Fall 2022 | Lecture 10

AI Game Playing

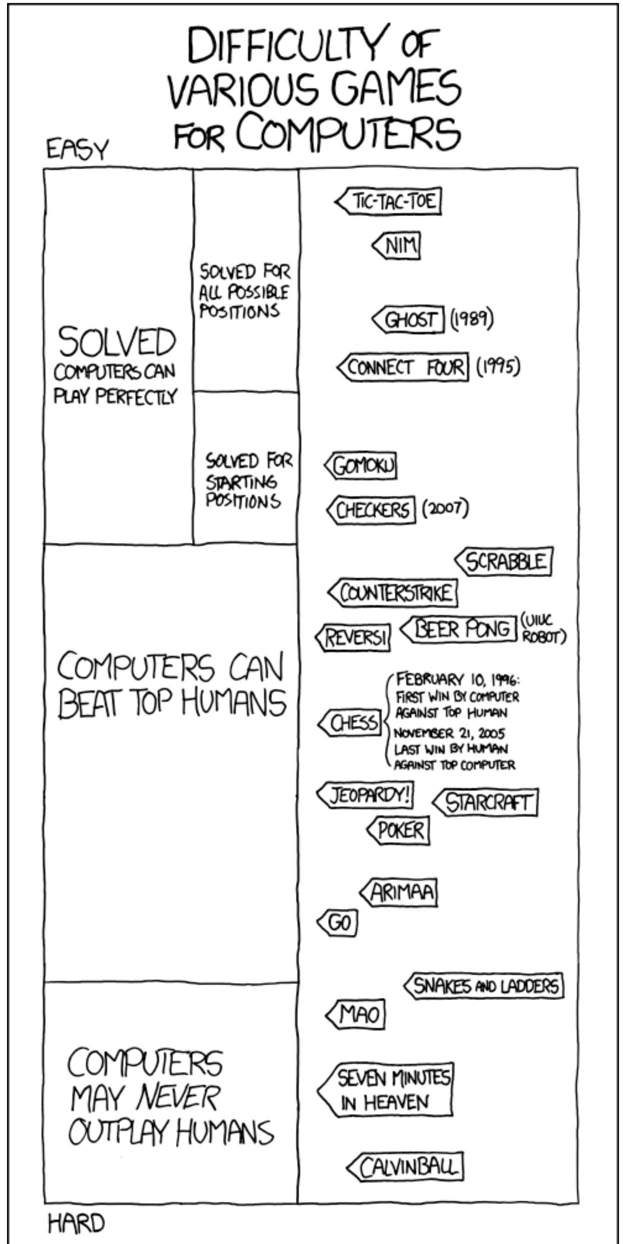
Ariel Procaccia | Harvard University

Kasparov vs. Deep Blue

1996-1997

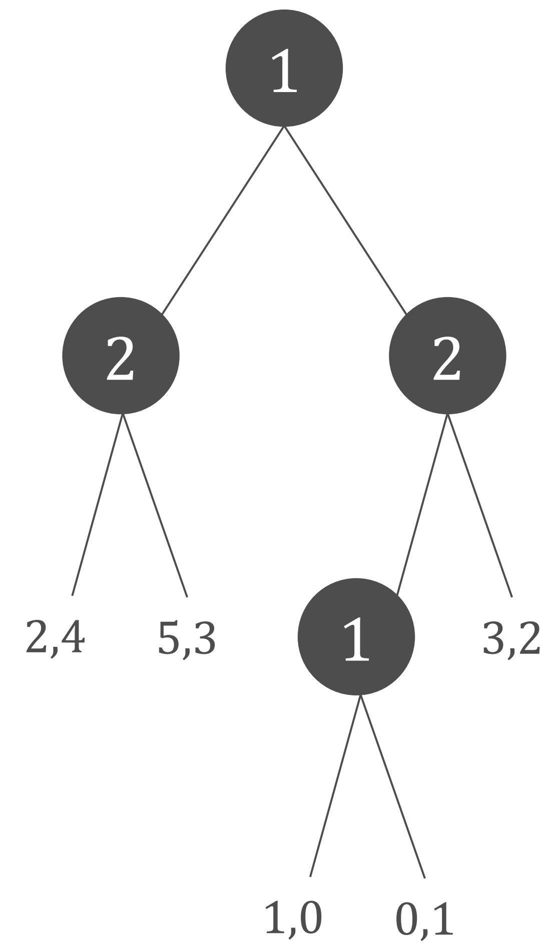
In 1997, the IBM supercomputer Deep Blue defeated Gary Kasparov. But Kasparov cried foul.

