Abdurrahim ESKİN/2016400387

CMPE 260 HW1 REPORT

In the question 0.1, I implemented the predicate allteams(L,N). L is the list containing all the teams in the database where N is the number of elements, I use findall function to find all teams each other after that I send all teams to TE array to keep the length of that array ı used length function and I used Permutation function

allteams(L,N):-

findall(A,team(A,\_),TE),

length(TE,N),

permutation(TE,L).

In the question 0.2 I implemented the following predicates:

wins(T,W,L,N) 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.

losses(T,W,L,N) 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.

draws(T,W,L,N) is very similar but now L involves the teams that team T could not defeat also

did not lose to.

I found two different situations; if a team scored more than got scored home and away matches and the team will be winner. I kept two different information in different arrays and append them each other, in other cases ı followed that ways and looked the situaiton that if a team got scored more than scored it is a looser team and in the last case if the scored and being got scored situations are same these teams are in draw situation.

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

findall(LT,(match(X,T,SC,LT,GS),X=<W,SC>GS),L1),

findall(LT,(match(X,LT,GS,T,SC),X=<W,SC>GS),L2),

append(L1,L2,L),

length(L,N).

loses(T,W,L,N):-

findall(LT,(match(X,T,SC,LT,GS),X=<W,SC<GS),L1),

findall(LT,(match(X,LT,GS,T,SC),X=<W,SC<GS),L2),

append(L1,L2,L),

length(L,N).

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

findall(LT,(match(X,T,SC,LT,GS),X=<W,SC=GS),L1),

findall(LT,(match(X,LT,GS,T,SC),X=<W,SC=GS),L2),

append(L1,L2,L),

length(L,N).

In question 0.3, I implemented:

scored(T,W,S) where S is the total number of goals scored by team T up to (and including)

week W.

conceded(T,W,C) where C is the total number of goals conceded by team T up to week W.

\*\*I assumed T and W are given as constants.

average(T,W,A) where A is the average (goals scored { goals conceded) of a team T gathered

up to (and including) week W.

\*\*I assumed T and W are given as constants.

In the first part I found the team's scores until selected week these scored contains, the all goals which have been scored in home and away matches to keep that information I looked

that in two different arrays and append that information to single array and used sum list method to sum all numbers of that array and assigned this summation to the S value

For other cases i used that way and evaluate conceded situation, in average part I calculated the GS(getting scored) and SC(scored) numbers for a team and subtracted these values to find that team's average in S value.

scored(T,W,S):-

findall(SC,(match(X,T,SC,\_,\_),X=<W),AV1),

findall(SC,(match(X,\_,\_,T,SC),X=<W),AV2),

append(AV1,AV2,L),

sumlist(L,S).

conceded(T,W,S):-

findall(GS,(match(X,T,\_,\_,GS),X=<W),AV1),

findall(GS,(match(X,\_,\_,GS,T),X=<W),AV2),

append(AV1,AV2,L),

sumlist(L,S).

average(T,W,S):-

findall(SC,(match(X,T,SC,\_,\_),X=<W),AV1),

findall(SC,(match(X,\_,\_,T,SC),X=<W),AV2),

append(AV1,AV2,L1),

sumlist(L1,S1),

findall(GS,(match(X,T,\_,\_,GS),X=<W),AV3),

findall(GS,(match(X,\_,\_,GS,T),X=<W),AV4),

append(AV3,AV4,L2),

sumlist(L2,S2),

S is S1-S2.

In question 0.4 I implement:

order(L,W) where W (week) is given as constant and league order in that week will be

retrieved in L.

Additionally, I implemented topThree([T1,T2, T3],W) where T1 T2 and T3 are the top

teams when we are in the given week W.

\*\*I assumed 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.

I found the names off all teams and also found averages of all names

%After that I sorted the teams according to their averages

order(L,W):-

findall(X,team(X,\_),LI),

teamavg(LI,W,L1),

sort(2, @>=, L1, L2),

tnames(L2,L3),

reverse(L3,L).

teamavg([],W,L1).

teamavg([X|T],W,L1):-

average(X,W,S),

teamavg(T,W,L2),

append(L2,[[X,S]],L1).

tnames([H|T],L3):-

select(H,H1),

tnames(T,L4),

append(L4,[H1],L3).

tnames([],L3).

topThree([X,Y,Z],W):-

order(L,W),

[X,Y,Z|\_]=L.

select([X|\_],X).