

E06 Queries on KB

17341015 Hongzheng Chen

October 12, 2019

Contents

1	Problem Description	2
2	Codes and Results	2

1 Problem Description

Given a KB `Restaurants.pl`, which describes the distribution of branches of 10 well-known restaurants in Guangzhou.

For example, `restaurant(ajukejiacai,2007,yuecai)` means that `ajukejiacai` was founded in 2007 and is a restaurant of `yuecai`. `branch(ajukejiacai,xintiandi)` means that `ajukejiacai` has a branch in `xintiandi`. `district(xintiandi,panyu)` means that `xintiandi` is an area of `panyu` district.

Please formulate each of the following questions as a query using Prolog's notation, pose it to Prolog, and obtain Prolog's answer:

1. What restaurants have branches in beigang?
2. What districts have restaurants of yuecai and xiangcai?
3. What restaurants have the least number of branches?
4. What areas have two or more restaurants?
5. Which restaurant has the longest history?
6. What restaurants have at least 10 branches?

Please define the new relation below using Prolog and test it.

- `sameDistrict(Restaurant1, Restaurant2)`: Restaurant1 and Restaurant2 have one or more branches in the same district.

You should write down a listing that shows the queries you submitted to Prolog, and the answer returned. Hand in a file named `E06_YourNumber.pdf`, and send it to `ai_201901@foxmail.com`

2 Codes and Results

The below listing shows `rules.pl` file.

- `numBranches/2`: Calculate the number of branches of a specific restaurant.
- `sameDistrict/2`: Check if two restaurants are in the same district.

```
numBranches(X,L) :- setof(Branch,Year^Type^(restaurant(X,Year,Type),branch(X,Branch)),Z),
    ↪ length(Z,L).

sameDistrict(X,Y) :- branch(X,Area1),branch(Y,Area2),district(Area1,Dist),district(Area2,
    ↪ Dist).
```

The below listing shows `sol.pl` program.

- Q1: Directly use `branch(X,beigang)`.
- Q2: Similar to `natural join` in MySQL, but remember to use `^` to eliminate verbose output.
- Q3: Use the predefined `numBranches/2` in `rules.pl`. If no restaurant has less branches than restaurant A, then restaurant A has the least number of branches. Here we use `\+setof` to test if the return is an empty set.

- Q4: Use `length` to get number of restaurants in one area, and make the length is bigger or equal to 2.
- Q5: Similar to Q3 but change number of branches to founded year.
- Q6: Reuse the `numBranches/2` function.

```
['Restaurants.pl', 'rules.pl'].

%% Q1
findall(Rest, branch(Rest, beigang), Z).

%% Q2
setof(Dist, Loc^Rest1^Rest2^Year1^Year2^Loc1^Loc2^(restaurant(Rest1, Year1, yuecai),
    ↪ restaurant(Rest2, Year2, xiangcai), branch(Rest1, Loc1), branch(Rest2, Loc2), district(
    ↪ Loc1, Dist), district(Loc2, Dist)), Res).

%% Q3
findall(MinRest, (numBranches(MinRest, MinNum), \+setof(SmallerRest, SmallerNum^(numBranches(
    ↪ SmallerRest, SmallerNum), MinNum > SmallerNum), Lst)), Z).

%% Q4
setof(Area, Lst^Len^(setof(Rest, branch(Rest, Area), Lst), length(Lst, Len), Len >= 2), Z).

%% Q5
findall(MinYearRest, (restaurant(MinYearRest, MinYear, T2), \+setof(ShorterRest, T1^Year^(
    ↪ restaurant(ShorterRest, Year, T1), Year < MinYear), Lst)), Z).

%% Q6
setof(Rest, Num^(numBranches(Rest, Num), Num >= 10), Z).
```

```

chhzh123@DESKTOP-PV2UBJL: /mnt/d/Assignments/ArtificialIntelligence/E06_Queries
chhzh123@DESKTOP-PV2UBJL:/mnt/d/Assignments/ArtificialIntelligence/E06_Queries$ prolog
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- ['Restaurants.pl', 'rules.pl'].
true.

?-
|   %% Q1
|   findall(Rest, branch(Rest, beigang), Z).
Z = [mixuebingcheng, huangmenjimifan, shaxianxiaochi].

?-
|   %% Q2
|   setof(Dist, Loc^Rest1^Rest2^Year1^Year2^Loc1^Loc2^(restaurant(Rest1, Year1, yuecai), restaurant(Rest2, Year2, xiangcai), branch(Rest1, Loc1), branch(Rest2, Loc2), district(Loc1, Dist), district(Loc2, Dist)), Res).
Res = [haizhu, liwan, panyu, tianhe, yuexiu].

?-
|   %% Q3
|   findall(MinRest, (numBranches(MinRest, MinNum), \+setof(SmallerRest, SmallerNum^(numBranches(SmallerRest, SmallerNum), MinNum > SmallerNum), Lst)), Z).
Z = [hongmenyan].

?-
|   %% Q4
|   setof(Area, Lst^Len^(setof(Rest, branch(Rest, Area), Lst), length(Lst, Len), Len >= 2), Z).
Z = [bainahui, beigang, dongpu, shiqiao, tianhebei, xintiandi, yongfu, yuancun].

?-
|   %% Q5
|   findall(MinYearRest, (restaurant(MinYearRest, MinYear, T2), \+setof(ShorterRest, T1^Year^(restaurant(ShorterRest, Year, T1), Year < MinYear), Lst)), Z).
Z = [huangmenjimifan].

?-
|   %% Q6
|   setof(Rest, Num^(numBranches(Rest, Num), Num >= 10), Z).
Z = [dagangxianmiaoshaoji, diandude, mixuebingcheng, muwushaokao, tongxianghui].

```

I test the sameDistrict/2 rule for several cases shown below.

```

chhzh123@DESKTOP-PV2UBJL: /mnt/d/Assignments/ArtificialIntelligence/E06_Queries
?- sameDistrict(diandude, mixuebingcheng).
true.

?- setof(X, sameDistrict(diandude, X), Z).
Z = [ajukejiacai, dagangxianmiaoshaoji, diandude, hongmenyan, huangmenjimifan, mixuebingcheng, muwushaokao, shaxianxiaochi, tongxianghui].

?- setof(X, sameDistrict(hongmenyan, X), Z).
Z = [ajukejiacai, dagangxianmiaoshaoji, diandude, hongmenyan, huangmenjimifan, mixuebingcheng, shaxianxiaochi, tongxianghui, yangguofu].

?- setof(pair(X, Y), (sameDistrict(X, Y), X \= Y), Z).
Z = [pair(ajukejiacai, dagangxianmiaoshaoji), pair(ajukejiacai, diandude), pair(ajukejiacai, hongmenyan), pair(ajukejiacai, huangmenjimifan), pair(ajukejiacai, mixuebingcheng), pair(ajukejiacai, muwushaokao), pair(ajukejiacai, shaxianxiaochi), pair(ajukejiacai, tongxianghui), pair(..., ...)].

?-

```