Şahin Akkaya
150170098

Problem 1

Constraint Satisfaction Problem

Variables = { E_{ij} for all edges }

Domains = { x : $0 \leq x < n$(Variables) }

Constraints

- $\forall E_{ij} \; \forall E_{xy}$, $|E_{ij} - E_{xy}| = 1 \Rightarrow i = y \vee j = x$
- $\forall E_{ij} \; \forall E_{xy}$, $i \neq x \wedge j \neq y \Rightarrow E_{ij} \neq E_{xy}$

Explanation: Variables are edges of the given graph.
Domain is range of sequence numbers from 0 to length of variables.
First constraint guarantees that if two edges are assigned 2 consequtive sequence numbers, they must share a common node. (i.e. lifting pencil is not possible)
Second constraint guarantees that each edge is assigned to a different number. Every value in domain is assigned to a single edge.

Problem 2

- FOL Representation
  - All dogs are animal.

    $\forall x \; Dog(x) \Rightarrow Animal(x)$
  - Not all robots can carry objects.

    $\neg \forall x \; Robot(x) \Rightarrow carries(x, object)$
  - Everyone who graduated from high school also graduated from prim. school.

    $\forall x \; Graduated(x, High\ school) \Rightarrow Graduated(x, Primary\ school)$
  - Some students did not take AI course.

    $\exists x \; Student(x) \wedge \neg Take(x, AI)$
  - There is only one table.

    $\exists x \forall y \; Table(x) \wedge Table(y) \Rightarrow x = y$
  - There is a teacher who only talk to other teachers that are teaching Phy.

    $\exists x \forall y \; Teacher(x) \wedge Teacher(y) \wedge Talks(x,y) \Rightarrow Teaches(y, Physics)$

- FOL and Resolution

a) Constructing the knowledge base

- $Student(Arda, X) \wedge Student(Cihan, X) \wedge Student(Gamze, X) \wedge University(U)$
- $Language(English) \wedge Language(French) \wedge Language(Russian) \wedge Language(Turkish)$
- $\forall X \; Student(X, U) \Rightarrow Speaks(X, Turkish)$
- $\forall x \; Student(x, U) \Rightarrow Speaks(X, English) \vee Speaks(X, Russian) \vee Speaks(X, French)$
- $Food(Fish) \wedge Food(Hamburger)$
- $Music(Classic) \wedge Music(Jazz) \wedge Music(Rock)$
- $\forall x \; Student(X, U) \wedge speaks(X, French) \Rightarrow Likes(X, Jazz) \wedge \neg Likes(X, Rock)$
- $\forall x \; Student(X, U) \wedge Speaks(X, Russian) \Rightarrow Likes(X, Rock)$
- $\forall x \; Student(X, U) \wedge Likes(X, Hamburger) \Rightarrow Speaks(X, English)$
- $\forall x \; Student(X, U) \wedge \neg Likes(X, Hamburger) \Rightarrow \neg Speaks(X, English)$
- $Likes(Arda, Jazz) \wedge Likes(Arda, Fish) \wedge \neg Likes(Arda, Classic) \wedge \neg Likes(Arda, Rock) \wedge \neg Likes(Arda, Hamburger)$
- $\forall x \; Music(X) \wedge Likes(Arda, X) \Rightarrow \neg Likes(Cihan, X)$
- $\forall x \; Music(X) \wedge \neg Likes(Arda, X) \Rightarrow Likes(Cihan, X)$
- $Likes(Cihan, Fish) \wedge \neg Likes(Cihan, Hamburger)$
- $Likes(Gamze, Fish) \wedge Likes(Gamze, Hamburger) \wedge Likes(Gamze, Classic) \wedge \neg Likes(Gamze, Jazz) \wedge \neg Likes(Gamze, Rock)$

Problem 3

In this question we are asked to implement a minimax agent to solve a turn-based sudoku game. To do so, I have created a class to keep the game state (grid and turn) and a class to perform required operations by minimax algorithm (successors, terminal_test, utility etc.) The code is documented so it should be self explanatory. The minimax and alpha-beta pruning algorithms are exact copy from lecture slides.

In order to run the code "numpy" should be installed.
python3 main.py input.txt a → alpha beta
python3 main.py input.txt m → minimax

I've tested both algoritms with given text file. Standard minimax couldn't find a solution. Alpha-beta pruning found the solution in 38.58 seconds by evaluating 367 770 nodes.