

# CPE/CSC 480 ARTIFICIAL INTELLIGENCE

## FINAL EXAM

### FALL 2005

---

**PROF. FRANZ J. KURFESS**

CAL POLY, COMPUTER SCIENCE DEPARTMENT

---

This is the Fall 2005 final exam for the CPE/CSC 480 class. It is a take-home exam, and you may use textbooks, course notes, or other material, but you must formulate the text for your answers yourself. You are not allowed to discuss the questions and answers of the exam with other students or anybody else. The use of calculators or computers is allowed for numerical calculations, but not for the execution of algorithms or programs to compute solutions for exam questions.

If you need clarifications about questions, you can contact me via email ([fkurfess@calpoly.edu](mailto:fkurfess@calpoly.edu)) or during my office hours on Monday Dec. 5, 10 am - 12 pm, and Tuesday, Dec. 6, 2-5 pm. The deadline for the exam is Tuesday, December 6, 2005, at 4:00 pm. If you submit the exam before noon on Monday, you will receive an early submission credit of 10% of your score in the exam.

You must submit a printed and signed copy of the exam, which you can either leave in the drop box in front of the CSC department office (it is emptied at 4:00 pm), or give it to me on Tuesday before 4:00 pm in my office.

**Student Name:**

**Signature:**

**Date:**

---

**PART 1: MULTIPLE CHOICE QUESTIONS**

---

Mark what you think is the best answer for the questions below. I tried to formulate the questions and answers such that there is only one suitable answer. Each question is worth 3 points.

- a) What is the *contingency problem* in the context of game-playing programs?
- a degree of uncertainty, introduced by the presence of an opponent or by chance elements
  - the outcome of a move may not be visible due to search limitations
  - the need for arbitration (e.g. by a referee) in some types of games
  - the elimination of branches that will never be explored
- b) What is the *horizon problem* in the context of game-playing programs?
- a degree of uncertainty, introduced by the presence of an opponent or by chance elements
  - the outcome of a move may not be visible due to search limitations
  - the need for arbitration (e.g. by a referee) in some types of games
  - the elimination of branches that will never be explored
- c) Why is it important in the *minimax algorithm* to generate the values of the evaluation function “bottom-up” rather than calculating them for each node when the node is visited first (“top-down”)?
- Minimax also works top-down, but the bottom-up method is more efficient.
  - This is the foundation for considering the possible countermoves by the opponent.
  - Because the value of the utility function can only be computed for leaf nodes.
  - Otherwise Minimax would have to be called Maximin.
- d) Which statement describes the *syntax* of a formal language for knowledge representation?
- It describes how a particular sentence relates to the facts in the world.
  - It allows the generation of new sentences that follow from a set of given sentences.
  - It specifies the admissible configurations of sentences in that language.
  - It makes sure that only truth-preserving sentences are admitted in the language.
- e) A logical sentence is called *satisfiable* if and only if
- it is constructed according to the syntactical specification of the language
  - it is true under all possible interpretations in all possible worlds
  - it can be used by an inference procedure to construct a proof
  - there exists at least one interpretation for which the sentence is true.

- f) Which of the following statements characterizes *predicate logic* (in contrast to propositional logic)?
- The world is described through sentences consisting of constants, symbols, connectives, and parentheses.
  - A simple logic in which truth tables are the only way of proving sentences.
  - The world is described through sentences specifying individual objects with properties, and relations between the objects.
  - A logic that relies on resolution as the only sound inference rule.
- g) Why are ontologies often organized around taxonomic hierarchies?
- It enables the use of inheritance for a concise, but flexible description of the concepts in the ontology.
  - It is used to represent elementary, undisputed facts and rules of a domain
  - It is a method used to formalize the semantics of sentences in a domain.
  - It greatly improves the efficiency of inference procedures by steering the selection of applicable inference rules.
- h) Why are decision trees relatively easy to learn?
- They are usually relatively small.
  - Decision trees represent Boolean functions, with a binary yes/no decision for queries.
  - Even for complicated trees, a small training set is sufficient for learning.
  - The hypothesis space can be traversed very effectively with simple decision tree learning procedures.
- i) What is an important limitation of the learning capability of perceptron networks?
- The training examples need to be presented in a specific order, otherwise the network may get confused.
  - They can only learn Boolean functions.
  - They are incapable of learning non-linearly separable functions.
  - They are very sensitive to noise.
- j) Which of the following describes the process of *classifying* a test example using a feed-forward neural network that uses back-propagation learning?
- Activation values are propagated from the input nodes through the hidden layers to the output nodes.
  - Activation values are propagated from the output nodes through the hidden layers to the input nodes.
  - Weights on the arcs are modified based on values propagated from input nodes to output nodes.
  - Weights on the arcs are modified based on values propagated from output nodes to input nodes.

