BLG-435E ARTIFICIAL INTELLIGENCE, FALL 2018-2019
Assignment #2 Report

ONAT ŞAHİN – 150150129

QUESTION – 1

a) I have modeled this problem like below:

Variables: Variables are the blocks required for a structure. Location of the blocks and whether they are placed horizontally or vertically is known. Their domain is an integer array that goes from 1 to the number of blocks, which represents the order of placement.

Constraints:

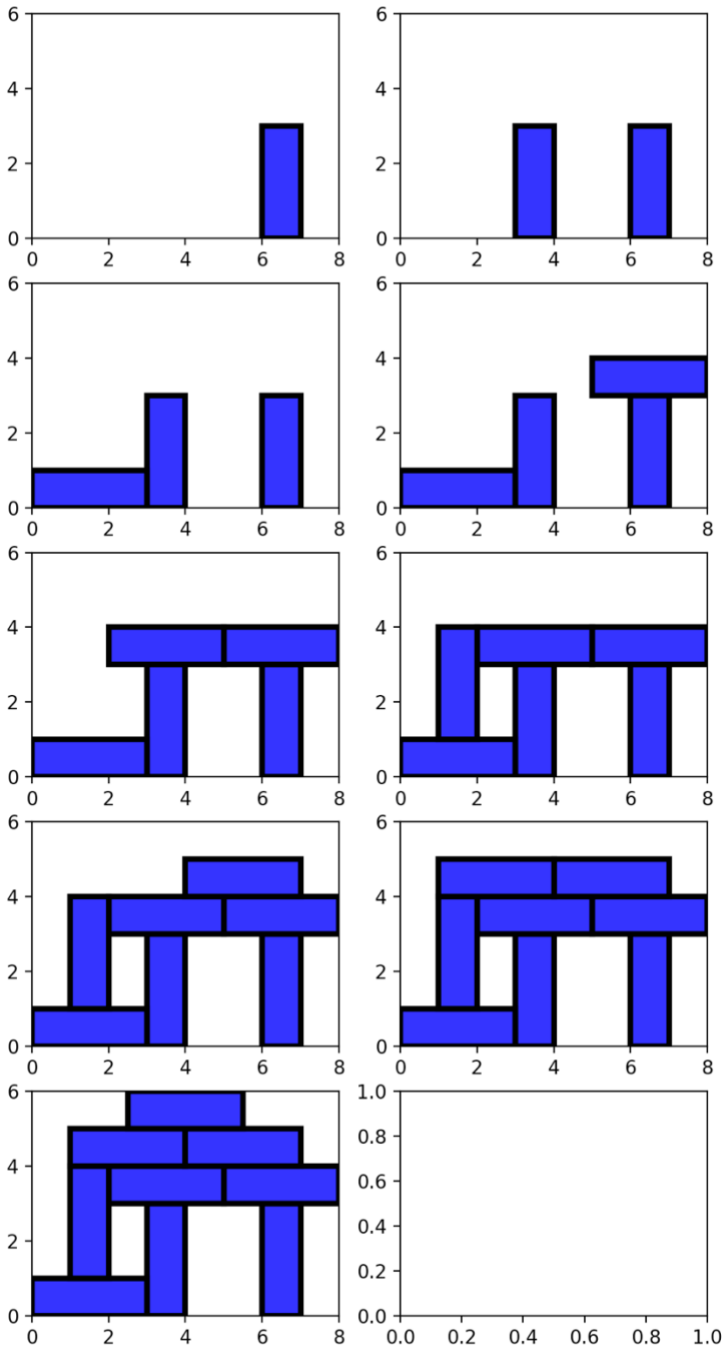
1. If a block is placed vertically, all of its down side must be supported by a block underneath it.
2. For a block to be placed horizontally, 2/3 of its down side or its center must be supported by the block(s) below.
3. Every block must have a unique placement order.

