SENG2200/6220 –Programming Languages & Paradigms
Computer Lab for Week 12, Semester 1, 2019

1. Consider the following Prolog program:

```
parent(X,Y) :- child(Y,X).
mother(X,Y) :- parent(X,Y),female(X).
father(X,Y) :- parent(X,Y),male(X).
grandparent(X,Y) :- grandchild(Y,X).
grandmother(X,Y) :- grandparent(X,Y),female(X).
grandfather(X,Y) :- grandparent(X,Y),male(X).
grandchild(X,Y) :- parent(Z,X),parent(Y,Z).
granddaughter(X,Y) :- grandchild(X,Y),female(X).
grandson(X,Y) :- grandchild(X,Y),male(X).
daughter(X,Y) :- child(X,Y),female(X).
son(X,Y) :- child(X,Y),male(X).

male(fred).
male(bill).
male(jim).
female(sally).
female(mary).
female(kate).
female(anne).

child(bill,fred).
child(bill,sally).
child(mary,fred).
child(mary,sally).
child(kate,jim).
child(kate,mary).
child(anne,bill).
```

Trace through the following queries:

a) ?- male(X).

Fred;
Bill;
Jim;
No;

b) ?- parent(X,bill)

Looks for child(Bill, x)

Fred;
Sally;
No;

c) ?- granddaughter(kate,X).

Looks for: grandchild(Kate,X),female(Kate).
female kate is true.
Grandchild looks for: grandchild(Kate, X) :- parent(Z,Kate),parent(X,Z).
parent looks for: child(kate,X).
x = jim
Jim has no children, so clause fails
x = mary
z= fred
Output Fred;
z = sally
Output Sally;
no more z, so Output No

d) ?- grandmother(X,kate),son(Y,X),child(Z,Y)

```
grandmother(X,Kate) :- grandparent(X,Kate),female(X).
    grandparent(X,Kate) :- grandchild(Kate,X).
        grandchild(Kate,X) :- parent(Z,Kate),parent(X,Z).
            parent(Z, Kate) = child(Kate, Z)
                Z = jim
            parent(X, Jim) = child(Jim, X)
                X = n/a
                backtrack
            next: parent(Z, Kate) = child(Kate, Z)
                Z = mary
            next: parent(X,mary) = child(mary, X)
                X = fred
        X = fred
    female(fred) - fails, backtrack
            next: parent(X,mary) = child(mary, X)
                X = sally
        X = sally
    female(sally) - true
Sally;
Continue searching
            No other values of X, try new value of Z
            No other values of Z
        No other X
    No other X
No;
```

e) ?- parent(jim,X),mother(Z,X),parent(Z,Y),father(W,Y)

```
parent(jim, x)
    child(x, Jim)
        x = kate
mother(Z, kate)
    parent(x, kate), female(x)
        child(kate, x)
            x = jim
        female(jim) = false
            x = sally
        female(sally) = true
    z = sally
parent(Sally, Y)
    child(Y, sally)
        y = bill
father(w, Bill)
    parent(x, Bill), male(w)
        child(Bill, x)
            x = fred
    male(fred) is true
```

output Fred;

Backtracks and finds nothing else.

Write a Prolog program to represent the following (note definitions not entirely accurate):

A mammal is a vertebrate that is warm-blooded and has hair.
A marsupial is a mammal that has a pouch.
A monotreme is a mammal that lays eggs.

There are 10 animals, a-j, which have the following properties:

All the animals except j are vertebrates.
All the animals except i are warm-blooded.
All animals except a have hair.
Animals e, f, g, and h have pouches.
Animals b, e, g, and j lay eggs.

```
Sample:
mammal(x) :- animal(x), vertebrate(x), warmBlooded(x), hair(x).
marsupial(x) :- mammal(x), pouch(x).
monotreme(x) :- mammal(x), eggs(x).

vertebrate(x) :- NOT(notVert(x)).
warmBlooded(x) :- NOT(notWarm(x)).
hair(x) :- NOT(hairless(x))

notVert(j).
notWarm(i).
hairless(a).
pouch(e).
pouch(f).
pouch(g).
pouch(h).

egg(b).
egg(e).
egg(g).
egg(j).

animal(a).
animal(b).
animal(c).
animal(d).
animal(e).
animal(f).
animal(g).
animal(h).
animal(i).
animal(j).
```

2. Using your program from Q2, write Prolog queries to determine the following:

a. Which animals are mammals?

```
?- mammal(X)
```

b. Which monotremes are also marsupials?

```
?-monotremes(X), marsupials(X)
```

c. Which marsupials are not monotremes?

```
?-marsupials(X), NOT(monotremes(X))
```