
The game had been tense, but friendly enough.
Then Steve tried to annex Portugal,
and the room became decidedly chillier.
ICANHASCHEEZBURGER.COM

Project 1

- Show expanded nodes?
  - isGoalState() can use visitedList to drawExpandedCells

Games

- Games vs. Game Theory?
- Difference from search?
- Use GAs?

Game Playing State-of-the-Art

- **Checkers:** Chinook ended 40-year-reign of human world champion Marion Tinsley in 1994. Used an endgame database defining perfect play for all positions involving 8 or fewer pieces on the board, a total of 443,748,401,247 positions. Checkers is now solved!

Game Playing State-of-the-Art

- **Checkers:** Chinook ended 40-year-reign of human world champion Marion Tinsley in 1994. Used an endgame database defining perfect play for all positions involving 8 or fewer pieces on the board, a total of 443,748,401,247 positions. Checkers is now solved!

- **Chess:** Deep Blue defeated human world champion Gary Kasparov in a six-game match in 1997. Deep Blue examined 200 million positions per second, used very sophisticated evaluation and undisclosed methods for extending some lines of search up to 40 ply. Current programs are even better, if less historic.

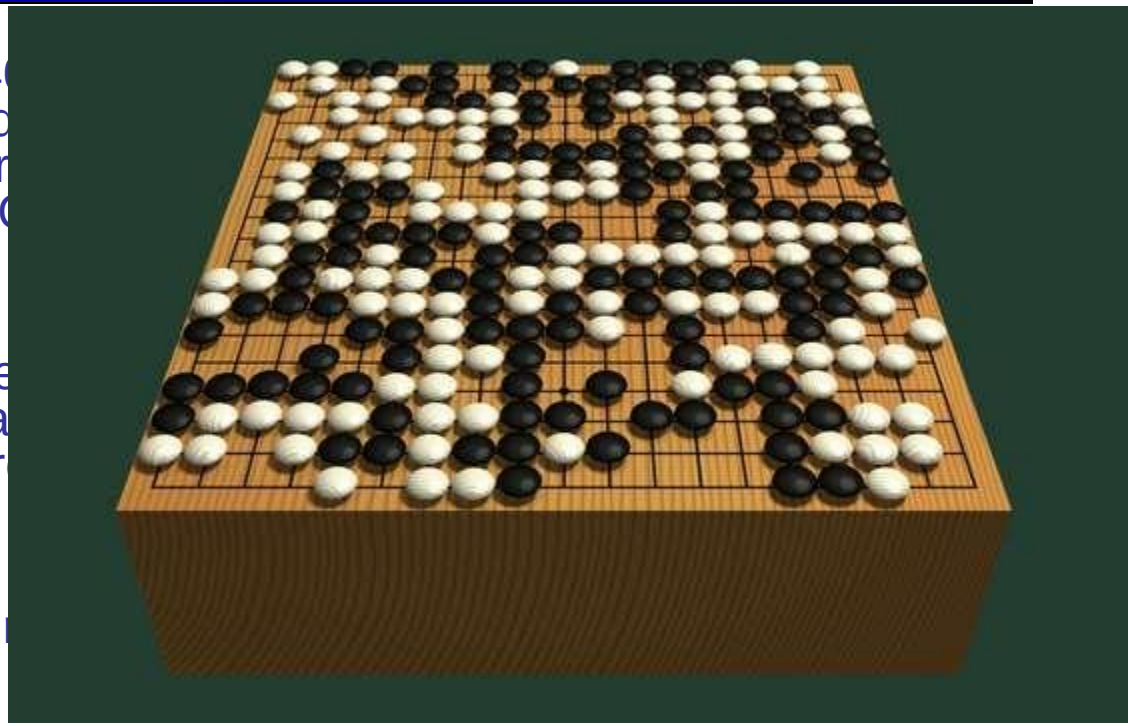
Game Playing State-of-the-Art

- **Checkers:** Chinook ended 40-year-reign of human world champion Marion Tinsley in 1994. Used an endgame database defining perfect play for all positions involving 8 or fewer pieces on the board, a total of 443,748,401,247 positions. Checkers is now solved!

