Task 1

Question 1: Is John a child or an adult?

* Goal: **adult(John)**
* Assumptions:
  + Only an adult can buy goods from a shop.
    - **customer(x) & store(y) & goods(z) & buys(x, z, y, t) -> adult(x).**
  + Child cannot be an adult
    - **child(x) -> - adult(x).**
* Situation Knowledge:
  + John went to the supermarket yesterday, and bought Ground beef, and tomatoes.
    - **buys(John, Tomatoes, Safeway, t).**
    - **buys(John, GroundBeef, Safeway, t).**

Question 2: Does John now have at least two tomatoes?

* Goal:
* Situation Knowledge:
  + John went to the supermarket yesterday, and bought Ground beef, and tomatoes.
    - **buys(John, Tomatoes, Safeway, t).**
    - **buys(John, GroundBeef, Safeway, t).**

Question 3: Did John buy any meat?

* Goal: **exists x (buys(John, x, Safeway, t) & meat(x))**
* Assumptions:
  + Produce derived from animals are considered meat
    - animal\_produce(x) -> meat(x).
* General Knowledge:
  + Groundbeef is a type of animal produce
    - animal\_produce(GroundBeef).
* Situation Knowledge:
  + John went to the supermarket yesterday, and bought Ground beef, and tomatoes.
    - **buys(John, Tomatoes, Safeway, t).**
    - **buys(John, GroundBeef, Safeway, t).**

Question 4: If Mary was buying tomatoes at the same time as John, did he see her?

* Goal: **sees(John, Mary).**
* Assumptions:
  + If a person x buys product p at a store s at the same time as a person y, it implies person x saw person y
    - **all x all p all s all t(buys(x, p, s, t) & buys(y, p, s, t) & products(z) & super\_market(s) -> sees(x, y)).**
* Situation Knowledge:
  + John went to the supermarket yesterday, and bought Ground beef, and tomatoes.
    - **buys(John, Tomatoes, Safeway, t).**
    - **buys(John, GroundBeef, Safeway, t).**
  + Mary was buying tomatoes at the same time as John
    - **buys(Mary, Tomatoes, Safeway, t).**

Question 5: Are the tomatoes made in the Supermarket?

* Goal: As Safeway doesn’t produce the tomatoes
  + **-produced(Tomatoes, Safeway).**
* General Knowledge:
  + **goods(x) & store(y) -> - produced(x, y).**

Question 6: What is John going to do with the tomatoes?

* Goal: John eats tomatoes
  + **eats(John, Tomatoes)**
* Assumptions:
  + Any food purchased at a supermarket can be eaten
    - **customer(x) & buys(x, y, z, t) & food(y) & super\_market(z) -> eats(x, y).**
* Situation Knowledge:
  + John went to the supermarket yesterday, and bought Ground beef, and tomatoes.
    - **buys(John, Tomatoes, Safeway, t).**
    - **buys(John, GroundBeef, Safeway, t).**

Question 7: Does Safeway sells deodorant?

* Goal: sells(Safeway, Deodrant)
* Assumptions:
  + Supermarket sells products
    - super\_market(x) & products(y) -> sells(x, y).
* General Knowledge:
  + Deodrant is a product
    - Product(Deodrant).

Question 8: Did John bring some money or a credit card to the supermarket?

* Goal: **brings(John, Cash, Safeway) | brings(John, Card, Safeway).**
* Assumptions:
  + If a customer buys something, it implies that person brings money
    - all x all y all z all c(customer(x) & goods(y) & buys(x, y, z, t) & money(c) -> brings(x, c, z)).
* Situation Knowledge:
  + John went to the supermarket yesterday, and bought Ground beef, and tomatoes.
    - **buys(John, Tomatoes, Safeway, t).**
    - **buys(John, GroundBeef, Safeway, t).**

Question 9: Does John have less money after going to the supermarket

exists x(money(x) & less(John, x)).

* Assumptions:
* Situation Knowledge:

Question 10: Are there are other people in Safeway while John is there?

* Goal:
* Assumptions
* Situation Knowledge

Question 11: Is John Vegetarian?

