Homework 1

CS 411 - Artificial Intelligence I - Fall 2019

Matteo Corain 650088272

September 5, 2019

1 Question 1

The environment in which the part-picking robot can possibly operate is described by the following properties:

- *Partially observable*: this environment may be considered only partially observable because the robot's capability of sensing the environment is essentially limited to a camera pointing towards the part to be moved. Agent has no knowledge of what is outside this restricted area under its sight; it simply analyzes and classifies one part at a time, whenever one gets inside the camera angle.
- *Single-agent*: the part-picking robot does not interact, within its environment, with other agents, neither in a competitive nor in a cooperative way. In fact, the robot simply interacts with the parts being classified, which are of course passive with respect to its actions.
- *Stochastic*: the environment in which the robot works cannot be considered completely deterministic since it is a physical environment, in which unforeseen events may occur; for this reason, we cannot have the certainty that, even when the environment is correctly sensed and the right action is chosen, the actual final state of the environment will correspond to the expected one.
- *Episodic*: the robot essentially carries out a classification task, which is implemented in a succession of atomic episodes, all of the same kind, which do not depend one on another. Each part is analyzed and classified in isolation, before the robot moves to the next one.
- *Dynamic*: the environment is dynamic since it evolves in autonomy, even without the intervention of the agent; for example, the conveyor belt is in continuous operation, regardless of whether the robot is picking and classifying correctly the parts or not.
- *Continuous*: the environment is continuous both in time and in states; for instance, the conveyor belt is in continuous operation, the characteristics of the classified parts vary with continuity and their relative positioning the belt may not be precisely defined. Consequently, also the percepts (e.g. camera continuously shooting) and actions (e.g. adjustment of the trajectory of the mechanical arm) of the robot must vary with continuity.

Having characterized the environment in which the part-picking robot operates, it is possible to describe its functioning by means of the PEAS framework:

- *Performance measure*: the performance measure for this kind of agent is related to the accuracy of the classification task, evaluated in terms of percentage of parts correctly classified.
- *Environment*: the environment in which the robot operates is made of a conveyor belt to move the parts underneath the robot and a set of bins in which the robot has to place the parts according to their category.
- *Actuators*: the robot acts on the environment thanks to a mechanical arm, capable of moving the parts from the belt to the correct bin.
- *Sensors*: the agent senses the environment essentially through a camera, whose data are elaborated in order to detct the distinctive features of each class of parts.

2 Question 2

An agent is defined *rational* when, given the information it was able to gather from the perception of its environment and possibly its prior knowledge, it acts on the same environment in the way that makes it most likely to achieve a defined goal.

Formally, a rational agent is described by an *agent function* $f^* : \mathcal{P}^* \to \mathcal{A}$ such that, for each element of the input set $\mathcal{P}^*$ (that is, for each finite sequence of percepts collected from the environment), selects from the output set $\mathcal{A}$ the most suitable action in order to maximize the *expected value* of its associated *reward function.* If we denote with r_t the value of the reward obtained at time t, a rational agent aims to maximize on a certain lifespan T the expression:

$$\mathbb{E}\left[\sum_{t=0}^{T} r_t\right]$$

It is important to underline that the goal of a rational agent is not to maximize the reward function itself, but instead its expected value; the rationality of an agent, in fact, is bounded by two factors:

- The agent has a limited knowledge of the world deriving from its current and past percepts: the selection of the action is therefore based on the percepts of the agent, even if they do not reflect the actual state of the world (no agent is *omniscient*).
- The agent cannot predict the future, so there is no certainty that, even in case the best action is always chosen, its outcomes will turn out to be the expected ones (no agent is *clairvoyant*).

Therefore, an agent may simply try to predict the consequences of its actions and, based on the expected outcomes, select the one that may possibly yield the best results. As a consequence, a rational agent is not the one that always does the *right* thing; instead, it is the one that performs the actions that are *most desirable* according to its performance measure and given its limited knowledge of the environment.

