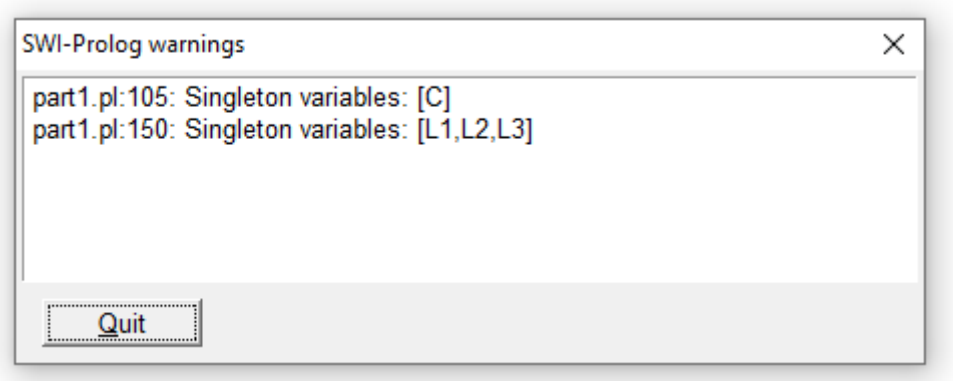
CSE 341 – PROGRAMMING LANGUAGES (FALL 2019)

HOMEWORK #4 – DOCUMENTATION

Gökhan HAS – 161044067

First, the program compile 2 warning occurs. The warningler does not prevent the program from running. This occurs because the defined variables are not used in some of them.



Şekil 1 Warnings

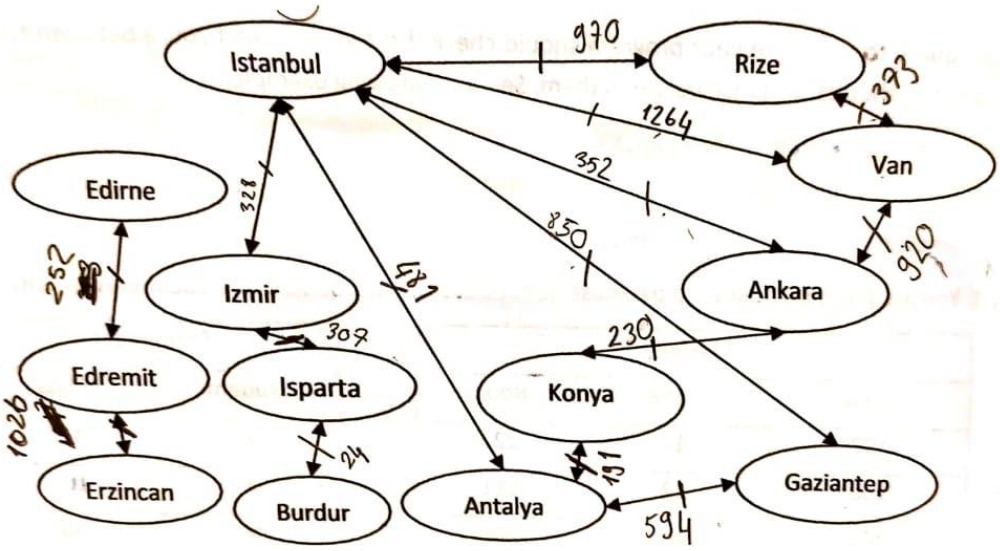
PART 1:

In this part, the facts were defined according to the figure. Then the route rule was written.

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

?- route(istanbuk,erzincan).
false.

?- route(istanbul,erzincan).
false.

?- route(rize,van).
true .

?- route(konya,X).
X = antalya .

?- route(konya,X).
X = antalya ;
X = ankara ;
X = izmir ;
X = gaziantep ;
X = istanbul ;
X = van.

?- route(gaziantep,edirne).
false.
```

PART 2:

In this part, first of all the distance facts were written. For this, the distances between cities were taken as follows:



The sroute rule statement, which gives the shortest flight between cities, is required. Then complete all the flights and distance are calculated.

As seen in the testbenches between Edremit Erzincan 1044 km, the shortest way between Istanbul and Konya 582 km from Ankara. The distance between Istanbul and Burdur is 659 km via Izmir and Isparta.

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

?- sroute(istanbul,burdur,X).
X = 659.

?- sroute(istanbul,konya,X).
X = 582.

?- sroute(istanbul,erzincan,X).
false.

?- sroute(istanbul,konya,582).
true.

?- sroute(istanbul,konya,600).
false.

?- sroute(ankara,van,X).
X = 920.
```

PART 3:

According to the statement given in the first part of this party, the facts were defined.

