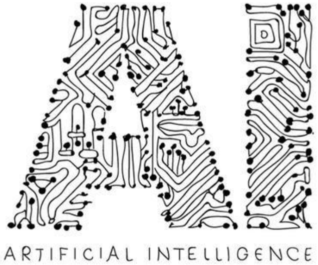
人工智能

沙瀛

信息学院

2020.3



如何用博弈论来预判他人行为？

- https://weibo.com/tv/v/lySlK0hVq?fid=1034:4482186494017557

•目录

博弈树的启发式搜索

- 博弈树
- 极大极小分析法
- α-β剪枝技术

博弈树概念

- "二人零和、全信息、非偶然"博弈的特点
  - 博弈A、B双方轮流行动，博弈结果只有A胜、A负、平局三种情况。

•概念

博弈树概念

- 博弈双方都了解当前和格局及过去的历史。
- 任何一方在采取行动前都要根据当前的实际情况进行得失分析，选取自己最有利而对对方最不利的对策，不存在“碰运气”的偶然因素。

•概念

博弈树

- 在博弈过程中，己方的各种攻击方案为“或”关系，而对方的应着方案为“与”关系。描述博弈过程的“与\或”树称为博弈树。

注：“与\或”树始终是站在某一方的立场上得出来的。

•概念

博弈树特点

- 博弈树中的初始节点为博弈的开始格局
- 在博弈树中，“或”节点和“与”节点逐层交替出现。自己一方扩展的节点为“或”节点，对方扩展的节点为“与”节点。
- 所有能使自己一方获胜的终局都是本原问题，相应的节点是可解节点；所有使对方获胜的终局都是不可解节点。

•概念

极大极小分析法

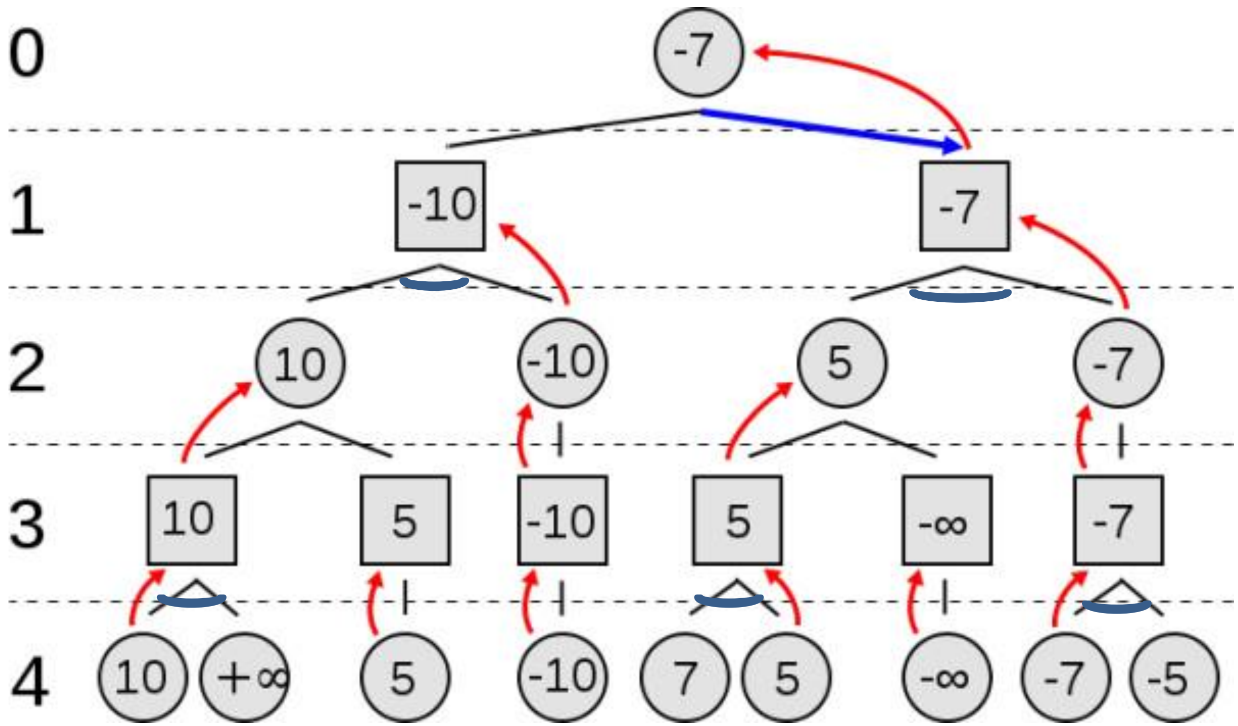
- 根据所求解问题的特殊信息设计合适的估价函数，计算当前博弈树中所有端节点的得分，该分数称为静态估价值。
- 根据端节点的估值推算出其父节点的分数：

极大极小分析法

- 若父节点为“或”节点，则其分数等于其所有子节点分数的最大值
- 若父节点为“与”节点，则其分数等于其所有子节点分数的最小值。
- 计算出的父节点的分数值称为**倒推值**。
- **如果一个方案能获得较大的倒推值，则它就是当前最好的行动方案。**