AI CRUSHES HUMANITY IN GAMING

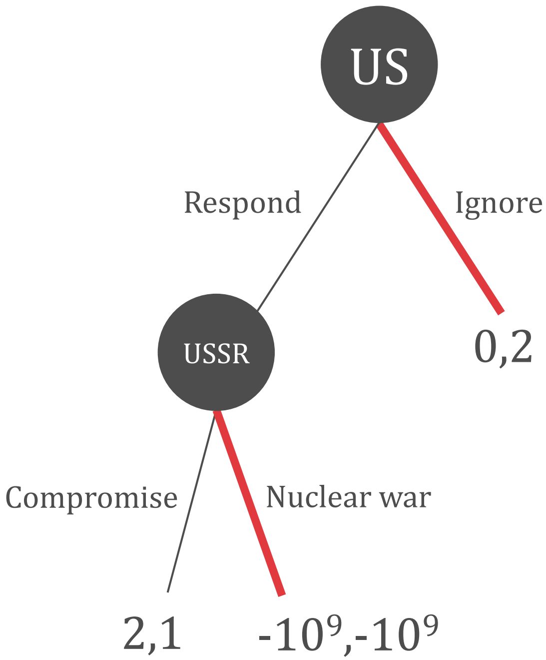

DIFFICULTY OF VARIOUS GAMES FOR COMPUTERS
EASY
SOLVED
COMPUTERS CAN PLAY PERFECTLY
SOLVED FOR ALL POSSIBLE POSITIONS
SOLVED FOR STARTING POSITIONS
TIC-TAC-TOE
NIM
GHOST (1989)
CONNECT FOUR (1995)
GOMOKU
CHECKERS (2007)
COMPUTERS CAN BEAT TOP HUMANS
SCRABBLE
COUNTERSTRIKE
REVERSI
BEER PONG (UIUC ROBOT)
CHESS
FEBRUARY 10, 1996: FIRST WIN BY COMPUTER AGAINST TOP HUMAN
NOVEMBER 21, 2005 LAST WIN BY HUMAN AGAINST TOP COMPUTER
JEOPARDY!
STARCRAFT
POKER
ARIMAA
GO
COMPUTERS MAY NEVER OUTPLAY HUMANS
SNAKES AND LADDERS
MAO
SEVEN MINUTES IN HEAVEN
CALVINBALL
HARD

EXTENSIVE-FORM GAMES

- Moves are done sequentially, not simultaneously
- Game forms a tree
- Nodes are labeled by players
- Leaves show payoffs



EXTENSIVE VS. NORMAL FORM

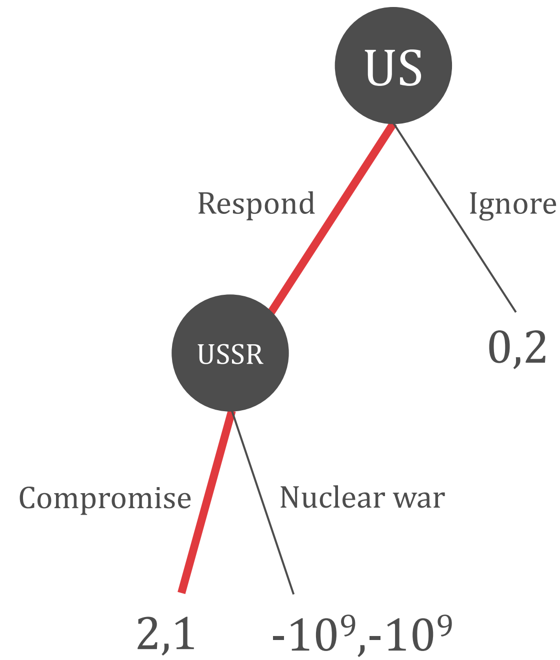


	Compromise	Nuclear war
Respond	2,1	$-10^9,-10^9$
Ignore	0,2	0,2

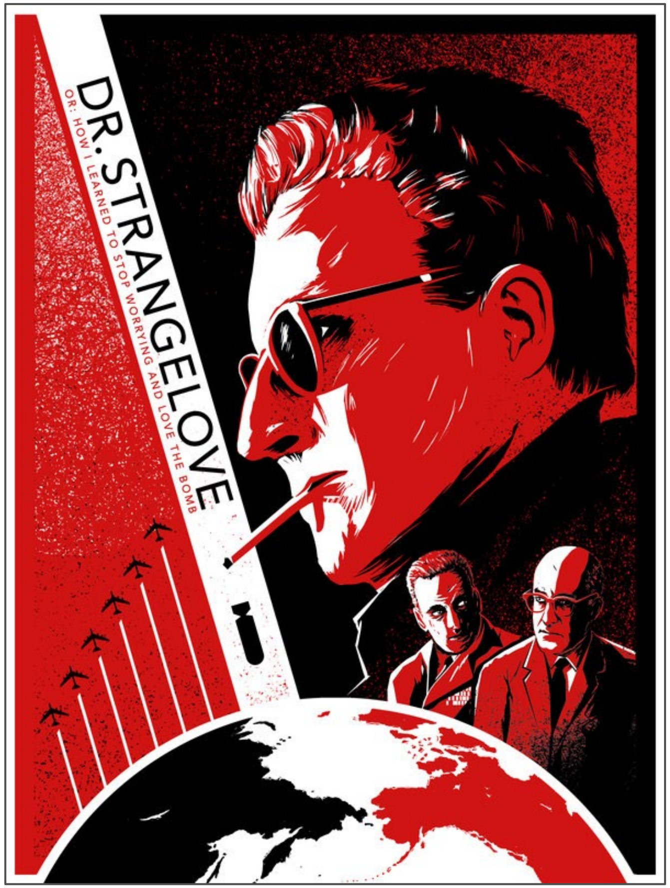
Problem: (ignore, nuclear war) is a Nash equilibrium, but threat isn't credible!

SUBGAME-PERFECT EQUILIBRIUM

- Each subtree forms a subgame
- A set of strategies is a subgame-perfect equilibrium if it is a Nash equilibrium in each subgame
- Players may be able to improve their equilibrium payoff by eliminating strategies!

