第二章 搜索技术

- 内容提要：
- 状态空间法
- 问题归约法

知识表示

- 盲目搜索
- 启发式搜索
- 与或图搜索

问题求解

1

前言

- 在学习本章内容之前，我们先了解一下有关知识及其表示的概念。

  人类的智能活动过程主要是一个获得并运用知识的过程，知识是智能的基础。为了使计算机具有智能，就必须使它具有知识。

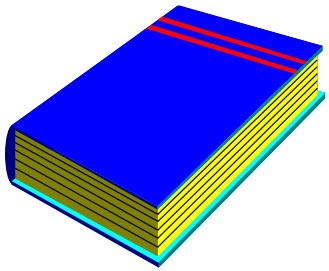
  那什么是知识呢？

2

知识一般概念

- 知识是人们在改造客观世界的实践中积累起来的认识和经验
- 认识：包括对事物现象、本质、属性、状态、关系、联系和运动等的认识
- 经验：包括解决问题的微观方法和宏观方法
- 微观方法：如步骤、操作、规则、过程、技巧等
- 宏观方法：如战略、战术、计谋、策略等
- 知识的有代表性的定义
- (1) Feigenbaum：知识是经过剪裁、塑造、解释、选择和转换了的信息
- (2) Bernstein：知识由特定领域的描述、关系和过程组成
- (3) Heyes-Roth：知识=事实+信念+启发式
- 知识、信息、数据及其关系
- 数据是信息的载体，本身无确切含义，其关联构成信息
- 信息是数据的关联，赋予数据特定的含义，仅可理解为描述性知识
- 知识可以是对信息的关联，也可以是对已有知识的再认识
- 常用的关联方式：if …… then ……

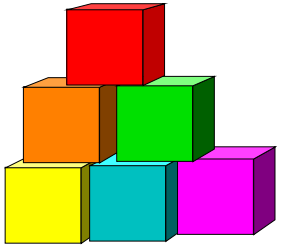
什么是知识?



- 一般来说，我们把有关信息关联在一起所形成的信息结构称为知识。
- 知识表示就是对知识的一种描述，一种计算机可以接受的用于描述知识的数据结构。

4

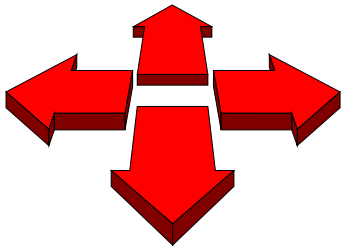
知识的要素



- 知识的要素是指构成知识的必需元素。在这里，我们关心的是一个人工智能系统所处理的知识的组成成分。一般而言，人工智能系统的知识包含事实、规则、控制和元知识。

5

知识的要素



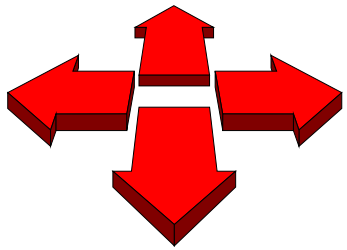
- 事实：事物的分类、属性、事物间关系、科学事实、客观事实等. 是有关问题环境的一些事物的知识，常以“---是---”形式出现，也是最低层的知识。例如：雪是白色的，人有四肢。

- 规则：事物的行动、动作和联系的因果关系知识。

这种知识是动态的，常以“如果---那么---”形式出现。例如启发式规则，如果下雨，则出门带伞。

6

知识的要素



- 控制：是有关问题的求解步骤、规划、求解策略等技巧性知识，告诉怎么做一件事。也包括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。
- 元知识：怎样使用规则、解释规则、校验规则、解释程序结构等知识。是有关知识的知识，是知识库中的高层知识。元知识与控制知识有时有重叠。

7

- 每种以**知识**和**符号操作**为基础的智能系统，其问题求解方法都需要某种**对解答的搜索**。
- 在搜索过程开始之前，必须先用某种方法或某几种方法的混和来表示问题。
- 问题求解技术主要涉及两个方面：

□ **问题的表示**

求解的方法

- **知识表示**方式是学习人工智能的中心内容之一。

8

知识表示的概念

- 什么是知识表示
- 是对知识的描述，即用一组符号把知识编码成计算机可以接受的某种结构。其表示方法不唯一。
- 知识表示的要求
- 表示能力：能否正确、有效地表示问题。包括：
- 表示范围的广泛性
- 领域知识表示的高效性
- 对非确定性知识表示的支持程度
- 可利用性：可利用这些知识进行有效推理。包括：
- 对推理的适应性：推理是根据已知事实利用知识导出结果的过程
- 对高效算法的支持程度：知识表示要有较高的处理效率
- 可实现性：要便于计算机直接对其进行处理
- 可组织性：可以按某种方式把知识组织成某种知识结构
- 可维护性：便于对知识的增、删、改等操作
- 自然性：符合人们的日常习惯
- 可理解性：知识应易读、易懂、易获取等

知识表示的一般方法

- 状态空间法
- 问题归约法
- 谓词逻辑法
- 语义网络
- 另外还有框架表示以及剧本表示，过程表示.
- 在表示和求解比较复杂的问题时，采用单一的表示方法是不够的，往往采用多种方法的混合表示.目前这仍是人工智能专家感兴趣的研究方向.

10

状态空间法

问题求解(problem solving)是个大课题，它涉及归约、推断、决策、规划、常识推理、定理证明和相关过程的核心概念。在分析了人工智能研究中运用的问题求解方法之后，就会发现许多问题求解方法是采用**试探搜索方法**的。也就是说，这些方法是**通过在某个可能的解空间内寻找一个解来求解问题的**。这种基于解答空间的问题表示和求解方法就是**状态空间法**，它是以**状态**和**算符**（operator）为基础来表示和求解问题的。

11

状态空间法

- 状态空间法

  状态 (State)

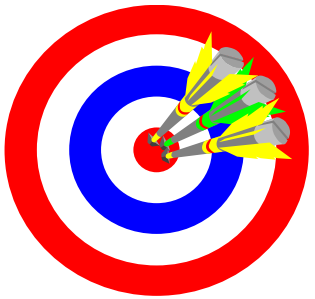
  算符 (Operator)

  状态空间方法 (Method on State Space)

□

12

状态

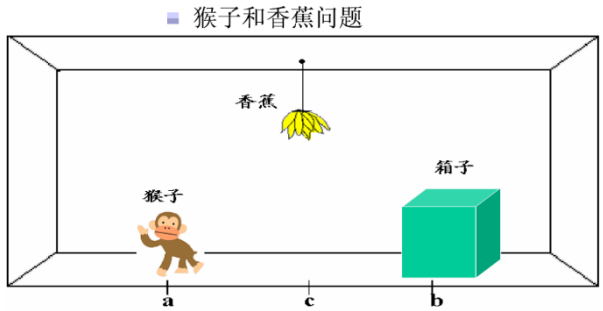


- **状态**（state）：为描述某类不同事物间的差别而引入的一组最少变量 q_0 q_1，…，q_n的有序集合.

  矢量形式： $Q=[q_0, q_1, \cdots q_n]^T$

□ 式中每个元素q_i为集合的分量，称为**状态变量**。

给定每个分量的一组值就得到**一个具体的状态**，如 $Q_k=[q_{0k}, q_{1k}, \cdots, q_{nk}]$

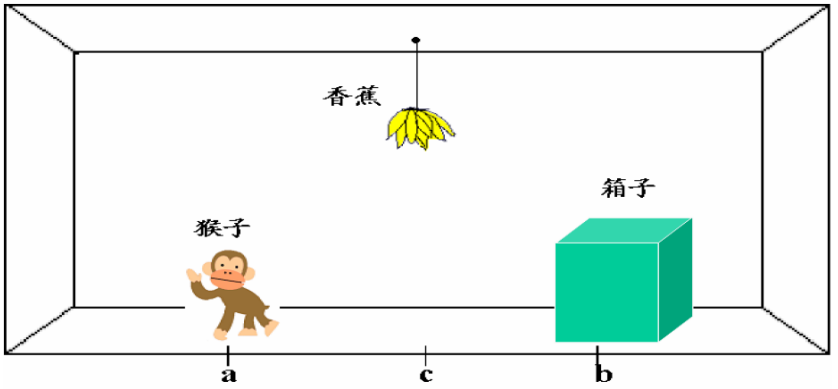


13

例：

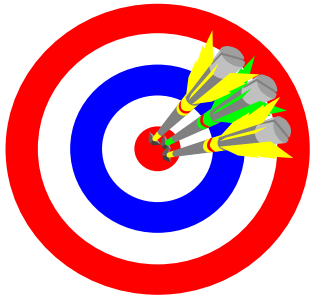
- 用一个四元表列（W，x，Y，z）来表示问题状态.
- 其中：W-猴子的水平位置；x－当猴子在箱子顶上时取x=1；否则取x=0；Y－箱子的水平位置；z-当猴子摘到香蕉时取z=1；否则取z=0。

猴子和香蕉问题

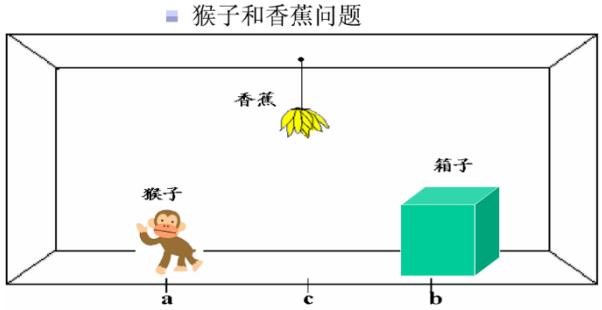


14

算符

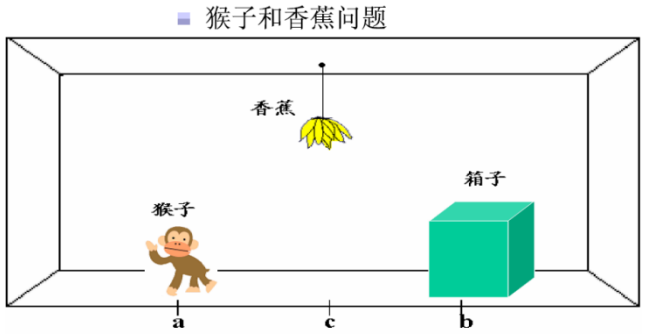


- **算符**（operator）：**把问题从一种状态变换为另一种状态的手段.**
- **算符可为走步、过程、规则、数学算子、运算符号或逻辑符号等。**



15

操作（算符）：



1. goto（U）表示猴子走到水平位置U

   或者用产生式规则表示为：

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

2. pushbox（V）猴子把箱子推到水平位置V，即有：

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

3. climbbox 猴子爬上箱顶，即有：

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

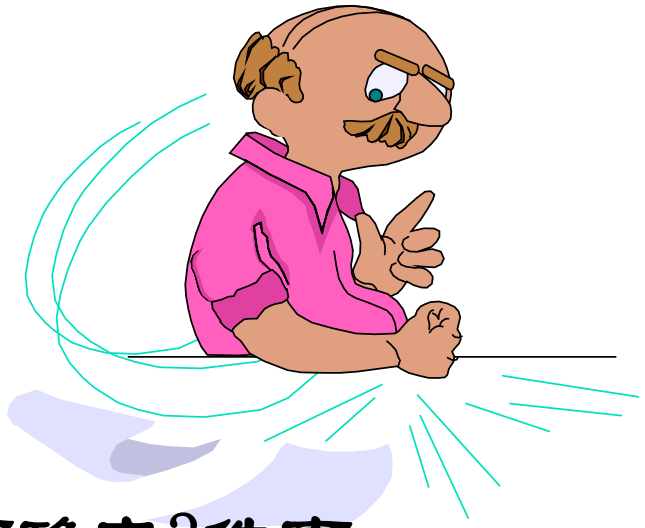
4. grasp 猴子摘到香蕉，即有：

$$(c,\ 1,\ c,\ 0) \xrightarrow{\text{grasp}} (c,\ 1,\ c,\ 1)$$

16

- 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。
- 它包含三种说明的集合，即三元状态（S，F，G）
- S — 初始状态集合；
- F — 操作符集合；
- G —目标状态集合。

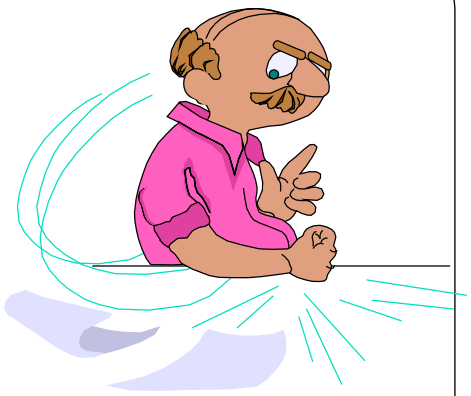
17



- 对一个问题的状态描述，必须确定3件事：
- (1) 该状态描述方式，特别是初始状态描述；
- (2) 操作符集合及其对状态描述的作用；
- (3) 目标状态描述的特性。

