

知能システム学

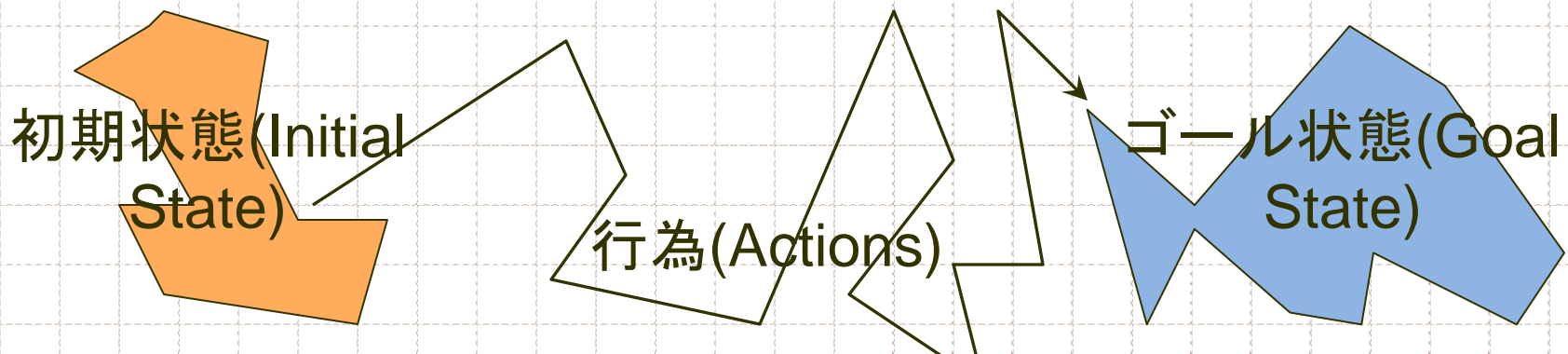
第12回

探索による問題解決(5)、
情報を持つ探索

ソフトウェア情報学部
David Ramamonjisoa

ゴールを用いるエージェントの構築

- エージェントの環境を状態で表現する (**state**)?
- エージェントのゴールは何か(**goal** to be achieved?)
- 可能な行為は何かある(What are the **actions**?)
- 問題を解決するためにどのような情報が状態と状態遷移に記述すべきか



探索戦略

◆ 探索戦略選択のための四つの基準

- 完全性: 解が存在するとき, それを見つけることが保証されているか?
- 最適性: いくつか異なる解があるとき, 戦略は最も良い解を見つけるか?
- 時間計算量: 解を見つけるまでにどれくらい時間が掛かるか?
- 空間計算量: 探索を行うためにどのくらいメモリを必要とするか?

◆ 情報を持つ探索(informed search), ある探索

- 現在の状態からゴールに至る順路の中で、最小コストの順路(最適順路)などを考えて効率的に行う。

ゲームプレイング

- ◆ 将棋、チェス、チェッカー、囲碁、オセロのような、二人ゲームを考える（ボードゲーム）
- ◆ そのようなゲームのコンピュータプログラムは、どうつくれば良いのか。
 - 探索問題の一種。
 - 良さそうな手を打つ。明らかに無駄な手は打たない。
 - 相手がこちらを妨害する。そのような状況で、最善の手を考える。
- ◆ どれくらい強いプログラムが作られたか。

情報をもつ探索戦略(Informed search strategies)

- ◆ **情報をもつ探索戦略**は問題定式化の使用可能情報のみ利用する。
- ◆ 敵対的探索(adversarial search)
 - 相手の手番が予測不可能ため、可能な手番をすべて探索する
 - ゲームの時間制限があるため、手番(ゴール)を近似する

目次

◆ ゲームプレイング

- ミニマックス法
- α - β 法

人口知能とゲームプログラミングの パイオニア

1949 Shannon paper

1951 Turing paper

1958 Bernstein program

55-60 Simon-Newell program

(α - β McCarthy?)

