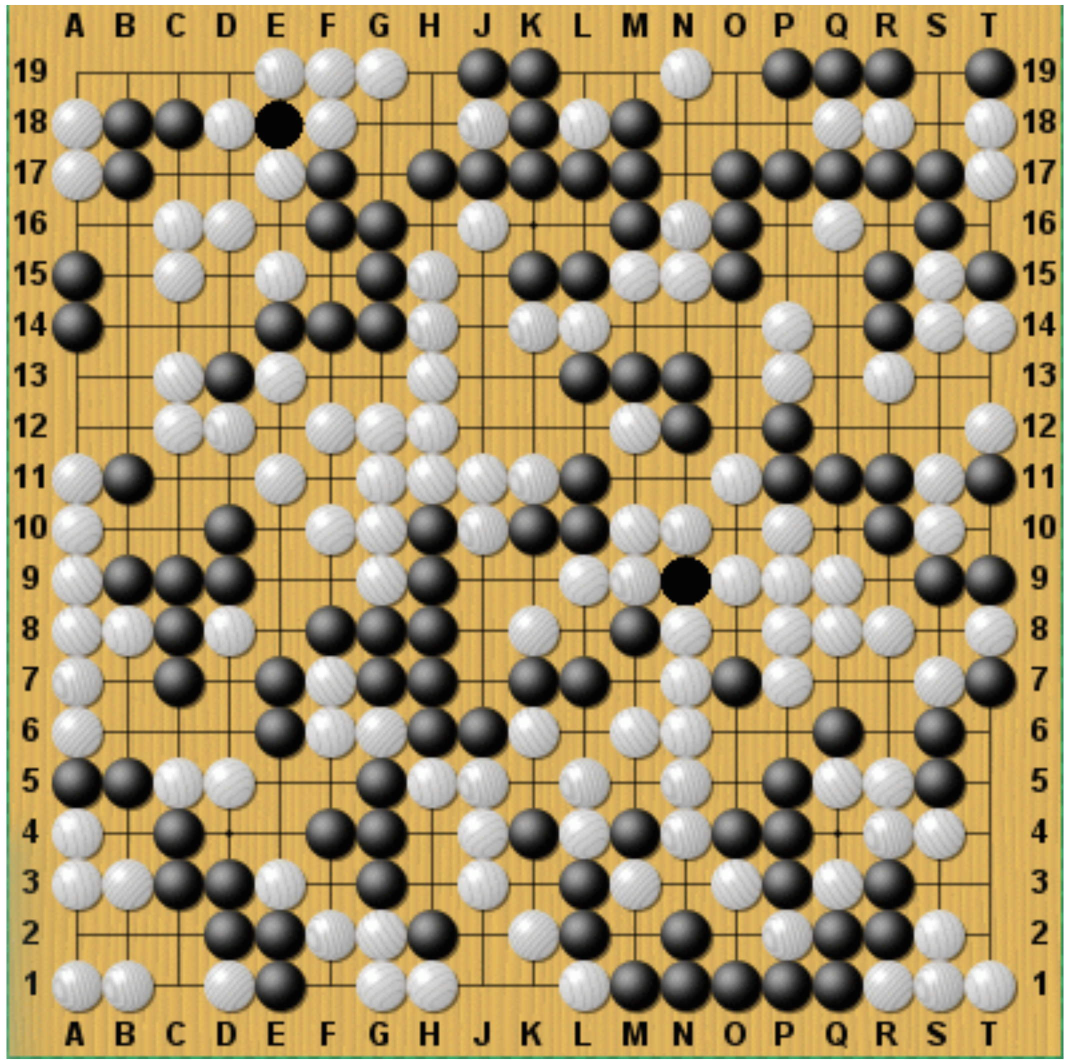
AlphaGo Zero

Part I Game Tree Search Techniques

- Game tree search
  - MinMax
  - Alpha-Beta pruning
- Monte-Carlo tree search
- AlphaGo Zero

Go



Some numbers

Chess

~10^46 - 10^49 legal positions

~10^123 game tree size

~ 35 branching 80 ply

Go

9x9

~10^38 legal positions

45 ply

19x19

208168199381979984699478633344862770286522453884530548425639456820927419612738015378525648451698519643907259916015628546089888314427129715319317557736620397247064840935

