CS 6364. 001

Question-Answering System

Radhika Kulkarni

(rxk180002)

Description of the System

The project aims at developing knowledge base to answer a series of questions based on a preliminary statement. This project has been adapted from Problems 12.5 and 12.6 of Artificial Intelligence – A Modern Approach textbook by Russell ad Norvig.

Initially, the content of the preliminary sentence is represented as a series of assertions. To answer the questions, background knowledge has been added to the knowledge base. The formalized domain is enough to answer a series of questions about the information in the sentence:

'Yesterday John went to the North Berkeley Safeway supermarket and bought two pounds of tomatoes and a pound of ground beef.'

The logical reasoning system demonstrates the sufficiency of the knowledge base. The knowledge base is especially designed to answer questions related to the 'meat' in the supermarket. The knowledge base considers the hypernyms, hyponyms and holonyms of 'meat', 'fish', 'chicken', 'beef'. A python script file is written to extract these features using WordNet.

The knowledge base contains a set of assertions which are queried sequentially by Prolog. Each query of the statement logically follows the previous one, and on successfully satisfying all the necessary conditions. Prolog is a programming language particularly well suited to logic and artificial intelligence programming. In Prolog, the program logic is expressed in terms of relations, represented as facts and rules.

Knowledge Base

- John shopped beef at North Berkeley Safeway yesterday.

  shop(john,beef,northBerkeleySafeway,yesterday).
- John shopped tomatoes at North Berkeley Safeway yesterday.

  shop(john,tomatoes,northBerkeleySafeway,yesterday).
- John lives alone

  livesAlone(john).
- Anyone who lives alone shops alone

  shopsAlone(X):-livesAlone(X).
- If someone shops alone and if shops, then it's an adult

  isa(X,adult):-shop(X,_,_,_),shopsAlone(X).
- If someone is not an adult, then he is a child

  isa(X,child):- !,\+ isa(X,adult).
- If someone is not an adult, then he is a child

  isa(X,child):- isa(X,adult), !, fail.
- Tomatoes have a unit weight of 0.5 pounds

  weightPerUnitOf (tomatoes,0.5).
- If John shopped for beef, he bought 1 pound of beef

  pounds(john,1,beef):-shop(john,beef,northBerkeleySafeway,yesterday).
- If John shopped for tomatoes, he bought 2 pounds of tomatoes

  pounds(john,2,tomatoes):-shop(john,tomatoes,northBerkeleySafeway,yesterday).
- Mary shopped for tomatoes, she bought 1 pound of tomatoes

  pounds(mary,1,tomatoes):-shop(mary,tomatoes,northBerkeleySafeway,yesterday).
- The number of items that someone has, is obtained by the no. of pounds/unit weight.

  number(P,X,I):- weightPerUnitOf (I,Y),pounds(P,Z,I),X is Z/Y.

- If N>=M then N is larger or equal to M.

  largerorequal(N,M):-N>=M.

- If someone has some number of items and if it is larger or equal to some number, then he has at least that many numbers of tomatoes.

  hasatleast(P,Y,I):- number(P,X,I),largerorequal(X,Y).

- If someone shopped for something somewhere sometime, then they bought that thing.

  bought(X,Y):-shop(X,Y,_,_).

- If John bought something and if that thing is meat, then John bought meat

  buys(X,Y):-bought(X,Z), typeOf(Z,Y).

- Mary was buying tomatoes at North Berkeley Safeway at the same time as john.

  shop(mary,tomatoes,northBerkeleySafeway,T):-shop(john,tomatoes,northBerkeleySafeway,T).

- If someone buys something at some store at some time, then that person is at that store at that time.

  isAtStore(X,Y,Z):-shop(X,_,Y,Z).

- If someone buys something at some stall at some time, then that person is at that stall at that time.

  isAtStall(X,Y,Z):-shop(X,Y,_,Z).

- If someone is at some stall at some store, and if someone else is also at the same stall and same store, then that person sees the other person

  sees(X,V):-isAtStore(X,Y,Z),isAtStall(X,U,Z),isAtStore(V,Y,Z),isAtStall(V,U,Z).

