

CS 440 Fall 2018 Homework Assignment 3

Rafal Stapinski

October 21 2018

1

Variables: $X = \{V_1, \dots, V_n\}$ for V in G where $G = (V, E)$

Domains: $D = \{c_1, c_2, \dots, c_k\}$

Constraints: $v_i \neq v_j$ if i, j are adjacent

2

Variables: $X = \{v_{1,1}, \dots, v_{1,9}, v_{2,1}, \dots, v_{2,9}, \dots, v_{9,1}, \dots, v_{9,9}\}$

Domain: $D = 1, 2, 3, 4, 5, 6, 7, 8, 9$

Constraints:

There is a one-to-one mapping of every $v_{i,1}$ through $v_{i,9}$ and the domain

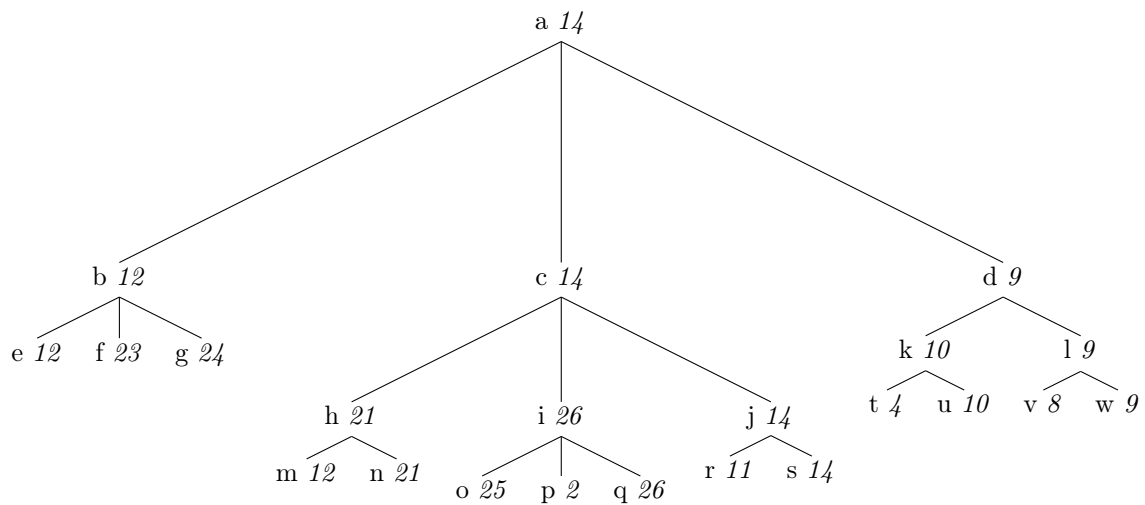
There is a one-to-one mapping of every $v_{1,j}$ through $v_{9,j}$ and the domain

There is a one-to-one mapping of every set of $\{v_{1,1}, \dots, v_{3,3}\}, \{v_{1,4}, \dots, v_{3,6}\}, \{v_{1,7}, \dots, v_{3,9}\}, \{v_{4,1}, \dots, v_{6,3}\}, \{v_{4,4}, \dots, v_{6,6}\}, \{v_{4,7}, \dots, v_{6,9}\}, \{v_{7,1}, \dots, v_{9,3}\}, \{v_{7,4}, \dots, v_{9,6}\}, \{v_{7,7}, \dots, v_{9,9}\}$ to the domain

The first constraint ensures that every row has a one-to-one mapping to the domain, so that there is exactly one of 1-9 in each row. The second constraint ensures the same mapping but for every column. The third constraint ensures the same mapping for each of the 9 squares.