An agent is defined *autonomous* when it is able to perform its operations based only on the information it was able to gather from the environment, rather than relying on some sort of built-in knowledge. Autonomy should be the goal of any rational agent and is achieved through a *learning process*, which, starting from the agent's percepts, transforms them into useful knowledge, making it more adapt to work in its environment.

3 Question 3

3.1 Sub-question a

The vacuum cleaner as presented implements a *simple reflex* agent program, characterized by the fact that the action to take is determined by the sole analysis of the current percepts, ignoring the succession of older ones. For this reason, in this case the introduction of a *NoOp* operation will have no benefit for the agent: in fact, the vacuum cleaner cannot remember whether the other square has already been cleaned or not.

Lacking this possibility, the agent is forced to keep moving between A and B, only to realize continuously that the square is already clean. This derives directly from the partial observability of the vacuum cleaner environment, which does not allow the agent to perceive contemporarily the presence of dirt in both locations.

3.2 Sub-question b

A *model-based reflex* agent would be a more suitable choice for this kind of environment. In this case, the vaccum cleaner may maintain an *internal state* information representing the dirt status in both locations, as observed on the last time the agent passed on each location. Under the given assumptions, which constitute the *model* of the world according to which the environment evolves both autonomously and consequently to the agent's actions, a simple set of *condition-action rules* to implement such an agent is described by the following algorithm:

```
function MODEL-BASED-VACUUM-AGENT(location,status)
    $map[location] \leftarrow status$
    if $status = Dirty$ then return $Suck$
    else if $location = A \wedge map[B] \neq Clean$ then return $Right$
    else if $location = B \wedge map[A] \neq Clean$ then return $Left$
    else return $NoOp$
    end if
end function
```

The *map* variable represents the known characteristics of the environment as previously described. Given the known geometry of the environment, initially all squares are set conservatively to the *Dirty* state; at each iteration, the agent first of all perceives its location and the dirt status, setting the appropriate value in the state variable. Then, it works as in the previous case, the only difference being that, if a location has already been considered for cleaning, it will not be considered again: assuming that clean squares remain clean, once the status of a location is set to *Clean*, it will never be changed afterwards, preventing the agent from passing there again. When all the locations will have been cleaned, the agent will simply perform *NoOp* actions, without needing to bounce back and forth between the two locations.

Using the proposed program, the obtained agent is indeed rational. In fact, according to the first assumption, the reward function that the agent has to maximize may be formalized as:

$$\sum_{t=0}^{1000} (r_{A,t} + r_{B,t})$$

Where the reward is defined, for each time step t and each square S, as:

$$r_{S,t} = \begin{cases} 1 & \text{if } S \text{ is clean at time } t \\ 0 & \text{otherwise} \end{cases}$$

Its expected value for each time step t and each square S can be computed as:

$$\mathbb{E}\left[r_{S,t}\right] = 1 * \mathbb{P}\left[S \text{ is clean at time } t\right] + 0 * \mathbb{P}\left[S \text{ is not clean at time } t\right]$$

Under the hypothesis that clean squares remain clean and based only on the observations of the agent, there are two possible values for the expected value of $r_{S,t}$:

- $map[S]$ is set to *Clean* at time t:

$$\mathbb{E}\left[r_{S,t}\right] = 1 * 100\% + 0 * 0\% = 1$$

- $map[S]$ is set to *Dirty* at time t:

$$\mathbb{E}\left[r_{S,t}\right] = 1 * 0\% + 0 * 100\% = 0$$

As long as there are squares which are supposed to contain dirt, the vacuum cleaner will act in a way that, in the following iterations, an additional point will expectedly be awarded, moving to the (assumedly) dirty location and sucking out dirt. After all squares have been considered and possibly cleaned, the agent is bound to perform no-operations, since it cannot expect to achieve a greater reward in doing any additional action.