~10^169 legal positions

~250 branching 150 ply

1996 Deep Blue defeats Gary Kasparov
(but looses the match 2-4)
1997 Deep Blue defeats Gary Kasparov
(wins the match 3 1/2-2 1/2)

2007 MoGo defeats Gua Juan (5P) in 9x9 Go

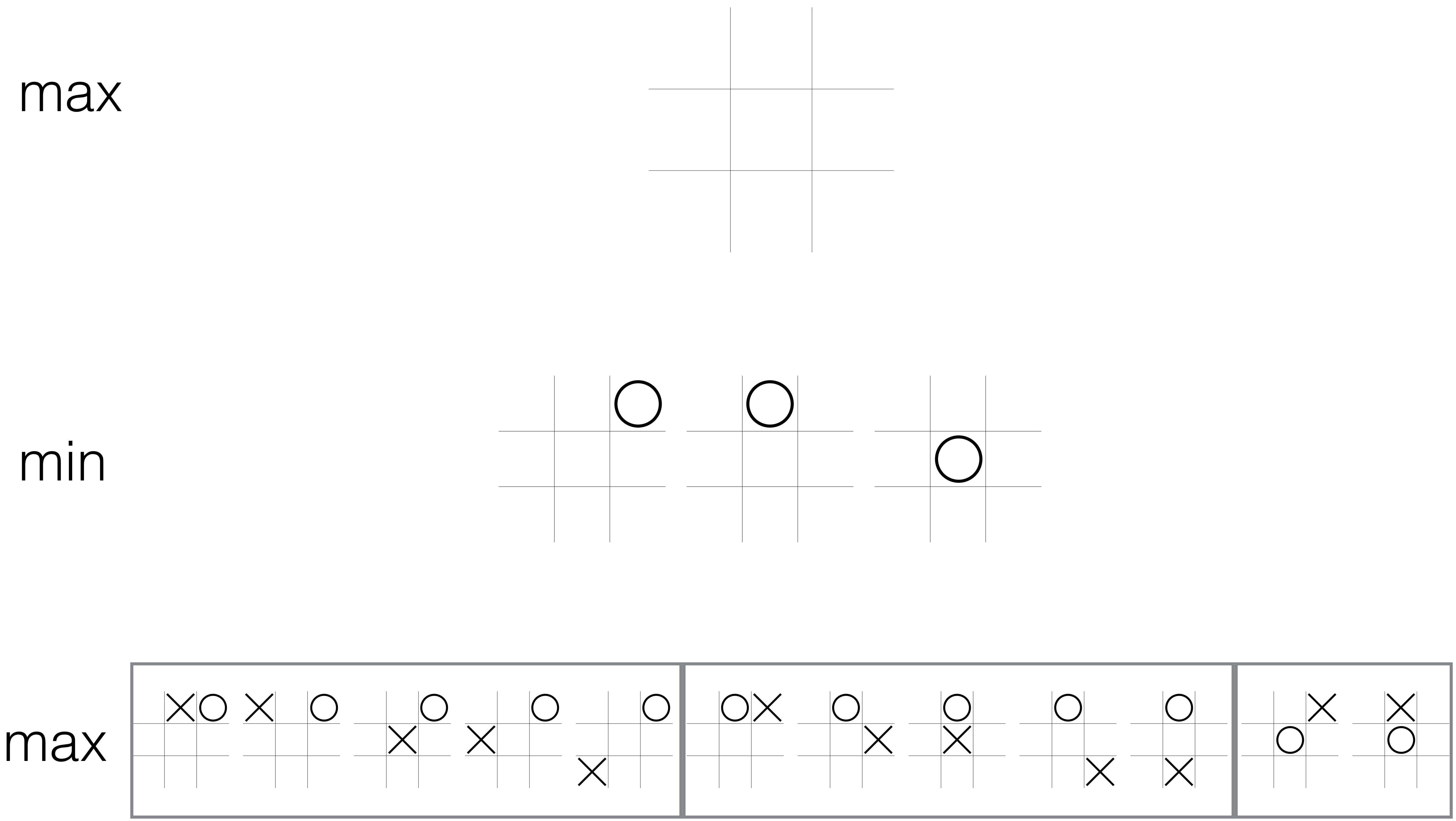
Arpad Rimmel, Olivier Teytaud, Chang-Shing Lee, Shi-Jim Yen, Mei-Hui Wang, et al.. "Current Frontiers in Computer Go". IEEE Transactions on Computational Intelligence and AI in games, IEEE Computational Intelligence Society, 2010

