Knowledge Representation

SEMESTER 1 2018 - Assignment 1

INSTRUCTOR: Dr Ruairí O'Reilly

ruairi.oreilly@cit.ie

Topics schedule

Week		
1	No lecture given	No lecture given
2	Intro to AI	Intelligent Agents
3	Logical Agents (i)	Logical Agents (ii)
4	Problem Solving & Search*	Problem Solving & Search
5	Visualisation with MatPlotLib (i)	Visualisation with MatPlotLib (i)
6	First Order Logic*	Inference in First-Order Logic
7	Knowledge Representation	Quantifying Uncertainty
8	Probabilistic Reasoning (i)	Probabilistic Reasoning (ii)
9	Assignment 1 due - 12/11/18 @ 23:59	

* Note: Delivery differs slightly between Full-Time and Part-Time

Completion Date

Submission date is based on your class group Full-time Day or Part-time Night:

Full-time Day - Monday the 12th of November @ 23:59

Part-time Night - Monday the 19th of November @ 23:59

Agent-based representation of knowledge

Getting started with agent-based representations

i) Create a simple agent-based game in a 1-dimensional world. Ideally, the objects within the world relate will relate to a knowledge-based representation scheme (Give context for the decisions you make in relation to the world, its agent, percepts and the actions carried out). Specify a PEAS description including at least four percepts and four actions of your choosing and specify the conditions under which the game is complete.

Implement this game in Python using any of the libraries made available from the AIMA python repo. The specification for the Agent, Environment, and Program to enable the game to run must be included. All code should be in a single python file.

NOTE: You should assume that your code will be executed in the same directory as the AIMA data repo. As such it is critical you use relative paths consistent with this.

Write a clear and concise description of the agent-based game. The purpose of this is to articulate an understanding of the underlying concepts being implemented both from a theoretical and practical perspective.

Extending the world

ii) Extend the world into a 2D environment and include a means of visualising your agent's progress within the world. Introduce a piece of bespoke functionality of your choosing (random spontaneous events, performance measure indicators, more complex manoeuvres - e.g. think a rope ladder that can cross a pit)

Inferring knowledge from an agent-based representation

iii) Using either the world you created as a solution to Part (ii) or the version of the wumpus world made available from AIMA, investigate different search-based approaches for inferring information about the world. Choose two of the search based approaches from the AIMA data repo and integrate them with your chosen World to map the potential states. Compare and contrast your results and critically evaluate the search based

approaches to representing knowledge in the world. Include the code and results of this investigation as part of the documentation submitted.

Submission

As a number of you are having trouble visualising the world state via an IDE such as PyCharm or Spyder - I would recommend implementing your solution using Jupyter notebook. This is not a significant challenge and will make your lives significantly easier:

a) It reduces development effort for Part (ii)
b) It enables you to combine code & text-based discussion. This will enable an incremental demonstration of the environment you've created. You can display pieces of code (e.g. in relation to PEAS env, performance measure, bespoke game features) and articulate your understanding.

[Getting started with Jupyter notebook tutorial]

Marking rubric

Qts	1h1 (>70%)	2H1 (60-69%)	Pass (40-59%)	Fail (<40%)
i - 40%	An advanced implementation of agent behavior, the integration of performance measure, runs automatically in a randomised environment.	A successful implementation of an agent, the integration of performance measure, runs automatically in a randomised world.	A limited attempt - successful implementation of agent, the integration of performance measure, only runs manually.	Little or no understanding, the code fails to run or is error-ridden.
ii - 30%	2D environment, complex bespoke game features included e.g. World traversal to identify objects enabling obstacles to be overcome, visualisation of the world state	2D environment, bespoke game features included e.g. World traversal to identify objects enabling obstacles to be overcome, visualisation of the world state	2D environment, the simple game features added, visualisation of the world state, only runs manually.	2D environment not implemented, no visualisation, no additional world features.
iii - 30%	Integration of search-based approaches to mapping world state, demonstrating the process/how it works. High level of understanding articulated via the presentation of results.	Integration of search-based approaches to mapping world state, demonstrating the process/how it works. Sufficient level of understanding articulated.	Integration of search-based approaches to a world. An understanding of Mapping world state demonstrated, level of understanding articulated. Limited level of understanding.	No Integration of search-based approaches. No mapping of the world state.

Doc req	Excellent understanding demonstrated via documentation and articulation of underlying concepts.	Good understanding demonstrated via documentation and articulation of underlying concepts.	Minimal understanding demonstrated, acceptable documentation and limited articulation of underlying concepts.	Limited understanding demonstrated, poor documentation and little/no articulation of underlying concepts.