AI Large Practical: Assignment 2 ctd

Alan Smaill

School of Informatics

Nov 6 2013

Organisation

School of informatics

- Remember that the hand-in date is Thursday 12th December.
- I will be available during drop-in sessions as before.
- You can use any additional programming tools and techniques that you think will help you –
  remember to acknowledge the source of ideas and code in your report.

Burden of proof and Carneades

A motivating example from the paper:

> *Let us illustrate the concept of argumentation schemes with the scheme for arguments from expert opinion, as formulated in [49, p. 210], with some minor notational changes:*
>
> *Major Premise* *Source E is an expert in the subject domain S containing proposition A.*
>
> *Minor Premise* *E asserts that proposition A in domain S is true*
>
> *Conclusion* *A may plausibly be taken as true.*
>
> *The six basic critical questions matching the appeal to expert opinion [49, p. 223] are the following.*
>
> *Gordon et al. p 5*

Motivating example ctd

School of informatics

1. *How credible is E as an expert source?*
2. *Is E an expert in the field that A is in?*
3. *Does E's testimony imply A?*
4. *Is E reliable?*
5. *Is A consistent with the testimony of other experts?*
6. *Is A supported by evidence?*

Gordon et al. p 5

In this case, we have a list of the 6 basic critical questions for expert testimony. In fact, (4) is used in the paper we used as starting point for the Haskell implementation.

Different sorts of premises

School of informatics

In a dialogue, use the Carneades classification of statements in an argument to distinguish where the burden of proof lies:

- **ordinary premises**: that must always be supported with further grounds
- **assumptions**: that can be assumed until they are questioned
- **exceptions**: that do not hold in absence of evidence to the contrary.

see Gordon et al., p 7

Burden of proof and Carneades

School of informatics

From the paper:

> *"In the case of exceptions, sometimes only the burden of production rests on the defendant; once this burden is satisfied, the plaintiff has the burden of persuading the court that the exception does not hold. This is the case, for instance, with all exceptions in criminal law."*
>
> *Gordon et al. p 8*

– the *"burden of production"* here **is** the burden of proof, for our purposes:

> *"We now see that, when discussing above how critical questions affect the burden of proof, we meant the burden of production in particular."*
>
> *Gordon et al. p 9*

Burden of Proof ctd

School of informatics

So the authors claim a strong link between types of premises (in Carneades), and the allocation of the burden of proof.

- If one side uses an argumentation scheme, then the burden is on that side to give evidence for:
  - the ordinary premises,
  - and, if challenged, the assumptions involved.

  after which the burden is on the other side to:
- defeat the argument by either:
  - finding a counter-argument to the conclusion (rebutting), or
  - pointing out exceptional circumstances

proof standards

Alongside the association of burden of proof with types of premises, the authors suggest starting with low proof standards assigned to statements of premisses:

> *The burden of production is distributed by dividing premises into different types: evidence for ordinary premises and (once challenged) assumptions must be produced by the proponent of the argument with these premises, while evidence for exceptions must be produced by the respondent. In addition some initially low proof standard needs to be assigned to the statement of each premise*
>
> *Gordon et al., p 9*

Dialogue

School of informatics

We can think of a dialogue here as a sequence of steps where arguments are put forward into a public arena; the state of the partial argument can be checked with the version of Carneades already implemented.
Here the notion of "speech acts" is in the background, where the state of play of the dialogue is updated by presentation of new claims. For background on this, see Stanford Encyclopedia of Philosophy:

`http://plato.stanford.edu/entries/speech-acts/`

Thus:

> *Speech acts can be modeled as functions which map a state of the dialogue to another state.*
>
> *Gordon et al., p 11*

The state of the dialogue includes at least the set of arguments in play, and the current assignment of burden of proof.

Critical questions and Carneades

School of informatics

Section 4 of Gordon et al. introduces some new proof standards and judgements; it is not required to implement these, but you may want to consider what role these may play in assigning burden of proof.

Sections 5 and 6 of Gordon et al. are important for the assignment. Section 5 discusses reasons for assigning premisses as (ordinary) premises, assumptions or exceptions.
The argument scheme for expert opinion is then worked out, based on the earlier characterisation of critical questions.