18

状态空间表示



- 典型的例子：
- 下棋、迷宫及各种游戏。

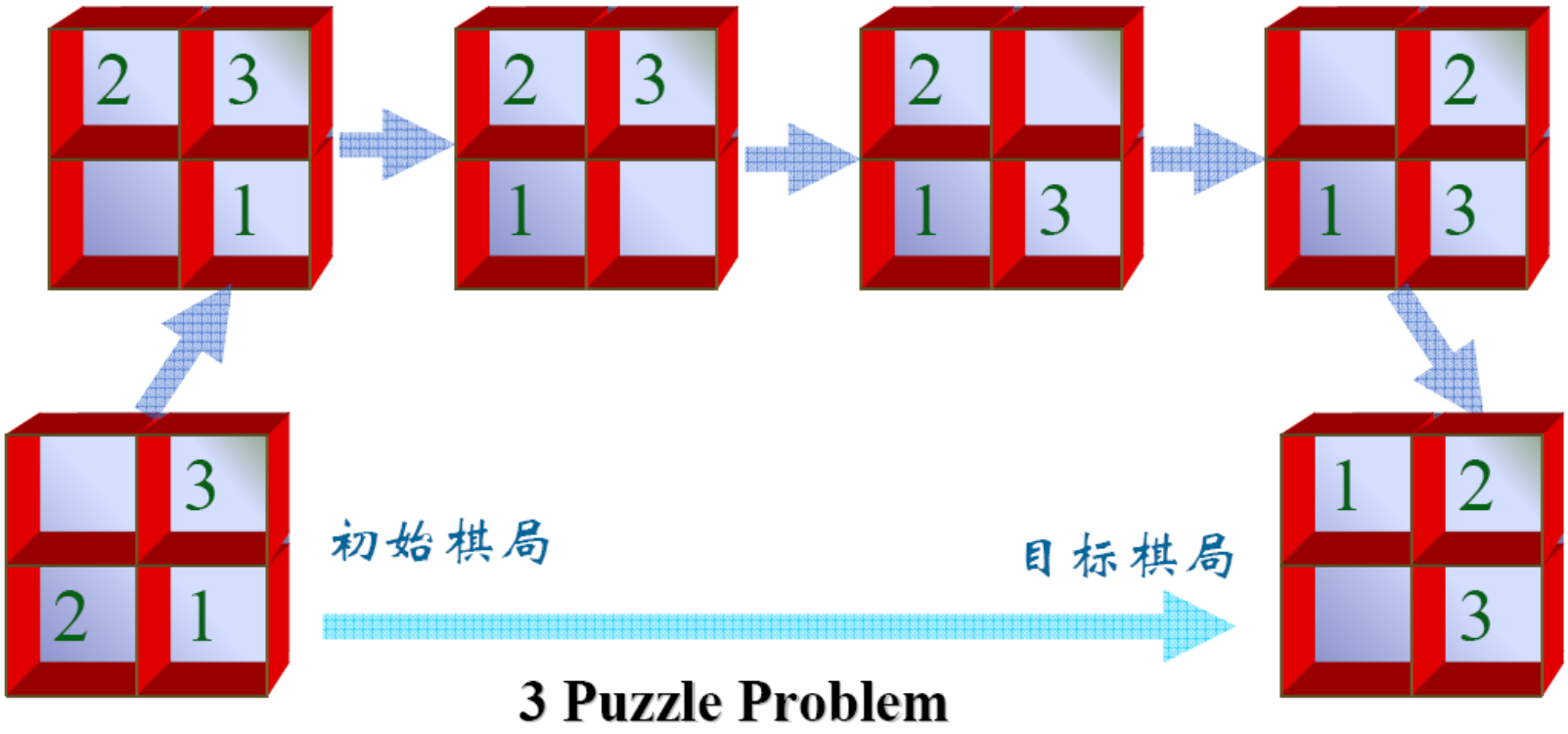
19

三数码难题

- 问题描述：
- 三数码难题由3个编有1-3并放在2×2方格棋盘上可走动的棋子组成。棋盘上总有一个空格，以便让空格周围的棋子走进来。直至从初始状态到达目标状态。

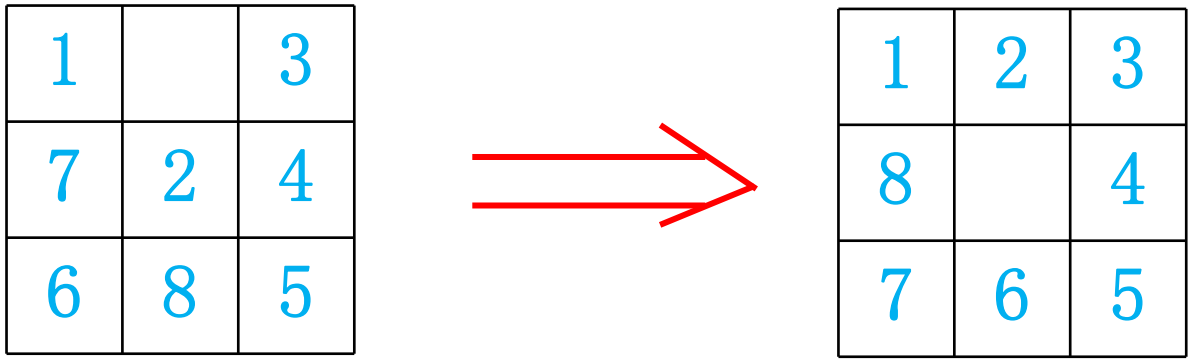
20

三数码难题



21

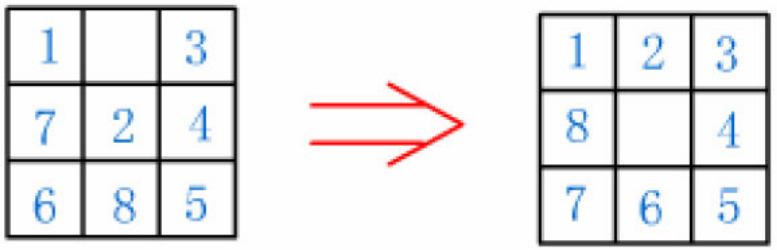
八数码难题



初始棋局　　目标棋局

22

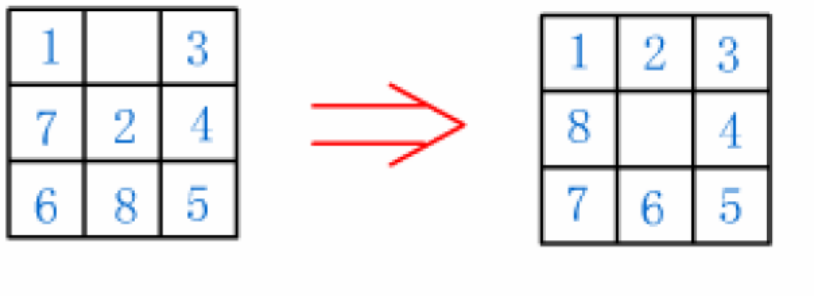
表示



•根据问题状态、操作算符和目标条件选择各种表示，是高效率求解必须的。在问题求解过程中，会不断取得经验，获得一些简化的表示。

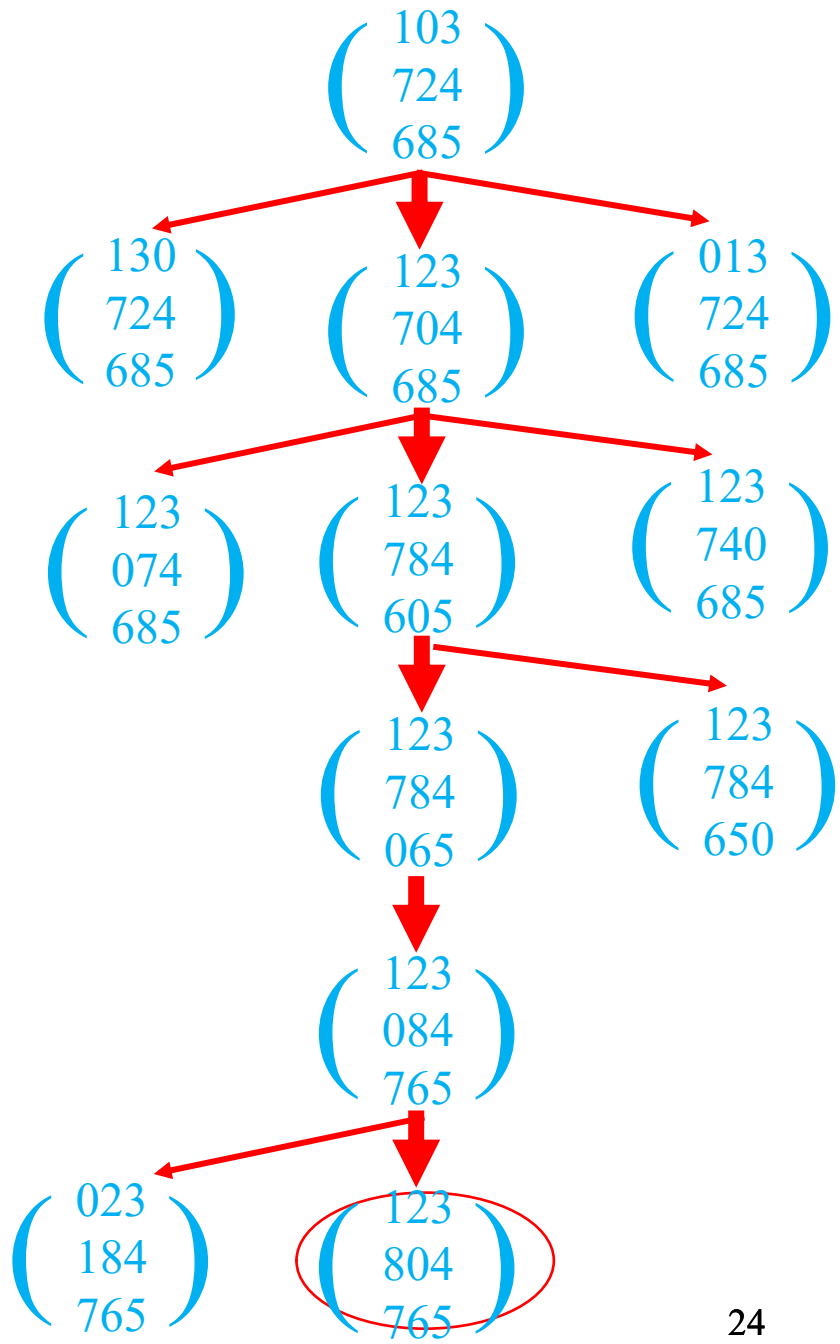
- 制定操作算符集：
- * 直观方法——为每个棋牌制定一套可能的走步：左、上、右、下四种移动。这样就需32个操作算子。
- * 简易方法——仅为空格制定这4种走步，因为只有紧靠空格的棋牌才能移动。
- * 空格移动的唯一约束是不能移出棋盘。

23



从初始棋局开始，试探由每一合法走步得到的各种新棋局，然后计算再走一步而得到的下一组棋局。这样继续下去，直至达到目标棋局为止。

把初始状态可达到的各状态所组成的空间设想为一幅**由各种状态对应的节点组成的图**。这种图称为**状态图**。图中每个节点标有它所代表的棋局。首先把适用的算符用于初始状态，以产生新的状态；然后，再把另一些适用算符用于这些新的状态；这样继续下去，直至产生目标状态为止。



八数码难题部分状态图

十五数码难题(思考)

11	9	4	15
1	3		12
7	5	8	6
13	2	10	14

初始状态

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

目标状态

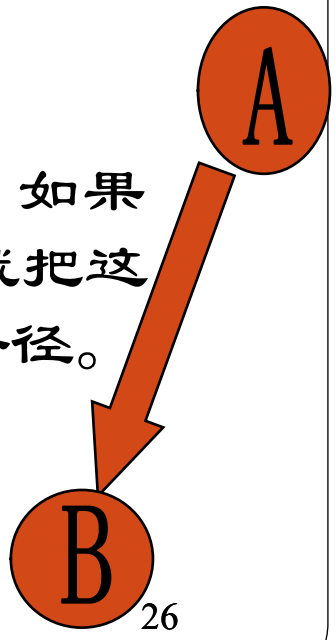
25

状态图示法

有向图(directed graph)

□ 图，由节点（不一定是有限的节点）和边的集合构成。有向图，是指图中的一对节点用弧线连接起来，从一个节点指向另一个节点。

- 路径
- 某个节点序列（$n_{i1}, n_{i2}, , \cdots, n_{ik}$）当j=2, 3, …k时，如果对于每一个$n_{ij-1}$都有一个后继节点$n_{ij}$存在，那么就把这个节点序列叫做从节点$n_{i1}$至节点$n_{ik}$的长度为k的路径。



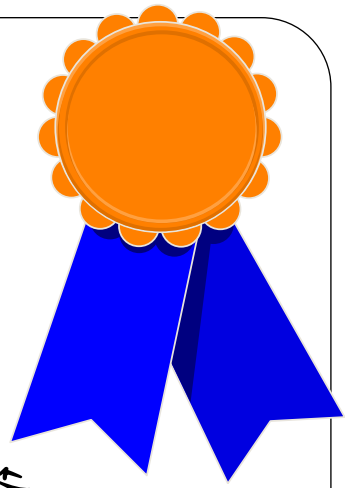
26

状态图示法

- 寻找从一种状态变换为另一种状态的某个算符序列问题就等价于寻求图的某一路径的问题.
- 代价　加在各弧线的指定数值，以表示加在相应算符上的代价。
- 图的显式说明　指各节点及其具有代价的弧线可以由一张表明确给出
- 图的隐式说明　指各节点及其具有代价的弧线不可以由一张表明确给出（起始节点＋后继算符，把后继算符应用于各节点，以扩展节点）

