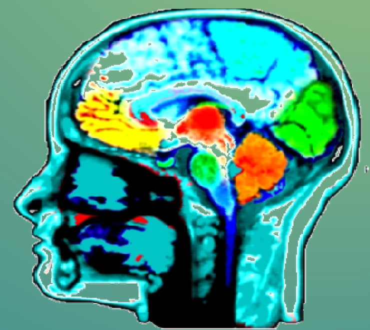
Introduction
To
Artificial Intelligence

Isfahan University of Technology (IUT)
1402



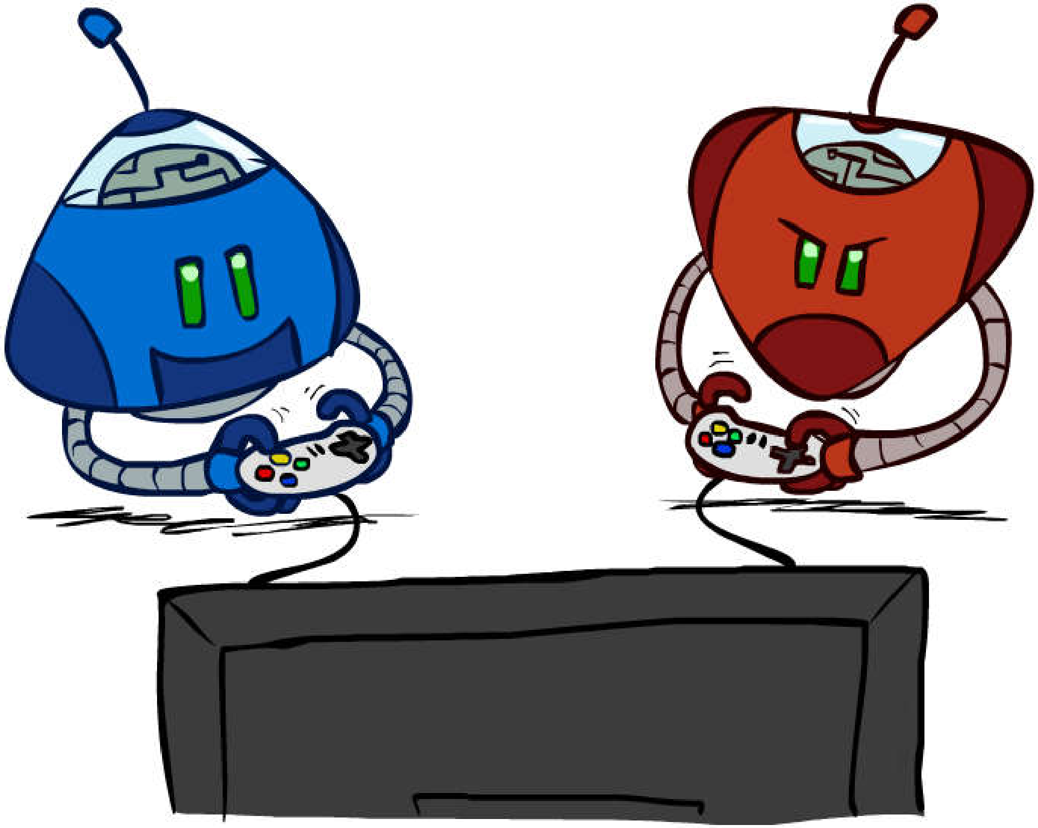
Adversarial Search

Dr. Hamidreza Hakim
hakim@iut.ac.ir

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.]

بسم الله الرحمن الرحیم

Adversarial Search



[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]

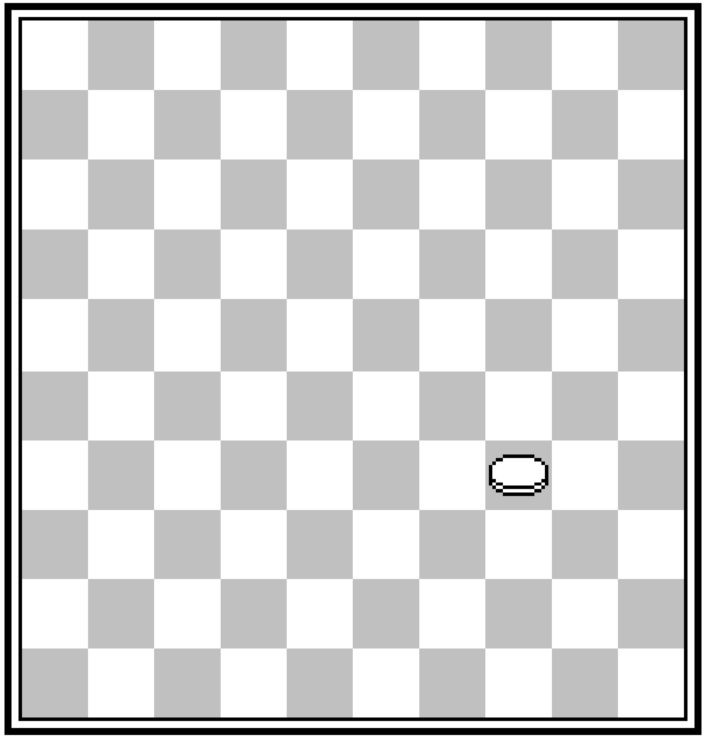
Game Playing State-of-the-Art

- **Checkers:** 1950: First computer player.

(Logical inference based)

1994: First computer champion: Chinook ended 40-year-reign of human champion Marion Tinsley using complete 8-piece endgame.

2007: Checkers solved! (Win strategy found)



Gain des Blancs.
S.R. 2003

Game Playing State-of-the-Art

- **Checkers:** 1950: First computer player.