- **Chess:** Deep Blue defeated human world champion Gary Kasparov in a six-game match in 1997. Deep Blue examined 200 million positions per second, used very sophisticated evaluation and undisclosed methods for extending some lines of search up to 40 ply. Current programs are even better, if less historic.

- **Othello:** Human champions refuse to compete against computers, which are too good.

Game Playing State-of-the-Art



- **Checkers:** Chinook ended 4
  Tinsley in 1994. Used an end
  positions involving 8 or fewer
  443,748,401,247 positions. C

- **Chess:** Deep Blue defeated
  six-game match in 1997. Dee
  second, used very sophistica
  extending some lines of sear
  better, if less historic.

- **Othello:** Human champions
  are too good.

- **Go:** Human champions are just beginning to be challenged by machines, though the best humans still beat the best machines. In go, b > 300, so most programs use pattern knowledge bases to suggest plausible moves, along with aggressive pruning.

Game Playing State-of-the-Art

- **Checkers:** Chinook ended 40-
  Tinsley in 1994. Used an endga
  positions involving 8 or fewer p
  443,748,401,247 positions. Ch

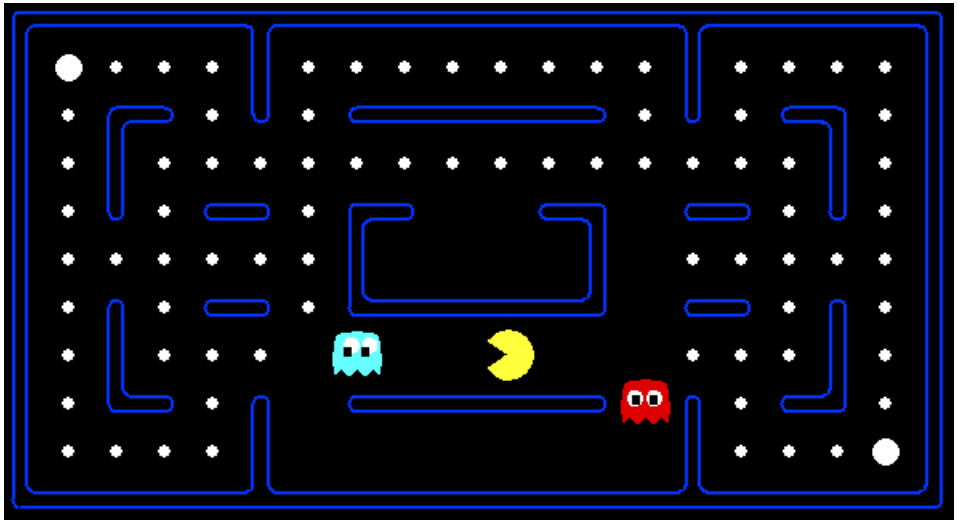
- **Chess:** Deep Blue defeated hu
  six-game match in 1997. Deep
  second, used very sophisticate
  extending some lines of search
  better, if less historic.

- **Othello:** Human champions refuse to compete against computers, which are too good.

- **Go:** Human champions are just beginning to be challenged by machines, though the best humans still beat the best machines. In go, b > 300, so most programs use pattern knowledge bases to suggest plausible moves, along with aggressive pruning.

- **Jeapordy**: Watson
- **Pacman**: ?

Adversarial Search?