•例子

极大极小分析法



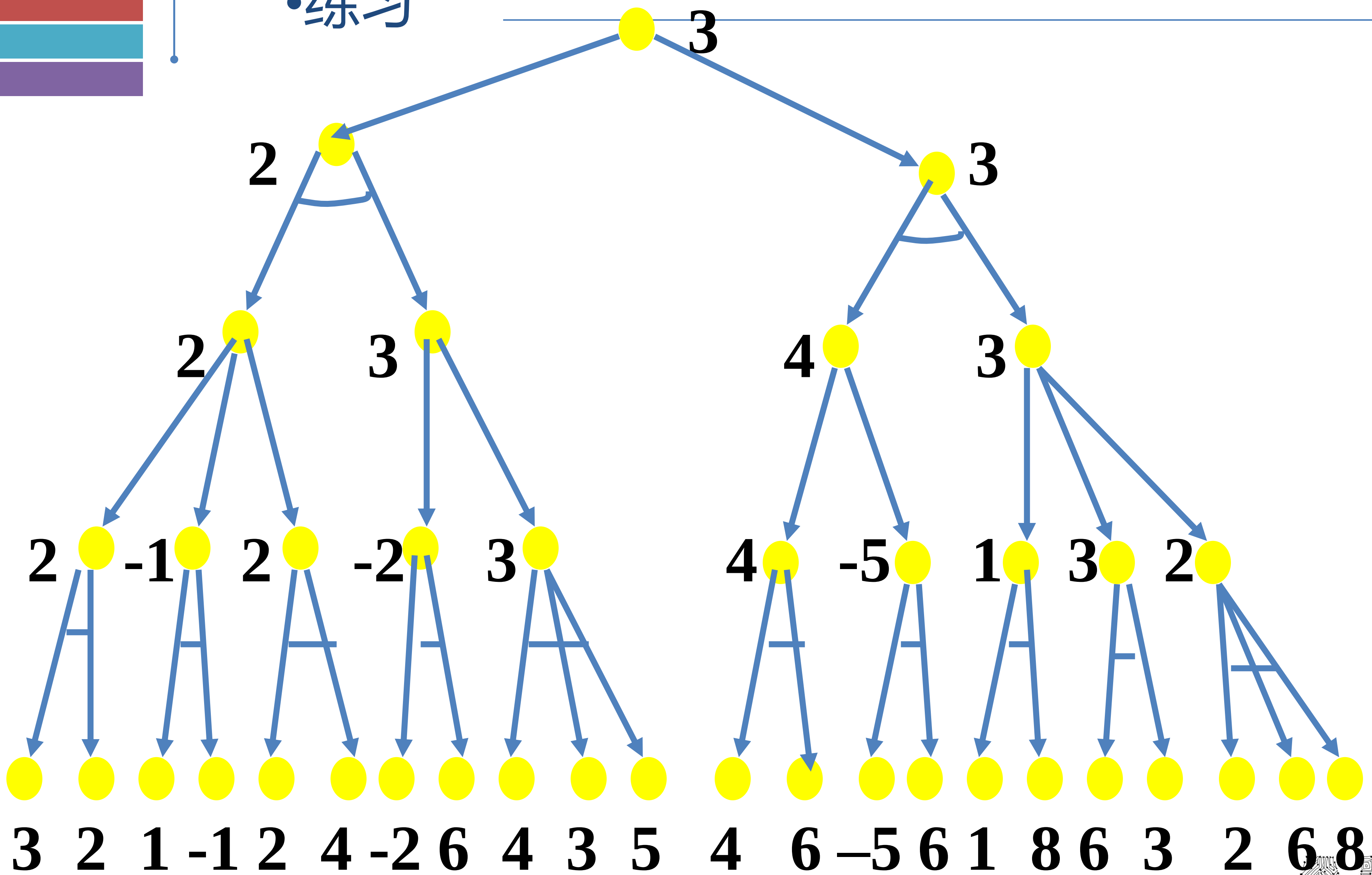
•练习

极大极小分析法

- 例、倒推值计算

博弈树搜索方法

- 练习



•例子

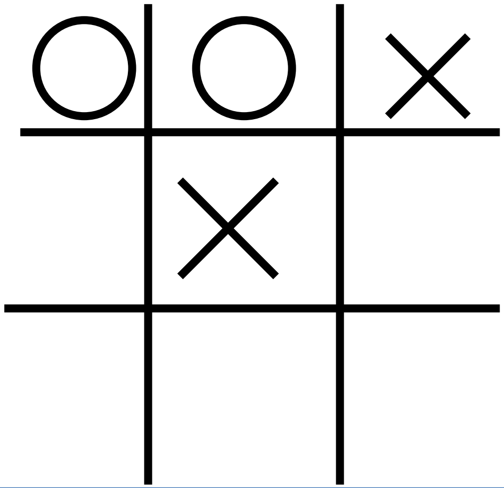
例、九宫棋游戏：由A,B二人轮流在九宫格中放置棋子，谁先使自己的三个棋子构成一条直线谁就获胜。

