

hw2

3210105327 万晨阳

Q 3.7

- a
 - **初始状态**: 未经染色的平面地图;
 - **目标测试**: 所有地区都被染色并且相邻地区的颜色不同;
 - **后继函数**: 对没有被染色的区域中的某一个进行染色, 保证其所选颜色与其相邻地区的颜色不同;
 - **耗散函数**: 在某状态下选择下一个未染色地区进行染色的耗散为一个单位。
- c
 - **初始状态**: 某个未经检测的文件, 一个计算机程序;
 - **目标测试**: 检测到某个记录不合法;
 - **后继函数**: 输入一个记录, 并检测是否合法;
 - **耗散函数**: 每输入一个记录耗散为一个单位。

Q 3.9

- a
 - **初始状态**: 六个人均在河的一边。
 - **目标测试**: 是否六个人都在河的另一边;
 - **后继函数**: 一人或者两人坐船从河的一边到河的另外一边;
 - **耗散函数**: 当前状态下船从一侧划到另一侧的耗散值为一个单位。

我们约定以一个三元组作为某一状态的描述:

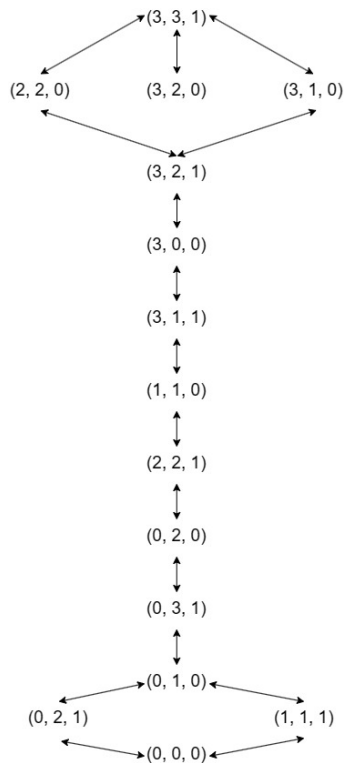
(河左岸的传教士数目, 河左岸的野人数目, 船的状态)

其中对于船的状态, 1表示船在左岸, 0表示船在右岸。

综上, 该问题的所有合法状态如下, 共15种:

```
(3, 3, 1)
(3, 2, 1)
(3, 2, 0)
(3, 1, 1)
(3, 1, 0)
(3, 0, 0)
(2, 2, 1)
(2, 2, 0)
(1, 1, 1)
(1, 1, 0)
(0, 3, 1)
(0, 2, 1)
(0, 2, 0)
(0, 1, 0)
(0, 0, 0)
```

状态空间转换图如下：



- b
 - 广度优先搜索。广度优先搜索是一种遍历图形结构的算法，它从起始状态开始，逐层探索每个可能的状态，直到找到目标状态。在传教士与野人问题中，广度优先搜索会从初始状态开始，生成所有可能的状态，并在每一步考虑所有合法的移动，然后继续探索这些状态，直到找到达到目标状态的路径。
 - 检测重复状态是一个好主意。因为检测重复状态能够通过避免重复探索相同的状态从而加快搜索速度，同时防止无限循环，保证算法终止。
- c
 - 虽然这个问题的状态空间比较简单，但是在求解的过程中需要对某一状态下所有可能的下一个合法状态进行列举，能够做到不漏举与避开非法状态是比较困难的。另外，在某一状态没有下一个合法状态的时候我们需要回溯进行求解，因而我们需要对于每一个状态维护其可能的转移策略，这都使得人为求解变得困难。