博弈树搜索

TA 陈姝睿

2020.11.27

博弈树搜索

- 理论课内容回顾
  - 二人零和博弈问题
  - 博弈树
  - Minimax搜索
  - Alpha-beta剪枝
- 实验课任务与报告提交

二人零和博弈问题

- 两名player轮流行动，行动的个数有限
- 确定性，不存在随机性
- 信息完备性，博弈双方知道所处状态的全部信息
- 零和性：结局有三种可能：playerA获胜、playerB获胜、平局（或两种可能，无平局），一方的损失相当于另一方的收益，总收益为0。

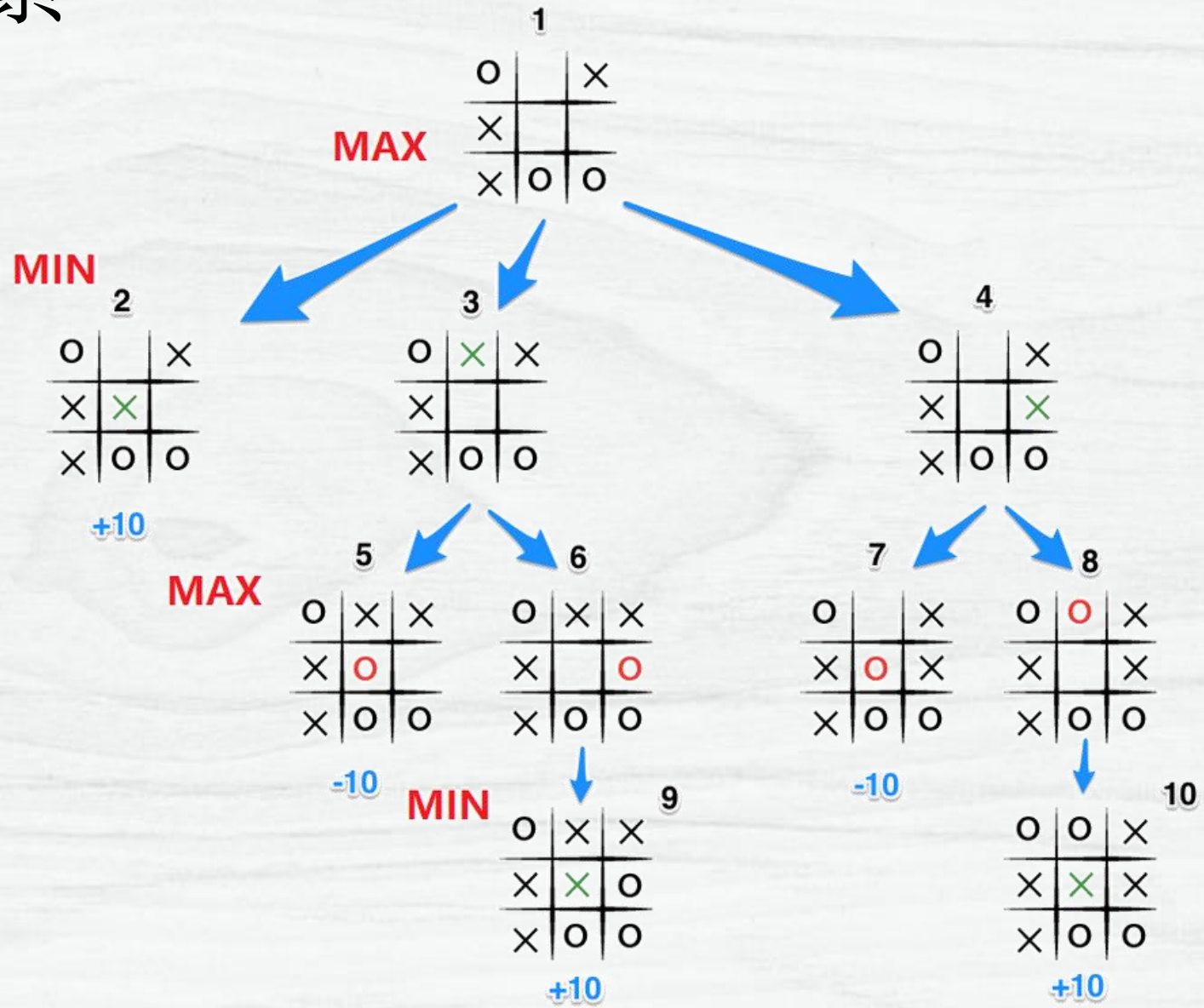
博弈树

- 内部节点（interior node）和叶子节点（leaf node）：表示问题的状态（state）
- 行动（action）：扩展节点。
- 双方轮流扩展节点：两个player的行动逐层交替出现。
- 评价函数（evaluator）：对当前节点的优劣得分。
- 博弈树的值（gametree value）：博弈树搜索的目的，找出对双方都是最优的子节点的值。

Minimax搜索

- playerA和playerB的行动逐层交替
- A和B的利益关系对立，即假设A要使分数更大，B就要使分数更小；
