AI Large Practical (2015–26)
Assignment 2

Stuart Anderson

28 October 2015

1 Introduction

Before launching into this assignment, it's useful to remind yourself of the learning outcomes for the course (see http://www.drps.ed.ac.uk/15-16/dpt/cxinfr09018.htm). As a result of successfully completing the course, you should have learnt how to do the following:

- Design and implement a complex system.
- Consider alternative designs, both for internal properties, and as ways of tackling a given problem.
- Read technical papers, and explain their relevance to the chosen approach.
- Design and carry out appropriate experiments, and explain the methodology involved.
- Write a scholarly report, suitably structured and with supporting evidence.

The various parts of the assignment address one or more of these outcomes.

2 Assignment Overview

Assignment 2 requires you to extend the functionality of the system that you developed in Assignment 1. The new functionality should allow you to model a situation where there is a proponent and opponent arguing about some conclusion, involving exchange of arguments and the notion of *burden of proof*.

This will involve the following tasks:

1. Look at some of the literature on the use of argumentation systems in dialogue, and in particular the notion of burden of proof; a starting point is [**GPW2007**].
2. Extend your implemented system to support the interchange of arguments following the burden of proof (discussed in more detail in section 4).

3. Write a report on the work you have carried out, placed in the context of prior work on AI approaches to argumentation.

The main credit for this assignment will go to a report that you are asked to write. The report should cover the practical work you have carried out in *both assignments*.

There are many ways that you could extend the your system from Assignment 1 to address task (2) above; there is no one 'right' answer. The way in which you approach the task and report on it are just as important as the actual software that you write.

You should write your own code and report. You are not permitted to

- copy code which someone else wrote for submission to this assignment;
- show your own programs or report to other students.

Outside these restrictions, you are encouraged to have discussions with your colleagues about concepts, techniques and tools.

For more information, please consult the School's plagiarism policy.

3 What to Submit

For this assignment, you are required to submit:

- program source code, ensuring
  - you make your code as readable as possible;
  - you provide appropriate docstrings for classes and methods;
  - where appropriate, you provide additional comments to help the reader understand the intention behind the code;
  - you provide some example scenarios treated by your program.
- your report, as a PDF document; *reports in other formats will not be accepted as valid submissions*.

Put all these in a single directory, compress it, and submit it using the following command in DICE:

```
% submit ailp 2 <zip-file-of-your-project-directory>
```

The deadline for Assignment 2 submission is **16:00 on Tuesday 22nd December 2015**. Monday 14th and Tuesday 15th December will be *Demo Days* where you can reserve a half hour slot to demo what you have done and talk through the structure of the report so I can provide timely formative feedback on your work. If there is demand for an earlier demo day I will schedule one in week beginning 7th December.

Proposer	Arg_id	Argument	Assumptions	*murder* accept?	BoP
P		*murder*		no	**P**
P	arg1	[*killing*, *malice*], ∼[§*187 excluded*] ⇒ *murder*	§187 *is valid*	yes	**D**
D	arg2	[*self-defense*], ∼[§*197 excluded*] ⇒ §*187 excluded*	§197 *is valid*	no	**D**
D	arg3	[*W1 'self-defense'*], ∼[*W1 not credible*] ⇒ *self-defense*	§197 *is valid*	no	**P**
P	arg4	[*W2 'time to run away'*], ∼[*W2 not credible*] ⇒ ¬*self-defense*	§197 *is valid*	no	**P**

Table 1: Shifting Burden of Proof

4 Burden of Proof

The first goal is to extend the system to allow a set of arguments to be put together cumulatively, as in a dialogue where a proponent aims to establish some conclusion, and an opponent aims to defeat it.

In a legal situation, the prosecution and defense each start with a body of evidence they bring to bear, and arguments of the relevance of that evidence. You are being asked to model this process, with the shifting burden of proof.

You should explicitly model where the burden of proof lies at each step. Ideally your system should also automatically select, from the set of available arguments, an appropriate argument to introduce; this is needed to get an A pass in this part of the course.

It is up to you how best to approach this modelling task, but your starting point should be the paper [**GPW2007**], which describes informally how the shifting burden of proof can be represented in the Carneades framework.

Table 1 provides a brief restatement of the discussion on pp. 888–890 of [**GPW2007**], showing how the burden of proof shifts as the prosecution and defense successively put forward arguments pro and con the conclusion of *murder*. The level of abstraction used in the table is adequate for your modelling task; in particular, you are not required to distinguish between the burden of persuasion and the burden of production. The mechanism for determining acceptability of a statement should be the one that was used in Assignment 1 (i.e., via the `CAES` method `acceptable()`.)

The table uses the following conventions:

- **P** and **D** stand for the prosecution and defense, respectively.
- '*murder* accept?' refers to whether *murder* is acceptable at this point in the discussion.
- 'BoP' indicates where the burden of proof lies.

As part of your system development, you are expected to provide examples of legal-style argumentation scenarios that allow you to evaluate the performance of your implementation.

5 The Report

As a rule of thumb, a report of around six pages should be sufficient for this assignment. In any case, your report should be *no longer than nine pages, including bibliography, figures and appendices*.

You do not have to use any particular word processing system, but a document skeleton in LaTeXis available on the course web page. This skeleton is *not* intended to be prescriptive, and you are free to use it or not.

Regardless of how you prepare your report, the submitted document *must* be in PDF format; other document formats will not be accepted.

Your report should address the following issues:

Context Explain the theoretical background to your system. This will in part be a 'literature review'. You are not expected to do extensive reading, but you should demonstrate that you have read at least **three** papers other than [**GPW2007**], and this will in any case give you a better understanding of the concepts involved in burden of proof. The bibliography of [**GPW2007**] is a good place to start.

Design Describe the functional requirements that your system is intended to meet (covering both Assignment 1 and Assignment 2), and give a high-level specification of the system that you have developed. Explain why you made your design decisions and describe possible alternatives.

Implementation Describe your implementation (covering both Assignment 1 and Assignment 2). You should not include large chunks of code in the body of your report; stick instead to presenting your approach at an algorithmic level, possibly including UML class diagrams if you wish.

Evaluation Consider how well your implementation satisfied the requirements presented in the *Design* part of your report. Describe tests that you ran, and whether your system performed as expected. If you decide to provide complete test runs, these will probably be best placed in an appendix.

Conclusion Provide a summary of your work, including its strengths and weaknesses and how it might possibly be improved.

You may find it useful to compare what you have developed with the implementation of Carneades available at `https://carneades.github.io/carneades/Carneades`, though this is not required.