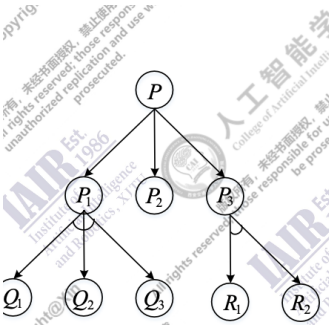
博弈搜索

一．博弈简介

1. 人工智能领域的“博弈”：有完整信息的，确定性的、轮流行动的，两个参与者的零和（Zero-sum）游戏（总值相等而符号相反的效用函数的对立导致对抗）。
2. 博弈的现实分类：1.同时博弈 2.相继博弈 3.边界博弈 4.规则博弈 5.策略性博弈
3. 根据博弈者数量分类：单人、两人、多人博弈（单人博弈是退化的博弈）
4. 按照博弈先后顺序分类：静态博弈（参与者同时行动，或者后者不知道前者是如何行动的；例：石头剪刀布，罚点球时的守门员与罚球手）、动态博弈（参与者有先后顺序；例：下棋，拍卖会）。
5. 根据博弈中的得益分类：零和博弈（利益始终对立）、常和博弈（利益是对立的且是竞争关系）、变和博弈（合作利益存在）。
6. 根据博弈中得益的信息分类：完全信息博弈（各方都了解所有博弈方各种情况下的得益）、不完全信息博弈（至少部分博弈方不完全了解）。
7. 根据博弈过程的信息：完美信息博弈（各方对每轮进程完全了解）、不完美信息博弈（至少部分博弈方不完全了解）。
8. 根据博弈者之间是否合作：合作博弈（集体利益最大化）、非合作博弈（个人利益最大化）

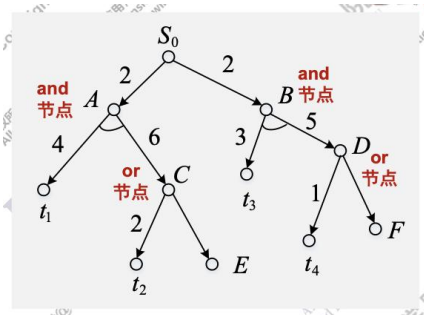
二．问题归约表示（与或树）

1. 本源问题：指无法（或者不需要）再分解，且可直接解答的问题。
2. 与或树相关的概念：



- **若一个问题既需要通过分解，又需要通过变换才能得到其原本问题，则其求解过程可用1个“与/或树”来表示**
- **事实上，大多数实际问题都需要用与/或树来表示，即在解决大多数问题时，对原问题的分解和变换是结合的**
  - **在与/或树中，根节点对应着待求解的原始问题**
  - **在与/或树中，没有子节点的节点称为端节点**
  - **本原问题对应的节点为终止节点**
  - **终止节点一定是端节点，但端节点不一定是终止节点**
- **可解节点：一个终止节点；一个“或”节点，且其子节点中至少有一个为有解的节点；一个“与”节点，且其子节点全部为有解的节点**

3. 与或树代价的求解：



- 若节点 n 为终止节点，则其代价 $h(n)=0$
- 若节点 n 为或节点，且其子节点为 $n_1,n_2,\cdots,n_k$，则其代价

$$h(n)=\min_{1\le i\le k}\{c(n,n_i)+h(n_i)\}$$

其中，$c(n,n_i)$ 节点 n 到其子节点 n_i 的代价

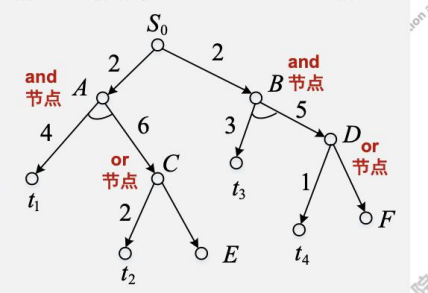
- 若节点 n 为与节点，且其子节点为 $n_1,n_2,\cdots,n_k$ ，则其代价

$$h(n)=\Sigma_{i=1}^{k}\{c(n,n_i)+h(n_i)\} \quad \text{或} \quad h(n)=\max_{1\le i\le k}\{c(n,n_i)+h(n_i)\}$$

