Constructive Logic
Fall 2000

Carnegie Mellon University
Frank Pfenning

Assignment 4 – Extra Credit

Out: Friday Sep 29

Due: Thursday Oct 5

5. More Primitive Recursion over nat (10 Points)

Specify and implement the following functions:

- $\mathit{half} : \mathbf{nat} \to \mathbf{nat}$ with the same semantics as in the lecture. But this time use an auxiliary function $\mathit{half}' : \mathbf{nat} \to \alpha \to \mathbf{nat}$ with an extra argument of a suitable type α.
- $\mathit{div} : \mathbf{nat} \to \mathbf{nat} \to \mathbf{nat}$. $\mathit{div}\ n\ m$ computes the quotient $n/(m+1)$.
- $\mathit{log} : \mathbf{nat} \to \mathbf{nat} \to \mathbf{nat}$. $\mathit{log}\ n\ b$ computes the logarithm of $n+1$ to the base $b+2$.

6. More Primitive Recursion over list (10 Points)

Specify and implement the following functions:

- $\mathit{flat} : \tau\,\mathbf{list}\,\mathbf{list} \to \tau\,\mathbf{list}$. The function *flat* gets a list of lists ll and "flattens" it into a simple list, by concatenating all lists contained in ll. This is to be implemented without the use of *append* and by looking at every element only once.
- $\mathit{merge} : \tau\,\mathbf{list} \to \tau\,\mathbf{list} \to \tau\,\mathbf{list}$. $\mathit{merge}\ xs\ ys$ takes the first element of xs, then the first of ys, then the second of xs etc. and puts them into a joint list. If one list is empty already, the remainder of the other list is simply appended.

7. Binary Representation of Natural Numbers (10 Points)

Introduce a new inductive datatype **bnat** for a binary representation of the natural numbers and specify and implement the successor function $\mathit{bsucc} : \mathbf{bnat} \to \mathbf{bnat}$ and addition $\mathit{badd} : \mathbf{bnat} \to \mathbf{bnat} \to \mathbf{bnat}$.

Good luck!