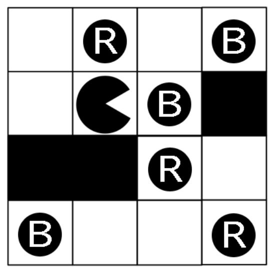
CS 188 Fall 2017 | Introduction to Artificial Intelligence | Exam Prep Solutions Search, CSPs, Game Trees

Q1. Foodie Pacman

There are two kinds of food pellets, each with a different color (red and blue). Pacman is only interested in tasting the two different kinds of food: the game ends when he has eaten 1 red pellet and 1 blue pellet (though Pacman may eat more than one of each pellet). Pacman has four actions: moving up, down, left, or right, and does not have a "stay" action. There are K red pellets and K blue pellets, and the dimensions of the board are N by M.



$K = 3$, $N = 4$, $M = 4$

(a) Give an efficient state space formulation of this problem. Specify the domain of each variable in your state space.

We need two variables to describe the location of pacman, one boolen variable showing whether pacmac already ate a red pellet, and another boolean variable for the blue pellets. Formally:

$(x \in [1 : N], y \in [1 : M], eaten_R \in \{T, F\}, eaten_B \in \{T, F\})$

(b) Give a tight upper bound on the size of the state space.

There are at most $N \times M$ possible locations for pacman and 4 possible assignments to the boolean variables so the size of the state space is upper bounded by $4 \times N \times M$

(c) Give a tight upper bound on the branching factor of the search problem.

Each state has at most four distinct successors corresponding to the four possible actions. The branching factor is at most 4.

(d) Assuming Pacman starts the game in position (x,y), what is the initial state?

(x, y, F, F). The two boolean state variables are both $false$.

(e) Define a goal test for the problem.

$(eaten_R == T)\&\&(eaten_B == T)$

(f) For each of the following heuristics, indicate (yes/no) whether or not it is admissible (a correct answer is worth 1 point, leaving it blank is worth 0 points, and an incorrect answer is worth -1 points).

Heuristic	Admissible?
The number of pellets remaining	1. No
The smallest Manhattan distance to any remaining pellet	2. Yes
The maximum Manhattan distance between any two remaining pellets	3. No
The minimum Manhattan distance between any two remaining pellets of opposite colors	4. No

1. Inadmissible because Pacman only cares about eating one pellet of each color. Adding extra pellets to the problem does not increase the optimal cost but it does increase this heuristic.

2. Admissible since Pacman needs to eat at least one extra pellet to reach the goal from any non-goal state and the cost of moving to that extra pellet is greater than of equal to the heuristic. Needs to be defined to equal zero at goal states.

3. Inadmissible. Adding extra pellets to the problem does not increase the optimal cost but it does increase this heuristic.

4. Inadmissible for the states where Pacman has already eaten one pellet.

Q2. Games: Three-Player Cookie Pruning

Three of your TAs, Alvin, Sergey, and James rent a cookie shuffler, which takes in a set number of cookies and groups them into 3 batches, one for each player. The cookie shuffler has three levers (with positions either UP or DOWN), which act to control how the cookies are distributed among the three players. Assume that 30 cookies are initially put into the shuffler.

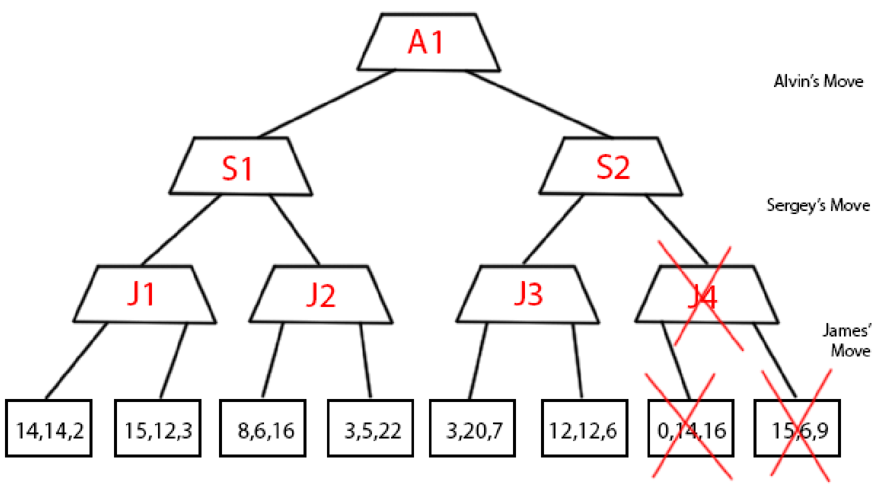
Each player controls one lever, and they act in turn. Alvin goes first, followed by Sergey, and finally James. Assume that all players are able to calculate the payoffs for every player at the terminal nodes. Assume the payoffs at the leaves correspond to the number of cookies for each player in their corresponding turn order. Hence, an utility of (7,10,13) corresponds to Alvin getting 7 cookies, Sergey getting 10 cookies, and James getting 13 cookies. No cookies are lost in the process, so the sum of cookies of all three players must equal the number of cookies put into the shuffler. Players want to maximize their own number of cookies.

(a) What is the utility triple propagated up to the root? 15,12,3

(b) Is pruning possible in this game? Fill in "Yes" or "No". If yes, cross out all nodes (both leaves and intermediate nodes) that get pruned. If no, explain in one sentence why pruning is not possible. Assume the tree traversal goes from left to right.

● Yes.