- 若节点 n 为非终止节点的端节点，则其代价 $h(n)=\infty$

4. 与节点有两种代价计算方式（sum/max），要注意审题，判断使用哪种。
5. 例题：

与或树求解的代价计算示例



- 终止节点：t_1,t_2,t_3,t_4
- 非终止端节点：E,F
- 边上旁的数字为该边的代价

$$h(t_1)=h(t_2)=h(t_3)=h(t_4)=0$$
$$h(E)=h(F)=\infty$$

■ 左侧子树：

- 按与点求和代价计算：

$$h(C)=2,\ h(A)=12,\ h(S_0)=14$$

- 按与点最大值代价计算：

$$h(C)=2,\ h(A)=8,\ h(S_0)=10$$

■ 右侧子树：

- 按与点求和代价计算：

$$h(D)=1,\ h(B)=9,\ h(S_0)=11$$

- 按与点最大值代价计算：

$$h(D)=1,\ h(B)=6,\ h(S_0)=8$$

6. 博弈树是一种特殊的与或树，特点有：
   1) 博弈的初始状态是初始节点
   2) 博弈中 or 节点（Max 方）和 and 节点(Min 方)逐层交替出现
   3) 整个博弈过程始终站在某一方立场上（Max 方或者 Min 方）。
7. 博弈树的子树搜索完成后的返回值是针对该子树的局部最优值，但并不一定是全局最优值。

三．极小极大搜索过程

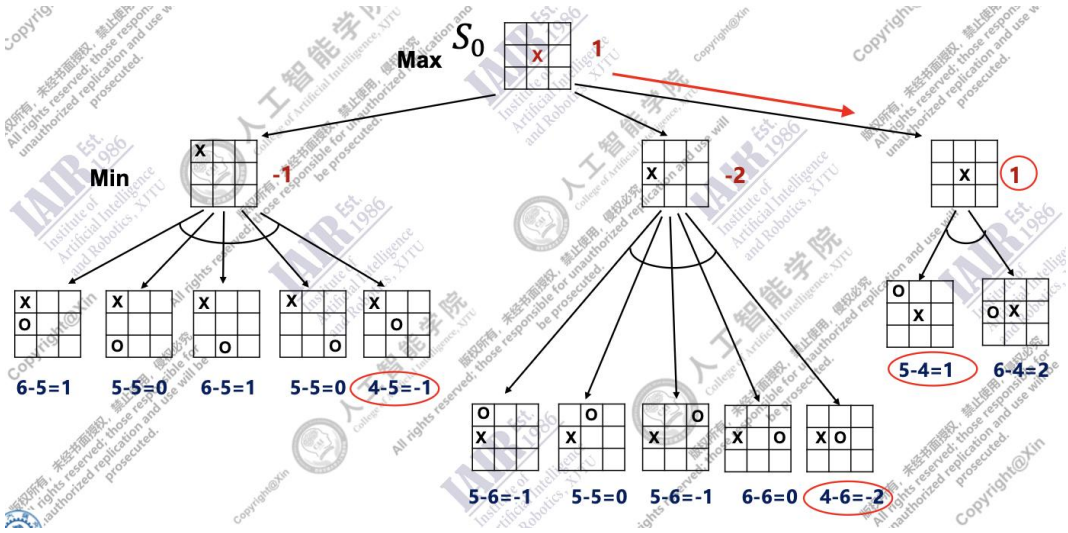
1. 生成 Max 方的 k 步博弈树
2. 定义评估函数，规定：

■ 估价函数 $e(p)$ 的规定：

- 1）若格局p是Max方获胜，则 $e(p)\to+\infty$
- 2）若格局怕是Min方获胜，则 $e(p)\to-\infty$
- 3）对任何一方都不是获胜，则 $e(p)=e_{\max}(p)-e_{\min}(p)$

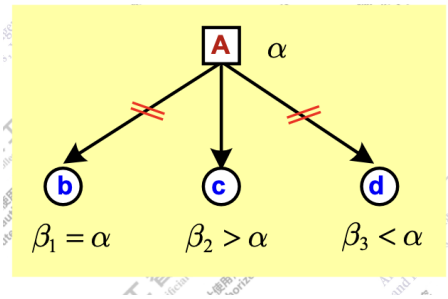
3. 用评估函数计算各叶节点的得分

4. 用极小极大运算自下而上逐层计算节点的得分（其中在 Max 处取最大值，Min 处取最小值）
5. 在根节点处结束操作
6. 例：（k=2 时的井字棋搜索过程）

