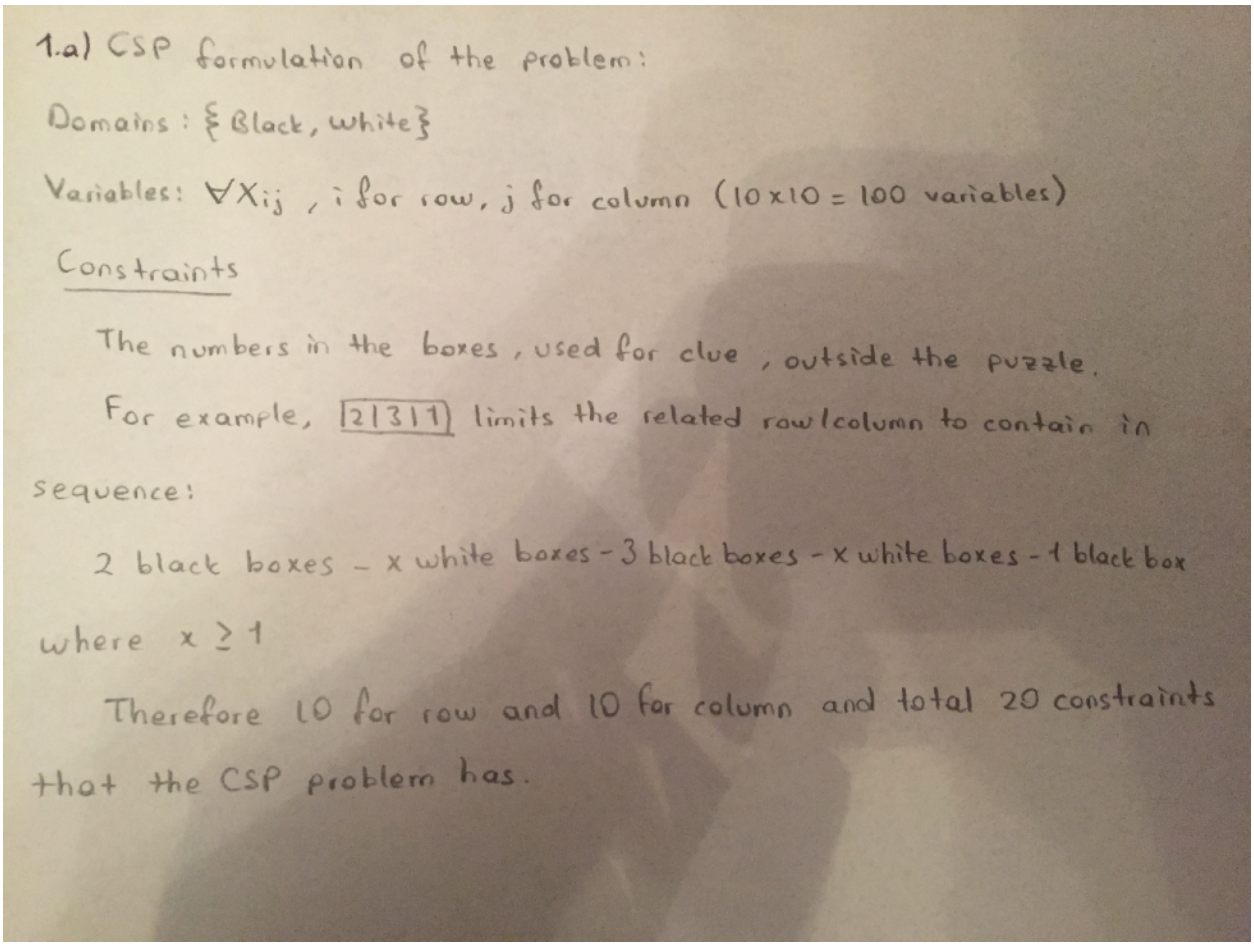
ARTIFICIAL INTELLIGENCE PROJECT 2 REPORT

MEHMET BARIŞ YAMAN - 150130136

Q1. A)



1.a) CSP formulation of the problem:

Domains : {Black, white}

Variables: $\forall X_{ij}$, i for row, j for column (10 x 10 = 100 variables)

Constraints

The numbers in the boxes, used for clue, outside the puzzle.

For example, [21311] limits the related row/column to contain in sequence:

2 black boxes – x white boxes – 3 black boxes – x white boxes – 1 black box

where $x \geq 1$

Therefore 10 for row and 10 for column and total 20 constraints that the CSP problem has.