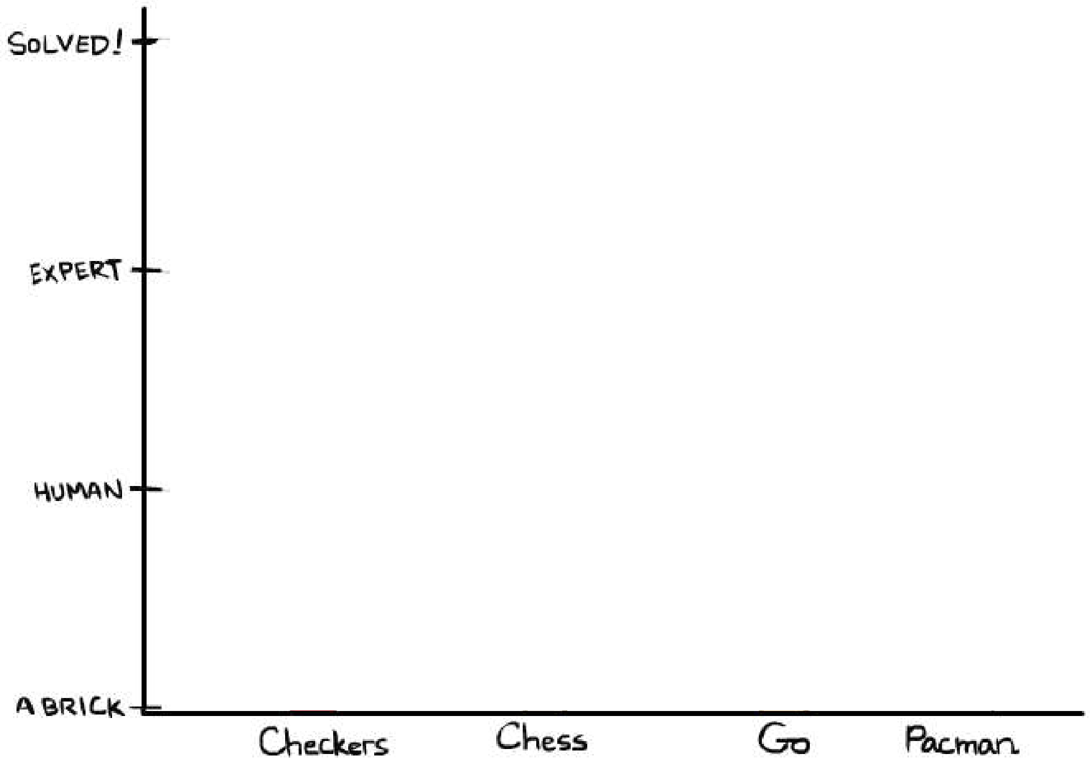
(Logical inference based)

1994: First computer champion: CHINOOK ended 40-year-reign of human champion Marion Tinsley using complete 8-piece endgame.

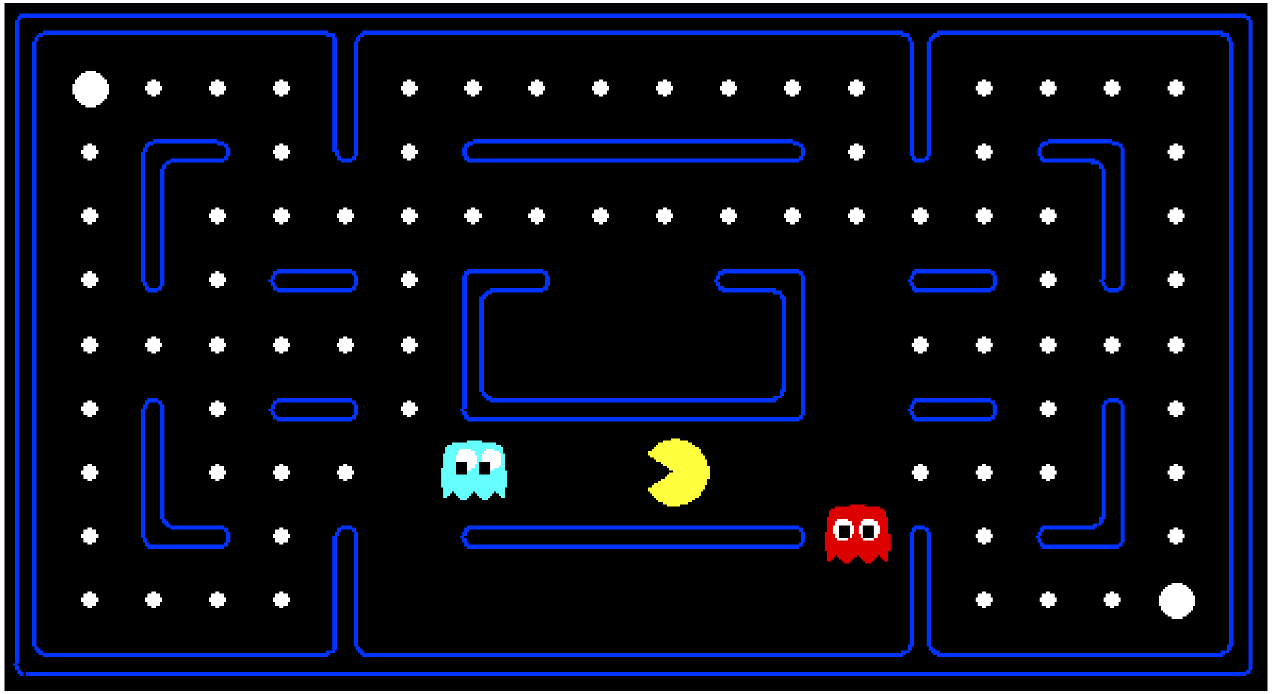
2007: Checkers solved! (Win strategy found)

