Last update: May 11, 2010

Review for the Final Exam

CMSC 421: Final Review

Final Exam

According to the university exam schedule, the final exam is on Wednesday, May 19, 10:30-12:30, in our usual classroom

- ♢ Open book, open notes
- ♢ No electronic devices

Summary of what we've covered

The midterm exam covered Chapters 1–6 and Common Lisp.

The final exam will include some of that, but will emphasize the following:

logic	7–9
planning	$\approx$ 11-12, but use my lecture slides, not the book
uncertainty	13
Bayesian networks	14
making decisions	16, 17
learning	18, 20.5

I won't ask you much about Chapters 22, 24, and 25

On the next few pages, I'll point out some topics you won't need to know

A few days before the exam, I'll post announcement(s) about other topics that you won't need to know

Chapter 1: Intelligent Agents

- ♢ What AI is:
- ♢ thinking versus acting
- ♢ humanly versus rationally

I won't ask any questions about Chapter 1

Chapter 2: Intelligent Agents

- ♢ Agents and environments
- ♢ Rationality
- ♢ PEAS (Performance measure, Environment, Actuators, Sensors)
- ♢ Environment types
- ♢ Agent types

I won't ask much (if anything) about Chapter 2

Chapter 3: Search

♢ Problem types: deterministic/nondeterministic, fully/partially observable
  example: vacuum world

♢ Tree-search algorithms
  Breadth-first search
  Uniform-cost search
  Depth-first search
  Depth-limited search, iterative deepening

♢ tree search versus graph search

Chapter 4: Informed Search and Exploration

- ♢ Heuristic search algorithms
  - Greedy search
  - A* (two versions)
  - IDA*
- ♢ Heuristic functions
  - admissibility
  - consistency
  - dominance
  - problem relaxation
- ♢ Iterative improvement algorithms
  - Hill climbing, simulated annealing,
  - local beam search, genetic algorithms

We didn't cover sections 4.4 (continuous spaces) and 4.5 (online search)

Common Lisp

- lists, atoms, list notation
- defining your own Lisp functions
- built-in Lisp operators (functions, predicates, special forms, macros)
- recursion, loops, and mapping functions
- passing functions as arguments
- operators for sequences (lists, vectors, strings)
- good programming style
  (no direct questions on this, but don't write sloppy code!)

Chapter 5: Constraint Satisfaction

- ♢ Definition: variables, constraints
- ♢ Representation: constraint graphs
- ♢ Backtracking search

- ♢ Variable selection heuristics:
  - MRV (minimum remaining values)
  - degree (most constraints on remaining variables)

- ♢ Value selection heuristic: least constraining value

- ♢ Pruning techniques
  - forward checking
  - arc consistency (constraint propagation)

- ♢ Problem structure:
  - independent subproblems
  - tree-structured CSPs
  - cutset conditioning

Chapter 6: Adversarial Search

- What type of game:
  deterministic, turn-taking, 2-player, perfect information, zero sum
- Game trees, minimax values
- Alpha-beta pruning
- Depth-bounded search, static evaluation functions
- Node ordering
- Nondeterministic game trees (e.g., backgammon) and expectiminimax

Chapter 7: Logical agents

- ♢ Knowledge-based agents
- ♢ Wumpus world
- ♢ Logic in general—models and entailment
- ♢ Propositional (Boolean) logic
- ♢ Equivalence, validity, satisfiability
- ♢ Inference rules and theorem proving
  - Horn clauses, forward chaining, backward chaining
  - resolution
- ♢ Completeness, complexity

Chapter 8: First-Order Logic

- ♢ Syntax: symbols, atomic sentences, quantifiers, equality, sentences
- ♢ Semantics: interpretations, models, truth
- ♢ Substitutions
- ♢ Wumpus world in FOL

Chapter 9: Inference in First-Order Logic

- ♢ Reducing first-order inference to propositional inference
- ♢ Unification
- ♢ Generalized Modus Ponens
- ♢ Forward and backward chaining
- ♢ Logic programming
- ♢ Resolution

Planning

Related to Chapters 11 and 12 of the book, but based mainly on my lecture slides

♢ Conceptual model, three main types of planners
I won't ask you about these

♢ Classical planning
- restrictive assumptions
- definitions, representation (blocks-world example)

