**CECS 545-50 Exercises 2.3, 2.5, 2.6**

**Ryan Schron**

**Exercise 2.3**

1. False. While an agent capable of observation cannot be perfect, it can be perfectly rational, as rationality is measured by expected performance, rather than actual performance. If a self-driving car lacks an understanding of traffic laws, it may be a rational decision to travel as fast as possible to reach the destination in an optimal time, given the vehicle has no concept of penalty for the decision.
2. True. Reflex agents respond poorly to partial-observability. Using an automated vacuum cleaner as an example, there may exist a general rule of “if device hits a wall, turn left 45 degrees”. Without a collision sensor to activate the rule, the device would continue to try and drive through a wall.
3. True. This would include environments in which the outcome is trivial. That is, there is no wrong answer.
4. False. An agent program only takes as input the current precept, whereas the agent function accepts the entire precept history.
5. False. Some tasks are limited by complexity or memory constraints, and cannot be implemented. Agents work best with a defined scope of action.
6. True. Like part c, this action would be rational if all actions are equally viable.
7. True. This is possible in instances in which the desired output is the same result between the environments.
8. False. The agent would not be rational in an unobservable environment if it did not act. It is rational in the case that the agent acts on the environment, even if it is unable to produce the desired outcome as it works as is expected.
9. False. A perfect agent can still suffer from bad luck, assuming it is playing against a opponent of equal skill (i.e. another perfect agent).

**Exercise 2.5**

Agent – Anything that takes inputs from various sensors and responds to the information using actuators. As an example, a thermostat fits this definition as it senses temperature and adjusts the environment accordingly.

Agent Function – A theoretical function that maps the input from an agent’s sensor to the appropriate actions to respond with.

Agent Program – The practical implementation of an agent function.

Rationality – The idea of behaving in a way that results in a desired outcome based on feedback from the environment. In the context of artificial intelligence, we describe agents as rational if their response to a precept effects the desired change in their environment.

Autonomy – Refers to the ability of an agent to learn from its environment and change it’s responses to precepts accordingly. An agent that lacks autonomy relies entirely on the knowledge of its designer or pre-defined knowledge of the environment.

Reflex Agent – Reflex agents are those that use pre-defined rules (based on the input data) to determine which actions to take. These are called condition-action rules. Reflex agents tend to be susceptible to infinite loops when the environment is only partially observable, as it results in missing precepts and an inability to return associated actions.

Model-Based Agent – Model-based Agents maintain and internal state, based on precept history, that helps define the environment. It also maintains a general set of rules that define the ‘laws of physics’ of the environment. These items together allow the agent to respond in a more rational way.

Goal-Based Agent – Goal-based Agents make decisions to reach a pre-defined destination. To do this, the agent must understand the consequences of an immediate action in addition to how it contributes to reaching the goals. This means the agent must plan future actions rather than simply respond to the current input.

Utility-Based Agent – These agents are concerned with how goals are met rather than just meeting them. They defined a utility function that helps determine actions based on the tradeoffs of decisions with regards to different performance metrics, providing a basis for learning.

Learning Agent – Agents that also include a learning component that evaluates the performance element. These agents take the results of actions as feed back to modify future performance. Another component to these agents is the problem generator, which forces the performance element to avoid always picking what it considers the optimal action so that it may learn for a long term benefit.

**Exercise 2.6**

1. More than one agent program can implement a given agent function given that the programs take a different approach to the solution. Using the first project as an example, a traveling salesman problem could be solved by checking each permutation of nodes and computing the distances as needed. Another program could take a dynamic programing approach with memoization to reduce the complexity of the problem, but both approaches would find the same result.
2. Yes. An agent function with excessive complexity or memory requirements cannot be implemented by any program.
3. Yes. The agent function implemented by the agent program is determined by the sequence of precepts it receives. The alternative is for there to be some variability in choice introduced based on the same sequence.
4. Given n bits there are 2n states possible. This would result in m2^n programs given there are m choices to make for each state.
5. Increasing computational power does not change the agent function, as it is defined by the environment. It could improve learning and other side functions for the agent program.