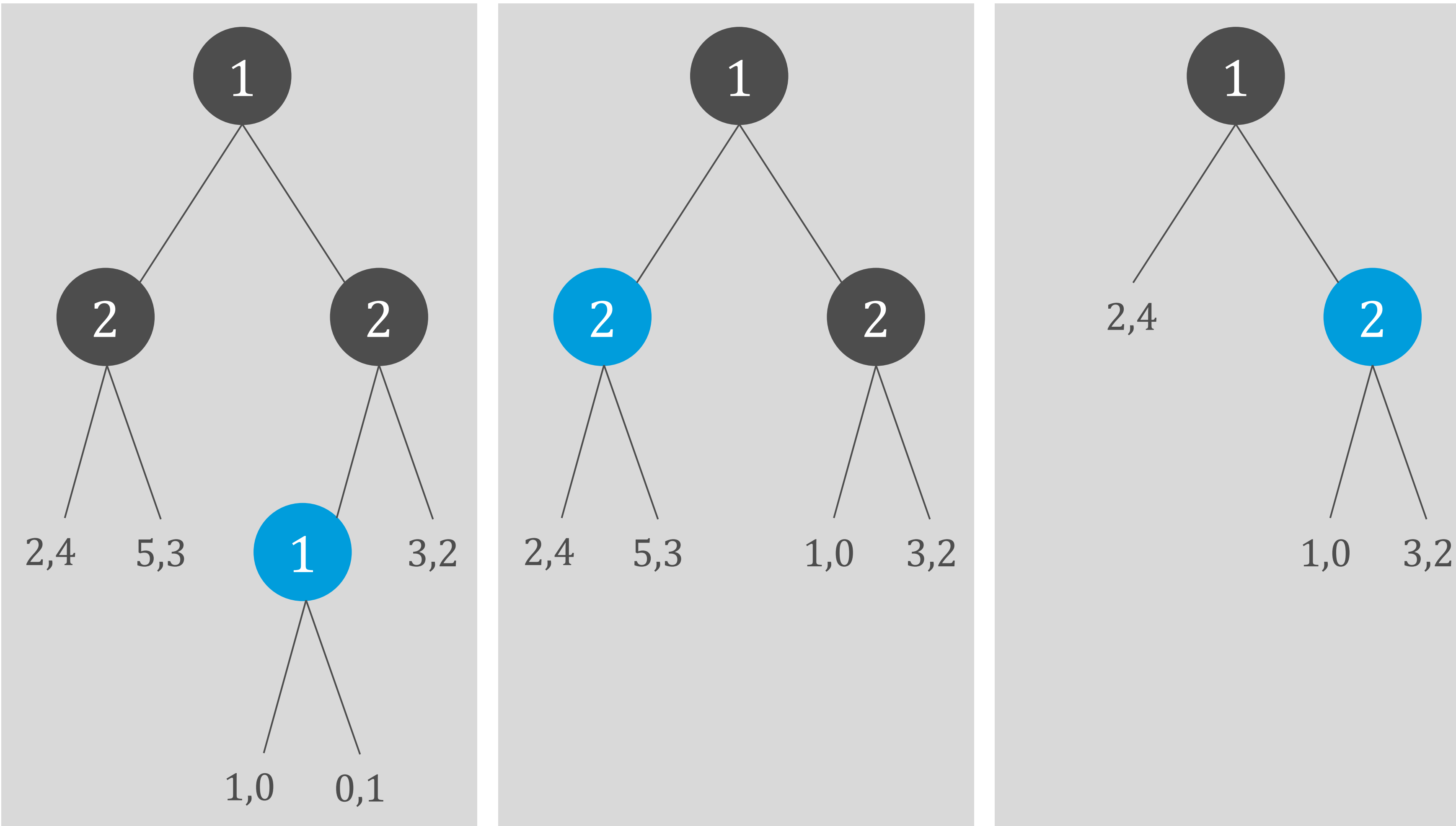


DOOMSDAY MACHINE

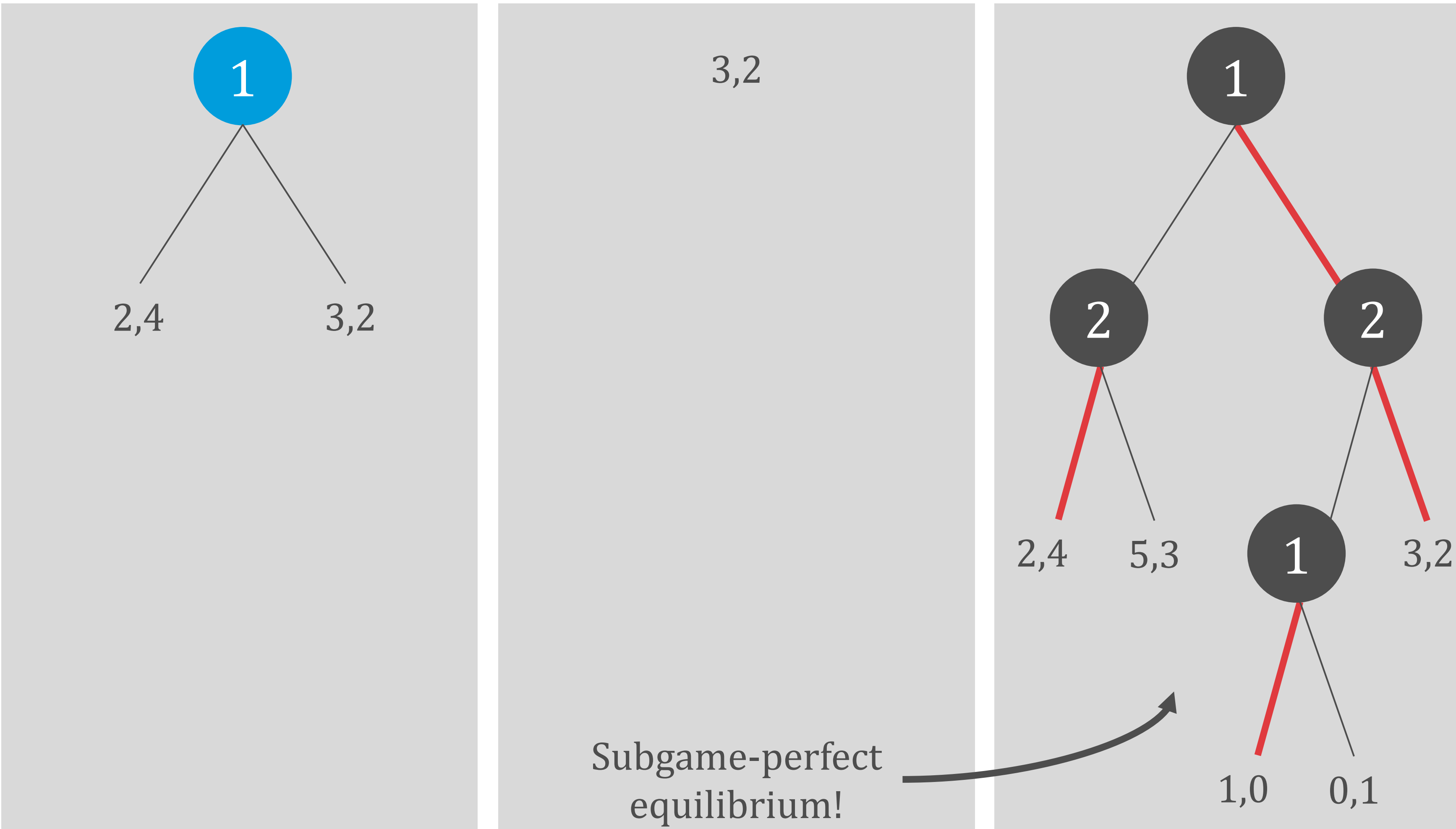


https://youtu.be/ozg7gEchjuM

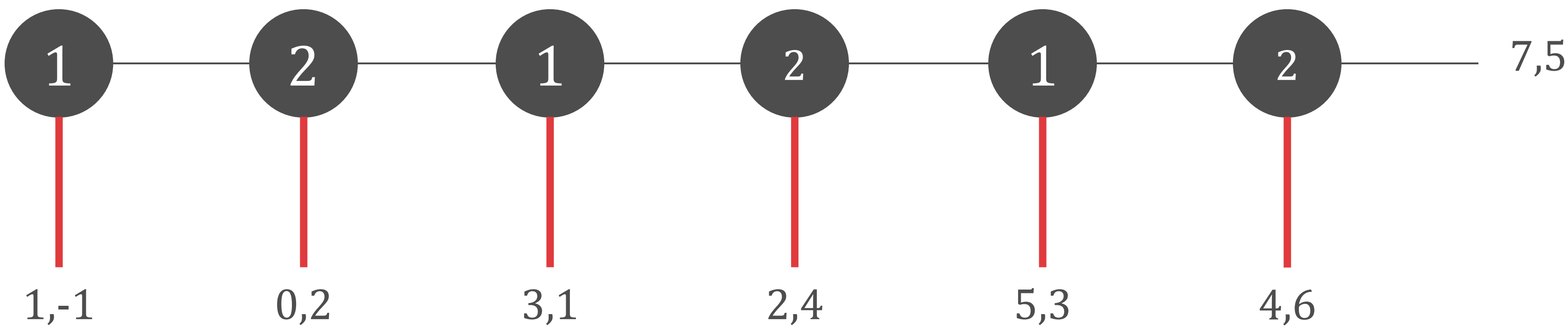
BACKWARD INDUCTION



BACKWARD INDUCTION



EXAMPLE: CENTIPEDE GAME

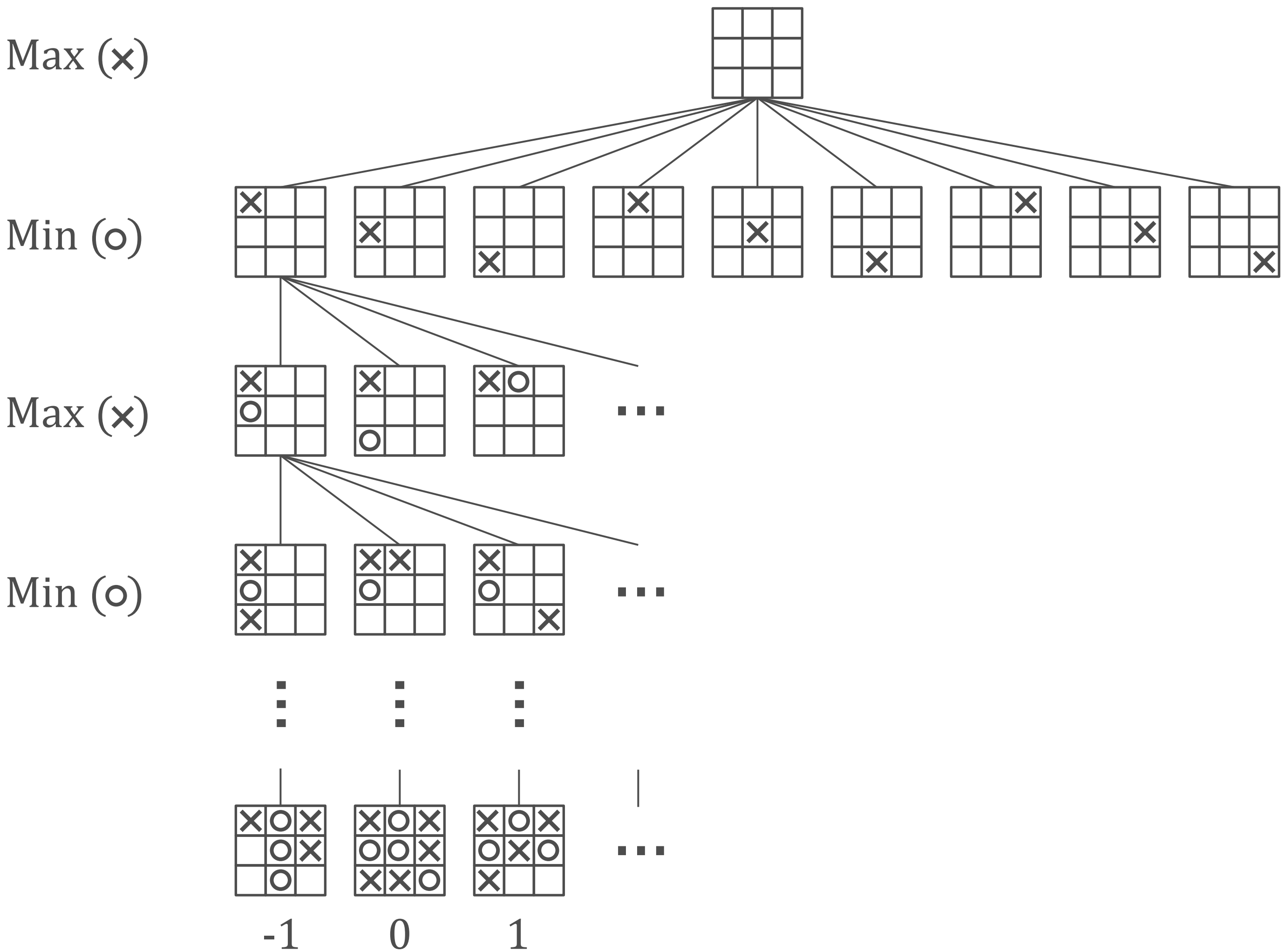


Even subgame-perfect equilibrium can lead to strange outcomes!

ZERO-SUM GAMES

- We represent games like chess, checkers, and go as two-player, zero-sum, extensive-form games between a max player and a min player
- A game position is a node in the game tree