- October 2015
  AlphaGo Fan defeats European Champion Fan Hui (2 professional Dan)
- March 2016
  AlphaGo Lee defeats Lee Sedol (9 professional Dan).
- 2016/17
  AlphaGo Zero defeats both those programs and all other existing Go programs ..
- 2017
  Alpha Zero defeats best chess playing program Stockfish

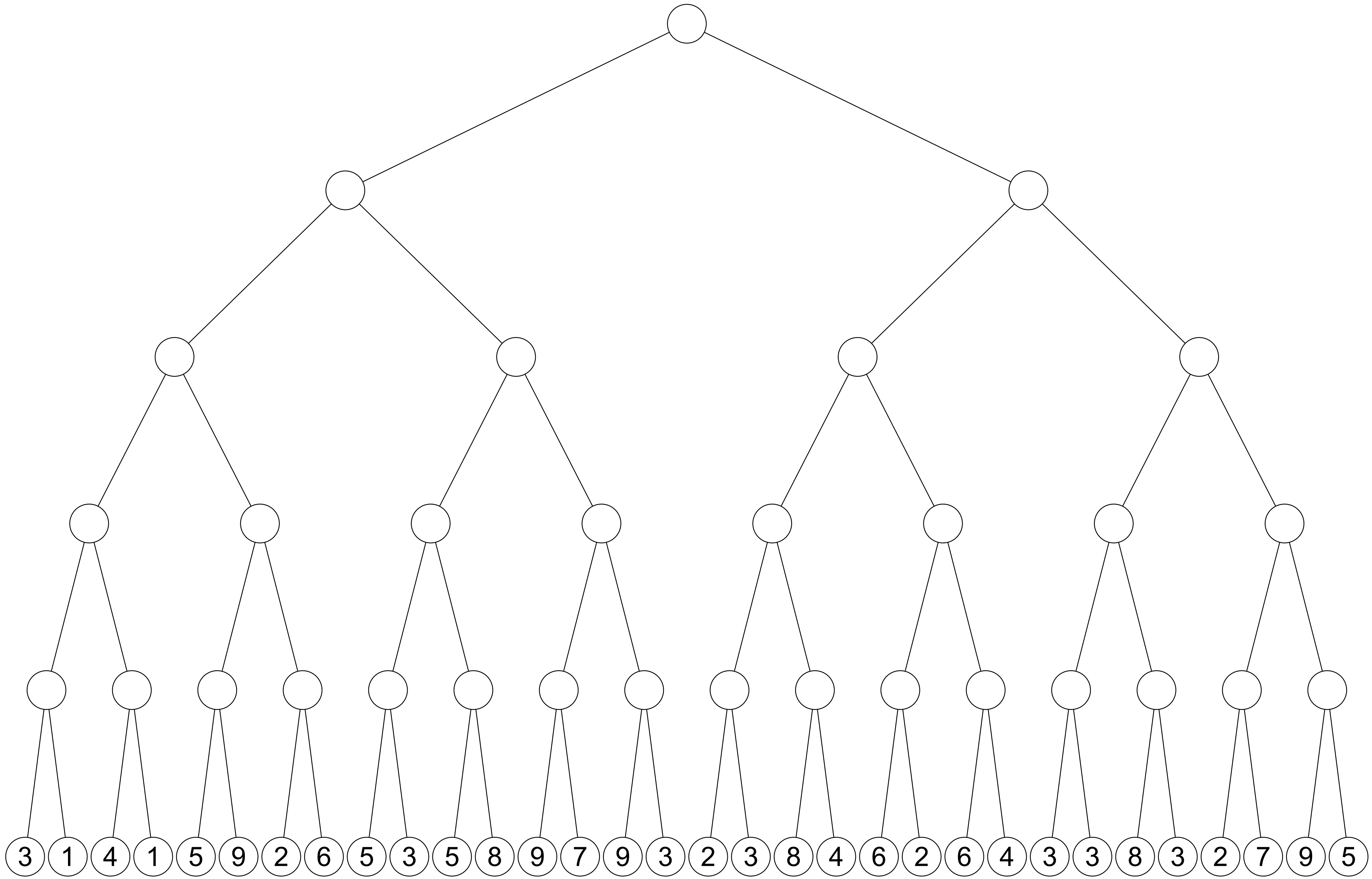
„Mastering the game of Go without human knowledge”, D. Silver et al., Nature **550**, 354–359 (2017)