- A和B均采取最优策略。

Minimax搜索

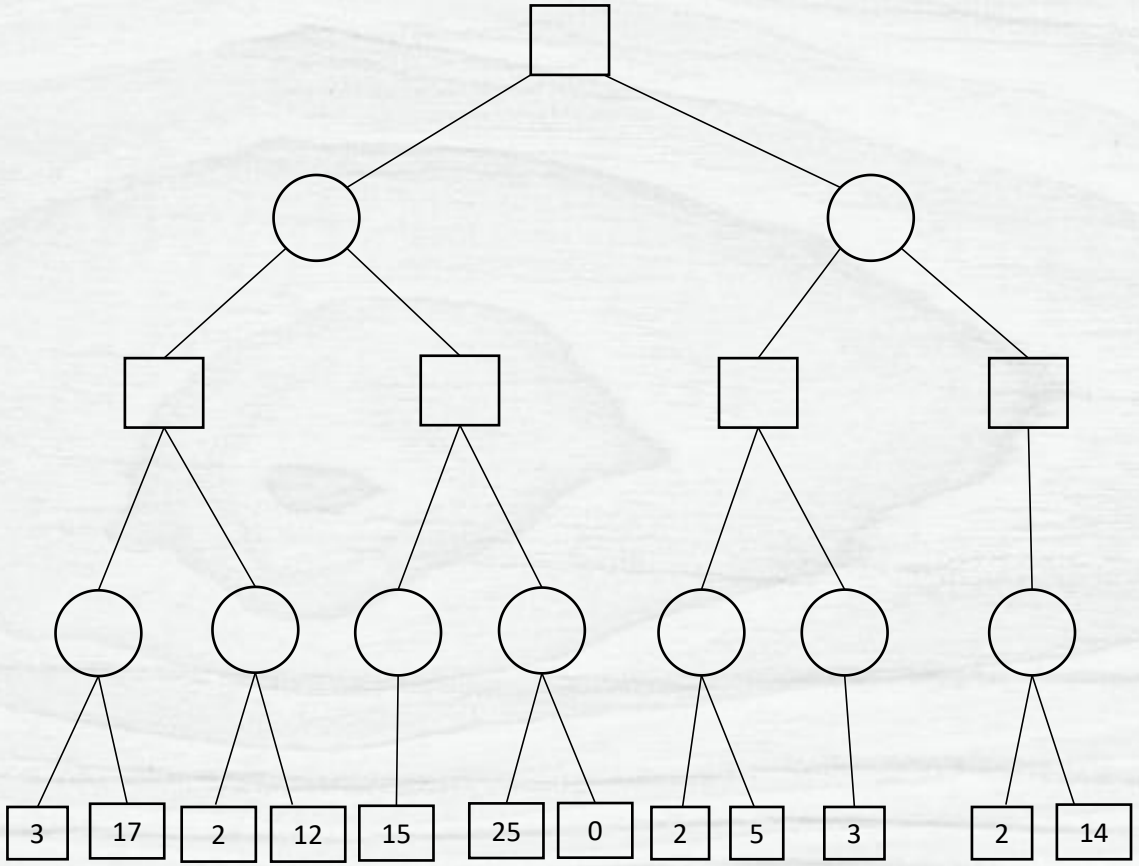


```
DFMiniMax(n, Player) //return Utility of state n given that
                     //Player is MIN or MAX

If n is TERMINAL
Return V(n) //Return terminal states utility
            //(V is specified as part of game)

//Apply Player's moves to get successor states.
ChildList = n.Successors(Player)
If Player == MIN
    return minimum of DFMiniMax(c, MAX) over c ∈ ChildList
Else //Player is MAX
    return maximum of DFMiniMax(c, MIN) over c ∈ ChildList
```

