CSE 341
HOMEWORK 4
M. YUSUF KOÇER
141004012

PART1

I implemented "Route():-" function with helper functions. There are two helper function to find path if exist. I sent cities and empty path list "P" to helper functions. First helper function checks if there is directed connection between cities. It just uses flight facts to confirm. If there is direct path between cities, it returns true and add travelled cities to path list. If not, second helper function is called. It checks there is an indirected path to reach the final city.It s called recursively until reach the goal. Then it recursively returns path between cities A and B.

The working principle:

```
helper(A,B)
    if ( flight(A,B) == true )
        return True
    else
        if ( flight(A,C) and helper(C,B) )
            return True
```

PART2

There is the "SRoute():-" function to find shortest route between cities A and B. It s implemented same with "Route" function referenced by part1. I just added cost values to calculate total cost of routes. When it reaches to finish, it returns cost values recursively to the top of function. Then I print cost values with paths on the screen.

The working principle:

```
helper(A,B,CostTop)
     if ( flight(A,B) == true )
        return True and CostTop
    else
        if ( flight(A,C) and helper(C,B) )
            return True and ( CostTop+Cost(C,B) )
```

PART3

I implemented facts with given tables in pdf. Then there are some functions are wanted from us.

schedule(A,P,T): to calculate schedule of a student. We need to find which classes students have. So I used to “enrolment” facts to list students classes. Then also need to find class times. There is a “when” facts to find which class is when.

Usage(R,T): to calculate which room has a class in which time. I used to “where” facts to find the classrooms have this class. And “when” facts to find when is it starts.

Conflict(X,Y): to find conflicts if X and Y conflicts due to classroom or time. I need to check times and classrooms if classes have same. I found times and rooms with “when” and “where” facts for X and Y. If their time or room values conflict. İt returns true.

Meet(X,Y): to find meetings if student X and student Y are present in the same classroom at the same time. I checked if X and Y enrolled the same classes with “enrolment” facts and if classes have same room or time with “when” and “where” facts. If they are confirmed true, there is a meeting between X and Y.

PART4

intersect():
It takes two set to compare if there are intersctions between elements. I checked the first element of set1 is also element of set2 with member function defined by prolog. If it s element of set2 too, I added it to empty set to return. Then I send the remaining elemtents of set1 and all of set2 to intersection function again. It Works recursively until there is no element in set1.
Union():
It takes two set to create a new set3 from union elements of set1 and set2. I checked the empty set3 in all steps and add element if set3 doesnt include it, it s added into the set3. I used member function to check elements in sets. I send remaining elements to travel recursively all lists.

Element():
I checked the first element of given list. If it s equal to given element, it returns true directly. Else, I call the “element” function again with given element and list with remaining elements without first element. It Works recursively until there is no element in list or it find the searching element. If it finds, it return true,if not it return false.

Equivalent():
I checked if set1 is a subset of set2 and also if set2 is a subset of set1. isSubset Works as if the first element of set1 is also a member of set2, it removes first element from set1 and calls isSubset func with tail of set1 and set2 recursively until there is no element in set1. If there is no fail while looking member function with elements of set1 at the end of function, it returns true. I also checked reversed version of this, because if sets sizes are different, one of them could be anothers subset but another one is not subset of it. So function must be verify they are both subset of themselves.