

FINAL PROJECT

Course: Introduction to Artificial Intelligence

Duration: 07 weeks

I. Formation

- The midterm project is conducted in groups of 04 – 05 students.
Student groups conduct required tasks and submit the project following instructions.

II. Tasks

a) Task 1 (3.0 points): 8x8 Tic-Tac-Toe

- Implement a program that allows users to play Tic-Tac-Toe with the computer on an 8 x 8 board, where whoever has 4 pieces in a row horizontally, vertically, or diagonally wins.
- The game operates on the console screen. The player selects a square by entering its coordinates from the keyboard. Students can update the board interface by erasing and redrawing it after each turn.
- The algorithm used for the computer is alpha-beta pruning.
- Students implement two classes
 - **Problem** → manage the problem
 - **SearchStrategy** with the following method for inferring the next action

alpha_beta_search(p: Problem) → action

- Students can implement supporting classes as needed.

b) Task 2 (3.0 points): N-Queens with CNFs

- Implement a program to place N queens on a board with the size of N x N. N is a positive integer, at least 4, provided by the user.
- Students assign a positive integer to each square on the board representing logical variables, where if the variable holds the value True, the square contains a piece, and if the variable holds the value False, the square is empty.

- Identify and represent the constraints between variables (squares) using logical propositions.
- Convert the propositions from the above step into Conjunctive Normal Form (CNFs).
- Utilize the Glucose3 library to find assignments for the variables and thereby deduce the solution to the problem.
- Students study about Glucose3 following the given link.

<https://pypi.org/project/python-sat/0.1.2dev4/>

- The complete program operates as follows:
 - Prompt the user to input a positive integer.
 - Initialize variables, generate constraints, and convert them into CNFs.
 - Find a set of values satisfying the CNF propositions using Glucose3.
 - If a solution exists, draw the resulting board on the console screen; otherwise, print a message indicating that there is no solution.
- Hints,

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

- For a 4 x 4 board, assign a positive integer to each square, each number is corresponding to a logical variable.
- Given an example constraint “there is a queen in square 1 if and only if square 2, 3, and 4 are empty”.
- The corresponding proposition: $1 \leftrightarrow \neg 2 \wedge \neg 3 \wedge \neg 4$ (\neg is **not**, \wedge is **and**, \leftrightarrow is biconditional).
- Convert the proposition into CNFs

$$p \leftrightarrow q \Leftrightarrow \begin{cases} p \rightarrow q \\ q \rightarrow p \end{cases} \Leftrightarrow \begin{cases} \neg p \vee q \\ \neg q \vee p \end{cases}$$

c) Task 3 (3.0 points): Decision Trees

- Students study the two libraries below to conduct the task.
 - Pandas: <https://www.w3schools.com/python/pandas/default.asp>
 - Decision Tree (Scikit-learn): <https://scikit-learn.org/stable/modules/tree.html>
- Given a data set in **dt_data.csv** with columns as below
 - # → row indices
 - Rank → academic ranks
 - Q1 – Q9 → nine attributes of scores
- Students perform two tasks:
 - Write a program allowing users to input the name of a score attribute. Calculate and display the Entropy (H), Average Entropy (AE), and Information Gain (IG) values of that attribute on the screen.
 - Implement, train, and evaluate a Decision Tree model with the provided dataset. Visualize the structure of the resulting decision tree.

d) Task 4 (1.0 point): Report

- Student groups compose a report.
- **THERE IS NO TEMPLATE. STUDENTS ARRANGE CONTENTS IN A LOGICAL STRUCTURE BY YOURSELVES.**
- The report must include below contents
 - Student list: Student ID, Full name, Email, Assigned tasks, Complete percentage.
 - Briefly present approaches to solve tasks, should make use of pseudo code/diagrams.
 - Avoid embedding raw source code in the presentation.
 - Study topics are introduced briefly with practical examples.
 - Advantages versus disadvantages
 - A table of complete percentages for each task.
 - References are presented in IEEE format.

- **Format requirements:** avoid using dark background/colorful shapes, students ensure contents are clear enough when printing in grayscale.

III. Submission

- Create a folder whose name is in the format
final_<Group ID>
 - **source:** source code files
 - **report.pdf:** report of the project
- Students maintain outputs of all cells in .ipynb files.
- Compress the folder into a zip file and submit by the deadline.

IV. Policy

- **Student groups submitting late get 0.0 points for each member.**
- **Copying source code on the internet/other students, sharing your work with other groups, etc. cause 0.0 points for all related groups.**
- **If there exist any signs of illegal copying or sharing of the assignment, then extra interviews are conducted to verify student groups' work.**

-- THE END --