- Or else they will not see each other.

  sees(X,V):-!, \+isAtStore(X,Y,Z); \+isAtStall(X,U,Z); \+isAtStore(V,Y,Z); \+isAtStall(V,U,Z), fail.

- If X sees Y, then Y also sees X.

  sees(Y,X):-sees(X,Y).

- If something is a vegetable, then it originates from nature.

  originates(X,nature):-typeOf(X,vegetable).

- If something is a skincare, then it does not originate in nature.

originates(X,nature):-typeOf(X,skincare), !, fail.

- If something is of form something which originates from man, then it is made in the supermarket.

  made(X,supermarket):- !,\+originates(X,nature).

- If something is a Vegetable or meat, it can be eaten.

  can(X,eaten):-typeOf(X,vegetable);typeOf(X,meat).

- Precedence of days

  after(yesterday,daybeforeyesterday).

  after(today, daybeforeyesterday).

  after(tomorrow, daybeforeyesterday).

  after(today,yesterday).

  after(tomorrow,yesterday).

  after(tomorrow,today).

- Someone has something at some time if he has shopped for it.

  has(X,Y,T):-shop(X,Y,_,T).

- Someone eats something at some time if someone has something at that time and if that thing can be eaten.

  whatWill(X,Y,T,eatThem):- has(X,Y,K), can(Y,eaten),after(T,K).

- Safeway has a branch called North Berkeley Safeway supermarket

  hasbranch(safeway,northBerkeleySafeway).

- Skin care products, vegetables, meat are available in supermarkets.

  available(skincare,supermarket).

  available(meat,supermarket).

  available(vegetable,supermarket).

- Something sells something if it has a branch which is a supermarket which has those class of products

  sells(X,Y):- hasbranch(X,U),typeOf(U,supermarket),typeOf(Y,Z),available(Z,supermarket);

  hasbranch(X,U),typeOf(U,supermarket),available(Y,supermarket).

- If anyone shops at someplace for anything at sometimes then they pay money for that thing at that time at that place.

  pay(X,Z,Y,T):-shop(X,_,Y,T).

- If someone has to pay money at some time at a store which is of some type, then they must bring that money to that place at that time.

  bring(X,Z,Y,T):-pay(X,Z,K,T),typeOf(K,Y),typeOf(Z,money).

- If anyone shops at someplace for anything at sometimes then they pay money for that thing at that time at that place.

  If someone who pays money at some time at someplace of some type, then he spent money that time in that type of place

  spentMoney(X,Z,T) :-pay(X,W,Y,T),typeOf(Y,Z),typeOf(W,money).

- Define after and before yesterday

  after(after,yesterday).

  after(yesterday,before).

- If someone spend money at someplace at some time, then he had less money after that time at the same place.

  moneyLeft(W,less,Y,T):- spentMoney(W,Y,U),after(T,U).

- If someone shops meat, then he is non-vegetarian

  nonVeg(X):- shop(X,Y,_,_),typeOf(Y,meat).

- If someone is not a non-vegetarian, then he is vegetarian

  veg(X):-!,\+nonVeg(X).

- If someone shopped 1 pound of something, then he bought 16 ounces of that thing

  ounces(X,Y,Z):-pounds(X,W,Z), Y is W*16.

- If John has some beef and if it is larger or equal to some number, then he has at least that many ounces of beef.

  hasAtleastOunces(X,U,Y):- ounces(X,V,Y),largerorequal(V,U).

- Capacity of John's car trunk is 4

  carTrunkCapacity(john,4).

- Volume occupied by each pound of tomato is 1

  volume(tomatoes,1).

- Volume occupied by each pound of beef is 1

  volume(beef,5).

- Space occupied by the item bought by a person is number of pounds of item bought times the volume occupied by the item

  spaceOccupied(P,I,X):- pounds(P,W,I),volume(I,V), X is W*V.

- If the space occupied by the number of pounds of item bought is than or equal to car trunk capacity of the person then the item fits in car.

  fitInCar(P,W):- spaceOccupied(P,W,Y),carTrunkCapacity(P,X),largerorequal(X,Y).

- If a person shops at supermarket then the supermarket is open at that time.

