GTU Department of Computer Engineering
CSE 341 – Fall 2022
Homework 4 Report

Emirkan Burak Yılmaz
1901042659

1. Expert System

First the knowledge base is created. Some of them facts are given below.

```
room(z23).
room(z11).
room(z06).
room(z10).

course(101).
course(102).
course(241).
course(222).
course(331).
course(341).
course(455).
course(452).

instructor(mehmet).
instructor(yakup).
instructor(erdogan).
instructor(yusuf).
instructor(alp).

student(student1).
student(student2).
student(student3).
student(student4).
student(student5).

% course time
when(101, 12).
when(102, 12).
when(241, 12).
when(222, 11).
when(331, 14).
when(341, 14).
when(455, 16).
when(452, 17).
```

```
% course room
where(101, z11).
where(102, z23).
where(241, z23).
where(222, z11).
where(331, z23).
where(341, z06).
where(455, z10).
where(452, z10).

% room provides the given special equipment
equipment(z23, projector).
equipment(z11, computer).
equipment(z06, projector).
equipment(z10, smart_board).
equipment(z10, projector).

% room has access for the handicapped students
access_for_handicapped(z10).
access_for_handicapped(z06).

% room capacity
capacity(z23, 120).
capacity(z11, 50).
capacity(z06, 150).
capacity(z10, 40).

% course capacity
capacity(101, 120).
capacity(102, 120).
capacity(241, 150).
capacity(222, 90).
capacity(331, 150).
capacity(341, 150).
capacity(455, 40).
capacity(452, 40).
```

Then the rules are defined as follows.

```
enroll(S, C) :-
    student(S),
    handicapped(S),
    course(C),
    where(C, R),
    access_for_handicapped(R).

enroll(S, C) :-
    student(S),
    \+ (handicapped(S)),
    course(C).

% checks if there is any scheduling conflict between two course
conflict(X, Y) :-
    course(X),
    course(Y),
    where(X, P),
    where(Y, P),
    when(X, T),
    when(Y, T),
    X \= Y.

% asssign room for the given course
assign(C, R) :-
    course(C),
    room(R),
    teaches(I, C),
    capacity(R, CR),
    capacity(C, CC),
    CC =< CR,
    forall(needs(C, E), equipment(R, E)),
    forall(prefers(I, E), equipment(R, E)).
```

Test Cases:

I. Check whether there is any scheduling conflict.
II. Check which room can be assigned to a given class.
III. Check which room can be assigned to which classes.
IV. Check whether a student can be enrolled to a given class.
V. Check which classes a student can be assigned.

Test ID	Result	Explanation	PASS/FAIL

I	`| ?- conflict(X, Y).`
	
	`X = 102`
	`Y = 241 ?`
	
	`yes`
		Both courses 102 and 241 are holding in room z23 at 12 am. So, there is conflict.	PASS

II	`| ?- assign(102, R). a`
	
	`R = z23 ?`
	
	`R = z06`
	
	`no`
		102 course requires a room with minimum capacity of 120. The capacity of z23 is 120 and it is 150 for z06. Since capacity of the z23 and z06 is enough for the 102, this course can be holding one of the two room.	PASS

III	`| ?- assign(C, z10). a`
	
	`C = 455 ?`
	
	`C = 452`
		Capacity of the z10 is 40. Both courses 455 and 452 requires a minimum capacity of 40. So those two courses can be hold at z10	PASS

IV	`| ?- enroll(student1, 102).`
	
	`true ?`
	
	`yes`
	`| ?- enroll(student2, 102).`
	
	`no`
	`| ?- enroll(student2, 341).`
	
	`true ?`
	
	`(16 ms) yes`
		Student1 can enroll the 102 course, however, student2 cannot enroll. Since student2 is handicapped and the course room z23 has no access for handicapped students. But student2 can enroll 341 course, since it's held at room z06 and this room has access for handicapped students.	PASS

V	`| ?- enroll(student2, C). a`
	
	`C = 341 ?`
	
	`C = 341`
	
	`C = 455`
	
	`C = 452`
	
	`no`
		Student2 is handicapped and can only enroll the courses that is hold in a class that has access for handicapped students. Those rooms are z10 and z06. The courses 341 is held at z10 and 455 and 452 are held at z06.	**PASS**

2. Possible Flights

First the knowledge base is created.

