**COMP316A-08 Exam Preparation**

Here are some suggestions that you should consider for exam preparation. These should all be done on paper. (Obviously, other material covered in the course is also relevant, so you should review everything we have covered, and not just what is mentioned here.)

*Heuristic search*

Create an abstract map containing towns in New Zealand, along with distances between them. Apply greedy-best first search and A* search with the straight-line distance to find routes.

*Adversarial search*

Draw the game tree for a simple game of your choice. Then apply minimax search to the game tree. For example, consider a simple two-player game, where, in each move, a player can add a number to the sum of numbers added so far. A player can either add 1, 3, or 5 to the current sum (starting with sum 0), and the player whose move makes the sum exceed 9 wins the game.

*Propositional logic*

Construct a knowledge base consisting of propositional definite Horn clauses and draw an AND-OR graph for it. Practice proving propositions using backward chaining.

*First-order logic*

Construct a knowledge base of first-order definite clauses and apply forward chaining to derive all implied facts.

*Probability theory*

Practice working with the product rule, Bayes' rule, and normalization by considering applications like the meningitis problem from the textbook. Make sure you understand the relationship between joint probabilities and conditional probabilities.

*Bayesian networks*

Practice computing conditional probabilities from a given Bayesian network, based on the global semantics of the network, the process of summing out variables, and normalization.

*Machine learning*

Make sure you understand the version space learning algorithm. Create simple Boolean datasets and apply the algorithm to learn conjunctions of positive literals.

Use the same data, using 0/1 values, to train a simple threshold perceptron (with a bias input). Make sure you also know how to classify examples.

*Grammar learning*

Make sure you understand how the Sequitur algorithm works. Create some arbitrary strings and apply the algorithm to construct a grammar for each string. (Note: you can find a web-based implementation of Sequitur at *www.sequitur.info.*)