- 解：设A方的棋子用×表示，B方的棋子用o表示，假设每次考虑两步，A为开局方，定义估价函数（设估价函数f(n)为启发函数h(n)）如下：

•例子

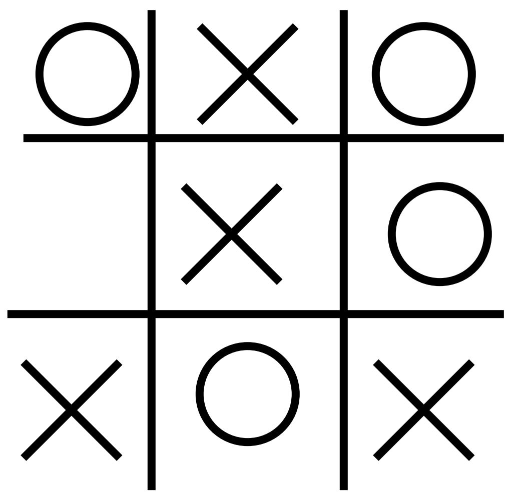
设：

若某条直线上只有某方(称x方)的棋子或无任何棋子，则称该直线为x方的一条赢线，win(x)表示x方的所有赢线数。如下图，win(A)=4, win(B)=2。



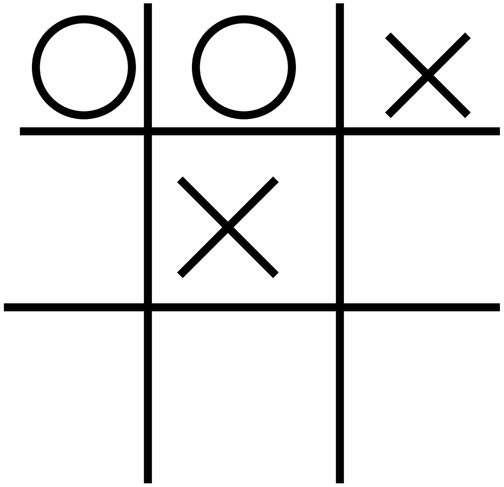
•例子

- 若n为非终局节点，则
- $h(n)=win(A)-win(B)$
- 若n为和局，则$h(n)=0$，如下图所示。



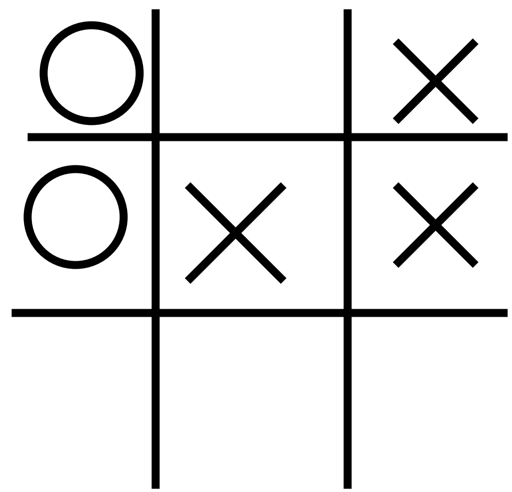
•例子

- 若n为A取胜的终局节点，则$h(n)=+\infty$，如下图所示。



•例子

