第三章 知识与知识表示方式

中科大 自动化系　郑志刚

2018.9

知识表示方法

- 概述
- 表示方法

知识的定义

- 知识的定义（难以给出明确的定义只能从不同侧面加以理解）
  - Feigenbaum：知识是经过消减、塑造、解释和转换的信息。
  - Bernstein：知识是由特定领域的描述、关系和过程组成的。
  - Hayes-roth：知识是事实、信念和启发式规则。
  - 从知识库的观点：知识是某领域中所涉及的各有关方面的一种符号表示。

知识的定义

- 从信息、数据出发给出的定义
  - 信息：物质的性质、物质物质之间的一种相互作用
    - 例如：年龄，高度，关系
  - 数据：信息的表示和载体
    - 可以量化的信息：真正的数据（温度，压力，年龄）
    - 不可以量化的信息：编码形成数据（汉字的内码）
  - 信息和数据的关系：
    - 信息是内容、实质；数据是信息的一种表示
  - 知识：
    - 有关联的信息的信息结构；例如：如果…则…
    - 人类在实践过程中对客观世界的认识与经验
    - AI系统中：事实、规则和策略都为知识

知识的特性

- 相对正确性：在一定的范围、条件和环境下是正确的
  - 例子
- 不确定性（非精确性）
  - 随机性：投币的例子
  - 模糊性：高矮的例子
  - 不完全性
  - 经验性

知识的描述

- 知识可以从范围、目的和有效性加以三维描述：
  - 知识的范围：由具体到一般；
  - 知识的目的：由说明到指定；
  - 知识的有效性：由确定到不确定；
  - 例如：
    - “为证明A->B，只需要证明A^~B是不可满足的”这种知识为一般性、指定性、确定性知识；
    - 而“桌子有四条腿”这种知识则为具体的、说明性，不确定性的；

知识的分类

- 范围
  - 常识性知识
  - 领域性知识
- 作用和表示
  - 事实性知识
    - 有关问题环境一些事实的知识
    - 形式：“…是…”
    - 关于事物的分类、属性、关系、事实
  - 过程性知识
    - 用于描述事物动作之因果关系的动态的知识
    - 形式：“如果…那么…”
  - 控制性知识
    - 如何使用过程性知识的知识
    - 表现为求解步骤和技巧

知识的分类

- 确定性
  - 精确性知识
  - 非精确性知识
- 结构与表现形式
  - 逻辑性知识
  - 形象性知识
- 知识的层次
  - 0级知识
  - 1级知识
  - 2级知识

知识的要素

- 事实：事物的分类、属性、事物间关系、科学事实、客观事实等。（最低层的知识）
- 规则：事物的行动、动作和联系的因果关系知识。（启发式规则）。
- 控制：当有多个动作同时被激活时，选择哪一个动作来执行的知识。（技巧性）
- 元知识：高层知识。怎样实用规则、解释规则、校验规则、解释程序结构等知识。

知识在AI问题求解中的作用

- 问题求解的前提
  - 前提知识存储在产生式系统的全局数据库和规则库中，它们是问题求解的依据，如果前提知识不充分，得到的解答就可能不完全
- 控制问题（搜索）的进程
  - 推理方法固然重要，但问题求解中的启发知识更重要；

讨论的主要问题

- “知识的不完全性”是认识论学派讨论最多的情况。推理者的知识是不完全，但却是一致的，其要点是在保持知识一致性的前提下得出新的结论。
- “知识不一致性”是常识的另一类性质。例：教友派教徒是和平主义者，共和党是好战分子。已知某教授是教友派教徒，且是共和党人。问他是和平主义者吗？
- “知识不确定性”是更复杂的常识问题。尽管Fuzzy、可信度理论、人工神经网络等丰富了对常识的不确定性研究方法。但还不能显现地表示“可废弃性”这个重要特征。大大限制了对智能行为“灵活性”的描述。因此，在复杂问题求解时，集成几种方法是有吸引力的想法。
- “常识的相对性”，目前在AI中研究甚少。如，理论集合是有限的，常识的集合是无限的。

表示观 —知识工程表示观

- 最常用的表示法都反映了知识工程表示观。
- 特点
  - 一般将表示理解为一类数据结构（逻辑）及在其上的操作。
  - 对知识的内容更强调与领域相关的，哪些是适合于这个领域的，来自领域专家经验知识。
  - 强调工程实现性。

表示观 —总结

结论：

无论持何种表示观的AI研究者都认为，表示是刻画智能行为的理论。

表示无论采用什么样的方式（包括数学的或程序的）所建立的表示方法和立足于什么样的表示观，均需要满足与智能现象一致的条件。

鉴于智能现象的复杂性，采用什么表示观，应当取决于所面临的问题。笼统地强调好的是没有什么意义的。

近几年一些研究者主张各种表示观应该互相渗透。

概述

- 知识表示：
  - 如何将已获得的有关知识以计算机内部代码形式加以合理地描述、存储，以使有效地利用这些知识便是知识表示；
  - 人工智能问题的求解是以知识为基础的；
  - 知识表示是人工智能研究中最基本的问题之一；
  - 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，既要考虑知识的存储又考虑知识的使用。
  - 知识表示可看作是一种描述事物的约定，以把人类知识表示成机器能处理的数据结构

知识表示问题

- AI系统要具备知识，首先遇到的问题就是知识表示问题
- 采用合适的描述手段（机器中的结构）描述知识
- 好的问题表示方法，不仅可以表示问题，而且对问题求解带来方便

定义

- 如果将AI问题到产生式系统“（GDB，RB，CS）”的映射统称为问题的表达，那么，问题域知识到全局数据库GDB和规则RB等结构上的映射，便是知识的表示问题。知识的表示是建立知识（库）系统要解决的首要问题。

知识表示主要涉及以下两个方面

- 语法（数据）结构：即知识存储和访问的形式。
- 语义（解释）过程：它给知识结构赋以含义。

知识表示问题（续）

- 知识表示研究用机器表示知识的可行性、有效性的一般方法。
- 知识表示是理智推理的部分理论。
- 知识表示是有效计算的载体
- 知识表示是交流的媒介（如语义网络）

选取知识表示的因素

- 表示范围是否广泛
- 是否适于推理
- 是否适于计算机处理
- 是否有高效的算法
- 能否表示不精确知识
- 能否模块化

- 总之 ……

- 知识和元知识能否用统一的形式表示
- 是否加入启发信息
- 过程性表示还是说明性表示
- 表示方法是否自然

选取知识表示的因素（续）

- 选取知识表示的因素

  ……

  - 总之，人工智能问题的求解是以知识表示为基础的。如何将已获得的有关知识以计算机内部代码形式加以合理地描述、存储、有效地利用便是表示应解决的问题。

好的知识表示方式的特点

- 充分表达
- 便于推理和运用
- 便于引入启发式信息，提高推理效率
- 便于知识获取

知识表示方式的评价准则

- 表示范围和准确性：能够正确反映领域知识，又可表达多种类型知识；
- 模块性和可理解性：模块化、便于修改、易于理解；
- 访问效率：知识库的组织形式能够有效地利用所表达的知识；
- 可扩充性：能够方便、灵活的对所表达的知识进行扩充；

知识表示研究的特点

- 知识表示研究的特点
  - 智能行为特有的灵活性。“常识问题”不能概括为一类简洁的理论，是大量小理论的集合。
  - AI的任务受到计算装置的约束。这导致了所采用的“表示”必须同时满足“刻画智能现象”与“计算装置可以接受”，这两个有时是矛盾的条件。

知识表示方法

- 概述
- 表示方法

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法

- 框架表示法
- 脚本方法
- 过程表示

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法

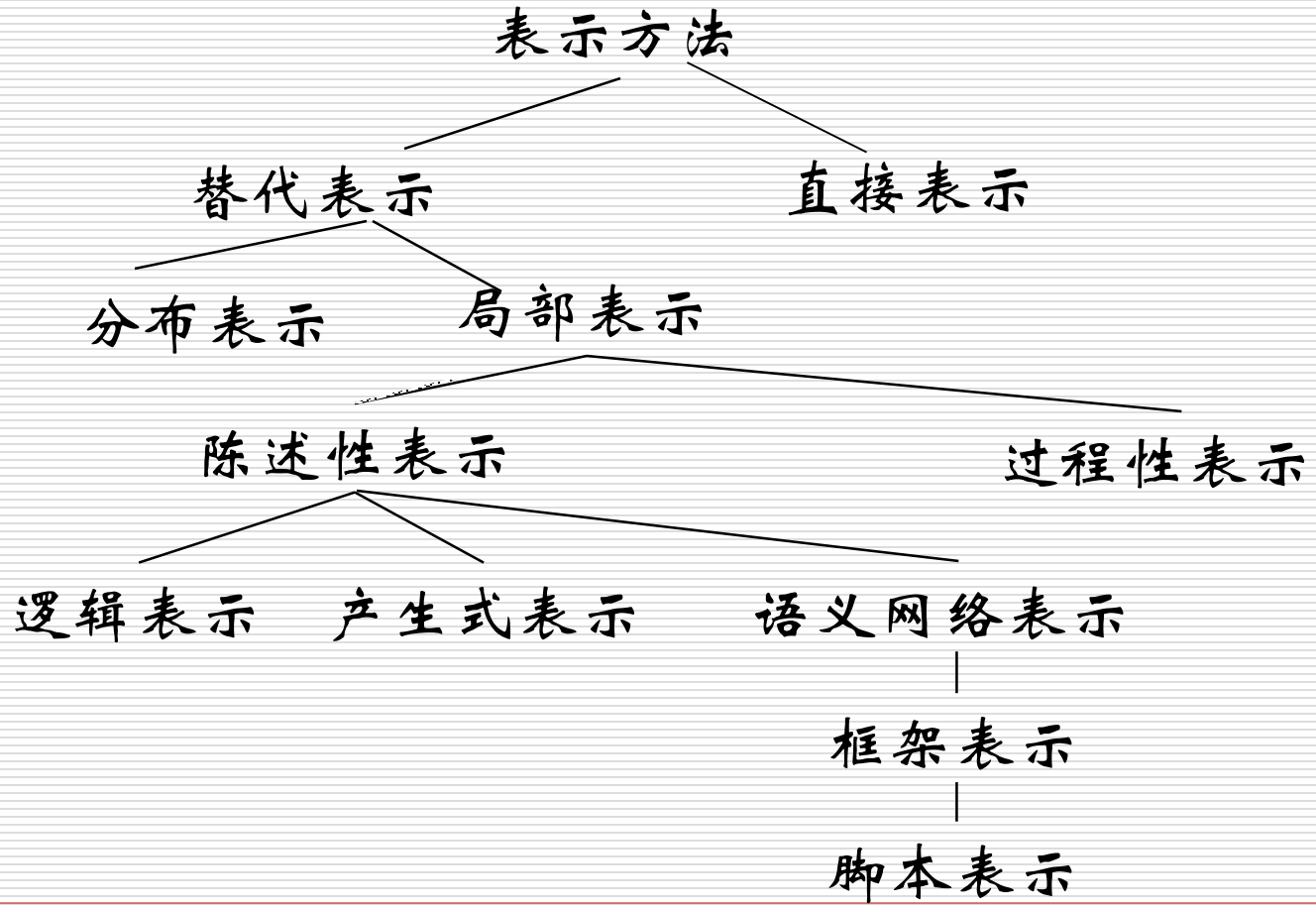
- 框架表示法
- 脚本方法
- 过程表示

表示方法 —概述

- 表示方法可以分成2类
  - 替代表示法
    - 局部表示类：最充分也是正统AI最经常使用的
    - 分布表示法：对局部表示法在智能行为表述尚不够充分而作的补充。
  - 直接表示法：

    正在引起越来越多AI研究者的注意。（不可完全独立：考虑到“任何表示方法必须被计算机所接受” 这个先决条件，直接表示需要借助局部或部分表示形式。）

表示方法 —概述


表示方法
替代表示
直接表示
分布表示
局部表示
陈述性表示
过程性表示
逻辑表示
产生式表示
语义网络表示
框架表示
脚本表示

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法

- 框架表示法
- 脚本方法
- 过程表示

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法

- 框架表示法
- 脚本方法
- 过程表示

表示方法 —直接表示

- 1963年由Gelernter提出的。用于基于传统欧氏几何证明的几何定理证明器。
- 它的输入是对前提和目标的陈述以及图示（图示是用一系列坐标来表示的）。
- 在证明过程中，证明器把图示作为启发式信息，排除在图示中不正确的子目标。从而大大地减少了搜索空间。
- 但……..

- 但，长期以来直接表示没有得到长足发展。原因如下：
  - 计算机对直接表示的信息难以处理。
  - 直接表示难以表示定量信息（语言设计失败）
  - 直接表示不能描述自然世界的全部信息
- 这两年直接表示有所发展，因为，现在认识到，可以用其它媒体表示的方法去补充直接表示的不足。——将被发展成多媒体。
- 引申的研究是临场AI与临境技术。近几年AI对自主智能系统研究（完全机器做人不干预）的失望，导致对建立人机一体智能系统的尝试。这样系统所需环境的要求是直接表示兴起的原因之一。

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法

- 框架表示法
- 脚本方法
- 过程表示

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法

- 框架表示法
- 脚本方法
- 过程表示

表示方法 —逻辑表示法

- 将人的自然语言转换成计算机中的数据，是一件非常困难的事情
- 但是又很需要，例如：信息的智能检索、机器人问题求解、数学定理证明等
- 一阶谓词逻辑表示法：一个很好的过渡
  - 能够表达很多自然语句
  - 又由于它的严密性容易被计算机实现

表示方法 —逻辑表示法

- 事实性知识：
  - 简单的采用原子谓词公式表示
  - 复杂的采用谓词公式的与或形表示
  - 常用于表示问题的状态
- 过程性知识
  - 蕴涵式表示
  - 常用于表示问题的操作

表示方法 —逻辑表示法

- 一阶谓词逻辑是谓词逻辑中最直观的一种逻辑。它以谓词形式来表示动作的主题、客体。客体可以多个。

  如：张三与李四打网球（Zhang and Li play tennis），可写为：play (Zhang, Li, tennis)

  这里谓词是play，动词主体是Zhang和 Li，而客体是tennis。

- 谓词逻辑规范表达式：

  P（ x_1, x_2, x_3, …），这里P是谓词，x_i是主体与客体。

表示方法 —逻辑表示法

- 谓词比命题更加细致地刻画知识：
  - 表达能力强
    - 如：北京是个城市， City(x)

      把城市这个概念分割出来。把“城市” 与“北京”两个概念连接在一起，而且说明“北京”是“城市”的子概念。（有层）
  - 谓词可以代表变化的情况
    - 如：City(北京),真。 City(煤球)，假
  - 在不同的知识之间建立联系…….