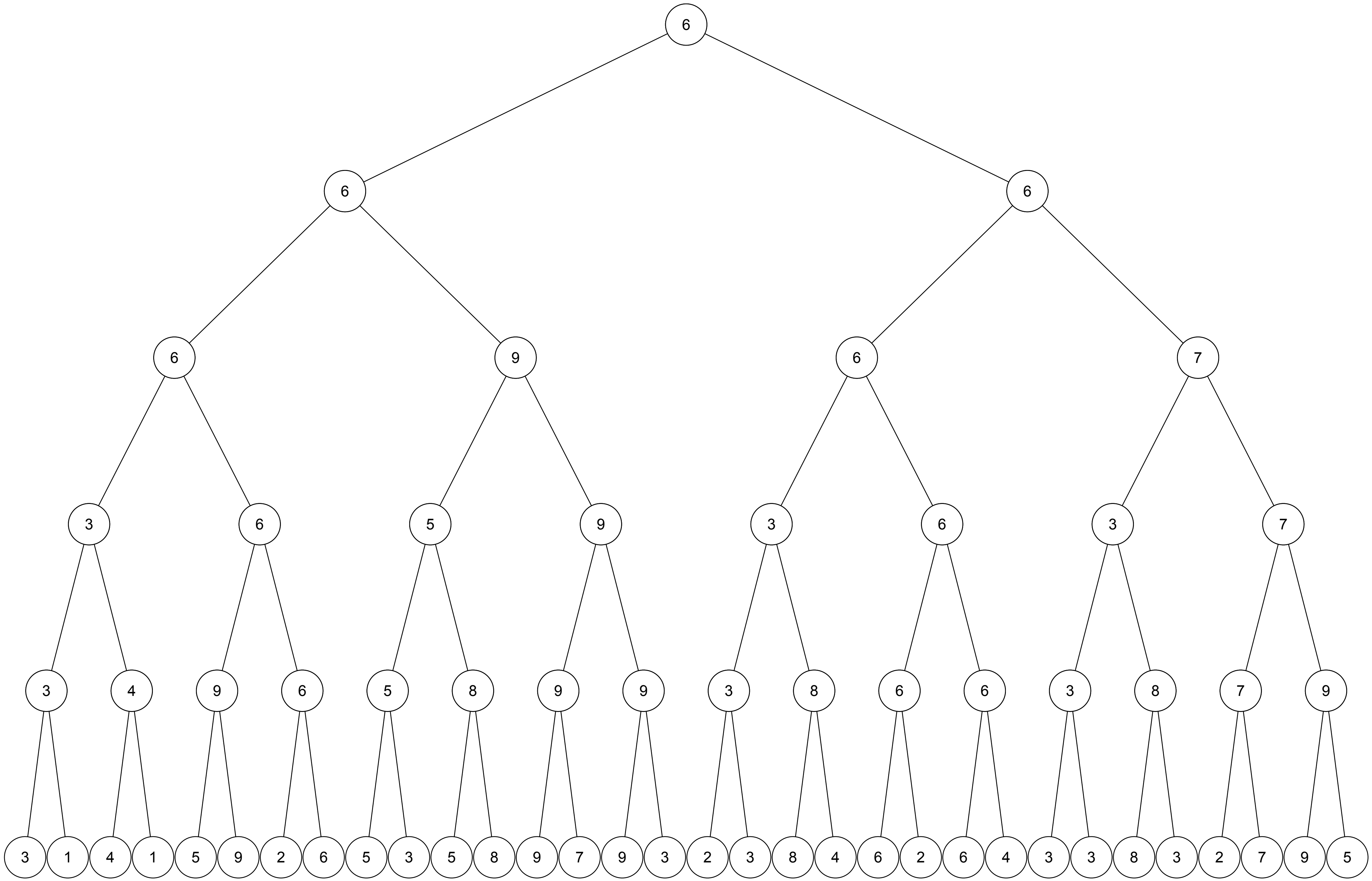
„Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm”, D. Silver et al., arXiv:17121815v1

