中，$Mother(x,y)$有两个实例
$Mother$(James, Ann)
$Mother$(James, Mike)

对于$Mother$(James, Mike)这一实例，x= James，y = Mike，将x和y代入$Father(x,y)$得到$Father$(James, Mike)，可知在训练样本中$Father$(James, Mike)是一个反例

知识图谱推理：FOIL (First Order Inductive Learner)

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x,y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x,y) \leftarrow$	$Mother(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 3$	0.32
	$Sibling(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 2$	0.74
	$Couple(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$\boldsymbol{Couple}(\boldsymbol{x},\boldsymbol{z})$	$\widehat{\boldsymbol{m}_+} = \mathbf{1}$	$\widehat{\boldsymbol{m}_-} = \mathbf{1}$	1.32
	$Couple(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} =0$	NA
	$Couple(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Couple(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Couple(z,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA

$Mother(x,y) \rightarrow Father(x,y)$

覆盖正例和反例数量分别为0和2，即$\widehat{m_+} = 0$，$\widehat{m_-} = 2$

由于$\widehat{m_+} = 0$，代入$FOIL_Gain$公式时会出现负无穷的情况，此时$FOIL_Gain$记为NA（Not Available）

知识图谱推理：FOIL (First Order Inductive Learner)

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x,y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x,y) \leftarrow$	$Mother(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 3$	0.32
	$Sibling(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 2$	0.74
	$Couple(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$\boldsymbol{Couple(x,z)}$	$\boldsymbol{\widehat{m_+} = 1}$	$\boldsymbol{\widehat{m_-} = 1}$	1.32
	$Couple(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Couple(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Couple(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Couple(z,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA

如果将$Couple(x,z)$作为前提约束谓词加入，可得到如下推理规则 $Couple(x,z) \rightarrow Father(x,y)$

在背景知识中，$Couple(x,z)$只有一个实例$Couple$(David, James)，即x=David，z= James，将其代入$Father(x,y)$得到$Father$(David, y)。

知识图谱推理：FOIL (First Order Inductive Learner)

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
目标谓词	前提约束谓词	正例	反例	信息增益值
$Father(x,y) \leftarrow$	空集	$m_+ = 1$	$m_- = 4$	$FOIL\_Gain$
$Father(x,y) \leftarrow$	$Mother(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Mother(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Mother(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 3$	0.32
	$Sibling(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$Sibling(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Sibling(z,y)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 2$	0.74
	$Couple(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$\boldsymbol{Couple(x,z)}$	$\boldsymbol{\widehat{m_+} = 1}$	$\boldsymbol{\widehat{m_-} = 1}$	1.32
	$Couple(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Couple(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$Couple(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 2$	NA
	$Couple(z,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA

在训练样本中存在正例$Father$(David, Mike)以及反例$\neg Father$(David, James)，即$Couple(x,z) \rightarrow Father(x,y)$覆盖正例和反例数量分别为1和1。信息增益值为：

$$\widehat{m_+} \cdot \left(\log_2 \frac{\widehat{m_+}}{\widehat{m_+} + \widehat{m_-}} - \log_2 \frac{m_+}{m_+ + m_-}\right)$$

$$= 1 \cdot \left(\log_2 \frac{1}{1+1} - \log_2 \frac{1}{1+4}\right) = 1.32$$

知识图谱推理：FOIL (First Order Inductive Learner)

Back-ground knowledge	Sibling(Ann, Mike) Couple(David, James) Mother(James, Ann) Mother(James, Mike)
Positive and negative samples	Father(David, Mike) ¬Father(David, James) ~~¬Father(James, Ann)~~ ~~¬Father(James, Mike)~~ ~~¬Father(Ann, Mike)~~

- $Couple(x, z)$ **加入后信息增益最大**
- **将** $Couple(x, z)$ **加入推理规则，得到** $Couple(x, z) \rightarrow Father(x, y)$**新推理规则**
- **将训练样例中与该推理规则不符的样例去掉。这里不符指当**$Couple(x, z)$**中**x**取值为David时，与**$Father$**(David,)或**$\neg Father$**(David,)无法匹配的实例。**
- **训练样本集中只有正例**$Father$**(David, Mike)和负例**$\neg Father$**(David, James)两个实例**

知识图谱推理：FOIL (First Order Inductive Learner)

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
现有规则	拟加入前提约束谓词	正例	反例	信息增益值
$Father(x,y) \leftarrow Couple(x,z)$		$m_+ = 1$	$m_- = 1$	1.32
$Father(x,y) \leftarrow Couple(x,z)$	$\wedge\, Mother(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Mother(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Mother(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Mother(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Mother(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\boldsymbol{\wedge\, Mother(z,y)}$	$\boldsymbol{\widehat{m_+} = 1}$	$\boldsymbol{\widehat{m_-} = 0}$	**1**
	$\wedge\, Sibling(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Sibling(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Sibling(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Sibling(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Sibling(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Sibling(z,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Couple(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$\wedge\, Couple(x,z)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 1$	0
	$\wedge\, Couple(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} =0$	NA
	$\wedge\, Couple(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Couple(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\, Couple(z,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA

- $Mother(z,y)$**加入信息增益最大**
- **将**$Mother(z,y)$**加入，得到新推理规则**

$$\boldsymbol{Mother(z,y) \wedge Couple(x,z) \rightarrow Father(x,y)}$$

- **当 x=David 、 y=Mike 、 z =James时，该推理规则覆盖训练样本集合中正例** $Father$**(David, Mike)且不覆盖任意反例，因此算法学习结束。**

知识图谱推理：FOIL (First Order Inductive Learner)