I chose the backtracking algorithm to solve this problem.

b) I implemented the code in Python 3 using the python-constraint module, which must be installed to run the codes. The module can be found in https://pypi.org/project/python-constraint/ This module allows you to create variable with domains and create constraints with custom functions. I used a built-in constraint to make each variable's value unique. For the constraints given in the assignment, I wrote a function and used that to create a constraint. For the first and second structures, I wrote csp_str_1.py and csp_str_2.py, respectively to set up the blocks, constraints and solve the problem. Both programs import block_construction_constraint.py, which includes the functions for the constraint and and the Block class. Implementation details can be found in the comments in the code files.

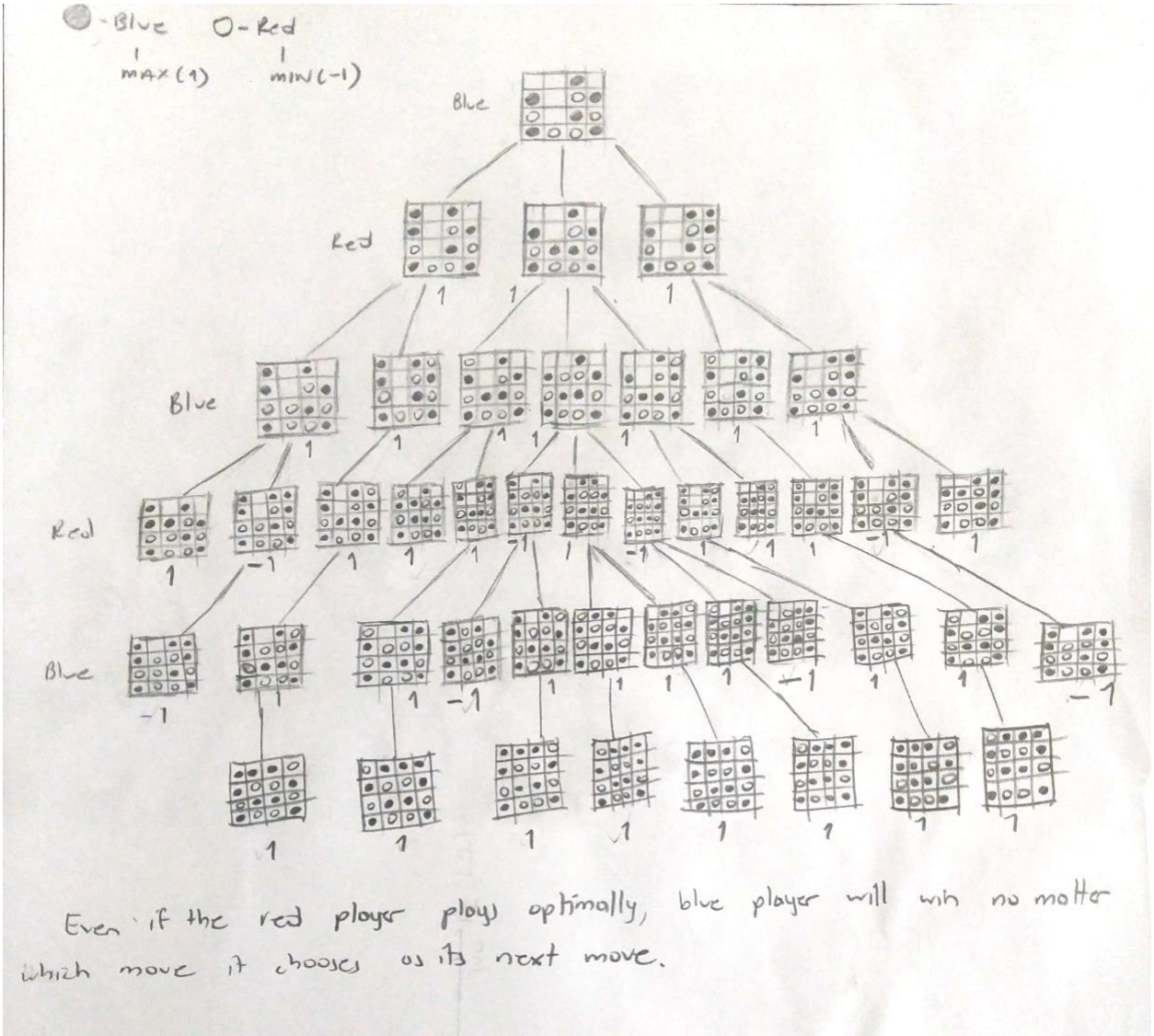
c) When csp_str_1.py is run, which solves the csp for the first structure, it finds and prints 720 different solutions. csp_str_2.py, which solves the csp for the second structure, however, is unable to find any solutions. The algorithm does not get stuck for these two structures.

d) I used one of the solutions found for the first structure to create the visualization below. I used Python 3's matplotlib module. The code is in 3 "visualize.py".