表示方法 —逻辑表示法

- 在不同的知识之间建立联系
  - 如：Human(x) → Lawed(x)， 人人都受法律管制，x是同一个人。

Commit(x) → Punished(x)， x不一定是人也可以是动物。

而，{[Human(x) → Lawed(x)]→[commit(x) → Punished(x)]}，

意为如果由于某个x是人而受法律管制，则这个人犯了罪就一定要受到惩罚。

使用谓词公式表示知识的过程

- 定义谓词
- 使用连接词连接谓词，构成谓词公式

一阶谓词表示法的例子

- 例子3.1
  - 知识：
    - 刘欢比他的父亲出名
    - 高阳是计算机系的一名学生，但是他不喜欢编程序
    - 人人爱劳动
  - 谓词的定义
    - BIGGER(x,y)：x比y出名
    - COMPUTER(x)：x为计算机系的学生
    - LIKE(x,y)：x喜欢y
    - LOVE(x,y)：x爱y
    - MAN(x)：x是人
  - BIGGER(liuhuan,father(liuhuan))
  - COMPUTER(Gaoyang) ∧¬LIKE(Gaoyang,programing)
  - (∀x)(MAN(x)) →LOVE(x,labour)

一阶谓词表示法的例子

- 例3.2
  - 自然数都是大于0的整数
  - 所有整数不是偶数就是奇数
  - 偶数除以2都是整数

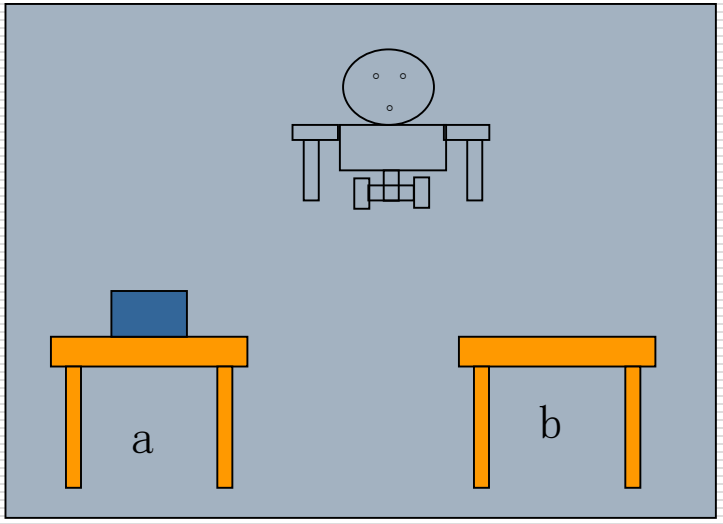
- 谓词的定义
  - N(x)： x是自然数
  - I(x) ： x是整数
  - E(x) ： x是偶数
  - O(x) ： x是奇数
  - GZ(x) ： x大于0

$$(\forall x)(N(x) \rightarrow GZ(x) \wedge I(x))$$

$$(\forall x)(I(x) \rightarrow E(x) \vee O(x))$$

$$(\forall x)(E(x) \rightarrow I(s(x)))$$

- 例3.3
  - 机器人行动规划
  - 谓词定义
    - TABLE(x)
    - EMPTY(y)
    - AT(y, z)
    - HOLDS(y, w)
    - ON(w, x)



一阶谓词表示法的例子

- 问题的初始状态
  - AT(robot, c)
  - EMPTY(robot)
  - ON(box, a)
  - TABLE(a)
  - TABLE(b)
- 问题的目标状态
  - AT(robot, c)
  - EMPTY(robot)
  - ON(box, b)
  - TABLE(a)
  - TABLE(b)

一阶谓词表示法的例子

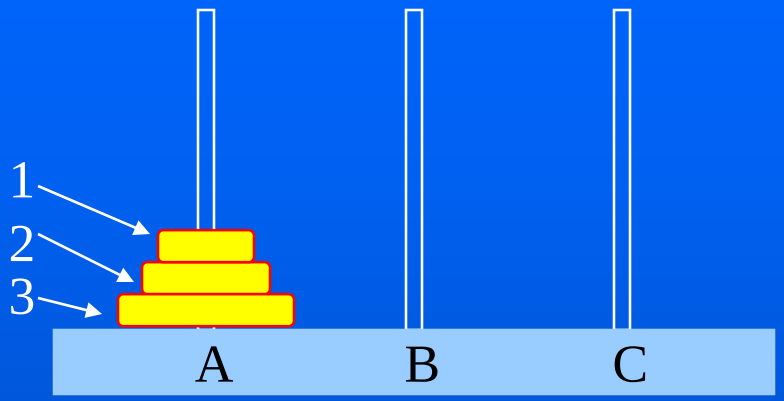
- 操作
  - GOTO(x, y)
    - 条件：AT(robot, x)
    - 动作：删除AT(robot, x);增加AT(robot, y)
  - PICK-UP(x)
    - 条件：ON(x) ∧TABLE(x)∧AT(robot,x) ∧EMPTY(robot)
    - 动作：删除EMPTY(robot) ∧ON(box,x)
      增加：HOLDS(robot,box)
  - SET-DOWN(x)
    - 条件： AT(robot,x) ∧TABLE(x) ∧ HOLDS(robot,box)
    - 动作：删除HOLDS(robot,box)
      增加EMPTY(robot) ∧ ON(box,x)

操作在某一个状态下是否能用

- 定理证明的过程
  - 检查当前状态是否可以是某一个操作的前提条件满足
  - 当前状态→操作前提条件
  - D →R永真

2.Hanoi塔问题：

Hanoi塔问题是由一叠在三根柱子上的木盘组成。这三根柱子固定在一个底板上。木盘具有不同的直径，并在中央有一个可插入柱子的孔。



开始时，所有的木盘都放在A柱子上，如图。

问题的目标是把所有的木盘都移到C柱子上，但一次只能移一个木盘，最终结果是将所有的木盘按原来的存放顺序放在C柱子上。可以把柱子B当作木盘的临时存放处，但任何时候，较大的木盘不能放在较小的木盘上。

求解策略：(1) 可直接移动一个木盘；

(2) 可按下面三个步骤来移动N个木盘：

a.把N-1个木盘移到中间柱子上；

b.把最后一个(第N个)木盘直接移到右边柱子上；

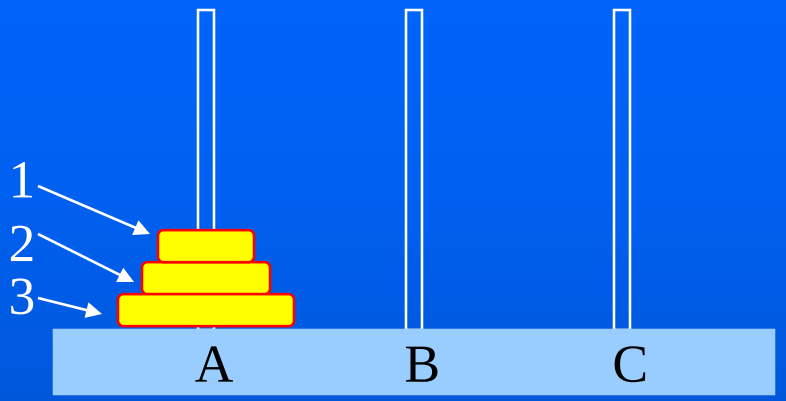
c.把N-1个木盘从中间柱子移到右边柱子上。

此问题用三个谓词描述：

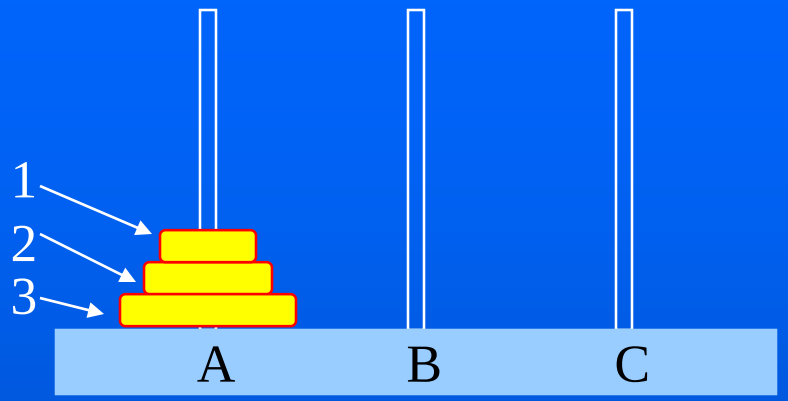
1.hanoi(N)　有N个木盘需要移动

2.move(N,A,B,C)　将N个木盘从A柱子移到C柱子，把B柱子做临时存放处。

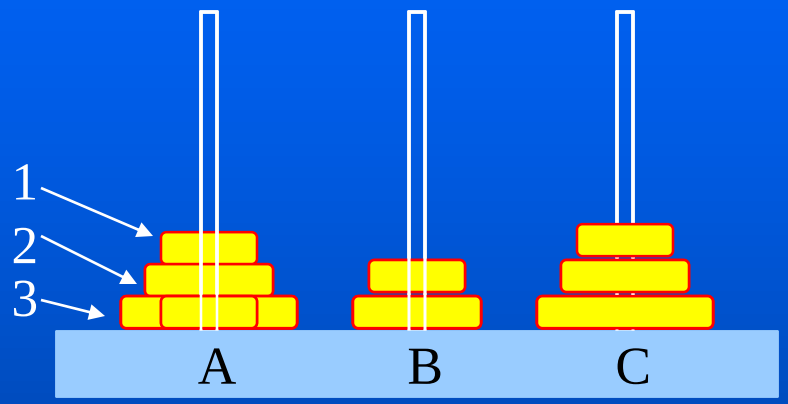
3.inform(A,C)将A柱子的顶端木盘移到C柱子。



```
domains
    loc =right;middle;left
predicates
    hanoi(integer)
    move(integer,loc,loc,loc)
    inform(loc,loc)
clauses
  hanoi(N):-
     move(N,left,middle,right).
  move(1,A,_,C):-
     inform(A,C),!.
```




```
 move(N,A,B,C):-
        N1=N-1,
        move(N1,A,C,B),
        inform(A,C),
        move(N1,B,A,C).
  inform(Loc1, Loc2):-
        write("\nMove a disk
from ", Loc1, " to ", Loc2).
goal
   hanoi(3).
```


1
2
3
A
B
C

表示方法 —逻辑表示法

- 谓词逻辑法是应用最广的方法之一，其原因是：
  - 谓词逻辑与数据库，特别是关系数据库就有密切的关系。在关系数据库中，逻辑代数表达式是谓词表达式之一。因此，如果采用谓词逻辑作为系统的理论背景，则可将数据库系统扩展改造成知识库。
  - 一阶谓词逻辑具有完备的逻辑推理算法。如果对逻辑的某些外延扩展后，则可把大部分的知识表达成一阶谓词逻辑的形式。（知识易表达）
  - ……..

谓词逻辑法应用最广的原因

- 谓词逻辑本身具有比较扎实的数学基础，知识的表达方式决定了系统的主要结构。因此，对知识表达方式的严密科学性要求就比较容易得到满足。这样对形式理论的扩展导致了整个系统框架的发展。
- 逻辑推理是公理集合中演绎而得出结论的过程。由于逻辑及形式系统具有的重要性质，可以保证知识库中新旧知识在逻辑上的一致性（或通过相应的一套处理过程检验）、和所演绎出来的结论的正确性。而其它的表示方法在这点上还不能与其相比。

表示方法 —逻辑表示法

- 谓词逻辑表示法在实际人工智能系统上得到应用。
- 优点
  - 自然性
  - 精密性
  - 严密性
  - 容易实现
- 存在问题：

  谓词表示越细，推力越慢、效率越低，但表示清楚。实际中是要折衷的。

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法

- 框架表示法
- 脚本方法
- 过程表示

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法
  - 框架表示法
  - 脚本方法
  - 过程表示

表示方法—产生式表示法

- 美国数学家Post，1943年提出了一种计算形式体系里所使用的术语。主要是使用类似文法的规则，对符号串做替换运算。这就是最早的一个产生式系统。
- 到了60年代，产生式系统成为认知心理学研究人类心理活动中信息加工过程的基础，由此心理学家认为，人脑对知识的存储就是产生式形式。因此，用它来建立人类认知模型。
- 到目前为止，产生式系统已发展成为人工智能系统中最典型最普遍的一种结构。产生式表示方法是专家系统的第一选择的知识表达方式。

产生式表示方法容易描述事实、规则以及它们的不确定性度量。

❖事实的表示：

实例1：香蕉是黄色的。语言变量——香蕉，值——黄色的

实例2：小李喜欢小莉。语言变量——小李、小莉，关系值——喜欢

一般使用三元组（对象，属性，值）或（关系，对象1，对象2）来表示事实，其中对象就是语言变量，若考虑不确定性就成四元组表示了。这种表示的机器内部实现就是一个表。

如：对事实“老李年龄今年是65岁”可表示成：

（Li，Age，65）

而“老赵和老张是朋友”可写成：

（Friend，Zhao，Zhang）

注：三元组（对象，属性，值）表示考虑了不确定性之后的四元表示为（对象，属性，值，不确定度量值）

❖规则的表示：

表示事物间的因果关系，以：“if condition then action”的单一形式来表示，将规则作为知识的单位。其中的Condition部分称作条件式前件或模式，而Action部分称作动作或后件或结论。条件部分常是一些事实的合取而结论常是某一事实B，如果考虑不确定性，需另附可信度度量值。

与谓词逻辑表示法中的蕴涵式的区别

- 谓词逻辑蕴涵式：表示精确性知识
- 产生式：可以表示非精确性的知识
  - 置信度
  - 前提部分的部分匹配

表示方法—产生式表示法

1、 产生式规则

产生式规则通常用于表示事物间的启发式关联，
其基本形式为：P => Q 或者：IF P then Q

P为规则激活使用的条件（或称前提）；

Q则指示规则激活时（即规则条件部分满足时）应该执行的动作（或应该得出的结论）。

依据规则右部的表示方式，可以把规则分类为条件-动作型和前提-结论型。

表示方法—产生式表示法

- 例如关于动物世界的产生式系统有规则：

  若 某动物是哺乳动物，

  且吃肉；

  则 这种动物是食肉动物。

  或表示为更便于计算机操作的形式化方式：

(Mammal, x) ∧ (Eat, x, Meat)

=> (Carnivore, x)

表示方法—产生式表示法

2、产生式系统组成

一般来讲，产生式系统由三个基本部分组成：规则库、综合数据库和控制系统。前二者构成产生式系统的问题表示（描述），后者则控制应用规则推出解答的全过程。一组规则（产生式）放在一起，它们相互配合，一个规则产生出的结论可以为另外一个规则的前提，用这种方式来求解问题的系统

表示方法—产生式表示法

1）规则库

规则库是产生式规则的集合，用于描述应用领域的常识和启发式知识，所以规则库就是产生式系统的知识库。

表示方法—产生式表示法

- 2） 综合数据库

综合数据库用于描述问题求解状态（简称问题状态），典型的情况下，可以是表示为谓词公式的事实元素集；但也可以是任何的数据结构，如向量、数组和表格等。例如， 产生式系统的基本组成关于动物世界的产生式系统有如下综合数据库：

…

(Mammal Dog)

(Eat Dog Meat)

…

从另一角度，综合数据库可视为推理过程中间结果的存贮池。随着中间结果的不断加入，使综合数据库描述的问题状态逐步转变为目标状态。

表示方法—产生式表示法

□3）控制系统

控制系统是产生式系统的推理机，又称规则的解释器，其驱动和控制整个系统的运行，基本的控制流程是：

识别--行动 循环，

在每个循环的识别阶段，控制系统在规则库中识别条件为真的规则，使这些规则激活；然后在该循环的行动阶段，执行激活的规则，即执行规则的右部指定的对于综合数据库的操作和任何其它合适的操作。

表示方法—产生式表示法

- 在一个循环的识别阶段，若有多于一条的规则激活，就称引起了一个冲突，所谓 冲突解决就是基于某种控制策略去选定需要执行的规则。冲突解决的策略可以分为三大类：

  ·First--选用首条激活的规则加以执行。

  ·Best--选用已激活规则中最好的加以执行，这里"最好"的评价依赖于应用领域制定的标尺。

  ·All--执行所有激活的规则。

表示方法—产生式表示法

- 例如，在关于动物世界的产生式系统中，已建立规则库，其包括上述形式化表示的规则；也建立了如上所述的综合数据库；则因该规则的左部二个匹配模式（Mammal, x）和（Eat, x, Meat）分别与综合数据库中的事实元素（Mammal Dog）和（Eat Dog Meat）匹配（能合一），从而得以激活，并将规则右部的结论（Carnivore Dog）插入综合数据库。