* Goal:
* Assumptions
* Situation Knowledge

Question 12: Who owns the deodorant in Safeway?

* Goal:
* Assumptions
* Situation Knowledge

Question 13: Did John have an ounce of ground beef?

* Goal:
* Assumptions
* Situation Knowledge

Question 14: Does the Shell station next door have any gas?

* Goal:
* Assumptions
* Situation Knowledge

Question 15: Do the tomatoes fit John’s car trunk?

* Goal:
* Assumptions
* Situation Knowledge

Task 2

Assumptions.txt

customer(John).

customer(Mary).

super\_market(Safeway).

gas\_station(Shell).

super\_market(x) -> store(x).

gas\_station(x) -> store(x).

vegetables(Tomatoes).

animal\_produce(GroundBeef).

money(Card).

money(Cash).

cosmetic(Deodrant).

fuel(Gas).

fuel(x) -> goods(x).

buys(John, Tomatoes, Safeway, t).

buys(John, GroundBeef, Safeway, t).

buys(Mary, Tomatoes, Safeway, t).

% If x is an animal produce then it is meat.

animal\_produce(x) -> meat(x).

% Supermarkets sells cosmetics, food, home\_decors

cosmetic(x) -> products(x).

meat(x) -> food(x).

vegetables(x) -> food(x).

food(x) -> products(x).

products(x) -> goods(x).

fuel(x) -> - eatable(x).

cosmetic(x) -> - eatable(x).

food(x) -> eatable(x).

% if a customer buys meat from the super market, he eats meat

customer(x) & buys(x, y, z, t) & food(y) & super\_market(z) -> eats(x, y).

% A person eats Anything eatable.

% person(x) & eatable(y) & - vegetarian(x) -> eats(x, y).

% Customer is a person

customer(x) -> person(x).

% person is a child or an adult

person(x) -> child(x) | adult(x).

child(x) -> - adult(x).

% Customer who buys goods from Store is an adult

customer(x) & store(y) & goods(z) & buys(x, z, y, t) -> adult(x).

% super\_market sells products

super\_market(x) & products(y) -> sells(x, y).

% gas\_station sells fuel

gas\_station(x) & fuel(y) -> sells(x, y).

% Anything that is food will be eaten by a person.

%customer(x) & goods(y) & store(z) & uses(x, m) & money(m) -> buys(x, y, z).

% Goods in the store are owned by the store

products(x) & super\_market(y) -> owns(y, x).

fuel(x) & super\_market(y) -> - owns(y, x).

super\_market(x) -> - gas\_station(x).

products(x) & gas\_station(y) -> - owns(y, x).

fuel(x) & gas\_station(y) -> owns(y, x).

% Goods in the store are not produced by the store

goods(x) & store(y) -> - produced(x, y).

%fuel(x) & gas\_station(y) -> - produced(x, y).

all x all p all s all t(buys(x, p, s, t) & buys(y, p, s, t) & products(z) & super\_market(s) -> sees(x, y)).

% Money is either cash or card

money(Cash).

money(Card).

all x all y all z all c(customer(x) & goods(y) & buys(x, y, z, t) & money(c) -> brings(x, c, z)).

eats(x, y) & meat(y) & person(x) -> -vegetarian(x).

buys(x, y, s, t) -> visits(x, s, t).

visits(x, y, t) & person(x) & store(y) -> at(x, y, t).

staff(x) & store(y) -> at(x, y, t).

staff(x) -> -customer(x).

weight(GroundBeef, One) & pound(One).

pound(One) -> ounces(Sixteen).

ounces(Sixteen) -> ounces(One).

all x (exists y buys(x, y, z, t) & products(y) -> have(x, y)).

all x (exists y products(x) -> pound(y) & weight(x, y)).

all x all y all s all t(buys(x, y, s, t)-> exists w(money(w) & less(x, w))).

% Any trunk has some capacity.

% Any product has some weight.

% A trunk can fit any product whose weight is less than the trunk's capacity.

% 2 is less than 100.

