CSE 643: Assignment 1

Abhimanyu Gupta - 2019226

Brief about working:

The prolog program takes a number of inputs based on user choices and returns back a number of electives suitable for the user.

Steps to run the program:

1. Load the program using consult or [code] in prolog terminal.
2. Run the advisory system using adviseElectiveCourses command.
3. Enter the department you are looking for from cse/mth.
4. Next, you would be asked a series of questions to determine your field of interest.
5. Quickly after it you would have to answer about electives you already did (using y/n).
6. At last, the program would return a list of electives suitable for you.

Working Examples:

1)

```
?- adviseElectiveCourses.
Welcome to Course Advisory System
Follow the directions to find the right elective for you
Select the department for which you want the course (cse/mth)
|: mth.
Are you interested in Pure Mathematics (y/n)
|: y.
Are you interested in Optimization (y/n)
|: y.
Are you interested in Probability and Statistics (y/n)
|: n.
Have you completed Real Analysis 1 course (y/n)?
|: y.
Have you completed Real Analysis 2 course (y/n)?
|: n.
Have you completed Abstract Alegbra 1 course (y/n)?
|: y.
Have you completed Abstract Alegbra 2 course (y/n)?
|: n.
Have you completed Discrete Mathematics course (y/n)?
|: n.
Have you completed Linear Optimisation course (y/n)?
|: y.
Have you completed Multivariate Calculus course (y/n)?
|: n.
Have you completed Graph Theory course (y/n)?
|: y.
Suitable Elective(s) for you in mth department:
Real Analysis 2
Abstract Alegbra 2
Discrete Mathematics
Multivariate Calculus
true .
```

2)

```
?- adviseElectiveCourses.
Welcome to Course Advisory System
Follow the directions to find the right elective for you
Select the department for which you want the course (cse/mth)
|: cse.
Are you interested in Security (y/n)
|: y.
Are you interested in Artificial Intelligence (y/n)
|: y.
Are you interested in Algorithms (y/n)
|: n.
Have you completed Computer Networks course (y/n)?
|: y.
Have you completed Foundation of Computer Security course (y/n)?
|: n.
Have you completed Network Security course (y/n)?
|: n.
Have you completed Artificial Intelligence course (y/n)?
|: y.
Have you completed Machine Learning course (y/n)?
|: y.
Have you completed Deep Learning course (y/n)?
|: n.
Have you completed Statistical Machine Learning course (y/n)?
|: n.
Suitable Elective(s) for you in cse department:
Foundation of Computer Security
Network Security
Deep Learning
Statistical Machine Learning
true .
```

