Multiagent Environments

Cooperative: car Competitive: chess

Adversarial Search Problems ≡ Games-competitive -2 agents -deterministic, fully observable

-zero sum game: utility=0, one gains one loses

\*strategy / policy / set of rules (not sequence of actions)

diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Games are modeled as:

1. search problems

2. heuristic evaluation function

**Zero-Sum Games**

▪ Adversarial Games that involve pure competition

Utility function – one maximizes, one minimizes

Ply – each move by a player

\*embedded thinking / backward reasoning

▪ initial state - 𝑺𝒐 - model of the state of the environment that agent starts from

▪ Player(s): Whose turn?

▪ Actions(s): returns the set of legal moves in state s

▪ Result(s, a): Transition model ( 𝑺 × 𝑨 → 𝑺 ) which defines the result of a move

▪ Terminal test: Is terminal? Game over? - States where game ends = terminal states

▪ Utility(s, p): utility function or objective function for a game that ends in terminal state s for player p

- In Chess, the outcome is a win, loss, or draw with values +1, 0, ½

- For tic-tac-toe we can use a utility of +1, -1, 0, if it's a win, loss, or draw, respectively

**Minimax Algorithm**

-searchs all the leaves

principle: compute the utility of being in a state, assuming both players play optimally from there until the end of the game

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

terminal nodeların hepsini explore etmeye gerek yok

𝜶 − 𝜷 pruning

𝜶 : largest value for MAX across seen children (current lower bound on MAX’s outcome)

𝜷 : lowest value for MIN across seen children (current upper bound on MIN’s outcome)

▪ Initialization: 𝜶 = −∞, 𝜷 = ∞

▪ Propagation: Send 𝜶,𝜷 values down during the search to be used for pruning

▪ Update 𝜶,𝜷 values by propagating upwards values of terminal nodes

▪ Update 𝜶 only at MAX nodes and update 𝜷 only at MIN nodes

▪ Pruning: Prune any remaining branches whenever 𝜶 ≥ �

Move Ordering

▪ Worst ordering: no pruning happens (best moves are on the right of the game tree)

▪ Ideal ordering: lots of pruning happens (best moves are on the left of the game tree)

**Stochastic Games (chance factor)**

Fully observble -monopolly

Chance nodes

**Expectiminimax algoirthm**