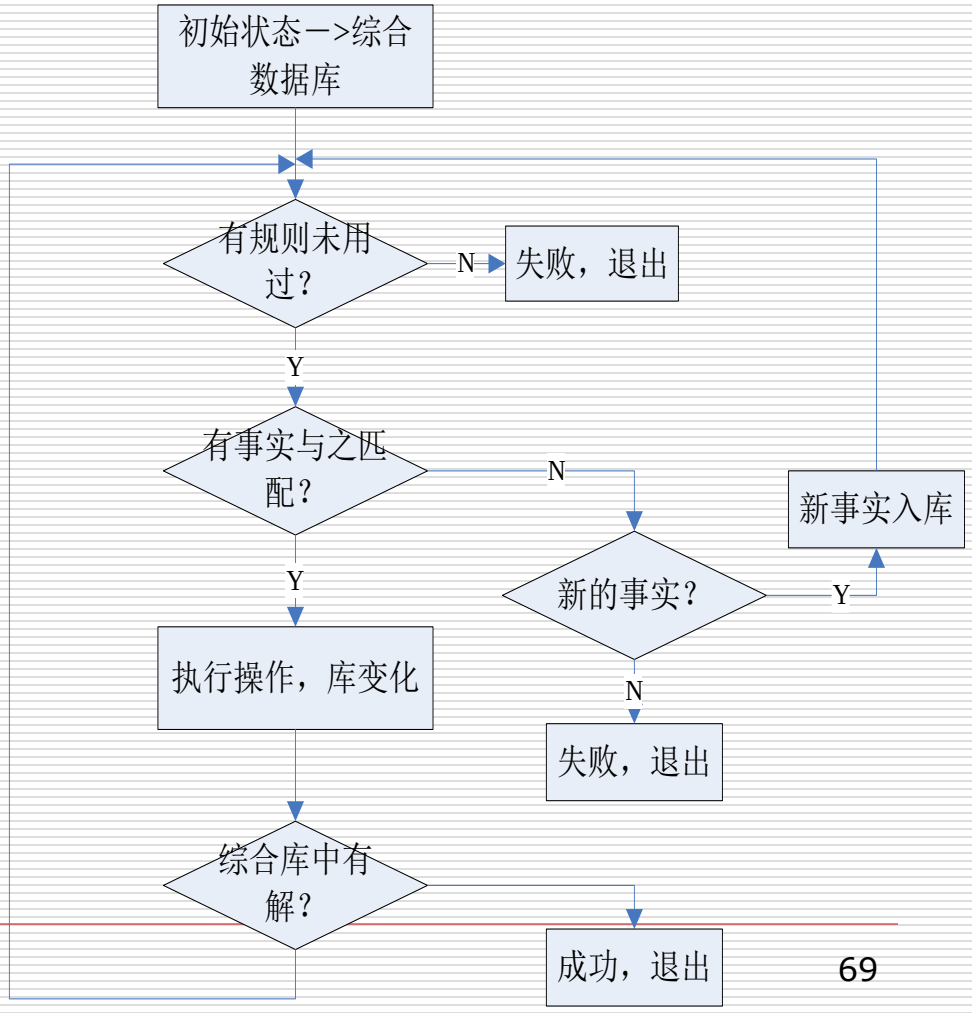
  实际上产生式系统的控制机制就是不断地挑选可激活的规则对综合数据库进行操作，直至得到解答（综合数据库内容转变为描述了目标状态），或失败结束。

推理机构的作用

1. 对于当前的综合数据库，选出适用规则
2. 冲突的解决，按照策略选出一条规则
3. 规则的作用
   - ① 结论：删除、加入一些状态
   - ② 操作：执行某些操作
4. 适当的停机机制

产生式表示法求解问题的过程

- 从规则库中选取规则，作用到综合数据库
- 库发生变化，前进一步
- 直到满足停机条件为止



产生式表示法—控制策略

- 从产生式系统的控制机制可知，选用合适的规则推进问题求解构成控制策略的主要内容。鉴于人工智能应用通常面向困难问题，系统设计时获取的关于求解问题的知识往往不足以确定最合适的规则；从而使问题求解在一定程度上表现为一种搜索过程，即仅对有希望导致最好解答的规则作尝试性使用。一旦发现试用失败，就需撤销试用规则对问题状态产生的影响，使综合数据库的内容恢复到执行该规则之前，再选用别的可激活规则作试探性推理；或者作某些弥补工作（即设法对已得到的部分解答进行修正），再继续进行推理；以期最终搜索到正确地解答。

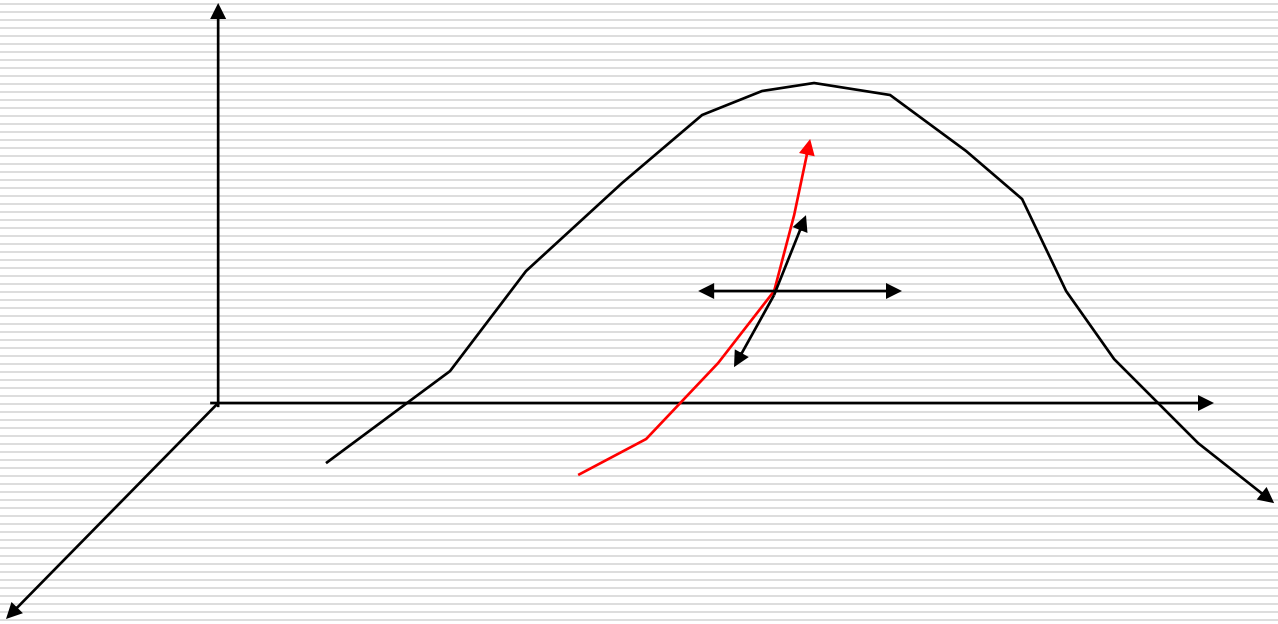
产生式系统的控制策略

- 不可撤回方式（Irrevocable)
- 试探性方式(Tentative)
  - 回溯方式（Backtracking)
  - 图搜索方式(Graph-Search)

不可撤回方式

- 这种方式是利用问题给出的局部知识来决定如何选取规则的，就是说根据当前可靠的局部知识选一条可应用规则并作用于当前综合数据库，接着再根据新状态继续选取规则，搜索过程一直进行下去，不必考虑撤回用过的规则。这是由于在搜索过程中如能有效利用局部知识，即使使用了一条不理想的规则，也不妨碍下一步选得另一条更合适的规则。这样不撤消用过的规则，并不影响求到解，只是解序列中可能多了一些不必要的规则。

爬山法（局部搜索法）



爬山问题

问题介绍

- 解决爬山问题就是确定如何一步一步前进达到顶峰。也就是一个在“爬山”过程中寻求函数的极大值问题。
- 利用高度随位置变化的爬山函数h(p)（当前点到目的点的距离）来引导爬山(h(p)-h(p0)大优先)，就可以实现不可撤回的控制方式。
- 限制：用不可撤回的方式(爬山法)来求解登山问题，只有在登单峰的时才总是有效的(即对单极值的问题可找到解)。对于比较复杂的情况，如碰到多峰、山脊或平顶的情况时，爬山搜索法并不总是有效的。

解题思路

- 建立一个描述综合数据库变化的函数，如果这个函数具有单极值，且这个极值对应的状态就是目标，则不可撤回的控制策略就是选择使函数值发生最大增长变化的那条规则作用于综合数据库，直到函数值最大，达到目标。

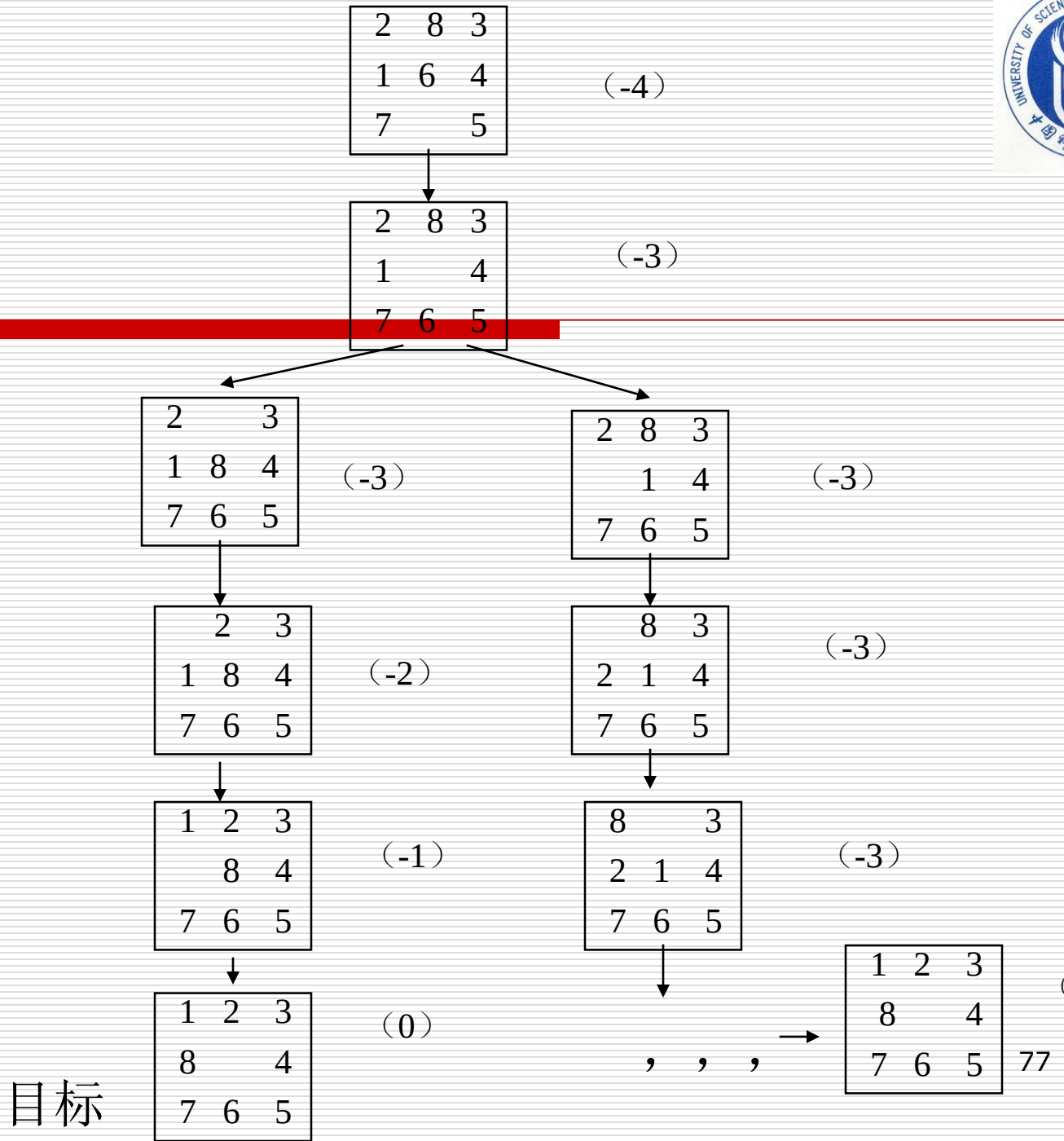
不可撤回方式示例：八数码问题

- 用“不在位”将牌个数并取其负值作为状态描述的函数：-W（n）（“不在位”将牌个数是指当前状态与目标状态对应位置逐一比较后有差异的将牌总个数，n表示任一状态）
- 解路径

1）上(-4)，上(-3)，左(-3)，下(-2)，右(-1)

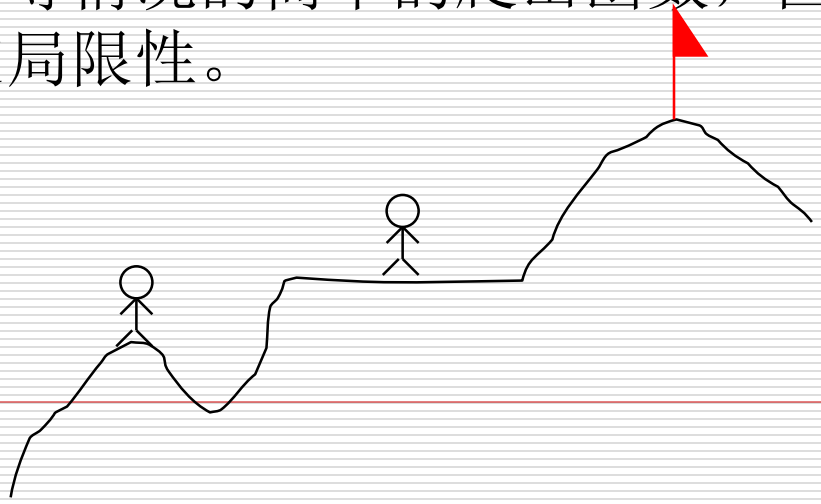
2）上(-4)，左(-3)，上(-3)，右(-3)，下(-3)，左(-3)，上(-3)，右(-2)，下(-1)

八数码游戏各状态的爬山函数值



优缺点

- 使用不可撤回的策略，虽然不可能对任何状态总能选得最优得规则，但是如果应用了一条不适合的规则之后，不去撤消它并不排除下一步应用一条合适的规则，那么只是解序列有些多余的规则而已，求得的解不是最优解，但控制较简单
- 有时很难对给定问题构造出任何情况下都能通用的即不具有多极值或“平顶”等情况的简单的爬山函数，因而不可撤回的方式具有一定局限性。



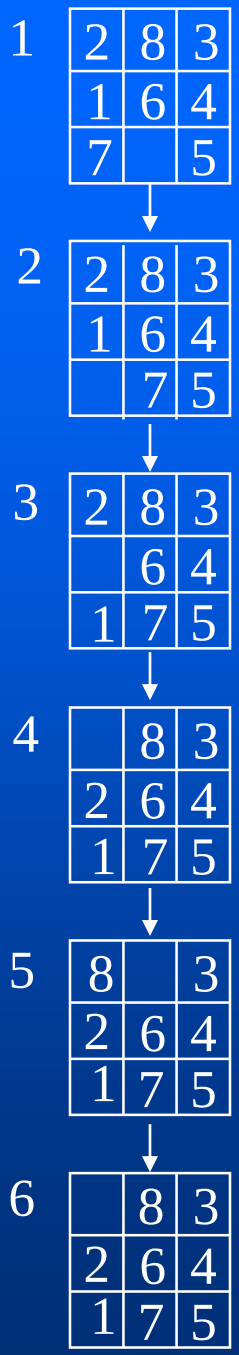
回溯方式

回溯方式

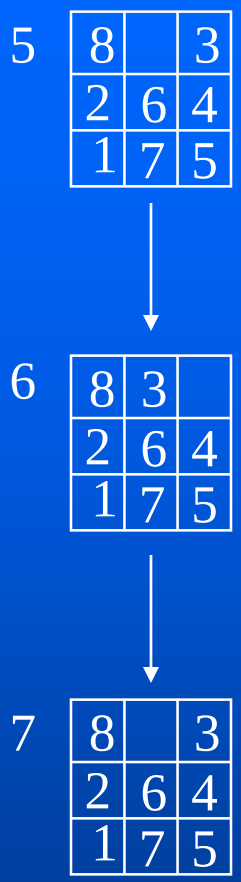
- 在问题求解过程中，有时会发现应用一条不合适的规则会阻挠或拖延达到目标的过程。在这种情况下，需要有这样的控制策略，先试一试某一条规则，如果以后发现这条规则不合适，则允许退回去，另选一条规则来试。
- 关键问题：确定回溯条件、利用有限知识对规则进行排序以减少回溯次数

以八数码问题为例

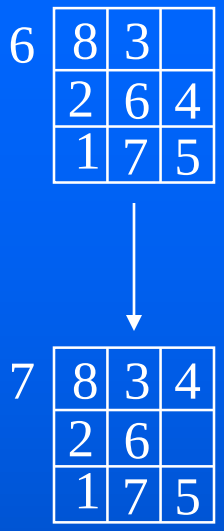
- 回溯点：①新生成的状态在通向初始状态的路径上已出现过；②从初始状态开始，应用的规则数目达到所规定的数目之后还未找到目标状态这一组规则的数目实际上就是搜索深度范围所规定的)；③对当前状态，再没有可应用的规则。



