2.3 习题解析

1. 搜索过程的三大要素是什么？对一个实际要求解问题，这三大要素分别是什么？

参考答案：

搜索过程的三大要素是搜索的对象、搜索的扩展规则和搜索的目标测试。

对一个实际要求解问题，搜索的对象是指在什么样的状态表示上进行搜索；搜索的扩展规则是指如何控制从一种状态变化为另一种状态，使得搜索得以前进；搜索的目标测试是指搜索在什么条件下终止。

2. 搜索求解问题的基本步骤是什么？对一个实际要求解问题，分别列举这些步骤。

参考答案：

搜索求解问题的基本步骤如下。

(1) 根据问题定义出相应的状态空间，确定出状态的一般表示，它含有相关对象各种可能的排列。然后给出问题的初始状态、目标状态，并能够给出所有其他状态的一般表示。

(2) 规定一组操作(算子)，能够作用于一个状态后过渡到另一个状态。

(3) 决定一种搜索策略，使得能够从初始状态出发，沿某个路径达到目标状态。

例如，编写一个下中国象棋的程序，这是典型的通过搜索来求解的问题。

棋局的状态空间就可以用一个二维数组表示，数组元素的取值就是该位置所放的棋子。问题的初始状态就是棋局开局时棋子摆放的那个状态；目标状态就将一方将死或和棋那个时候棋子的摆放状态；一般状态就是下棋时按下棋规则可以走出的棋局。

该问题的操作(算子)就是象棋所规定的下棋规则；该问题的搜索策略就是各棋手自己下棋的策略。

3. 列举一些实际问题，哪些问题是可分解的？哪些问题分解后是可以按任何次序独立求解子问题的？哪些不能按任何次序求解？

参考答案：

很多实际问题可以分解后变成若干小问题，对小问题再求解。这就是通常所说的"大事化小，小事化了"问题。"化了"就是可以直接求解。例如，高等数学中符号积分可以使用不定积分的计算规则，对一个复杂的积分公式进行分解。这些规则有分部积分规则、和式分解规则、因子规则、代数代换规则、相除化简规则、三角函数的代换规则等。

又如，大型矩阵相乘计算也可以将其中一个矩阵按行或列划分成若干矩阵，分别进行计算，这也是对复杂问题进行分解。

上述列举的两个例子都是问题分解后可以按任何次序独立求解子问题。对于堆积木的问题，如果限定机器人一次只能拿一块积木，而要求积木按一定的次序摆放，那么就不能按任何次序求解。

4. 从数理逻辑的角度解释一下，为什么要求解问题的知识库是相容的。

参考答案：

知识库如果不相容，也就是可以推导出矛盾 F，出现矛盾 F 后，按照数理逻辑的推理规则，F 作为前件，公式 $F \rightarrow ?$ 中"?"不管为真或为假，该公式都为真，这就意味着，该知识库什么结论都可以推导出来。

2.4 补充习题

1. 设有3只琴键开关一字排开，初始状态为“关、开、关”，问连按3次后是否会出现“开、开、开”或“关、关、关”的状态？要求每次必须按下一个开关，而且只能按一个开关。请画出状态空间图。

参考答案：

用(K_1,K_2,K_3)表示3个开关的状态，取值为0时表示闭合，为1时表示打开，则初始状态为(0,1,0)。根据题设要求，一个状态与下一个状态之间只能有一位取值不同(此即状态转换规则)，据此可以画出如图2.1所示的状态空间图。

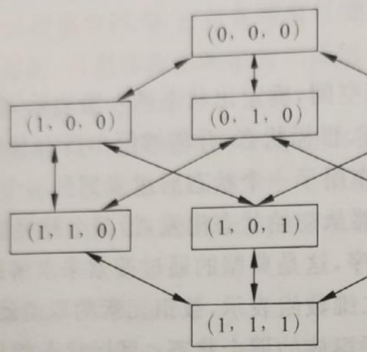

图2.1　状态空间图

从此状态图不难看出，经过连续三步有状态(0,1,0)只能到达状态(0,0,0)而不能到达状态(1,1,1)，即会出现状态“关，关，关”，但不会出现“开，开，开”。