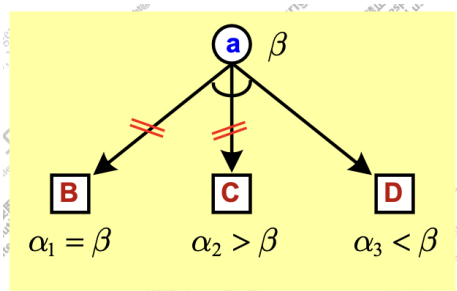


四． α-β 剪枝

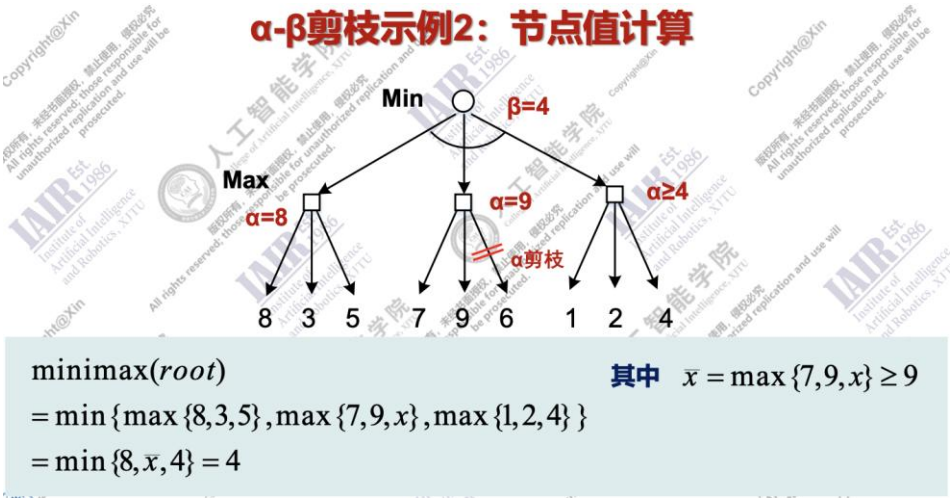
1. 基本思想：
   - **对于一个“或”（or）节点（Max方），取当前子节点中最大的倒推值作为其都推值的下界值，称为α（α≥该最大值）**
   - **对于一个“与”（and）节点（Min方），取当前子节点中最小的倒推值作为其都推值得上界值，称为β（β≤该最大值）**
   - **α：Max方可以搜索到的最好值，β：Min方可以接受的最坏值**
2. α剪枝规则：子节点（and 节点）的β值<=该节点（or 节点）的α值时可以直接删去该子节点。



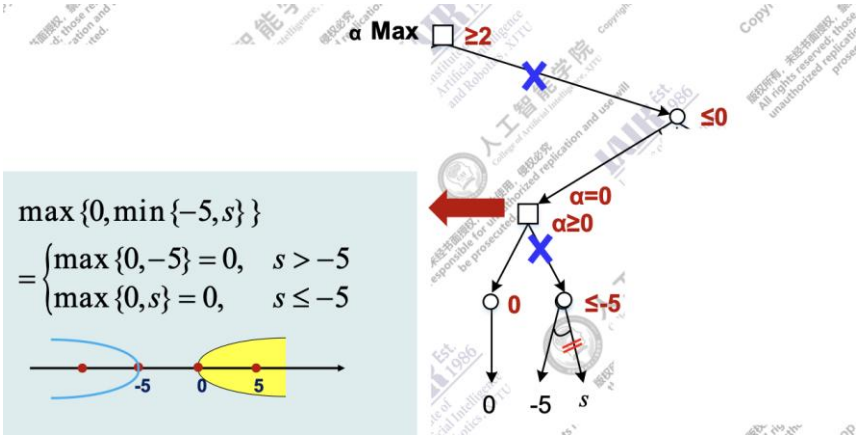
3. β剪枝规则：子节点（or 节点）的α值>=该节点（and 节点）β值时可以直接删去该子节点。



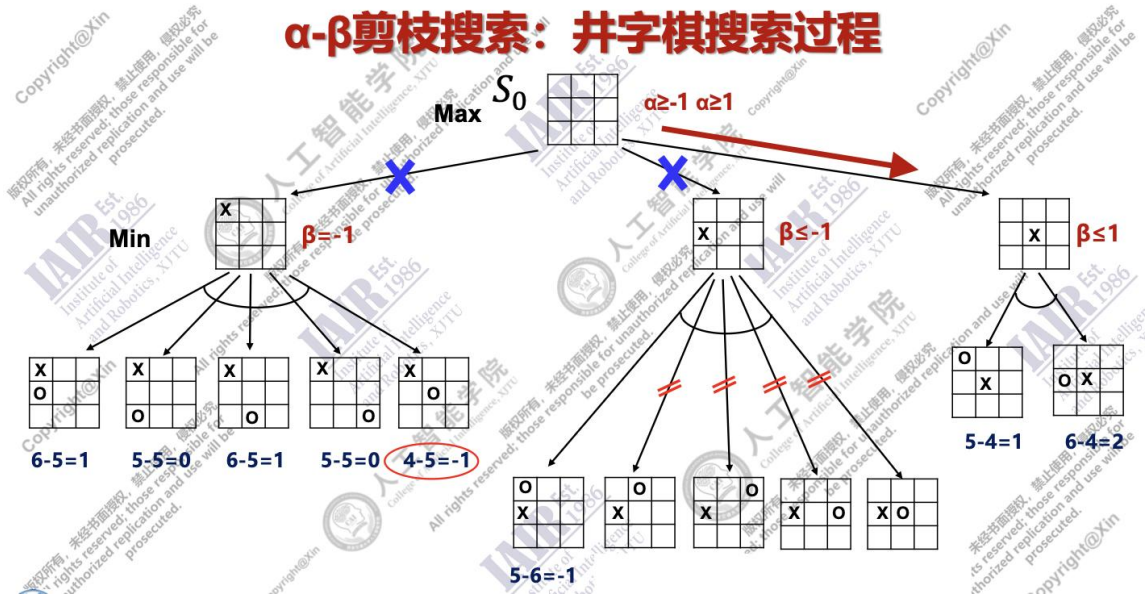
4. 可以用 max 与 min 函数来表示：设未知量为 x，发现 x 并不影响最终结果即可把 x 所在子树剪掉。



