CMSC671 - Principles of AI
Computer Science Department
University of Maryland, Baltimore County
Spring 1992

FINAL EXAMINATION

Name ____
SS# ____

1 Warm Up Exercise (0 points)

Is *no* your answer to this question.

2 Lisp (10 points)

Write recursive Lisp functions *height* and *weight* each of which take one s-expression and return an integer for the height and weight of their argument, where: takes an s-expression and returns its *height*, where

- the *height* of an atom is 0.
- the *height* of a list is one plus the maximum height of any of the elements of the list.
- the *weight* of a non-nil atom is 1.
- the *weight* of the empty list is 0.
- the *weight* of a list is the sum of the weights of its elements.

3 True/False (20 points)

Instructions: Place a T or an F in the space before each statement to indicate whether the statement is True or False. There is no penalty for incorrect answers (but then, there are no points for incorrect answers either).

1. ___ First-order predicate calculus allows quantified variables to refer to objects in the domain of discourse, and not to predicates or functions.
2. ___ Every sound inference rule is complete and every complete inference rule is sound.
3. ___ Two expressions in the propositional calculus are equivalent if they have the same value under all truth value assignments.
4. ___ If X is a variable and s is a propositional calculus sentence, then $\exists X s$ is a propositional calculus sentence.
5. ___ If D is a non-empty set representing the domain of discourse, then in an interpretation over D, each function f of arity m is defined on m arguments of D and defines a mapping from D^n into $\{T, F\}$.
6. ___ Unification is an algorithm for determining the substitutions needed to make two predicate calculus expressions match.
7. ___ A most general unifier is a unifier that is not restricted to first-order predicate calculus.
8. ___ If sentence A unifies with sentence B, and sentence B unifies with sentence C, then sentence A unifies with sentence C.
9. ___ If sentence A unifies with sentence B, and sentence A unifies with sentence C, then sentence B unifies with sentence C.
10. ___ It is possible for Algorithm A^* to expand a non-optimal solution node.
11. ___ Algorithm A will perform a breadth-first search if $H(n) = G(n)$.
12. ___ Algorithm A will perform a depth-first search if the heuristic function $H(n) = -G(n)$.
13. ___ Semantics networks lack adaquate expressive power for representing natural language menaing because one can not represent non-binary relations, since every arc connects exactly two nodes.
14. ___ The addition of inheritance to a semantic net system will usually *reduce* the size of the knowledge base.
15. ___ Of the three major inference systems, *deduction*, *induction*, and *abduction*, only the third is not sound.
16. ___ MYCIN is a goal-driven expert system.
17. ___ The main reason why MYCIN is not used by doctors today is that it can not learn to diagnose new diseases.

4 Inference (10 points)

Suppose that the letters A, B, C, and D represent the following sentences:

A:	All grapes have elbows.
B:	Belinda has elbows.
C:	Everything that has elbows can dance.
D:	Belinda is a grape.

Fill in each of the following blanks with a letter from A to D to make each of the following statements true:

1. Given sentences ___ and ___, deduction allows us to infer sentence ___.
2. Given sentences ___ and ___, induction allows us to infer sentence ___.
3. Given sentences ___ and ___, abduction allows us to infer sentence ___.

5 Unification (10 points)

For each of the following pairs of literals, state whether they unify. If they do unify, state the binding of each of the variables. If they do not unify, give the reason.

a)	p(f(Z), Y, g(Y))	p(f(X), c, g(X))
b)	p(f(X, X), a)	p(f(Y, f(Y, a)), a)
c)	p(a, X, c, X)	p(b, c, Z, c)
d)	p(f(Y), Y, g(a, Z), b)	p(f(b), W, g(X, X), W)
e)	p(X, Y)	p(a, b, c)

6 Prolog (10 points)

Consider two possible definitions of the *member* predicate:

```
member1(X,[X|Tail]).                        member2(X,[X|Tail]) :- !.
member1(X,[Y|Tail]) :- member1(X,Tail).     member2(X,[Y|Tail]) :- member2(X,Tail).
```

Both of these definitions are in fact useful. Describe in high level terms (i) how they differ; (ii) why one would chose to use one definition rather than the other and (iii) show all possible solutions to the two goals $member1(X,[1,2,3])$. and $member2(X,[1,2,3])$.

7 Expert Systems (10 Points)

Give at least four reasons why one might need to represent and reason with some certainty metric in an expert system.

8 Knowledge Representation (20 points)

Suppose we wanted to implement a simple frame-based representation language in Prolog. We might choose to represent the basic type/subtype hierarchy with an *isa*1 predicate that is true if the frame named by its first argument is an immediate subtype of on instance of the frame named by its second. Note that it is possible for a frame to have multiple parent frames.

```
% isa(<frame>,<frame>).
isa1(person,mammal).   isa1(woman,person).    isa1(woman,femaleThing).
isa1(man,person).      isa1(georgeBush,man).  isa1(man,femaleThing).
```

Write a recursive definition of the predicate *isa*, where $isa(X,Y)$ is true if X and Y are identical or there is a chain of *isa* links between frames X and Y. In the above example, we want $isa(georgeBush, mammal)$ to succeed.

We can represent slots of a frame using a predicate *has1*:

```
% has1(<frame>,<slot>,<value>).
has1(mammal,bloodtemp,warm).         has1(georgeBush,job,president).
has1(person,intelligence,high).     has1(georgeBush,intelligence,very_high).
```

Define the predicate $has(F,S,V)$ that is true if frame F has or inherits a slot S with value V and does not have or inherit a slot S with value $V2$ (different from V) that is "closer" than the value V. You should support the following dialogue:

```
:- has(georgeBush,bloodtemp,T)                :- has(goergeBush,intelligence, high).
T = warm                                      no.
yes
:- has(georgeBush,intelligence, I).
I = very_high;
no.
```

9 Explanations in Expert Systems (5 points)

Give three reasons why it is considered important for expert systems to have an explanation capability.

10 Probability in Mycin? (5 points)

What would be the advantages and disadvantages of using probability theory in a rule-based expert system like MYCIN to represent and reason about uncertain data and knowledge instead of the more ad hoc system actually used in MYCIN.