  If the supermarket is open, then the staff is present

  Hence when a person is at the supermarket and shops any item then there are other people in Safeway

  isOpen(P,L, T):- shop(P,_,L,T).

  staffPresent(P,L,T):-isOpen(P,L,T).

  peopleAtStore(P,L,T):- isAtStore(P,L,T),isOpen(P,L, T),staffPresent(P,L,T).

- If someone sells something, then he owns that thing

  owns(X,Y):-sells(X,Y).

- North Berkeley Supermarket is a type of Supermarket.

  typeOf(northBerkeleySafeway,supermarket).

- Soap is a type of skin care product.

  typeOf(soap,skincare).

- Tomatoes are type of vegetables

  typeOf(tomatoes,vegetable).

- Deodorant is type of skincare

  typeOf(deodorant,skincare).

- Similarly, we can list following typeOf axioms

  typeOf(creditcard,money).

  typeOf(cash,money).

  typeOf(sausage_meat,meat).

typeOf(stockfish,meat).

typeOf(anchovy,meat).

typeOf(pemmican,meat).

typeOf(horsemeat,meat).

typeOf(spawner,meat).

typeOf(spring_chicken,meat).

typeOf(roaster,meat).

typeOf(variety_meat,meat).

typeOf(hereford,meat).

typeOf(cattle,meat).

typeOf(shad,meat).

typeOf(veal,meat).

typeOf(ground_beef,meat).

typeOf(jerky,meat).

typeOf(eel,meat).

typeOf(cartilaginous_fish,meat).

typeOf(panfish,meat).

typeOf(rail,meat).

typeOf(cattalo,meat).

typeOf(rock_salmon,meat).

typeOf(sausage,meat).

typeOf(seine,meat).

typeOf(pastrami,meat).

typeOf(game,meat).

typeOf(stuff,meat).

typeOf(pork,meat).

typeOf(haddock,meat).

typeOf(quintessence,meat).

typeOf(trout,meat).

typeOf(dark_meat,meat).

typeOf(galloway,meat).

typeOf(mullet,meat).

typeOf(chicken,meat).

typeOf(cut,meat).

typeOf(raw_meat,meat).

typeOf(fingerling,meat).

typeOf(northern_snakehead,meat).

typeOf(fryer,meat).

typeOf(stew_meat,meat).

typeOf(cock,meat).

typeOf(mouthbreeder,meat).

typeOf(climbing_perch,meat).

typeOf(alewife,meat).

typeOf(smelt,meat).

typeOf(dominique,meat).

typeOf(broiler,meat).

typeOf(quiddity,meat).

typeOf(hypostasis,meat).

typeOf(young_fish,meat).

typeOf(angle,meat).

typeOf(food_fish,meat).

typeOf(bare_bones,meat).

typeOf(halal,meat).

typeOf(net_fish,meat).

typeOf(shrimp,meat).

typeOf(brail,meat).

typeOf(shark,meat).

typeOf(trawl,meat).

typeOf(charolais,meat).

typeOf(wimp,meat).

typeOf(bottom-feeder,meat).

typeOf(rhode_island_red,meat).

typeOf(bony_fish,meat).

typeOf(aberdeen_angus,meat).

typeOf(chick,meat).

typeOf(hake,meat).

typeOf(spatchcock,meat).

typeOf(longhorn,meat).

typeOf(carbonado,meat).

typeOf(still-fish,meat).

typeOf(bottom_lurkers,meat).

typeOf(escargot,meat).

typeOf(santa_gertrudis,meat).

typeOf(durham,meat).

typeOf(schrod,meat).

typeOf(meat,meat).

typeOf(gripe,meat).

typeOf(orpington,meat).

typeOf(bird,meat).

typeOf(game_fish,meat).

typeOf(hen,meat).

typeOf(crab,meat).

typeOf(prawn,meat).

typeOf(lamb,meat).

typeOf(capon,meat).

typeOf(cold_cuts,meat).

typeOf(bully_beef,meat).

typeOf(red_meat,meat).

typeOf(scallop,meat).

typeOf(fish,meat).

typeOf(salmon,meat).

