

# 情報工学実験1

## (7-9班, 4/15)

アルゴリズムに関する実験と評価

木谷 裕紀

# 本講義(木谷担当)について

(3限の開始と4限の開始に)出欠の確認->40分程度の説明  
(次回は30分程度)

授業中に発言を求めることはありません.

提出期限はしっかり守ってください.

他者のプログラムや類似したものを提出する行為や指定された部分以外のAIツールを使用したプログラムの作成は減点します.  
一方で学生同士の相談などは自由に行ってください.

# 出欠の確認, ネットワークトラブルについて

対面参加者は直接出欠を取ります.

zoom参加者は参加者リストのスクリーンショットを取ります.

遅れて参加した場合は必ず名乗り出てください. 出欠が取られていない可能性があります.

私も含めてネットワークトラブルにより, 途中入退出する可能性が考えられます. 基本的には私が途中でオフラインになった場合は10分後に授業を再開します.

# 本実験の実験報告書の書き方(ガイダンス参照)

- (a) 目的(簡潔で良い).
- (b) 解説 (必須ではない, 本実験ではこのパートの代替として「結果予測(仮説をたてること)」を行う).
- (c) 使用機器 (基本的には記述の必要はないが, 実験結果に計算速度などの要素が入る場合は, PCのスペックなどの記述を入れる場合がある).
- (d) 実験方法 (実験概要に含まれるため簡潔で良い).
- (e) 実験結果. プログラムを参照するなどして実験結果をかく.
- (f) 検討考察. 実験結果に対して検討, 考察や解析したことから記述する.

卒業研究ではしばしば仮説をたてた上でその検証を行うことがある。

事前に適切な(あるいは尤もらしい)仮説をたてて検証をすることによって、余分に多くの実験をすることを避けられたり、結果の説明力を高められたりすることがある

- 経験則
- データ解析
- etc...



# 本実験概要

本実験ではゲームを最適に振る舞った場合の勝敗に関する予想を行い、その予想を実際に探索アルゴリズムを走らせることによって検証することを行います。

# 課題1(7-9班, 共円ゲーム)

1. 「共円ゲーム」を $3 \times 3$ の正方格子盤面で行うとき、『必ず先手が勝てる』か『必ず後手が勝てる』かのどちらに当てはまるか予想をしてください。
2.  $3 \times 3$ の正方格子盤面上に頂点をプロットする時、ちょうど4点を通るような円となる4点組は何通りあるか、教えてください
3. 「共円ゲーム」を $3 \times 3$ の正方格子盤面で行うとき、『必ず先手が勝てる』か『必ず後手が勝てる』かのどちらに当てはまるかアルゴリズム実装などを通して、検証してください。

「3 to 15」は『必ず引き分けに持ち込める』ことが知られています。  
0のカードを追加した「3 to 15」に対して、『必ず先手が勝てる』か  
『必ず後手が勝てる』か『必ず引き分けに持ち込める』のうち、ど  
れに当てはまるか予想してください。

0のカードを追加した「3 to 15」に対して、『必ず先手が勝てる』か  
『必ず後手が勝てる』か『必ず引き分けに持ち込める』のうち、ど  
れに当てはまるかアルゴリズム実装などを通して、検証してください。



# 提出に関して

最終提出物はTexやWordなどを用いて作成し， pdfファイルをMoodle上に提出する．

その際， 実装したプログラムと一緒に別ファイルとして提出を行うか， pdf文書ファイル内に入れること

生成AIの使用や参考は探索アルゴリズムの知見の取得の目的を鑑み， 全く推奨されないが， 完全に禁止はしない． 使用及び参考にした場合， 実験方法などの項に使用した生成AI， プロンプト(ファイルを入力していればそのファイルの中身)， その出力を全て記入すること． エラーチェッカー機能は使用アプリケーションのみ記載すれば良いこととする．

Moodle上の該当提出先に提出.

ファイル名は(学籍番号)\_report.pdfとする.

ファイルサイズが大きすぎる場合、moodleで受け付けられない可能性があるため、気をつけること. また、その場合は事前に相談すること.

# 二人零和有限確定完全情報ゲーム

ゲームのジャンルの一つで

1. 二人で行い(二人),
  2. 非協力ゲームで(零和),
  3. ゲームの長さが有限で(有限),
  4. サイコロなどの偶然要素がなく(確定),
  5. 隠されている情報がない(完全情報)
- ゲーム.

ゲームの終了状態は(あるプレイヤーにとって, )勝ち, 負け, 引き分けのいずれか

---

# 必ず勝つ局面

相手の任意の着手に対して、終了状態が勝ちの局面となるような着手が存在する時、その局面を必ず勝つ局面と呼ぶ。

二人零和有限確定完全情報ゲームは(+逐次手番という制約を加えると)お互いが最善を尽くしたとき、同じ盤面からは『必ず先手が勝てる』か『必ず後手が勝てる』か『必ず引き分けに持ち込める』のいずれか(ノイマン・ツェルメロの定理)。

# ○×ゲーム(三目並べ)

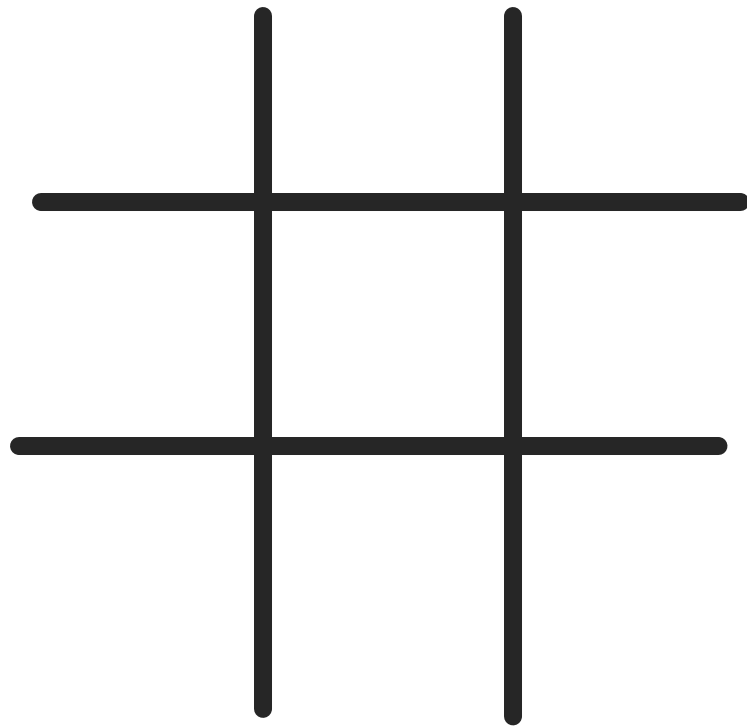
二人零和有限確定完全情報ゲームの一つ.

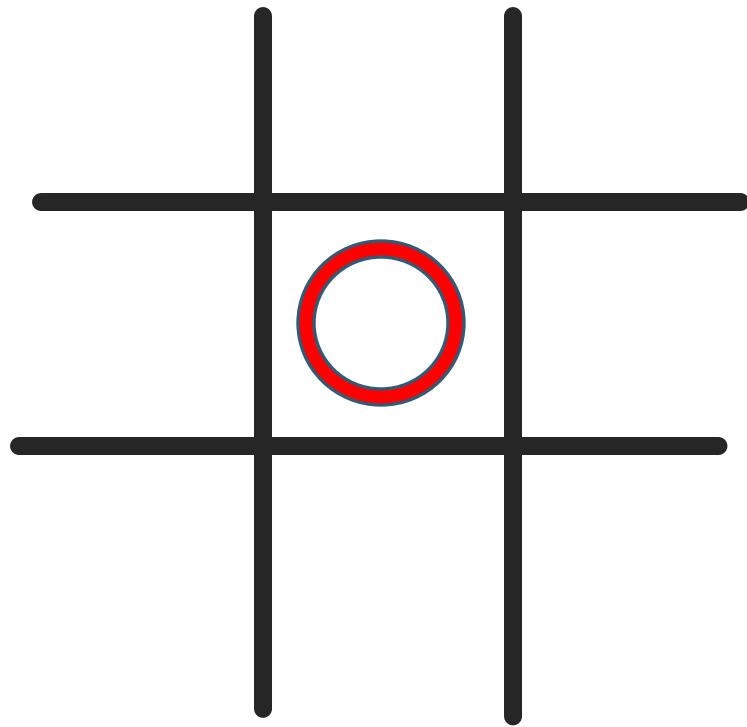
○プレイヤーと×プレイヤーに分かれてゲームが進む.

3 \* 3 のボードに○と×を空いてるマスに交互に着手を行う.

