Name ______________________________

Introduction to Expert Systems

EE 4785

Final Examination

11 May 1995, 10:00–12:00 CDT

Problem 1	________	(20 pts)
Problem 2	________	(30 pts)
Problem 3	________	(20 pts)
Problem 4	________	(30 pts)
Exam Total	________	(100 pts)

Good Luck!

Problem 1: Part of an expert system must check the parts for a top-secret military project. (Sorry, I can't tell you the details.) Parts are instances of a class `part`. The class has attributes `color, shape,` and `texture`. A proposition, `partsCheck` will be true if for each part: (1) if the part's `color` is red then its `texture` is soft or its `shape` is round, and (2) if the part's `color` is blue then there exists a different part with the same `texture`. Write a rule or rules for this proposition for the `shellOAVbc` shell. (20 pts)

The following might be data in the database:

```
data(part1,isa,part,init).
data(part1,color,blue,init).
data(part1,texture,rough,init).
data(part1,shape,cube,init).
data(part2,isa,part,init).
data(part2,color,orange,init).
```

⋮

Problem 2: Write a Prolog program to transpose a matrix. The goal `trans(Matrix,TMatrix)` should succeed if `TMatrix` and `Matrix` are bound to lists of lists (as described below) in which `TMatrix` is a transposed version of `Matrix`. Matrix $\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$ is represented in Prolog by the list of lists

$[[a_{11}, a_{12}, \dots, a_{1n}],\ [a_{21}, a_{22}, \dots, a_{2n}], \dots,\ [a_{m1}, a_{m2}, \dots, a_{mn}]]$.

(A matrix is transposed by moving the element at row i column j to row j column i.) (30 pts)

For example, let $T = \begin{pmatrix} a & b & c \\ 1 & 2 & 3 \end{pmatrix}$ transpose of T is then $X = \begin{pmatrix} a & 1 \\ b & 2 \\ c & 3 \end{pmatrix}$. In Prolog:

```
 ?- trans([[a,b,c],[1,2,3]],X).

X = [[a,1],[b,2],[c,3]]

yes
```

Let $T = \begin{pmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \end{pmatrix}$. The transpose is $X = \begin{pmatrix} 11 & 21 & 31 \\ 12 & 22 & 32 \\ 13 & 23 & 33 \end{pmatrix}$. In Prolog:

```
 ?- trans([[11,12,13],[21,22,23],[31,32,33]],X).

X = [[11,21,31],[12,22,32],[13,23,33]]

yes
```

Problem 3: Modify the OAV shell, `shellOAVbc` so that the quantified expressions bind a variable with the percentage of qualified expressions which meet the condition. The code to be modified appears below. (20 pts) For example, consider:

```
r1:if forCount(O:.isa=thing,O:.speed=fast and O:.loose=true,O:.height=3,P)

 then n:=P.
```

Suppose there were a total of 12 things, 5 of the things were fast and loose, and 2 of those had a height of 3. Then `P` would be bound to $2/5 = .4$.

```
deduce(Goal,Val):-
        Goal=..[Amt,O:.isa=Class,M,G],  % Extract parts of quantifier cmd.
        var(O),atom(Class),
        member(Amt,[forAll,forEach      % Check for valid quantifier.
        ,forSome,forNone]),
        !,                              % (Below) Get list of objects.
        setof(O,[M,From]^(data(O,isa,Class,From),deduce(M,true)),Os),
        newName(doAll,Name),            % Make a outcome str. name.
        T=doAllRec(Name,Outcome),       % Make an outcome structure.
        (
         member(O,Os),                  % Get an object.
         once (deduce(G,Outcome)),      % Test condition on object.
         once ( T ; asserta(T) ),fail   % Update outcome if new;
         ;                              %  get next object.
         true
        ),                              % (Below) Check outcomes.
        once ( retract(doAllRec(Name,true)),S=t ; true ),
        once ( retract(doAllRec(Name,false)),F=t ; true ),
        once (
          ( Amt=forNone,var(S) ; Amt=forSome,not((var(S),nonvar(T))) ;
            Amt=forAll,var(F)  ; Amt=forEach
          ),!,Val=true
          ;
          Val=false
        ).
```

Problem 4: Answer each question below.

(a) Show the state of the agenda and database before each rule fires in the execution of the following forward-chaining rules. Indicate which rule will fire in each step. The database initially contains a single fact, `start`. A higher priority number indicates higher priority. State any reasonable assumptions made about how rules are chosen.

(10 pts)

```
r1:if start then [assert a,b,c] priority 100

r2:if a or b then [assert d, retract c] priority 200

r3:if a and c then [assert e, retract a] priority 100

r4:if a and d then [assert c] priority 100

r5:if c and e then [assert f] priority 100

r6:if a or c then [assert g] priority 100
```

(b) Convert the each of the following rules to logical expressions. (10 pts)

```
r1:if a or b then c.

r2:if a and d then c.

r3:if not c then e.
```

(c) Suppose the logical expressions from the problem above are to be interpreted as axioms. When `a` and `b` are both false the rules, as interpreted by `shellPropBC`, and the axioms do not agree. Explain how and why. (10 pts)