Expert witness and critical questions

The argument scheme:

Premise	E is an expert in the subject domain S containing the proposition A.
Premise	E asserts A.
Assumption	E is a credible expert.
Exception	E is not reliable.
Exception	A is not consistent with the testimony of other experts.
Assumption	A is based on evidence

Conclusion A.

Example in paper

School of informatics

Look at the example in section 6.

This shows a small example of combination of argument schemes following the ideas already presented.

1. First the prosecution gives an initial argument that murder has occurred;
   Carneades finds that on its own it is convincing.
2. Burden of proof passes to defence, who claim that self-defence is involved, and supply witness.
   This is enough at this stage to block the murder conclusion.
3. This is attacked in turn by the prosecution; however at this stage the evidence is not strong enough to establish murder.

Comments on example

Note that the defence put forward a combination of arguments, since it had a burden of proof to establish a premise of its own argument; only then does burden of proof pass to the prosecution.

So care is needed in setting out arguments.
The paper makes use of *argument schemes* – these are general patterns which then give particular arguments in different cases. A full implementation could make this explicit, but it's enough here to take care on particular instances.

What should implementation provide?

School of informatics

- A sequence of dialogue states, which should be
- recorded in some form (as trace? how much info??)

In a given state, it should be clear what the current set of arguments is, and what the state of the top-level assertion is.

The different sides of the argument should start with some arguments they can bring to the table, and deploy them in a relevant way, following the burden of proof.

The example in Gordon et al., section 6, would be a good development example.

Shell scripts

School of informatics

Shell scripts can be useful in helping to organise experimental runs, and keeping track of data. There is absolutely no requirement to use these, but it may help you (and markers!) to use your code easily.
Some pointers — the first should be enough for present needs.

- Bash Guide for Beginners
  `http://tldp.org/LDP/Bash-Beginners-Guide/html/`
- Advanced Bash-Scripting Guide
  `http://tldp.org/LDP/abs/html/`

Script example

In the simplest case, set up a simple sequence of commands
make your script executable by yourself and others:

```
=> chmod a+x myScript.sh
```

```
#! /bin/bash

# example script:
#  just a sequence of shell commands
# -- here just put extra line on end of each file,
#  and output in a different directory

TESTDIR=testData
RESULTDIR=results

cat ${TESTDIR}/a1 > ${RESULTDIR}/a1_result
echo "results obtained" >> ${RESULTDIR}/a1_result
cat ${TESTDIR}/a2 > ${RESULTDIR}/a2_result
echo "results obtained" >> ${RESULTDIR}/a2_result
cat ${TESTDIR}/a3 > ${RESULTDIR}/a3_result
echo "results obtained" >> ${RESULTDIR}/a3_result
```

Script example 2

Not too hard to arrange for all the test files in a directory to be processed as above:

```
#! /bin/bash

# example to run procedure on all files in a directory,
# output results to result directory
# -- here just put extra line on each file

# relative paths to test/result directories

TESTDIR=testData
RESULTDIR=results

# use result of bash command
for file in ‘ls ${TESTDIR}‘
do
#  next line commented out
#  cmd [option] $file >> results.out
    cat ${TESTDIR}/$file > ${RESULTDIR}/${file}_result
    # above overwrites any data
    echo "the results" >> ${RESULTDIR}/${file}_result
    #  this appends new data
done
```

LaTeX template and help

School of informatics

- There is a zip file on the course web page with LaTeX template for writing the report.
- You can use whatever you like to produce the report; but your submitted report must be in **PDF** format.
  Use `pdflatex` to produce PDF directly.
- Some resources for LaTeX:
  obvious place to look is the LaTeX project site:

  http://www.latex-project.org/

  — in particular the short introduction.

Using LaTeX ctd

Look through the template; this should be self-explanatory.

There is a useful emacs mode for editing LaTeX source, `auctex`. To enable this, put the following in your `.emacs`

```
(load-library "auctex")
(TeX-PDF-mode 1)                ;turn on PDF mode
```

This will give helpful menus for compiling and viewing pdf documents.

You are recommended to use `pdflatex` to produce output pdf from LaTeX.

However you write your report, you should submit **pdf** document; just about any system you might use allows this.

Summary

School of informatics

- Modelling burden of proof in Carneades
- bash scripts
- using LaTeX