---

**PART 2: SHORT QUESTIONS**

---

In this part of the exam, you should answer the questions in about one or two paragraphs. Please note that the number of points differs for the questions.

1. Explain the general concept of *probably approximately correct (PAC) learning*, and discuss its possible use for a vacuum bot agent. The main emphasis for the agent is to learn from its past experiences about the likelihood of finding dirt or encountering obstacles in various parts of a room.

10 points

- a) PAC Learning Explanation

- 
- 
- 
- 
- 

- b) PAC Learning for a Vacuum Bot Agent

- 
- 
- 
- 
-

## 2. Logic and Neural Networks

- a) The **NAND** function of  $n$  binary inputs,  $x_1, x_2, \dots, x_n$  is defined as  $\neg(x_1 \wedge x_2 \wedge \dots \wedge x_n)$ . Let True be represented by 1, False by 0, and  $n$  is a known, fixed constant. Can this function be represented by a Perceptron? If so, describe the structure of a Perceptron that does it; if not, argue why not.

5 points

•

•

•

•

•

- b) Another function that we could call the **SAME** function of two inputs,  $x_1, x_2$ , is defined to be 1 if the inputs are both the same (i.e., both 0 or both 1), and 0 otherwise. Can this function be represented by a Perceptron? If so, construct a Perceptron that does it; if not, argue why not.

5 points

•

•

•

•

•

## 3. Capabilities and Limitations of Neural Networks

- a) People often say that *local minima* can cause problems in feed-forward neural networks with back-propagation as training method. What are the function and its arguments that these local minima refer to (e.g., the input function and the respective weights, or the activation function)? Also explain why this is a problem in terms of the search method used.

5 points

•

•

•

•

•

- b) Assume that we have a feed-forward neural network with one hidden layer. Typically, such networks use a sigmoid function as activation function associated with every hidden node and every output node. Your team mate suggests to use a linear function instead in order to reduce the training time for the network. That is, the activation function is  $g(x) = ax + b$ . You can assume the same linear function is used at every node (i.e.,  $a$  and  $b$  are constants). Is this a good idea? What is the impact on the expressiveness of the neural network with such a linear activation function?

5 points

•

•

•

•

•

---

## PART 3: GRAPH-BASED WUMPUS WORLD

---

In this part of the exam, you need to explore and evaluate the capabilities of an agent in a Wumpus World that has as its underlying topology a graph instead of a grid-based maze. Due to this topology, the navigation task for the agent is also somewhat different: it moves in a different way, and the assumptions for reasoning about properties of nodes may have to be changed.

The basic setup is similar: The agent has to find the gold, pick it up, and return to the starting point. It has as percepts {**Breeze**, **Smell**, **Glitter**, **Scream**}, and as actions {**Move-Next**, **Shoot**, **Grab**, **Climb**}. The **Move-Next** action transports the agent from the current node to another node. If the agent maintains a queue of nodes to be explored, it moves to the first node in the queue (which does not necessarily have to be adjacent to the current node), otherwise a node is selected randomly from the adjacent ones.

In this part, you are provided with an agent that has the following characteristics:

- The agent works with two queues for nodes to be explored: A *safe queue* for nodes that are safe to enter, and a *risky queue* containing nodes that may be dangerous for the agent (e.g. a pit, or the wumpus may be on the node).
- The agent prefers to move to the next node in its safe queue; if the safe queue is empty, it will select the next node from the risky queue. It explores adjacent nodes in a clockwise fashion, starting from the “noon” position, and follows a breadth-first search method.
- The agent keeps a knowledge base with useful statements for later use.

And here are a few assumptions to clarify ambiguous situation; if necessary, you can make additional assumptions, but make sure to state them.

- Once the wumpus is dead, its smell still can be perceived.
- The wumpus can not be on the same node as a pit.

