FALL 1994
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN – MADISON
PH.D. QUALIFYING EXAMINATION

Artificial Intelligence

Monday, September 19, 1994
3:30 – 7:30 PM
1240 Computer Sciences

GENERAL INSTRUCTIONS:

1. Answer each question in a separate book.
2. Indicate on the cover of *each* book the area of the exam, your code number, and the question answered in that book. On *one* of your books, list the numbers of *all* the questions answered. *Do not write your name on any answer book.*
3. Return all answer books in the folder provided. Additional answer books are available if needed.

SPECIFIC INSTRUCTIONS:

Answer:

- either one (1) of the questions B540-1 or B540-2, *and*
- both (2) questions in the section labeled B760, B761, B762, or B766 corresponding to your chosen focus area, *and*
- any two (2) additional question in the sections B760, B761, B762, and B766, where these two questions need *not* come from the same section, *and*
- both (2) questions in the section labeled A76x that corresponds to your focus area.

Hence, you are to answer a total of *seven* (7) questions.

POLICY ON MISPRINTS AND AMBIGUITIES:

The Exam Committee tries to proofread the exam as carefully as possible. Nevertheless, the exam sometimes contains misprints and ambiguities. If you are convinced that a problem has been stated incorrectly, mention this to the proctor. If necessary, the proctor can contact a representative of the area to resolve problems during the *first hour* of the exam. In any case, you should indicate your interpretation of the problem in your written answer. Your interpretation should be such that the problem is nontrivial.

Answer either one (1) of the questions B540-1 or B540-2.

B540 – Introduction to AI Questions

B540-1. Say we want to use the *situation calculus* for solving a planning problem. Let the predicate $holds(s, f)$ indicate that f is true in situation s. $Result(s1, e) = s2$ means that if event e occurs in situation $s1$, then the new situation $s2$ will result. Assume our problem is represented by the following three axioms in the situation calculus:

i. $holds(0, f)$
ii. $\forall s, holds(s, f) \Rightarrow holds(result(s, a), g)$
iii. $\forall s, [holds(s, g) \wedge holds(s, f)] \Rightarrow holds(result(s, b), h)$

a. Determine s such that $holds(s, g)$.

b. Add the necessary *frame axiom* so that $holds(result(0, a), f)$.

c. Prove $holds(result(result(0, a), b), h)$ using the frame axiom from (b).

B540-2. Suppose we have a *computer program playing a game*, and at each move, the computer decides its move by doing a 3-ply heuristic search. Assume that each player has exactly 4 possible legal moves, and that at the given game state, neither player can win after 3 more plays. Then, applying a MIN-MAX search, the computer will examine $4^3 = 64$ leaf positions. How many leaf positions will the computer examine if it does an alpha-beta search in:

a. the best case?

b. the worst case?

Give an explicit example to illustrate each case.

Answer both (2) of the questions in the section labeled B76x that corresponds to your chosen focus area. Also answer any two (2) additional questions in any of the other sections (these two questions need NOT occur in the same section).

B760 – Machine Learning Basic Questions

B760-1. A common technique in *decision-tree induction* is to prune the tree that is produced during the process of completely separating the training examples. Explain why pruning is done, then briefly outline an algorithm for doing so.

Is pruning guaranteed to increase, or at least not decrease, generalization (i.e., accuracy on examples not in the training set)? Explain you answer.

B760-2. Define the task of *reinforcement learning* (RL) and list one (1) of its major strengths and one (1) of its major weaknesses. Are there any conditions where RL is equivalent to the more traditional problem of learning from a set of teacher-classified examples?

B761 – Deductive Problem Solving & Logic Programming Basic Questions

B761-1. Derive the empty clause from the following, using *binary resolution plus paramodulation.* x, y, z are variables and a, b are constants.

$$x \cdot (y \cdot z) = y$$
$$a \neq b$$
$$x = x$$

B761-2. In many *Prolog* systems, "=" and basic list operations are built in. So, for example, the interactions:

? - f(a,Y) = f(X,b).
X = a , Y = b

and

? - length([p,q,r], X).
X = 3

will take place even with no program at all consulted. But, suppose you were working with a very primitive Prolog, which contained the usual pure logic, plus some elementary arithmetic, but did not know about list notation or "=". How would you program "="? What would you use as a substitute for lists, and how would you then define "length"?

