Birla Institute of Technology & Science-Pilani, Hyderabad Campus First Semester 2024-2025

Principles of Programming languages (CS F301)

Due Date for Assignment 2: 25-11-2024

Max marks:20

This assignment will give you a flavor of Functional and Logic programming paradigms. You have a total of 4 questions to code. Two are using Prolog, and 2 using Scheme. This is a group assignment, and the group from Assignment 1 will continue. Please read the instructions (uploaded separately) carefully before proceeding to the questions.

Prolog questions:

Q1: Department Assignment Puzzle

You are tasked with solving a logic puzzle to determine the department, favorite language, and project preference of four friends. Each friend has a unique department, favorite language, and project preference. Your goal is to use the given clues to match each friend to their department, language, and project.

Attributes: Each friend has three attributes:

- **Department**: research, development, design, testing
- **Favorite Language**: python, java, c++, javascript
- **Project Preference**: AI, web, embedded, data

Task: Determine the correct department, favorite language, and project preference for each friend (Alex, Ben, Carla, and Dana) based on the provided clues.

Constraints:

- Each friend has a unique value for each attribute.
- Use the clues provided to derive each friend's department, favorite language, and project preference.
- **Avoid hard-coding** the solution; instead, use logical deduction based on the clues.

Clues:

1. The friend who prefers the AI project does not work in research and is not Dana.
2. Alex does not work in development and does not prefer C++ as a language.
3. The friend working in the testing department likes Python as their favorite language.
4. Ben prefers web projects and does not work in testing.
5. The friend whose favorite language is Java works in the design department.
6. Carla's project preference is not embedded, and her department is not development.

7. Dana’s favorite language is not Java, but she works in either research or testing.
8. The friend who prefers data projects works in research and does not use JavaScript.

Example Output: For each friend, provide their department, favorite language, and project preference in the following format:

Alex = [research, python, AI]

Ben = [development, java, web]

Carla = [design, c++, embedded]

Dana = [testing, javascript, data]

Note: This is only a template and may not represent the correct solution.

Expected Solution: Assign each friend to the correct department, favorite language, and project preference to satisfy all given clues. The output should be ordered as follows: Alex, Ben, Carla, Dana.

Please note that Prolog may generate multiple valid solutions based on different variable orderings. Any correct solution that satisfies all given clues and constraints will be accepted. You need to print any one correct solution.

Code Template: You are required to PRINT the solution by writing the logic inside the main predicate. Please use the following template:
https://github.com/aditya4107/PPLAssignment2Templates/blob/main/id_q1.pl

NOTE:

- Only modify the main predicate; all other parts of your code should remain unchanged.

Q2: Form a Valid Arithmetic Equation:

Given a list of integers, your task is to insert arithmetic operators (+, -, *, /, =) between the numbers in such a way that the result is a correct equation. You may use parentheses to enforce the desired order of operations. The goal is to form a valid equation from the given list of numbers, using each number exactly once and in the order they appear.

Input:

- A list of integers representing the numbers.

Output:

- Any valid equation formed by inserting arithmetic operators between the numbers.

Example:

- For the list [4, 2, 5, 3], a possible valid equation is "4*2=5+3" (return it as a string).

Constraints:

- All numbers must appear in the equation in the same order as given in the list.
- The operators to be used are +, -, *, /, and =.
- Division must result in an integer.
- Assume that a valid solution exists for the given input.
- The = operator can be used only once in the equation.

Code Template: You are required to RETURN the solution by writing the logic inside the generate_equation predicate. Please use the following template: https://github.com/aditya4107/PPLAssignment2Templates/blob/main/id_q2.pl

Note:

- In Prolog, "return" means to unify the answer with the Equation variable provided in the template.
- Only modify the generate_equation predicate; all other parts of your code should remain unchanged.

Scheme questions:

Q3: Graph Traversal using Depth-First Search (DFS)

Write a Scheme program that performs Depth-First Search (DFS) on a directed graph. The graph is represented as an adjacency list, and your program should traverse the graph starting from a given node, visiting all reachable nodes.

You will represent the graph using an adjacency list, where each node points to a list of nodes it is connected to. The program should implement a function that takes the graph and a starting node as input and returns a list of nodes visited during the DFS traversal.

Supported Operations:

1. Graph Representation: The graph will be represented as a list of pairs, where each pair represents a node and its list of adjacent nodes.
    - Example graph: ((a (b c)) (b (d)) (c (d e)) (d ()) (e ()))
2. DFS Traversal: The program should implement the DFS algorithm to traverse the graph and return the list of visited nodes in the order they are visited.
3. Handling Cycles: The graph may contain cycles, so the program should ensure that no node is visited more than once.

Constraints:

- The graph is directed and may have cycles or disconnected components.
- You can assume the graph is non-empty and has at least one node.

- Any valid DFS traversal order is accepted
- Each node is represented by a single English character.

Input Format:

- The input is an adjacency list representing the graph and a starting node for the DFS traversal.

Output Format:

- The output should be a list of nodes in the order they were visited during the DFS traversal.

Example Input:

- ((a (b c)) (b (d)) (c (d e)) (d ()) (e ()))
- Starting node: a

Example Output:

- (a b d c e)

Code Template: You are required to PRINT the solution by implementing the logic in the print-dfs-result function. Please use this template: https://github.com/aditya4107/PPLAssignment2Templates/blob/main/id_q3.rkt

Note:

- Only modify the print-dfs-result function; all other parts of your code should remain unchanged.

Q4. Conditionally Insert an Element into a Nested List

Write a Scheme program that defines a function insert-if-short. The function takes three inputs: an element x, a list of lists lst, and a number n. The function should insert x at the beginning of each sublist within lst only if the length of that sublist is less than n. If a sublist has a length equal to or greater than n, leave it unchanged.

The insertion should occur at all levels of nesting within lst.

Requirements:

- The function should handle arbitrary levels of nested lists.
- Maintain the original order of elements in each list after adding x.
- Do not use built-in functions for list length or manipulation; instead, implement your own recursive logic.

Constraints:

- If lst is an empty list, return an empty list.
- If lst contains any empty lists, consider those for the length criteria.
- Each node is represented by a digit.

Input Format:

Your program should prompt the user for input in the following format:

- An element x (a Scheme atom).
- A list of lists lst (entered in the standard Scheme list format).
- A number n (an integer).

Output Format:

Your program should output the modified list after the function has processed the input.

Example Input:

- x = 42
- lst = ((1 2) (3 4 5) (6) ())
- n = 3

Example Output:

- ((42 1 2) (3 4 5) (42 6) ())

Code Template: You are required to PRINT the solution by implementing the logic in the print-insert-result function. Please use this template: https://github.com/aditya4107/PPLAssignment2Templates/blob/main/id_q4.rkt

Note:

- Only modify the print-insert-result function; all other parts of your code should remain unchanged.

Incase you have any queries please contact the TA

Aditya Kumar Sharma f20213112@hyderabad.bits-pilani.ac.in