27

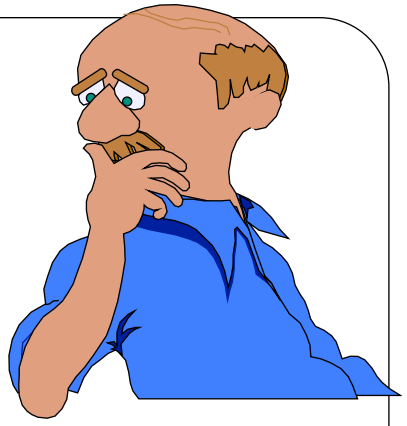
状态图示法



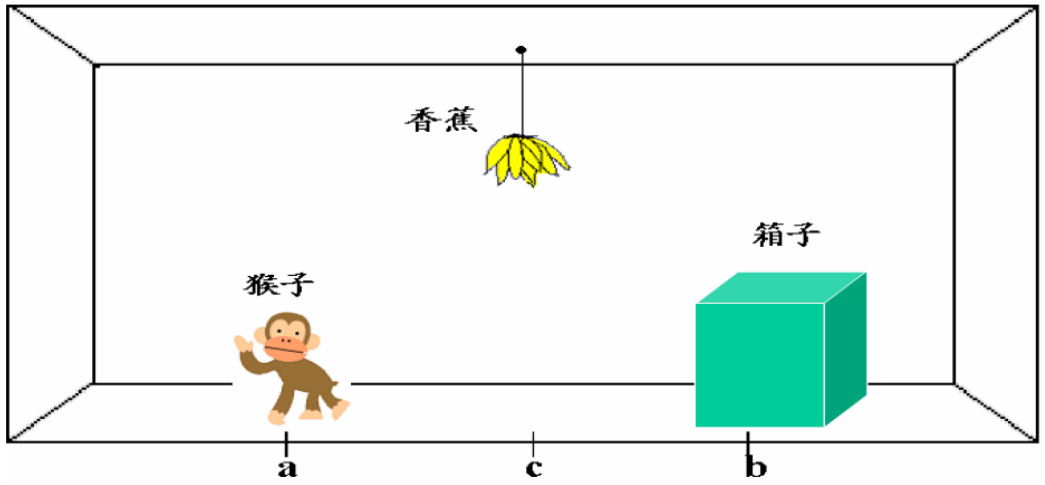
- 搜索某个状态空间以求得算符序列的一个解答的过程，就对应于使隐式图足够大的一部分变为显示图以便包含目标的过程。
- 这样的搜索图是状态空间问题求解的主要基础。

28

状态空间表示举例



- 猴子和香蕉问题

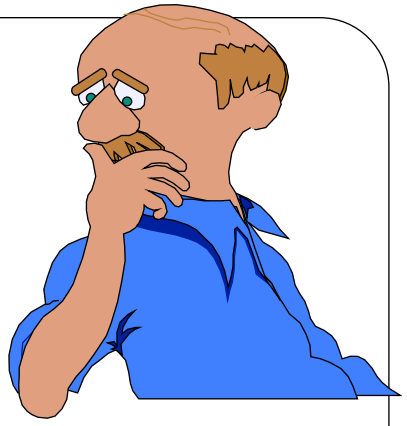


问题描述

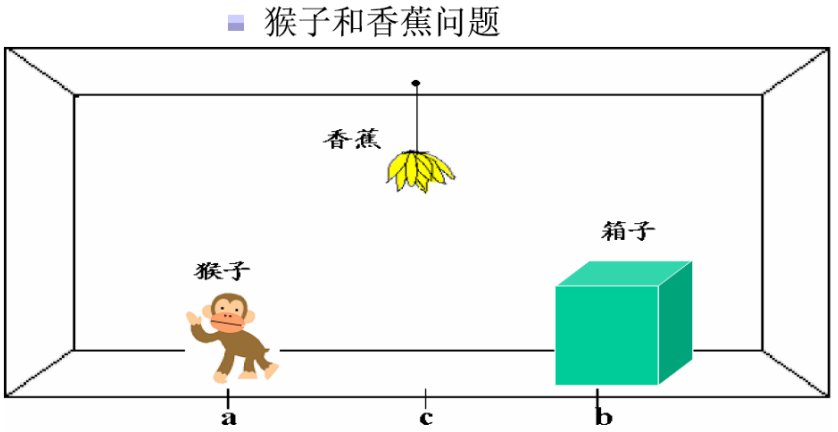
在一个房间内有一只猴子(可把这只猴子看做一个机器人)、一个箱子和一束香蕉。香蕉挂在天花板下方，但猴子的高度不足以碰到它。那么这只猴子怎样才能摘到香蕉呢?

29

解题过程

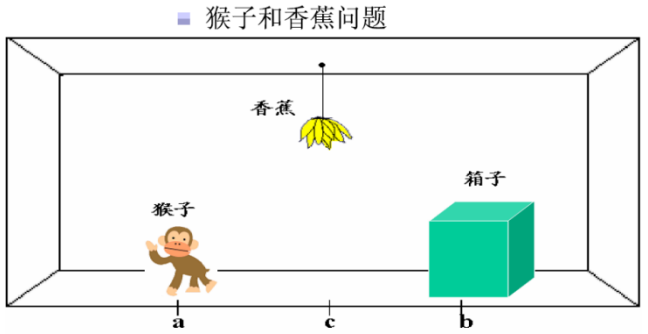


- 用一个四元表列（W，x，Y，z）来表示问题状态.
- 其中：W-猴子的水平位置；x－当猴子在箱子顶上时取x=1；否则取x=0；Y－箱子的水平位置；z-当猴子摘到香蕉时取z=1；否则取z=0。