B762 – Knowledge Representation & Expert Systems Basic Questions

B762-1. Theoretical logicians have long been concerned with the *soundness and completeness of axiomatic systems.* As those systems have been used as a basis for the implementation of automatic deductive inference procedures on modern computers and for the design of the deductive components of AI systems (e.g., the inference engines of expert systems), soundness and completeness have often been investigated as important properties of these practical systems.

a. Precisely what do the terms "sound" and "complete" mean when applied to theoretical axiomatic systems?

b. Precisely what do they mean when applied to practical computer-based deduction systems?

c. What are the dangers of using an unsound computer-based deduction system? What about an incomplete one? Are these dangers such as to present practical difficulties, or are they only of theoretical interest? Illustrate your answers with specific examples.

B762-2. *Expert systems* that use knowledge bases consisting of large numbers of precisely formalized expressions are hard to implement because the task of expression formalization is so labor intensive. How do expert systems implemented as case-based reasoners approach this problem? Does the approach work? To what extent does it limit the kinds of tasks to which the resulting systems can be applied, and how does it handle the knowledge-base validation problem? How might the use of neural networks be an alternative?

B766 – Computer Vision Basic Questions

B766-1. The major difficulty in computational *stereo* is the *correspondence problem.*

a. If we assume that a single continuous surface is being looked at, what geometric constraint must be satisfied by the point correspondences?

b. Describe two other types of constraints that have been used to help solve the correspondence problem, one being a geometric constraint imposed by the imaging system, and the other a geometric constraint arising from the assumptions about the surfaces in the scene.

c. Why doesn't the method called *photometric stereo* have a correspondence problem? Explain why photometric stereo cannot recover the 3D coordinates of points in the scene and, instead, what it does recover.

NOTE: This part of this question mistakenly refers to material that was not on the basic reading list and therefore was graded assuming the student had no knowledge of photometric stereo.

B766-2.

NOTE: This question mistakenly refers to material that was not on the basic reading list and therefore was graded assuming the student had no knowledge of active vision.

a. Explain what the *active vision* paradigm is, and what its purported main advantages are.

b. Select a task, e.g., recovering the 3D coordinates of a set of feature points, and then describe an active approach to solving this task. Be sure to clearly indicate what parts of the approach are the key ones in making it an "active" method.

Answer both (2) of the questions in the section labeled A76x that corresponds to your chosen focus area.

A760 – Machine Learning Advanced Questions

A760-1. Imagine you are assigned the task of creating a *software agent that learns.* Specifically, assume that this agent's task is to wander the World-Wide Web, looking for papers that are likely to be of interest to its "owner." Propose a design for this system, building upon existing machine-learning paradigms and algorithms as much as possible.

What do you feel is the most significant open issue in machine learning that this task faces? Justify your choice.

A760-2. An appealing idea is that of *life-long learning.* Namely, a learner should be able to use previously learned concepts when facing the task of learning a new one. Describe how one could use ideas from theory refinement to create a life-long learner. Refer to published algorithms where appropriate.

Discuss one (1) major strength and one (1) major weakness of applying the theory-refinement view to the problem of life-long learning.

A761 – Deductive Problem Solving & Logic Programming Advanced Questions

A761-1. In the *Boyer-Moore prover*, a simple way to represent sets is by lists, so that every list represents the set of its members. Then, one can define `subsetp` so that in an `r-loop`, you would get, for example:

```
*(subsetp '(3 1 2 1 2) '(4 1 2 3))
T
* (subsetp '(1 2 3) '(1 2))
F
*(subsetp '((1 2) 1 2) '(1 2))
F
```

How exactly would you define `subsetp`? How would you get the prover to accept the following lemma?

```
(prove-lemma non-subset ()
     (not (subsetp (cons lst lst) lst)))
```

A761-2. Explain positive and negative *hyper-resolution.* Sketch a proof of the completeness theorem for these. Give an example to show that factoring is necessary for completeness. How is hyper-resolution related to Prolog? Why can Prolog get by without doing factoring?