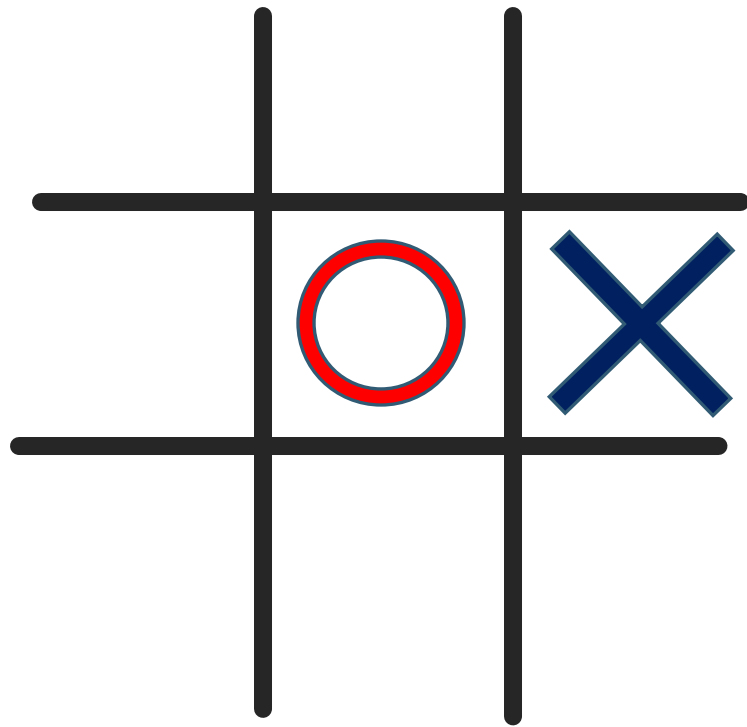
縦か横か斜めに自分のマークのみを一行に並べたプレイヤーが勝ち.

どちらのプレイヤーも着手できなくなったら引き分け.



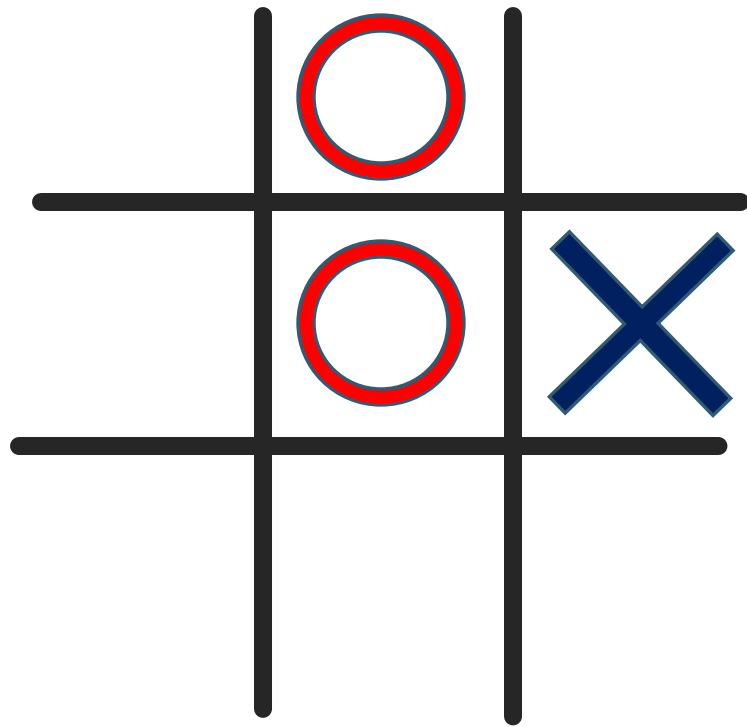


# ゲーム進行例

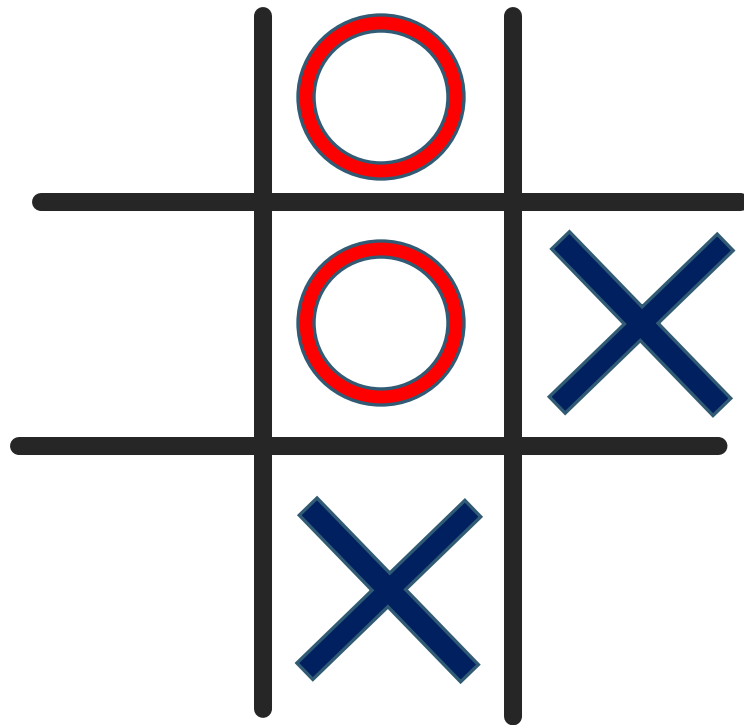




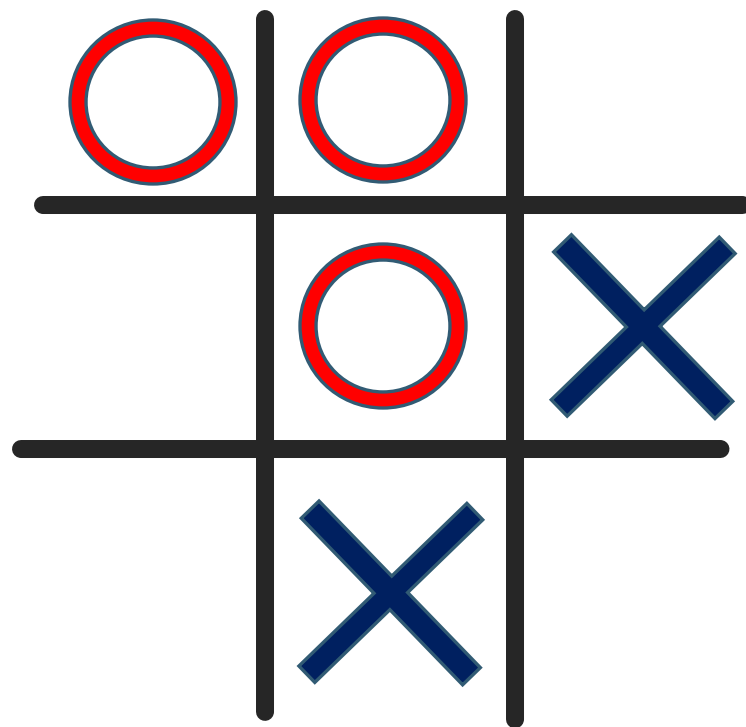
# ゲーム進行例



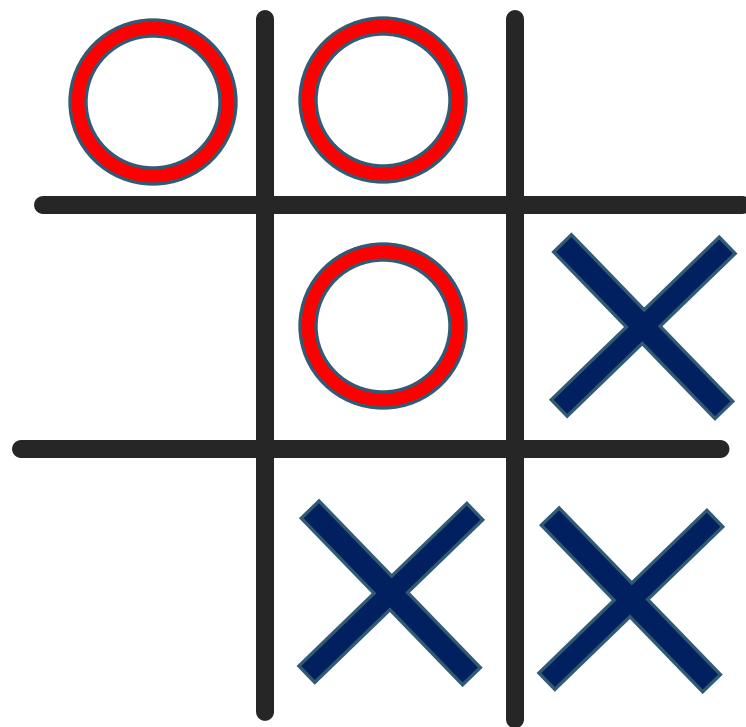
# ゲーム進行例



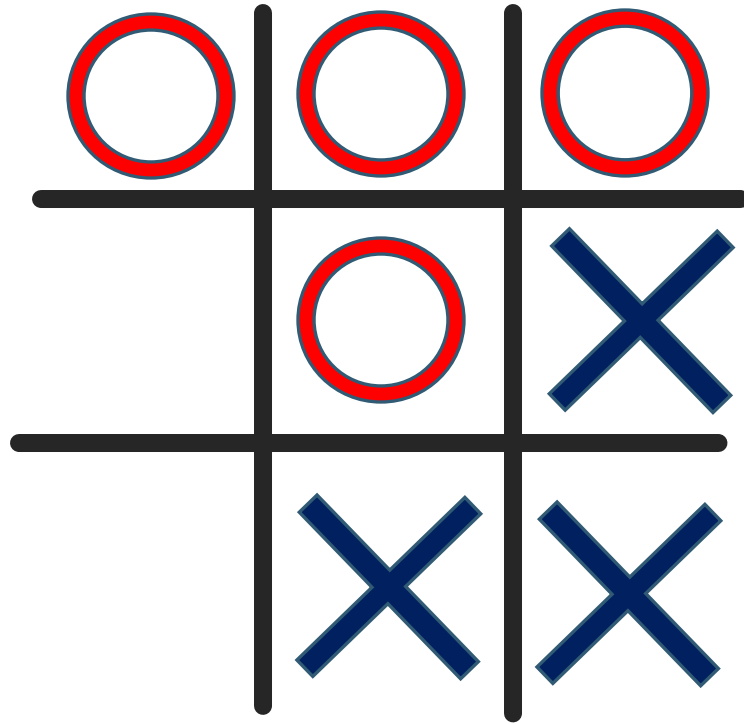
# ゲーム進行例



# ゲーム進行例



# ゲーム進行例



3つ揃ったので○プレイヤーの勝ち.