- 若n为A失败的终局节点，则$h(n)=-\infty$，如下图所示。

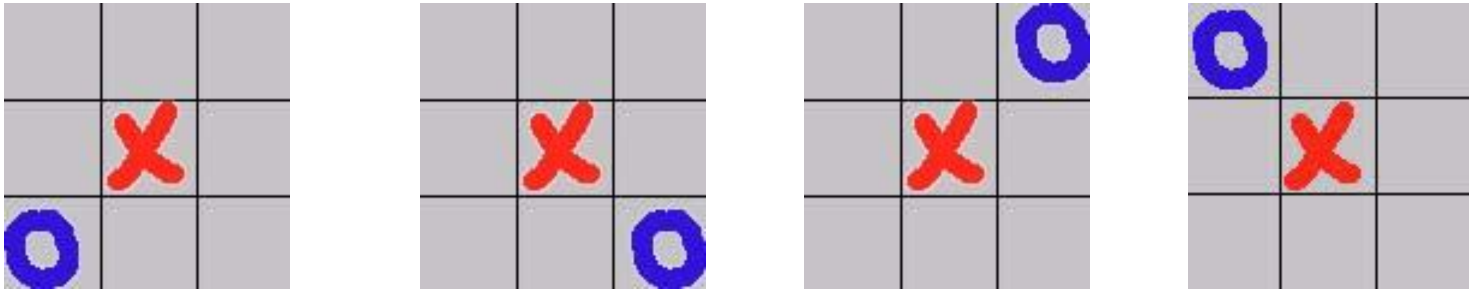


•例子

求解过程

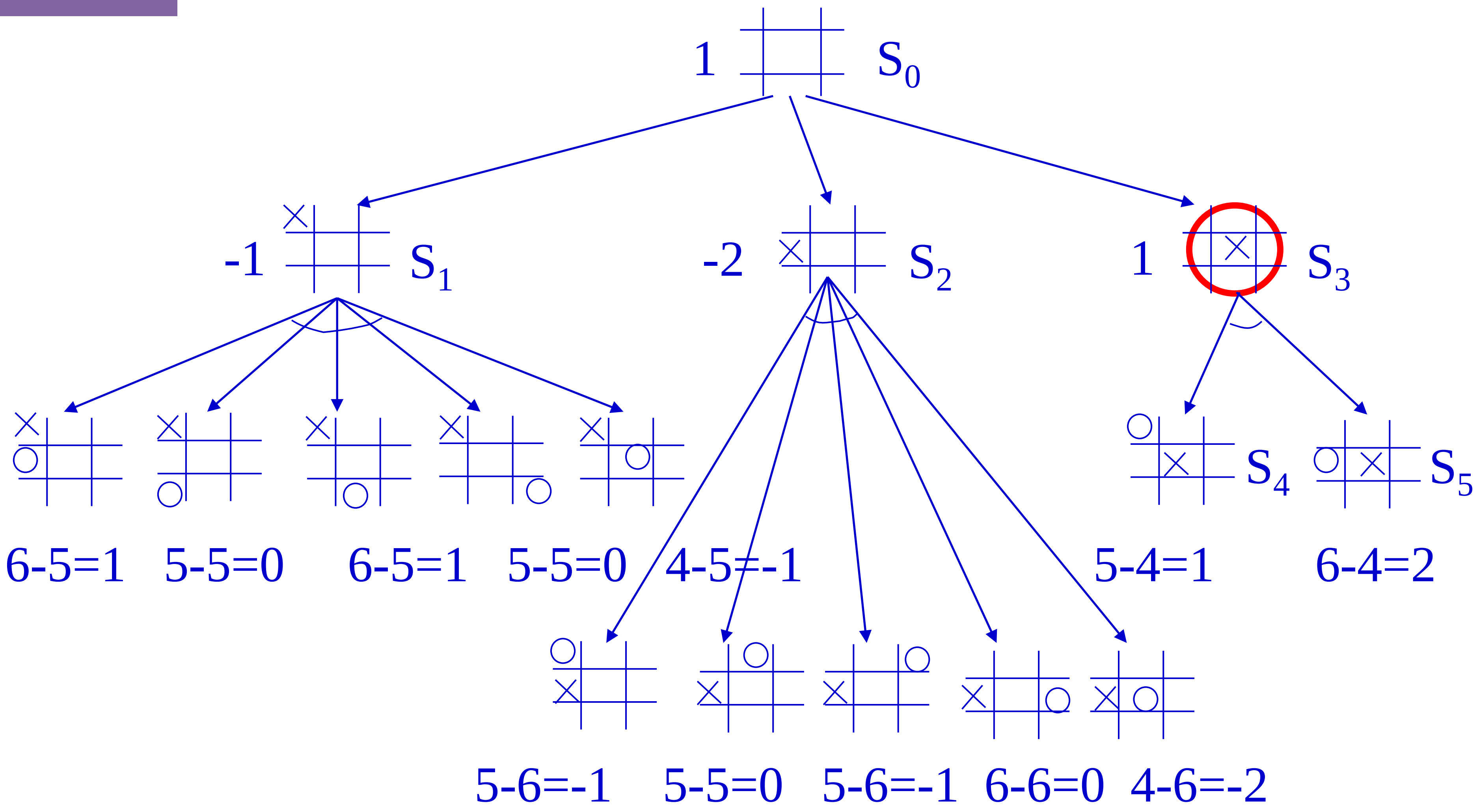
- 则第1步棋的搜索树如下：

注意：对称棋局认为是相同棋局



博弈树搜索方法

• 例子



1 S0
-1 S1
-2 S2
1 S3
S4
S5
6-5=1
5-5=0
6-5=1
5-5=0
4-5=-1
5-4=1
6-4=2
5-6=-1
5-5=0
5-6=-1
6-6=0
4-6=-2

•概念

α-β 剪枝技术

- 一个“与”节点取当前子节点的最小值作为其倒推值的上界，称该值为β值。
- 一个“或”节点取当前子节点的最大值作为其倒推值的下界，称该值为α值。

•概念

α-β剪枝算法

– 如果“或”节点x的α值不能降低其父节点的β值，即：