Game Playing

- Many different kinds of games!
- Axes:
  - Deterministic or stochastic?
  - One, two, or more players?
  - Perfect information (can you see the state)?
- Want algorithms for calculating a strategy (policy) which recommends a move in each state

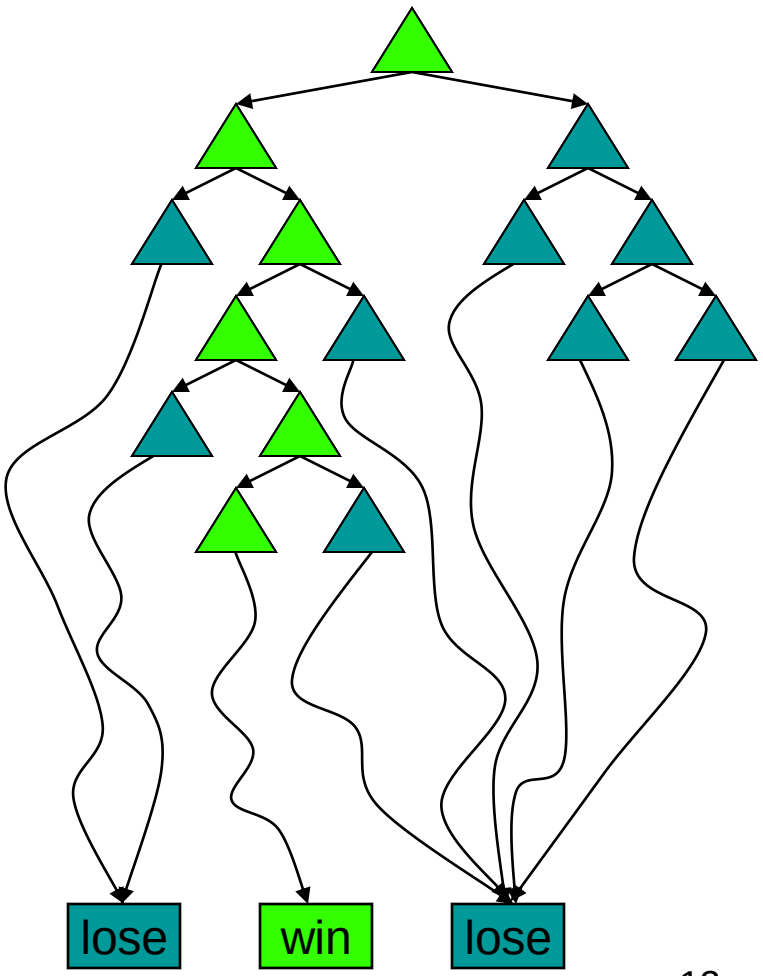
Deterministic Games

- Many possible formalizations, one is:
  - States: S (start at s_0)
  - Players: P={1...N} (usually take turns)
  - Actions: A (may depend on player / state)
  - Transition Function: SxA $\rightarrow$ S
  - Terminal Test: S $\rightarrow$ {t,f}
  - Terminal Utilities: SxP $\rightarrow$ R

- Solution for a player is a policy: S $\rightarrow$ A

Deterministic Single-Player?

- Deterministic, single player, perfect information:
  - Know the rules
  - Know what actions do
  - Know when you win
  - E.g. Freecell, 8-Puzzle, Rubik's cube
- … it's just search!
- Slight reinterpretation:
  - Each node stores a value: the best outcome it can reach
  - This is the maximal outcome of its children (the max value)
  - Note that we don't have path sums as before (utilities at end)
- After search, can pick move that leads to best node

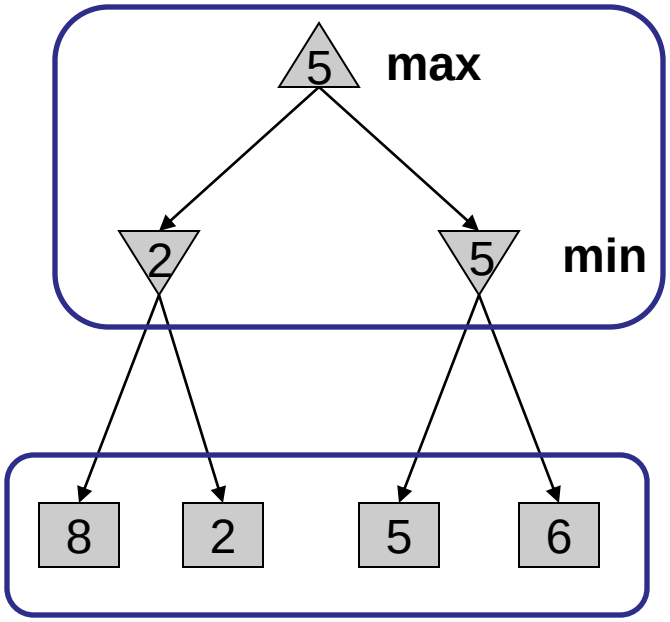


Adversarial Games

- Deterministic, zero-sum games:
  - Tic-tac-toe, chess, checkers
  - One player maximizes result
  - The other minimizes result

