Deep Patel
Intro to AI - HW 3
10/21/18

Problem 1

For a generic graph G(V,E):
Variables, $X = \{v \in V\}$
Domains, $D = \{c_1 \ldots c_k\}$
Constraints, $C = \forall\ e_{i,j} \in E$ of the form (v_i,v_j) for some $v_i,v_j \in V$, $i \neq j$, $color(v_i) \neq color(v_j)$, where color(x) specifies the color that a vertex x is assigned.

Problem 2

For a 9x9 Sudoku game:
Variables, $X = \{v_{i,j}$, where $1 \leq i \leq 9$ and $1 \leq j \leq 9\}$
Domains, $D = \{1,2,3,4,5,6,7,8,9\}$
Constraints
$C_1 = v_{a,j} \neq v_{b,j}$ where $1 \leq a \leq 9$ and $1 \leq b \leq 9$, $a \neq b$ for $1 \leq j \leq 9$ (column values are different)
$C_2 = v_{i,a} \neq v_{i,b}$ where $1 \leq a \leq 9$ and $1 \leq b \leq 9$, $a \neq b$ for $1 \leq i \leq 9$ (row values are different)
C_3 = for any two vertices, $v_{a,b} = v_{c,d}$, $v_{a,b}, v_{c,d} \notin B_i$ for $1 \leq a \leq 9$, $1 \leq b \leq 9$, $1 \leq c \leq 9$, $1 \leq d \leq 9$, $1 \leq i \leq 9$, $a,b \neq c,d$ where B_i denotes any one of the 9 major 3x3 blocks. The blocks are between rows and columns as follows: $(1 \leq i \leq 3, 1 \leq j \leq 3)$, $(4 \leq i \leq 6, 1 \leq j \leq 3)$, $(7 \leq i \leq 9, 1 \leq j \leq 3)$, $(1 \leq i \leq 3, 4 \leq j \leq 6)$, $(4 \leq i \leq 6, 4 \leq j \leq 6)$, $(7 \leq i \leq 9, 4 \leq j \leq 6)$, $(1 \leq i \leq 3, 7 \leq j \leq 9)$, $(4 \leq i \leq 6, 7 \leq j \leq 9)$, $(7 \leq i \leq 9, 7 \leq j \leq 9)$.
Basically these constraints mean that each row, column, and 3x3 major block must contain different values. Because you can only choose from the nine values in the domain and there exactly nine positions to choose for each constraint, by the pigeonhole principle, it is the same thing as having exactly one occurrence of each of the elements of the domain {1,2,3,4,5,6,7,8,9} in each row, column, and major block.

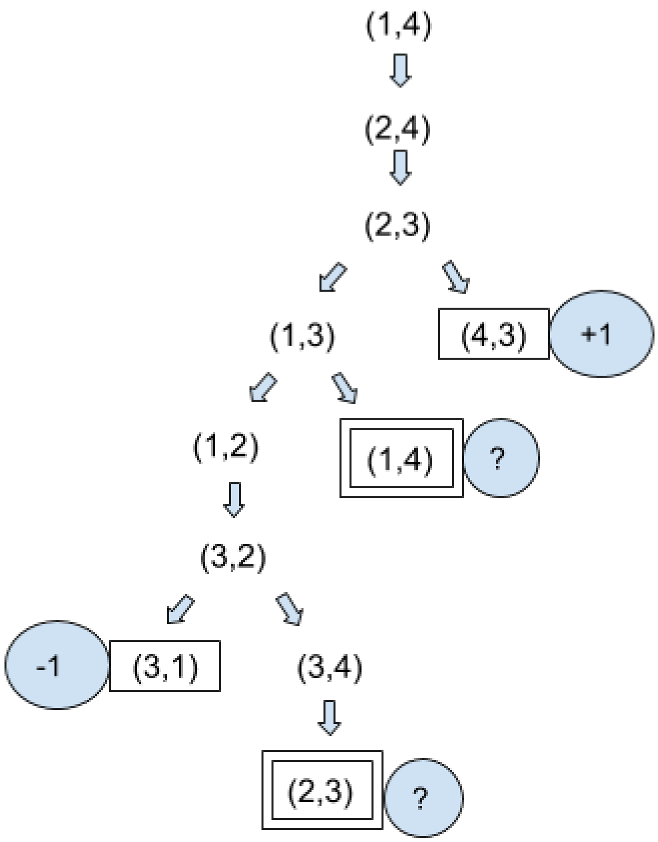
Problem 3

a) **a = 14**
   b = 12, c = 14, d = 9
   e = 12, f = 23, g = 24, **h = 21, i = 26, j = 14, k = 10, l = 9**
   m = 12, n = 21, o = 25, p = 2, q = 26, r = 11, s = 14, t = 4, u = 10, v = 8, w = 9
   At the max node a, the best sequence of moves is **a-c-j-s**, which yields a utility of **14**.
b) **a, b, e, f, g, c, h, m, n, i, o (termination), j, r, s, d, j, t, u, l, v, w**