30

操作（算符）：



1. goto（U）表示猴子走到水平位置U

   或者用产生式规则表示为：

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

2. pushbox（V）猴子把箱子推到水平位置V，即有：

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

3. climbbox 猴子爬上箱顶，即有：

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

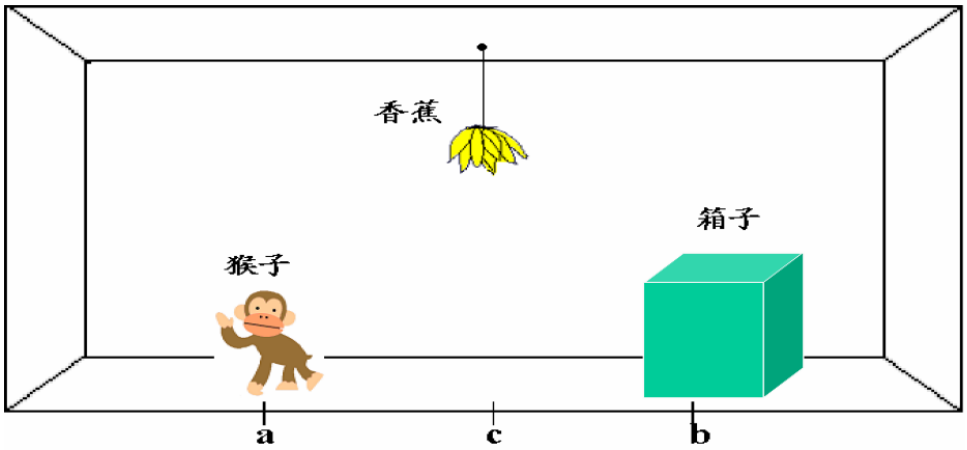
4. grasp 猴子摘到香蕉，即有：

$$(c,\ 1,\ c,\ 0) \xrightarrow{\text{grasp}} (c,\ 1,\ c,\ 1)$$

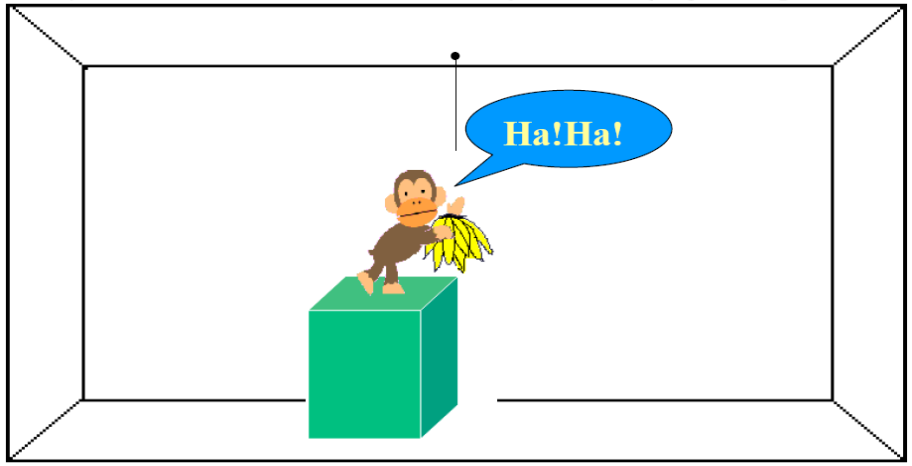
31

- 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

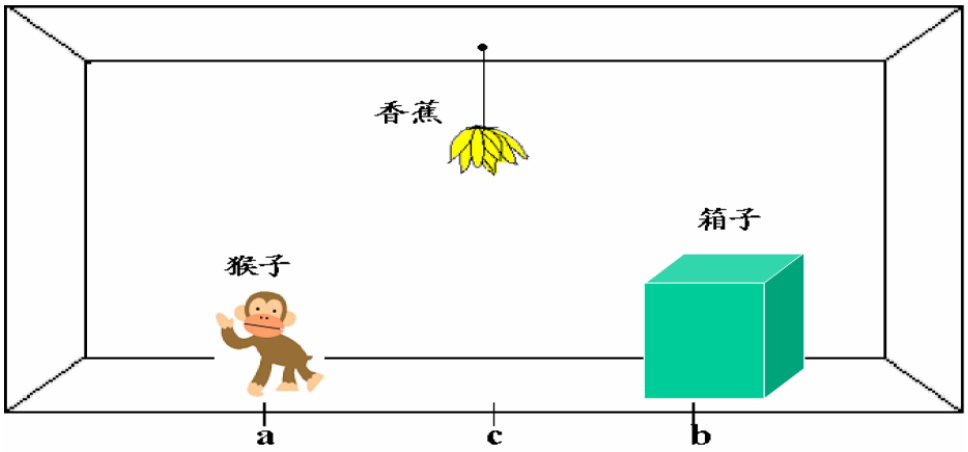


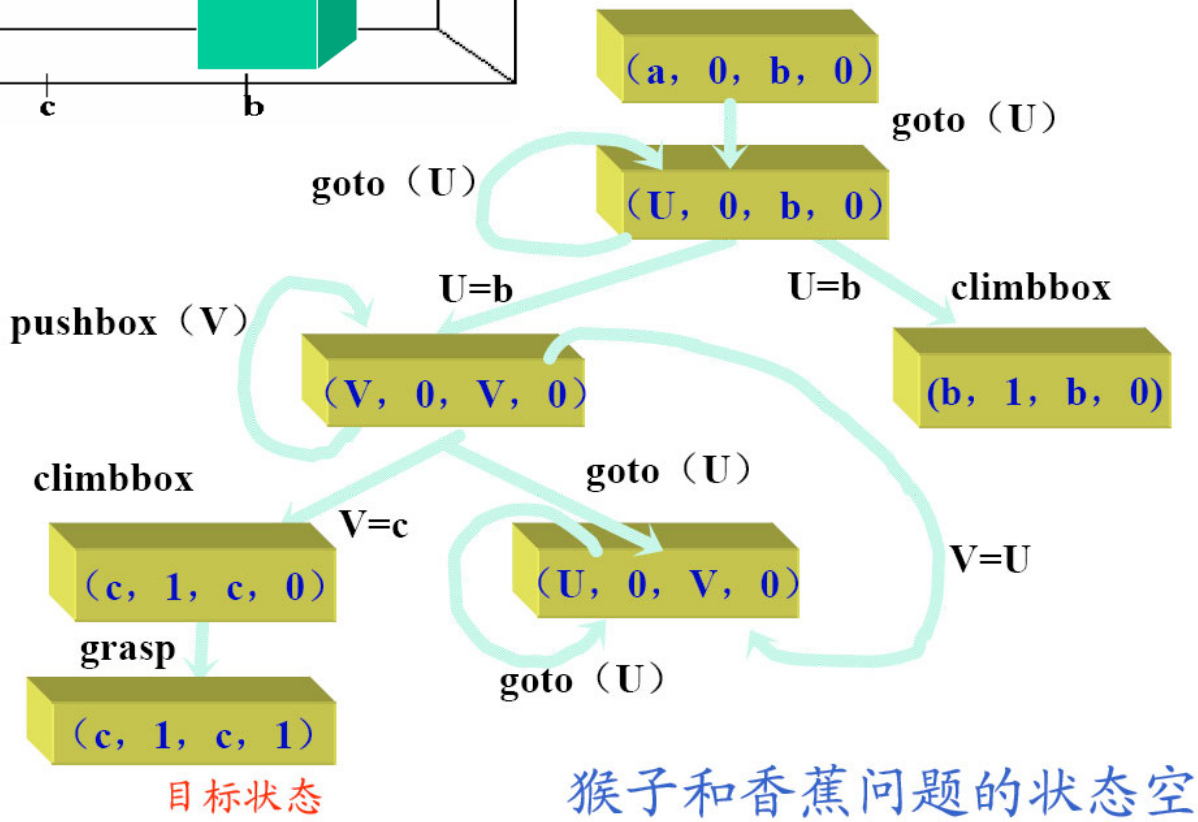
猴子和香蕉问题的演示过程



32

- 猴子和香蕉问题





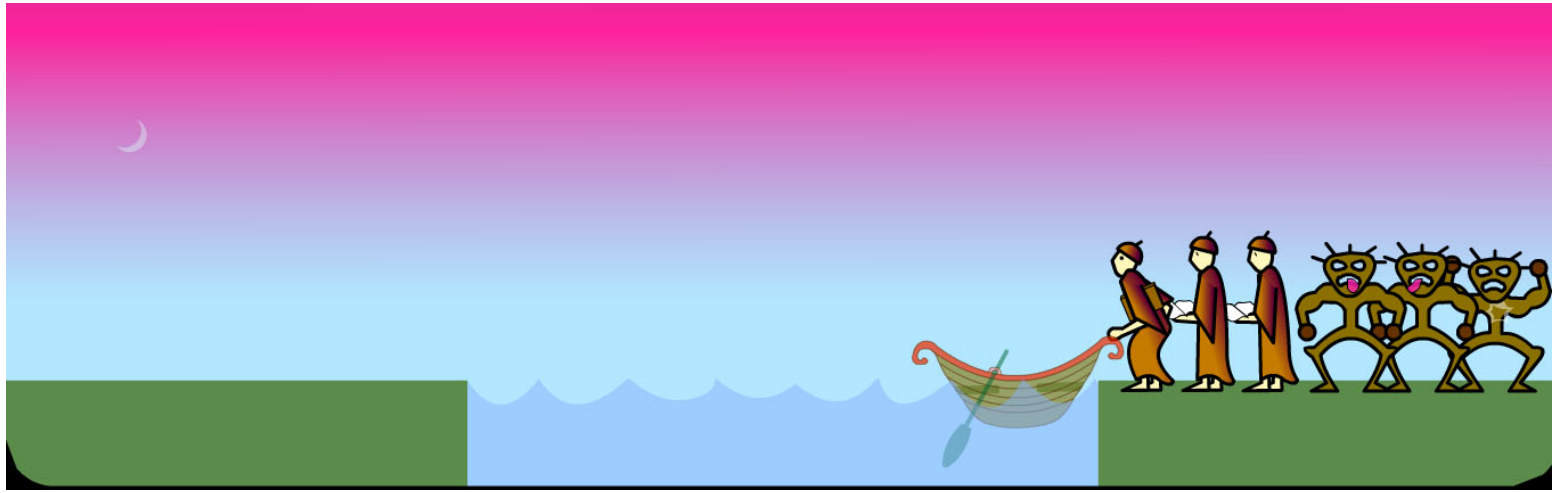
猴子和香蕉问题的状态空间图

33

状态空间表示举例2

传教士野人问题（Missionaries& Cannibals，MC问题）

有三个传教士M和三个野人C过河，只有一条能装下两个人的船，在河的一方或者船上，如果野人的人数大于传教士的人数，那么传教士就会有危险，你能不能提出一种安全的渡河方法呢？



- 状态：问题在某一时刻所处的“位置”，“情况”等
- 根据问题所关心的因素，一般用向量形式表示，每一位表示一个因素

状态可有多种表示方法：

（左岸传教士数，右岸传教士数，左岸野人数，右岸野人数，船的位置）

或

（左岸传教士数，左岸野人数，船的位置）

初始状态：（0，0，0）

目标状态：（3，3，1）

0：右岸

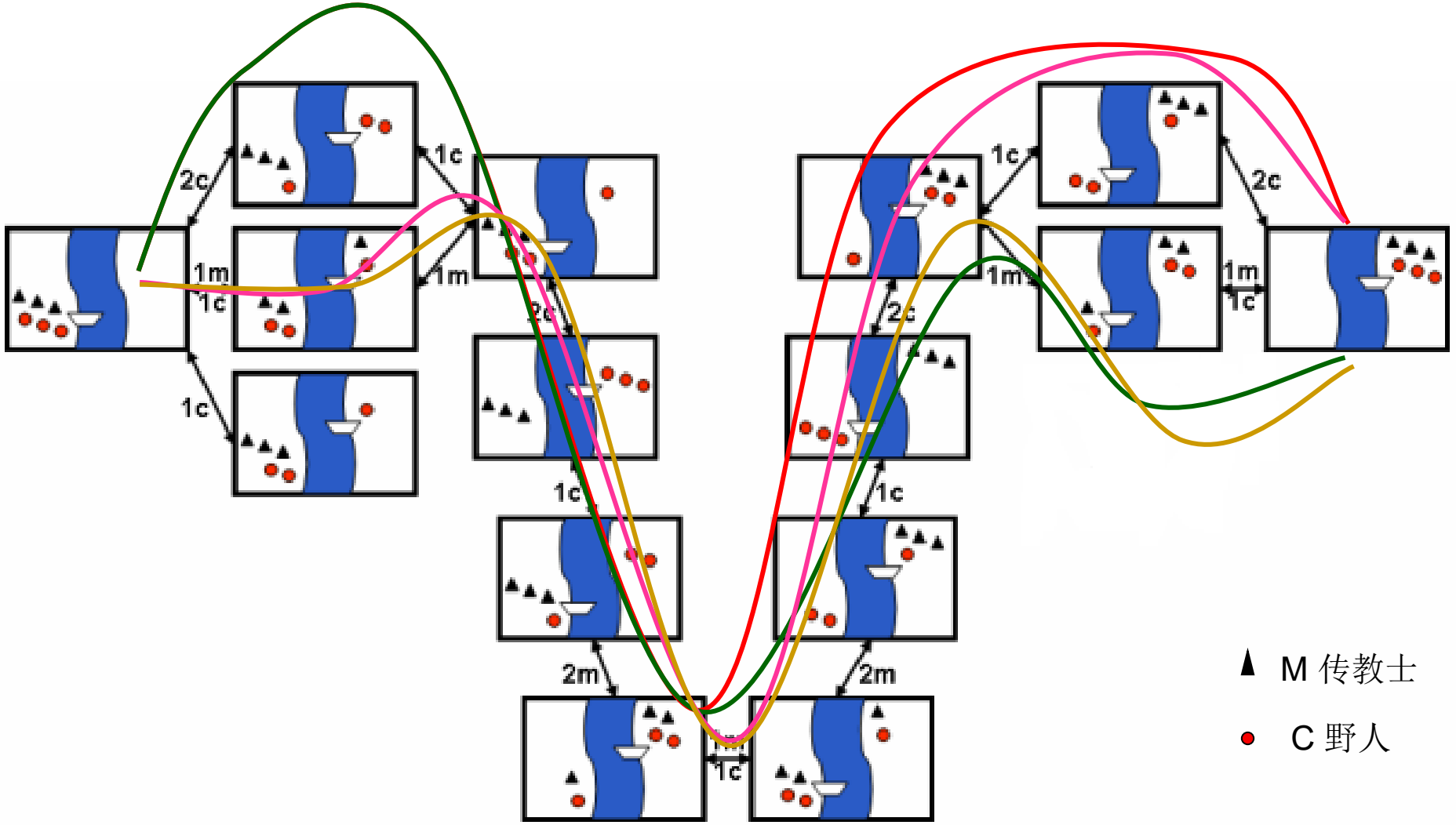
1：左岸

哪些操作能导致状态变化？

状态的转换

- 算子（算符，操作符）——使状态发生改变的操作
- MC问题中的算子
  - 将传教士或野人运到河对岸
  - Move-1m1c-lr：将一个传教士(m)一个野人(c)从左岸(l)运到右岸(r)
  - 所有可能操作
    - Move-1m1c-lr　Move-1m1c-rl　Move-2c-lr
      Move-2c-rl　Move-2m-lr　Move-2m-rl
      Move-1c-lr　Move-1c-rl　Move-1m-lr
      Move-1m-rl

传教士野人问题状态空间图



- 例2.1 推销员旅行问题（旅行商问题）
- 一个推销员计划出访推销产品。他从一个城市（如A）出发，访问每个城市一次，且最多一次，然后返回城市A。要求寻找最短路线。

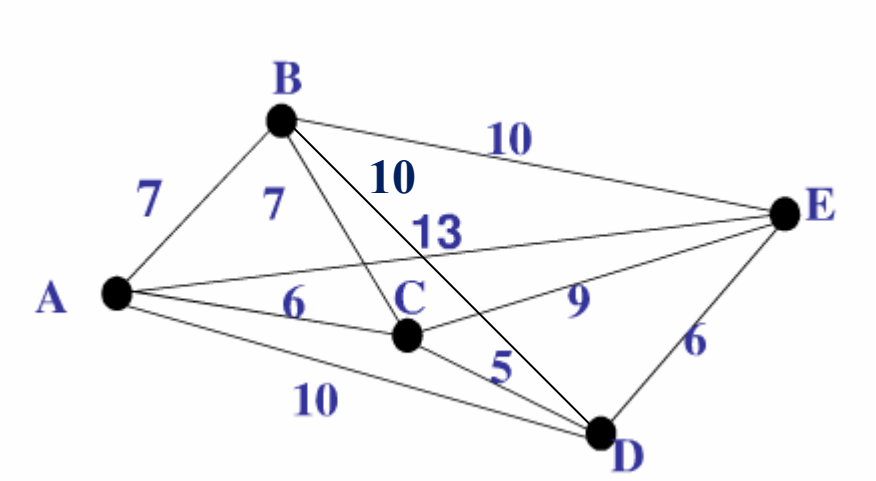
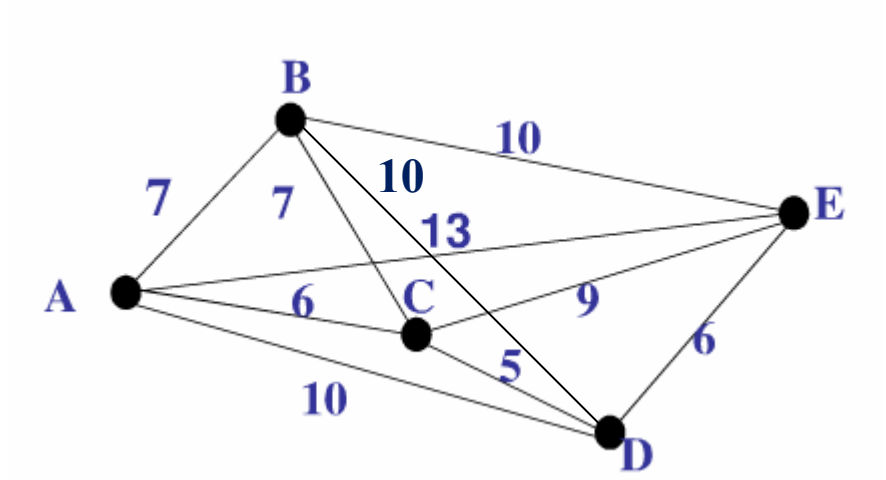


图2.3 推销员旅行问题

- 状态描述：目前为止访问过的城市列表 (A…)
- 初始状态： (A)
- 目标状态： (A……A)