- Minimax search:
  - A state-space search tree
  - Players alternate turns
  - Each node has a minimax value: best achievable utility against a rational adversary

Minimax values: computed recursively



Terminal values: part of the game

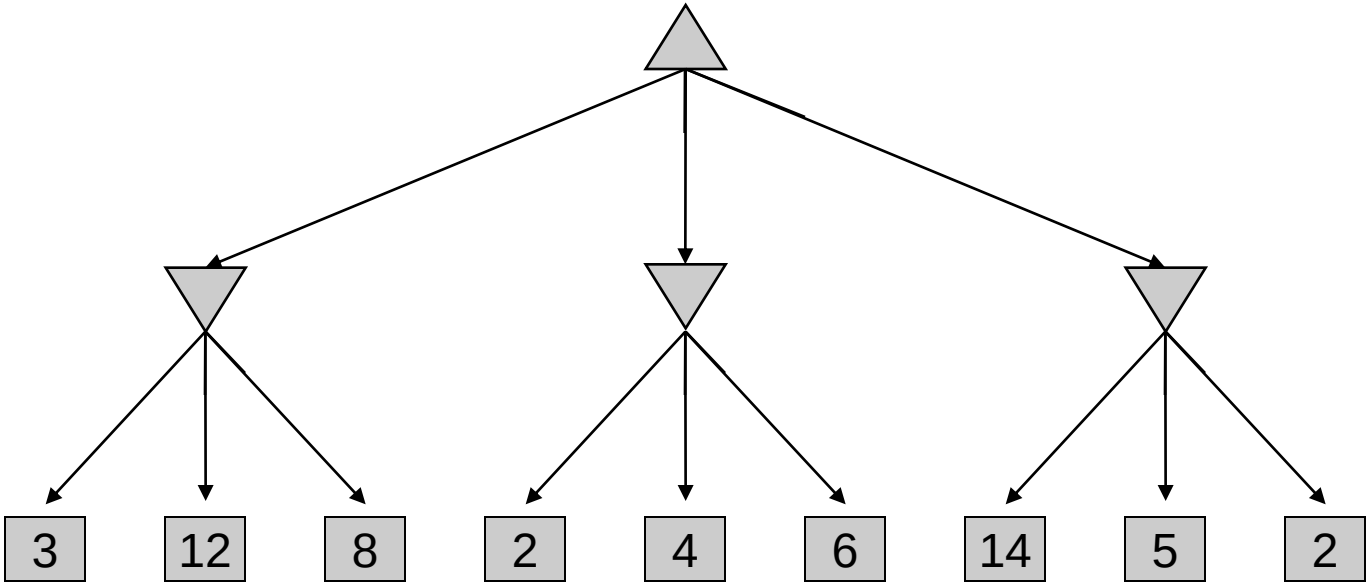
Computing Minimax Values

- Two recursive functions:
  - max-value maxes the values of successors
  - min-value mins the values of successors