Game tree



MinMax

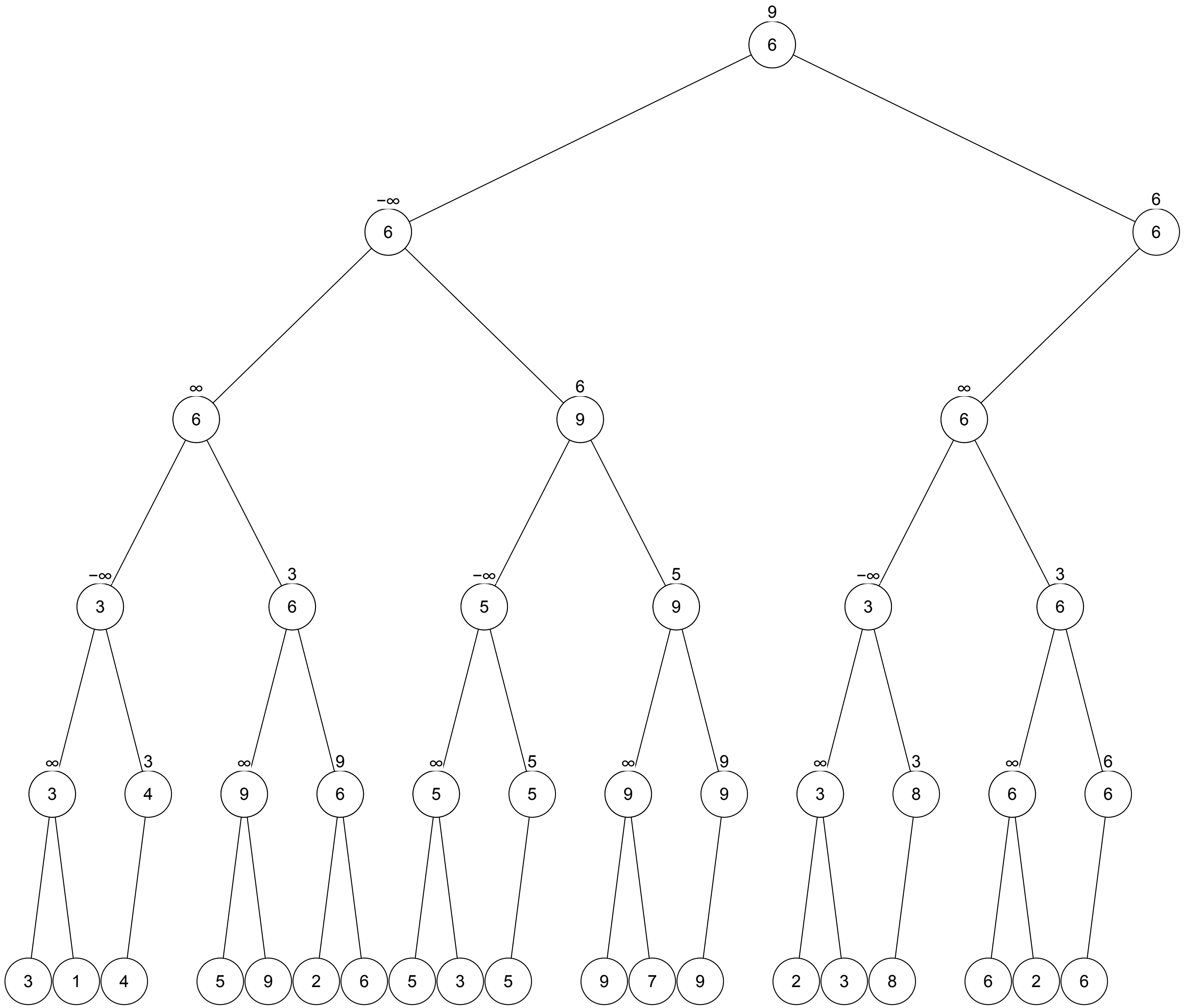



6
6
6
6
9
6
7
3
6
5
9
3
6
3
7
3
4
9
6
5
8
9
9
3
8
6
6
3
8
7
9
3
1
4
1
5
9
2
6
5
3
5
8
9
7
9
3
2
3
8
4
6
2
6
4
3
3
8
3
2
7
9
5

