Homework 12

April 18, 2014

Instruction

Write the answers to all questions in one text file of the name `hwk12.pl`.

Prolog programming

In this homework, you will implement Prolog program to generate class schedules with the least number of excess seats in assigned classrooms. Make sure that your program can be loaded into Prolog console and yield the correct output. I have provided a template file with some facts about class rooms, instructors' teaching assignment, the course requirements, and the lists of class starting time and end time.

You should implement the following predicates:

1. Define the predicate `excess_capacity_one(schedule(C, R, _), X)` so that X is the number of excess seats in room R for cource C.
2. Define the predicate `excess_capacity(L, X)` to find the total excess seats in the list L of schedules.
3. Define the predicate `minc(L, SofarC, Sofar, X)` so that X is the schedule with minimum excess seats in the list L of schedules while `SofarC` is the minimum so far and `SoFar` is the schedule associated with `SofarC`.
4. Define the predicate `min_excess_capacity(L, X)` so that X is the best schedule in L with the least number of excess seats.
5. Define the predicate `best_schedule(X, C)` to return the the schedule X with the least number of excess seats C.

Make sure the predicates can be used in the following test cases:

```
?- excess_capacity_one(schedule(c1, r1, _), C).
C = 0

?- excess_capacity_one(schedule(c3, r1, _), C).
```

```
C = 5

?- excess_capacity([schedule(c1, r3, _), schedule(c2, r3, _),
                    schedule(c3, r1, _), schedule(c4, r1, _),
                    schedule(c5, r3, _), schedule(c6, r1, _)],
                   C).

C = 15.

?- minc([[schedule(c1, r3, _), schedule(c2, r3, _), schedule(c3, r1, _)],
         [schedule(c4, r1, _), schedule(c5, r3, _), schedule(c6, r1, _)]],
        30, _, X).

X = [schedule(c4, r1, _), schedule(c5, r3, _), schedule(c6, r1, _)] ;
false

?- min_excess_capacity(
         [[schedule(c1, r3, _), schedule(c2, r3, _), schedule(c3, r1, _)],
          [schedule(c4, r1, _), schedule(c5, r3, _), schedule(c6, r1, _)]],
        X).

X = [schedule(c4, r1, _), schedule(c5, r3, _), schedule(c6, r1, _)].

?- best_schedule(X, C).
X = [schedule(c1, r1, between(14, 15)), schedule(c2, r3, between(15, 16.5)),
     schedule(c3, r1, between(11, 12.5)), schedule(c4, r3, between(14, 15)),
     schedule(c5, r3, between(11, 12.5)), schedule(c6, r1, between(15, 16.5))],
C = 15 ;

X = [schedule(c1, r3, between(14, 15)), schedule(c2, r3, between(15, 16.5)),
     schedule(c3, r1, between(11, 12.5)), schedule(c4, r1, between(14, 15)),
     schedule(c5, r3, between(11, 12.5)), schedule(c6, r1, between(15, 16.5))],
C = 15 ;

...

?- num_best_schedule(N).
N = 8.
```

Note that when you define the predicate `minc` you have to use $\geq$ or $\leq$ to compare `SofarC` with other excess capacity. If you just use $>$ or $<$, you may just get one best schedule instead of all schedules with the least excess capacity.