typeOf(pisces,meat).

typeOf(mouton,meat).

typeOf(rough_fish,meat).

typeOf(meat,cattle).

typeOf(meat,person).

typeOf(meat,content).

typeOf(meat,aquatic_vertebrate).

typeOf(meat,weakling).

typeOf(meat,domestic_fowl).

typeOf(meat,complain).

typeOf(meat,contest).

typeOf(meat,catch).

typeOf(meat,plant_part).

typeOf(meat,poultry).

typeOf(meat,search).

typeOf(meat,objection).

typeOf(meat,food).

typeOf(chicken,food).

typeOf(fish,food).

typeOf(beef,food).

Questions

Section 12.5

a) Is John a child or an adult?

Answer:

Explanation:

John lives alone. Anyone who lives alone shops alone. So, John shops alone.

John shopped tomatoes and beef at North Berkeley Safeway yesterday.

If someone has to shop alone and shops, then it is an adult. Hence John is an adult.

Syntax Used:

john: Person 'John'

livesAlone(X): X lives alone

shopsAlone(X): X has to shop alone

shop(W, X, Y, Z): W shops for X at Y during Z

isa(X, Y): X is Y

Knowledge Base:

```
%John shopped beef at North Berkeley Safeway yesterday
shop(john,beef,northBerkeleySafeway,yesterday).
%John shopped tomatoes at North Berkeley Safeway yesterday
shop(john,tomatoes,northBerkeleySafeway,yesterday).
%John lives alone
livesAlone(john).
%Anyone who lives alone shops alone
shopsAlone(X):-livesAlone(X).
%If someone has to shop alone and shops, then it's an adult.
isa(X,adult):-shop(X,_,_,_),shopsAlone(X).
%If someone is not an adult, then he is a child
isa(X,child):- !,\+ isa(X,adult).
%If someone is not an adult, then he is a child
isa(X,child):- isa(X,adult), !, fail.
```

Prolog Demonstration:

```
% d:/Spring 2020/AI/Project/AISupermarket/kb.pl compiled 0.02 sec, 178 clauses
?- isa(john,X).
X = adult ;

?- isa(john,adult).
true ;

?- isa(john,child).
false.
```

b) Does John now have at least two tomatoes?

Answer**:**

Explanation:

If John shopped tomatoes at North Berkeley Safeway yesterday, then he bought 2 pounds of tomatoes. The unit weight of tomatoes is 0.5. If each tomato weighs 0.5 pounds and if John has 2 pounds, then John has 4 tomatoes. If John has some number of tomatoes and if it is larger or equal to some number, then he has at least that many numbers of tomatoes. Hence John has at least 2 tomatoes.

Syntax Used:

weightPerUnitOf(X, Y): 1 unit of X weighs Y pounds

pounds(X, Y, Z): X has Y pounds of Z

number(X, Y, Z): X has Y amount of Z

largerorequal(X, Y): X is larger of equal to Y

hasatleast(X, Y, Z): X has at least Y amounts of Z

Knowledge Base:

```
% Tomatoes have a unit weight of  0.5 pounds
weightPerUnitOf(tomatoes,0.5).
% If John shopped for tomatoes, he bought 2 pounds of tomatoes and 1 pound of
beef.
pounds(john,2,tomatoes):-shop(john,tomatoes,northBerkeleySafeway,yesterday).
% The number of tomatoes that john has is obtained by the no. of pounds/unit
weight.
number(P,X,I):-weightPerUnitOf(I,Y),pounds(P,Z,I),X is Z/Y.
% If N>=M then N is larger or equal to M.
largerorequal(N,M):-N>=M.
% If John has some number of tomatoes and if it is larger or equal to some number
then he has at least that many number of tomatoes.
hasatleast(P,Y,I):- number(P,X,I),largerorequal(X,Y).
```

Prolog Demonstration:

```
% d:/Spring 2020/AI/Project/AISupermarket/kb.pl compiled 0.00 sec, 0 clauses
?- hasatleast(john,2,tomatoes).
true.
```

c) Did John buy any meat?