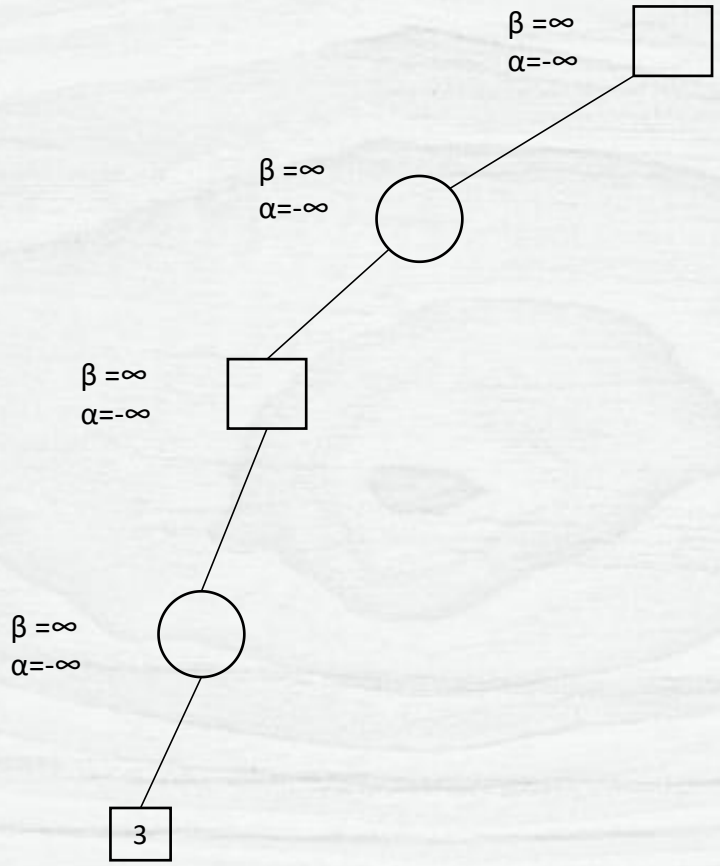
Alpha-beta剪枝

- Minimax搜索：必须检查的游戏状态的数目随着博弈的进行呈指数增长；
- Alpha-beta剪枝：剪掉不可能影响决策的分支，尽可能地消除部分搜索树。

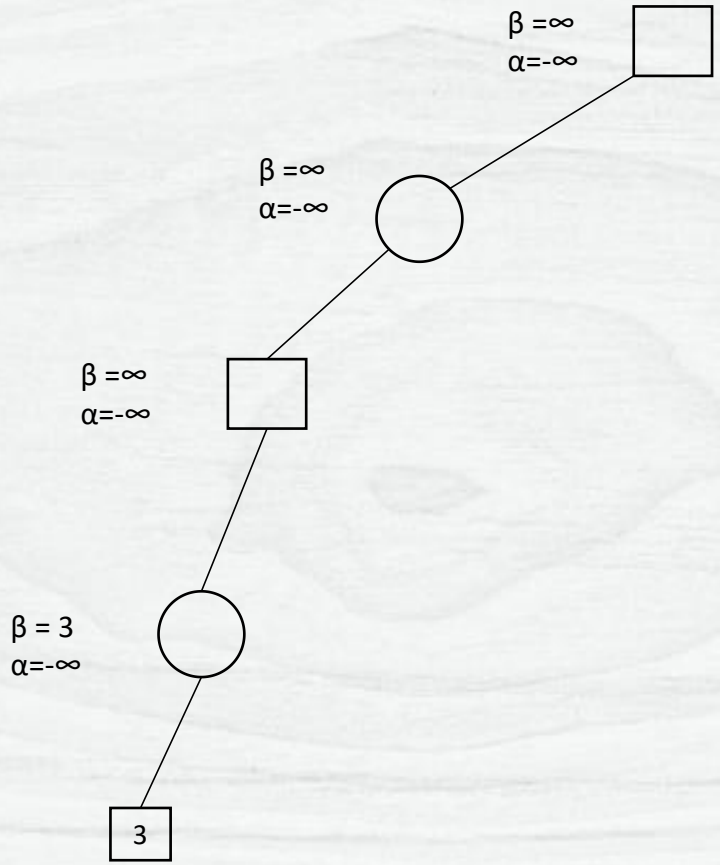
Alpha-beta剪枝