61 Soviet program

66 – 67 MacHack 6 (MIT AI)

70's NW Chess 4.5

80's Cray Blitz

90's Belle, Hitech, Deep Thought,
Deep Blue

どれくらい強いプログラムが作られたか

- ◆ チェッカー、オセロ、バックギャモンなどは、コンピュータプログラムのほうが人間よりも強い。
- ◆ 1997年春、IBMで開発されたDeep Blueと呼ばれる専用並列コンピュータが、人間のチェスチャンピオン、Gary Kasparovと戦い、3.5対2.5でDeep Blueが勝った。

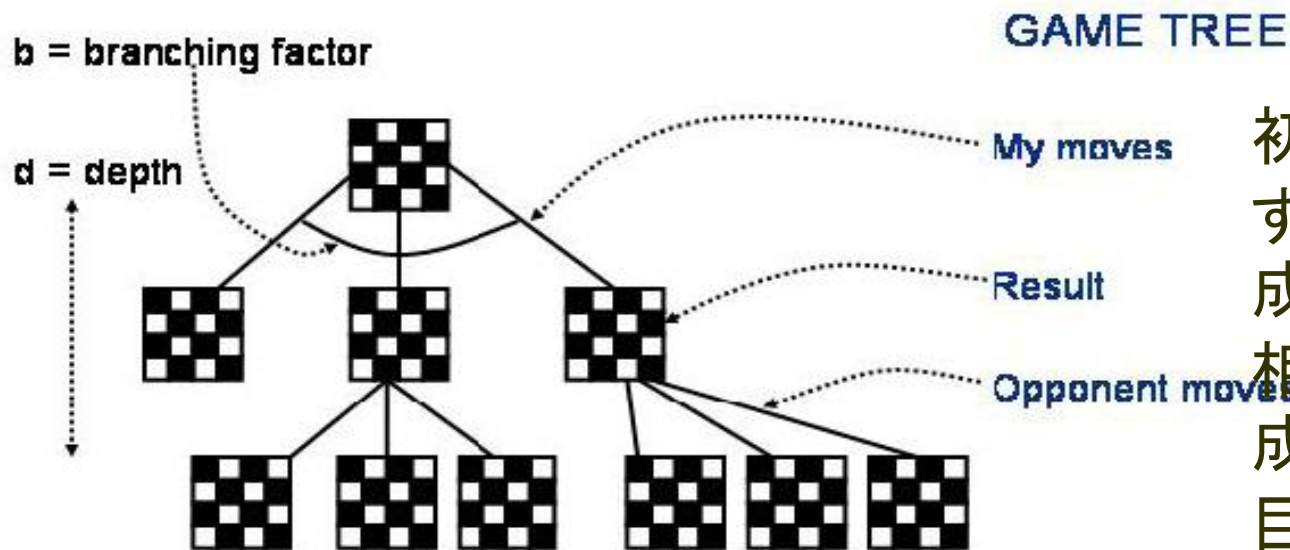
ゲーム木の探索

- ◆ 初期状態: 盤上の初期位置、プレイヤー(どちらの手番かの指示)
- ◆ オペレータ: プレイヤーが指すことのできる合法手を定義する。
- ◆ 終端テスト: ゲームの終了を決定する。
- ◆ 効用関数: ゲームの結果を数値として与える。チェスでは結果は勝ち、負け、引き分けのどれかので、値は+1,-1,0で表す。
- ◆ ゲーム木: ゲームに可能な状態を表す

ゲーム木の探索

- ◆ 探索の解は経路ではなく、指し手列を探索し、次の指したい手のみを決定する
- ◆ 相手の手によって最善(ベスト)の指し手が決定できる。
- ◆ ゲーム木では、1手(1 move)は2つの半手(half-move)あるいは2つのプライ(ply)から成る。日本で言う1手は1プライと呼ぶ。西欧ではmoveは先手後手合わせたものを指す。