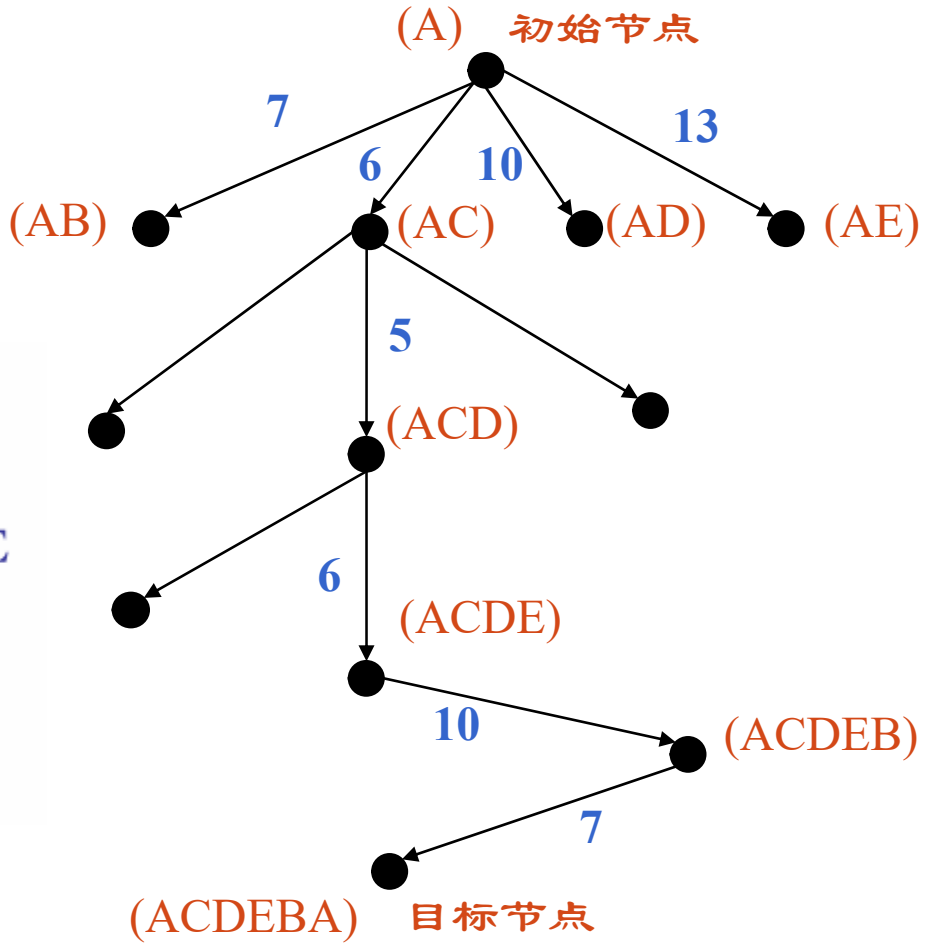
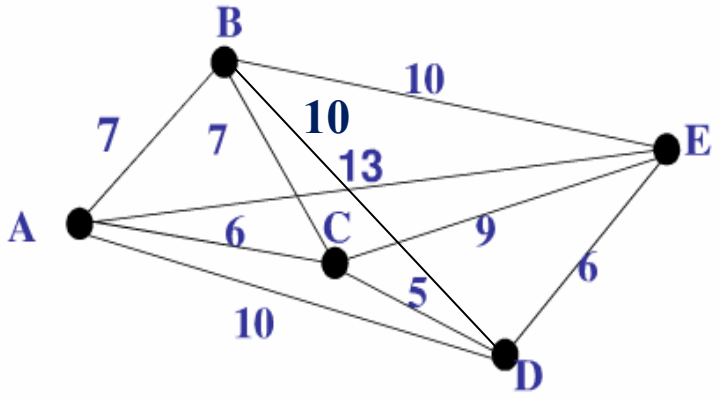


图2.4 推销员旅行问题状态空间图

- 算符：下一步走向的城市(a) (b) (c) (d) (e)
- 约束：每个城市只能走过一次，A除外

课后练习

- 下图表示了五个城市间的交通关系，用状态空间法规划一个最短的旅行路程：此旅程从城市A开始，访问其他城市不多于一次，并返回A。选择一个状态表示，表示出所求得的状态空间的节点及弧线，标出适当的代价，并指明图中从起始节点到目标节点的最佳路径。



问题归约法

- 已知问题的描述，通过一系列变换把此问题最终变为一个子问题集合；这些子问题的解可以直接得到，从而解决了初始问题。
- 该方法也就是从目标(要解决的问题)出发逆向推理，建立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平凡的本原问题集合。这就是问题归约的实质。

42

问题归约法的组成部分

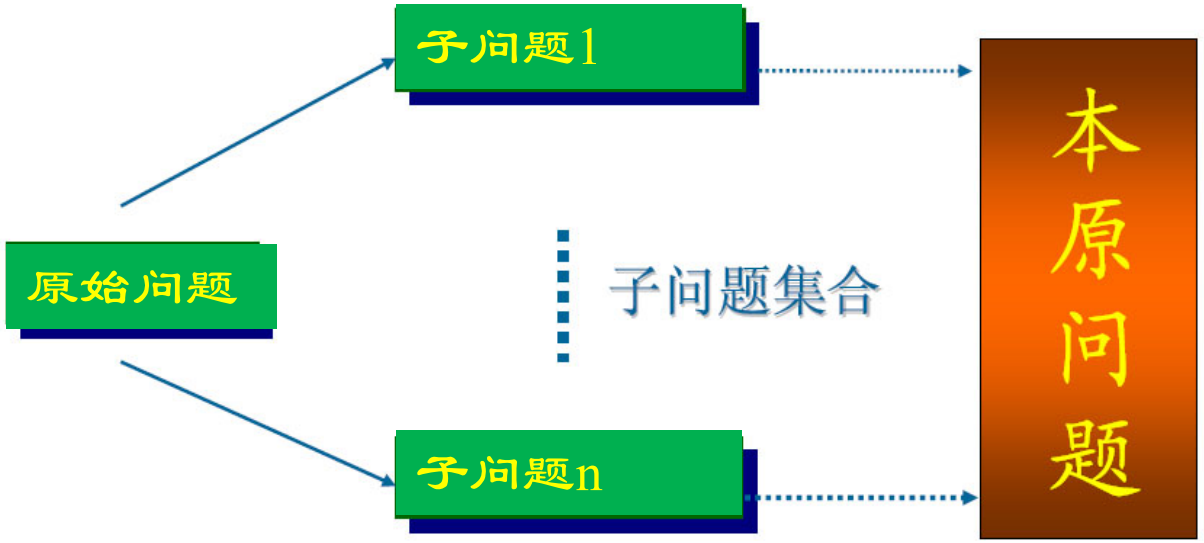
- (1) 一个初始问题描述；
- (2) 一套把问题变换为子问题的操作符；
- (3) 一套本原问题描述。

43

问题规约法图解

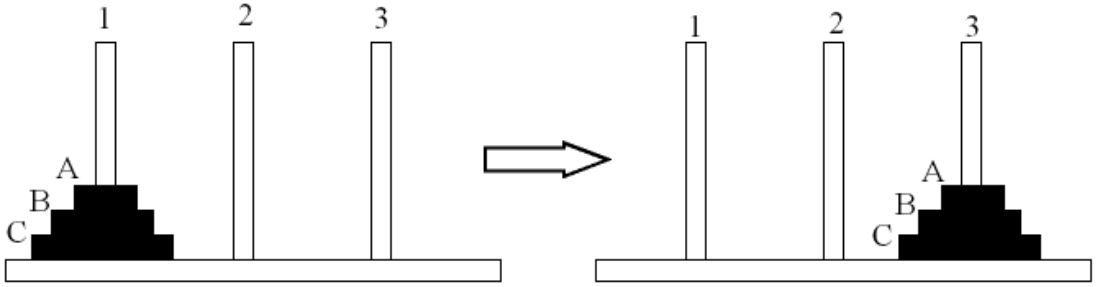
问题归约法

Problem Reduction Representation



44

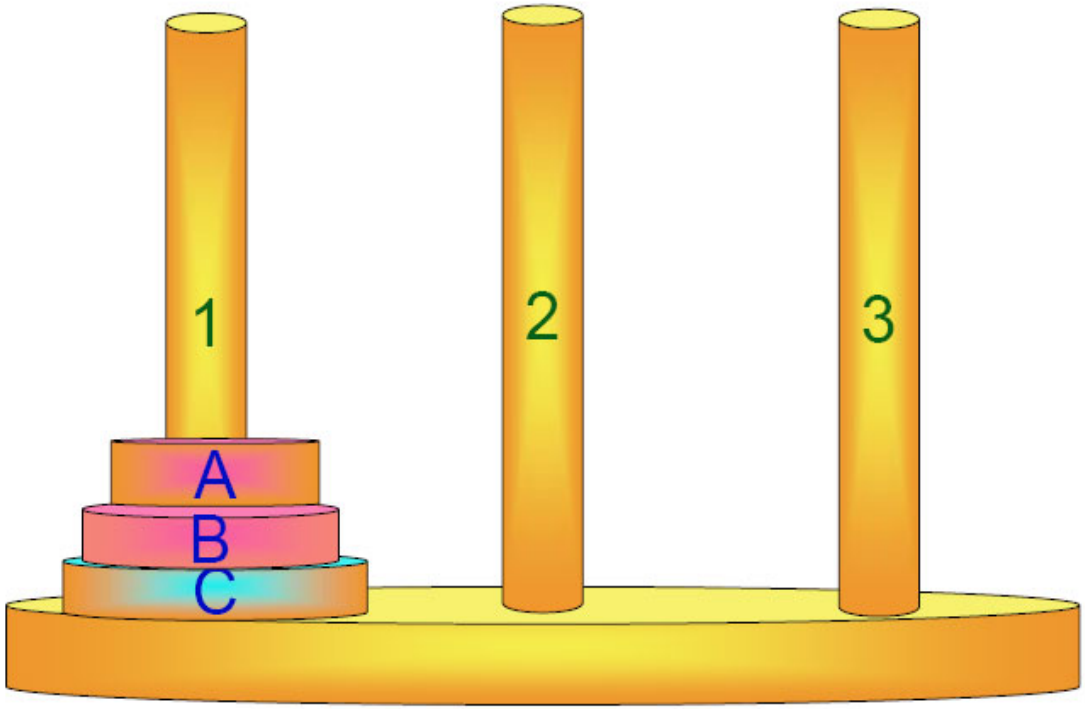
梵塔难题



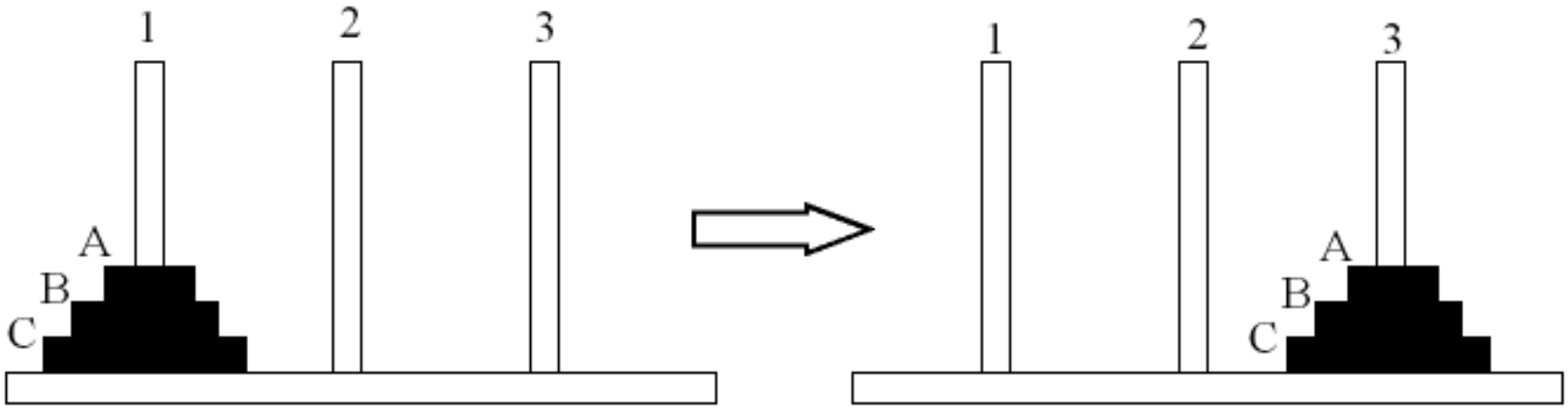
- 有3个柱子(1，2，3)和3个不同尺寸的圆盘(A，B，C)。在每个圆盘的中心有个孔，所以圆盘可以堆叠在柱子上。最初，全部3个圆盘都堆在柱子1上：最大的圆盘C在底部，最小的圆盘A在顶部。要求把所有圆盘都移到柱子3上，每次只许移动一个，而且只能先搬动柱子顶部的圆盘，还不许把尺寸较大的圆盘堆放在尺寸较小的圆盘上。这个问题的初始配置和目标配置如图所示。