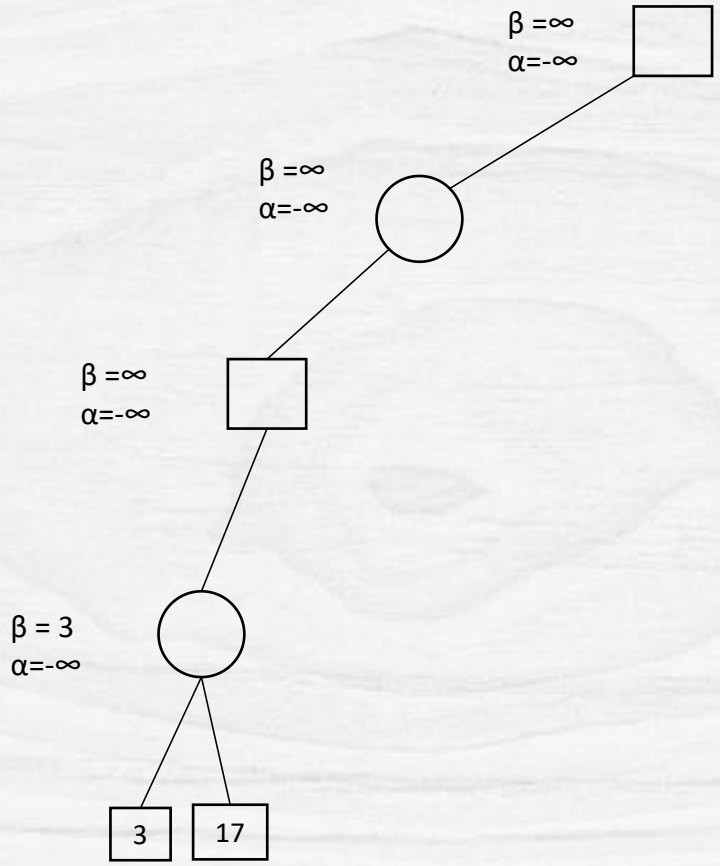
Alpha-beta剪枝



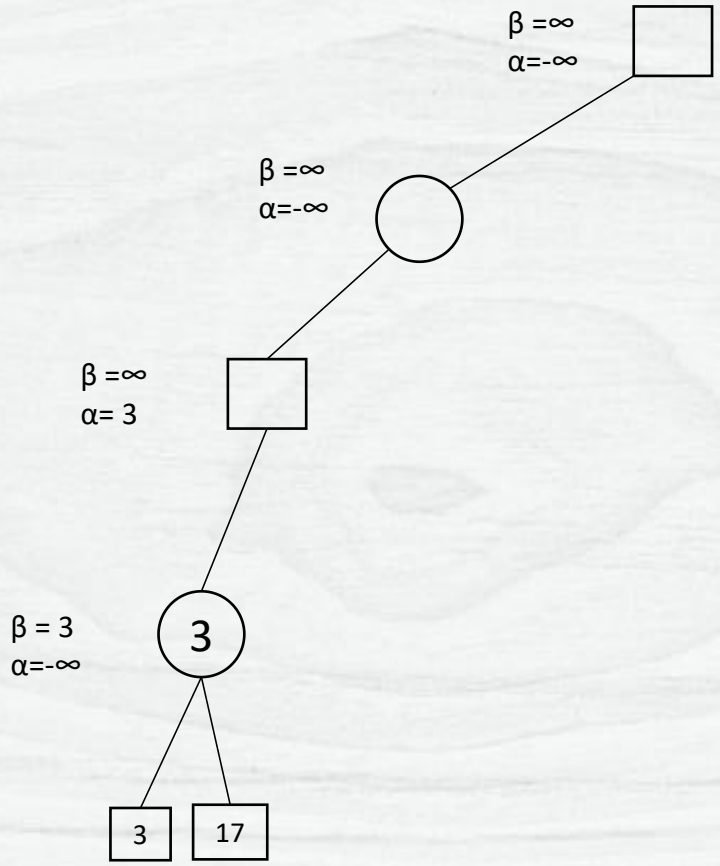
Alpha-beta剪枝



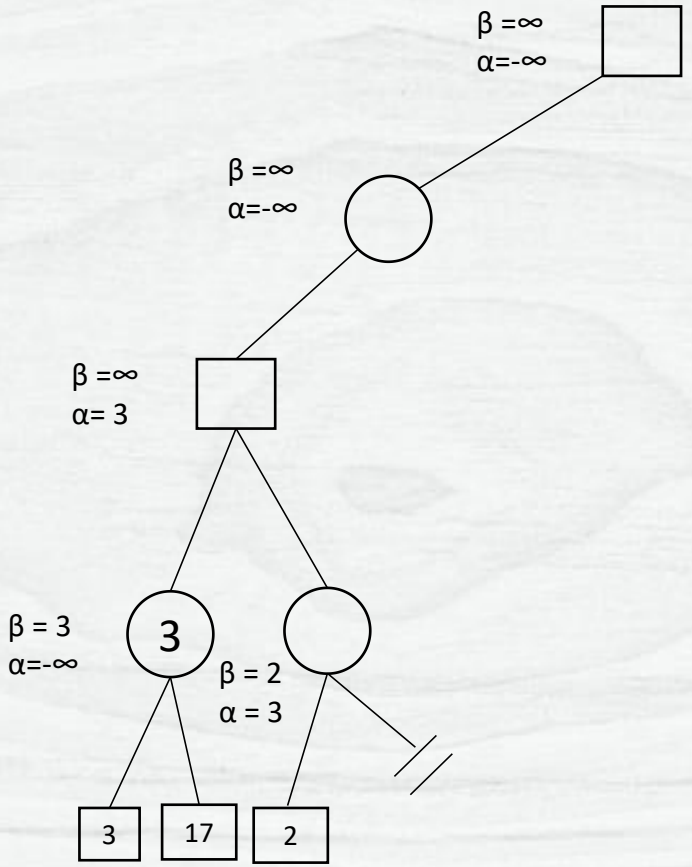
Alpha-beta剪枝



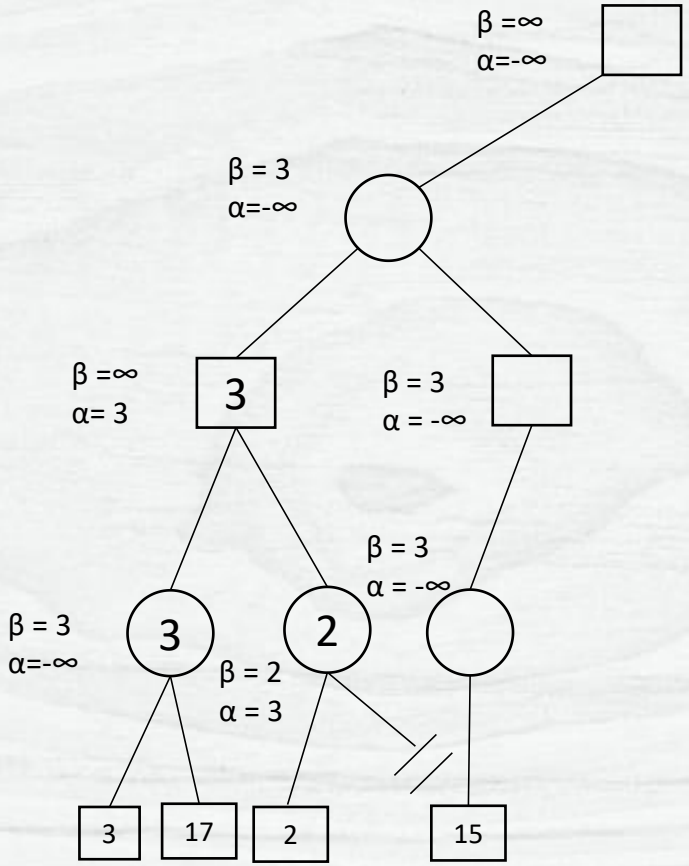
