Knowledge Representation - W4 Lab: Logical Agents

Ruairi D. O'Reilly

I. Introduction

This weeks lab is based on the lecture on Logical Agents [1]. This weeks lab aims to enhance your understanding of logical agents 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. AIMA

Which of the following are correct?

1) $False \models True$.
2) $True \models False$.
3) $(A \wedge B) \models (A \Leftrightarrow B)$.
4) $A \Leftrightarrow B \models A \vee B$.
5) $A \Leftrightarrow B \models \neg A \vee B$.
6) $(A \wedge B) \Rightarrow C \models (A \Rightarrow C) \vee (B \Rightarrow C)$.
7) $(C \vee (\neg A \wedge \neg B)) \equiv ((A \Rightarrow C) \wedge (B \Rightarrow C))$.
8) $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B)$.
9) $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B) \wedge (\neg D \vee E)$.
10) $(A \vee B) \wedge \neg(A \Rightarrow B)$ is satisfiable.
11) $(A \Leftrightarrow B) \wedge (\neg A \vee B)$ is satisfiable.
12) $(A \Leftrightarrow B) \Leftrightarrow C$ has the same number of models as $(A \Leftrightarrow B)$ for any fixed set of proposition symbols that includes A, B, C.

C. Propositional logic and models

Consider the problem of deciding whether a propositional logic sentence is true in a given model.

a Write a recursive algorithm PL_TRUE(s,m) that returns true if and only if the sentence s is true in the model m (where m assigns a truth value for every symbol in s). The algorithm should run in time linear in the size of the sentence. (Alternatively, use a version of this function from the online code repository.)

b Give three examples of sentences that can be determined to be true or false in a partial model that does not specify a truth value for some of the symbols.

c Show that the truth value (if any) of a sentence in a partial model cannot be determined efficiently in general.

d Modify your PL-TRUE? algorithm so that it can sometimes judge truth from partial models, while retaining its recursive structure and linear run time. Give three examples of sentences whose truth in a partial model is not detected by your algorithm.

e Investigate whether the modified algorithm makes TT-ENTAILS? more efficient.

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.

"W4_Lab_OReilly_Ruairi_R123456.py"

References

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