% Capacity of trunk is 100.

weight(Tomatoes, Two).

capacity(t, Hundred) & trunk(t).

all x (exists z products(x) -> weight(x, z)).

all x (exists y trunk(x) -> capacity(x, y)).

all x all z (exists y exists t weight(x, z) & capacity(t, y) & products(x) & trunk(t) & less\_weight(z, y) -> fits(t, x)).

less\_weight(Two, Hundred).

Goals.txt

adult(John).

eats(John, Tomatoes).

-produced(Tomatoes, Safeway).

-vegetarian(John).

sells(Shell, Gas).

sees(John, Mary).

sells(Safeway, Deodrant).

owns(Safeway, Deodrant).

brings(John, Cash, Safeway) | brings(John, Card, Safeway).

exists x (buys(John, x, Safeway, t) & meat(x)).

all t exists x(at(John, Safeway, t) & at(x, Safeway, t) & -customer(x) -> staff(x)).

have(John, GroundBeef) & weight(GroundBeef, One) & ounces(One).

exists x(money(x) & less(John, x)).

exists t(fits(t, Tomatoes) & trunk(t)).

Task 3

1. adult(John).

-------- Proof 1 --------

THEOREM PROVED

------ process 7800 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.01) seconds.

% Length of proof is 25.

% Level of proof is 8.

% Maximum clause weight is 0.000.

% Given clauses 0.

1 super\_market(x) -> store(x) # label(non\_clause). [assumption].

7 vegetables(x) -> food(x) # label(non\_clause). [assumption].

8 food(x) -> products(x) # label(non\_clause). [assumption].

9 products(x) -> goods(x) # label(non\_clause). [assumption].

17 customer(x) & store(y) & goods(z) & buys(x,z,y,t) -> adult(x) # label(non\_clause). [assumption].

43 adult(John) # label(non\_clause) # label(goal). [goal].

45 customer(John). [assumption].

48 -customer(x) | -store(y) | -goods(z) | -buys(x,z,y,t) | adult(x). [clausify(17)].

51 -super\_market(x) | store(x). [clausify(1)].

52 super\_market(Safeway). [assumption].

66 -vegetables(x) | food(x). [clausify(7)].

67 vegetables(Tomatoes). [assumption].

89 food(Tomatoes). [resolve(66,a,67,a)].

90 -food(x) | products(x). [clausify(8)].

96 -products(x) | goods(x). [clausify(9)].

106 products(Tomatoes). [resolve(89,a,90,a)].

118 -store(x) | -goods(y) | -buys(John,y,x,t) | adult(John). [resolve(48,a,45,a)].

127 goods(Tomatoes). [resolve(106,a,96,a)].

166 -adult(John). [deny(43)].

170 -store(x) | -buys(John,Tomatoes,x,t) | adult(John). [resolve(127,a,118,b)].

175 store(Safeway). [resolve(51,a,52,a)].

184 -store(x) | -buys(John,Tomatoes,x,t). [resolve(170,c,166,a)].

205 buys(John,Tomatoes,Safeway,t). [assumption].

211 -buys(John,Tomatoes,Safeway,t). [resolve(184,a,175,a)].

218 $F. [resolve(211,a,205,a)].

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

2. eats(John, Tomatoes).

-------- Proof 1 --------

THEOREM PROVED

------ process 7802 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 16.

% Level of proof is 6.

% Maximum clause weight is 0.000.

% Given clauses 0.

7 vegetables(x) -> food(x) # label(non\_clause). [assumption].

13 customer(x) & buys(x,y,z,t) & food(y) & super\_market(z) -> eats(x,y) # label(non\_clause). [assumption].

43 eats(John,Tomatoes) # label(non\_clause) # label(goal). [goal].

44 -customer(x) | -buys(x,y,z,t) | -food(y) | -super\_market(z) | eats(x,y). [clausify(13)].

45 customer(John). [assumption].

52 super\_market(Safeway). [assumption].

58 -buys(John,x,y,t) | -food(x) | -super\_market(y) | eats(John,x). [resolve(44,a,45,a)].

66 -vegetables(x) | food(x). [clausify(7)].

67 vegetables(Tomatoes). [assumption].

89 food(Tomatoes). [resolve(66,a,67,a)].

92 -buys(John,x,Safeway,t) | -food(x) | eats(John,x). [resolve(58,c,52,a)].