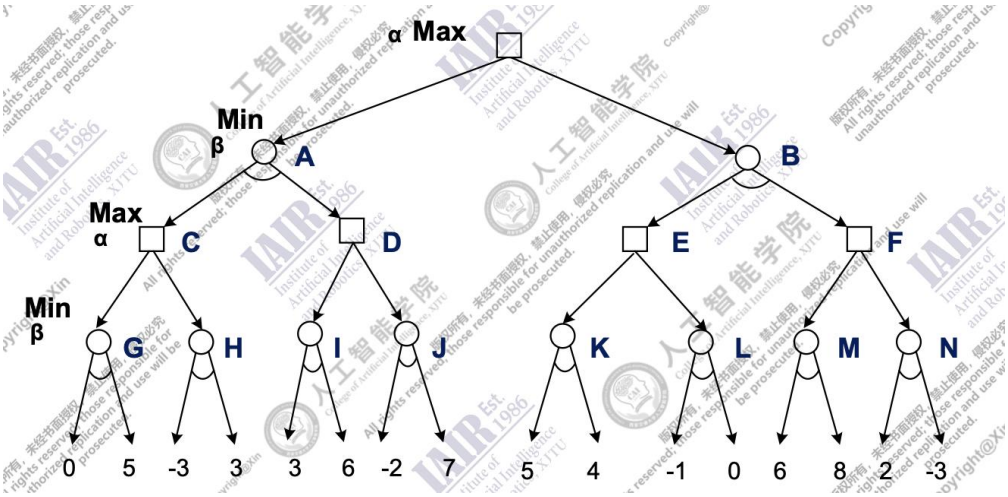
5. 每一次对节点的α或β值进行更改后，要从下到上更新所有节点的值。
6. 相关节点值计算验证：对该节点的子节点中的未知数用字母表示，然后分类讨论取值，并在数轴上表示。