Alpha-beta剪枝



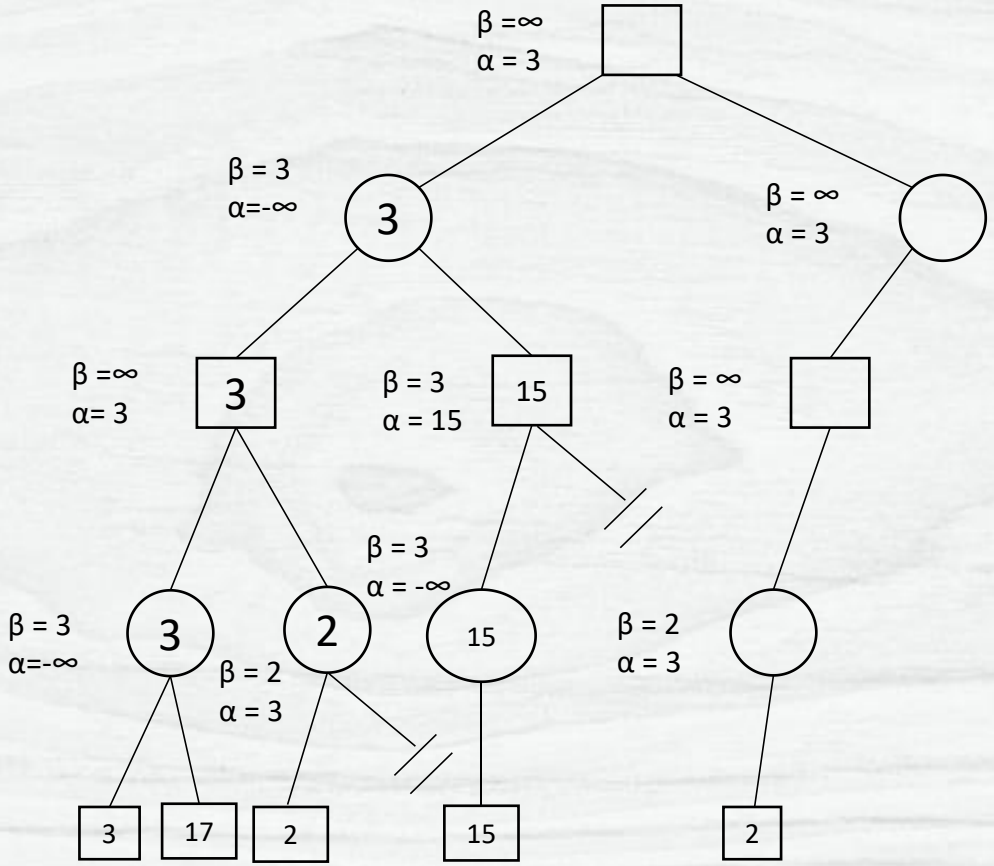
Alpha-beta剪枝



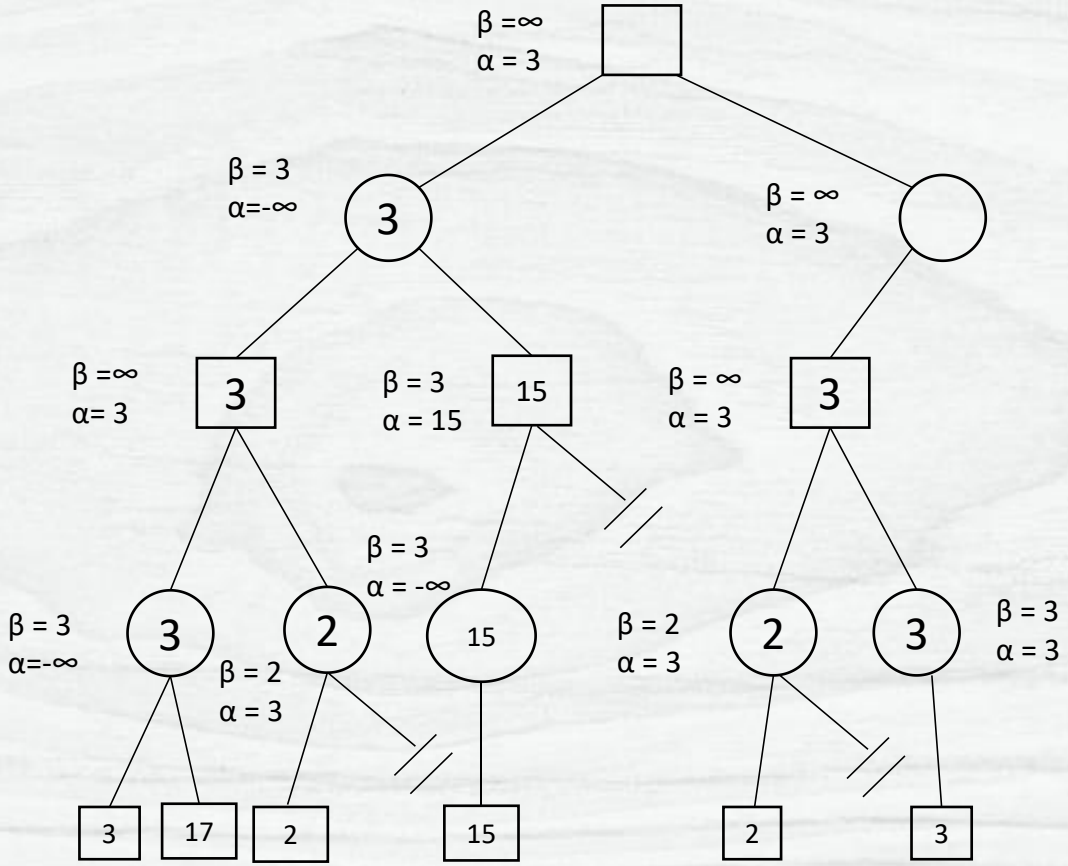
Alpha-beta剪枝



Alpha-beta剪枝



Alpha-beta剪枝



```
AlphaBeta(n,Player,alpha,beta) //return Utility of state
If n is TERMINAL
    return V(n) //Return terminal states utility
ChildList = n.Successors(Player)
If Player == MAX
 for c in ChildList
    alpha = max(alpha, AlphaBeta(c,MIN,alpha,beta))
    If beta <= alpha
        break  return alpha
Else //Player == MIN
  for c in ChildList
    beta = min(beta, AlphaBeta(c,MAX,alpha,beta))
    If beta <= alpha
        break  return beta
```