2. 有一个农夫带一条狼、一只羊和一框青菜欲从河的左岸乘船到右岸，但受到下列条件的限制。

(1) 船太小，农夫每次只能带一样东西过河。

(2) 如果没有农夫看管，则狼要吃羊，羊要吃青菜。

请设计一个过河方案，使得农夫、狼、羊都能不受损失地过河，画出相应的状态空间图。

提示：

(1) 用四元组(农夫、狼、羊、青菜)表示状态，其中每个元素都为0或1，用1表示在左岸，用0表示在右岸。

(2) 把每次过河的一种安排作为一种操作，每次过河都必须有农夫，因为只有他可以划船。

参考答案：

第一步，定义问题的描述形式。

用四元组$S=(f,w,s,v)$表示问题状态，其中，f、w、s和v分别表示农夫、狼、羊和青菜是否在左岸，它们都可以取1或0，取1表示在左岸，取0表示在右岸。

第二步，用所定义的问题状态表示方式，把所有可能的问题状态表示出来，包括问题的初始状态和目标状态。

由于状态变量有 4 个，每个状态变量都有两种取值，因此有以下 16 种可能的状态。

$S_0=(1,1,1,1)$，　$S_1=(1,1,1,0)$，　$S_2=(1,1,0,1)$，　$S_3=(1,1,0,0)$

$S_4=(1,0,1,1)$，　$S_5=(1,0,1,0)$，　$S_6=(1,0,0,1)$，　$S_7=(1,0,0,0)$

$S_8=(0,1,1,1)$，　$S_9=(0,1,1,0)$，　$S_{10}=(0,1,0,1)$，　$S_{11}=(0,1,0,0)$

$S_{12}=(0,0,1,1)$，　$S_{13}=(0,0,1,0)$，　$S_{14}=(0,0,0,1)$，　$S_{15}=(0,0,0,0)$

其中，状态 S_3、S_6、S_7、S_8、S_9、S_{12} 是不合法状态，S_0 和 S_{15} 分别为初始状态和目标状态。

第三步，定义操作，即用于状态变换的算符组 F。

由于每次过河船上都必须有农夫，且除农夫外船上只能载狼、羊和青菜中的一种，因此算符定义如下。

$L(i)$表示农夫从左岸将第 i 样东西送到右岸（$i=1$ 表示狼，$i=2$ 表示羊，$i=3$ 表示青菜，$i=0$ 表示船上除农夫外不载任何东西）。由于农夫必须在船上，因此对农夫的表示省略。

$R(i)$表示农夫从右岸将第 i 样东西带到左岸（$i=1$ 表示狼，$i=2$ 表示羊，$i=3$ 表示青菜，$i=0$ 表示船上除农夫外不载任何东西）。同样，对农夫的表示省略。

这样，所定义的算符组 F 可以有以下 8 种算符。

$L(0)$，　$L(1)$，　$L(2)$，　$L(3)$

$R(0)$，　$R(1)$，　$R(2)$，　$R(3)$

第四步，根据上述定义的状态和操作进行求解。

该问题求解过程的状态空间图如图 2.2 所示。

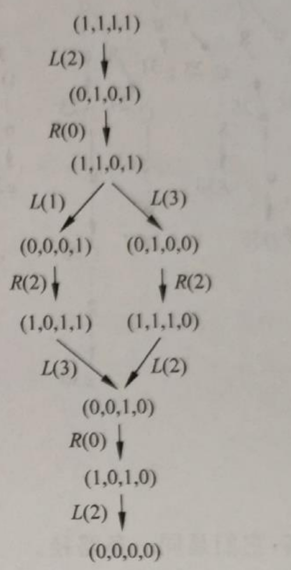

图 2.2　过河问题的状态空间图

3. 图 2.3 是 5 个城市的交通图，城市之间的连线旁边的数字是城市之间路程的费用。要求从 A 城出发，经过其他各城市一次且仅一次，最后回到 A 城，请找出一条最优线路。

