CMPE 260

PRINCIPLES OF PROGRAMMING LANGUAGES

BATURALP YÖRÜK

2015400036

PROLOG PROJECT

22/04/2018

INTRODUCTION

The predicates that will be used are as follows:

team(teamName, hometown).

match(week, homeTeam, homeTeamScore, awayTeam, awayTeamScore).

The following is a portion of a sample database illustrating the two relations that are defined:

team(realmadrid, madrid).

team(juventus, torinp).

team(galatasaray, istanbul).

team(kobenhavn, kopenag).

match(1, galatasaray, 1, realmadrid, 6).

match(1, kobenhavn, 1, juventus, 1).

match(2, juventus, 2, galatasaray, 2).

match(2, realmadrid, 4, kobenhavn, 0).

The first predicate shows the teams currently existing in the database. For example

team (galatasaray,istanbul).

means that galatasaray is a team that plays home matches in istanbul.

The second predicate implies that there has been a match between the two given teams in the given week. Score is also given in the predicate. To illustrate, *match(2, realmadrid, 4, kobenhavn, 0).* implies that “In the 2nd week, realmadrid defeated kobenhavn with the score of 4-0.”

PROGRAM INTERFACE

Copy the file cl_base.pl posted into your working directory. The file contains a database of facts about 4 teams and the played matches that you may use to test and debug your program. Also put the predicates.pl file to the same directory. Then run it with swipl.exe program by double clicking on the file. Then you can get the predicates results from that interface. (i.e Write “*order(L,6).*” and then press enter.) You will see the answer just under what you have written.

INPUT and OUTPUT

INPUT OF THE PROGRAM MUST BE IN THESE TWO FORMATS:

1) team(teamName, hometown).

2) match(week, homeTeam, homeTeamScore, awayTeam, awayTeamScore).

SAMPLE INPUT:

team(realmadrid, madrid).

team(juventus, torino).

team(galatasaray, istanbul).

team(kobenhavn, copenhagen).

team(manutd, manchester).

team(realsociedad, sansebastian).

team(shaktard, donetsk).

team(bleverkusen, leverkusen).

team(omarseille, marseille).

team(arsenal, london).

team(fcnapoli, napoli).

team(bdortmund, dortmund).

match(1, galatasaray, 1, realmadrid, 6).

match(1, kobenhavn, 1, juventus, 1).

match(1, manutd, 4, bleverkusen, 2).

match(1, realsociedad, 0, shaktard, 2).

match(1, omarseille, 1, arsenal, 2).

match(1, fcnapoli, 2, bdortmund, 1).

match(2, juventus, 2, galatasaray, 2).

match(2, realmadrid, 4, kobenhavn, 0).

match(2, shaktard, 2, manutd, 3).

match(2, bleverkusen, 1, realsociedad, 1).

match(2, bdortmund, 3, omarseille, 0).

match(2, arsenal, 2, fcnapoli, 0).

match(3, galatasaray, 3, kobenhavn, 1).

match(3, realmadrid, 2, juventus, 1).

match(3, manutd, 1, realsociedad, 0).

match(3, bleverkusen, 4, shaktard, 0).

match(3, omarseille, 1, fcnapoli, 2).

match(3, arsenal, 1, bdortmund, 2).

match(4, kobenhavn, 1, galatasaray, 0).

match(4, juventus, 2, realmadrid, 2).

match(4, bleverkusen, 0, manutd, 5).

match(4, shaktard, 4, realsociedad, 0).

match(4, fcnapoli, 4, omarseille, 2).

match(4, bdortmund, 0, arsenal, 1).

match(5, realmadrid, 4, galatasaray, 1).

match(5, juventus, 3, kobenhavn, 1).

match(5, realsociedad, 0, manutd, 0).

match(5, shaktard, 0, bleverkusen, 0).

match(5, bdortmund, 3, fcnapoli, 1).

match(5, arsenal, 2, omarseille, 0).

match(6, galatasaray, 1, juventus, 0).

match(6, kobenhavn, 0, realmadrid, 2).

match(6, manutd, 1, shaktard, 0).

match(6, realsociedad, 2, bleverkusen, 0).

match(6, omarseille, 1, bdortmund, 2).

match(6, fcnapoli, 2, arsenal, 0).

SAMPLE OUTPUT:

?- allTeams(L,N).

