ACI Assignment 2
Question 2 A

Part 1: Constraint Satisfaction Problem (CSP):

Problem statement: Time Tabler Agent

There are 6 teaching staff members in the Department of Computer Science: one Head Of the Department (HOD), two Associate Professors and three Assistant Professors. Classes can be delivered within restricted 5 sessions per working day from Monday to Friday. The duration of one session can be either 1 hour or 50 minutes. Every day classes may be scheduled to start at 9AM with an allotted duration of one hour per class in the forenoon session. At 12 Noon there is a lunch-break spanning 1.15 hours. Post lunch only two classes could be scheduled with an allowable duration of 50-minutes each. Since the Head of the department is expected to execute managerial activities, the first hour of the day shouldn't be allotted for the HOD. Similarly the two associate professors shouldn't be assigned classes in the second hour since they are allotted monitoring activity during that period. HOD can be assigned 11 hours per week. Both Associate and Assistant professors can be assigned 12 hours each.

There are 4 years student sections (1st, 2nd, 3rd and 4th year) with a total of 20 courses to be scheduled. Out of the 20 courses there are few core courses that need to be allotted 4 hours each. The distribution of core courses are as follows:

1st year → 1 core course in computer science subject
2nd , 3rd & 4th year →2 core courses each in computer science subjects

In addition to above, the first-year students have one Mathematics and one English course that are handled by other department staff (included in the 20 courses). Out of the remaining 11 courses, 5 are elective theory courses with 4 hours each in class .The rest are skill-based electives in the lab. Four of the lab courses require an allotment of 3 hours each and the rest of the two lab courses need 2 hours each.

Design a CSP agent for faculty allocation and time table scheduling. Assume that all the faculty can instruct any course in their department.

a. Draw the constraint satisfaction graph to represent this problem.

b. Implement Python code for the design under part a, using Arc Consistency (AC3 Algorithm)

c. Print the Faculty wise allotment and year wise class scheduling details.

(Part 2 is in next page)

Part 2: Logic Design and Inferencing:

Problem Statement: Haberman's Survival:

The shared excel sheet contains Haberman's Survival Data Set with below attributes, on survival of patients who had undergone surgery for breast cancer.
1. Age: Age of patient at time of operation (numerical)
2. Op_Year : Patient's year of operation (numerical)
3. axil_nodes: Number of positive axillary nodes detected (numerical)
4. Surv_status: Survival status (class attribute)
1 = the patient survived 5 years or longer
2 = the patient died within 5 year

Python code for Decision Tree is already shared in CANVAS. Use that code snippet for the given dataset. Obtain the output of the Decision Tree function and use it as the input for the logic design in Prolog.

<u>Evaluations will be based on the following.</u> [All below has EQUAL WEIGHTAGE]

1. Illustration of Constraint Graph with justification to the design of <VARIABLES, DOMAIN,CONSTRAINTS> .
2. Encode & Design of Variables/Domain/Constraints of the given problem in python.
3. Implementation of Heuristics for the problem in python.
4. For CSP each assignment & backtrack, corresponding (variable,value) and/or conflicting constraints should be legibly printed during execution.
5. Code the rules into a Prolog Knowledge base.
6. Dynamically get Patient information as input from the user and classify whether the patient survived 5 years and longer or the patient died within 5 years.

NOTE:

- You are provided with the python notebook template which stipulates the structure of code and documentation. Use well intended python code.
- Use a separate MS word or PDF document for explaining the theory part. Do not include the theory part in the executables.
- The implemented executables must be completely original and executable.
- Please keep your work (code, documentation) confidential. If your code is found to be plagiarized, you will be penalized severely. Parties involved in the copy will be considered equal partners and will be penalized severely.**Collaboration among different group members will also be considered as plagiarism.**