GTU Department of Computer Engineering
CSE341 - Fall 2020
Homework 4(Prolog) Report

Akif KARTAL
171044098

1) Parts or properties you have implemented and you did not implemented.

I implemented part 1,2,3 and 4 because of time constraint.

I didn't implement part 5 and 6.

2) Assumptions about the parts you have implemented. Ambiguities and and how you handled them.

Part 1:

Pdf says your program should check if there is a direct route between two given cities but example gives any routes. Therefore, I wrote very simple query to handle this. Also I only have one direct rules since my program goes infinity loop I have define all routes.

Results;

```
?- route(edirne,X).
X = edremit ;
X = erzincan ;
false.

?- route(istanbul,X).
X = izmir ;
X = antalya ;
X = gaziantep ;
X = ankara ;
X = van ;
X = rize ;
X = isparta ;
X = burdur ;
X = antalya ;
X = konya ;
X = van ;
X = antalya ;
X = van ;
false.
```

Part2:

In this part to find a solution I used simple prolog list properties with my own knowledge base.

Results:

```
?- sroute(edremit,erzincan,X).
X = 1044.

?- sroute(istanbul,izmir,X).
X = 329.

?- sroute(rize,van,X).
X = 373.
```

Part3:

3.1;

In this part in pdf file given examples creates wrong result. Normally in schedule(S,P,T) query P denotes place but result gives class name instead of room name. This is the ambiguity of this part.

My results;

```
?- schedule(a,P,T).
P = z23,
T = 10 ;
P = z11,
T = 12.

?- schedule(b,P,T).
P = z23,
T = 10.

?- schedule(c,P,T).
P = z11,
T = 12.

?- schedule(d,P,T).
P = z06,
T = 14.
```

3.2:

In this part in pdf file given examples creates wrong result. Normally in usage(P,T) query T denotes Time but result gives class name instead of time . This is the ambiguity of this part.

My results;

```
?- usage(207,T).
T = 16 ;
T = 17.

?- usage(z23,T).
T = 10.

?- usage(z11,T).
T = 12.

?- usage(z06,T).
T = 14.
```

3.3;

No example but my results is;

```
?- conflict(455,452).
true .

?- conflict(455,341).
false.
```

3.4;

No example but my results is;

```
?- meet(a,b).
true .

?- meet(a,c).
true.

?- meet(a,e).
false.
```

Part4:

In this part, in order to make set operations I used Lists in prolog,

4.1

```
?- element(8,[5,6,7,8]).
true .

?- element(1,[2,3,54]).
false.
```

4.2

```
?- union([1,2,3,4],[2,3,5,7,8],[1,2,3,4,5,7,8]).
true .

?- union([1,2,3,4],[2,3,5,7,8],[1,2,3,4,5,7,8,9]).
false.
```

4.3

```
?- intersect([1,2,3,4],[2,3,5,7,8],[2,3]).
true .

?- intersect([1,2,3,4],[2,3,5,7,8],[2,3,5,7]).
false.
```

4.4

```
?- equivalent([1,2,3,4],[1,2,3,4]).
true .

?- equivalent([1,2,3,4],[1,2,3,4,5]).
false.
```