# ○×ゲーム(三目並べ)

二人零和有限確定完全情報ゲームの一つ.

○プレイヤーと×プレイヤーに分かれてゲームが進む.

3 \* 3 のボードに○と×を空いてるマスに交互に着手を行う.

縦か横か斜めに自分のマークのみを一行に並べたプレイヤーが勝ち.

どちらのプレイヤーも着手できなくなったら引き分け.

後手が最善を尽くすと引き分けに持ち込めることが知られている.

(先手がどんなにうまくても後手が上手なら引き分け. )

---

ただし，必要に応じて---や---を使っても良い.

# ゲームに対するアルゴリズム設計

二人零和有限確定完全情報ゲーム(組合せゲーム)に対して、作りたいアルゴリズムは？

- 最善(より良い)手を出力する(より早い)アルゴリズム
- 最善手の結果として、どちらが勝つか出力するアルゴリズム

最も基本的なこれらのアルゴリズムとして、MiniMax探索を用いる手法であるMiniMax法が知られている。



相手が最善手を打ち続ける過程の下，最も結果が良くなる着手を探索する方法.

具体的にはお互いのプレイヤーの手番において，1手先の盤面の盤面評価を行い，自分にとって最も「良い」盤面評価となるように着手を選択する.

今回扱うような零和ゲームに対しては先手が必ず勝つ局面 $=+1$ ，引き分け $=0$ ，先手が負け $= -1$ として盤面を評価することができる.

先手は盤面評価が高い盤面を目指し，後手は盤面評価が低い盤面を目指す(先手を最大化プレイヤー，後手を最小化プレイヤーと呼ぶ).

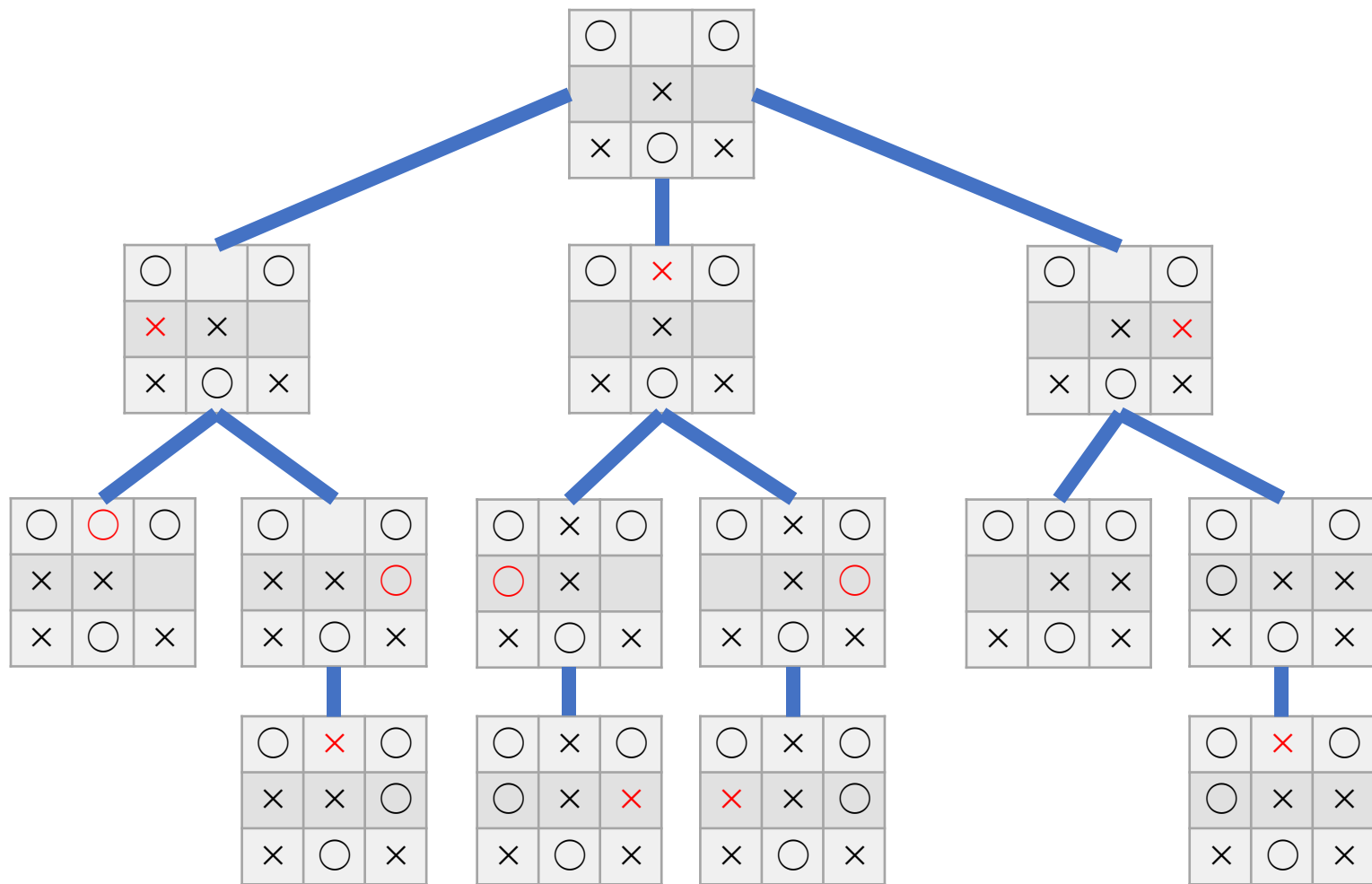
全ての盤面を正確に評価できる時に用いると，厳密解が得られる.

Minimax法の最も基本的な実装手順は以下となる.

1. ゲーム木を作成する.
2. ゲーム木の葉に評価をつける.
3. 根に向かってノードの評価を行う.

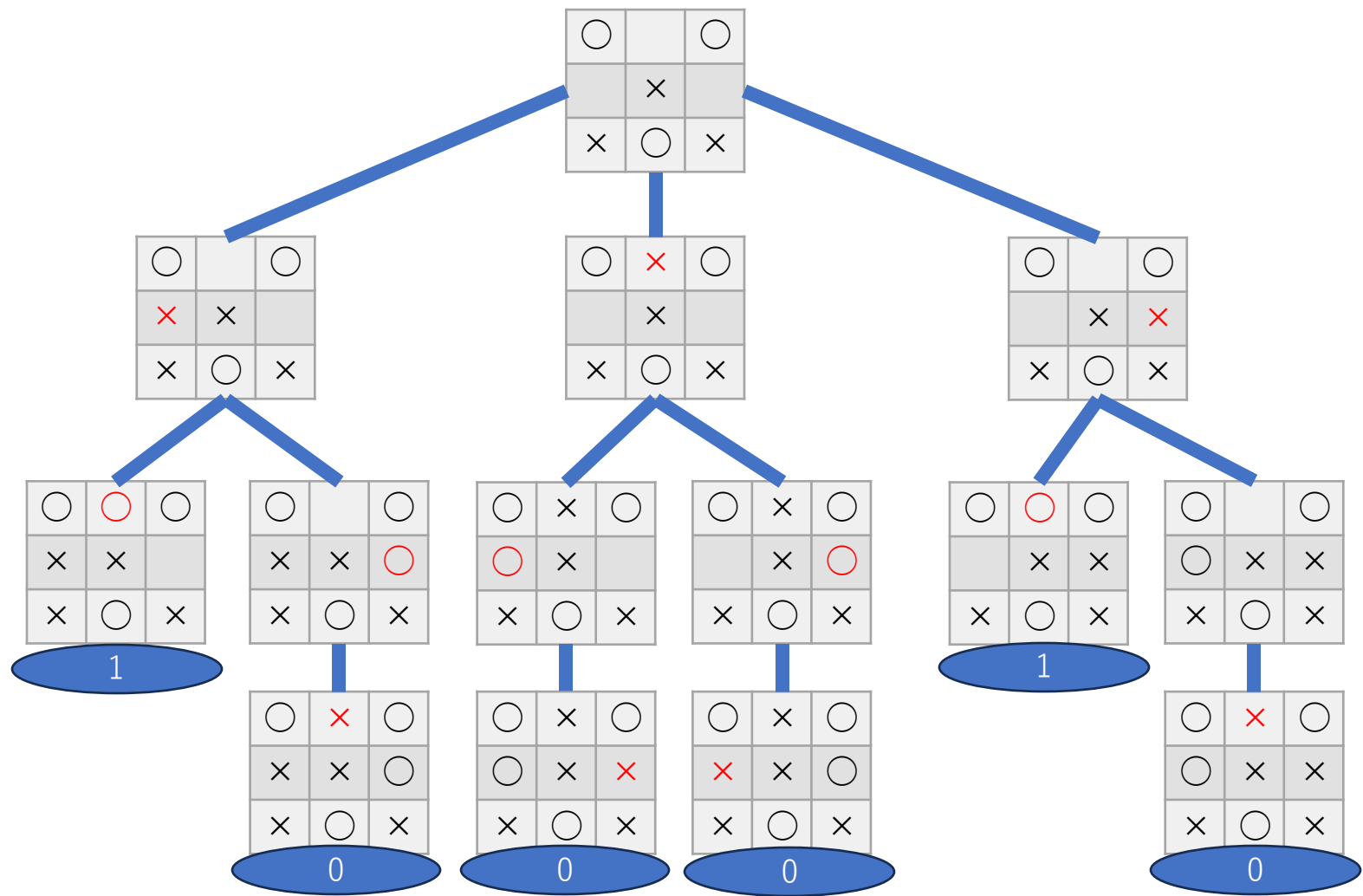
Minimax法の最も基本的な実装手順は以下となる.