45

- 梵塔难题



梵塔难题



(a) 初始状态　　(b) 目标状态

47

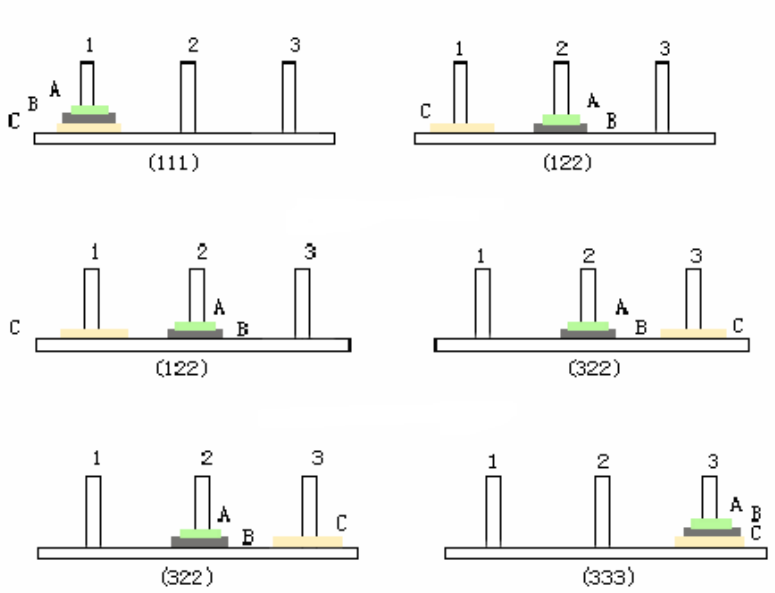
分析

- 原始问题归约（简化）为三个子问题

  1、移动A，B盘至柱子2的双圆盘难题

  2、移动圆盘C至柱子3的单圆盘问题

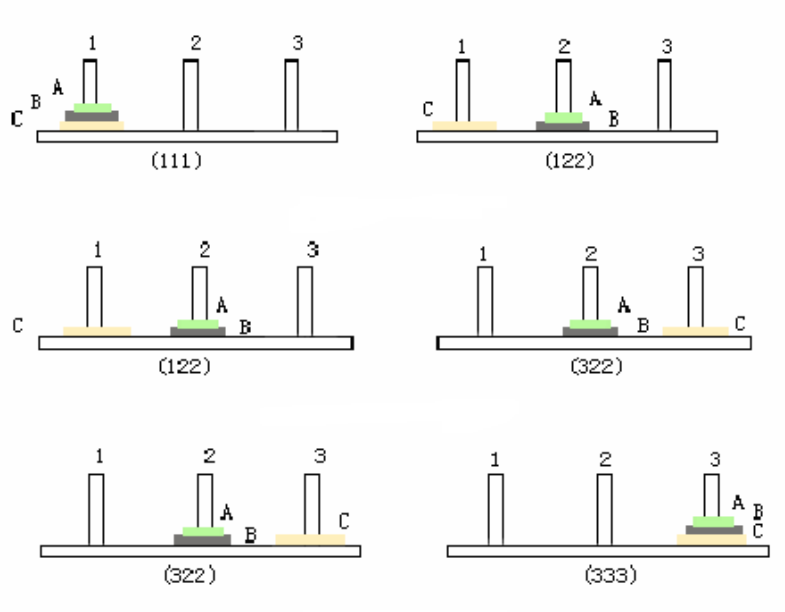
  3、移动A，B盘至柱子3的双圆盘难题



48

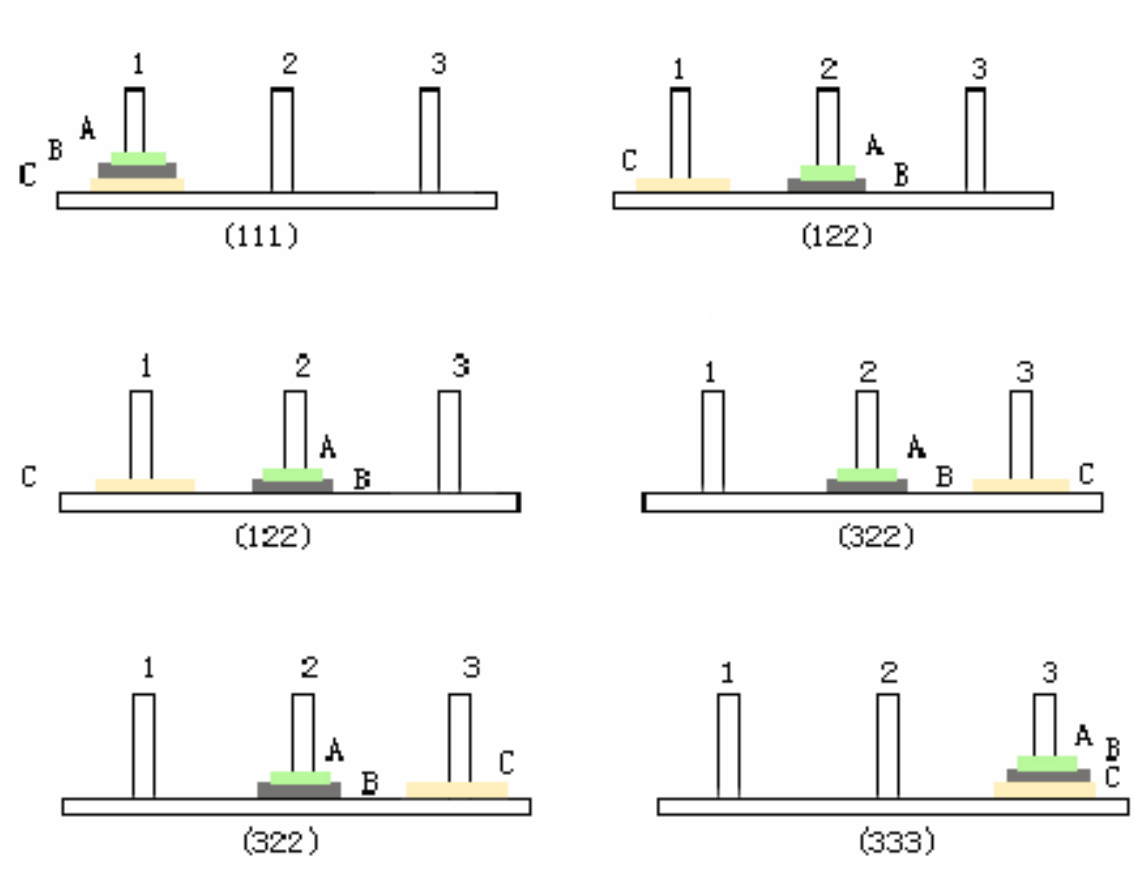
分析

- 对于梵塔问题，子问题[(111)→(122)]，[(122)→(322)]以及[(322)→(333)]规定了最后解答路径将要通过的脚踏石状态(122)和(322)。