$\alpha \geqslant \beta$

则应停止搜索节点x的其余子节点，并使x的倒推值为α。这种技术称为β剪枝。

•概念

α-β剪枝算法

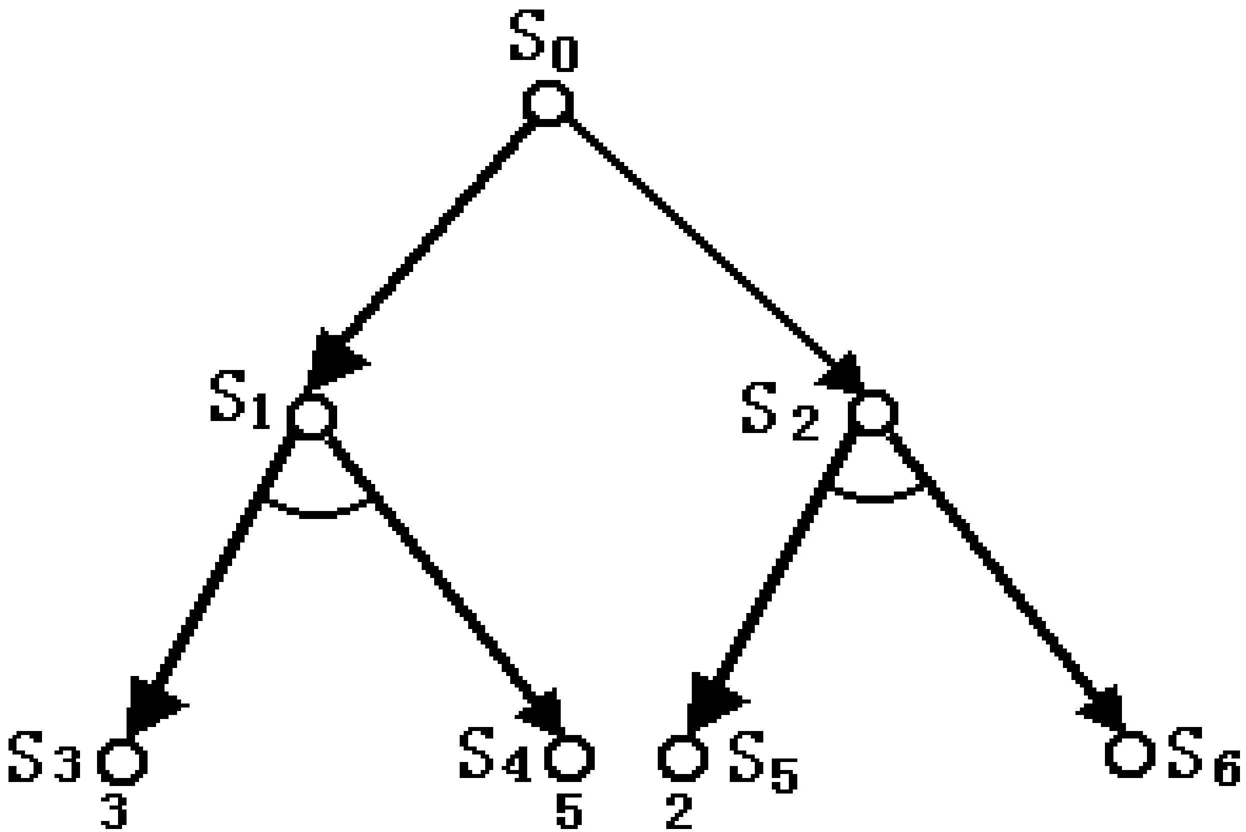
- 如果"与"节点X的β值不能升高其父节点的α值，即：

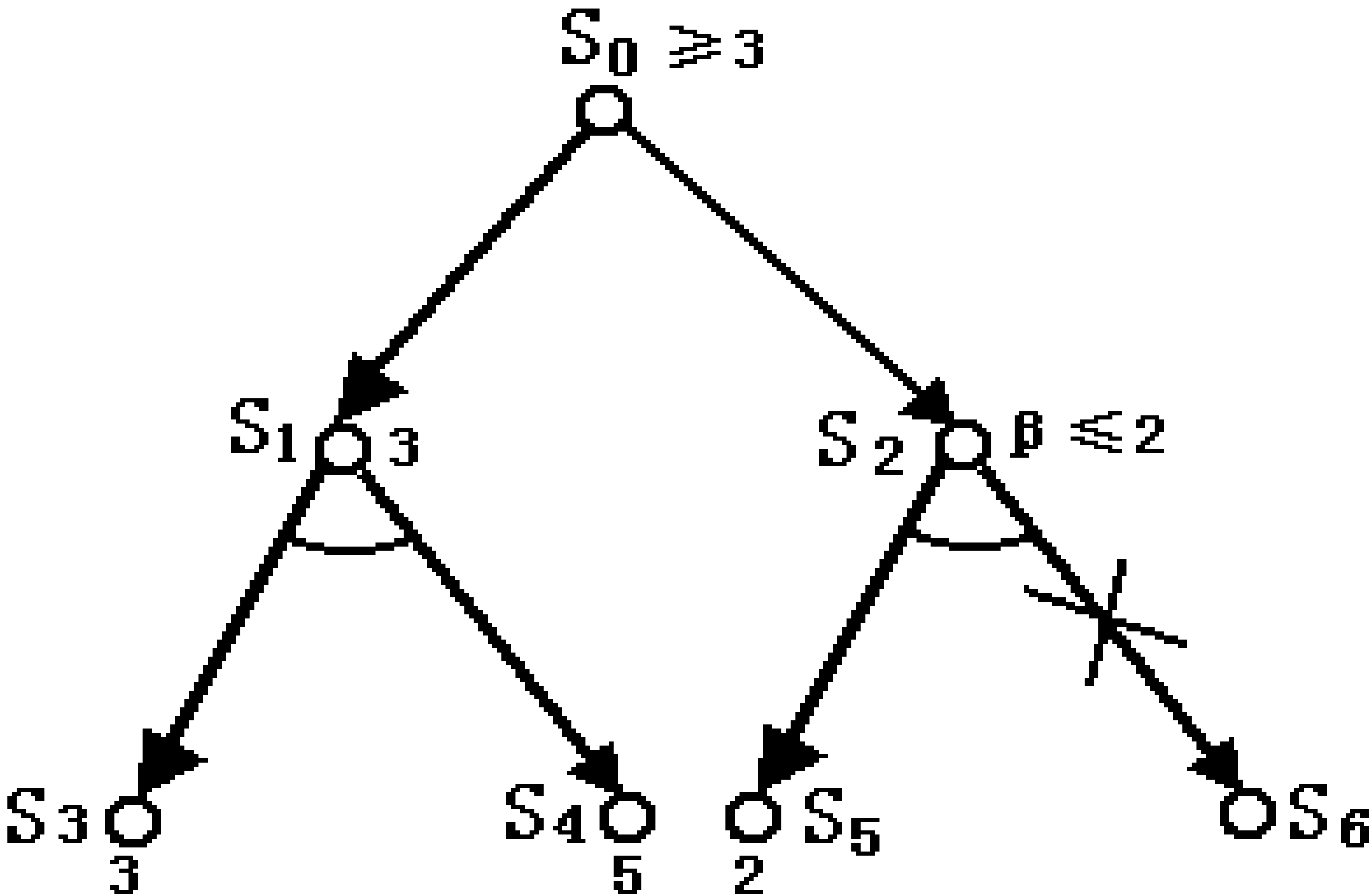
$\beta \leqslant \alpha$

则应停止搜索节点X的其余子节点，并使X的倒推值为β。这种技术称为α剪枝。

例子

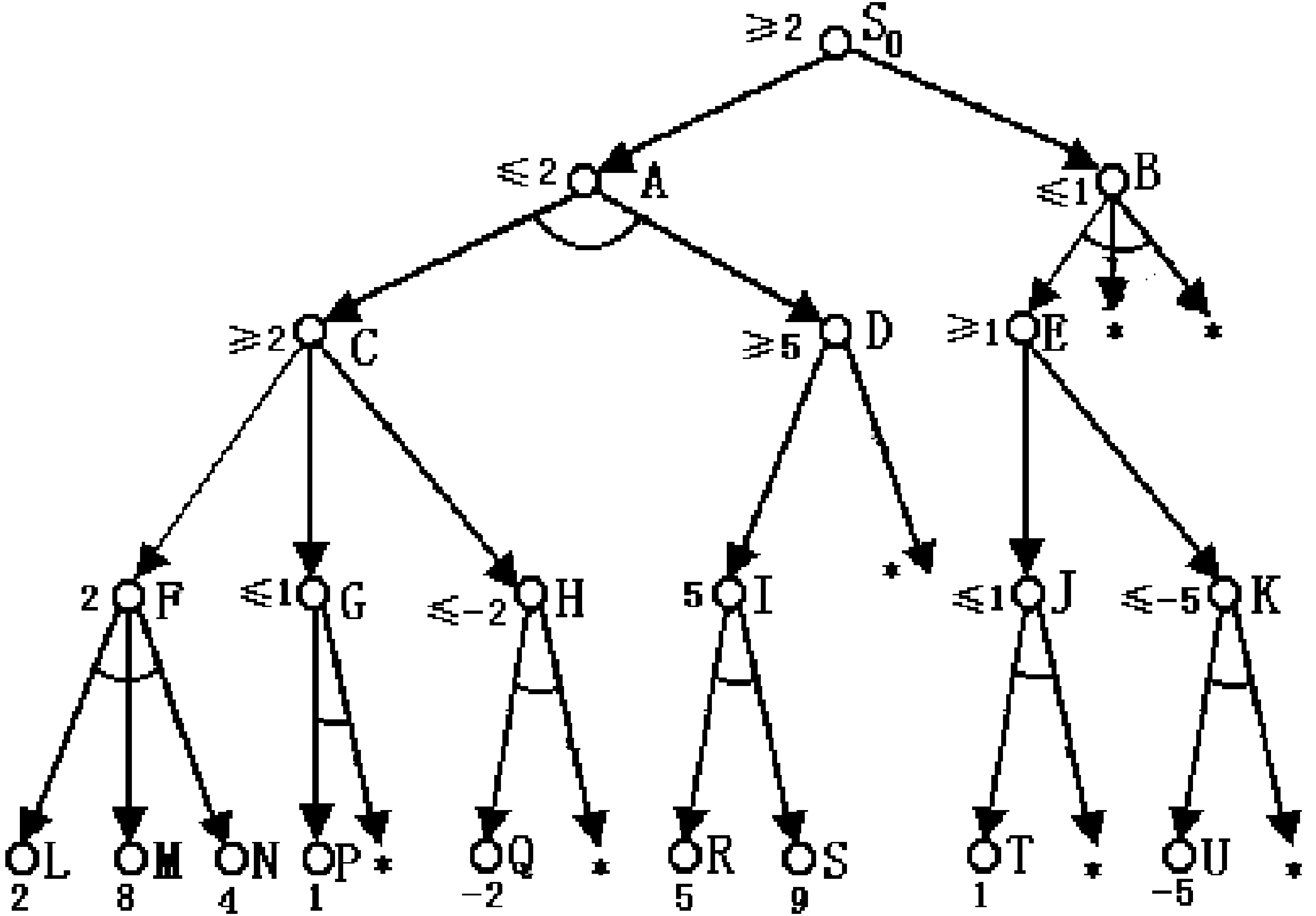
- 例、对于下面的博弈树a进行剪枝



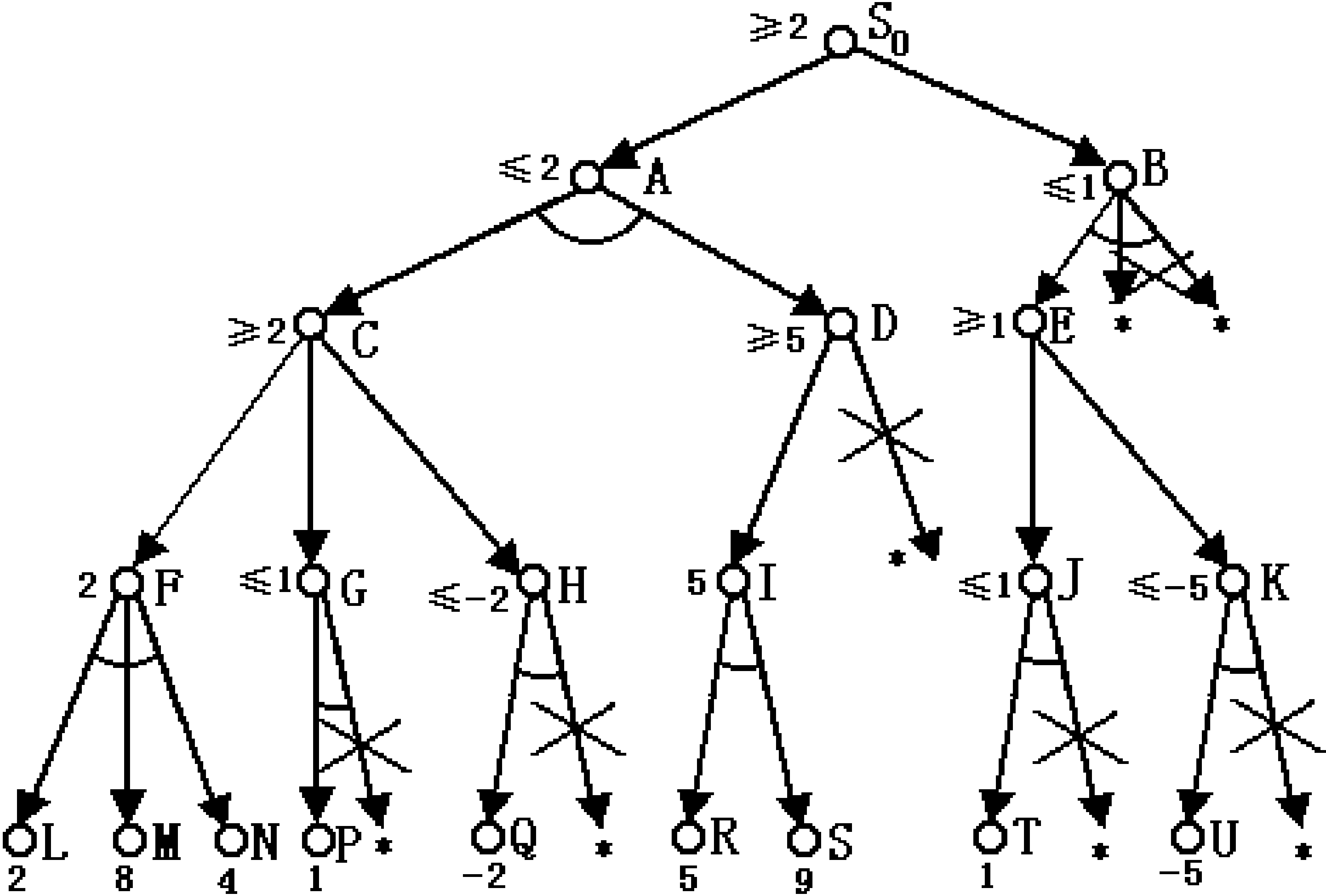

S_0 ≥3
S_1 3
S_2 β ≤2
S_3
3
S_4
5
2
S_5
S_6

