CS 161: Fundamentals of Artificial Intelligence

Spring 2017 – Final Study Guide

Material

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 RBFS, MA* and SMA*)
5. Heuristics: Section 3.6
6. Local search strategies: Sections 4.1.2 and 4.1.4
7. Constraint satisfaction: Sections 6.0–6.3.2, 6.5 (until tree decomposition)
8. Game playing: Sections 5.0–5.4.2, 5.4.4, and 5.5
9. Propositional logic: Sections 7.0–7.6.2
10. First-order logic: Section 8.0–8.3.2
11. First-order reasoning: Section 9.0-9.2.2, 9.5–9.5.4
12. Reasoning under uncertainty: Section 13.0–13.5
13. Bayesian networks: Section 14.0–14.2, 14.4–14.4.2
14. Machine learning: Section 18.0–18.3.5, high level of 18.6–18.7.4

Form

Questions will test for insight. Some questions will be factual, but their answers can be found by understanding the material, not by memorizing. The exam is closed book.

The final will 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 the multiple-choice question. The following items will almost certainly be covered.

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. Compute minimax values to solve a game.

9. Perform steps of constraint satisfaction backtracking search, for various choices of variable order, value selection, and constraint propagation.

10. Model a problem as a propositional or first-order knowledge base, or as a Bayesian network.

11. Convert a propositional or first-order logic sentence to CNF. Perform Skolemization. Apply standard logical rewritings.

12. Reason using possible worlds/models (decide satisfiability, validity, compute probabilities, etc.).

13. Perform propositional or first-order resolution, unification, apply deductive inference rules, and perform simple DPLL, forward, or backward chaining.

14. Basic probabilistic reasoning (inclusion-exclusion, marginalization, conditioning, Bayes rule) and checking properties (conditional independence).

15. Identify conditional independence assumptions and joint distribution encoded by a Bayesian network (its semantics).

16. Perform inference by enumeration. Multiply factors and sum out a variable from a factor.

17. Compute the size of a hypothesis space.

18. 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, for a small number of points.