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.

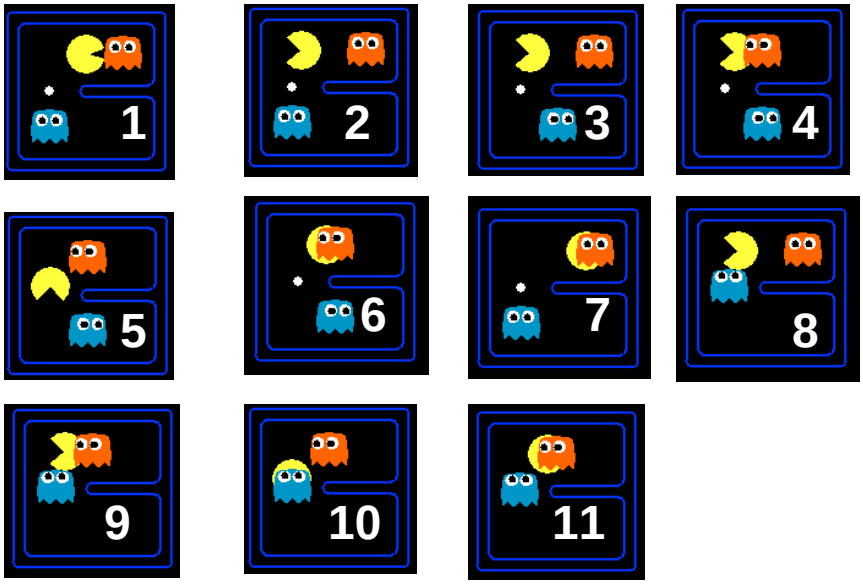
Given the following 'trapped' maze (which is 3x3 excluding the external walls), build the expectimax tree, with max and chance nodes clearly identified. Use the game score as the evaluation function at the leaves. If you don't want to draw little drawings, all the possible outcomes of this game have 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

