Logic Project 2: Normal Form Transformation

Adrian Crăciun
acraciun@info.uvt.ro

December 11, 2013

The Aim of the Project

Given an arbitrary propositional formula, construct its normal forms: Conjunctive Normal Form or Disjunctive Normal Form, or both, according to an option to be passed to the program.

Requirements

- Using the implementation, the user should be able to introduce a propositional formula :
  - files are preferred for the input, to allow the user prior preparation of the input,
  - other input methods (visual forms, etc.) are not restricted, but "live" console input is not encouraged (due to limitations);
- the program should transform the formula to the intermediary negation normal form (NNF),
- the NNF transformation should be done with simplifications (contradiction checks, subsumptions);
- the program should then transform the NNF into the required form, and print it (on the screen and/or in a file).

Programming Language

C or Python strongly suggested. However, other languages will be allowed[1].

Delivery of the Project

The project will be delivered in the following form:

1. A paper, describing the implementation, that should contain the following:
   - the statement of the problem,

[1] This should be discussed with me before the start of the project and properly motivated

- user manual, i.e. how to use the program,
- a description of the solution (structure of the program, functions, data structures),
- an argument (proof) that the program is correct,
- a test suite[2] for the program, i.e. examples of inputs, including non-trivial ones,
- test runs (e.g. screen captures, or algorithm printouts), with timing,
- conclusions (whatever the author(s) experience was);

2. Annexes to this paper, containing:

   - the license of the program (see `http://en.wikipedia.org/wiki/Software_licenses`); also provide a reason for the choice of license,
   - the code of the program,
   - the installation manual for the program.

   NOTE. The paper and annexes will be delivered in .pdf format, by email or made available online (preferred).

3. The program (source + executable) should be sent by email or made available online (preferred).

4. The program will be demonstrated at the University (arrangements to be discussed).

Working in Teams

Working in teams (max. 3 people) is allowed, and even encouraged. However, this should be a learning experience. Therefore, people with experience in programming are encouraged to team up with the people which are less familiar with this activity. At delivery, any of the team members should be able to answer any question connected to the project.

[2] The test suite should include all homeworks (where it applies)