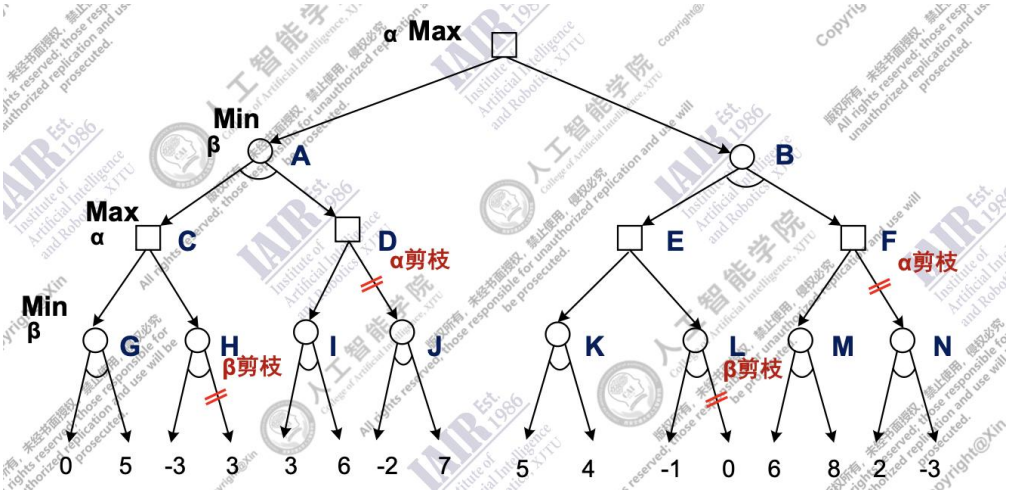


7. 例子：井字棋的 α-β 剪枝：



8. 例题：（课堂练习）





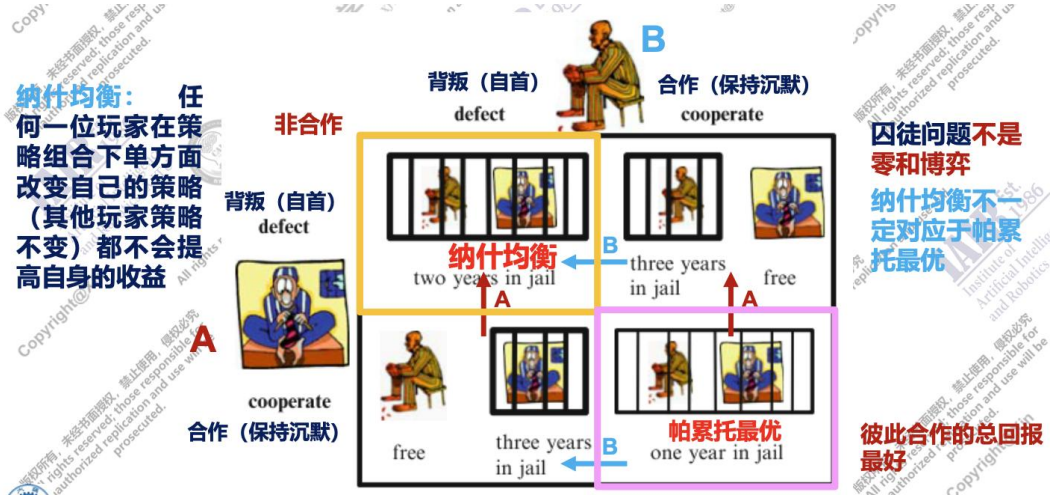
五．非合作博弈

1. 非合作博弈的分类以及对应的均衡概念：

2. 纳什均衡的概念：纳什均衡是在非合作博弈条件下，可能会形成的一种均衡状态，当参与人选定的策略组成纳什均衡后，就会形成一个平衡的局面，在这个平衡的局面中，任何一个参与人单方面地改变自己的策略，只

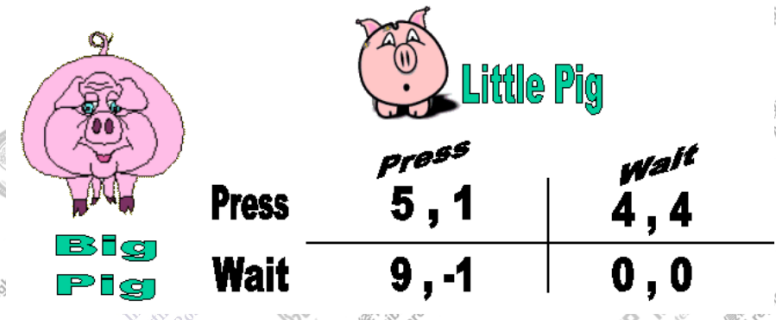
能是自己的收益下降（或不变），绝不可能使自己的收益增加

3. 并不是所有的非合作博弈都有纳什均衡，有可能没有均衡状态，而有纳什均衡存在的博弈也不见得只有一个均衡，可能存在两个或者多个均衡。
4. 囚徒困境：



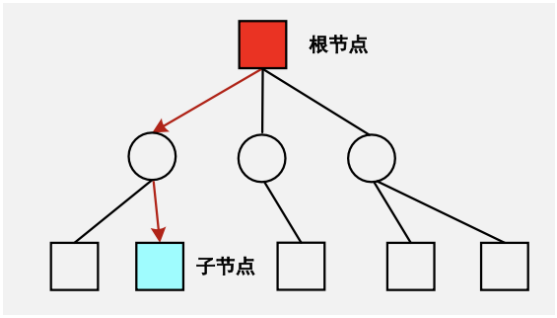
收益矩阵：可以看到，不管在 B 做出什么选择的情况下，A 选择背叛操作能够获得的收益是最大的（对 B 也是同理）。但是我们作为上帝视角可以看到，彼此合作的总回报才是最好的（帕累托最优）。

