ASSIGNMENT 3:

Assigned: 03/3/16 Due: Wednesday 3/9/16 at 11:55 PM (submit via moodle) Submit only pdf or txt files (in a zip if you have multiple files)

Update: A few parts in Problems 4 and 5 have been clarified and denoted with an underline.

Written/drawn:

Problem 1. (10 points)
Find all pure Nash equilibrium and Pareto optimum in this game. Show work why you think your claims are true.

<table>
<thead>
<tr>
<th>P1 Action 1</th>
<th>P2 Action 1</th>
<th>P2 Action 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 Action 1</td>
<td>P1 = 10, P2 = 5</td>
<td>P1 = 2, P2 = 7</td>
</tr>
<tr>
<td>P1 Action 2</td>
<td>P1 = 3, P2 = 8</td>
<td>P1 = 4, P2 = -1</td>
</tr>
</tbody>
</table>

Problem 2. (20 points)
Find the mixed strategy Nash equilibrium for Problem 1 for both players. Show (for both players) that by playing the Nash equilibrium, the opponent's reward for the game is fixed (i.e. the same for all actions).

Problem 3. (40 points)
Consider the following CSP of jobs under the following constraints:
J1 takes 3 time units to complete
J2 takes 2 time units to complete
J3 takes 1 time unit to complete
J1 must happen before J3
J2 cannot happen at time 1 or 2
All tasks must finish before time 7 (i.e. it is valid if J3 starts at time 6, but not if J2 starts at time 6 as it would end before time 8 yet after time 7.)

The domain (i.e. valid values) for all jobs is {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} (these are the time units).

1. Convert the last constraint (“All tasks must finish before time 7”) into multiple unary and/or binary constraints.
2. Apply all unary constraints and show the resulting domains for all variables.
3. Apply all 2-consistency constraints (assuming you already applied part (1)) and the resulting domains for all variables (i.e. run AC-3 on this problem).
4. Apply all 3-consistency constraints (assuming you already applied part (1) and (2)) and the resulting domains for all variables.
5. Based on the modifications to search, what variable and with what value should be expanded first using the information from part (3).
Problem 4. (10 points)
If you do not know how to play checkers, go read this: https://simple.wikipedia.org/wiki/Checkers.
Assume you have a a 4x6 board (with players starting on the length 4 sides) as such:

```
 X  X  O  O
 X  O  O
 X  X  O
```

Distance from X side:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Assume you have a mid-state evaluation as:

\[
\sum_{p \in \{\text{all your pieces}\}} DistanceFromYourSideTo(p) - \sum_{q \in \{\text{opponent’s pieces}\}} 1
\]

(1) What is the value of the following state:

```
 X  O  X
 X  X  O
```

(2) Should you use minimax to find the best move using the mid-state evaluation? Explain your reasoning clearly.

Problem 5. (20 points)
Assume you and an opponent are playing the following Prisoner's dilemma game:

<table>
<thead>
<tr>
<th></th>
<th>P2 confess</th>
<th>P2 lie</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 confess</td>
<td>P1 = -8, P2 = -8</td>
<td>P1 = 0, P2 = -10</td>
</tr>
<tr>
<td>P1 lie</td>
<td>P1 = -10, P2 = 0</td>
<td>P1 = -1, P2 = -1</td>
</tr>
</tbody>
</table>

You are going to play this game for 3 rounds with/against a tit-for-tat agent. Use alpha-beta pruning to prune the search tree (similar to alpha-beta pruning) to find the best course of action in a manner that will never remove an optimal solution. The “cooperative” action in this game is to “lie”, while the “competitive” action is to “confess”. The value of a terminal state is the sum of the rewards from the 3 games along the path.
(Hint: think about which branch you should go down first.)

(Lisp) programming:

Problem 6. (10 points)
Run the n-queens CSP problem for board sizes: 5, 10, 15 and 20 with both the backtracking and forward-checking algorithms. Use the built-in lisp function (time ...) to determine: the number of nodes searched, the memory used and the time to solve. For this part, simply report these 24 values in an organized manner, such as a table. (Use a decent computer.)