- **Chess:** 1997: Deep Blue defeats human champion GARY KASPAROV in a six-game match. Deep Blue examined 200M 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.
- More branching factor

- **Go:** Human champions are now starting to be challenged by machines, though the best humans still beat the best machines. In go, branchfacotr > 300! Classic programs use pattern knowledge bases, but big recent advances use Monte Carlo (randomized) expansion methods.
- **Pacman**

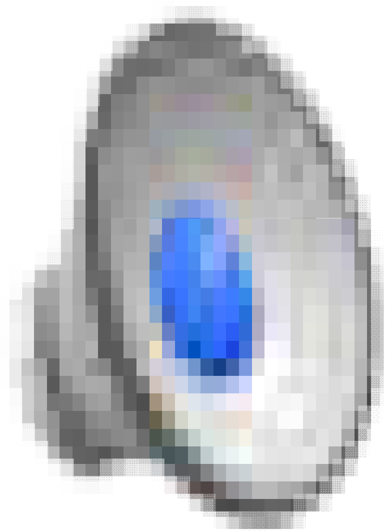


Behavior from Computation

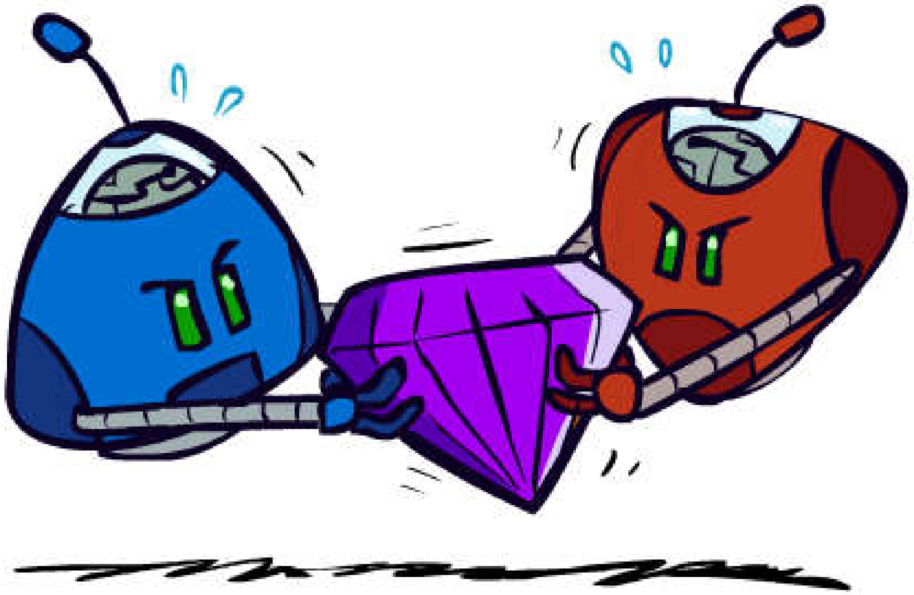


[Demo: mystery pacman (L6D1)]

Video of Demo Mystery Pacman



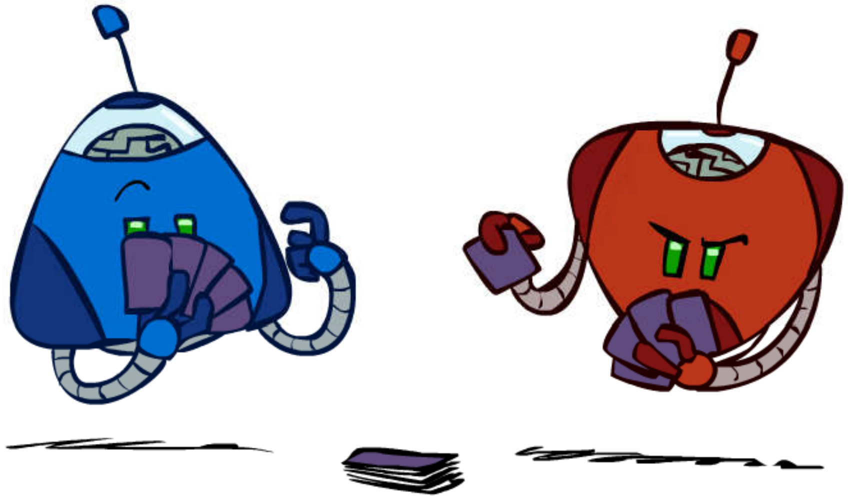
Adversarial Games



Types of Games

- Many different kinds of games!

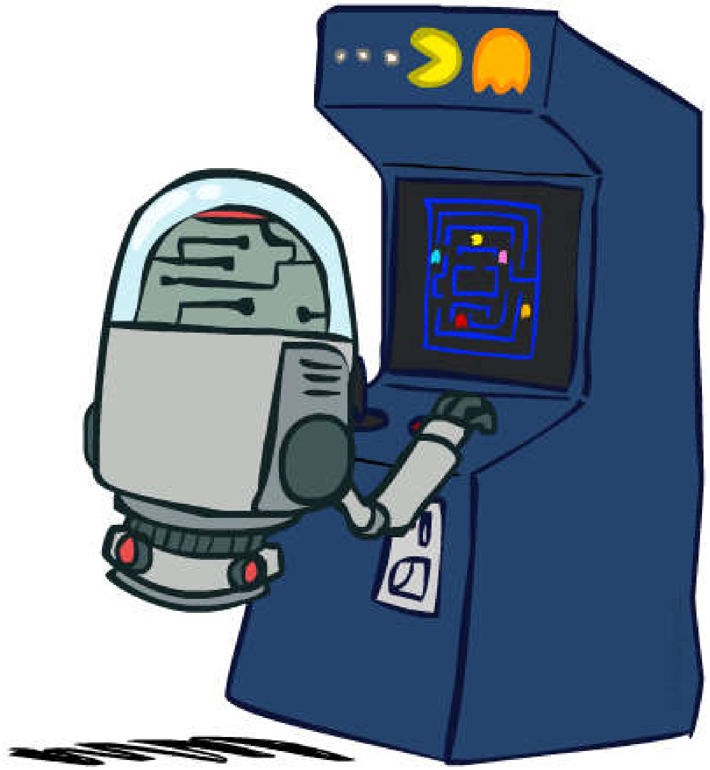
- Axes:
  - Deterministic or stochastic?
  - One, two, or more players?
  - Zero sum?
  - Perfect information (can you see the state)?



