COMP 3031 Assignment 3

Logic Programming

Fall 2019

Due: 5PM on Nov 29 Friday

Instructions

- There are five problems in this assignment. Each problem counts for two points.
- Write your prolog program according to the problem definition, 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 predicates, including built-in predicates available on the lab computers, but *not* any external modules that require downloading.
- 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.

An H × W matrix consists of H rows and W columns. In this assignment, we use a list of H lists each of which consists of W numbers to represent an H × W matrix. You can assume all input are valid in the PROLOG interpreter.

1. Get

Define a relation `mat_get(M, I, J, R)` where R is the number located at the i-th row and j-th column of matrix M.

Examples:

?- mat_get([[1,2],[3,4]],0,1,R).
R = 2.
?- mat_get([[1,2],[3,4]],0,0,R).
R = 1.
?- mat_get([[1,2,5],[3,4,7]],1,2,R).
R = 7.
?- mat_get([[1,2,5],[3,4,7]],1,2,7).
true.
?- mat_get([[1,2,5],[3,4,7]],1,2,6).
false.

2. Mean

Define a relation `mat_mean(M, R)` in which R is the mean value of a matrix M.

Examples:

?- mat_mean([[1,2],[3,4]],R).
R = 2.5.
?- mat_mean([[0,0,1],[0,0,2]],R).
R = 0.5.
?- mat_mean([[0,0,1],[0,0,2]],0.5).
true.
?- mat_mean([[0,0,1],[0,0,2]],0.6).
false.

3. Transpose

Define a relation `mat_trans(M,R) where` matrices M and R are the transpose of each other.

?- mat_trans([[1,2]],R).
R = [[1], [2]].
?- mat_trans(R, [[1],[2]]).
R = [[1, 2]].
?- mat_trans([[1,2,3]],[[1],[2],[3]]).
true.
?- mat_trans([[1,2,3]],[[3],[2],[1]]).
false.

4. Blend

Define a relation `mat_blend(A, B, W, C)` in which C is the blending result of two given matrices A and B with the weight W, i.e., C = W*A + (1-W)*B where each element C(i,j) is the sum of W*A(i,j) and (1-W)*B(i,j).

Examples:

?- mat_blend([[1.0,2.0],[3.0,4.0]],[[2.0,3.0],[1.0,2.0]],0.5,C).
C = [[1.5, 2.5], [2.0, 3.0]].
?- mat_blend([[1,2],[3.0,4.0]],[[2.0,3.0],[1.0,2.0]],0.4,C).
C = [[1.6, 2.5999999999999996], [1.8000000000000003, 2.8]].
?- mat_blend([[1.0,2.0],[3.0,4.0]],[[2.0,3.0],[1.0,2.0]],0.5,[[1.5, 2.5], [2.0, 3.0]]).
true.
?- mat_blend([[1.0,2.0],[3.0,4.0]],[[2.0,3.0],[1.0,2.0]],0.5,[[10.0, 2.5], [2.0, 3.0]]).
false.

5. Dot product

Define a relation `mat_dot(A, B, C)` where matrix C is the dot product of two given matrices A and B. Specifically, in the output matrix C = A x B, each element C(i,j) is the dot product of the i-th row of A and the j-th column of B.

Examples:

?- mat_dot([[1.0,2.0,3.0],[4.0,5.0,6.0],[7.0,8.0,9.0]],[[1.0,2.0,3.0],[4.0,5.0,6.0],[7.0,8.0,9.0]], R).

R = [[30.0, 36.0, 42.0], [66.0, 81.0, 96.0], [102.0, 126.0, 150.0]].

?- mat_dot([[0.0,1.0,1.0],[2.0, 2.0, 2.0]], [[1.0,3.0,1.0],[2.0, 2.0, 0.0], [3.0, 1.0, 1.0]], R).

R = [[5.0, 3.0, 1.0], [12.0, 12.0, 4.0]].

?- mat_dot([[0.0,1.0,1.0],[2.0, 2.0, 2.0]], [[1.0,3.0,1.0],[2.0, 2.0, 0.0], [3.0, 1.0, 1.0]], [[5.0, 3.0, 1.0], [12.0, 12.0, 4.0]]).

true.

?- mat_dot([[0.0,1.0,1.0],[2.0, 2.0, 2.0]], [[1.0,3.0,1.0],[2.0, 2.0, 0.0], [3.0, 1.0, 1.0]], [[5.0, 3.0, 1.0], [12.0, 12.0, 3.0]]).

false.