例子

- 例、对于下面的博弈树进行剪枝



α－β 剪枝示例

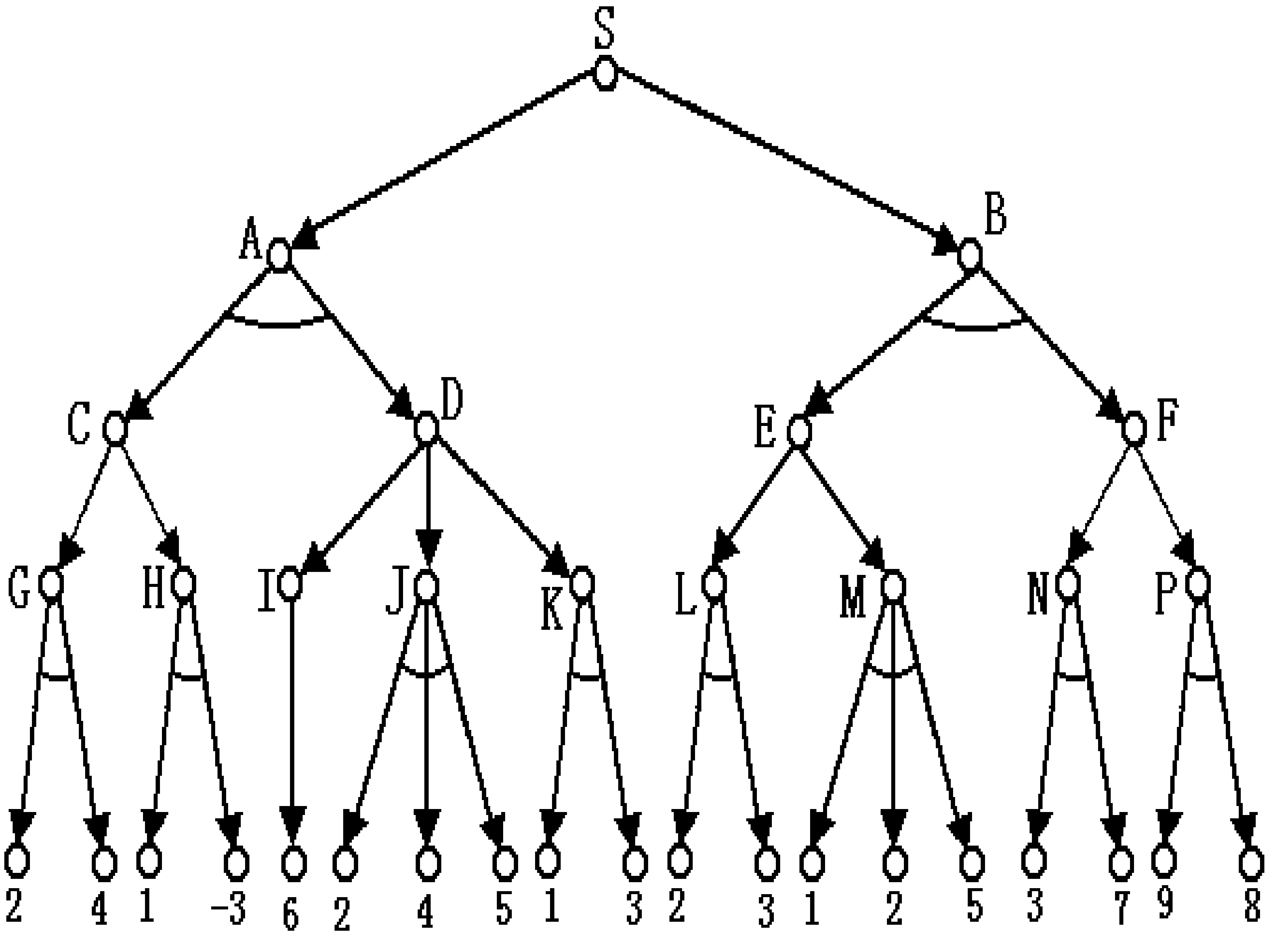


α − β 剪枝示例

练习

课堂练习

- 请根据端节点的估值计算下述博弈树的各节点的倒推值，并利用α-β剪枝技术剪去不必要的分枝。


S
A
B
C
D
E
F
G
H
I
J
K
L
M
N
P
2
4
1
-3
6
2
4
5
1
3
2
3
1
2
5
3
7
9
8

蒙特卡洛树搜索

- 蒙特卡洛树搜索就是一种随机搜索算法
- 蒙特卡洛树搜索（MCTS）是所有现代围棋程序的核心组件。在此之上可以加入各种小技巧（如UCT，RAVE/AMAF，Progressive Bias，Virtual win & lose，Progressive Widening，LGR，Criticality 等等）和大改进（如 AlphaGo 的策略网络和价值网络）

