Constructive Logic
Fall 2000

Carnegie Mellon University
Frank Pfenning

Assignment 6

Out: Friday Oct 20

Due: Thursday Oct 26

The use of the tutch proof checker is recommended for checking the proof terms, as soon as it is available for arithmetic and lists. Please check the course home page `http://www.cs.cmu.edu/~fp/courses/logic/` or the course bulletin board for an announcement.

1. Program Extraction (40 Points)

1. Give an informal proof of

$$\forall x{\in}\mathbf{nat}.\ \forall y{\in}\mathbf{nat}.\ y < \mathbf{s}(x) \supset \exists z{\in}\mathbf{nat}.\ \mathit{plus}\ y\ z =_N x$$

2. Give a proof term corresponding to your informal proof.

3. Annotate your proof term with brackets to match the specification

$$\forall x{\in}\mathbf{nat}.\ \forall y{\in}\mathbf{nat}.\ [y < \mathbf{s}(x)] \supset \exists z{\in}\mathbf{nat}.\ [\mathit{plus}\ y\ z =_N x]$$

   If your proof term does not allow such an annotation, please revise your proof so it does!

4. Give the type and the term which result from erasing the bracketed expression, as shown in lecture and the lecture notes. What does your function compute?

2. Structural Induction (30 Points)

For each of the following propositions, first give an informal proof using structural induction, then show the proof term. You may use earlier lemmas to establish later results.

1. $\mathit{refll} : \forall l{\in}\tau\ \mathbf{list}.\ l =_L l$.

2. $\mathit{appapp} : \forall k{\in}\tau\ \mathbf{list}.\ \forall l{\in}\tau\ \mathbf{list}.\ \forall m{\in}\tau\ \mathbf{list}.\ \mathit{app}\ (\mathit{app}\ l\ k)\ m =_L \mathit{app}\ l\ (\mathit{app}\ k\ m)$.

3. Generalizing the Induction Hypothesis (30 Points)

Prove informally, that

$$\forall l{\in}\tau\,\mathbf{list}.\ \mathit{reverse}(\mathit{reverse}\ l) =_L l$$

Make sure to appropriately generalize the induction hypothesis and exhibit the structure of your inductive argument. You may use the theorems from Problem 2, and the lemmas

$$\begin{array}{lcl} \mathit{appnil} & : & \forall{\in}\tau\,\mathbf{list}.\ \mathit{app}\ l\ \mathbf{nil} =_L l \\ \mathit{apprev} & : & \forall l{\in}\tau\,\mathbf{list}.\ \forall k{\in}\tau\,\mathbf{list}.\ \forall m{\in}\tau\,\mathbf{list}.\ \mathit{app}\ (\mathit{rev}\ l\ k)\ m =_L \mathit{rev}\ l\ (\mathit{app}\ k\ m) \end{array}$$

proven in lecture.