Name:
SSN:

CMSC471/671 -- Artificial Intelligence
Final Exam -- Fall 1998

A philosopher, which is what I am supposed to be, is a sort of intellectual yokel who gapes and stares at what sensible people take for granted, a person who cannot get rid of the feeling that the barest facts of everyday life are unbelievably odd.
-- Alan Watts, "Does It Matter?", 1968.

Please put your name and SSN on this exam and your name on all of the exam booklets you use. Hand in this exam and all of the exam booklets you use. You are free to use any books or notes you want. The points add up to 120 and you have 120 minutes to finish the exam. Good luck.

1	
2	
3	
4	
5	
6	
7	
8	
----	

1) Prolog I [20]

Write recursive prolog predicates height/2 and weight/2, which compute the "height" and "weight" of prolog lists.

height(+List, ?N) is true if List is a prolog list and it's "height" is the integer N.
weight(+List, ?N) is true if List is a prolog list and it's "weight" is the integer N.

where notions of the height and weight of a list are defined as follows:

- the height of a term, which is not a list, 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 term, which is not a list, is 1.
- the weight of the empty list is 0.
- the weight of a list is the sum of the weights of its elements.

2) Prolog II [15]

Consider two possible definitions of the member predicate:

Version one:

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

Version two:

```
member2(X,[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]).

3) True/False [15]

Place a T or an F in the space before each statement to indicate
whether the statement is True or False.

a) Algorithm A* is simply and instance of Algorithm A with an additive evaluation function of the form f = g + h.

b) Algorithm A will perform a breadth-first search if H(n) = G(n).

c) Algorithm A will perform a depth-first search if the heuristic function H(n) = -G(n).

d) Alpha-Beta Search is Minimax search in which alpha > beta.

e) Given a poor plausible move generator (i.e. one that generates all the legal moves properly but does a bad job of ordering them from best to worst), the alpha-beta algorithm can prevent spectacular moves such as (in chess) a queen sacrifice leading to a checkmate.

f) First-order predicate calculus allows quantified variables to refer to objects in the domain of discourse, and not to predicates or functions.

g) Every sound inference rule is complete and every complete inference rule is sound.

h) Two expressions in the propositional calculus are equivalent if they have the same value under all truth value assignments.

i) If X is a variable and S is a propositional calculus sentence, then $\exists X\ S$ is a propositional calculus sentence.

j) A most general unifier is a unifier that is not restricted to first-order predicate calculus.

k) Strips is a sound planner but not a complete planner.

l) One way a partial order planner can fail is to find a complete partial-order plan that satisfies the constraints but for which no linear order can be computed.

m) The ID3 decision tree induction algorithm is guaranteed to find the optimal decision tree consistent with a given training set.

n) One drawback of the ID3 decision tree induction algorithm is that it can not be used if the training data contains values which are drawn from finite sets, i.e. real numbers are not allowed.

4) Backward vs. Forward Chaining [10]

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 to a backward chaining one.

5) Iterative Deepening [10]

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

6) Unification [10]

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.

1.	p(f(Z), Y, g(Y))	p(f(X), c, g(X))
2.	p(f(X, X), a)	p(f(Y, f(Y, a)), a)
3.	p(a, X, c, X)	p(b, c, Z, c)
4.	p(f(Y), Y, g(a, Z), b)	p(f(b), W, g(X, X), W)
5.	p(X, Y)	p(a, b, c)

7) STRIPS operators and the situation calculus [15]

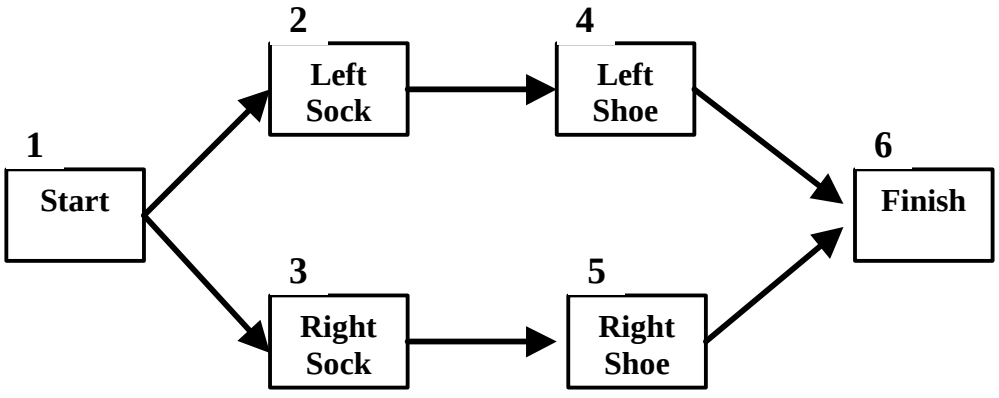
Translate the following situation calculus axioms into *one* STRIPS operator:

$\forall s, x, p$ *Edible* $(x) \wedge$ *Holding* $(p, x, s) \Rightarrow$
Inside $(x, p,$ *Result* $($*Eat* $(p, x), s))$
$\forall s, x, p$ *Edible* $(x) \wedge$ H*olding* $(p, x, s) \Rightarrow$
$\neg$*Holding* $(p, x,$ *Result* $($*Eat* $(p, x), s))$
$\forall s, x, y, p$ *Holding* $(p, y, s) \wedge \neg(x=y) \Rightarrow$
Holding $(p, y,$ *Result* $($*Eat* $(p, x), s))$
$\forall s, x, y, p$ *Inside* $(y, p, s) \Rightarrow$
Inside $(y, p,$ *Result* $($*Eat* $(p, x), s))$
$\forall s, x, y, p \neg$*Inside* $(y, p, s) \neg(x=y) \Rightarrow$
$\neg$*Inside* $(y, p,$ *Result* $($*Eat* $(p, x), s))$

making reasonable assumptions about what the predicates edible, holding and eat mean.

8) Partial-Order Planning I [5]

List all possible totally-ordered step sequence plans that are represented by the following partially-ordered plan containing six steps for putting your shoes and socks on. Indicate a step using its number. You are not allowed to touch or look at your feet in working out the solution to this problem.



9) **Partial-Order Planning II [20]**

Consider the following partially-ordered plan. Solid arcs are causal links; the letter labeling the arc is the literal that is added by the source step and a precondition of the destination step. Dotted arcs are temporal ordering constraint links. Literals at the upper-left corner of a step are the preconditions of the step; literals at the bottom-right corner of a step are the effects of the step.

Positive effects indicate additions and negated effects denote deletions. As usual, for each causal link, there is an implicit temporal constraint link between the same two nodes that is not drawn to keep the figure less cluttered.

(a) [4] List all the *linearizations* (i.e., total orderings) implicit in the above partially ordered plan.

(b) [4] Are all of the preconditions of the Finish step guaranteed to be achieved in the above partial-order plan? Explain briefly why or why not.

(c) [4] Which step(s) *threaten* the causal link connecting step B to step C?

(d) [4] Modify the partial-order plan in the *above figure* to *remove the threat(s)* to the causal link connecting B to C. Do *not* delete anything already in the figure. Mark your addition(s) with "**(d)**".

(e) [4] Modify the partial-order plan in the *above figure* by *adding a new step* to fix a problem in the plan. Add whatever nodes and/or arcs that are necessary. Do *not* delete anything already in the figure. Mark your change(s) with "(e)" on the figure.