- Children of a node are positions that are reachable in one move
- Each leaf specifies a payoff to the max player

EXAMPLE: TIC-TAC-TOE



MINIMAX SEARCH

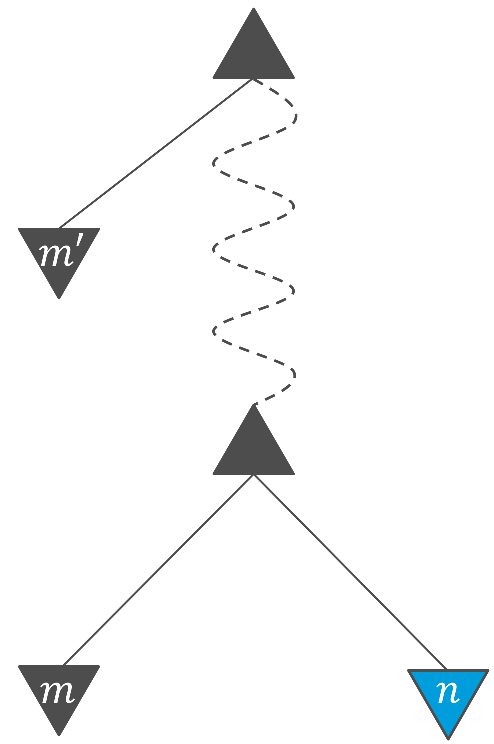
```
function MaxEval (node n)
      if n is a leaf then return PAYOFF(n)
      v ← −∞
      for all children n' of n
            v ← MAX(v, MINEVAL(n'))
      return v

function MinEval (node n)
      if n is a leaf then return PAYOFF(n)
      v ← ∞
      for all children n' of n
            v ← MIN(v, MAXEVAL(n'))
      return v
```

CHECKERS IS SOLVED

