Problem Set 6 Due Thursday November 11, 2010 CS4860

Reading: Please read *Proofs as Programs*, the pdf version of the article posted with Lecture 19, Tue Nov 2.

1 Problem

Show that *Complete Induction* for $\mathbb{N}$ and *Standard Induction* for $\mathbb{N}$ are equivalent principles in the sense that each one implies the other in Refinement Logic. Write the arguments first informally and then write them in Refinement Logic.

1. Complete $\forall x : \mathbb{N}.\ ((\forall y : \mathbb{N}.\ y < x \ \Rightarrow \ P(y)) \ \Rightarrow P(x)) \ \Rightarrow \ \forall n : \mathbb{N}.P(n)$.
2. Standard $P(0) \ \wedge \ \forall y : \mathbb{N}.(P(y) \ \Rightarrow P(y+1)) \ \Rightarrow \ \forall n : \mathbb{N}.P(n)$.

2 Problem

Prove in Refinement Logic that the *Least Number Principle* is true, using one of the induction rules from the above problem. Here is the principle.

$$\exists x : \mathbb{N}.P(x) \ \Rightarrow \ \exists y : \mathbb{N}.\ (P(y) \ \wedge \ \forall z : \mathbb{N}.(z < y \ \Rightarrow \ \neg P(z))).$$

3 Problem

Write an algorithm to solve the Stamps Problem from Lecture 20. Use the proof from lecture to guide your efforts and explain how the proof helps you write a correct algorithm. *Try to make the algorithm follow the proof as closely as you can.* You could even use the "induction forms" from Refinement Logic, nat-ind and list-ind, to write your program or you can make up a programming notation for the problem. Of course, you can also use a programming language, but make sure the notation is not cluttered with unnecessary syntax and that you *follow the proof.* Here is a statement of the problem.

Suppose we have stamps of two different denominations, 3 cents and 5 cents. We want to show that it is possible to make up exactly any postage of 8 cents or more using stamps of these two denominations. We want to show that if it is possible to make up exact postage of n cents using 3-cent and 5-cent stamps, then it is also possible to make up exact postage of n + 1 cents using 3-cent and 5-cent stamps for n larger than or equal to 8.

4 Problem

Represent rational numbers, $\mathbb{Q}$, as lists with exactly two elements, [a,b] to represent a/b.

(a) Define the equality relation on these numbers, call it $=_{\mathbb{Q}}$.
(b) Define the addition of two rational numbers.
(c) Prove informally that addition of rational numbers, q_1 and q_2 is commutative, that is $q_1 \ + \ q_2 \ =_{\mathbb{Q}} \ q_2 \ + \ q_1$.

(d) Write in Refinement Logic the *statement* that addition is commutative; assume that the equality relation on these numbers, $=_{\mathbb{Q}}$, and the addition operator + are already defined in the logic.

5 Problem

Explain how to implement the Least Number Principle from problem 2 as a "while rule" in the sense of Hoare's while rule that we used to guide our thinking about integer square roots. Assume that the relation P satisfies the condition that we can prove without the law of excluded middle, $(P \vee \sim P)$, this assertion:
$\forall x : \mathbb{N}.\ (P(x) \vee \sim P(x))$.