- Want algorithms for calculating a strategy (policy) which recommends a move from each state

Deterministic Games

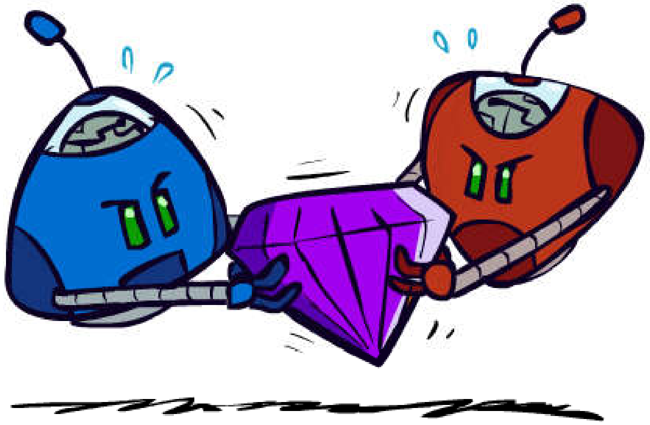
- Many possible formalizations, one is:
  - States: S (start at s_0)
  - Players: P={1...N} (usually take turns)
    - here N=2
  - Actions: A (may depend on player / state)
  - Transition Function: $S \times A \rightarrow S$
  - Terminal Test: $S \rightarrow$ {t,f(finish game)}
  - Terminal Utilities: $S \times P \rightarrow R$ (utility)
    - Bigger R means better Game
    - Utility define for terminal states.
- Solution(goal) is not Path (why?)
- Solution for a player is a policy: $S \rightarrow A$
  - Find policy to win game in every game

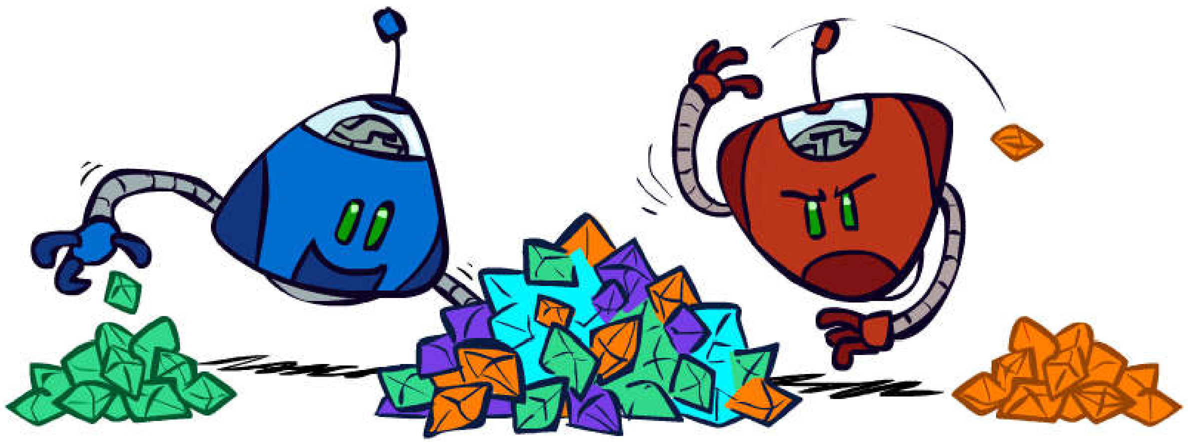


Game vs. Search Problem

- "Unpredictable" opponent ⇒ solution is a strategy specifying a action(move) for every possible state/opponent reply(actions)
- Time limits ⇒ unlikely to find goal, must approximate