参考答案：

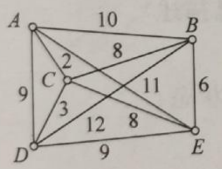

图 2.3　交通费用图

这个问题又称为旅行商问题（Travelling Salesman Problem，TSP）或货郎担问题，是一个较有普遍性的实际应用问题。根据数学理论，对 n 个城市的旅行商问题，其封闭路径的排列总数为

$$(n!)/n=(n-1)!$$

其计算量相当大。例如，当 $n=20$ 时，要穷举其所有路径，即使用一个每秒一亿次的计算机来算也需要 350 年。因此，对这类问题只能用智能搜索的方法来解决。

图 2.4 是对图 2.3 按最小代价搜索所得到的搜索树，树中的节点为城市名称，节点边上的数字为该节点的代价 g。其计算公式为

$$g(n_{i+1})=g(n_i)+c(n_i,n_{i+1})$$

其中，$c(n_i,n_{i+1})$为节点 n_i 到节点 n_{i+1} 的边代价。

从图 2.4 中可以看出，其最短路径为

$$A\rightarrow C\rightarrow D\rightarrow E\rightarrow B\rightarrow A$$

或

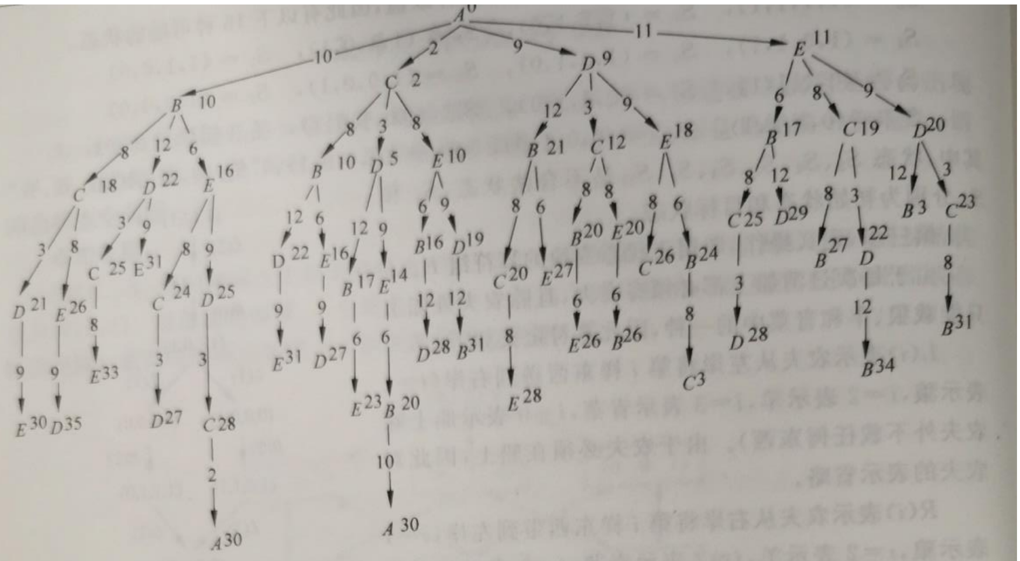

图 2.4　最小代价搜索树

$$A \rightarrow B \rightarrow E \rightarrow D \rightarrow C \rightarrow A$$

其实，它们是同一条路径。

4. 什么是状态空间法?

参考答案：

状态空间(State Space)是利用状态变量和操作符号，表示系统或问题的有关知识的符号体系，状态空间是一个四元组(S,O,S_0,G)。

其中S为状态集合；O为操作算子集合；S_0为初始状态，$S_0 \subset S$；G为目标状态，$G \subset S$（G可为满足某些性质的若干具体状态，也可为一系列状态的路径信息描述）。

从S_0节点到G节点的路径被称为解路径。

状态空间的解是一有限操作算子序列，它使初始状态转换为目标状态：

$$S_0 \xrightarrow{O_1} S_1 \xrightarrow{O_2} S_2 \xrightarrow{O_3} \cdots \xrightarrow{O_k} G$$

其中，$O_1, O_2, \cdots, O_k$即为状态空间的一个解（解往往不是唯一的）。