Min Max

```
minS[node_] := node
maxS[node_] := node
maxS[node_List] := Max[minS /@ node]
minS[node_List] := Min[maxS /@ node]
```

Branch and cut



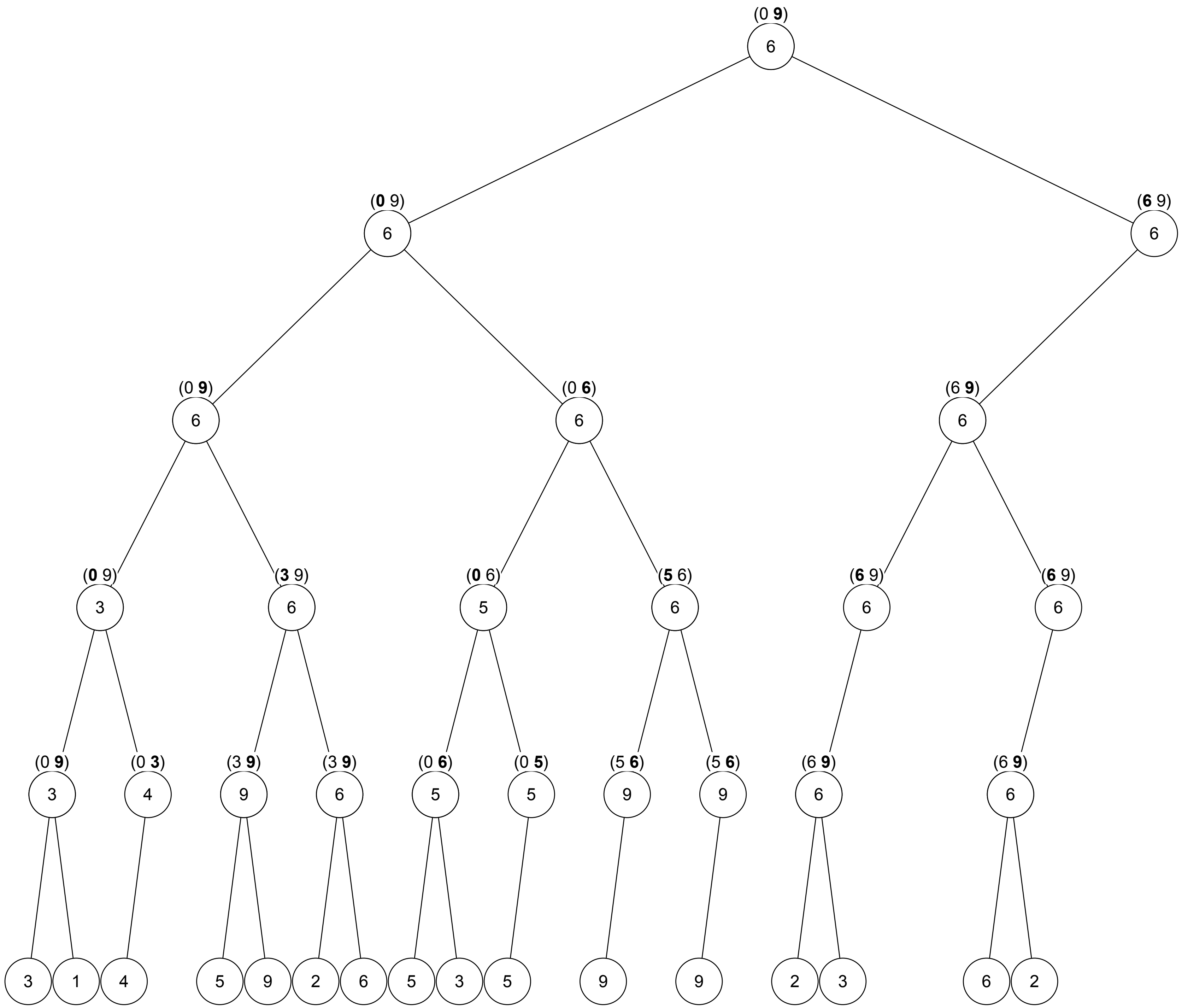
Branch and cut

```
BBMinS[node_, bound_] := node
BBMaxS[node_, bound_] := node

BBMaxS[node_List, bound_] :=
 Module[{m = -Infinity, t, i, n,  nodes = {}} ,
  For[i = 1, i <= Length[node], i++,
   t = BBMinS[node[[i]], m];
   m = If[t > m, t, m];
   If[m >= bound, Break[]]
   ];
  m
  ]

BBMinS[node_List, bound_] :=
 Module[{m = Infinity, t, i, n,  nodes = {}} ,
  For[i = 1, i <= Length[node], i++,
   t = BBMaxS[node[[i]], m];
   m = If[t < m, t, m];
   If[m <= bound, Break[]]
   ];
  m
  ]
```