165 -buys(John,Tomatoes,Safeway,t) | eats(John,Tomatoes). [resolve(89,a,92,b)].

166 -eats(John,Tomatoes). [deny(43)].

200 -buys(John,Tomatoes,Safeway,t). [resolve(165,b,166,a)].

201 buys(John,Tomatoes,Safeway,t). [assumption].

223 $F. [resolve(200,a,201,a)].

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

3. -produced(Tomatoes, Safeway).

-------- Proof 1 --------

THEOREM PROVED

------ process 7804 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 21.

% Level of proof is 7.

% Maximum clause weight is 0.000.

% Given clauses 0.

1 super\_market(x) -> store(x) # label(non\_clause). [assumption].

7 vegetables(x) -> food(x) # label(non\_clause). [assumption].

8 food(x) -> products(x) # label(non\_clause). [assumption].

9 products(x) -> goods(x) # label(non\_clause). [assumption].

25 goods(x) & store(y) -> -produced(x,y) # label(non\_clause). [assumption].

43 -produced(Tomatoes,Safeway) # label(non\_clause) # label(goal). [goal].

51 -super\_market(x) | store(x). [clausify(1)].

52 super\_market(Safeway). [assumption].

66 -vegetables(x) | food(x). [clausify(7)].

67 vegetables(Tomatoes). [assumption].

89 food(Tomatoes). [resolve(66,a,67,a)].

90 -food(x) | products(x). [clausify(8)].

96 -products(x) | goods(x). [clausify(9)].

106 products(Tomatoes). [resolve(89,a,90,a)].

117 -goods(x) | -store(y) | -produced(x,y). [clausify(25)].

127 goods(Tomatoes). [resolve(106,a,96,a)].

166 produced(Tomatoes,Safeway). [deny(43)].

168 -store(x) | -produced(Tomatoes,x). [resolve(127,a,117,a)].

171 store(Safeway). [resolve(51,a,52,a)].

182 -store(Safeway). [resolve(168,b,166,a)].

222 $F. [resolve(182,a,171,a)].

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

4. -vegetarian(John).

-------- Proof 1 --------

THEOREM PROVED

------ process 7806 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.01) seconds.

% Length of proof is 27.

% Level of proof is 7.

% Maximum clause weight is 0.000.

% Given clauses 0.

4 animal\_produce(x) -> meat(x) # label(non\_clause). [assumption].

6 meat(x) -> food(x) # label(non\_clause). [assumption].

13 customer(x) & buys(x,y,z,t) & food(y) & super\_market(z) -> eats(x,y) # label(non\_clause). [assumption].

14 customer(x) -> person(x) # label(non\_clause). [assumption].

28 eats(x,y) & meat(y) & person(x) -> -vegetarian(x) # label(non\_clause). [assumption].

43 -vegetarian(John) # label(non\_clause) # label(goal). [goal].

44 -customer(x) | -buys(x,y,z,t) | -food(y) | -super\_market(z) | eats(x,y). [clausify(13)].

45 customer(John). [assumption].

47 -customer(x) | person(x). [clausify(14)].

52 super\_market(Safeway). [assumption].

58 -buys(John,x,y,t) | -food(x) | -super\_market(y) | eats(John,x). [resolve(44,a,45,a)].

68 -animal\_produce(x) | meat(x). [clausify(4)].

69 animal\_produce(GroundBeef). [assumption].

86 meat(GroundBeef). [resolve(68,a,69,a)].

87 -meat(x) | food(x). [clausify(6)].

88 -eats(x,y) | -meat(y) | -person(x) | -vegetarian(x). [clausify(28)].

92 -buys(John,x,Safeway,t) | -food(x) | eats(John,x). [resolve(58,c,52,a)].

94 food(GroundBeef). [resolve(86,a,87,a)].

108 person(John). [resolve(47,a,45,a)].

112 -eats(x,GroundBeef) | -person(x) | -vegetarian(x). [resolve(86,a,88,b)].

165 -eats(John,GroundBeef) | -vegetarian(John). [resolve(112,b,108,a)].