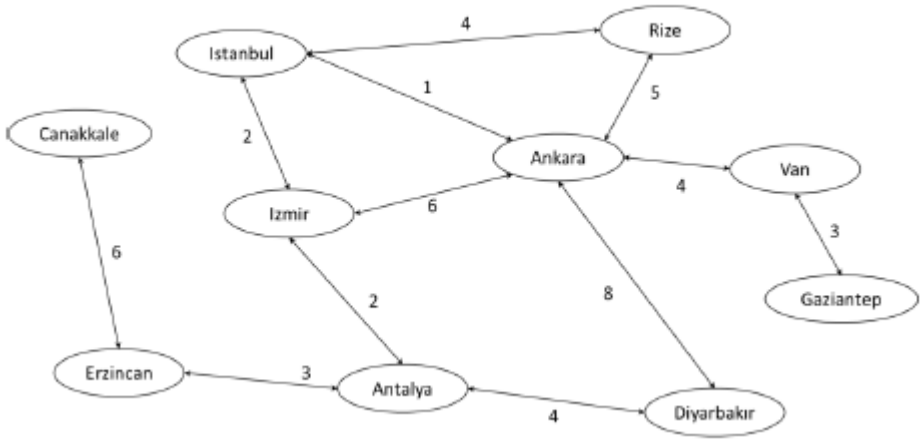
```
schedule(istanbul, ankara, 1).
schedule(istanbul, izmir, 2).
schedule(istanbul, rize, 4).
schedule(istanbul, mardin, 15).
schedule(ankara, rize, 5).
schedule(ankara, van, 4).
schedule(ankara, diyarbakir, 8).
schedule(ankara, izmir, 6).
schedule(izmir, antalya, 2).
schedule(izmir, manisa, 4).
schedule(van, gaziantep, 3).
schedule(antalya, erzincan, 3).
schedule(antalya, diyarbakir, 4).
schedule(erzincan, canakkale, 6).
```

Then the rules are defined as follows.

```
path(S, D, C) :-
    schedule(S, D, C).

path(S, D, C) :-
    schedule(D, S, C).

find_path(S, D, C, Visited) :-
    path(S, D, C).

find_path(S, D, C, Visited) :-
    path(S, B, C1),                    % there is a path between S and B with cost C1
    \+ (member(B, Visited)),           % B is not visited before
    find_path(B, D, C2, [B | Visited]),
    C is C1 + C2,
    S \= D.

% there exist a route between X and Y with cost C
connection(X, Y, C) :-
    find_path(X, Y, C, [X]).
```

Test Cases:

I. Different path costs between two cities.
II. All the different routes from a city to other cities with cost 8
III. All the different routes from a city to other cities



Test cases are tested on the above graph

Test ID	Result	PASS/FAIL

I	`| ?- connection(ankara, antalya, C). a`
	`C = 13 ?`
	`C = 12`
	`C = 8`
	`C = 5`
	`no`
		PASS

II		PASS

```
| ?- connection(S, D, 8). a

D = diyarbakir
S = ankara ?

D = ankara
S = diyarbakir

D = gaziantep
S = istanbul

D = ankara
S = istanbul

D = diyarbakir
S = istanbul

D = antalya
S = ankara

D = istanbul
S = ankara

D = erzincan
S = ankara

D = rize
S = izmir

D = antalya
S = rize

D = izmir
S = rize

D = ankara
S = antalya

D = rize
S = antalya

D = istanbul
S = gaziantep

D = ankara
S = erzincan

D = istanbul
S = diyarbakir

no
```

III		PASS

```
| ?- connection(istanbul, antalya, C). a

C = 13 ?

C = 9

C = 4

C = 20

C = 21

C = 17

no
```