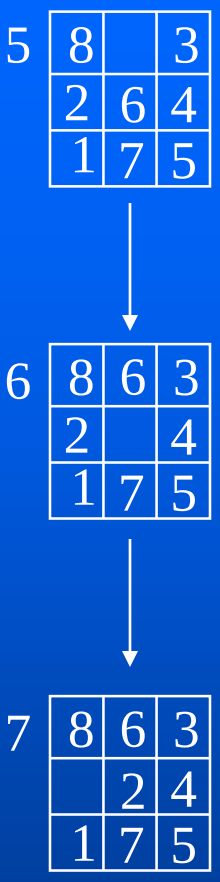
6和4相同回溯到5



7和5相同回溯到6



7已超过深度，6已没有未试过的规则，回溯到5。



7已超过深度，回溯6继续。

8谜问题回溯控制

优缺点

- 回溯过程是一种可试探的方法，从形式上看不论是否存在对选择规则有用的知识，都可以采用这种策略。

- 如果没有有用的知识来引导规则的选取，那么规则可按任意方式（固定排序或随机）选取；如果有好的知识可用，那么用这种知识来引导规则选取，就会减少盲目性，降低回溯次数，甚至不回溯就能找到解

图搜索方式

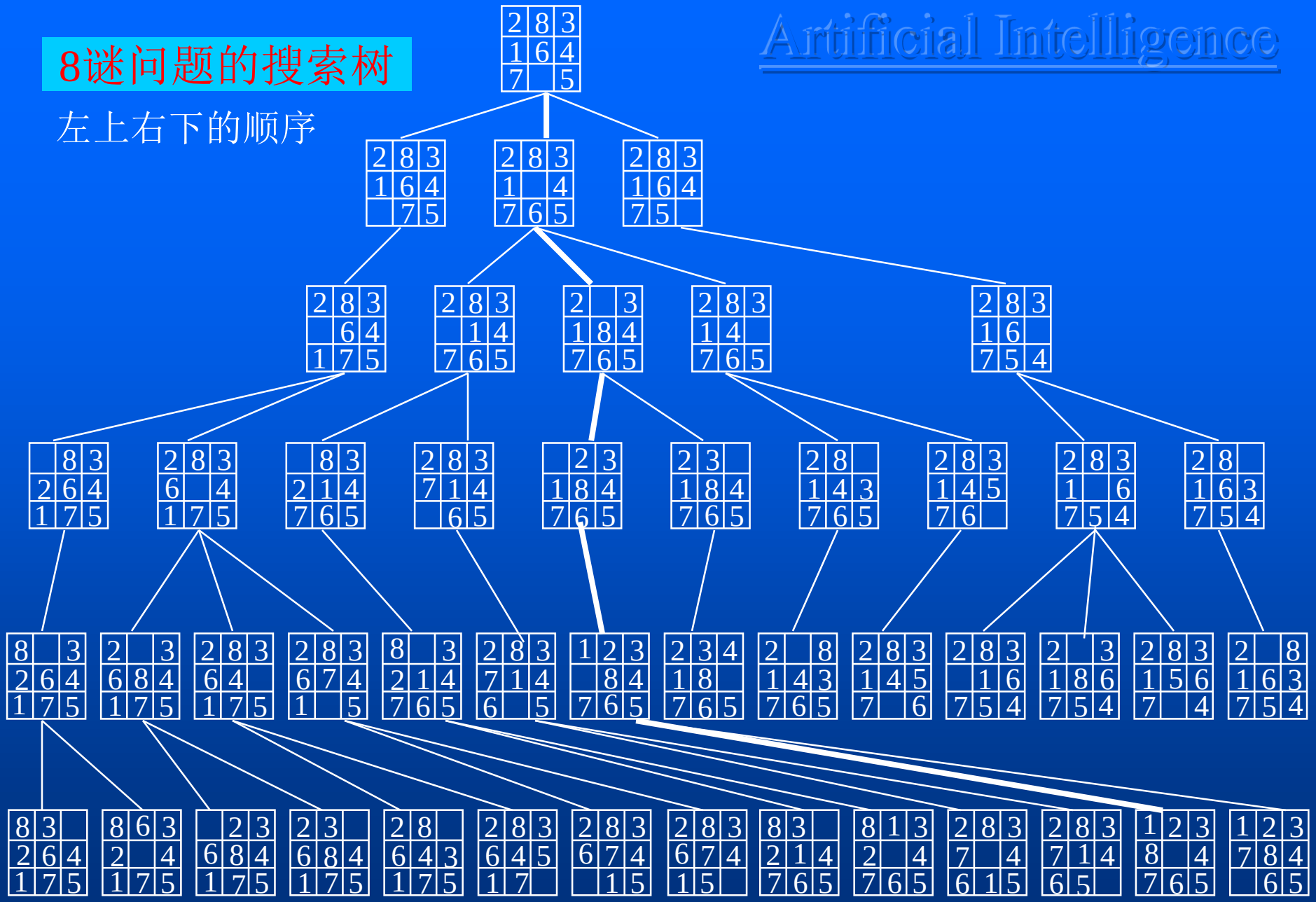
图搜索方式

- 如果把问题求解过程用图或树的这种结构来描述，即图中的每一个节点代表问题的状态，节点间的弧代表应用的规则，那么问题的求解空间就可由隐含图来描述。图搜索方式就是用某种策略选择应用规则，并把状态变化过程用图结构记录下来，一直到得出解为止，也就是从隐含图中搜索出含有解路径的子图来。

八数码问题的搜索树（穷举法）

8谜问题的搜索树

左上右下的顺序



三种搜索策略总结

- 不可撤回方式相当于沿着单独的一条路向下延伸搜索下去； 回溯方式则不保留完整的搜索树结构，只记住当前工作的一条路径，回溯就是对这条路径进行修正； 图搜索方式则记下完整的搜索树

- 对一个要求解的具体问题，有可能用不同的方式都能得到解，选那种方法还要根据其它一些实际的要求考虑决定

表示方法—产生式表示法

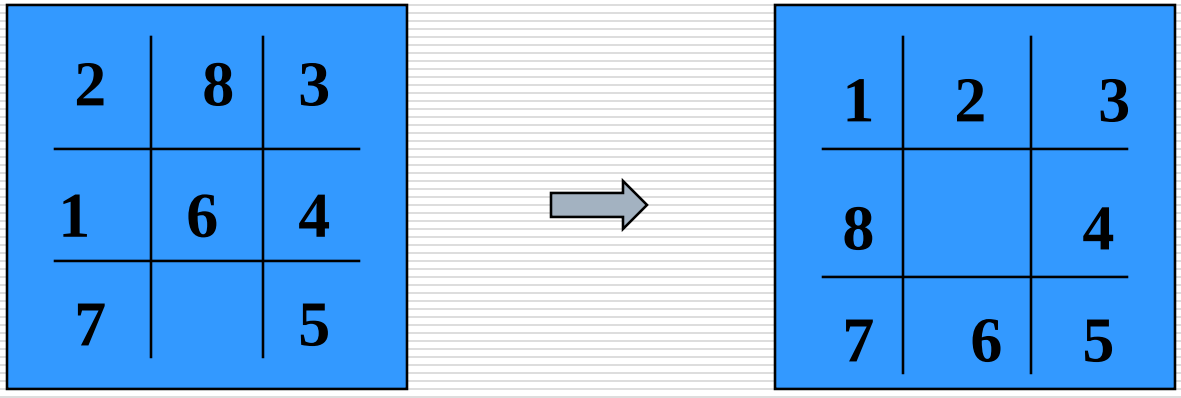
一般来讲，产生式系统由三个基本部分组成：规则库、综合数据库和控制系统。前二者构成产生式系统的问题表示（描述），后者则控制应用规则推出解答的全过程。一组规则（产生式）放在一起，它们相互配合，一个规则产生出的结论可以为另外一个规则的前提，用这种方式来求解问题的系统

八数码问题

- 八数码游戏（Eight-Puzzle）

  在3*3组成的九宫棋盘上，摆有8个将牌，每个将牌都刻有1-8中的某个数码。棋盘中留有一个空格，允许其周围的某个将牌向空格移动。给定一种初始布局和一个目标布局，为如何移动将牌，实现从初始状态到目标状态的转变，给出走步序列。

如：



八数码问题 （续1）

一、综合数据库

用二维数组(Sij)来表示将牌的布局，其中1 ≤ i、j ≤ 3，Sij∈{0,1...8}，且互不相等。

二、规则集：用以下4条规则来模拟将牌空格向左、上、右、下移动的走法

1， IF jo-1>=1 then Siojo:=Sio(jo-1), Si0(jo-1):=0
（Siojo向左）

2， IF io-1>=1 then Siojo:=S(io-1)jo, S(i0-1)jo:=0
（Siojo向上）

3, IF jo+1<=3 then Siojo:=Sio(jo+1), Si0(jo+1):=0
（Siojo向右）

4, IF io+1<=3 then Siojo:=S(io+1)jo, S(i0+1)jo:=0
（Siojo向下）

八数码问题 （续2）

三、搜索策略

是从规则中选取规则并作用于状态的一种广义选取函数

四、问题的解

即实现目标的一个走步序列（即规则序列），如（上、上、左、下、右）

传教士与野人问题

- 传教士与野人问题（M-C问题）

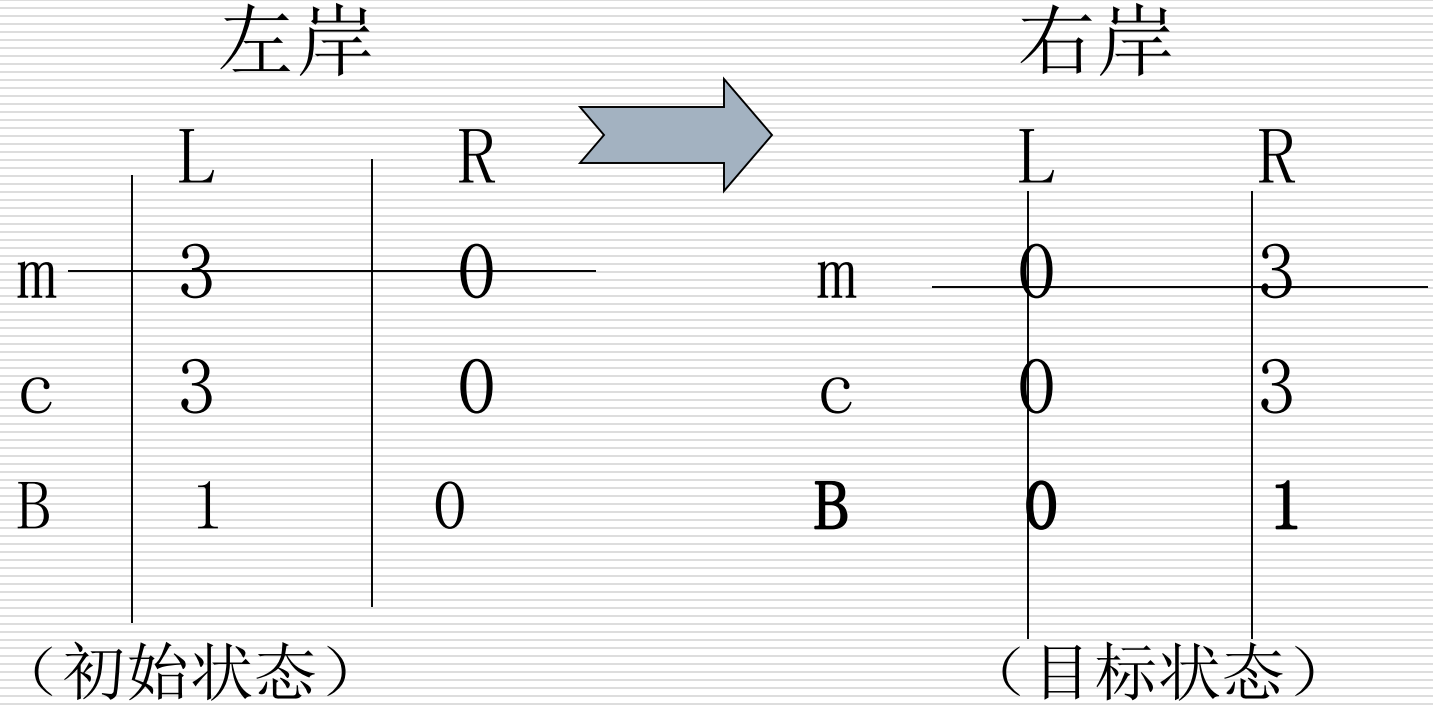
  问题：N个传教士，N个野人，一条船，可同时乘坐k个人乘渡。

  问：传教士为安全起见，应如何规定摆渡方案，使得任何时刻，河两岸以及船上的野人数目总是不超过传教士的数目。

- 以N=3，k=2为例求解。

M-C问题（续１）

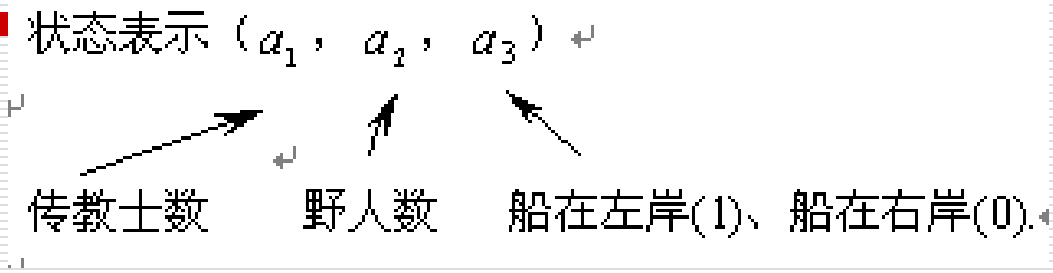
图中L和R表示左岸和右岸，B=1或0表示有船或无船，约束条件是：两岸上M>=C,船上M+C<=2：



M-C问题（续２）

1，综合数据库

（m，c，b），



其中：0≤m，c≤3，b ∈{0，1}

2，初始状态

（3，3，1）

3，目标状态（结束状态）

（0，0，0）

M-C问题（续3）

4，规则集

IF (m, c, 1) THEN (m-1, c, 0)

IF (m, c, 1) THEN (m, c-1, 0)

IF (m, c, 1) THEN (m-1, c-1, 0)

IF (m, c, 1) THEN (m-2, c, 0)

IF (m, c, 1) THEN (m, c-2, 0)

IF (m, c, 0) THEN (m+1, c, 1)

IF (m, c, 0) THEN (m, c+1, 1)

M-C问题（续４）

IF (m, c, 0) THEN (m+1, c+1, 1)

IF (m, c, 0) THEN (m+2, c, 1)

IF (m, c, 0) THEN (m, c+2, 1)

也可以定义为：

IF (m, c, 1) AND 1 ≤i+j≤2 THEN (m-i, c-j, 0)

IF (m, c, 0) AND 1 ≤i+j≤2 THEN (m+i, c+j, 1)