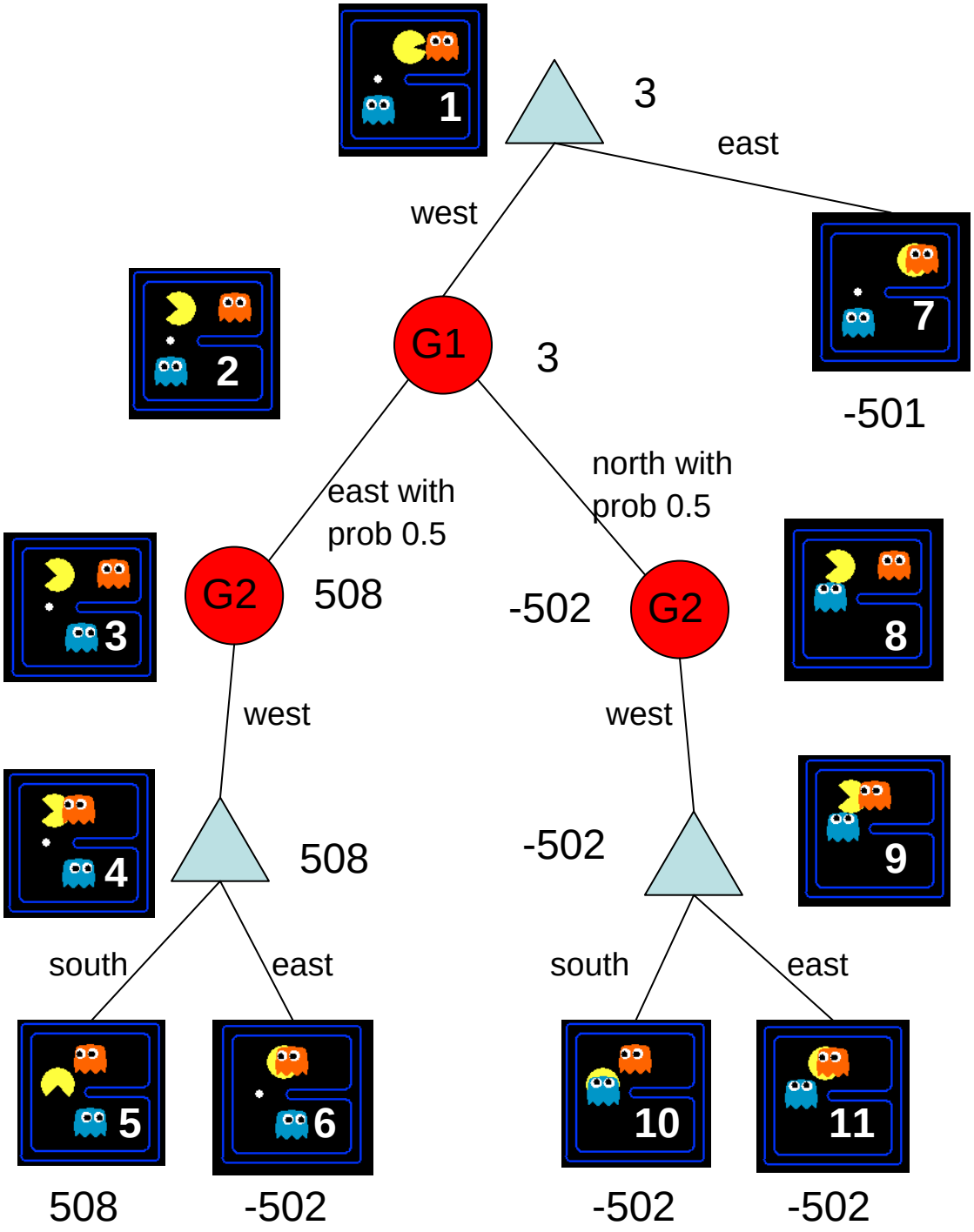
SOLUTIONS - Section Handout #4 Expectiminimax and MDPs

Klein, Fall 2007

Positions:



1. Draw the expectimax tree here.


1
3
east
west
7
2
G1
3
-501
east with
prob 0.5
north with
prob 0.5
3
G2
508
-502
G2
8
west
west
4
508
-502
9
south
east
south
east
5
6
10
11
508
-502
-502
-502

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

It would go east since the value of going west is -502 in the worst case vs. -501 to go east.

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?

Yes. The value of going west becomes -2.6 in expectation (the average of -1.7 and -3.5) with this evaluation function vs. -2.5 for going east. This shows that a monotonic transformation of the utility doesn't preserve necessarily the ordering of the actions when doing expectimax (vs. it is preserved for minimax). Only linear transformation of the utilities will preserve the ordering of the actions for 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)?

Yes. Minimax is a special case of expectimax where the opponent chooses the worst action with probability 1 (i.e. the action which minimizes the utility). In our case, if the ghost flees from pacman with probability less than 1/1010 (about 0.1%), then expectimax will choose the same suicidal action as minimax. In this case, expectimax will conclude that the probability of trying to win is too low to counterbalance the cost of making one more move.

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:

The configurations of the board, as given in a GameState object, including food locations, agent positions, etc.

b. Actions:

A subset of {north, south, east, west, stop}, depending by state.

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

The movement of pacman is deterministic; the randomness comes from the fact that the ghosts move at random here (the ghosts are part of the environment). So the distribution on *s'* is uniform over all possible states reachable with legal moves by the ghosts from *s* with pacman having moved according to *a* from *s.*

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

Reward = -1 + (500 if pacman has won) + (-500 if pacman has lost) + (10 if pacman just ate food) + (200 if pacman just ate a ghost)

e. Discount factor:

Because of the -1 reward at each time step and the limited amount of resources, the utility is bounded above, so there is no need of a discount factor to compare alternatives: use 1 as the multiplicative discount factor.