3

3.1



The best path is: a, c, j, s

3.2

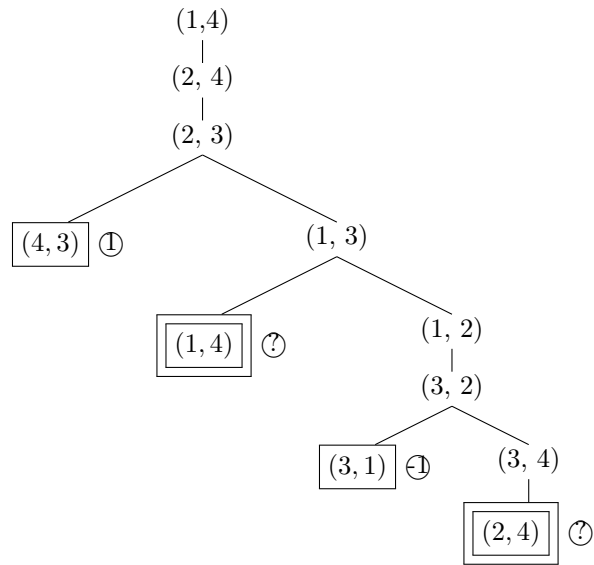
Visit A

Visit B

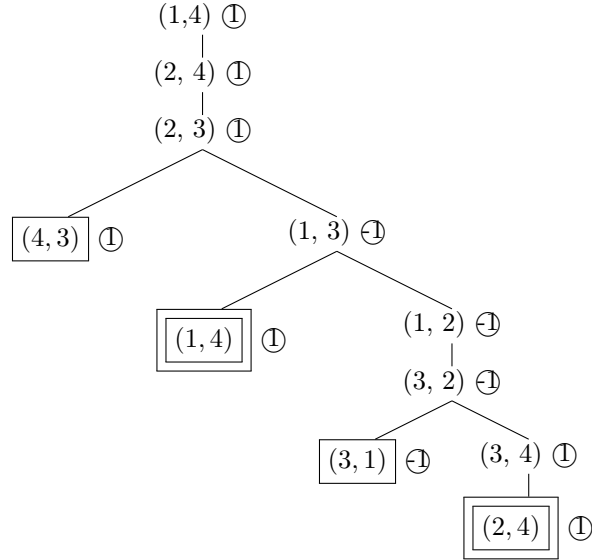
Visit E
 Visit F
 Visit G
 Visit C
 Visit H
 Visit N
 Visit I
 Visit O
 Skip P, Q, α of I = 25 β of I = 21
 Visit J
 Visit R
 Visit S
 Visit D
 Visit K
 Visit T
 Visit U
 Visit L
 Visit V
 Visit W

4

4.1



4.2



The two looped states occur at (1,4) and (2,4) where there is only one possible move. Because of this, there would not be any change in predicted outcome. Both states only have one meaningful child state (2,3). In this child state, a max occurs which always results in a 1 due to the (4,3) terminal state, so both of the loop states were set to 1.

4.3

The standard MinMax algorithm would fail as it would continue infinitely at the first loop state (1,4). To work on any tree with loop states, the MinMax algorithm should be modified to initially skip over any loop states it encounters. When returning to a state that infers from a deterministic and loop state, the algorithm can now choose the appropriate min/max of that state based on whether the parent loop state is a min/max. This would work in all cases, inferring the optimal min/max at a current loop state based on what the optimal choice would be of the parent/original state this would go to.

4.4

In cases where n is even, A and B meet become adjacent because of B's turn. At this point, A begins by skipping over B, thus always being closer to its goal from that point on. In cases where n is odd, B skips over A first thus being closer to its goal. In the case where $n=3$, A always wins on the first turn.

5

5.1

For Player 1, the dominant strategy is 3. For Player 2 there is no dominant strategy. For Player 1, regardless of choice of Player 2, strategy 3 provides the highest payoff. For Player 2, the choices of Player 1 change the highest payoff strategy.

5.2

(1,b), (2,a), (3,b) are Nash equilibria. In these cases, neither player has anything to gain by changing their choice alone. In all three cases, these are the choices the player would make had the other player made their move already.

6

6.1

A = system armed
M = motion detected
S = alarm will sound
F = fire

$$\begin{aligned}A \wedge M &= S \\(A \vee F) &\Rightarrow S \\F &\Rightarrow S \\M &= T\end{aligned}$$

6.2

$$\begin{aligned}\neg(A \vee F) &= \neg A \wedge \neg F \\(A \vee F) \Rightarrow S &= \neg(A \vee F) \vee S = \neg A \wedge \neg F \vee S \\&\{\neg A\}\{\neg F, S\}\end{aligned}$$

7

Expectation of X = 6

Probability of tails on first flip is 1/2, total number of flips requires is X+1
Probability of heads then tails is 1/4, with flips required being x+2
Probability of two heads is 1/4 and the required flips is 2

Adding the variables we get:

$$\begin{aligned}X &= (1/2)(X+1) + (1/4)(X+2) + (1/4)*2 \quad X = X/2 + 1/2 + X/4 + 1/2 + 1/2 \quad X = 3X/4 + 3/2 \\X/4 - 3/2 &= 0 \quad X = 6\end{aligned}$$