10/14/2018

|  |
| --- |
| Faculty  **Dr. A. F. M. Saifuddin Saif**  Assistant Professor |



|  |  |
| --- | --- |
| Artificial Intelligence | Rashid, Md.Mamunur  ID:15-29161-1 |

**1. )Explore the differences between agent functions and agent program.**

*1. Can there be more than one agent program that implements a given agent functions? Given an example, or show why one is not possible.*

**Ans:** Yes. Assume we are given an agent function whose actions only depend on the previous p percepts. One program can remember the previous p percepts to implement the agent function, while another could remember greater than p percepts and still implement the same agent function.

***2*.** Are there agent functions that cannot be implemented by any agent program?

**Ans*:*** Yes. Consider an agent whose only action is to return an integer, and who perceives a bit each turn. It gains a point of performance if the integer returned matches the value of the entire bit string perceived so far. Eventually, any agent program will fail because it will run out of memory.

***3.*** Given a fixed machine architecture, does each agent program implement exactly one agent function?

**Ans:** Yes. Given a percept sequence, an agent program will select an action. To implement multiple agent functions this would require the agent program to select different actions, given the same percept sequence.

**2).**Show that, simple vacuum cleaner agent function described previously is indeed rational under the some assumptions. Write down the assumptions in your opinion.

**Ans**: To be rational in these circumstances it maximizes its performance measure 1 point per each clean square. If there is dirt here, suck gains you a sure point now and doesn’t impact your chances of getting other dirt in the future, so suck. If there’s no dirt here, the only possible way of getting another point in the future is to move to the other square

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

**Ans**: For the case in which each movement costs one point, the agent should stop after checking both squares A and B and removing any dirt found in order to reduce needless movements and loss of points. The corresponding agent program would require internal state in this case, as it would need to remember checking both squares before it stops performing its task. If it finds dirt at square A, cleans it, then moves to square B and finds no dirt, and then does not remember whether or not it checked square A, it will check A again, and then repeat with square B and continue making unnecessary movements, thus losing points.

If the squares did not permanently remain clean, however, the agent could stop for some fixed amount of time after checking and cleaning both squares and repeat its task. It could work at some time interval, checking both squares, cleaning any dirt found, stopping for 1 hour or so, and then repeating the process. This would allow the agent to minimize movements and point loss while keeping the squares clean if they were to become dirty again.

**4)**. For each of the following agents, develop a PEAS description of the task environment:

*1. Robot soccer player:*

Performance measure: Win / Lose

Environment: Field

Actuators: Legs, head, upper body

Sensors: Eyes, ears

Partially observable, multiagent, stochastic, sequential, dynamic, continuous

***2***. Internet book shopping agent:

Performance measure: Cost of book, quality/relevance/correct edition

Environment: Internet’s used book shops

Actuators: Key entry, cursor

Sensors: Website interfaces, browser

Partially observable, multiagent, stochastic, sequential, dynamic, continuous

***3.*** Autonomous Mars rover:

Performance measure: Surface area mapped, extraterrestrial life found

Environment: Surface of Mars

Actuators: Steering, accelerator, break, probe arm

Sensors: Camera, sonar, probe sensors

Partially observable, single agent, stochastic, sequential, dynamic, continuous

***4***. Mathematician’s theorem-proving assistant***:***

Performance measure: Time requirement, degree of correction

Environment: The theorem to prove, existing axioms

Actuators: Accept the right theorem, reject the wrong theorem, infer based on axioms and facts

Sensors: Input device that reads the theorem to prove

Partially observable, single agent, stochastic, sequential, dynamic, continuous

***5).*** Can a simplex agent be perfectly rational for any specific environment? Explain***.***

**Ans:** A simple reflex agent cannot be perfectly rational in this environment because the agent never stops and its score will continue downward. It also has no idea whether there are even any unclean spaces before moving.

6). Can a simple reflex agent with a randomized agent function out performs a simple reflex agent? Design such an agent and measure its performance on several environments.

**Ans:** Yes, because it can avoid the stuck position mentioned above for the simple reflex agent. One possible design cleans up dirt and otherwise moves randomly. This design would be: “If Dirty, then Suck, else randomly move either Left, Right, Up and Down.” It works reasonably well in compact environments.

**7**).The vacuum environments in the preceding exercises have all been deterministic. Discuss possible agent programs for each of the following stochastic versions:

*A. Murphy’s law: twenty- five percent of the time, the suck fails to clean the floor if it dirty and deposits dirt onto the floor if the floor is clean. How is your agent program affected if the dirt sensor gives the wrong answer 10% of the time.*

**Ans:** For a reflex agent, this presents no additional challenge, because the agent will continue to suck as long as the current location remains dirty. For an agent that constructs a sequential plan, every Suck action would need to be replaced by “Suck until clean”. If the dirt sensor can be wrong on each step, then the agent might want to wait for a few steps to get a more reliable measurement before deciding whether to Suck or move on to a new square. Obviously, there is a trade-off because waiting too long means that dirt remains on the floor (incurring a penalty), but acting immediately risks either dirtying a clean square or ignoring a dirty square (if the sensor is wrong). A rational agent must also continue touring and checking the squares in case it missed one on a previous tour (because of bad sensor readings). It is not immediately obvious how the waiting time at each square should change with each new tour. These issues can be clarified by experimentation, which may suggest a general trend that can be verified mathematically

***B.*** Small children: At each time step, each clean square has a 10% chance of becoming dirty. Can you come up with a rational agent design for this case.

**Ans:** In this case, the agent must keep touring the squares indefinitely. The probability that a square is dirty increases monotonically with the time since it was last cleaned, so the rational strategy is, roughly speaking, to repeatedly execute the shortest possible tour of all squares. This problem is also a partially observable Markov decision process.