- Zermelo's Theorem [1913]: In any two-player, zero-sum, extensive-form game, either white can force a win, or black can force a win, or both sides can force a draw
- Proof: Minimax search ■
- Schaeffer solved checkers in 2007, after 18 years of computation: It's a tie!
- Checkers game tree has 10^{20} nodes; chess has 10^{40}; go has 10^{170}

ALPHA-BETA PRUNING

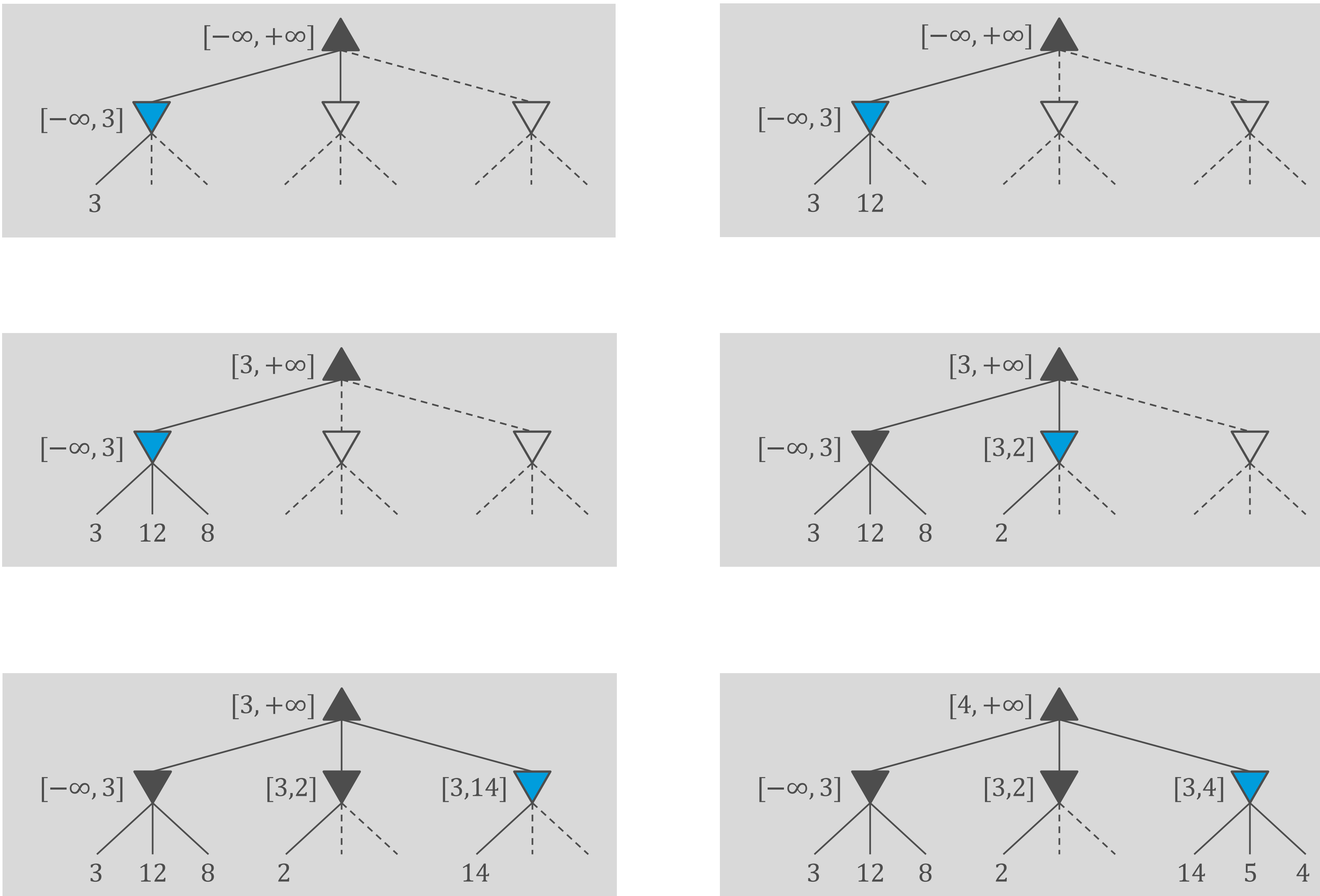
- Sometimes it's possible to prune large parts of the search tree that are irrelevant
- For a node n, if there's a better node at the same level (e.g., m) or at a different level (e.g., m') then optimal play wouldn't choose n