α - β prunning



Alpha - Beta pruning

```
AlphaBetaMinS[node_, \[Alpha]_, \[Beta]_] := node
AlphaBetaMaxS[node_, \[Alpha]_, \[Beta]_ ] := node

AlphaBetaMaxS[node_List, \[Alpha]_, \[Beta]_] :=
 Module[{m = \[Alpha], t, i, n,  nodes = {}} ,
  For[i = 1, i <= Length[node], i++,
   t = AlphaBetaMinS[node[[i]], m, \[Beta]];

   m = If[t > m, t, m];
   If[m >= \[Beta], Break[]]
   ];
  m
  ]

AlphaBetaMinS[node_List, \[Alpha]_, \[Beta]_] :=
 Module[{m = \[Beta], t, i, n,  nodes = {}} ,
  For[i = 1, i <= Length[node], i++,
   t = AlphaBetaMaxS[node[[i]], \[Alpha], m];

   m = If[t < m, t, m];
   If[m <= \[Alpha], Break[]]
   ];
  m
  ]
```

D.E. Knuth, R.W. Moore
„An Analysis of Alpha-Beta Pruning”
Artificial Intelligence **6** (1975) 293.

Heuristics

In most games tree cannot be evaluated completelly

Enhancements

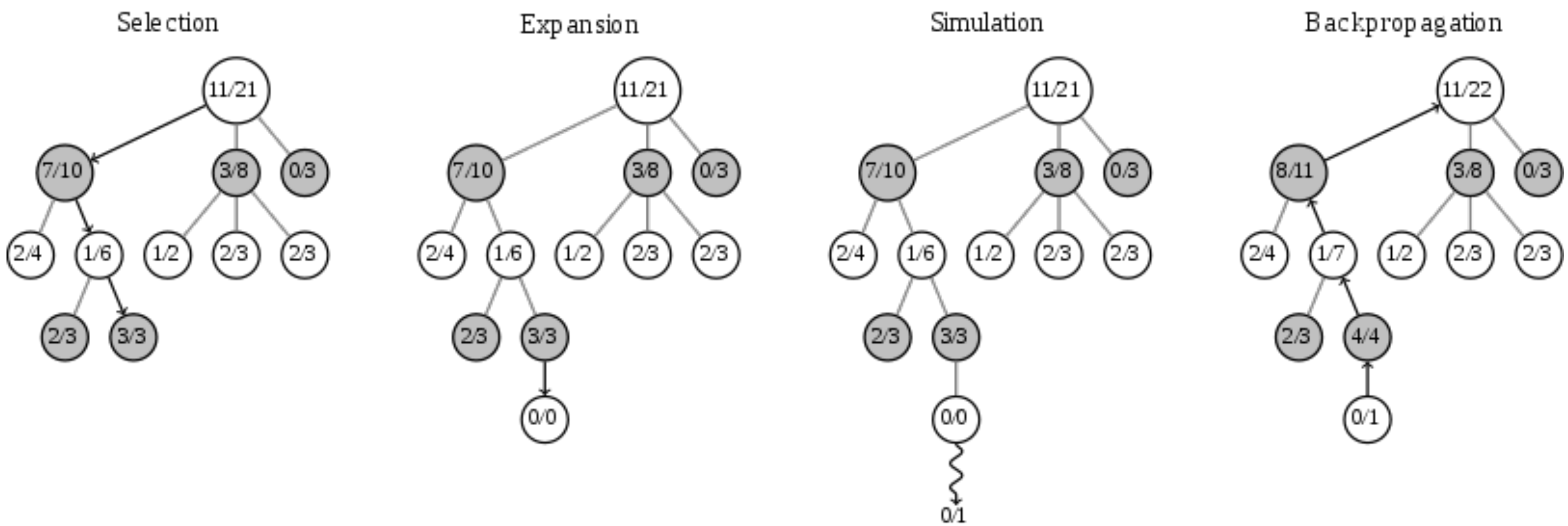
Move ordering

- Iterative deepening

Transposition tables

Parallelisation

Monte-Carlo tree search



