CS 470
Sam Hopkins
Hw 8

1. a) For sudoku, we would have each square connected to every square in it's column and row and local area. when we input a value, we update the domains for all of them by removing our inputted value from their domain

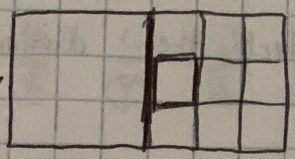
X_{23} =

5	6	9
2		
		1

Domain (row & column & local) = {4,7}
row = {1,8,3,2} column = {9,1,2,5} local = {1,2,5,6,9}

ii.) With so many possible outcomes (around 10^{21}), AC-3 would be extremely slow as it tries to solve the puzzle multiple times. It may have to back track all the way to the beginning because it can't detect all of the errors

iii) X_{24} →



We can put 1 there because in it's surrounding columns and rows there are 1's, so a 1 could not be placed anywhere else in that square

AC3 wouldn't catch that imediately because we draw our conclusion based on neighboring values, not ones that are available to it while performing forward traking. Its domain at that time is {1,6}, and only later would it realize that only 1 works

iv) Minimum value - Pick the space with the smallest domain
Degree - Pick the space that will effect the smallest amount of other domains. If it affects the least number of squares, then pick that square. Domains it would affect are rows, columns, and local squares.

b) i) R_i^m cannot equal R_i^m, so if R_2^1 than it cannot = R_2^1 for any other row.
R_i^m cannot equal C_i^n, [illegible]
more than once for each Row and column
R_i^m cannot equal G_i^m more than once for 3x3 grid

ii) The Domain would be {4}, after checking constraints on G_2^1

iii) Domain G_6^2 = {4} as well after checking all constraints

2a) i) The variables = a word in a row/column of the puzzle
Each has a domain is the N words in the dictionary

ii) There are 12 variables, one for each column and row

iii) 36 constraints, a letter in each word for each square that corresponds to a letter of the crossing word on either the column or row.
Each letter at position X needs to be a letter for a different word also at position X. ex.

P
S a T
T

iv) 36, one for each letter in puzzle

b) i) We check each _ square and check the dictionary. If dictionary ever returns 0, we backtrack

ii) We will choose the space with the most-constrained variable, or lowest number of values taken from the dictionary

c) We can quickly determine the domain of letters for each square, which would help as find dead ends quicker with the help of already inputted values. When a set is empty we backtrack,

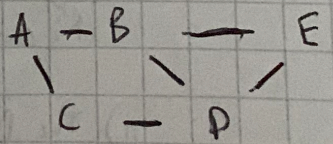
3.a) i) variables: Different state A-E
ii) Each variable has a Domain of {Green, Red, yellow}

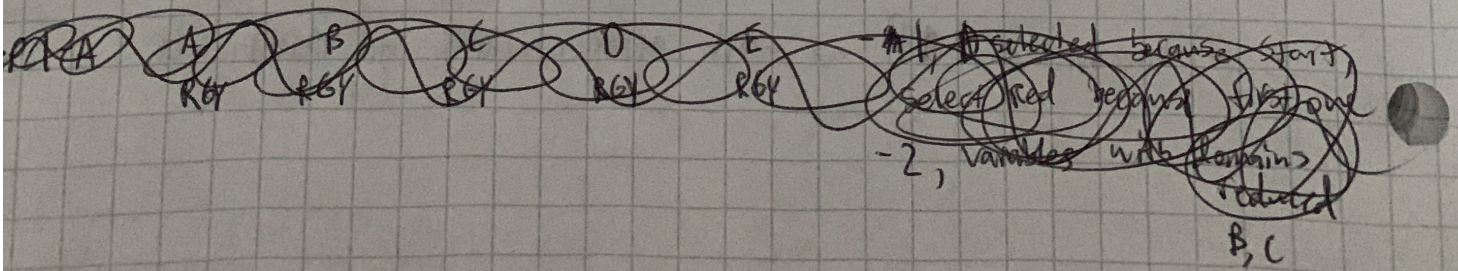
iii) constraints are that no two neighboring states will have the same two jersey colors.

b)

A	B	C	D	E
RGY	RGY	RGY	RGY	RGY

- Step 0

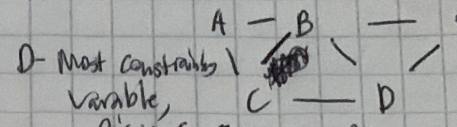




	A	B	C	D	E
Step 1,2:	~~RGY~~	~~RGY~~ (R)	GY	GY	GY
Step 3,4:	GY	(R)	Y	(G)	Y
Step 5,6:	G	(R)	(Y)	(G)	Y
Step 7,8:	G	(R)	(Y)	(G)	(Y)
Step 9,10	(G)	(R)	(Y)	(G)	(Y)

choose B, because has most constraining variable heuristic, choose Red to start
-Domains update = ~~R~~, C, E, A, B, D

D- Most constraining variable, pick G ~~R~~ to start
-Domains update: E, C



C - Most constraint variable, only one value
Domains updated: A

E - Most constraint? Tied with A, only one value, no domains updated

A - last only minimum Remaing, has only one - G

No domains updated