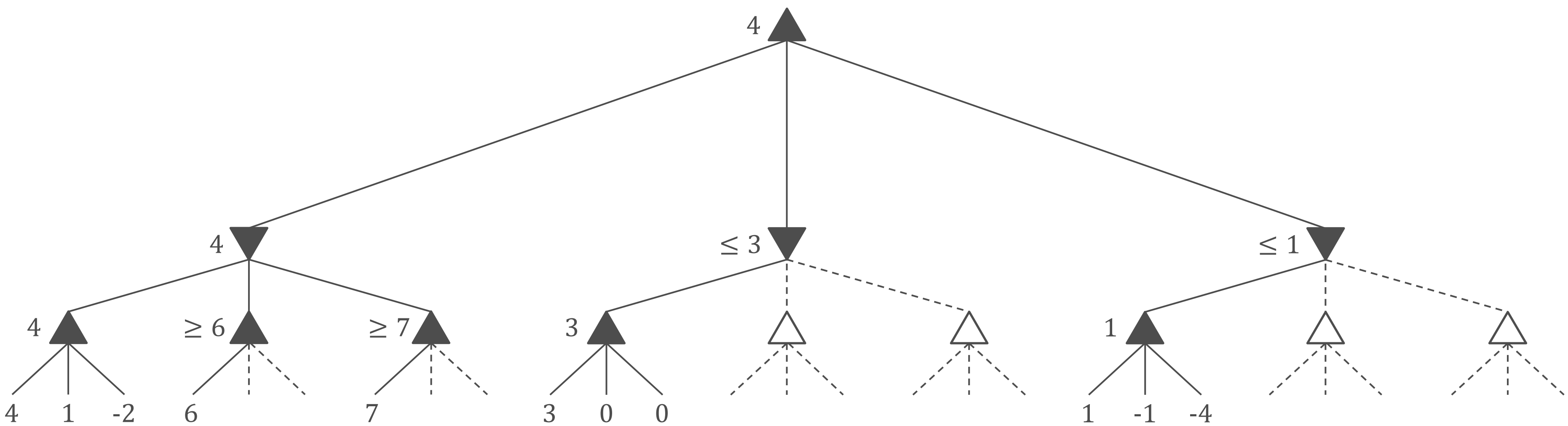
ALPHA-BETA PRUNING

```
function MaxEval (node n, numbers α,β)
        // max can guarantee ≥ α
        // min can guarantee ≤ β
        if n is a leaf then return PAYOFF(n)
        v ← α
        for all children n' of n
                v ← MAX(v, MINEVAL(n', v, β))
                if v ≥ β then return v
        return v

function MinEval (node n, numbers α,β)
        if n is a leaf then return PAYOFF(n)
        v ← β
        for all children n' of n
                v ← MIN(v, MAXEVAL(n', α, v))
                if v ≤ α then return v
        return v
```

ALPHA-BETA PRUNING: EXAMPLE



BEST-CASE EXAMPLE



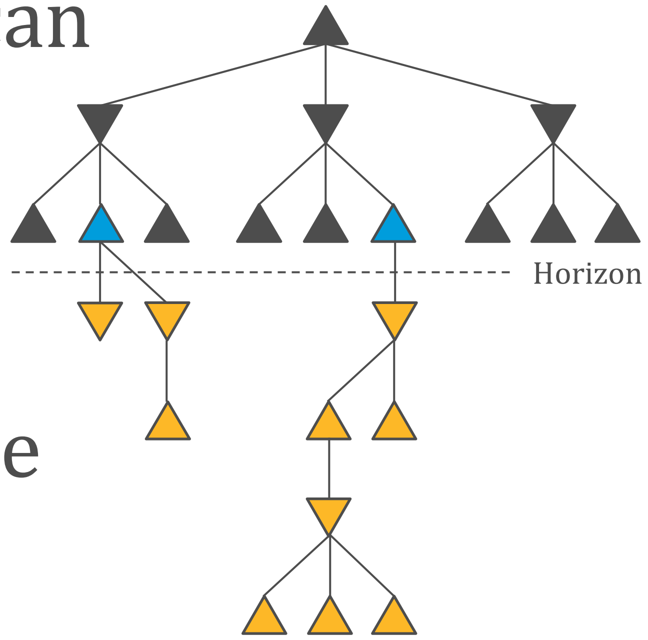
Optimal pruning of the game tree essentially requires solving the game!

