Last name:________________ First name:________________ SID:__________ Class account login:__________

Collaborators: ____

CS188 Spring 2010 Written 2: CSP's and Min-Max Search

Due: Thursday 2/11 in 283 Soda Drop Box by 11:59pm (no slip days)

Policy: Can be solved in groups (acknowledge collaborators) but must be written up individually.

1 [11 pts] Airport

You are in charge of scheduling arrivals for ATI (Alan Turing International). Today, there will be 4 arrivals, each of which you must assign a runway and a time slot in the grid below.

	T1	T2	T3	T4
R				
L				

Here are the requirements for the schedule:

1. Air Force One (AF1) must land on runway R due to motorcade and secret service logistics.
2. The airport closes (no landings) for one timeslot before, one during, and one after the arrival of AF1.
3. The Blue Angels (BA) will land in formation, which requires both runways at the same time.
4. The Blue Angels are exempt from AF1 related airport closures.
5. The new Boeing 777 Dreamliner (B777) must land before the Blue Angels
6. The B777 must land on runway L.
7. The Cessna (C) must not land beside or one timestep after the B777, due to turbulence considerations.
8. No two planes may land on the same runway at the same time (!!)

[2 pt] (a) Represent the problem with 4 variables: AF1, B777, C, and BA. The domain of BA is a time between 1 and 4 that the Blue Angels will arrive; the domain of the other three variables is a time between 1 and 4 plus 'R' or 'L' to specify the runway being landed on, as shown in part (c). Enumerate separately the unary and binary constraints in this problem. For the binary constraints, you may use pseudocode for implicit functions, like $beside(\cdot, \cdot)$

Unary Constraints | Binary Constraints

[1 pt] (b) Write constraint 5 in explicit form:

[1 pt] (c) Enforce all *unary* constraints by crossing out values in the table below.

AF1	R1	R2	R3	R4	L1	L2	L3	L4
B777	R1	R2	R3	R4	L1	L2	L3	L4
C	R1	R2	R3	R4	L1	L2	L3	L4
BA	1	2	3	4				

[1 pt] (d) Transfer your answer from part (c) to the table below. Then run arc-consistency.

AF1	R1	R2	R3	R4	L1	L2	L3	L4
B777	R1	R2	R3	R4	L1	L2	L3	L4
C	R1	R2	R3	R4	L1	L2	L3	L4
BA	1	2	3	4				

[5 pt] (e) Assuming you have not yet found a unique solution, perform backtracking search, and maintain arc-consistency after each variable assignment. Use the Minimum Remaining Values (MRV) heuristic to choose which variable to assign first, breaking ties in the order AF1, B777, C, BA. Draw a search tree illustrating which variable assignments are being made during search.

[grids provided for your convenience; you may not need all of them.]

AF1	R1	R2	R3	R4	L1	L2	L3	L4
B777	R1	R2	R3	R4	L1	L2	L3	L4
C	R1	R2	R3	R4	L1	L2	L3	L4
BA	1	2	3	4				

AF1	R1	R2	R3	R4	L1	L2	L3	L4
B777	R1	R2	R3	R4	L1	L2	L3	L4
C	R1	R2	R3	R4	L1	L2	L3	L4
BA	1	2	3	4				

AF1	R1	R2	R3	R4	L1	L2	L3	L4
B777	R1	R2	R3	R4	L1	L2	L3	L4
C	R1	R2	R3	R4	L1	L2	L3	L4
BA	1	2	3	4				

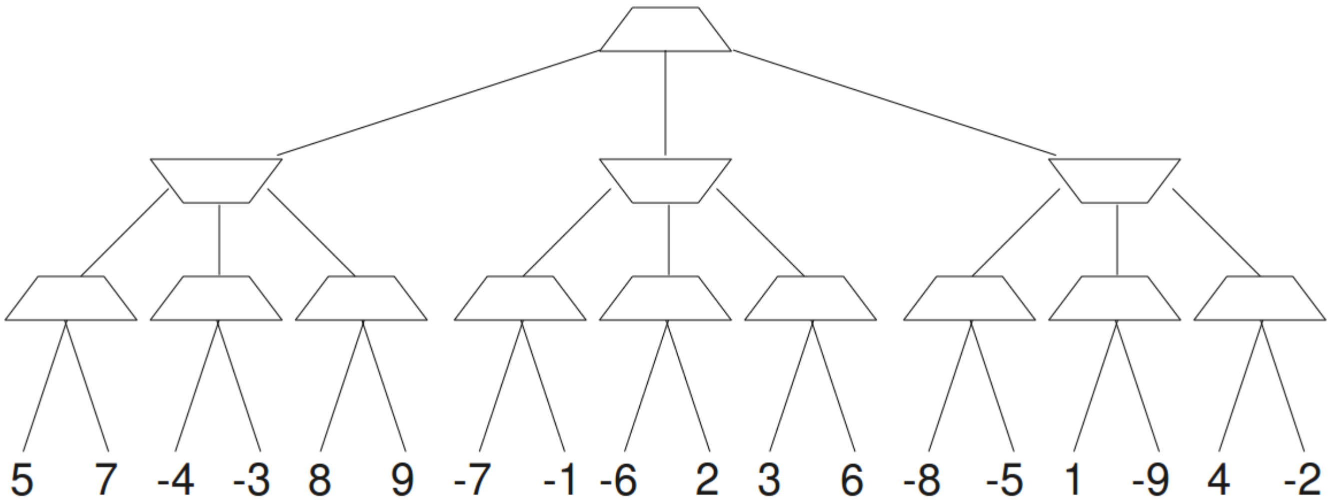
[1 pt] (f) How many solutions does this Constraint Satisfaction Problem have? If any, list all of them.