Answer**:**

Explanation:

John shopped beef at North Berkeley Safeway yesterday. If someone shopped for something someday then they bought that thing. So, John bought beef. Beef is a type of meat. If someone bought something and if that thing is of type X, then that person bought X. John bought beef which is of type meat. Hence, John bought meat.

Syntax Used:

typeOf(X, Y): X is a type of Y

bought(X, Y): X has bought Y

buys(X, Y): X buys Y

Knowledge Base:

```
% Beef is a type of meat
typeOf(beef,meat).
% If someone shopped for something somewhere sometime, then they bought that
thing.
bought(X,Y):-shop(X,Y,_,_).
% If John bought something and if that thing is meat then John bought meat
buys(X,Y):-bought(X,Z), typeOf(Z,Y).
```

Prolog Demonstration:

```
% d:/Spring 2020/AI/Project/AISupermarket/kb.pl compiled 0.02 sec, 0 clauses
?- buys(john,meat).
true .
```

d) If Mary was buying tomatoes at the same time as John, did he see her?

Answer**:**

Explanation:

John shopped for beef and tomatoes at North Berkeley Safeway yesterday. Mary shopped tomatoes yesterday. If someone buys something at some stall at some time, then that person is at that stall at that time. Hence John and Mary North Berkeley store yesterday. If someone buys something at some stall at some time, then that person is at that stall at that time. Hence John and Mary are at the tomatoes stall yesterday. If someone is at some stall

at some store, and if someone else is also at the same stall and same store, then that person sees the other person. If X sees Y, then Y also sees X. Hence Mary and John see each other. If they are not at the same store at the same stall at the same time, then they do not see each other.

Syntax Used:

isAtStore(X, Y, Z): X is at Y store on Z day

isAtStall(X, Y, Z): X is at Y stall on Z day

sees(X,Y): X sees Y

Knowledge Base:

```
% Mary was buying tomatoes at North Berkeley Safeway at the same time as john.
shop(mary,tomatoes,northBerkeleySafeway,T):-
shop(john,tomatoes,northBerkeleySafeway,T).
% If someone buys something at some store at some time then that person is at that
store at that time.
isAtStore(X,Y,Z):-shop(X,_,Y,Z).
% If someone buys something at some stall at some time then that person is at that
stall at that time.
isAtStall(X,Y,Z):-shop(X,Y,_,Z).
% If someone is at some stall at some store, and if someone else is also at the
same stall and same store, then that person sees the other person
sees(X,V):-isAtStore(X,Y,Z),isAtStall(X,U,Z),isAtStore(V,Y,Z),isAtStall(V,U,Z).
% Or else they will not see each other.
sees(X,V):-!, \+isAtStore(X,Y,Z); \+isAtStall(X,U,Z); \+isAtStore(V,Y,Z);
\+isAtStall(V,U,Z), fail.
% If X sees Y then Y also sees X.
sees(Y,X):-sees(X,Y).
```

Prolog Demonstration:

```
% d:/spring 2020/ai/project/aisupermarket/kb compiled 0.00 sec, 0 clauses
?- sees(john,mary).
true .

?- sees(mary,john).
true .
```

e) Are the tomatoes made in the supermarket?

Answer**:**

Explanation:

Tomatoes are vegetables. If something is a vegetable, then it originates in nature. Hence tomatoes originate in nature. Soap is a skincare item. If something is skincare, then it cannot originate in nature. If something is made in nature, then it cannot be made in the

supermarket. Hence tomatoes cannot be made by the supermarket, but soap can be made by the supermarket.

Syntax Used:

originates(X, Y): X originates from Y

made(X, Y): X is made in Y

Knowledge Base:

```
% Soap is a type of skin care product.
typeOf(soap,skincare).
%Tomatoes are type of vegetables
typeOf(tomatoes,vegetable).
% If something is a vegetable then it originates from nature.
originates(X,nature):-typeOf(X,vegetable).
% If something is a skincare then it does not originate in nature.
originates(X,nature):-typeOf(X,skincare), !, fail.
%If something is of form something which originates from man, then it is made in
the supermarket.
made(X,supermarket):- !,\+originates(X,nature).
```

Prolog Demonstration:

```
% d:/spring 2020/ai/project/aisupermarket/kb compiled 0.00 sec, 0 clauses
?- made(tomatoes,supermarket).
false.

?- made(soap,supermarket).
true.
```

f) What is John going to do with the tomatoes?