ゲーム木の探索



初期状態から
すべて合法手を生成し、
相手の合法手も生成する。
巨大な探索木になる。

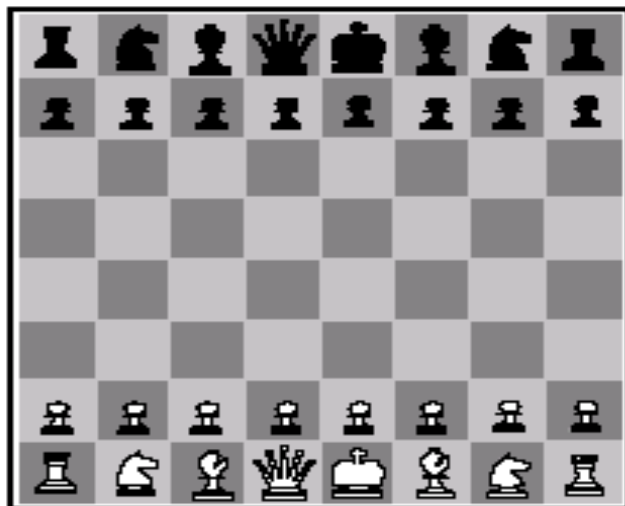
Chess

b = 36

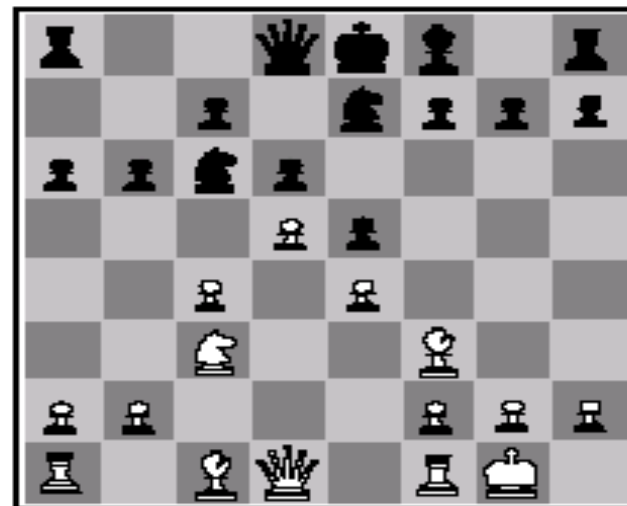
d > 40

36⁴⁰

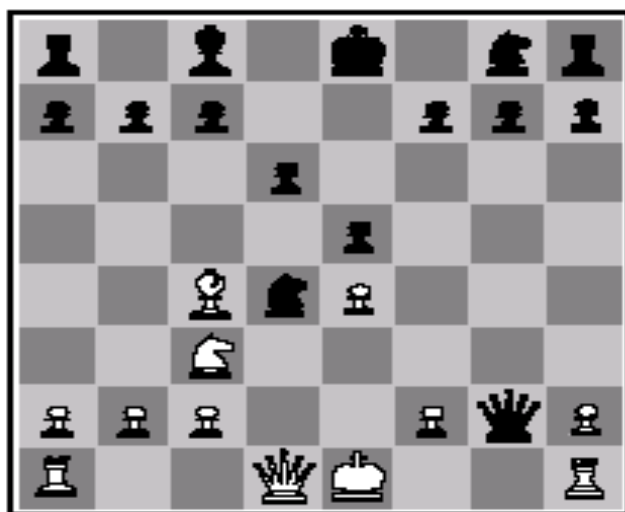
is big!



