CIS 471: Homework 2

Zach Domke

2/16/19

Q1:

1. {X(i,j) ≠ X(k,j) | k ∈ S:{1,2,…n} ^ k ≠ i,

X(i,j) ≠ X(i,k) | k ∈ S:{1,2,…n} ^ k ≠ j,

X(1,2) > X(2,2),

X(3,2) > X(2,2),

X(3,1) > X(4,1),

X(4,2) > X(4,3),

X(3,4) > X(4,4)}

1. {X(3,4) > X(4,4),

X(3,4) ≠ 3,

X(4,4) ≠ 3}

Domains: X(3,4) = {1,2,3,4} X(4,4) = {1,2,3,4}

X(3,4) = {1,2,4} X(4,4) = {1,2,4}

X(3,4) = {2,4} X(4,4) = {1,2}

1. Maximum domain is 2
2. Maximum domain adjacent to an inequality is 2
3. Arc consistency will not find any other requirements because there are 2 inequalities

Q2:

1. False
2. ∞, It is possible for backtracking to get caught in an infinite loop without a proper ordering heuristic.
3. O(nd2), with a proper ordering heuristic, backtracking should only need to backtrack a few times as it will have some more information going into the decision process.
4. O(nd2) because backtracking should only need to analyze each node once as well as analyzing each node below for each color. This is faster because we don’t need to backtrack very often. Should be around O(nd2)
5. We should select a value for the middle node and use arc consistency from there. This cutset will help us determine a solution fastest.

Q3:

1. X≤1

As long as x is at most 1, then the min node will pick a value that is at most 1 and the max node will always pick the left node (1).

1. None

The rightmost node could always be something larger than either the left or middle nodes.

1. X≥2

If x<2 then we need to see what the last value is in order to make a fully educated decision. However, if we know that our left min-node is picking a value of at least 2, then once the right min-node sees the 2, we know that the right min-value will be at most 2. Therefore, the max-node will pick the left min-node always because it will be equal to or greater than the right min-node.

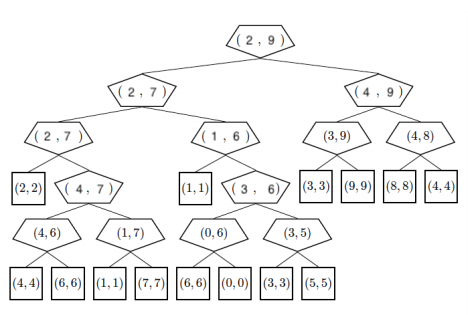
1. X≥3

The left min-node will pick a number of at least 3 and the right min-node will pick a value of at most 3. In this case the root max-node will not even need to consider the right min-node since it is at most 3.

1. None

The max-node will always need to see that right child in order to make an informed decision.

Q4:



1. Possible. No branches should be pruned.
2. If Pacman knew the ghost was friendly, the root would be 9
3. i. = A1

ii. = B8

iii. = A5

iv. = B8

1. i. = C1

ii. = D5

iii. = C8

iv. = D4

Q5:

1. (5)

There are no possible situations in which we can prune any of these nodes as it is necessary in order for us to determine which node is the median.

1. (3) and (4)

There are situations where we will not need to know the value of V7 or V8. One such case is if the left max-node chooses a value that will be the smallest of the left med-node’s choices.

1. (4)

We are sometimes able to prune V12 in the cases that V11 is larger than the value chosen by the left max-node.

1. (4)

We are sometimes able to prune V16 if we know that V15 is smaller than the middle min-node. In this case, we know that the right min-node will be the smallest value when the med-node makes it’s evaluations.