AI Large Practical: Assignment 2 ctd

Alan Smaill

School of Informatics

Oct 26 2016

Today: Assignment 2

School of informatics

- Submission, assessment factors
- Any big problems?
- reminder on plagiarism
- recent news item

Reminder of submission

School of informatics

- Your are required to submit program source code (clearly and suitably commented), and
- your test files;

- To help the tester, please also provide:
  - a README explaining how to run the program to read in and evaluate the test arguments;
    - the simpler, the better
    - try to make this run out of the box – watch out for permissions
  - some version of results of sample queries.

Your system should run under DICE !

Structuring your development

School of informatics

As mentioned before, you should consider the overall design of your program, and in particular how it may be broken into suitable modules.
There will be credit for this aspect of your work.

Test files

School of informatics

- The basic requirement is that your test files fit the required format, and are properly processesd.
- You should aim for tests that make sense as argumentation examples, rather than just syntactically.

- Eg, case of footballer who was found not guilty of racial abuse in court, yet fined by football authorities for the same incident.

Assessment criteria

School of informatics

This part of the assignment is worth 50% of the overall marks for the course.
You submitted system will be tested, following your README instructions; the following aspects will be taken into consideration:

- whether the program runs
- content of test files
- whether the results are reasonable
- the structuring of the development
- appropriate comments and explanations

(Remember you will want to build on this system for the second part of the assignment.)

Assessment ctd

School of informatics

- To pass: a running program, able to read in at least some of the required inputs.
- For C: appropriately presented so as to be easy to run;
- For B: also needs well structured and documented code, well chosen test examples
- For an A, require results to be OK, and comments, error-reporting on input files, structuring and explanation as requested.

Efficiency is not a primary concern here; but extra credit is available for efficiency and style.

Remember, you will build on this implementation in the second part of the course, so re-usability is important.

Reminder: write your own code

The submissions will be run through a checker to get some idea of similarities between submissions.

As always, it is fine to discuss the AILP with fellow students, but your must do your own implementation (and examples).

News item on computers and legal jusgements

The following may be relevant for assignment 2, as background:

`http://preview.tinyurl.com/jlk2q6t`

The Guardian, 24th October 2016:
"Artificial intelligence 'judge' developed by UCL computer scientists"

> *"Software program can weigh up legal evidence and moral questions of right and wrong to predict the outcome of trials."*

News item ctd

School of informatics

This has been reported in several papers in the last few days; the work has been published in an academic journal.
Announcement from UCL:

`http://www.ucl.ac.uk/news/news-articles/1016/241016-AI-predicts-outcomes-human-rights-trials`

This system apparently works from natural language accounts of cases (an area this AILP does not look at). But this aspect is complementary to the argumentation which underlies the judgements that are made.

Future meetings

School of informatics

- There will be no Wednesday class next week;
- In two weeks, introduction to Assignment 3
- Drop-in sessions will continue as usual
- My office hours: Thursdays, 10:30–11:30, 2.03 IF.
- or see me by appointment.