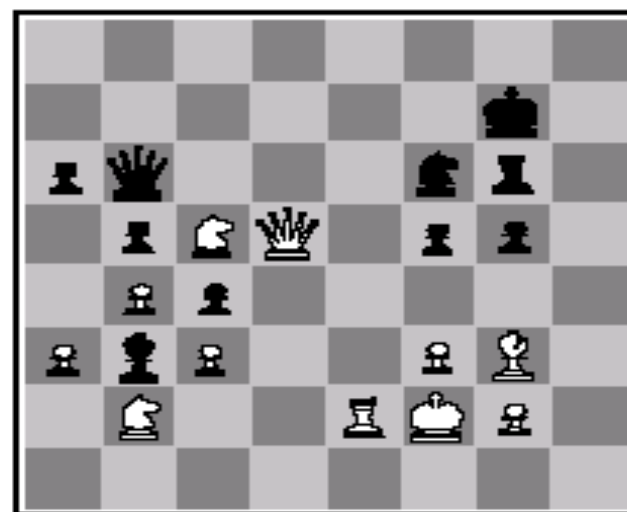
(a) White to move
Fairly even



(b) Black to move
White slightly better



(c) White to move
Black winning

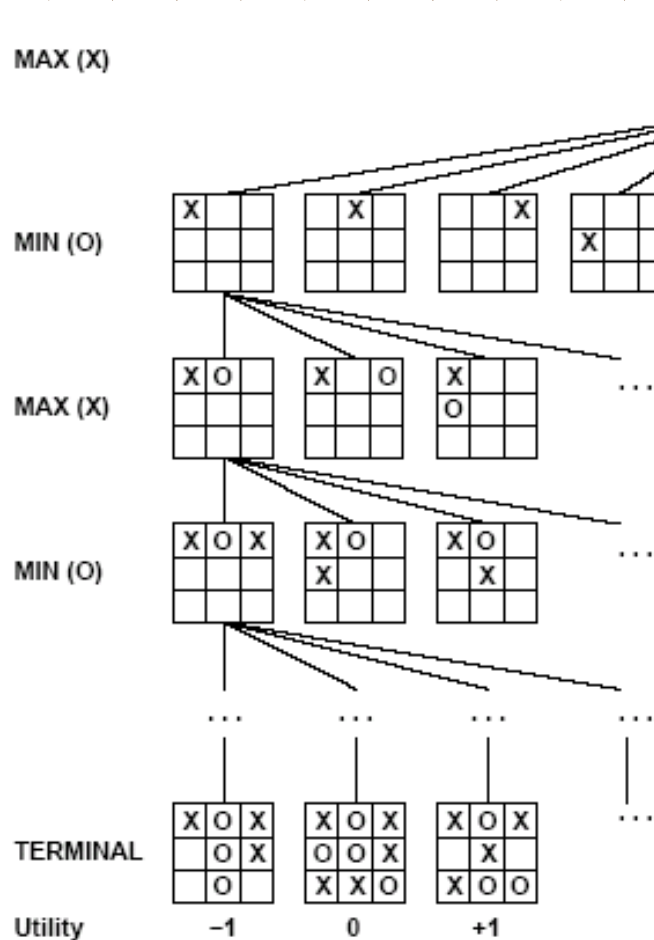


(d) Black to move
White about to lose

ゲーム木の探索

- ◆ 2つのプレイヤー: MAXとMIN、MAXが先手、MINが後手にする。
- ◆ 後はゲームが終るまで交互に指し続ける。ゲームが終わったときにある得点が勝者に与えられる(敗者に罰が与えられるときもある。)
- ◆ MINがどんな手を指そうと勝ちの終端状態に達するような戦略を見つける必要がある。
- ◆ 仮定: MAXとMINは最適な手を行う。