蒙特卡洛树搜索

- MCTS四步法

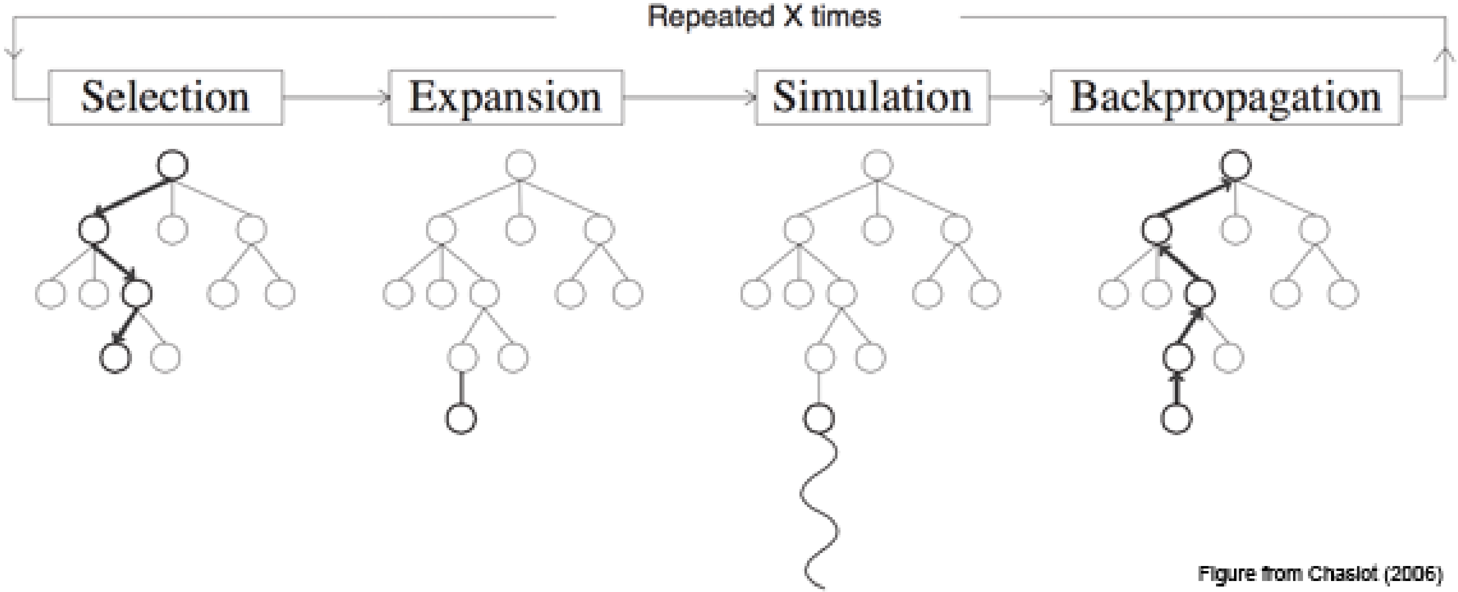
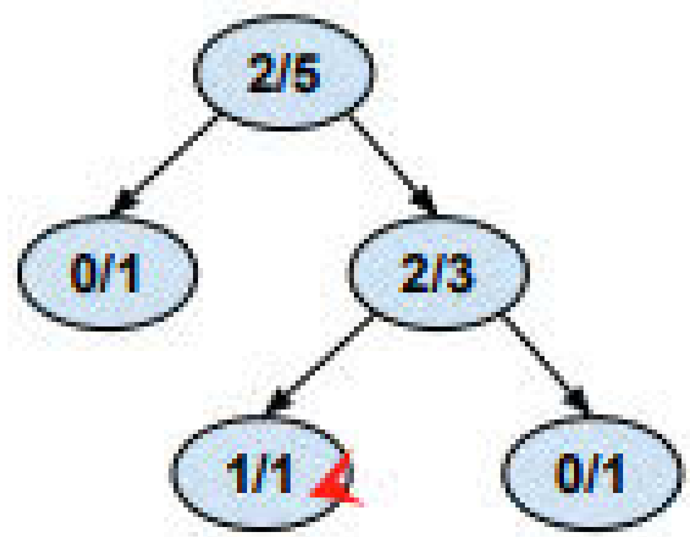


Figure from Chaslot (2006)

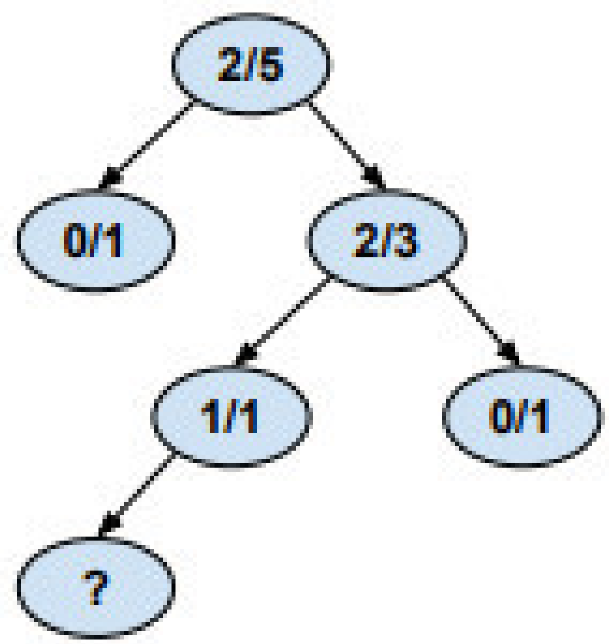
1、Selection

- 从根节点开始，检查当前状态下所有可行的动作，按照某种策略选择一种动作，进入下一个状态
- 节点中的信息为胜利次数/总的访问次数



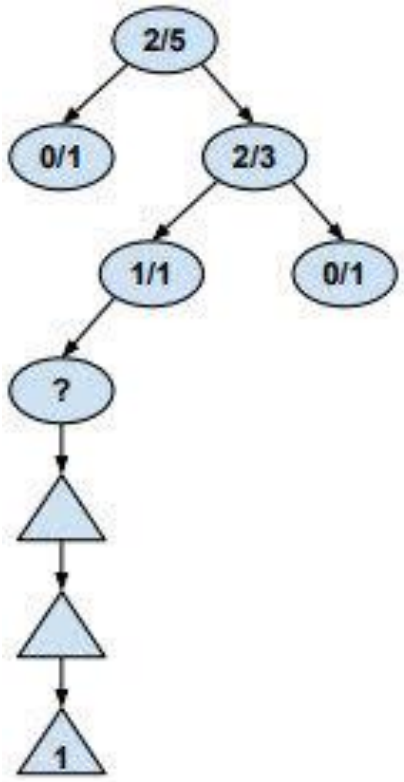
2、Expansion

- Expansion即从找到未被expansion的动作的那个节点处创建一个新的孩子节点表示执行该动作后的状态
- 注意每次只会在expansion阶段创建出新的节点