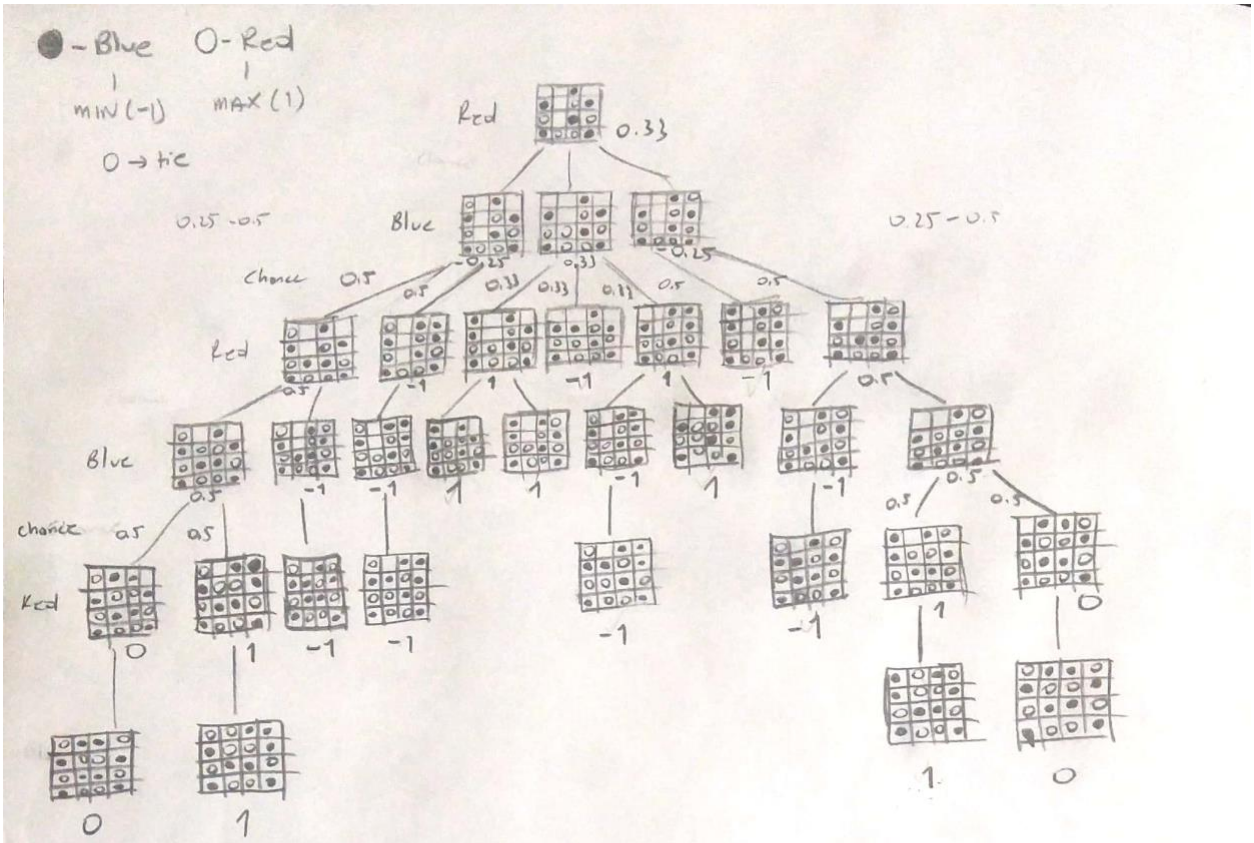
QUESTION – 2

a)

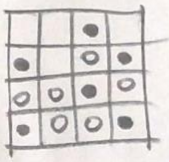


Even if the red player plays optimally, blue player will win no matter which move it chooses as its next move.

