Final Study Guide

Final Time and Place:

- Thursday, May 3, 7-10pm
- Neville 003

Format:

You can expect the following types of questions: true/false, short answer, and smaller versions of homework problems. Although you will have three hours to complete the final, it will only be about twice as long as the midterm. It will be closed book and closed notes. However, you may bring one 8 ½ x 11" "cheat sheet" with handwritten notes on both sides. All PDAs, portable audio players (e.g., iPods) and cell phones must be put away for the duration of the test, but you may use a simple, non-programmable calculator.

Coverage:

The test will be comprehensive, however approximately two-third of the questions will be on subjects covered since the midterm. In general, anything from the assigned reading or lecture could be on the test. In order to help you focus, I have provided a **partial list** of topics that you should know below. In some cases, I have explicitly listed topics that you do not need to know. In addition, you do not need to memorize the pseudo-code for any algorithm, but you should be able to apply the principles of the major algorithms to a problem as we have done in class and on the homework.

- Ch. 1 – Introduction
  - rationality
  - definitions of "artificial intelligence"
  - The Turing Test
  - **you do not need to know:**
    - dates and history
- Ch. 2 - Agents
  - PEAS descriptions
    - performance measure, environment, actuators, sensors
  - properties of task environments
    - fully observable vs. partially observable, deterministic vs. stochastic vs, strategic, episodic vs. sequential, static vs. dynamic, discrete vs. continuous, single agent vs. multiagent
  - agent architectures
    - simple reflex agents, goal-based agents, utility-based agents, learning agents
- Ch. 3 – Search
  - problem description
    - initial state, actions (successor function), goal test, path cost, step cost
  - tree search
    - expanding nodes, fringe
    - branching factor

- uninformed search strategies
  - breadth-first, depth-first, uniform cost
  - similarities and differences / benefits and tradeoffs between strategies
  - evaluation criteria
    - completeness, optimality, time complexity, space complexity
- **you do not need to know:**
  - depth-limited, iterative deepening or bidirectional search
  - the exact *O*() for any strategy's time/space complexity *(but you should know relative complexity)*
  - sensorless planning

- Ch. 4 – Informed Search (Sect. 4.1-4.2)
  - best first search
  - evaluation function, heuristics
  - strategies
    - greedy search, A*
    - admissible heuristics
    - similarities and differences / benefits and tradeoffs between strategies
  - **you do not need to know:**
    - details of proof that A* is optimal if h(n) is admissible
    - memory bounded heuristic search
    - learning heuristics from experience
- Ch. 6 - Game playing (Sect. 6.1-6.2, 6.4, 6.6-6.8)
  - two-player zero-sum game
  - problem description
    - initial state, actions (successor function), terminal test, utility function
  - minimax algorithm
  - optimal decision vs. imperfect real-time decisions
  - evaluation function, cutoff-test
  - **you do not need to know:**
    - alpha-beta pruning
- Ch. 7 – Logical Agents (Sect. 7.1-7.4)
  - knowledge-based agents
    - TELL, ASK
  - propositional logic
    - syntax and semantics
  - entailment, models, truth tables
  - valid, satisfiable, unsatisfiable
  - inference algorithms
    - criteria: sound, complete
  - model checking
  - **you do not need to know:**
    - details of the Wumpus world

- Ch. 8 – First-Order Logic
  - syntax and semantics
    - be able to translate English sentences into logic sentences
  - quantification
    - existential, universal
  - domain, model, interpretation
- Ch. 9 – Inference in First-Order Logic (Sect. 9.1-9.2, 9-4)
  - substitution, unification
    - most general unifier
  - backward-chaining
    - pros / cons
  - negation as failure
  - **you do not need to know:**
    - inference rules, skolemization
    - constraint logic programming
- "Intro to Prolog Programming" Reading, Ch. 1
  - syntax
    - be able to write rules and facts in Prolog
    - translating to FOL and vice versa
  - backward-chaining, depth-first search
    - be able to find the answers to a goal given a simple Prolog program
  - closed world assumption
- Ch 10 – Knowledge Representation (Sect. 10.1-10.2, 10.5-10.6)
  - categories
    - unary predicate vs. object representation
  - semantic networks
    - inheritance
    - compared to FOL
  - **you do not need to know:**
    - description logic
    - Semantic Web
    - OWL
- Ch 11 – Planning (Sect. 11.1-11.3)
  - problem description
    - initial state, goal state, actions
  - The STRIPS language
    - preconditions and effects
  - forward state-space search
    - applicable actions, result states
  - backward state-space search
    - relevant and consistent actions, predecessor states
  - partial-order planning
    - least-commitment
    - causal links
    - propositional example, resolve conflicts
    - linearizations

    - **you do not need to know:**
        - ADL
        - the actions for any specific planning problem given in the book
- Ch. 12 – Planning and Acting in the Real World (Sect. 12.3-12.6)
    - bounded / unbounded indeterminacy
    - conditional planning
        - disjunctive effects, conditional effects
        - AND-OR graphs
        - knowledge propositions
        - automatic vs. active sensing
    - continuous planning
    - when is each form of planning applicable?
    - **you do not need to know:**
        - details of execution monitoring and replanning
- Ch. 13 - Uncertainty
    - Boolean, discrete and continuous random variables
    - prior probability and conditional probability
    - full joint distribution, atomic events
        - calculate probability of an event from the full joint
    - independent variables
    - conditionally independence
- Ch. 14 - Bayesian Networks (Sect. 14.1-14.2, 14.4)
    - understand network structure
    - compute probability of an atomic event
    - **you do not need to know:**
        - variable elimination algorithm
        - clustering algorithms
- Ch. 16 - Making Simple Decisions (Sect. 16.1 – 16.4)
    - utility function
    - maximum expected utility
    - **you do not need to know:**
        - the axioms of utility theory
        - multiattribute utility
- Ch. 18 - Learning (Sect. 18.1-18.2)
    - types of learning
        - supervised vs. reinforcement vs. unsupervised
    - inductive learning
        - hypothesis
        - training set vs. test set
        - positive vs. negative examples

- Ch 19 - Logical Formulation of Learning (Sect. 19.1)
  - classification and description sentences
  - candidate definition
  - false positive, false negative
  - generalize/specialize hypotheses
  - types
    - current-best hypothesis
    - version space learning
  - **you do not need to know:**
    - how to apply version space learning to a specific problem
- Ch. 20 - Neural Networks (Sect. 20.5)
  - activation functions
  - perceptron
    - linearly-separable functions
    - supervised learning method
      - learning rate, epoch, error
  - multi-layer feed-forward networks
    - be able to calculate output
    - what can be represented?
  - **you do not need to know:**
    - back-propagation
    - recurrent networks