166 vegetarian(John). [deny(43)].

196 -eats(John,GroundBeef). [resolve(165,b,166,a)].

199 -buys(John,GroundBeef,Safeway,t) | eats(John,GroundBeef). [resolve(94,a,92,b)].

203 buys(John,GroundBeef,Safeway,t). [assumption].

220 -buys(John,GroundBeef,Safeway,t). [resolve(196,a,199,b)].

223 $F. [resolve(220,a,203,a)].

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

5. sells(Shell, Gas).

-------- Proof 1 --------

THEOREM PROVED

------ process 7808 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.01) seconds.

% Length of proof is 9.

% Level of proof is 4.

% Maximum clause weight is 0.000.

% Given clauses 0.

19 gas\_station(x) & fuel(y) -> sells(x,y) # label(non\_clause). [assumption].

43 sells(Shell,Gas) # label(non\_clause) # label(goal). [goal].

61 gas\_station(Shell). [assumption].

62 -gas\_station(x) | -fuel(y) | sells(x,y). [clausify(19)].

81 fuel(Gas). [assumption].

84 -fuel(x) | sells(Shell,x). [resolve(62,a,61,a)].

165 sells(Shell,Gas). [resolve(84,a,81,a)].

166 -sells(Shell,Gas). [deny(43)].

206 $F. [resolve(165,a,166,a)].

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

6. sees(John, Mary).

-------- Proof 1 --------

THEOREM PROVED

------ process 7810 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 15.

% Level of proof is 5.

% Maximum clause weight is 10.000.

% Given clauses 5.

5 cosmetic(x) -> products(x) # label(non\_clause). [assumption].

26 (all x all p all s all t (buys(x,p,s,t) & buys(y,p,s,t) & products(z) & super\_market(s) -> sees(x,y))) # label(non\_clause). [assumption].

43 sees(John,Mary) # label(non\_clause) # label(goal). [goal].

52 super\_market(Safeway). [assumption].

57 -buys(x,y,z,u) | -buys(w,y,z,u) | -products(v5) | -super\_market(z) | sees(x,w). [clausify(26)].

77 -cosmetic(x) | products(x). [clausify(5)].

78 cosmetic(Deodrant). [assumption].

95 products(Deodrant). [resolve(77,a,78,a)].

104 -buys(x,y,Safeway,z) | -buys(u,y,Safeway,z) | -products(w) | sees(x,u). [resolve(57,d,52,a)].

165 -buys(x,y,Safeway,z) | -buys(u,y,Safeway,z) | sees(x,u). [resolve(95,a,104,c)].

166 -sees(John,Mary). [deny(43)].

201 buys(John,Tomatoes,Safeway,t). [assumption].

203 buys(Mary,Tomatoes,Safeway,t). [assumption].

206 -buys(John,x,Safeway,y) | -buys(Mary,x,Safeway,y). [resolve(165,c,166,a)].

209 $F. [resolve(206,a,201,a),unit\_del(a,203)].

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

7. sells(Safeway, Deodrant).

-------- Proof 1 --------

THEOREM PROVED

------ process 7812 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 12.

% Level of proof is 4.

% Maximum clause weight is 0.000.

% Given clauses 0.

5 cosmetic(x) -> products(x) # label(non\_clause). [assumption].

18 super\_market(x) & products(y) -> sells(x,y) # label(non\_clause). [assumption].

43 sells(Safeway,Deodrant) # label(non\_clause) # label(goal). [goal].

52 super\_market(Safeway). [assumption].

53 -super\_market(x) | -products(y) | sells(x,y). [clausify(18)].

77 -cosmetic(x) | products(x). [clausify(5)].

78 cosmetic(Deodrant). [assumption].

95 products(Deodrant). [resolve(77,a,78,a)].

102 -products(x) | sells(Safeway,x). [resolve(53,a,52,a)].

166 -sells(Safeway,Deodrant). [deny(43)].

167 sells(Safeway,Deodrant). [resolve(95,a,102,a)].

206 $F. [resolve(167,a,166,a)].

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

8. owns(Safeway, Deodrant).

-------- Proof 1 --------

THEOREM PROVED

------ process 7814 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 12.

