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

CMSC671 Final Examination
May 17, 1993

Name _____
SS# _____

1 True/False (10 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. ___ Unification is an algorithm for determining the substitutions needed to make two predicate calculus expressions match.
3. ___ If a Prolog term A unifies with term B, and term B unifies with term C, then term A unifies with term C.
4. ___ If a Prolog term A unifies with term B, and term A unifies with term C, then term B unifies with term C.
5. ___ It is possible for Algorithm A^* to expand a non-optimal solution node.
6. ___ Semantics networks lack adequate expressive power for representing natural language meaning because one can not represent non-binary relations, since every arc connects exactly two nodes.
7. ___ The addition of inheritance to a semantic net system will usually *reduce* the size of the knowledge base.
8. ___ MYCIN is a goal-driven expert system.
9. ___ MYCIN can not learn to diagnose new diseases.
10. ___ MYCIN assumes that each of its production rules provides independent evidence for its conclusion(s).

2 Lists in Prolog (10 points)

Assume the predicate *pick(List,Element,Remainder)* is true iff *Element* is a member of the list *List* and *Remainder* is a list equivalent to *List* with (one occurrence) of *Element* removed. Thinking of this procedurally, you could say that *pick/3* picks one element from its first argument and returns it and the rest of the list in the next two arguments. That is:

```
?- pick([a,b,c],X,L).
  X=a, L=[b,c];
  X=b, L=[a,c];
  X=c, L=[a,b];
  no.
```

1. Write a prolog predicate defining pick/3. Define any predicates you need (e.g. member, delete, etc).
2. Show the appropriate mode declaration for your definition.

3 Dominos (15 points)

Write a prolog predicate to play dominos. Represent a domino by a term I:J where I is the number of dots on one half and J is the number on the other. A row is is list of dominos such that, for every pair of adjacent dominos in the row, A:B and C:D, B=C.

1. Write a predicate valid_row(L) which is true iff L is a list representing a valid row of dominos. For example,

```
?- valid_row([1:2,2:5,5:4]).     ?- valid_row([1:2,2:4,5:4]).
yes                              no
?- valid_row([]).
yes
```

2. Write a predicate make_row(+List,-Row) which is true if List is a random list of dominos and Row is a valid domino row which contains each of the dominos in List. (You may find the pick/3 predicate useful in this.) Examples:

```
?- make_row([1:2,3:1,2:4],L).      ?- make_row([2:4,1:2,4:1],L).
L = [3:1,1:2,2:4];                 L = [2:4,4:1,1:2];
no                                 L = [4:1,1:2,2:4];
?- make_row([1:2,3:4],L).          L = [1:2,2:4,4:1];
no                                 no
```

4 Backward vs. Forward Chaining (5 points)

General methods can be given for both backward chaining and forward chaining which are both sound and complete. Give at least three different reasons why one might prefer to use a forward chaining algorithm over a backward chaining one.

5 Iterative Deepening (5 points)

Briefly describe the advantages of the *iterative deepening* search algorithm for trees over (a) depth first search and (b) breadth first search.

6 Justification Based TMS (5 points)

Assuming a forward chaining system with a justification-based truth maintenance system like the one described in class, describe what would happen as the following clauses are added to the knowledge base.

```
(1) p, ~q => r.
(2) p, ~r => q.
(3) p.
```

7 Algorithm A Search (5 points)

Given an algorithm A type search with a function of the standard form $F(n) = G(n) + H(n)$, show what form the $H(n)$ function would have to be to have the algorithm perform:

1. Breadth first search.
2. Depth first search.
3. Best first search.

8 Knowledge Representation (10 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)

Briefly describe 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.

11 Knowledge Representation in KL-ONE (10 points)

Draw a KL-ONE diagram which represents the following sentences:

- A *point* is a primitive concept which is a kind of *thing*.
- A *line segment* is a primitive concept which is a kind of *thing* with exactly two *ends* both of which must be points.
- A *polygon* is a primitive concept which is a kind of *thing* with at least three *sides*, all of which must be *line segments*, and exactly one *area*, which is a real number.
- A *triangle* is a polygon with exactly three sides.