CIS 500 Software Foundations

Homework Assignment 6,

Simple Extensions of Simply-Typed Lambda Calculus

Due: Monday, November 1, 2004, by noon

Submission instructions: Same as last time.

1 **Exercise** Prove type safety for variants. You need to extend the following proofs with cases only for the rules in Figure 11-11. Also note that in this figure the value syntax for variants is missing. There is another form of values in the language and that form is $< l = v >$ *as* T.

1. Extend the inversion lemma of the typing relation (9.3.1).
2. Extend the canonical forms lemma (9.3.4).
3. Extend the progress lemma (9.3.5).
4. Extend the substitution lemma (9.3.8). As this lemma is stated in the textbook, it must be proved by induction on the depth of the typing derivation. Also to prove this lemma, you need to extend the definition for substitution within variants. You may assume that the permutation and weakening lemmas hold for the lambda calculus extended with variants.
5. Extend the preservation theorem (9.3.9).

2 **Exercise** In O'Caml, we can write the following program:

```
let rec take (l : bool list) : bool list =
  match l with
    [] -> []
  | hd::tl -> hd :: skip tl
and skip (l : bool list) : bool list =
  match l with
    [] -> []
  | hd::tl -> take tl
in
  take [true,false,true]
```

How would you write this program in the language of Chapter 11?

3 **Exercise** TAPL 11.12.2

4 **Debriefing**

1. How many hours (per person) did you spend on this assignment?
2. Would you rate it as easy, moderate, or difficult?
3. Did everyone in your study group participate?
4. How deeply do you feel you understand the material it covers (0%–100%)?

If you have any other comments, we would like to hear them; please send them to `cis500@cis.upenn.edu`.