4.4

(20 points) Use Prover9 to perform automatically the refutation. Submit a report with three parts:
 I. Assumptions and goal;
 II. The input and output of prover9 (The input of prover 9 should be in plain text)
 III. Conclusion

4.4.1 ASSUMPTIONS AND GOAL:

Following are the assumptions made

1. $S_1 : \neg Rides(x_1, x_2) \vee \neg Harley(x_2) \vee Rough(x_1)$

2. $S_{21} : \neg Biker(x_3) \vee Rides(x_3, f(x_3))$

3. $S_{22} : \neg Biker(x_3) \vee Harley(f(x_3)) \vee BMW(f(x_3))$

4. $S_3 : \neg Rides(x_5, x_6) \vee \neg BMW(x_6) \vee Yuppie(x_5)$

5. $S_4 : \neg Yuppie(x_7) \vee Lawyer(x_7)$

6. $S_5 : \neg Nice(x_8) \vee \neg Rough(x_9) \vee \neg Date(x_8, x_9)$

7. $S_{61} : Nice(Mary)$

8. $S_{62} : Biker(John)$

Following are the goals considered

1. $S_7 : Lawyer(John) \vee -Date(Mary, John)$

4.4.2 INPUT AND OUTPUT OF PROVER9

INPUT

```
1  formulas(assumptions).    % synonym for formulas(sos).
2    -Rides(x,y) | -Harley(y) | Rough(x).
3    -Biker(x) | Rides(x,f(x)).
4    -Biker(x) | Harley(f(x)) | BMW(f(x)).
5    -Rides(x,y) | -BMW(y) | Yuppie(x).
6    -Yuppie(x) | Lawyer(x).
7    -Nice(x) | -Rough(y) | -Date(x,y).
8    Nice(Mary).
9    Biker(John).
10 end_of_list.
11
12 formulas(goals).           % to be negated and placed in the sos list
13    Lawyer(John) | -Date(Mary, John).
14 end_of_list.
```

OUTPUT

```
1  ============================== prooftrans ==============================
2  Prover9 (32) version Dec-2007, Dec 2007.
3  Process 12132 was started by ROHITH PEDDI on DESKTOP-T2S851L,
4  Mon Nov  9 22:03:58 2020
5  The command was "/cygdrive/c/Program Files (x86)/Prover9-Mace4/bin-win32/prover9".
6  ============================== end of head ==============================
7
8  ============================== end of input ==============================
9
10 ============================== PROOF ==============================
11
12 % -------- Comments from original proof --------
13 % Proof 1 at 0.01 (+ 0.06) seconds.
14 % Length of proof is 23.
15 % Level of proof is 6.
16 % Maximum clause weight is 0.
17 % Given clauses 0.
18
19 1 Lawyer(John) | -Date(Mary,John) # label(non_clause) # label(goal).  [goal].
20 2 -Biker(x) | Rides(x,f(x)).  [assumption].
21 3 -Rides(x,y) | -Harley(y) | Rough(x).  [assumption].
```

```
22  4 -Rides(x,y) | -BMW(y) | Yuppie(x).  [assumption].
23  5 Biker(John).  [assumption].
24  6 -Biker(x) | Harley(f(x)) | BMW(f(x)).  [assumption].
25  7 -Biker(x) | -Harley(f(x)) | Rough(x).  [resolve(2,b,3,a)].
26  8 -BMW(f(x)) | Yuppie(x) | -Biker(x).  [resolve(4,a,2,b)].
27  9 -BMW(f(John)) | Yuppie(John).  [resolve(8,c,5,a)].
28  10 -Yuppie(x) | Lawyer(x).  [assumption].
29  11 Nice(Mary).  [assumption].
30  12 -Nice(x) | -Rough(y) | -Date(x,y).  [assumption].
31  13 -BMW(f(John)) | Lawyer(John).  [resolve(9,b,10,a)].
32  14 -Lawyer(John).  [deny(1)].
33  15 -Rough(x) | -Date(Mary,x).  [resolve(11,a,12,a)].
34  16 Date(Mary,John).  [deny(1)].
35  17 -Harley(f(John)) | Rough(John).  [resolve(7,a,5,a)].
36  18 Harley(f(John)) | BMW(f(John)).  [resolve(5,a,6,a)].
37  19 Rough(John) | BMW(f(John)).  [resolve(17,a,18,a)].
38  20 -BMW(f(John)).  [resolve(13,b,14,a)].
39  21 Rough(John).  [resolve(19,b,20,a)].
40  22 -Rough(John).  [resolve(15,b,16,a)].
41  23 $F.  [resolve(21,a,22,a)].
42
43  ============================= end of proof ==========================
```

4.4.3 CONCLUSION:

1. Prover9 proved the query using resolution technique

2. Knowledge base entails the given query

5 EXTRA CREDIT

(30 points) Use Prover9 to automatically perform the refutation of the following:

The Pigs and Balloons Puzzle

(1) All, who neither dance on tight ropes nor eat penny-buns, are old.

