Homework 3

PB17000297 罗晏宸

March 16 2020

1 Exercise 5.8

考虑图1中描述的两人游戏。

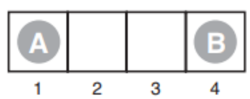

图 1: 一个简单游戏的初始棋局

选手 A 先走。两个选手轮流走棋，每个人必须把自己的棋子移动到任一方向上的相邻空位中。如果对方的棋子占据着相邻的位置，你可以跳过对方的棋子到下一个空位。（例如，A 在位置 3，B 在位置 2，那么 A 可以移回 1。）当一方的棋子移动到对方的端点时游戏结束。如果 A 先到达位置 4，A 的值为 $+1$；如果 B 先到达位置 1，A 的值为 -1。

a 根据如下约定画出完整博弈树：

- 每个状态用 (s_A, s_B) 表示，其中 s_A 和 s_B 表示棋子的位置。
- 每个终止状态用方框画出，用圆圈写出它的博弈值。
- 把循环状态（在到根结点的路径上已经出现过的状态）画上双层方框。由于不清楚他们的值，在圆圈里标记一个“?”。

b 给出每个结点倒推的极小极大值（也标记在圆圈里）。解释怎样处理“?”值和为什么这么处理。

c 解释标准的极小极大算法为什么在这棵博弈树中会失败，简要说明你将如何修正它，在 (b) 的图上画出你的答案。你修正后的算法对于所有包含循环的游戏都能给出最优决策吗？

d 这个 4-方格游戏可以推广到 n 个方格，其中 $n>2$。证明如果 n 是偶数 A 一定能赢，而 n 是奇数则 A 一定会输。

解

a 博弈树如图2所示

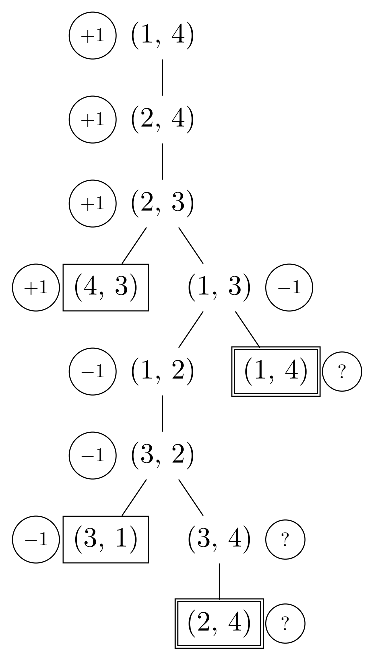

图 2: 两人游戏的博弈树

b 在圆圈中的博弈值后附上相应的极小与极大值，如图3所示 对于“?”值采取的处理办法是，如果 Agent 能够选择则总是会选择获胜（即博弈值为 $+1$），因此除非后继结点均为“?”，可以将“?”视为介于 -1 与 $+1$ 之间的值。

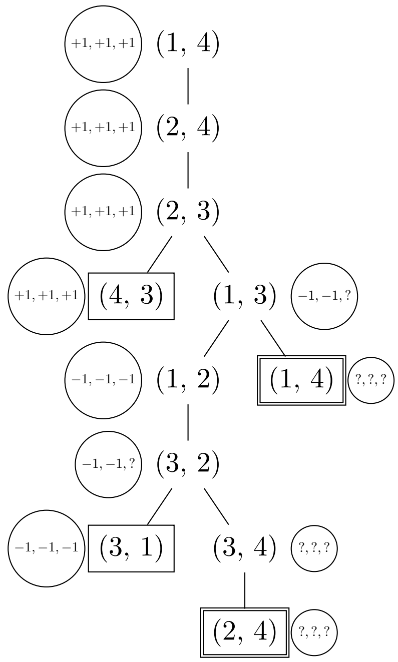

图 3: 标注每个结点倒推极小极大值的博弈树

c 标准的极小极大算法是深度优先的，会在遇到循环状态时陷入无限循环。一个可行的修正方式是带记录的深度优先搜索，通过维护一个队列记录已遍历的结点，如果遇到已在队列中的重复状态，就直接返回“?”，通过前述的处理方法来解决比较。答案如所示。修正后的算法并不能给出所有包含循环的游戏的最优决策，因为这样的处理方法不能区分不同的“?”值，在获胜情况并不唯一，或者有不同获胜程度的状态时，直接返回可能会丢失其他获胜程度较深的状态。

d 当 $n = 3$ 和 $n = 4$ 时，A 分别输掉和赢得游戏，很显然对于 $n \geq 4$ 的情况，A 和 B 在开始都会向对方走一格，如果 A 在 $n-2$ 的情况获胜，那么在这种情况（A 处于位置 1，B 处于位置 $n-1$）下，A 将会在 B 到达位置 1 之前到达位置 $n-1$，那么 A 也将会在 n 的情况下获胜，在这种情况下 A 不会向着自己的原始位置行棋；同理如果 A 在 $n-2$ 的情况失败，意味着 B 获胜，那么 B 也将会在 n 的情况获胜。因此如果 n 是偶数 A 一定能赢，而 n 是奇数则 A 一定会输。