♢ Classical planning algorithms:
- GraphPlan (dinner example)
- FastForward

♢ Task-list planning
- the TFD algorithm (travel examples)

Chapter 13: Uncertainty

- ♢ Random variables, propositions
- ♢ Prior and conditional probability
- ♢ Inference by enumeration
- ♢ Independence and conditional independence
- ♢ Bayes' rule
- ♢ Wumpus example

Chapter 14: Bayesian networks

- ♢ Syntax - what the networks look like
- ♢ Global semantics: joint distribution
- ♢ Local semantics: conditional independence, Markov blanket
- ♢ constructing Bayesian networks
- ♢ Exact inference: enumeration, variable elimination

We didn't cover these sections:
14.3 (hybrid networks),
14.5 (approximate inference),
14.6 (fist-order representations)

Chapter 16, Making Simple Decisions

- ♢ Rational preferences
- ♢ Utilities
- ♢ Multiattribute utilities
- ♢ Human utilities, and the utility of money *(not on the final exam)*
- ♢ Decision networks *(not on the final exam)*
- ♢ Value of information

We didn't cover Section 16.7 (decision-theoretic expert systems)

Sections 17.1–17.3: MDPs

- ♢ Markov decision processes
- ♢ Policies
- ♢ Value iteration
- ♢ Policy iteration

We didn't cover these sections:
17.4 (Partially observable MDPs)
17.5 (decision-theoretic agents)

Section 17.6: Game theory

- ♢ Prisoner's Dilemma
- ♢ Strategies, strategy profiles
- ♢ Dominance, dominant strategy equilibria
- ♢ Pareto optimality
- ♢ Mixed strategies, expected utility
- ♢ Nash equilibria (for both pure and mixed strategies)
- ♢ finding Nash equilibria
  - Battle of the sexes, soccer penalty kicks, morra, Braess's paradox
- ♢ p-beauty contest, iterated elimination of dominant strategies

The final exam won't include the following topics:
roshambo, the IPD with noise, the DBS algorithm

Chapter 18: Learning from Observations

We only covered Sections 18.1–18.3:

- ♢ Inductive learning *(not on the final)*
- ♢ Ockham's razor *(not on the final)*
- ♢ Decision tree learning: attributes, information gain
- ♢ Performance measurement

We didn't cover these sections:
18.4 (ensemble learning)
18.5 (computational learning theory)

Section 20.5: Neural Networks

- ♢ analogy to brain computation
- ♢ nodes/units
- ♢ activation functions: threshold (step), logistic (sigmoid)
- ♢ learning rule
- ♢ perceptrons (single-layer networks with threshold units)
- ♢ perceptron learning rule
- ♢ multi-layer feedforward networks
- ♢ error-backpropagation learning
- ♢ Examples: Nettalk, OCR, ALVINN *(not on the final)*

Chapter 22: Communication and Language

- ♢ Communication *(not on the final)*
- ♢ Grammar, parse trees
- ♢ logical grammars *(not on the final)*
- ♢ Problems presented by real language
  (grammaticality, ambiguity, anaphora, indexicality, vagueness
  discourse structure, metonymy, metaphor, noncompositionality)
  (not on the final)
- ♢ part-of-speech tagging
  - tagsets
  - stochastic tagging
  - Bayes' rule, computing conditional probabilities

If there are any questions about this material, they will be relatively simple

Chapter 24: Vision

- ♢ Perception generally
- ♢ Vision "subsystems"
- ♢ Image formation, color vision
- ♢ Edge detection, noise, smoothing
- ♢ Inferring shape from motion, stereo, texture
- ♢ Inferring shape from edges (Huffman-Clowes line labeling)
- ♢ Object recognition, digit recognition
- ♢ Shape context matching

I might ask a question about Huffman-Clowes line labeling, but not about anything else

Chapter 25, Robotics

- ♢ definition, various examples
- ♢ hand coding of robot controllers
- ♢ path and motion planning
- ♢ configuration parameters, configuration space
- ♢ cell decomposition, voronoi diagrams
- ♢ probabilistic roadmaps: how to generate and use them
- ♢ robot control: sensory-motor functions, modalities

I might ask a question about roadmaps, but not about anything else