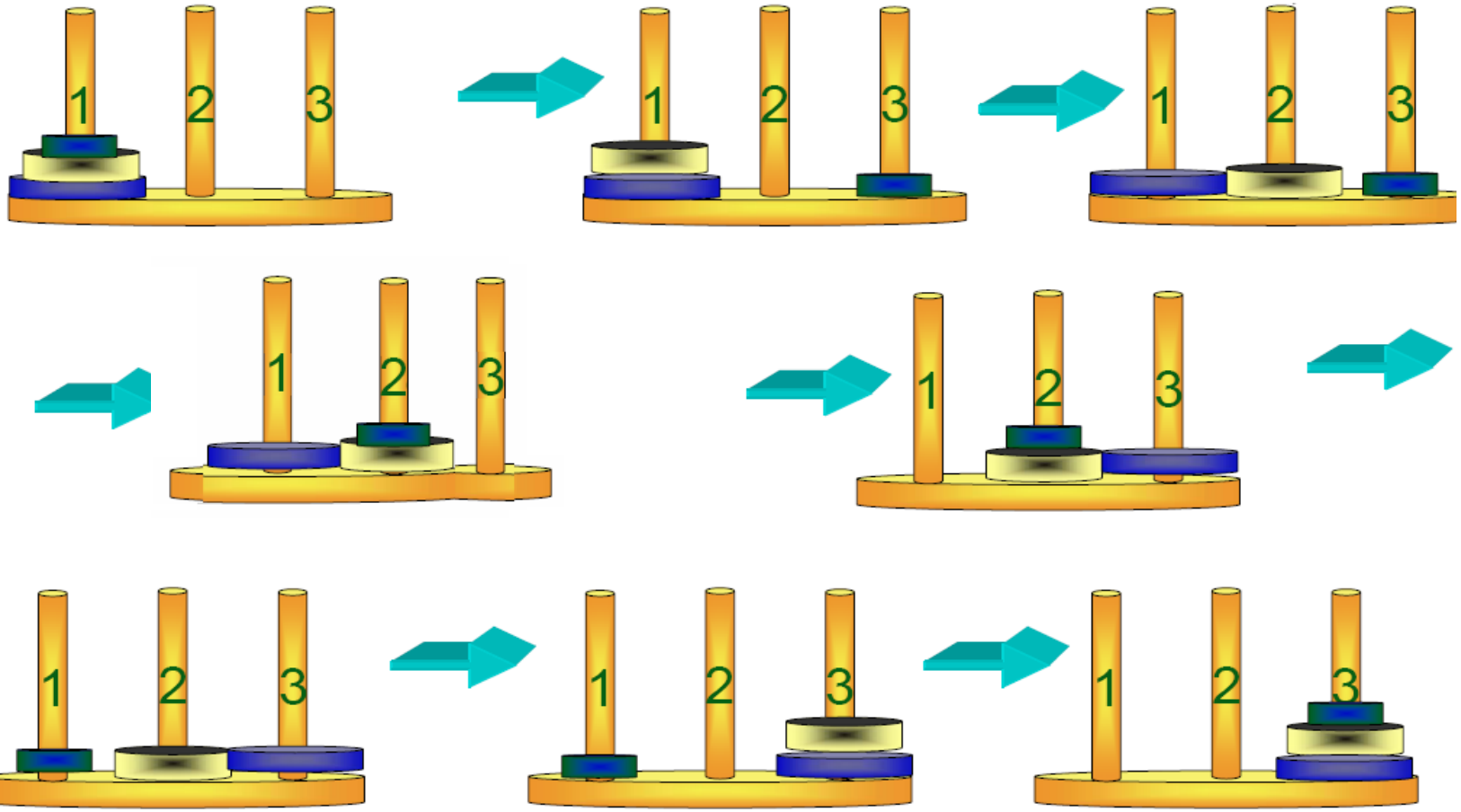
49

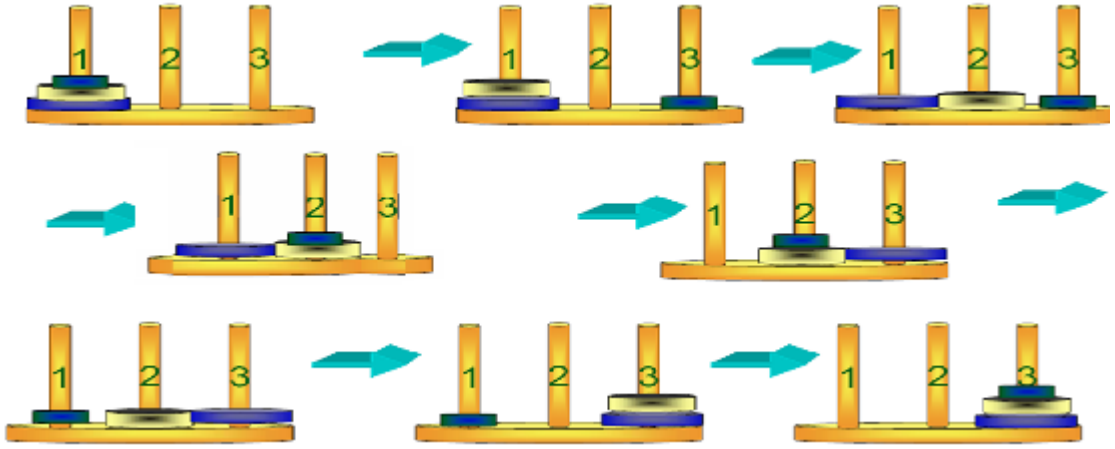
分析



50

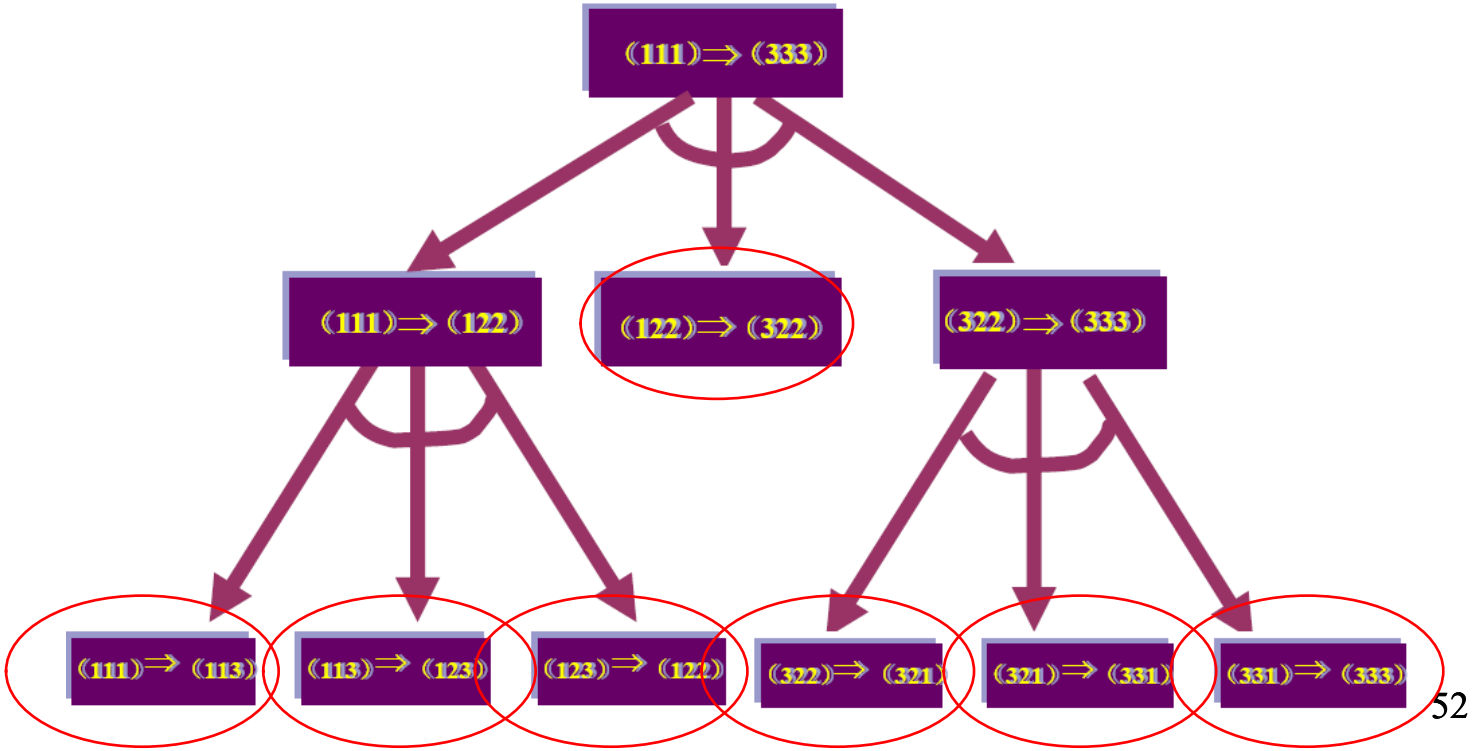
具体解题过程





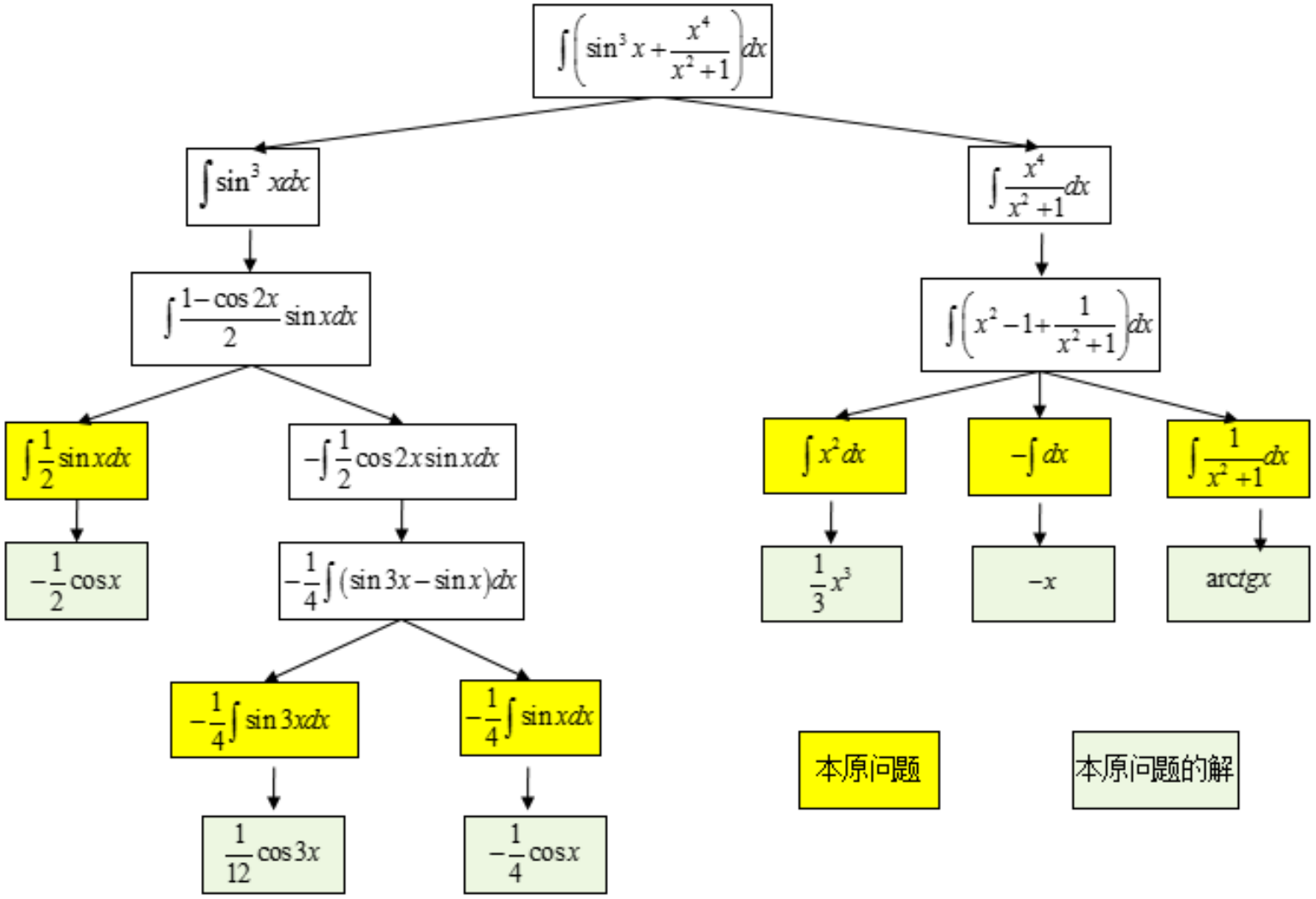
n=1?
n=2?
n=3?
⋮

梵塔问题归约图



52

不定积分的归约过程


$\int\left(\sin^3 x+\frac{x^4}{x^2+1}\right)dx$
$\int\sin^3 x dx$
$\int\frac{x^4}{x^2+1}dx$
$\int\frac{1-\cos 2x}{2}\sin x dx$
$\int\left(x^2-1+\frac{1}{x^2+1}\right)dx$
$\int\frac{1}{2}\sin x dx$
$-\int\frac{1}{2}\cos 2x\sin x dx$
$\int x^2 dx$
$-\int dx$
$\int\frac{1}{x^2+1}dx$
$-\frac{1}{2}\cos x$
$-\frac{1}{4}\int(\sin 3x-\sin x)dx$
$\frac{1}{3}x^3$
$-x$
arctgx
$-\frac{1}{4}\int\sin 3x dx$
$-\frac{1}{4}\int\sin x dx$
本原问题
本原问题的解
$\frac{1}{12}\cos 3x$
$-\frac{1}{4}\cos x$

由该问题归约的与或图，可读出该不定积分的解为：

$$\int\left(\sin^3 x+\frac{x^4}{x^2+1}\right)dx=-\frac{1}{2}\cos x+\frac{1}{12}\cos 3x-\frac{1}{4}\cos x+\frac{1}{3}x^3-x+\operatorname{arctg}x$$

$$=-\frac{3}{4}\cos x+\frac{1}{12}\cos 3x+\frac{1}{3}x^3-x+\operatorname{arctg}x$$

与或图表示

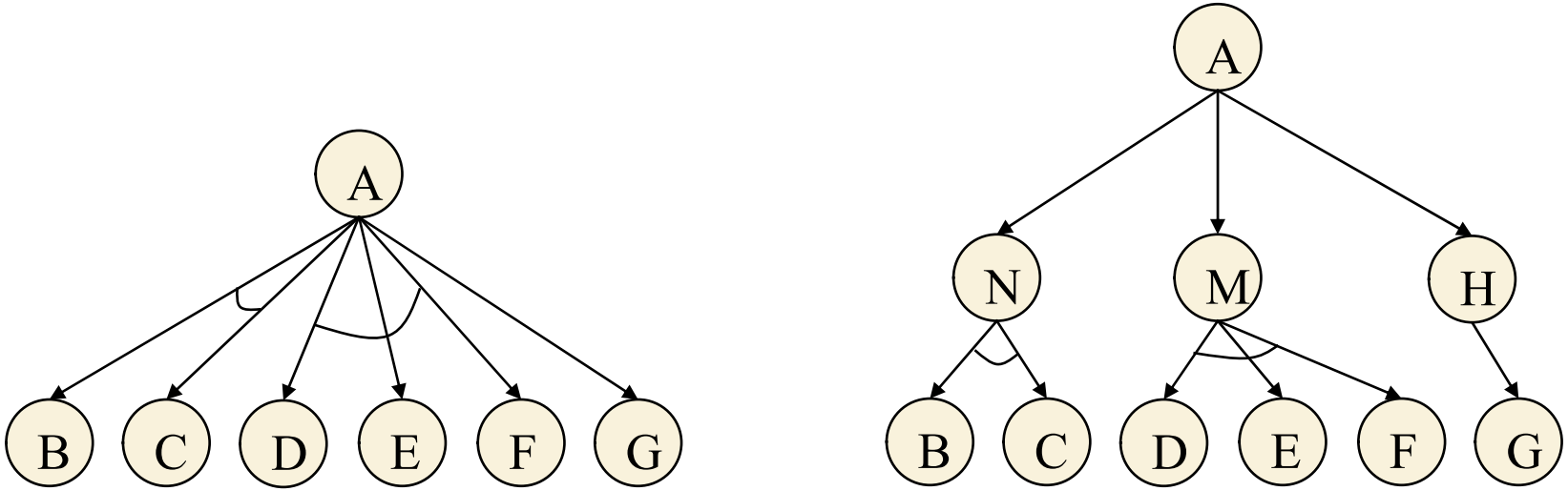
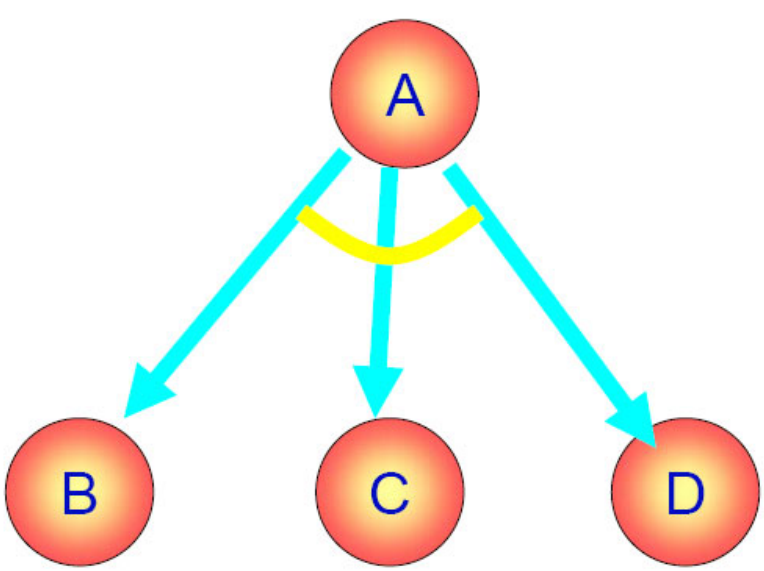