5. 智猪博弈：不管大猪怎么选，小猪总是要选择等待才会获得最大的收益（但对大猪不是这样，所以不是纳什均衡）。

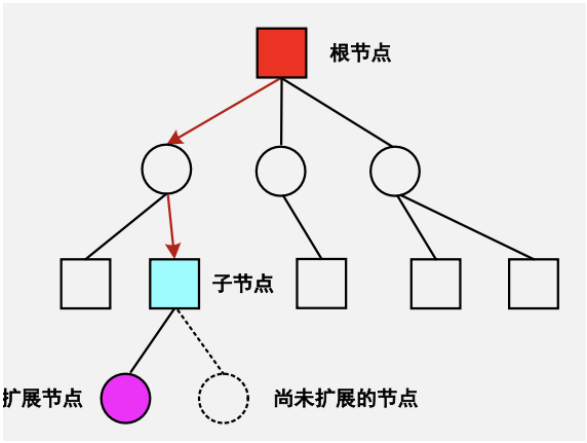


六．蒙特卡洛树搜索

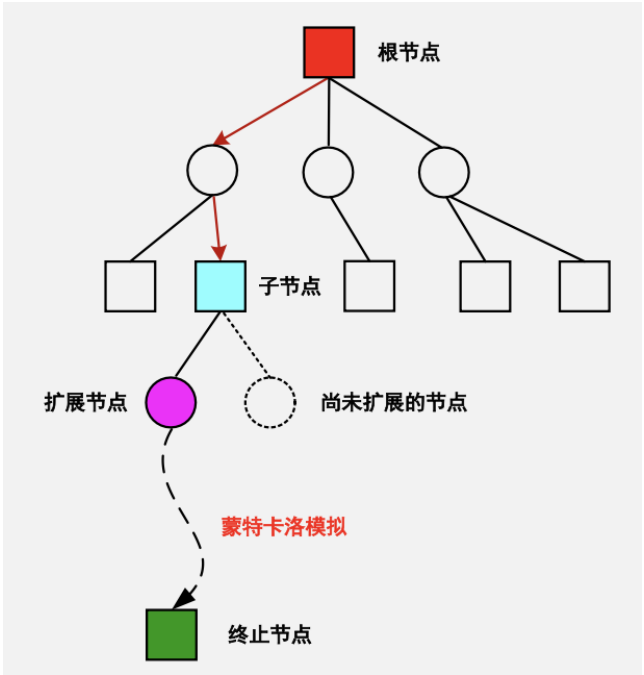
1. 不同于普通的树搜索，蒙特卡洛树搜索边模拟、边探索、边调整。
2. 既有树木搜索的精度，又具有随机抽样的普遍性。
3. 蒙特卡洛树搜索基于两个基本概念：
   - 可以使用随机模拟近似动作的真实值
   - 这些模拟值可以有效地用于将策略调整为最佳优先策略
4. 蒙特卡洛树搜索步骤：
   a) 选择：自上而下找到最急迫需要可扩展节点（非终节点且未被访问就是可拓展的节点；利用置信上限（UCT）公式从上到下找到“最急迫”节点）



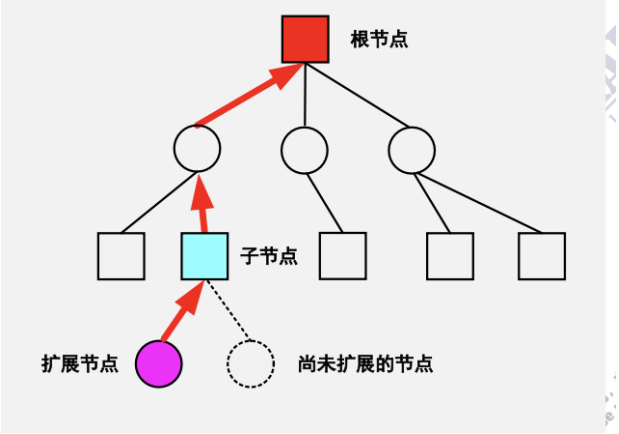
b) 扩展：在选定的节点上随机扩展一个或多个未被扩展的子节点以扩展搜索树。



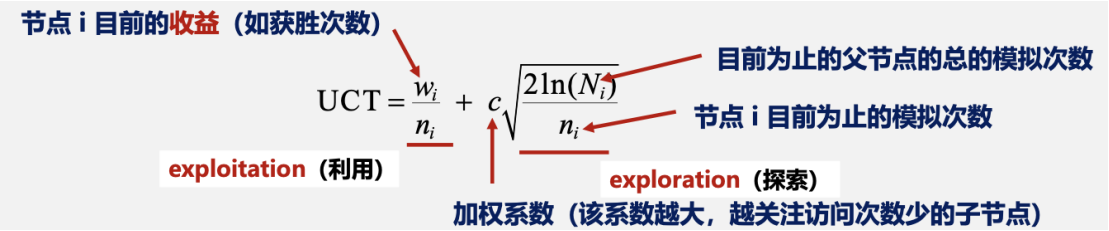
c) 模拟：在新节点上进行多次蒙特卡洛方法模拟，根据结果评估扩展节点的值函数。（蒙特卡洛模拟先随机抽样，再计算在目标区域内出现的次数）