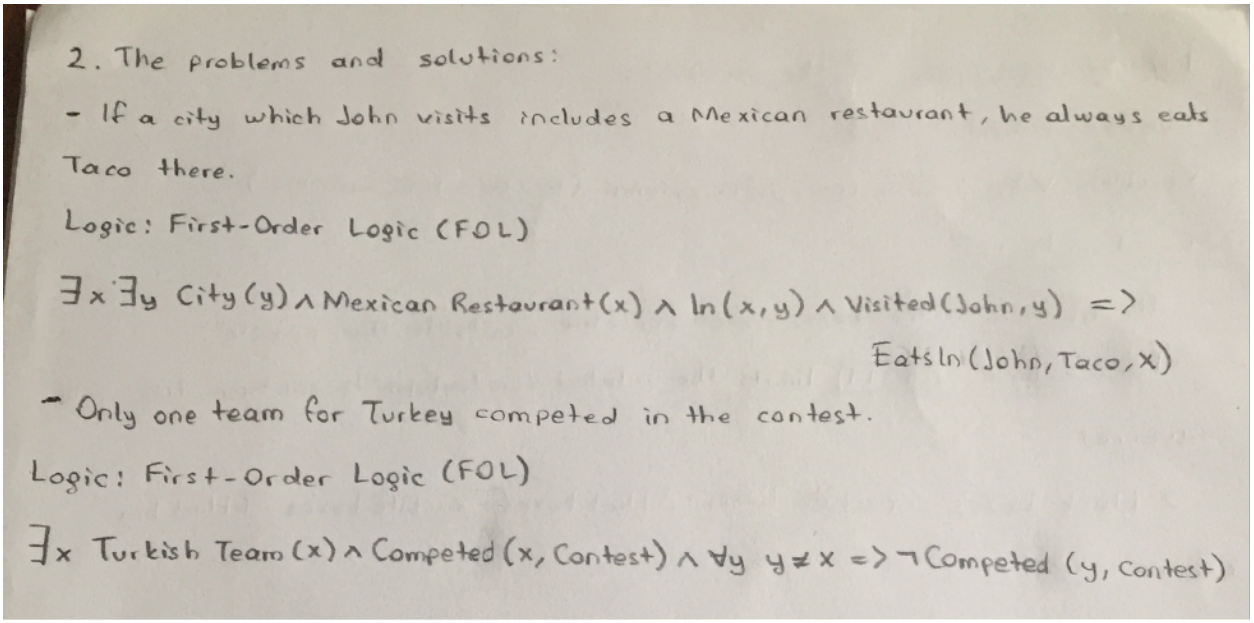
B)

I selected min_conflict algorithm from the SimpleAI library, to solve the problem. The function returns a dictionary value if it reaches the solution. Then, I print the dictionary returned from the algorithm. I did not give any initial assignment and limit to the algorithm since the solution is given in the Assignment 2 pdf, and after approximately 10 seconds, the algorithm finds the exact solution that is same as given in the assignment.

The function basically assign the variables that causes minimum conflicts between the variable domains by assigning the value to the assignment array. After that, it checks for all conflicts may be caused by calling the function namely "_find_conflicts". If it finds conflicts it selects a conflicted variable randomly and assigns a value that may cause minimum conflict. If there is no limit or the limit is not reached it calls _count_conflict function and breaks while loop corresponding to the value namely "run".

The SimpleAI Library used can be found on the link:
http://simpleai.readthedocs.io/en/latest/_modules/simpleai/search/csp.html#min_conflicts

Q2.



2. The problems and solutions:

- If a city which John visits includes a Mexican restaurant, he always eats Taco there.

Logic: First-Order Logic (FOL)

$\exists x \exists y \; City(y) \wedge Mexican\ Restaurant(x) \wedge In(x, y) \wedge Visited(John, y) \Rightarrow EatsIn(John, Taco, x)$

- Only one team for Turkey competed in the contest.

Logic: First-Order Logic (FOL)

$\exists x \; Turkish\ Team(x) \wedge Competed(x, Contest) \wedge \forall y \; y \neq x \Rightarrow \neg Competed(y, Contest)$

Q3. A)

3. a) Knowledge Base

LR: Living Room , G: Garden

$\neg Guilty(Ayse) \Leftrightarrow \neg In(Cem, LR) \wedge In(Baris, LR)$

$\neg Guilty(Baris) \Leftrightarrow In(Baris, G)$

$\neg Guilty(Cem) \Leftrightarrow \neg In(Baris, G) \wedge \neg In(Ayse, G)$

$Guilty(Ayse) \Leftrightarrow \neg Guilty(Baris) \wedge \neg Guilty(Cem)$

$Guilty(Baris) \Leftrightarrow \neg Guilty(Ayse) \wedge \neg Guilty(Cem)$

$Guilty(Cem) \Leftrightarrow \neg Guilty(Ayse) \wedge \neg Guilty(Baris)$

$Guilty(x) \Rightarrow In(x, LR)$ $\qquad \neg(Guilty(x) \wedge Guilty(y))$

$In(x, G) \Rightarrow \neg In(x, LR)$

$In(x, LR) \Rightarrow \neg In(x, G)$

Knowledge Base In CNF Form

$[Guilty(Ayse) \vee \neg In(Cem, LR)] \wedge [Guilty(Ayse) \vee In(Baris, LR)]$ (1)

$Guilty(Baris) \vee In(Baris, G)$ (2)

$[Guilty(Cem) \vee \neg In(Baris, G)] \wedge [Guilty(Cem) \vee \neg In(Ayse, G)]$ (3)

