Final Exam Study Guide

CS161 Spring 2022

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 exam:
1. What are common definitions of AI? What is the Turing test?
2. Basic LISP programming, lists, recursion
3. Systematic search strategies:
Edition 3 & 4: Sections 3.0-3.4
4. Informed search strategies:
Edition 3: Section 3.5, excluding consistency, RBFS, MA* and SMA*
Edition 4: Section 3.5.0-3.5.3, excluding consistency; Section 3.5.5, excluding RBFS, MA* and SMA*
5. Heuristics:
Edition 3: Section 3.6
Edition 4: Section 3.6.0-3.6.3, excluding effective depth
6. Local search strategies:
Edition 3 & 4: Section 4.1
7. Game playing:
Edition 3: Sections 5.0-5.4.2, 5.4.4, and 5.5
Edition 4: Sections 5.0-5.3.2, 5.3.4, and 5.5
8. Constraint satisfaction:
Edition 3 & 4: Sections 6.0-6.2.2, 6.3.0-6.3.2, 6.5
9. Propositional logic:
Edition 3 & 4: Sections 7.0-7.5.3, 7.6-7.6.1
10. First-order logic:
Edition 3 & 4: Section 8.0-8.2.7, 8.3-8.3.2, 9.1.2
11. Reasoning under uncertainty:
Edition 3: Section 13.1-13.5 (except "probability density functions")
Edition 4: Section 12.1-12.6 (except "probability density functions")
12. Bayesian networks:
Edition 3: Section 14.0-14.2, 14.4-14.4.2
Edition 4: Section 13.0-13.2.2 (except "d-separation")

Form

The following items will almost certainly be on the final and determine a large part of your score.

1. A simple LISP programming exercise
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, or space complexity for an algorithm related to the ones we saw in class.
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 game tree.
8. Model a problem as a propositional or first-order knowledge base, or as a Bayesian network.
9. Convert a propositional logic sentence to CNF. Apply standard logical rewritings.
10. Reason using possible worlds/models (decide satisfiability, validity, compute probabilities, etc.).
11. Perform propositional and first-order resolution, apply inference rules.
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).