d) 回溯：根据模拟结果，向上更新路径上的节点。

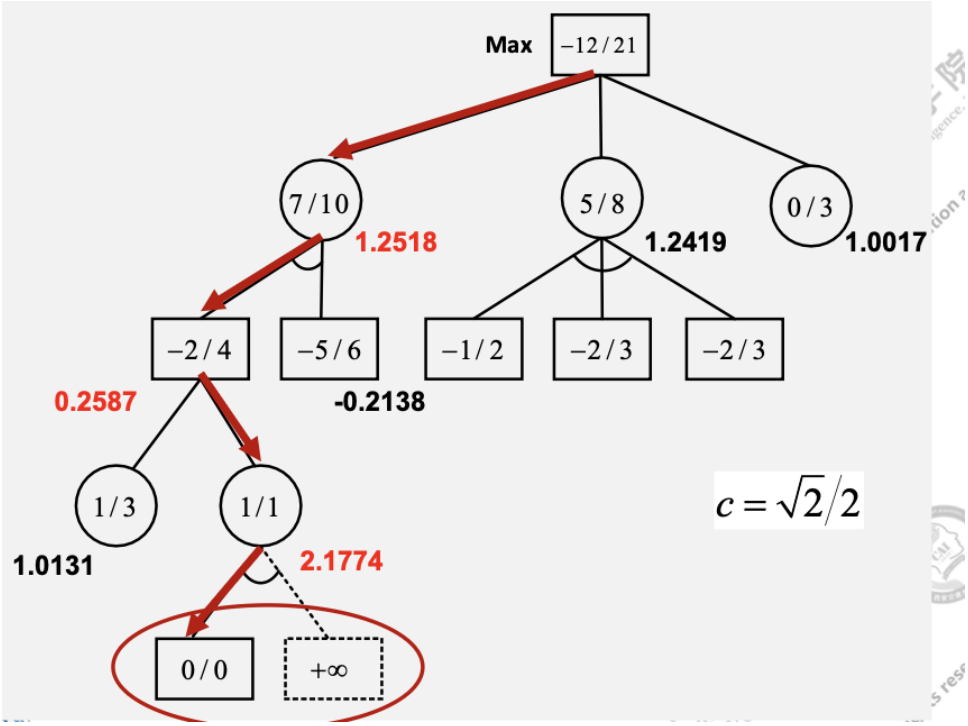


5. UCT（Upper Confidence Bounds for Trees）的计算公式（综合考虑当前胜率 exploitation、潜在胜率 exploration）：

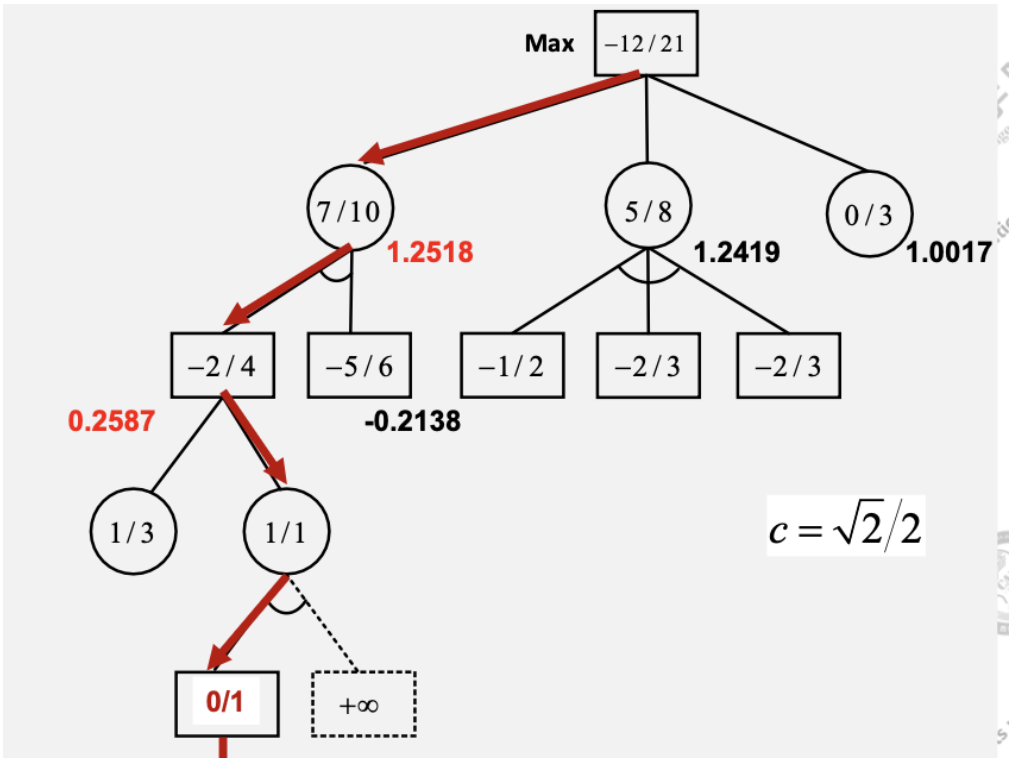


6. 搜索实例：

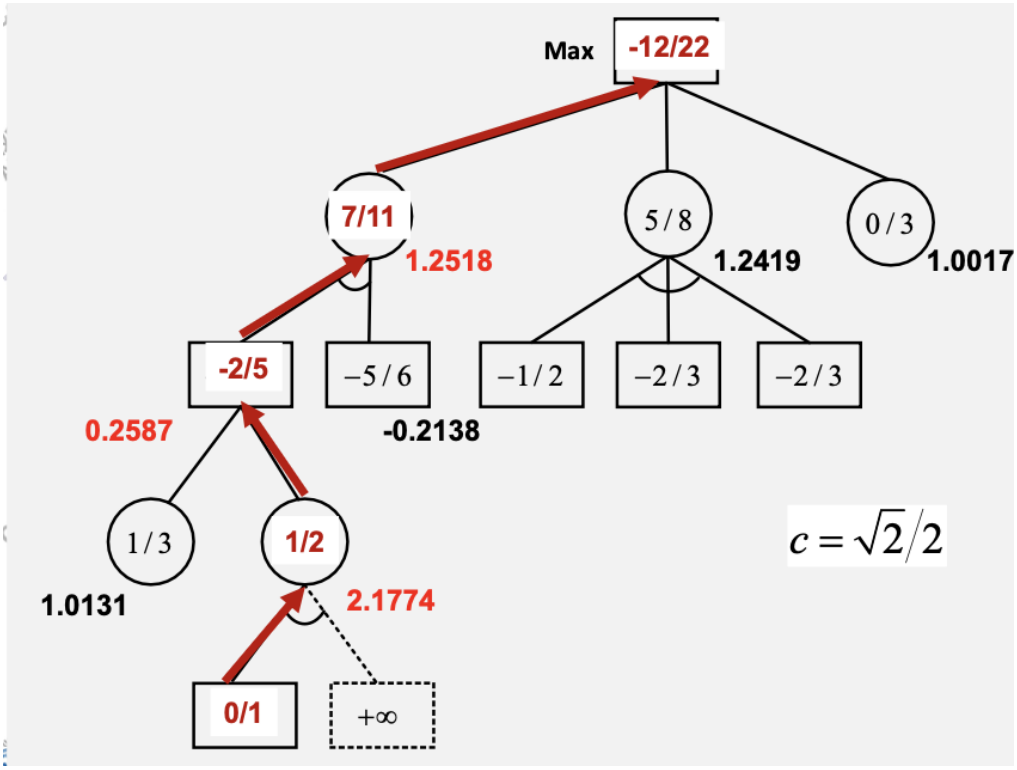