1. ゲーム木を作成する.
2. ゲーム木の葉に評価をつける.
3. 根に向かってノードの評価を行う.



Minimax法の最も基本的な実装手順は以下となる.

1. ゲーム木を作成する.
2. ゲーム木の葉に評価をつける.
3. 根に向かってノードの評価を行う.



Minimax法の最も基本的な実装手順は以下となる.

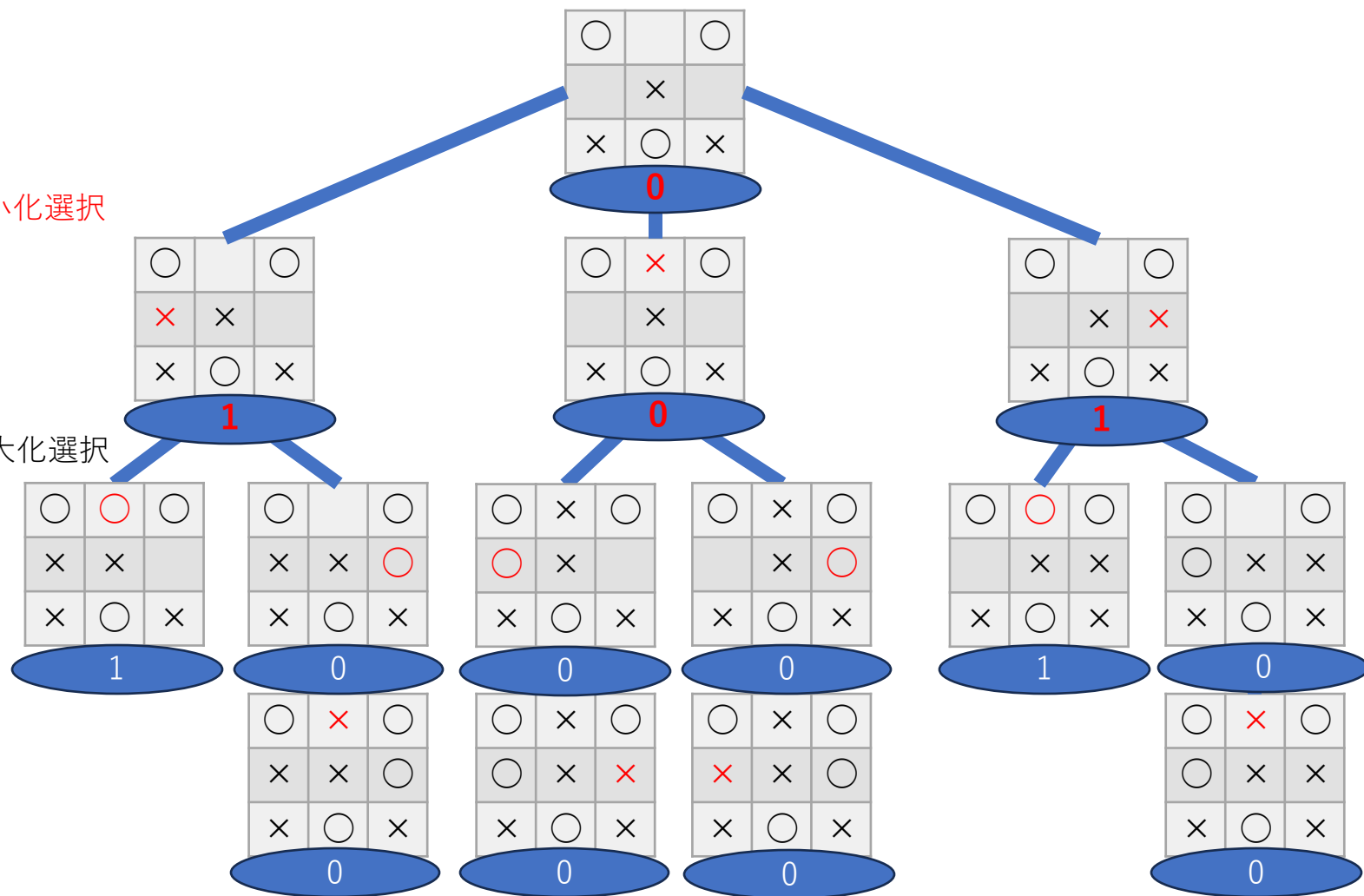
1. ゲーム木を作成する.
2. ゲーム木の葉に評価をつける.
3. 根に向かってノードの評価を行う.





