Birla Institute of Technology & Science, Pilani
1st Semester 2022-2023
CS F214 - Logic in Computer Science
Assignment

Instructions

- This assignment contains two questions on Prolog. The first question carries 3 marks and the second carries 7 marks (total 10 marks).
- You must use SWI prolog for writing your programs. Code written in any other flavor is unacceptable.
- Your code must handle all possible boundary cases.
- Marks shall be given for the logic part (50%) and successful execution of the test cases (50%).
- You must submit at least 2 working test cases along with your program.
- There must be a readme file giving instructions on how to run your program. It should also contain all assumptions made explicitly for each problem.
- Put your files in separate directories for each problem. Create a single zip file consisting of both directories and upload it on Nalanda before **11:59 PM** on **24 November 2022**. The submission link will only accept one submission. So, make sure that you are uploading the right submission on time.
- Any kind of plagiarism shall not be tolerated.
- Assignment submission has to be made on the link given on Nalanda.
- **Only one of the team members should submit the assignment. Others need not.**

Problem 1 **(3M)**

Given a weighted undirected graph G = (V, E). The weight on an edge (v_1, v_2) represents the distance between the vertices v_1 and v_2 {$v_i \in V$; and (v_i, v_j) $\in$ E}. Given a set of weights of edges in the form of prolog clauses:

```
weight(p, q, 20).
weight(q, r, 5).
```

Write a prolog program to check if there exists a path between two given vertices. Print the path and its length, if it exists. For example:

```
?- findPath(p, r, Path, L).
```

the prolog interpreter should print

```
Path = [p, q, r]
L = 25
```

Note that the user should print an alternative path if it exists on each press of a semi colon and print **no** if no path exists.

Your program should avoid traversing through cycles, if any.

Marking Scheme:

50% for logic and 50% for code execution. Marks will be considered only for the logic if code is not executing as intended by the problem statement.

Problem 2 **(1+2+2+2 = 7M)**

Write a Prolog predicate for each of the following operations on a list:

a) `sublist(X,Y)` : true if list `X` is a sublist of list `Y`. A sublist is the original list, in the same order, but with no/some elements removed.
b) `has_triplicate(X)` : true if list `X` contains at least three copies of an element. It also prints the element which is triplicated.
c) `remove_nth(N,X,Y)` : prints list `Y` which is the list `X` with its N^{th} element removed. If `X` does not have an N^{th} element then the predicate should fail. Assume that `N > 0`.
d) `remove_every_other(X,Y)` : prints a list `Y` which is the list `X` with every other element removed (the two lists should have the same first element).

Note: Each predicate should be able to print the alternative solutions if they exist.

====== End of the document ======