Answer**:**

Explanation:

Tomatoes are type of vegetables and beef is a type of meat. If something is a vegetable or meat, it can be eaten. Hence tomatoes and beef can be eaten. Someone has something at some time if he has shopped it. Hence John has tomatoes and beef. Someone eats something at some time if someone has something at that time and if that thing can be eaten. Today and Tomorrow comes after yesterday. Hence John eats beef or tomatoes at any time after yesterday that is today or tomorrow.

Syntax Used:

can(X, Y): X can be Y

after(X, Y): X comes after Y

has(X, Y, T): X has Y on T

whatWill(W, X, Y, Z): W will do Z on X on Y day

Knowledge Base:

```
% If something is a vegetable or meat, it can be eaten.
can(X,eaten):-typeOf(X,vegetable);typeOf(X,meat).
% Precedence of days
after(yesterday,daybeforeyesterday).
after(today, daybeforeyesterday).
after(tomorrow, daybeforeyesterday).
after(today,yesterday).
after(tomorrow,yesterday).
after(tomorrow,today).
% Someone has something at some time if he has shopped for it.
has(X,Y,T):-shop(X,Y,_,T).
% Someone eats something at some time if someone has something at that time and if
that thing can be eaten.
whatWill(X,Y,T,eatThem):- has(X,Y,K), can(Y,eaten),after(T,K).
```

Prolog Demonstration:

```
% d:/Spring 2020/AI/Project/AISupermarket/kb.pl compiled 0.03 sec, 0 clauses
?- whatWill(john,tomatoes,today,X).
X = eatThem ;
```

g) Does Safeway sell deodorant?

Answer**:**

Explanation:

Safeway has a branch called North Berkeley supermarket. North Berkeley Safeway supermarket is a type of supermarket. Hence North Berkeley Safeway is also a type of supermarket. Skin care products, vegetables, meat are available in supermarkets. Deodorant is a type of skin care product. Something sells something if it has a branch which is a supermarket which has those class of products. Hence North Berkeley Safeway sells deodorant because North Berkeley Safeway is a supermarket and supermarkets have skin care products and deodorant is a skin care product.

Syntax Used:

hasBranch(X, Y): X has a branch called Y

available(X, Y): X is available at Y

sells(X, Y): X sells Y

Knowledge Base:

```
% North Berkeley Supermarket is a type of Supermarket.
typeOf(northBerkeleySafeway,supermarket).
%Deodorant is type of skincare
typeOf(deodorant,skincare).
% Safeway has a branch called North Berkeley Safeway supermarket
hasbranch(safeway,northBerkeleySafeway).
% Skin care products, vegetables, meat are available in supermarkets.
available(skincare,supermarket).
available(meat,supermarket).
available(vegetable,supermarket).
% Something sells something if it has a branch which is a supermarket which has
those class of products
sells(X,Y):-
hasbranch(X,U),typeOf(U,supermarket),typeOf(Y,Z),available(Z,supermarket);
hasbranch(X,U),typeOf(U,supermarket),available(Y,supermarket).
```

Prolog Demonstration:

```
% d:/spring 2020/ai/project/aisupermarket/kb compiled 0.00 sec, 0 clauses
?- sells(safeway,beef).
true .

?- sells(safeway,socks).
false.
```

h) Did John bring some money or a credit card to the supermarket?

Answer**:**

Explanation:

If anyone shops at someplace for anything at sometimes then they pay money for that thing at that time at that place. John shops at North Berkeley Safeway tomatoes and beef yesterday. Hence John pays money for tomatoes and beef yesterday. If someone has to pay money at some time at a store which is of some type, then they must bring that money to that place at that time. North Berkeley Safeway is a supermarket. Hence Johns brings money to the supermarket yesterday.