(2) Pigs, that are liable to giddiness, are treated with respect.

(3) A wise balloonist takes an umbrella with him.

(4) No one ought to lunch in public who looks ridiculous and eats penny-buns.

(5) Young creatures, who go up in balloons, are liable to giddiness.

(6) Fat creatures, who look ridiculous, may lunch in public, provided that they do not dance on tight ropes.

(7) No wise creatures dance on tight ropes, if liable to giddiness.

(8) A pig looks ridiculous, carrying an umbrella.

(9) All, who do not dance on tight ropes, and who are treated with respect are fat. Show that no wise young pigs go up in balloons.

-Lewis Carroll, Symbolic Logic,

Submit a report with three parts: I. Assumptions and goal; II. The input and output of prover9 (The input of prover 9 should be in plain text) III. Conclusion

5.0.1 ASSUMPTIONS AND GOAL

Following are the assumptions considered :

1. $Dances\_on\_tightropes(x) \lor Eats\_pennybuns(x) \lor Old(x)$

2. $\neg Pig(x) \lor \neg Liable\_to\_giddiness(x) \lor Treated\_with\_respect(x)$

3. $\neg Wise(x) \lor \neg Balloonist(x) \lor Has\_umbrella(x)$

4. $\neg Looks\_ridiculous(x) \lor \neg Eats\_pennybuns(x) \lor \neg Eats\_lunch\_in\_public(x)$

5. $\neg Balloonist(x) \lor \neg Young(x) \lor Liable\_to\_giddiness(x)$

6. $\neg Fat(x) \lor \neg Looks\_ridiculous(x) \lor Dances\_on\_tightropes(x) \lor Eats\_lunch\_in\_public(x)$

7. $\neg Liable\_to\_giddiness(x) \lor \neg Wise(x) \lor \neg Dances\_on\_tightropes(x)$

8. $\neg Pig(x) \lor \neg Has\_umbrella(x) \lor Looks\_ridiculous(x)$

9. $Dances\_on\_tightropes(x) \lor \neg Treated\_with\_respect(x) \lor Fat(x)$

10. $Young(x) \lor Old(x)$

11. $\neg Young(x) \lor \neg Old(x)$

Following are the goals considered

1. $\neg Wise(x) \lor \neg Young(x) \lor \neg Pig(x) \lor \neg Balloonist(x)$

5.0.2 INPUT AND OUTPUT

INPUT

```
formulas(assumptions).    % synonym for formulas(sos).
  Dances_on_tightropes(x) | Eats_pennybuns(x) | Old(x).
  -Pig(x)|-Liable_to_giddiness(x) | Treated_with_respect(x).
  -Wise(x) | -Balloonist(x) | Has_umbrella(x).
  -Looks_ridiculous(x) | -Eats_pennybuns(x) | -Eats_lunch_in_public(x).
  -Balloonist(x) | -Young(x) | Liable_to_giddiness(x).
  -Fat(x) | -Looks_ridiculous(x) | Dances_on_tightropes(x) | Eats_lunch_in_public(x).
  -Liable_to_giddiness(x) | -Wise(x) | -Dances_on_tightropes(x).
  -Pig(x) | -Has_umbrella(x) | Looks_ridiculous(x).
  Dances_on_tightropes(x) | -Treated_with_respect(x) | Fat(x).
  Young(x) | Old(x).
  -Young(x) | -Old(x).
end_of_list.

formulas(goals).          % to be negated and placed in the sos list
  -Wise(x) | -Young(x) | -Pig(x) | -Balloonist(x).
end_of_list.
```