3、Simulation

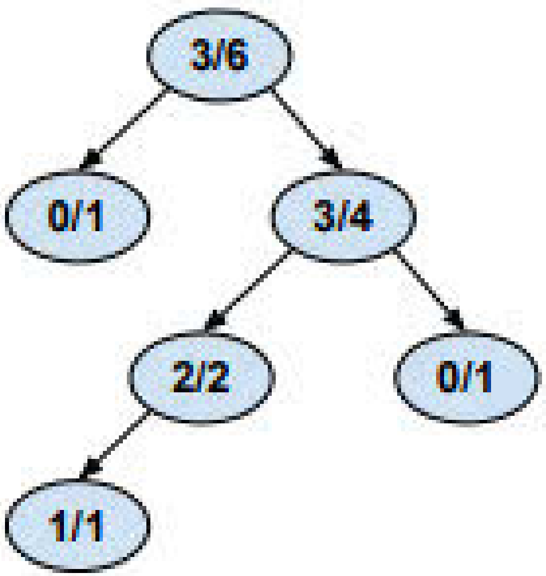
- Simulation就是做推演模拟，即转入到新创建的状态节点，按照某种策略选择动作，如随机选择可行的动作或按照某种指导原则选择可行动作，直到状态结束，即胜利或失败
- 然后update新节点中的相关状态，若此次simulation的结果为胜利则设胜利次数为1，若为失败则设为，设该节点的访问次数为1。



应用

4、Backpropagation

- 最后把新的simulation的结果反向传播回根节点，即从改叶节点原路返回至根节点，在返回路径上的每个节点访问次数和胜利次数加上新的叶节点的访问次数和胜利次数。



上限置信区间

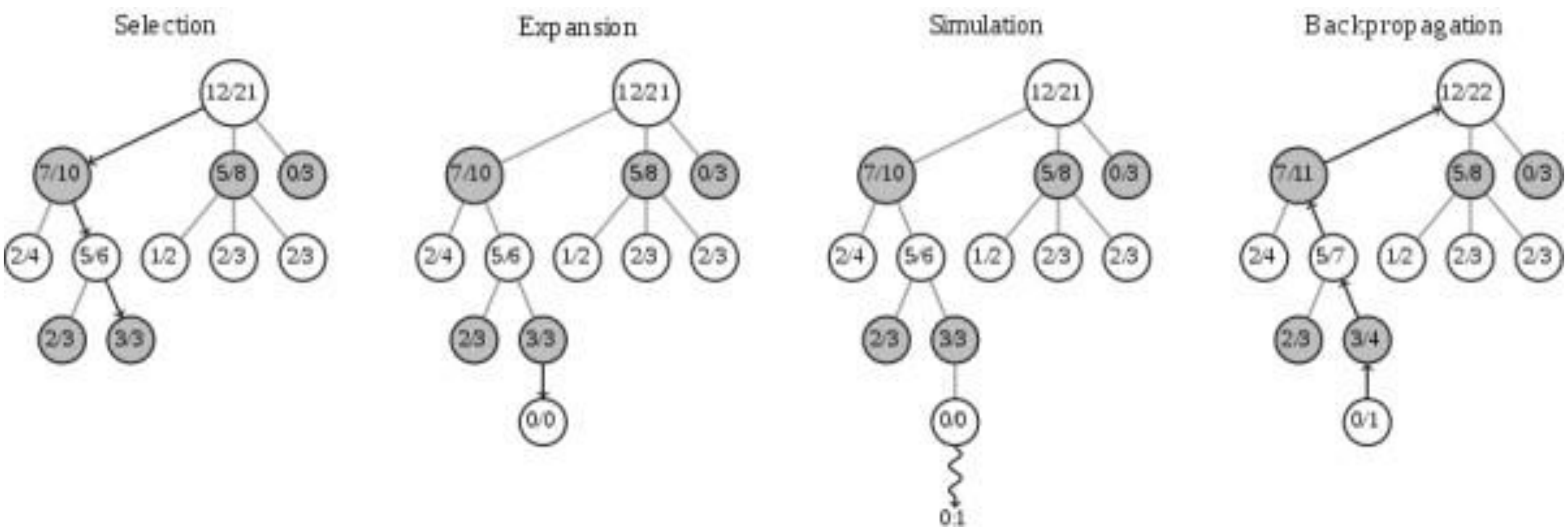
- 上面selection中提到用一种策略选择一种动作，那么这个策略的到底是什么呢?
- 常用的上限置信区间策略（Upper Confidence Bound）

$$score = x_{child} + C \cdot \sqrt{\frac{\log(N_{parent})}{N_{child}}}$$

上限置信区间

- 其中X是节点的当前胜率估计（注意，要考虑当前是黑棋走还是白棋走），N是节点的访问次数。C是一个常数。C越大就越偏向于广度搜索，C越小就越偏向于深度搜索。

上限置信区间的例子


Selection
12/21
7/10
5/8
0/3
2/4
5/6
1/2
2/3
2/3
2/3
3/3
Expansion
12/21
7/10
5/8
0/3
2/4
5/6
1/2
2/3
2/3
2/3
3/3
0/0
Simulation
12/21
7/10
5/8
0/3
2/4
5/6
1/2
2/3
2/3
2/3
3/3
0/0
0/1
Backpropagation
12/22
7/11
5/8
0/3
2/4
5/7
1/2
2/3
2/3
2/3
3/4
0/1

UCB计算过程

- 那么我们首先需要在 7/10、5/8、0/3 之间选择：

1. 其中 7/10 对应的分数为 $7/10 + C \cdot \sqrt{\frac{\log(21)}{10}} \approx 0.7 + 0.55C$。
2. 其中 5/8 对应的分数为 $5/8 + C \cdot \sqrt{\frac{\log(21)}{8}} \approx 0.625 + 0.62C$。
3. 其中 0/3 对应的分数为 $0/3 + C \cdot \sqrt{\frac{\log(21)}{3}} \approx 0 + 1.00C$。
4. 可以注意到，C越大，就会越照顾访问次数相对较少的子节点。

UCB计算过程

- 如果 C 比较小，我们将会选择 7/10，接着就要在 2/4 和 5/6 间选择。注意，由于现在是白棋走，需要把胜率估计倒过来：

1. 其中 2/4 对应的分数为 $(1-2/4)+C\cdot\sqrt{\frac{\log(10)}{4}}\approx 0.5+0.76C$。
2. 其中 5/6 对应的分数为 $(1-5/6)+C\cdot\sqrt{\frac{\log(10)}{6}}\approx 0.17+0.62C$。

AlphaGo的蒙特卡洛树搜索

- AlphaGo的策略网络，可以用于改进上述的分数公式，让我们更准确地选择需扩展的节点。而AlphaGo的价值网络，可以与快速走子策略的模拟结果相结合，得到更准确的局面评估结果

本节结束！