Syntax Used:

pay(W, X, Y, Z): W pays X to Y on Z

brings(W, X, Y, Z): W brings X to Y on Z

Knowledge Base:

```
% Credit card is a type of money
typeOf(creditcard,money).
% Cash is a type of money
typeOf(cash,money).
% If anyone shops at someplace for anything at sometimes then they pay money for
that thing at that time at that place.
pay(X,Z,Y,T):-shop(X,_,Y,T).
```

```
% If someone has to pay money at some time at a store which is of some type then
they must bring that money to that place at that time.
bring(X,Z,Y,T):-pay(X,Z,K,T),typeOf(K,Y),typeOf(Z,money).
```

Prolog Demonstration:

```
% d:/Spring 2020/AI/Project/AISupermarket/kb.pl compiled 0.00 sec, 0 clauses
?- bring(john,money,supermarket,yesterday).
true .

?- bring(john,cash,supermarket,yesterday).
true .
```

i) Does John have less money after going to the supermarket?

Answer**:**

Explanation:

If anyone shops at someplace for anything at sometimes then they pay money for that thing at that time at that place. John shops at North Berkeley Safeway tomatoes and beef yesterday. Hence John pays money for tomatoes and beef yesterday. If someone who pays money at some time at someplace of some type, then he spent money that time in that type of place. North Berkeley Safeway is a supermarket. Hence Johns spent money in a supermarket. If someone spend money at someplace at some time, then he had less money after that time at the same place. After going to supermarket is after yesterday. John spent money in that supermarket yesterday. Hence John had less money after going to the supermarket.

Syntax Used:

spentMoney(X, Y, Z): X spends money on Y at Z type of place

moneyLeft(W, X, Y, Z): W has X left at Y after Z time

Knowledge Base:

```
% If anyone shops at someplace for anything at sometimes then they pay money for
that thing at that time at that place.
% If someone who pays money at some time at someplace of some type then he spent
money that time in that type of place
spentMoney(X,Z,T) :-pay(X,W,Y,T),typeOf(Y,Z),typeOf(W,money).
% Define after and before yesterday
after(after,yesterday).
after(yesterday,before).
% If someone spend money at someplace at some time then he had less money after
that time at the same place.
moneyLeft(W,less,Y,T):- spentMoney(W,Y,U),after(T,U).
```

Prolog Demonstration:

```
% d:/Spring 2020/AI/Project/AISupermarket/kb.pl compiled 0.00 sec, 0 clauses
?- moneyLeft(john,X,supermarket,after).
X = less .
```

Section 12.6

a) Are there other people in Safeway while John is there?

Answer**:**

Explanation:

If someone shops at a store at a time, then the person is at that store at that time. If a person shops at the supermarket then the supermarket is open at that time. If a person is at the supermarket and the supermarket is open, then the staff will be present at that time. Hence, other people will be there at the store. John shops beef and tomatoes yesterday. As he shops, North Berkeley Safeway was open yesterday. As North Berkeley Safeway was open staff was present yesterday. Hence, people are there with John yesterday.

Syntax Used:

isOpen(P,L,T): L is open at T when P was there

staffPresent(P,L,T): staff is present in L at T when P was there

peopleAtStore(P,L,T): Other people are present with P at L during T

Knowledge Base:

```
%If a person shops at supermarket then the supermarket is open at that time.
isOpen(P,L,T):- shop(P,_,L,T).
%If the supermarket is open then the staff is present
staffPresent(P,L,T):-isOpen(P,L,T).
% When a person is at the supermarket and supermarket is open any item and staff
is people then there are other people in supermarket
peopleAtStore(P,L,T):-isAtStore(P,L,T),isOpen(P,L,T),staffPresent(P,L,T).
```

Prolog Demonstration:

```
% d:/spring 2020/ai/project/aisupermarket/kb compiled 0.00 sec, 0 clauses
?- peopleAtStore(john,northBerkeleySafeway,yesterday).
true .
```

b) Is John a vegetarian?

Answer**:**

Explanation:

Someone shops something. If type of that thing is meat, then he is non-vegetarian. If a person is non-vegetarian, then he is not a vegetarian. John shops beef. Beef is a type of meat. So, John is a non-vegetarian. Hence, John is not a vegetarian.

