CS520 Problems for Assignment 3

Posted: 24 Sept 2004
Due: 1 October 2004

Problem 1: Construct a term "eq" of $\boldsymbol{\lambda}$**NB** such that

$$
\begin{array}{lll}
\mathsf{eq}\ m\ n \Rightarrow^* \mathsf{if} & & (\mathsf{iszero}\ m) \\
& \mathsf{then} & \mathsf{if}\ (\mathsf{iszero}\ n) \\
& & \mathsf{then\ true} \\
& & \mathsf{else\ false} \\
& \mathsf{else} & \mathsf{if}\ (\mathsf{iszero}\ n) \\
& & \mathsf{then\ false} \\
& & \mathsf{else}\ (\mathsf{eq}\ (\mathsf{pred}\ m)\ (\mathsf{pred}\ n))
\end{array}
$$

Follow the method outlined in Handout 07. Test your code on input values $m = n = 2$.

Problem 2: There are propositional formulas that are tautologies in standard **propositional logic** (PL) but not in **implicational logic** (IL). One such formula is

$$T \triangleq ((\mathsf{a} \to \mathsf{b}) \to \mathsf{a}) \to \mathsf{a}$$

T is a tautology in PL just by giving all possible assignments of "true" and "false" to a and b. Show that T is not a tautology of IL, i.e., there is no formal proof of T in IL.

We propose an outline for solving this problem and the different results that you will need for it. Your task is to supply the details:

1. Assume the opposite of the desired conclusion, i.e., that there is a proof of T in IL, with the goal of getting a contradiction at the end.

2. Use the *formulas-as-types correspondence* [Handout, page 12].

3. Although we have not discussed it in lecture yet, use the Normalization Theorem [Pierce, page 152] which says that every simply-typable term is normalizable (according to call-by-value evaluation). In fact, there is a stronger version of the theorem, not in [Pierce], which says that every simply-typable term is *strongly normalizable* according to full beta-reduction.

4. Use the Preservation Theorem [Pierce, Theorem 9.3.9; Handout 06, page 7].

5. Show there is no closed term t in normal form in $\boldsymbol{\lambda}_\to$ which has type T, i.e., $\vdash t : T$ is not derivable in $\boldsymbol{\lambda}_\to$.