İTÜ

Homework 2

BLG 435E

CSP, Logic and Games

Question 1 is about CSPs. **This question will be answered in handwriting.**

Question 2 is about the knowledge representation and reasoning. **This question will be answered in handwriting.** In the first part, you need to represent some sentences in first-order logic. In the second part, you are going to use the resolution inference algorithm.

In the third question, you need to implement a solution to a described decision problem. To do so, you are going to implement the **minimax** algorithm that has been covered in the class.

Problem 1 - Constraint Satisfaction Problems (10 points)

You are given the continuous shape drawing problem. In this problem, given a graph (a number of nodes and edges), the objective is finding a sequence of connected edges to draw all of the given edges of the graph exactly once (i.e., drawing without lifting pencil off from the paper and without duplications of edges). A sample graph is given in the figure below.

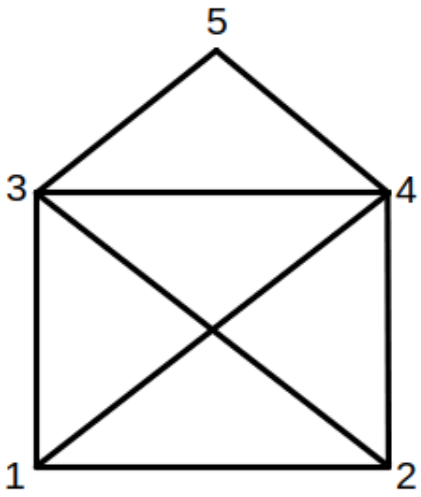

Figure 1: An example graph for continuous shape drawing problem

Formulate this problem as a CSP, state your variables, their domains and the constraints.

Problem 2 - First-Order Logic Representation and Inference via Resolution (30 points)

FOL Representation

Represent the following sentences in first-order logic (FOL). You may define predicates as you like.

- All dogs are animal.
- Not all robots can carry objects.
- Everyone who graduated from high school also graduated from primary school.
- Some students did not take AI course.
- There is only one table.
- There is a teacher who only talks to other teachers that are teaching physics.

FOL and Resolution

Arda, Cihan and Gamze are students in the same university.

English, French, Russian and Turkish belong to language category (Same as "English, French, Russian and Turkish are languages").

Each student in the university speaks Turkish.

Each student in the university speaks at least one of the foreign languages: English, French, Russian.

Fish and hamburger belong to food category.

Classic, jazz and rock belong to music category.

Students who speak French like jazz music and dislike rock music. All students who speak Russian like rock music.

All students who like hamburger speak English. Students who do not like hamburger do not speak English.

Arda likes jazz music and fish but dislikes classic music, rock music and hamburger.

Cihan dislikes whatever music Arda likes, and he likes whatever music Arda dislikes. He likes fish and dislikes hamburger.

Gamze likes fish, hamburger and classic music but dislikes jazz music and rock music.

(**a**) Construct a knowledge-base using the given facts.

(**b**) Use resolution to answer the following queries:

(1) Arda speaks French.

(2) Cihan speaks Russian.

(3) Gamze speaks Russian.

Problem 3 - Optimal Decision in Games (60 points)

Sadık likes to play games. One of his favorite games is a turn-based sudoku game which is modified to be a two-player game. In this game, in each turn, a player picks one of the empty squares on the sudoku grid and writes one of the available numbers for that square. When there is not a valid move left, player who performs the last move wins the game. Since he is very bad at playing games, he likes to use a computer program to gain advantage against his opponents when playing a game. You need to help him by creating a **minimax** agent that finds the best move to play in a given game state. Your program is going to read initial game state from an input file. Some squares on the given sudoku grid is already occupied at the start.

The game has the following rules:

- Standard sudoku rules apply for numbers. That is, you can't place a number if it already exists in the same row/column/box.
- A player can choose any empty square to make a move (if there is a valid number available for that square).
- Player who performed the last move wins the game.

You need to:

(a) Write a program that reads the game grid from an input file.

(b) Implement a basic minimax agent to solve the game.

(c) Implement a minimax agent with alpha-beta pruning to solve the game.

(d) Compare two agents in terms of:

- The number of nodes evaluated in the search tree
- The running time

Some important aspects about your implementation:

- Your program will calculate the result for the first player (player max).
- For the assignment, you are required to implement the standard minimax algorithm, that is, your evaluation function will always evaluate a given position from the perspective of the player max. If the player max is going to win the game from the given state, the utility of that state must be higher compared to utilities of losing states.
- Any other form of the minimax algorithm is not allowed and you will not get any grade.
- Do not use any heuristic functions for state evaluation. Value of states must be calculated exactly by searching down to the game tree and propogating terminal state values appropriately.
- Your program **must** accept a command line argument, input file to read initial game state.

Input File Format

A text file with 9 lines, each line contains 9 space separated numbers. Each number represent the initial number of a square on the sudoku board. If a number is 0, it means that square is initially empty.

Output Format

You will only print a single digit. Print 1 if the first player is going to win, else 2.

Constraints

Normal sudoku constraints apply for placing numbers to the empty squares:

- Each number can only occur once in the same row
- Each number can only occur once in the same column
- Each number can only occur once in one of the 9 3x3 areas.

$1 \leq$ #empty squares in the given grid ≤ 15

Example Input

Example input 0:

4 3 6 1 0 0 9 8 2
5 2 1 3 9 8 4 6 7
8 7 9 2 6 0 1 5 3
6 4 2 8 5 9 3 7 1
9 5 7 4 1 3 8 2 6
0 8 0 6 2 7 0 9 4
2 0 8 7 4 1 6 3 5
7 0 5 9 3 0 2 4 0
3 0 0 5 0 2 0 1 0

4	3	6	1			9	8	2
5	2	1	3	9	8	4	6	7
8	7	9	2	6		1	5	3
6	4	2	8	5	9	3	7	1
9	5	7	4	1	3	8	2	6
	8		6	2	7		9	4
2		8	7	4	1	6	3	5
7		5	9	3		2	4	
3			5		2		1	

Figure 2: Visualization of grid from example input 0.

Implementation and Submission Details

Submit your homework files through Ninova. Please zip and upload all your files using filename BLG435E_HW_2_STUDENTID.zip. You are going to submit:

1. All your code files for implementation of Q3.
2. A pdf file containing answers of Q1 and Q2 (**answered in handwriting**), the required explanations/analyses about Q3 (also specify which grids you performed the analyzes on) and instructions to compile/run your code.

There is no restriction on the programming language.

Your code must be able to compile and run on ITU's Linux Servers without any errors, otherwise your code may not be evaluated (If you can not run your program on ITU servers due to some requirement for your programming language, like JDK for java, just make sure it can run on a Linux machine, and explain this situation in your report).

Important Ethical Considerations: In Q3, your solution can rely on existing minimax algorithms. However, you need to explain how the algorithm work in this problem and perform the requested analyses with sufficient explanations in your report. **Code usage without relevant references will be considered as plagiarism.**

Note that submitted homeworks may also be tested with different initial grids other than those that are given with the assignment files.

Please pay attention to the deadline (date/time) of the homework. **Late submissions will not be accepted, without any exception**.

In case of any questions, feel free to send an e-mail to unlut@itu.edu.tr.