Carl Stevenson

Unit 5 notes

* Game Playing
  + Features:
  + Deterministic vs stochastic
    - stochastic = random
    - deterministic = no random elements
  + adversarial vs cooperative
    - adversarial = opponents
    - cooperative = working together against the game
  + perfect vs partial information
    - perfect = see all the game components
    - partial = opposite (stratego)
  + number of players
  + interaction scheduling
    - turn based
    - simultaneous
* As state spaces
  + State variables describe game features
  + Start states define initial setup
  + States are legal game positions
  + Moves are transitions between states
  + Goal determines terminal states
  + If there are two players, it is a bi-partite state space
* Perfect information
  + Players know everything there is to know
    - No hidden information
    - No random events
  + Players need not have the same set of moves
* Contingency Problem
  + Opponent has many possible moves
  + Our solution must cover them
  + Opponents behavior introduces uncertainty
    - Has a different set of possible moves than you (tic-tac-toe)
  + Assume a rational opponent
    - Takes best move (maximises their utility function)
* Minimax algorithm (for games)
  + Have a state space
    - For a tree, nodes represent states
    - For the board positions, have transitions that lead to the other states
  + Ply decision making
    - Looking ahead to choose based on overall outcome, instead of immediate
    - Evaluate each sub-tree to promote the values up.
  + Assumes both players play perfectly
  + Min wants lowest possible score, or chooses the move which will minimize the score resulting from Max choosing the maximizing move.
  + Max does the reverse, chooses the move that maximizes the score given Min's minimizing move
  + Procedure:
    - Statically evaluate states at depth d
    - Work upwards
      * At Max level, choose maximum of child node values
      * At Min level, choose minimum of child node values
    - Can code depth first, so space efficient
  + Minimax evaluation shows the futility of the Token Game for red
* Evaluation problems
  + Tic Tac Toe – 9! states
  + Checkers- 10^40 states
  + Chess – 10^120 states
  + Go- 361! states
* Minimax search
  + In interesting games, state space is large
    - Too large to reach terminal states
  + Use heuristic evaluation of partial paths
  + Use estimated cost of internal nodes
  + Deeper search gets closer to terminal states
  + Also, problem is often online search
  + Does extra work
* Alpha-Beta pruning
  + Start propagating costs as soon as a leaf node is generates
  + Don't explore nodes which we know are no better than the best found so far
  + Alpha-Beta Values
    - Alpha value- associated with Max, best score found, never decreases
    - Beta value – associated with Min, best score found, never increases
    - Each used to prune tree

Little more than 2/3