C. Brown et al. "A Survey of Monte Carlo Tree Search Methods" IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. **4**, NO. 1, MARCH 2012

Monte-Carlo tree search

- Selection
- Expansion
- Simulation
- Backpropagation

Multi armed bandit

K - distribution

Maximize gain

Minimize regret

$$nE(X_{i_{max}}) - E(\sum_{t=1}^{n} X_{i_t})$$

Exploitation vs. Exploration

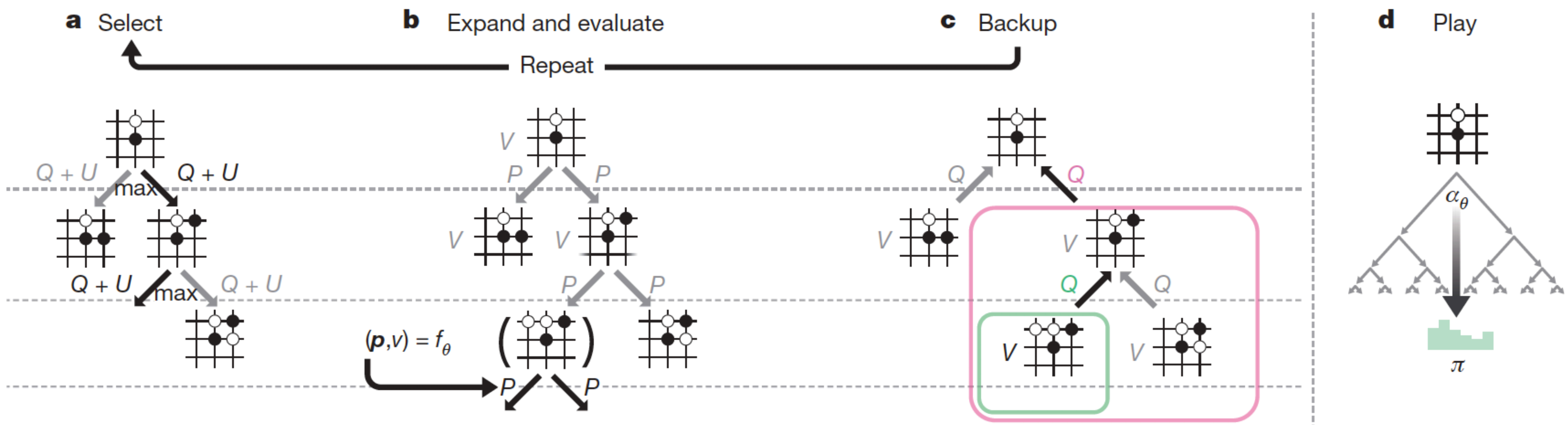
UCB1

$$I_t = \underset{i \in \{1,\ldots,K\}}{\operatorname{argmax}} \left\{ \overline{X}_{i,T_i(t-1)} + \sqrt{\frac{2\ln(t-1)}{T_i(t-1)}} \right\}$$

UCT

$$I_t = \operatorname*{argmax}_{v' \in \text{children of } v} \left\{ \frac{Q(v')}{N(v')} + +c\sqrt{\frac{2\ln N(v)}{N(v')}} \right\}$$

AlphaGo Zero



$$\pi(a|s) = \frac{N(a,s)^{\frac{1}{\tau}}}{\sum_b N(b,s)^{\frac{1}{\tau}}}$$

Alpha Go Zero PUCT

$$(\mathbf{p}, \nu) = f_\theta(s) \quad p_a = P(a|s)$$

$$\operatorname*{argmax}_a \left\{ \frac{Q(s,a)}{N(s,a)} + c_{puct} p_a \frac{\sum_b N(s,b)}{N_(s,a)} \right\}$$