   a) 先找到 UCT 最大的未拓展节点并拓展其子节点



   b) 再对其子节点进行蒙特卡洛模拟（该题模拟了一次）

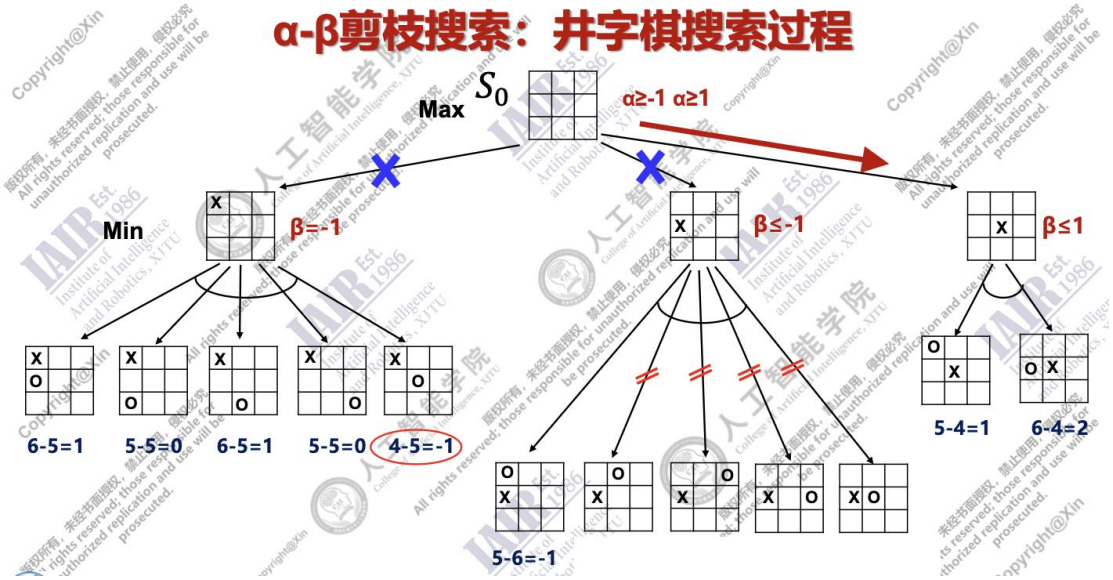


c) 再向上进行更新回溯



七．2024 年真题

1. （选择）囚徒困境不是零和博弈（一般来说是变和博弈）。
2. （简答）用极大极小化搜索求一个节点的取值，用$\alpha - \beta$剪枝对四层的博弈树进行剪枝（PPT 原图，改数字）：



想做练习的话可以看“四-8”，有课上的题目与答案。