% Level of proof is 4.

% Maximum clause weight is 0.000.

% Given clauses 0.

5 cosmetic(x) -> products(x) # label(non\_clause). [assumption].

20 products(x) & super\_market(y) -> owns(y,x) # label(non\_clause). [assumption].

43 owns(Safeway,Deodrant) # label(non\_clause) # label(goal). [goal].

52 super\_market(Safeway). [assumption].

54 -products(x) | -super\_market(y) | owns(y,x). [clausify(20)].

77 -cosmetic(x) | products(x). [clausify(5)].

78 cosmetic(Deodrant). [assumption].

95 products(Deodrant). [resolve(77,a,78,a)].

103 -products(x) | owns(Safeway,x). [resolve(54,b,52,a)].

166 -owns(Safeway,Deodrant). [deny(43)].

168 owns(Safeway,Deodrant). [resolve(95,a,103,a)].

202 $F. [resolve(168,a,166,a)].

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

9. brings(John, Cash, Safeway) | brings(John, Card, Safeway).

-------- Proof 1 --------

THEOREM PROVED

------ process 7816 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 22.

% Level of proof is 7.

% Maximum clause weight is 0.000.

% Given clauses 0.

7 vegetables(x) -> food(x) # label(non\_clause). [assumption].

8 food(x) -> products(x) # label(non\_clause). [assumption].

9 products(x) -> goods(x) # label(non\_clause). [assumption].

27 (all x all y all z all c (customer(x) & goods(y) & buys(x,y,z,t) & money(c) -> brings(x,c,z))) # label(non\_clause). [assumption].

43 brings(John,Cash,Safeway) | brings(John,Card,Safeway) # label(non\_clause) # label(goal). [goal].

45 customer(John). [assumption].

49 -customer(x) | -goods(y) | -buys(x,y,z,t) | -money(u) | brings(x,u,z). [clausify(27)].

66 -vegetables(x) | food(x). [clausify(7)].

67 vegetables(Tomatoes). [assumption].

70 -goods(x) | -buys(John,x,y,t) | -money(z) | brings(John,z,y). [resolve(49,a,45,a)].

71 money(Card). [assumption].

89 food(Tomatoes). [resolve(66,a,67,a)].

90 -food(x) | products(x). [clausify(8)].

96 -products(x) | goods(x). [clausify(9)].

106 products(Tomatoes). [resolve(89,a,90,a)].

120 -goods(x) | -buys(John,x,y,t) | brings(John,Card,y). [resolve(70,c,71,a)].

127 goods(Tomatoes). [resolve(106,a,96,a)].

167 -brings(John,Card,Safeway). [deny(43)].

179 -buys(John,Tomatoes,x,t) | brings(John,Card,x). [resolve(127,a,120,a)].

226 buys(John,Tomatoes,Safeway,t). [assumption].

230 -buys(John,Tomatoes,Safeway,t). [resolve(179,b,167,a)].

247 $F. [resolve(230,a,226,a)].

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

10. exists x (buys(John, x, Safeway, t) & meat(x)).

-------- Proof 1 --------

THEOREM PROVED

------ process 7818 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 9.

% Level of proof is 4.

% Maximum clause weight is 0.000.

% Given clauses 0.

4 animal\_produce(x) -> meat(x) # label(non\_clause). [assumption].

43 (exists x (buys(John,x,Safeway,t) & meat(x))) # label(non\_clause) # label(goal). [goal].

68 -animal\_produce(x) | meat(x). [clausify(4)].

69 animal\_produce(GroundBeef). [assumption].

86 meat(GroundBeef). [resolve(68,a,69,a)].

89 -buys(John,x,Safeway,t) | -meat(x). [deny(43)].

199 -buys(John,GroundBeef,Safeway,t). [resolve(86,a,89,b)].

201 buys(John,GroundBeef,Safeway,t). [assumption].

222 $F. [resolve(199,a,201,a)].

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

11. all t exists x(at(John, Safeway, t) & at(x, Safeway, t) & -customer(x) -> staff(x)).

-------- Proof 1 --------

THEOREM PROVED

------ process 7820 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 4.

