CMPT 383
Quiz #5
November 8, 2005

1) Translate the following statement into Prolog rule(s): "Everybody who has a child is happy" (introduce a one-argument relation **happy**).

```
happy(X) :- has_child(Y).      or
happy(X) :- child(Y,X).        or
happy(X) :- parent(X,Y).
```

2) Define the relation **grandchild** using the **parent** relation.

```
grandchild(Z,X) :- parent(X,Y), parent(Y,Z).
```

3) Which of the following are syntactically correct Prolog terms? What kinds of terms are they?

a) Diana **(variable)**
b) diana **(atom)**
c) 'Diana' **(atom)**
d) _diana **(variable)**
e) 'Diana goes south' **(atom)**
f) goes(diana,south) **(structure)**
g) 45 **(number)**
h) 5(X,Y) **(invalid)**
i) +(north,west) **(structure)**
j) three(Black(Cats)) **(invalid)**

4) Will the following matching operations succeed of fail? If they succeed, what are the resulting instantiations of variables?

a) point(A,B) = point(1,2) **{ A = 1, B = 2 }**
b) point(A,B) = point(X, Y, Z) **fail**
c) plus(2,2) = 4 **fail**
d) +(2,D) = +(E,2) **{ D = 2, E = 2 }**
e) triangle(point(-1,0),P2,P3) = triangle(P1,point(1,0),point(0,Y))
{ P1 = point(-1,0), P2 = point(1,0), P3 = point(0,Y) }

5) Rewrite the following program without using the semicolon notation:

```
translate(Number,Word) :-
            Number = 1, Word = one;
            Number = 2, Word = two;
            Number = 3, Word = three.
translate(1,one).
translate(2,two).
translate(3,three).
```

6) Define the relation

```
max(X,Y,Max)
```

so that `Max` is the greater of two numbers `X` and `Y`.

```
max(X,Y,X) :- X >= Y.
max(X,Y,Y) :- X < Y.
```

or

```
max(X,Y,X) :- X >= Y, !.
max(_,Y,Y).
```

7) Let a program be:

```
p(1).
p(2) :- !.
p(3).
```

Write all Prolog's answer to the following questions:

a) `?- p(X).`

```
            X = 1;
            X = 2.
```

b)
```
?- p(X),p(Y).
            X = 1, Y = 1;
            X = 1, Y = 2;
            X = 2, Y = 1;
            X = 2, Y = 2.
```

c)
```
?- p(X), !, p(Y).
            X = 1, Y = 1;
            X = 1, Y = 2.
```

8) Given the following fact `located_in(austin,texas)`. A beginning Prolog student has the following dialogue with the computer:

```
?- located_in(austin,X).
X = texas
?- write(X).
X is unistantiated
```

Why did not the computer print `'texas'` the second time?

They are two unrelated queries. So, the instantiation of variable X of the first query (`located_in(austin,X)`) finishes when the query is completed. The second query uses a new uninstantiated variable.

9) Explain which of the following queries succeed, fail, or raise error conditions, and why:

a) `5 is 2+3.` **succeeds**
b) `5 =:= 2+3.` **succeeds**
c) `5 = 2+3.` **fails**
d) `4+1 is 2+3.` **fails**
e) `4+1 =:= 5.` **succeeds**
f) `What is 2+3.` **succeeds {What = 5}**
g) `What =:= 2+3.` **error**
h) `What is 5.` **succeeds {What = 5}**
i) `What = 5.` **succeeds {What = 5}**