Classes		
Class	Time	Room
102	10	z23
108	12	z11
341	14	z06
455	16	207
452	17	207

Enrollment	
Student	Class
a	102
a	108
b	102
c	108
d	341
e	455

```
whenX(102,10).
whenX(108,12).
whenX(341,14).
whenX(455,16).
whenX(452,17).

where(102,z23).
where(108,z11).
where(341,z06).
where(455,207).
where(452,207).

enroll(a,102).
enroll(a,108).
enroll(b,102).
enroll(c,108).
enroll(d,341).
enroll(e,455).
```

IMPORTANT : WHY whenX?

Because the expression "when" gives the following error because it is a defined expression in the prolog. Therefore, it must be called whenX.

```
Singleton variables: [C]
ERROR: c:/users/gokha/desktop/prolog/part1.pl:119:
       No permission to redefine imported_procedure 'when:when/2'
ERROR: c:/users/gokha/desktop/prolog/part1.pl:120:
       No permission to redefine imported_procedure 'when:when/2'
ERROR: c:/users/gokha/desktop/prolog/part1.pl:121:
       No permission to redefine imported_procedure 'when:when/2'
ERROR: c:/users/gokha/desktop/prolog/part1.pl:122:
       No permission to redefine imported_procedure 'when:when/2'
ERROR: c:/users/gokha/desktop/prolog/part1.pl:123:
       No permission to redefine imported_procedure 'when:when/2'
```

3.1: Predicate “schedule(S,P,T)” that associates a student to a place and time of class.

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

```
?- schedule(a,z23,10).
true ;
```

```
?- scheule(b,P,T).
Correct to: "schedule(b,P,T)"?
Please answer 'y' or 'n'? yes
P = z23,
T = 10 .
```

3.2: Another predicate “usage(P,T)” that gives the usage times of a classroom. See the example query and its result.

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

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

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

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

?- usage(207,17).
true.
```

3.3: Another predicate “conflict(X,Y)” that gives true if X and Y conflicts due to classroom or time. And I added class 402 (conflict 341) time 14 for only used in this part(3.3).

```
?- conflict(102,108).
false.

?- conflict(341,108).
false.

?- conflict(455,452).
false.

?- conflict(402,341).
true.
```

3.4: Another predicate “meet(X,Y)” that gives true if student X and student Y are present in the same classroom at the same time.

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

?- meet(a,b).
true.

?- meet(a,d).
false.

?- meet(a,c).
true.

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

PART 4:

4.1: Define a Prolog predicate “element(E,S)” that returns true if E is in S.

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

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

4.2: Define a Prolog predicate “union(S1,S2,S3)” that returns true if S3 is the union of S1 and S2.

IMPORTANT : WHY unionx?

Because the expression "union" gives the following error because it is a defined expression in the prolog. Therefore, it must be called unionx.

```
ERROR: c:/users/gokha/desktop/prolog/part1.pl:152:
    No permission to redefine imported_procedure `lists:union/3'
ERROR: c:/users/gokha/desktop/prolog/part1.pl:153:
    No permission to redefine imported_procedure `lists:union/3'
ERROR: c:/users/gokha/desktop/prolog/part1.pl:154:
    No permission to redefine imported_procedure `lists:union/3'
```

```
?- unionx([3,5,2],[1,5],[3,2,1,5]).
true .

?- unionx([3,5,2],[1,5],U).
U = [3, 2, 1, 5] .

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

4.3: Define a Prolog predicate “intersect(S1,S2,S3)” that returns true if S3 is the intersection of of S1 and S2.

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

?- intersection([1,2,3,4],[2,3],I).
I = [2, 3].

?- intersection([1,2,3,4],[2,3],[7,5]).
false.

?- intersection([3,7,5,2],[2,5],[5,2]).
true.

?- intersection([3,7,5,2],[2,5],I).
I = [5, 2].

?- intersection([3,7,5,2],[2,5],[7,13]).
false.
```

4.4: Define a Prolog predicate “equivalent(S1,S2)” that returns true if S1 and S2 are equivalent sets.

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

?- equalivalent([1,2,3],[3,2,1]).
true.

?- equalivalent([1,2,3],[3,1,2]).
true.

?- equalivalent([1,2,3],[6,7,5]).
false.
```