图 2.14 与或图

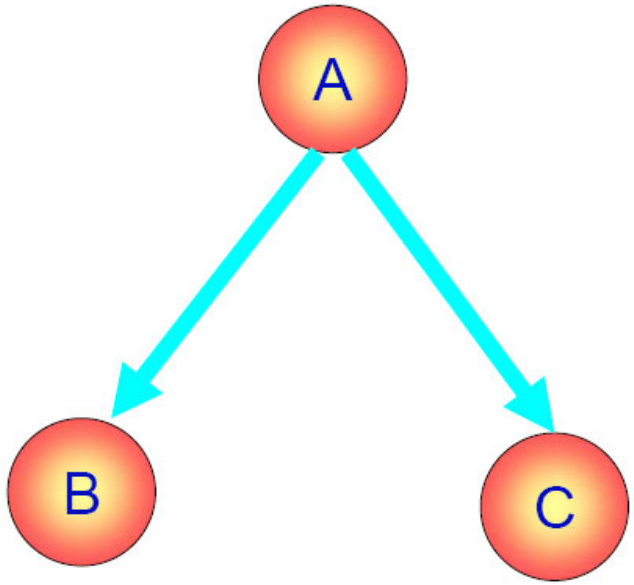
55

与或图表示

- 与图、或图、与或图



与图

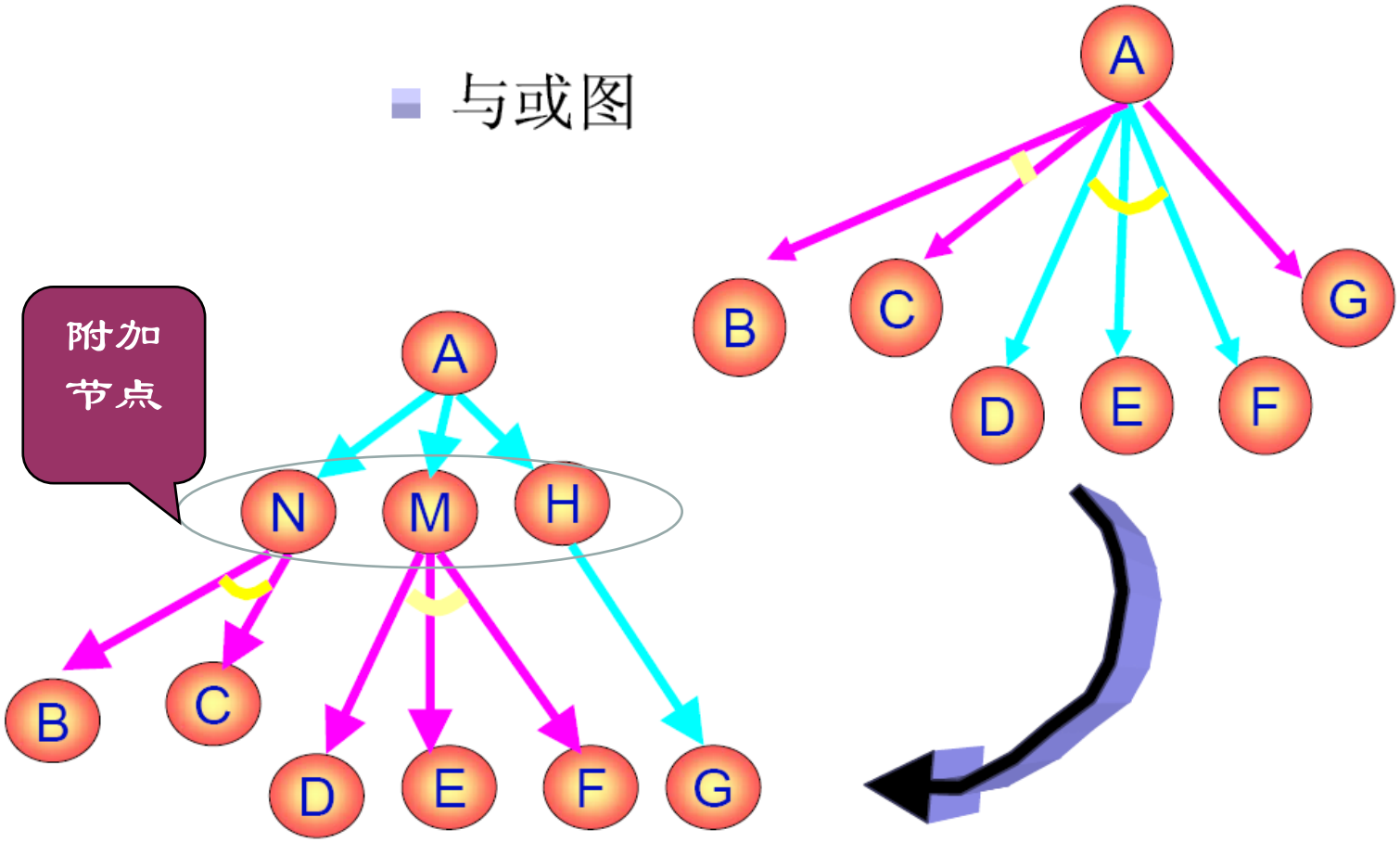


或图

56

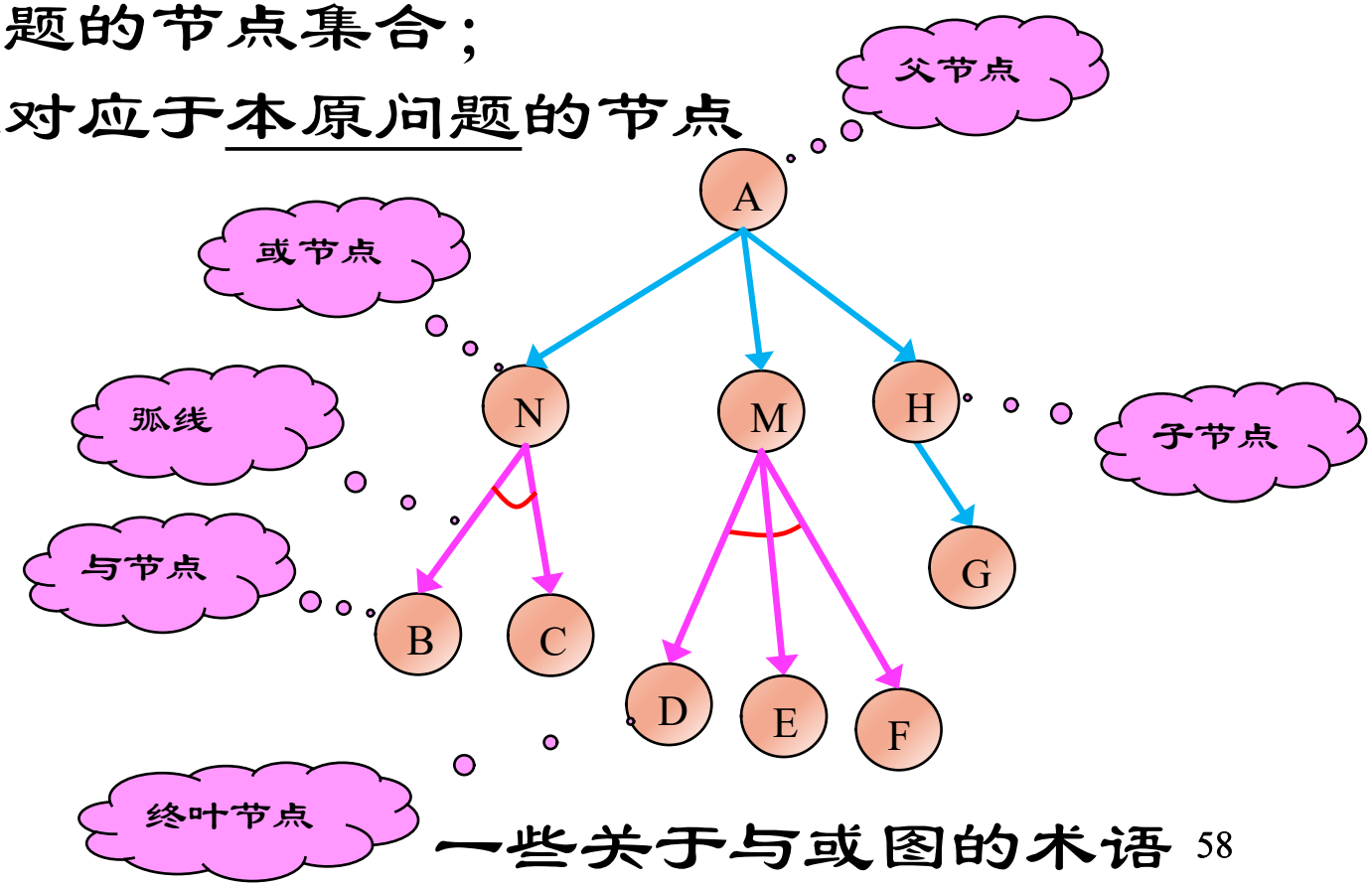
与或图表示

- 与或图



57

- 如果某条弧线从节点a指向节点b，那么节点a叫做节点b的**父辈节点**；节点b叫做节点a的**后继节点**或**后裔**；
- **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；
- **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；
- **终叶节点**，是对应于本原问题的节点



一些关于与或图的术语

58

- 可解节点：与或图中一个可解节点的一般定义可以归纳如下：

1、 终叶节点是可解节点(因为它们与本原问题相关连)。

2、如果某个非终叶节点含有**或后继节点**，那么只有当其后继节点至少有一个是可解的时，此非终叶节点才是可解的。

3、如果某个非终叶节点含有**与后继节点**，那么只要当其后继节点全部为可解时，此非终叶节点才是可解的。

59

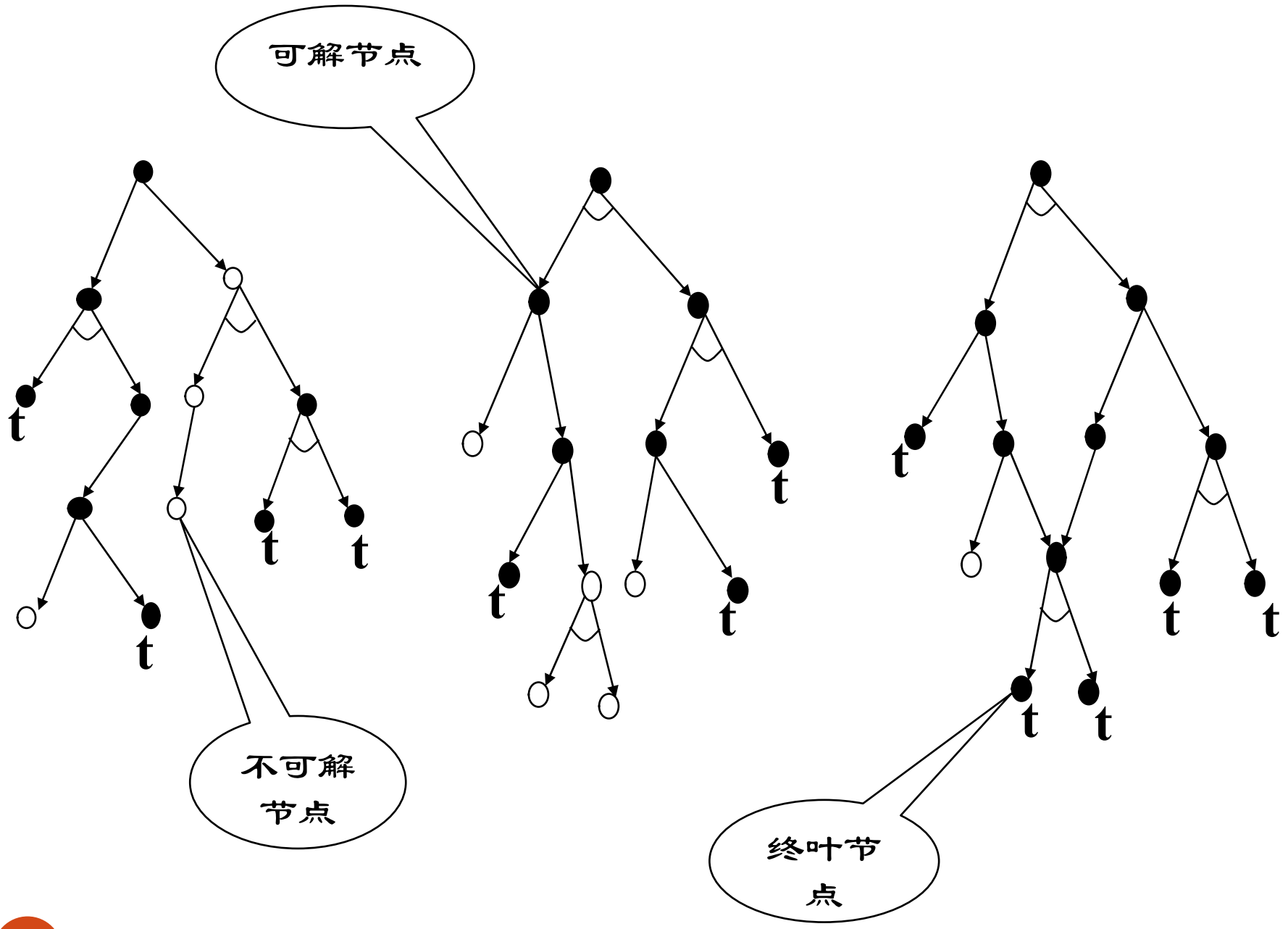
- 不可解节点 不可解节点的一般定义归纳于下：

1、没有后裔的非终叶节点为不可解节点。

2、如果某个非终叶节点含有或后继节点，那么只有当其全部后裔为不可解时，此非终叶节点才是不可解的。

3、如果某个非终叶节点含有与后继节点，那么只要当其后裔至少有一个为不可解时，此非终叶节点才是不可解的。

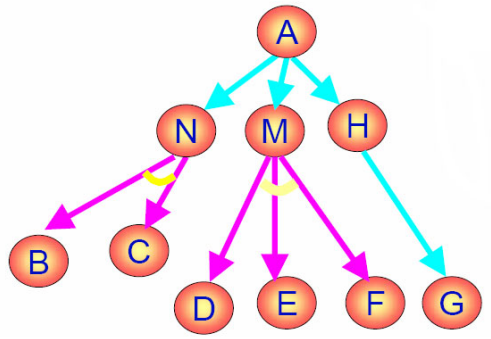
60

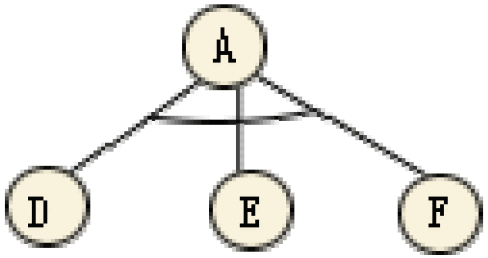


61

与或图构成规则

(1) 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。图中所含起始节点对应于原始问题。(2) 对应于本原问题的节点，叫做终叶节点，它没有后裔。(3) 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。（或节点）(4) 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。由于只有当集合中所有的项都有解时，这个子问题的集合才能获得解答，所以这些子问题节点叫做与节点。(5) 在特殊情况下，当只有一个算符可应用于问题A，而且这个算符产生具有一个以上子问题的某个集合时，由上述规则3和规则4所产生的图可以得到简化。因此，代表子问题集合的中间或节点可以被略去。





62

课后练习

- 试用四元数列结构表示四圆盘梵塔问题，并画出求解该问题的与或图。

63

小结

- 1. 状态空间法是一种基于解答空间的问题表示和求解方法，它是以状态和操作符为基础的。在利用状态空间图表示时，我们从某个初始状态开始，每次加一个操作符，递增地建立起操作符的试验序列，直到达到目标状态为止。由于状态空间法需要扩展过多的节点，容易出现"组合爆炸"，因而只适用于表示比较简单的问题。

64

小结

- 2. 问题归约法从目标(要解决的问题)出发，逆向推理，通过一系列变换把初始问题变换为子问题集合和子-子问题集合，直至最后归约为一个平凡的本原问题集合。这些本原问题的解可以直接得到从而解决了初始问题，用与或图来有效地说明问题归约法的求解途径。

65