

实验4

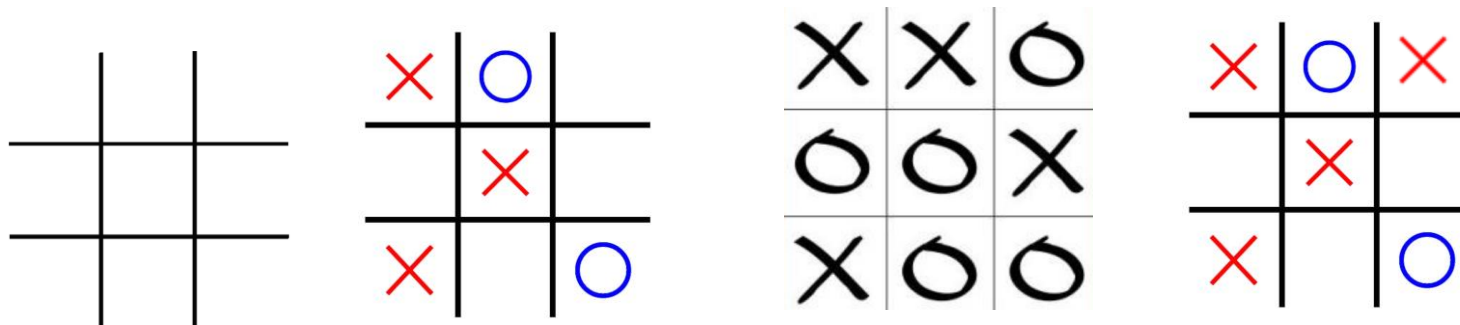
实验名称： 博弈树搜索算法实现

实验内容： 使用博弈树搜索算法实现tic-tac-toe游戏。

基本要求： 使用MINIMAX算法。 （如果使用ALPHA-BETA剪枝有附加分，请在实验报告中说明。）

Game: tic-tac-toe (井字棋)

- 两个玩家，一个叫MAX,一个叫MIN，MAX每次放一个X，MIN每次放一个O，交替走步，直到结束状态。
- **井字棋赢的定义：** 棋盘有3个X或者O在同一行或者同一列或者对角线上。
- 结束状态：MAX赢或者MIN赢，或者棋盘被摆满（平局）。



MINIMAX算法

MINIMAX算法可参考参考《人工智能：一种现代方法（第3版）》
：5.2.1节，图5.3

↑ ⓘ 不安全 | 202.195.149.12:9088/front/book/search/page

江南大学图书馆馆藏数字资源服务平台

图书

图书分类

- 田 经典理论
- 田 哲学、宗教
- 田 社会科学总论
- 田 政治、法律
- 田 军事
- 田 经济
- 田 文化、科学、教育、体育
- 田 语言、文字
- 田 文学
- 田 艺术
- 田 历史、地理

人工智能一种现代

搜索 二次搜索 高级搜索

☒ 书名 ☐ 作者 ☐ 主题词

共搜到4本图书 排序： 书名 出版日期 当前为：第1页，共1页



国外著名高等院校信息科学与技术优秀教材 **人工智能：一种现代方法（第2版）**
作者：（美）STUART RUSSELL PETET NORVING著
出版日期：2004
主题词：人工智能-高等学校-教材
索书号：TP18/117

馆藏借阅



人工智能一种现代的方法（第3版）
作者：STUART J.RUSSELL
出版日期：2013
主题词：人工智能-教材
索书号：TP18

馆藏借阅

MINIMAX算法

function MINIMAX-DECISION(*state*) *returns an action*
 return $\arg \max_{a \in \text{ACTIONS}(s)} \text{MIN-VALUE}(\text{RESULT}(\text{state}, a))$

function MAX-VALUE(*state*) *returns a utility value*
 if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)
 $v \leftarrow -\infty$
 for each *a* **in** ACTIONS(*state*) **do**
 $v \leftarrow \text{MAX}(v, \text{MIN-VALUE}(\text{RESULT}(s, a)))$
 return *v*

function MIN-VALUE(*state*) *returns a utility value*
 if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)
 $v \leftarrow \infty$
 for each *a* **in** ACTIONS(*state*) **do**
 $v \leftarrow \text{MIN}(v, \text{MAX-VALUE}(\text{RESULT}(s, a)))$
 return *v*

Figure 5.3 An algorithm for calculating minimax decisions. It returns the action corresponding to the best possible move, that is, the move that leads to the outcome with the best utility, under the assumption that the opponent plays to minimize utility. The functions MAX-VALUE and MIN-VALUE go through the whole game tree, all the way to the leaves, to determine the backed-up value of a state. The notation $\arg \max_{a \in S} f(a)$ computes the element *a* of set *S* that has the maximum value of *f(a)*.

实验注意事项

1、不要花时间去设计复杂的交互界面，精力放在博弈树搜索算法上。可以采用下面的方式设计界面，也可以自行定义其他交互界面。

先询问是人先下还是机器先下：

Human first?[y/n]

如果是人先下，再询问人选择X还是0?

you choose X or 0?

实验注意事项

在人选择好先下后下、X还是O以后，界面上显示空白棋盘。例如可以用下面的界面代表空白棋盘（你也可以自己采用其他的棋盘表示形式）。

. . .

. . .

. . .

提示“please input row and col”（例如输入1 3代表第1行第3列），那么棋局就变成：

. . X

. . .

. . .

2、当人或者机器选择将棋放在某个位置时，需要考虑此位置是否已经摆放了棋子。

3、需要考虑搜索图的深度（往下推算的步数）及叶子结点的估价函数定义。

最简单的做法，最大深度设为9，必然已经到达最终状态了，可以直接得出叶子状态是人赢了还是机器赢了。注意，机器永远希望它自己的得分最大，所以叶子节点的估价函数可以设置成：

$f(\text{board})=1$, if 机器赢；

$f(\text{board})=-1$, if 人赢；

$f(\text{board})=0$, if 平局。

也可以设置小一点的搜索图的最大深度值（例如深度阈值为3或者4），将叶子结点的估价函数定义为：

$f(\text{board})=\text{机器的赢线数}-\text{人的赢线数}$ ，if 叶子节点不是最终棋局；

$f(\text{board})=\text{正无穷}$ ，if 叶子节点是机器赢；

$f(\text{board})=\text{负无穷}$ ，if 叶子节点是人赢；

$f(\text{board})=0$ ，if 叶子节点是平局。

你也可以自己定义叶子节点的估价函数。

实验报告内容:

- ①实验目的
- ②实验内容
- ③估价函数定义、搜索树深度等
- ④算法流程图（或伪代码）
- ⑤实验运行过程截图、实验结果
- ⑥实验过程中遇到的问题
- ⑦实验心得体会。

另外需要提交完整的源代码（需有注释说明，单独作为文件，不要放到实验报告word文档里面）。

注意事项:

- 1、可以相互讨论，但必须独立完成代码和实验报告，若与网上或者其他同学雷同，按不及格处理。
- 2、实验报告请使用老师提供的实验模板。实验报告命名：完整学号_姓名_AI_project4.doc；
例： 0304120101_张三_AI_project4.doc；
每个同学建立个人文件夹放实验报告和源代码，文件夹名“完整学号_姓名_AI_project4”。
- 3、报告提交时间：第12周周四。