CSc 8710 DDLP, Fall 2008 **Name:** ______________________

Exam #1

Thursday, September 25, 2008

1. Consider the following schema:

```
suppliers(SID,sname,address).
parts(PID,pname,color).
supplies(SID,PID,cost).
```

   The primary keys are shown in uppercase. The `SID` column in `supplies` is a foreign key referring to `SID` in `suppliers` and the `PID` column in `supplies` is a foreign key referring to `PID` in `parts`. The `supplies` table lists the prices charged for parts by suppliers. Write one of the following queries in **DRC** and the other in **Relational Algebra**.

   (a) Get the pnames of parts supplied by at least 3 different suppliers.

   (b) Get the pnames of the most expensive parts supplied by suppliers named *Yosemite Sham*.

2. Suppose we have a relation `Male(name)` that holds the set of names of males. There is also a relation `Marriages(name1, name2)` that relates pairs of married people. It is not determined which of a married couple is listed first, and you should not assume that `Marriages(a, b)` implies `Marriages(b, a)`. A Bachelor is a male who is not married. Write a **Datalog** and **Relational Algebra** queries that define the set of Bachelors.

3. Convert the following Datalog rule into an expression of **Relational Algebra and DRC** for relation P.
   `p(X,Y) :- r(X,Z,Z), s(Z,Y,Y).`

4. For each of the following sets of formulas, use the MGU algorithm to determine if it has a most general unifier. If yes, what is the MGU?

```
(a) { [[X,Y],Z|R], [[a,b],[1,2],[c,d]] }
(b) { k(X,g(Y,a),Y), k(g(g(U,Z),Z),x,g(Z,a)) }
(c) { p(X,X), p(f(a,Y),f(Z,b)) }
```

5. Consider the following Prolog program (Note: Rule numbers are not part of the program - you should cite them in the SLD refutation tree):

```
(R1) p(a).                        (R7)  q(X) :- s(X).
(R2) p(X) :- q(X), r(X).
(R1) p(X) :- u(X).                (R8)  r(a).
                                  (R9)  r(b).
(R4) s(a).
(R5) s(b).                        (R10) u(d).
(R6) s(c).
```

   Construct the entire SLD Refutation tree for the query: `?- p(X).` Show rule numbers and MGUs at each edge of the tree.

6. Consider the problem of manipulating matrices in Prolog. A NxM matrix of numbers is represented in Prolog as a list of rows, where each row is a list of numbers. For example, thhe matrix $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ is represented in Prolog as the list `[[1,2,3],[4,5,6]]`. Write Prolog programs for the following:

   (a) **Transpose:** The transpose of a matrix is defined as a matrix obtained by turning the rows of the given matrix into columns and the columns into rows. For instance, the transpose of $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ is $\begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$

   Here are two sample runs in SWI-Prolog:

```
?- transpose([[1,2,3],[4,5,6]],M).
M = [[1,4],[2,5],[3,6]]
Yes
?- transpose([[2,3,4,5],[7,8,9,10],[8,7,6,5]],M).
M = [[2, 7, 8], [3, 8, 7], [4, 9, 6], [5, 10, 5]]
Yes
```

   (b) **Matrix multiplication:** Recall Matrix multiplication from your Math classes. Here is an example:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \times \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1{*}5{+}2{*}7 & 1{*}6{+}2{*}8 \\ 3{*}5{+}4{*}7 & 3{*}6{+}4{*}8 \end{pmatrix} = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$

   Here are two sample runs in SWI-Prolog:

```
?- matrixMul([[1,2],[3,4]],[[5,6],[7,8]],M).
M=[[19,22],[43,50]]
Yes
?- matrixMul([[2,3,4],[5,6,7]],[[1,2,3,4],[2,3,5,6],[7,1,2,4]],M).
M = [[36, 17, 29, 42], [66, 35, 59, 84]]
Yes
```

   You do not have to check for the proper dimensions of the input matrices.

   Hint: Try using transpose.