2 [11 pts] Min-Max 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.

[1 pt] (a) Assuming both opponents act optimally, carry out the min-max search algorithm. Write the value of each node inside the corresponding trapzoid. What move should you make now? How much is the game worth to you?



[3 pt] (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?

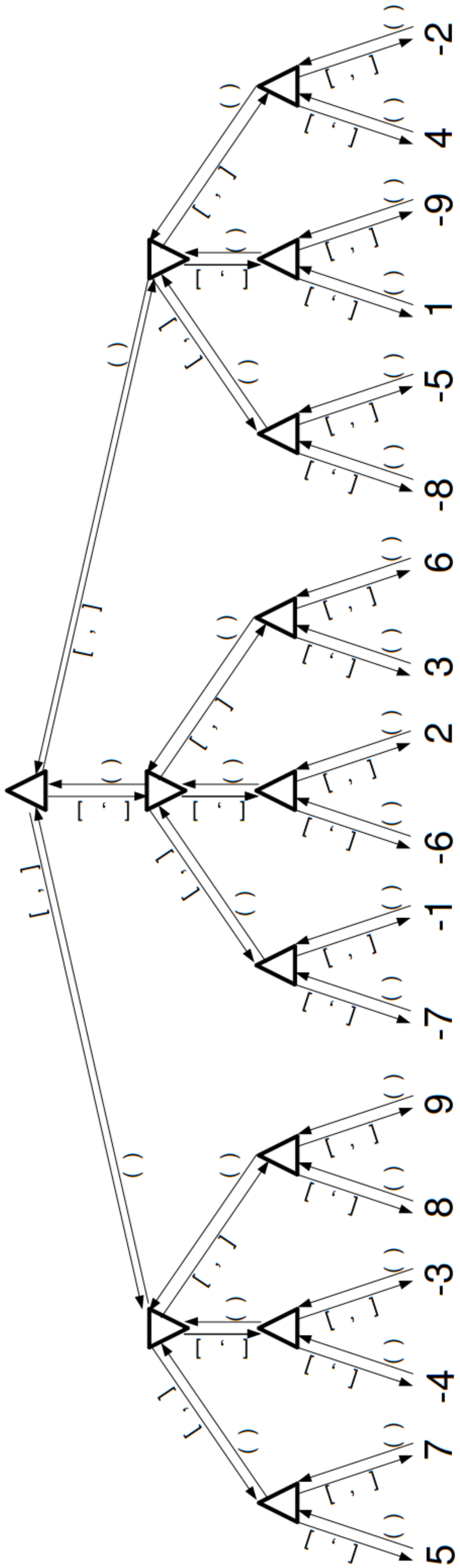
[1 pt] (c) Did α-β pruning find the *optimal* move in part (b)? In general, are i) minimax and ii) α-β pruning guaranteed to find the *optimal* move?

[4 pt] (d) Consider again the same game tree, searched using α-β pruning. This time, rather than expanding successors from left to right assume you can decide the order in which you expand successors. Which order would be optimal (result in exploring as few nodes as possible for this particular game)? As in part (b), record the $[\alpha, \beta]$ values passed down the tree, and the (v) return values passed up. Circle visited leaf nodes, and put an 'X' through pruned edges.

[2 pt] (e) Assume you have an evaluation function which for each node can provide an estimate of the minimax value. How can you use these minimax value estimates to guide the order in which successors are expanded, with the goal of minimizing the number of leaf nodes visited while running the α-β pruning algorithm?

.

(b)



(d)