ゲーム木の探索:三目ならべ

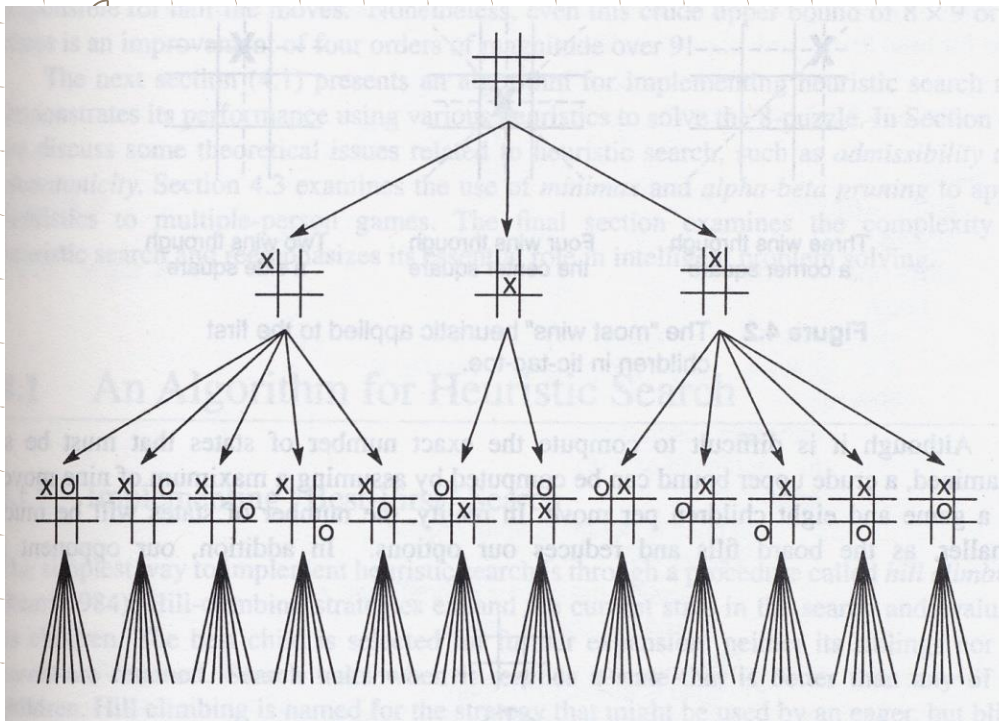


三目ならべゲームの(部分的)探索木。最も上の節点は初期状態で、MAXが先手でどこかに×を置く。MIN(O)とMAX(×)を交互に置いていく。ゲームのルールに応じた効用が割り当てられた終端状態へ最終的に達するまで続く。

状態空間:

可能な状態数: $3^9 = 19,683$ 、
探索木の状態数: $9! = 362,880$

三目ならべのヒューリスティック

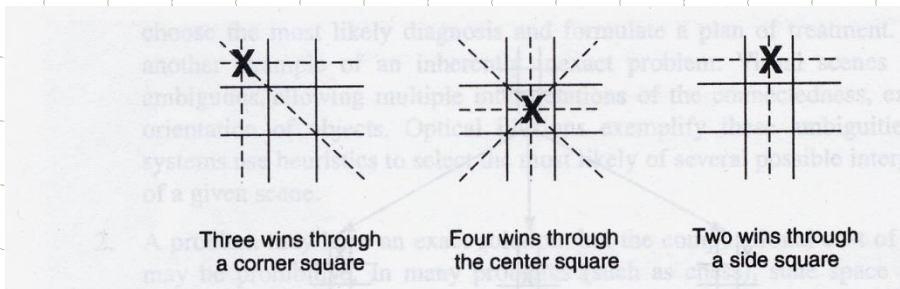


三目ならべゲームの(部分的)探索木。

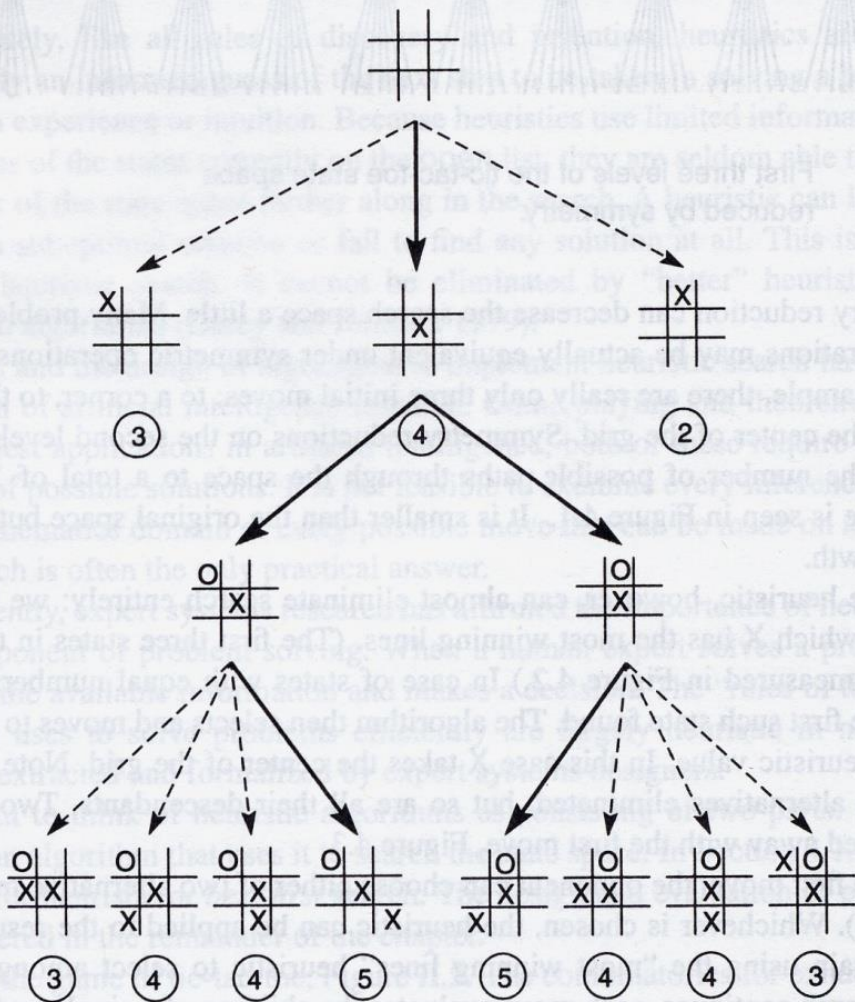
ポジションの回転と反射が同じと見なされる場合、わずか26,830のゲームしかありません。

MAX(×)のヒューリスティックは勝ち状態の数で評価する。

左上:3、間中:4、間中上:2



三目ならべのヒューリスティック



三目ならべゲームの(部分的)探索木。

MAX(×)のヒューリスティックは勝ち状態の数で評価する。

深さ1:

左上:3、間中:4、間中上:2

深さ3:

左部分:3,4,4,5

右部分:5,4,4,3

この時点ではMINIMAX最善手は、

MAX(×)は①間中、MIN(O)は①左上、MAX(×)②右下

ミニマックス法

- ◆ 評価値を考え、相手の手番には(相手が最もよい手をとると予想されるので)最小値(ミニ)、自分の手番には最大値を取るように木を逆にたどって各局面での指し手を決める手法をミニマックス法と呼ぶ。
- ◆ ミニマックス法では、評価値が正しい限り、最善の手を保証する。
- ◆ 先読みの手数に対して調べるべき節点の数は、指数関数的に増大する。

