

1. Which of the elements between columns “A” and “B” are mutually exclusive and why?

- 1-d: Known and Unknown Environments are mutually exclusive because in a known environment the value of possible outcomes are known, if the environment is unknown, the agent don't know the environment and will have to learn how it works, therefore they can't exist at the same time since they are opposite.
- 2-g: Single-Agent and Multiagent Environments are mutually exclusive because in a single-agent environment there is only one agent that is making decision/action, the decision is made by itself and not affected by another agent. In multiagent environment there is two or more factors involve or being affected when taking an action.
- 3-b: Continuous and Discrete Environments are mutually exclusive because in a discrete environment there are only a set of moves possible, like in a game of chess. While in a continuous environment, the possibilities are infinite, reroutes can happen anytime, and the number of actions performed cannot be count.
- 4-e: Deterministic and Non-deterministic Environments are mutually exclusive because in a deterministic environment, the next state is being determined by the current state therefore we know the next state. However, in a non-deterministic environment, there's a lot of uncertainty of what the next state could be, and that we don't know what's the next state.
- 5-f: Unobservable and Partially Observable Environments are mutually exclusive because they can't exist simultaneously. An unobservable environment is the agent having no sensors at all, while a partially observable environment has sensors, but the sensors might not be accurate, there's missing parts, or there's other factors affecting the sensors.
- 6-a: Dynamic and Static Environments are mutually exclusive because a static environment only focuses on itself when making an action, it doesn't ask for permission. Dynamic environment on the other hand needs to constantly asked the agent what it wants to do.
- 7-h: Competitive and Cooperative Environments are mutually exclusive because a taxi-driving environment can be both competitive and cooperative. It can be cooperative trying to avoid collisions with other cars, however it can also be competitive, whenever there's only one car that fits in a certain lane or parking spot.
- 8-c: Episodic and Sequential are mutually exclusive because episodic environments does not need to think ahead, the next action does not depend on the previous action. However, in sequential action, your current action will affect the next action or all future decisions.

2. Give PEAS description for the robotic-soccer environment. [2\*4]

- Performance Measure – safety when playing with other players, play well (kicking ball to correct team member, kicking ball to the goal, taking the ball from

opponent), stop playing when referee tells to, guard other players even if they don't have the ball

- Environment – soccer field, teammates, opponents, people watching the game, referee
- Actuators – robotic legs, robotic hands, camera/display for robot to see where to go
- Sensors – touch sensors (legs, hands), camera, hear sensor (for referee whistle)

3. Define rational agent and autonomous agent in your own words.

- A rational agent acts depending on what is the best solution. Autonomous agent has fixed outcome no matter what the uncertainty is. For example, an autonomous weapon, once you launch the missile, it will only focus on the target and even you change your mind it will not stop. Not unless you destroy the agent itself. While a robotic vacuum cleaner is a rational agent, it will only clean dirt if it sees dirt. Therefore, if it doesn't see dirt, it will not clean. Rational agent's goal is to achieve the best outcome unlike autonomous agents, no matter what the situation is, it has a fixed outcome.

4. You have the agent program for simple reflex agent as shown below

If status = Dirty then suck Else if location = A then return right Else if location = B then return left
--

Now suppose you can use one more action "NoOp" which does nothing.

- a. Would it be desirable to use this action in this agent program to prevent vacuum cleaner from moving after all squares are cleaned? If yes, modify the simple reflex agent program adding "NoOp" action and if not give an argument that despite the option of "NoOp" the agent is bound to keep moving. [4].

Yes, it would be desirable to use NoOp function. Simple reflex agent program with "NoOp" action, this agent flags location when it's already visited, so when it checks that the location has already been visited or has been flag, the vacuum will do nothing or "NoOp"

if status = Dirty then return suck else if location = flag, then return NoOp else if location = A then flag location A, return Right if location = flag, then return NoOp
--

- b. If you have a choice of using other type of agents, which one would you choose and how would it prevent vacuum from moving after all squares are cleaned? [3]

I would choose goal-based agent because my agent will get input from the actions it makes, how the environment evolves, and other factors. With this my agent will know if the environment is clean since it gets input on how the environment evolves (clean/dirty), and will be aware of what it's doing, whether the agent already clean the area or not.