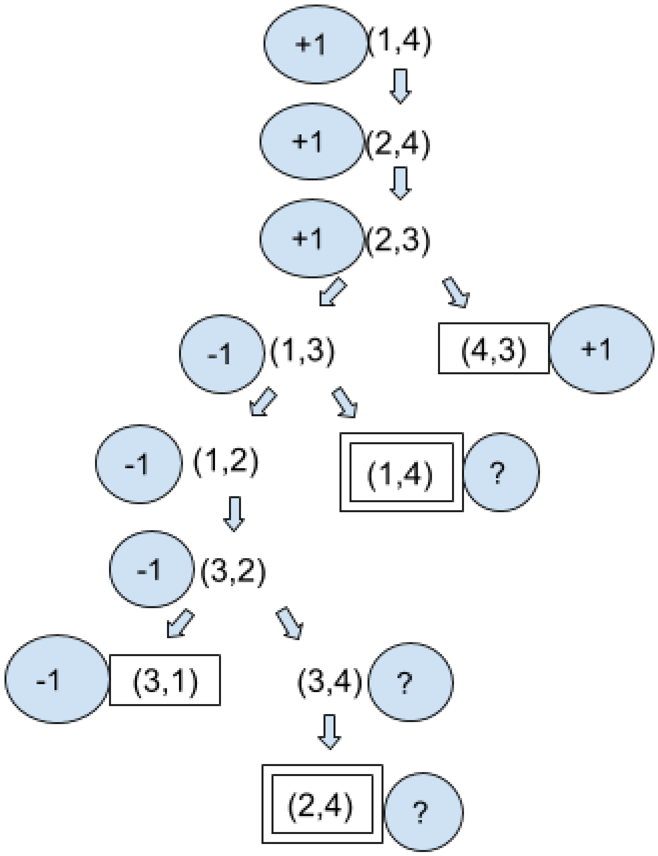
o - is terminated because its value, 25, is greater than the beta (min value) for c, which is already set to 21, which came from the b branch. So there is no need to look further into the actions of i, because the max value is at least 25, which will be ignored when getting the min for c. The alpha and beta values at this point are **[-∞, 21] for c** (and [12, +∞] for a).

Problem 4

a)



b)



The “?”s are handled by assuming that if player A or B have the choice of winning the game and entering a “?” loop, it will always choose to win. Thus, the value would be max(?,+1) = +1 and min(?,-1) = -1 depending on whose turn it is. if all the successors are “?” then the backed up value is “?”.

c) The standard Minimax algorithm would fail because as a depth-first search algorithm, it can go into an infinite loop. To correct this, we can keep a stack of visited states and whenever a state is revisited, return a “?”, which is handled as specified in part b. This modified algorithm still does not give optimal solutions to all games with loops because some games may not have the same values for terminal states (like a draw or win of different degree). Sometimes, being in an infinite loop may be prefered by both players in the game, in which case the game can be called a draw.

d) Using a proof by induction, we can trivially see that for the base case of n=3, A will lose and for n=4 A will win. For any game of size n>4, the initial moves for A and B are the same. Both must move one step closer to the opposite direction. This effectively makes a subgame of size n-2. It is clear that whoever wins this subgame will win the game of size n because in the next step they can simple move one step to the goal. In the case that a player moves back to their home square, it is obvious that that this strategy will have them lose because the game will become a subgame of size n - 2k, closer to the loser's home where, where k is a positive integer.

Problem 5

a) The dominant strategy for Player 1 is strategy **3** because if Player 2 chooses a, then strategy 1 would lead to a payoff of 1, which is lower than 6 for strategy 3. Also, if Player 2 chooses b, then strategy 2 would lead to payoff of 3 instead of 4 for strategy 3. Thus in either case, Player 1 is better off with strategy 3, so it is the dominant strategy. For Player 2, a dominant strategy does not exist because the payoffs rely heavily on Player 1's decision. In other words, there is no one choice for Player 2 that will always lead to a better payoff than the others. If Player 2 chooses strategy b then it will be better off if Player 1 chooses either strategies 1 or 3, but not if it chooses 2. Thus, there is no dominant strategy for Player 2.

b)

i) (1,b) because (2,b) and (3,b) lead to equal or less payoff for Player 1 and (1,a) leads to a lower payoff (from 4 to 2) for Player 2.

ii) (3,b) because (1,b) and (2,b) lead to equal or less payoff for Player 1 and (3,a) leads to a lower payoff (from 7 to 3) for Player 2.

iii) (2,a) because (1,a) and (3,a) lead to equal or less payoff for Player 1 and (2,b) leads to a lower payoff (from 3 to 1) for Player 2.

Problem 6

a) Lets define the following statements:

m = motion is detected,
s = system in armed,
f = there is a fire, and
a = alarm goes off.

Then the following propositional statements can be inferred:

$s \land m \Rightarrow a$
$a \Rightarrow s \lor f$
$f \Rightarrow a$
m

b) $(s \wedge m \Rightarrow a) \wedge m$

$s \Rightarrow a$

$(s \Rightarrow a) \wedge (f \Rightarrow a)$

$s \vee f \Rightarrow a$

Thus,
$(s \vee f \Rightarrow a) \wedge (a \Rightarrow s \vee f)$

$\mathbf{a \Leftrightarrow s \vee f}$

Problem 7

The possible initial tosses can be 2 heads, in which case the number of tosses is 2 and we are done, or heads then tails, in which case we wasted 2 tosses and start from scratch, or start with a tail, in which case we waste 1 move and start over. This can be put into a formula like below to get that the expected value of X is 6.

E[X] = P(HH)*2 + P(HT)*(E[X]+2) + P(T)*(E[X]+1)
= 0.25*2 + 0.25*(E[X]+2) + 0.5*(E[X]+1)
= 0.5 + 0.5 + 0.5 + 0.75E[X]

0.25E[X] = 1.5

E[X] = 6