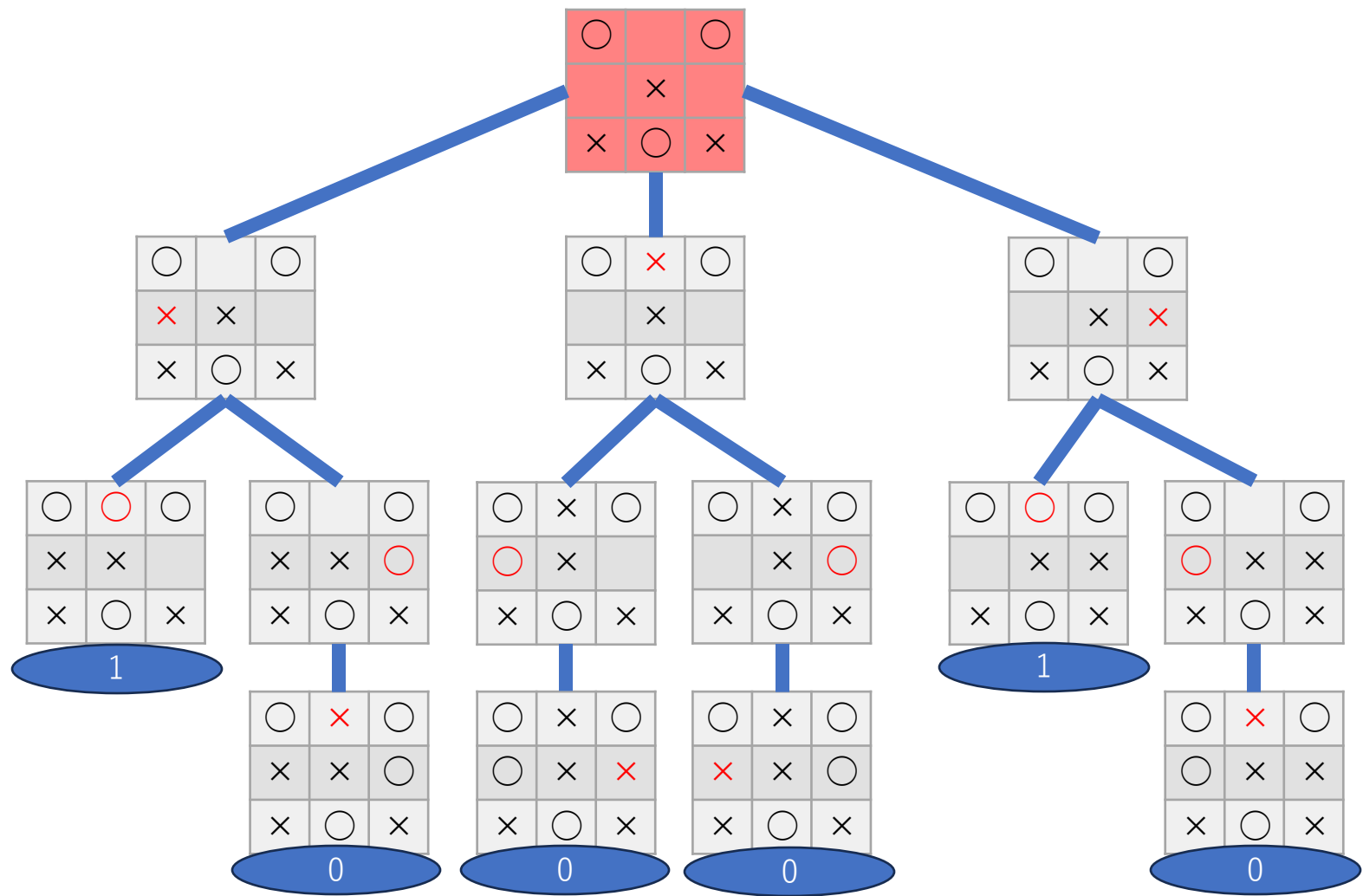
最小化選択

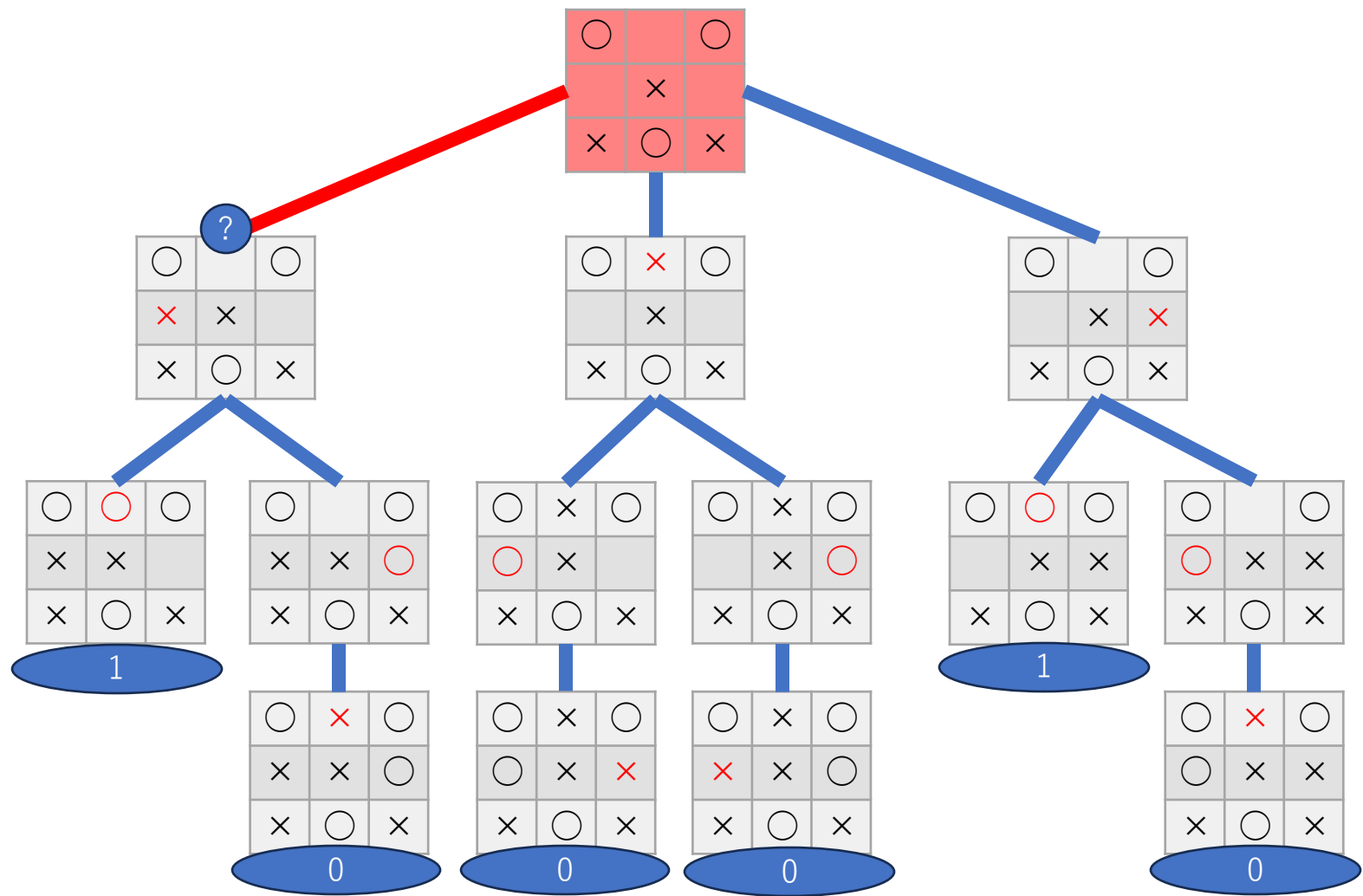
最大化選択

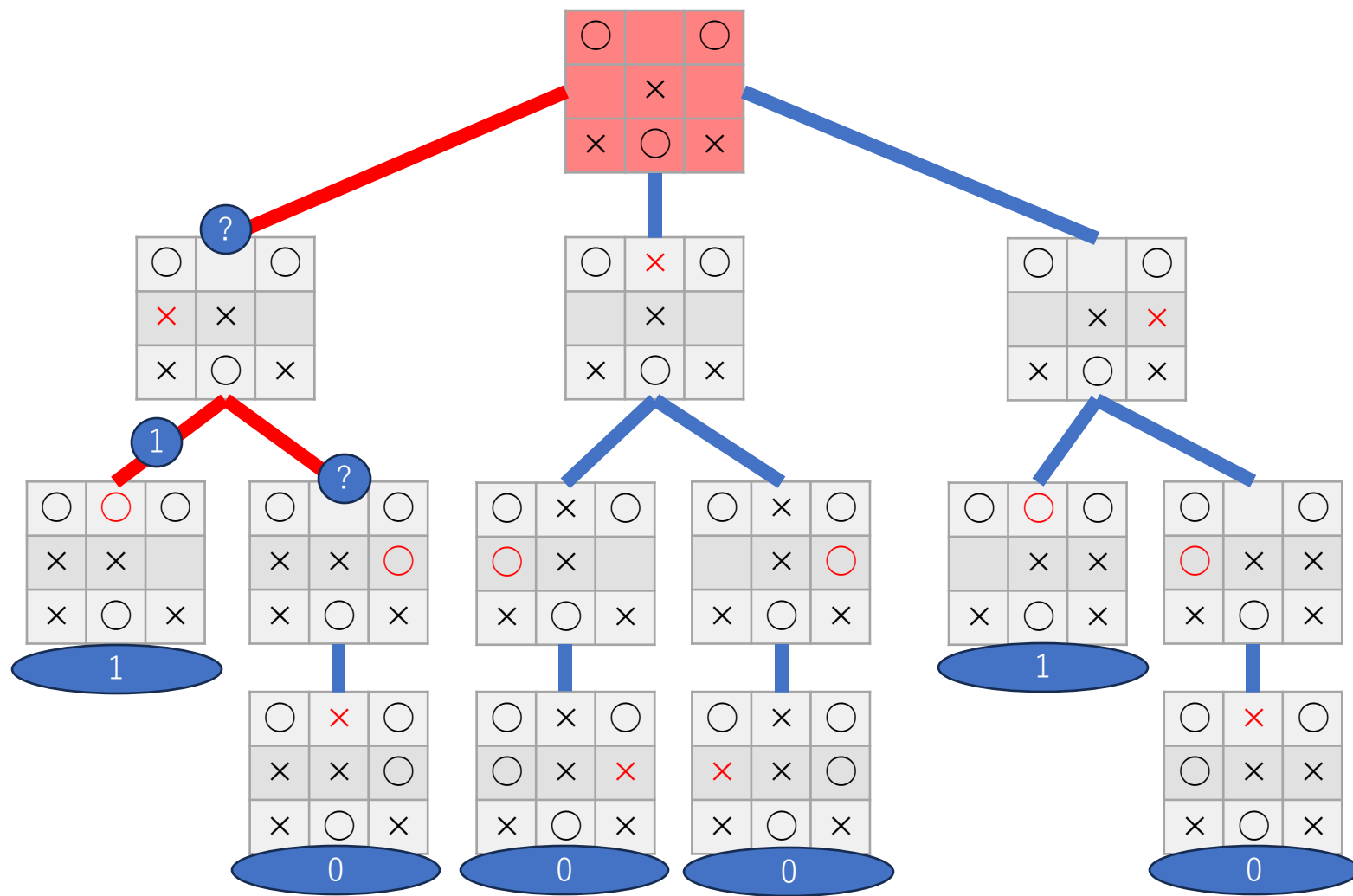


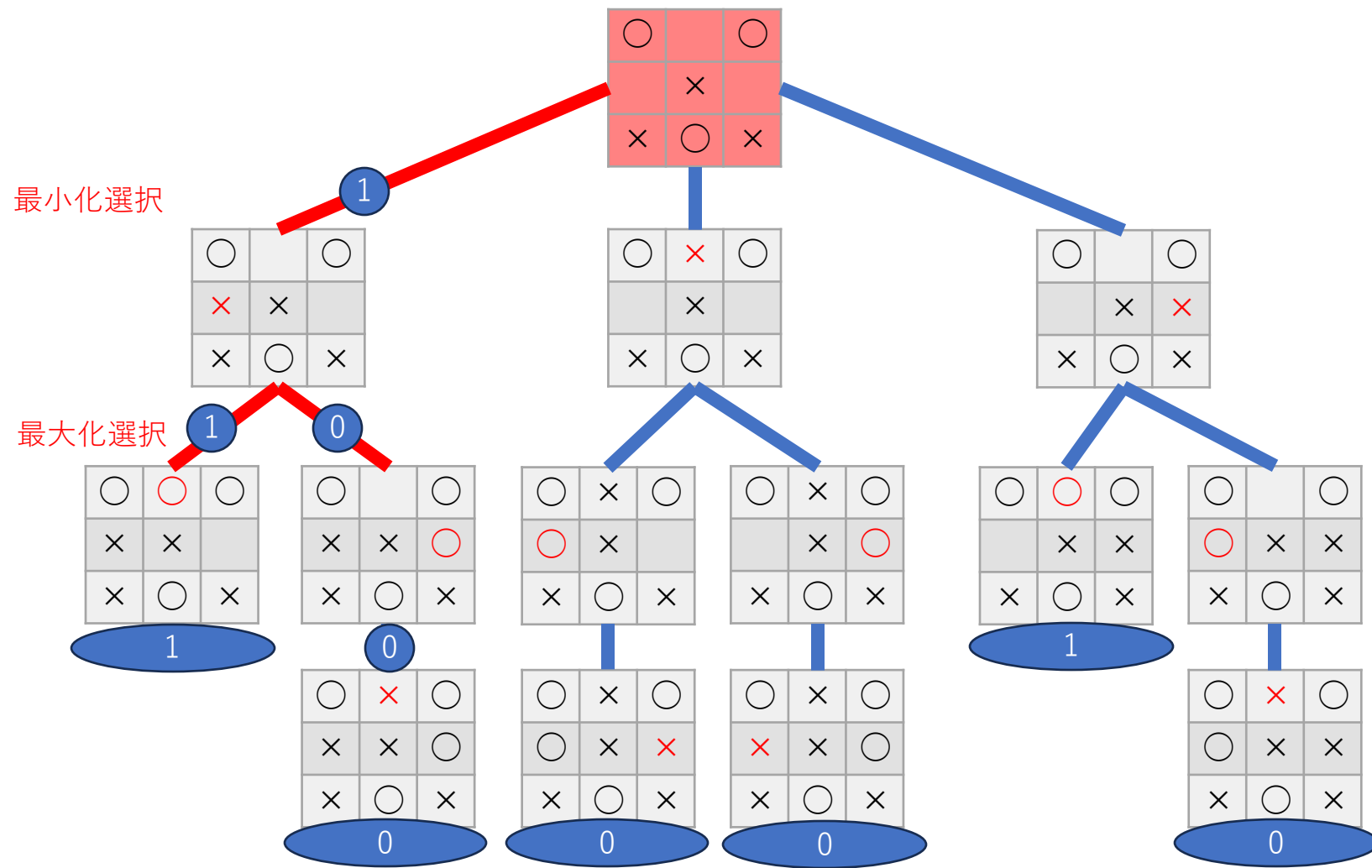
