**1.Define in your own words the following terms: agent, agent function, agent program, rationality, autonomy, reflex agent, model-based agent, goal-based agent, utility-based agent, learning agent. [5 points]**

* Agent: an agent is anything that have sensors to perceive environment and use effectors/actuators to act in the environment.
* Agent function: agent function is something abstract that use to describe how the agent work in the environment with what action. From percept sequences to actions.
* Agent program: agent program means the actual program in the agent. For instance a specific implementation in a system.
* Rationality: rationality is the actions that can be considered as “right”, or make the agent to be most successful, to maximize its expected utility. And rationality should be based on the performance measure, agent’s prior knowledge of the environment, actions that the agent can perform and agent’s percept sequence to date.
* Autonomy: autonomy means the agent can inherent knowledge. Do decisions based on its own experience without direct human interference.
* Reflex agent: reflex agent is a simple agent that based on the condition-action rules, without inherent knowledge, only use current percept to do actions. And it only works in a fully observable environment.
* Model-based agent: This agent uses percept history and its actions are selected based on the model and percepts. Model-based agent has internal states
  + Goal-based agent: The goal helps agent to select actions. It’s a expand on the model-based agents. The right actions are the actions that can help agent to accomplish its goal.
* Utility-based agent: a more complex goal-based agent. Because the previous one can only distinguish action to the goal or not. But in a multi-goal agent, can use utility function maps a state to a measure of the utility of the state. Utility-based agent can use utility functions to define how desirable a state is.
* Learning agents: an agent that can use learning element to help making improvements and performance element to select external actions. A learning agents can learn from its own experiences.

**2.Both the performance measure and the utility function measure how well an agent is doing. Explain the difference between the two.**

Performance measure: this is how we evaluate an agent behavior. It has two parts: One is subjective which if agent evaluates himself. Another one is evaluation done by observer; people define what actions should be considered as successful.

Utility function: this is just the function that is inside an agent. The agent uses to evaluate its own performance and how desirable the states are.

**3. In this question we explore further the differences between agent functions and agent programs. [2 points]**

**(a) Can there be more than one agent program that implements a given agent function?**

Yes, for example, a e-directory program’s agent functions are to get the explanation of given words. In order to reach that goal, there could be two agent programs: one is to received words and search their meaning through the internet. Another agent programs could be store data in local, then after receiving words, search local database instead of through internet.

**(b) Are there agent functions that cannot be implemented by any agent program?**

Yes, for example, the agent functions can specify if a machine Is broken, then let the machine send an alert message. But this can’t be done because the machine is broken now, and it can’t do anything.

**(c) Given a fixed machine architecture, does each agent program implement exactly one agent function?**

Yes, given a percept sequence, then the agent program will select an action.

**(d) Given an architecture with n bits of storage, how many possible agent programs are there? Is this enough, that is, might there be environments for which there are no good agent programs?**

2^n possible agent program. In most cases, 2^n is greater than n which means for most of the times this environment would lack of memory and some programs have to wait until previous programs done. So it shouldn’t be enough.

**4. Let us examine the rationality of various vacuum-cleaner agent functions in various environments. [2 points]**

**(a) Describe a rational agent function for the modified performance measure that deducts one point for each movement. Does the corresponding agent program require internal state?**

Check if the current position is dirty, if so, suck, then do the check action again. Whenever after doing check action and find the position is clean, then do move action. After each movement, do all the same “check, if dirty suck, if clean move” action. After cleaning all the positions, stop working.

Because each movement will deduct one point, this agent program needs an internal state. Otherwise, if after clean A and move to B position, then the vacuum-cleaner doesn’t remember whether or not it cleaned position A, the machine will keep working around all positions and those needless movements will cost lots of points.

**(b) Discuss possible agent designs for the cases in which clean squares can become dirty and the geography of the environment is unknown. Does it make sense for the agent to learn from its experience in these cases? If so, what should it learn?**

Yes, it make sense. If from agent’s experience, clean squares can become dirty, then the machine can set a time interval and check all cleaned squares periodically. Compared to keep checking all the time, this method could minimizing movements.

If the geography of the environment is unknow, then the agent can not only do the simple move right one step than move back things. Instead, it should keep a internal model and build its own map. Then keep exploring the environment. For example, locate its initial position, then go straight until something(like walls)stop it, then change direction and keep going. In the end the agent would have the map of the environment and finally can clean all the squares without any unknown place.

**Exercise 1.2 (10 points) The following exercises all concern the implementation of environments and agents for the vacuum-cleaner world. Programs need to be written in C/C++, Java or Python.**

1. **Implement a performance-measuring environment simulator A B for the vacuum-cleaner world depicted in the figure on the right (Figure 2.2 on page 36 of our textbook). Your implementation should be modular so that the sensors, actuators, and environment characteristics (size, shape, dirt placement, etc.) can be changed easily.**

This world would contains two part:

Firstly the environment part should only contains two location, location A and location B. Each time there will be some dirt randomly set on these locations.

Another part is the vacuum-cleaner. It should be a machine that only has two movements: move right or left. And it also has two actions: check the current square if it’s clean and suck.

After running the following program, the vacuum would randomly appear on one of the squares. It will know its location and check if the square is clean or not and suck if it’s dirty. Then it will move to another location. If both squares are clean, The machine will rest for some seconds(can set up by ourselves) and then work again.