WORST-CASE ANALYSIS

- Poll: If the game tree has height h and branching factor b, how many leaves would Alpha-Beta Pruning check in the worst case?
  - b
  - $h \cdot b$
  - $b^{\sqrt{h}}$
  - b^h
- In practice, effectiveness is highly dependent on the order in which states are examined
- For chess, simple ordering functions give performance close to the theoretical best case… which is still impractical

HEURISTIC ALPHA-BETA SEARCH

- To reduce computation we can cut off the search at horizon d and apply an evaluation function
- This requires a minor change to the pseudocode:




```
if n is a leaf then return Payoff(n)
```

$$\Downarrow$$

```
if Cutoff(n, d) then return Eval(n)
```

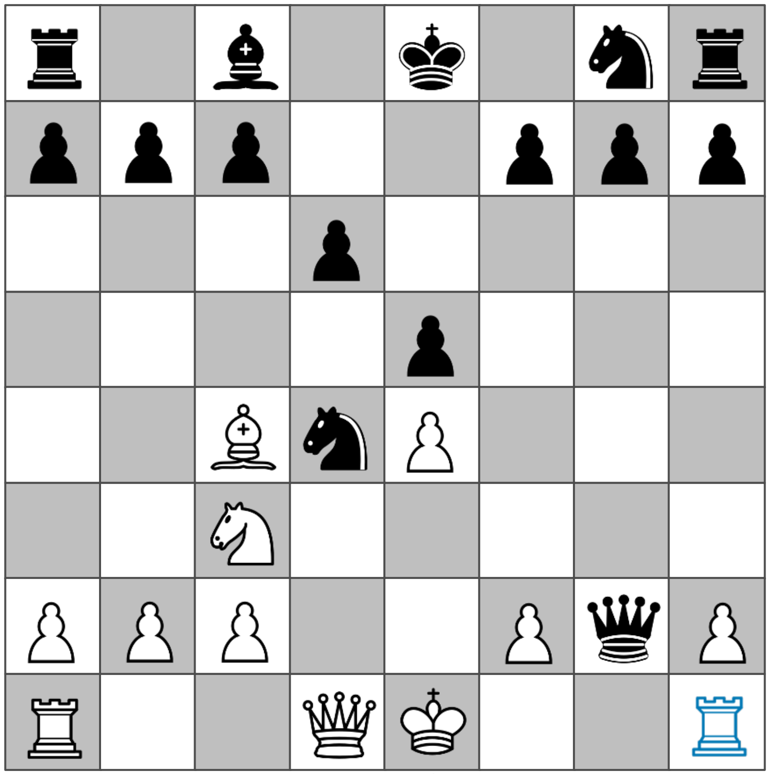
EVALUATION FUNCTIONS

- The evaluation function returns an estimate of the payoff at the given node, and should be easy to compute
- Evaluation functions typically employ features of the nodes, such as the number of pieces of each type
- The classic chess strategy of assigning a material value to each piece corresponds to a weighted linear evaluation function

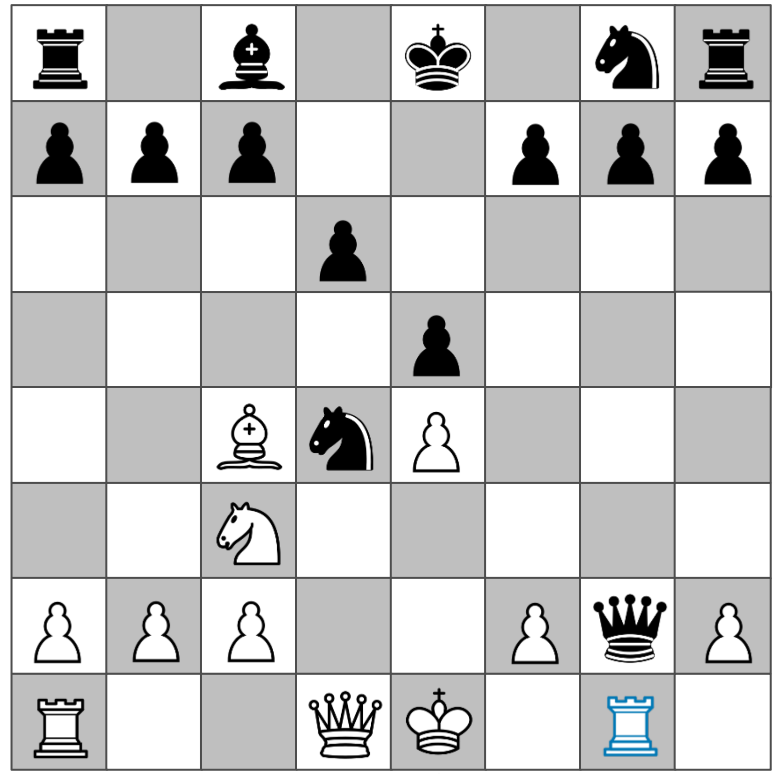
QUIESCENCE SEARCH

- In some nodes the evaluation function may be off because a pending move will make a big impact
- Quiescence search only cuts off the search in "quiescent states," otherwise CUTOFF is false

