Intelligent Agents

Problems for Final Viva

Part 1. Search and Optimization

Each of the following problems is something you may consider solving using some form of graph search or local search or optimization approach:

1. An agent that controls all of the traffic signals in one area. It has sensors that detect traffic flow rate on roads and others that detect when a vehicle is stopped at a signal (but only the first stopped vehicle). The idea is to keep the traffic flowing as much as possible and to reduce the time drivers spend waiting at signals.
2. An agent that is playing two-player Tetris against another agent (or a human for that matter).
3. A business is trying to arrange for the coverage of its working hours by a range of sessional staff. Many of the sessional staff are students or parents (or both) and thus have relatively limited availability. The business is open from 7am to 8pm but needs staff in store from 6am and until midnight for cleaning and stocking. Staff are further divided into counter staff (who serve customers), general staff (who do the cleaning and stocking) and administrative staff (who do management, budgets and similar).
4. Connect Four

For each of the above problems, consider the application of each of the following search methods. Which of the methods below are more appropriate and which are possibly not appropriate (or impossible)? Why? [Hint: The reason should relate to the problem, not your (dis)like of a search strategy.]

- A*
- Hill Climbing with Random Restart
- Minimax with alpha-beta pruning
- Genetic Algorithms

For each problem, choose one appropriate method from the list above and apply that, either by coding it or on paper. If no methods are applicable at all, be prepared to justify this. If you do not have enough information to apply the chosen method, consider (in detail) how you would apply it for some different scenarios. As with the tutorials, you don't need to do all of it, just enough to get an idea of what's happening. You should be able to answer questions about how the method is applied and how it performs on the problem.

Part 2. Constraint Satisfaction and Logic

There are four bridges in four cities, each with a location, a name, a type and a length with details as follows:

1. Of the 11,500 ft structure and the cantilever one, one is in Rome and the other is Star Bridge.
2. Avery Bridge is 8,500 ft.
3. The bridge in Paris is 3,000 feet shorter than North Bridge.
5. The suspension bridge is shorter than the one in Copenhagen.
6. The 11,500 ft one is the camelback bridge.
7. The bridge in Zagreb is 1,500 feet shorter than the suspension one.

Model this using constraint satisfaction. Answer the following questions using AC-3:

- Where is the Bay Bridge?
- How long is the cable-stayed bridge?

You will need to explain your working, including possibly stepping through parts of the process and explaining why you are making certain choices.

Now model this again using logic. Use either Propositional or First-Order Logic, whichever is more appropriate. Presumably you will come up with the same answers. Again, you will need to be able to justify your choices and explain your reasoning.

(This puzzle is taken from puzzlebaron.com.)

Part 3. Logic and Planning

You will be asked specific questions about using Logic to run an agent in Wumpus World. We have talked about this theoretically, but you can deepen your understanding with some coding. Experimenting with the code at https://github.com/s-webber/prolog-wumpus-world may assist you, but only if you look at how it achieves answers rather than just what the results are. Also note that this code has some limits.

You will be asked questions relating to the following, which are not necessarily specific to the code (but the code may help you determine answers).

1. The differences between the use of Propositional and First-Order logics for the deduction of pit locations in Wumpus World.
2. There are parts of the PROLOG in this code that are not “pure” logic. You will need to do some reading on this. The questions will be about the strengths and weaknesses of non-pure logic (such as the “cut”) and why it may be beneficial in some cases.
3. You should familiarize yourself with the use of successor-state axioms for Wumpus World (the code won’t help you with this). You should understand them well enough to be able to create axioms for modifications to the Wumpus World rules.