The detailed tasks you need to solve are described below, with the respective number of points. The total number of points for this task is 40.

### Additional Assumptions

If necessary, state additional assumptions that you are making here.

- 
- 
- 
- 
-

a) In the table below, describe the following important aspects of the agent outlined above. [8 points]

- *Behavior*  
What are the observable activities of the agent for an outside observer? Could that observer judge the “intelligence” of the agent without knowing anything about how the agent works internally?
- *Internal storage and retrieval of information (memory)*  
How does the agent keep track of previous percepts and possibly other important information? Does it construct an internal representation of the environment?
- *Reasoning and decision making*  
How does the agent make decisions about which action to take? Is it possible for the agent to plan ahead for several steps?
- *Learning*  
Is the agent capable of learning from observations or experiences? If not, would it be possible to incorporate a learning component into the agent? What advantages and problems might result from that?

**AGENT ASPECTS**

|                  |            |
|------------------|------------|
| <i>Behavior</i>  | •<br><br>• |
| <i>Memory</i>    | •<br><br>• |
| <i>Reasoning</i> | •<br><br>• |
| <i>Learning</i>  | •<br><br>• |



b) What type of agent is described above? Check the appropriate box, and explain your answer!

- simple reflex agent
  - model-based agent
  - goal-based agent
  - utility-based agent
  - knowledge-based agent
  - learning agent
- Explanation:

[4 points]

c) What is the simplest type of agent that can successfully solve the task of finding the gold, and returning to the starting point? The characteristics of this type may be different from the ones listed above. Explain your answer!

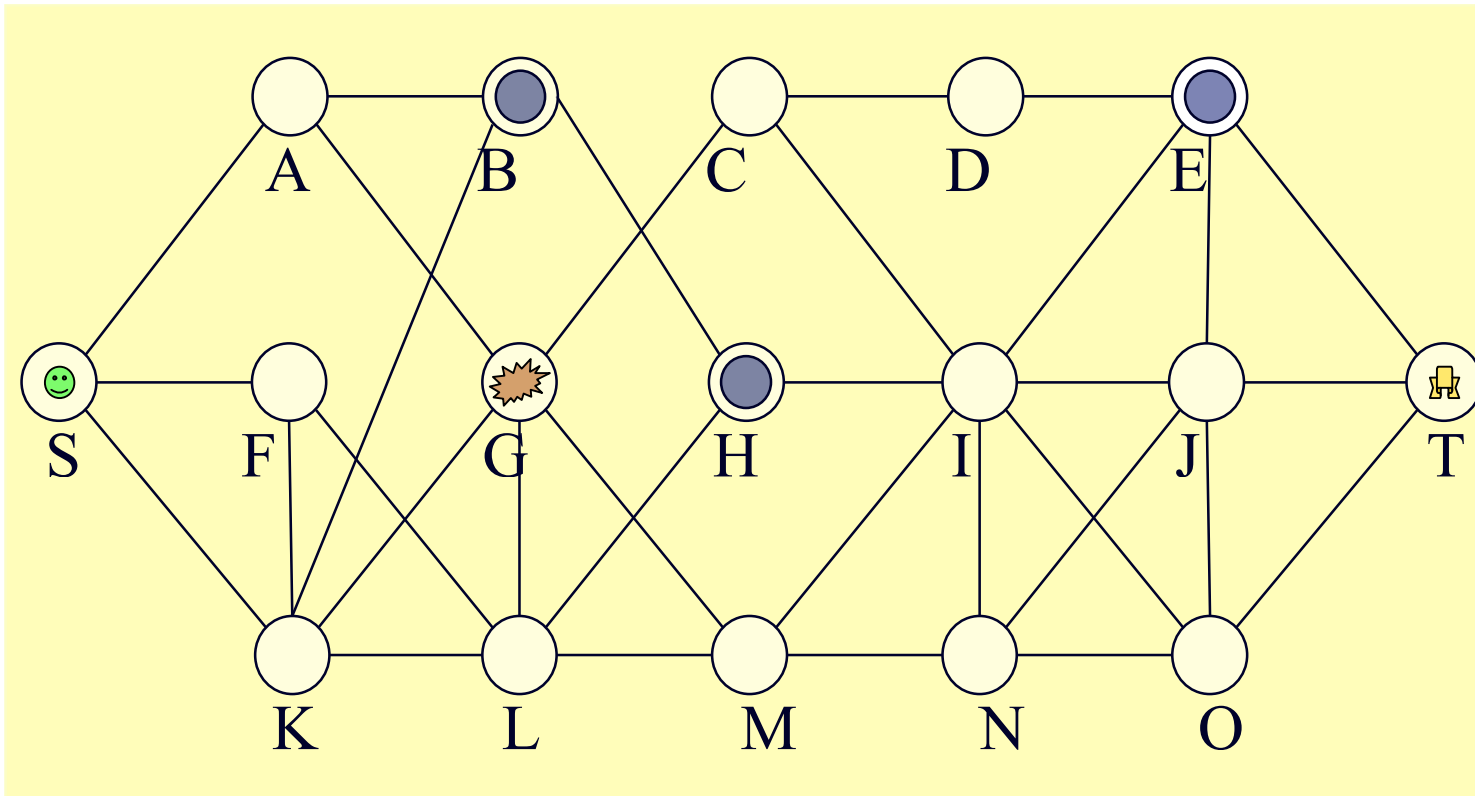
- simple reflex agent
  - model-based agent
  - goal-based agent
  - utility-based agent
  - knowledge-based agent
  - learning agent
- Explanation:

[4 points]

d) Indicate in the diagram a possible path for the agent from the starting point to the goal, and describe the important steps taken and critical decisions made by the agent. You don't need to perform the actions necessary for the agent to return to the starting point and climb out, but you should maintain information about a return path for the agent. Use the diagram and the table below. The agent initially is at node S, and the gold at node T. The wumpus is at node G, and nodes B, E, and H are pits. Please use the following notation:

- mark the nodes where a breeze or a smell can be perceived with appropriate colors (e.g. highlighters)
- mark the nodes known to be safe to the agent with “ok”
- mark the nodes visited by the agent by crossing them out
- mark the nodes that the agent thinks could be pits with “?Pit”
- mark the nodes where the agent thinks the wumpus might be with “?Wumpus”
- if the agent ascertains the presence of a pit or the wumpus, cross out the question mark, and write down an exclamation mark (e.g. “!Pit”)
- if the agent ascertains the absence of a pit or the wumpus, cross out the question mark, and write down a negation sign (e.g. “¬Pit”)

If you want to, you can replicate the graph below to indicate the progress of the agent. [12 points]



- e) Use the same notation to record the activities of the agent in the table, listing the steps (e.g. movements, important reasoning steps) of the agent. You can assume that knowledge from previous steps is accessible to the agent, and that “newer” knowledge replaces “older” knowledge. As an alternative, you can replicate the complete knowledge base of the agent in each step; this is less confusing, but requires more space. The size of the table does not necessarily correspond to the number of steps your agent may need.

[12 points]

| <i>Step</i> | <i>Node</i> | <i>Percept</i> | <i>Visited</i> | <i>Return Path</i> | <i>Safe Queue</i> | <i>Risky Queue</i> | <i>Knowledge</i>                    | <i>Action</i> |
|-------------|-------------|----------------|----------------|--------------------|-------------------|--------------------|-------------------------------------|---------------|
| 0           | S           | None           | S              | S                  | [A, F, K]         | []                 | ¬Pit (A, F, K)<br>¬Wumpus (A, F, K) | Move-Next     |
| 1           |             |                |                |                    |                   |                    |                                     |               |
| 2           |             |                |                |                    |                   |                    |                                     |               |
| 3           |             |                |                |                    |                   |                    |                                     |               |
| 4           |             |                |                |                    |                   |                    |                                     |               |
| 5           |             |                |                |                    |                   |                    |                                     |               |
| 6           |             |                |                |                    |                   |                    |                                     |               |
| 7           |             |                |                |                    |                   |                    |                                     |               |
| 8           |             |                |                |                    |                   |                    |                                     |               |
| 9           |             |                |                |                    |                   |                    |                                     |               |
| 10          |             |                |                |                    |                   |                    |                                     |               |
| 11          |             |                |                |                    |                   |                    |                                     |               |

|    |
|----|
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 29 |
| 21 |
| 22 |
| 23 |
| 24 |

**Total Points:**