$[Guilty(Ayse) \vee \neg Guilty(Baris)] \wedge [Guilty(Ayse) \vee \neg Guilty(Cem)]$ (4)

$[Guilty(Baris) \vee \neg Guilty(Ayse)] \wedge [Guilty(Baris) \vee \neg Guilty(Cem)]$ (5)

$[Guilty(Cem) \vee \neg Guilty(Ayse)] \wedge [Guilty(Cem) \vee \neg Guilty(Baris)]$ (6)

$In(Cem, LR) \vee \neg In(Baris, LR) \vee \neg Guilty(Ayse)$ (7)

$\neg In(Baris, G) \vee \neg Guilty(Baris)$ (8)

$In(Baris, G) \vee In(Ayse, G) \vee \neg Guilty(Cem)$ (9)

$Guilty(Baris) \vee Guilty(Cem) \vee Guilty(Ayse)$ (10)

$\neg In(x, G) \vee \neg In(x, LR)$ (11)

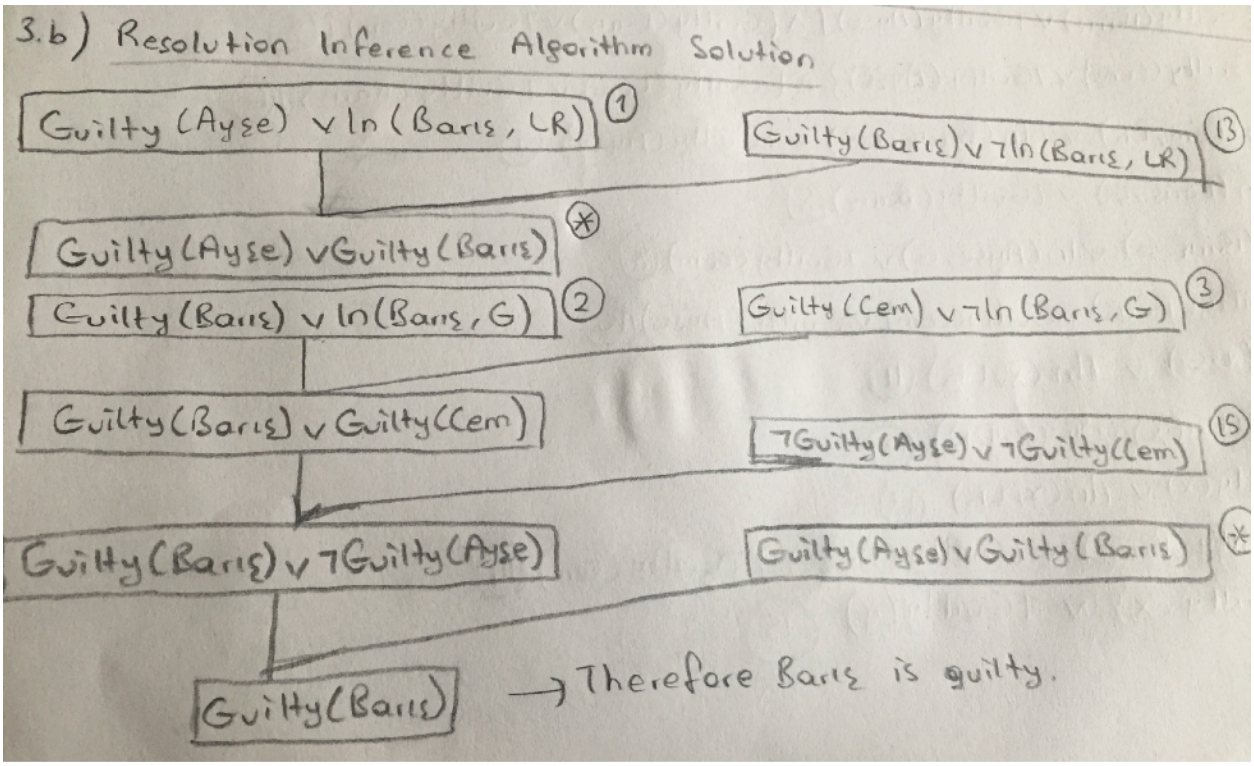
$\neg Guilty(x) \vee In(x, LR)$ (12)

$Guilty(x) \vee \neg In(x, LR)$ (13)

$\neg Guilty(Baris) \vee \neg Guilty(Ayse) \vee \neg Guilty(Cem)$ (14)

$\neg Guilty(x) \vee \neg Guilty(y)$ (15)

B)


3.b) Resolution Inference Algorithm Solution
Guilty (Ayse) v In (Baris, LR) 1
Guilty (Baris) v ¬In (Baris, LR) 13
Guilty (Ayse) v Guilty (Baris) *
Guilty (Baris) v In (Baris, G) 2
Guilty (Cem) v ¬In (Baris, G) 3
Guilty (Baris) v Guilty (Cem)
¬Guilty (Ayse) v ¬Guilty (Cem) 15
Guilty (Baris) v ¬Guilty (Ayse)
Guilty (Ayse) v Guilty (Baris) *
Guilty (Baris)
→ Therefore Baris is guilty.