Week 1: Tasks and Agent Types + Prolog Programming (Open Learning Week 2)

Tutorial 2: Tasks and Agents

2.1: a) PEAS Descriptions

a) Present to your tutorial group the **PEAS** model (Performance measure, Environment, Actuators, Sensors) (Russell & Norvig 2.3.1) you developed for the activity from "The PEAS Model of an Agent".

Were there any additional aspects of features of the task that need to be added to your model?

Image source Netflix, Spam, Roomba

1. Stock market trading system
2. 7-11 food delivery drone
3. Taxi driver
4. Recommender System Online Dating
5. Recommender System Books or Movies (Netflix, Amazon)
6. Piano teacher
7. Fighter drone
8. SPAM filtering
9. ROOMBA - robotic vacuum cleaner

2.1: b) PAGE Descriptions

Task Environments - Properties of Environments

b) Discuss PAGE (Percepts, Actions, Goals, Environment) (Russell & Norvig, 1995, 2.3) pecifications for the following agents (there is not necessarily a single correct answer):

(i) Robocup dog
(ii) (ii) Self-driving car

(iii) Air traffic controller
(iv) Poker playing bot

In each case, decide whether the environment is *fully or partially observable*, d*eterministic* or *stochastic*, *episodic* or *sequential*, *static or dynamic*, and *discrete or continuous.*

What architecture(s) would be appropriate for these agents?

2.2: Classifying Environments & Choice of Agent Type

Use the PEAS model to classify an environment from the tutorial in Week 1.

image source Cairo Traffic

Spin to be assigned a random task environment to classify from the options shown above.

For the task that you received from the spinner above, please complete the following:

1. Classify the task according to the properties given in lectures in the Classifying Environments page of Week 2 module. Give a one-line justification for each choice.

2. Select a suitable agent type (or, discuss the relative merits of the different agent types presented).

Classifying Tasks

Task:

Simulated or Situated/Embodied ? Why ?

Static or Dynamic ? Why ?

Discrete or Continuous ? Why ?

Fully Observable or Partially Observable ? Why ?

Deterministic or Stochastic ? Why ?

Episodic or Sequential ? Why ?

Known or Unknown ? Why ?

Single-Agent or Multi-Agent ? Why ?

Agent:

What type of Agent would be most suitable? Why

Programming Question: The students will work on Question 2.3a, and 2.3b alone and in the Lab

2.3.a Sorting in Prolog

1. Write a prolog predicate insert(Num, List, NewList) that takes a number Num along with a list of numbers List which is already sorted in increasing order, and binds NewList to the list obtained by inserting Num into List so that the resulting list is still sorted in increasing order.

2. Write a predicate isort(List,NewList) that takes a List of numbers in any order, and binds NewList to the list containing the same numbers sorted in increasing order. Hint: your predicate should call the insert() predicate from part .

3. Write a predicate split(BigList,List1,List2) which takes a list BigList and divides the items into two smaller lists List1 and List2, as evenly as possible (i.e. so that the number of items in List1 and List2 differs by no more that 1). Can it be done without measuring the length of the list?

4. Write a predicate merge(Sort1,Sort2,Sort) which takes two lists Sort1 and Sort2 that are already sorted in increasing order, and binds Sort to a new list which combines the elements from Sort1 and Sort2, and is sorted in increasing order.

2.3.b Bonus Challenge:

This exercise is intended for students who have already studied sorting algorithms in another programming language.

Write a predicate mergesort(List,NewList) which has the same functionality as the isort() predicate from part above, but uses the MergeSort algorithm. Hint: you will need to use the split() and merge() predicates from parts and above.

2.4 Compare the following two definitions of 'autonomy' for agents: (i) the agent's behaviour is partly determined by its perceptual experience rather than the knowledge of the designer (Russell and Norvig), and (ii) the agent can control its own behaviour (from the lectures). Give an example of an agent that satisfies definition (i) but not definition (ii), and vice versa.

2.5 Consider the Delivery Robot from Poole and Mackworth. Does the simple strategy of always turning left when encountering an obstacle enable the robot to always reach its goal(s), or can the robot become stuck?