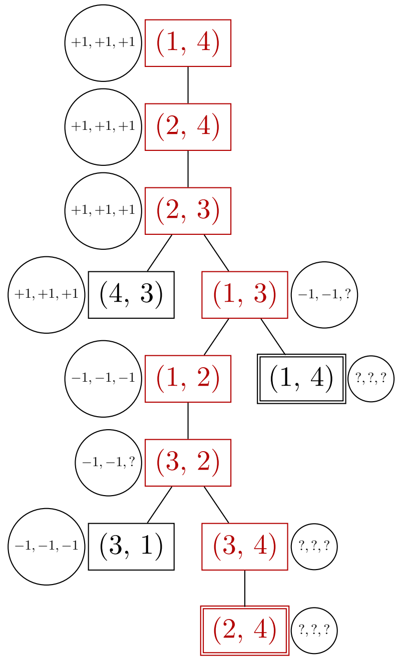

图 4: 修正后求解两人游戏的答案

2 Exercise 5.9

本题以井字棋（圈与十字游戏）为例练习博弈中的基本概念。定义 X_n 为恰好有 n 个 X 而没有 O 的行、列或者对角线的数目。同样 O_n 为正好有 n 个 O 的行、列或者对角线的数目。效用函数给 $X_3 = 1$ 的棋局 $+1$，给 $O_3 = 1$ 的棋局 -1。所有其他终止状态效用值为 0。对于非终止状态，使用线性的评估函数定义为 $Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$。

a 估算可能的井字棋局数。

b 考虑对称性，给出从空棋盘开始的深度为 2 的完整博弈树（即，在棋盘上一个 X 一个 O 的棋局）。

c 标出深度为 2 的棋局的评估函数值。

d 使用极小极大算法标出深度为 1 和 0 的棋局的倒推值，并根据这些值选出最佳的起始行棋。

e 假设结点按对 α-β 剪枝的最优顺序生成，圈出使用 α-β 剪枝将被剪掉的深度为 2 的结点。

解

a 井字棋最多可能将棋盘的 9 个格子下满，最少需要双方轮流执棋两合后先手获胜，因此可能的局数在 $\frac{9!}{4!}$ 到 $9!$ 之间。

b 完整博弈树如图5所示

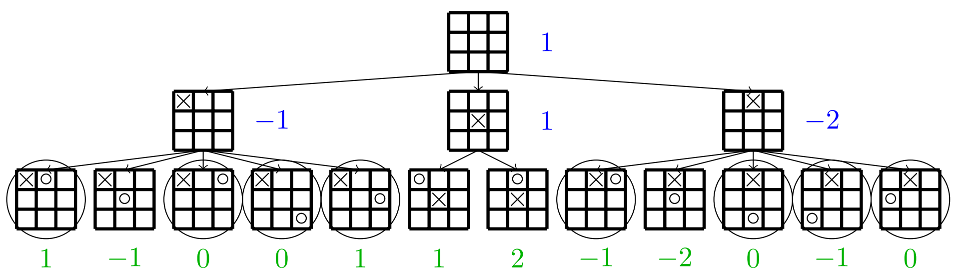

图 5: 从空棋盘开始的深度为 2 的完整博弈树

c 评估函数值已在图5中以绿色标出

d 倒推值已在图5中以蓝色标出，可知最佳的起始行棋是在棋盘中央落子。

e 被剪掉的结点已在图5中圈出。

3 Exercise 5.13

请给出 α-β 剪枝正确性的形式化证明。要做到这一点需考虑图6。问题为是否要剪掉结点 n_j，它是一个 MAX 结点，是 n_1 的一个后代。基本的思

路是当且仅当 n_1 的极小极大值可以被证明独立于 n_j 的值时，会发生剪枝。

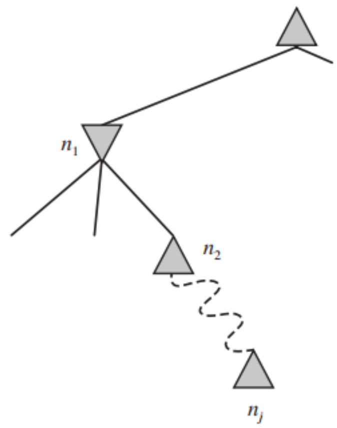

图 6: 是否剪掉结点 n_j 时的情形

a n_1 的值时所有后代结点的最小值：$n_1 = \min(n_2, n_{2\,1}, \cdots, n_{2\,b_2})$。请为 n_2 找到类似的表达式，以得到用 n_j 表示的 n_1 的表达式。

b 深度为 i 的结点 n_i 的极小极大值已知，l_i 是在结点 n_i 左侧结点的极小值（或者极大值）。同样，r_i 是在 n_i 右侧的未探索过的结点的极小值（或者极大值）。用 l_i 和 r_i 的值重写 n_1 的表达式。

c 现在重新形式化表达式，来说明为了向 n_1 施加影响，n_j 不能超出有 l_i 值得到的某特定界限。

d 假设 n_j 是 MIN 结点的情况，请重复上面的过程。

解

a 类似的，有 $n_2 = \max(n_3, n_{3\,1}, \cdots, n_{3\,b_3})$，得到

$$n_1 = \min(\max(\cdots \max(n_j, n_{j\,1}, \cdots, n_{j\,b_j})), n_{2\,1}, \cdots, n_{2\,b_2})$$

b 改写 n_1 如下

$$\begin{aligned} n_1 &= \min(l_2,\, n_2,\, r_2) \\ &= \min(l_2,\, \max(l_3,\, n_3,\, r_3),\, r_2) \\ &\vdots \\ &= \min(l_2,\, \max(l_3,\, \max(\cdots\max(l_j,\, n_j,\, r_j),\, \cdots),\, r_3),\, r_2) \end{aligned}$$

c 由前述可知

$$n_i = \min(l_{i+1},\, \max(l_{i+2},\, \max(\cdots\max(l_j,\, n_j,\, r_j),\, \cdots),\, r_{i+2}),\, r_{i+1})$$

由于 n_j 是一个 MAX 结点,为了向 n_1 施加影响,n_j 不能超出 $\min(l_2, l_4, \cdots, l_j)$。

d 假设 n_j 是 MIN 结点，只需要交换前述中的 max 与 min 即可。