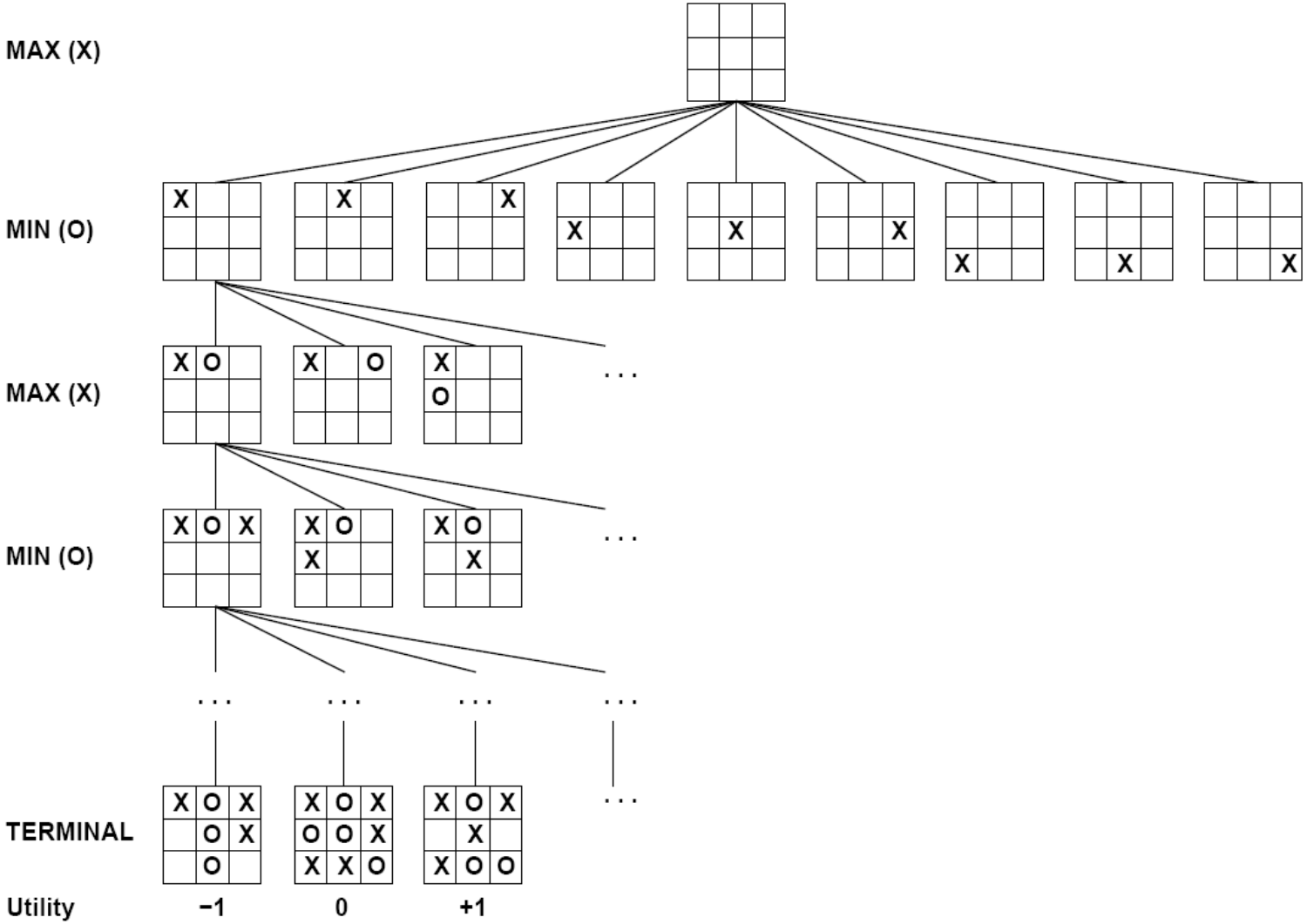
```
def value(state):
    If the state is a terminal state: return the state's utility
    If the next agent is MAX: return max-value(state)
    If the next agent is MIN: return min-value(state)

def max-value(state):
    Initialize max = -∞
    For each successor of state:
        Compute value(successor)
        Update max accordingly
    Return max
```

Minimax Example



Tic-tac-toe Game Tree



Minimax Properties

- Optimal against a perfect player. Otherwise?
- Time complexity?
  - $O(b^m)$
- Space complexity?
  - $O(bm)$
- For chess, $b \approx 35$, $m \approx 100$
  - Exact solution is completely infeasible
  - But, do we need to explore the whole tree?