Zero-Sum Games





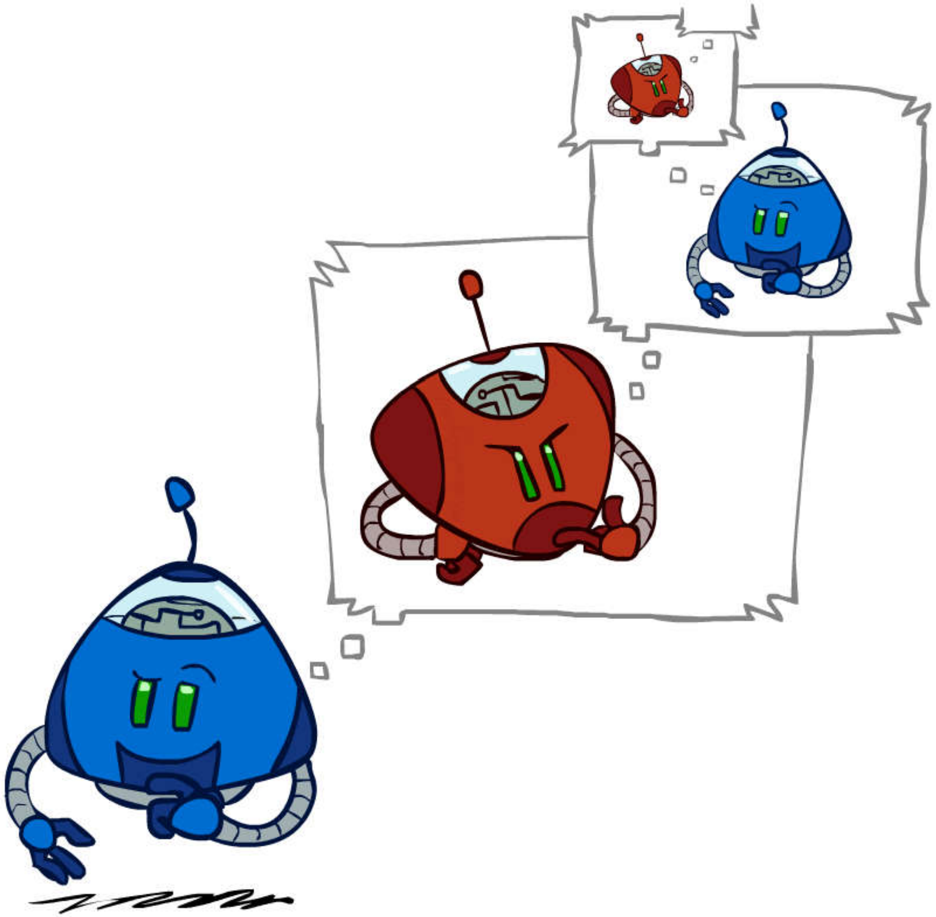
1.Zero-Sum Games

- Same utility for opposite
- Agents have opposite utilities (values on outcomes)
- Lets us think of a single value that one maximizes and the other minimizes
- Adversarial, pure competition

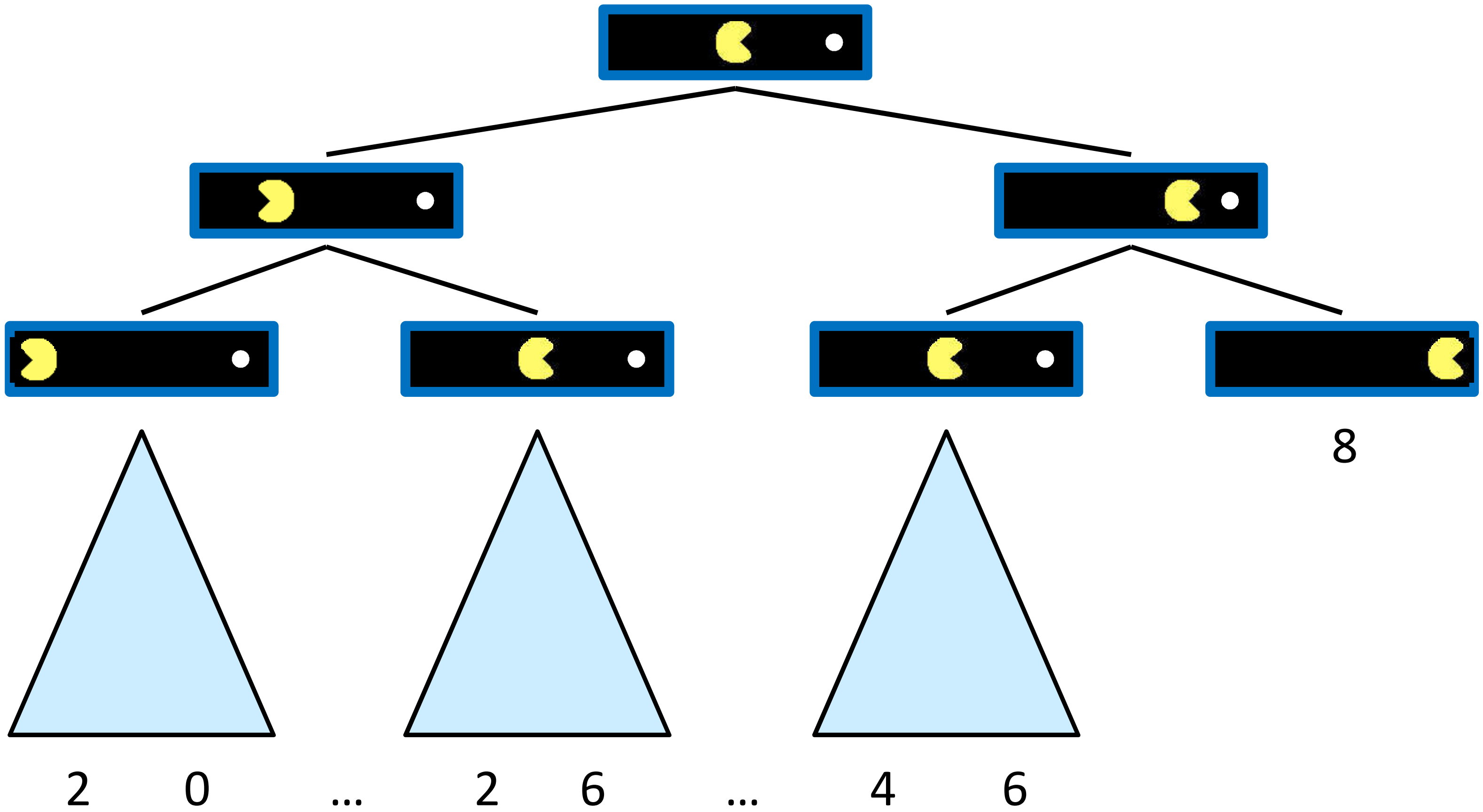
2. General Games

- Agents have independent utilities (values on outcomes)
- Cooperation, indifference, competition, and more are all possible
- More later on non-zero-sum games