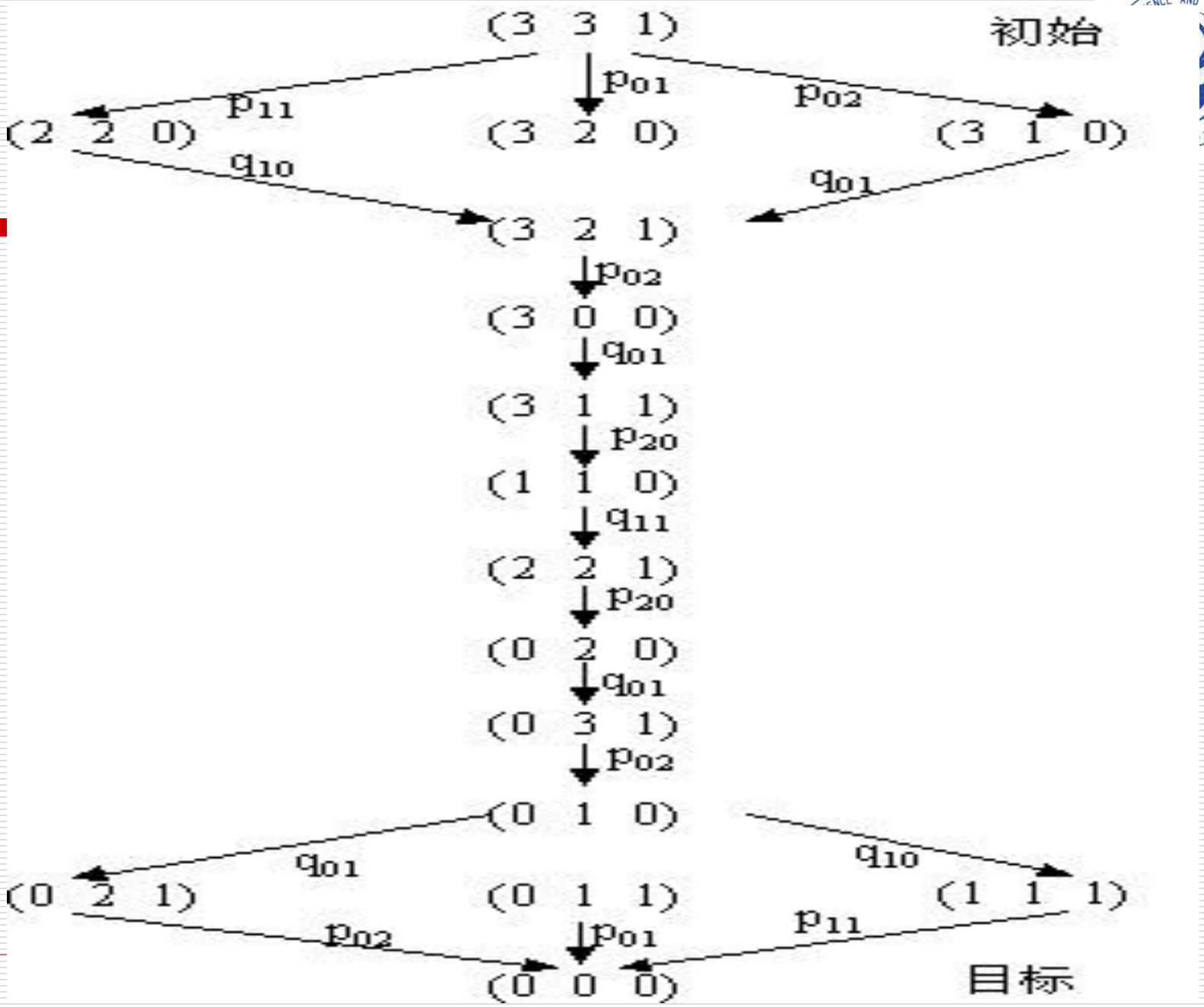
５，控制策略：（略）

M-C问题（续5）

N＝3的M-C问题，状态空间的总状态数为4X4X2＝32，根据约束条件的要求，可以看出只有20个合法状态。再进一步分析后，又发现有4个合法状态实际上是不可能达到的。因此实际的问题空间仅由16个状态构成。下表列出分析的结果：

(0 0 1)达不到，(0 0 0)，(0 1 1)，(0 1 0)，(0 2 1)，(0 2 0)，(0 3 1)，(0 3 0)达不到，(1 0 1)不合法，(1 0 0)不合法，(1 1 1)，(1 1 1)，(1 2 1)不合法，(1 2 0)不合法，(1 3 1)不合法，(1 3 0)不合法，(2 0 1)不合法，(2 0 0)不合法，(2 1 1)不合法，(2 1 0)不合法，(2 2 1)，(2 2 0)，(2 3 1)不合法，(2 3 0)不合法，(3 0 1)达不到，(3 0 0)，(3 1 1)，(3 1 0)，(3 2 1)，(3 2 0)，(3 3 1)，(3 3 0)达不到

M-C问题状态空间图


(3 3 1)
初始
p01
p11
p02
(2 2 0)
(3 2 0)
(3 1 0)
q10
q01
(3 2 1)
p02
(3 0 0)
q01
(3 1 1)
p20
(1 1 0)
q11
(2 2 1)
p20
(0 2 0)
q01
(0 3 1)
p02
(0 1 0)
q01
q10
(0 2 1)
(0 1 1)
(1 1 1)
p02
p01
p11
(0 0 0)
目标

字符转换

- 问题：设字符转换规则

  $A \wedge B \rightarrow C$

  $A \wedge C \rightarrow D$

  $B \wedge C \rightarrow G$

  $B \wedge E \rightarrow F$

  $D \rightarrow E$

  已知：A，B

  求：F

字符转换 （续1）

一、综合数据库

{x}，其中x为字符

二、规则集

1，IF A∧B THEN C
2，IF A∧C THEN D
3，IF B∧C THEN G
4，IF B∧E THEN F
5，IF D THEN E

字符转换 （续２）

三、控制策略

顺序排队

四、初始条件

{A，B}

五、结束条件

F∈{x}

求解过程

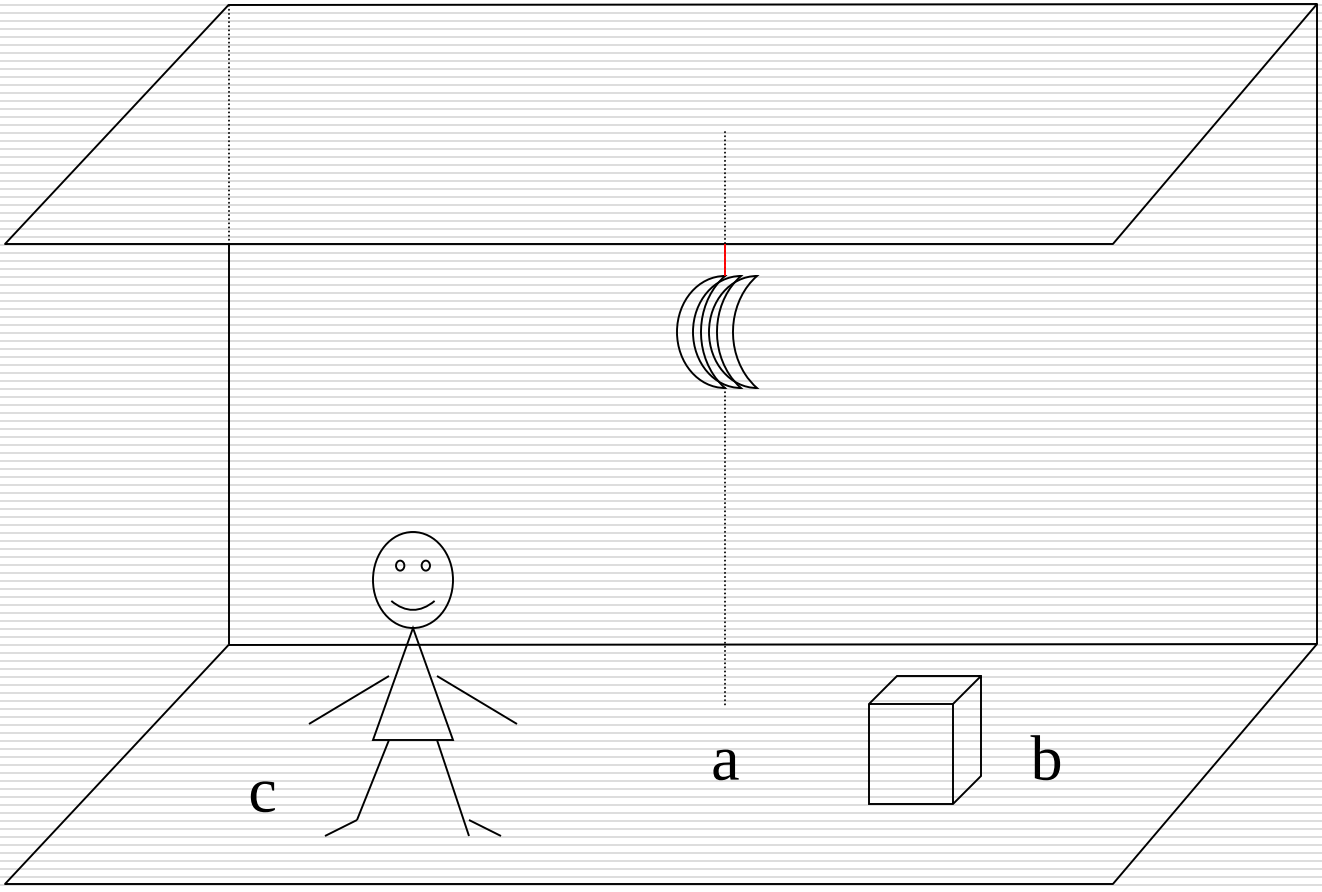
数据库	可触发规则	被触发规则
A，B	（1）	（1）
A，B，C	（2） （3）	（2）
A，B，C，D	（3） （5）	（3）
A，B，C，D，G	（5）	（5）
A，B，C，D，G，E	（4）	（4）
A，B，C，D，G，E，F		

1，IF A∧B THEN C　　2，IF A∧C THEN D

3，IF B∧C THEN G　　4，IF B∧E THEN F

5，IF　D　THEN E

猴子摘香蕉问题



猴子摘香蕉问题（续1）

1，综合数据库

（M, B, Box, On, H）

M：猴子的位置

B：香蕉的位置

Box：箱子的位置

On=0：猴子在地板上

On=1：猴子在箱子上

H=0：猴子没有抓到香蕉

H=1：猴子抓到了香蕉

猴子摘香蕉问题（续2）

2，初始状态

（c，a，b，0，0）

3，结束状态

（x1，x2，x3，x4，1）

其中x1～x4为变量。

猴子摘香蕉问题（续3）

4，规则集

r1: IF (x, y, z, 0, 0) THEN (w, y, z, 0, 0)

r2: IF (x, y, x, 0, 0) THEN (z, y, z, 0, 0)

r3: IF (x, y, x, 0, 0) THEN (x, y, x, 1, 0)

r4: IF (x, y, x, 1, 0) THEN (x, y, x, 0, 0)

r5: IF (x, x, x, 1, 0) THEN (x, x, x, 1, 1)

其中x，y，z，w为变量

表示方法—产生式表示法

- 产生式系统的基本特征：
  - 一组规则，即产生式本身。

    每个规则分左边右边。

    如：天上下雨 → 地上湿

    一般左边表示情况，即什么条件。发生时产生式被调用。通常用匹配方法核实情况。匹配成功时，执行右边规定的动作。

    ……

表示方法—产生式表示法

产生式系统的基本特征：

- 数据库

  存放的数据是构成产生式的基本元素，又是产生式作用的对象。这里的数据是广义的常量、变量、多元组谓词、表、图像等。往往事实或断言——知识元

- 一个解释程序

  从匹配成功的规则（可能不止一个）中选出一个加以执行。

表示方法—产生式表示法

产生式系统基本结构

- 工作存储器：存放当前已知的数据，包括推理过程中形成的中间结论。数据是广义的，可以是常量、多元数组、谓词、表示结构等。
- 产生式规则：每条产生式规则分为左右两个部分。左部表示激活该产生式规则的条件，右部表示调用该产生式规则后所作的动作。 条件是一组复杂的模式，规则之间的控制也不是语句的传递，而且满足条件的规则被激活但不一定立即执行，取决于产生式系统的冲突消解策略。

表示方法—产生式表示法

产生式系统基本结构

……

规则解释程序

- 匹配器：判断规则条件是否成立。
- 冲突消解器：选择可调用的规则。
- 解释器：执行规则的动作。并且在满足结束条件时终止产生式系统运行。

表示方法—产生式表示法

- 推理方法：

  正向，向前推理，数据驱动

  反向，向后推理，目标驱动

  双向，

表示方法—产生式表示法

正向推理方法：

从已知事实出发，逐步推导出最后结论。其推理过程大致是：

- 用工作存储器中的事实与产生式规则的前提条件进行匹配。
- 按冲突消解策略从匹配的规则中选择一条规则。
- 执行选中规则的动作，依次。修改工作存储器。
- 用更新后的工作存储器，重复上述工作，直到得出结论或工作存储器不再发生变化为止。

表示方法—产生式表示法

反向推理方法：

首先提出假设，然后验证这些假设的真假性，找到假设成立的所有证据或事实。其推理过程大致是：

- 看假设是否在工作存储器中，若在，则假设成立，推理结束。
- 找出结论与此假设匹配的规则。
- 按冲突消解策略从匹配的规则实例中选择一条规则。
- 将选中的规则的前提条件作为新的假设，重复上述工作，直到假设的真假性被验证或不存在激活的规则。

表示方法—产生式表示法

- 双向推理方法：

  即自顶向下、又自底向上作双向推理，直至某个中间界面上两方向结果相符便成功结束。

  该方法较正向或反向推理所形成的推理网络小，从而推理效果更高。

表示方法—产生式表示法

推理方法的选择

推理方法的选择取决于推理的目标和搜索空间的形状。

- 如果目标是从一组给定事实出发，找出所有可能的结论，那么，通常使用正向推理。
- 如果目标是证实或否定某一特定结论，那么，通常使用反向推理，否则，从一组初始事实出发盲目地正向推理，可能得出许多和所要证实的结论无关的结论。

表示方法—产生式表示法

- 推理空间形状
  - 方向分支因素。即从一结点可以直接到达的平均结点数。一般从分支因素低的方向开始推理更加有效。
  - 开始状态数与终止状态数。即当两个方向的分支因素没有明显差异时，从小的状态集出发朝大的状态集推理，这样找解比较容易。

表示方法—产生式表示法

- 正向、逆向、双向产生式系统
  - 正向产生式系统是从初始状态出发朝着目标状态这个方向来使用规则，即正推的方式工作的，我们称这些规则为F规则。反之称为B规则；
- 可交换的产生式系统
  - 可交换性是指问题在求解过程中可任意交换可应用规则的次序而不影响求解；
- 可分解的产生式系统
  - 能够分解产生式系统的综合数据库和结束条件的产生式系统称为可分解的产生式系统。

表示方法—产生式表示法

（1）正向产生式系统——这种系统通过检查前提是否满足当前问题状态（与综合数据库内容匹配）来决定规则的激活，由此实现正向推理方式，并推动问题求解从初始状态向目标状态逼近。以正向推理方式使用的规则称为正向规则，或F规则（Forward rule）。

规则1: IF P1 THEN P2

规则3: IF P2 THEN P3

规则2: IF P3 THEN q3

表示方法—产生式表示法

（2）逆向产生式系统——这种系统通过检查结论是否满足当前问题状态来决定规则的激活，由此实现逆向推理方式，并推动问题求解从目标状态向初始状态逼近。以逆向推理方式使用的规则称为逆向规则，或B规则（Backward rule）。

规则1: IF P1 THEN P2

规则3: IF P2 THEN P3

规则2: IF P3 THEN q3

表示方法—产生式表示法

（3）双向产生式系统——这种系统以双向推理方式（正、逆向同时进行）去求解问题。

双向系统的综合数据库必须有两套数据结构，分别描述从初始状态出发推得的中间状态——正向状态，和从目标状态出发推得的中间状态——逆向状态。换言之，综合数据库 = 正向状态描述 +逆向状态描述，以便于F、B规则分别作用于不同的状态描述。

表示方法—产生式表示法

选用那种推理方向，主要取决于问题的特征，最重要的是推理的分支因素，即每个识别-行动循环激活的规则数。显然规则数越大，分支越多，规则的激活检查和启发式评价的工作量也就越大。所以，应选择分支因素小的推理方向。在正、逆向分支因素接近的情况下，双向可提高效率，但由于需求解的问题本身的复杂性和关于规则选用的启发式知识往往不完善，易于发生双向推理不相交的情况，导致比单向更低的推理效率