% Level of proof is 2.

% Maximum clause weight is 0.000.

% Given clauses 0.

43 (all t exists x (at(John,Safeway,t) & at(x,Safeway,t) & -customer(x) -> staff(x))) # label(non\_clause) # label(goal). [goal].

45 customer(John). [assumption].

51 -customer(x). [deny(43)].

98 $F. [resolve(51,a,45,a)].

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

12. have(John, GroundBeef) & weight(GroundBeef, One) & ounces(One).

-------- Proof 1 --------

THEOREM PROVED

------ process 7822 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 29.

% Level of proof is 8.

% Maximum clause weight is 4.000.

% Given clauses 2.

4 animal\_produce(x) -> meat(x) # label(non\_clause). [assumption].

6 meat(x) -> food(x) # label(non\_clause). [assumption].

8 food(x) -> products(x) # label(non\_clause). [assumption].

33 weight(GroundBeef,One) & pound(One) # label(non\_clause). [assumption].

34 pound(One) -> ounces(Sixteen) # label(non\_clause). [assumption].

35 ounces(Sixteen) -> ounces(One) # label(non\_clause). [assumption].

36 (all x ((exists y buys(x,y,z,t)) & products(y) -> have(x,y))) # label(non\_clause). [assumption].

43 have(John,GroundBeef) & weight(GroundBeef,One) & ounces(One) # label(non\_clause) # label(goal). [goal].

68 -animal\_produce(x) | meat(x). [clausify(4)].

69 animal\_produce(GroundBeef). [assumption].

86 meat(GroundBeef). [resolve(68,a,69,a)].

87 -meat(x) | food(x). [clausify(6)].

90 -food(x) | products(x). [clausify(8)].

94 food(GroundBeef). [resolve(86,a,87,a)].

97 -buys(x,y,z,t) | -products(u) | have(x,u). [clausify(36)].

107 products(GroundBeef). [resolve(94,a,90,a)].

132 -have(John,GroundBeef) | -weight(GroundBeef,One) | -ounces(One). [deny(43)].

133 weight(GroundBeef,One). [clausify(33)].

141 -pound(One) | ounces(Sixteen). [clausify(34)].

142 pound(One). [clausify(33)].

200 -have(John,GroundBeef) | -ounces(One). [resolve(132,b,133,a)].

203 -buys(x,y,z,t) | have(x,GroundBeef). [resolve(107,a,97,b)].

205 buys(John,Tomatoes,Safeway,t). [assumption].

225 -ounces(One) | -buys(John,x,y,t). [resolve(200,a,203,b)].

226 -ounces(Sixteen) | ounces(One). [clausify(35)].

227 ounces(Sixteen). [resolve(141,a,142,a)].

228 -ounces(One). [resolve(225,b,205,a)].

229 ounces(One). [ur(226,a,227,a)].

230 $F. [resolve(229,a,228,a)].

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

13. exists x(money(x) & less(John, x)).

-------- Proof 1 --------

THEOREM PROVED

------ process 7824 exit (max\_proofs) ------

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

% Proof 1 at 0.01 (+ 0.00) seconds.

% Length of proof is 10.

% Level of proof is 5.

% Maximum clause weight is 5.000.

% Given clauses 0.

38 (all x all y all s all t (buys(x,y,s,t) -> (exists w (money(w) & less(x,w))))) # label(non\_clause). [assumption].

43 (exists x (money(x) & less(John,x))) # label(non\_clause) # label(goal). [goal].

70 -money(x) | -less(John,x). [deny(43)].

75 -buys(x,y,z,u) | money(f1(x,y,z,u)). [clausify(38)].

145 -buys(x,y,z,u) | less(x,f1(x,y,z,u)). [clausify(38)].

147 -less(John,f1(x,y,z,u)) | -buys(x,y,z,u). [resolve(70,a,75,b)].

205 buys(John,GroundBeef,Safeway,t). [assumption].

209 -buys(John,x,y,z) | -buys(John,x,y,z). [resolve(147,a,145,b)].

210 -buys(John,x,y,z). [copy(209),merge(b)].

211 $F. [resolve(210,a,205,a)].

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