L = [galatasaray, realmadrid, juventus, kobenhavn, manutd, realsociedad, shaktard, bleverkusen,

omarseille, arsenal, fcnapoli, bdortmund]

N = 12;

L = [galatasaray, realmadrid, juventus, kobenhavn, manutd, realsociedad, shaktard, bleverkusen,

omarseille, arsenal, bdortmund, fcnapoli]

N = 12;

L = [galatasaray, realmadrid, juventus, kobenhavn, manutd, realsociedad, shaktard, bleverkusen,

omarseille, bdortmund, arsenal, fcnapoli]

N = 12

True

?- wins(galatasaray,4,L,N).

L = [kobenhavn]

N = 1 ;

False

?- losses(galatasaray,4,L,N).

L = [realmadrid, kobenhavn]

N = 2 ;

False

?- draws(galatasaray,4,L,N).

L = [juventus]

N = 1 ;

False

?- draws(galatasaray,4,[juventus],N).

N = 1 ;

False

?- draws(galatasaray,4,L,1).

L = [juventus] ;

False

?- draws(galatasaray,4,[juventus],1).

True

?- scored(juventus,5,S).

S = 9

?- conceded(juventus,5,C).

C = 8

?- average(kobenhavn, 3, A).

A = -6

?- average(kobenhavn, 6, A).

A = -9

?- order(L, 6).

L = [realmadrid, manutd, bdortmund, arsenal, fcnapoli, shaktard, juventus, bleverkusen, galatasaray, realsociedad, kobenhavn, omarseille]

?-topThree(L, 6).

L = [realmadrid, manutd, bdortmund]

PROGRAM STRUCTURE

1) allTeams(L,N)

L is the list containing all the teams in the database where N is the number of elements in the list.

2) wins(T,W,L,N)

This predicate implies that L involves the teams defeated by team T when we are in week W and N is the number of elements in L.

3) losses(T,W,L,N)

This predicate implies that L involves the teams that defeated team T when we are in week W and N is the number of elements in L.

4) draws(T,W,L,N)

This predicate is very similar but now L involves the teams that team T could not defeat also did not lose to.

5) scored(T,W,S)

This predicate implies that S is the total number of goals scored by team T up to (and including) week W.

6) conceded(T,W,C)

This predicate implies that C is the total number of goals conceded by team T up to (and including) week W.

7) average(T,W,A)

This predicate implies that A is the average (goals scored - goals conceded) of a team T gathered up to (and including) week W.

8) order(L,W)

This predicate implies that W (week) is given as constant and league order in that week will be retrieved in L. Assuming that the order is decided according to average i.e the one with the highest average will be at the top. If the two teams have the same average then the order can be in any order.

9) topThree([T1,T2,T3],W)

This predicate implies that T1, T2 and T3 are the top teams when we are in the given week W.

EXAMPLES

```
?- allTeams(L,N).
L = [realmadrid, juventus, galatasaray, kobenhavn, manutd, realsociedad, shaktard, bleverkusen, omarseille|...],
N = 12 ;
L = [realmadrid, juventus, galatasaray, kobenhavn, manutd, realsociedad, shaktard, bleverkusen, omarseille|...],
N = 12 ;
L = [realmadrid, juventus, galatasaray, kobenhavn, manutd, realsociedad, shaktard, bleverkusen, omarseille|...],
N = 12 ;
L = [realmadrid, juventus, galatasaray, kobenhavn, manutd, realsociedad, shaktard, bleverkusen, omarseille|...],
N = 12 ;
L = [realmadrid, juventus, galatasaray, kobenhavn, manutd, realsociedad, shaktard, bleverkusen, omarseille|...],
N = 12 ;
L = [realmadrid, juventus, galatasaray, kobenhavn, manutd, realsociedad, shaktard, bleverkusen, omarseille|...],
N = 12 ;
L = [realmadrid, juventus, galatasaray, kobenhavn, manutd, realsociedad, shaktard, bleverkusen, arsenal|...],
N = 12 .
```

```
?- wins(galatasaray,4,L,N).
L = [kobenhavn],
N = 1.
```

```
?- draws(galatasaray,4,L,N).
L = [juventus],
N = 1.
```

```
?- losses(galatasaray,4,L,N).
L = [realmadrid, kobenhavn],
N = 2.
```

```
?- conceded(juventus,5,C).
C = 8.
```

```
?- scored(juventus,5,S).
S = 9.
```

```
?- average(kobenhavn,3,A).
A = -6.
```

```
?- order(L,6).
L = [realmadrid, manutd, bdortmund, arsenal, fcnapoli, shaktard, juventus, realsociedad, bleverkusen|...].

?- topThree(L,6).
L = [realmadrid, manutd, bdortmund].
```

IMPROVEMENTS and EXTENSIONS

This program could be shorter and it could be more naively implemented.