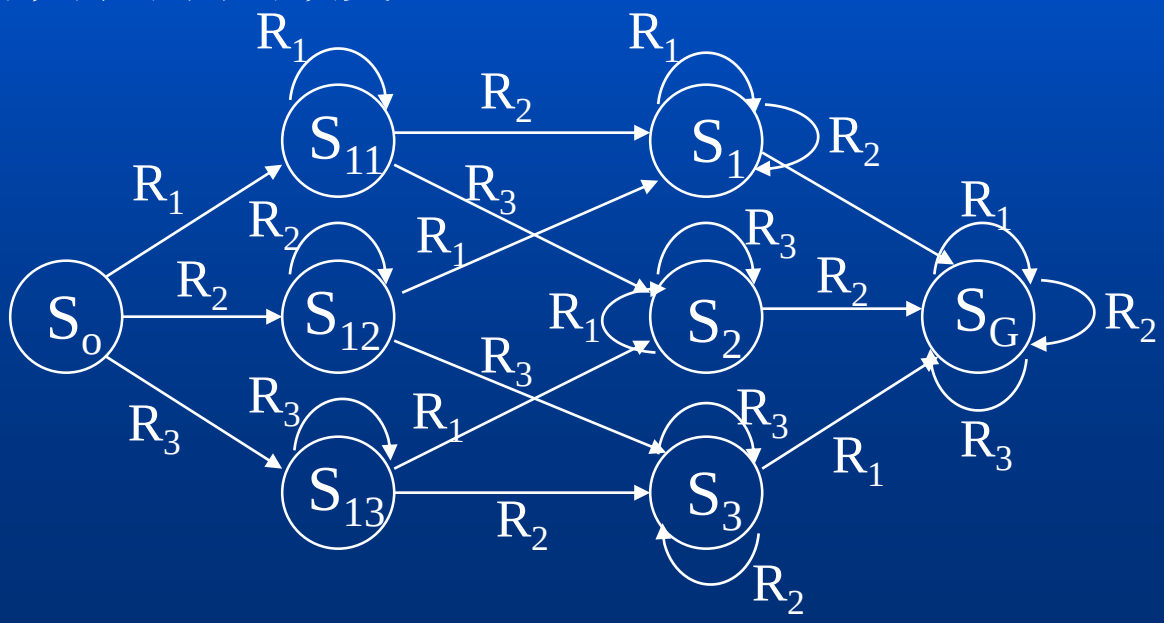
可交换的产生式系统

一个产生式系统称作是可交换的，如果它对任意状态描述D具有如下性质：

(1)每一条对D可应用的规则，对于对D应用一条可应用的规则后所产生的状态描述仍是可用的。

(2)如果D满足目标条件，则对D应用任何一条可应用的规则所产生的状态描述也满足目标条件。

(3)由可应用于D的规则所构成的规则序列应用于D后，所产生的状态描述不因序列的次序不同而改变。



可交换系统简例：

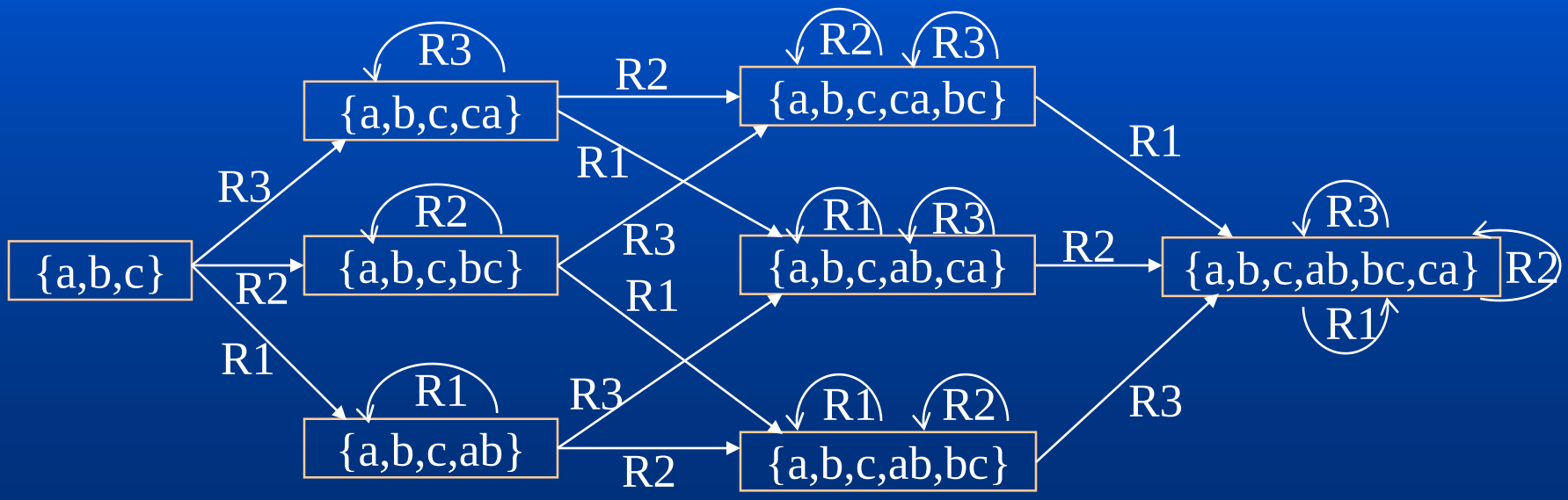
给定一个整数集合{a,b,c}，可通过把集合中任意一对元素的乘积作为新元素添加到集合中，来扩大该整数集，要求通过若干次操作后能生成所需的集合来。　　用产生式系统表示并求解此问题：

初始数据库：{a,b,c}　目标数据库：{a,b,c,ab,bc,ca}

规则集：R1:if{a,b,c}then {a,b,c,ab}

R2:if{a,b,c}then {a,b,c,bc}

R3:if{a,b,c}then {a,b,c,ca}



可分解的产生式系统

例：一产生式系统的初始状态描述为：(C,B.Z)目标状态描述为：(M,M,M……)

用图搜索方式求解：

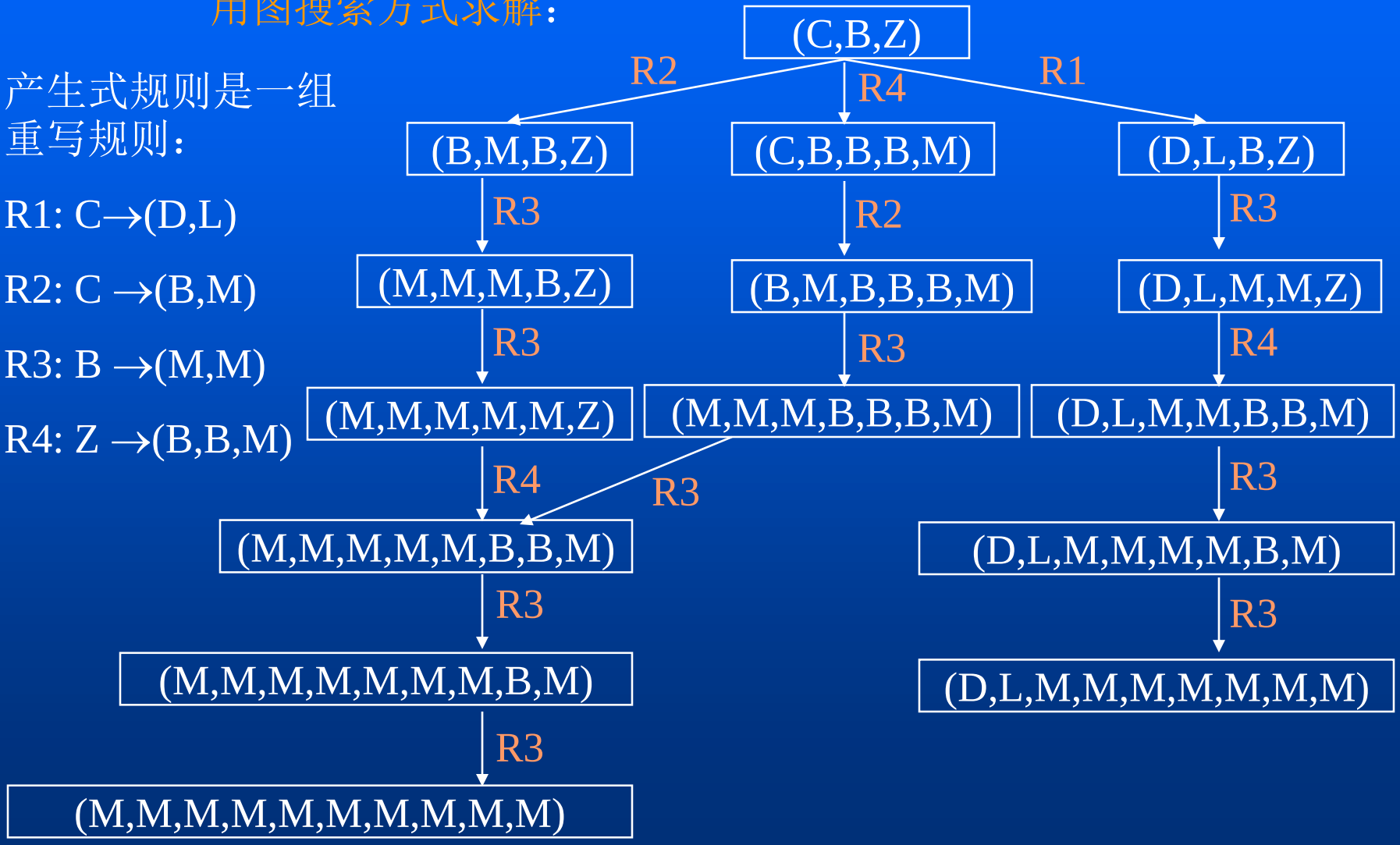
产生式规则是一组
重写规则：

R1: C→(D,L)

R2: C →(B,M)

R3: B →(M,M)

R4: Z →(B,B,M)



上例中，初始状态可以分解成C,B和Z，然后把产生式规则应用于这些组成部分，应用规则后得到的结果状态又可进一步分裂，这样不断地分解，不断地应用规则，直到所有的状态全由M组成为止。

如果一个产生式系统能够分解成若干组成部分，产生式规则可以分别用在各组成部分上，整个系统的终止条件可以用各组成部分的终止条件表示出来。这样的产生式系统叫做可分解的产生式系统。

可分解产生式系统的基本算法：

1.DATA←初始状态描述

2.$\{D_i\}$ ←DATA的分解结果；这里把每一个D_i看成是独立的状态描述

3.until所有的$\{D_i\}$满足终止条件，do:

4.begin

5.在$\{D_i\}$中选择一个不满足终止条件的D*

6.在$\{D_i\}$中删除D*

7.在规则集合中选出一个可应用于D*的规则R

8.D ← 把R应用于D*的结果

9.$\{d_i\}$ ← D的分解结果

10.把$\{d_i\}$加入$\{D_i\}$中

11.end

与/或图的结构对于描述可分解系统的算法十分有用。

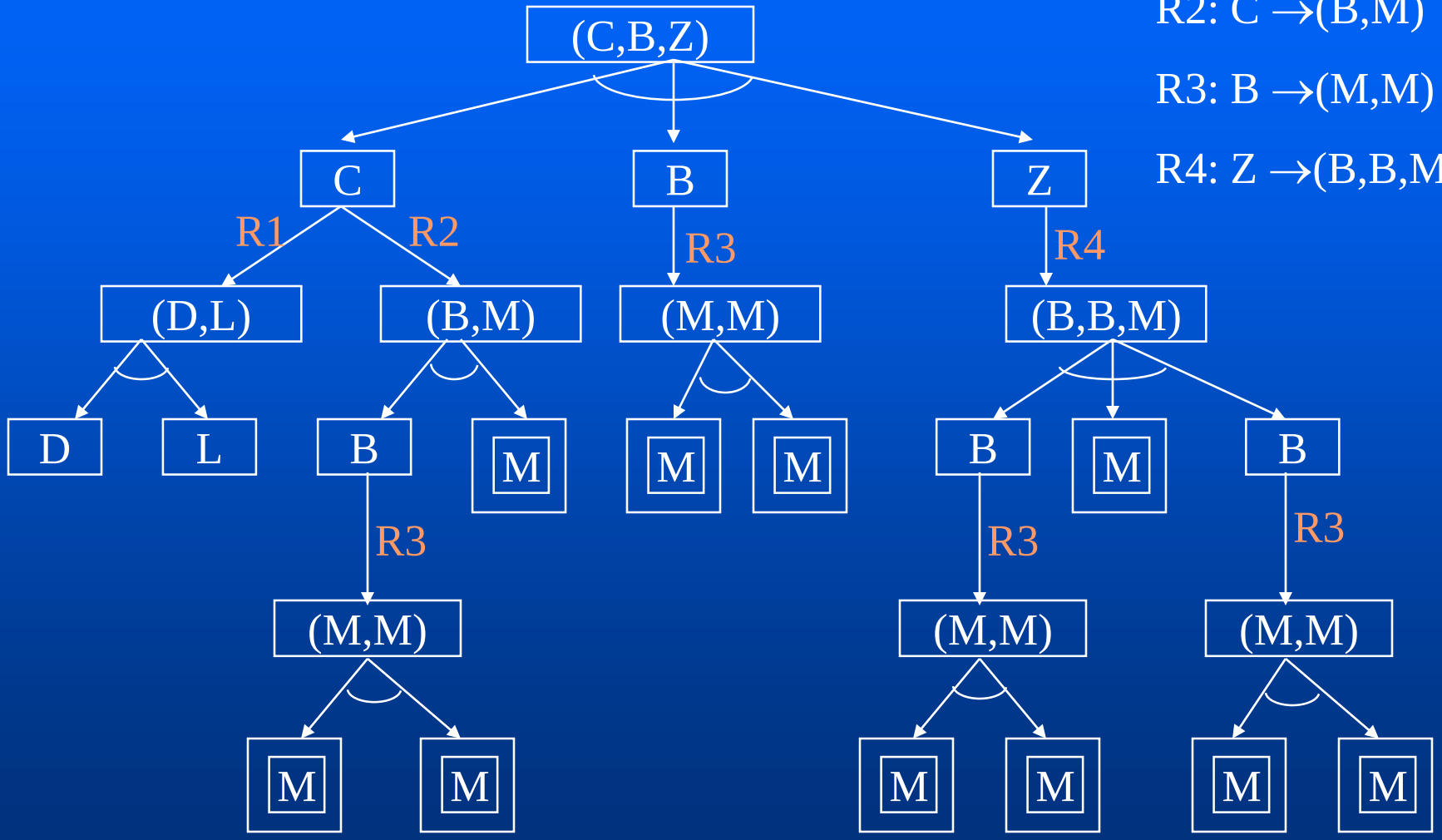
以上重写问题的与/或树如下：

R1: C→(D,L)

R2: C →(B,M)

R3: B →(M,M)

R4: Z →(B,B,M)



表示方法—产生式表示法

特点

- 用产生式系统结构求解问题的过程和人类求解问题时的思维很相像。因而可以用它来模拟人类求解问题的思维过程。
- 可以把产生式系统作为人工智能系统的基本结构单元或基本模型看待。就好像是积木世界中的积木块一样。因而研究产生式系统的基本问题就具有一般意义。
- 表示的格式固定、形式单一、规则间相互独立。所以建立容易；推理方式单纯、知识库与推理机分离，修改方便、容易理解。

表示方法—产生式表示法

- 优点
  - 模块性。

    规则与规则之间相互独立
  - 灵活性。

    知识库易于增加、修改、删除
  - 自然性。

    方便地表示专家的启发性知识与经验
  - 透明性。

    易于保留动作所产生的变化、轨迹

表示方法—产生式表示法

- 缺点：
  - 知识库维护难。
  - 效率低。为了模块一致性
  - 理解难。由于规则一致性彼此之间不能调用。
- 应用实例：
  - 用于化工工业测定分子结构的DENDRAL
  - 用于诊断脑膜炎和血液病毒感染的MYCIN
  - 估计矿藏的PROSPECTOR

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法
  - 框架表示法
  - 脚本方法
  - 过程表示

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法
- 框架表示法
- 脚本方法
- 过程表示

表示方法—语义网络表示法

概述

- 1968年R.Quillian的博士论文建议用一种语义网络来描述人对事物的认知，实际上是对人脑功能的模拟。他主张在处理自然语言词义理解问题时，必须把语义放在第一位，一个词的含义只有根据所属的上下文环境才能准确地把握。Simon于1970年正式提出语义网络的概念。
- 逻辑和产生式表示方法常用于表示有关领域中各个不同状态间的关系。然而用于表示一个事物同其各个部分间的分类知识就不方便了。
- Duda等人设计的著名专家系统PROSPECTOR采用了语义网络作为知识表示方式。
- 语义网络同一阶逻辑有相同的能力。多用于自然语言处理。
- 分类：命题语义网络、数据语义网络、语言语义网络、结构、分类等

