CS 5541: Artificial Intelligence (4)

Catalog Description:
Principles and programming methods of artificial intelligence. Knowledge representation methods, state space search strategies, and use of logic for problem solving. Applications chosen from among expert systems, planning, natural language understanding, uncertainty reasoning, machine learning, and robotics. Lectures and labs will utilize suitable high-level languages (e.g., Python or Lisp).

Textbook: Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 2nd Ed. Prentice-Hall, 2003.

Course Goals:
This course introduces the field of artificial intelligence (AI). Students learn about AI methods in machine problem solving (a) through search and reasoning, (b) through learning algorithms (e.g., neural networks), and (c) through algorithms dependent on particular instructors (e.g., robotics related techniques). Students also use a new programming language *Python* and/or *Lisp*, and learn to think and write about the issues which dominate the field.

Prerequisites by Course & Topic
CS 3512: Computer Science Theory – large scale software issues including management, design, modularity, and data abstraction (via 3512 prerequisite of 2511), predicates and quantifiers, sets, relations, proof techniques, recursion and induction, combinatorial counting techniques, graph theory

Major Topics Covered in the Course
- Python and/or Lisp function and data definitions
- Knowledge representation, empiricist and rationalist approaches, and agent concepts
- State space search, informed and uninformed search techniques
- Iterative improvement techniques (e.g., genetic algorithms)
- Propositional logic and inference in logic
- Learning algorithms: decision trees, neural networks
- Case studies (e.g., Deep blue, Speech translation).

Class/Laboratory Schedule: Lecture: 3 hours per week, Laboratory: 1

Course Outcomes
1. Familiarity with philosophical and representational issues directly relevant to AI.
   a. Understanding the agent-based approach to AI.
   b. Understanding empiricist or data-oriented approaches (e.g., Brooks).
   c. Understanding rationalist or reasoning-based approaches (e.g., Newell & Simon).
   d. Understanding the physical symbol system hypothesis.
   e. Understanding the physical grounding hypothesis.
   f. Understanding distributed versus local representation.
2. Proficiency in search techniques for finding solutions to formally specified goals.
   a. Understand a range of search techniques given problem start states and goal states (e.g., breadth first, depth first search).
   b. Analyze traditional AI micro-world problems, such as the water-jugs problem and for these problems, formulate the operators and specify a start state and goal state needed to solve the problem via search.
3. Proficiency with analysis and methods in propositional logic.
   a. Analyze natural language statements into propositional logic terms.
   b. Apply inference methods in propositional logic to prove statements.
4. Proficiency in the Python/Lisp programming language.
   a. Ability to write recursive programs in Python/Lisp based on an algorithmic specification.
   b. Designing program solutions to problems assigned in class.
5. Ability to analyze problems using learning and iterative improvement algorithms.

a. Utilize matrix algebra to determine the outputs of a neural network given a network structure and particular activation function.
c. Construct a decision tree on the basis of information theory techniques.
b. Specify a fitness function for a genetic algorithm given a statement of a goal to be achieved.

Relationship to Program Outcomes

Students who have completed computer science theory may take this course. CS 5541 contributes to meeting the following program outcomes:

1. *Students understand the mathematics and statistics that underlie scientific applications.*
   Students learn mathematical techniques related to neural networks and information theory in decision trees. Course outcome 5 maps to this program outcome.
2. *Students can design, develop, and analyze significant software systems.*
   Student programs must demonstrate good design and well-structured implementation. Course outcomes 1-5 map to this program outcome.
3. *Students understand the fundamentals of computer organization and architecture, data structures and related algorithms, and programming languages*
   Students increase their proficiency in algorithms and data structures by learning the intricacies of neural networks and genetic algorithms. They learn how to use search techniques to find solutions to formally specified goals. They also learn Python and/or Lisp – an additional programming language. Course outcomes 1-5 map to this program outcome.
4. *Students can apply computer science principles and practices to a variety of problems.*
   Students apply knowledge gained in this course to AI problem solving methods, artificial neural networks, robotics and machine vision. Course outcomes 1-5 map to this program outcome.

Assessment Plan for Course:
This course is assessed every third year by the instructor and a course assessment document covering all of the course outcomes and their effect on the program outcomes is prepared.

Estimate CSAB Category Content

	CORE	ADVANCED
Data Structures		1
Algorithms		1
Software Design		1

	CORE	ADVANCED
Computer Organization and Architecture		
Concept of Programming Languages		1

Coordinator/Prepared by: C. Prince