CS5860

Fall 2011 Examples of mini-projects

Instead of doing one large project you could do four or so smaller projects as topics of interest arise. I will grade them as they are submitted. Here are some possibilities.

1. Write the Euclid gcd algorithm in classic ML and give an argument with the program that it is correct using the method of treating an assertion (proposition) as a type. Cite the articles of Gries and McCarthy to explain your approach. Connect your explanation to your understanding of computational logic.

2. Give ML programs that are evidence terms for ten interesting propositions in the Propositional Calculus, say from the examples in Smullyan posted on the Web.

Each of these projects could be easily extended as we progress, say to other programs that can be verified, other logics whose formulas have evidence terms.

3. Read one of the suggested readings in detail and relate it to the lectures, say in the form of a written report. For example, my article on Core Computational Type Theory explains dependent types. You could use these types to explain a programming task. You could study Prof. Abraham's article on Peterson's algorithm and express his argument in type theory.

CS 5860
Fall 2011 Examples of mini-projects continued 2

4. You could elaborate points from the lectures and make your observations available to the class after I have edited them. For example, there are many interesting examples of statements of the form $(A \Rightarrow B)$ which are true even though both A and B are unknown. You could analyze one in detail such as (ERH $\Rightarrow$ SAT $\in$ PTime) and illustrate what it means to know such an implication.

5. You could build a collection of proofs starting with propositional logic using Tableau and Refinement style and relating the formal proof to a natural language proof. This could become a valuable course resource.