Syntax Used:

nonVeg(X): X is non vegetarian

veg(X): X is vegetarian

Knowledge Base:

```
% If someone shops meat then he is non vegetarian
nonVeg(X):- shop(X,Y,_,_),typeOf(Y,meat).
% If someone is not a non-vegetarian then he is vegetarian
veg(X):-!,\+nonVeg(X).
```

Prolog Demonstration:

```
% d:/Spring 2020/AI/Project/AISupermarket/kb.pl compiled 0.00 sec, 0 clauses
?- isVeg(john).
false.
```

c) Who owns the deodorant in Safeway?

Answer**:**

Explanation:

If someone sells something, then he owns that thing. Safeway sells deodorant hence Safeway owns deodorant.

Syntax Used:

owns(X,Y): X owns Y

Knowledge Base:

```
% If someone sells something then he owns that thing
owns(X,Y):-sells(X,Y).
```

Prolog Demonstration:

```
% d:/spring 2020/ai/project/aisupermarket/kb compiled 0.00 sec, 0 clauses
?- owns(X,deodorant).
X = safeway ;
```

d) Did John have an ounce of ground beef?

Answer**:**

Explanation:

If someone shopped 1 pound of something, then he bought 16 ounces of that thing. If John has some beef and if it is larger or equal to some number, then he has at least that many ounces of beef. Thus, John has at least 1 ounce of beef.

Syntax Used:

ounces(X,Y,Z): X has Y ounces of Z

hasAtleastOunces(X,U,Y): X has at least U ounces of Z

Knowledge Base:

```
% If someone shopped 1 pound of something, then he bought 16 ounces of that thing.
ounces(X,Y,Z):-pounds(X,W,Z), Y is W*16.
% If John has some beef and if it is larger or equal to some number then he has at
least that many ounces of beef.
hasAtleastOunces(X,U,Y):- ounces(X,V,Y),largerorequal(V,U).
```

Prolog Demonstration:

```
% d:/Spring 2020/AI/Project/AISupermarket/kb.pl compiled 0.00 sec, 0 clauses
?- hasAtleastOunces(john,1,beef).
true .
```

f) Do the tomatoes fit in John's car trunk?

Answer**:**

Explanation:

The capacity of John's car trunk is 4. The volume occupied by each pound of tomatoes is 1. John shopped 2 pounds of tomatoes. Space occupied by 2 pounds of tomatoes is 2 * 1 = 2. The space occupied by tomatoes is less than car capacity. Hence tomatoes fit in John's car trunk.

Syntax Used:

carTrunkCapacity(X,Y): Y is the capacity of X's car trunk

volume(X,Y): Y is the volume occupied by X

spaceOccupied(P,I,X): X is the space occupied by the number of pounds of I bought by P

fitInCar(P,W): W fits in P's Car

Knowledge Base:

```
% Capacity of John's car trunk is 4
carTrunkCapacity(john,4).
% Volume occupied by each pound of tomato is 1
volume(tomatoes,1).
% Volume occupied by each pound of beef is 1
volume(beef,5).
%Space occupied by the item bought by a person is number of pounds of item bought
times the volume occupied by the item
spaceOccupied(P,I,X):- pounds(P,W,I),volume(I,V), X is W*V.
%If the space occupied by the number of pounds of item bought is than or equal to
car trunk capacity of the person
%then the item fits in car.
fitInCar(P,W):- spaceOccupied(P,W,Y),carTrunkCapacity(P,X),largerorequal(X,Y).
```

Prolog Demonstration:

```
% d:/Spring 2020/AI/Project/AISupermarket/kb.pl compiled 0.02 sec, 0 clauses
?- fitInCar(john,tomatoes).
true.
```

System Performance

The system performance for various questions listed in Sections 12.5 and 12.6 can be observed as shown above. The system works best for `meat` related products as the holonyms, hypernyms and hyponyms of 'meat' are considered.

References

- 'Artificial Intelligence – A Modern Approach', Third Edition, book by Stuart Russell and Peter Norvig.
- SWI Prolog Software https://www.swi-prolog.org/
- Prolog https://en.wikipedia.org/wiki/Prolog