Adversarial Search



Single-Agent Trees

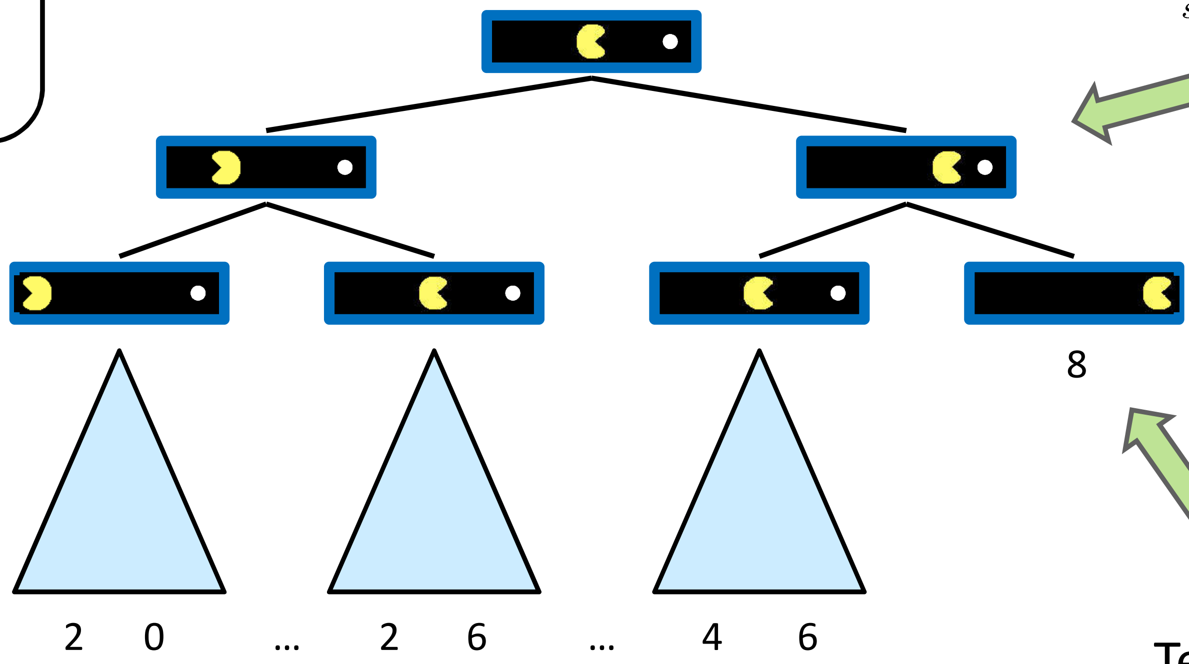


Value of a State

Value of a state: The best achievable(future) outcome (utility) from that state (Recursive)

Non-Terminal States:

$$V(s) = \max_{s' \in \text{children}(s)} V(s')$$



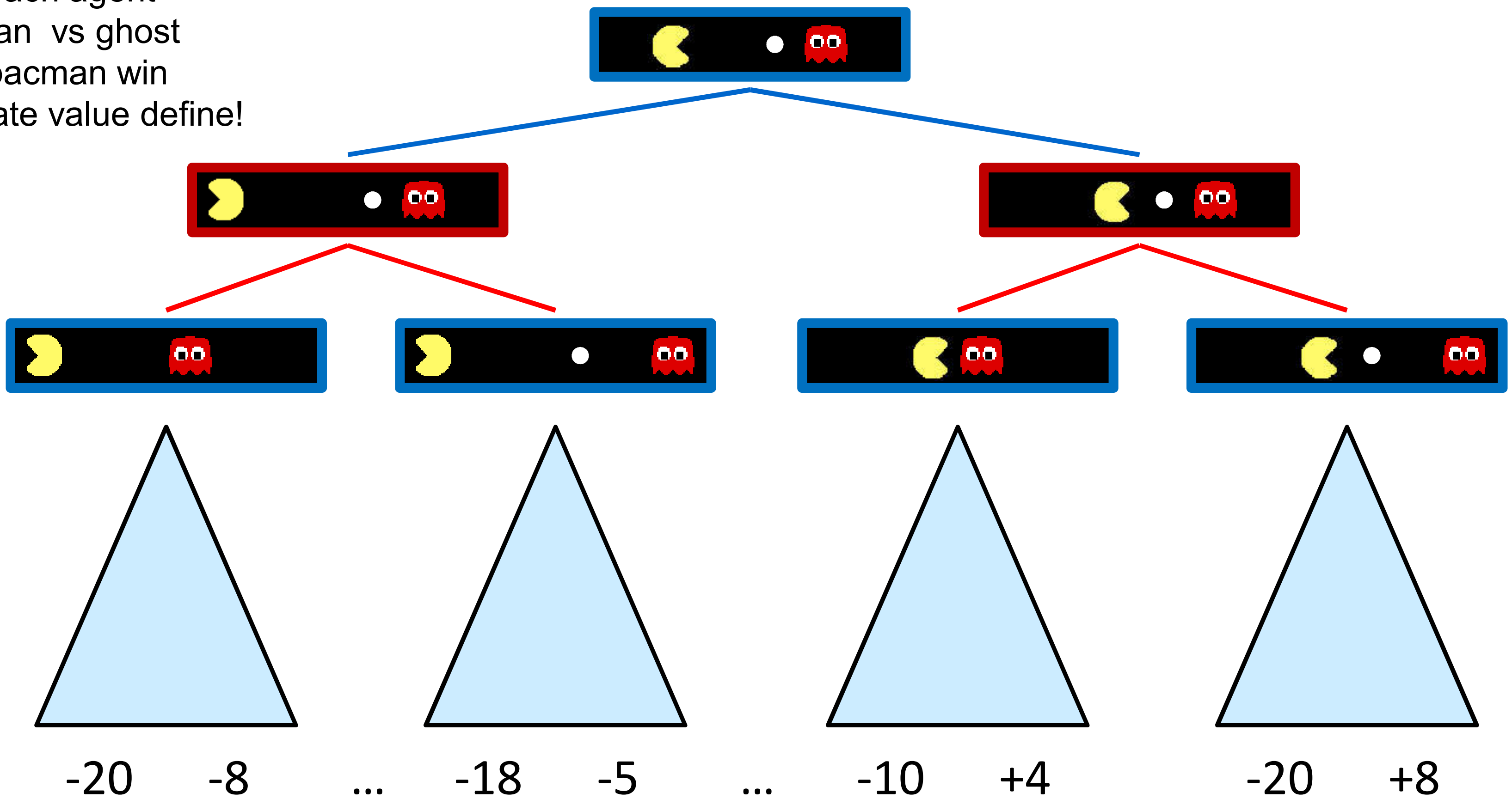
Terminal States:

$$V(s) = \text{known}$$

policy: S $\rightarrow$ A best action == best frontier State value

Adversarial Game Trees

- Each level for each agent
  pacman vs ghost
- Positive utility pacman win
- Ambiguity in state value define!



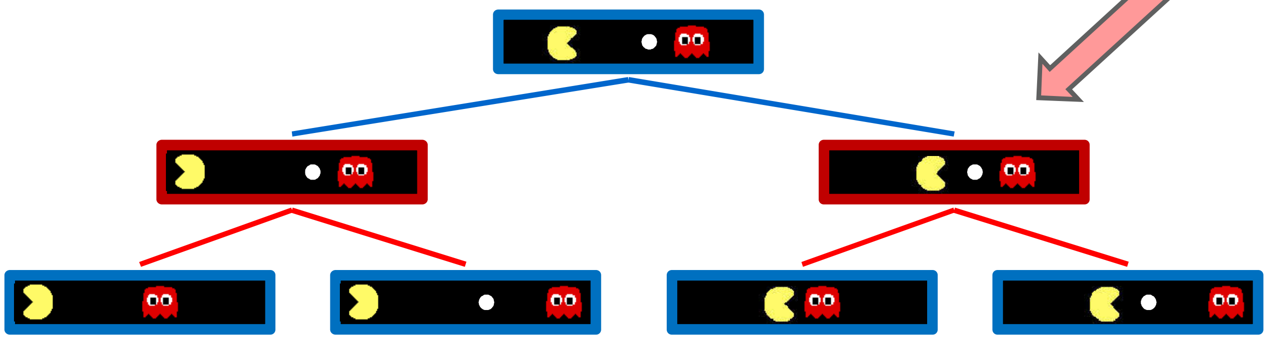
Minimax Values

States Under Agent's Control:

$$V(s) = \max_{s' \in \text{successors}(s)} V(s')$$

States Under Opponent's Control:

$$V(s') = \min_{s \in \text{successors}(s')} V(s)$$



-8 -5 -10 +8

Terminal States:

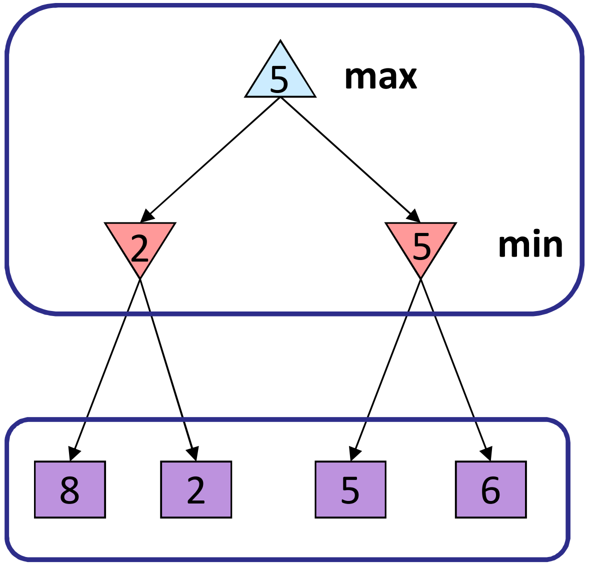
$$V(s) = \text{known}$$

Adversarial Search (Minimax)

- 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
  - Compute each node's minimax value: the best achievable utility against a rational (optimal) adversary

