CS 6364: Artificial Intelligence

Homework 2: Logic Puzzles

Assumptions:

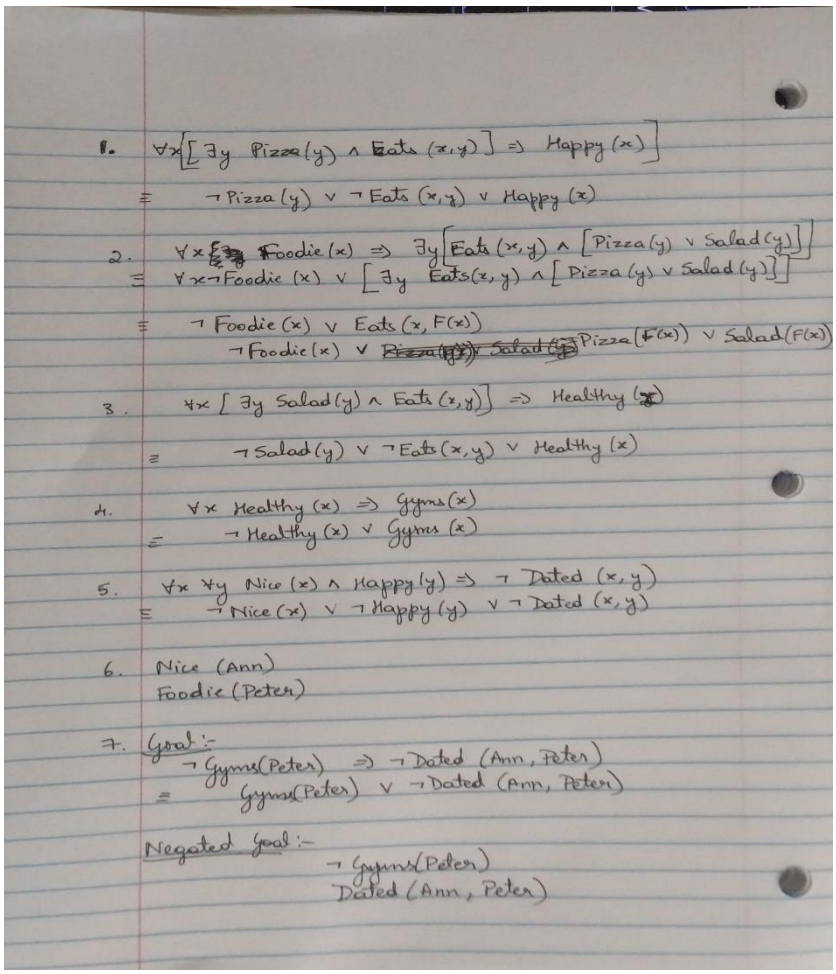
1. Anyone who eats a pizza is a happy character
2. Every foodie eats [something that is] either a pizza or a salad
3. Anyone who eats a salad is healthy
4. Every healthy person goes to the gym
5. Any nice girl does not date anyone who is a happy character
6. Ann is a nice girl, and Peter is a foodie

Goal:

To prove/show that “If Peter does not go to gym, then Ann does not date Peter.”

First Order Logic and Clause form:

The first order logic and clause form for the above assumptions and goal are as shown in the image below:



Input for Prover9:

```
formulas(assumptions).

-Pizza(y) | -Eats(x,y) | Happy(x).

-Foodie(x) | Eats(x,F(x)).

-Foodie(x) | Pizza(F(x)) | Salad(F(x)).

-Salad(y) | -Eats(x,y) | Healthy(x).

-Healthy(x) | Gyms(x).

-Nice(x) | -Happy(y) | -Dated(x,y).

Nice(Ann).

Foodie(Peter).

end_of_list.


formulas(goals).

Gyms(Peter) | -Dated(Ann, Peter).

end_of_list.
```

Output of Prover9:

```
============================== prooftrans ============================

Prover9 (32) version Dec-2007, Dec 2007.

Process 2072 was started by hp on LAPTOP-U8QNP91K,

Sun Mar  8 20:28:01 2020

The command was "/cygdrive/c/Program Files (x86)/Prover9-Mace4/bin-win32/prover9".

============================== end of head ===========================


============================== end of input ==========================


============================== PROOF =================================


% -------- Comments from original proof --------
```

% Proof 1 at 0.00 (+ 0.03) seconds.

% Length of proof is 21.

% Level of proof is 7.

% Maximum clause weight is 4.

% Given clauses 0.

1 Gyms(Peter) | -Dated(Ann,Peter) # label(non_clause) # label(goal). [goal].

2 -Foodie(x) | Pizza(F(x)) | Salad(F(x)). [assumption].

3 -Pizza(x) | -Eats(y,x) | Happy(y). [assumption].

4 Foodie(Peter). [assumption].

5 -Foodie(x) | Eats(x,F(x)). [assumption].

6 -Foodie(x) | Salad(F(x)) | -Eats(y,F(x)) | Happy(y). [resolve(2,b,3,a)].

7 Salad(F(Peter)) | -Eats(x,F(Peter)) | Happy(x). [resolve(6,a,4,a)].

8 -Salad(x) | -Eats(y,x) | Healthy(y). [assumption].

9 -Eats(x,F(Peter)) | Happy(x) | -Eats(y,F(Peter)) | Healthy(y). [resolve(7,a,8,a)].

10 -Healthy(x) | Gyms(x). [assumption].

11 Nice(Ann). [assumption].

12 -Nice(x) | -Happy(y) | -Dated(x,y). [assumption].

13 -Eats(x,F(Peter)) | Happy(x) | -Eats(y,F(Peter)) | Gyms(y). [resolve(9,d,10,a)].

14 -Gyms(Peter). [deny(1)].

15 -Happy(x) | -Dated(Ann,x). [resolve(11,a,12,a)].

16 Dated(Ann,Peter). [deny(1)].

17 -Happy(Peter). [resolve(15,b,16,a)].

18 -Eats(x,F(Peter)) | Happy(x) | -Eats(Peter,F(Peter)). [resolve(13,d,14,a)].

19 Eats(Peter,F(Peter)). [resolve(4,a,5,a)].

20 -Eats(Peter,F(Peter)) | -Eats(Peter,F(Peter)). [resolve(17,a,18,b)].

21 $F. [copy(20),merge(b),unit_del(a,19)].

============================== end of proof ==========================

Conclusion:

In the above output, we can see that Line 1 is the goal, while Lines 14 and 16 are the negation of the goal. Lines 2-5, 8, 10-12 are the sentences in the assumption. Rest of the lines are binary resolutions between two sentences occurring above it (not necessarily immediately above).

Line 21 indicates a contradiction between sentences 20 and 19, **so we can conclude that if Peter does not go to gym, then Ann does not date Peter**.