Moodleのサンプルファイル参照(python, C++).

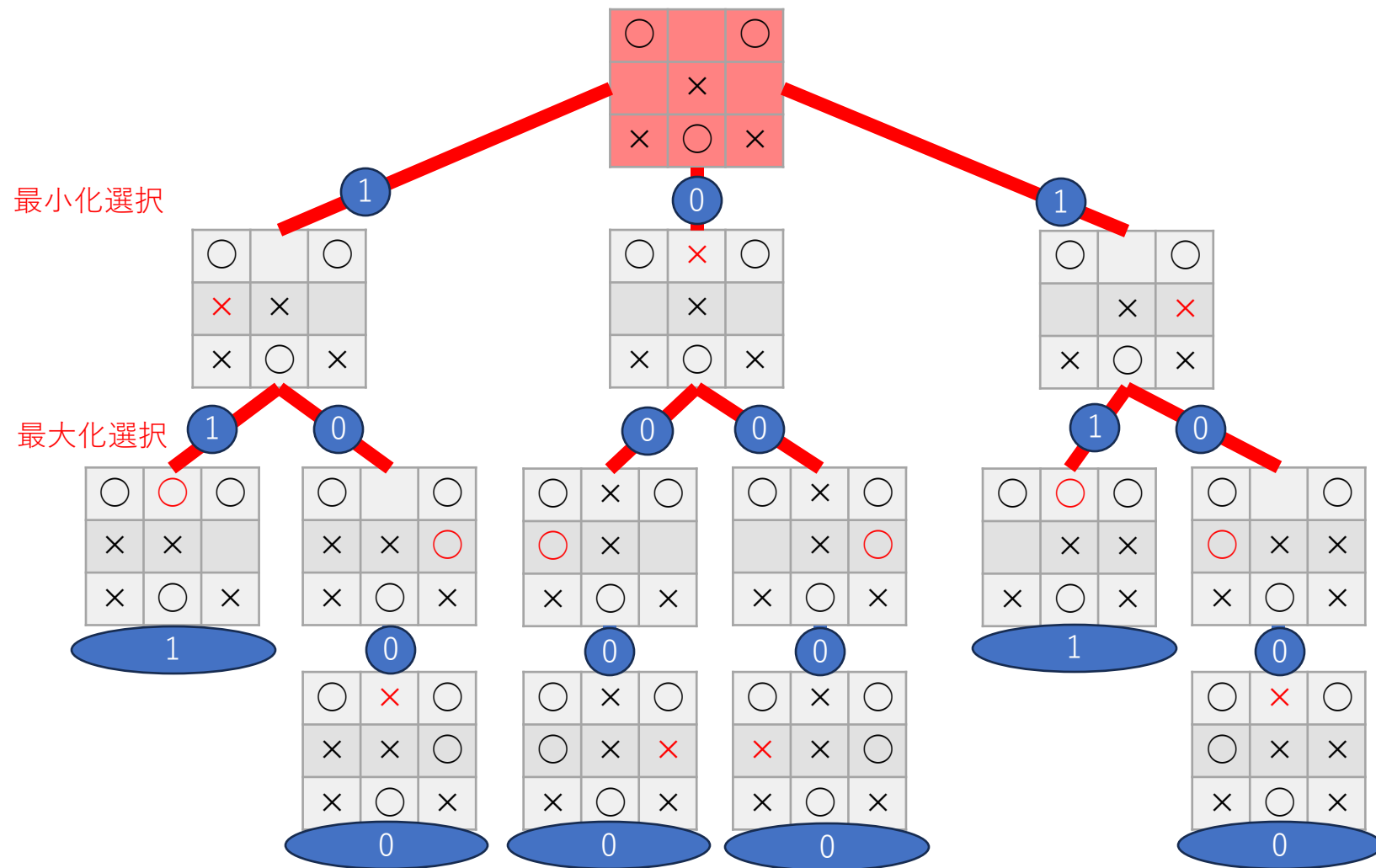
現在のプレイヤーが○プレイヤーの場合, 1手先の局面全てに対し, 盤面を評価し, その盤面の評価が定まっていればその値を, 定まっていなければ, 同様にその一手先の盤面を探索する.

