

# Homework1

Artificial Intelligence

PB20020480 王润泽

## Q1

**3.7** Give the initial state, goal test, successor function, and cost function for each of the following. Choose a formulation that is precise enough to be implemented.

a. You have to color a planar map using only four colors, in such a way that no two adjacent regions have the same color.

b. A 3-foot-tall monkey is in a room where some bananas are suspended from the 8-foot ceiling. He would like to get the bananas. The room contains two stackable, movable, climbable 3-foot-high crates.

d. You have three jugs, measuring 12 gallons, 8 gallons, and 3 gallons, and a water faucet. You can fill the jugs up or empty them out from one to another or onto the ground. You need to measure out exactly one gallon.

## A1

a.

初始状态：没有颜色的地图

目标测试：地图地区颜色涂满且两个相邻地区颜色不同

后继函数：用4种颜色的一种涂一个地图颜色后依然满足任意两个相邻地区颜色不同的地图状态

耗散函数：涂色次数

b.

初始状态：猴子和箱子任意处于某个位置

目标测试：箱子叠放，且猴子攀登后可以获取香蕉

后继函数：产生通过箱子移动、叠放、攀登后箱子的合法状态

耗散函数：每一步动作的耗散值为1，整个过程的耗散值为总步数

d.

初始状态：三个水壶容量为12加仑，8加仑和3加仑，且所有水壶为空

目标测试：存在一个水壶内只有1加仑水

后继函数：经过某个水壶装满、倒空，或从一个水壶向另一个水壶倒水后的合法状态（即不存在水壶里的水超过容积）

耗散函数：每一步动作的耗散值为1，整个过程的耗散值为总步数

## Q2

3.9 The **missionaries and cannibals** problem is usually stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place. This problem is famous in AI because it was the subject of the first paper that approached problem formulation from an analytical viewpoint (Amarel, 1968).

- Formulate the problem precisely, making only those distinctions necessary to ensure a valid solution. Draw a diagram of the complete state space.
- Implement and solve the problem optimally using an appropriate search algorithm. Is it a good idea to check for repeated states?
- Why do you think people have a hard time solving this puzzle, given that the state space is so simple?

## A2

a. 目标: 从河岸A将3位教士和3个野人运到河岸B.

状态抽象: 河岸A当前位教士数目  $C$ , 野人数目  $m$ , 是否有船  $b$

用三元组  $[C, m, b]$  表示.

初始状态:  $[3, 3, 1]$

目标状态:  $[0, 0, 0]$

约束:  $C \geq m$  且  $3-C \geq 3-m \Leftrightarrow C=m$ .

或  $m=0, C=1, 2, 3$

或  $C=0, m=1, 2, 3$

后继函数:  $b=1 \rightarrow b=0$  且  $\begin{cases} A=A-1, C=C-1 \\ A=A-2 \\ C=C-2 \\ A=A-1 \\ C=C-1 \end{cases}$

$b=0 \rightarrow b=1$  且  $\begin{cases} A=A+1, C=C+1 \\ A=A+2 \\ C=C+2 \\ A=A+1 \\ C=C+1 \end{cases}$

且满足约束

完整状态空间  $[3, 3, 1], [3, 2, 1/0], [3, 1, 1/0], [3, 0, 1/0], [2, 2, 1/0], [1, 1, 1/0], [0, 3, 1/0], [0, 2, 1/0], [0, 1, 1/0], [0, 0, 0]$  共18种状态

b. 搜索算法: DFS 策略.

从初始  $[3, 3, 1]$  开始, 在操作后, 检查是否为状态空间中未一个.

如果有, 改变状态, 检查新状态, 继续操作, 直到处于  $[0, 0, 0]$  状态.

如果没有, 则回溯到前一步状态 (深搜停止).

需检查重复状态, 防止出现死循环

c. 状态空间简单, 但反复检测重复状态是困难的.