

COMP 3031 Assignment 3

Logic Programming

Fall 2017

Due: 5PM on Nov 30 Thursday

Instructions

- There are five problems in this assignment. Each problem counts for two points.
- Write your prolog program according to the definition of the problem, with the same predicate name and number of arguments as specified. Write all the solutions in a single file named “ass3.pl”. You can use any helper predications, including built-in predicates, e.g., member/2, var/1, in the solutions.
- Submit your code through Canvas.
- No late submissions will be accepted.
- Your submission will be run on a lab 2 machine with the following command:
“?- [ass3].”.

Please make sure your submission is executable. If it is not, a significant number of points will be deducted.

For the five problems in this assignment, we define a relation `enroll(Stu, Courses)` and a relation `teach(Prof, Courses)`. In `enroll(Stu, Courses)`, `Stu` is a student ID, and `Courses` is a list of course IDs the student `Stu` enrolls in. In `teach(Prof, Courses)`, `Prof` is a professor id, and `Courses` is a list of course IDs taught by professor `Prof`. The course IDs in each enroll fact and each course fact are distinct.

Assume all input data are correct.

Examples are based on the following database:

```
/* The database of enroll and teach facts */
enroll(1701, [c01, c10]).
enroll(1602, [c21]).
enroll(1711, [c01, c21, c10]).
enroll(1501, []).
teach(p01, [c01, c21]).
teach(p02, [c23]).
teach(p03, [c10]).
teach(p04, []).
```

1. Listing all the professors of a student's enrollments

Define a relation `prof_ids(Stu, L)` that specifies a list `L` of **all** the professor IDs of a student's enrollment. The order of professor IDs in `L` corresponds to the order of courses taught by these professors in `enroll(Stu, Courses)`.

Examples:

```
?- prof_ids(1701, L).
L = [p01, p03].

?- prof_ids(1501, L).
L = [].

?- prof_ids(Stu, [p01]).
Stu = 1602 ;
false.

?- prof_ids(Stu, [p01, p02]).
false.

?- prof_ids(Stu, [p01, p03]).
Stu = 1701 ;
false.

?- prof_ids(Stu, [p01, p03, p04]).
false.

?- prof_ids(Stu, L).
Stu = 1701,
L = [p01, p03] ;
Stu = 1602,
L = [p01] ;
Stu = 1711,
L = [p01, p01, p03] ;
Stu = 1501,
L = [].
```

2. Listing common enrollments of two students

Write a relation `common_enroll(X, Y, L)` that specifies a list `L` of courses that two **different** students `X` and `Y` both enroll in.

Examples:

```
?- common_enroll(1701, 1602, L).  
L = [].  
  
?- common_enroll(1701, 1501, L).  
L = [].  
  
?- common_enroll(1701, 1711, L).  
L = [c01, c10].  
  
?- common_enroll(1701, 1711, []).  
true.  
  
?- common_enroll(1701, 1711, [c01]).  
true.  
  
?- common_enroll(1701, 1711, [c01, c23]).  
false.  
  
?- common_enroll(1701, 1711, [c10, c01]).  
true.  
  
?- common_enroll(1701, Y, L).  
Y = 1602,  
L = [] ;  
Y = 1711,  
L = [c01, c10] ;  
Y = 1501,  
L = [] ;  
false.  
  
?- common_enroll(X, 1711, L).  
X = 1701,  
L = [c01, c10] ;  
X = 1602,  
L = [c21] ;  
X = 1501,  
L = [] ;  
false.
```

3. Listing distinct students that enroll in a professor's course(s)

Write a relation `student_list(Prof, L)` that specifies a list `L` of distinct students that enroll in at least one course taught by the professor `Prof`.

Examples:

```
?- student_list(p01, L).
L = [1711, 1602, 1701].

?- student_list(p02, L).
L = [].

?- student_list(p04, L).
L = [].

?- student_list(p01, []).
true.

?- student_list(p01, [1701, 1711]).
true.

?- student_list(p01, [1501, 1711]).
false.

?- student_list(X, L).
X = p01,
L = [1711, 1602, 1701] ;
X = p02,
L = [] ;
X = p03,
L = [1711, 1701] ;
X = p04,
L = [].
```

4. Listing courses taught by the professors

Write a relation `course_list(L)` that specifies a list `L` of courses taught by the professors in the teach facts.

Examples:

```
?- course_list(L).
L = [c01, c21, c23, c10].

?- course_list([]).
true.
```

```

?- course_list([c01]).
true.

?- course_list([c23, c21]).
true.

?- course_list([c23, c02]).
false.

?- course_list([c01, c21, c23, c10]).
true.

```

5. Counting the number of students that enroll in a course

Write a relation `count_students(C, N)` that counts the number of students that enroll in the course `C`.

Examples:

```

?- count_students(c01, N).
N = 2.

?- count_students(c01, 2).
true.

?- count_students(c01, 0).
false.

?- count_students(c23, N).
N = 0.

?- count_students(c10, N).
N = 2.

?- count_students(X, 2).
X = c01 ;
X = c21 ;
X = c10.

?- count_students(X, 0).
X = c23 ;
false.

?- count_students(X, 1).

```

false.

?- count_students(X, N).

X = c01,

N = 2 ;

X = c21,

N = 2 ;

X = c23,

N = 0 ;

X = c10,

N = 2.