FALL 1993
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN—MADISON
PH. D. QUALIFYING EXAMINATION

Artificial Intelligence
Depth Examination

Monday September 20, 1993
3:00 – 7:00 PM
2365 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

- any two (2) of the questions B540-1, B540-2, and B540-3, *and*
- both questions in any *one* of the sections labeled B760, B761, B762, or B766, *and*
- any two (2) additional questions in the sections B760, B761, B762, and B766, where these two questions need *not* come from the section, *and*
- both questions in the section labeled D76x that corresponds to the same area in which you answered both questions in section B76x.

Hence, you are to answer a total of *eight* (8) questions.

POLICY ON MISPRINTS AND AMBIGUITIES:

The Exam Committee tires to proofread the exam as carefully as possible. Nevertheless, the exam sometimes contains misprints and ambiguities. If you are convinced 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 any two (2) of the three questions labeled B540-x.

B540 – Introduction to AI Questions

B540-1. Explain **alpha-beta search** and give an explicit example. Try to keep your example simple, but it should demonstrate how cutoffs work to prune the tree three levels down as well as one level down.

B540-2. Explain syntactic ambiguity as occurs in **natural language parsing**, using as a specific example the sentence, "I killed the mosquito with a fly swatter." In particular, describe a context-free grammar that produces two parses for this sentence. Then, give another example of syntactic ambiguity that does not involve prepositional phrases.

B540-3. Define the **frame problem** and explain its significance to the AI task of planning. Describe how the frame problem is addressed by (a) STRIPS and (b) situation calculus. Illustrate your answer by showing how a concrete, real-world action could be formalized in each of these two formalisms.

Answer BOTH of the questions in any ONE of the sections labeled B76x. (Note: You must then answer BOTH of the questions in the corresponding section D76x.) Also answer any two additional questions in any of the other sections (these two questions need NOT occur in the same section).

B760 – Machine Learning Breadth Questions

B760-1. Suppose a machine learning researcher is given a new stock-market dataset and is asked to produce an accurate stock-market advisor. As a first step, this person divides the data into disjoint training and testing sets. Next, the researcher uses backpropagation to independently train 25 **neural networks**, each with one layer of hidden units. The same training examples are always used, but in each run the initial weights, number of hidden units, learning rate, etc. are randomly chosen. Each network is trained until the accuracy on the training set reaches 95%.

Following training, this researcher measures the correctness of each of the 25 networks on the test set. The researcher discards all but the network with the best test-set accuracy and then reports the remaining network's test-set accuracy as the prediction of its future performance.

Critique this *experimentation methodology*. Describe and justify any modifications you would make.

B760-2. State, define, and motivate the two major types of **hypothesis-space biases** used in inductive learning. Give the bias of (a) ID3 and (b) backpropagation. Justify your answers.

B761 – Deductive Problem Solving & Logic Programming Breadth Questions

B761-1. Most **Prolog** systems do not do an *occur check*. Explain what this means, and give an example where it can lead to nonsense results. Also explain why this is usually not a problem in practical Prolog AI programming.

B761-2. If **R** is a binary relation that is irreflexive (*x* **R** *x* is always false) and transitive, then it can never have a 3-element cycle (*x* **R** *y* and *y* **R** *z* and *z* **R** *x*). Express this inference in Resolution format, and then **derive the empty clause** using paramodulation plus binary resolution.

B762 – Knowledge Representation & Expert Systems Breadth Questions

B762-1. One popular architecture for expert systems emphasizes an inference engine and a knowledge base as the two primary components.

(a) Portray the structure and function of each in the case where the knowledge of an expert system is represented as **predicate calculus** expressions.

(b) Portray the structure and function of each in the case where the knowledge of an expert system is represented as **frames** (or objects).

B762-2. The GOFAI (good old-fashioned AI) approach is perhaps best summed up in Newell and Simon's **physical symbol system hypothesis**.

(a) Describe the hypothesis.

(b) Explain how the connectionist approach either fits within the GOFAI approach, changes part of it, or changes all of it.

B766 – Computer Vision Breadth Questions

B766-1. It is often recommended that **edge detectors** be applied at multiple scales to an image and these outputs then be "integrated" in some way.

(a) Give an example of a single image where detecting edges at two different scales is necessary to detect different salient structural features.

(b) Given a single image, how are multiple resolution images usually produced? Be as specific as possible. Is it appropriate or inappropriate for this step to use isotropic (i.e., rotationally-invariant) operators? Explain briefly.