语义网络的结构

语义网络是一种用实体及其语义关系来表达知识的有向图。

结点：代表实体，表示各种事物、概念、情况、属性、状态、事件、动作等；

弧：代表语义关系，表示它所连接的两个实体之间的语义联系。

在语义网络中，每一个结点和弧都必须带有标识，这些标识用来说明它所代表的实体或语义。

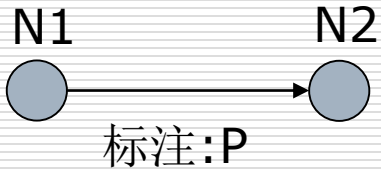
语义网络的结构

从结构上来看，语义网络一般由一些最基本的语义单元组成。这些最基本的语义单元被称为语义基元。可用如下三元组来表示：（结点1，弧，结点2）

注意：在语义网络中，弧是有向弧，方向不能随意调换。语义网络表示法和产生式表示法及谓词逻辑表示法之间有着对应的表示能力。

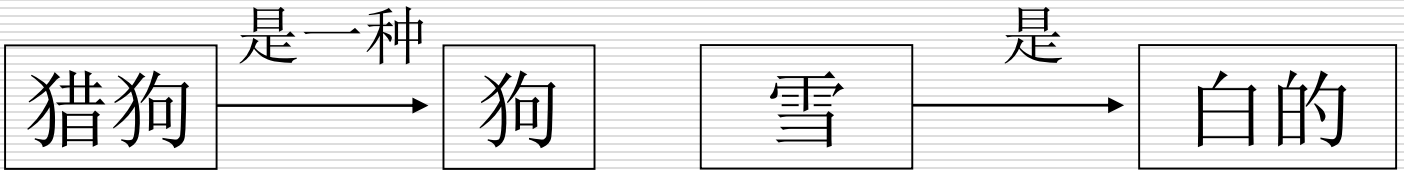
语义网络的结构

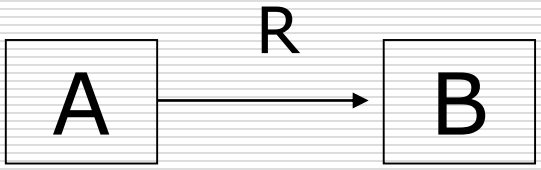
- 弧
  - 方向：
    - 体现节点的主从性
    - 节点1主节点，节点2：从节点
  - 标注：
    - N1有N2的性质
    - N1和N2之间具备某种关系



- 语义网络的2元组表示：P(N1,N2)

语义网络的例子





从功能上说，语义网络可以描述任何事物间的任意复杂关系。从一些基本的语义关系组合成任意复杂的语义关系是可行的。

个体为中心的语义联系

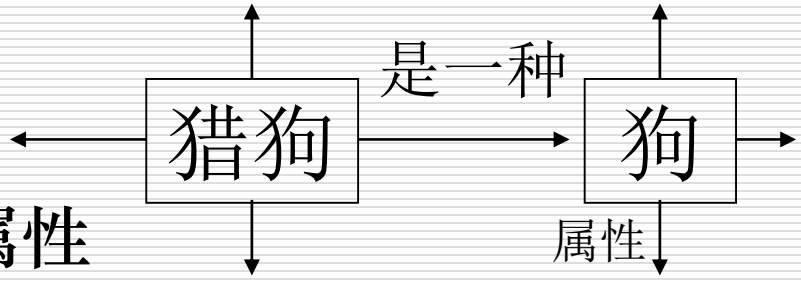
- 实例联系：小华是女孩　ISA
- 泛化联系：一类结点与更抽象结点　AKO
- 聚集联系：个体与其组成部分　PARTOF
- 属性联系：个体、属性、值之间的关系 age，…

常用的节点之间关系的语义网络表示

- 分类关系
- 聚类关系
- 属性关系
- 推论关系
- 时间位置关系
- 合取、析取关系

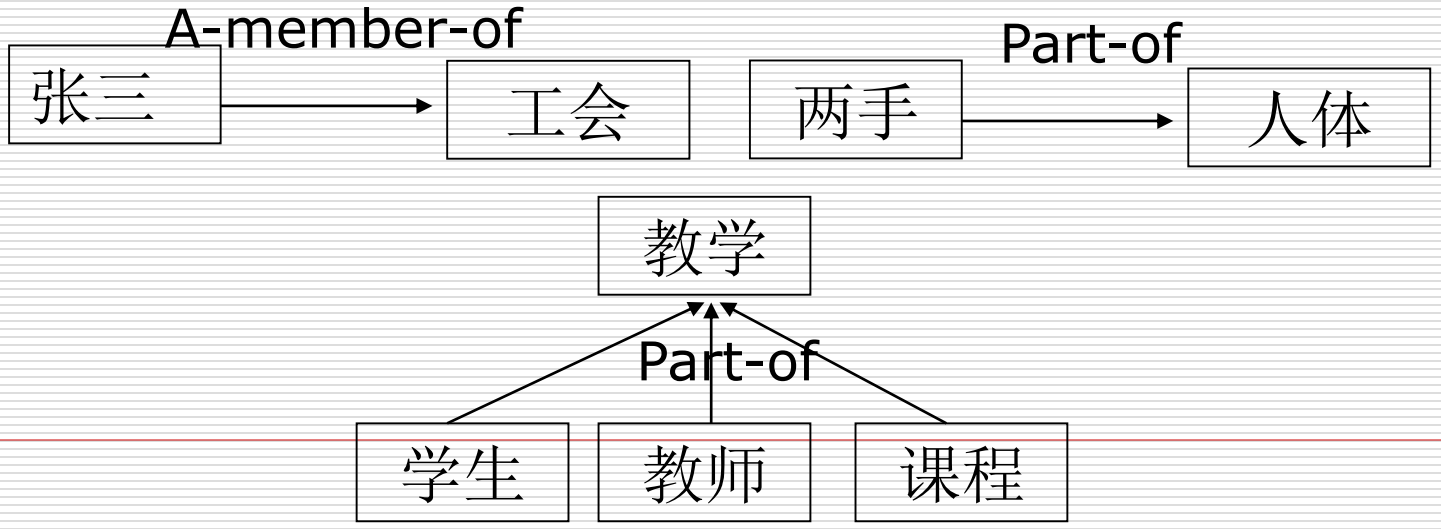
分类关系

- 标注：is a ; is kind of
- 表示节点之间的隶属关系，抽象与具体的关系
- 例子：
  - 猎狗可以继承狗的一些属性
  - 猎狗可以具有它特有的属性
  - 猎狗可以对狗共有的一些属性细化、补充和变异



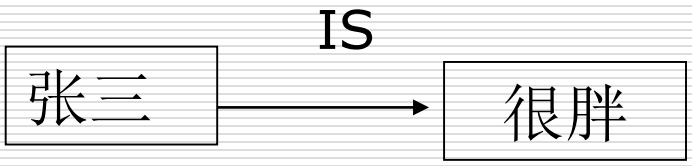
聚类关系

- 标注：part of ; a member of
- 常用于表示部分和全体之间的关系，或包含关系
- 在聚类关系中，N1和N2的属性很不相同



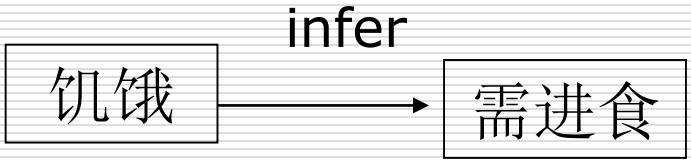
属性关系

- 标注：IS
- 表示N1节点的属性为N2



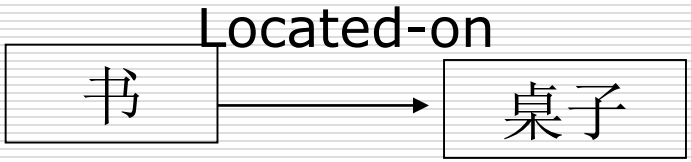
推论关系

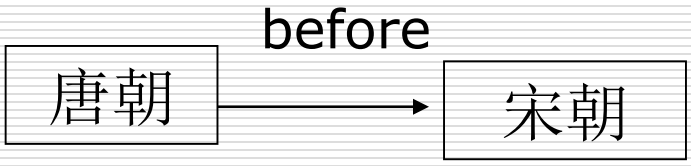
- 标注：infer
- 表示N1节点和N2节点之间存在因果关系，由N1可以推出N2



时间、位置关系

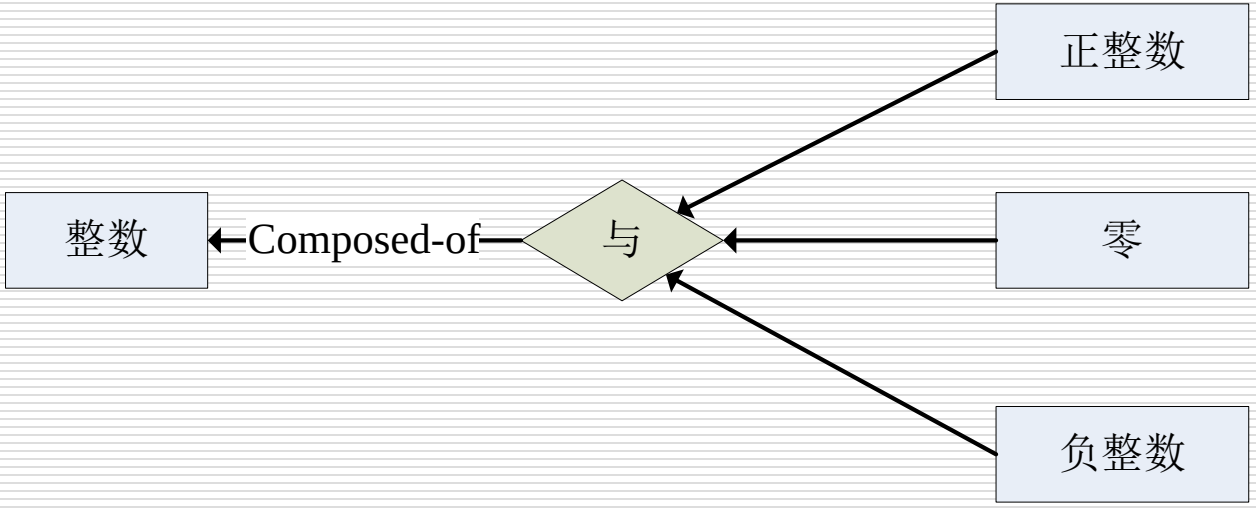
- 标注：at, before, after, located-on , located-under, located-at
- 表示节点之间的时间和位置关系



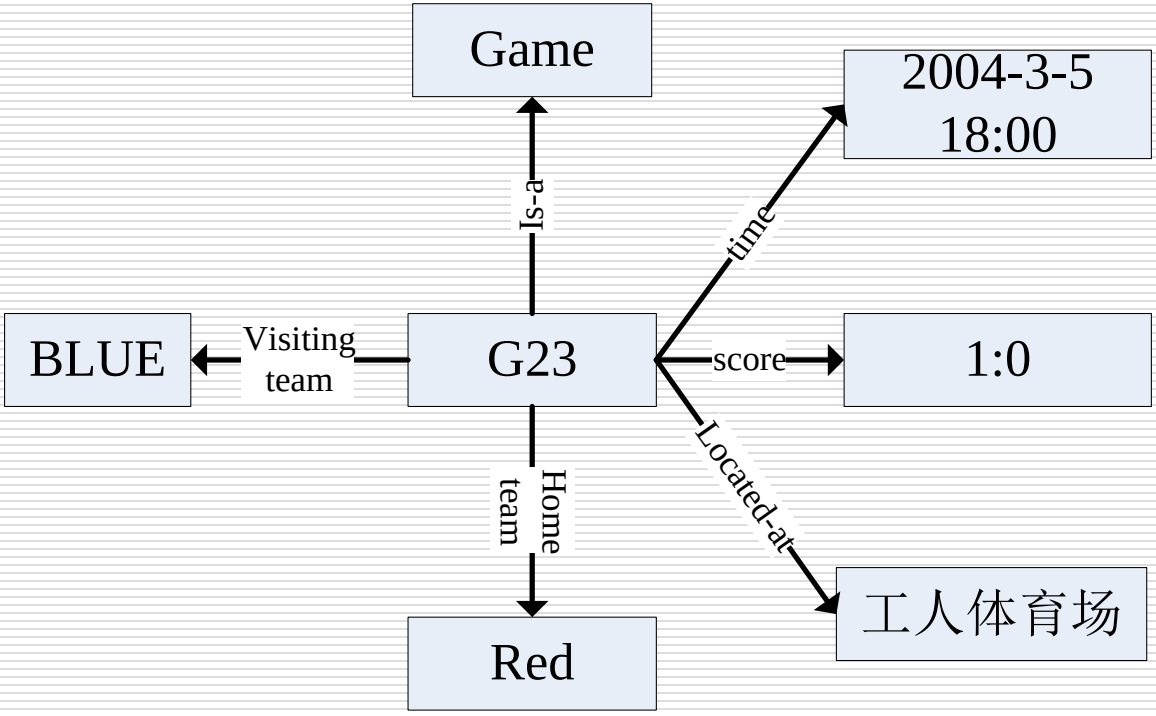


合取、析取关系

- 表示：引入合取、析取节点
- 表示节点之间的合取与析取的关系



语义网络的复杂例子

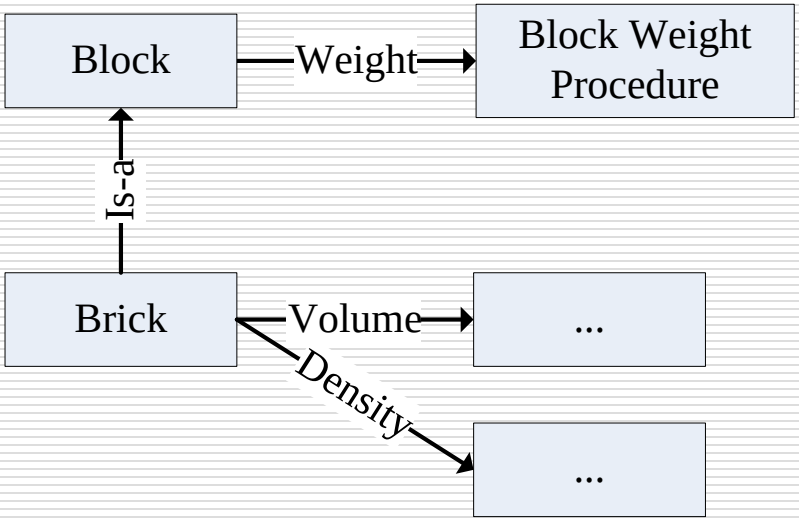

Game
2004-3-5
18:00
Is-a
time
BLUE
Visiting
team
G23
score
1:0
Home
team
Located-at
工人体育场
Red

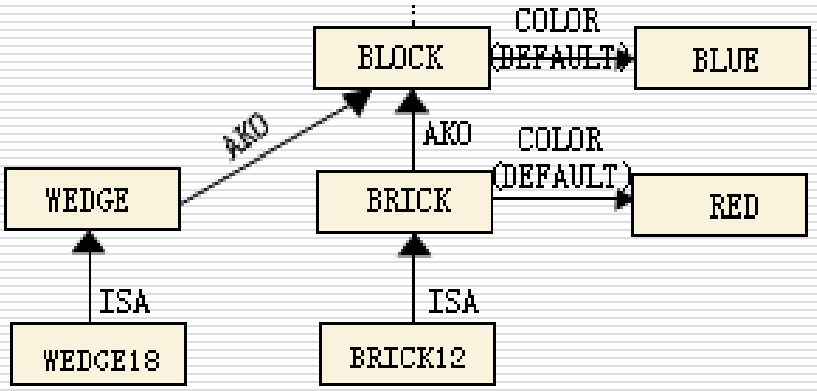
语义网络中的推理

- 继承
  - 将节点的属性由抽象节点传递到具体节点
  - 三种类型的继承
    - 值继承：将抽象节点的属性传递到具体节点
      - 属性值：数值，非数值
    - 如果需要继承：可以从抽象节点那里继承属性的计算方法，再根据其它属性值计算该属性值
    - 默认继承
      - 默认值：将具有相当程度的真实性，但有不能够十分肯定的值；例如：法官是诚实的；宝石是昂贵的；
      - 默认继承：将默认值从抽象节点继承到具体节点

继承的例子

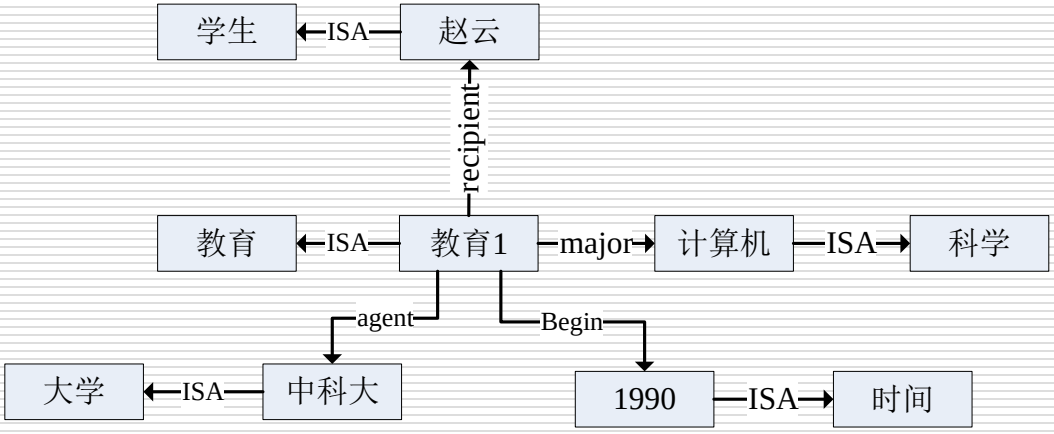

Robin
Is-a
bird
Is-a
animal

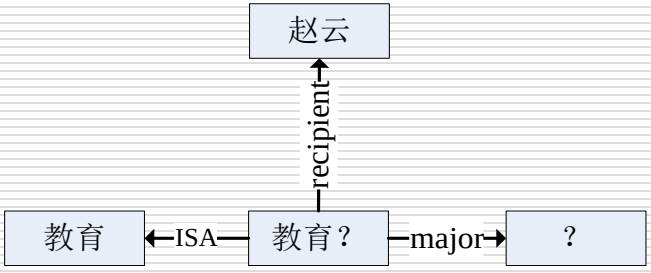

Block
Weight
Block Weight Procedure
Is-a
Brick
Volume
...
Density
...