Minimax values: computed recursively

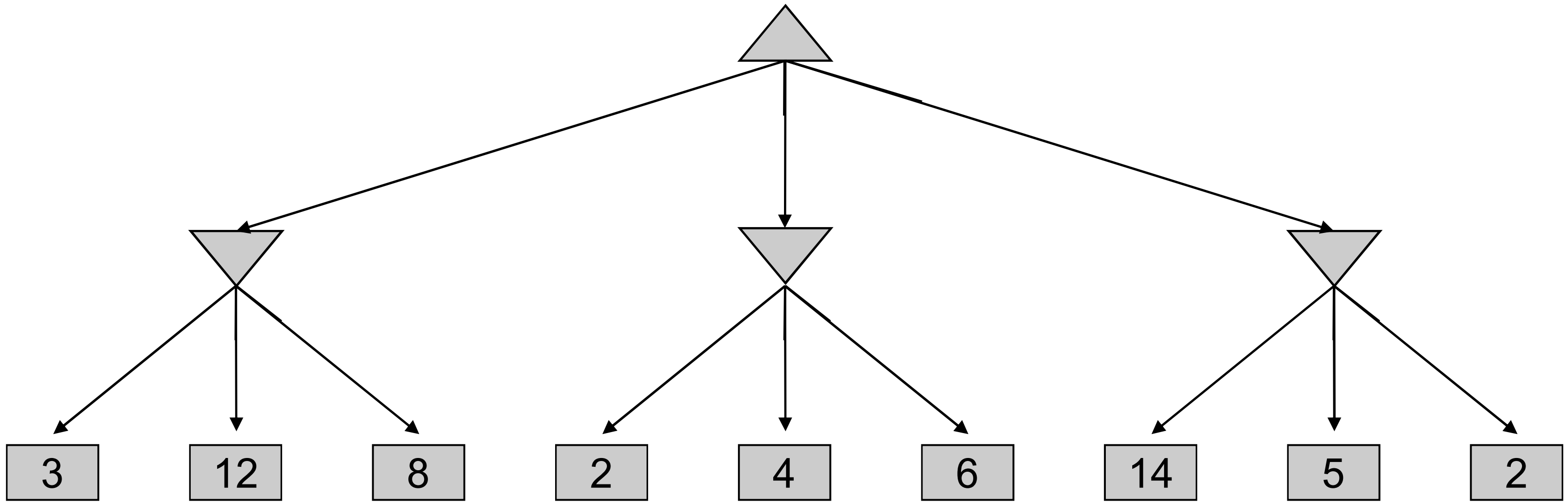


Terminal values: part of the game

MinMax Strategy

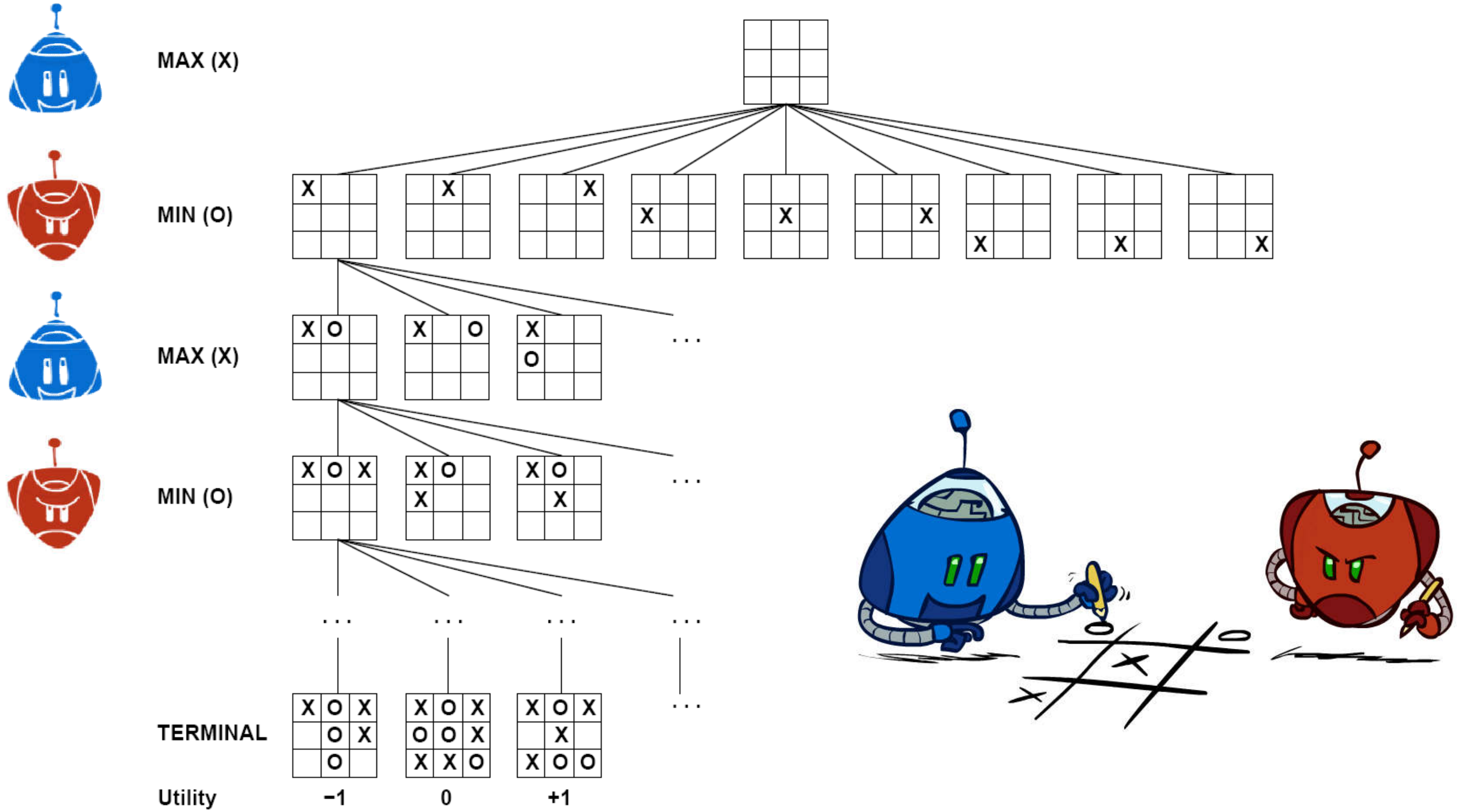
- Perfect play for deterministic, perfect-information games
- Idea: choose move to position with highest minimax value = best achievable payoff against best play

Minimax Example



Best value solution obtain Again Best opponent actions(Min Select)
What happen if opponent Select non minimum action?

Tic-Tac-Toe Game Tree



What is Max value?