(c) Briefly describe a method for combining the results of edge detection at multiple scales into a single representation.

B766-2. Labeling a line drawing of a polyhedral scene can be formulated as a "consistent labeling" problem, assigning a label to each edge subject to constraints at the vertices.

(a) Describe how the **Waltz filtering** algorithm works by showing the initial configuration and several snapshots of the algorithm as it is applied to a simple scene of your choosing.

(b) Characterize how the Waltz filtering algorithm differs from (i) hypothesize-and-test, and (ii) backtracking tree search procedures for solving this problem.

Answer BOTH of the questions in the section labeled D76x that corresponds to the same area in which you answered BOTH questions in section B76x.

D760 – Machine Learning Depth Questions

D760-1. Define and motivate the task addressed by Watkin's **Q-learning algorithm**. Like Q-learning, SOAR and PRODIGY are approaches by which an intelligent agent learns how to better choose appropriate actions. What do you see as the most important strength of Q-learning over SOAR (or PRODIGY)? What would you claim is the major advantage of SOAR (or PRODIGY) over Q-learning? Finally, propose a method by which prior, domain-specific knowledge could be used to improve some weakness of Q-learning.

D760-2. Describe and motivate the **minimal-description-length (MDL) principle**. Sketch how it has been used in (a) decision-tree induction and (b) neural-network training. How would you use MDL in a learner whose input is a set of training examples plus an approximately-correct domain theory?

D761 – Deductive Problem Solving & Logic Programming Depth Questions

D761-1. The **Clark completion** is a first step towards describing the semantics of Prolog programs with negation. Explain why the completion correctly explains the semantics of the program

```
p :-  not(r).
q.
```

but not of the program

```
p :-  not(p).
q.
```

Then, describe some approach that correctly handles the semantics of the second program.

D761-2. With the **Boyer-Moore prover** you can define the factorial function in the standard way by

```
(defn fact (n) (if (zerop n) 1 (times n (fact (sub1 n)))))
```

It will then succeed in proving some elementary facts. For example, if you tell it

```
(prove-lemma increasing () (implies
    (and (lessp 0 x) (lessp x y) )
    (lessp (fact x) (fact y))))
```

it finds the correct induction without any help. However, it fails if you give it

```
(prove-lemma one-one () (implies
    (and (lessp 0 x) (lessp 0 y) (equal (fact x) (fact y)))
    (equal  x  y)))
```

Why? What would you have to give it for hints and/or auxiliary lemmas to get this to work?

D762 – Knowledge Representation & Expert Systems Depth Questions

D762-1. (a) Characterize the distinction between knowledge represented **procedurally** and knowledge represented **declaratively**. Give examples of each. Briefly list the merits claimed for each.

(b) Explain why the distinction is or is not the same as the distinction between programs and data.

(c) Consider the representation of knowledge as Prolog statements. Explain why you consider this procedural representation, declarative representation, or both.

(d) Consider the representation of knowledge as production rules. Explain why you consider this procedural representation, declarative representation, or both.

D762-2. An advantage held out for building computer systems as **expert systems** is that they can be built step by step, that they are potentially useful after a few initial steps, and that the building steps can continue indefinitely (that is, the system may never be fully completed). Such a method of system construction is called *incremental prototyping*.

(a) Explain how the method works.

(b) How does it relate to the method of "fast prototyping" used in program construction?

(c) What are the implications of the method for the problem of system validation?

D766 – Computer Vision Depth Questions

D766-1. Many stereo and tracking algorithms require matching points detected in multiple images. This requires an operator that detects the same physical point in the images so that the correspondence problem can then be addressed.

(a) Characterize at least two necessary properties of a good **interest operator** for detecting points in images when they are to be used by stereo or tracking algorithms.

(b) Define a candidate operator that satisfies the properties in (a). Briefly explain why this operator is an appropriate one to use for this purpose.

(c) Where might this type of interest operator *fail* to work correctly? Give an example and an explanation. Also briefly indicate what might be done to detect and then eliminate bad points.

D766-2. (a) What the **active vision** approach in computer vision? Give an example of a problem that has used this approach.

(b) How might an active vision approach be used to find the *occluding contours* of a set of smooth objects in a scene, distinguishing this set of edge points from edge points produced by surface markings, texture, shadows, and surface-orientation discontinuities? Briefly outline an algorithm, explaining in particular the parts that utilize the "active" capabilities you've chosen.