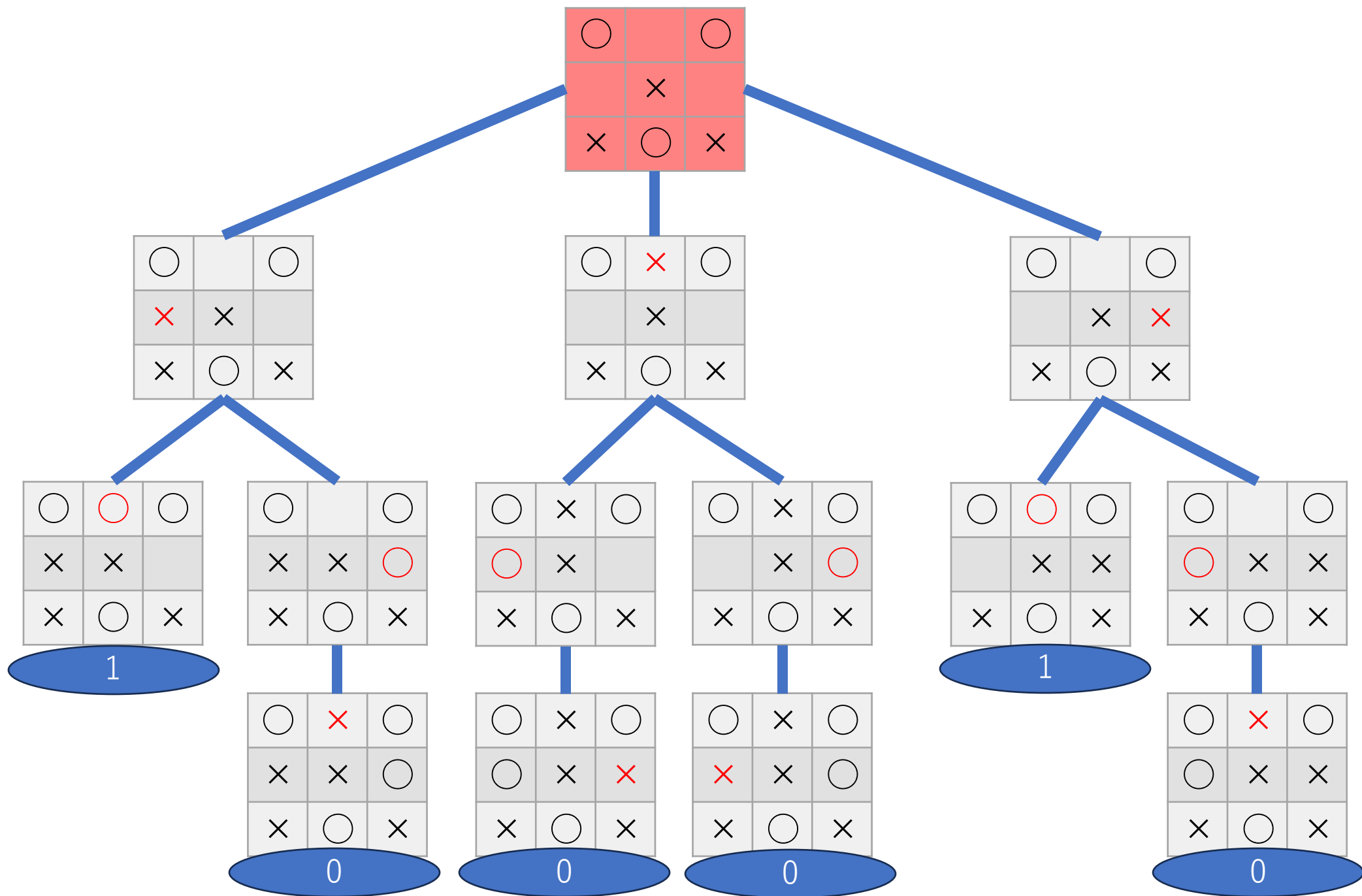




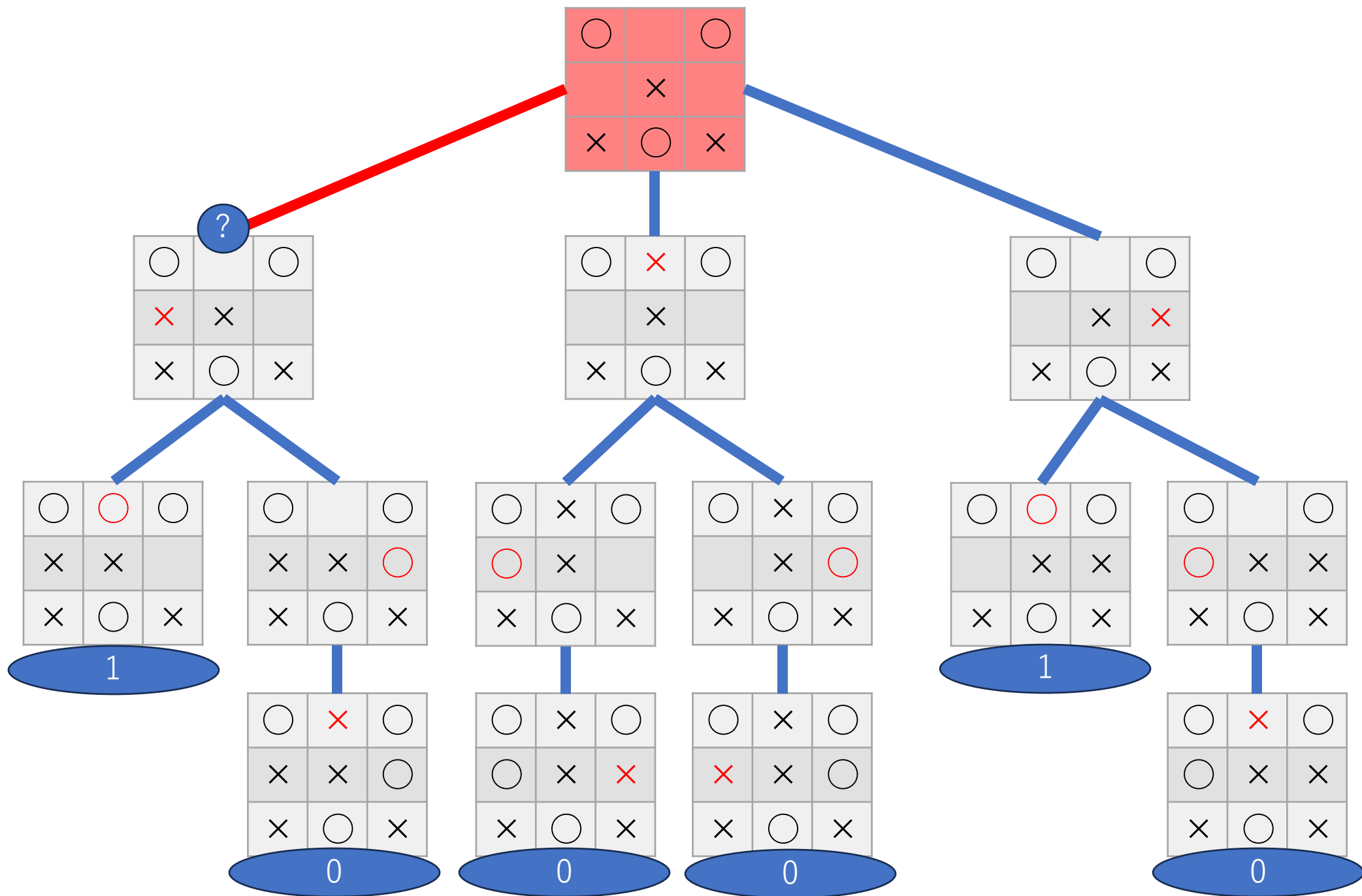


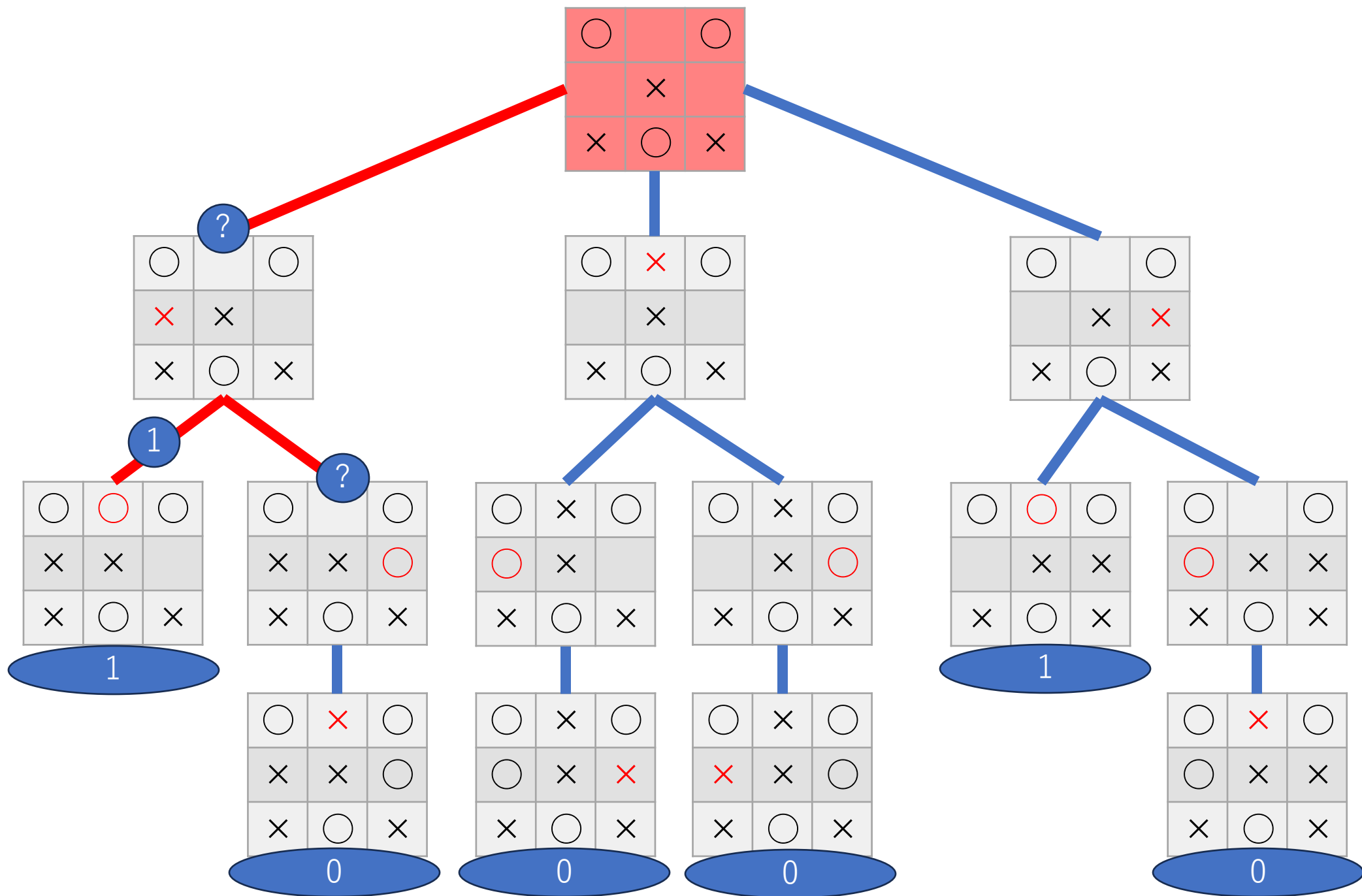








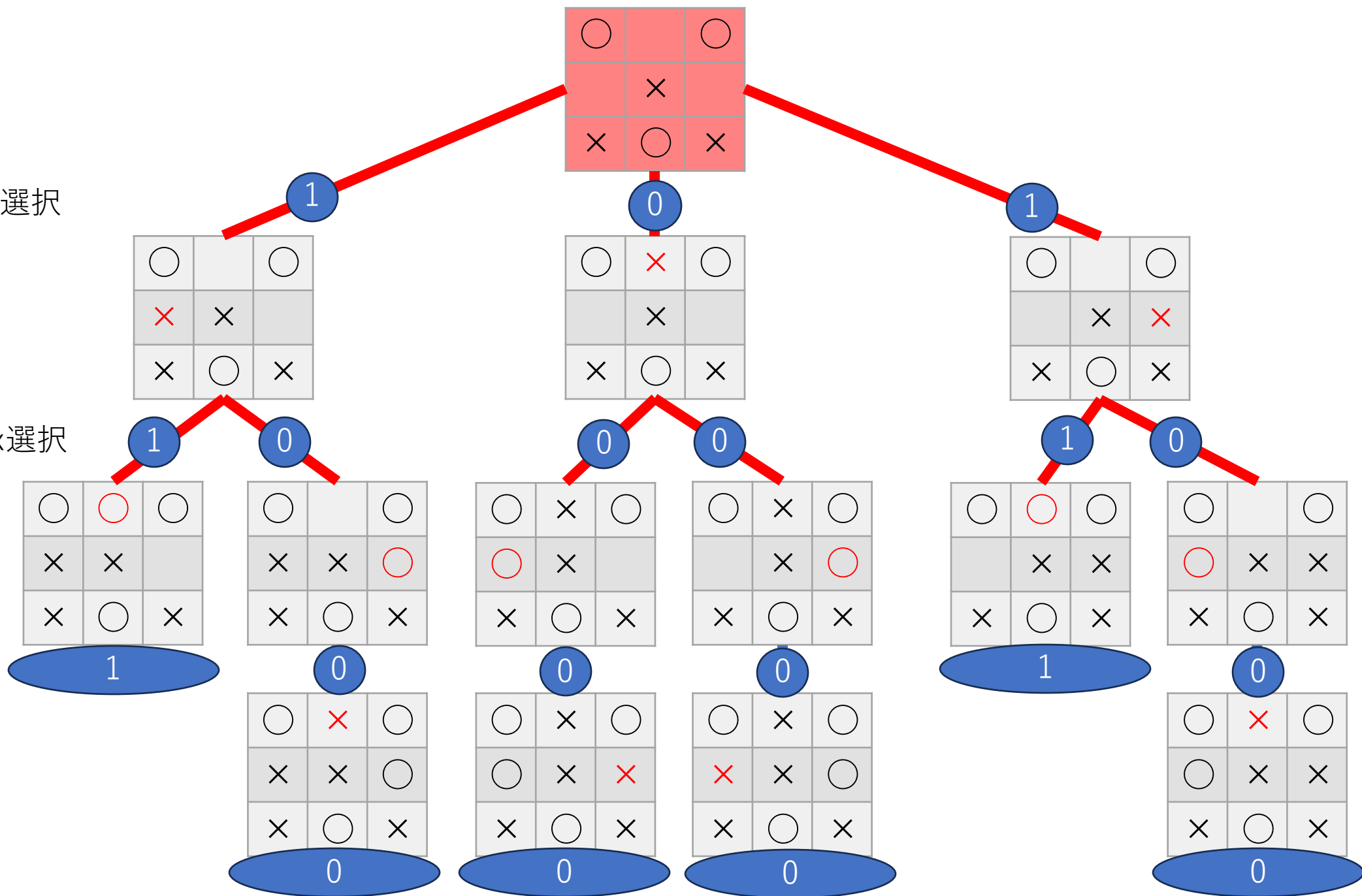






Min選択

Max選択



# サンプルプログラム(最小化プレイヤー手番)

1. 初期値(best\_score)を十分大きく設定
2. 1手先の着手可能箇所に×を置いた盤面を考える.  
その盤面の評価が定まっていればbest\_scoreと比較し,  
小さい方を採用する. 定まっていなければその盤面  
に対し, 最大化プレイヤーのMiniMaxを行う.
3. 2を全ての着手可能箇所に対して繰り返す.

```
int best_score = 2;  
for (const auto& move : available_moves(board)) {  
    board[move] = 'X';  
    best_score = min(best_score, minimax(board, true, depth - 1));  
    board[move] = move + '0';  
}  
return best_score;  
}
```

# サンプルプログラム(最大化プレイヤー手番)

1. 初期値(best\_score)を十分小さく設定
2. 1手先の着手可能箇所に○を置いた盤面を考える. その盤面の評価が定まっていればbest\_scoreと比較し, 大きい方を採用する. 定まっていなければその盤面に対し, 最小化プレイヤーのMiniMaxを行う.
3. 2を全ての着手可能箇所に対して繰り返す.

```
if (Oplayer) {  
    int best_score = -2;  
    for (const auto& move : available_moves(board)) {  
        board[move] = 'O';  
        best_score = max(best_score, minimax(board, false, depth - 1));  
        board[move] = move + '0';  
    }  
}
```

3\*3の正方格子を場としておきます.

先手プレイヤーから交互に一つずつ格子点に石を置いていきます. そのときに同一円周上に四点以上石があるような状態 (共円) を作ってしまうと即座に負けとなってしまいます.

負けなかったプレイヤー, つまり, 相手に共円を作らせたプレイヤーの勝ちとなります.

(実際のゲームでは相手が共円を作ったとしても, 指摘がなければゲームは続きますが, 今回は考慮しません)