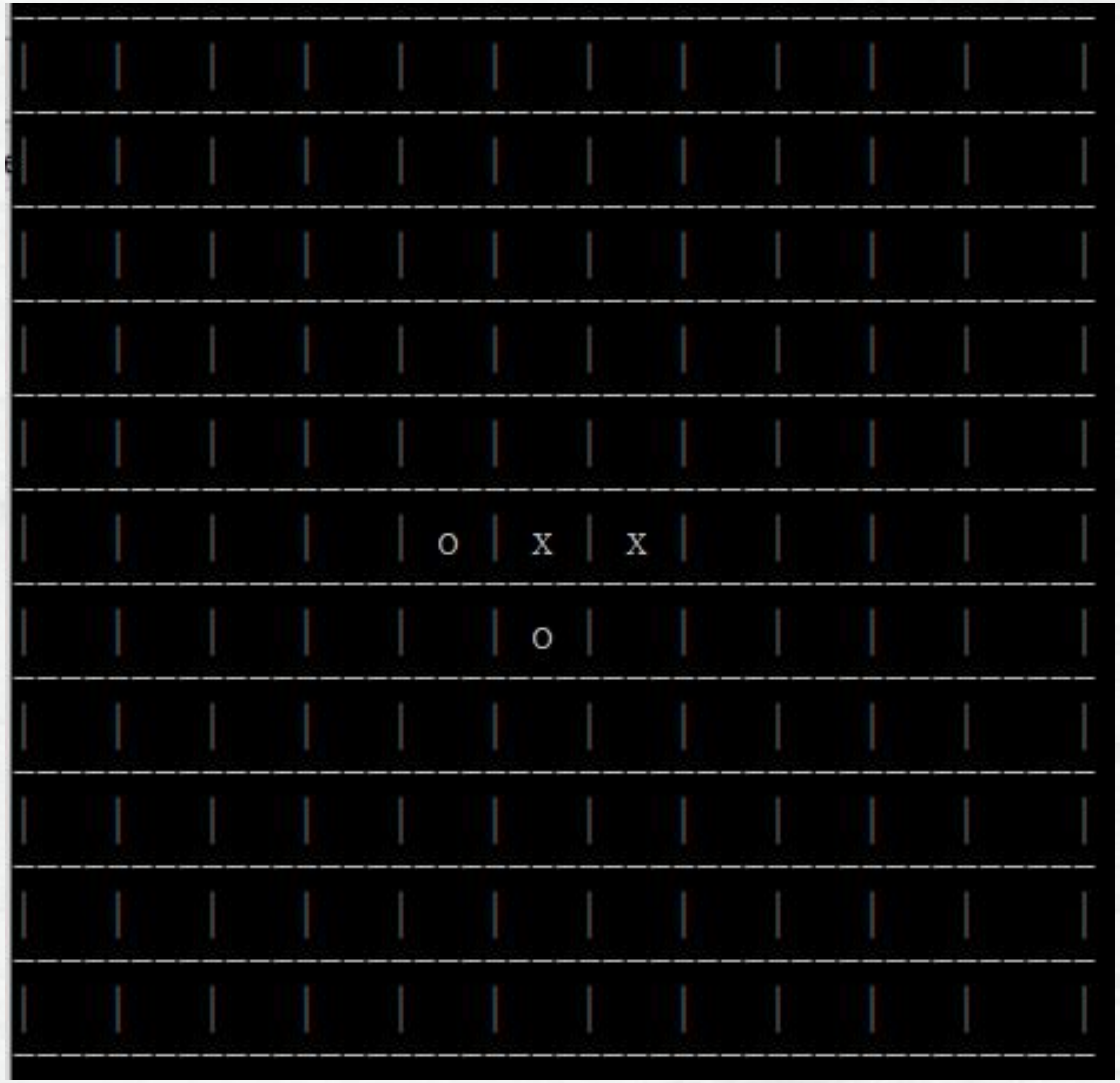
实验任务与报告提交

- 实现N*N的五子棋的人机对战，要求：
  - N=11，横排、竖排、对角线均可连线；
  - 要求使用alpha-beta剪枝；
  - 搜索深度和评价函数不限，自己设计。在报告中说明清楚自己的评价函数及搜索策略；
  - 实验结果要求展示至少连续三个回合（人和机器各落子一次指一回合）的棋局分布情况，并输出每步落子的得分。
- 提交文件

① 实验报告：18******_wangxiaoming.pdf

② 代码：18******_wangxiaoming.zip 如果代码分成多个文件，需要写readme

- DDL：2020-12-10 23:59:59



初始状态已下好四个点，X先手，o后手。

要求实现人和电脑下棋，即实现人类下棋的输入接口函数。

主函数中要求判断人类先手或电脑先手的模式。若模式为电脑先手，则下一步电脑先下，模式为人类先手，则下一步人类先下。

cons_initial.cpp

```cpp
int main()
{
    int len = 14;
    char a[len][len] ;
    for(int i=0;i<len;i++)
    {
        for(int j=0;j<len;j++)
        {
            a[i][j] =' ';
        }
    }
    a[5][5] = 'x';
    a[5][6] = 'x';
    a[6][5] = 'o';
    a[5][4] = 'o';

    print_board((char**)a,len);
}
```

```cpp
void print_board(char** a,int N)
{
    for(int j=0;j<N;j++)
    {
        cout<<"----";
    }
    cout<<"--";
    cout<<endl;
    for(int i=0;i<N;i++)
    {
        for(int j=0;j<N;j++)
        {
            cout<<"| "<<*((char*)a + N*i + j)<<" ";
        }
        cout<<" |";
        cout<<endl;
        for(int j=0;j<N;j++)
        {
            cout<<"----";
        }
        cout<<"--";
        cout<<endl;
    }
}
```