

# Knowledge Representation - W6

## Lab: First-Order Logic

Ruairi D. O'Reilly

### I. Introduction

This weeks lab is based on the lecture on First Order Logic [1]. This weeks lab aims to enhance your understanding of First-Order Logic, the associated expressiveness from both a practical and theoretical perspective.

#### A. Prep-work

Review and complete the following up as far as First-Order Logic Knowledge Bases (we cover this next week):

- logic.ipynb - Jupyter notebook
- logic.py - underlying library containing majority of implementation code.

Note: Your solution should be generated in Spyder as a .py file importing code as required from the AIMA repo (See the file provided for an example of importing it form a sub-directory). I will not accept .ipynb submissions and your code should import what it needs from a parent directory (no libraries to be submitted along with solutions).

#### B. logic.ipynb

Pay particular attention to the following sections as it will aid your understanding of FOL:

- Inference in Propositional Knowledge Base
  - Proof by Resolution
  - Forward and backward chaining
  - Effective Propositional Model Checking
- First-Order Logic Knowledge Bases: FolKB
  - Criminal KB

#### C. FOL

Formulate a paragraph of your own choosing to represent a knowledge base (based on Criminal KB). Implement a “FolKB”based on this. This will be used in next weeks lab with regards inference in FOL.

E.g. If you follow rugby a paragraph describing the conditions under which a would be decided for the Six Nations. You have free reign, start simple.

### II. Submission

Submit your solution by the due date as a single “.py” file using the following naming convention.

“W<Week\_num>\_Lab\_<Surname>\_<First name>\_<Student Number>.py”

e.g.

“W6\_Lab\_OReilly\_Ruairi\_R123456.py”

### References

- [1] S. Russell and P. Norvig, “Ai a modern approach,” Learning, 2005.