QUIESCENCE SEARCH: EXAMPLE

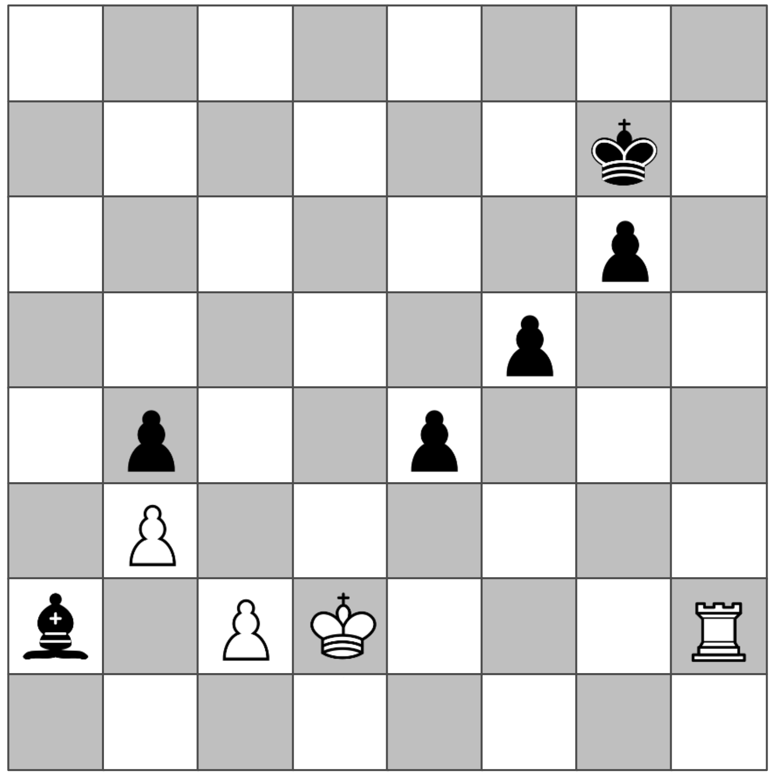


Black has a significant advantage of a knight and two pawns, so a black win is likely



White rook will capture the black queen, leading to a probable white win

THE HORIZON EFFECT



Black will surely lose the bishop but may push their loss beyond the search horizon by sacrificing pawns through repeated checks

MODERN AI CHESS

Modern chess engines train a neural network to configure an evaluation function

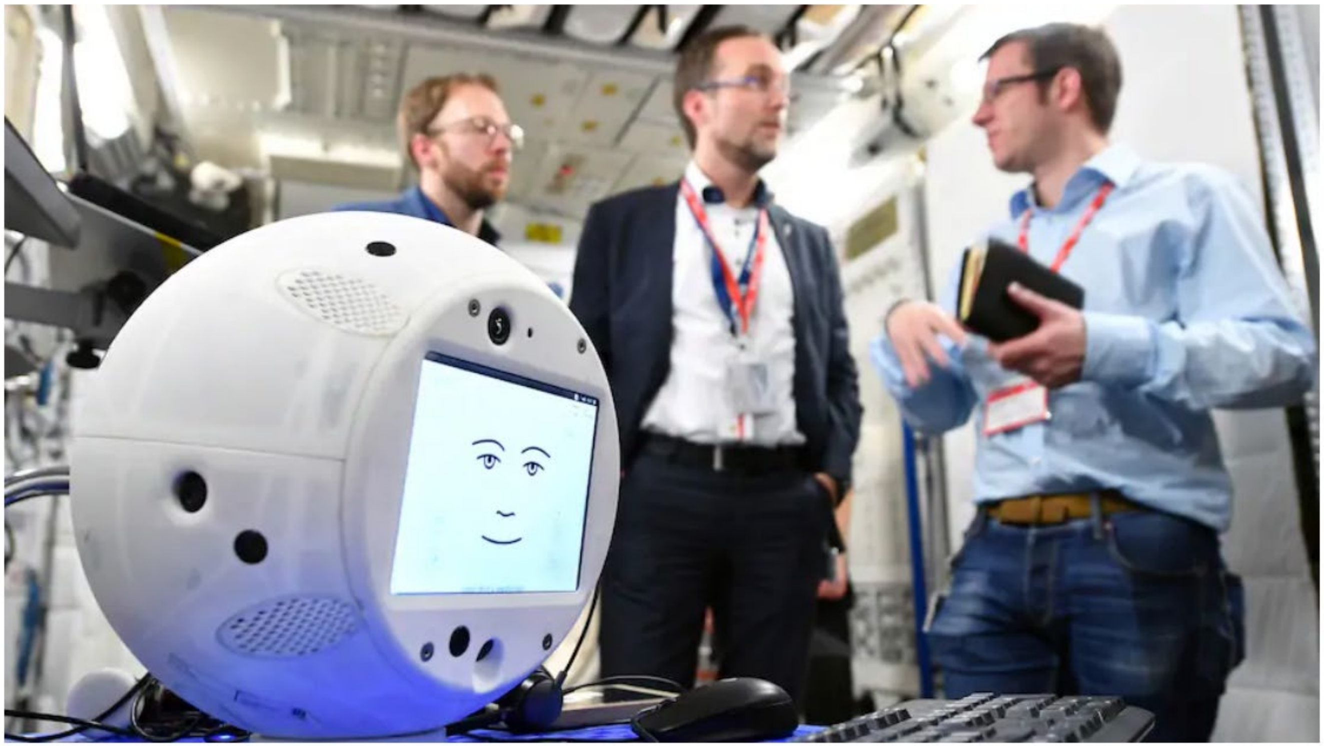
Stockfish uses Alpha-Beta Search, similarly to the techniques we discussed

Deepmind's AlphaZero uses Monte-Carlo Tree Search, which is a bit different

The Washington Post
Democracy Dies in Darkness

Opinions

It's time for AI to outgrow gaming



The Cimon (Crew Interactive MObile companioN) robot is shown during a communications test at the ESA European Astronaut Center in Cologne-Porz, Germany, on Jan. 30, 2018. (T. Bourry/AP)

Opinion by **Ariel Procaccia**

March 20, 2020 at 8:52 a.m. EDT