

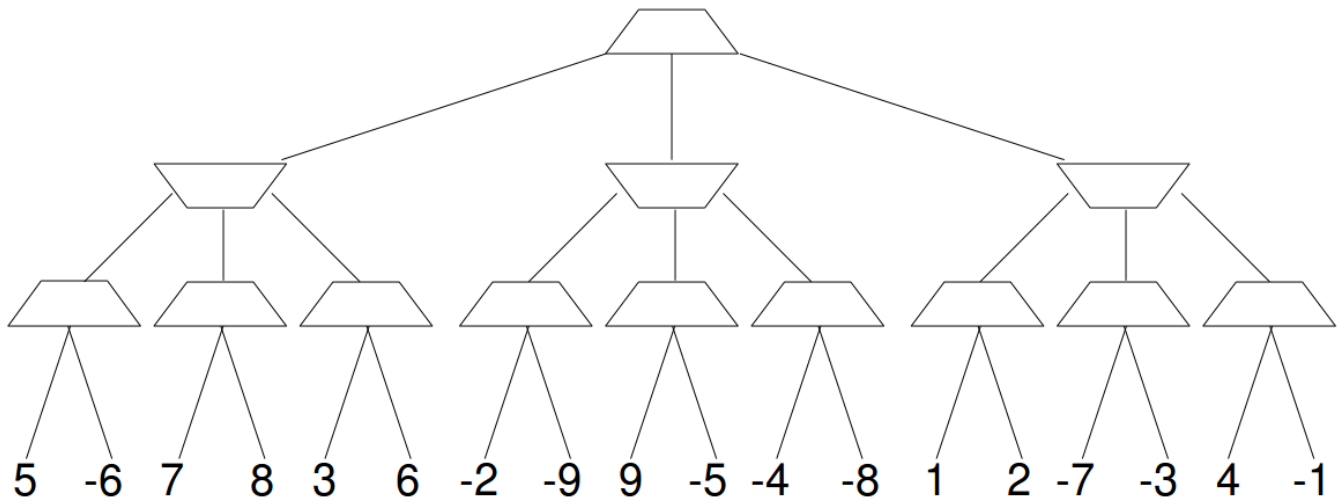
CS188 Spring 2012 Section 4: Games

1 Minimax Search

In this problem, we will explore adversarial search.

Consider the zero-sum game tree shown below. Trapezoids that point up, such as at the root, represent choices for the player seeking to maximize; trapezoids that point down represent choices for the minimizer. Outcome values for the maximizing player are listed for each leaf node. It is your move, and you seek to maximize the expected value of the game.

(a) Assuming both opponents act optimally, carry out the minimax search algorithm. Write the value of each node inside the corresponding trapezoid. What move should you make now? How much is the game worth to you?



(b) Now reconsider the same game tree, but use α - β pruning (the tree is printed on the next page). Expand successors from left to right. In the brackets $[,]$, record the $[\alpha, \beta]$ pair that is passed down that edge (through a call to MIN-VALUE or MAX-VALUE). In the parentheses $()$, record the value (v) that is passed up the edge (the value returned by MIN-VALUE or MAX-VALUE). **Circle all leaf nodes that are visited. Put an 'X' through edges that are pruned off.** How much is the game worth according to α - β pruning?

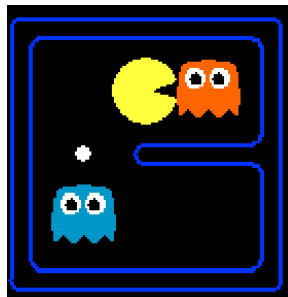
2 Suicidal Pacman

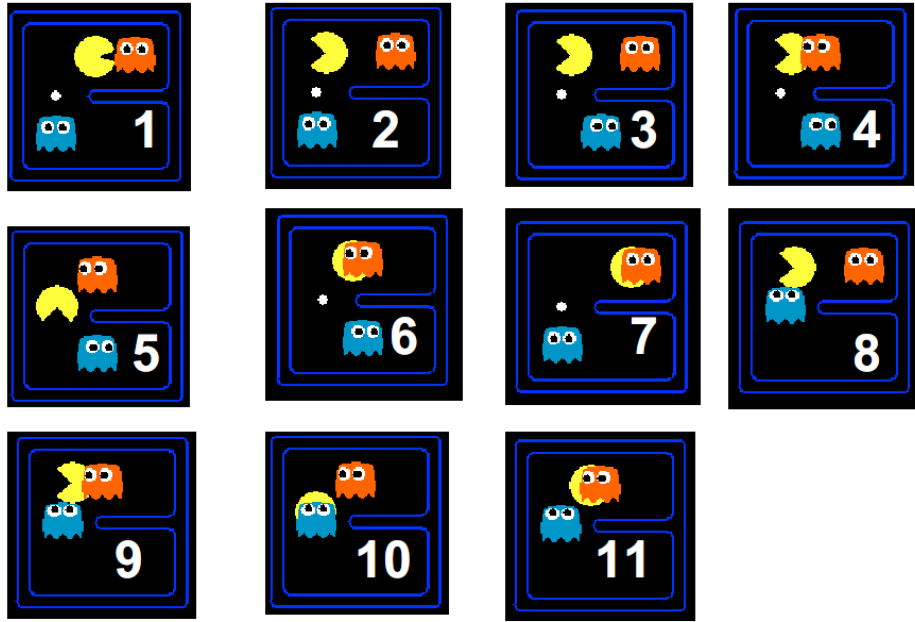
Pacman is sometimes suicidal when doing a minimax search because of its worst case analysis. We will build here a small expectimax tree to see the difference in behavior.

Consider the following rules:

- Ghosts cannot change direction unless they are facing a wall. The possible actions are east, west, south, and north (not stop). Initially, they have no direction and can move to any adjacent square.
- We use random ghosts which choose uniformly between all their legal moves.
- Assume that Pacman cannot stop
- If Pacman runs into a space with a ghost, it dies before having the chance to eat any food which was there.
- The game is scored as follows:
 - -1 for each action Pacman takes
 - 10 for each food dot eaten
 - -500 for losing (if Pacman is eaten)
 - 500 for winning (all food dots eaten)

Given the following “trapped” maze, build the expectimax tree with max and chance nodes clearly identified. Use the game score as the evaluation function at the leaves. If you don’t want to make little drawings, all possible states of the game have been labeled for you on the next page: use them to identify the states of the game. Pacman moves first, followed by the lower left ghost, then the top right ghost.





(a) Build the expectimax tree. What is Pacman's optimal move?

(b) What would pacman do if it was using minimax instead?

(c) By changing the probabilities of action for the ghosts, can you get expectimax to make the same decision as minimax?

(d) Now say you are using the following alternate game score components:

- -1 for Pacman making a move
- -1.5 for losing
- 0 for eating food
- 0.3 for winning

Use this new game score as your evaluation function at the leaves. Note this yields a monotonic transformation of the original utilities: a function which preserves the ordering of the state according to their utility. Could this change the decision of Pacman using expectimax?