DIFFICULTIES ENCOUNTERED

I encountered some difficulties while implementing this project because it was my first time with Prolog thus I was not into it. Those difficulties was mostly about Prolog's built-in functions and recursive way of the predicates.

CONCLUSION

As a conclusion, I implemented my first Prolog project and it worked succesfully with all the testcases with cl_base input.

APPENDICES

```
%% Baturalp Yoruk
%% 2015400036
%% PROLOG PROJECT


%_____________________________________________________________________________
_____ALLTEAMS PREDICATE
_____________________________________________________


%% Finds the teams' names from team(_,_) predicates. Also finds number of
teams.
%% Also prints different permutations of the list.
allTeams(L,N):- findall(X,team(X,_),Z), permutation(Z, L), length(L,N).


%% This is for some predicates which don't need different permutations of
the teams list.
stableAllTeams(L,N):- findall(X,team(X,_),L), length(L,N).


%_____________________________________________________________________WINS
PREDICATE AND IT'S HELPER PREDICATES
_____________________________________________


%% This keeps a number between 1 and W (both inclusive).
smaller(A,W):- between(1,W,A).


%% Finds the team T's wins at home.
helper_wins(T,W,L,N):- smaller(A,W), match(A,T,S1,_,S2), S1>S2,
findall(X,match(A,T,S1,X,S2),L), length(L,N), W is W.


%% Finds the team T's wins at away.
helper_wins2(T,W,L,N):- smaller(A,W), match(A,_,S1,T,S2), S2>S1,
findall(X,match(A,X,S1,T,S2),L), length(L,N), W is W.


%% Puts all wins at home to a list L.
helper_wins3(T,W,L,N):- findall(X, helper_wins(T,W,X,N), L).


%% Puts all wins at away to a list L.
helper_wins4(T,W,L,N):- findall(X, helper_wins2(T,W,X,N), L).
```

```
%% Puts all wins to a list and finds its numbers.

helper_wins5(T,W,C,N):- helper_wins3(T,W,L1,N), helper_wins4(T,W,L2,N),
append(L1,L2,Z), append(Z,C), length(C,N).


%% Gives you the list of wins for team T and wins number.

wins(T,W,L,N):- helper_wins5(T,W,L,N).


%_____________________________________________________________________LOSSES
PREDICATE AND IT'S HELPER
PREDICATES_____________________________________________


%% Finds the team T's losses at home.

helper_losses(T,W,L,N):- smaller(A,W), match(A,T,S1,_,S2), S1<S2,
findall(X,match(A,T,S1,X,S2),L), length(L,N), W is W.


%% Finds the team T's losses at away.

helper_losses2(T,W,L,N):- smaller(A,W), match(A,_,S1,T,S2), S2<S1,
findall(X,match(A,X,S1,T,S2),L), length(L,N), W is W.


%% Puts all losses at home to a list L.

helper_losses3(T,W,L,N):- findall(X, helper_losses(T,W,X,N), L).


%% Puts all losses at away to a list L.

helper_losses4(T,W,L,N):- findall(X, helper_losses2(T,W,X,N), L).


%% Puts all losses to a list and finds its numbers.

helper_losses5(T,W,A,N):- helper_losses3(T,W,L1,N),
helper_losses4(T,W,L2,N), append(L1,L2,Z), append(Z,A), length(A,N).


%% Gives you the list of losses for team T and losses number.

losses(T,W,L,N):- helper_losses5(T,W,L,N).


%_____________________________________________________________________DRAWS
PREDICATE AND IT'S HELPER
PREDICATES_____________________________________________


%% Finds the team T's draws at home.

helper_draws(T,W,L,N):- smaller(A,W), match(A,T,S1,_,S2), S1=:=S2,
findall(X,match(A,T,S1,X,S2),L), length(L,N), W is W.
```

```
%% Finds the team T's draws at away.

helper_draws2(T,W,L,N):- smaller(A,W), match(A,_,S1,T,S2), S2=:=S1,
findall(X,match(A,X,S1,T,S2),L), length(L,N), W is W.


%% Puts all losses at home to a list L.

helper_draws3(T,W,L,N):- findall(X, helper_draws(T,W,X,N), L).


%% Puts all losses at away to a list L.

helper_draws4(T,W,L,N):- findall(X, helper_draws2(T,W,X,N), L).


%% Puts all draws to a list and finds its numbers.