b)



If we are the red player and the blue player plays randomly, minimax tree and values will be as above and the optimal next value for red will be:



QUESTION – 3

a) Knowledge Base

Student (Ayşe)
Student (Barış)
Student (Cem)
Club (cinema)
Club (game)
club (literature)
Item (cola)
Item (popcorn)

$\exists x \ Club(x) \wedge Attend(Ayşe, x)$
$\exists x \ Club(x) \wedge Attend(Barış, x)$
$\exists x \ Club(x) \wedge Attend(Cem, x)$

$\forall x \ (Student(x) \wedge Attend(x, literature)) \Rightarrow \neg Like(x, cola)$
$\forall x \ (Student(x) \wedge Attend(x, cinema)) \Rightarrow Like(x, popcorn)$

Like (Ayşe, cola)
¬Like (Ayşe, popcorn)
Like (Barış, popcorn)
Like (Barış, cola)

$\forall x \ (Item(x) \wedge \neg Like(Ayşe, x)) \Rightarrow Like(Cem, x)$

b) Knowledge-base in CNF

1. student (Ayşe)
2. student (Barış)
3. student (Cem)
4. Club (cinema)
5. Club (Game)
6. club (literature)
7. Item (cola)
8. Item (Popcorn)

9. $Club(x) \wedge Attend(Ayşe, x)$
10. $Club(x) \wedge Attend(Barış, x)$
11. $Club(x) \wedge Attend(Cem, x)$
12. $\neg(Student(x) \wedge Attend(x, literature)) \vee \neg Like(x, cola)$
    $\neg Student(x) \vee \neg Attend(x, literature) \vee \neg Like(x, cola)$
13. $\neg Student(x) \vee \neg Attend(x, cinema) \vee Like(x, popcorn)$
14. Like (Ayşe, cola)
15. ¬Like (Ayşe, popcorn)
16. Like (Barış, popcorn)
17. Like (Barış, cola)
18. $\neg Item(x) \vee Like(Ayşe, x) \vee Like(Cem, x)$

- Checking if Ayse is attending the game club (α: Attend(Ayse, game))

19 - $\neg\alpha$: $\neg$ Attend(Ayse, game) SHOW $KB \wedge \neg\alpha$ is unsatisfiable

20 - $\neg$ Attend(Ayse, cinema) by using 17, 13, 15 (x: Ayse)

21 - $\neg$ Attend(Ayse, literature) by using 1, 12, 14 (x: Ayse)

22 - Attend(Ayse, game) by using 5, 9, 20, 21 (x: game)

23 - {} empty clause by 19 and 22.

since $KB \wedge \neg\alpha$ is unsatisfiable, Ayse attends the game club.

- Checking if Barış is attending the game club (α: Attend(Barış, game))

SHOW $KB \wedge \neg\alpha$ is unsatisfiable

19 - $\neg\alpha$: $\neg$ Attend(Barış, game)

20 - $\neg$ Attend(Barış, literature) by using 2, 12, 17

21 - Attend(Barış, cinema) $\vee$ Attend(Barış, game) by using 10, 20

22 - $KB \wedge \neg\alpha$ is satisfiable if Attend(Barış, cinema) $\wedge \neg$ Attend(Barış, game), so Barış may or may not attend the game club.

- Checking if Cem is attending the game club (α: Attend(Cem, game))

SHOW $KB \wedge \neg\alpha$ is unsatisfiable

19 - $\neg\alpha$: $\neg$ Attend(Cem, game)

20 - Like(Cem, popcorn) by using 8, 17, 18 (x: popcorn)

21 - Like(Cem, cola) $\vee \neg$ Like(Cem, cola) by using 16 (no restraint on cola)

22 - Attend(Cem, literature) $\vee \neg$ Attend(Cem, literature) by using 21, 12, 3

23 - Attend(Cem, cinema) $\vee \neg$ Attend(Cem, cinema) by using 3, 13, 20

24 - $KB \wedge \neg\alpha$ is satisfiable if Attend(Cem, literature) $\vee$ Attend(Cem, cinema).

Therefore, Cem may or may not attend the game club.