<https://yambi.jp/kyouen/>

次のゲームは**3 to 15**と呼ばれるゲームです.

最初に1-9までの1枚ずつのカードを2人の真ん中に（以降「場」とします）置きます.

先手と後手を決めて先手から場のカードを1枚ずつ取り自分のカードとします.

自分のカードのうちいずれかちょうど3枚で合計が15になれば勝ち.

10枚ともとり終えたとき(先手が5枚, 後手は5枚)どちらも15を作れなければ引き分け.

---



ゲーム進行例(先手の手札を右に，後手を左に表示)



4

1 2 3

5 6 7 8 9

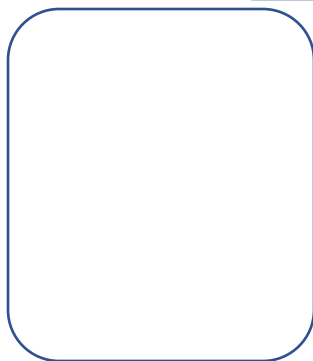
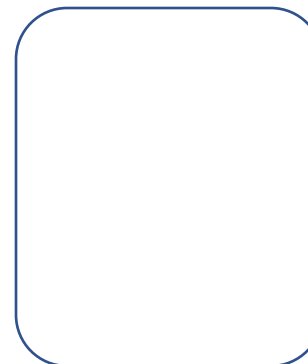
5

4

1

2

3



6

7

8

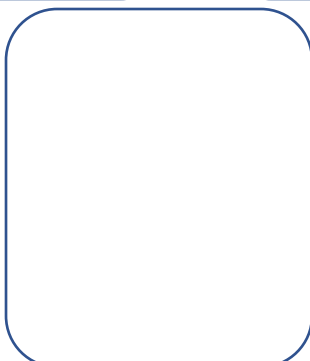
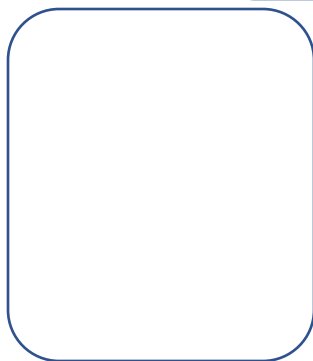
9

5

1

2

3



7

8

9

4

6

5

8

1

2

3

7

9

4

6

5

8

1

3

7

9

4

6

2

5

8

7

1

3

9

4

6

2

5

8

7

1

3

右(先手)の勝ち

4

6

2

9



# 課題1(7-9班, 共円ゲーム)

1. 「共円ゲーム」を $3 \times 3$ の正方格子盤面で行うとき、『必ず先手が勝てる』か『必ず後手が勝てる』かのどちらに当てはまるか予想をしてください。
2.  $3 \times 3$ の正方格子盤面上に頂点をプロットする時、ちょうど4点を通るような円となる4点組は何通りあるか、教えてください
3. 「共円ゲーム」を $3 \times 3$ の正方格子盤面で行うとき、『必ず先手が勝てる』か『必ず後手が勝てる』かのどちらに当てはまるかアルゴリズム実装などを通して、検証してください。

「3 to 15」は『必ず引き分けに持ち込める』ことが知られています。  
0のカードを追加した「3 to 15」に対して、『必ず先手が勝てる』か  
『必ず後手が勝てる』か『必ず引き分けに持ち込める』のうち、ど  
れに当てはまるか予想してください。

0のカードを追加した「3 to 15」に対して、『必ず先手が勝てる』か  
『必ず後手が勝てる』か『必ず引き分けに持ち込める』のうち、ど  
れに当てはまるかアルゴリズム実装などを通して、検証してください。

Moodle上の該当提出先に提出.

ファイル名は(半角で学籍番号)\_report.pdfとする.

ファイルサイズが大きすぎる場合, moodleで受け付けられない可能性があるため, 気をつけること. また, その場合は事前に相談すること.

締め切りは5/13 23:59とします.