推理规则		推理规则涵盖的正例和反例数		FOIL信息增益值
现有规则	拟加入前提约束谓词	正例	反例	信息增益值
$Father(x,y) \longleftarrow Couple(x,z)$		$m_+ = 1$	$m_- = 1$	1.32
$Father(x,y) \longleftarrow Couple(x,z)$	$\wedge\ Mother(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Mother(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Mother(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Mother(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Mother(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ \boldsymbol{Mother(z,y)}$	$\boldsymbol{\widehat{m_+} = 1}$	$\boldsymbol{\widehat{m_-} = 0}$	**1**
	$\wedge\ Sibling(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Sibling(x,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Sibling(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Sibling(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Sibling(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Sibling(z,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Couple(x,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 1$	NA
	$\wedge\ Couple(x,z)$	$\widehat{m_+} = 1$	$\widehat{m_-} = 1$	0
	$\wedge\ Couple(y,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Couple(y,z)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Couple(z,x)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA
	$\wedge\ Couple(z,y)$	$\widehat{m_+} = 0$	$\widehat{m_-} = 0$	NA

$$\boldsymbol{Mother(z,y) \wedge Couple(x,z) \rightarrow Father(x,y)}$$

已知：

$Mother$**(James, Ann)**

$Couple$**(David,James)**

于是： $Father$**(David, Ann)**

知识图谱推理：FOIL (First Order Inductive Learner)

$$(\forall x)(\forall y)(\forall z)\big(Mother(z,y) \wedge Couple(x,z) \rightarrow Father(x,y)\big)$$

前提约束谓词
（学习得到）

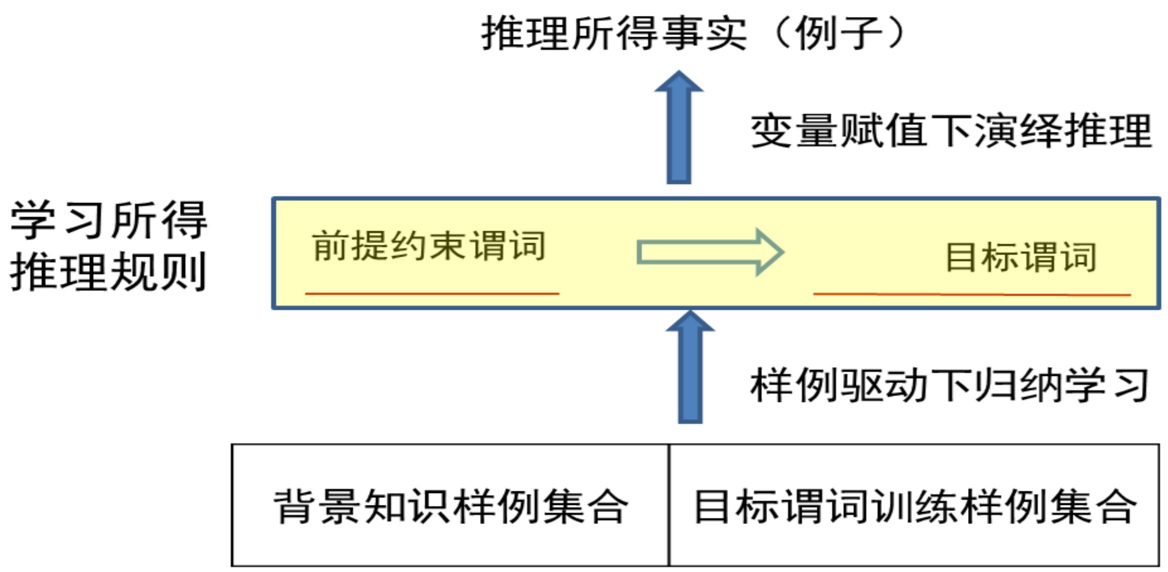
目标谓词
（已知）

推理手段： *positive examples* + *negative examples* + *background knowledge examples* ⟹ **hypothesis**

背景知识样例集合	Sibling (Ann, Mike) Couple (David, James) Mother (James, Ann) Mother (James, Mike)	目标谓词训练样例集合	Father(David, Mike) ¬Father(David, James) ¬Father(James, Ann) ¬Father(James, Mike) ¬Father(Ann, Mike)

给定目标谓词，FOIL算法从实例（正例、反例、背景样例）出发，不断测试所得到推理规则是否还包含反例，一旦不包含负例，则学习结束，展示了“归纳学习”能力。

知识图谱推理：FOIL (First Order Inductive Learner)



给定目标谓词，FOIL算法从实例（正例、反例、背景知识样例）出发，不断测试所得推理规则是否还包含反例，一旦不包含，则学习结束，由此充分展示了“归纳学习”的能力。在学得推理规则后，再给推理规则中的变量赋予具体例子，经过“演绎”得到新的知识

人工智能（功能模拟）基本问题

- 知识表示
- 知识获取
- 知识应用

* 20世纪70年代，由于研究和建造专家系统的需要而出现了知识工程（Knowledge Engineering）。
* 它主要关注了知识的表示和知识的演绎推理（由知识生成智能策略）的问题，至于如何获取专家系统所需要的知识（即知识的获取问题），则几乎完全依靠专家系统设计者的手工操作。

知识发现
知识获取
知识演绎

* 基于数据库的<u>**知识发现**</u>(KDD)和**数据挖掘**还存在着混淆。

* KDD表示将低层数据转换为高层知识的整个过程。

* 机器学习/数据挖掘一般可认为是观察数据中模式或模型的抽取。数据挖掘是知识发现过程的核心，但它通常仅占KDD的一部分(大约是15％ 到25％)。

人工智能 （功能模拟）
基本问题
知识工程三要素

- 知识表示
- 知识获取（知识发现）主流技术：数据挖掘/机器学习
- 知识应用（逻辑推理；搜索求解）