CS 161: Fundamentals of Artificial Intelligence

Fall 2019 – Final Study Guide

Material

The final covers all the material but will be more heavily focused on the material that was not covered in the midterm. The following items are in the scope of the final:

1. What is the Turing test? What is a Winograd Schema?
2. Basic LISP programming, lists, recursion
3. Systematic search strategies: Sections 3.0–3.4
4. Informed search strategies: Sections 3.5 (excluding consistency, RBFS, MA* and SMA*)
5. Heuristics: Section 3.6
6. Local search strategies: Sections 4.1
7. Game playing: Sections 5.0–5.4.2, and 5.5
8. Constraint satisfaction: Sections 6.0–6.3.2, 6.5 (until tree decomposition)
9. Propositional logic: Sections 7.0–7.6.1
10. First-order logic: Section 8.1.2–8.3.2, 9.2.2, 9.5–9.5.3
11. Reasoning under uncertainty: Section 13.1 – 13.5 (except "probability density functions", and "where do probabilities come from?")
12. Bayesian networks: Section 14.0–14.2, 14.4–14.4.2
13. Machine learning: Section 18.0–18.3.5, high level of 18.6.4–18.7.4

I highly recommend studying from the book, it contains almost everything we covered, including detailed intuitions and examples. Do not only study from the slides, which are not meant to be reading material, they are only meant to facilitate teaching.

Form

The final will entirely consist of true/false and multiple-choice questions. They will still require that you solve more open-ended exercises (as on the midterm) before answering a multiple-choice question about the result. The exam is closed book. You are allowed to bring a simple calculator but it will not be necessary. The following items will almost certainly be on the final, with more emphasis on the material covered after the midterm.

1. A simple LISP programming exercise (one recursive function).

2. Formalize a real-world problem as a search or constraint satisfaction problem. Come up with an admissible heuristic. Determine branching factors and solution depths.
3. Label nodes in a search tree according to the order in which they will be expanded/generated for any of the search algorithms.
4. Determine completeness, optimality, time, and space complexity for any of the search algorithms.
5. Perform steps of constraint satisfaction backtracking search, for various choices of variable order, value selection, and constraint propagation.
6. Compute minimax or expectiminimax values to solve a game.
7. Perform α-β pruning on a given game tree.
8. Model a problem as a propositional or first-order knowledge base, or as a Bayesian network.
9. Convert a propositional or first-order logic sentence to CNF. Perform Skolemization. Apply standard logical rewritings.
10. Reason using possible worlds/models (decide satisfiability, validity, compute probabilities, etc.).
11. Perform propositional or first-order resolution, unification, apply deductive inference rules, and perform simple DPLL
12. Basic probabilistic reasoning (inclusion-exclusion, marginalization, conditioning, Bayes rule) and checking properties (conditional independence).
13. Identify conditional independence assumptions and joint distribution encoded by a Bayesian network (its semantics).
14. Perform Bayesian network inference by enumeration. Multiply factors and sum out a variable from a factor.
15. Compute the size of a hypothesis space.
16. Learn a decision tree from data and identify optimal tests.

Beyond these items, I may ask a few short questions about other parts of the material, in particular as true/false questions.