helper_draws5(T,W,A,N):- helper_draws3(T,W,L1,N), helper_draws4(T,W,L2,N),
append(L1,L2,Z), append(Z,A), length(A,N).


%% Gives you the list of draws for team T and losses number.

draws(T,W,L,N):- helper_draws5(T,W,L,N).


%_____________________________________________________________________SCORED
PREDICATE AND IT'S HELPER
PREDICATES_____________________________________________


%% Finds the goals that team T scored at home.

helper_scored(T,W,L,N,S):- smaller(A,W), match(A,T,S1,_,S2),
findall(X,match(A,T,S1,X,S2),L), length(L,N), W is W, S is S1.


%% Finds the goals that team T scored at away.

helper_scored2(T,W,L,N,S):- smaller(A,W), match(A,_,S1,T,S2),
findall(X,match(A,X,S1,T,S2),L), length(L,N), W is W, S is S2.


%% Gives the list L (all goals list) that team T scored at home.

helper_scored3(T,W,L,N,S):- findall(S, helper_scored(T,W,_,N,S), L).


%% Gives the list L (all goals list) that team T scored at away.

helper_scored4(T,W,L,N,S):- findall(S, helper_scored2(T,W,_,N,S), L).


%% Sums the goals that team T scored until week W (inclusive) and gives
that number S.

scored(T,W,S):- helper_scored3(T,W,L1,N,B), helper_scored4(T,W,L2,N,B),
append(L1,L2,Z), sum_list(Z,S).
```

```
%_____CONCEDED PREDICATE AND IT'S HELPER PREDICATES_____

%% Finds the goals that team T conceded at home.
helper_condeded(T,W,L,N,S):- smaller(A,W), match(A,T,S1,_,S2),
findall(X,match(A,T,S1,X,S2),L), length(L,N), W is W, S is S2.

%% Finds the goals that team T conceded at away.
helper_condeded2(T,W,L,N,S):- smaller(A,W), match(A,_,S1,T,S2),
findall(X,match(A,X,S1,T,S2),L), length(L,N), W is W, S is S1.

%% Gives the list L (all goals list) that team T conceded at home.
helper_condeded3(T,W,L,N,S):- findall(S, helper_condeded(T,W,_,N,S), L).

%% Gives the list L (all goals list) that team T conceded at away.
helper_condeded4(T,W,L,N,S):- findall(S, helper_condeded2(T,W,_,N,S), L).

%% Sums the goals that team T conceded until week W (inclusive) and gives
that number C.
conceded(T,W,C):- helper_condeded3(T,W,L1,N,B),
helper_condeded4(T,W,L2,N,B), append(L1,L2,Z), sum_list(Z,C).

%_____AVERAGE PREDICATE_____

%% Substract the conceded goals from scored goals of the given team T
(finds that teams average) at week W(inclusive).
average(T,W,A):- scored(T,W,S), conceded(T,W,C), A is S-C.

%_____ORDER PREDICATE AND IT'S HELPER PREDICATES_____

%% Pairs the teams with their own average.
helper_order(W,A,E) :- stableAllTeams(L,N), Z is N-1, between(0,Z,M),
nth0(M,L,X), average(X, W, A), append([A],[X],E).

%% Gives the paired list (Z) of all teams.
helper_order2(W,Z) :- findall(X, helper_order(W,_,X),L), sort(L,P),
reverse(P,Z).
```

```
%% Appends the list above.
helper_order3(W,L) :- helper_order2(W,Z), append(Z,L).


%% Gives the Xth element of that list.
helper_order4(W,Q,X) :- helper_order3(W,L), nth0(X,L,Q).


%% Increment the given number by two. This method stands for taking the
listed teams names without their averages.
incr(X,X1) :- X1 is X+2.


%% Gives the Xth and jumps one element after it and goes like that until
the end of the list.
helper_order5(W,Q,X) :- helper_order4(W,Z,X), incr(X,X1),
stableAllTeams(_,N), X1<(N*2)+2, findall(C, helper_order5(W, C, X1), Z1),
append([Z],Z1 , Q).


%% Flattens the above list.
helper_order6(W,Q,X) :- helper_order5(W,L,X), flatten(L,Q).


%% Gives the list L which contains the order of the teams at the end of
that week(W).
order(L,W) :- helper_order6(W,L,1).


%_____________________________________________________________________TOP
THREE
PREDICATE_____________________________________________________________


%% Gives the first three teams after the given week(W).
topThree([T1,T2,T3],W) :- order(L1,W), nth0(0,L1,T1), nth0(1,L1,T2),
nth0(2,L1,T3).
```