Problem Set 7 Due Thursday December 2, 2010 CS4860

1 Problem

(a) Prove the following formula using a Gentzen system proof that first moves all information into the hypothesis list leaving $false$ as the only goal to prove.

$$(\sim B \Rightarrow \sim A) \;\Rightarrow\; (\sim\sim A \Rightarrow \sim\sim B)$$

(b) Prove this formula by Refinement Logic and exhibit the reduced proof term. Note that in Refinement Logic we define $\sim A$ to mean $A \Rightarrow false$.
(c) Discuss the advantages and disadvantages of each proof style.

2 Problem

Write Refinement proofs for the following formulas from Smullyan page 56 and produce the proof expressions in reduced form.

1. $\forall x(Px \;\Rightarrow\; \exists x P(x))$
2. $\exists x(Px \;\vee\; Qx) \;\Rightarrow \exists x Px \;\vee\; \exists x Qx$
3. $\exists x(Px \;\vee\; Qx) \;\Leftarrow \exists x Px \;\vee\; \exists x\; Qx$
4. $\exists x(Px \;\wedge\; Qx) \;\Rightarrow\; \exists x Px \;\wedge\; \exists x Qx)$
5. $\exists y((\exists x Px) \;\Rightarrow\; Py)$

3 Problem

In Lecture 24 we stated the computation rule for the *Standard Induction* proof expression. It is this:

$ind(0; b; u, i.p(u, i)) \;=\; b$
$ind(n + 1; b; u, i.p(u, i)) \;=\; p(n, ind(n; b; u, i.p(u, i)).$

This is an example of a *primitive recursive* function definition.

Give a proof expression for *Complete Induction* (from the last problem set) and its reduction rule, similar to the one above for Standard Induction.

4 Problem

(a) Give a Refinement style proof of the following simple fact about the factorial function defined as in lecture. Make the proof as close to an actual Refinement Proof as possible.

$$\forall n{:}\mathbb{N}.\exists y{:}\mathbb{N}.fact(n) = y.$$

You can use a rule that allows you to expand the definition of factorial in a proof step, call it Definition Rule.

(b) Show the extract for this proof as readably as possible.

5 Problem

Here is a recursive predicate about lists of natural numbers which defines what it means for a natural number to be on a list. Recall that a list is either *nil* or is a head h "consed" on to a tail t, that is $h.t$ where h is a number and t is a list.

$ListMembership == \forall x : \mathbb{N}.\forall h : \mathbb{N}.\forall t : List.(OnList(x, nil) = false \wedge OnList(x, h.t) = (x = h \vee OnList(x, t))$

Prove the following formula in Refinement Logic using list induction and a rule for using the OnList(x,l) predicate.

$$\forall l{:}List.\exists y : \mathbb{N}.\forall x{:}\mathbb{N}.(OnList(x, l) \Rightarrow x \ \leq \ y).$$