OUTPUT

```
1  =============================== prooftrans ============================
2  Prover9 (32) version Dec-2007, Dec 2007.
3  Process 23656 was started by ROHITH PEDDI on DESKTOP-T2S851L,
4  Mon Nov  9 22:14:08 2020
5  The command was "/cygdrive/c/Program Files (x86)/Prover9-Mace4/bin-win32/prover9".
6  ============================= end of head ============================
7
8  ============================= end of input ===========================
9
10 ============================= PROOF ==================================
11
12 % -------- Comments from original proof --------
13 % Proof 1 at 0.00 (+ 0.06) seconds.
14 % Length of proof is 42.
15 % Level of proof is 8.
16 % Maximum clause weight is 2.
17 % Given clauses 0.
18
19 1 -Wise(x) | -Young(x) | -Pig(x) | -Balloonist(x) # label(non_clause) # label(goal).  [goal].
20 2 -Liable_to_giddiness(x) | -Wise(x) | -Dances_on_tightropes(x).  [assumption].
21 3 Dances_on_tightropes(x) | Eats_pennybuns(x) | Old(x).  [assumption].
22 4 -Fat(x) | -Looks_ridiculous(x) | Dances_on_tightropes(x) | Eats_lunch_in_public(x).  [
      assumption].
23 5 Dances_on_tightropes(x) | -Treated_with_respect(x) | Fat(x).  [assumption].
24 6 Pig(c1).  [deny(1)].
25 7 -Pig(x) | -Liable_to_giddiness(x) | Treated_with_respect(x).  [assumption].
26 8 -Pig(x) | -Has_umbrella(x) | Looks_ridiculous(x).  [assumption].
27 9 Wise(c1).  [deny(1)].
28 10 -Wise(x) | -Balloonist(x) | Has_umbrella(x).  [assumption].
29 11 -Liable_to_giddiness(x) | -Wise(x) | Eats_pennybuns(x) | Old(x).  [resolve(2,c,3,a)].
30 12 -Liable_to_giddiness(x) | -Wise(x) | -Fat(x) | -Looks_ridiculous(x) | Eats_lunch_in_public(
      x).  [resolve(2,c,4,c)].
31 13 -Treated_with_respect(x) | Fat(x) | -Liable_to_giddiness(x) | -Wise(x).  [resolve(5,a,2,c)
      ].
32 14 -Has_umbrella(c1) | Looks_ridiculous(c1).  [resolve(6,a,8,a)].
33 15 -Looks_ridiculous(x) | -Eats_pennybuns(x) | -Eats_lunch_in_public(x).  [assumption].
34 16 -Liable_to_giddiness(c1) | -Fat(c1) | -Looks_ridiculous(c1) | Eats_lunch_in_public(c1).  [
      resolve(12,b,9,a)].
35 17 Balloonist(c1).  [deny(1)].
36 18 -Balloonist(x) | -Young(x) | Liable_to_giddiness(x).  [assumption].
37 19 -Balloonist(c1) | Has_umbrella(c1).  [resolve(9,a,10,a)].
38 20 -Young(x) | -Old(x).  [assumption].
39 21 Young(x) | Old(x).  [assumption].
40 22 Young(c1).  [deny(1)].
41 23 -Young(c1) | Liable_to_giddiness(c1).  [resolve(17,a,18,a)].
42 24 Liable_to_giddiness(c1) | Old(c1).  [resolve(23,a,21,a)].
43 25 -Liable_to_giddiness(c1) | Treated_with_respect(c1).  [resolve(6,a,7,a)].
44 26 -Liable_to_giddiness(c1) | Eats_pennybuns(c1) | Old(c1).  [resolve(11,b,9,a)].
45 27 -Treated_with_respect(c1) | Fat(c1) | -Liable_to_giddiness(c1).  [resolve(13,d,9,a)].
46 28 -Liable_to_giddiness(c1) | -Fat(c1) | Eats_lunch_in_public(c1) | -Has_umbrella(c1).  [
      resolve(16,c,14,b)].
47 30 Has_umbrella(c1).  [resolve(19,a,17,a)].
48 31 -Has_umbrella(c1) | -Eats_pennybuns(c1) | -Eats_lunch_in_public(c1).  [resolve(14,b,15,a)].
49 32 Old(c1) | -Fat(c1) | Eats_lunch_in_public(c1) | -Has_umbrella(c1).  [resolve(24,a,28,a)].
50 34 Old(c1) | -Treated_with_respect(c1) | Fat(c1).  [resolve(24,a,27,c)].
51 35 Old(c1) | Treated_with_respect(c1).  [resolve(24,a,25,a)].
52 38 -Eats_pennybuns(c1) | -Eats_lunch_in_public(c1).  [resolve(30,a,31,a)].
53 39 Old(c1) | Eats_pennybuns(c1) | Old(c1).  [resolve(24,a,26,a)].
54 40 Old(c1) | Fat(c1) | Old(c1).  [resolve(34,b,35,b)].
55 41 Old(c1) | -Fat(c1) | Eats_lunch_in_public(c1).  [resolve(32,d,30,a)].
56 44 Old(c1) | Old(c1) | Old(c1) | Eats_lunch_in_public(c1).  [resolve(40,b,41,b)].
57 45 -Eats_lunch_in_public(c1) | Old(c1) | Old(c1).  [resolve(38,a,39,b)].
58 47 -Old(c1).  [resolve(22,a,20,a)].
59 48 Old(c1) | Old(c1) | Old(c1) | Old(c1) | Old(c1).  [resolve(44,d,45,a)].
60 49 $F.  [copy(48),merge(b),merge(c),merge(d),merge(e),unit_del(a,47)].
61
62 ============================= end of proof ===========================
```

5.0.3 CONCLUSION:

1. Prover9 proved the query using resolution technique

2. Knowledge base entails the given query