CS221 Fall 2015 Homework [number]

SUNet ID: waba
Name: Alexander Carlisle
Collaborators: Roger Song

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 0

(a) The CSP is as following. There are m variables corresponding to each of the m buttons. Each variable has a domain of 0,1 where a 1 indicates the button is turned on and a 0 indicates the button is turned off. There will be n constraints, one for each bulb, where the ith constraint ensures that the number of times the ith bulb has been toggled is odd. This is because if all bulbs start off and whenever any button is switch it toggles all bulbs in its subset, each bulb will only be on if it has been toggled an odd number of times.

(b) i. There are two consistent assignments, one where X1 and X3 are 1 and X2 is 0, and another where X1 and X3 are 0 and X2 is 1. ii. The picture ATTACHED TO BACK OF PDF shows how it makes 9 calls to backtrack without a heuristic

iii. The second PICTURE ATTACHED TO BACK OF PDF shows that with AC-3 it only makes 7 calls

(c) in Code

Problem 1

(a) All code

(b) Code

(c) in code

Problem 2

(a) I would implement a strategy similar to the one we used in lecture. I would introduce auxiliary variables A1,A2, and A3. Each variable would be a pair. A1[0] would store 0, and A1[1] would have a domain of all sums of 0 and any value in X1. A2[0] would have the domain equal to A1[1]'s, and A2[1] would have the domain equal to all possible sums between each number in A2[1]'s domain and each number in X2's domain. Basically the second part of each pair has the domain of all possible sums of all the X's up to that point. NOTE for the domain, only include values less than or equal to K. If A3 is defined similarly to A1 and A2 all we have to do is add constraints and we are done. We need the binary constraint that A1[1] equals A2[0] and another binary constraint that A2[1] must equal A3[0]. We also need 3 binary constraints one between (A1,X1), one between (A2,X2), and one between (A3,X3) such that the Ai[1] -Ai[0] is equal to Xi. Lastly, if we constrain A3[1] to be less than or equal to K we are done. Basically we made auxilary variables and gave them domains of possible additions of the X's and ensured that the Aux variables never exceeded max sum. By making the difference between pre and post for an Aux variable be the corresponding X variable, we ensured that if we could find values for the Aux variables that worked,than we are ensured that the X variables worked because they are constrained to be the difference between pre and post for each aux variable.

(b) Problem 3

(a) Coded

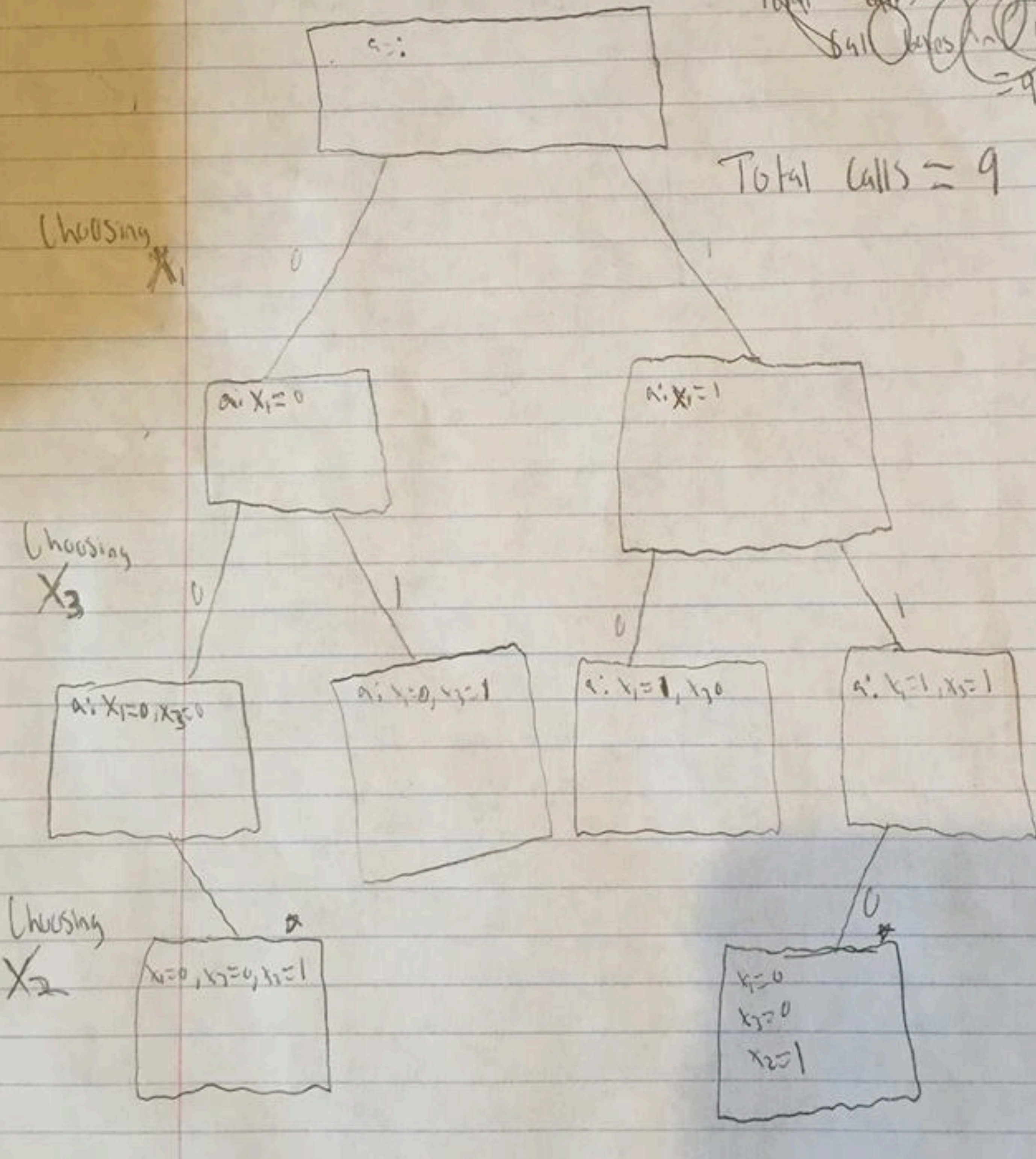
(b) Coded

(c) It worked out well! It was able to find classes for the winter and spring under my given constraints.

No heuristic.

~~Total calls = 51 ... all full boxes ... total place~~ = 9.

Total calls = 9



With heuristic

~~Short Calls = 7~~

Total Calls = 7