○ No, Reasoning:



Left lowest subtree: James chooses the node (15,12,3) at node J1. This gets propagated up to S1. Sergey doesn't know what value he can get on his right child, so we explore that. Upon propagating (8,6,16) to J2, we must explore J2's right child since there could be a triple better for James' best option (greater than 16) and better than Sergey's best option. (greater than 12). (15,12,3) gets propagated up to A1.

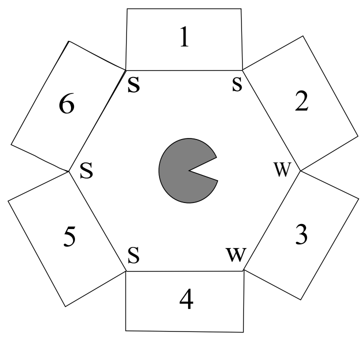
Alvin might have a good option (greater than 15) in the right subtree, so we explore down. On the (3,20,7) node, we propagate this up to J3. We need to continue going down this path because Sergey doesn't know if he can get more than 20, and Alvin doesn't know if he can get more than 15 (A1's value). Hence, (12,12,6) is explored. (3,20,7) is propagated to S2. Now, we can guarantee that Sergey will prefer any cookie count over 20. But, because the sum of cookies must be 30, this means that Alvin can get no more than 10 cookies in the right subtree. Hence, we can immediately prune any children of S2.

Q3. CSPs: Trapped Pacman

Pacman is trapped! He is surrounded by mysterious corridors, each of which leads to either a pit (P), a ghost (G), or an exit (E). In order to escape, he needs to figure out which corridors, if any, lead to an exit and freedom, rather than the certain doom of a pit or a ghost.

The one sign of what lies behind the corridors is the wind: a pit produces a strong breeze (S) and an exit produces a weak breeze (W), while a ghost doesn't produce any breeze at all. Unfortuantely, Pacman cannot measure the the strength of the breeze at a specific corridor. Instead, he can stand *between* two adjacent corridors and feel the max of the two breezes. For example, if he stands between a pit and an exit he will sense a strong (S) breeze, while if he stands between an exit and a ghost, he will sense a weak (W) breeze. The measurements for all intersections are shown in the figure below.

Also, while the total number of exits might be zero, one, or more, Pacman knows that two neighboring squares will *not* both be exits.



Pacman models this problem using variables X_i for each corridor i and domains P, G, and E.

(a) State the binary and/or unary constraints for this CSP (either implicitly or explicitly).

From the breezes, we get the following constraints:

Binary		Unary
$X_1 = P$ or $X_2 = P$,	$X_2 = E$ or $X_3 = E$,	$X_2 \neq P$
$X_3 = E$ or $X_4 = E$,	$X_4 = P$ or $X_5 = P$,	$X_3 \neq P$
$X_5 = P$ or $X_6 = P$,	$X_1 = P$ or $X_6 = P$,	$X_4 \neq P$

And there is another binary constraint: If adjacent(i, j), then $\neg(X_i = E \wedge X_j = E)$.

(b) Cross out the values from the domains of the variables that will be deleted in enforcing arc consistency.

X_1	P		
X_2		G	E
X_3		G	E
X_4		G	E
X_5	P		
X_6	P	G	E

(c) According to MRV, which variable or variables could the solver assign first?

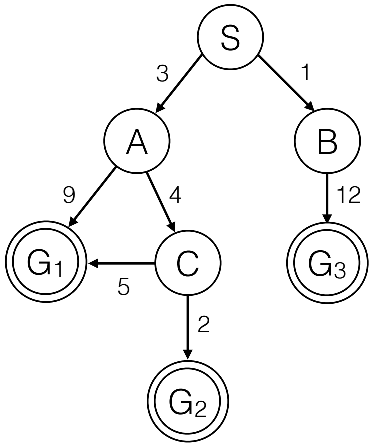
X_1 or X_5 (tie breaking)

(d) Assume that Pacman knows that $X_6 = G$. List all the solutions of this CSP or write *none* if no solutions exist.

(P,E,G,E,P,G)
(P,G,E,G,P,G)

Don't forget that exits cannot be adjacent to each other, and that it takes at least one exit to generate a weak breeze.

Q4. Search: Algorithms



	A	B	C	S
H-1	0	0	0	0
H-2	6	7	1	7
H-3	7	7	1	7
H-4	4	7	1	7

(a) Consider the search graph and heuristics shown above. Select **all** of the goals that **could** be returned by each of the search algorithms below. For this question, if there is a tie on the fringe, assume the tie is broken **randomly**.

(i) DFS — G_1 ● G_2 ● G_3 ●

(ii) BFS — G_1 ● G_2 ○ G_3 ●

(iii) UCS — G_1 ○ G_2 ● G_3 ○

(iv) Greedy with H-1 — G_1 ● G_2 ● G_3 ●

(v) Greedy with H-2 — G_1 ● G_2 ○ G_3 ○

(vi) Greedy with H-3 — G_1 ● G_2 ○ G_3 ●

(vii) A* with H-2 — G_1 ○ G_2 ● G_3 ○

(viii) A* with H-3 — G_1 ○ G_2 ● G_3 ○

(b) For each heuristic, indicate whether it is consistent, admissible, or neither (select more than one option if appropriate):

(i) H-1 — Consistent ● Admissible ● Neither ○

(ii) H-2 — Consistent ○ Admissible ● Neither ○

(iii) H-3 — Consistent ○ Admissible ○ Neither ●

(iv) H-4 — Consistent ● Admissible ● Neither ○