Question 1 (Class) – Expectiminimax – Saving Suicidal Pacman!

You will see during your assignment that pacman is sometimes suicidal when doing a minimax search because of its worst-case analysis. We will build here a small expectimax tree (there is no min node, hence the name) to see the difference in behavior.

Consider the following rules for pacman (a bit more restrictive than in Project 2):

- Ghosts cannot reverse 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.
- For here (to reduce the branching factor), assume that pacman cannot stop.
- As a convention here, if pacman runs into a space with a ghost, it dies before having the chance of eating any food which was there (in the current code implementation, pacman eats food before getting eaten by a ghost, and so can win this way).
- Assume we are using random ghosts which choose uniformly between all their legal moves.
- The score of the game is: -1 for each action that pacman takes, -500 when pacman is eaten by a ghost, 500 when pacman wins, and 10 for each food dot eaten.

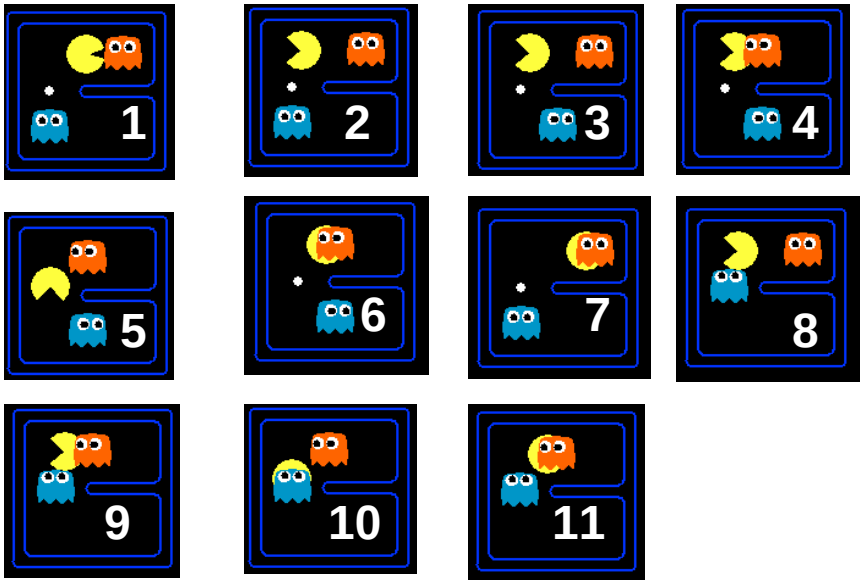
Given the following 'trapped' maze (which is 3x3 excluding the external walls), build the expectimax tree, with max and chance nodes clearly identified. Use the game score as the evaluation function at the leaves. If you don't want to draw little drawings, all the possible outcomes of this game have already been labeled for you on the following page; use them to identify the states in your tree. Pacman starts with the first move, followed by the lower left ghost and then the top right one.

CS188 – Introduction to Artificial Intelligence

Section Handout #4 Expectiminimax and MDPs

Klein, Fall 2007

Positions:



1. Draw the expectimax tree here.

2. What would pacman do if it was doing minimax instead?

3. 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*

and use this new game score as your evaluation function at the leaves.

[Note that this yields a monotonic transformation of the original utilities – i.e. a function which preserves the ordering of the state according to their utility.]

Could this change the decision of pacman using expectimax?

4. By changing the probabilities of action for the ghost, can you get expectimax to make the same decision as minimax, when using the original utility function from 1)?

Question 2 (Homework) - MDP

For the case where ghosts move uniformly at random amongst legal moves, describe pacman's decision problem (with project 2 rules) as an **MDP.** Each component should be answered in at most one sentence (just give the general idea).

Hint: What is the source of randomness for the transition function in this case?

a. *States:*

b. *Actions:*

c. *Transition model T(s,a,s'):*

d. *Reward function R(s,a,s'):*

e. *Discount factor:*