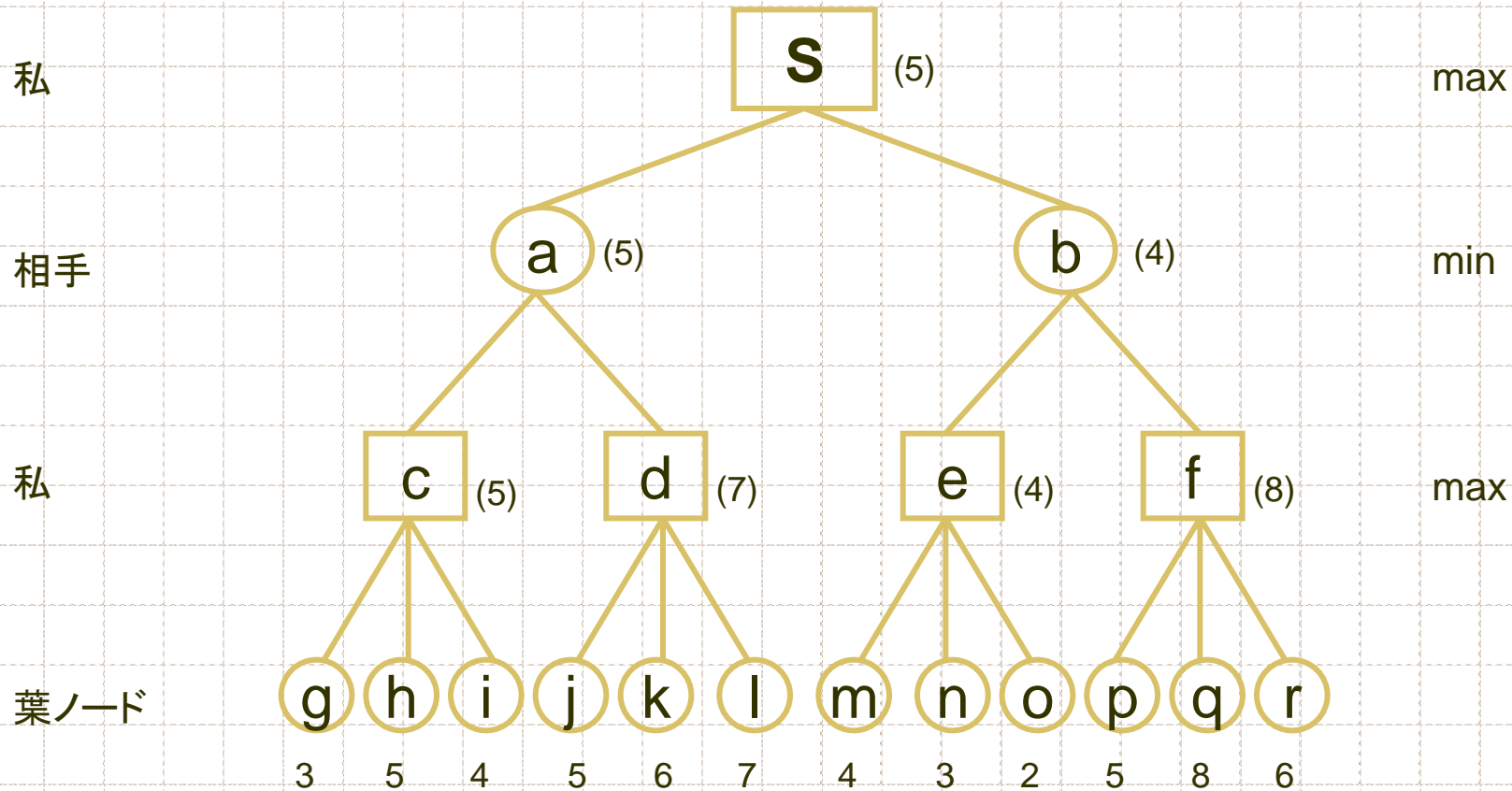
ミニマックス法 (Minimax algorithm)

```
function MINIMAX-DECISION(state) returns an action  
   $v \leftarrow \text{MAX-VALUE}(\textit{state})$   
  return the action in SUCCESSORS(state) with value  $v$ 
```

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

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

ゲーム木



ミニマックス法の性質(Properties of minimax)

- ◆ 完全(Complete)? Yes (if tree is finite)
- ◆
- ◆ 最適(Optimal)? Yes (against an optimal opponent)
- ◆
- ◆ 時間計算量(Time complexity)? $O(b^m)$
- ◆
- ◆ 空間計算量(Space complexity)? $O(bm)$ (depth-first exploration)
- ◆ チェスのゲーム: $b \approx 35$, $m \approx 100$ マスターレベル
→ 正解の解を求めるのは不可能である。

ミニマックス法のまとめ

- ◆ ゲームは魅力的であり、ゲームプログラミングはそれ以上に魅力的である。AIにおけるゲームプログラミングは自動車産業におけるF1レース、自動運転に相当すると言える。
- ◆ ゲームは、初期状態、作用素、終端集合、効用関数から定義される。
- ◆ 完全情報の2人ゲームでは、ミニマックスアルゴリズムにより、ゲーム木を完全に教え上げることで一方のプレイヤーにとっての最善手を決定できる。
- ◆ ミニマックス法の探索は深さ優先探索である。

参考ページ

- ◆ <https://ja.wikipedia.org/wiki/ゲーム木>
- ◆ <https://ja.wikipedia.org/wiki/ミニマックス法>