BLOCK
COLOR
(DEFAULT)
BLUE
AKO
AKO
COLOR
(DEFAULT)
WEDGE
BRICK
RED
ISA
ISA
WEDGE18
BRICK12

语义网络中的推理

- 匹配
  - 知识库使用语义网络表示
  - 将待解的问题形成语义网络片断
  - 将语义网络片断与网络进行匹配，如果成功，则可以获得问题的解





语义网络的特点

- 优点
  - 结构性：能够有机地表示结构性知识
  - 联想性：能够体现联想思维过程
  - 信息共享：共享抽象节点的公共属性
  - 自然性：符合人们表达事物间关系的习惯
- 缺点
  - 不能够保证推理的严密性和有效性
  - 不便于表达深层次的知识

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法
- 框架表示法
- 脚本方法
- 过程表示

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法
- 框架表示法
  - 脚本方法
  - 过程表示

表示方法—框架表示法

- 概述
  - 1975年 Minsky在论文中提出了框架理论。他从心理学的证据出发，认为人脑已存储有大量的典型情景，当人们面临新的情景时，就从记忆中选择一个称作框架的基本知识结构，这个框架是以前记忆的一个知识空框，而其具体内容依照新的情景而改变，对这空框的细节加工修改和补充，形成对新情景的认识又记忆于人脑中。

表示方法—框架表示法

概述

- 人类对于一件事的了解，表现在对于这件实物的诸方面，即属性的了解。掌握了事物的属性，也就有了关于事物的知识，知识表示是从属性描述开始的。

- 槽和填槽表示方法便于表示这种分类知识。这种表示方法包括语义网络、框架、概念从属和脚本。语义网络方法的特点就在于提出了槽和填槽的结构。

表示方法—框架表示法

- 定义
  - 框架是由若干个结点和关系（统称为槽）构成的网络。是语义网络的一般化形式的一种结构。同语义网络没有本质的区别。如书上的所示如将语义网络结点间弧上的标注也放到槽内就成了框架表示形式。
- 表示形式：
  - 由框架名、槽名、侧面、值组成。

表示方法—框架表示法

性质

- 对事物进行描述。而且对其中某些细节做进一步描述。则可将其扩充为另外一些框架。

  如：汽车载货或人

- 可以通过它对一些从感官中没有直接得到的信息进行预测，对于人来说这种功能是很强的。如：一想到桌子就可以想到它腿的形状与位置。
- 可以在它基础上进行判断推理。
- 可通过它来认识某一类事物。
- 可以通过一系列实例来修正框架对某些事物的不完整描述。（填充空的框架，修改默认值）

框架的一般形式

- 构成
  - 框架名：用于表示某个概念、对象或事件
  - 若干的槽：每一个槽用于表示概念某方面的属性，例子：教室的窗户、讲台等
    - 槽名
    - 槽值
  - 每一个槽分成许多侧面
    - 侧面名
    - 侧面值
    - 几种常见的侧面：值侧面，如果需要侧面，默认侧面

框架的一般形式

- 〈框架名〉
  - 〈槽名1〉
    - 〈侧面名11〉（值111，值112，…）
    - 〈侧面名12〉（值121，值122，…）
    - …
  - 〈槽名2〉
    - 〈侧面名21〉（值211，值212，…）
    - 〈侧面名22〉（值221，值222，…）
    - …

槽值或侧面的值

- 槽值
  - 数值、字符侧面、布尔型侧面
  - 动作、过程

- 槽值之间的约束
  - 框架描述中，还可以给出约束条件，指明槽值之间的应该满足某种约束条件，防止出现不合理情况

侧面的作用如下：

·Value--记载类的个体相应属性的公共值或典型值，作为缺省值；

·If-Needed--在不可能提供统一缺省值的情况下，提供计算函数或推理知识去产生相应属性的一个值，简称执行了If-Needed操作；

·If-Added--当给类的某个体的一个属性赋值或修改时，提供计算函数或推理知识去作必要的后继处理，包括对其它相关槽的赋值和修改处理，以及任何需要的附加处理；简称执行了If-Added操作。

框架的例子

框架名：〈教师〉

姓名：单位（姓、名）

年龄：单位（岁）

性别：范围（男、女）

缺省：男

职称：（教授、副教授、讲师、助教）

缺省：讲师

部门：单位（系，教研室）

住址：〈住址框架〉

工资：〈工资框架〉

开始工作时间：单位（年，月）

截止日期：单位（年，月）

缺省：现在

实例框架的例子

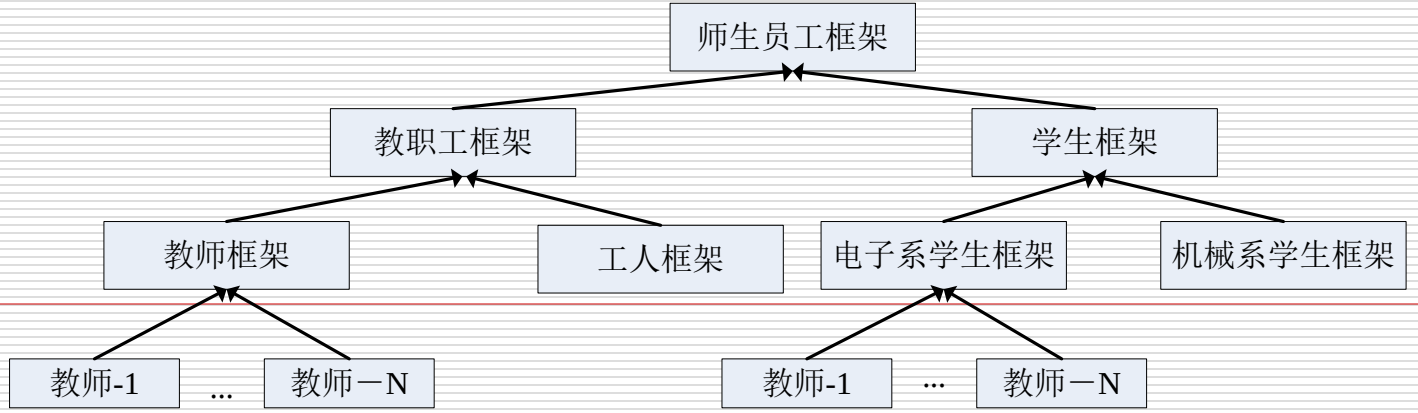
框架名：〈教师－1〉

姓名：夏冰
年龄：36
性别：女
职称：副教授
部门：计算机系软件教研室
住址：〈adr-1〉
工资：〈sal-1〉
开始工作时间：1988.9
截止日期：1996.7

- 实例框架，槽必须给出槽值

框架网络

- 纵向关系
  - 抽象－具体框架之间的关系
  - 例子：从教师、工人和学生框架中抽象出教职员工框架
  - 注意：在抽象框架中出现的属性，在具体框架中无须出现，除非
    - 属性的变异
    - 属性的细化
    - 属性的补充
  - 纵向关系的体现：下层框架中设立一个专用的槽，其槽值指向上层框架



框架网络

- 横向关系
  - 表现:某一个槽值为另外一个框架的名字
  - 非纵向关系的框架之间的联系

- 框架网络
  - 多个互连的框架连接起来构成的框架系统称之为框架网络

框架表示的问题求解过程

- 通过匹配和填槽来实现
  - 问题框架
  - 匹配
    - 匹配过程：问题框架与知识库中的框架的槽值逐个比较，相等或相容
    - 匹配的结果：与知识库中的框架进行匹配，指出一个或几个预选框架，搜索进一步的信息，接收一个或几个框架
  - 填槽：根据匹配出的框架，将问题框架中的未知槽值填上槽值，形成问题的解

框架表示的特点

- 结构性
  - 适合表达知识之间的宏观结构关系
- 继承性
  - 表示抽象概念和具体概念之间的继承关系
  - 下层框架可以继承上层框架的值
  - 也可以对上层框架进一步修改和补充
- 自然性
  - 与人的思维规律有相似之处
- 不足：不太适合表示过程性知识

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法
- 框架表示法
  - 脚本方法
  - 过程表示

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法
  - 框架表示法
  - 脚本方法
  - 过程表示

脚本表示法

- 框架知识表示法的一种特例，也是由一组**固定的槽**构成
- 可用来描述特殊范围内的事件发生的序列
- 1975年，shank提出，由概念依赖理论提出的一种知识表示法

- 概念依赖理论
  - 将现实生活中发生的故事情节分解澄一个个不能再分解的基本概念，称之为原子概念，确定这些原子概念间的依赖关系
  - 然后将所发生的故事都用这些原子概念表示出来
  - 遇到问题，从知识库中找出剧本，按照剧本中事件发生的序列理解故事

原子概念

- 对原子概念的要求
  - 不能有二义性
  - 相互独立
- Shank抽取的关于动作的一些原子概念

✓Propel
✓Grasp
✓Move
✓Atrans
✓Ptrans
✓Attend
✓Ingest
✓Expel
✓Speak
✓Mtrans
✓Mbuild

脚本

- 描述特定范围内事件的结构和顺序
- Shank用这11个原子动作表示生活中发生的事件，形成脚本
- 一个脚本对应一类事件

- 脚本的应用
  - 发生了一个事件，就找出一个与之匹配的脚本，并按照脚本来理解故事

脚本的结构

1. 进入条件：事件发生的前提条件
2. 角色：事件中可能出现的人物
3. 道具：事件中可能出现的物体
4. 场景：用于描述时间发生的序列，有若干场景
5. 结局：事件发生后，一定满足条件的结果

脚本槽的构成是固定的

场景的槽值是由原子动作的序列构成的

例子：餐厅脚本

- **开场条件**
  (a)顾客饿了，需要进餐；　　(b)顾客有足够的钱
- **角色**
  顾客，服务员，厨师，老板
- **道具**
  食品，桌子，菜单，钱
- **场景**
  场景1 进入餐厅
  (a) 顾客走入餐厅。(b) 寻找桌子。(c) 在桌子旁坐下
  场景2 点菜
  (a) 服务员给顾客菜单。(b) 顾客点菜。(c) 顾客把菜单还给服务员。(d) 顾客等待服务员送菜。

例子：餐厅脚本

场景3 等待

（a）服务员把顾客所点的菜告诉厨师。（b）厨师做菜。

场景4 吃菜

（a）厨师把做好的菜给服务员。（b）服务员给顾客送菜。（c）顾客吃菜。

场景5 离开

（a）服务员拿来帐单。（b）顾客付钱给服务员。（c）顾客离开餐厅。

- **结果**

（a）顾客吃了饭，不饿了。（b）顾客花了钱。（c）老板挣了钱。（d）餐厅食品少了。

脚本的应用与特点

- 脚本用于描述一类事件
- 当发生了某个事件，从脚本库中找出与之匹配的脚本
- 按照脚本的内容（动作的序列）理解事件
  - 预测没有显式提及的事件的发生
  - 给出已明确提到事件的关系

- 特点
  - 比较适合表示动作的序列
  - 比较呆板，多用于自然语言理解

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法
  - 框架表示法
  - 脚本方法
  - 过程表示

表示方法

- 概述
- 直接表示
- 逻辑表示
- 产生式规则表示法
- 语义网络表示法
  - 框架表示法
  - 脚本方法
  - 过程表示

表示方法—过程表示法

- 前面的几种知识表示方法均是知识和事实的一种静止的表示方法。我们称这类知识表示方式为陈述式表达。它所强调的是事物所涉及的对象是什么，是对事物有关知识的静态描述，是知识的一种显式、说明性知识表达形式。
- 说明性表示知识给出事物本身的属性及事物之间的相互关系。对问题的解答就隐含在这些知识之中。而过程性知识则给出解决一个问题的具体过程。

表示方法—过程表示法

- 说明性知识和过程性知识相比：
  - 说明性知识比较简要、清晰、可靠、便于修改。但往往效率低。
  - 过程性知识比较直截了当，效率高。但由于详细地给出了解决过程，使这种知识表示显得复杂、不直观、容易出错、不便于修改。
  - 实际上，说明性表示和过程性表示实际上没有绝对的分界线。因此，任何说明性知识如果要被实际使用，必须有一个相应的过程去解释执行它。对于一个以使用说明性表示为主的系统来说，这种过程往往是隐含在系统之中，而不是面向用户。

表示方法—过程表示法

- 知识过程性的两个含义：
  - 含义1：把解决一个问题的过程描述出来。可以称它为解题知识的过程表示。
  - 含义2：把客观事物的发展过程用某种方式表示出来。
  - 在某些情况下，这两种含义是很难决然分开的。如，任何一个解题系统的基本构成都是一个数据集，一组运算符和一个解释程序。过程性知识使用状态来表示，在状态空间运作。

表示方法—过程表示法

- 过程式表示定义：
  - 过程式表示就是将有关某一问题领域的知识连同如何使用这些知识的方法均隐式地表达为一个求解过程。
  - 它所给出的是事物的一些客观规律，表达的是如何求解问题，知识的描述形式就是程序。所有信息均隐含在程序中——效率高、没有固定形式。
  - 如何描述知识完全取决定于具体的问题。

实际上的系统都是陈述与过程观点的结合。陈述之中多少包含了过程方法。

知识表示

- 结论：

本章介绍了若干种知识表达方式，绝大多数在应用中得到了很好的验证。但实际工作中，如果要建立一个人工智能系统、专家系统时，可能还是要根据具体情况提出一个混合性的知识表达方式。

小结

知识是一切智能行为的基础，也是人工智能的重要研究对象。要使计算机具有智能，就必须使它具有知识，而要使计算机具有知识，就必须解决知识表示的问题。

本章就知识和知识表示的概念进行了较详细的阐述，对各种知识表示方法进行了简单的介绍。各种不同的知识表示方法没有什么本质的差别，目的无非是使知识的运用更合理和高效。一般而言，一个智能系统都由推理系统和知识库构成，知识表示的优劣直接影响推理的进行，二者是相辅相成的。因此，选择什么样的知识表示法时，不仅要考虑知识本身的结构，还要考虑推理系统的构成。

作业

- 3.7
- 3.9
- 3.16
- 3.21