Code snippets:

```
% Abhimanyu Gupta - 2019226
% CSE643 - Artificial Intelligence - Assignment 1

adviseElectiveCourses :-
    clear_all,

    write('Welcome to Course Advisory System\n'),
    write('Follow the directions to find the right elective for you\n'),

    write('Select the department for which you want the course (cse/mth)\n'),

    read(Department),
    (
        (Department = 'cse' -> check_cse_domains_interest);
        (Department = 'mth' -> check_mth_domains_interest);
        write('Invalid department!!, program exiting!\n'),fail
    ),

    forall(interested(X), check_available_electives(X)),

    recommend_courses(Courses),
    (
        is_empty(Courses) -> write('Sorry, nothing to recommend at the moment\n');
        format('Suitable Elective(s) for you in ~w department:\n', [Department]), show_list(Courses)
    ),

    clear_all.


:- dynamic interested/1, course_done/1, course_not_done/1, is_empty/1, recommend_course/1.

clear_all :-
    retractall(interested(_)),
    retractall(course_done(_)),
    retractall(course_not_done(_)),
    retractall(recommend_course(_)).


recommend_courses([Head | Tail]) :-
    retract(recommend_course(Head)), recommend_courses(Tail).
recommend_courses([]).

show_list([Head | Tail]) :-
    format('~w~n', [Head]), show_list(Tail).
show_list([]).

isempty([]).
```

```
check_course(Q) :-
    (
        course_done(Q) -> true;
        (
            course_not_done(Q) -> fail;
            ask_question_to_check(Q)
        )
    ).

ask_question_to_check(Q) :-
    format('Have you completed ~w course (y/n)?\n', [Q]),
    read(A),
    (
        (A = 'y') -> assert(course_done(Q));
        assert(course_not_done(Q))
    ).


check_cse_domains_interest :-
    write('Are you interested in Security (y/n)\n'),
    read(S),
    (
        (S = 'y') -> assert(interested(s));
        !
    ),

    write('Are you interested in Artificial Intelligence (y/n)\n'),
    read(AI),
    (
        (AI = 'y') -> assert(interested(ai));
        !
    ),

    write('Are you interested in Algorithms (y/n)\n'),
    read(A),
    (
        (A = 'y') -> assert(interested(a));
        !
    ).

check_mth_domains_interest :-
    write('Are you interested in Pure Mathematics (y/n)\n'),
    read(PM),
    (
        (PM = 'y') -> assert(interested(pm));
        !
    ),

    write('Are you interested in Optimization (y/n)\n'),
    read(O),
    (
        (O = 'y') -> assert(interested(o));
        !
    ),
```

```
    write('Are you interested in Probability and Statistics (y/n)\n'),
    read(PS),
    (
        (PS = 'y') -> assert(interested(ps));
        !
    ).

check_available_electives(s) :-

    check_course('Computer Networks'),
    (
        course_not_done('Computer Networks') -> assert(recommend_course('Computer Networks'));
        (
            check_course('Foundation of Computer Security'),
            (
                course_not_done('Foundation of Computer Security') -> assert(recommend_course('Foundation of Computer Security'));
                !
            )
        )
    ),

    check_course('Computer Networks'),
    (
        course_not_done('Computer Networks') -> assert(recommend_course('Computer Networks'));
        (
            check_course('Network Security'),
            (
                course_not_done('Network Security') -> assert(recommend_course('Network Security'));
                !
            )
        )
    ),

    check_course('Foundation of Computer Security'),
    (
        course_not_done('Foundation of Computer Security') -> assert(recommend_course('Foundation of Computer Security'));
        (
            check_course('Security Engineering'),
            (
                course_not_done('Security Engineering') -> assert(recommend_course('Security Engineering'));
                !
            )
        )
    ).

check_available_electives(ai) :-

    check_course('Artificial Intelligence'),
    (
        course_not_done('Artificial Intelligence') -> assert(recommend_course('Artificial Intelligence'));
        !
    ),

    check_course('Machine Learning')
```

```
    check_course('Machine Learning'),
    (
        course_not_done('Machine Learning') -> assert(recommend_course('Machine Learning'));
        (
            check_course('Deep Learning'),
            (
                course_not_done('Deep Learning') -> assert(recommend_course('Deep Learning'));
                !
            )
        )
    ),

    check_course('Statistical Machine Learning'),
    (
        course_not_done('Statistical Machine Learning') -> assert(recommend_course('Statistical Machine Learning'));
        !
    ).


check_available_electives(a) :-

    check_course('Algorithm Design and Analysis'),
    (
        course_not_done('Algorithm Design and Analysis') -> assert(recommend_course('Algorithm Design and Analysis'));
        (
            check_course('Modern Algorithm and Design'),
            (
                course_not_done('Modern Algorithm and Design') -> assert(recommend_course('Modern Algorithm and Design'));
                !
            )
        )
    ),

    check_course('Algorithm Design and Analysis'),
    (
        course_not_done('Algorithm Design and Analysis') -> assert(recommend_course('Algorithm Design and Analysis'));
        (
            check_course('Randomised Algorithm'),
            (
                course_not_done('Randomised Algorithm') -> assert(recommend_course('Randomised Algorithm'));
                !
            )
        )
    ),

    check_course('Algorithm Design and Analysis'),
    (
        course_not_done('Algorithm Design and Analysis') -> assert(recommend_course('Algorithm Design and Analysis'));
        (
            check_course('Introduction to Graduate Algorithm'),
            (
                course_not_done('Introduction to Graduate Algorithm') -> assert(recommend_course('Introduction to Graduate Algorithm'));
```

```
    check_course('Algorithm Design and Analysis'),
    (
        course_not_done('Algorithm Design and Analysis') -> assert(recommend_course('Algorithm Design and Analysis'));
        (
            check_course('Introduction to Graduate Algorithm'),
            (
                course_not_done('Introduction to Graduate Algorithm') -> assert(recommend_course('Introduction to Graduate Algorithm'));
                !
            )
        )
    ).


check_available_electives(pm) :-

    check_course('Real Analysis 1'),
    (
        course_not_done('Real Analysis 1') -> assert(recommend_course('Real Analysis 1'));
        (
            check_course('Real Analysis 2'),
            (
                course_not_done('Real Analysis 2') -> assert(recommend_course('Real Analysis 2'));
                !
            )
        )
    ),

    check_course('Abstract Alegbra 1'),
    (
        course_not_done('Abstract Alegbra 1') -> assert(recommend_course('Abstract Alegbra 1'));
        (
            check_course('Abstract Alegbra 2'),
            (
                course_not_done('Abstract Alegbra 2') -> assert(recommend_course('Abstract Alegbra 2'));
                !
            )
        )
    ),

    check_course('Discrete Mathematics'),
    (
        course_not_done('Discrete Mathematics') -> assert(recommend_course('Discrete Mathematics'));
        (
            check_course('Graph Theory'),
            (
                course_not_done('Graph Theory') -> assert(recommend_course('Graph Theory'));
                !
            )
        )
    ).

check_available_electives(o) :-

```

```
            course_not_done('Linear Optimisation') -> assert(recommend_course('Linear Optimisation'));
            !
        ),

        check_course('Multivariate Calculus'),
        (
            course_not_done('Multivariate Calculus') -> assert(recommend_course('Multivariate Calculus'));
            (
                check_course('Convex Optimisation'),
                (
                    course_not_done('Convex Optimisation') -> assert(recommend_course('Convex Optimisation'));
                    !
                )
            )
        ),

        check_course('Graph Theory'),
        (
            course_not_done('Graph Theory') -> assert(recommend_course('Graph Theory'));
            !
        ).

    check_available_electives(ps) :-

        check_course('Probability & Statistics'),
        (
            course_not_done('Probability & Statistics') -> assert(recommend_course('Probability & Statistics'));
            (
                check_course('Stochastic Processes and Applications'),
                (
                    course_not_done('Stochastic Processes and Applications') -> assert(recommend_course('Stochastic Processes and Applications'));
                    !
                )
            )
        ),

        check_course('Probability & Statistics'),
        (
            course_not_done('Probability & Statistics') -> assert(recommend_course('Probability & Statistics'));
            (
                check_course('Statistical Inference'),
                (
                    course_not_done('Statistical Inference') -> assert(recommend_course('Statistical Inference'));
                    !
                )
            )
        ).


    check_available_electives(X) :-
        true.
```